DEPARTMENT OF PHYSICS AND GEOPHYSICAL SCIENCES
SCHOOL OF SCIENCES
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA

(NASA-CR-143286) COMPUTER SIMULATION OF PLASMA AND N-BODY PROBLEMS Annual Report, Jun. 1974 - May 1975 (Old Dominion Univ. Research Foundation) 222 p HC $7.25 CSCL 20I G3/75
N75-29894
Unclas 32355

COMPUTER SIMULATION OF PLASMA AND N-BODY PROBLEMS

By

Wynford L. Harries

and

John B. Miller

Annual Report

Prepared for the
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia

Under
Grant NSG 1040

August 1975

TABLE OF CONTENTS

INTRODUCTION

Research during the period June 1974 through May 1975 has resulted in the following Fortran language computer codes: (1) more efficient two- and three-dimensional central force potential solvers; (2) a three-dimensional simulator of an isolated galaxy which incorporates the aforementioned potential solver; (3) a two-dimensional particle-in-cell simulator of the Jeans instability in an infinite self-gravitating compressible gas; and (4) a two-dimensional particle-in-cell simulator of a rotating self-gravitating compressible gaseous system of which rectangular coordinate and superior polar coordinate versions were written.

POTENTIAL SOLVERS

The two- and three-dimensional potential solvers listed in Appendices A and B respectively decrease computing cost as compared to the previous versions by 50 and 75 percent respectively. Their methods of operation were detailed in the Semi-Annual Status Report for the period June 1974 through November 1974. The three-dimensional potential solver (pages B-4 through B-17 of this annual report) is virtually self-explanatory due to extensive use of comment cards.

THREE-DIMENSIONAL ISOLATED GALAXY SIMULATOR

Through multiple overlays the three-dimensional potential solver has been incorporated into an efficient n-body dynamic code, which provides for plotting of various views of the particle distribution in position and velocity space as well as printed diagnostics. A set of balanced initial conditions have been tested for a thin disk of particles. In creating the initial conditions, the particles are first randomly distributed according to an axi-symmetric radial density distribution which varies as $[1 - (r/r_0)^2]^{1/2}$, where r and r_0 are the radius and maximum radius respectively, and according to a Gaussian axial density distribution. After solving for the potential, the particles are assigned a Maxwellian velocity distribution after determining the component standard deviations. That of the axial velocity is determined by balancing the axial components of the gradients of the potential and the pressure. The standard deviation of the radial and azimuthal velocity distributions are determined by dividing the galaxy into layers of constant axial position and by applying to each layer the Toomre* stability criterion for infinitesimally thin disks. An average azimuthal velocity is calculated for each radius and height by reducing the square of the angular velocity of the balanced cold disk by terms involving (a) the radial pressure gradient, (b) the difference in the squares of the standard deviations of the radial and azimuthal velocities, and (c) the axial derivative of the expectation value of the product of the radial and axial velocities.

This code is listed in Appendix B; position and velocity space plots of a set of initial conditions and the first few cycles of a run are presented in Appendix C.

Longer runs on larger meshes (up to 64 x 64 x 16) with initial conditions computed with this method and probably other methods will be made in the near future.

*Toomre, Alar: On the Gravitational Stability of a Disk of Stars. Astrophys. J., vol. 139, no. 4, May 15, 1964, pp. 1217-1238.

TWO-DIMENSIONAL GASEOUS JEANS

INSTABILITY SIMULATOR

The particle-in-cell method attempts to combine the advantages of the Lagrangian and Eulerian approaches to non-steady compressible gas simulation. The basic method as described in The Particle-in-Cell Method for the Calculation of the Dynamics of Compressible Fluids by A. A. Amsden, Los Alamos Scientific Laboratory Report LA-3466 of February 1966, was modified to include self-gravitation and periodic boundary conditions. Self-gravitation was implemented by adding gravitational terms to the cell calculation of new velocity and specific internal energy. Periodic boundary conditions were implemented by (a) requiring that particles leaving one boundary carry their mass, energy, and momenta into the opposite boundary and (b) finite differencing pressure and potential near the boundary in such a way that a value required from a cell just outside the boundary is obtained from a cell just inside the opposite boundary.

This code is listed in Appendix D and plots from two full length runs (150 cycles) made on a small mesh (32 x 32 cells) are presented in Appendix E. These runs verify the analytic prediction that an increase in initial temperature decreases both the number of instabilities per unit area and their rate of evolution. That the code conserves total energy is evident from the plots of Appendix E while printed diagnostics (not shown) verify conservation of linear momentum.

When point masses of a collisionless gravitational n-body code were given an initial position distribution identical to that given the gas "particles" in this particle-in-cell fluid code and an initial velocity dispersion corresponding to the initial specific internal energy of the cells in this fluid code, an almost identical pattern of instabilities resulted. The dynamics of the instabilities occuring in these two codes will be compared in the near future. Also planned is the combination of these two codes for use in studying stellar-gaseous gravitational two-stream instabilities.

POLAR COORDINATE SIMULATOR FOR A ROTATING GAS

The status report for June 1974 - November 1974 described the modification of a two-dimensional particle-in-cell code with square cells and linear dynamics to a code with square cells but with angular and radial momentum transfer during the particle movement phase. Although this modification conserved angular momentum and greatly reduced artificial heating, extensive testing demonstrated that the combination of square cells and polar dynamics produced a large assymetric radial outward acceleration. A "quick fix" of rotating the meshes (+) or (-) 45° on alternate cycles served only to symmetrize the artificial radial acceleration.

Recently a code was written which combines a rectangular mesh potential solver with polar meshes for the cell values of radial and angular momentum, specific internal energy, mass, x and y coordinates of the cell center of mass, and the average radius squared for the center and end of each time step. At the center of mass of each polar cell a combination finite differencing and bilinear interpolation of the rectangular potential mesh yields the x and y components of gravitational field which are then resolved into radial and gravitational components. The particle movement phase depends on the principle that for a collection of particles moving with a uniform angular velocity, e.g., a cell, both the rotational kinetic energy and the angular momentum are proportional to the average of the radius squared (radius of gyration squared) times a power of the angular velocity. During the particle movement phase, the new mass density is built up simultaneously in the polar mass density mesh and in the rectangular mesh which was used to store the potential during the cell calculations. The possibility of high random cell velocities near the center of the gas was provided for by making the radial and azimuthal dimensions of all cells roughly equal to one unit; this was accomplished by increasing the number of cells per mass ring from four for the first ring to 128 for the 31st. ring. Since the basic particle-in-cell method is a leap frog method, it is critically important that all positions, forces and accelerations be computed at the center of the time step and that all velocities, momenta, kinetic energies and the associated radii of gyration be computed at the end of the time step; a verification of this code is that identical results were produced by alternate procedures for leap frogging velocity and radius of gyration one half time step ahead of particle position.

The above polar coordinate code is listed in Appendix F. Appendix G presents plots produced by runs of the polar and earlier rectangular codes, describes their identical initial conditions, and demonstrates the superiority of the polar code by a brief analysis. The polar code suffers from a very small radial outward acceleration and this is being investigated with a view toward elimination.

To conserve computer resources, the system state at the end of a run may be stored on magnetic tape and the run may later be continued if desired. If the time scaled velocity of any cell exceeds the dimension of that cell, the run is terminated with a complete set of plots and a long printout of cell quantities and a continuation tape is generated for possible later use with a smaller time step.

Plans for the use of this code include investigation of the evolution of (a) a purely gaseous system (listing of Appendix F with one parameter change (b) a gaseous system acting under the influence of an analytically computed constant stellar central force component (Appendices F and G) and (c) a gaseous system interacting through the gravitational potential with an existing collisionless stellar n-body code. The code may be modified to allow star formation and/or permit internal energy loss by electromagnetic radiation.

APPENDIX A

Listing of the Two-Dimensional Potential Solver of Increased Efficiency.

This subroutine decreases cost by 50% by replacing central memory storage with disk file storage. To minimize file buffer size and peripheral process time it utilizes the input/output routine MEMDISK and associated assembly language subroutine MDFUNC which were written by R. Bulle of the Analysis and Computations Division of the Langley Research Center.

```
      SUBROUTINE GETPHI
      COMMON Z(1025),Y(1025),RHO(128,128),I2A,ITEST,F(8385)
      DIMENSION RHO1(128,64),RHO2(128,64),HIN21(128),HJN21(128)
      DIMENSION IPAR(5)
      EQUIVALENCE (RHO(1,1),RHO1(1,1)),(RHO(1,65),RHO2(1,1))
      IF(ITEST.EQ.0) GO TO 70
      ITEST=0
      I2B=I2A-1
      N=2**I2A
      NO2=N/2
      N21=NO2+1
      NO4=N/4
      N34=NO2+NO4
      NN24=NO2*NO4
      RNI=1./(N*N)
      DO 5 J=1,NO2
      DO 5 I=1,NO2
      IF(I.EQ.1.AND.J.EQ.1) GO TO 5
      R2=(I-1)*(I-1)+(J-1)*(J-1)
      R=SQRT(R2)
      RHO(I,J)=RNI/R
    5 CONTINUE
      RHO(1,1)=RHO(1,2)
      NO2SQ=NO2*NO2
      DO 10 J=1,NO2
      R2=NO2SQ+(J-1)*(J-1)
      R=SQRT(R2)
      HIN21(J)=RNI/R
   10 HJN21(J)=HIN21(J)
      R2=NO2SQ+NO2SQ
      R=SQRT(R2)
      HIJN21=RNI/R
      CALL GETSET(2,I2B)
      DO 20 J=1,NO2
      DO 15 I=1,NO2
   15 Z(I)=RHO(I,J)
      Z(N21)=HIN21(J)
      CALL FTRANS(2,I2B)
```

```
      HIN21(J)=Y(N21)
      DO 20 I=1,NO2
   20 RHO(I,J)=Y(I)
      DO 25 I=1,NO2
   25 Z(I)=HJN21(I)
      Z(N21)=HIJN21
      CALL FTRANS(2,I2B)
      HIJN21=Y(N21)
      DO 30 I=1,NO2
   30 HJN21(I)=Y(I)
      DO 40 I=1,NO2
      DO 35 J=1,NO2
   35 Z(J)=RHO(I,J)
      Z(N21)=HJN21(I)
      CALL FTRANS(2,I2B)
      HJN21(I)=Y(N21)
      DO 40 J=1,NO2
   40 RHO(I,J)=Y(J)
      DO 45 J=1,NO2
   45 Z(J)=HIN21(J)
      Z(N21)=HIJN21
      CALL FTRANS(2,I2B)
      HIJN21=Y(N21)
      DO 50 J=1,NO2
   50 HIN21(J)=Y(J)
      DO 55 I=1,NO2
      DO 55 J=1,I
      M=I+(J-1)*N21-(J-1)*J/2
   55 F(M)=RHO(I,J)
      DO 60 J=1,NO2
      M=N21+(J-1)*N21-(J-1)*J/2
   60 F(M)=HIN21(J)
      NFM=N21+NO2*N21-NO2*N21/2
      F(NFM)=HIJN21
      IPAR(1)=1
      IPAR(2)=0
      IPAR(3)=NFM
      IPAR(4)=5LTAPE5
      CALL MEMDISK(F(1),IPAR)
```

```
      REWIND  5
      GO TO 200
   70 IPAR(1)=0
      IPAR(3)=NFM
      IPAR(4)=5LTAPE5
      CALL MEMDISK(F(1),IPAR)
      REWIND 5
   72 IPAR(1)=1
      IPAR(3)=NN24
      IPAR(4)=5LTAPE3
      CALL MEMDISK(RHO2(1,1),IPAR)
      REWIND 3
      CALL GETSET(3,I2A)
      DO 80 J=1,NO4
      DO 75 I=1,NO2
      Z(I)=RHO1(I,J)
   75 Z(NO2+I)=0.
      CALL FTRANS(3,I2A)
      DO 80 I=1,NO2
      RHO1(I,J)=Y(I)
   80 RHO2(I,J)=Y(NO2+I)
      IPAR(1)=1
      IPAR(4)=5LTAPE1
      CALL MEMDISK(RHO1(1,1),IPAR)
      REWIND 1
      IPAR(4)=5LTAPE2
      CALL MEMDISK(RHO2(1,1),IPAR)
      REWIND 2
      IPAR(1)=0
      IPAR(4)=5LTAPE3
      CALL MEMDISK(RHO1(1,1),IPAR)
      REWIND 3
      DO 90 J=1,NO4
      DO 85 I=1,NO2
      Z(I)=RHO1(I,J)
   85 Z(NO2+I)=0.
      CALL FTRANS(3,I2A)
```

```
      DO 90 I=1,NO2
      RHO1(I,J)=Y(I)
   90 RHO2(I,J)=Y(NO2+I)
      IPAR(1)=1
      IPAR(4)=5LTAPE4
      CALL MEMDISK(RHO2(1,1),IPAR)
      REWIND 4
      IPAR(1)=0
      IPAR(4)=5LTAPE1
      CALL MEMDISK(RHO2(1,1),IPAR)
      REWIND 1
      DO 115 I=1,NO2
      DO  95 J=1,NO4
      Z(J)=RHO2(I,J)
      Z(NO4+J)=RHO1(I,J)
      Z(NO2+J)=0.
   95 Z(N34+J)=0.
      CALL GETSET(3,I2A)
      CALL FTRANS(3,I2A)
      DO 100 J=2,NO2
      IF(I.LT.J)GO TO 96
      M=I+(J-1)*N21-(J-1)*J/2
      GO TO 97
   96 M=J+(I-1)*N21-(I-1)*I/2
   97 Z(J)=Y(J)*F(M)
  100 Z(NO2+J)=Y(NO2+J)*F(M)
      Z(1)=Y(1)*F(I)
      M=N21+(I-1)*N21-(I-1)*I/2
      Z(N21)=Y(N21)*F(M)
      CALL GETSET(4,I2A)
      CALL FTRANS(4,I2A)
      DO 115 J=1,NO4
      RHO1(I,J)=Y(J)
  115 RHO2(I,J)=Y(NO4+J)
      IPAR(1)=1
      IPAR(4)=5LTAPE1
      CALL MEMDISK(RHO1(1,1),IPAR)
```

```
      REWIND 1
      IPAR(4)=5LTAPE3
      CALL MEMDISK(RHO2(1,1),IPAR)
      REWIND 3
      IPAR(1)=0
      IPAR(4)=5LTAPE2
      CALL MEMDISK(RHO1(1,1),IPAR)
      REWIND 2
      IPAR(4)=5LTAPE4
      CALL MEMDISK(RHO2(1,1),IPAR)
      REWIND 4
      DO 130 I=1,NO2
      DO 120 J=1,NO4
      Z(J)=RHO1(I,J)
      Z(NO4+J)=RHO2(I,J)
      Z(NO2+J)=0.
  120 Z(N34+J)=0.
      CALL GETSET(3,I2A)
      CALL FTRANS(3,I2A)
      IF(I.EQ.1)GO TO 125
      DO 123 J=2,NO2
      IF(I.LT.J)GO TO 121
      M=I+(J-1)*N21-(J-1)*J/2
      GO TO 122
  121 M=J+(I-1)*N21-(I-1)*I/2
  122 Z(J)=Y(J)*F(M)
  123 Z(NO2+J)=Y(NO2+J)*F(M)
      Z(1)=Y(1)*F(I)
      M=N21+(I-1)*N21-(I-1)*I/2
      Z(N21)=Y(N21)*F(M)
      GO TO 128
  125 DO 127 J=2,NO2
      M=N21+(J-1)*N21-(J-1)*J/2
      Z(J)=Y(J)*F(M)
  127 Z(NO2+J)=Y(NO2+J)*F(M)
      Z(1)=Y(1)*F(N21)
      Z(N21)=Y(N21)*F(NFM)
```

```
128 CALL GETSET(4,I2A)
    CALL FTRANS(4,I2A)
    DO 130 J=1,NO4
    RHO1(I,J)=Y(J)
130 RHO2(I,J)=Y(NO4+J)
    IPAR(1)=1
    IPAR(4)=5LTAPE2
    CALL MEMDISK(RHO1(1,1),IPAR)
    REWIND 2
    IPAR(1)=0
    IPAR(4)=5LTAPE3
    CALL MEMDISK(RHO1(1,1),IPAR)
    REWIND 3
    DO 140 J=1,NO4
    DO 135 I=1,NO2
    Z(I)=RHO1(I,J)
135 Z(NO2+I)=RHO2(I,J)
    CALL FTRANS(4,I2A)
    DO 140 I=1,NO2
140 RHO1(I,J)=Y(I)
    IPAR(1)=1
    IPAR(4)=5LTAPE3
    CALL MEMDISK(RHO1(1,1),IPAR)
    REWIND 3
    IPAR(1)=0
    IPAR(4)=5LTAPE1
    CALL MEMDISK(RHO1(1,1),IPAR)
    REWIND 1
    IPAR(4)=5LTAPE2
    CALL MEMDISK(RHO2(1,1),IPAR)
    REWIND 2
    DO 150 J=1,NO4
    DO 145 I=1,NO2
    Z(I)=RHO1(I,J)
145 Z(NO2+I)=RHO2(I,J)
    CALL FTRANS(4,I2A)
```

```
      DO 150 I=1,NO2
  150 RHO1(I,J)=Y(I)
      IPAR(1)=0
      IPAR(4)=5LTAPE3
      CALL MEMDISK(RHO2(1,1),IPAR)
      REWIND 3
  200 RETURN
      END
```

```
      SUBROUTINE MEMDISK(IBUFF,IPARAMS)
      DIMENSION JPAR(5)
      EQUIVALENCE
     1(JPAR(1),IFUNCT ),
     2(JPAR(2),IMODE  ),
     3(JPAR(3),LENGTH ),
     4(JPAR(4),IAGTN  ),
     5(JPAR(5),ISTATUS)
      DIMENSION IPARAMS(5)
      DIMENSION IBUFF(130)
C *** ************************
C      IBUFF   - BUFFER
C      IPARAMS - PARAMETER LIST
C       1 IFUNCT    = FUNCTION   0=READ   ,1=WRITE
C       2 IMODE     = MODE       0=BINARY,1=CODED
C       3 LENGTH    = LENGTH OF MESSAGE
C       5 ISTATUS   = ERROR STATUS
C            0        NO ERROR
C            1        INVALID MODE
C            2        INVALID FUNCTION
C *** ************************
      IFUNCT =IPARAMS(1)
      IMODE  =IPARAMS(2)
      LENGTH =IPARAMS(3)
      IAGTN  =IPARAMS(4)
      ISTATUS=0
      IF(IMODE)44,45,43
   43 IF(IMODE-1)45,45,44
   44 ISTATUS=1
      GO TO 999
   45 CONTINUE
      IF(IFUNCT)9,7,8
    7 IFUN=10B
      IF(IMODE.EQ.0)IFUN=12B
      GO TO 11
    8 IF(IFUNCT-1)9,10,9
    9 ISTATUS=2
      GO TO 999
   10 IFUN=34B
      IF(IMODE.EQ.0)IFUN=36B
   11 CONTINUE
```

7-5

```
CALL MDSETF(IBUFF,JPAR)
CALL MDFUNC(IFUN.L,ISTAT)
CONTINUE
IPARAMS(5)=ISTATUS
RETURN
END
```

```
         IDENT   MDFUNC
         ENTRY   MDFUNC
         ENTRY   MDSETF
RAPLUS1  MACRO
+        SA1     RA+1
         NZ      X1,*
         ENDM
RA       EQU     0
FN       BSS     0
FET      DATA    0
FST      DATA    0
IN       DATA    0
OUT      DATA    0
LIM      DATA    0
CIOCALL  VFD     18/3LCIO,1/0,1/1,22/0,18/FET
         VFD     42/0HMDFUNC,18/3
MDFUNC   DATA    0
*
*     CALL MDFUNC(IFUNCT,L,STAT)
*
         SA1     B1                .X1=IFUNCT
         MX0     42                .X0=42BITS
         SA2     FET               .X2=FET
         BX3     -X0*X1            .X3=18R(IFUNCT)
         BX4     X0*X2             .X4=42L(FET)
         SB7     1                 .B7=1
         BX6     X4+X3             .X6=42L(FET),18R(IFUNCT)
         SA6     A2                .FFT=FET+IFUNCT
         SA1     CIOCALL           .X1=CIO
         BX6     X1                .X6=CIO
         RAPLUS1                   .ISRA+1 CLEAR
         SA6     B7                .CALL CIO
         XJ                        .EXCHANGE JUMP TO MONITOR
+        SA1     A2                .X1=FET
         SA2     IN                .X2=IN
         BX6     -X0*X1            .X6=CODE AND STATUS
         SA3     A2+B7             .X3=OUT
         SA6     B3                .STAT=CODE AND STATUS
         IX7     X2-X3             .X7=IN-OUT
         PL      X7,FFT1           .IF(IN.GE.OUT)GO TO FFT1
```

```
         SA1     A2-B7          .X1=FIRST
         SA4     A3+B7          .X4=LIMIT
         IX7     X7-X1          .X7=IN-FIRST-OUT
         IX7     X7+X4          .X7=IN-FIRST+LIMIT-OUT
FFT1     SA7     B2             .L=IN-FIRST+LIMIT-OUT
         EQ      MDFUNC
         VFD     42/0HMDSETF,18/2
MDSETF   DATA    0
*
*     CALL MDSETF(IBUFF,JPARAMS)
*
         SB6     3             .B6=3
         SB4     1             .B4=1
         SA1     B2+B6         .X1=AGT NAME
         SB5     B4+B4         .B5=2
         SB3     FET           .B3=FET
         BX6     X1            .X6=AGT NAME
         SB7     B4+B6         .B7=4
         SA6     B3            .FET=AGT NAME
         SA1     B2            .X1=IFUNCT
         SX6     B1            .X6=IBUFF
         ZR      X1,READ       .IF(IFUNCT.EQ.0)GO TO READ
WRITE    SA5     B2+B5         .X5=LENGTH
         IX7     X6+X5         .X7=IBUFF+LENGTH
         SA6     B3+B6         .OUT=IBUFF
         SA7     B3+B5         .IN =IBUFF+LENGTH
         EQ      BOTH          .GO TO BOTH
READ     SA6     B3+B6         .OUT=IBUFF
         SA6     B3+B5         .IN =IBUFF
BOTH     SA5     B2+B5          .X5=LENGTH
         SX4     B4             .X4=1
         IX7     X6+X5          .X7=IBUFF+LENGTH
         SA6     B3+B4          .1ST=IBUFF
         IX7     X7+X4          .X7=IBUFF+LENGTH+1
         SA7     B3+B7          .LIM=IBUFF+LENGTH+1
         EQ      MDSETF         .RETURN
         END
```

APPENDIX B

Listing of the Three-Dimensional Isolated Galaxy Dynamic Simulator. The initial conditions are for a thin balanced disk the velocity dispersion of which satisfies the Toomre* local stability criterion.

Overlay programs GETH and GETPHI constitute an improved three-dimensional potential solver which decreases cost by 75% by replacing central memory storage with disk file storage.

To make the listing easier to understand, use of the input/output routine listed in Appendix A has been replaced in this listing by the less efficient Fortran binary READ and WRITE statements and their accompanying larger disk file buffers.

