NATIONAL BUREAU OF STANDARDS REPORT

9881

COMPUTER PROGRAM TO CALCULATE NATURAL CONVECTION FLOWS IN A ROOM

July 18, 1968

NBS

U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] was established by an act of Congress March 3, 1901. Today, in addition to serving as the Nation's central measurement laboratory, the Bureau is a principal focal point in the Federal Government for assuring maximum application of the physical and engineering sciences to the advancement of technology in industry and commerce. To this end the Bureau conducts research and provides central national services in three broad program areas and provides central national services in a fourth. These are: (1) basic measurements and standards, (2) materials measurements and standards, (3) technological measurements and standards, and (4) transfer of technology.

The Bureau comprises the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, and the Center for Radiation Research.

THE INSTITUTE FOR BASIC STANDARDS provides the central basis within the United States of a complete and consistent system of physical measurement, coordinates that system with the measurement systems of other nations, and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of an Office of Standard Reference Data and a group of divisions organized by the following areas of science and engineering:

Applied Mathematics—Electricity—Metrology—Mechanics—Heat—Atomic Physics—Cryogenics[2]—Radio Physics[2]—Radio Engineering[2]—Astrophysics[2]—Time and Frequency.[2]

THE INSTITUTE FOR MATERIALS RESEARCH conducts materials research leading to methods, standards of measurement, and data needed by industry, commerce, educational institutions, and government. The Institute also provides advisory and research services to other government agencies. The Institute consists of an Office of Standard Reference Materials and a group of divisions organized by the following areas of materials research:

Analytical Chemistry—Polymers—Metallurgy — Inorganic Materials — Physical Chemistry.

THE INSTITUTE FOR APPLIED TECHNOLOGY provides for the creation of appropriate opportunities for the use and application of technology within the Federal Government and within the civilian sector of American industry. The primary functions of the Institute may be broadly classified as programs relating to technological measurements and standards and techniques for the transfer of technology. The Institute consists of a Clearinghouse for Scientific and Technical Information,[3] a Center for Computer Sciences and Technology, and a group of technical divisions and offices organized by the following fields of technology:

Building Research—Electronic Instrumentation — Technical Analysis — Product Evaluation—Invention and Innovation— Weights and Measures — Engineering Standards—Vehicle Systems Research.

THE CENTER FOR RADIATION RESEARCH engages in research, measurement, and application of radiation to the solution of Bureau mission problems and the problems of other agencies and institutions. The Center for Radiation Research consists of the following divisions:

Reactor Radiation—Linac Radiation—Applied Radiation—Nuclear Radiation.

---

[1] Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D. C. 20234.
[2] Located at Boulder, Colorado 80302.
[3] Located at 5285 Port Royal Road, Springfield, Virginia 22151.

NATIONAL BUREAU OF STANDARDS REPORT

NBS PROJECT 421 2650

July 18, 1968

NBS REPORT 9881

COMPUTER PROGRAM TO CALCULATE NATURAL CONVECTION FLOWS IN A ROOM

by

Kenneth E. Torrance
Factory Mutual Research Associate
Fire Research Section
Institute for Applied Technology
National Bureau of Standards
Washington, D. C. 20234

IMPORTANT NOTICE

NBS

U.S. DEPARTMENT OF COMMERCE

NATIONAL BUREAU OF STANDARDS

COMPUTER PROGRAM TO CALCULATE

NATURAL CONVECTION FLOWS

IN A ROOM

by

Kenneth E. Torrance
Factory Mutual Research Associate
Fire Research Section
Institute for Applied Technology
National Bureau of Standards
Washington, D. C. 20234

July 18, 1968

COMPUTER PROGRAM TO CALCULATE NATURAL CONVECTION FLOWS IN A ROOM

by

Kenneth E. Torrance

ABSTRACT

A computer program is described for calculating time-dependent, laminar natural convection flows in a room. The complete Fortran language program as used on the NBS Univac 1108 computer is given. A discussion of the program and instructions for its use are facilitated by the aid of a flowchart and an example.

COMPUTER PROGRAM TO CALCULATE NATURAL CONVECTION FLOWS IN A ROOM

1. INTRODUCTION

This report provides a description of the computer program EXCON1 and instructions for its use. The program was written in ASA Fortran language (reference 1) and coded for use on the NBS Univac 1108 computer. The program was developed to calculate time-dependent, laminar natural convection flows in a room. The room is in the form of a vertical circular cylinder, with the heat source which induces the flow located in the center of the floor (see figure 1). The computer program is based upon mathematical principles which are discussed in references 2 and 3. A comparison of several programs for calculating flows is presented in reference 2, which highlights the particular advantages, as well as the limitations, of EXCON1 (known therein as numerical method V). An application of EXCON1 is presented in reference 3, in which calculated results are compared with the experimental flows of reference 4.

Figure 1 illustrates a cylindrical room of height a and radius b. In the center of the floor, a disk of radius c is held at a temperature ΔT above the remaining walls of the enclosure. The flow is assumed to be axisymmetric two-dimensional; i.e., without variations in the azimuthal direction. A grid is constructed as shown of M+1 horizontal lines and N+1 vertical lines spaced at intervals of ΔX and ΔR. The program EXCON1 is employed to calculate the temperature, vorticity, stream function and vertical and radial velocities at the intersections of these grid lines (called grid or mesh points). By rotation about the centerline, the resulting fields of temperature, velocity, etc., are translated to any azimuthal plane.

2. DISCUSSION OF EXCON 1

An annotated listing of the complete program EXCON1 is presented in figure 3. A flowchart of the main program is presented in figure 2. The next two sections describe the main program and the subroutines, respectively.

2.1 Main Program

The logical sequence of the main program (pp 1-7 of figure 3) can be grasped most readily by first considering the flowchart in figure 2. A list of input parameters is read in on data cards (an example of which is shown on p 16 of figure 3) as follows:

RB	=	aspect ratio of enclosure = b/a.
RS	=	relative size of heat source = c/a.
GR	=	Grashof number = $g\beta\Delta Ta^3/\nu^2$, where g = gravity, β = volume expansion coefficient of fluid, and ν = kinematic viscosity of fluid.
PR	=	Prandtl number of fluid in enclosure = ν/κ, where κ = thermal diffusivity.
M	=	number of vertical mesh points minus one.
N	=	number of radial mesh points minus one.
TAUZER	=	Time level at start of calculation.
DTAU	=	Time step for first time advancement. If JDATA = 0, DTAU is automatically calculated within the program.
TAUMAX	=	Computation to proceed in time up to this value.
TAUNOM	=	Nominal time level at start of program. Output is indexed with respect to this, so as to have output at convenient time values.
RELAX	=	relaxation parameter for iteration of stream function field.
JDATA	=	If = 0, calculation starts from quiescent initial conditions; if $\neq 0$, it starts from input data fields.
ITMAX	=	maximum number of stream function iteration cycles permitted at each time step. Each cycle consists of two iterative sweeps over the field; one vertically and one radially.
FKTR	=	heat transfer rates calculated and printed at times TAU = TAUNOM + n · FKTR, where n = 0, 1, 2, ···.
FPRINT	=	fields of temperature, velocity, vorticity and stream function printed at times TAU = TAUNOM + n · FPRINT, where n = 0, 1, 2, ···.
FTAPE	=	fields put on magnetic tape at times TAU = TAUNOM + n · FTAPE, where n = 0, 1, 2, ···.
FPUNCH	=	fields punched onto cards at times TAU = TAUNOM + n · FPUNCH, where n = 0, 1, 2, ···.

FPLOT = not used in present program.

INTAPE = logical I/O unit on which input tape is mounted (if used).

OUTAPE = logical I/O unit on which output tape is mounted (if used).

INBLK = data block to be read from input tape.

OUTBLK = data block in which writing starts on output tape.

After reading the input parameters the initial data fields are either generated within the program or read in, according to the value of JDATA. The program then enters the computation cycle as shown in figure 2.

2.2 Subroutines

SUBROUTINE VEL (p 8 of figure 3): This subroutine is called from the main program and computes the velocities U and V at node points with central difference approximations. At the solid boundaries, U = V = 0. The output fields, however, have nonzero values for U along the vertical wall and for V along the floor and ceiling. These values pertain to a location $\Delta R/2$ or $\Delta X/2$ away from the boundary and are computed with one sided differences.

SUBROUTINE MEAN (p 9 of figure 3): This subroutine is called from the main program and computes the following "mean" velocities:

$$UM(I,J) = (U(I,J) + U(I+1,J))/(2\Delta X)$$
$$VM(I,J) = (V(I,J) + V(I, J+1))/(2\Delta R).$$

The grid spacings ΔX and ΔR appear in the denominators and eliminate several multiplications in the main program.

SUBROUTINE RESLTS (p 10 of figure 3): This subroutine is called from the main program and controls printing of the fields of temperature, velocity, vorticity and stream function.

SUBROUTINE BLOCK (p 11 of figure 3): This is called from subroutine RESLTS to break up and print the output fields in blocks ten or eleven columns wide. The top and bottom of each block corresponds to the ceiling and floor, respectively. The printed output is illustrated in the example, section 3.

SUBROUTINE PUN (p 12 of figure 3): This subroutine is called from the main program and punches the fields of temperature, velocity, etc., onto cards. The first card serves to identify the data field. When used as input for the main program, this first card replaces the first of the three cards (shown on p 16 of figure 3) which contain the input parameter list. This first card is then followed by two cards containing suitable values for the remaining input parameters, and the data field cards.

SUBROUTINE TAP (p 13 of figure 3): When called from the main program, this subroutine puts the fields of temperature, velocity, etc., onto magnetic tape, followed by an end of file mark. (An end of file mark is always placed at the beginning of a new tape). The first record identifies the data block and is of the same format as the first input data card of the main program. This record is used to locate data blocks on tape. The information in this record, in the same format, is also punched onto two cards.

SUBROUTINE CHECK (p 14 of figure 3): When called from the main program, this subroutine scans the first record after every end of file mark on the input tape, until a data block corresponding to the first input data card of the main program is located. Four input data cards are required for the main program, the first and fourth cards being identical.

SUBROUTINE NUSSEL (pp 15-16 of figure 3): When called from the main program, this subroutine calculates two rates of heat transfer: (a) from heat source to fluid in enclosure; and (b) from fluid in enclosure to cold walls. Results are printed as a dimensionless ratio, $\dot{Q}_{in\ or\ out}/\lambda a \Delta T$, where λ is the thermal conductivity of the fluid in the enclosure. Nusselt numbers are also computed by dividing the dimensionless rates of heat transfer by the dimensionless heating area (A_{in}/a^2) or cooling area (A_{out}/a^2).

3. EXAMPLE

Results obtained from the execution of the program shown in figure 3 are presented in figure 4. Page 1 of the figure is the heading for the output results, which lists the pertinent input parameters for the problem. These are some of the parameters read in from the data cards on p 16 of figure 3. Following this heading, heat transfer results and calculated fields are periodically printed out. To save space, only the printed output for TAU = 0.25 is shown (pp 2-6 of figure 4). The computed fields are listed in the following order: temperature, vertical velocity U, radial velocity V, vorticity, and stream function. A 21x21 grid was employed, and for each field, 441 numbers are printed. For the first block of numbers, 21x10, the left-most column pertains to the centerline, and columns to the right correspond to increasing values of radius, up to R = 0.45. The top and bottom numbers in each column correspond respectively to the ceiling and the floor. The second block of numbers, 21x11, covers the radius range from R = 0.5 (left column) to R = 1.0 (right column). Again, ceiling and floor correspond to the top and bottom of each column.