*Toomre, Alar: On the Gravitational Stability of a Disk of Stars. Astrophys. J., vol. 139, no. 4, May 15, 1964, pp. 1217-1238.

```
      OVERLAY(IFILE,0,0)
      PROGRAM STARS3D(OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,TAPE5,TAPE6,TAPE7,
     1    TAPE8,TAPE9,TAPE10,TAPE11,TAPE12)
      COMMON/ALLCOM/N,NO2,N21,NO4,N41,N34,NH,NHO2,NH21,NCH,NRHO,NHH,
     1    I2A,I2B,I3A,I3B,NHO4
      COMMON/HN21COM/HN21(33,9)
      COMMON/INITCOM/MASSD(16,4)
      COMMON/ADVCOM/NBR,NBS,RI,XM,DT,DTE2,NO4M1,NO4M2,VXYMAX,VZMAXI,
     1    CY,CYY,NPLOT,NPRINT,IN(2),XMIN,XMAX,YMIN,YMAX,ZMIN,
     2    ZMAX,RMIN,RMAX,VRMIN,VRMAX,VTMIN,VTMAX,VZMIN,VZMAX,XPP(3),
     3    YP,ZP,YPP(3),RPP(3),VTP,VRP,VZP,ITAPX,PI,MASK1,S2,JT,JS,NMK,
     4    NBS3,CXY,CZ,VZAV1,DR,ITEST,JTFILE,JSFILE,DDD,NO4P1
      INTEGER CY,CYY
      REAL MASSD
C NAME OVERLAY FILES
C     NAME THE FILE FOR THE GETH AND INITIAL OVERLAYS
      IFILE=5LIFILE
C     NAME GETPHI OVERLAY FILE
      GPFILE=6LGPFILE
C     NAME STARS OVERLAY FILE
      SFILE=5LSFILE
C  SET DIMENSIONS
      I2A=6
      I3A=4
      I2B=I2A-1
      I3B=I3A-1
      N=2**I2A
      NO2=N/2
      N21=NO2+1
      NO4=N/4
      NO4P1=NO4+1
      N41=NO4+1
      N34=NO2+NO4
      NH=2**I3A
      NHO2=NH/2
      NH21=NHO2+1
      NHO4=NH/4
      NCH=NO2*NO2*NHO2
      NRHO=NO4*NO4*NHO2
      NHH=NO4*NO4*NH21
C CALL GETH OVERLAY IN ORDER TO COMPUTE THE TRANSFORMED GREENS FUNCTION AND
C STORE CHUNKS OF IT ON DISK FILE 9 IN THE ORDER APPROPRIATE FOR LATER USE IN
```

```
C THE GETPHI OVERLAY
      CALL OVERLAY(IFILE,1,0,6HRECALL)
      ITEST=0
C CALL THE INITIAL OVERLAY IN ORDER TO SET INITIAL VALUES, GENERATE INITIAL STAR
C POSITIONS, AND COMPUTE INITIAL DENSITY.
      CALL OVERLAY(IFILE,3,0,6HRECALL)
C CALL GETPHI OVERLAY IN ORDER TO COMPUTE THE INITIAL POTENTIAL FROM THE INITIAL
C DENSITY.
      CALL OVERLAY(GPFILE,2,0,6HRECALL)
      ITEST=1
      CYY=0
C CALL INITIAL OVERLAY IN ORDER TO ASSIGN INITIAL VELOCITIES TO STARS, HALF
C TIME STEP THE INITIAL STAR POSITIONS, COMPUTE A NEW INITIAL DENSITY, AND MAKE
C AN INITIAL SET OF PLOTS.
      CALL OVERLAY(IFILE,3,0,6HRECALL)
      CYY=1
10    PRINT 12, CYY
12    FORMAT(7H CYCLE=,I8)
      CALL SECOND(T1)
C CALL GETPHI OVERLAY IN ORDER TO COMPUTE POTENTIAL
      CALL OVERLAY(GPFILE,2,0,6HRECALL)
      CALL SECOND (T2)
      TGETPHI=T2-T1
      PRINT 27, TGETPHI
27    FORMAT(12H FIELD TIME=, E16.8)
C CALL STARS OVERLAY IN ORDER TO COMPUTE NEW STAR POSITIONS AND VELOCITIES AND
C A NEW DENSITY MESH.
      CALL OVERLAY(SFILE,4,0,6HRECALL)
      CALL SECOND(T3)
      TSTARS=T3-T2
      PRINT 42, TSTARS
42    FORMAT(19H STAR ADVANCE TIME=,E16.8)
C IF IT IS NOT THE LAST CYCLE (TIME STEP) CY +NCREMENT THE CYCLE NUMBER CYY AND
C GO TO STATEMENT 10.
      IF(CYY.GE.CY) GO TO 50
45    CYY = CYY + 1
      GO TO 10
50    STOP
      END
```

```
      OVERLAY(IFILE,1,0)
      PROGRAM GETH
C THIS OVERLAY PERFORMS A COSINE ANALYSIS OF THE THREE-DIMENSIONAL GREENS
C FUNCTION ARRAY.  IT THEN WRITES CHUNKS OF THIS ARRAY ON DISK FILE 9 IN THE
C ORDER IN WHICH THEY WILL BE READ INTO THE HH ARRAY DURING THE GETPHI
C OVERLAY. VALUES FOR I=N/2+1 AND J=N/2+1 ARE TRANSFERRED TO THE HN21 ARRAY
C WHICH IS IN COMMON WITH THE GETPHI OVERLAY.
      COMMON/ALLCOM/N,NO2,N21,NO4,N41,N34,NH,NHO2,NH21,NCH,NRHO,NHH,
     1    I2A,I2B,I3A,I3B,NHO4
      COMMON/HN21COM/HN21(33,9)
      COMMON Z(1025), Y(1025)
      DIMENSION H(33,33,9)
      RNI=1./(N*N*NH)
      DO 1 K=1,NH21
      DO 1 J=1,N21
      DO 1 I=1,N21
      RI=(K-1)*(K-1)+(J-1)*(J-1)+(I-1)*(I-1)
      IF(RI.LT.1.) RI=1.
      H(I,J,K)=RNI/SQRT(RI)
    1 CONTINUE
      CALL GETSET(2,I2B)
      DO 2 K=1,NH21
      DO 2 J=1,N21
      DO 3 I=1,N21
    3 Z(I)=H(I,J,K)
      CALL FTRANS(2,I2B)
      DO 4 I=1,N21
    4 H(I,J,K)=Y(I)
    2 CONTINUE
      DO 5 K=1,NH21
      DO 5 I=1,N21
      DO 6 J=1,N21
    6 Z(J)=H(I,J,K)
      CALL FTRANS(2,I2B)
      DO 7 J=1,N21
    7 H(I,J,K)=Y(J)
    5 CONTINUE
      CALL GETSET(2,I3B)
      DO 10 J=1,N21
      DO 10 I=1,N21
      DO 8 K=1,NH21
```

```
 8 Z(K)=H(I,J,K)
   CALL FTRANS(2,I3B)
   DO 9 K=1,NH21
 9 H(I,J,K)=Y(K)
10 CONTINUE
   NO4P1=N41
   WRITE(9) (((H(I,J,K),I=1,NO4),J=1,NO4),K=1,NH21)
   WRITE(9) (((H(I,J,K),I=1,NO4),J=NO4P1,NO2),K=1,NH21)
   WRITE(9) (((H(I,J,K),I=1,NO4),J=1,NO4),K=1,NH21)
   WRITE(9) (((H(I,J,K),I=NO4P1,NO2),J=1,NO4),K=1,NH21)
   WRITE(9) (((H(I,J,K),I=NO4P1,NO2),J=NO4P1,NO2),K=1,NH21)
   WRITE(9) (((H(I,J,K),I=NO4P1,NO2),J=1,NO4),K=1,NH21)
   REWIND 9
   DO 15 K=1,NH21
   DO 15 I=1,N21
15 HN21(I,K)=H(I,N21,K)
   RETURN
   END
```

```
      OVERLAY(GPFILE,2,0)
      PROGRAM GETPHI
C THIS OVERLAY SOLVES FOR THE POTENTIAL MESH (DIMENSIONED N/2 BY N/2 BY NH/2)
C DUE TO A DENSITY MESH (DIMENSIONED N/2 BY N/2 BY NH/2) BY DOING A PERIODIC
C ANALYSIS OF THE DENSITY AND THEN A PERIODIC SYNTHESIS OF THE PRODUCT OF THE
C TRANSFORMED GREENS FUNCTION (DIMENSIONED (N/2+1) BY (N/2+1) BY (NH/2+1)) AND
C THE TRANSFORMED DENSITY.  FORMALLY SPEAKING, EACH OF THE TRANSFORMS EXCEPT
C THE COSINE ANALYSIS OF THE GREENS FUNCTION, WHICH IS PERFORMED IN THE GETH
C OVERLAY) REQUIRES AN ARRAY DIMENSIONED N BY N BY NH/2.  TO REDUCE CORE
C STORAGE THIS OVERLAY PERFORMS THESE TRANSFORMS IN CHUNKS BY THE ALIGNMENT
C OF FOUR SMALLER ARRAYS NAMED RHO1,RHO2,RHO3, AND RHO4, EACH OF WHICH IS
C DIMENSIONED N/4 BY N/4 BY NH/2.  THE CHUNKS OF THE LARGER ARRAY NOT IN CORE
C AT ANY ONE TIME ARE STORED ON DISK FILES 1 THROUGH 8.  THE FOLLOWING ARE TWO
C TOP VIEWS OF THE LARGER ARRAY BOTH OF WHICH DESIGNATE THE CHUNKS AS IROW AND
C JCOLUMN.  IROW 1 AND 2 OF JCOLUMN 1 AND 2 CONSTITUTE THE ACTIVE MESH.  IN
C THE DIAGRAM ON THE LEFT THE NUMBERS WITHIN THE CHUNKS OF JCOLUMN 1 AND 2
C INDICATE THE DISK FILES ON WHICH THOSE CHUNKS ARE STORED.  (NO DISK STORAGE
C IS REQUIRED FOR JCOLUMN 3 OR 4.)  REFERRING TO THE DIAGRAM ON THE RIGHT,  THE
C NUMBERS WITHIN THE CHUNKS ARE THE ORDER IN WHICH CHUNKS OF THE TRANSFORMED
C DENSITY ARE MULTIPLIED (ELEMENT BY ELEMENT) BY THE APPROPRIATE PORTION OF THE
C TRANSFORMED GREENS FUNCTION WHICH HAS BEEN READ FROM DISK FILE 9 INTO ARRAY
C HH(N/4,N/4,NH/2+1).  (AN EXCEPTION IS THE SET OF TRANSFORMED GREENS FUNCTION
C BOUNDARY VALUES FOR I=N/2+1 AND J=N/2+1 WHICH REMAIN AT ALL TIMES IN COMMON
C IN THE ARRAY HN21(N/2+1,NH/2+1).)  A PLUS IN A CHUNK INDICATES THAT NEW VALUES
C MUST BE READ INTO ARRAY HH BEFORE THAT CHUNK IS MULTIPLIED BY HH.  THIS SYSTEM
C MINIMIZES PERIPHERAL PROCESS TIME BY UTILIZING THE PERIODICITY OF THE
C TRANSFORMED GREENS FUNCTION.
```

```
C
C
C
C     TWO TOP VIEWS OF LARGER MESH (N BY N BY NH/2) - IROW 1 AND 2 OF JCOLUMN
C         1 AND 2 CONSTITUTE THE ACTIVE MESH (N/2 BY N/2 BY NH/2).  THE
C         DIRECTIONS ARE X AND I - DOWN ON PAGE, Y AND J - TO RIGHT ON PAGE,
C         Z AND K - OUT OF PAGE.
C
C
C                   JCOLUMN                                         JCOLUMN
C                 1   2   3   4                                   1   2   3   4
C
C               *****************                               *****************
C               *   *   *   *   *                               * + * + *   *   *
C       IROW=1  * 1 * 5 *   *   *                       IROW=1  * 1 * 3 * 2 * 4 *
C               *****************                               *****************
C               *   *   *   *   *                               * + * + *   *   *
C       IROW=2  * 2 * 6 *   *   *                       IROW=2  * 9 *11 *10 *12 *
C               *****************                               *****************
C               *   *   *   *   *                               * + *   *   *   *
C       IROW=3  * 3 * 7 *   *   *                       IROW=3  * 7 * 5 * 8 * 6 *
C               *****************                               *****************
C               *   *   *   *   *                               * + *   *   *   *
C       IROW=4  * 4 * 8 *   *   *                       IROW=4  *15 *13 *16 *14 *
C               *****************                               *****************
C
C       DISK FILES ON WHICH CHUNKS                          ORDER IN WHICH CHUNKS ARE
C       ARE STORED                                          MULTIPLIED BY APPROPRIATE
C                                                           PORTION OF TRANSFORMED GREENS
C                                                           FUNCTION
C
C
C
      COMMON/ALLCOM/N,NO2,N21,NO4,N41,N34,NH,NHO2,NH21,NCH,NRHO,NHH,
     1     I2A,I2B,I3A,I3B,NHO4
      COMMON/TRANCOM/RHO1(16,16,8),RHO2(16,16,8),RHO3(16,16,8),
     1     RHO4(16,16,8),HH(16,16,9)
      COMMON/HN21COM/HN21(33,9)
C THE INITIAL CONDITIONS OVERLAY (PROGRAM INITIAL) OR PARTICLE ADVANCING
C OVERLAY (PROGRAM STARS) STORES THE DENSITY CHUNKS OF IROW 1 AND 2 FOR
```

```
C JCOLUMN=1 ON DISK FILES 1 AND 2 RESPECTIVELY AND FOR JCOLUMN=2 ON DISK FILES
C 5 AND 6 RESPECTIVLEY.  THE GETPHI OVERLAY REPLACES THE DENSITY ON THESE DISK
C FILES WITH THE CORRESPONDING VALUES OF POTENTIAL WHICH ARE THEN USED IN THE
C PARTICLE ADVANCING OVERLAY.  THIS IS ACCOMPLISHED THROUGH CALLING SUBROUTINES
C ANLX(JCOLUMN), ANLSYN(IROW), AND SYNX(JCOLUMN) AS DETAILED BELOW.
C
C
C SUBROUTINE ANLX(JCOLUMN) READS RESPECTIVELY IROW 1 AND 2 FROM THE FOLLOWING
C DISK FILES - 1 AND 2 FOR JCOLUMN=1 AND 5 AND 6 FOR JCOLUMN=2.  IT THEN
C PERFORMS A PERIODIC ANALYSIS IN THE X DIRECTION OVER JCOLUMN FOR I=1,N AND
C WRITES THE RESULTS RESPECTIVELY FOR IROW 1,2,3, AND 4 ON THE FOLLOWING DISK
C FILES - 1,2,3, AND 4 FOR JCOLUMN=1 AND 5,6,7, AND 8 FOR JCOLUMN=2.
      CALL ANLX(1)
      CALL ANLX(2)
C SUBROUTINE ANLSYN(IROW) READS RESPECTIVLEY JCOLUMN 1 AND 2 FROM THE FOLLOWING
C DISK FILES - 1 AND 5 FOR IROW=1,2 AND 6 FOR IROW=2,3 AND 7 FOR IROW=3, AND
C 4 AND 8 FOR IROW=4.  IT THEN PERFORMS A PERIODIC ANALYSIS IN THE Y DIRECTION
C OVER IROW FOR J=1,N.   FOR EACH CHUNK IT THEN PERFORMS A PERIODIC ANALYSIS IN
C THE Z DIRECTION FOR K=1,NH, ELEMENT BY ELEMENT MULTIPLICATION WITH A
C SIMILARLY SHAPED CHUNK OF THE TRANSFORMED GREENS FUNCTION AND THEN A PERIODIC
C SYNTHESIS IN THE Z DIRECTION FOR K=1,NH.  THE RESULT FOR K=1,NH/2 IS THEN
C PERIODICALLY SYNTHESIZED IN THE Y DIRECTION OVER IROW FOR J=1,N.  THIS LAST
C RESULT FOR JCOLUMN 1 AND 2 IS WRITTEN RESPECTIVLEY ON THE FOLLOWING DISK
C FILES - 1 AND 5 FOR IROW=1, 2 AND 6 FOR IROW=2, 3 AND 7 FOR IROW=3, AND 4 AND
C 8 FOR IROW=4.  THE ORDER IN WHICH ANLSYN IS CALLED FOR IROW 1 THROUGH 4
C MINIMIZES READING FROM DISK FILE 9 OF CHUNKS OF THE TRANSFORMED GREENS
C FUNCTION AS MENTIONED ABOVE.
      CALL ANLSYN(1)
      CALL ANLSYN(3)
      CALL ANLSYN(2)
      CALL ANLSYN(4)
C SUBROUTINE SYNX(JCOLUMN) READS RESPECTIVELY IROW 1,2,3, AND 4 FROM THE
C FOLLOWING DISK FILES - 1,2,3, AND 4 FOR JCOLUMN=1 AND 5,6,7, AND 8 FOR
C JCOLUMN=2. IT THEN PERFORMS A PERIODIC SYNTHESIS IN THE X DIRECTION OVER
C JCOLUMN FOR J=1,N.  IT THEN WRITES THE RESULT RESPECTIVELY FOR IROW 1 AND 2
C ON THE FOLLOWING DISK FILES - 1 AND 2 FOR JCOLUMN=1 AND 5 AND 6 FOR JCOLUMN=2.
      CALL SYNX(1)
      CALL SYNX(2)
      RETURN
      END
```

```
      SUBROUTINE ANLX(JCOLUMN)
      COMMON/ALLCOM/N,N02,N21,N04,N41,N34,NH,NH02,NH21,NCH,NRHO,NHH,
     1    I2A,I2B,I3A,I3B,NH04
      COMMON/TRANCOM/RH01(16,16,8),RH02(16,16,8),RH03(16,16,8),
     1    RH04(16,16,8),HH(16,16,9)
      COMMON Z(1025), Y(1025)
      IF(JCOLUMN.EQ.2) GO TO 2
      READ(1) RH01
      REWIND 1
      READ(2) RH02
      REWIND 2
      GO TO 3
    2 READ(5) RH01
      REWIND 5
      READ(6) RH02
      REWIND 6
    3 CONTINUE
      CALL GETSET(3,I2A)
      DO 10 K=1,NH02
      DO 10 J=1,N04
      DO  5 I=1,N04
      Z(I)=RH01(I,J,K)
      Z(N04+I)=RH02(I,J,K)
      Z(N02+I)=0.
    5 Z(N34+I)=0.
      CALL FTRANS(3,I2A)
      DO 10 I=1,N04
      RH01(I,J,K)=Y(I)
      RH02(I,J,K)=Y(N04+I)
      RH03(I,J,K)=Y(N02+I)
   10 RH04(I,J,K)=Y(N34+I)
      IF(JCOLUMN.EQ.2) GO TO 12
      WRITE(1) RH01
      REWIND 1
      WRITE(2) RH02
      REWIND 2
      WRITE(3) RH03
      REWIND 3
      WRITE(4) RH04
      REWIND 4
```

```
      RETURN
   12 WRITE(5) RHO1
      REWIND 5
      WRITE(6) RHO2
      REWIND 6
      WRITE(7) RHO3
      REWIND 7
      WRITE(8) RHO4
      REWIND 8
      RETURN
      END
```

```
      SUBROUTINE ANLSYN(IROW)
      COMMON/ALLCOM/N,NO2,N21,NO4,N41,N34,NH,NHO2,NH21,NCH,NRHO,NHH,
     1     I2A,I2B,I3A,I3B,NHO4
      COMMON/TRANCOM/RHO1(16,16,8),RHO2(16,16,8),RHO3(16,16,8),
     1     RHO4(16,16,8),HH(16,16,9)
      COMMON/HN21COM/HN21(33,9)
      COMMON Z(1025), Y(1025)
      GO TO (1,2,3,4) IROW
    1 READ(1) RHO1
      REWIND 1
      READ(5) RHO2
      REWIND 5
      GO TO 5
    2 READ(2) RHO1
      REWIND 2
      READ(6) RHO2
      REWIND 6
      GO TO 5
    3 READ(3) RHO1
      REWIND 3
      READ(7) RHO2
      REWIND 7
      GO TO 5
    4 READ(4) RHO1
      REWIND 4
      READ(8) RHO2
      REWIND 8
    5 CALL GETSET(3,I2A)
      DO 10 K=1,NHO2
      DO 10 I=1,NO4
      DO  7 J=1,NO4
      Z(J)=RHO1(I,J,K)
      Z(NO4+J)=RHO2(I,J,K)
      Z(NO2+J)=0.
    7 Z(N34+J)=0.
      CALL FTRANS(3,I2A)
      DO 10 J=1,NO4
      RHO1(I,J,K)=Y(J)
      RHO2(I,J,K)=Y(NO4+J)
      RHO3(I,J,K)=Y(NO2+J)
```

```
 10 RHO4(I,J,K)=Y(N34+J)
    GO TO(49,49,75,75) IROW
 49 READ(9) HH
 50 JCOLUMN=1
    DO 70 I=1,NO4
    DO 70 J=1,NO4
    DO 52 K=1,NHO2
    Z(K)=RHO1(I,J,K)
 52 Z(NHO2+K)=0.
    CALL GETSET(3,I3A)
    CALL FTRANS(3,I3A)
    IF(IROW.NE.3) GO TO 300
    IF(I.NE.1)GO TO 300
    LL=J
    GO TO 200
 54 DO 70 K=1,NHO2
 70 RHO1(I,J,K)=Y(K)
    GO TO 100
 74 READ(9) HH
 75 JCOLUMN=2
    DO 95 I=1,NO4
    DO 95 J=1,NO4
    DO 77 K=1,NHO2
    Z(K)=RHO2(I,J,K)
 77 Z(NHO2+K)=0.
    CALL GETSET(3,I3A)
    CALL FTRANS(3,I3A)
    IF(IROW.NE.3) GO TO 300
    IF(I.NE.1)GO TO 300
    LL=NO4+J
    GO TO 200
 79 DO 95 K=1,NHO2
 95 RHO2(I,J,K)=Y(K)
    GO TO 125
100 JCOLUMN=3
    DO 120 I=1,NO4
    DO 120 J=1,NO4
    DO 101 K=1,NHO2
    Z(K)=RHO3(I,J,K)
101 Z(NHO2+K)=0.
```

```
      CALL GETSET(3,I3A)
      CALL FTRANS(3,I3A)
      GO TO(103,105,107,115) IROW
  103 IF(J.NE.1)GO TO 300
      LL=I
      GO TO 200
  105 IF(J.NE.1)GO TO 300
      LL=N04+I
      GO TO 200
  107 IF(I.NE.1.AND.J.NE.1)GO TO 300
      IF(I.EQ.1.AND.J.EQ.1)GO TO 111
      IF(I.EQ.1)GO TO 109
      LL=I
      GO TO 200
  109 LL=J
      GO TO 200
  111 LL=N21
      GO TO 200
  115 IF(J.NE.1) GO TO 300
      LL=N04+I
      GO TO 200
  117 DO 120 K=1,NH02
  120 RH03(I,J,K)=Y(K)
      GO TO (74,74,400,390) IROW
  125 JCOLUMN=4
      DO 145 I=1,N04
      DO 145 J=1,N04
      DO 127 K=1,NH02
      Z(K)=RH04(I,J,K)
  127 Z(NH02+K)=0.
      CALL GETSET(3,I3A)
      CALL FTRANS(3,I3A)
      IF(IROW.NE.3) GO TO 300
      IF(I.NE.1)GO TO 300
      LL=N04+J
      GO TO 200
  129 DO 145 K=1,NH02
  145 RH04(I,J,K)=Y(K)
      GO TO (400,400,49,49) IROW
  200 DO 205 K=2,NH02
```

```
      Z(K)=Y(K)*HN21(LL,K)
  205 Z(NH02+K)=Y(NH02+K)*HN21(LL,K)
      Z(1)=Y(1)*HN21(LL,1)
      Z(NH21)=Y(NH21)*HN21(LL,NH21)
      GO TO 310
  300 DO 305 K=2,NH02
      Z(K)=Y(K)*HH(I,J,K)
  305 Z(NH02+K)=Y(NH02+K)*HH(I,J,K)
      Z(1)=Y(1)*HH(I,J,1)
      Z(NH21)=Y(NH21)*HH(I,J,NH21)
  310 CALL GETSET(4,I3A)
      CALL FTRANS(4,I3A)
      GO TO (54,79,117,129) JCOLUMN
  390 REWIND 9
  400 CALL GETSET(4,I2A)
      DO 410 K=1,NH02
      DO 410 I=1,N04
      DO 405 J=1,N04
      Z(J)=RH01(I,J,K)
      Z(N04+J)=RH02(I,J,K)
      Z(N02+J)=RH03(I,J,K)
  405 Z(N34+J)=RH04(I,J,K)
      CALL FTRANS(4,I2A)
      DO 410 J=1,N04
      RH01(I,J,K)=Y(J)
  410 RH02(I,J,K)=Y(N04+J)
      GO TO (415,420,425,430) IROW
  415 WRITE(1) RH01
      REWIND 1
      WRITE(5) RH02
      REWIND 5
      RETURN
  420 WRITE(2) RH01
      REWIND 2
      WRITE(6) RH02
      REWIND 6
      RETURN
  425 WRITE(3) RH01
      REWIND 3
      WRITE(7) RH02
```

```
      REWIND 7
      RETURN
  430 WRITE(4) RHO1
      REWIND 4
      WRITE(8) RHO2
      REWIND 8
      RETURN
      END
```

```
      SUBROUTINE SYNX(JCOLUMN)
      COMMON/ALLCOM/N,NO2,N21,NO4,N41,N34,NH,NHO2,NH21,NCH,NRHO,NHH,
     1     I2A,I2B,I3A,I3B,NHO4
      COMMON/TRANCOM/RHO1(16,16,8),RHO2(16,16,8),RHO3(16,16,8),
     1     RHO4(16,16,8),HH(16,16,9)
      COMMON Z(1025),Y(1025)
      IF(JCOLUMN.EQ.2) GO TO 1
      READ(1) RHO1
      REWIND 1
      READ(2) RHO2
      REWIND 2
      READ(3) RHO3
      REWIND 3
      READ(4) RHO4
      REWIND 4
      GO TO 4
    1 READ(5) RHO1
      REWIND 5
      READ(6) RHO2
      REWIND 6
      READ(7) RHO3
      REWIND 7
      READ(8) RHO4
      REWIND 8
    4 CALL GETSET(4,I2A)
      DO 10 K=1,NHO2
      DO 10 J=1,NO4
      DO  5 I=1,NO4
      Z(I)=RHO1(I,J,K)
      Z(NO4+I)=RHO2(I,J,K)
      Z(NO2+I)=RHO3(I,J,K)
    5 Z(N34+I)=RHO4(I,J,K)
      CALL FTRANS(4,I2A)
      DO 10 I=1,NO4
      RHO1(I,J,K)=Y(I)
   10 RHO2(I,J,K)=Y(NO4+I)
      IF(JCOLUMN.EQ.2) GO TO 12
      WRITE(1) RHO1
      REWIND 1
      WRITE(2) RHO2
```

```
      REWIND 2
      RETURN
12    WRITE(5) RHO1
      REWIND 5
      WRITE(6) RHO2
      REWIND 6
      RETURN
      END
```

```
      OVERLAY(IFILE,3,0)
      PROGRAM INITIAL
C THIS OVERLAY SPECIFIES INITIAL CONDITIONS, GENERATES INITIAL STAR POSITIONS
C AND VELOCITIES AND CREATES THE INITIAL DENSITY MESH.
      COMMON/ALLCOM/N,NO2,N21,NO4,N41,N34,NH,NHO2,NH21,NCH,NRHO,NHH,
     1    I2A,I2B,I3A,I3B,NHO4
      COMMON/ADVCOM/NBR,NBS,RI,XM,DT,DTE2,NO4M1,NO4M2,VXYMAX,VZMAXI,
     1    CY,CYY,NPLOT,NPRINT,IN(2),XMIN,XMAX,YMIN,YMAX,ZMIN,
     2    ZMAX,RMIN,RMAX,VRMIN,VRMAX,VTMIN,VTMAX,VZMIN,VZMAX,XPP(3),
     3    YP,ZP,YPP(3),RPP(3),VTP,VRP,VZP,ITAPX,PI,MASK1,S2,JT,JS,NMK,
     4    NBS3,CXY,CZ,VZAV1,DR,ITEST,JTFILE,JSFILE,DDD,NO4P1
      COMMON/INITCOM/MASSD(16,4)
      DIMENSION PHI(32,32,8)
      DIMENSION XS(6),RS(6)
      DIMENSION XPACK(2048),YPACK(2048),ZPACK(2048)
      DIMENSION RPACK(2048),TPACK(2048)
      DIMENSION E(16,4),SIGMAVZ(16,4),VTR(16,4),VRR(16,4),TH(16,4),
     1     VRVZ(16,4),SIGMAVR(16,4)
      EQUIVALENCE (XS(1),X),(XS(2),VX),(XS(3),Y),(XS(4),VY),(XS(5),Z),
     1    (XS(6),VZ)
      EQUIVALENCE(RS(1),R),(RS(2),VR),(RS(3),THETA),(RS(4),VT),
     1    (RS(5),ZR),(RS(6),VZR)
      EQUIVALENCE (XPACK(1),RPACK(1)),(YPACK(1),TPACK(1))
      INTEGER Q,CY,CYY
      REAL MASSD
      CALL PSEUDO
C IF THIS IS THE SECOND CALL TO THIS OVERLAY GO TO 70.
      IF(ITEST.EQ.1) GO TO 70
C SET INITIAL VALUES
C     SET TOTAL NUMBER OF STARS
      NBR=20480
C     SET NUMBER OF STARS PER READ OF STAR FILE
      NBS=2048
C     SET INITIAL RADIUS(MUST BE .LE. N/4-3)
      RI=12.
C     SET INITIAL STANDARD DEVIATION OF Z AT ZERO RADIUS
      SIGMAZ=.6
C     SET INITIAL MAXIMUM DEVIATION OF Z AT ZERO RADIUS(MUST BE .LE. NH/4-1)
      ZMAXI=1.2
C     SET NUMBER OF TIME STEPS PER ROTATION
```

```
      DDD=50
C     SET SCALED MASS
      XM=3.55E6/NBR
C     SET NUMBER OF POINTS PER PLOT
      NP=NBR
C     SET INITIAL MAXIMUM ABSOLUTE VALUE OF VX AND VY
      VXYMAX =3.
C     SET INITIAL MAXIMUM ABSOLUTE VALUE OF VZ
      VZMAXI=.5
C     SET TOTAL NUMBER OF TIME STEPS
      CY=8
C     SET NUMBER OF TIME STEPS BETWEEN PLOTS(MUST BE.GE.6)
      NPLOT=6
C     SET NUMBER OF TIME STEPS BETWEEN PRINTED DIAGNOSTICS
      NPRINT=2
C SET PLOTTING SPECIFICATIONS
C     SET PLOT TITLE
      IN(1)=10HJ. MILLER
      IN(2)=10H3D-TEST
C     SET MAXIMUM AND MINIMUM VALUES TO BE PLOTTED
      XMIN=-8.
      XMAX=40.
      YMIN=-8.
      YMAX=40.
      ZMIN=-8.
      ZMAX=40.
      RMIN=0.
      RMAX=24.
      VRMIN=-10.
      VRMAX=10.
      VTMIN=-10.
      VTMAX=10.
      VZMIN=-10.
      VZMAX=10.
C     SET COORDINATE LABELS
      XPP(1)=10HX,
      XPP(2)=10HCYCLE=
      YP=10H    Y
      ZP=10H    Z
      YPP(1)=10HY,
```

```
      YPP(2)=10HCYCLE=
      RPP(1)=10HR.
      RPP(2)=10HCYCLE=
      VTP=10H   VT 'ETA
      VRP=10H   VR
      VZP=10H   VZ
C     SET TAPE NUMBER FOR DDIPLT
      ITAPX=6LTAPE22
C SET CONSTANTS
      PI=3.1415926536
      MASK1=07777777770000000000
      S2=SQRT(2.)
C     INITIALIZE STAR TAPE NUMBERS
      JT=10
      JS=11
      REWIND 10
      REWIND 11
      NMK=NP-NBS
      NBS3=3*NBS
      CXY=NO4
      CZ=NHO4+.5
      NO4M1=NO4-1
      NO4M2=NO4-2
      DR=XM/2
      DR=DR.AND.MASK1
      RI2=RI*RI
C END OF CONSTANTS
C SET PHI TO ZERO FOR BUILD UP OF X-Y-Z MASS DENSITY
      DO 25 K=1,NHO2
      DO 25 J=1,NO2
      DO 25 I=1,NO2
   25 PHI(I,J,K)=0.
C SET MASSD TO ZERO FOR BUILD UP OF AN AXISYMMETRIC (R-Z) MASS DENSITY
C WHICH WILL BE USED LATER IN INITIALIZING PARTICLE VELOCITIES.
      DO 30 IZ=1,NHO4
      DO 30 IR=1,NO4M1
   30 MASSD(IR,IZ)=0.
C GENERATE RANDOM STAR POSITIONS WITH A RADIAL DENSITY DISTRIBUTION OF
C SQRT(1-(RADIUS/MAXRADIUS)**2) AND A MAXWELLIAN Z DENSITY DISTRIBUTION. BUILD
C UP AN X-Y-Z MASS DENSITY IN PHI AND AN R-Z MASS DENSITY IN MASSD.  TEMPORARILY
```

```
C ASSIGN ZEROES AS THE STELLAR VELOCITY COMPONENTS.  PACK THE POSITION AND
C VELOCITY COMPONENTS INTO EACH WORD OF XPACK, YPACK, AND ZPACK AND WRITE
C THOSE ARRAYS ON DISK FILE 10.
      NS2=0
      X=URAN(72737.)
   35 DO 60 IS=1,NBS
   40 X=2.*(URAN(0.)-.5)
      Y=2.*(URAN(0.)-.5)
      Z7=2.*(URAN(0.)-.5)
      RR=X*X+Y*Y+Z7*Z7
      IF(RR.GT.1.) GO TO 40
      X=RI*X
      Y=RI*Y
      R2=X*X+Y*Y
      R=SQRT(R2)
      IF(IS.EQ.NBS) R=(NS2+NBS)*RI/NBR
      IF(R.LT.1.E-5) X=0.0001
      THETA=ATAN2(-Y,X)
      X=R*COS(THETA)+CXY
      Y=-R*SIN(THETA)+CXY
      ZR=SQRT(1-R*R/RI2)
      ZZ=S2*ZR*SIGMAZ*SQRT(-ALOG(1.-URAN(0.)))
      Z=ZZ*COS(2.*PI*URAN(0.))
      ZM=ZR*ZMAXI
      IF(ABS(Z).GT.ZM) Z=SIGN(ZM,Z)
      Z=Z+CZ
      VX=0.
      VY=0.
      VZ=0.
      IR=R+1.5
      IZ=ABS(Z-CZ)+1.5
C     BUILD UP R-Z DENSITY IN MASSD (DIVISION BY AREA OF MASS RINGS WILL BE
C     PERFORMED LATER).
      MASSD(IR,IZ)=MASSD(IR,IZ)+DR
      IX=X+.5
      JY=Y+.5
      KZ=Z+.5
C     BUILD UP X-Y-Z DENSITY IN LAST HALF OF EACH WORD OF PHI.
      CALL DENS(DR,PHI(IX,JY,KZ))
      CALL PACK(XS(1),XPACK(IS),YPACK(IS),ZPACK(IS))
```

```
   60 CONTINUE
      WRITE(10) XPACK,YPACK,ZPACK
      NS2=NS2+NBS
      IF(NS2.LT.NBR) GO TO 35
      REWIND 10
      GO TO 800
C THE SECOND CALL TO THIS OVERLAY BEGINS HERE.
C READ THE POTENTIAL INTO THE PHI MESH FROM DISK FILES 1,2,5, AND 6.
   70 READ(1) (((PHI(I,J,K),I=1,NO4),J=1,NO4),K=1,NHO2)
      REWIND 1
      READ(2) (((PHI(I,J,K),I=NO4P1,NO2),J=1,NO4),K=1,NHO2)
      REWIND 2
      READ(5) (((PHI(I,J,K),I=1,NO4),J=NO4P1,NO2),K=1,NHO2)
      REWIND 5
      READ(6) (((PHI(I,J,K),I=NO4P1,NO2),J=NO4P1,NO2),K=1,NHO2)
      REWIND  6
C SET LEAST SIGNIFICANT HALF OF EACH WORD OF PHI MESH EQUAL TO ZERO.
      DO 80 K=1,NHO2
      DO 80 J=1,NO2
      DO 80 I=1,NO2
   80 PHI(I,J,K)=PHI(I,J,K).AND.MASK1
C AVERAGE THE Z COMPONENT OF GRAVITATIONAL FIELD ALONG FOUR RADIAL DIRECTIONS.
C STORE IT IN THE E ARRAY AS A FUNCTION OF R AND Z AND PRINT IT.
      DO 130 I=1,NO4M1
      DO 120 K=2,NHO4
      KP=NHO4+K
      KM=NHO4+1-K
      EIPKP=2.*(PHI(NO4+I,NO4,KP-1)-PHI(NO4+I,NO4,KP))
      EIMKP=2.*(PHI(NO4-I,NO4,KP-1)-PHI(NO4-I,NO4,KP))
      EJPKP=2.*(PHI(NO4,NO4+I,KP-1)-PHI(NO4,NO4+I,KP))
      EJMKP=2.*(PHI(NO4,NO4-I,KP-1)-PHI(NO4,NO4-I,KP))
      EKP=(EIPKP+EIMKP+EJPKP+EJMKP)/4.
      EIPKM=2.*(PHI(NO4+I,NO4,KM+1)-PHI(NO4+I,NO4,KM))
      EIMKM=2.*(PHI(NO4-I,NO4,KM+1)-PHI(NO4-I,NO4,KM))
      EJPKM=2.*(PHI(NO4,NO4+I,KM+1)-PHI(NO4,NO4+I,KM))
      EJMKM=2.*(PHI(NO4,NO4-I,KM+1)-PHI(NO4,NO4-I,KM))
      EKM=(EIPKM+EIMKM+EJPKM+EJMKM)/4.
  120 E (I+1,K)=(EKP+EKM)/2.
  130 E(I+1,1)=0.
      DO 140 K=2,NHO4
```

```
      KP=NH04+K
      KM=NH04+1-K
      EKP=2.*(PHI(N04,N04,KP-1)-PHI(N04,N04,KP))
      EKM=2.*(PHI(N04,N04,KM+1)-PHI(N04,N04,KM))
  140 E (1,K)=(EKP+EKM)/2.
      E (1,1)=0.
      PRINT 910
  910 FORMAT(1H0 *Z COMPONENT OF FIELD*)
      PRINT 920,((E(IR,IZ),IZ=1,NH04),IR=1,N04M1)
  920 FORMAT(1H ,4E14.7)
C DIVIDE THE RING MASSES IN MASSD BY RING AREA TO STORE MASS DENSITY IN MASSD.
      DO 160 IZ=1,NH04
      DO 150 IR=2,N04M1
  150 MASSD(IR,IZ)=MASSD(IR,IZ)/(4.*PI*(IR-1))
  160 MASSD(1,IZ)=MASSD(1,IZ)/(.50*PI)
C COMPUTE A STANDARD DEVIATION OF THE Z VELOCITY WHICH WILL CREATE A PRESSURE
C GRADIENT IN Z JUST SUFFICIENT TO BALANCE THE Z COMPONENT OF THE GRAVITATIONAL
C FIELD.  STORE THE STANDARD DEVIATION IN SIGMAVZ AS A FUNCTION OF R AND Z.
      NH04P1=NH04+1
      DO 180 IR=1,N04M1
      SIGMAVZ(IR,NH04)=0.
      DO 180 IIZ=2,NH04
      IZ=NH04P1-IIZ
  180 SIGMAVZ(IR,IZ)=SIGMAVZ(IR,IZ+1)+MASSD(IR,IZ+1)*E(IR,IZ+1)
      DO 190 IR=1,N04M1
      DO 190 IZ=1,NH04
      IF(MASSD(IR,IZ).EQ.0.) GO TO 185
      IF(SIGMAVZ(IR,IZ).LT.0.) SIGMAVZ(IR,IZ)=0.
      SIGMAVZ(IR,IZ)=SQRT(SIGMAVZ(IR,IZ)/MASSD(IR,IZ))
      GO TO 190
  185 SIGMAVZ(IR,IZ)=0.
  190 CONTINUE
C AVERAGE THE RADIAL COMPONENT OF THE GRAVITATIONAL FIELD ALONG FOUR RADIAL
C DIRECTIONS, STORE IT IN THE E ARRAY AS A FUNCTION OF R AND Z, AND PRINT IT.
      DO 200 IZ=1,NH04
      KP=NH04+IZ
      KM=NH04+1-IZ
      DO 200 IR=1,N04M2
      IP=N04+IR
      IM=N04-IR
```

```
      EIPKP =PHI(IP-1,NO4,KP)-PHI(IP+1,NO4,KP)
      EIMKP =PHI(IM+1,NO4,KP)-PHI(IM-1,NO4,KP)
      EJPKP =PHI(NO4,IP-1,KP)-PHI(NO4,IP+1,KP)
      EJMKP =PHI(NO4,IM+1,KP)-PHI(NO4,IM-1,KP)
      EKP= (EIPKP+ EIMKP+ EJPKP+ EJMKP)/4.
      EIPKM =PHI(IP-1,NO4,KM)-PHI(IP+1,NO4,KM)
      EIMKM =PHI(IM+1,NO4,KM)-PHI(IM-1,NO4,KM)
      EJPKM =PHI(NO4,IP-1,KM)-PHI(NO4,IP+1,KM)
      EJMKM =PHI(NO4,IM+1,KM)-PHI(NO4,IM-1,KM)
      EKM=( EIPKM+ EIMKM+ EJPKM+ EJMKM)/4.
  200 TH(IR+1,IZ)=(EKP+EKM)/2.
      DO 220 IR=2,NO4M1
      DO 210 IZ=2,NH04
  210 E(IR,IZ)=(TH(IR,IZ)+TH(IR,IZ-1))/2.
  220 E(IR,1)=3./2.*TH(IR,2)-.5*TH(IR,3)
      DO 230 IZ=1,NH04
  230 E (1,IZ)=0.
      PRINT 930
  930 FORMAT(1H0,*R COMPONENT OF FIELD*)
      PRINT 920,((E(IR,IZ),IZ=1,NH04),IR=1,NO4M1)
C COMPUTE THE ANGULAR VELOCITY ANGV REQUIRED TO BALANCE THE RADIAL COMPONENT
C OF THE GRAVITATIONAL FIELD AT HALF OF THE INITIAL RADIUS.
      IR02=RI/2.+1.5
      ANGV=SQRT( ABS(E(IR02,1))/(IR02-1))
C COMPUTE THE TIME OF ROTATION TROT.
      TROT=2.*PI/ANGV
C USING THE NUMBER OF TIME STEPS PER ROTATION DDD, COMPUTE THE TIME STEP DT.
      DT=TROT/DDD
      PRINT 242,XM
  242 FORMAT(1H0,3HXM=,E16.8)
      PRINT 244,DT
  244 FORMAT(1H ,3HDT=,E16.8)
C TIME SCALE THE PARTICLE MASS DR AND SET ITS LEAST SIGNIFICANT DIGITS EQUAL
C TO ZERO.
      DT2=DT**2
      DR=DR*DT2
      DR=DR.AND.MASK1
C COMPUTE DTE2 WHICH IS LATER USED TO APPLY A CENTRAL GRAVITATIONAL FORCE TO
C STARS WHICH ARE OUTSIDE OF THE MESH.
      DTE2=-XM*NBR*DT2
```

```
C TIME SCALE THE RADIAL GRAVITATIONAL FIELD E, THE STANDARD DEVIATION OF THE
C Z COMPONENT OF VELOCITY SIGMAVZ AND THE R-Z MASS DENSITY MASSD.
      DO 246 IZ=1,NH04
      DO 246 IR=1,N04M1
      E(IR,IZ)=E(IR,IZ)*DT2
      SIGMAVZ(IR,IZ)=SIGMAVZ(IR,IZ)*DT
  246 MASSD(IR,IZ)=MASSD(IR,IZ)*DT2
C FOR EACH R-Z MESH POINT COMPUTE THE MINIMUM STANDARD DEVIATION OF THE RADIAL
C VELOCITY WHICH SATISFIES THE TOOMRE STABILITY CRITERION AND STORE IN SIGMAVR.
C COMPUTE THE ANGULAR VELOCITY REQUIRED TO BALANCE THE RADIAL GRAVITATIONAL
C FORCE AND STORE IN VTR. (COMMENTS WILL HENCEFORTH REFER TO THIS QUANTITY AS
C THE ANGULAR VELOCITY OF THE COLD DISK.)
      DO 260 IZ=1,NH04
      DO 250 IR=2,N04M2
      SIGMAVR(IR,IZ)=(E(IR+1,IZ)-E(IR-1,IZ))/2.+E(IR,IZ)*3./(IR-1.)
      VTR(IR,IZ)=SQRT(E(IR,IZ)/(IR-1.))
  250 SIGMAVR(IR,IZ)=3.36*MASSD(IR,IZ)*2.000/SQRT(ABS(SIGMAVR(IR,IZ)))
      SIGMAVR(1,IZ)=2.*SIGMAVR(2,IZ)-SIGMAVR(3,IZ)
      SIGMAVR(N04M1,IZ)=SIGMAVR(N04M2,IZ)
      VTR(1,IZ)=2.*VTR(2,IZ)-VTR(3,IZ)
  260 VTR(N04M1,IZ)=VTR(N04M2,IZ)
C FOR EACH R-Z MESH POINT COMPUTE THE RATIO OF THE STANDARD DEVIATIONS OF THE
C AZIMUTHAL AND RADIAL VELOCITIES AND STORE IN TH.
      DO 280 IZ=1,NH04
      DO 270 IR=2,N04M2
  270 TH(IR,IZ)=SQRT(ABS(1.+(IR-1.)*(VTR(IR+1,IZ)-VTR(IR-1,IZ))/
     1    (4.*VTR(IR,IZ))))
      TH(1,IZ)=1.
  280 TH(N04M1,IZ)=TH(N04M2,IZ)
C FOR EACH R-Z MESH POINT COMPUTE THE PRODUCT OF THE MASS DENSITY AND THE
C SQUARE OF THE STANDARD DEVIATION OF THE RADIAL VELOCITY AND STORE IN VTR.
      DO 290 IZ=1,NH04
      DO 290 IR=1,N04M1
  290 VTR(IR,IZ)=MASSD(IR,IZ)*SIGMAVR(IR,IZ)**2
C FOR EACH R-Z MESH POINT COMPUTE THE DIFFERENCE IN THE SQUARES OF THE ANGULAR
C VELOCITIES OF THE COLD AND WARM DISKS AND STORE IN VRR.  THIS WILL NOT INCLUDE
C A TERM INVOLVING THE AVERAGE OF THE PRODUCT OF THE AXIAL AND RADIAL VELOCITIES
C WHICH WILL BE COMPUTED LATER.
      DO 310 IZ=1,NH04
      DO 300 IR=2,N04M2
```

```
      IF(MASSD(IR,IZ).EQ.0.) GO TO 295
      VRR(IR,IZ)=(VTR(IR+1,IZ)-VTR(IR-1,IZ))/((IR-1.)*MASSD(IR,IZ)*2.)+
     1    VTR(IR,IZ)*(1.-TH(IR,IZ)**2)/((IR-1.)*(IR-1.)*MASSD(IR,IZ))
      GO TO 300
  295 VRR(IR,IZ)=0.
  300 CONTINUE
      VRR(1,IZ)=2.*VRR(2,IZ)-VRR(3,IZ)
      IF(MASSD(3,IZ).EQ.0.) VRR(1,IZ)=VRR(2,IZ)
      IF(MASSD(1,IZ).EQ.0.) VRR(1,IZ)=0.
      VRR(NO4M1,IZ)=VRR(NO4M2,IZ)
      IF(MASSD(NO4M1,IZ).EQ.0.) VRR(NO4M1,IZ)=0.
  310 CONTINUE
C SET TO ZERO THE R-Z MESH VRVZ IN WHICH WILL BE SUMMED THE PRODUCT OF THE
C AXIAL AND RADIAL VELOCITIES.
      DO 320 IZ=1,NHO4
      DO 320 IR=1,NO4M1
  320 VRVZ(IR,IZ)=0.
C READ THE PACKED STAR POSITION AND VELOCITY COMPONENTS FROM TAPE 10.  UNPACK
C THE POSITION AND VELOCITY COMPONENTS.  DO A BILINEAR INTERPOLATION OVER THE
C R-Z MESH POINTS OF THE STANDARD DEVIATION OF THE RADIAL VELOCITY IN SIGMAVR
C AND STORE THE VALUE IN VR.  DO A BILINEAR INTERPOLATION OF THE RATIO OF THE
C STANDARD DEVIATIONS OF THE AZIMUTHAL AND RADIAL VELOCITIES CONTAINED IN TH
C AND STORE THE VALUE IN VT.  DO A BILINEAR INTERPOLATION OF THE STANDARD
C DEVIATION OF THE AXIAL VELOCITY AND STORE IN VZ.  USING THESE INTERPOLATED
C VALUES OF THE STANDARD DEVIATIONS OF THE VELOCITIES IN THE RADIAL AZIMUTHAL
C AND AXIAL DIRECTIONS, GIVE THE PARTICLES MAXWELLIAN VELOCITY DISTRIBUTIONS
C IN THESE DIRECTIONS AND STORE IN VR, VT, AND VZ.  IN THE VRVZ MESH SUM THE
C PRODUCT OF THE RADIAL AND AXIAL VELOCITIES. COMPUTE VX AND VY FROM VR AND VT.
C PACK   X  AND  VX, Y AND VY, AND Z AND VZ AND STORE ON DISK FILE 11.
      X=URAN(1234567.)
      VZAV1=0.
      NS2=0
  330 READ(10) XPACK,YPACK,ZPACK
      DO 360 IS=1,NBS
      CALL UNPACK(XS(1),XPACK(IS),YPACK(IS),ZPACK(IS))
      XC=X-CXY
      YC=Y-CXY
      R2=XC*XC+YC*YC
      R=SQRT(R2)
      IR=R
```

```
      DRR=R-IR
      IR=IR+1
      ZC=ABS(Z-CZ)
      IZ=ZC
      DZZ=ZC-IZ
      IZ=IZ+1
      D11=1-DZZ-DRR+DRR*DZZ
      D12=DZZ*(1-DRR)
      D21=DRR*(1-DZZ)
      D22=DRR*DZZ
      IF(MASSD(IR,IZ).EQ.0.) D11=0.
      IF(MASSD(IR,IZ+1).EQ.0.) D12=0.
      IF(MASSD(IR+1,IZ).EQ.0.) D21=0.
      IF(MASSD(IR+1,IZ+1).EQ.0.) D22=0.
      DSUM=D11+D12+D21+D22
      VR=D11*SIGMAVR(IR,IZ)+D12*SIGMAVR(IR,IZ+1)+D21*SIGMAVR(IR+1,IZ)+
     1    D22*SIGMAVR(IR+1,IZ+1)
      VR=VR/DSUM
      VT=D11*TH(IR,IZ)+D12*TH(IR,IZ+1)+D21*TH(IR+1,IZ)+D22*TH(IR+1,IZ+1)
      VT=VT/DSUM
      VZ=D11*SIGMAVZ(IR,IZ)+D12*SIGMAVZ(IR,IZ+1)+D21*SIGMAVZ(IR+1,IZ)+
     1    D22*SIGMAVZ(IR+1,IZ+1)
      VZ=VZ/DSUM
      VR=VR*SQRT(-ALOG(1.-URAN(0.)))*S2
      THET1=2.*PI*URAN(0.)
      VT=VT*VR*SIN(THET1)
      VR=VR*COS(THET1)
      THETA=ATAN2(-YC,XC)
      VX=VR*COS(THETA)-VT*SIN(THETA)
      IF(ABS(VX).GT.VXYMAX) VX=SIGN(VXYMAX,VX)
      VY=-VR*SIN(THETA)-VT*COS(THETA)
      IF(ABS(VY).GT.VXYMAX) VY=SIGN(VXYMAX,VY)
      VZ=VZ*SQRT(-ALOG(1.-URAN(0.)))*S2
      VZ=VZ*COS(2.*PI*URAN(0.))
      IF(ABS(VZ).GT.VZMAXI) VZ=SIGN(VZMAXI,VZ)
      VZAV1=VZAV1+VZ
      IZ=ZC+1.5
      IR=R+1.5
      VRVZ(IR,IZ)=VRVZ(IR,IZ)+VR*VZ
  360 CALL PACK(XS(1),XPACK(IS),YPACK(IS),ZPACK(IS))
```

```
      WRITE(11) XPACK,YPACK,ZPACK
      NS2=NS2+NBS
      IF(NS2.LT.NBR) GO TO 330
      REWIND 10
      REWIND 11
      VZAV1=VZAV1/NBR
C ADD THE AXIAL PARTIAL DERIVATIVE OF THE AVERAGE VALUE OF THE PRODUCT OF THE
C RADIAL AND AXIAL VELOCITIES TO WHAT IS ALREADY IN THE VRR MESH TO FORM THE
C DIFFERENCE IN THE SQUARES OF THE BALANCED ANGULAR VELOCITIES OF THE COLD
C AND WARM DISKS.
      NH04M1=NH04-1
      DO 385 IR=2,N04M1
      DO 383 IZ=2,NH04M1
      IF(MASSD(IR,IZ).EQ.0.) GO TO 383
      VRR(IR,IZ)=VRR(IR,IZ)+(VRVZ(IR,IZ+1)-VRVZ(IR,IZ-1))*DR/
     1    (2.*(IR-1)*MASSD(IR,IZ))
  383 CONTINUE
      VRR(IR,1)=2.*VRR(IR,2)-VRR(IR,3)
      IF(MASSD(IR,3).EQ.0.) VRR(IR,1)=VRR(IR,2)
      IF(MASSD(IR,1).EQ.0.) VRR(IR,1)=0.
      VRR(IR,NH04)=VRR(IR,NH04M1)
      IF(MASSD(IR,NH04).EQ.0.) VRR(IR,NH04)=0.
  385 CONTINUE
      VRR(1,1)=2.*VRR(2,1)-VRR(3,1)
      IF(MASSD(3,1).EQ.0.) VRR(1,1)=VRR(2,1)
      IF(MASSD(1,1).EQ.0.) VRR(1,1)=0.
C ENCODE CYCLE NUMBER CYY IN PREPARATION FOR PLOTTING.
      ENCODE(10,388,XPP(3)) CYY
      ENCODE(10,388,YPP(3)) CYY
      ENCODE(10,388,RPP(3)) CYY
  388 FORMAT(I10)
C READ THE STAR POSITIONS AND VELOCITIES FROM DISK FILE 11 AND UNPACK.  DO A
C BILINEAR INTERPOLATION OVER R-Z OF THE RADIAL GRAVITATIONAL FIELD IN E AND
C STORE IN ER.  DIVIDE BY THE RADIUS TO OBTAIN THE SQUARE OF THE BALANCED
C ANGULAR VELOCITY OF THE COLD DISK AND STORE IN ANGV.  DO A BILINEAR
C INTERPOLATION OF VRR AND SUBTRACT FROM ANGV AND TAKE SQRT TO YIELD AVERAGE
C BALANCED ANGULAR VELOCITY AT THAT POINT.  ADD THE RESOLVED COMPONENTS OF THIS
C VALUE TO VX AND VY.  SUBTRACT THE AVERAGE VALUE OF VZ FROM VZ.  TIME CENTER
C THE POSITIONS.  COMPUTE A NEW DENSITY IN THE LAST HALF OF EACH WORD OF THE PHI
C MESH.  PACK THE STAR POSITIONS AND VELOCITIES AND WRITE ON DISK FILE 10.  MAKE
C AN X-Y PLOT OF STAR POSITIONS.
```

```
      NS2=0
  390 READ(11) XPACK,YPACK,ZPACK
      DO 420 IS=1,NBS
      CALL UNPACK(XS(1),XPACK(IS),YPACK(IS),ZPACK(IS))
      XC=X-CXY
      YC=Y-CXY
      R=SQRT(XC*XC+YC*YC)
      IR=R
      DRR=R-IR
      IR=IR+1
      ZC=ABS(Z-CZ)
      IZ=ZC
      DZZ=ZC-IZ
      IZ=IZ+1
      D11=1-DZZ-DRR+DRR*DZZ
      D12=DZZ*(1-DRR)
      D21=DRR*(1-DZZ)
      D22=DRR*DZZ
      ER= D11*E(IR,IZ)+D12*E(IR,IZ+1)+D21*E(IR+1,IZ)+D22*E(IR+1,IZ+1)
      ER=ABS(ER)
      RR=R
      IF(RR.LT.0.11) RR=RR+0.05
      ANGV=ER/RR
      IF(MASSD(IR,IZ).EQ.0.) D11=0.
      IF(MASSD(IR,IZ+1).EQ.0.) D12=0.
      IF(MASSD(IR+1,IZ).EQ.0.) D21=0.
      IF(MASSD(IR+1,IZ+1).EQ.0.) D22=0.
      DSUM=D11+D12+D21+D22
      VP=D11*VRR(IR,IZ)+D12*VRR(IR,IZ+1)+D21*VRR(IR+1,IZ)+
     1    D22*VRR(IR+1,IZ+1)
      VP=VP/DSUM
      ANGV=ANGV-ABS(VP)
      IF(ANGV.LT.0.) ANGV=0.
      ANGV=SQRT(ANGV)
      VY=VY-ANGV*XC
      VX=VX+ANGV*YC
      VZ=VZ-VZAV1
      THETA=ATAN2(-YC,XC)
      X=R*COS(ANGV/2.+THETA)+CXY
```

```
      Y=-R*SIN(ANGV/2.+THETA)+CXY
      IX=X+.5
      JY=Y+.5
      KZ=Z+.5
      CALL DENS(DR,PHI(IX,JY,KZ))
  420 CALL PACK(XS(1),XPACK(IS),YPACK(IS),ZPACK(IS))
      Q=0
      IF(NS2.EQ.NMK) Q=1
      CALL DDIPLT(Q,IN,NBS,XPACK,YPACK,XMIN,XMAX,YMIN,YMAX,
     1   3,XPP,1,YP,13,ITAPX)
      WRITE(10) XPACK,YPACK,ZPACK
      NS2=NS2+NBS
      IF(NS2.LT.NBR) GO TO 390
      REWIND 10
      REWIND 11
C READ THE POSITIONS AND VELOCITIES FROM DISK FILE 10 AND MAKE AN X-Z PLOT OF
C STAR POSITIONS.
      NS2=0
  430 READ(10) XPACK,YPACK,ZPACK
      Q=0
      IF(NS2.EQ.NMK) Q=1
      CALL DDIPLT(Q,IN,NBS,XPACK,ZPACK,XMIN,XMAX,ZMIN,ZMAX,
     1   3,XPP,1,ZP,13,ITAPX)
      NS2=NS2+NBS
      IF(NS2.LT.NBR) GO TO 430
      REWIND 10
C READ FROM DISK FILE 10 AND MAKE A Y-Z PLOT OF STAR POSITIONS.  CONVERT THE
C X AND Y POSITIONS AND VELOCITIES TO RADIAL AND AZIMUTHAL POSITIONS AND
C VELOCITIES AND WRITE THEM ON DISK FILE 12.
      NS2=0
  450 READ(10) XPACK,YPACK,ZPACK
      Q=0
      IF(NS2.EQ.NMK) Q=1
      CALL DDIPLT(Q,IN,NBS,YPACK,ZPACK,YMIN,YMAX,ZMIN,ZMAX,3,YPP,
     1    1,ZP,13,ITAPX)
      DO 480 IS=1,NBS
      CALL UNPACK(XS(1),XPACK(IS),YPACK(IS),ZPACK(IS))
      XC=X-CXY
      YC=Y-CXY
      R=SQRT(XC*XC+YC*YC)
```

```
      THETA=ATAN2(-YC,XC)
      VR=(XC*VX+YC*VY)/R
      VT=-(XC*VY-YC*VX)/R
      ZR=Z
      VZR=VZ
  480 CALL PACK(RS(1),RPACK(IS),TPACK(IS),ZPACK(IS))
      WRITE(12) RPACK,TPACK,ZPACK
      NS2=NS2+NBS
      IF(NS2.LT.NBR) GO TO 450
      REWIND 10
      REWIND 12
C READ FROM DISK FILE 12 AND MAKE A PLOT OF RADIAL VELOCITY VS RADIUS.
      NS2=0
  490 READ(12) RPACK,TPACK,ZPACK
      DO 510 IS=1,NBS
  510 TPACK(IS)=RPACK(IS)
      CALL SHIFT2(RPACK(1) ,NBS)
      Q=0
      IF(NS2.EQ.NMK) Q=1
      CALL DDIPLT(Q,IN,NBS,TPACK,RPACK,RMIN,RMAX,VRMIN,VRMAX,3,RPP,
     1     1,VRP,13,ITAPX)
      NS2=NS2+NBS
      IF(NS2.LT.NBR) GO TO 490
      REWIND 12
C READ FROM DISK FILE 12 AND MAKE A PLOT OF AZIMUTHAL VELOCITY VS RADIUS.
      NS2=0
  520 READ(12) RPACK,TPACK,ZPACK
      CALL SHIFT2(TPACK(1),NBS)
      Q=0
      IF(NS2.EQ.NMK) Q=1
      CALL DDIPLT(Q,IN,NBS,RPACK,TPACK,RMIN,RMAX,VTMIN,VTMAX,3,RPP,
     1     1,VTP,13,ITAPX)
      NS2=NS2+NBS
      IF(NS2.LT.NBR) GO TO 520
      REWIND 12
C READ FROM DISK FILE 12 AND MAKE A PLOT OF AXIAL VELOCITY VS RADIUS.
      NS2=0
  540 READ(12) RPACK,TPACK,ZPACK
      CALL SHIFT2(ZPACK(1),NBS)
      Q=0
```

```
      IF(NS2.EQ.NMK) Q=1
      CALL DDIPLT(Q,IN,NBS,RPACK,ZPACK,RMIN,RMAX,VZMIN,VZMAX,3,RPP,
     1    1,VZP,13,ITAPX)
      NS2=NS2+NBS
      IF(NS2.LT.NBR) GO TO 540
      REWIND 12
C SHIFT THE DENSITY TO THE MOST SIGNIFICANT HALF OF EACH WORD OF THE PHI MESH.
  800 CALL SHIFT2(PHI(1,1,1),NCH)
C WRITE THE DENSITY FROM THE PHI MESH ONTO DISK FILES 1,2,5, AND 6.
      WRITE(1) (((PHI(I,J,K),I=1,NO4),J=1,NO4),K=1,NHO2)
      REWIND 1
      WRITE(2) (((PHI(I,J,K),I=NO4P1,NO2),J=1,NO4),K=1,NHO2)
      REWIND 2
      WRITE(5) (((PHI(I,J,K),I=1,NO4),J=NO4P1,NO2),K=1,NHO2)
      REWIND 5
      WRITE(6) (((PHI(I,J,K),I=NO4P1,NO2),J=NO4P1,NO2),K=1,NHO2)
      REWIND 6
      RETURN
      END
```

```
      OVERLAY(SFILE,4,0)
      PROGRAM STARS
C THIS OVERLAY USES THE GRAVITATIONAL POTENTIAL COMPUTED IN THE GETPHI OVERLAY
C TO COMPUTE NEW STAR POSITIONS AND VELOCITIES AND TO COMPUTE A DENSITY MESH
C USED IN THE GETPHI OVERLAY.  IT ALSO PRINTS OUT CERTAIN DIAGNOSTICS EVERY
C CYCLE AND PERIODICALLY PRINTS OUT MORE EXTENSIVE DIAGNOSTICS AND MAKES PLOTS.
      COMMON/ALLCOM/N,NO2,N21,NO4,N41,N34,NH,NHO2,NH21,NCH,NRHO,NHH,
     1     I2A,I2B,I3A,I3B,NHO4
      COMMON/ADVCOM/NBR,NBS,RI,XM,DT,DTE2,NO4M1,NO4M2,VXYMAX,VZMAXI,
     1     CY,CYY,NPLOT,NPRINT,IN(2),XMIN,XMAX,YMIN,YMAX,ZMIN,
     2     ZMAX,RMIN,RMAX,VRMIN,VRMAX,VTMIN,VTMAX,VZMIN,VZMAX,XPP(3),
     3     YP,ZP,YPP(3),RPP(3),VTP,VRP,VZP,ITAPX,PI,MASK1,S2,JT,JS,NMK,
     4     NBS3,CXY,CZ,VZAV1,DR,ITEST,JTFILE,JSFILE,DDD,NO4P1
      DIMENSION PHI(32,32,8),MASSDR(16),VR2AVGR(16),VRAVGR(16),
     1     VT2AVGR(16),VTAVGR(16),VZ2AVGR(16),VZAVGR(16),MASSDZ(8),
     2     VZ2AVGZ(8),VZAVGZ(8)
      DIMENSION XS(6)
      DIMENSION XPACK(2048),YPACK(2048),ZPACK(2048)
      EQUIVALENCE (XS(1),X),(XS(2),VX),(XS(3),Y),(XS(4),VY),(XS(5),Z),
     1     (XS(6),VZ)
      INTEGER Q,CY,CYY,DDM
      REAL KE,MASSDR,MASSDZ
      CALL PSEUDO
      PRINT 20
   20 FORMAT(40H STAR NUMBER            X                 VX     ,
     1     56H              Y                  VY                 Z                 VZ     ,
     2     14H          R   /)
C READ THE POTENTIAL INTO THE PHI MESH FROM DISK FILES 1,2,5, AND 6.
      READ(1) (((PHI(I,J,K),I=1,NO4),J=1,NO4),K=1,NHO2)
      REWIND 1
      READ(2) (((PHI(I,J,K),I=NO4P1,NO2),J=1,NO4),K=1,NHO2)
      REWIND 2
      READ(5) (((PHI(I,J,K),I=1,NO4),J=NO4P1,NO2),K=1,NHO2)
      REWIND 5
      READ(6) (((PHI(I,J,K),I=NO4P1,NO2),J=NO4P1,NO2),K=1,NHO2)
      REWIND 6
C SET TO ZERO THE LEAST SIGNIFICANT HALF OF EACH WORD OF THE PHI MESH.
      DO 60 K=1,NHO2
      DO 60 J=1,NO2
      DO 60 I=1,NO2
```

```
   60 PHI(I,J,K)=PHI(I,J,K).AND.MASK1
C DETERMINE WHICH PLOTS AND/OR PRINTED DIAGNOSTICS (IF ANY) ARE TO BE MADE THIS
C CYCLE.
      IPRINT=1
      IF(CYY-CYY/NPRINT*NPRINT.EQ.0..OR.CYY.EQ.1) IPRINT=2
      IPLOT=1
      IF(CYY+5-(CYY+5)/NPLOT*NPLOT.EQ.0.)IPLOT=2
      IF(CYY+4-(CYY+4)/NPLOT*NPLOT.EQ.0.)IPLOT=3
      IF(CYY+3-(CYY+3)/NPLOT*NPLOT.EQ.0.)IPLOT=4
      IF(CYY+2-(CYY+2)/NPLOT*NPLOT.EQ.0.)IPLOT=5
      IF(CYY+1-(CYY+1)/NPLOT*NPLOT.EQ.0.)IPLOT=6
      IF(CYY-CYY/NPLOT*NPLOT.EQ.0.)IPLOT=7
C IF NECESSARY ENCODE THE CYCLE NUMBER CYY INTO PLOT AXES LABELS.
      IF(IPLOT.EQ.2.OR.IPLOT.EQ.3) ENCODE(10,65,XPP(3)) CYY
   65 FORMAT(I10)
      IF(IPLOT.EQ.4) ENCODE(10,65,YPP(3)) CYY
      IF(IPLOT.GE.5) ENCODE(10,65,RPP(3)) CYY
C IF LONG DIAGNOSTIC PRINTING IS TO BE DONE THIS CYCLE SET CONSTANT R MESHES
C AND CONSTANT Z MESHES TO ZERO.
      IF(IPRINT.EQ.1) GO TO 90
      DO 70 IR=1,NO4
      MASSDR(IR)=0.
      VR2AVGR(IR)=0.
      VRAVGR(IR)=0.
      VT2AVGR(IR)=0.
      VTAVGR(IR)=0.
      VZ2AVGR(IR)=0.
   70 VZAVGR(IR)=0.
      DO 80 KZ=1,NHO2
      MASSDZ(KZ)=0.
      VZ2AVGZ(KZ)=0.
   80 VZAVGZ(KZ)=0.
C READ THE PACKED STAR POSITIONS AND VELOCITIES FROM DISK FILE JT.  IF LONG
C DIAGNOSTIC PRINTING IS NOT TO BE DONE THIS CYCLE, THE STARS ARE PROCESSED
C THROUGH A FIRST DO LOOP ENDING WITH LINE 150.  IF LONG DIAGNOSTIC PRINTING
C IS TO BE DONE THIS CYCLE THE STARS ARE PROCESSED THROUGH A SECOND DO LOOP
C ENDING WITH LINE 210.  BOTH LOOPS TAKE THE GRADIENT OF THE POTENTIAL IN THE
C PHI MESH AND USE IT TO COMPUTE NEW STAR POSITIONS AND VELOCITIES.  ADVANCE
C THE PARTICLES WHICH ARE OUTSIDE OF A CYLINDER TANGENT TO THE PHI MESH BY
C ASSUMING THAT THE TOTAL MASS OF THE GALAXY RESIDES AT ITS CENTER.  SUM THE
```

```
C NUMBER OF PARTICLES OUTSIDE OF THIS CYLINDER.  COMPUTE THE DENSITY AND STORE
C IT IN THE LEAST SIGNIFICANT HALF OF EACH WORD OF THE PHI MESH.  PACK THE
C NEW STAR POSITIONS AND VELOCITIES AND WRITE THEM ON DISK FILE JS.  DEPENDING
C ON THE VALUE OF IPLOT (WHICH DEPENDS ON CYCLE NUMBER AND PLOTTING FREQUENCY)
C ONE OF SIX PLOTS MAY BE MADE.
      KE=0.
      PEIN=0.
      PEOUT=0.
   90 NUMBER=1
      DDM=0
      NS2=0
   95 READ(JT) XPACK,YPACK,ZPACK
      IF(IPRINT.EQ.2) GO TO 155
C     BEGINNING OF STAR ADVANCING LOOP FOR NO LONG DIAGNOSTIC PRINTING
      DO 150 IS=1,NBS
      CALL UNPACK(XS(1),XPACK(IS),YPACK(IS),ZPACK(IS))
      XC=X-NO4
      YC=Y-NO4
      R=SQRT(XC*XC+YC*YC)
      IX=X+.5
      JY=Y+.5
      KZ=Z+.5
      IR=R+1.5
      IF(KZ.LE.1) GO TO 110
      IF(KZ.GE.NHO2) GO TO 110
      IF(IR.GE.NO4) GO TO 110
      VX=VX+PHI(IX+1,JY,KZ)-PHI(IX-1,JY,KZ)
      VY=VY+PHI(IX,JY+1,KZ)-PHI(IX,JY-1,KZ)
      VZ=VZ+(Z-KZ+.5)*(PHI(IX,JY,KZ+1)-PHI(IX,JY,KZ))
     1    +(KZ-Z+.5)*(PHI(IX,JY,KZ)-PHI(IX,JY,KZ-1))
      GO TO 120
  110 ZC=Z-CZ
      R2=XC*XC+YC*YC+ZC*ZC
      R32=R2*SQRT(R2)
      E=DTE2/R32
      VX=VX+E*XC
      VY=VY+E*YC
```

```
      VZ=VZ+E*ZC
  120 X=X+VX
      Y=Y+VY
      Z=Z+VZ
      XC=X-N04
      YC=Y-N04
      R=SQRT(XC*XC+YC*YC)
      IX=X+.5
      JY=Y+.5
      KZ=Z+.5
      IR=R+1.5
      IF(KZ.LT.1) GO TO 140
      IF(KZ.GT.NH02) GO TO 140
      IF(IR.GT.N04) GO TO 140
      CALL DENS(DR,PHI(IX,JY,KZ))
      GO TO 150
  140 DDM=DDM+1
  150 CALL PACK(XS(1),XPACK(IS),YPACK(IS),ZPACK(IS))
      GO TO 220
C     BEGINNING OF STAR ADVANCING LOOP USED FOR LONG DIAGNOSTIC PRINTING.  THE
C     FOLLOWING MESHES SUM THE DESIGNATED QUANTITY FOR A PARTICULAR MASS RING -
C         VR2AVGR(IR) - RADIAL VELOCITY SQUARED
C         VRAVGR(IR)  - RADIAL VELOCITY
C         VT2AVGR(IR) - AZIMUTHAL VELOCITY SQUARED
C         VTAVGR(IR)  - AZIMUTHAL VELOCITY
C         VZ2AVGR(IR) - AXIAL VELOCITY SQUARED
C         VZAVGR(IR)  - AXIAL VELOCITY
C         MASSDR(IR)  - NUMBER OF PARTICLES
C     THE FOLLOWING MESHES SUM THE DESIGNATED QUANTITY FOR A PARTICULAR AXIAL
C     MASS LAYER -
C         MASSDZ(KZ)  - NUMBER OF PARTICLES
C         VZ2AVGZ(KZ) - AXIAL VELOCITY SQUARED
C         VZAVGZ(KZ)  - AXIAL VELOCITY
C     THE LOOP ALSO SUMS THE POTENTIAL AND KINETIC ENERGIES
  155 DO 210 IS=1,NBS
      CALL UNPACK(XS(1),XPACK(IS),YPACK(IS),ZPACK(IS))
      XC=X-N04
      YC=Y-N04
      R=SQRT(XC*XC+YC*YC)
      IX=X+.5
      JY=Y+.5
      KZ=Z+.5
```

```
      IR=R+1.5
      IF(KZ.LE.1) GO TO 160
      IF(KZ.GE.NHO2) GO TO 160
      IF(IR.GE.NO4) GO TO 160
      VX=VX+PHI(IX+1,JY,KZ)-PHI(IX-1,JY,KZ)
      VY=VY+PHI(IX,JY+1,KZ)-PHI(IX,JY-1,KZ)
      VZ=VZ+(Z-KZ+.5)*(PHI(IX,JY,KZ+1)-PHI(IX,JY,KZ))
     1    +(KZ-Z+.5)*(PHI(IX,JY,KZ)-PHI(IX,JY,KZ-1))
      GO TO 170
  160 ZC=Z-CZ
      R2=XC*XC+YC*YC+ZC*ZC
      R32=R2*SQRT(R2)
      E=DTE2/R32
      VX=VX+E*XC
      VY=VY+E*YC
      VZ=VZ+E*ZC
  170 X=X+VX
      Y=Y+VY
      Z=Z+VZ
      XC=X-NO4
      YC=Y-NO4
      R=SQRT(XC*XC+YC*YC)
      IX=X+.5
      JY=Y+.5
      KZ=Z+.5
      IR=R+1.5
      IF(KZ.LT.1) GO TO 200
      IF(KZ.GT.NHO2) GO TO 200
      IF(IR.GT.NO4) GO TO 200
      CALL DENS(DR,PHI(IX,JY,KZ))
      MASSDR(IR)=MASSDR(IR)+1.
      IF(R.EQ.0.) GO TO 180
      VR=(XC*VX+YC*VY)/R
      VT=(XC*VY-YC*VX)/R
      GO TO 190
  180 VR=SQRT(VX*VX+VY*VY)
      VT=0.
  190 VR2AVGR(IR)=VR2AVGR(IR)+VR*VR
      VRAVGR(IR)=VRAVGR(IR)+VR
```

```
      VT2AVGR(IR)=VT2AVGR(IR)+VT*VT
      VTAVGR(IR)=VTAVGR(IR)+VT
      VZ2=VZ*VZ
      VZ2AVGR(IR)=VZ2AVGR(IR)+VZ2
      VZAVGR(IR)=VZAVGR(IR)+VZ
      MASSDZ(KZ)=MASSDZ(KZ)+1.
      VZ2AVGZ(KZ)=VZ2AVGZ(KZ)+VZ2
      VZAVGZ(KZ)=VZAVGZ(KZ)+VZ
      PEIN=PEIN+PHI(IX,JY,KZ)
      GO TO 205
  200 DDM=DDM+1
      ZC=Z-CZ
      RR=SQRT(XC*XC+YC*YC+ZC*ZC)
      PEOUT=PEOUT+1./RR
      VZ2=VZ*VZ
  205 KE=KE+VX*VX+VY*VY+VZ2
  210 CALL PACK(XS(1),XPACK(IS),YPACK(IS),ZPACK(IS))
  220 PRINT 225,NUMBER,X,VX,Y,VY,Z,VZ,R
  225 FORMAT(1H ,I8,7E14.7)
      WRITE(JS) XPACK,YPACK,ZPACK
      Q=0
      IF(NS2.EQ.NMK) Q=1
C     DETERMINE WHICH PLOTS IF ANY ARE TO BE MADE.
      GO TO(300,240,250,260,270,280,290) IPLOT
C     MAKE AN X-Y STAR POSITION PLOT.
  240 CALL DDIPLT(Q,IN,NBS,XPACK,YPACK,XMIN,XMAX,YMIN,YMAX,
     1    3,XPP,1,YP,13,ITAPX)
      GO TO 300
C     MAKE AN X-Z STAR POSITION PLOT
  250 CALL DDIPLT(Q,IN,NBS,XPACK,ZPACK,XMIN,XMAX,ZMIN,ZMAX,
     1    3,XPP,1,ZP,13,ITAPX)
      GO TO 300
C     MAKE A Y-Z STAR POSITION PLOT
  260 CALL DDIPLT(Q,IN,NBS,YPACK,ZPACK,YMIN,YMAX,ZMIN,ZMAX,
     1    3,YPP,1,ZP,13,ITAPX)
      GO TO 300
C     MAKE A RADIAL VELOCITY VS. RADIUS PLOT
  270 DO 275 IS=1,NBS
      CALL UNPACK(XS(1),XPACK(IS),YPACK(IS),ZPACK(IS))
      XC=X-NO4
```

```
      YC=Y-N04
      R=SQRT(XC*XC+YC*YC)
      IF(R.EQ.0.) GO TO 272
      YPACK(IS)=(XC*VX+YC*VY)/R
      GO TO 275
  272 YPACK(IS)=SQRT(VX*VX+VY*VY)
  275 XPACK(IS)=R
      CALL DDIPLT(Q,IN,NBS,XPACK,YPACK,RMIN,RMAX,VRMIN,VRMAX,
     1     3,RPP,1,VRP,13,ITAPX)
      GO TO 300
C     MAKE AN AZIMUTHAL VELOCITY VS. RADIUS PLOT
  280 DO 285 IS=1,NBS
      CALL UNPACK(XS(1),XPACK(IS),YPACK(IS),ZPACK(IS))
      XC=X-N04
      YC=Y-N04
      R=SQRT(XC*XC+YC*YC)
      IF(R.EQ.0.) GO TO 282
      YPACK(IS)=-(XC*VY-YC*VX)/R
      GO TO 285
  282 YPACK(IS)=0.
  285 XPACK(IS)=R
      CALL DDIPLT(Q,IN,NBS,XPACK,YPACK,RMIN,RMAX,VTMIN,VTMAX,
     1     3,RPP,1,VTP,13,ITAPX)
      GO TO 300
C     MAKE AN AXIAL VELOCITY VS. RADIUS PLOT
  290 CALL SHIFT2(ZPACK(1),NBS)
      DO 295 IS=1,NBS
      XC=XPACK(IS)-N04
      YC=YPACK(IS)-N04
  295 XPACK(IS)=SQRT(XC*XC+YC*YC)
      CALL DDIPLT(Q,IN,NBS,XPACK,ZPACK,RMIN,RMAX,VZMIN,VZMAX,
     1     3,RPP,1,VZP,13,ITAPX)
  300 NS2=NS2+NBS
      NUMBER=NUMBER+1
      IF(NS2.LT.NBR) GO TO 95
C REWIND DISK FILES JS AND JT AND EXCHANGE THEIR TAPE NUMBERS
      REWIND JT
      REWIND JS
      JSAVE=JS
```

```
      JS=JT
      JT=JSAVE
      PRINT 305,DDM
  305 FORMAT(1H0,40HNUMBER OF PARTICLES OUTSIDE OF CYLINDER=,I6)
      IF(IPRINT.EQ.1) GO TO 500
C IF LONG DIAGNOSTIC  PRINTING IS TO BE DONE, PRINT TOTAL KINETIC ENERGY, TOTAL
C POTENTIAL ENERGY AND TOTAL ENERGY.  ALSO, COMPUTE AND PRINT AVERAGED
C QUANTITIES AS FUNCTIONS OF R, Z, AND R-Z.
      DT2=DT*DT
      KE=KE/(8.*DT2)
      PE=(DTE2*PEOUT-PEIN)/DT2
      ET=KE+PE
      PRINT 310,KE,PE,ET
  310 FORMAT(1H0,3HKE=,E16.8,6H   PE=,E16.8,10H  TOTAL E=,E16.8)
C     COMPUTE AND PRINT THE FOLLOWING AVERAGED QUANTITIES AS FUNCTIONS OF
C     RADIUS - MASS DENSITY, AVERAGE RADIAL VELOCITY, STANDARD DEVIATION OF
C     RADIAL VELOCITY, AVERAGE AZIMUTHAL VELOCITY, STANDARD DEVIATION OF
C     AZIMUTHAL VELOCITY, AVERAGE AXIAL VELOCITY AND STANDARD DEVIATION OF
C     AXIAL VELOCITY.
      DO 320 IR=1,NO4
      IF(MASSDR(IR).LT.1.) GO TO 320
      VR2AVGR(IR)=SQRT((VR2AVGR(IR)-VRAVGR(IR)*VRAVGR(IR)/MASSDR(IR))
     1     /(MASSDR(IR)*DT2))
      VRAVGR(IR)=VRAVGR(IR)/(MASSDR(IR)*DT)
      VT2AVGR(IR)=SQRT((VT2AVGR(IR)-VTAVGR(IR)*VTAVGR(IR)/MASSDR(IR))
     1     /(MASSDR(IR)*DT2))
      VTAVGR(IR)=-VTAVGR(IR)/(MASSDR(IR)*DT)
      VZ2AVGR(IR)=SQRT((VZ2AVGR(IR)-VZAVGR(IR)*VZAVGR(IR)/MASSDR(IR))
     1     /(MASSDR(IR)*DT2))
      VZAVGR(IR)=VZAVGR(IR)/(MASSDR(IR)*DT)
  320 CONTINUE
      MASSDR(1)=MASSDR(1)*XM/(.25*PI)
      XMO2PI=XM/(2*PI)
      DO 330 IR=2,NO4
      R=IR-1
  330 MASSDR(IR)=MASSDR(IR)*XMO2PI/R
      PRINT 340
```

```
340 FORMAT(1H0,38HRADIUS    MASS DENSITY          AVG VR,
   1    48H          SIGMA VR           AVG VT        SIGMA VT,
   2    32H          AVG VZ          SIGMA VZ/)
    DO 350 IR=1,NO4
    R=IR-1
350 PRINT 360,R,MASSDR(IR),VRAVGR(IR),VR2AVGR(IR),VTAVGR(IR),
   1    VT2AVGR(IR),VZAVGR(IR),VZ2AVGR(IR)
360 FORMAT(1H ,F6.1,7E16.8)
    DO 370 KZ=1,NHO2
C   COMPUTE AND PRINT THE FOLLOWING AVERAGE QUANTITIES AS FUNCTION
C   OF AXIAL POSITION(HEIGHT) - MASS DENSITY, AVERAGE AXIAL VELOCITY, AND
C   STANDARD DEVIATION OF AXIAL VELOCITY.
    IF(MASSDZ(KZ).LT.1.) GO TO 370
    VZ2AVGZ(KZ)=SQRT((VZ2AVGZ(KZ)-VZAVGZ(KZ)*VZAVGZ(KZ)/MASSDZ(KZ))
   1    /(MASSDZ(KZ)*DT2))
    VZAVGZ(KZ)=VZAVGZ(KZ)/(MASSDZ(KZ)*DT)
    MASSDZ(KZ)=MASSDZ(KZ)*XM
370 CONTINUE
    PRINT 380
380 FORMAT(1H0,42HHEIGHT(Z)     MASS DENSITY          AVG VZ,
   1    16H         SIGMA VZ/)
    DO 390 KZ=1,NHO2
    Z=KZ
390 PRINT 400,Z,MASSDZ(KZ),VZAVGZ(KZ),VZ2AVGZ(KZ)
400 FORMAT(1H ,F6.1,3E16.8)
C   COMPUTE AND PRINT AVERAGE RADIAL GRAVITATIONAL FIELD AS A FUNCTION OF R-Z.
    PRINT 410
410 FORMAT(1H0,50HRADIAL FIELD AS A FUNCTION OF R AND Z(KZ=2,NHO2-1))
    PRINT 420
420 FORMAT(7H RADIUS)
    NHO2M1=NHO2-1
    DO 440 IR=2,NO4M1
    R=IR-1
    NO4PIR=NO4+IR
    DO 430 KZ=2,NHO2M1
430 MASSDZ(KZ)=PHI(NO4PIR+1,NO4,KZ)-PHI(NO4PIR-1,NO4,KZ)
```

```
  440 PRINT 445,R,(MASSDZ(KZ),KZ=2,NHO2M1)
  445 FORMAT(1H ,F6.1,8E16.8)
C     COMPUTE AND PRINT AVERAGE AXIAL GRAVITATIONAL FIELD AS A FUNCTION OF R-Z.
      PRINT 450
  450 FORMAT(1H0,45HZ FIELD AS A FUNCTION OF R AND Z(KZ=2,NHO2-1))
      PRINT 460
  460 FORMAT(7H RADIUS)
      DO 463 KZ=2,NHO2M1
  463 MASSDZ(KZ)=PHI(NO4,NO4,KZ+1)-PHI(NO4,NO4,KZ-1)
      PRINT 465,(MASSDZ(KZ),KZ=2,NHO2M1)
  465 FORMAT(7H    0.0,8E16.8)
      DO 480 IR=2,NO4M1
      R=IR-1
      NO4PIR=NO4+IR
      DO 470 KZ=2,NHO2M1
  470 MASSDZ(KZ)=PHI(NO4PIR,NO4,KZ+1)-PHI(NO4PIR,NO4,KZ-1)
  480 PRINT 445,R,(MASSDZ(KZ),KZ=2,NHO2M1)
C SHIFT THE DENSITY TO THE MOST SIGNIFICANT HALF OF EACH WORD OF THE PHI MESH.
  500 CALL SHIFT2(PHI(1,1,1),NCH)
C WRITE  THE DENSITY FROM THE PHI MESH ONTO DISK FILES 1,2,5, AND 6.
      WRITE(1) (((PHI(I,J,K),I=1,NO4),J=1,NO4),K=1,NHO2)
      REWIND 1
      WRITE(2) (((PHI(I,J,K),I=NO4P1,NO2),J=1,NO4),K=1,NHO2)
      REWIND 2
      WRITE(5) (((PHI(I,J,K),I=1,NO4),J=NO4P1,NO2),K=1,NHO2)
      REWIND 5
      WRITE(6) (((PHI(I,J,K),I=NO4P1,NO2),J=NO4P1,NO2),K=1,NHO2)
      REWIND 6
      RETURN
      END
```

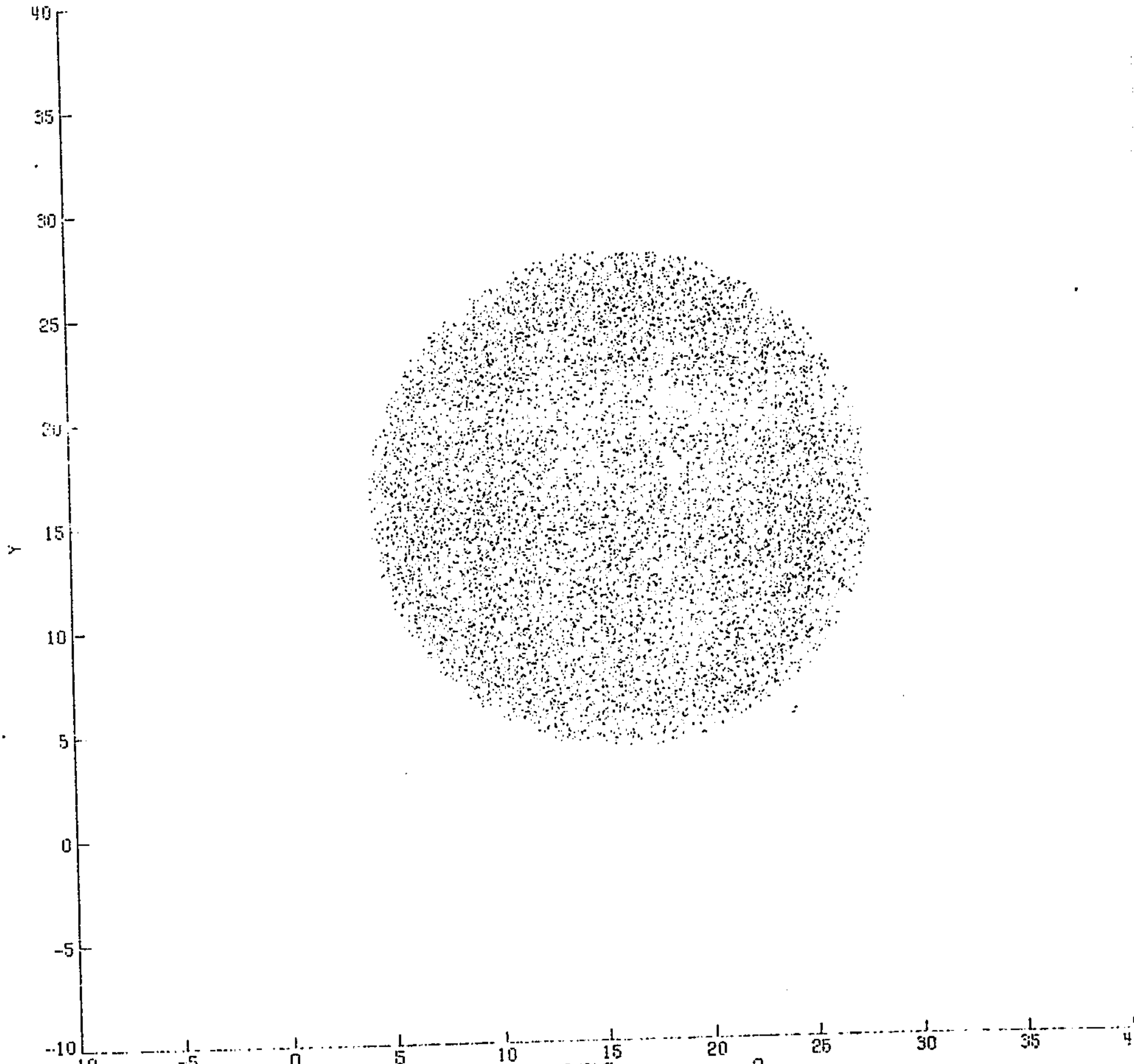
APPENDIX C

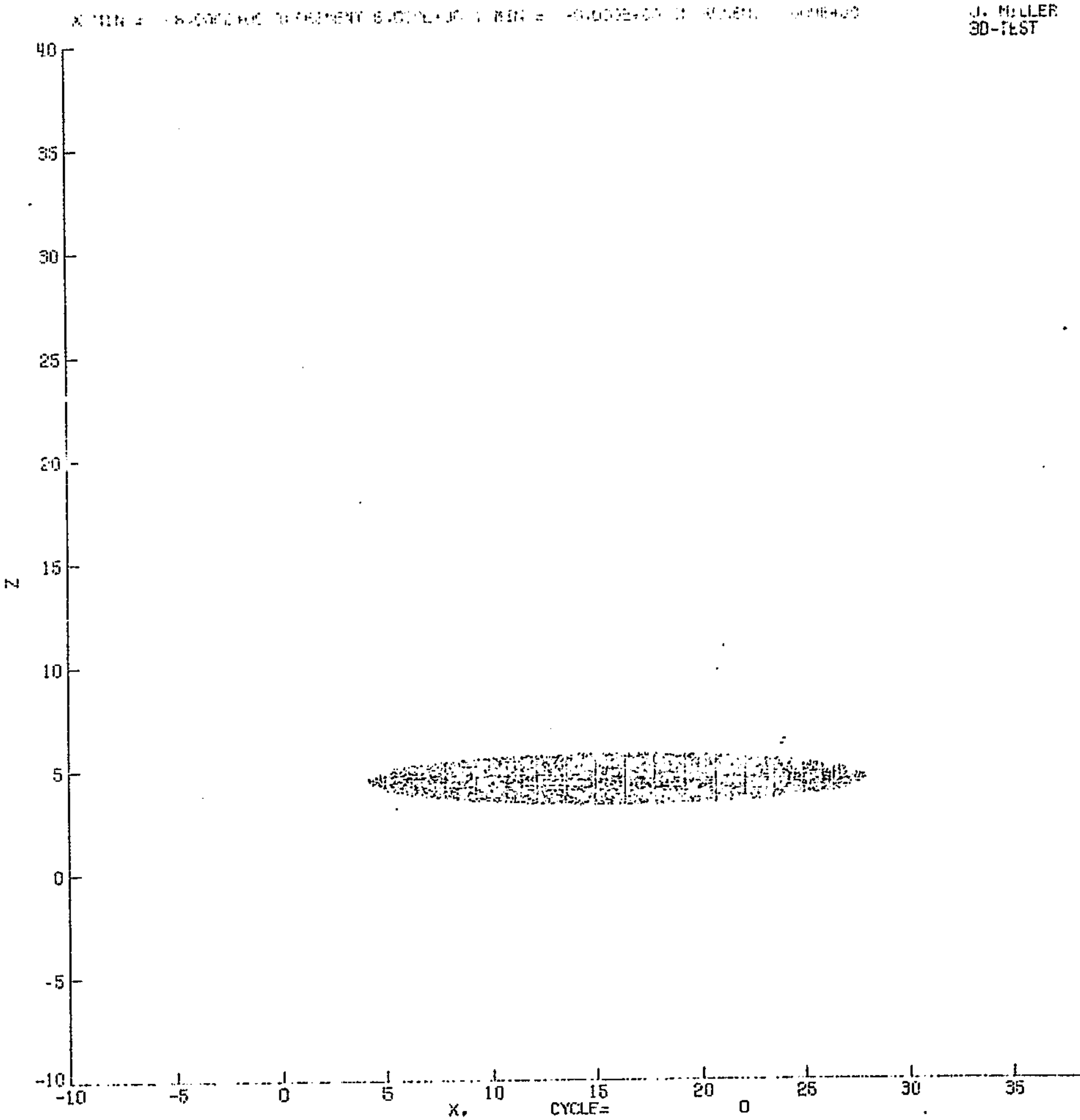
Computer Plots Produced by the Three-Dimensional Galaxy Simulator of Appendix B.

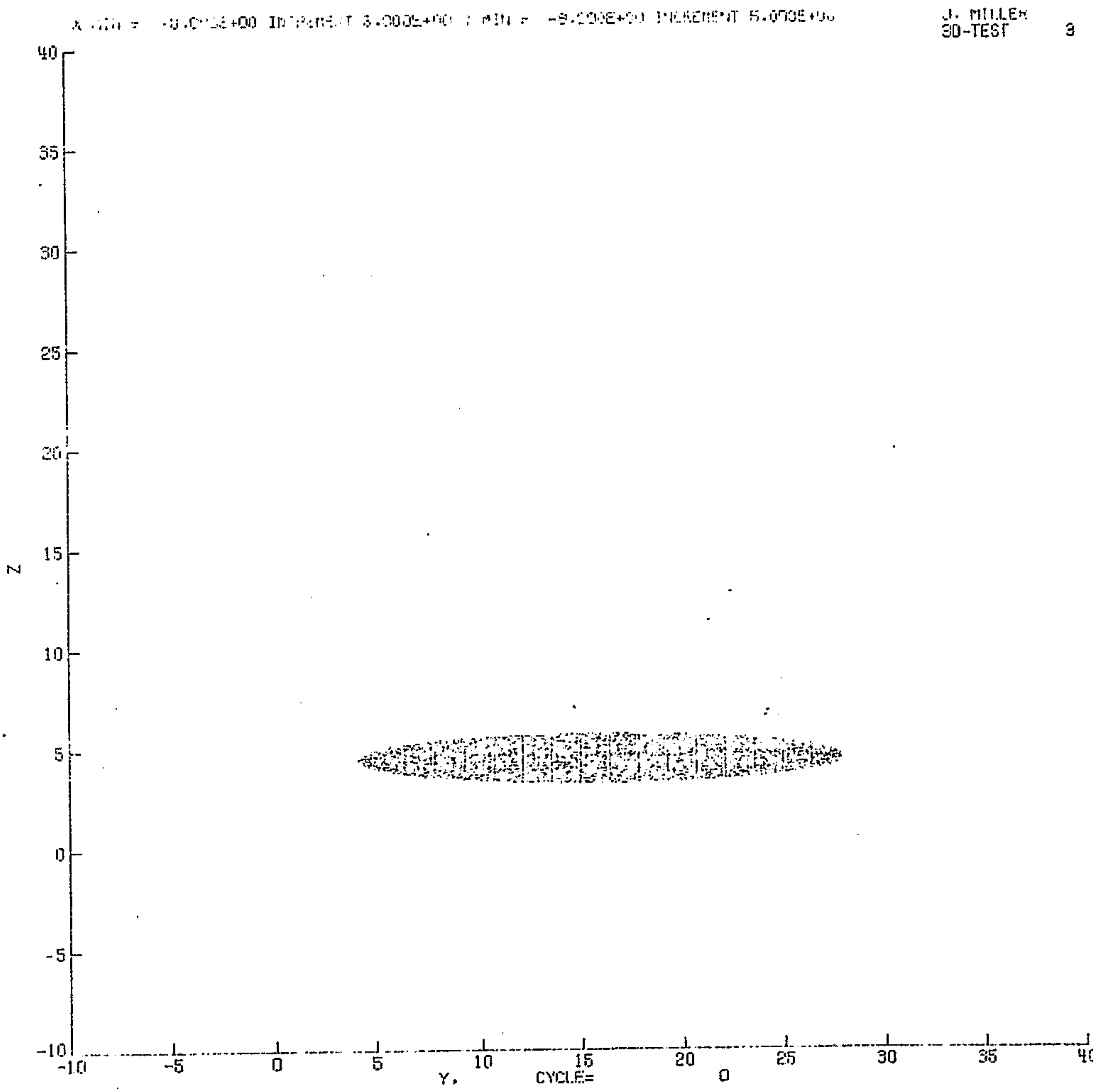
A long run has yet to be made. These are sample plots showing a set of initial conditions and eight subsequent cycles (time steps). The rough appearance of the initial velocity distributions is due to the small mesh size (32 x 32 x 8) which necessitated making the galaxy only about two cells thick. For an actual run, the dimensions will be increased to 64 x 64 x 16. It may prove necessary to generate initial conditions in whole or part by analytic means.

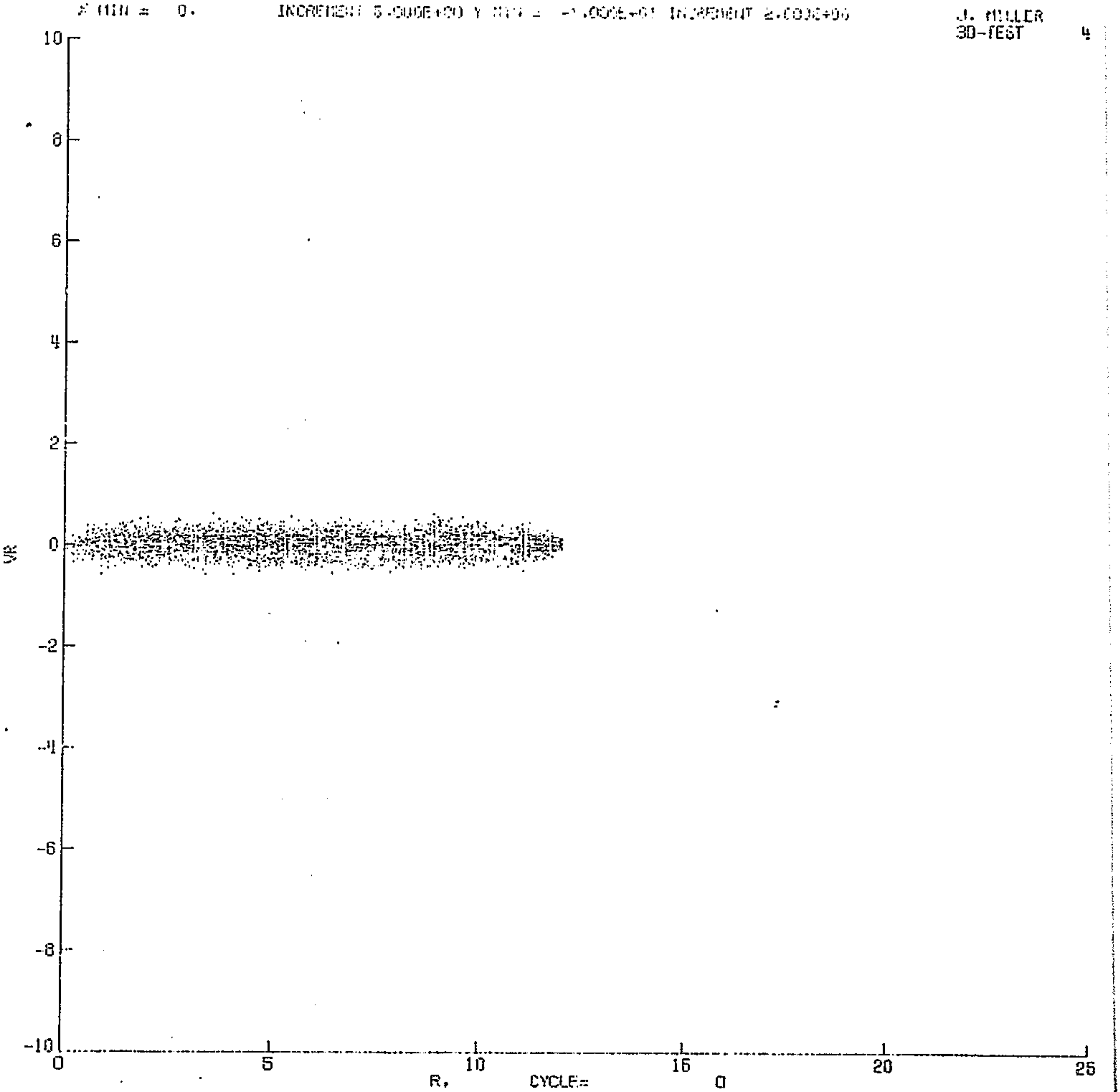
Cycle No.	Plot Type (Stars in Position/Velocity Space)	Page No.
0	y position vs x position	C-2
0	z position vs x position	C-3
0	z position vs y position	C-4
0	r velocity vs r position	C-5
0	azim. velocity vs r position	C-6
0	z velocity vs r position	C-7
1	y position vs x position	C-8
2	z position vs x position	C-9
3	z position vs y position	C-10
4	r velocity vs r position	C-11
5	azim. velocity vs r position	C-12
6	z velocity vs r position	C-13
7	y position vs x position	C-14
8	z position vs x position	C-15