As noted in the discussion of SUBROUTINE VEL, the right-most column of vertical velocities U is nonzero. These values pertain to R = 0.975 (not R = 1.0). Similarly, the radial velocities V along the floor and ceiling are nonzero. These values respectively correspond to X = 0.025 and X = 0.975.

Following the stream function field, some additional output is shown (p 6 of figure 4). The first entry is the number of cycles required for convergence of the stream function iteration (in this case, one). The next three entries constitute printed output from the subroutines PUN, TAP, and NUSSEL. Similar output results results whenever these subroutines are called. The words CARDS PUNCHED indicate that the temperature, velocity, etc., fields have been punched onto cards. The next line indicates that these fields have been written on tape in BLOCK 9. The first identifying record of the data block on tape is printed between the asterisks. The remaining two lines on the page are printed by subroutine NUSSEL, and provide the time TAU, the next time step DTAU, and the heat transfer rates.

4. APPLICATION EXPERIENCE

Most of the calculations to date with EXCON1 have employed fixed values of the following parameters: RB = 1, RS = 0.1, and PR = 0.7. This includes reference 2, which presents information on truncation errors and the effects of grid size. A range of GR from 1×10^5 to 4×10^7 is covered. Reference 3 presents a study of the effect of varying GR from 4×10^4 to 4×10^{10} with M = 50 and N = 20. The computed flows are all laminar. The validity of the calculations for GR $\lesssim 1 \times 10^9$ is confirmed by comparison with the experiments of reference 4. At higher values of GR the experiments indicate turbulence, although the calculated flows are laminar. This difference is due to the assumed axisymmetry and to numerical truncation errors. The latter increase with both GR and coarser grids. From figure 5, it is apparent that the M x N = 50 x 20 grid employed in reference 3 required computing times of several hours at high values of GR.

Some calculations have also been attempted at GR = 4×10^8 with an aspect ratio of RB = 10. Preliminary runs with 10 x 100 and 20 x 200 grids indicate that the required computing times to reach steady state are 81 minutes and 17 x 81 minutes, respectively. Both grids are too coarse to adequately resolve details of the flow. Furthermore, the finer grid increases the amount of computer time required roughly as the ratio of the squares of the number of grid points.

From experience, the following conclusion can be noted about the use of EXCON1: Meaningful results can be obtained only when the grid is fine enough to resolve details of the flows. If the grid is too coarse, details are not resolved and truncation errors become important. The use of fine grids to improve the calculation, however, can quickly lead to prohibitively large demands on computer time and storage. The application of EXCON1 to very high GR natural convection flows does not yet seem feasible, because of the foregoing requirements. Nevertheless, the program can be effectively and successfully applied to calculate flows when the mesh size is chosen judiciously and does not exceed computer storage limitations.

5. REFERENCES

1. D. D. McCracken, A guide to Fortran IV programming (John Wiley and Sons, Inc., New York, N.Y., 1965).

2. K. E. Torrance, Comparison of finite-difference procedures for integrating the equations of fluid flow, submitted to the J. Research (NBS) (1968).

3. K. E. Torrance and J. A. Rockett, Numerical study of natural convection in an enclosure with localized heating from below-creeping flow to the onset of laminar instability, submitted to the J. Fluid Mechanics (1968).

4. K. E. Torrance, L. Orloff and J. A. Rockett, Experiments on natural convection in enclosures with localized heating from below, submitted to the J. Fluid Mechanics (1968).

APPENDIX A. APPLICATION TO THE FACTORY MUTUAL IBM 360 COMPUTER IN NORWOOD, MASSACHUSETTS

The program listed in figure 3 must be modified for use on the FM IBM 360. The IBM compiler uses a "basic Fortran" language and does not recognize logical IF statements, which are used extensively throughout EXCON1. These statements would all have to be reprogrammed. An improved IBM compiler is being developed, which, when available, will recognize the logical IF. Only minor modifications would then be required.

Any decks of cards punched for or on the Univac 1108 have to be converted for use on the IBM 360. A few symbols are punched differently for the two computers, such as parentheses, plus sign, etc. Dr. John A. Rockett has developed a program to effect such a conversion.

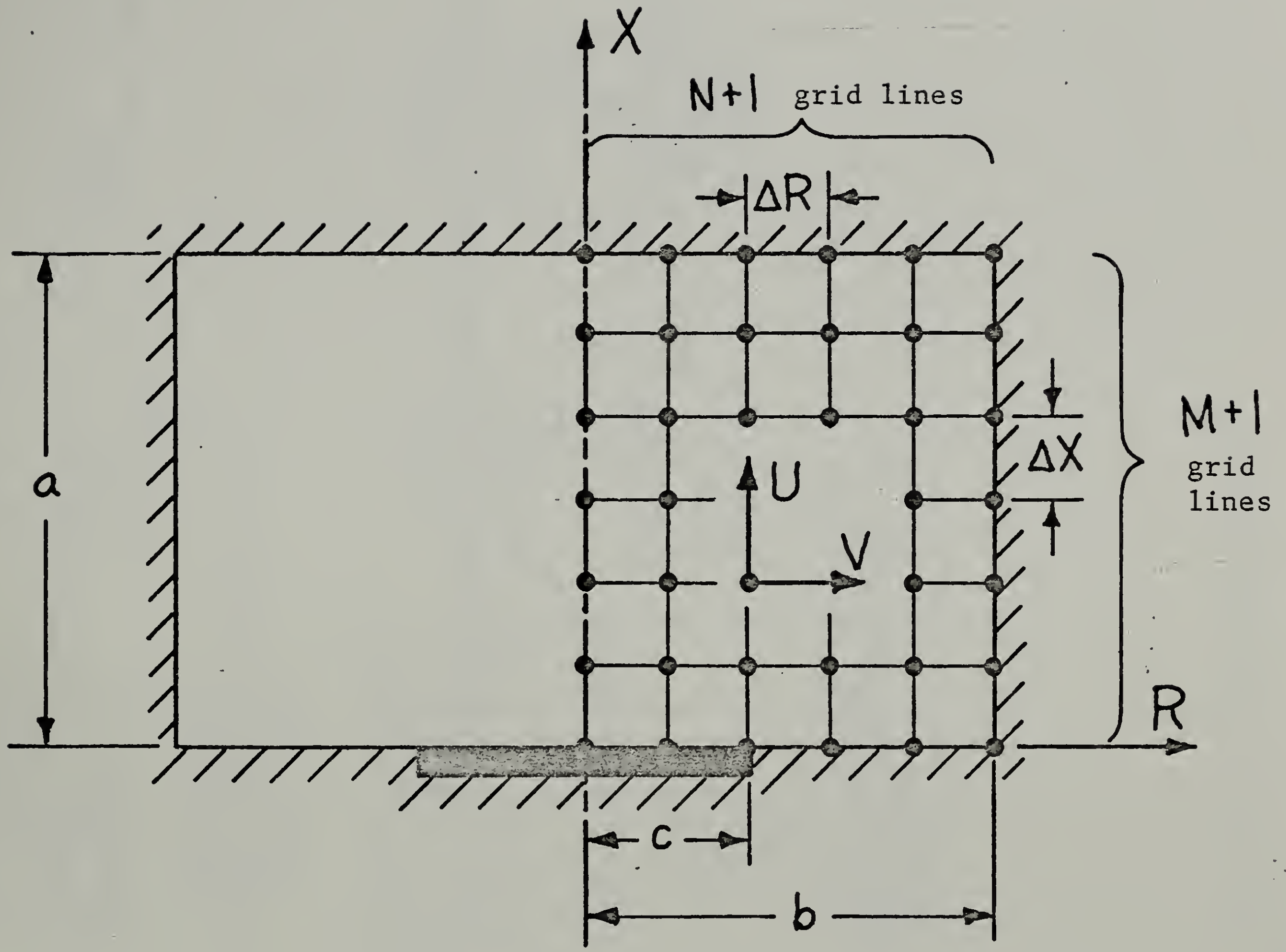

FIGURE 1. Cylindrical room and grid system.

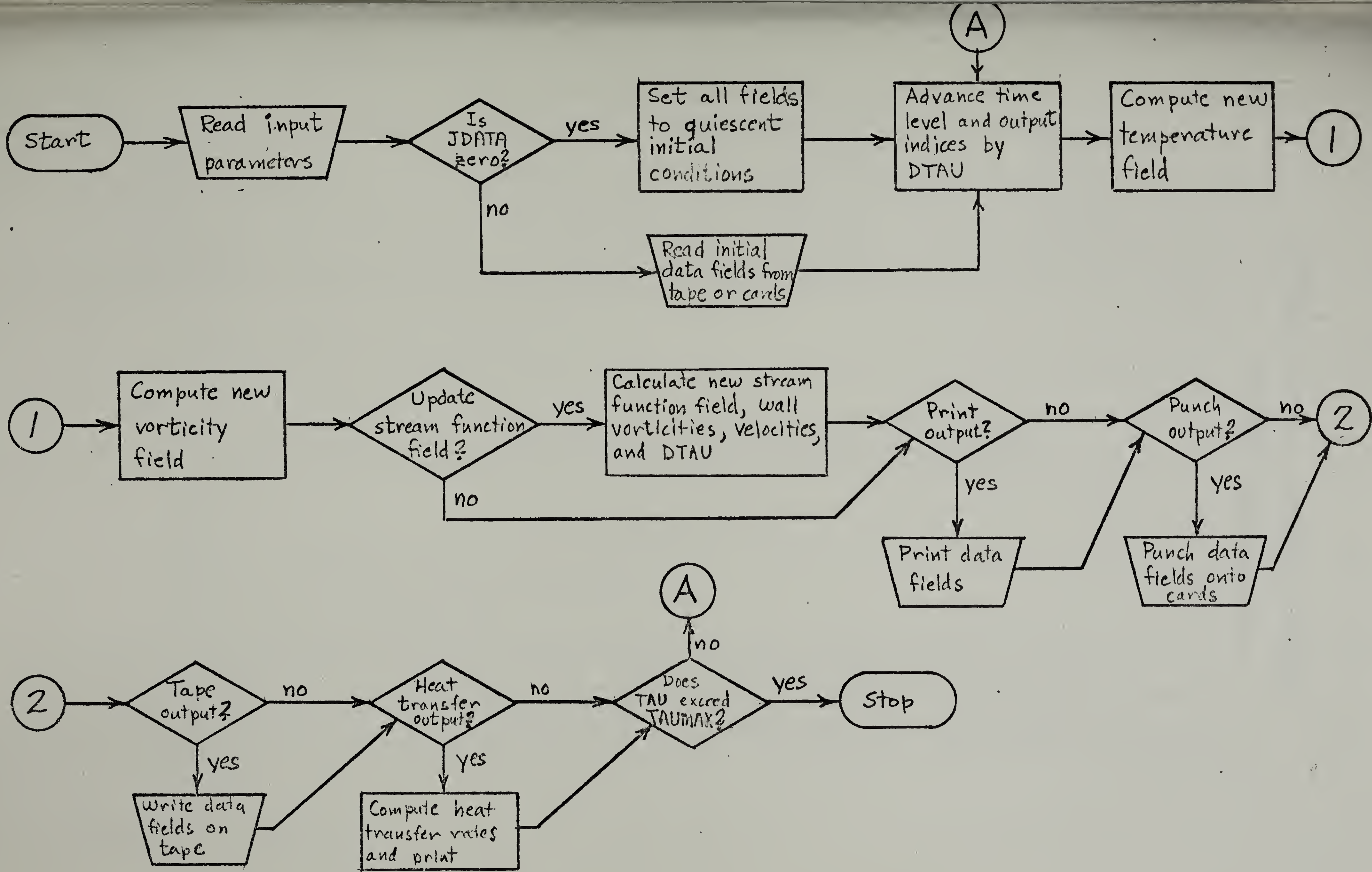

FIGURE 2. Flowchart of natural convection calculation. The corresponding computer program is shown in figure 3.

FIGURE 3 (16 pages). Listing of EXCON1

Fortran language program.

```
' RUN TORRAN,19350,10,130,1400
'IT FOR EXCON1,EXCON1
C
C  THE PROBLEM OF NATURAL CONVECTION IN A ROOM INDUCED BY A HEAT SOURCE AT
C  FLOOR LEVEL. EXPLICIT CALCULATION SCHEME, WITH THE DIRECTION OF CONVECTIVE
C  TRANSPORT BETWEEN NODES DETERMINED BY THE MEAN VELOCITY BETWEEN THE
C  NODES.  STATISFIES MASS CONSERVATION APPROXIMATELY, TRANSPORT TERMS OF
C  ENERGY AND VORTICITY EQUATIONS CONSERVATIVE.  TIME STEP USED TO ADVANCE
C  THE CALCULATION IS LIMITED BY STABILITY CONSIDERATIONS.  TEMPERATURE,
C  VELOCITIES, STREAM FUNCTION AND VORTICITY ARE OBTAINED AS FUNCTIONS OF
C  CYLINDRICAL COORDINATE POSITION (X,R) AND TIME (TAU).
C
C      PROGRAM ROOM FLOW
       INTEGER OUTAPE,OUTBLK
C
C  READ INPUT PARAMETERS
    1 READ(5,5) RB,RS,GR,PR,M,N,TAUZER,DTAU,TAUMAX,TAUNOM,RELAX,JDATA,
     1ITMAX,FKTR,FPRINT,FTAPE,FPUNCH,FPLOT,INTAPE,OUTAPE,INBLK,OUTBLK
    5 FORMAT(F8.0,F7.0,F15.0,F10.0,2I5,2E15.8/3F6.0,2I6,5F10.0/4I10)
      MM=M+1
      NN=N+1
      COMMON T(101,21), Z(101,21), U(101,21), V(101,21), UM(101,21),
     1VM(101,21), PSI(101,21)
      DIMENSION TSTAR(101,21), ZSTAR(101,21)
      WRITE (6,10) RB,RS,GR,PR
   10 FORMAT(1H1//////10X,81HNATURAL CONVECTION IN A ROOM ARISING FROM A
     1 LOCALIZED HEAT SOURCE AT FLOOR LEVEL.///20X,33HHEAT SOURCE AT TEM
     2PERATURE T = 1./ 20X,48HCEILING, WALLS, AND REMAINDER OF FLOOR AT
     3T = 0.///15X,16HINPUT PARAMETERS///20X,'RADIUS/HEIGHT = B/A = RB =
     4 ',F7.3,6X, 'HEAT SOURCE RADIUS/RADIUS = C/B = RS =',F7.3 //20X,
     5'GRASHOF BASED ON HEIGHT = GR =',1PE11.3,10X,'PRANDTL = PR =',
     60PF7.3)
C
C  COMPUTE CONSTANTS
      DX = 1.0/FLOAT(M)
      DR = RB/FLOAT(N)
      DXSQ=DX*DX
      DRSQ=DR*DR
      A1 = 1./DX
      A2 = 1./DR
      A3 = 1./DXSQ
      A4 = 1./DRSQ
```

```
      A5 = 2./DXSQ
      A6 = 2./DRSQ
      A8 = 4./DR
      A9 = 4./DRSQ
      A10 = PR/DXSQ
      A11 = PR/DRSQ
      A12 = 2.*PR/DXSQ
      A13 = 0.5*GR*PR*PR/DR
      A14 = 0.5/(DRSQ*RB)
      A15 = 0.5/(DR*DXSQ)
      RELAX1 = 1. - RELAX
      DTAUMX = .95/(A5 + A9)
      IF(JDATA.EQ.0)  DTAU =DTAUMX
      WRITE (6,15)  M,N,DX,DR,TAUZER,DTAU,TAUMAX,RELAX,ITMAX,FKTR,
     1FPRINT,FTAPE,FPUNCH
   15 FORMAT (/,20X, 4HM = , I4, 5X, 4HN = , I4, 5X, 4HDX =, F9.6, 5X,
     14HDR =, F9.6//20X, 8HTAUZER =,1PE14.7,5X,10HDTAUZERO =,1PE14.7,5X,
     28HTAUMAX =,0PF10.8//20X,7HRELAX =,F6.2,2X,8HITMAX = , I2, 2X,
     35HKTR =,F9.7,2X,7HPRINT =,F9.7,2X,6HTAPE =,F9.7,2X,7HPUNCH =,F6.4,
     4//////////)
C
C  INITIALIZE VELOCITIES, VORTICITY, STREAM FUNCTION AND TEMPERATURE
      KTOTAL = MM*NN
      JRS= FLOAT(N)*RS + 1.001
      IF(JDATA.NE.0) GO TO 30
      DO 20 I=1,MM
      DO 20 J= 1,NN
      T(I,J)  = 0.
      TSTAR(I,J)=T(I,J)
      Z(I,J)  = 0.
      ZSTAR(I,J)=Z(I,J)
      U(I,J)  = 0.
      V(I,J)  = 0.
      UM(I,J) = 0.
      VM(I,J) = 0.
   20 PSI(I,J) = 0.
      DO 25 J=1,JRS
      T(1,J) = 1.0
      IF(J.EQ.JRS) T(1,J) = 0.5
   25 TSTAR(1,J) = T(1,J)
      GO TO 50
   30 CALL CHECK(INTAPE,INBLK)
```

```
      READ(INTAPE,35)(I,J,T(I,J), U(I,J), V(I,J), Z(I,J), PSI(I,J) ,
     1 K = 1, KTOTAL)
   35 FORMAT (2I5,  5E14.7)
C  INSERT CARD TO SKIP OVER ONE END OF FILE MARK ON INTAPE
      DO 40 I=1,MM
      ZSTAR(I,1) = Z(I,1)
   40 TSTAR(I,NN) = T(I,NN)
      DO 45 J=1,NN
      TSTAR(1,J)=T(1,J)
   45 TSTAR(MM,J) = T(MM,J)
      ZSTAR(1,NN) = Z(1,NN)
      ZSTAR(MM,NN) = Z(MM,NN)
      CALL MEAN (MM,NN,A1,A2)
   50 TAU = TAUZER
      GPRINT = 0.
      GPLOT = 0.
      GPUNCH=0.
      GTAPE = 0.
      GKTR = 0. + FKTR
      ITE = 0
      KTRVEL = 40
      IF (JDATA.EQ.0)  GO TO 51
      GPRINT = TAU - TAUNOM
      GPLOT = TAU - TAUNOM
      GPUNCH = TAU - TAUNOM
      GTAPE = TAU - TAUNOM
      GKTR = TAU - TAUNOM + FKTR
   51 CALL NUSSEL (M, N, JRS,RB, TAU,DTAU,GKTR,FKTR,DX,DR)
      ITE = ITMAX
C
C  START COMPUTATION LOOP
   52 TAU = TAU + DTAU
      KTRVEL = KTRVEL + 1
      GPRINT = GPRINT + DTAU
      GPLOT = GPLOT + DTAU
      GPUNCH = GPUNCH + DTAU
      GTAPE = GTAPE + DTAU
      GKTR = GKTR + DTAU
      A7=1.0/DTAU
C
C  COMPUTE NEW TEMPERATURE FIELD TSTAR
C
```

```
      D3 = A7 - (A5 + A9)
      DO 70 I = 2,M
      D1 = A3
      D2 = A3
      D31 = D3
      D4 = A9
      IF(UM(I,1).GT.0.)   D31 = D31 - UM(I,1)
      IF(UM(I,1).LT.0.)   D1  = D1  - UM(I,1)
      IF(UM(I-1,1).GT.0.)  D2 = D2 + UM(I-1,1)
      IF(UM(I-1,1).LT.0.) D31 = D31+ UM(I-1,1)
      IF(VM(I,1).GT.0.)   D31 = D31 - 4.*VM(I,1)
      IF(VM(I,1).LT.0.)   D4  = D4  - 4.*VM(I,1)
   69 TSTAR(I,1) = DTAU*(D1*T(I+1,1) + D2*T(I-1,1)+D31*T(I,1)+D4*T(I,2))
   70 CONTINUE
      IF(ABS(TSTAR(M,1)-.5).GT.0.51)  GO TO 250
      D3 = A7 - (A5 + A6)
      DO 80 J = 2,N
      FJ = 1./FLOAT(J-1)
      FJ1 = 1. + .5*FJ
      FJ2 = 1. - .5*FJ
      D4 = FJ1*A4
      D5 = FJ2*A4
      DO 80 I = 2,M
      D1 = A3
      D2 = A3
      D31 = D3
      D41 = D4
      D51 = D5
      IF(UM(I,J).GT.0.)  D31 = D31 - UM(I,J)
      IF(UM(I,J).LT.0.)  D1  = D1  - UM(I,J)
      IF(UM(I-1,J).GT.0.)  D2 = D2+ UM(I-1,J)
      IF(UM(I-1,J).LT.0.)D31 =D31 + UM(I-1,J)
      IF(VM(I,J).GT.0.)  D31 = D31 - FJ1*VM(I,J)
      IF(VM(I,J).LT.0.)  D41 = D41 - FJ1*VM(I,J)
      IF(VM(I,J-1).GT.0.)   D51 = D51 + FJ2*VM(I,J-1)
      IF(VM(I,J-1).LT.0.)   D31 = D31 + FJ2*VM(I,J-1)
   79 TSTAR(I,J) = DTAU*(D1*T(I+1,J) + D2*T(I-1,J) + D31*T(I,J)
     1 + D41*T(I,J+1) + D51*T(I,J-1))
   80 CONTINUE
      IF(ABS(TSTAR(3,3)-.5).GT.0.51)  GO TO 250
C
C  COMPUTE NEW VORTICITY FIELD ZSTAR
```

```
C
      DO 100 J=2,N
      FJ = FLOAT(J)
      FJ1 = (FJ-1.)/(FJ-.5)
      FJ2 = (FJ-1.)/(FJ-1.5)
      FJ3= FJ/(FJ-.5)
      FJ4 = (FJ-2.)/(FJ-1.5)
      E3 = A7 - A12 - A11*(FJ1 + FJ2)
      E4 = FJ3*A11
      E5 = FJ4*A11
      DO 100  I = 2,M
      E1 = A10
      E2 = A10
      E31 = E3
      E41 = E4
      E51 = E5
      IF(UM(I,J).GT.0.)  E31 = E31 - UM(I,J)
      IF(UM(I,J).LT.0.)  E1  = E1  - UM(I,J)
      IF(UM(I-1,J).GT.0.)  E2  = E2  + UM(I-1,J)
      IF(UM(I-1,J).LT.0.)  E31 = E31 + UM(I-1,J)
      IF(VM(I,J).GT.0.)  E31 = E31 - VM(I,J)
      IF(VM(I,J).LT.0.)  E41 = E41 - VM(I,J)
      IF(VM(I,J-1).GT.0.)  E51 = E51 + VM(I,J-1)
      IF(VM(I,J-1).LT.0.)  E31 = E31 + VM(I,J-1)
  100 ZSTAR(I,J) = DTAU*(E1*Z(I+1,J) + E2*Z(I-1,J) + E31*Z(I,J) +E41*
     1 Z(I,J+1) + E51*Z(I,J-1) -A13*(TSTAR(I,J+1)-TSTAR(I,J-1)))
C
C  EQUATE T AND Z TO NEW VALUES TSTAR AND ZSTAR
  129 DO 130 I = 2,M
      DO 130 J = 1,N
      T(I,J) = TSTAR(I,J)
  130 Z(I,J) = ZSTAR(I,J)
C
C  CHECK FOR UPDATE OF STREAM FUNCTION FIELD.  THE FOLLOWING CARD SPECIFIES
C  AN EMPIRICAL VALUE OF ITEVEL.  AT HIGH GR, ITEVEL=0.25*(ITMAX-ITE)+1.
      ITEVEL = 1
  132 IF(KTRVEL.LE.ITEVEL)  GO TO 237
      KTRVEL = 0
C
C  CALCULATE NEW STREAM FUNCTION FIELD
      DO 160 ITER = 1,ITMAX
      REFPSI = 0.
```

```
      DELPSI = 0.
      DO 150 J=2,N
      FJ= FLOAT(J-1)
      D1 = FJ*DR
      D2 = A4*FJ/(FJ+.5)
      D3 = A4*FJ/(FJ-.5)
      REL = RELAX/(A5 + D2 + D3)
      DO 150 I=2,M
      PSI(I,J) = RELAX1*PSI(I,J) + REL*(D1* Z(I,J) + A3*(PSI(I+1,J) +
     1 PSI(I-1,J)) + D2*PSI(I,J+1) + D3*PSI(I,J-1))
  150 CONTINUE
      DO 155 I=2,M
      DO 155 J=2,N
      FJ= FLOAT(J-1)
      D1 = FJ*DR
      D2 = A4*FJ/(FJ+.5)
      D3 = A4*FJ/(FJ-.5)
      REL = RELAX/(A5 + D2 + D3)
      P = PSI(I,J)
      PSI(I,J) = RELAX1*PSI(I,J) + REL*(D1* Z(I,J) + A3*(PSI(I+1,J) +
     1 PSI(I-1,J)) + D2*PSI(I,J+1) + D3*PSI(I,J-1))
      IF(PSI(I,J).GT.REFPSI)  REFPSI = PSI(I,J)
      DEL = ABS(PSI(I,J) - P)
      IF(DEL.GT.DELPSI) DELPSI=DEL
  155 CONTINUE
      ITE = ITER
      IF((DELPSI/REFPSI).LT.0.0001)  GO TO 210
  160 CONTINUE
C
C  COMPUTE NEW VORTICITIES AT BOUNDARIES
  210 DO 220 I=2,M
  220 Z(I,NN) =(-8.*PSI(I,N) + PSI(I,N-1))*A14
      DO 230 J=2,N
      FJ = A15/FLOAT(J-1)
      Z(1,J) =(-8.*PSI(2,J) + PSI(3,J))*FJ
  230 Z(MM,J) =(-8.*PSI(M,J) + PSI(M-1,J))*FJ
C
C  CALCULATE NEW VELOCITIES U, V, UM AND VM
      CALL VEL(MM,NN,DX,DR,RB)
      CALL MEAN(MM,NN,A1,A2)
C
C  CALCULATE NEW DTAU
```

```
      VELMXC = 0.
      VELMX  = 0.
      DO 232 I= 2,M
      VELTST = 0.
      IF(UM(I,1).GT.0.)  VELTST = VELTST + UM(I,1)
      IF(VM(I,1).GT.0.)  VELTST = VELTST + 4.*VM(I,1)
      IF(UM(I-1,1).LT.0.)  VELTST = VELTST - UM(I-1,1)
      IF(VELTST.GT.VELMXC)  VELMXC = VELTST
  232 CONTINUE
      VELMXC = VELMXC + A5 + A9
      DO 235 J=2,N
      FJ1 = 1. + .5/FLOAT(J-1)
      DO 235 I = 2,M
      VELTST = 0.
      IF(UM(I,J).GT.0.)  VELTST = VELTST + UM(I,J)
      IF(VM(I,J).GT.0.)  VELTST = VELTST + FJ1*VM(I,J)
      IF(UM(I-1,J).LT.0.)  VELTST = VELTST - UM(I-1,J)
      IF(VM(I,J-1).LT.0.)  VELTST = VELTST - VM(I,J-1)
      IF(VELTST.GT.VELMX)  VELMX = VELTST
  235 CONTINUE
      VELMX = VELMX + A5 + A6
      IF(VELMXC.GT.VELMX)  GO TO 236
      DTAU = .95/VELMX
      GO TO 238
  236 DTAU = .95/VELMXC
  238 IF(DTAU.GT.DTAUMX)  DTAU = DTAUMX
C
C  LOOP COMPLETE - - CHECK FOR OUTPUT
  237 IF(GPRINT.GT.FPRINT) CALL RESLTS(MM,NN,TAU,DTAU,GPRINT,FPRINT,ITE)
      IF(GPUNCH.GT.FPUNCH)CALL PUN(M,N,RB,RS,GR,PR,TAU,DTAU,GPUNCH,FPUNC
     1H)
      IF(GTAPE.GT.FTAPE)CALL TAP(M,N,RB,RS,GR,PR,TAU,DTAU,GTAPE,FTAPE,
     1 OUTAPE,OUTBLK)
  239 IF(GKTR.GT.FKTR)CALL NUSSEL(M,N,JRS,RB,TAU,DTAU,GKTR,FKTR,DX,DR)
C
C  CHECK FOR RETURN TO START OF LOOP
      IF (TAU.GT.TAUMAX)  GO TO 245
  240 GO TO 52
  245 CONTINUE
  250 STOP
      END
```