3D-TEST 1

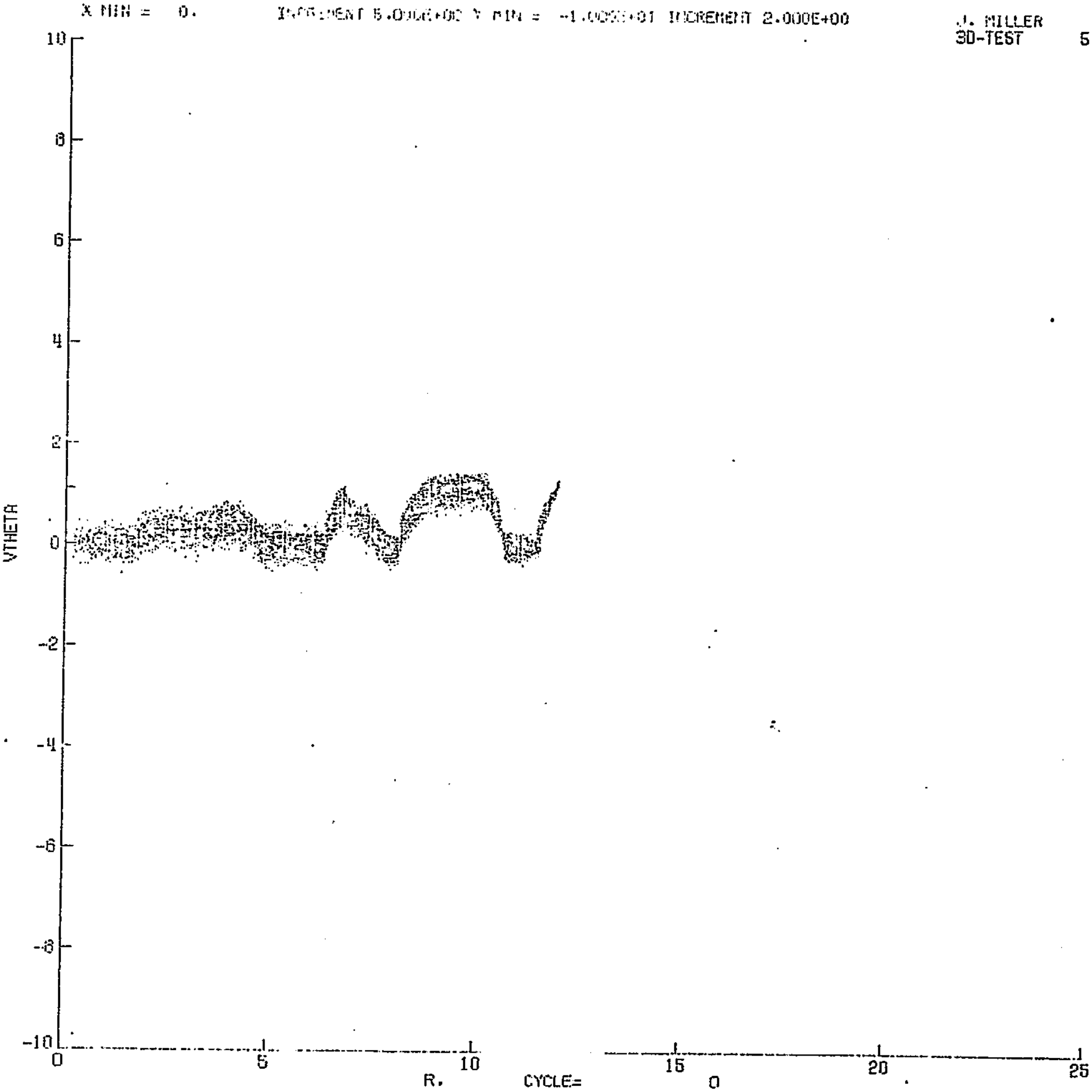



J. MILLER
3D-TEST
Z
40
35
30
25
20
15
10
5
0
-5
-10
-10
-5
0
5
10
15
20
25
30
35
X,
CYCLE=
0

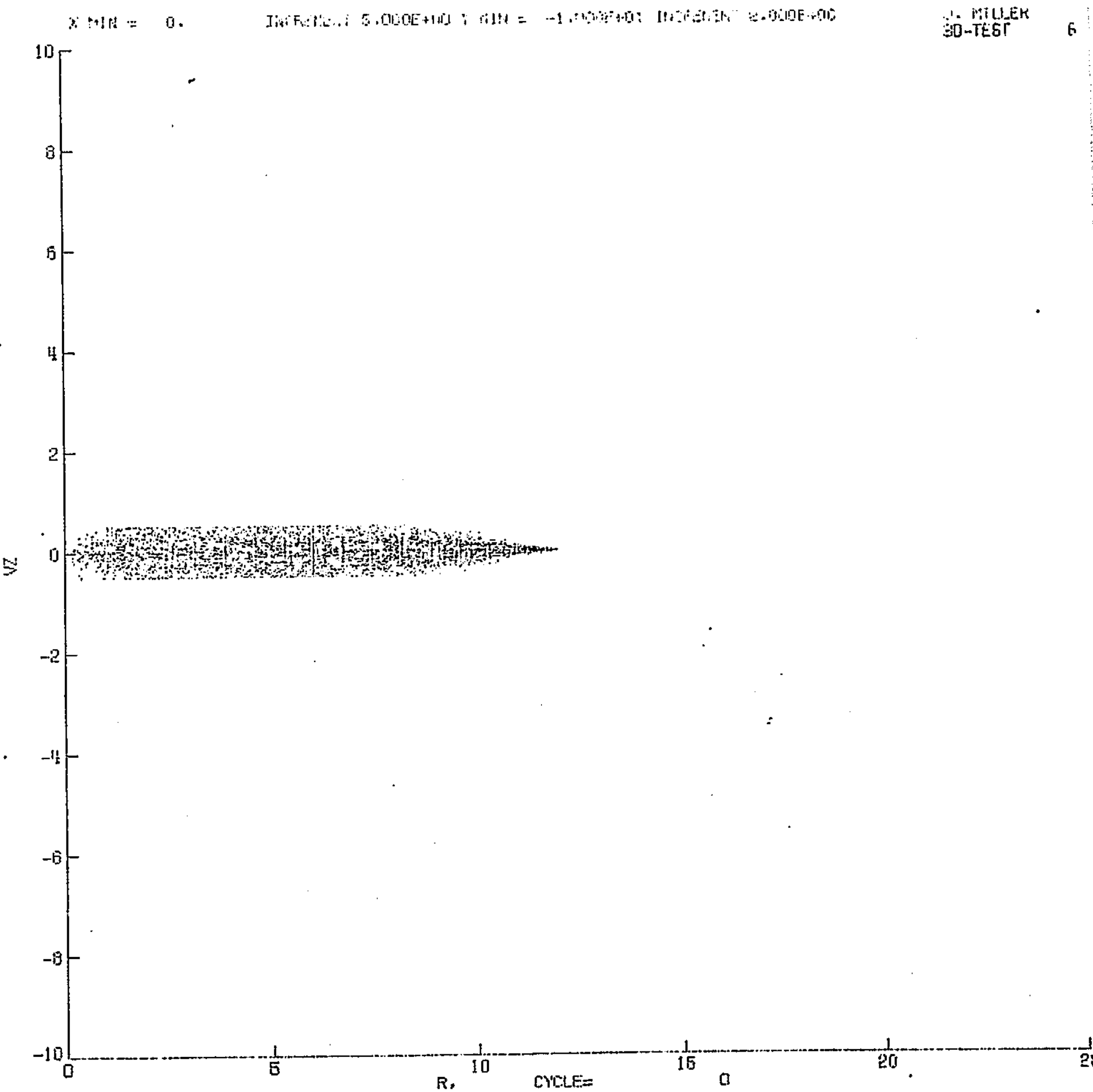

J. MILLER
3D-TEST
3
Z
Y,
CYCLE=
0
40
35
30
25
20
15
10
5
0
-5
-10



ORIGINAL PAGE IS
OF POOR QUALITY

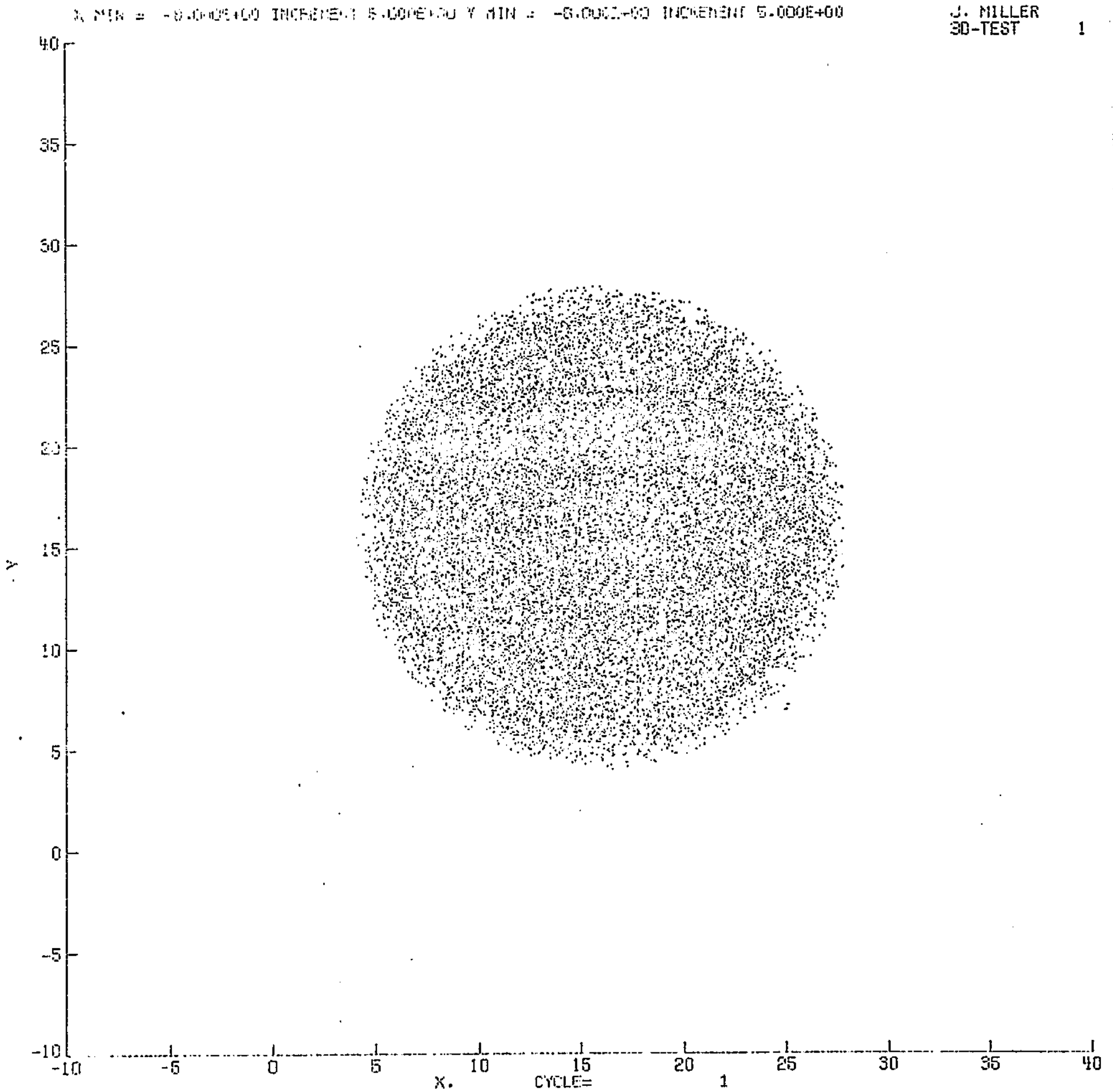


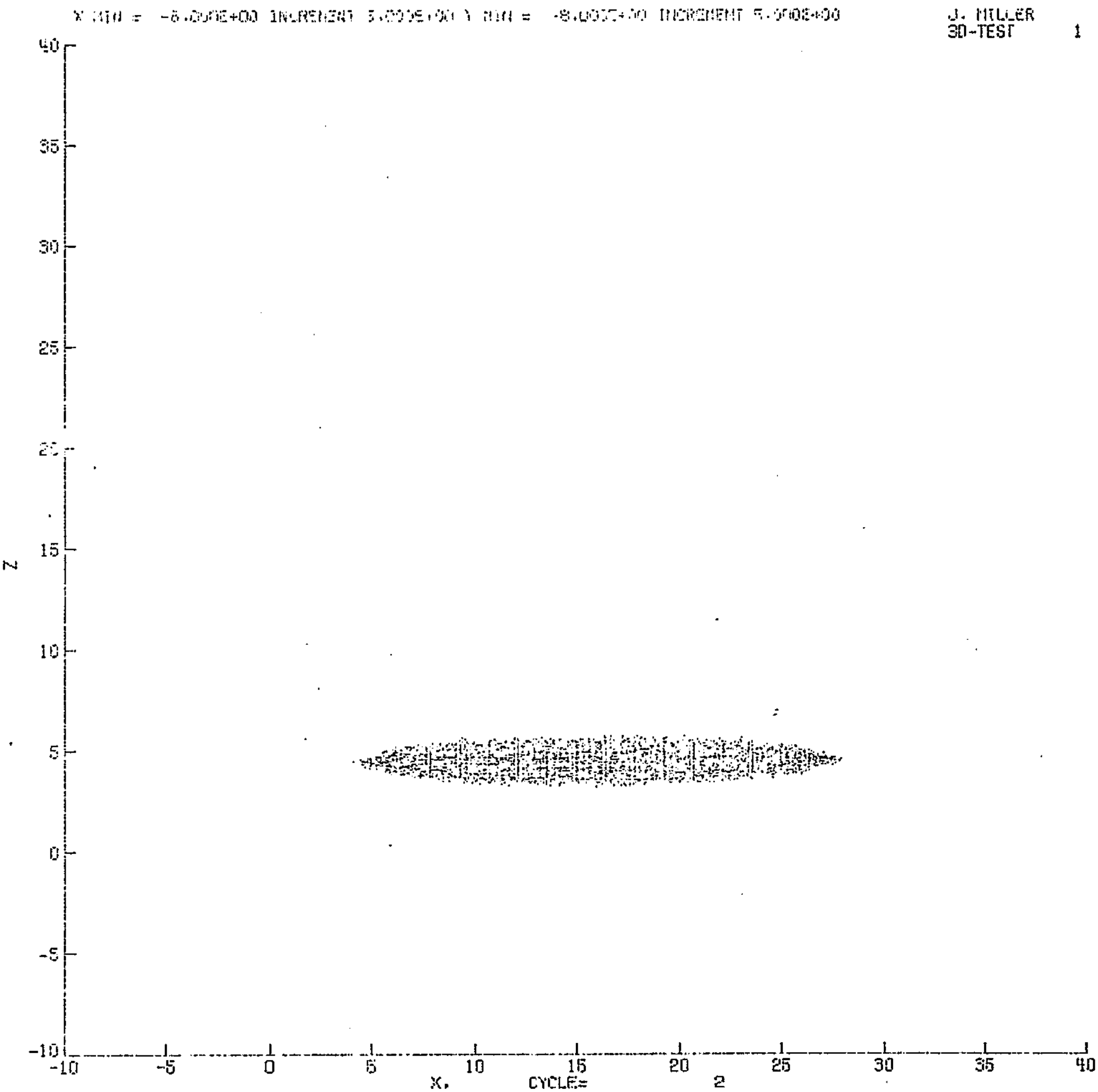
c-6



ORIGINAL PAGE IS
OF POOR QUALITY

C-7

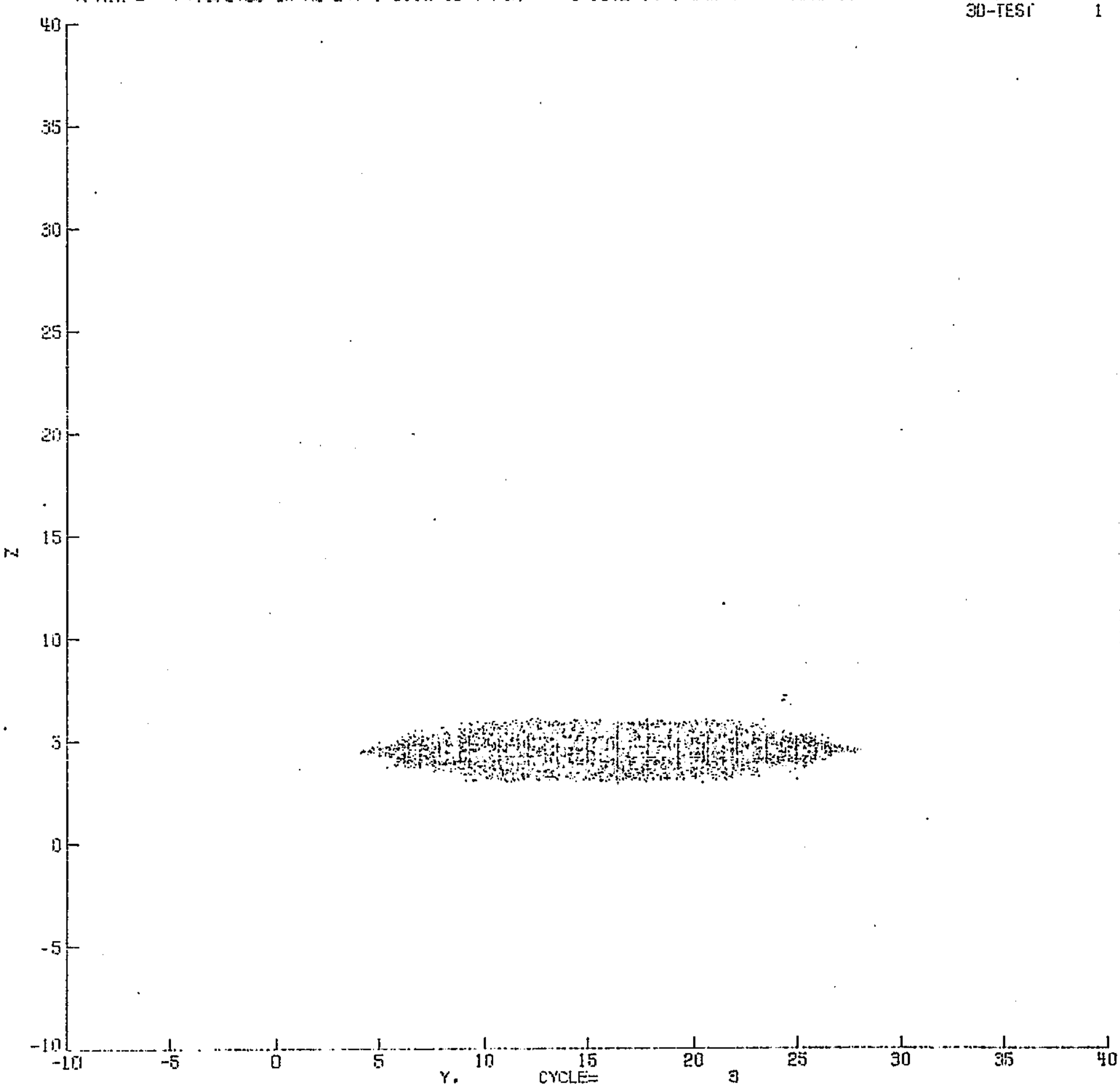

INCREMENT 5.000E+00
J. MILLER
3D-TEST 1
Y.
X.
CYCLE= 1
40
35
30
25
15
10
5
0
-5
-10
-10
-5
0
5
10
15
20
25
30
35
40

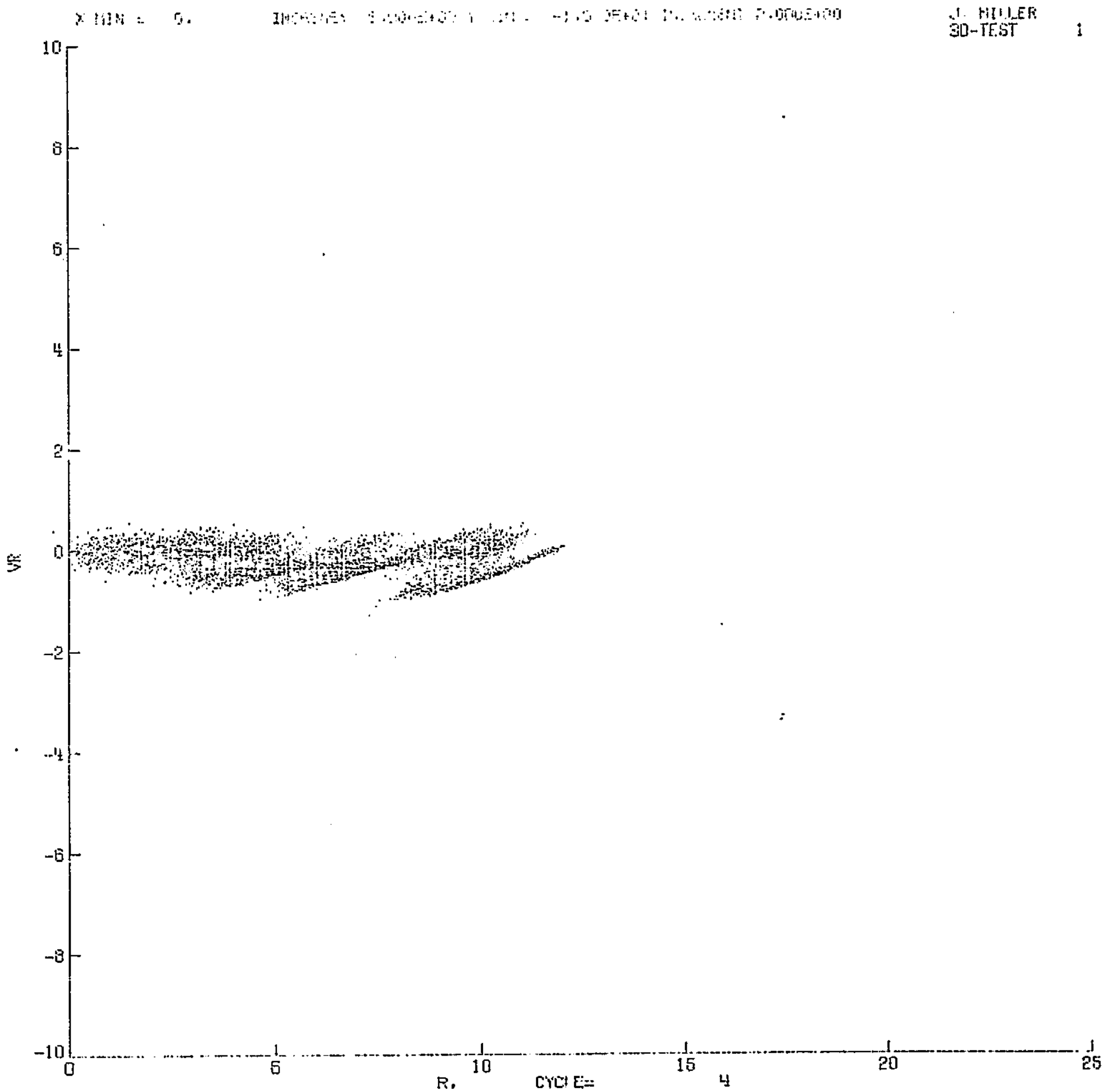


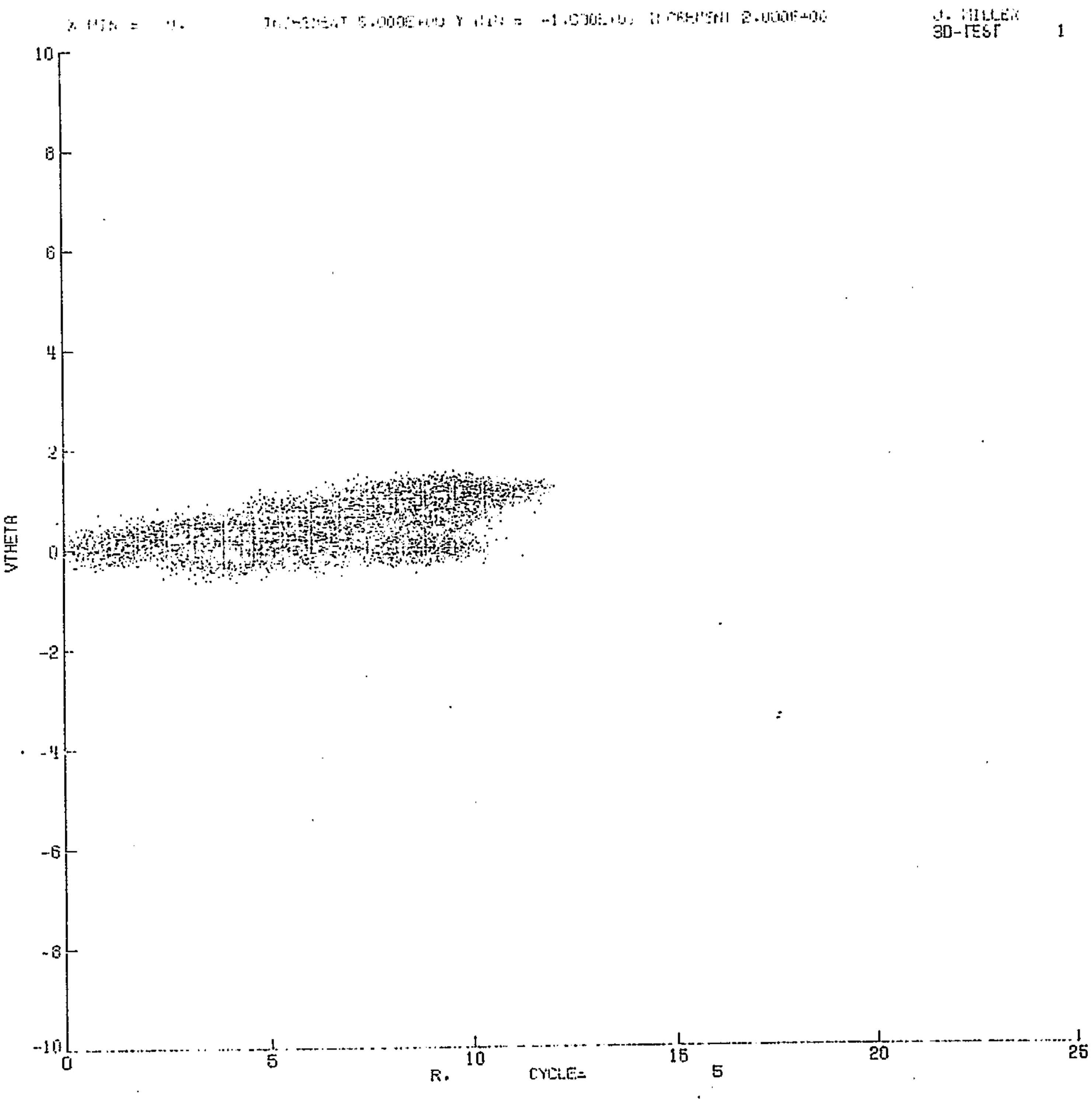
ORIGINAL PAGE IS
OF POOR QUALITY

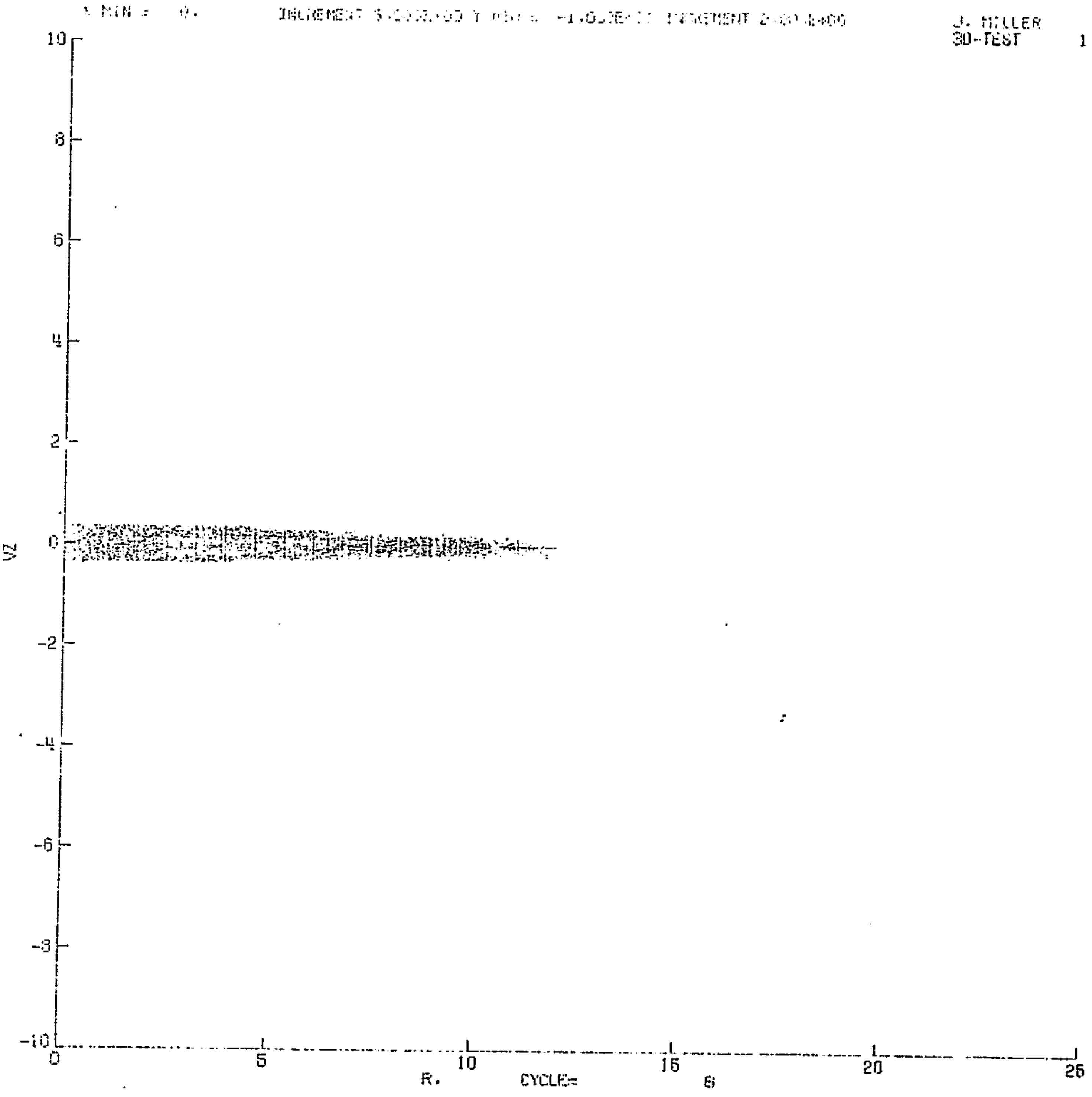
X MIN = -7.000E+00 INCREMENT 5.000E+00 Y MIN = -2.000E+00 INCREMENT 5.000E+00

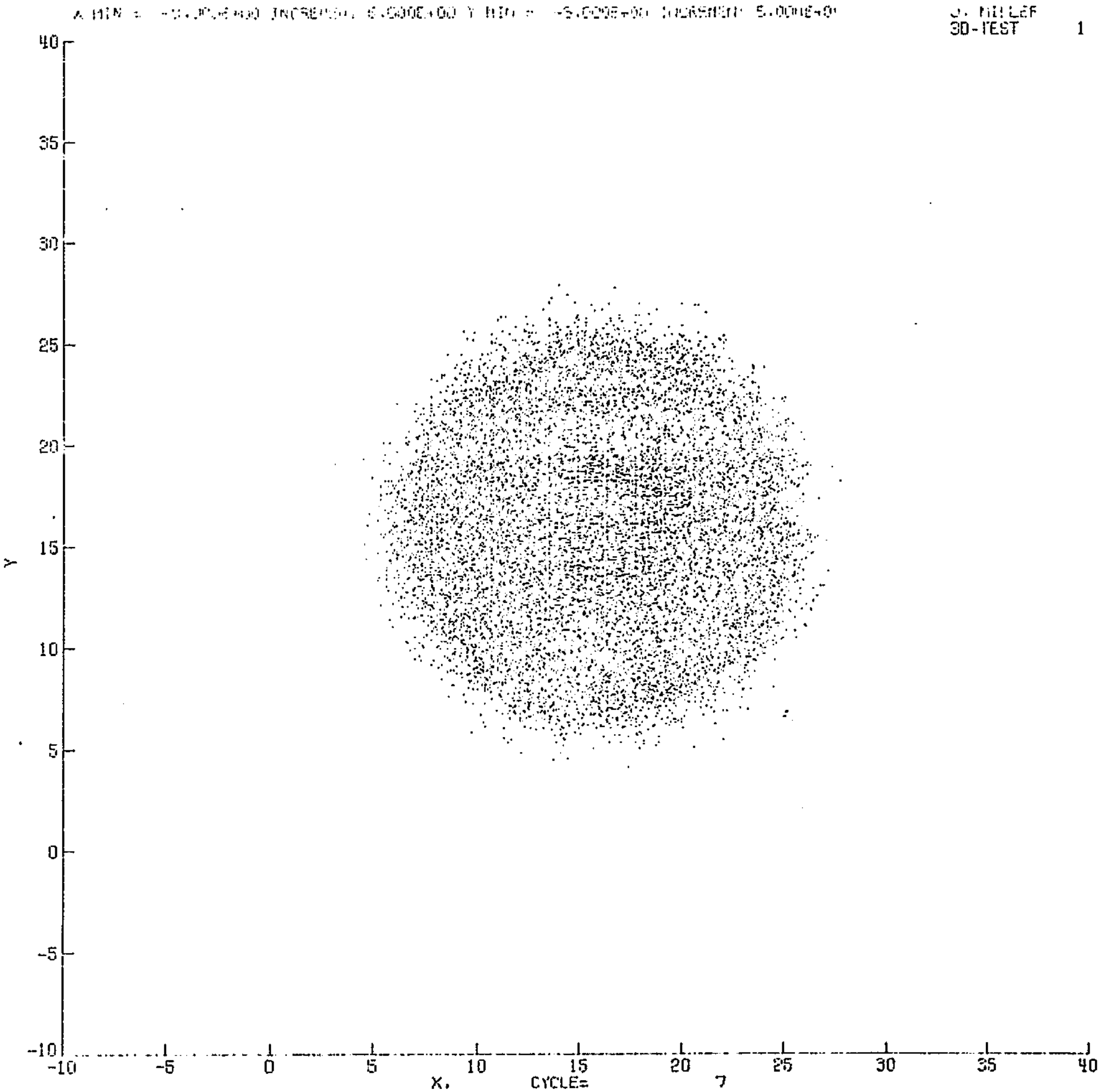
J. MILLER
3D-TEST 1

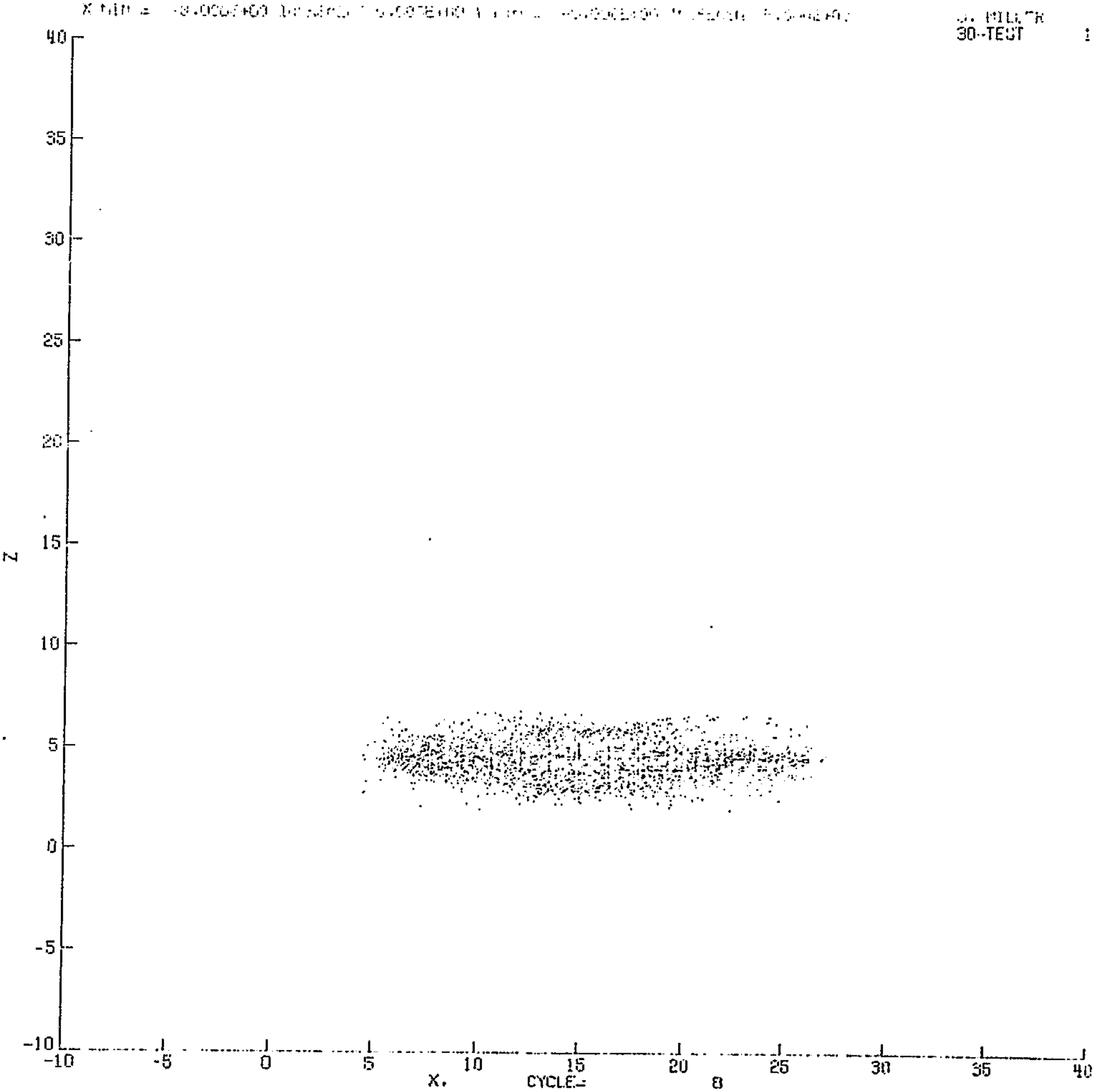



J. MILLER
3D-TEST
1
10
8
6
4
2
0
-2
-4
-6
-8
-10
VR
0
5
10
15
20
25
R.
CYCLE=
4


J. MILLER
3D-TEST 1
10
8
6
4
2
0
-2
-4
-6
-8
-10
VTHETA
0
5
10
15
20
25
R.
CYCLE= 5


J. MILLER
3D-TEST
1
10
8
6
4
2
0
-2
-4
-6
-8
-10
VZ
0
5
10
15
20
25
R.
CYCLE=
6


J. MILLER
3D-TEST 1
Y
X, CYCLE= 7
40
35
30
25
20
15
10
5
0
-5
-10
-10 -5 0 5 10 15 20 25 30 35 40



ORIGINAL PAGE IS
OF POOR QUALITY

APPENDIX D

Listing of the Two-Dimensional Particle-in-Cell Simulator of the Jeans Instability in an Infinite Self-Gravitating Compressible Gas.

Comment cards necessary to make this listing self explanatory will be added later.

```
      OVERLAY(IFILE,0,0)
      PROGRAM GASJ(OUTPUT,TAPE1,TAPE8,TAPE9,TAPE10,TAPE11)
      COMMON/ALLCOM/I2A,ITEST,N,CY,CYY,RHO(33,33)
      COMMON/GASCOM/NPLOTG,NPRINTG,NPRNTMG,NO2SQ,NO2,NO2P1,NO2M1,DMG,
     1    GM1O2,AUO2,AUO4,SAVPE(200),SAVIE(200),SAVKE(200),SAVTE(200),
     2    CYMIN,NBRG,NBSG,JTG,JSG,NMKG,PI,MASK1,ING(2),XMING,XMAXG,
     3    YMING,YMAXG,XPG(2),YPG,ITAPXG,XYL(3),PLP(2),PLT(2),PLM(2),
     4    RCOS,RSIN,XPR(5),YPR(5),XMAX4,XMIN4,YMAX4,YMIN4,SPHIM,SPHIT,
     5    SPHIP,EMIN,EMAX,SCALM,DT,DT2,DT3,DT4,SUMMAS,SUMPX,SUMPY,
     6    SUMKE,SUMIE,SPHI,PL(2)
      INTEGER CY,CYY
      IFILE=5LIFILE
      GPFILE=6LGPFILE

      AGFILE=6LAGFILE
      CALL OVERLAY(IFILE,2,0,6HRECALL)
      ITEST=1
   10 PRINT 5,CYY
    5 FORMAT(1H0,6HCYCLE=,I8)
      CALL SECOND(T1)
      CALL OVERLAY(GPFILE,1,0,6HRECALL)
      CALL SECOND(T2)
      T3=T2-T1
      PRINT 15, T3
   15 FORMAT(12H FIELD TIME=,E16.8)
      CALL OVERLAY(AGFILE,3,0,6HRECALL)
      CALL SECOND(T4)
      T5=T4-T2
      PRINT 35,T5
   35 FORMAT(14H ADVANCE TIME=, E16.8)
      IF(CYY.GE.CY) GO TO 40
      CYY=CYY+1
      GO TO 10
   40 CALL OVERLAY(IFILE,4,0,6HRECALL)
      END
```

```
      OVERLAY(GPFILE,1,0)
      PROGRAM GETPHI
      COMMON/ALLCOM/I2A,ITEST,N,CY,CYY,RHO(33,33)
      COMMON Z(1025),Y(1025)
      DIMENSION G(33,33)
      NO2=N/2
      N21=NO2+1
      IF(ITEST.EQ.0) GO TO 40
      I2B=I2A-1
      RNI=1./(N*N)
      NO2P1=N21
      DO 1 J=1,N
      DO 1 I=1,N
      IF(I.EQ.1.AND.J.EQ.1) GO TO 1
      G(I,J)=RNI/SQRT((I-1.)*(I-1.)+(J-1.)*(J-1.))
    1 CONTINUE
      G(1,1)=G(1,2)
      CALL GETSET(2,I2B)
      DO 3 J=1,NO2P1
      DO 4 I=1,NO2P1
    4 Z(I)=G(I,J)
      CALL FTRANS(2,I2B)
      DO 10 I=1,NO2P1
   10 G(I,J)=Y(I)
    3 CONTINUE
      DO 2 I=1,NO2P1
      DO 19 J=1,NO2P1
   19 Z(J)=G(I,J)
      CALL FTRANS(2,I2B)
      DO 22 J=1,NO2P1
   22 G(I,J)=Y(J)
    2 CONTINUE
      WRITE(1) G
      REWIND 1
   40 READ(1) G
      REWIND 1
      CALL GETSET(3,I2A)
      DO 7 J=1,N
      DO 8 I=1,N
    8 Z(I)=RHO(I,J)
```

```
      CALL FTRANS(3,I2A)
      DO 9 I=1,N
    9 RHO(I,J)=Y(I)
    7 CONTINUE
      DO 25 I=1,N
      DO 26 J=1,N
   26 Z(J)=RHO(I,J)
      CALL FTRANS(3,I2A)
      DO 27 J=1,N
   27 RHO(I,J)=Y(J)
   25 CONTINUE
      DO 11 J=2,NO2
      DO 11 I=2,NO2
      I1=I+NO2
      J1=J+NO2
      RHO(I,J)=RHO(I,J)*G(I,J)
      RHO(I1,J)=RHO(I1,J)*G(I,J)
      RHO(I,J1)=RHO(I,J1)*G(I,J)
      RHO(I1,J1)=RHO(I1,J1)*G(I,J)
   11 CONTINUE
        DO 37 I=2,NO2
      RHO(I+NO2,1)=RHO(I+NO2,1)*G(I,1)
      RHO(I+NO2,N21)=RHO(I+NO2,N21)*G(I,N21)
      RHO(I,1)=RHO(I,1)*G(I,1)
   37 RHO(I,N21)=RHO(I,N21)*G(I,N21)
      DO 38 J=2,NO2
      RHO(1,J)=RHO(1,J)*G(1,J)
      RHO(1,J+NO2)=RHO(1,J+NO2)*G(1,J)
      RHO(N21,J)=RHO(N21,J)*G(N21,J)
   38 RHO(N21,J+NO2)=RHO(N21,J+NO2)*G(N21,J)
      RHO(1,1)=RHO(1,1)*G(1,1)
      RHO(N21,1)=RHO(N21,1)*G(N21,1)
      RHO(1,N21)=RHO(1,N21)*G(1,N21)
      RHO(N21,N21)=RHO(N21,N21)*G(N21,N21)
      CALL GETSET(4,I2A)
      DO 14 J=1,N
      DO 12 I=1,N
   12 Z(I)=RHO(I,J)
      CALL FTRANS(4,I2A)
      DO 13 I=1,N
```

```
13 RHO(I,J)=Y(I)
14 CONTINUE
   DO 28 I=1,N
   DO 29 J=1,N
29 Z(J)=RHO(I,J)
   CALL FTRANS(4,I2A)
   DO 30 J=1,N
30 RHO(I,J)=Y(J)
28 CONTINUE
   DO 81 I=1,N
   RHO(N+1,I)=RHO(1,I)
81 RHO(I,N+1)=RHO(I,1)
     ITEST=0
   RETURN
   END
```

```
      OVERLAY(IFILE,2,0)
      PROGRAM INITGAS
      COMMON/ALLCOM/I2A,ITEST,N,CY,CYY,RHO(33,33)
      COMMON/GASCOM/NPLOTG,NPRINTG,NPRNTMG,NO2SQ,NO2,NO2P1,NO2M1,DMG,
     1    GM1O2,AUO2,AUO4,SAVPE(200),SAVIE(200),SAVKE(200),SAVTE(200),
     2    CYMIN,NBRG,NBSG,JTG,JSG,NMKG,PI,MASK1,ING(2),XMING,XMAXG,
     3    YMING,YMAXG,XPG(2),YPG,ITAPXG,XYL(3),PLP(2),PLT(2),PLM(2),
     4    RCOS,RSIN,XPR(5),YPR(5),XMAX4,XMIN4,YMAX4,YMIN4,SPHIM,SPHIT,
     5    SPHIP,EMIN,EMAX,SCALM,DT,DT2,DT3,DT4,SUMMAS,SUMPX,SUMPY,
     6    SUMKE,SUMIE,SPHI,PL(2)
      DIMENSION A1(32,32),A2(32,32),A3(32,32),A4(32,32),XHOLD(528),
     1    YHOLD(528)
      EQUIVALENCE (RHO(1,1),XHOLD(1)),(RHO(1,17),YHOLD(1))
      INTEGER CY,CYY
      REAL INITIE
C SET INITIAL VALUES
C     SET SIZE WHERE N=NO2=2**I2A IS THE SIZE OF THE ACTIVE MESH
      I2A=5
C     SET TOTAL NUMBER OF GAS PARTICLES
      NBRG=12672
C     SET NUMBER OF GAS PARTICLES PER READ OF GAS PARTICLE FILE
      NBSG=528
C     SET ARTIFICIAL VISCOSITY CONSTANT OF GAS
      AU=0.
C     SET MASS PER GAS PARTICLE
      XMG=1.
C     SET THE INITIAL SPECIFIC INTERNAL ENERGY
      INITIE=5.
C     SET TOTAL NUMBER OF TIME STEPS
      CY=150
C     SET NUMBER OF POINTS PER GAS PLOT
      NPG=NBRG
C     SET PERIOD OF GAS LONG PRINTING
      NPRINTG=50
C     SET PERIOD OF GAS DENSITY SHORT PRINTING
      NPRNTMG=25
C     SET PERIOD OF GAS PLOTTING
      NPLOTG=25
C     SET RATIO OF SPECIFIC HEAT AT CONSTANT VOLUME TO SPECIFIC HEAT AT CONSTANT
C     PRESSURE. (GAMMA MUST BE GREATER THAN OR EQUAL TO 1. GAMMA IS EQUAL TO 2.0
```

```
C     FOR NORMAL SIMULATION OF MONATOMIC GAS IN TWO DIMENSIONS. GAMMA MAY BE SET
C     EQUAL TO 1.0 FOR A SIMULATION WITHOUT PRESSURE TERMS.)
      GAMMA=2.0
C     SET ITAPE=1 TO START RUN. SET ITAPE=2 TO CONTINUE RUN WITH PICK UP TAPE
      ITAPE=1
C SET CONSTANTS
      PI=3.1415926536
      MASK1=07777777770000000000
      JTG=9
      JSG=10
      GM1O2=(GAMMA-1.)/2.
      AUO2=AU/2.
      AUO4=AU/4.
      NMKG=NPG-NBSG
      N=2**I2A
C     NOTE FOLLOWING DEFINITION
      NO2=N
      NO2P1=NO2+1
      NO2M1=NO2-1
      NO2SQ=NO2*NO2
      NO4=NO2/2
      GMU=XMG*NBRG/NO2SQ
      TAU=1/SQRT(GMU)
      DT=TAU/50.
      DT2=DT*DT
      DT3=DT2*DT
      DT4=DT2*DT2
C SET PLOTTING SPECIFICATIONS
      ING(1)=10H2D-GAS
      ING(2)=10H
      XMING=-10
      XMAXG=40
      YMING=-10
      YMAXG=40
      XPG(1)=10HX, CYCLE=
      YPG=10H    Y
      ITAPXG=6LTAPE23
      XYL(1)=10HX-Y PLANE,
      XYL(2)=10H  CYCLE =
      PL(1)=10HPOTENTIAL*
      PLP(1)=10H PRESSURE*
```

```
      PLT(1)=10H      TEMP*
      PLM(1)=10H  DENSITY*
      ANG=PI/3.
      RCOS=COS(ANG)
      RSIN=SIN(ANG)
      IC=NO2+1
      XPR(1)=1.+NO2*RCOS
      XPR(2)=1.
      XPR(3)=1.+NO2
      XPR(4)=IC+NO2*RCOS
      XPR(5)=XPR(1)
      YPR(1)=1.+NO2*RSIN
      YPR(2)=1.
      YPR(3)=1.
      YPR(4)=1.+NO2*RSIN
      YPR(5)=YPR(1)
      XMAX4=IC+NO2*RCOS
      XMIN4=-1.
      YMIN4=-1.
      YMAX4=XMAX4
      HT=IC+NO2*RCOS-NO2*RSIN
      PMAX=2.*NBRG*XMG*DT2/NO4
      PMAXM=10.*GMU*DT2
      PMAXT=50.*INITIE*DT2
      PMAXP=.2*PMAXM*PMAXT
      SPHI=HT/PMAX
      ENCODE(10,25,PL(2)) SPHI
      SPHIM=HT/PMAXM
      ENCODE(10,25,PLM(2)) SPHIM
   25 FORMAT(F10.3)
      SPHIT=HT/PMAXT
      ENCODE(10,25,PLT(2)) SPHIT
      SPHIP=HT/PMAXP
      ENCODE(10,30,PLP(2)) SPHIP
   30 FORMAT(F10.0)
      EMIN=-((XMG*NBRG)**2)/NO4
      EMAX=-EMIN
      SCALM=GMU*DT2/7.
      INITIE=INITIE*DT2
      DMG=XMG*D↑2
      DMG=DMG.AND.MASK1
```

```
      IF(ITAPE.EQ.2) GO TO 350
      CYY=1
      DO 105 I=1,NO2
      DO 105 J=1,NO2
      A4(I,J)=0.
  105 CONTINUE
      X=URAN(7654321.)
      NS2=0
  110 DO 130 IS=1,NBSG
      X=NO2*URAN(0.)+.5
      Y=NO2*URAN(0.)+.5
      IX=X+.5
      JY=Y+.5
      IF(IX.GE.1) GO TO 115
      X=X+NO2
      IX=X+.5
      GO TO 120
  115 IF(IX.LE.NO2) GO TO 120
      X=X-NO2
      IX=X+.5
  120 IF(JY.GE.1) GO TO 125
      Y=Y+NO2
      JY=Y+.5
      GO TO 128
  125 IF(JY.LE.NO2) GO TO 128
      Y=Y-NO2
      JY=Y+.5
  128 A4(IX,JY)=A4(IX,JY)+DMG
      XHOLD(IS)=X
      YHOLD(IS)=Y
  130 CONTINUE
      WRITE(9) XHOLD,YHOLD
      NS2=NS2+NBSG
      IF(NS2.LT.NBRG) GO TO 110
      REWIND 9
      SUMMAS=0
      SUMIE=DMG*NBRG*INITIE
      SUMKE=0.
      SUMPX=0.
      SUMPY=0.
```

```
      DO 140 I=1,NO2
      DO 140 J=1,NO2
      IF(A4(I,J).EQ.0.) GO TO 135
      SUMMAS=SUMMAS+A4(I,J)
      A3(I,J)=INITIE
      GO TO 138
  135 A3(I,J)=0
  138 A1(I,J)=0.
      A2(I,J)=0.
      RHO(I,J)=.5*A4(I,J)
  140 CONTINUE
      DO 150 I=1,NO2P1
      RHO(I,NO2P1)=0.
      RHO(NO2P1,I)=0.
  150 CONTINUE
      GO TO 400
C STATEMENTS 350 TO 400 ENABLE RUN TO BE CONTINUED WITH PICK UP TAPE.
  350 NS2=0
  355 READ(11) XHOLD,YHOLD
      WRITE(9) XHOLD,YHOLD
      NS2=NS2+NBSG
      IF(NS2.LT.NBRG) GO TO 355
      REWIND 9
      READ(11)A1,A2,A3,A4,CYY,SUMMAS,SUMPX,SUMPY,SUMKE,SUMIE
      CY=CY+CYY
      CYY=CYY+1
      DO 360 I=1,NO2
      DO 360 J=1,NO2
      RHO(I,J)=.5*A4(I,J)
  360 CONTINUE
      DO 370 I=1,NO2P1
      RHO(I,NO2P1)=0.
      RHO(NO2P1,I)=0.
  370 CONTINUE
  400 CYMIN=CYY-1
      CALL EVICT(6LTAPE11)
C WRITE U,V,I, AND M ONTO TAPE8
      WRITE(8)A1,A2,A3,A4
      REWIND 8
      RETURN
      END
```

```
      OVERLAY(AGFILE,3,0)
      PROGRAM ADVGAS
      COMMON/ALLCOM/I2A,ITEST,N,CY,CYY,RHO(33,33)
      COMMON/GASCOM/NPLOTG,NPRINTG,NPRNTMG,NO2SQ,NO2,NO2P1,NO2M1,DMG,
     1    GM1O2,AUO2,AUO4,SAVPE(200),SAVIE(200),SAVKE(200),SAVTE(200),
     2    CYMIN,NBRG,NBSG,JTG,JSG,NMKG,PI,MASK1,ING(2),XMING,XMAXG,
     3    YMING,YMAXG,XPG(2),YPG,ITAPXG,XYL(3),PLP(2),PLT(2),PLM(2),
     4    RCOS,RSIN,XPR(5),YPR(5),XMAX4,XMIN4,YMAX4,YMIN4,SPHIM,SPHIT,
     5    SPHIP,EMIN,EMAX,SCALM,DT,DT2,DT3,DT4,SUMMAS,SUMPX,SUMPY,
     6    SUMKE,SUMIE,SPHI,PL(2)
      DIMENSION A1(32,32),A2(32,32),A3(32,32),A4(32,32),A1HORZ(32),
     1    A1VERT(32),A2HORZ(32),A2VERT(32),A3HORZ(32),A3VERT(32),
     2    XHOLD(528),YHOLD(528),PX(32,32),PY(32,32),ETOTAL(32,32)
      EQUIVALENCE (RHO(1,1),XHOLD(1)),(RHO(1,17),YHOLD(1))
      INTEGER CYY,CY,Q
      REAL MC,MIP,MIM,MJP,MJM,IE,KE,IC
      REAL IEIP,IEIPJP,IEIPJM,IEJP,IEJM,IEIM,IEIMJP,IEIMJM
C IF GAS PLOTTING IS TO BE DONE THIS CYCLE SET IPLOTG=1, OTHERWISE SET IPLOTG=0
      IPLOTG=0
      IF(CYY-CYY/NPLOTG*NPLOTG.EQ.0.OR.CYY.EQ.1) IPLOTG=1
      CALL PSEUDO
      ENCODE(10,21,XPG(2)) CYY
      ENCODE(10,21,XYL(3)) CYY
   21 FORMAT(I10)
      IF(IPLOTG.NE.1) GO TO 120
C MAKE A CONTOUR PLOT OF THE GRAVITATIONAL POTENTIAL
      CALL DDIPLT(0,ING,5,XPR,YPR,XMIN4,XMAX4,YMIN4,YMAX4,3,XYL(1),
     1    2,PL(1),14,ITAPXG)
      DX=0.
      DY=1.
      DO 50 J=2,NO2
      K=0
      DX=DX+RCOS
      DY=DY+RSIN
      DU=0.
      DV=1.
      DO 40 I=2,NO2
      K=K+1
      DU=DU+RCOS
      DV=DV+RSIN
```

```
      A1HORZ(K)=J+DU
      A1VERT(K)=SPHI*RHO(J,I)+DV
      A2HORZ(K)=I+DX
   40 A2VERT(K)=SPHI*RHO(I,J)+DY
      CALL DDIPLT(0,ING,K,A2HORZ,A2VERT,XMIN4,XMAX4,YMIN4,YMAX4,
     1    3,XYL(1),2,PL(1),14,ITAPXG)
      CALL DDIPLT(0,ING,K,A1HORZ,A1VERT,XMIN4,XMAX4,YMIN4,YMAX4,
     1    3,XYL(1),2,PL(1),14,ITAPXG)
   50 CONTINUE
      K=0
      DO 60 I=2,NO2
      XI=I
      K=K+1
      A2HORZ(K)=XI+RCOS
      A2VERT(K)=1.+RSIN
   60 CONTINUE
      CALL DDIPLT(1,ING,K,A2HORZ,A2VERT,XMIN4,XMAX4,YMIN4,YMAX4,
     1    3,XYL(1),2,PL(1),13,ITAPXG)
C IF GAS PRINTING IS TO BE DONE THIS CYCLE SET IPRINTG=1,OTHERWISE SET IPRINTG=0
  120 IPRINTG=0
      IF(CYY-CYY/NPRINTG*NPRINTG.EQ.0.OR.CYY.EQ.1)  IPRINTG=1
C IF GAS DENSITY SHORT PRINTING IS TO BE DONE THIS CYCLE SET IPRNTMG=1,
C OTHERWISE SET IPRNTMG=0.
      IPRNTMG=0
      IF(CYY-CYY/NPRNTMG*NPRNTMG.EQ.0.OR.CYY.EQ.1) IPRNTMG=1
C READ U,V,I,AND M FROM TAPE8 INTO A1,A2,A3, AND A4 RESPECTIVELY
      READ(8) A1,A2,A3,A4
      REWIND 8
      NHIVEL=0
      SUMPE=0.
      DO 200 J=1,NO2
      DO 200 I=1,NO2
      MC=A4(I,J)
      IF(MC.LT.DMG) GO TO 183
      SUMPE=SUMPE+MC*RHO(I,J)
      UC=A1(I,J)
      VC=A2(I,J)
      PC=A3(I,J)*MC
      IF(I.EQ.1) GO TO 130
      IF(I.EQ.NO2) GO TO 140
```

```
      UIP=A1(I+1,J)
      MIP=A4(I+1,J)
      PIP=A3(I+1,J)*A4(I+1,J)
      UIM=A1(I-1,J)
      MIM=A4(I-1,J)
      PIM=A3(I-1,J)*A4(I-1,J)
      GX=RHO(I+1,J)-RHO(I-1,J)
      GO TO 150
  130 UIP=A1(2,J)
      MIP=A4(2,J)
      PIP=A3(2,J)*A4(2,J)
      UIM=A1(NO2,J)
      MIM=A4(NO2,J)
      PIM=A3(NO2,J)*A4(NO2,J)
      GX=RHO(2,J)-RHO(NO2,J)
      GO TO 150
  140 UIP=A1(1,J)
      MIP=A4(1,J)
      PIP=A3(1,J)*A4(1,J)
      UIM=A1(NO2M1,J)
      MIM=A4(NO2M1,J)
      PIM=A3(NO2M1,J)*A4(NO2M1,J)
      GX=RHO(1,J)-RHO(NO2M1,J)
  150 IF(J.EQ.1) GO TO 160
      IF(J.EQ.NO2) GO TO 170
      VJP=A2(I,J+1)
      MJP=A4(I,J+1)
      PJP=A3(I,J+1)*A4(I,J+1)
      VJM=A2(I,J-1)
      MJM=A4(I,J-1)
      PJM=A3(I,J-1)*A4(I,J-1)
      GY=RHO(I,J+1)-RHO(I,J-1)
      GO TO 175
  160 VJP=A2(I,2)
      MJP=A4(I,2)
      PJP=A3(I,2)*A4(I,2)
      VJM=A2(I,NO2)
      MJM=A4(I,NO2)
      PJM=A3(I,NO2)*A4(I,NO2)
      GY=RHO(I,2)-RHO(I,NO2)
      GO TO 175
```

```
170 VJP=A2(I,1)
    MJP=A4(I,1)
    PJP=A3(I,1)*A4(I,1)
    VJM=A2(I,NO2M1)
    MJM=A4(I,NO2M1)
    PJM=A3(I,NO2M1)*A4(I,NO2M1)
    GY=RHO(I,1)-RHO(I,NO2M1)
175 QIP=MIP*(UC-UIP)/(MC+MIP)
    IF(QIP.LE.0.) QIP=0.
    QIM=MIM*(UIM-UC)/(MC+MIM)
    IF(QIM.LE.0.) QIM=0.
    GM=GM102/MC
    IF(PC.LT.0.) PC=0.
    PIPB=PC+PIP
    PIMB=PC+PIM
    PJPB=PC+PJP
    PJMB=PC+PJM
    IF(PIP.LE.0.) PIPB=0.
    IF(PIM.LE.0.) PIMB=0.
    IF(PJP.LE.0.) PJPB=0.
    IF(PJM.LE.0.) PJMB=0.
    UT=UC+GM*(PIMB-PIPB)+AUO2*(QIM-QIP)+GX
    QJP=MJP*(VC-VJP)/(MC+MJP)
    IF(QJP.LE.0.) QJP=0.
    QJM=MJM*(VJM-VC)/(MC+MJM)
    IF(QJM.LE.0.) QJM=0.
    VT=VC+GM*(PJMB-PJPB)+AUO2*(QJM-QJP)+GY
    UIPB=UC+UIP
    UIMB=UC+UIM
    VJPB=VC+VJP
    VJMB=VC+VJM
    IE=A3(I,J)+.5*(UC*UC+VC*VC+(UC+UT)*GX+(VC+VT)*GY)
   1    +AUO4*(QIM*UIMB-QIP*UIPB+QJM*VJMB-QJP*VJPB)
   2    +GM*(PIMP*UIMB-PIPB*UIPB+PJMB*VJMB-PJPB*VJPB)/2.
   3   -.5*(UT*UT+VT*VT)
    IE=MC*IE
    IF(MC.LT.5.0*DMG) GO TO 180
    IF(UT.GE..375.OR.VT.GE..375) NHIVEL=NHIVEL+1
180 IF(ABS(UT).LT.1.0.AND.ABS(VT).LT.1.0) GO TO 185
    CY=CYY
```

```
      IPLOTG=1
      IPRINTG=1
      IPRNTMG=1
      PRINT 182,UT,VT,I,J
  182 FORMAT(1H0*UT=*E16.8*    AND VT=*E16.8*    FOR I=*I2*   AND J=*I2)
      GO TO 185
  183 UT=99.
      VT=99.
      IE=0.
  185 PX(I,J)=UT
      PY(I,J)=VT
      ETOTAL(I,J)=IE
  200 CONTINUE
      IF(NHIVEL.GT.0) PRINT 205,NHIVEL
  205 FORMAT(1H0*NUMBER OF CELLS CONTAINING GREATER THAN 5 PARTICLES AND
     1 HAVING SCALED VELOCITY COMPONENT(S) GREATER THAN .375=   *I5)
      DO 210 I=1,NO2
      DO 210 J=1,NO2
      A1(I,J)=PX(I,J)
      PX(I,J)=0.
      A2(I,J)=PY(I,J)
      PY(I,J)=0.
      A3(I,J)=ETOTAL(I,J)
      ETOTAL(I,J)=0.
  210 CONTINUE
      DO 222 I=1,NO2
      DO 222 J=1,NO2
  222 RHO(I,J)=0.
      DO 224 I=2,NO2M1
      DO 224 J=2,NO2M1
      IF(A3(I,J).GE.0.) GO TO 224
      IE=A3(I,J)
      IEIP=A3(I+1,J)
      IF(IEIP.LT.0.) IEIP=0.
      IEIPJP=A3(I+1,J+1)
      IF(IEIPJP.LT.0.) IEIPJP=0.
      IEIPJM=A3(I+1,J-1)
      IF(IEIPJM.LT.0.) IEIPJM=0.
      IEJP=A3(I,J+1)
      IF(IEJP.LT.0.) IEJP=0.
```

```
      IEJM=A3(I,J-1)
      IF(IEJM.LT.0.) IEJM=0.
      IEIM=A3(I-1,J)
      IF(IEIM.LT.0.) IEIM=0.
      IEIMJP=A3(I-1,J+1)
      IF(IEIMJP.LT.0.) IEIMJP=0.
      IEIMJM=A3(I-1,J-1)
      IF(IEIMJM.LT.0.) IEIMJM=0.
      SUM=IEIP+IEIPJP+IEIPJM+IEJP+IEJM+IEIM+IEIMJP+IEIMJM
      IF(SUM.LT.-IE) GO TO 223
      QUOTNT=IE/SUM
      RHO(I+1,J)=RHO(I+1,J)+QUOTNT*IEIP
      RHO(I+1,J+1)=RHO(I+1,J+1)+QUOTNT*IEIPJP
      RHO(I+1,J-1)=RHO(I+1,J-1)+QUOTNT*IEIPJM
      RHO(I,J+1)=RHO(I,J+1)+QUOTNT*IEJP
      RHO(I,J-1)=RHO(I,J-1)+QUOTNT*IEJM
      RHO(I-1,J)=RHO(I-1,J)+QUOTNT*IEIM
      RHO(I-1,J+1)=RHO(I-1,J+1)+QUOTNT*IEIMJP
      RHO(I-1,J-1)=RHO(I-1,J-1)+QUOTNT*IEIMJM
  223 RHO(I,J)=RHO(I,J)-IE
  224 CONTINUE
      DO 228 I=1,NO2
      DO 228 J=1,NO2
      IF(A4(I,J).LT.DMG) GO TO 226
      A3(I,J)=(A3(I,J)+RHO(I,J))/A4(I,J)
      IF(A3(I,J).LT.0.) A3(I,J)=0.
      GO TO 227
  226 A3(I,J)=0.
  227 A4(I,J)=0.
  228 CONTINUE
      PXX=SUMPX/DT3
      PYY=SUMPY/DT3
      TM=SUMMAS/DT2
      PE=-SUMPE/DT4
      IE=SUMIE/DT4
      KE=SUMKE/DT4
      TE=PE+IE+KE
      PRINT 232
  232 FORMAT(1H0 *THE VALUES ON THE NEXT TWO LINES ARE FOR THE END OF TH
     1E LAST CYCLE*)
```

```
      PRINT 234,PXX,PYY,TM
  234 FORMAT(1H *PX=*E14.7*    PY=*E14.7*    TM=*E14.7)
      PRINT 236,PE,IE,KE,TE
  236 FORMAT(1H ,*PE=*E14.7*    IE=*E14.7*     KE=*E14.7*    TE=*E14.7)
      ICYPLT=CYY-CYMIN
      SAVPE(ICYPLT)=PE
      SAVIE(ICYPLT)=IE
      SAVKE(ICYPLT)=KE
      SAVTE(ICYPLT)=TE
      PRINT 250
  250 FORMAT(1H0,47HLAST PARTICLE OF EACH SET OF NBSG GAS PARTICLES)
      PRINT 260
  260 FORMAT(1H *NUMBER       X                    Y                   UEFF
     1             VEFF*)
      NS2=0
      NUMBER=1
  270 READ(JTG) XHOLD,YHOLD
      DO 350 IS=1,NBSG
      X=XHOLD(IS)
      Y=YHOLD(IS)
      IXOLD=X+.5
      JYOLD=Y+.5
      IX=X
      JY=Y
      DX=X-IX
      DY=Y-JY
      D11=1-DY-DX+DX*DY
      D12=DY*(1-DX)
      D21=DX*(1-DY)
      D22=DX*DY
      IF(IX.LT.1) IX=IX+NO2
      IF(JY.LT.1) JY=JY+NO2
      IXP1=IX+1
      JYP1=JY+1
      IF(IXP1.GT.NO2) IXP1=IXP1-NO2
      IF(JYP1.GT.NO2) JYP1=JYP1-NO2
      IF(A1(IX,JY).GT.98.) D11=0.
      IF(A1(IX,JYP1).GT.98.) D12=0.
      IF(A1(IXP1,JY).GT.98.) D21=0.
      IF(A1(IXP1,JYP1).GT.98.) D22=0.
```

```
      DSUM=D11+D12+D21+D22
      UEFF=(D11*A1(IX,JY)+D12*A1(IX,JYP1)+D21*A1(IXP1,JY)
     1      +D22*A1(IXP1,JYP1))/DSUM
      VEFF=(D11*A2(IX,JY)+D12*A2(IX,JYP1)+D21*A2(IXP1,JY)
     1      +D22*A2(IXP1,JYP1))/DSUM
      X=X+UEFF
      Y=Y+VEFF
      IXNEW=X+.5
      JYNEW=Y+.5
      IF(IXNEW.GE.1) GO TO 310
      X=X+NO2
      IXNEW=X+.5
      GO TO 320
  310 IF(IXNEW.LE.NO2) GO TO 320
      X=X-NO2
      IXNEW=X+.5
  320 IF(JYNEW.GE.1) GO TO 330
      Y=Y+NO2
      JYNEW=Y+.5
      GO TO 340
  330 IF(JYNEW.LE.NO2) GO TO 340
      Y=Y-NO2
      JYNEW=Y+.5
  340 DPX=DMG*A1(IXOLD,JYOLD)
      DPY=DMG*A2(IXOLD,JYOLD)
      DE=DMG*(A3(IXOLD,JYOLD)+.5*(A1(IXOLD,JYOLD)*A1(IXOLD,JYOLD)
     1      +A2(IXOLD,JYOLD)*A2(IXOLD,JYOLD)))
      PX(IXNEW,JYNEW)=PX(IXNEW,JYNEW)+DPX
      PY(IXNEW,JYNEW)=PY(IXNEW,JYNEW)+DPY
      ETOTAL(IXNEW,JYNEW)=ETOTAL(IXNEW,JYNEW)+DE
      A4(IXNEW,JYNEW)=A4(IXNEW,JYNEW)+DMG
      XHOLD(IS)=X
  350 YHOLD(IS)=Y
      WRITE(JSG) XHOLD,YHOLD
      IF(NUMBER.GT.10) GO TO 365
      PRINT 360,NUMBER,X,Y,UEFF,VEFF
  360 FORMAT(1H ,I6,4E16.8)
  365 NUMBER=NUMBER+1
      IF(IPLOTG.EQ.0) GO TO 370
      Q=0
```

```
      IF(NS2.EQ.NMKG) Q=1
      CALL DDIPLT(Q,ING,NBSG,XHOLD,YHOLD,XMING,XMAXG,YMING,YMAXG,
     1     2,XPG,1,YPG,13,ITAPXG)
  370 NS2=NS2+NBSG
      IF(NS2.LT.NBRG) GO TO 270
      REWIND JTG
      REWIND JSG
C EXCHANGE TAPE NUMBERS OF JTG AND JSG
      JSAVE=JSG
      JSG=JTG
      JTG=JSAVE
C COMPUTE .5*DENSITY AND STORE IN RHO
C COMPUTE UP AND STORE IN A1. COMPUTE VP AND STORE IN A2. COMPUTE IP AND STORE
C IN A3.
      SUMMAS=0.
      SUMPX=0.
      SUMPY=0.
      SUMKE=0.
      SUMIE=0.
      DO 380 I=1,NO2
      DO 380 J=1,NO2
      MC=A4(I,J)
      IF(MC.EQ.0.) GO TO 376
      RHO(I,J)=.5*MC
      SUMMAS=SUMMAS+MC
      SUMPX=SUMPX+PX(I,J)
      SUMPY=SUMPY+PY(I,J)
      A1(I,J)=PX(I,J)/MC
      A2(I,J)=PY(I,J)/MC
      KE=.5*MC*(A1(I,J)*A1(I,J)+A2(I,J)*A2(I,J))
      SUMKE=SUMKE+KE
      IE=ETOTAL(I,J)-KE
      SUMIE=SUMIE+IE
      A3(I,J)=IE/MC
      GO TO 380
  376 A1(I,J)=0.
      A2(I,J)=0.
      A3(I,J)=0.
      RHO(I,J)=0.
  380 CONTINUE
      DO 390 I=1,NO2P1
```

```
      RHO(I,NO2P1)=0.
      RHO(NO2P1,I)=0.
  390 CONTINUE
      WRITE(8) A1,A2,A3,A4
      REWIND 8
C IF GAS PRINTING IS NOT TO BE DONE THIS CYCLE, GO TO 485
      IF(IPRINTG.NE.1) GO TO 485
  400 FORMAT(1H ,I3,8E15.7)
      NO2M7=NO2-7
      DO 420 IMIN=1,NO2M7,8
      IMAX=IMIN+7
      PRINT 410,IMIN,IMAX,NO2
  410 FORMAT(1H0*J            ((U(I,J),I=*I3*,*I3*),J=1,*I3*)*)
  420 PRINT 400,(J,(A1(I,J),I=IMIN,IMAX),J=1,NO2)
      DO 440 IMIN=1,NO2M7,8
      IMAX=IMIN+7
      PRINT 430,IMIN,IMAX,NO2
  430 FORMAT(1H0*J            ((V(I,J),I=*I3*,*I3*),J=1,*I3*)*)
  440 PRINT 400,(J,(A2(I,J),I=IMIN,IMAX),J=1,NO2)
      DO 460 IMIN=1,NO2M7,8
      IMAX=IMIN+7
      PRINT 450,IMIN,IMAX,NO2
  450 FORMAT(1H0*J            ((I(I,J),I=*I3*,*I3*),J=1,*I3*)*)
  460 PRINT 400,(J,(A3(I,J),I=IMIN,IMAX),J=1,NO2)
      DO 480 IMIN=1,NO2M7,8
      IMAX=IMIN+7
      PRINT 470,IMIN,IMAX,NO2
  470 FORMAT(1H0*J             ((M(I,J),I=*I3*,*I3*),J=1*I3*)*)
  480 PRINT 400,(J,(A4(I,J),I=IMIN,IMAX),J=1,NO2)
C IF GAS PLOTTING IS NOT TO BE DONE THIS CYCLE, RETURN.
  485 IF(IPLOTG.NE.1) GO TO 600
C PLOT VELOCITY VECTORS
      XNO2=NO2
      A=0.
      CALL DDIPLT(0,ING,1,A,A,0,XNO2,0,XNO2,2,XPG,1,YPG,13,ITAPXG)
      XYSCALE=10./NO2
      VSCALE=.1/DT
      DO 500 I=1,NO2
      DO 500 J=1,NO2
      IF(A4(I,J).LT.DMG) GO TO 500
```

C-2

```
      XA=XYSCALE*I
      YA=XYSCALE*J
      XB=XYSCALE*(I+VSCALE*A1(I,J))
      YB=XYSCALE*(J+VSCALE*A2(I,J))
      CALL PARROW(XA,YA,XB,YB,1)
  500 CONTINUE
      CALL DDIPLT(1,ING,1,A,A,0,XNO2,0,XNO2,2,XPG,1,YPG,13,ITAPXG)
C MAKE A CONTOUR PLOT OF THE DENSITY
      CALL DDIPLT(0,ING,5,XPR,YPR,XMIN4,XMAX4,YMIN4,YMAX4,3,XYL(1),
     1    2,PLM(1),14,ITAPXG)
      DX=0.
      DY=1.
      DO 515 J=2,NO2
      K=0
      DX=DX+RCOS
      DY=DY+RSIN
      DU=0.
      DV=1.
      DO 510 I=2,NO2
      K=K+1
      DU=DU+RCOS
      DV=DV+RSIN
      A1HORZ(K)=J+DU
      A1VERT(K)=SPHIM*A4(J,I)+DV
      A2HORZ(K)=I+DX
  510 A2VERT(K)=SPHIM*A4(I,J)+DY
      CALL DDIPLT(0,ING,K,A2HORZ,A2VERT,XMIN4,XMAX4,YMIN4,YMAX4,
     1    3,XYL(1),2,PLM(1),14,ITAPXG)
      CALL DDIPLT(0,ING.K,A1HORZ,A1VERT,XMIN4,XMAX4,YMIN4,YMAX4,
     1    3,XYL(1),2,PLM(1),14,ITAPXG)
  515 CONTINUE
      K=0
      DO 520 I=2,NO2
      XI=I
      K=K+1
      A2HORZ(K)=XI+RCOS
      A2VERT(K)=1.+RSIN
  520 CONTINUE
      CALL DDIPLT(1,ING,K,A2HORZ,A2VERT,XMIN4,XMAX4,YMIN4,YMAX4,
```

```
     1     3,XYL(1),2,PLM(1),13,ITAPXG)
C MAKE A CONTOUR PLOT OF THE TEMPERATURE
      CALL DDIPLT(0,ING,5,XPR,YPR,XMIN4,XMAX4,YMIN4,YMAX4,3,XYL(1),
     1     2,PLT(1),14,ITAPXG)
      DX=0.
      DY=1.
      DO 540 J=2,NO2
      K=0
      DX=DX+RCOS
      DY=DY+RSIN
      DU=0.
      DV=1.
      DO 530 I=2,NO2
      K=K+1
      DU=DU+RCOS
      DV=DV+RSIN
      A1HORZ(K)=J+DU
      A1VERT(K)=SPHIT*A3(J,I)+DV
      A2HORZ(K)=I+DX
  530 A2VERT(K)=SPHIT*A3(I,J)+DY
      CALL DDIPLT(0,ING,K,A2HORZ,A2VERT,XMIN4,XMAX4,YMIN4,YMAX4,
     1     3,XYL(1),2,PLT(1),14,ITAPXG)
      CALL DDIPLT(0,ING,K,A1HORZ,A1VERT,XMIN4,XMAX4,YMIN4,YMAX4,
     1     3,XYL(1),2,PLT(1),14,ITAPXG)
  540 CONTINUE
      K=0
      DO 545 I=2,NO2
      XI=I
      K=K+1
      A2HORZ(K)=XI+RCOS
      A2VERT(K)=1.+RSIN
  545 CONTINUE
      CALL DDIPLT(1,ING,K,A2HORZ,A2VERT,XMIN4,XMAX4,YMIN4,YMAX4,
     1     3,XYL(1),2,PLT(1),13,ITAPXG)
C MAKE A CONTOUR PLOT OF THE PRESSURE
      CALL DDIPLT(0,ING,5,XPR,YPR,XMIN4,XMAX4,YMIN4,YMAX4,3,XYL(1),
     1     2,PLP(1),14,ITAPXG)
      DX=0.
      DY=1.
      DO 560 J=2,NO2
```

```
      K=0
      DX=DX+RCOS
      DY=DY+RSIN
      DU=0.
      DV=1.
      DO 550 I=2,NO2
      K=K+1
      DU=DU+RCOS
      DV=DV+RSIN
      A1HORZ(K)=J+DU
      A1VERT(K)=SPHIP*A3(J,I)*A4(J,I)+DV
      A2HORZ(K)=I+DX
  550 A2VERT(K)=SPHIP*A3(I,J)*A4(I,J)+DY
      CALL DDIPLT(0,ING,K,A2HORZ,A2VERT,XMIN4,XMAX4,YMIN4,YMAX4,
     1     3,XYL(1),2,PLP(1),14,ITAPXG)
      CALL DDIPLT(0,ING,K,A1HORZ,A1VERT,XMIN4,XMAX4,YMIN4,YMAX4,
     1     3,XYL(1),2,PLP(1),14,ITAPXG)
  560 CONTINUE
      K=0
      DO 570 I=2,NO2
      XI=I
      K=K+1
      A2HORZ(K)=XI+RCOS
      A2VERT(K)=1.+RSIN
  570 CONTINUE
      CALL DDIPLT(1,ING,K,A2HORZ,A2VERT,XMIN4,XMAX4,YMIN4,YMAX4,
     1     3,XYL(1),2,PLP(1),13,ITAPXG)
  600 IF(CYY.LT.CY) GO TO 650
C IF(CYY.EQ.CY) WRITE INFORMATION TO BE SAVED ON TAPE1
C     TRANSFER PARTICLE POSITIONS FROM TAPE JTG ONTO TAPE1.
      NS2=0
  610 READ(JTG) XHOLD,YHOLD
      WRITE(1) XHOLD,YHOLD
      NS2=NS2+NBSG
      IF(NS2.LT.NBRG) GO TO 610
      WRITE(1) A1,A2,A3,A4,CYY,SUMMAS,SUMPX,SUMPY,SUMKE,SUMIE
      REWIND 1
  650 IF(IPRNTMG.EQ.0) GO TO 700
      PRINT 652,SCALM
  652 FORMAT(1H1*THE FOLLOWING IS A PLOT OF MASS DENSITY (ALREADY SCALED
```

```
   1 WITH DT SQUARED) WHICH HAS BEEN DIVIDED BY *E16.8* AND THEN*)
      PRINT 653
  653 FORMAT(1H *INTEGERIZED, WITH VALUES BELOW 1 OR ABOVE 9 REPLACED BY
     1 BLANK OR STAR RESPECTIVELY*/)
      DO 670 I=1,NO2
      DO 670 J=1,NO2
      IM=A4(I,J)/SCALM
      IF(IM.EQ.0) GO TO 660
      IF(IM.GT.9) GO TO 665
      ENCODE(10,655,A4(I,J)) IM
  655 FORMAT(I10)
      GO TO 670
  660 A4(I,J)=10H
      GO TO 670
  665 A4(I,J)=10H         *
  670 CONTINUE
      DO 672 I=1,NO2
      A4(I,1)=10H         -
      A4(I,NO2)=10H-
  672 CONTINUE
      DO 674 J=2,NO2M1
      A4(1,J)=10H         I
      A4(NO2,J)=10HI
  674 CONTINUE
      DO 693 JJ=1,NO2
      J=NO2P1-JJ
  693 PRINT 695,(A4(I,J),I=1,NO2)
  695 FORMAT(1H ,64R2)
      PRINT 697
  697 FORMAT(1H1,1H )
  700 RETURN
      END
```

```
      OVERLAY(IFILE,4,0)
      PROGRAM GASPLOT
      COMMON/ALLCOM/I2A,ITEST,N,CY,CYY,RHO(33,33)
      COMMON/GASCOM/NPLOTG,NPRINTG,NPRNTMG,NO2SQ,NO2,NO2P1,NO2M1,DMG,
     1     GM102,AUO2,AUO4,SAVPE(200),SAVIE(200),SAVKE(200),SAVTE(200),
     2     CYMIN,NBRG,NBSG,JTG,JSG,NMKG,PI,MASK1,ING(2),XMING,XMAXG,
     3     YMING,YMAXG,XPG(2),YPG,ITAPXG,XYL(3),PLP(2),PLT(2),PLM(2),
     4     RCOS,RSIN,XPR(5),YPR(5),XMAX4,XMIN4,YMAX4,YMIN4,SPHIM,SPHIT,
     5     SPHIP,EMIN,EMAX,SCALM,DT,DT2,DT3,DT4,SUMMAS,SUMPX,SUMPY,
     6     SUMKE,SUMIE,SPHI,PL(2)
      DIMENSION XDATA(200),LABLEN(8)
      DIMENSION ENDCYY(2),ENDPE(2),ENDIE(2),ENDKE(2),ENDTE(2)
      INTEGER CY
      CALL PSEUDO
      CYMAX=CY
      NCYY=CY-CYMIN
      DO 20 I=1,NCYY
   20 XDATA(I)=CYMIN+I
      LABLCYY=10HCYCLES
      LABLEN(1)=10HENERGY***C
      LABLEN(2)=10HIRCLE-POTE
      LABLEN(3)=10HNTIAL,SQUA
      LABLEN(4)=10HRE-INTERNA
      LABLEN(5)=10HL,DIAMOND-
      LABLEN(6)=10HKINETIC,TR
      LABLEN(7)=10HIANGLE-TOT
      LABLEN(8)=10HAL
      ENDCYY(1)=CYMIN+1
      ENDCYY(2)=CY
      ENDPE(1)=SAVPE(1)
      ENDPE(2)=SAVPE(NCYY)
      ENDIE(1)=SAVIE(1)
      ENDIE(2)=SAVIE(NCYY)
      ENDKE(1)=SAVKE(1)
      ENDKE(2)=SAVKE(NCYY)
      ENDTE(1)=SAVTE(1)
      ENDTE(2)=SAVTE(NCYY)
      IF(NCYY.LT.2) GO TO 40
      CALL DDIPLT(0,ING,2,ENDCYY,ENDPE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,1)
```

```
      CALL DDIPLT(0,ING,2,ENDCYY,ENDIE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,2)
      CALL DDIPLT(0,ING,2,ENDCYY,ENDKE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,3)
      CALL DDIPLT(0,ING,2,ENDCYY,ENDTE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,4)
   40 CALL DDIPLT(0,ING,NCYY,XDATA,SAVPE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,0)
      CALL DDIPLT(0,ING,NCYY,XDATA,SAVIE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,0)
      CALL DDIPLT(0,ING,NCYY,XDATA,SAVKE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,0)
      CALL DDIPLT(1,ING,NCYY,XDATA,SAVTE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,0)
      RETURN
      END
```

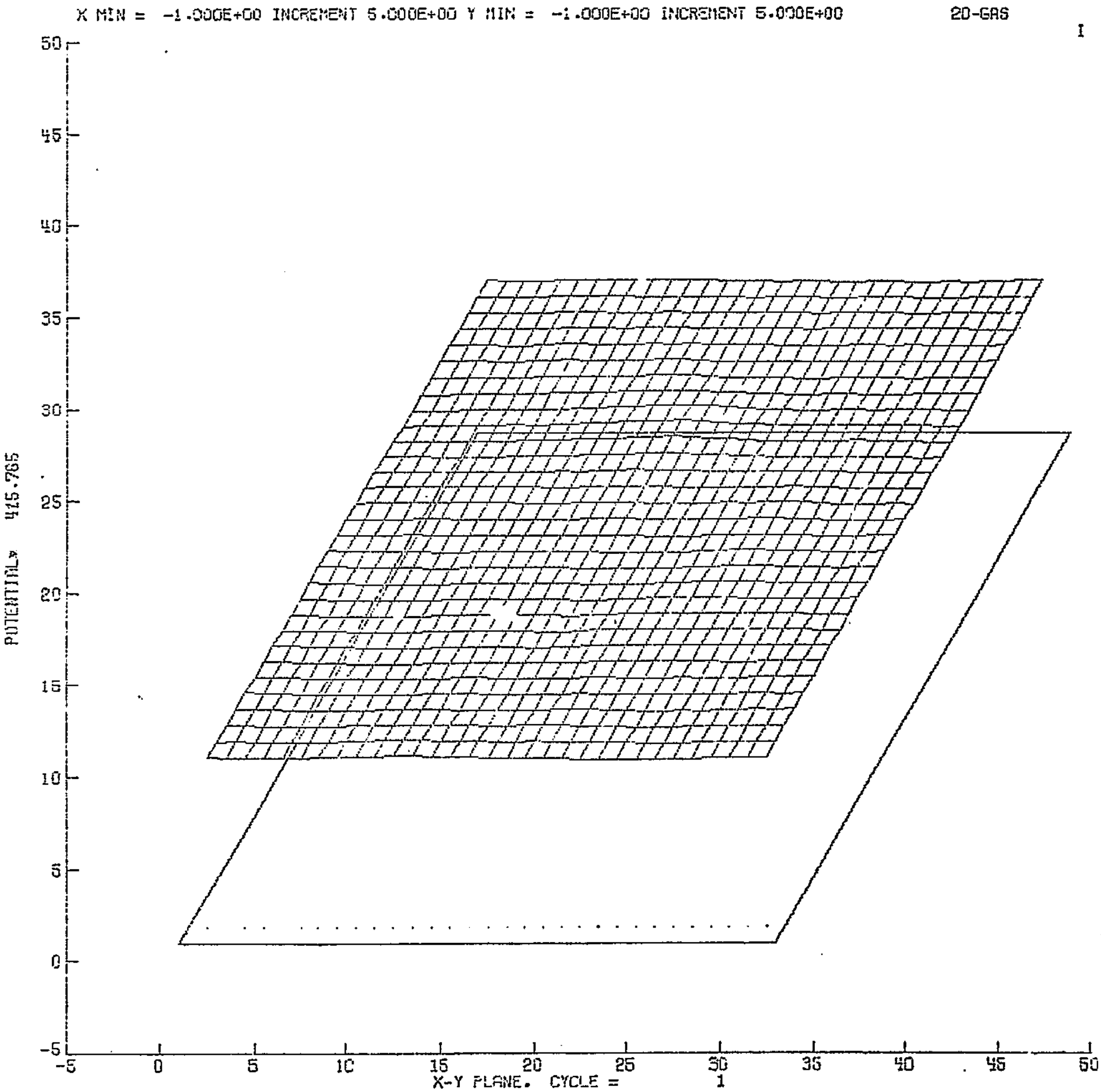
APPENDIX E

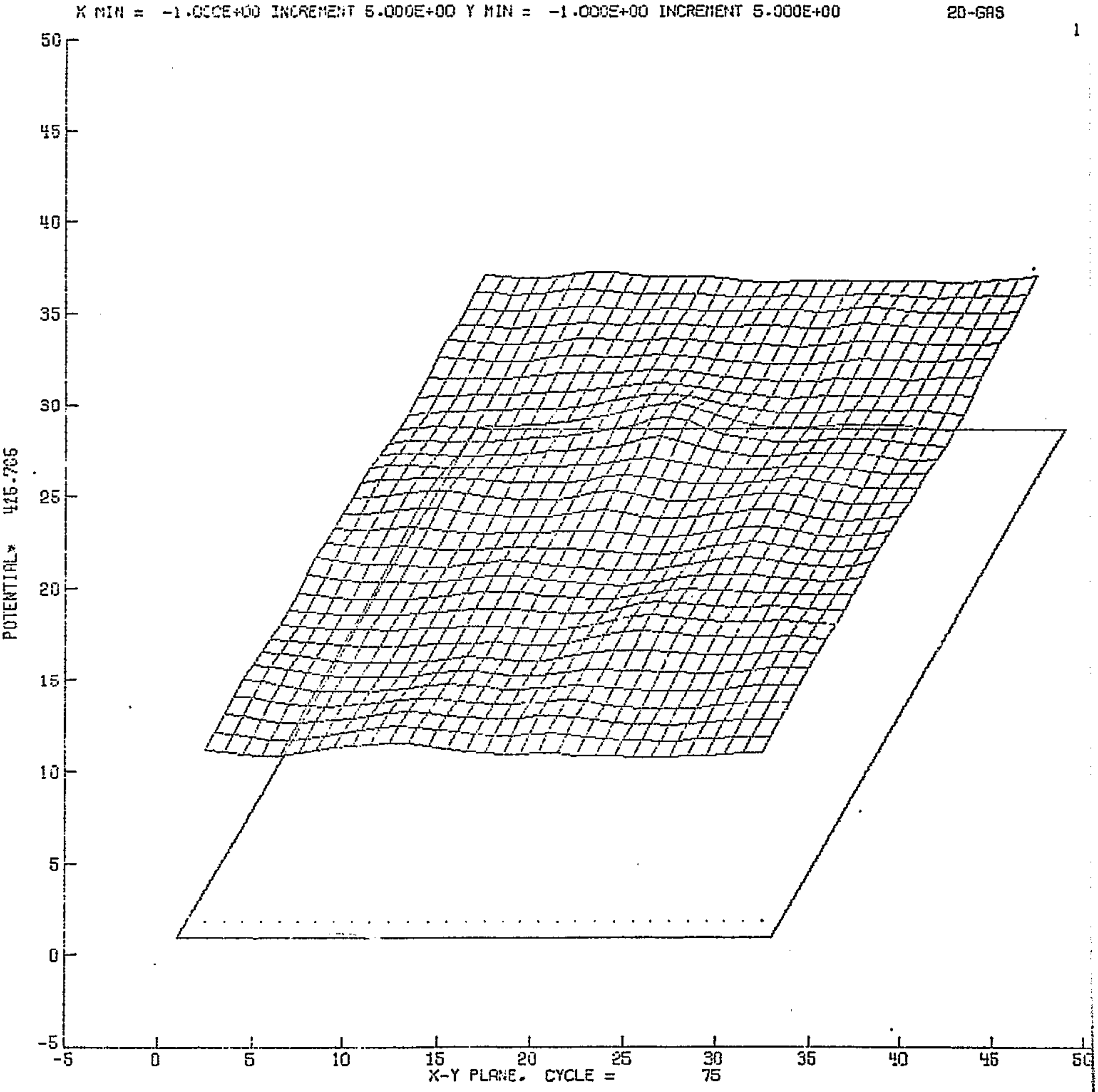
Computer Plots Produced by the Jeans Instability Simulator of Appendix D.

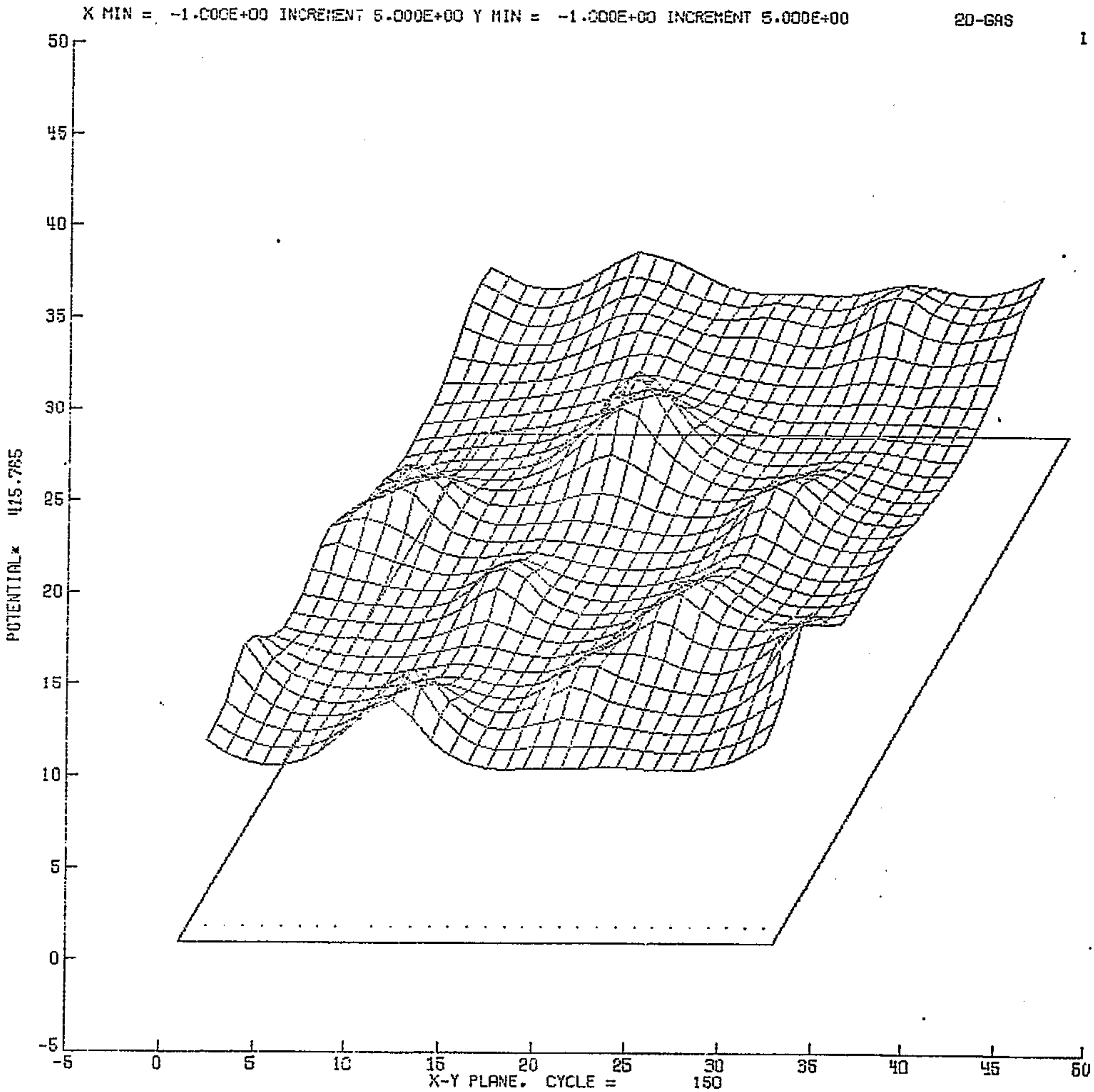
These plots are from two full length runs with identical initial conditions except for the ratio of total internal to total potential energy. The mesh size is 32 x 32. Later runs will be made with meshes of 64 x 64 and possibly 128 x 128.

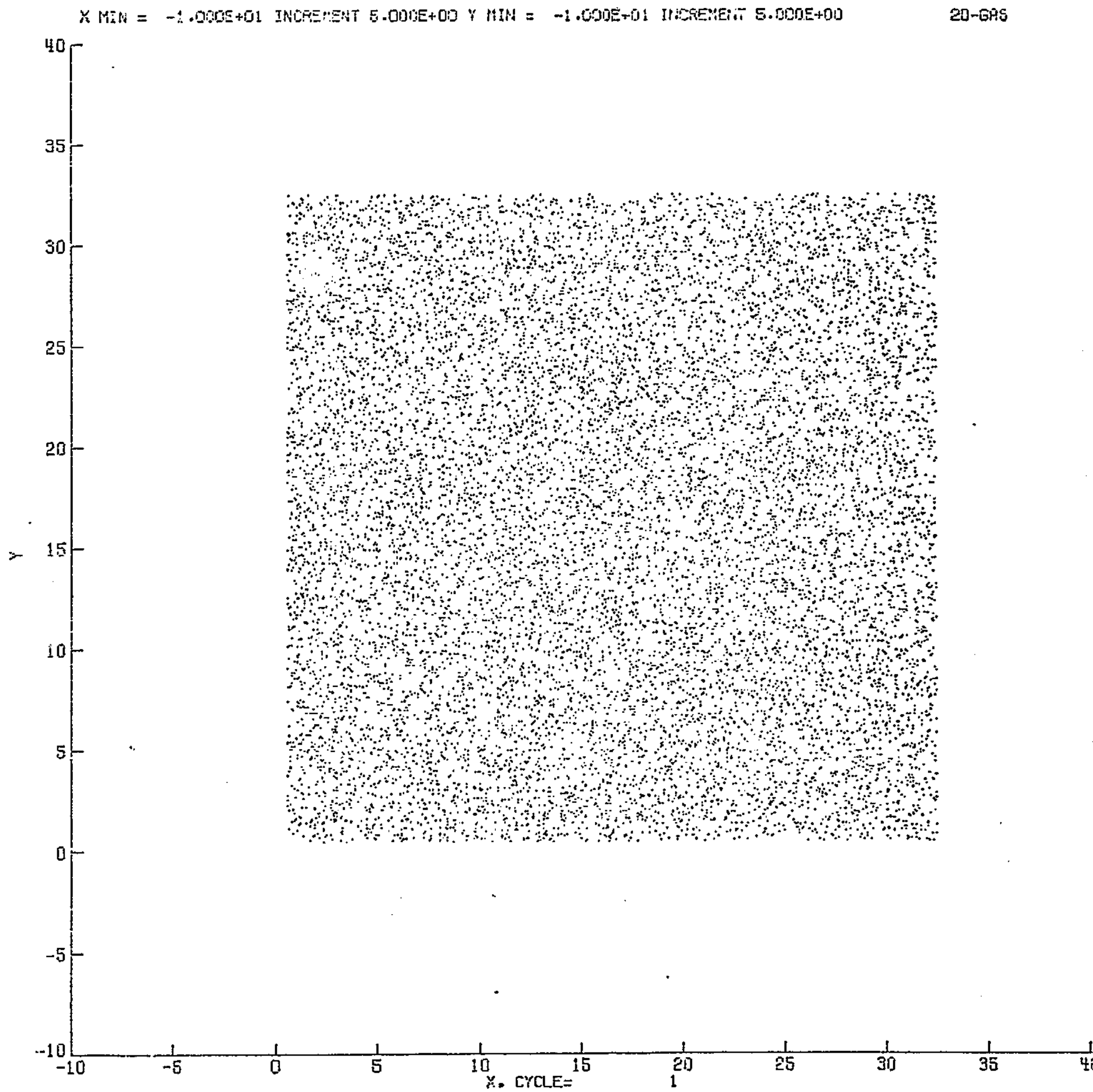
The last plot of each run demonstrates that despite rapid changes in total potential, internal, and kinetic energies, the code conserves total energy almost exactly.

Cycle No.	Plot Type	Page No. IE/PE =.00735	Page No. IE/PE =.0367
150	cell pressure over x-y plane	E-20	E-39
1-150	total potential, internal, kinetic, and total energy vs. cycle (time)	E-21	E-40

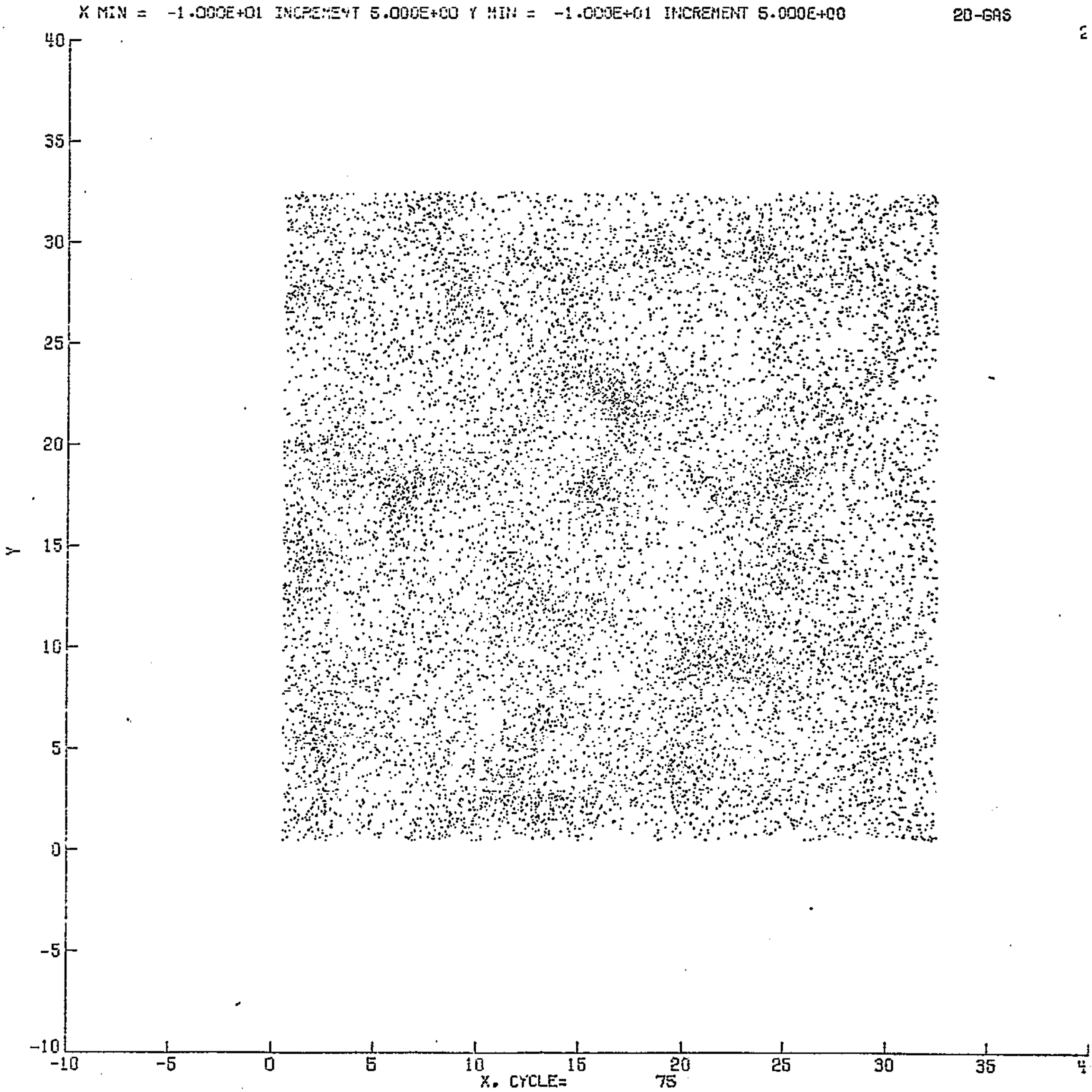

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
I
POTENTIAL* 415.765
50
45
40
35
30
25
20
15
10
5
0
-5
-5
0
5
10
15
20
25
30
35
40
45
50
X-Y PLANE. CYCLE = 1

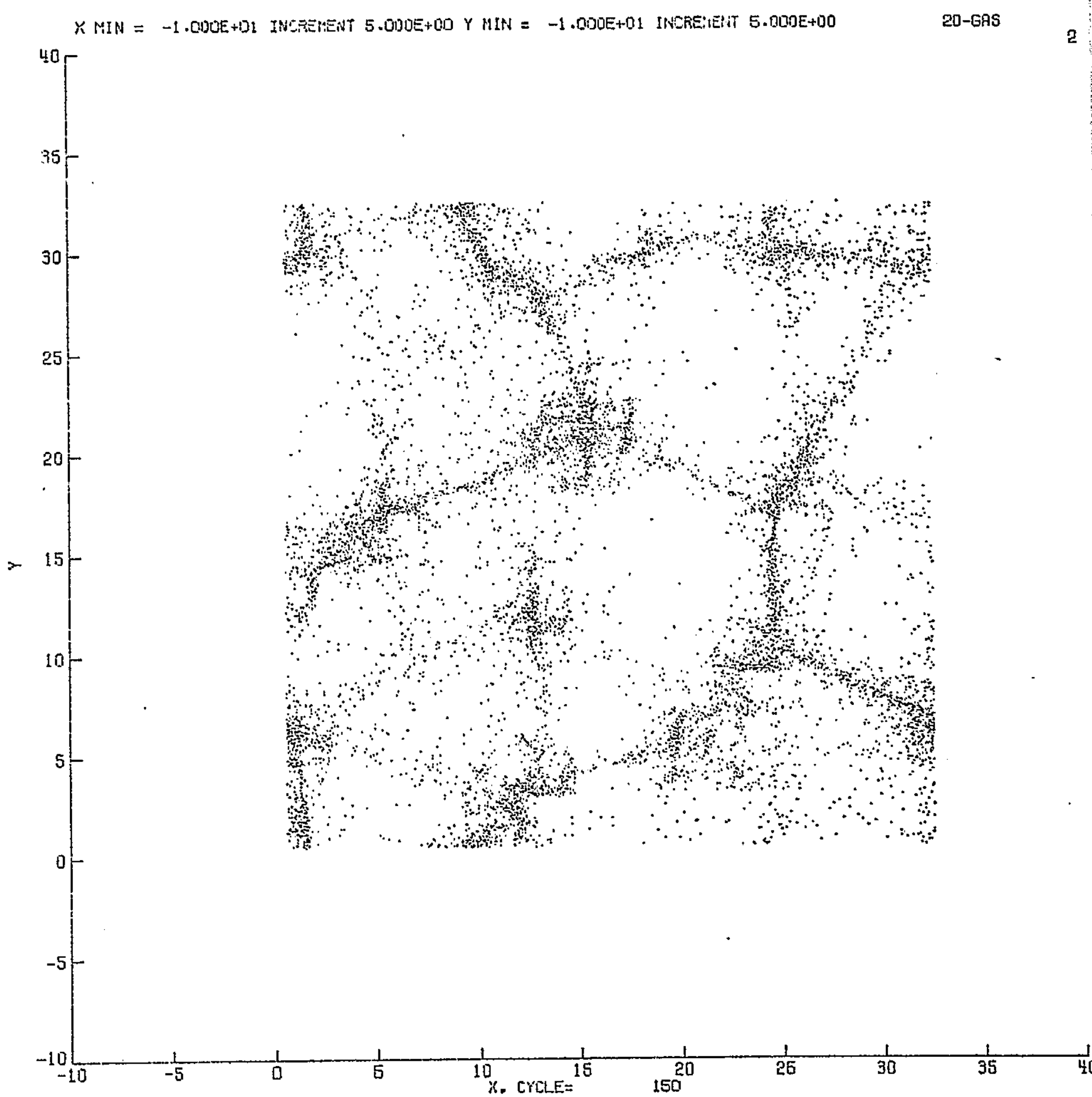

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
1
POTENTIAL* 415.705
X-Y PLANE. CYCLE = 75

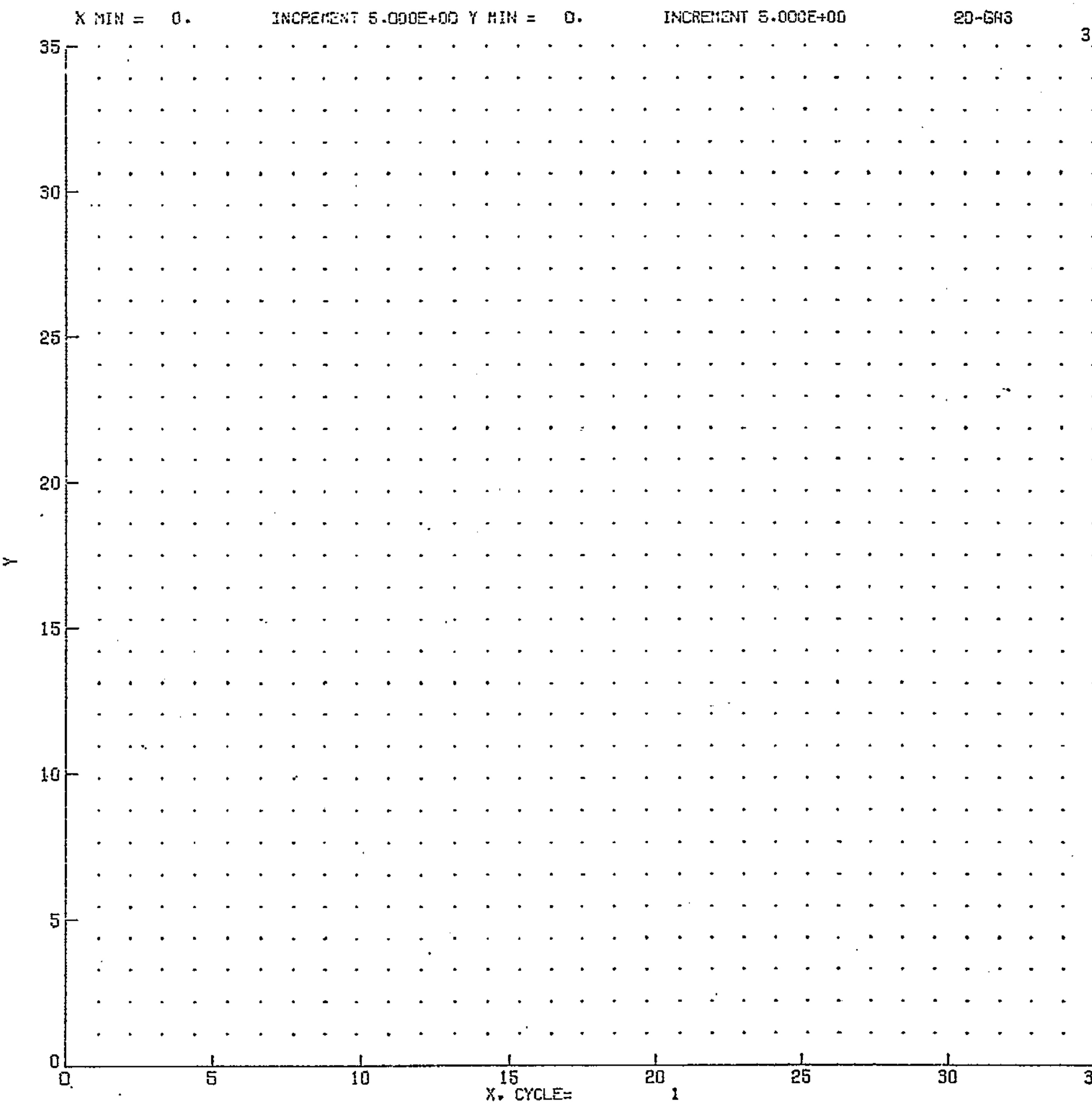

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
I
POTENTIAL* 415.765
X-Y PLANE. CYCLE = 150
-5
0
5
10
15
20
25
30
35
40
45
50

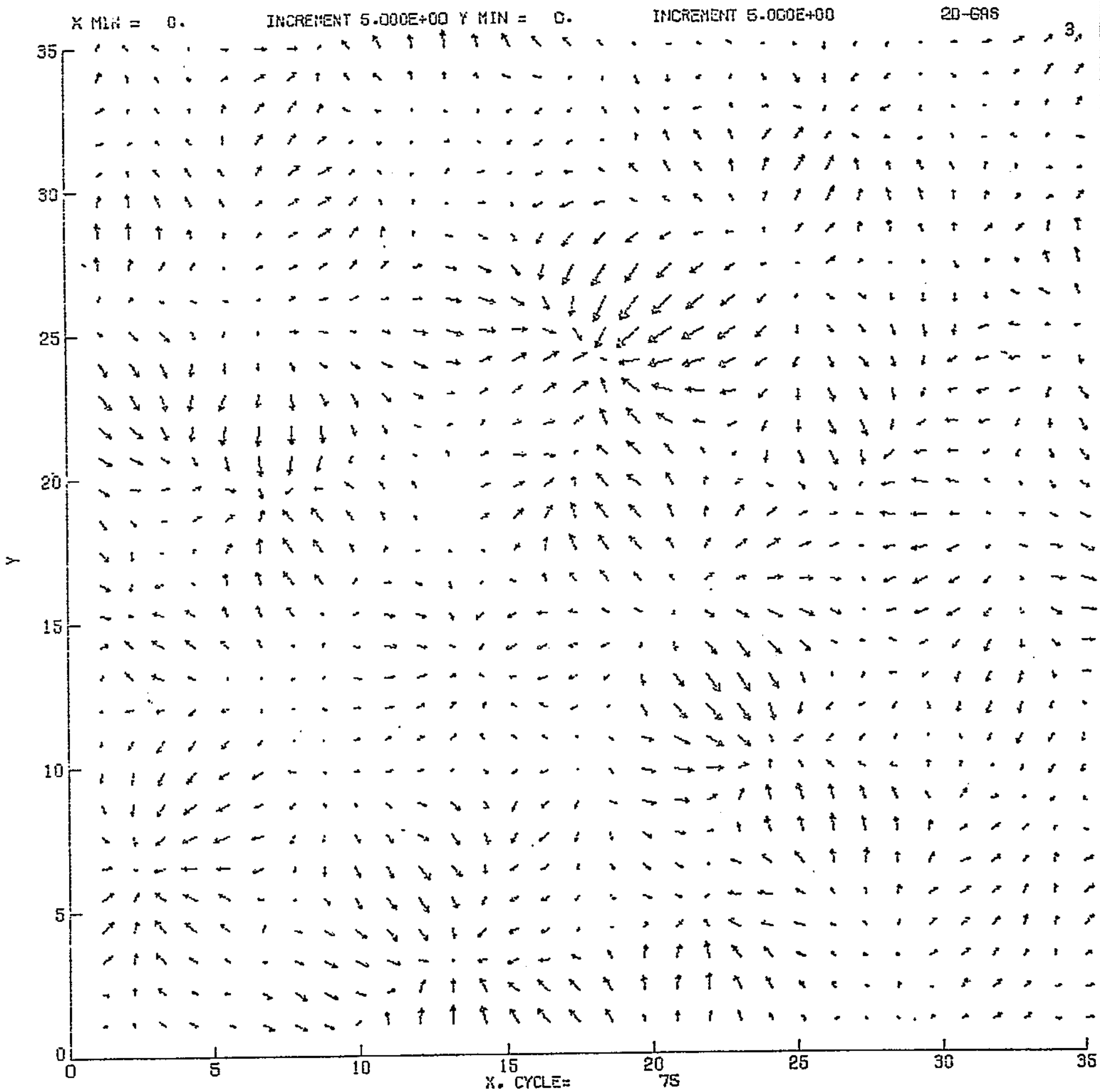


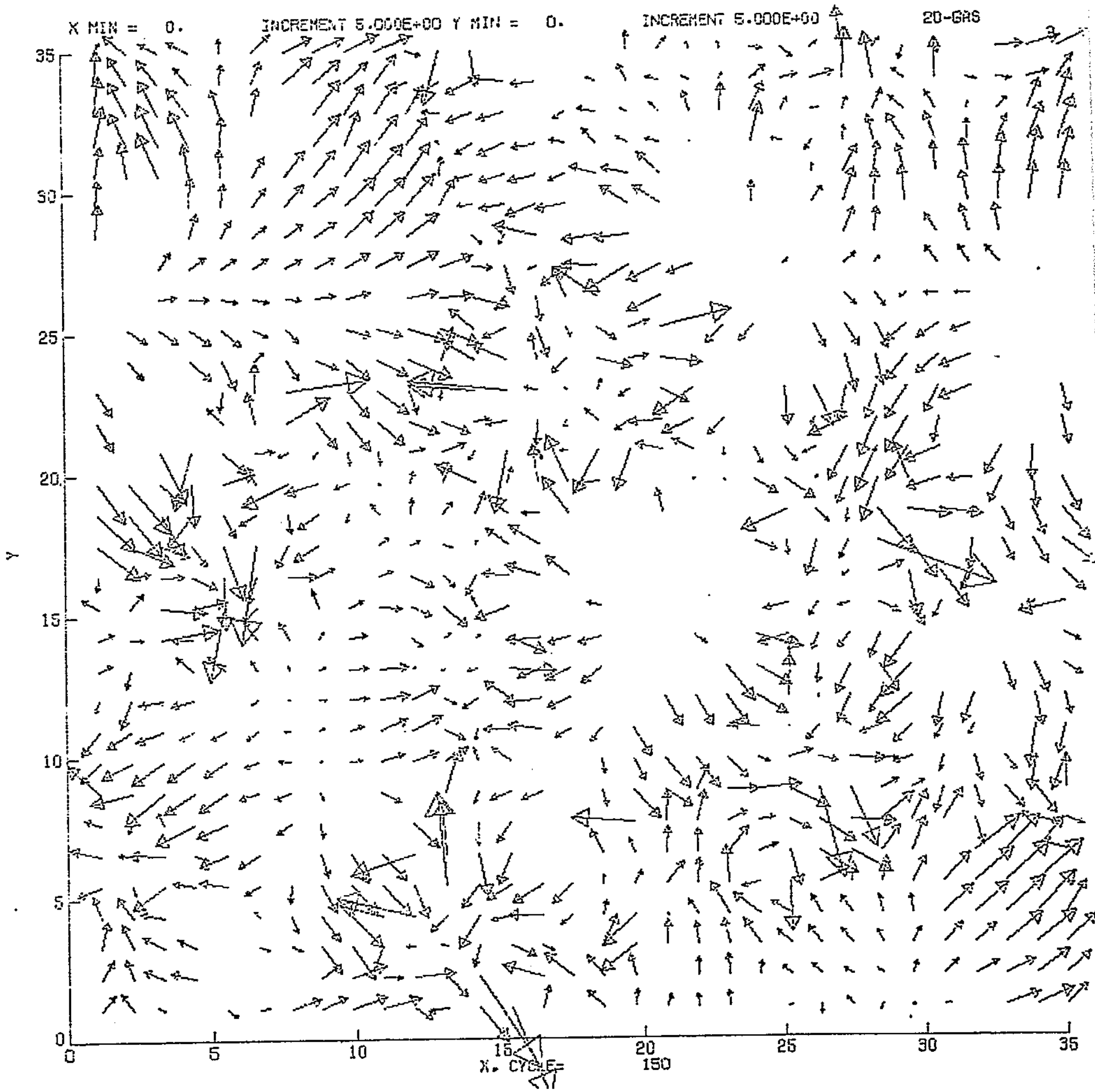
X MIN = -1.000E+01 INCREMENT 5.000E+00 Y MIN = -1.000E+01 INCREMENT 5.000E+00
2D-GAS
Y
X. CYCLE= 1