FIGURE 3 (cont'd)

```
'IT FOR VELCD,VELCD
C
C  PROGRAM TO COMPUTE VELOCITIES U AND V AT NODE POINTS
C
      SUBROUTINE VEL(MM,NN,DX,DR,RB)
      COMMON T(101,21), Z(101,21), U(101,21), V(101,21), UM(101,21),
     1VM(101,21), PSI(101,21)
      C1 =  0.5/(DR*DR)
      C2 = -0.5/(DX*DR)
      C3 = 2./(DR*DR)
      C4 = -1./(DR*(RB-DR/4.))
      C5 = 1./(DX*DR)
      DO 5  I = 1,MM
      U(I,1) = C3*PSI(I,2)
    5 V(I,1) = 0.
      DO 30 J = 2,NN
      FJINV = 1./FLOAT(J-1)
      C6 = C1*FJINV
      C7 = C2*FJINV
      C8 = C5*FJINV
      DO 30 I = 1,MM
      IF(J.LT.NN) GO TO 10
      U(I,J) = C4*PSI(I,J-1)
      GO TO 15
   10 U(I,J) = C6*(PSI(I,J+1) - PSI(I,J-1))
   15 IF(I.GT.1.AND.I.LT.MM)  GO TO 20
      IF(I.EQ.1) V(I,J) = -C8*PSI(I+1,J)
      IF(I.EQ.MM) V(I,J) = C8*PSI(I-1,J)
      GO TO 30
   20 V(I,J) = C7*(PSI(I+1,J) - PSI(I-1,J))
   30 CONTINUE
      END
```

```
'IT FOR MEANAV,MEANAV
C
C  PROGRAM TO COMPUTE INTER-NODE TRANSPORT VELOCITIES UM AND VM
C
      SUBROUTINE MEAN(MM,NN,A1,A2)
      COMMON T(101,21), Z(101,21), U(101,21), V(101,21), UM(101,21),
     1VM(101,21), PSI(101,21)
      A1A = A1*.5
      A2A = A2*.5
      DO 30 J = 1,NN
      DO 30 I = 1,MM
      IF(I.EQ.MM) GO TO 20
      UM(I,J) = (U(I,J) + U(I+1,J))*A1A
   20 IF(J.EQ.NN)  GO TO 30
      VM(I,J) = (V(I,J) + V(I,J+1))*A2A
   30 CONTINUE
      END
```

FIGURE 3 (cont'd) page 9

```
'IT FOR PRINT,PRINT
C
C  PROGRAM TO CONTROL PRINTING OF TEMPERATURE, VELOCITY, VORTICITY AND STREAM
C  FUNCTION FIELDS FOR ANY NUMBER OF POINTS (N) ACROSS THE ENCLOSURE.
C
      SUBROUTINE RESLTS(MM,NN,TAU,DTAU,GPRINT,FPRINT,ITE)
      COMMON T(101,21), Z(101,21), U(101,21), V(101,21), UM(101,21),
     1VM(101,21), PSI(101,21)
      WRITE (6,600)    TAU
  600 FORMAT(1H1      10X,9HAT A TIME//10X,6HTAU = ,1PE15.7// 10X, 33HTHE
     1 TEMPERATURE FIELD IS GIVEN BY)
      CALL BLOCK (MM, NN,     T, 1)
      WRITE (6,620)
  620 FORMAT (//10X,45HTHE FIELD OF VELOCITY COMPONENT U IS GIVEN BY)
      CALL BLOCK (MM, NN,     U, 2)
      WRITE (6,630)
  630 FORMAT (//10X,45HTHE FIELD OF VELOCITY COMPONENT V IS GIVEN BY)
      CALL BLOCK (MM, NN,     V, 2)
      WRITE (6,640)
  640 FORMAT(//10X,34HTHE FIELD OF VORTICITY IS GIVEN BY)
      CALL BLOCK (MM, NN,     Z, 2)
      WRITE (6,650).
  650 FORMAT(//10X,37HTHE STREAM FUNCTION FIELD IS GIVEN BY)
      CALL BLOCK (MM, NN,  PSI, 2)
      WRITE (6,660)   ITE
  660 FORMAT (10X, 29HSTREAM FUNCTION ITERATIONS  =, I4)
  670 GPRINT = GPRINT - FPRINT
      END
```