X MIN = -1.000E+01 INCREMENT 5.000E+00 Y MIN = -1.000E+01 INCREMENT 5.000E+00
2D-GAS
Y
X, CYCLE= 75
40
35
30
25
20
15
10
5
0
-5
-10
-10
-5
0
5
10
15
20
25
30
35

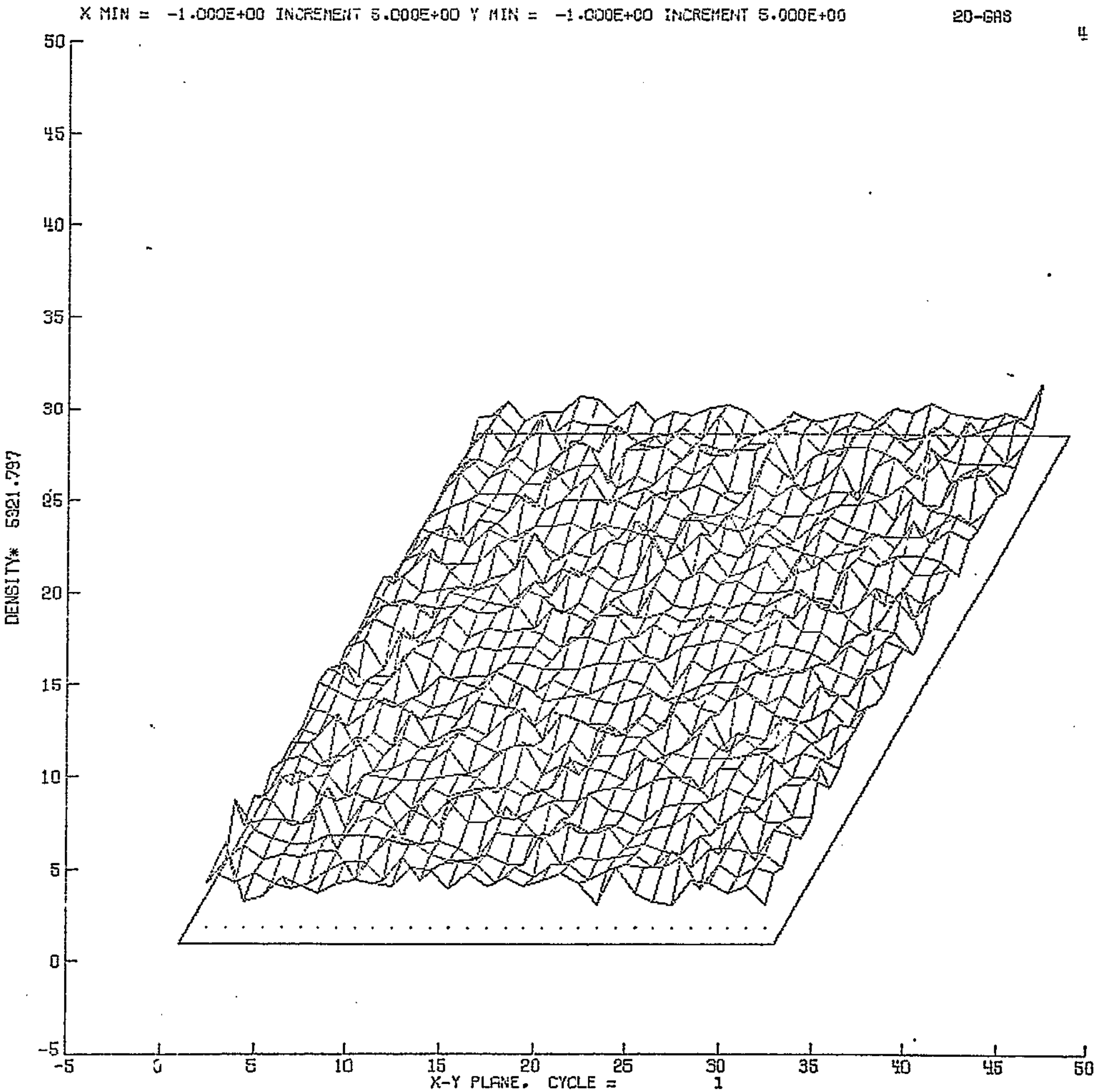

X MIN = -1.000E+01 INCREMENT 5.000E+00 Y MIN = -1.000E+01 INCREMENT 5.000E+00
20-GAS
2
Y
X, CYCLE= 150
40
35
30
25
20
15
10
5
0
-5
-10
-10
-5
0
5
10
15
20
25
30
35
40


X MIN = 0. INCREMENT 5.000E+00 Y MIN = 0. INCREMENT 5.000E+00 2D-GAS
3
35
30
25
20
15
10
5
0
Y
0
5
10
15
20
25
30
35
X, CYCLE= 1


X MIN = 0. INCREMENT 5.000E+00 Y MIN = 0. INCREMENT 5.000E+00
2D-GAS
3,
Y
0
5
10
15
20
25
30
35
X. CYCLE= 75

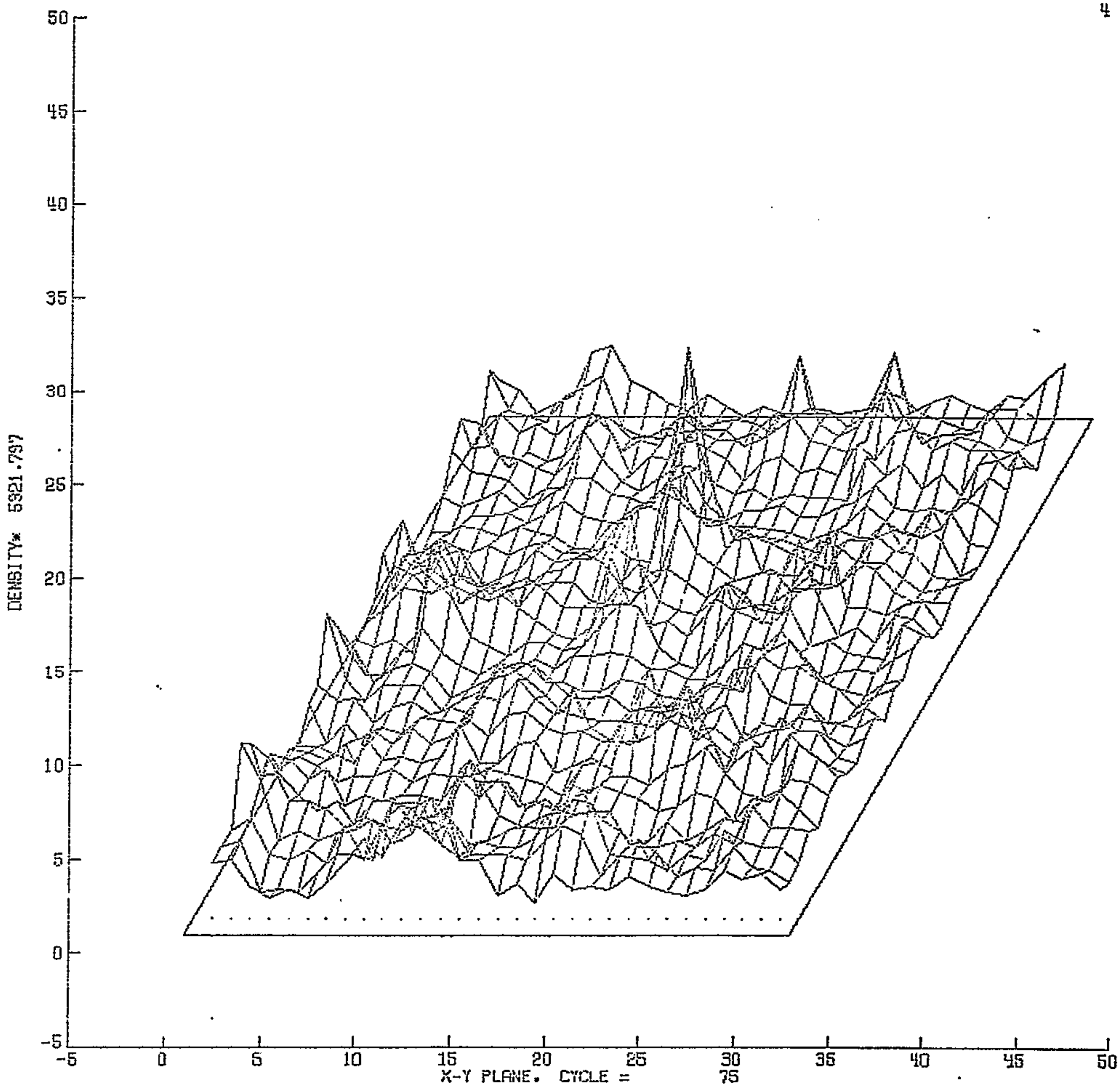


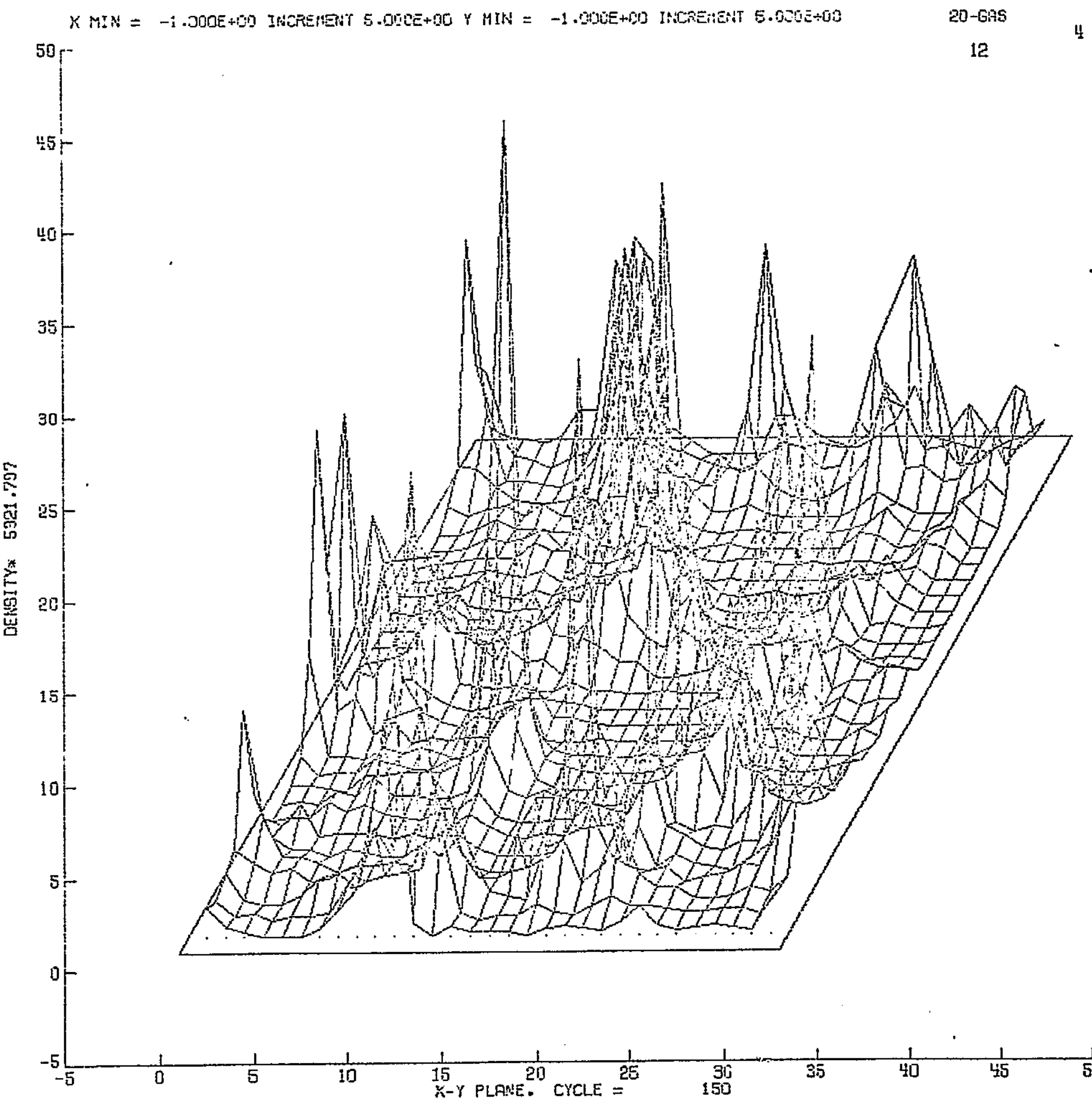
X MIN = 0.
INCREMENT 5.000E+00 Y MIN = 0.
INCREMENT 5.000E+00
2D-GAS
Y
X. CYCLE= 150
0
5
10
15
20
25
30
35


X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
20-GAS
DENSITY* 5321.797
X-Y PLANE, CYCLE = 1
50
45
40
35
30
25
20
15
10
5
0
-5
-5
0
5
10
15
20
25
30
35
40
45
50

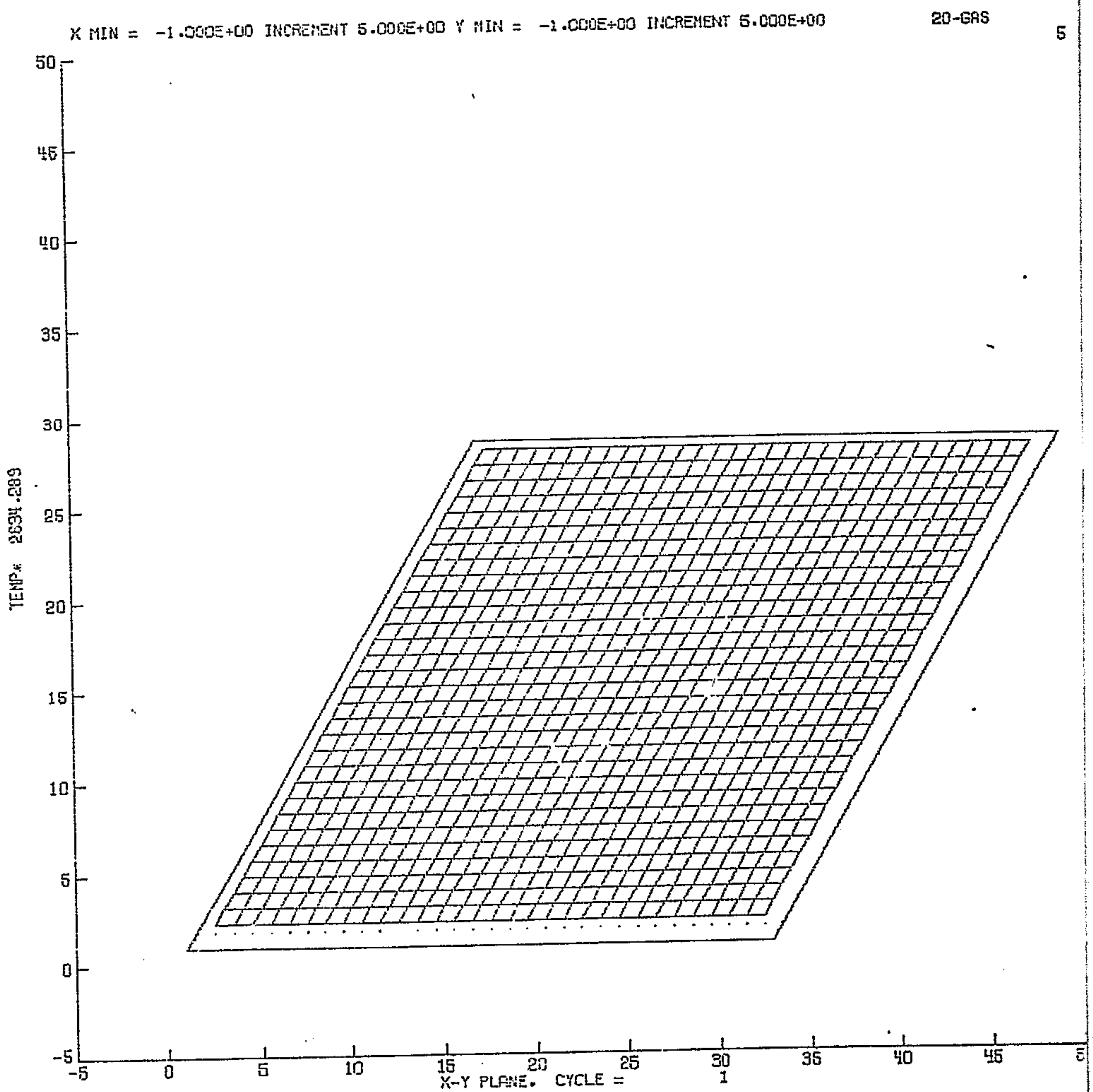
X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00 20-GAS

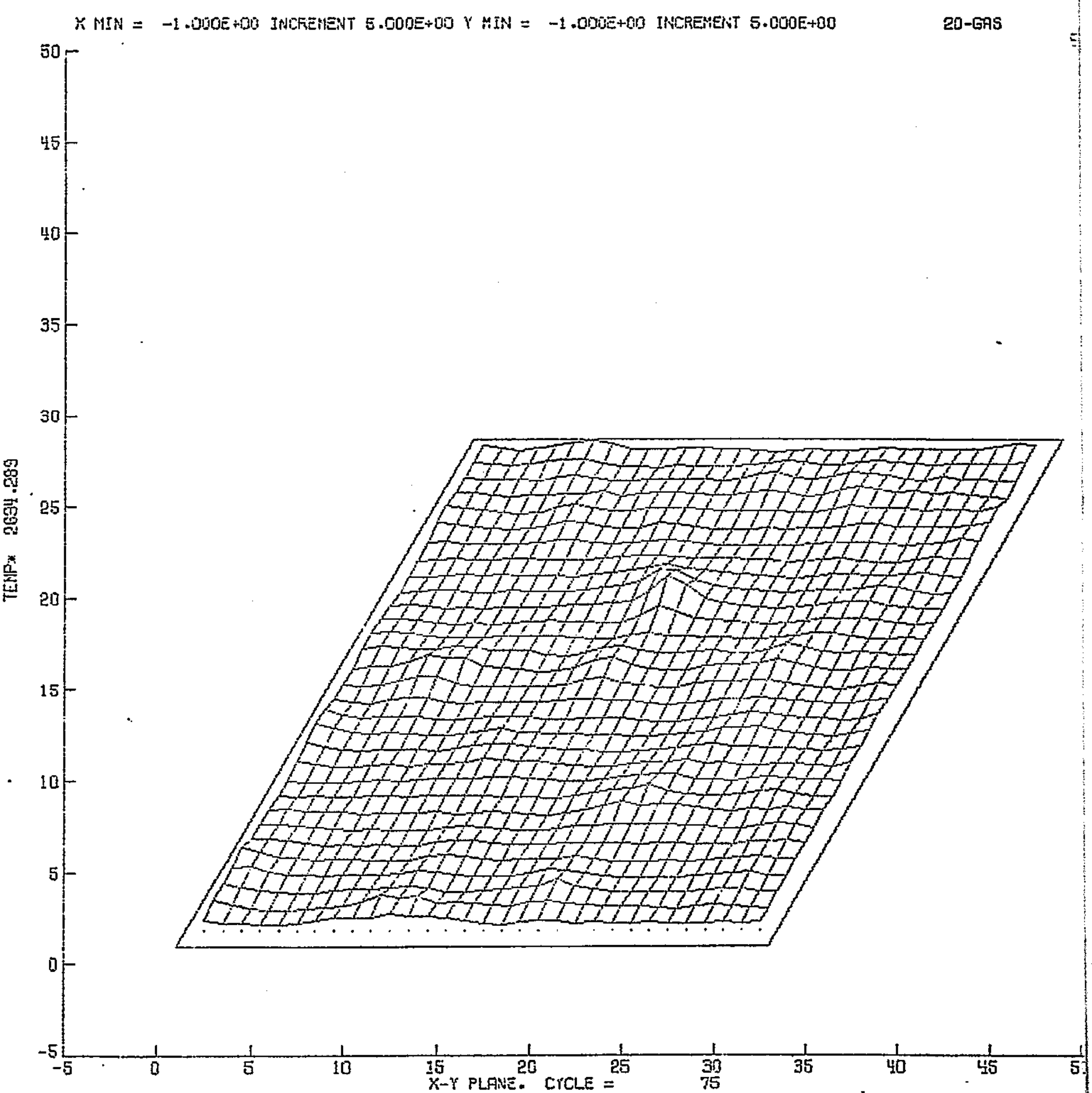
4

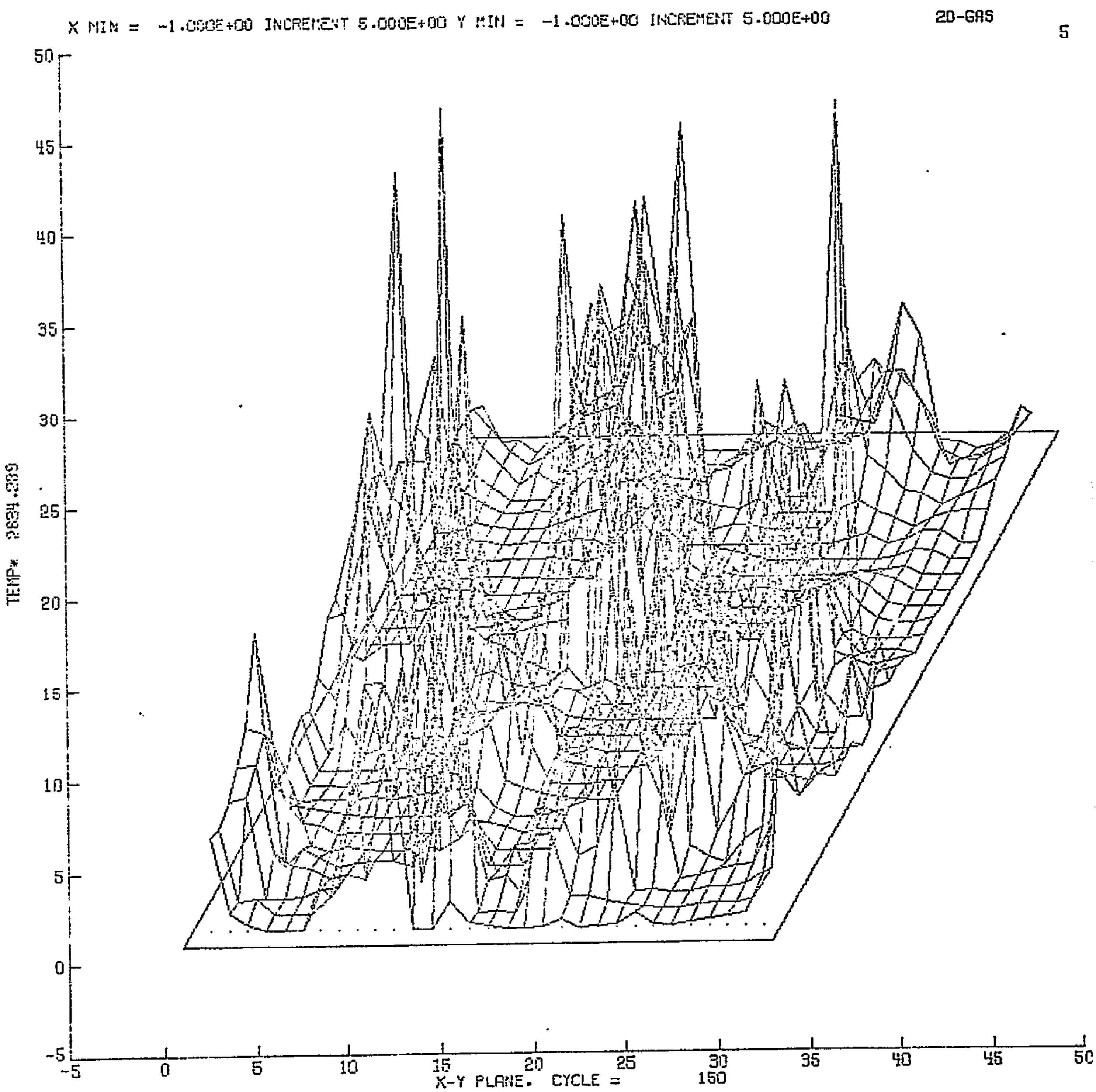




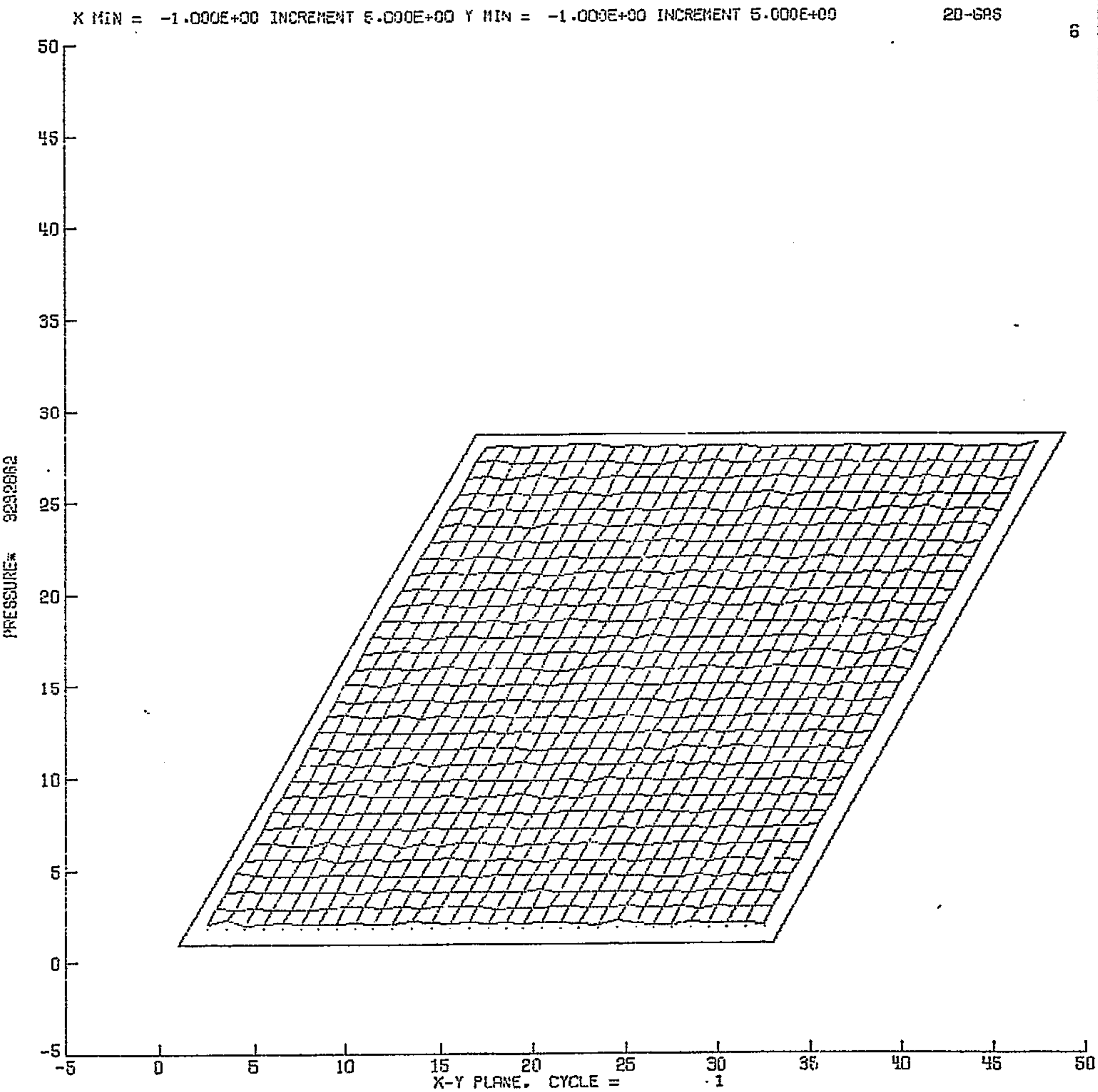
X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
20-GAS
4
12
DENSITY* 5321.797
50
45
40
35
30
25
20
15
10
5
0
-5
-5
0
5
10
15
20
25
30
35
40
45
50
X-Y PLANE. CYCLE = 150

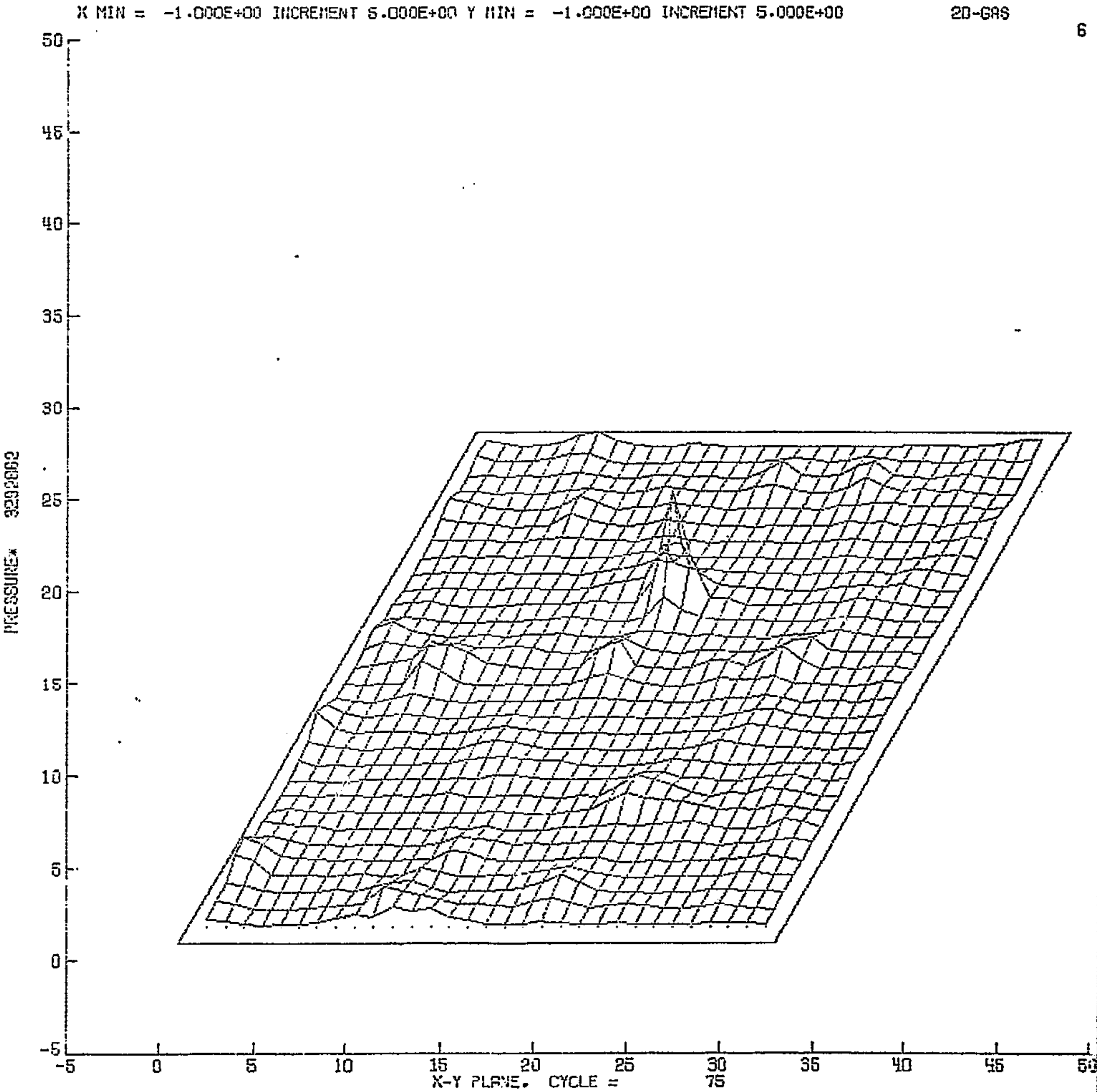

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
20-GAS
5
TEMP= 2634.289
X-Y PLANE. CYCLE = 1

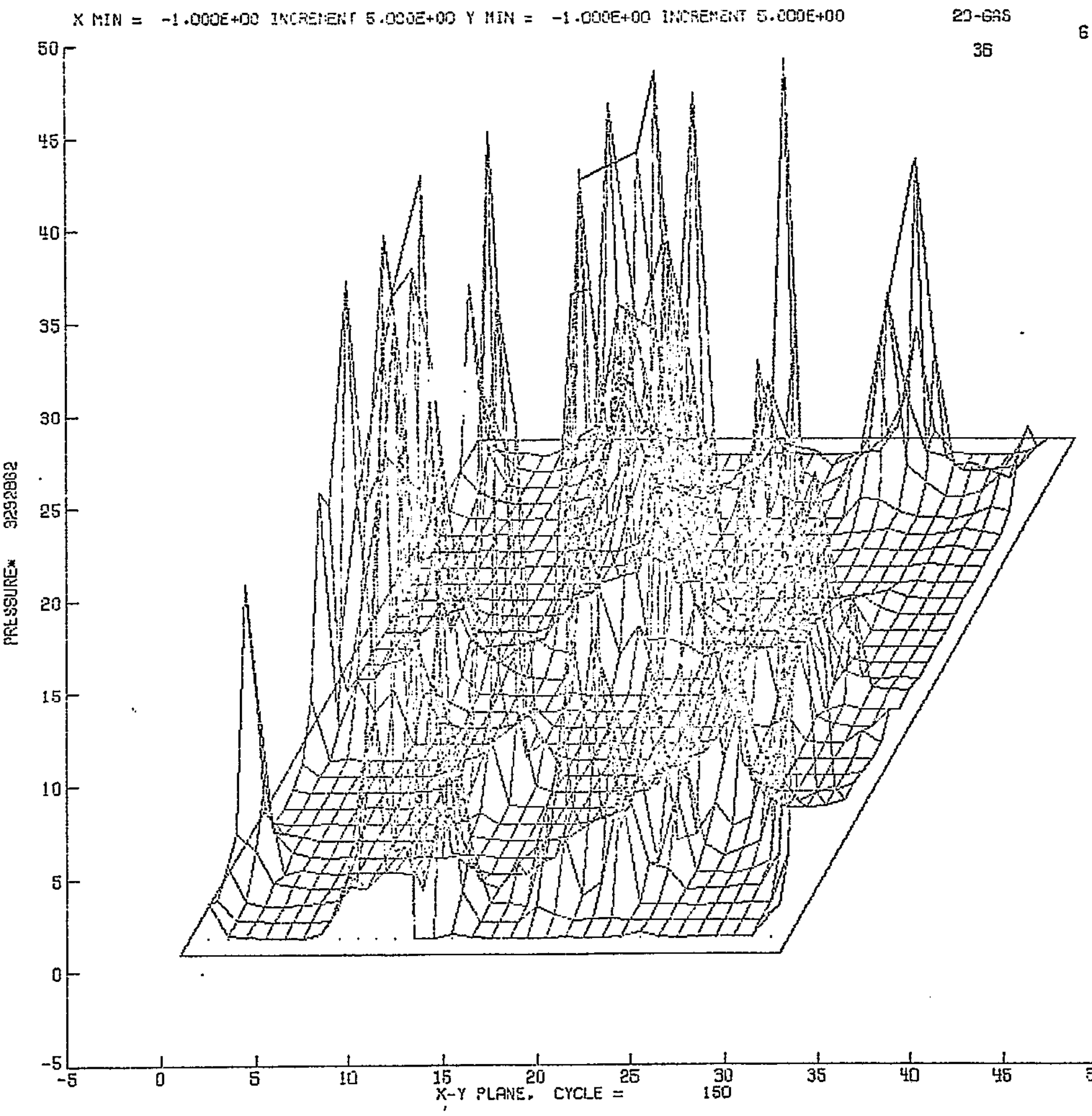

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
20-GAS
TEMP* 2634.289
X-Y PLANE. CYCLE = 75
-5
0
5
10
15
20
25
30
35
40
45
50

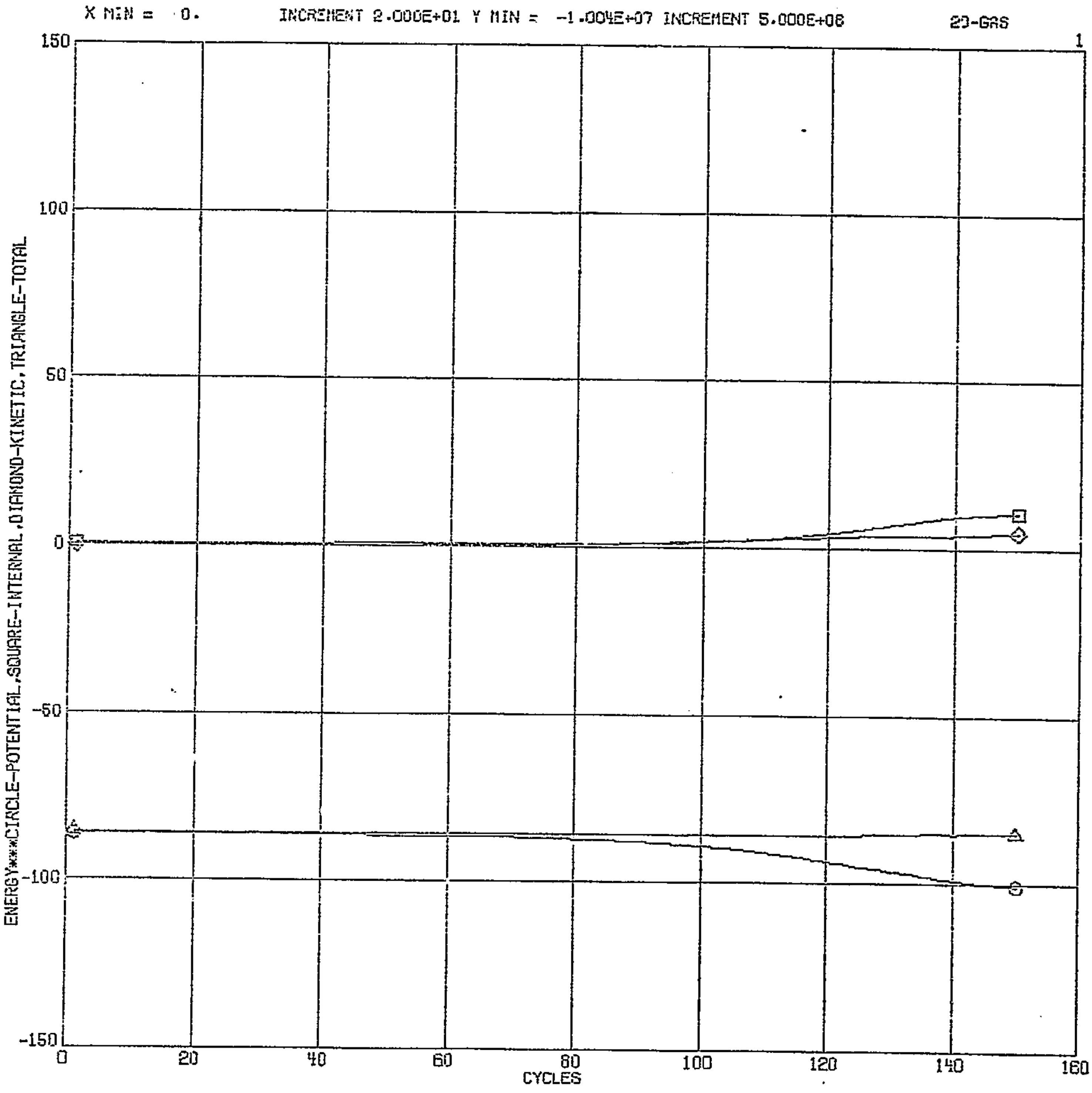


X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
5
TEMP* 2834.339
X-Y PLANE. CYCLE = 150

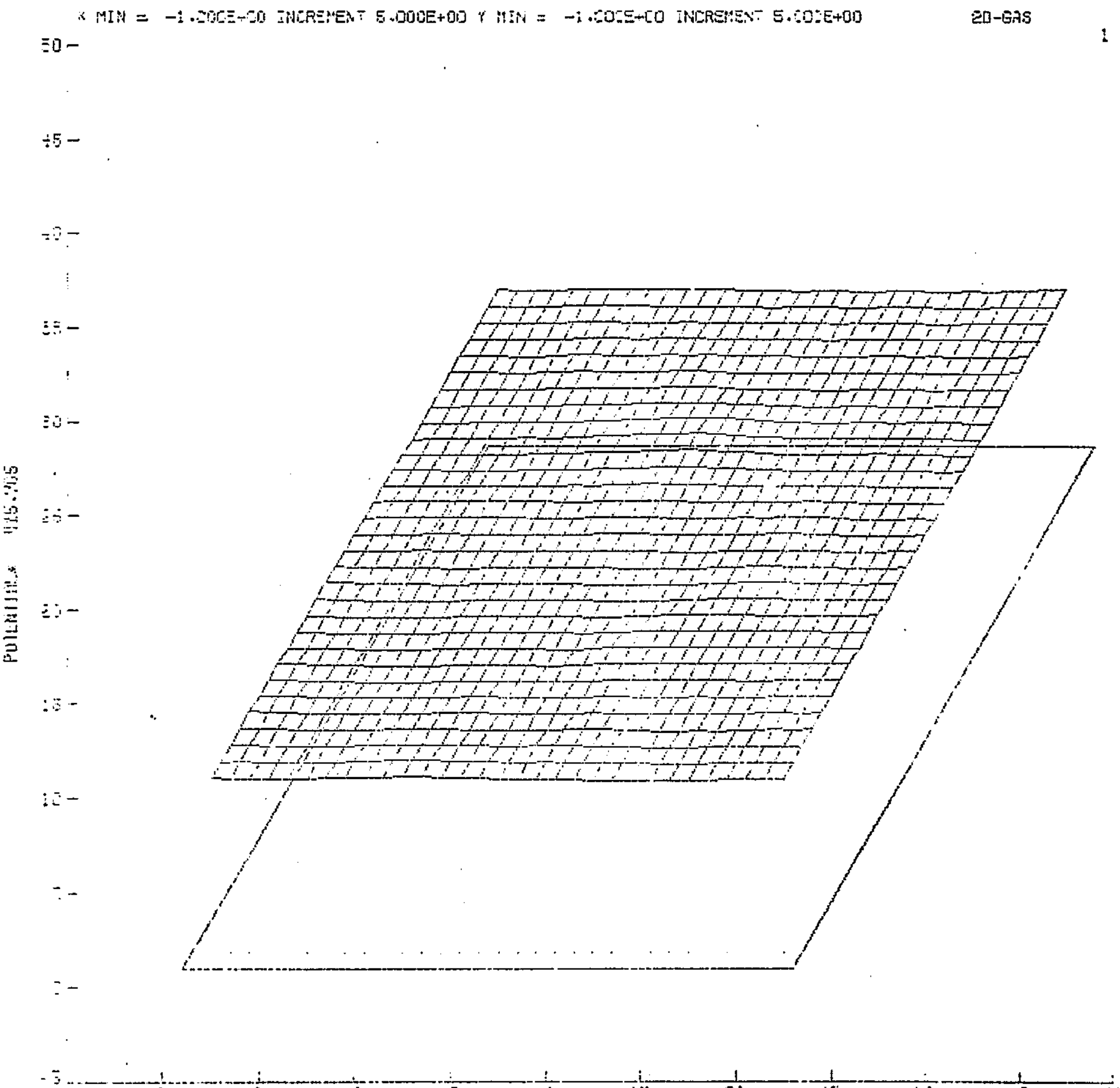

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
6
PRESSURE* 3292062
X-Y PLANE. CYCLE = 1


X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
6
PRESSURE* 3292062
X-Y PLANE. CYCLE = 75
-5
0
5
10
15
20
25
30
35
40
45
50


X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
6
36
PRESSURE* 9292862
50
45
40
35
30
25
20
15
10
5
0
-5
-5
0
5
10
15
20
25
30
35
40
45
X-Y PLANE. CYCLE = 150

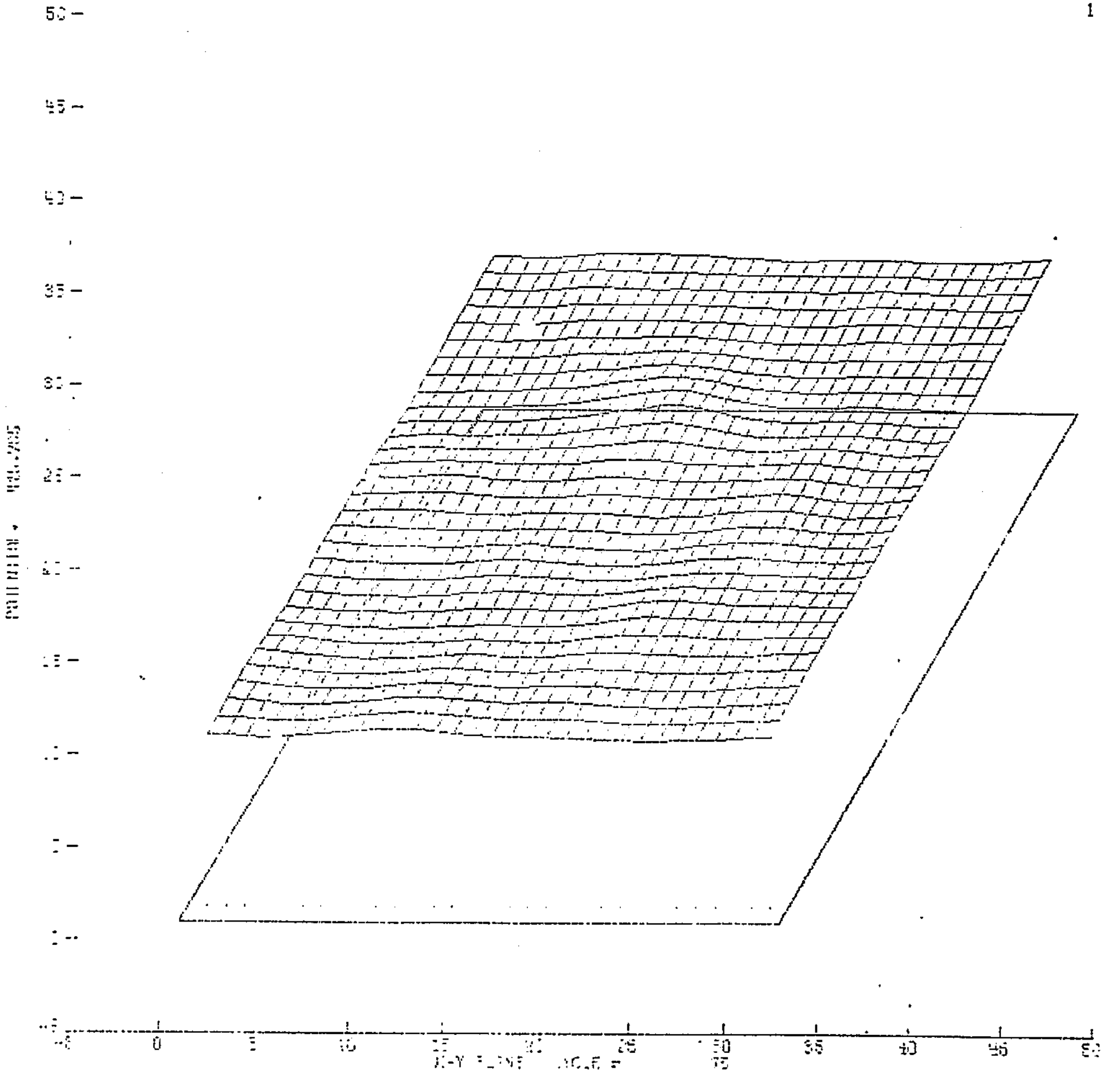


X MIN = 0. INCREMENT 2.000E+01 Y MIN = -1.004E+07 INCREMENT 5.000E+06
2D-GAS
1
150
100
50
0
-50
-100
-150
0
20
40
60
80
100
120
140
160
CYCLES
ENERGY***CIRCLE-POTENTIAL,SQUARE-INTERNAL,DIAMOND-KINETIC,TRIANGLE-TOTAL