FIGURE 3 (cont'd)

```
'IT FOR BLOCK,BLOCK
C
C  PROGRAM TO PRINT BLOCKS OF OUTPUT DATA.  CALLED FROM SUBROUTINE RESLTS.
C
      SUBROUTINE BLOCK (MM, NN, DUMMY, KFOR)
C  THE DIMENSION OF DUMMY AND THE FIELDS IN COMMON MUST CORRESPOND
      DIMENSION DUMMY(101,21)
      M = MM - 1
      N = NN - 1
      NUMBER = N/10
    1 FORMAT (3X, 10F10.6)
    2 FORMAT (3X, 1P10E10.3)
    3 FORMAT (3X, 11F10.6)
    4 FORMAT (3X, 1P11E10.3)
      DO 700 K = 1,NUMBER
      IND1 = 1 + 10*(K-1)
      IND2 = 10*K
      PRINT 650
  650 FORMAT (1H )
      DO 700 I = 1, MM
      II = MM + 1 - I
      IF(K.EQ.NUMBER) GO TO 690
      IF                (KFOR.EQ.1) PRINT 1,(DUMMY(II,J),J=IND1,IND2)
      IF                (KFOR.EQ.2) PRINT 2,(DUMMY(II,J),J=IND1,IND2)
      GO TO 700
  690 IF                (KFOR.EQ.1) PRINT 3,(DUMMY(II,J),J=IND1,NN)
      IF                (KFOR.EQ.2) PRINT 4,(DUMMY(II,J),J=IND1,NN)
  700 CONTINUE
      END
```

```
'IT FOR PUNCH,PUNCH
C
C  PROGRAM TO PUNCH RESULTS OF TEMPERATURE, VELOCITIES, VORTICITY AND STREAM
C  FUNCTION WHICH ARE ACCEPTABLE AS INPUT DATA
C
      SUBROUTINE PUN(M,N,RB,RS,GR,PR,TAU,DTAU,GPUNCH,FPUNCH)
      COMMON T(101,21), Z(101,21), U(101,21), V(101,21), UM(101,21),
     1VM(101,21), PSI(101,21)
      MM = M+1
      NN = N+1
      PUNCH 710,    RB,RS,GR,PR,M,N,TAU,DTAU
  710 FORMAT(F8.3,F7.3,F15.0,F10.3,2I5,2E15.8)
      PUNCH 720   (( I,J, T(I,J), U(I,J), V(I,J), Z(I,J), PSI(I,J),
     1I=1,MM),J=1,NN)
  720 FORMAT (2I5,  5E14.7)
      WRITE (6,730)
  730 FORMAT  (1H /10X,13HCARDS PUNCHED)
      GPUNCH = GPUNCH - FPUNCH
      END
```

```
'IT FOR TAPE,TAPE
C
C  PROGRAM TO PUT RESULTS ON TAPE
C
      SUBROUTINE TAP(M,N,RB,RS,GR,PR,TAU,DTAU,GTAPE,FTAPE,IO,IBLOCK)
      COMMON T(101,21), Z(101,21), U(101,21), V(101,21), UM(101,21),
     1VM(101,21), PSI(101,21)
      MM = M+1
      NN = N+1
      IF(IBLOCK.GT.1)  GO TO 5
C  INSERT CARD TO PLACE AN END OF FILE MARK ON IO TAPE
    5 DO 20 I = 1,2
      PUNCH  10,     RB,RS,GR,PR,M,N,TAU,DTAU
   10 FORMAT(F8.3,F7.3,F15.0,F10.3,2I5,2E15.8)
   20 CONTINUE
      WRITE(IO,10)     RB,RS,GR,PR,M,N,TAU,DTAU
      WRITE(IO,35)(( I,J, T(I,J), U(I,J), V(I,J), Z(I,J), PSI(I,J),
     1I=1,MM),J=1,NN)
   35 FORMAT (2I5,  5E14.7)
C  INSERT CARD TO PLACE AN END OF FILE MARK ON IO TAPE
      WRITE(6,40)IBLOCK,RB,RS,GR,PR,M,N,TAU,DTAU
   40 FORMAT (1H /    'RESULTS ON TAPE IN BLOCK ',I3, '*****',F8.3,F7.3,
     1 F15.0,F10.3,2I5,2E15.8,'*****')
      IBLOCK = IBLOCK + 1
      GTAPE = GTAPE - FTAPE
      END
```

```
'IT FOR CHECK,CHECK
C
C  PROGRAM TO FIND AND VERIFY DATA BLOCK ON IOTAPE
C
      SUBROUTINE CHECK(IOTAPE,IOBLK)
      DIMENSION M(80), N(80)
    5 FORMAT(80A1)
    6 FORMAT(//10X,'DATA BLOCK', I3, '   *****',80A1,'*****'//)
    7 FORMAT(//10X,'**************************************************'/
     1  10X,  '*',48X,'*'/  10X,'*      THIS DATA BLOCK IS MISNUMBERED ON
     2 TAPE      *'/  10X,'*',48X,'*'/   10X,'**************************
     3************************'//)
      READ(5,5) M
      KPLUS = IOBLK + 5
      DO 30 K = 1,KPLUS
C  INSERT CARD TO SKIP OVER ONE END OF FILE MARK ON IOTAPE
      READ(IOTAPE,5) N
      DO 10 I = 1,80
      IF(M(I).NE.N(I))  GO TO 20
   10 CONTINUE
      IF(K.NE.IOBLK)  PRINT 7
      PRINT 6, K,N
      GO TO 40
   20 IF(K.LT.IOBLK) GO TO 30
      PRINT 7
      PRINT 6, K,N
      IF(K.EQ.KPLUS) GO TO 45
   30 CONTINUE
   40 RETURN
   45 STOP
      END
```

```
'IT FOR NUSSEL,NUSSEL
C
C  CALCULATION OF HEAT TRANSFER RATES WITHIN ENCLOSURE
C
      SUBROUTINE NUSSEL(M,N,JRS,RB,TAU,DTAU,GKTR,FKTR,DX,DR)
      COMMON T(101,21), Z(101,21), U(101,21), V(101,21), UM(101,21),
     1VM(101,21), PSI(101,21)
      PI = 3.14159265
      A1 = 1./DX
      A2 = 1./DR
      AJ1 = PI*DR*DR/4.
      AJ2 = 2.*PI*DR*DR
      DTDX = 0.
      WRITE (6,10)  TAU, DTAU
   10 FORMAT(/2X,15HAT A TIME TAU =,1PE15.7,25X,11HNEXT DTAU =,1PE15.7)
      JRS1 = JRS + 1
      DO 30 J=1,JRS
      FJ = FLOAT(J-1)
      IF(J.EQ.1)  AREA = AJ1
      IF(J.GT.1)  AREA = AJ2*FJ
      IF(UM(1,J).GT.0.)  DTDX = DTDX + UM(1,J)*DX*AREA*T(1,J)
      IF(UM(1,J).LT.0.)  DTDX = DTDX + UM(1,J)*DX*AREA*T(2,J)
   30 DTDX =  AREA*A1*(T(1,J) - T(2,J)) + DTDX
      AREA = PI*DX*DR*(FLOAT(JRS) - .5)
      QEDGE = AREA*A2*(T(1,JRS) - T(1,JRS1))
      IF(VM(1,JRS).GT.0.) QEDGE=QEDGE+AREA*VM(1,JRS)*DR*T(1,JRS)
      IF(VM(1,JRS).LT.0.) QEDGE=QEDGE+AREA*VM(1,JRS)*DR*T(1,JRS1)
      DTDX = QEDGE + DTDX
      DTDN = QEDGE
      DO 50 J = 1,N
      FJ = FLOAT(J-1)
      IF(J.EQ.1)  AREA = AJ1
      IF(J.GT.1)  AREA = AJ2*FJ
      IF(J.LT.JRS1) GO TO 40
      IF(UM(1,J).GT.0.)  DTDN = DTDN - UM(1,J)*DX*AREA*T(1,J)
      IF(UM(1,J).LT.0.)  DTDN = DTDN - UM(1,J)*DX*AREA*T(2,J)
      DTDN = -AREA*A1*(T(1,J) - T(2,J)) + DTDN
   40 IF(UM(M,J).GT.0.)  DTDN = DTDN + UM(M,J)*DX*AREA*T(M,J)
   50 DTDN = AREA*A1*T(M,J) + DTDN
      AREA = PI*(2.*RB-DR)*DX
      DO 60 I = 2,M
      IF(VM(I,N).GT.0.) DTDN = DTDN + VM(I,N)*DR*AREA*T(I,N)
```

```
    60 DTDN = AREA*A2*T(I,N) + DTDN
       AREA1 = PI*DR*DR*(FLOAT(JRS) - .5)**2
       AREA2 = PI*2.*RB*(RB+1.) - AREA1
       QHS = DTDX
       FNUHS = QHS/AREA1
       QCW = DTDN
       FNUCW = QCW/AREA2
    65 WRITE(6,70)      FNUHS, QHS, FNUCW, QCW
    70 FORMAT(    1X,  11H2-PT APPROX  , 4X, 12HNUSSELT HS =   , F11.4,
      13X, 12HQHS/K*A*DT =  ,F10.4, 4X, 12HNUSSELT CW = , F9.4, 3X,
      2 12HQCW/K*A*DT = , F9.4)
       GKTR = GKTR - FKTR
       END
C
C
' XQT EXCON1
C
C  DATA CARDS CONTAINING INPUT PARAMETER LIST
C
     1.0   .100          100000.        .700   20   20   .00000000+00   .00000000+00
  .2500 .0000  1.75      0      15    .001000     .01000     .01000        .250      0.100
         7          7          0          0
C
' FIN
```

FIGURE 3 (cont'd) page 16

NATURAL CONVECTION IN A ROOM ARISING FROM A LOCALIZED HEAT SOURCE AT FLOOR LEVEL.

HEAT SOURCE AT TEMPERATURE T = 1.
CEILING, WALLS, AND REMAINDER OF FLOOR AT T = 0.

INPUT PARAMETERS

RADIUS/HEIGHT = B/A = RB = 1.000 HEAT SOURCE RADIUS/RADIUS = C/B = RS = .100

GRASHOF BASED ON HEIGHT = GR = 1.000+05 PRANDTL = PR = .700

M = 20 N = 20 DX = .050000 DR = .050000

TAUZER = 0.0000000 DTAUZERO = 3.9583333-04 TAUMAX = .25000000

RELAX = 1.75 ITMAX = 15 KTR = .0010000 PRINT = .0100000 TAPE = .0100000 PUNCH = .2500

AT A TIME

TAU = 2.5084057-01

THE TEMPERATURE FIELD IS GIVEN BY

.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000
.018008	.017492	.016278	.014639	.012827	.011022	.009331	.007809	.006477	.005338
.031463	.030607	.028581	.025868	.022882	.019901	.017080	.014496	.012177	.010124
.040075	.038952	.036248	.032699	.028906	.025229	.021828	.018748	.015976	.013483
.045579	.044157	.040686	.036234	.031657	.027425	.023694	.020449	.017604	.015067
.049712	.047919	.043514	.037979	.032496	.027667	.023643	.020336	.017573	.015185
.053551	.051306	.045786	.038984	.032463	.026962	.022616	.019246	.016588	.014397
.057682	.054890	.048058	.039811	.032147	.025926	.021232	.017778	.015200	.013183
.062460	.058998	.050610	.040723	.031833	.024867	.019825	.016275	.013746	.011861
.068184	.063880	.053603	.041831	.031634	.023953	.018572	.014905	.012388	.010590
.075199	.069797	.057210	.043307	.031691	.023239	.017498	.013699	.011172	.009428
.083998	.077164	.061682	.045220	.031991	.022700	.016580	.012643	.010094	.008387
.095277	.086511	.067253	.047621	.032510	.022294	.015779	.011702	.009129	.007452
.110060	.098606	.074259	.050580	.033220	.021973	.015047	.010837	.008247	.006600
.129982	.114676	.083230	.054206	.034092	.021679	.014331	.010003	.007410	.005802
.157796	.136839	.095080	.058667	.035075	.021331	.013560	.009145	.006578	.005028
.198451	.169056	.111535	.064203	.036041	.020777	.012620	.008186	.005697	.004238
.261623	.219653	.136308	.071008	.036575	.019672	.011295	.006996	.004689	.003387
.368587	.309063	.178818	.078191	.035222	.017166	.009162	.005372	.003455	.002420
.573223	.498650	.268151	.076410	.027010	.011361	.005536	.003062	.001892	.001288
1.000000	1.000000	.500000	.000000	.000000	.000000	.000000	.000000	.000000	.000000