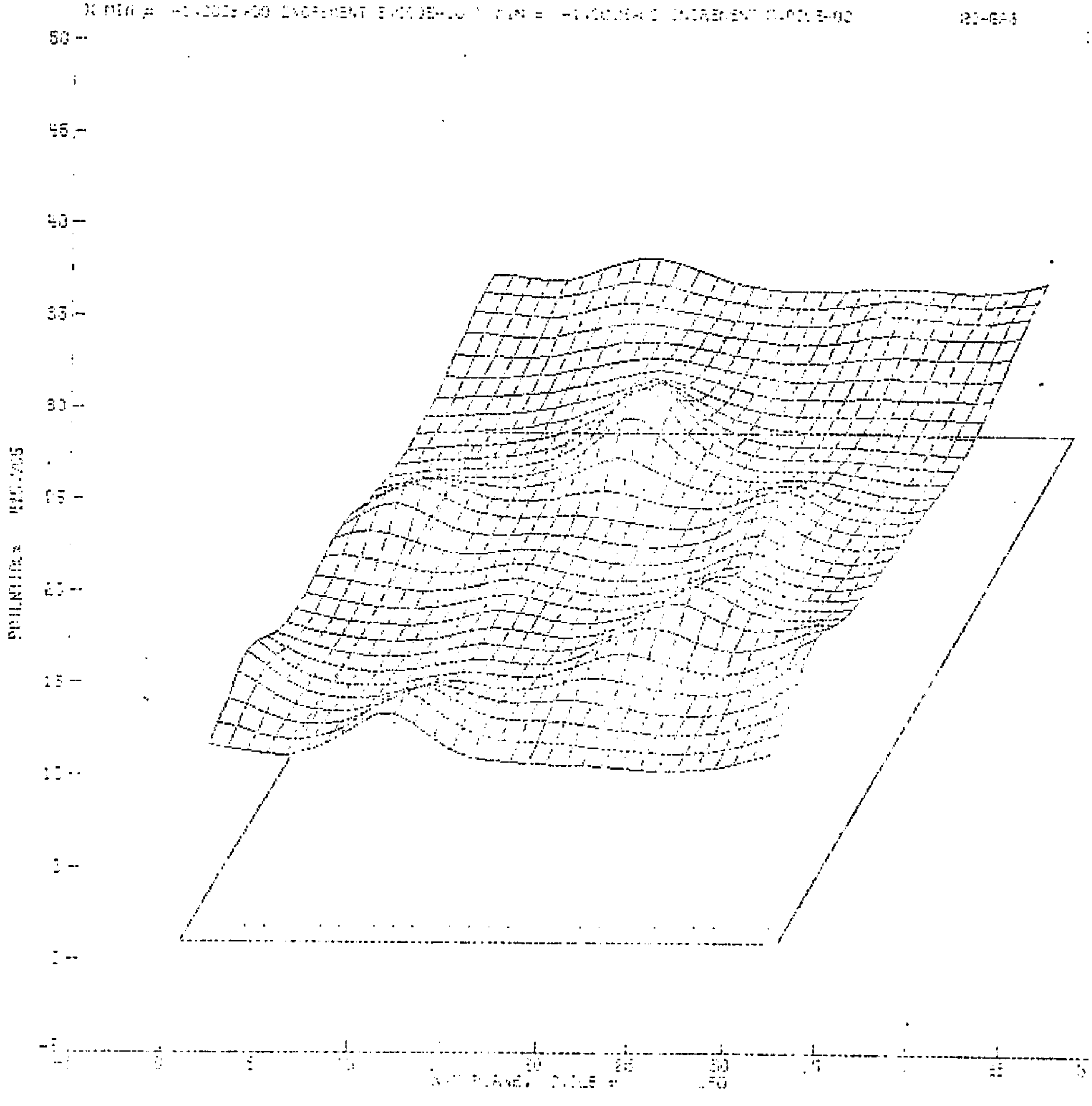


X MIN = -1.200E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
1
50
45
35
30
POTENTIAL
0
5
10
30
35

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00 2D-GAS 1

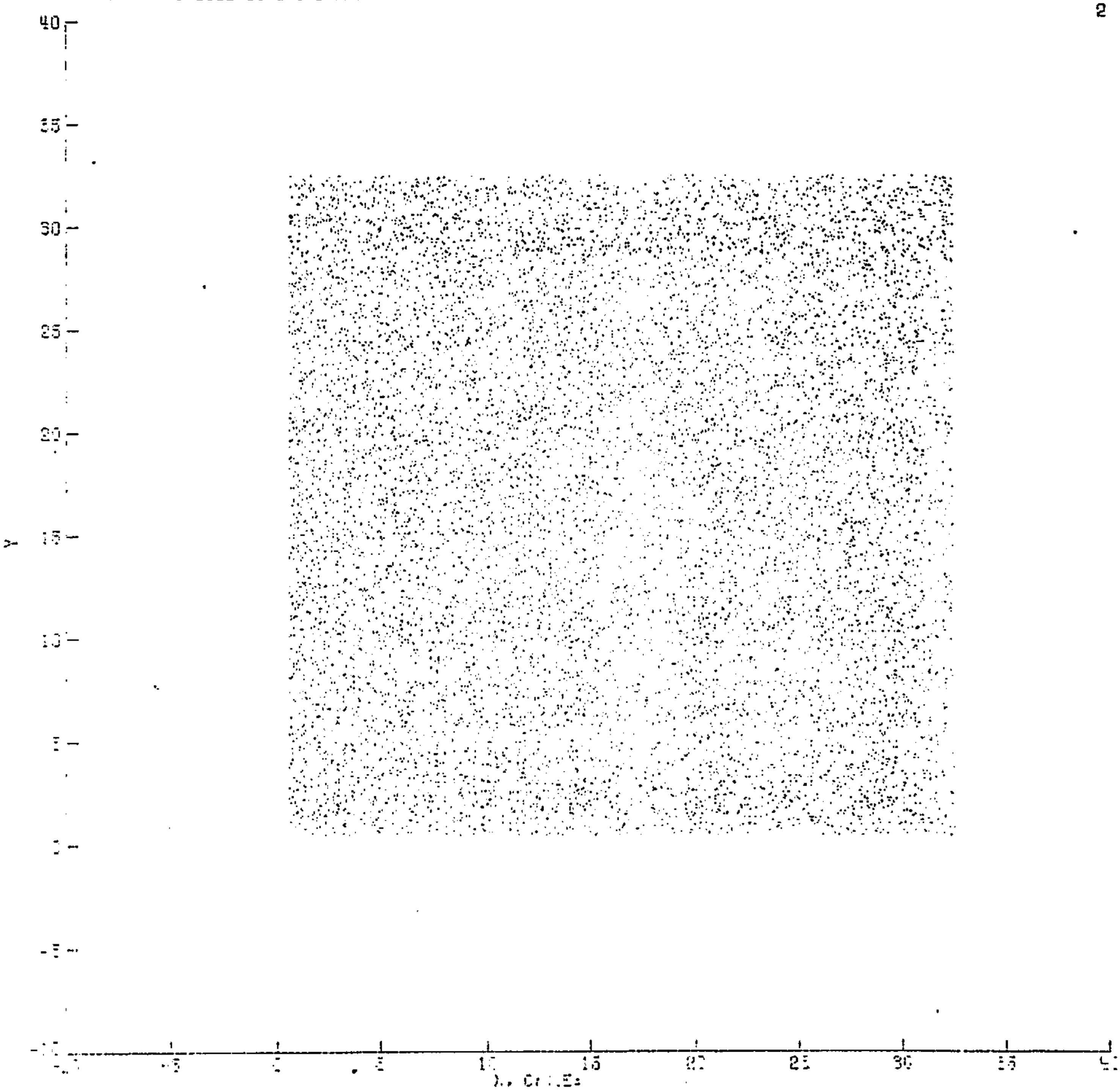


E-23

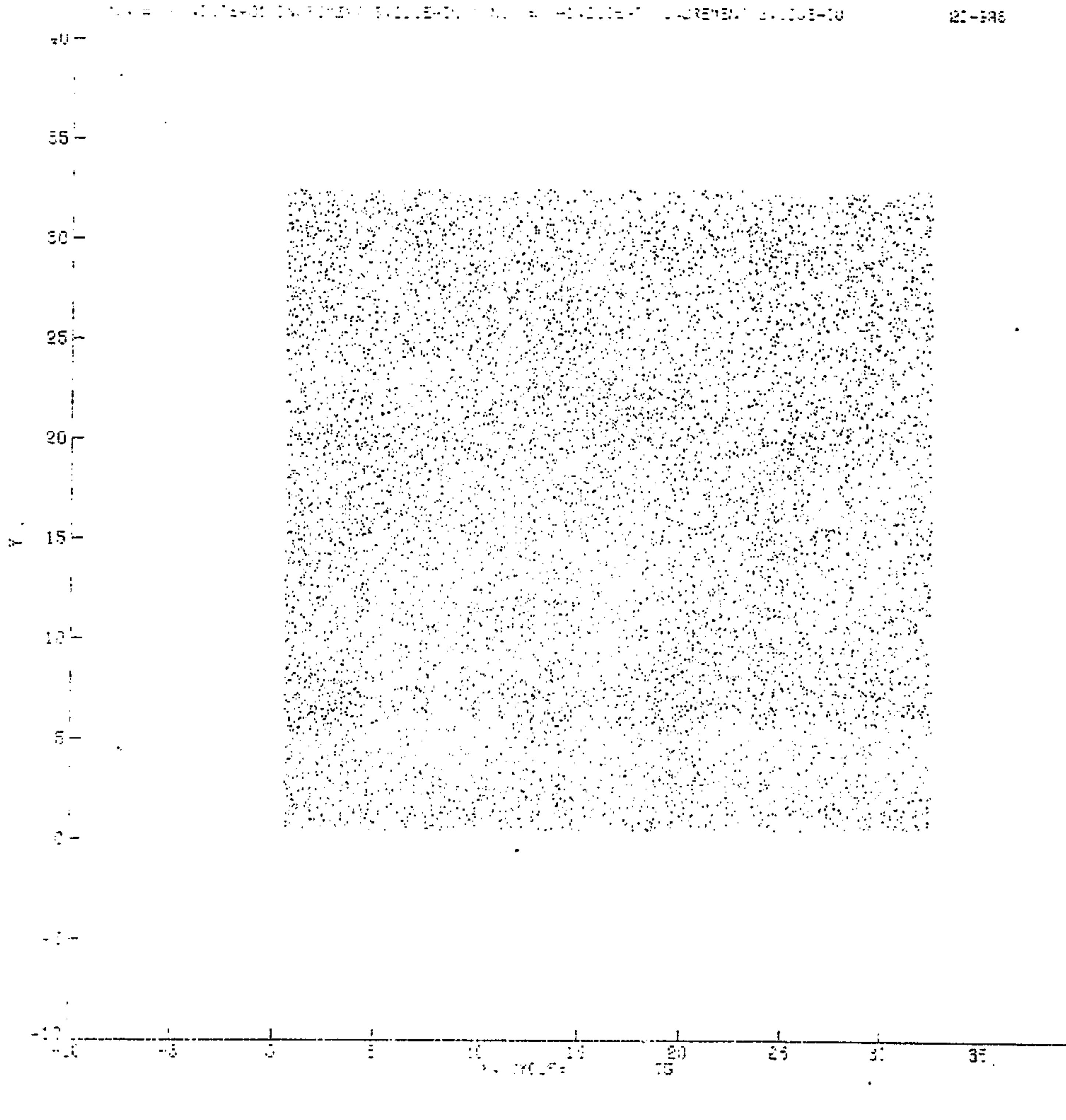


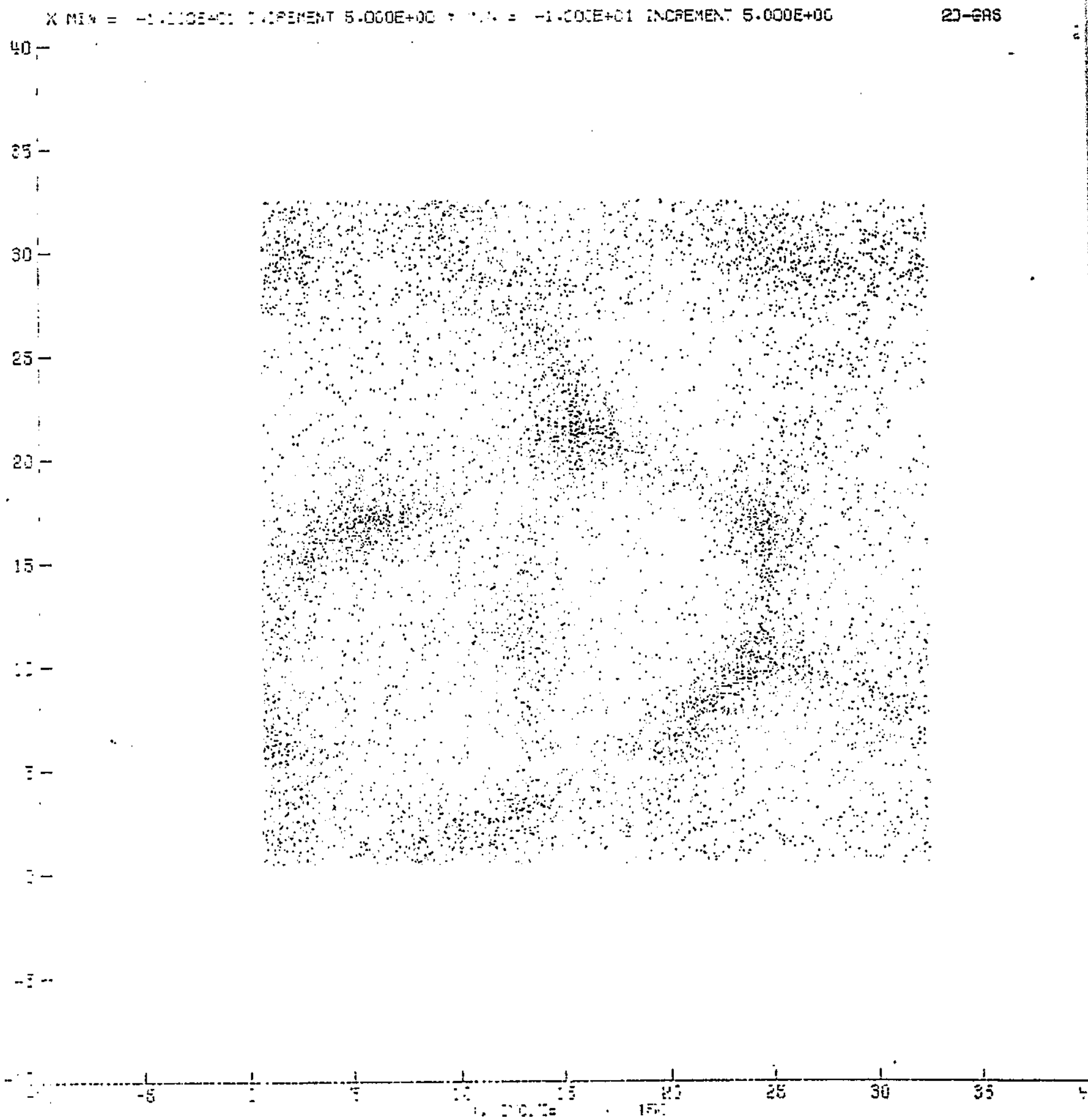
X MIN = -1.000E+01 INCREMENT 5.000E+00 Y MIN = -1.000E+01 INCREMENT 5.000E+00 2D-GAS

2

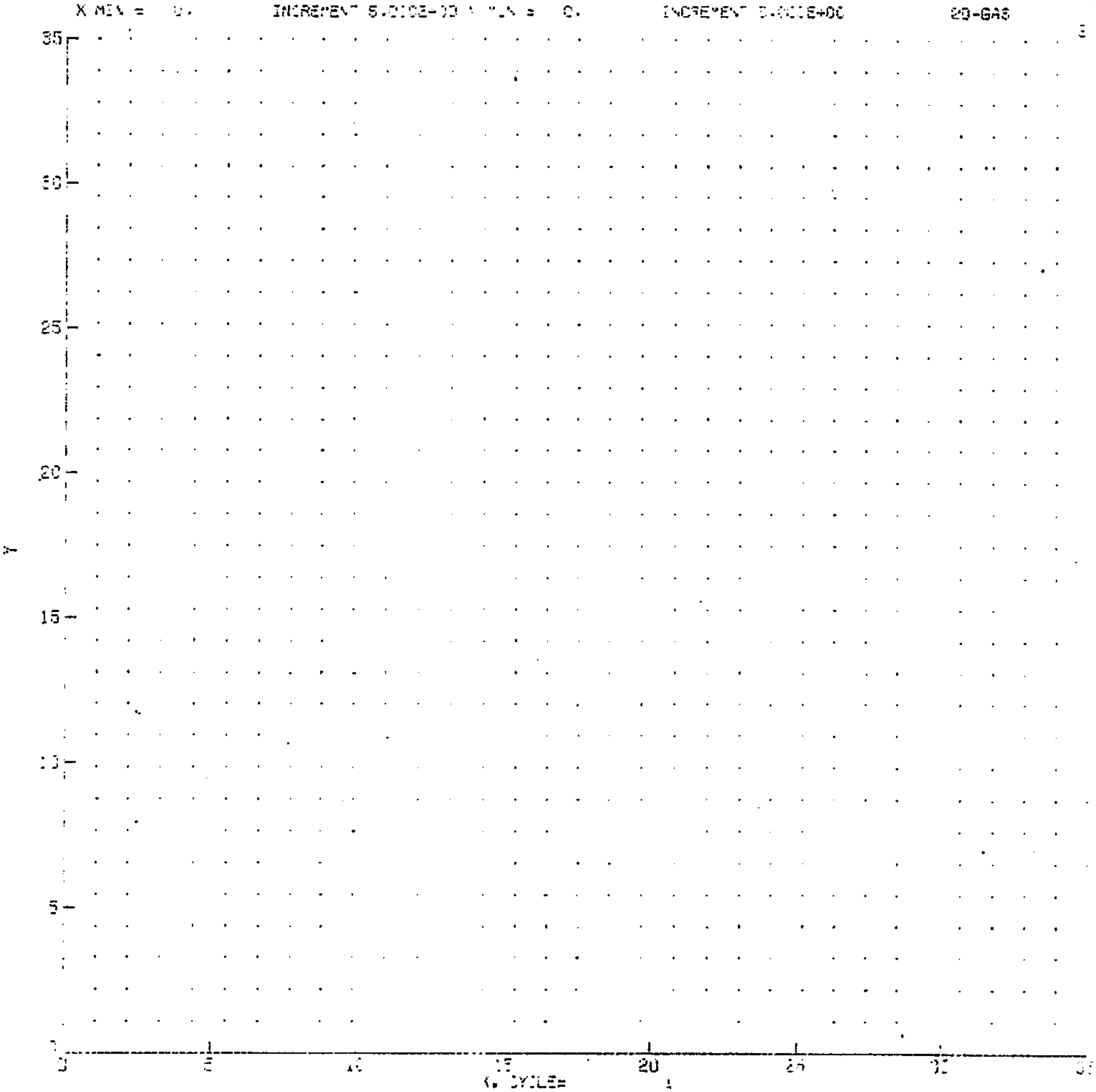


E-25

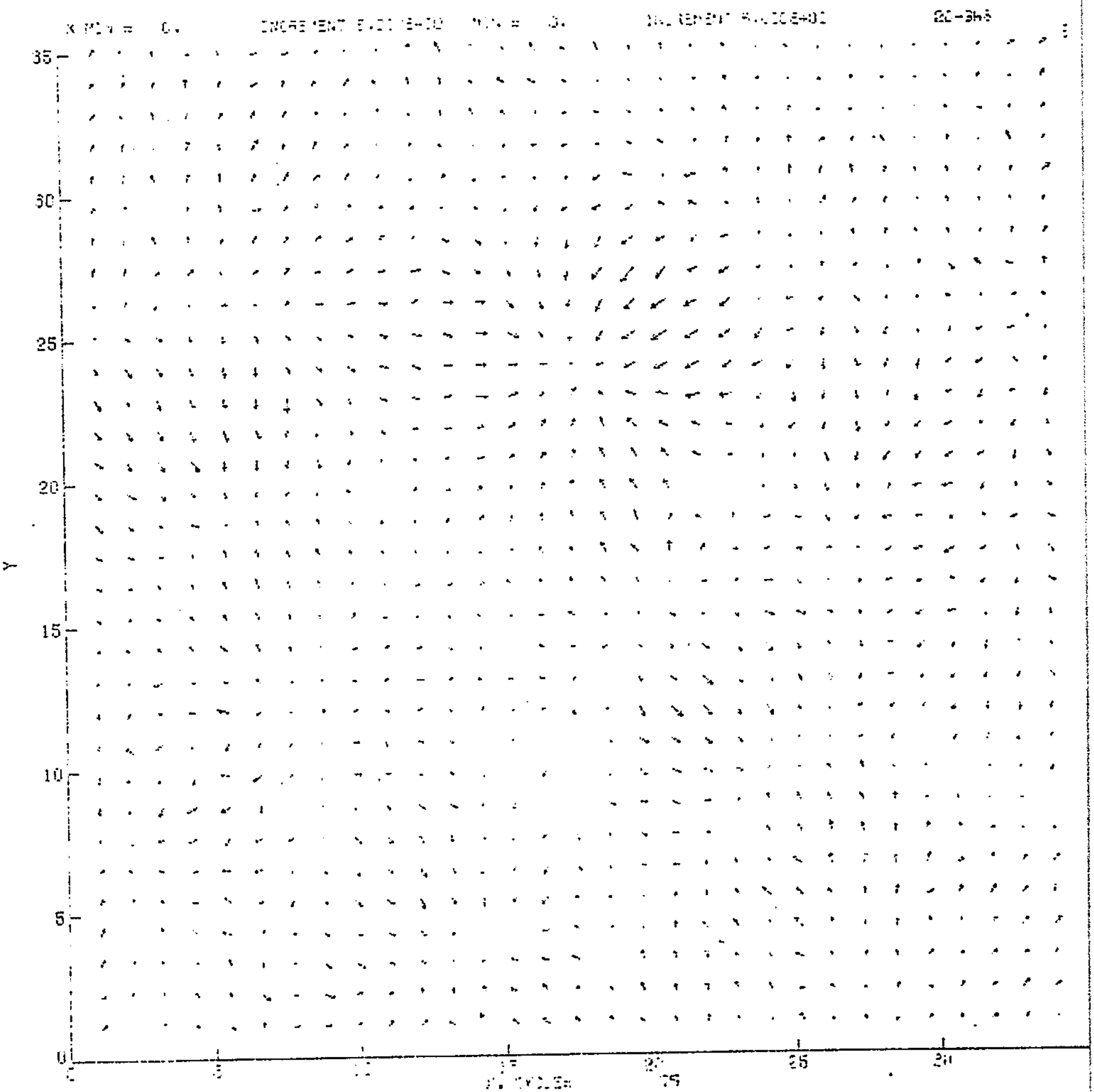

30
25
20
15
5
0
25
35

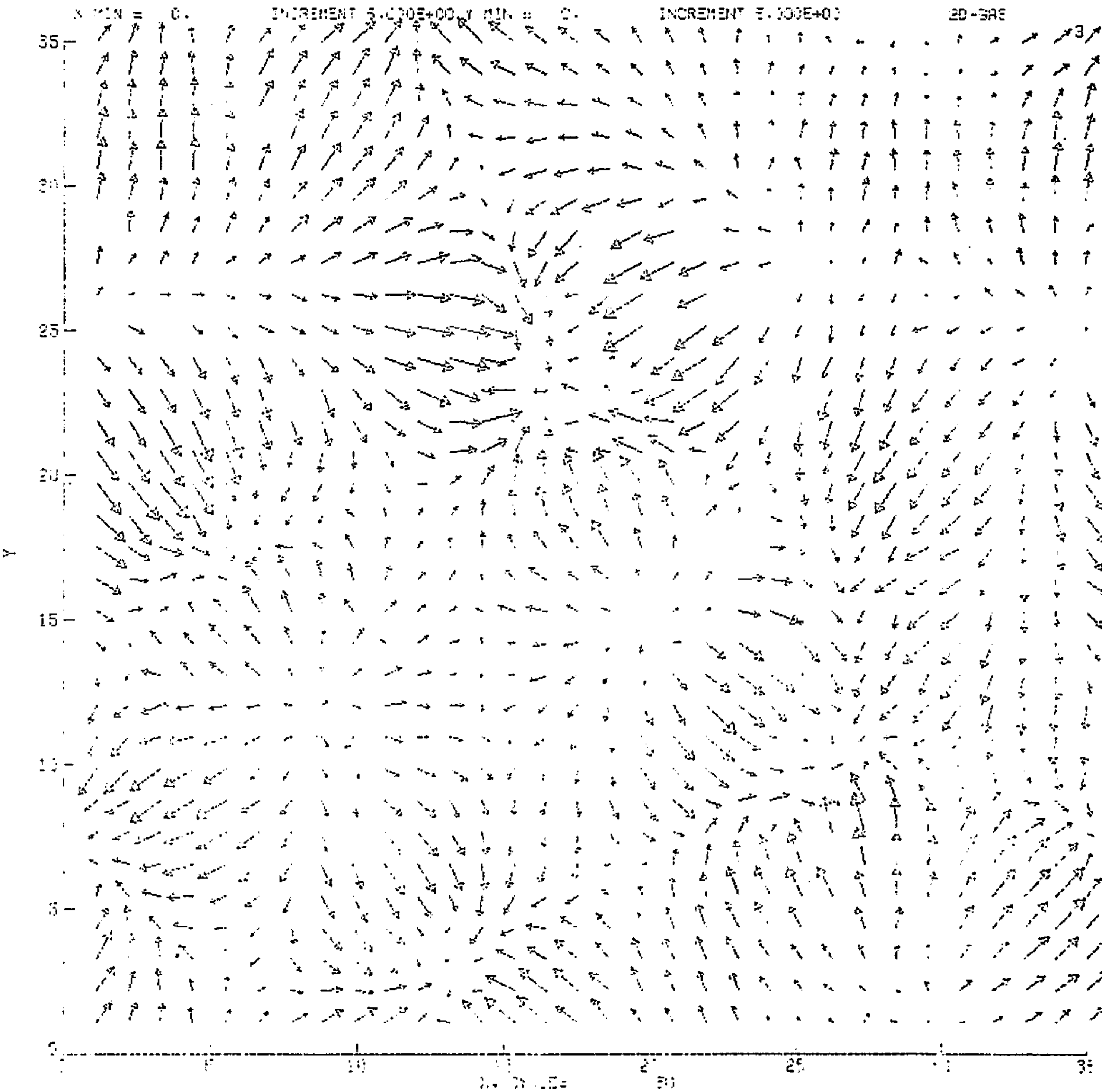


E-27

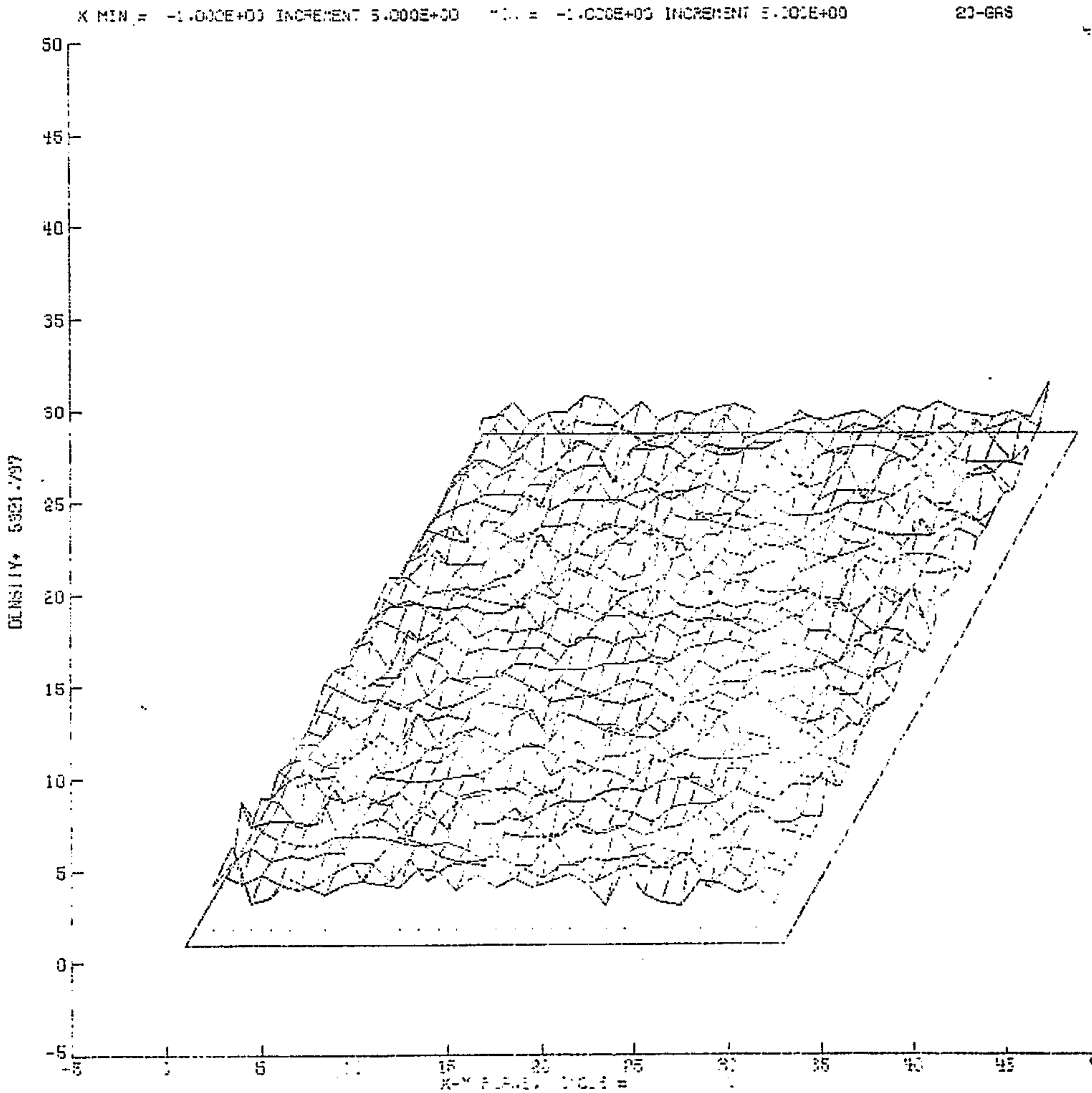


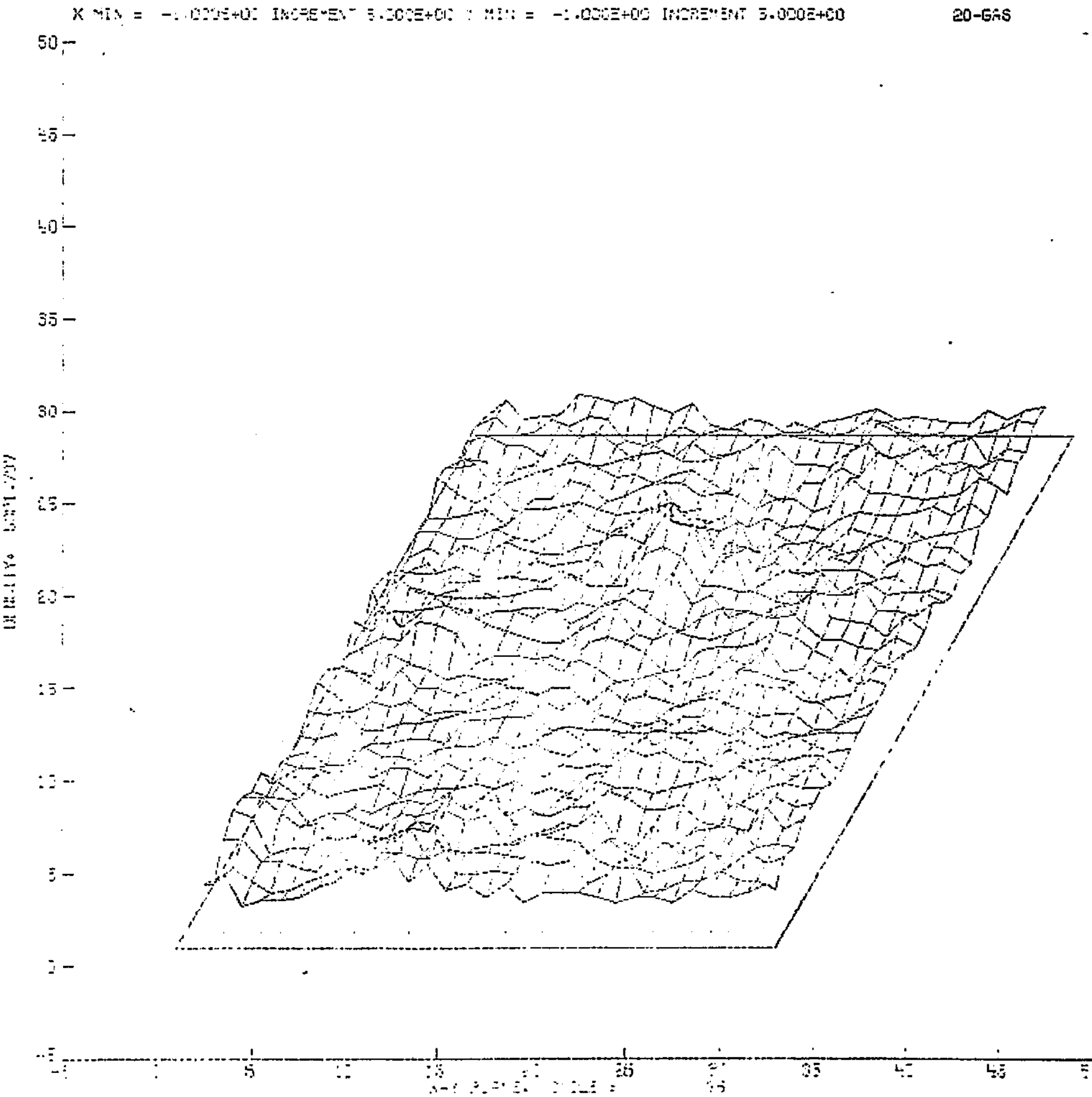
X MIN = 0.
INCREMENT 5.000E-00
Y MIN = 0.
INCREMENT 5.000E+00
2D-GAS
35
30
25
20
15
10
5
0
0
5
10
15
20
25
30
35
X, CYCLE=


X MIN = 0.
INCREMENT
Y
X, CYCLES

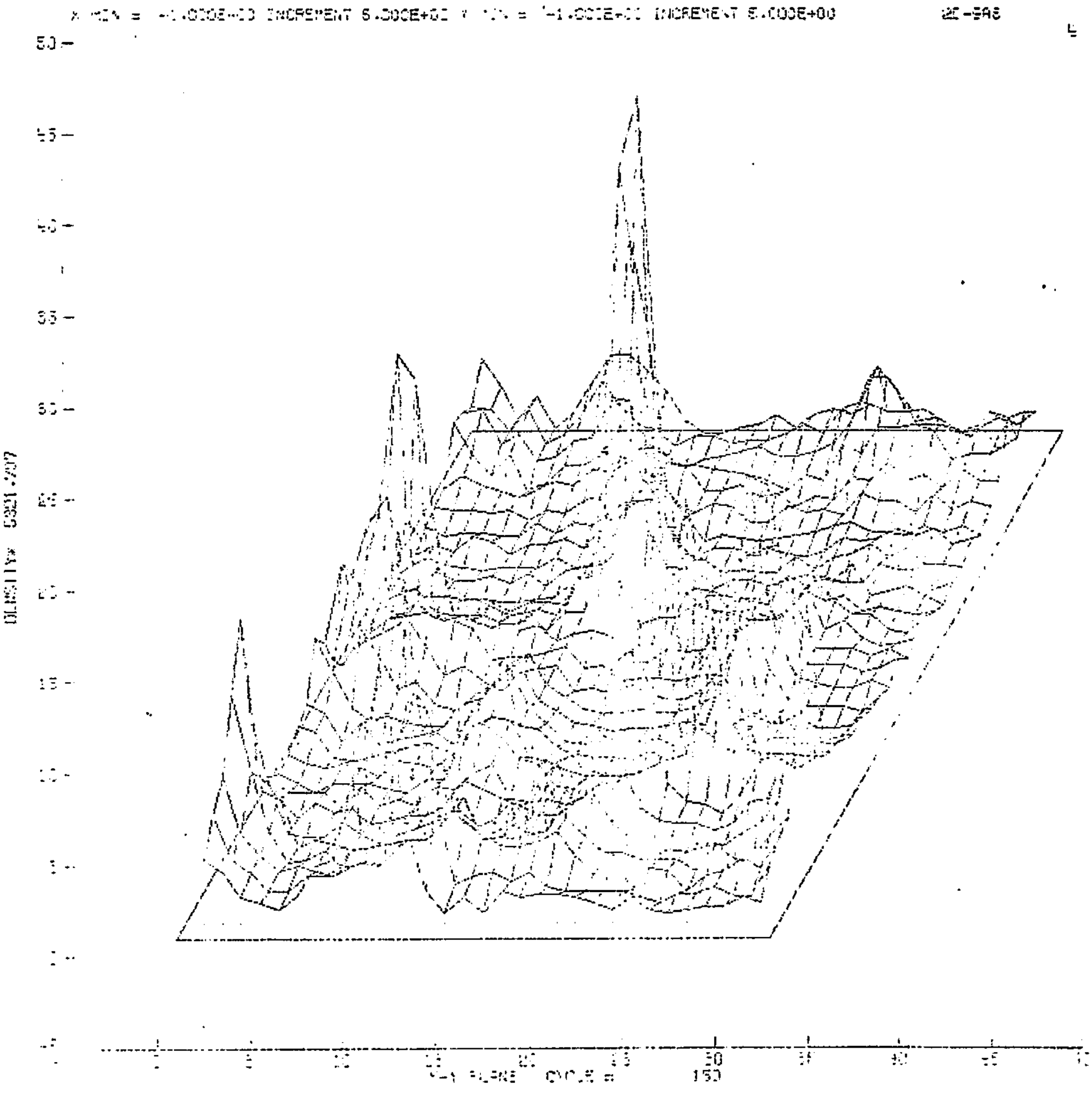


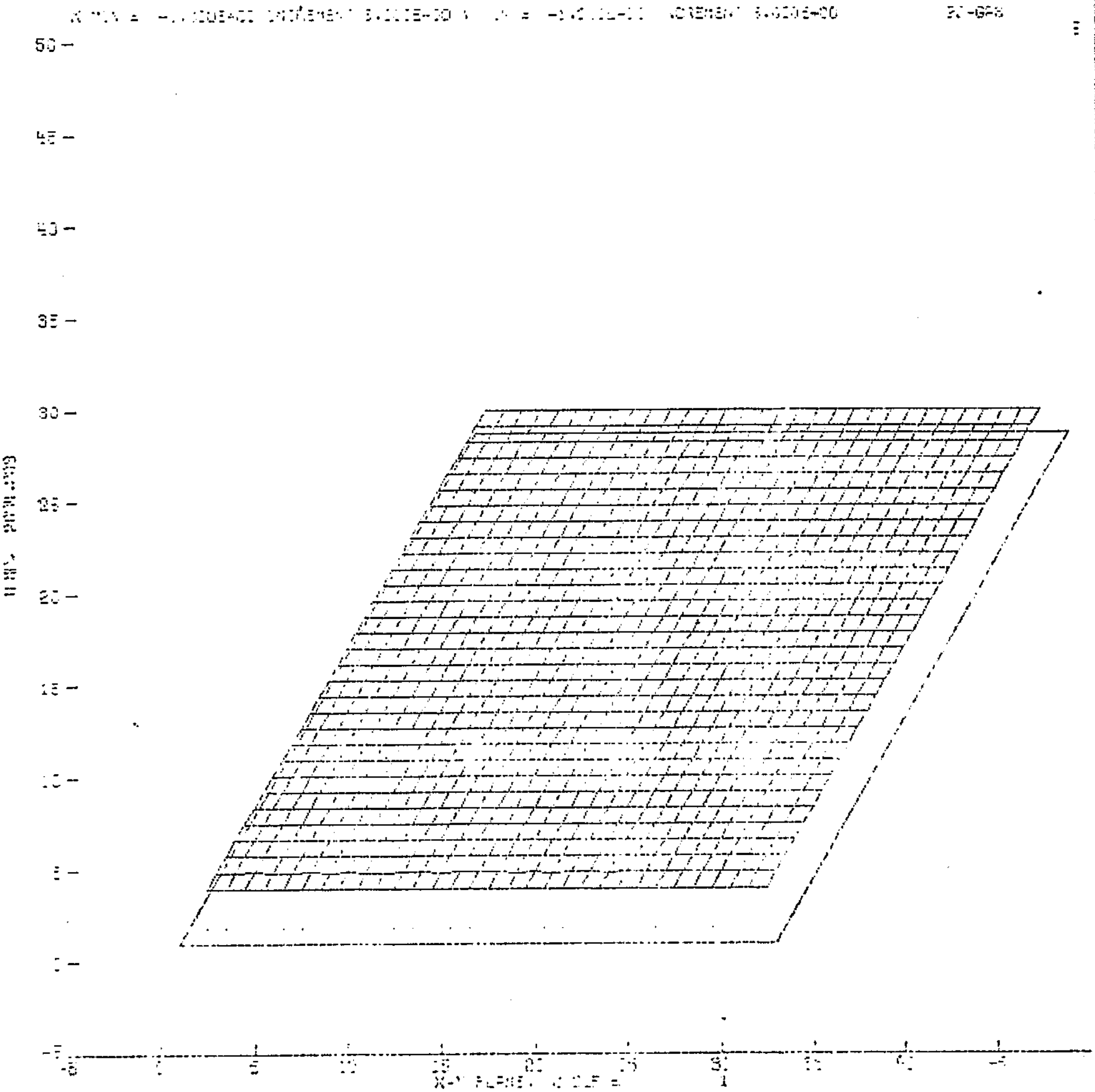
X MIN = 0. INCREMENT 5.000E+00 Y MIN = 0. INCREMENT 5.000E+00
2D-GAS
3
35
30
25
20
15
10
5
0
5
10
15
20
25
30
35
Y


X MIN = -1.000E+03 INCREMENT 5.000E+00
= -1.000E+03 INCREMENT 5.000E+00
20-GAS
DENSITY+ 5321.797
50
45
40
35
30
25
20
15
10
5
0
-5
-5
5
15
20
25
35
40
45

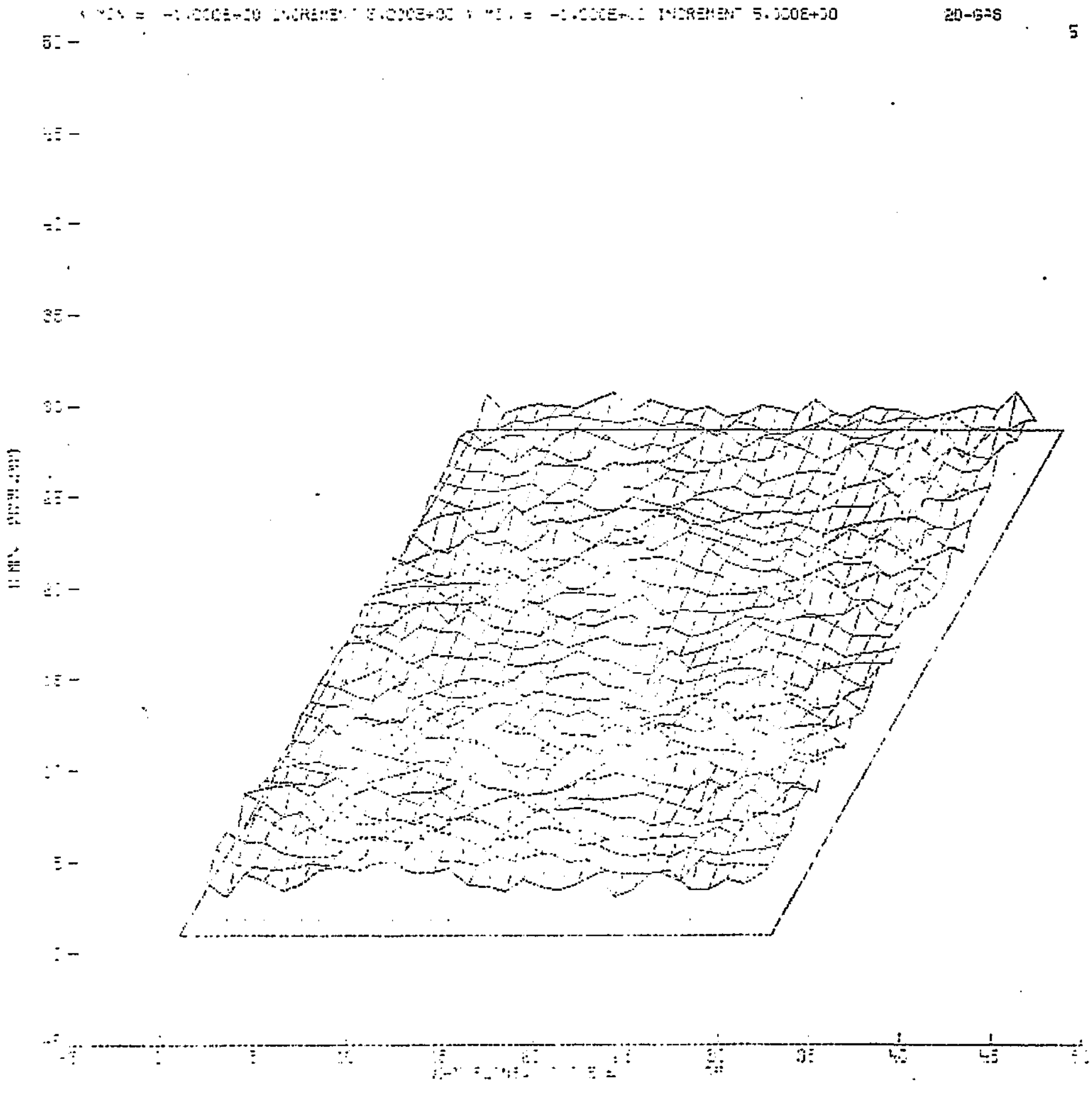


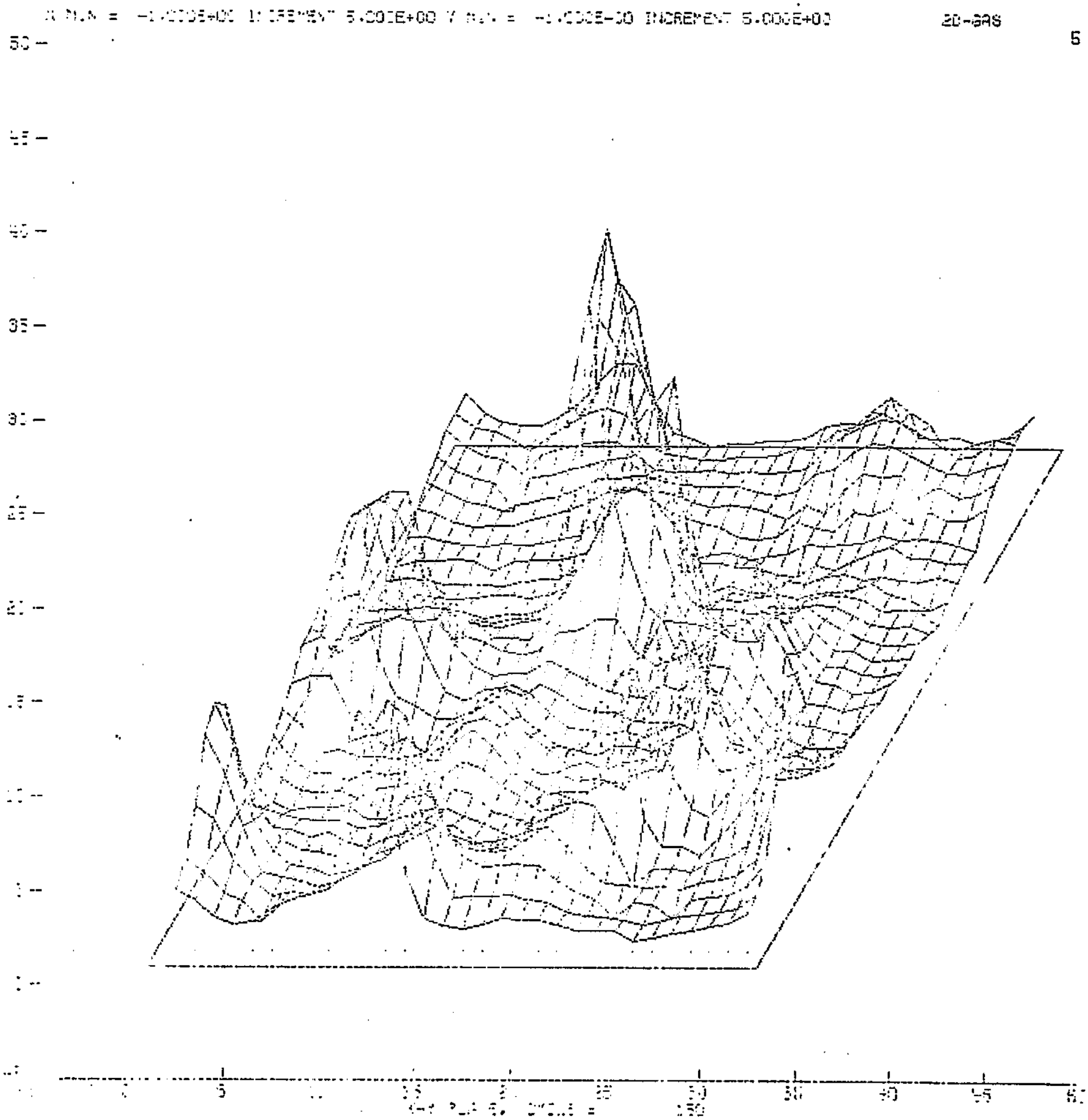
X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
20-GAS