.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000
.004378	.003562	.002865	.002271	.001768	.001342	.000984	.000683	.000426	.000203	.000000
.008353	.006824	.005507	.004378	.003414	.002596	.001905	.001322	.000825	.000393	.000000
.011252	.009283	.007555	.006049	.004747	.003628	.002673	.001860	.001164	.000555	.000000
.012760	.010670	.008792	.007118	.005639	.004346	.003225	.002257	.001418	.000677	.000000
.013040	.011061	.009242	.007580	.006077	.004732	.003542	.002496	.001576	.000755	.000000
.012487	.010730	.009088	.007553	.006131	.004827	.003648	.002590	.001644	.000789	.000000
.011493	.009975	.008549	.007193	.005908	.004703	.003588	.002567	.001638	.000789	.000000
.010350	.009038	.007815	.006642	.005514	.004433	.003413	.002460	.001578	.000762	.000000
.009210	.008057	.007009	.006007	.005031	.004082	.003168	.002300	.001483	.000718	.000000
.008144	.007112	.006204	.005347	.004510	.003687	.002883	.002106	.001365	.000663	.000000
.007173	.006235	.005437	.004699	.003983	.003275	.002577	.001893	.001232	.000600	.000000
.006296	.005432	.004722	.004083	.003470	.002865	.002264	.001670	.001091	.000532	.000000
.005494	.004696	.004061	.003504	.002980	.002466	.001954	.001446	.000947	.000463	.000000
.004751	.004014	.003446	.002962	.002515	.002082	.001652	.001225	.000803	.000393	.000000
.004041	.003369	.002866	.002449	.002073	.001714	.001360	.001009	.000662	.000324	.000000
.003337	.002740	.002306	.001957	.001649	.001360	.001078	.000800	.000525	.000257	.000000
.002609	.002107	.001752	.001475	.001237	.001017	.000804	.000596	.000391	.000191	.000000
.001822	.001447	.001189	.000993	.000828	.000678	.000535	.000396	.000259	.000127	.000000
.000950	.000744	.000605	.000502	.000417	.000340	.000268	.000198	.000129	.000063	.000000
.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000	.000000

THE FIELD OF VELOCITY COMPONENT U IS GIVEN BY

0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5.112+00	4.733+00	4.086+00	3.220+00	2.319+00	1.508+00	8.505-01	3.607-01	2.444-02	-1.866-01
1.456+01	1.363+01	1.202+01	9.832+00	7.464+00	5.219+00	3.276+00	1.710+00	5.298-01	-2.953-01
2.385+01	2.245+01	2.005+01	1.673+01	1.310+01	9.575+00	6.427+00	3.774+00	1.649+00	3.915-02
3.139+01	2.963+01	2.659+01	2.242+01	1.784+01	1.340+01	9.393+00	5.936+00	3.065+00	7.684-01

3.696+01	3.490+01	3.134+01	2.649+01	2.123+01	1.618+01	1.164+01	7.708+00	4.379+00	1.627+00
4.082+01	3.848+01	3.446+01	2.904+01	2.328+01	1.784+01	1.302+01	8.872+00	5.342+00	2.368+00
4.332+01	4.070+01	3.626+01	3.036+01	2.420+01	1.853+01	1.360+01	9.415+00	5.869+00	2.857+00
4.477+01	4.188+01	3.702+01	3.069+01	2.424+01	1.844+01	1.352+01	9.420+00	5.979+00	3.058+00
4.541+01	4.223+01	3.698+01	3.026+01	2.359+01	1.776+01	1.293+01	8.996+00	5.743+00	2.997+00
4.543+01	4.195+01	3.630+01	2.924+01	2.242+01	1.662+01	1.196+01	8.248+00	5.240+00	2.728+00
4.496+01	4.116+01	3.511+01	2.773+01	2.082+01	1.513+01	1.070+01	7.267+00	4.549+00	2.313+00
4.404+01	3.991+01	3.346+01	2.580+01	1.887+01	1.336+01	9.223+00	6.126+00	3.739+00	1.811+00
4.269+01	3.820+01	3.135+01	2.346+01	1.659+01	1.136+01	7.599+00	4.891+00	2.871+00	1.276+00
4.083+01	3.597+01	2.875+01	2.071+01	1.402+01	9.187+00	5.883+00	3.622+00	2.003+00	7.598-01
3.829+01	3.308+01	2.556+01	1.751+01	1.118+01	6.886+00	4.147+00	2.390+00	1.197+00	3.093-01
3.471+01	2.928+01	2.164+01	1.383+01	8.105+00	4.555+00	2.491+00	1.283+00	5.196-01	-2.821-02
2.955+01	2.420+01	1.684+01	9.698+00	4.945+00	2.366+00	1.072+00	4.162-01	4.608-02	-2.110-01
2.189+01	1.731+01	1.105+01	5.316+00	2.028+00	6.312-01	1.087-01	-7.963-02	-1.604-01	-2.203-01
1.096+01	8.340+00	4.622+00	1.438+00	6.654-02	-1.989-01	-1.882-01	-1.427-01	-1.113-01	-9.879-02
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
-3.096-01	-3.753-01	-4.032-01	-4.077-01	-3.992-01	-3.833-01	-3.592-01	-3.176-01	-2.405-01	-1.159-01	-4.293-02
-8.292-01	-1.148+00	-1.315+00	-1.380+00	-1.380+00	-1.338+00	-1.255+00	-1.111+00	-8.635-01	-4.741-01	-2.443-01
-1.100+00	-1.848+00	-2.294+00	-2.522+00	-2.597+00	-2.563+00	-2.430+00	-2.170+00	-1.719+00	-9.917-01	-5.540-01
-9.752-01	-2.215+00	-3.032+00	-3.516+00	-3.749+00	-3.784+00	-3.642+00	-3.290+00	-2.642+00	-1.558+00	-8.958-01
-5.716-01	-2.240+00	-3.428+00	-4.206+00	-4.648+00	-4.808+00	-4.704+00	-4.297+00	-3.482+00	-2.073+00	-1.204+00
-9.024-02	-2.050+00	-3.531+00	-4.575+00	-5.231+00	-5.539+00	-5.503+00	-5.075+00	-4.134+00	-2.467+00	-1.434+00
3.138-01	-1.785+00	-3.446+00	-4.683+00	-5.516+00	-5.958+00	-5.997+00	-5.570+00	-4.547+00	-2.708+00	-1.569+00
5.634-01	-1.545+00	-3.271+00	-4.609+00	-5.554+00	-6.093+00	-6.192+00	-5.777+00	-4.713+00	-2.795+00	-1.609+00
6.437-01	-1.374+00	-3.065+00	-4.416+00	-5.401+00	-5.988+00	-6.122+00	-5.721+00	-4.656+00	-2.745+00	-1.569+00
5.788-01	-1.277+00	-2.858+00	-4.146+00	-5.107+00	-5.691+00	-5.834+00	-5.448+00	-4.415+00	-2.586+00	-1.468+00
4.106-01	-1.237+00	-2.653+00	-3.822+00	-4.706+00	-5.246+00	-5.375+00	-5.005+00	-4.036+00	-2.347+00	-1.323+00
1.851-01	-1.222+00	-2.440+00	-3.454+00	-4.224+00	-4.692+00	-4.791+00	-4.442+00	-3.559+00	-2.055+00	-1.150+00
-5.362-02	-1.204+00	-2.204+00	-3.041+00	-3.677+00	-4.059+00	-4.122+00	-3.798+00	-3.022+00	-1.732+00	-9.619-01
-2.659-01	-1.154+00	-1.931+00	-2.585+00	-3.082+00	-3.372+00	-3.399+00	-3.110+00	-2.454+00	-1.394+00	-7.679-01
-4.169-01	-1.051+00	-1.613+00	-2.090+00	-2.452+00	-2.655+00	-2.655+00	-2.408+00	-1.880+00	-1.056+00	-5.750-01
-4.780-01	-8.810-01	-1.248+00	-1.566+00	-1.807+00	-1.937+00	-1.919+00	-1.722+00	-1.326+00	-7.315-01	-3.906-01
-4.323-01	-6.456-01	-8.516-01	-1.036+00	-1.177+00	-1.250+00	-1.227+00	-1.088+00	-8.212-01	-4.384-01	-2.242-01
-2.584-01	-3.697-01	-4.585-01	-5.433-01	-6.099-01	-6.425-01	-6.260-01	-5.473-01	-4.010-01	-2.002-01	-9.122-02
-1.021-01	-1.168-01	-1.379-01	-1.605-01	-1.793-01	-1.887-01	-1.834-01	-1.583-01	-1.105-01	-4.622-02	-1.231-02
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

THE FIELD OF VELOCITY COMPONENT V IS GIVEN BY

0.000	2.556+00	4.733+00	6.300+00	7.197+00	7.490+00	7.311+00	6.808+00	6.114+00	5.339+00
0.000	3.640+00	6.813+00	9.228+00	1.078+01	1.151+01	1.154+01	1.103+01	1.015+01	9.048+00
0.000	4.685+00	8.860+00	1.220+01	1.456+01	1.594+01	1.643+01	1.617+01	1.531+01	1.402+01
0.000	4.207+00	8.002+00	1.112+01	1.344+01	1.497+01	1.578+01	1.594+01	1.553+01	1.465+01
0.000	3.277+00	6.221+00	8.621+00	1.043+01	1.168+01	1.246+01	1.281+01	1.279+01	1.239+01
0.000	2.358+00	4.424+00	6.032+00	7.184+00	7.970+00	8.489+00	8.804+00	8.936+00	8.872+00
0.000	1.590+00	2.904+00	3.811+00	4.355+00	4.665+00	4.860+00	5.012+00	5.139+00	5.222+00
0.000	9.863-01	1.699+00	2.036+00	2.081+00	1.987+00	1.885+00	1.847+00	1.892+00	2.003+00
0.000	5.220-01	7.646-01	6.536-01	3.085-01	-9.880-02	-4.387-01	-6.436-01	-6.961-01	-6.125-01
0.000	1.658-01	3.935-02	-4.234-01	-1.068+00	-1.712+00	-2.230+00	-2.562+00	-2.697+00	-2.650+00
0.000	-1.140-01	-5.374-01	-1.283+00	-2.165+00	-2.987+00	-3.632+00	-4.050+00	-4.237+00	-4.211+00
0.000	-3.475-01	-1.024+00	-2.013+00	-3.090+00	-4.048+00	-4.776+00	-5.237+00	-5.439+00	-5.408+00
0.000	-5.651-01	-1.481+00	-2.695+00	-3.941+00	-4.998+00	-5.766+00	-6.226+00	-6.404+00	-6.334+00
0.000	-8.020-01	-1.969+00	-3.408+00	-4.801+00	-5.919+00	-6.679+00	-7.091+00	-7.201+00	-7.058+00
0.000	-1.102+00	-2.560+00	-4.230+00	-5.745+00	-6.871+00	-7.562+00	-7.867+00	-7.859+00	-7.606+00
0.000	-1.530+00	-3.344+00	-5.246+00	-6.830+00	-7.881+00	-8.414+00	-8.536+00	-8.357+00	-7.958+00

0.000	-2.185+00	-4.440+00	-6.538+00	-8.079+00	-8.908+00	-9.153+00	-8.999+00	-8.591+00	-8.022+00
0.000	-3.206+00	-5.985+00	-8.130+00	-9.379+00	-9.740+00	-9.522+00	-8.999+00	-8.334+00	-7.604+00
0.000	-4.646+00	-7.928+00	-9.696+00	-1.016+01	-9.720+00	-8.910+00	-8.023+00	-7.172+00	-6.380+00
0.000	-5.473+00	-8.655+00	-9.192+00	-8.315+00	-7.138+00	-6.069+00	-5.192+00	-4.482+00	-3.695+00
0.000	-5.481+00	-8.340+00	-7.990+00	-6.327+00	-4.900+00	-3.886+00	-3.178+00	-2.665+00	-2.274+00

4.556+00	3.805+00	3.108+00	2.475+00	1.907+00	1.400+00	9.501-01	5.592-01	2.446-01	4.462-02	0.000
7.853+00	6.649+00	5.492+00	4.412+00	3.426+00	2.536+00	1.743+00	1.056+00	5.005-01	1.270-01	0.000
1.245+01	1.075+01	9.026+00	7.352+00	5.774+00	4.320+00	3.008+00	1.863+00	9.245-01	2.656-01	0.000
1.338+01	1.185+01	1.018+01	8.445+00	6.738+00	5.109+00	3.602+00	2.261+00	1.144+00	3.386-01	0.000
1.166+01	1.062+01	9.355+00	7.940+00	6.455+00	4.968+00	3.543+00	2.243+00	1.141+00	3.377-01	0.000
8.589+00	8.063+00	7.308+00	6.362+00	5.282+00	4.130+00	2.976+00	1.893+00	9.596-01	2.799-01	0.000
5.218+00	5.077+00	4.765+00	4.279+00	3.641+00	2.898+00	2.108+00	1.340+00	6.710-01	1.897-01	0.000
2.135+00	2.233+00	2.242+00	2.129+00	1.890+00	1.545+00	1.134+00	7.143-01	3.459-01	9.071-02	0.000
-4.305-01	-2.012-01	1.850-02	1.819-01	2.642-01	2.646-01	2.037-01	1.157-01	3.857-02	2.172-04	0.000
-2.455+00	-2.154+00	-1.800+00	-1.442+00	-1.113+00	-8.323-01	-5.973-01	-3.768-01	-2.202-01	-7.333-02	0.000
-4.005+00	-3.657+00	-3.212+00	-2.714+00	-2.202+00	-1.703+00	-1.232+00	-7.994-01	-4.195-01	-1.281-01	0.000
-5.177+00	-4.782+00	-4.264+00	-3.661+00	-3.011+00	-2.348+00	-1.699+00	-1.091+00	-5.603-01	-1.652-01	0.000
-6.056+00	-5.605+00	-5.017+00	-4.328+00	-3.574+00	-2.791+00	-2.014+00	-1.283+00	-6.500-01	-1.875-01	0.000
-6.706+00	-6.184+00	-5.526+00	-4.763+00	-3.930+00	-3.063+00	-2.201+00	-1.393+00	-6.984-01	-1.986-01	0.000
-7.158+00	-6.553+00	-5.824+00	-5.000+00	-4.109+00	-3.189+00	-2.280+00	-1.434+00	-7.135-01	-2.011-01	0.000
-7.395+00	-6.704+00	-5.912+00	-5.043+00	-4.121+00	-3.180+00	-2.261+00	-1.413+00	-6.989-01	-1.961-01	0.000
-7.341+00	-6.578+00	-5.747+00	-4.863+00	-3.947+00	-3.025+00	-2.135+00	-1.325+00	-6.515-01	-1.823-01	0.000
-6.840+00	-6.049+00	-5.231+00	-4.389+00	-3.534+00	-2.687+00	-1.879+00	-1.154+00	-5.609-01	-1.556-01	0.000
-5.636+00	-4.920+00	-4.212+00	-3.504+00	-2.797+00	-2.105+00	-1.453+00	-8.756-01	-4.140-01	-1.101-01	0.000
-3.387+00	-2.925+00	-2.484+00	-2.052+00	-1.625+00	-1.209+00	-8.192-01	-4.776-01	-2.113-01	-4.741-02	0.000
-1.954+00	-1.675+00	-1.414+00	-1.162+00	-9.142-01	-6.727-01	-4.461-01	-2.484-01	-9.757-02	-1.279-02	-0.000

THE FIELD OF VORTICITY IS GIVEN BY

0.000	-1.317+02	-2.424+02	-3.195+02	-3.602+02	-3.690+02	-3.542+02	-3.241+02	-2.862+02	-2.461+02
0.000	-3.328+01	-6.650+01	-9.812+01	-1.253+02	-1.456+02	-1.573+02	-1.605+02	-1.560+02	-1.452+02
0.000	2.655+01	4.260+01	4.669+01	4.030+01	2.656+01	9.044+00	-8.976+00	-2.491+01	-3.705+01
0.000	5.488+01	9.521+01	1.190+02	1.275+02	1.241+02	1.120+02	9.438+01	7.385+01	5.271+01
0.000	6.662+01	1.166+02	1.473+02	1.614+02	1.630+02	1.561+02	1.431+02	1.259+02	1.058+02
0.000	7.218+01	1.254+02	1.564+02	1.694+02	1.705+02	1.647+02	1.546+02	1.415+02	1.257+02
0.000	7.621+01	1.306+02	1.590+02	1.676+02	1.649+02	1.573+02	1.477+02	1.370+02	1.249+02
0.000	8.037+01	1.354+02	1.601+02	1.632+02	1.555+02	1.445+02	1.338+02	1.240+02	1.143+02
0.000	8.521+01	1.407+02	1.614+02	1.586+02	1.456+02	1.312+02	1.188+02	1.089+02	1.003+02
0.000	9.095+01	1.469+02	1.631+02	1.544+02	1.367+02	1.192+02	1.051+02	9.453+01	8.635+01
0.000	9.775+01	1.540+02	1.654+02	1.511+02	1.290+02	1.087+02	9.298+01	8.170+01	7.350+01
0.000	1.056+02	1.621+02	1.684+02	1.484+02	1.222+02	9.934+01	8.226+01	7.032+01	6.196+01
0.000	1.147+02	1.713+02	1.719+02	1.460+02	1.157+02	9.060+01	7.239+01	5.995+01	5.148+01
0.000	1.251+02	1.815+02	1.755+02	1.431+02	1.089+02	8.177+01	6.272+01	5.002+01	4.157+01
0.000	1.367+02	1.923+02	1.784+02	1.390+02	1.008+02	7.202+01	5.246+01	3.977+01	3.159+01
0.000	1.491+02	2.031+02	1.797+02	1.323+02	9.019+01	6.011+01	4.047+01	2.819+01	2.062+01
0.000	1.609+02	2.121+02	1.771+02	1.205+02	7.466+01	4.395+01	2.498+01	1.378+01	7.386+00
0.000	1.675+02	2.143+02	1.651+02	9.787+01	4.919+01	1.957+01	2.940+00	-5.785+00	-9.856+00
0.000	1.549+02	1.930+02	1.273+02	4.909+01	2.509+00	-2.058+01	-3.057+01	-3.370+01	-3.330+01
0.000	7.025+01	7.845+01	2.553+00	-6.830+01	-8.877+01	-8.831+01	-8.140+01	-7.313+01	-6.494+01
0.000	-3.290+02	-4.941+02	-4.553+02	-3.398+02	-2.493+02	-1.895+02	-1.504+02	-1.236+02	-1.040+02

-2.074+02	-1.714+02	-1.388+02	-1.098+02	-8.405+01	-6.129+01	-4.114+01	-2.362+01	-9.562+00	-1.030+00	0.000
-1.301+02	-1.129+02	-9.507+01	-7.757+01	-6.100+01	-4.580+01	-3.231+01	-2.095+01	-1.222+01	-6.267+00	-1.190+00
-4.369+01	-4.563+01	-4.390+01	-3.956+01	-3.364+01	-2.713+01	-2.097+01	-1.603+01	-1.292+01	-1.156+01	-1.029+01
3.336+01	1.785+01	6.541+00	-8.497-01	-5.090+00	-7.243+00	-8.534+00	-1.021+01	-1.331+01	-1.836+01	-2.492+01
8.411+01	6.347+01	4.546+01	3.076+01	1.928+01	1.028+01	2.632+00	-5.061+00	-1.422+01	-2.602+01	-4.117+01

1.0[illegible]3+02	8.768+01	6.860+01	5.127+01	3.612+01	2.290+01	1.074+01	-1.671+00	-1.588+01	-3.345+01	-5.572+01
1.109+02	9.482+01	7.783+01	6.103+01	4.506+01	2.997+01	1.518+01	-3.676-01	-1.813+01	-3.967+01	-6.644+01
1.037+02	9.133+01	7.744+01	6.265+01	4.752+01	3.220+01	1.634+01	-8.553-01	-2.055+01	-4.403+01	-7.247+01
9.193+01	8.251+01	7.150+01	5.906+01	4.549+01	3.090+01	1.509+01	-2.487+00	-2.263+01	-4.621+01	-7.400+01
7.912+01	7.160+01	6.287+01	5.261+01	4.080+01	2.745+01	1.245+01	-4.518+00	-2.391+01	-4.619+01	-7.181+01
6.684+01	6.046+01	5.333+01	4.484+01	3.472+01	2.287+01	9.203+00	-6.432+00	-2.420+01	-4.427+01	-6.681+01
5.559+01	4.992+01	4.390+01	3.680+01	2.821+01	1.794+01	5.912+00	-7.913+00	-2.351+01	-4.082+01	-5.989+01
4.531+01	4.015+01	3.498+01	2.903+01	2.183+01	1.317+01	2.941+00	-8.802+00	-2.193+01	-3.627+01	-5.180+01
3.565+01	3.100+01	2.659+01	2.169+01	1.583+01	8.767+00	4.327-01	-9.099+00	-1.965+01	-3.099+01	-4.309+01
2.609+01	2.206+01	1.848+01	1.465+01	1.014+01	4.725+00	-1.665+00	-8.940+00	-1.690+01	-2.533+01	-3.418+01
1.587+01	1.270+01	1.013+01	7.530+00	4.495+00	8.133-01	-3.569+00	-8.567+00	-1.399+01	-1.961+01	-2.537+01
3.900+00	2.010+00	8.130-01	-2.665-01	-1.587+00	-3.346+00	-5.600+00	-8.285+00	-1.122+01	-1.418+01	-1.696+01
-1.121+01	-1.114+01	-1.041+01	-9.487+00	-8.690+00	-8.200+00	-8.102+00	-8.379+00	-8.896+00	-9.375+00	-9.384+00
-3.112+01	-2.803+01	-2.453+01	-2.092+01	-1.740+01	-1.415+01	-1.131+01	-8.982+00	-7.151+00	-5.551+00	-3.404+00
-5.713+01	-4.963+01	-4.232+01	-3.513+01	-2.810+01	-2.138+01	-1.520+01	-9.881+00	-5.730+00	-2.771+00	-9.400-02
-8.858+01	-7.547+01	-6.347+01	-5.195+01	-4.064+01	-2.964+01	-1.931+01	-1.032+01	-3.580+00	-7.507-02	0.000

THE STREAM FUNCTION FIELO IS GIVEN 8Y

0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
0.000	6.390-03	2.367-02	4.725-02	7.197-02	9.362-02	1.097-01	1.191-01	1.223-01	1.201-01
0.000	1.820-02	6.813-02	1.384-01	2.156-01	2.877-01	3.461-01	3.860-01	4.060-01	4.072-01
0.000	2.982-02	1.123-01	2.303-01	3.632-01	4.922-01	6.026-01	6.850-01	7.347-01	7.510-01
0.000	3.924-02	1.481-01	3.052-01	4.844-01	6.620-01	8.195-01	9.438-01	1.027+00	1.066+00
0.000	4.620-02	1.745-01	3.596-01	5.718-01	7.842-01	9.763-01	1.133+00	1.246+00	1.309+00
0.000	5.103-02	1.924-01	3.956-01	6.281-01	8.613-01	1.074+00	1.252+00	1.385+00	1.466+00
0.000	5.415-02	2.035-01	4.168-01	6.589-01	9.008-01	1.122+00	1.309+00	1.452+00	1.544+00
0.000	5.596-02	2.094-01	4.262-01	6.697-01	9.110-01	1.131+00	1.317+00	1.460+00	1.556+00
0.000	5.676-02	2.112-01	4.266-01	6.650-01	8.984-01	1.109+00	1.286+00	1.424+00	1.516+00
0.000	5.679-02	2.098-01	4.198-01	6.483-01	8.682-01	1.064+00	1.227+00	1.352+00	1.437+00
0.000	5.619-02	2.058-01	4.073-01	6.217-01	8.237-01	1.000+00	1.145+00	1.254+00	1.327+00
0.000	5.505-02	1.995-01	3.896-01	5.865-01	7.669-01	9.205-01	1.044+00	1.135+00	1.193+00
0.000	5.337-02	1.910-01	3.669-01	5.429-01	6.987-01	8.270-01	9.267-01	9.982-01	1.042+00
0.000	5.104-02	1.798-01	3.385-01	4.905-01	6.190-01	7.202-01	7.955-01	8.469-01	8.756-01
0.000	4.786-02	1.654-01	3.034-01	4.280-01	5.270-01	6.002-01	6.514-01	6.838-01	6.992-01
0.000	4.339-02	1.464-01	2.598-01	3.539-01	4.219-01	4.677-01	4.967-01	5.126-01	5.175-01
0.000	3.693-02	1.210-01	2.054-01	2.664-01	3.043-01	3.256-01	3.364-01	3.402-01	3.383-01
0.000	2.736-02	8.655-02	1.379-01	1.663-01	1.784-01	1.821-01	1.817-01	1.793-01	1.753-01
0.000	1.370-02	4.170-02	5.992-02	6.327-02	6.125-02	5.829-02	5.561-02	5.330-02	5.116-02
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1.139-01	1.046-01	9.325-02	8.045-02	6.675-02	5.250-02	3.800-02	2.377-02	1.101-02	2.120-03	0.000
3.927-01	3.657-01	3.295-01	2.868-01	2.398-01	1.902-01	1.395-01	8.978-02	4.504-02	1.206-02	0.000
7.364-01	6.959-01	6.348-01	5.583-01	4.709-01	3.765-01	2.787-01	1.821-01	9.422-02	2.735-02	0.000
1.062+00	1.018+00	9.400-01	8.357-01	7.115-01	5.733-01	4.276-01	2.820-01	1.480-01	4.423-02	0.000
1.319+00	1.280+00	1.196+00	1.074+00	9.227-01	7.491-01	5.621-01	3.728-01	1.969-01	5.944-02	0.000
1.491+00	1.461+00	1.379+00	1.249+00	1.081+00	8.831-01	6.657-01	4.429-01	2.344-01	7.082-02	0.000
1.580+00	1.559+00	1.482+00	1.353+00	1.178+00	9.664-01	7.308-01	4.867-01	2.573-01	7.745-02	0.000
1.598+00	1.584+00	1.513+00	1.388+00	1.213+00	9.990-01	7.565-01	5.036-01	2.655-01	7.944-02	0.000
1.559+00	1.548+00	1.483+00	1.364+00	1.196+00	9.863-01	7.471-01	4.965-01	2.608-01	7.748-02	0.000
1.475+00	1.465+00	1.405+00	1.294+00	1.136+00	9.365-01	7.087-01	4.699-01	2.457-01	7.247-02	0.000
1.358+00	1.347+00	1.290+00	1.188+00	1.042+00	8.585-01	6.485-01	4.285-01	2.230-01	6.531-02	0.000
1.216+00	1.202+00	1.149+00	1.056+00	9.247-01	7.604-01	5.728-01	3.771-01	1.953-01	5.678-02	0.000
1.056+00	1.039+00	9.894-01	9.066-01	7.918-01	6.492-01	4.873-01	3.195-01	1.645-01	4.749-02	0.000
8.811-01	8.623-01	8.177-01	7.464-01	6.496-01	5.307-01	3.967-01	2.588-01	1.324-01	3.791-02	0.000
6.977-01	6.784-01	6.400-01	5.816-01	5.041-01	4.100-01	3.049-01	1.976-01	1.003-01	2.839-02	0.000

AGE: 37

5.114-01	4.936-01	4.629-01	4.187-01	3.611-01	2.922-01	2.159-01	1.387-01	6.949-02	1.928-02	0.000
3.307-01	3.166-01	2.952-01	2.655-01	2.278-01	1.831-01	1.341-01	8.498-02	4.165-02	1.107-02	0.000
1.694-01	1.609-01	1.490-01	1.334-01	1.137-01	9.067-02	6.553-02	4.059-02	1.901-02	4.504-03	0.000
4.855-02	4.605-02	4.243-02	3.778-02	3.200-02	2.523-02	1.785-02	1.056-02	4.391-03	6.076-04	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

STREAM FUNCTION ITERATIONS = 1

CARDS PUNCHED

RESULTS ON TAPE IN BLOCK 9***** 1.000 .100 100000. .700 20 20 .25004057-00 .28714466-03*****

AT A TIME TAU = 2.5004057-01 NEXT DTAU = 2.8714466-04

2-PT APPROX NUSSELT HS = 12.810950 QHS/K*A*DT = .628856 NUSSELT CW = .050083 QCW/K*A*DT = .626905

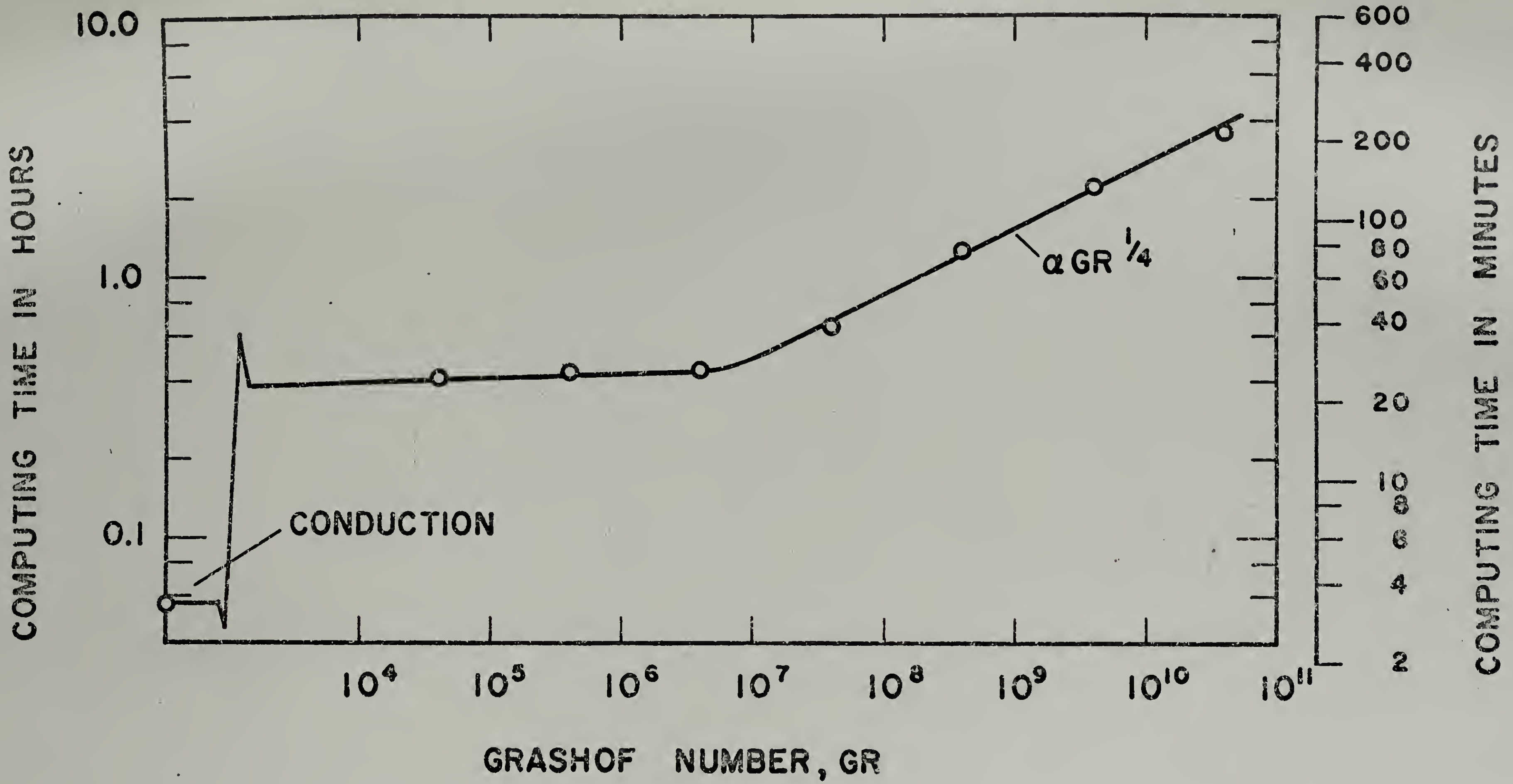

FIGURE 5

Computing time required for EXCON1 at several Grashof numbers (GR), numerical study of reference 3. Calculations carried from quiescent initial conditions to steady state with a $M \times N = 50 \times 20$ grid.