INCREMENT 5.000E+00
CYCLE = 150



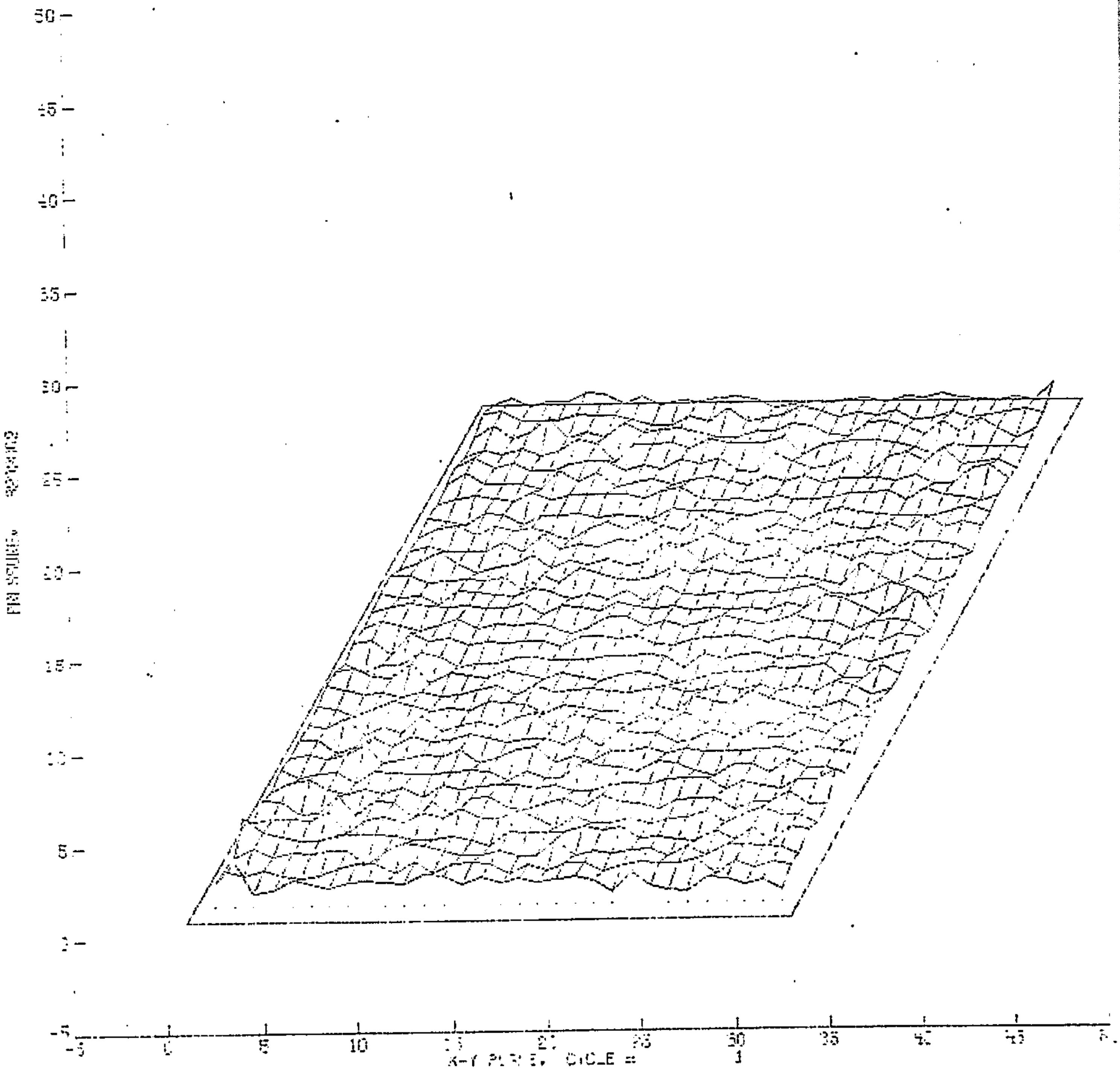
50
45
40
35
30
25
20
15
10
5
0
-5
5
30


INCREMENT 5.000E+00
20-GAS
50
45
40
35
30
25
20
15
10
5
0
5
10
15
20
25
30
35
40
45
50

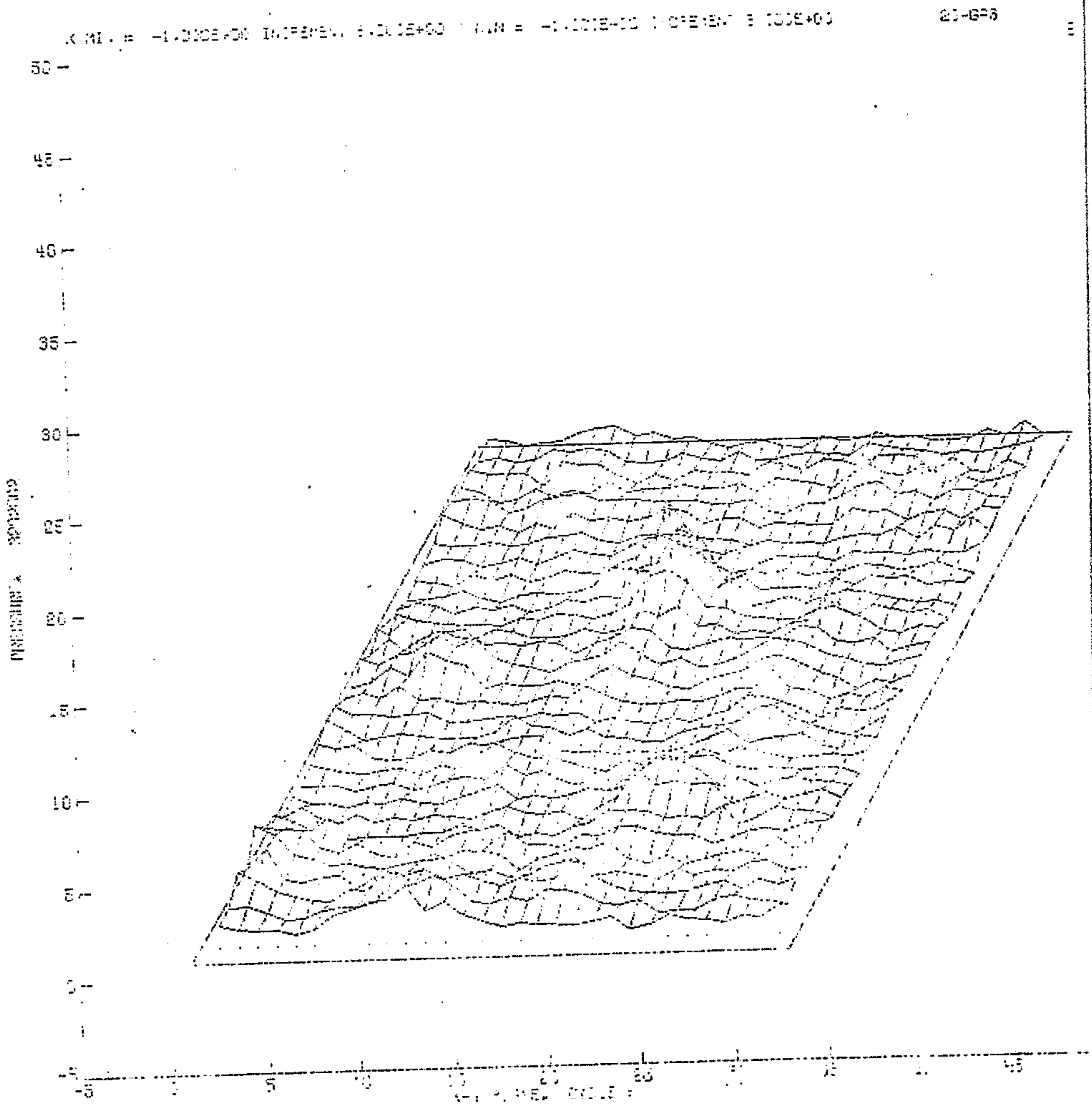

INCREMENT 5.000E+00
2D-BAS
50
45
40
35
30
5

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00 2D-GAS

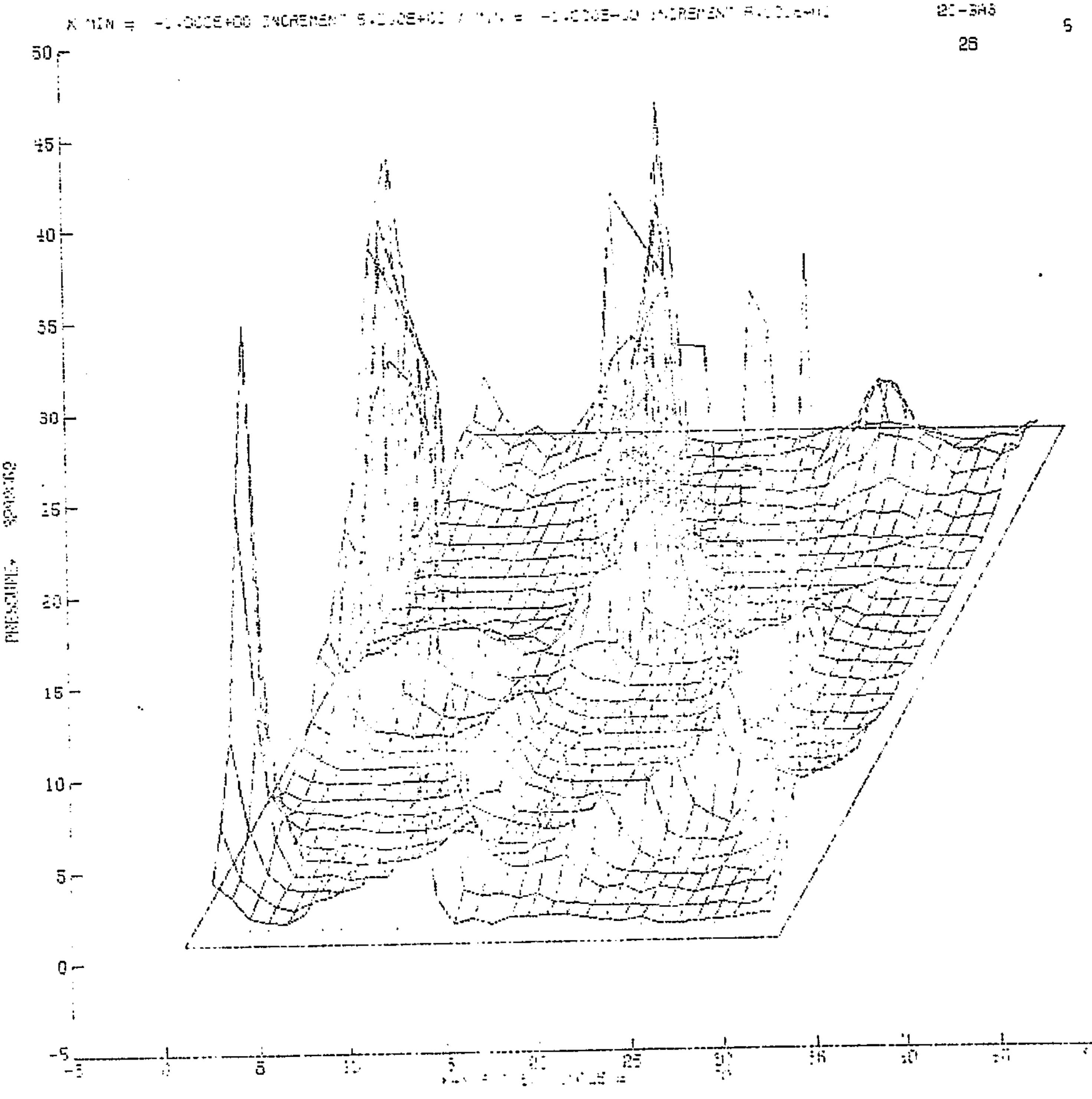
6

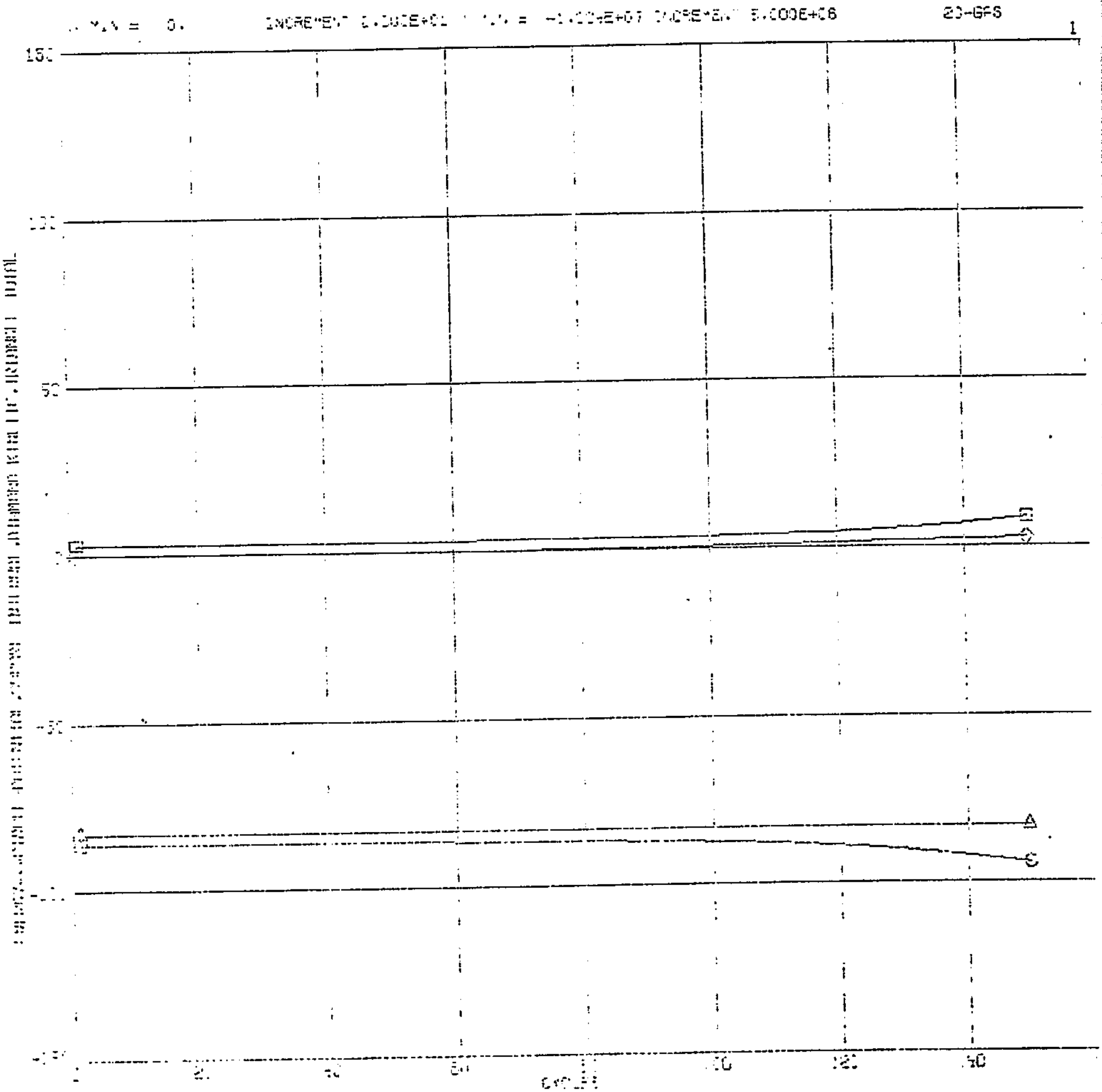


E-37



50
45
40
35
30
25
20
10


20-3A5
5
26
PRESSURE, SPACING
50
45
40
35
30
25
20
15
10
5
0
-5
-5
0
5
10
15
20
25
30
35
40
45


150
50
B
D
A
C

APPENDIX F

Listing of the Two-Dimensional Polar Coordinate Particle-in-Cell Simulator of a Rotating Self-Gravitating Compressible Gas.

Comment cards necessary to make this listing self explanatory will be added later.

```
      OVERLAY(IFILE,0,0)
      PROGRAM POLR(OUTPUT,TAPE1,TAPE8,TAPE9,TAPE10,TAPE11)
      COMMON/ALLCOM/I2A,ITEST,N,NO2,CY,CYY,RHO(32,32)
      COMMON/GASCOM/NTH(15),DTH(15),AREA(15),NPLOTG,NPRNTG,NRING,NO4,
     1    NO4M1,NO4M2,RCC,GMC,GMCP1,GMCP2,DMG,DT,DT2,DT3,DT4,NRTCEL,
     2    RMAX,CENTER,SUMPTH,SUMMAS,SUMIE,SUMKE,PESOLD,PEFOLD,XMG,
     3    SAVPTH(200),SAVPE(200),SAVIE(200),SAVKE(200),SAVTE(200),
     4    MASK1,NBSG,NBRG,JTG,JSG,ING(2),XMING,XMAXG,YMING,YMAXG,XPG(2),
     5    YPG,ITAPXG,GM1,NMKG,XYL(3),XPR(5),YPR(5),XMIN4,XMAX4,YMIN4,
     6    YMAX4,PLM(2),RCOS,RSIN,SPHIM,PLT(2),SPHIT,PLP(2),SPHIP,TWOPI,
     7    PTHMIN,PTHMAX,EMIN,EMAX,CYMIN,PI,PL(2),SPHI,MTEST,ITAPE
      INTEGER CY,CYY
      IFILE=5LIFILE
      GPFILE=6LGPFILE
      AGFILE=6LAGFILE
      MTEST=1
      ITEST=1
      CALL OVERLAY(IFILE,2,0,6HRECALL)
      IF(ITAPE.EQ.1) CALL OVERLAY(GPFILE,1,0,6HRECALL)
      IF(ITAPE.EQ.1) CALL OVERLAY(IFILE,2,0,6HRECALL)
    5 PRINT 10,CYY
   10 FORMAT(1H0*CYCLE = *I8)
      CALL SECOND(T1)
      CALL OVERLAY(GPFILE,1,0,6HRECALL)
      CALL SECOND(T2)
      T3=T2-T1
      PRINT 15,T3
   15 FORMAT(1H *FIELD TIME = *E16.8)
      CALL OVERLAY(AGFILE,3,0,6HRECALL)
      CALL SECOND(T4)
      T5=T4-T2
      PRINT 35,T5
   35 FORMAT(1H *ADVANCE TIME = *E16.8)
      IF(CYY.GE.CY) GO TO 40
      CYY=CYY+1
      GO TO 5
   40 CALL OVERLAY(IFILE,4,0,6HRECALL)
      STOP
      END
```

```
      OVERLAY(GPFILE,1,0)
      PROGRAM GETPHI
      COMMON/ALLCOM/I2A,ITEST,N,NO2,CY,CYY,RHOA(32,32)
      COMMON Z(1025),Y(1025)
      DIMENSION G(33,33),RHO(64,32)
      I2B=I2A-1
      N21=NO2+1
      IF(ITEST.EQ.0) GO TO 10
      ITEST=0
      RNI=1./(N*N)
      DO 1 J=1,N21
      DO 1 I=1,N21
      IF(I.EQ.1.AND.J.EQ.1) GO TO 1
      G(I,J)=RNI/SQRT((I-1.)*(I-1.)+(J-1.)*(J-1.))
    1 CONTINUE
      G(1,1)=G(1,2)
      CALL GETSET(2,I2B)
      DO 2 J=1,N21
      DO 3 I=1,N21
    3 Z(I)=G(I,J)
      CALL FTRANS(2,I2B)
      DO 4 I=1,N21
    4 G(I,J)=Y(I)
    2 CONTINUE
      DO 5 I=1,N21
      DO 6 J=1,N21
    6 Z(J)=G(I,J)
      CALL FTRANS(2,I2B)
      DO 7 J=1,N21
    7 G(I,J)=Y(J)
    5 CONTINUE
      WRITE(1) G
      REWIND 1
   10 READ(1) G
      REWIND 1
      CALL GETSET(3,I2A)
      DO 11 J=1,NO2
      DO 8 I=1,NO2
      Z(I)=RHOA(I,J)
    8 Z(NO2+I)=0.
```

```
      CALL FTRANS(3,I2A)
      DO 9 I=1,N
    9 RHO(I,J)=Y(I)
   11 CONTINUE
      DO 12 I=1,N
      DO 13 J=1,NO2
      Z(J)=RHO(I,J)
   13 Z(J+NO2)=0.
      CALL GETSET(3,I2A)
      CALL FTRANS(3,I2A)
      IF(I.GT.N21) GO TO 14
      DO 15 J=2,NO2
      Z(J)=Y(J)*G(I,J)
   15 Z(J+NO2)=Y(J+NO2)*G(I,J)
      Z(1)=Y(1)*G(I,1)
      Z(N21)=Y(N21)*G(I,N21)
      GO TO 16
   14 DO 17 J=2,NO2
      Z(J)=Y(J)*G(I-NO2,J)

   17 Z(J+NO2)=Y(J+NO2)*G(I-NO2,J)
      Z(1)=Y(1)*G(I-NO2,1)
      Z(N21)=Y(N21)*G(I-NO2,N21)
   16 CONTINUE
      CALL GETSET(4,I2A)
      CALL FTRANS(4,I2A)
      DO 18 J=1,NO2
   18 RHO(I,J)=Y(J)
   12 CONTINUE
      DO 19 J=1,NO2
      DO 20 I=1,N
   20 Z(I)=RHO(I,J)
      CALL FTRANS(4,I2A)
      DO 21 I=1,NO2
   21 RHOA(I,J)=Y(I)
   19 CONTINUE
      RETURN
      END
```

```
      OVERLAY(IFILE,2,0)
      PROGRAM INIGAS
      COMMON/ALLCOM/I2A,ITEST,N,NO2,CY,CYY,RHO(32,32)
      COMMON/GASCOM/NTH(15),DTH(15),AREA(15),NPLOTG,NPRNTG,NRING,NO4,
     1    NO4M1,NO4M2,RCC,GMC,GMCP1,GMCP2,DMG,DT,DT2,DT3,DT4,NRTCEL,
     2    RMAX,CENTER,SUMPTH,SUMMAS,SUMIE,SUMKE,PESOLD,PEFOLD,XMG,
     3    SAVPTH(200),SAVPE(200),SAVIE(200),SAVKE(200),SAVTE(200),
     4    MASK1,NBSG,NBRG,JTG,JSG,ING(2),XMING,XMAXG,YMING,YMAXG,XPG(2),
     5    YPG,ITAPXG,GM1,NMKG,XYL(3),XPR(5),YPR(5),XMIN4,XMAX4,YMIN4,
     6    YMAX4,PLM(2),RCOS,RSIN,SPHIM,PLT(2),SPHIT,PLP(2),SPHIP,TWOPI,
     7    PTHMIN,PTHMAX,EMIN,EMAX,CYMIN,PI,PL(2),SPHI,MTEST,ITAPE
      DIMENSION RADIUS(1000),THETA(1000),RDOT(15,64),TDOT(15,64),
     1    IE(15,64),MASS(15,64),PRES(15,64),XNWAV(15,64),YNWAV(15,64),
     2    R2MDAV(15,64),R2NWAV(15,64),GR(15),KAPPA2(15)
      INTEGER CY,CYY
      REAL IE,MINIE,KAPPA,KAPPA2,MASS,NUMBER
C SET INITIAL VALUES
C     SET I2A WHERE (2**I2A)/2 IS THE SIZE OF THE ACTIVE MESH
      I2A=6
C     SET NUMBER OF CYCLES
      CY=60
C     SET TOTAL NUMBER OF GAS PARTICLES
      NBRG=1000
C     SET NUMBER OF GAS PARTICLES PER READ OF GAS PARTICLE FILE
      NBSG=10
C     SET INITIAL RADIUS OF GAS
      RIG=10.
C     SET TOTAL MASS OF GALAXY
      GMC=3.55E6
C     SET FRACTION OF GALAXY MASS WHICH IS GAS
      PERC=.05
C     SET RADIUS OF FIXED SPHERE OF STARS
      RCC=6.
C     SET INITIAL MAXIMUM RATIO OF TIME SCALED ANGULAR VELOCITY TO ANGULAR CELL
C     WIDTH.
      ARATIO=.3
C     SET NUMBER OF POINTS PER GAS PLOT
      NPG=NBRG
C     SET PERIOD OF GAS LONG PRINTING
      NPRNTG=60
C     SET PERIOD OF PRINTING OF GAS ANGULARLY AVERAGED (RING) QUANTITIES
```

```
      NRING=30
C     SET PERIOD OF GAS PLOTTING
      NPLOTG=30
C     SET RATIO OF SPECIFIC HEAT AT CONSTANT VOLUME TO SPECIFIC HEAT AT
C     CONSTANT PRESSURE. (GAMMA MUST BE GREATER THAN OR EQUAL TO ONE.
C     GAMMA IS EQUAL TO 2.0 FOR NORMAL SIMULATION OF MONATOMIC GAS IN
C     TWO DIMENSIONS. GAMMA MAY BE SET EQUAL TO 1.0 FOR A SIMULATION
C     WITHOUT PRESSURE TERMS.)
      GAMMA=2.0
C     SET RATIO OF THE VELOCITY DISPERSION OF THE GAS (SQRT OF SPECIFIC
C     INTERNAL ENERGY) TO THAT REQUIRED FOR LOCAL STABILITY
      QGAS=1.0
C     SET MINIMUM INITIAL SPECIFIC INTERNAL ENERGY
      MINIE=1.
C     SET ITAPE=1 TO START RUN. SET ITAPE=2 TO CONTINUE RUN WITH PICK UP TAPE.
      ITAPE=1
C SET CONSTANTS
      PI=3.1415926536
      TWOPI=2.*PI
      MASK1=077777777770000000000
      N=2**I2A
      NO2=N/2
      NO4=N/4
      NO4M1=NO4-1
      NO4M2=NO4-2
      NRTCEL=N*NO4M1
      JTG=9
      JSG=10
      GM1=GAMMA-1.
      RIG2=RIG**2
      NMKG=NPG-NBSG
      XMG=GMC*PERC/NBRG
      XMG=XMG.AND.MASK1
      GMC=GMC*(1.-PERC)
      CENTER=NO4+.5
      RMAX=NO4-2.0001
      NTH(1)=4
      NTH(2)=8
      NTH(3)=16
      NTH(4)=16
```

```
      NTH(5)=32
      NTH(6)=32
      NTH(7)=32
      NTH(8)=32
      NTH(9)=64
      NTH(10)=64
      NTH(11)=64
      NTH(12)=64
      NTH(13)=64
      NTH(14)=64
      NTH(15)=64
      IF(NO4.EQ.16) GO TO 20
      NTH(16)=64
      DO 10 I=17,31
      NTH(I)=128
   10 CONTINUE
   20 DO 30 I=1,NO4M1
      DTH(I)=TWOPI/NTH(I)
      AREA(I)=DTH(I)*(I-.5)
   30 CONTINUE
C SET PLOTTING SPECIFICATIONS
      ING(1)=10H2D-GAS
      ING(2)=10H
      XMING=-10
      XMAXG=40
      YMING=-10
      YMAXG=40
      XPG(1)=10HX,CYCLE=
      YPG=10H    Y
      ITAPXG=6LTAPE23
      XYL(1)=10HX-Y PLANE,
      XYL(2)=10H CYCLE =
      PL(1)=10HPOTENTIAL*
      PLP(1)=10H PRESSURE*
      PLT(1)=10H     TEMP*
      PLM(1)=10H  DENSITY*
      ANG=PI/3.
      RCOS=COS(ANG)
      RSIN=SIN(ANG)
      IC=NO2+1
```

```
      XPR(1)=1.+NO2*RCOS
      XPR(2)=1.
      XPR(3)=1.+NO2
      XPR(4)=IC+NO2*RCOS
      XPR(5)=XPR(1)
      YPR(1)=1.+NO2*RSIN
      YPR(2)=1.
      YPR(3)=1.
      YPR(4)=1.+NO2*RSIN
      YPR(5)=YPR(1)
      XMAX4=IC+NO2*RCOS
      XMIN4=-1.
      YMIN4=-1.
      YMAX4=XMAX4
      HT=IC+NO2*RCOS-NO2*RSIN
C END OF INITIAL DATA
      IF(ITAPE.EQ.2) GO TO 350
      CYY=1
      IF(MTEST.EQ.0) GO TO 160
      MTEST=0
      DO 105 I=1,NO4M1
      NNTH=NTH(I)
      DO 105 J=1,NNTH
      RDOT(I,J)=0.
      MASS(I,J)=0.
      XNWAV(I,J)=0.
      YNWAV(I,J)=0.
      R2MDAV(I,J)=0.
      R2NWAV(I,J)=0.
  105 CONTINUE
      DO 107 I=1,NO2
      DO 107 J=1,NO2
      RHO(I,J)=0.
  107 CONTINUE
      X=URAN(7654321.)
      NS2=0
  110 DO 130 IS=1,NBSG
  120 X=2.*(URAN(0.)-.5)
      Y=2.*(URAN(0.)-.5)
      Z7=2.*(URAN(0.)-.5)
```

```
      R2=X*X+Y*Y+Z7*Z7
      IF(R2.GT.1.) GO TO 120
      XC=RIG*X
      YC=RIG*Y
      IF(XC.EQ.0.) XC=.00001
      R2=XC*XC+YC*YC
      R=SQRT(R2)
      TH=ATAN2(YC,XC)
      IF(TH.LT.0.) TH=TH+TWOPI
      IF(TH.GE.TWOPI) TH=TH-TWOPI
      IR=R+1
      JT=TH/DTH(IR)+1
      IX=CENTER+XC+.5
      JY=CENTER+YC+.5
      XNWAV(IR,JT)=XNWAV(IR,JT)+XC
      YNWAV(IR,JT)=YNWAV(IR,JT)+YC
      R2MDAV(IR,JT)=R2MDAV(IR,JT)+R2
      R2NWAV(IR,JT)=R2NWAV(IR,JT)+R2
      MASS(IR,JT)=MASS(IR,JT)+XMG
      RHO(IX,JY)=RHO(IX,JY)+.5*XMG
      RADIUS(IS)=R
      THETA(IS)=TH
  130 CONTINUE
      WRITE(9) RADIUS,THETA
      NS2=NS2+NBSG
      IF(NS2.LT.NBRG) GO TO 110
      REWIND 9
      DO 140 I=1,NO4M2
      NNTH=NTH(I)
      DO 140 J=1,NNTH
      IF(MASS(I,J).EQ.0.) GO TO 140
      NUMBER=MASS(I,J)/XMG
      XNWAV(I,J)=XNWAV(I,J)/NUMBER
      YNWAV(I,J)=YNWAV(I,J)/NUMBER
      R2MDAV(I,J)=R2MDAV(I,J)/NUMBER
      R2NWAV(I,J)=R2NWAV(I,J)/NUMBER
  140 CONTINUE
      WRITE(8) RDOT,MASS,XNWAV,YNWAV,R2MDAV,R2NWAV
      REWIND 8
      RETURN
```

```
160 READ(8) RDOT,MASS,XNWAV,YNWAV,R2MDAV,R2NWAV
    REWIND 8
    DO 170 I=1,NO4M2
    GR(I)=0.
    NNTH=NTH(I)
    DO 165 J=1,NNTH
    R=I-.5
    TH=(J-.5)*DTH(I)
    XCELL=R*COS(TH)
    YCELL=R*SIN(TH)
    X=CENTER+XCELL
    Y=CENTER+YCELL
    IX=X
    IY=Y
    XX=IX-X
    YY=IY-Y
    RAD3=R*R*R
    IF(R.LT.RCC) RAD3=RCC*RCC*RCC
    GX=(YY+1.)*((XX+1.)*(RHO(IX+1,IY)-RHO(IX-1,IY))-XX*(RHO(IX+2,IY)-R
   1HO(IX,IY)))-YY*((XX+1.)*(RHO(IX+1,IY+1)-RHO(IX-1,IY+1))-XX*(RHO(IX
   2+2,IY+1)-RHO(IX,IY+1)))-GMC*XCELL/RAD3
    GY=(XX+1.)*((YY+1.)*(RHO(IX,IY+1)-RHO(IX,IY-1))-YY*(RHO(IX,IY+2)-R
   1HO(IX,IY)))-XX*((YY+1.)*(RHO(IX+1,IY+1)-RHO(IX+1,IY-1))-YY*(RHO(IX
   2+1,IY+2)-RHO(IX+1,IY)))-GMC*YCELL/RAD3
    GR(I)=GR(I)+(XCELL*GX+YCELL*GY)/R
165 CONTINUE
    GR(I)=GR(I)/NNTH
170 CONTINUE
    NO4M3=NO4-3
    DO 175 I=2,NO4M3
    R=I-.5
    KAPPA2(I)=(GR(I+1)-GR(I-1))/2.+3.*GR(I)/R
175 CONTINUE
    KAPPA2(1)=(GR(2)-0.)/2.+3.*GR(1)/.5
    KAPPA2(NO4M2)=(GR(NO4M2)-GR(NO4M3))/1.+3.*GR(NO4M2)/(NO4M2-.5)
    KAPPA2(NO4M1)=0.
    DO 185 I=1,NO4M1
    KAPPA2(I)=-KAPPA2(I)
    IF(KAPPA2(I).LT.0.) KAPPA2(I)=0.
    KAPPA=SQRT(KAPPA2(I))
    NNTH=NTH(I)
```

```
      DO 185 J=1,NNTH
      IF(MASS(I,J).EQ.0.) GO TO 182
      IF(KAPPA.EQ.0.) GO TO 178
      IE(I,J)=(3.36*QGAS*MASS(I,J)/(AREA(I)*KAPPA))**2+MINIE
      GO TO 180
  178 IE(I,J)=MINIE
  180 PRES(I,J)=GM1*IE(I,J)*MASS(I,J)/AREA(I)
      GO TO 185
  182 IE(I,J)=0.
      PRES(I,J)=0.
  185 CONTINUE
      AMAX=0.
      ISAV=1
      DO 240 I=1,NO4M2
      IPTEST=0
      IF(I.EQ.1) IPTEST=1
      IF(I.EQ.2) IPTEST=1
      IF(I.EQ.4) IPTEST=1
      IF(I.EQ.8) IPTEST=1
      IF(I.EQ.16) IPTEST=1
      IMTEST=0
      IF(I.EQ.2) IMTEST=1
      IF(I.EQ.3) IMTEST=1
      IF(I.EQ.5) IMTEST=1
      IF(I.EQ.9) IMTEST=1
      IF(I.EQ.17) IMTEST=1
      NNTH=NTH(I)
      DO 240 J=1,NNTH
      IF(MASS(I,J).EQ.0.) GO TO 230
      XNEW=XNWAV(I,J)
      YNEW=YNWAV(I,J)
      RNEW=SQRT(XNEW*XNEW+YNEW*YNEW)
      X=CENTER+XNEW
      Y=CENTER+YNEW
      IX=X
      IY=Y
      XX=IX-X
      YY=IY-Y
      RAD3=RNEW*RNEW*RNEW
      IF(RNEW.LT.RCC) RAD3=RCC*RCC*RCC
      GX=(YY+1.)*((XX+1.)*(RHO(IX+1,IY)-RHO(IX-1,IY))-XX*(RHO(IX+2,IY)-R
```

```
   1HO(IX,IY)))-YY*((XX+1.)*(RHO(IX+1,IY+1)-RHO(IX-1,IY+1))-XX*(RHO(IX
   2+2,IY+1)-RHO(IX,IY+1)))-GMC*XNEW/RAD3
    GY=(XX+1.)*((YY+1.)*(RHO(IX,IY+1)-RHO(IX,IY-1))-YY*(RHO(IX,IY+2)-R
   1HO(IX,IY)))-XX*((YY+1.)*(RHO(IX+1,IY+1)-RHO(IX+1,IY-1))-YY*(RHO(IX
   2+1,IY+2)-RHO(IX+1,IY)))-GMC*YNEW/RAD3
    GRR=(XNEW*GX+YNEW*GY)/RNEW
    IF(IPTEST.EQ.1) GO TO 200
    PIP=(PRES(I,J)+PRES(I+1,J))/2.
    IF(MASS(I+1,J).EQ.0.) PIP=0.
    GO TO 210
200 PIP1=(PRES(I,J)+PRES(I+1,2*J-1))/2.
    IF(MASS(I+1,2*J-1).EQ.0.) PIP1=0.
    PIP2=(PRES(I,J)+PRES(I+1,2*J))/2.
    IF(MASS(I+1,2*J).EQ.0.) PIP2=0.
    PIP=(PIP1+PIP2)/2.
210 IF(I.NE.1) GO TO 215
    JP2=J+2
    IF(JP2.GT.4) JP2=JP2-4
    PIM=(PRES(1,J)+PRES(1,JP2))/2.
    IF(MASS(1,JP2).EQ.0.) PIM=0.
    GO TO 220
215 JJ=J
    IF(IMTEST.EQ.1) JJ=(J+1)/2
    PIM=(PRES(I,J)+PRES(I-1,JJ))/2.
    IF(MASS(I-1,JJ).EQ.0.) PIM=0.
220 TDOT2=-((PIM-PIP)/MASS(I,J)+GRR)/RNEW
    IF(TDOT2.LT.0.) TDOT2=0.
    TDOT(I,J)=-SQRT(TDOT2)
    ASAV=-TDOT(I,J)/DTH(I)
    IF(ASAV.LT.AMAX) GO TO 240
    AMAX=ASAV
    ISAV=I
    GO TO 240
230 TDOT(I,J)=0.
240 CONTINUE
    NNTH=NTH(NO4M1)
    DO 245 J=1,NNTH
    TDOT(NO4M1,J)=0.
245 CONTINUE
    DT=ARATIO/AMAX
    PRINT 250,DT
```

```
250 FORMAT(1H0*TIME STEP DT = *E16.8)
    DDDG=NTH(ISAV)/ARATIO
    PRINT 253,ISAV,DDDG
253 FORMAT(1H0*AT R+.5 = *I3*  THERE ARE *F10.3*  TIME STEPS PER ROTAT
   1ION*)
    DT2=DT*DT
    DT3=DT2*DT
    DT4=DT2*DT2
    DMG=XMG*DT2
    DMG=DMG.AND.MASK1
    PRINT 255,DMG
255 FORMAT(1H0*TIME SCALED PARTICAL MASS = *E16.8)
    GMC=GMC*DT2
    GMCP1=3.*GMC/(2.*RCC)
    GMCP2=GMC/(2.*RCC**3)
    DO 260 I=1,NO4M2
    NNTH=NTH(I)
    DO 260 J=1,NNTH
    IF(MASS(I,J).EQ.0.) GO TO 260
    TDOT(I,J)=TDOT(I,J)*DT
    IE(I,J)=IE(I,J)*DT2
    MASS(I,J)=MASS(I,J)*DT2
260 CONTINUE
    ANGVG=TWOPI/(DDDG*DT)
    EMAX=.2*XMG*NBRG*RIG2*ANGVG*ANGVG
    EMIN=-10.*EMAX
    PTHMAX=0.
    PTHMIN=-.5*XMG*NBRG*RIG2*ANGVG
    PMAX=2.*(2.*RHO(NO4,NO4)*DT2+GMCP1)
    PMAXM=10.*DMG*NBRG/(PI*RIG2)
    PMAXT=100.*(IE(6,1)+IE(6,2)+IE(6,3)+IE(6,4))/4.
    PMAXP=.2*PMAXM*PMAXT
    SPHI=HT/PMAX
    ENCODE(10,275,PL(2)) SPHI
275 FORMAT(F10.3)
    SPHIM=HT/PMAXM
    ENCODE(10,275,PLM(2)) SPHIM
    SPHIT=HT/PMAXT
    ENCODE(10,275,PLT(2)) SPHIT
    SPHIP=HT/PMAXP
```

```
      ENCODE(10,280,PLP(2)) SPHIP
  280 FORMAT(F10.0)
      SUMMAS=0.
      SUMPTH=0.
      SUMKE=0.
      SUMIE=0.
      DO 300 I=1,NO4M2
      NNTH=NTH(I)
      DO 300 J=1,NNTH
      IF(MASS(I,J).EQ.0.) GO TO 300
      R2MID=R2MDAV(I,J)
      SUMMAS=SUMMAS+MASS(I,J)
      SUMPTH=SUMPTH+MASS(I,J)*R2MID*TDOT(I,J)
      SUMKE=SUMKE+.5*MASS(I,J)*R2MID*TDOT(I,J)*TDOT(I,J)
      SUMIE=SUMIE+MASS(I,J)*IE(I,J)
  300 CONTINUE
      GO TO 400
C STATEMENTS 350 TO 400 ENABLE RUN TO BE CONTINUED WITH PICK UP TAPE
  350 NS2=0
  355 READ(11) RADIUS,THETA
      WRITE(9) RADIUS,THETA
      NS2=NS2+NBSG
      IF(NS2.LT.NBRG) GO TO 355
      REWIND 9
      READ(11) RDOT,TDOT,IE,MASS,XNWAV,YNWAV,R2MDAV,R2NWAV,SUMMAS,
     1    SUMPTH,SUMIE,SUMKE,PESOLD,PEFOLD,CYY,DT,DT2,DT3,DT4,DMG,GMC,
     2    GMCP1,GMCP2,SPHIM,SPHIT,SPHIP,PLM(2),PLT(2),PLP(2),EMAX,EMIN,
     3    PTHMAX,PTHMIN,SPHI,PL(2)
      CY=CY+CYY
      CYY=CYY+1
  400 CYMIN=CYY-1
      CALL EVICT(6LTAPE11)
      DO 405 I=1,NO2
      DO 405 J=1,NO2
      RHO(I,J)=0.
  405 CONTINUE
      NS2=0
  410 READ(9) RADIUS,THETA
      DO 420 IS=1,NBSG
```

```
      IF(RADIUS(IS).GT.RMAX) GO TO 420
      IX=RADIUS(IS)*COS(THETA(IS))+CENTER+.5
      JY=RADIUS(IS)*SIN(THETA(IS))+CENTER+.5
      RHO(IX,JY)=RHO(IX,JY)+.5*DMG
  420 CONTINUE
      NS2=NS2+NBSG
      IF(NS2.LT.NBRG) GO TO 410
      REWIND 9
      WRITE(8) RDOT,TDOT,IE,MASS,XNWAV,YNWAV,R2MDAV,R2NWAV
      REWIND 8
      RETURN
      END
```

```
      OVERLAY(AGFILE,3,0)
      PROGRAM ADVGAS
      COMMON/ALLCOM/I2A,ITEST,N,NO2,CY,CYY,RHO(32,32)
      COMMON/GASCOM/NTH(15),DTH(15),AREA(15),NPLOTG,NPRNTG,NRING,NO4,
     1    NO4M1,NO4M2,RCC,GMC,GMCP1,GMCP2,DMG,DT,DT2,DT3,DT4,NRTCEL,
     2    RMAX,CENTER,SUMPTH,SUMMAS,SUMIE,SUMKE,PESOLD,PEFOLD,XMG,
     3    SAVPTH(200),SAVPE(200),SAVIE(200),SAVKE(200),SAVTE(200),
     4    MASK1,NBSG,NBRG,JTG,JSG,ING(2),XMING,XMAXG,YMING,YMAXG,XPG(2),
     5    YPG,ITAPXG,GM1,NMKG,XYL(3),XPR(5),YPR(5),XMIN4,XMAX4,YMIN4,
     6    YMAX4,PLM(2),RCOS,RSIN,SPHIM,PLT(2),SPHIT,PLP(2),SPHIP,TWOPI,
     7    PTHMIN,PTHMAX,EMIN,EMAX,CYMIN,PI,PL(2),SPHI,MTEST,ITAPE
      COMMON/NEGCOM/IE(15,64),MASS(15,64),XNWAV(15,64)
      DIMENSION RADIUS(1000),THETA(1000),RDOT(15,64),
     1    TDOT(15,64),YNWAV(15,64),R2MDAV(15,64),
     2    R2NWAV(15,64),PRES(15,64),RDDOT(15,64),PRAD(15,64),
     3    PTHETA(15,64),ETOTAL(15,64),XPLOT(1000),YPLOT(1000),GR(15),
     4    RNEWR(15),DENSR(15),PRESR(15),MASSR(15),IER(15),KAPPA(15),
     5    XPLOT1(32),YPLOT1(32),XPLOT2(32),YPLOT2(32)
      EQUIVALENCE (RADIUS(1),XPLOT(1)),(THETA(1),YPLOT(1))
      INTEGER CY,CYY,Q
      REAL IE,INE,IER,KAPPA,KAPPA2,KE,MASS,MAS,MASSR,NUMBER
C IF GAS PLOTTING IS TO BE DONE THIS CYCLE SET IPLOTG=1, OTHERWISE SET IPLOTG=0.
      IPLOTG=0
      IF(CYY-CYY/NPLOTG*NPLOTG.EQ.0.OR.CYY.EQ.1) IPLOTG=1
      ENCODE(10,15,XPG(2)) CYY
      ENCODE(10,15,XYL(3)) CYY
   15 FORMAT(I10)
      CALL PSEUDO
C IF GAS LONG PRINTING IS TO BE DONE THIS CYCLE SET IPRNTG=1, OTHERWISE
C SET IPRNTG=0
      IPRNTG=0
      IF(CYY-CYY/NPRNTG*NPRNTG.EQ.0.OR.CYY.EQ.1) IPRNTG=1
C IF GAS RING AVERAGES ARE TO BE PRINTED THIS CYCLE SET IRING=1 ,OTHERWISE SET
C IRING=0.
      IRING=0
      IF(CYY-CYY/NRING*NRING.EQ.0.OR.CYY.EQ.1) IRING=1
      READ(8) RDOT,TDOT,IE,MASS,XNWAV,YNWAV,R2MDAV,R2NWAV
      REWIND 8
```

```
      IF(IPLOTG.NE.1) GO TO 40
C MAKE A CONTOUR PLOT OF THE GRAVITATIONAL POTENTIAL
      CALL DDIPLT(0,ING,5,XPR,YPR,XMIN4,XMAX4,YMIN4,YMAX4,3,XYL(1),
     1    2,PL(1),14,ITAPXG)
      DX=0.
      DY=1.
      DO 30 J=2,NO2
      K=0
      DX=DX+RCOS
      DY=DY+RSIN
      DU=0.
      DV=1.
      DO 25 I=2,NO2
      K=K+1
      DU=DU+RCOS
      DV=DV+RSIN
      XC=I-CENTER
      YC=J-CENTER
      R2=XC*XC+YC*YC
      R=SQRT(R2)
      EGC=GMCP1-GMCP2*R2
      IF(R.GT.RCC) EGC=GMC/R
      XPLOT1(K)=J+DU
      YPLOT1(K)=SPHI*(2.*RHO(J,I)+EGC)+DV
      XPLOT2(K)=I+DX
   25 YPLOT2(K)=SPHI*(2.*RHO(I,J)+EGC)+DY
      CALL DDIPLT(0,ING,K,XPLOT2,YPLOT2,XMIN4,XMAX4,YMIN4,YMAX4,
     1    3,XYL(1),2,PL(1),14,ITAPXG)
      CALL DDIPLT(0,ING,K,XPLOT1,YPLOT1,XMIN4,XMAX4,YMIN4,YMAX4,
     1    3,XYL(1),2,PL(1),14,ITAPXG)
   30 CONTINUE
      K=0
      DO 35 I=2,NO2
      XI=I
      K=K+1
      XPLOT2(K)=XI+RCOS
```

```
      YPLOT2(K)=1.+RSIN
   35 CONTINUE
      CALL DDIPLT(1,ING,K,XPLOT2,YPLOT2,XMIN4,XMAX4,YMIN4,YMAX4,
     1     3,XYL(1),2,PL(1),13,ITAPXG)
   40 IF(IRING.NE.1) GO TO 120
      DO 50 I=1,NO4M2
      GRR=0.
      NNTH=NTH(I)
      DO 45 J=1,NNTH
      R=I-.5
      TH=(J-.5)*DTH(I)
      XCELL=R*COS(TH)
      YCELL=R*SIN(TH)
      X=CENTER+XCELL
      Y=CENTER+YCELL
      IX=X
      IY=Y
      XX=IX-X
      YY=IY-Y
      RAD3=R*R*R
      IF(R.LT.RCC) RAD3=RCC*RCC*RCC
      GX=(YY+1.)*((XX+1.)*(RHO(IX+1,IY)-RHO(IX-1,IY))-XX*(RHO(IX+2,IY)-R
     1HO(IX,IY)))-YY*((XX+1.)*(RHO(IX+1,IY+1)-RHO(IX-1,IY+1))-XX*(RHO(IX
     2+2,IY+1)-RHO(IX,IY+1)))-GMC*XCELL/RAD3
      GY=(XX+1.)*((YY+1.)*(RHO(IX,IY+1)-RHO(IX,IY-1))-YY*(RHO(IX,IY+2)-R
     1HO(IX,IY)))-XX*((YY+1.)*(RHO(IX+1,IY+1)-RHO(IX+1,IY-1))-YY*(RHO(IX
     2+1,IY+2)-RHO(IX+1,IY)))-GMC*YCELL/RAD3
      GRR=GRR+(XCELL*GX+YCELL*GY)/R
   45 CONTINUE
      GR(I)=GRR/(NNTH*DT2)
   50 CONTINUE
      PRINT 52
   52 FORMAT(1H0*ANGULARLY AVERAGED QUANTITIES FOR THE END OF THE LAST C
     1YCLE WHICH HAVE BEEN COMPUTED AT CELL GEOMETRIC CENTERS*)
      PRINT 53
   53 FORMAT(1H0*    I    R                GR            KAPPA(-GR)*)
      DO 60 I=1,NO4M2
      R=I-.5
      IF(I.EQ.1)GO TO 54
      IF(I.EQ.NO4M2) GO TO 56
      KAPPA2=(GR(I+1)-GR(I-1))/2.+3.*GR(I)/R
```

```
      GO TO 58
   54 KAPPA2=(GR(2)-0.)/2.+3.*GR(1)/R
      GO TO 58
   56 KAPPA2=GR(NO4M2)-GR(NO4-3)+3.*GR(NO4M2)/R
   58 KAPPA2=-KAPPA2
      IF(KAPPA2.LT.0.) KAPPA2=0.
      KAPPA(I)=SQRT(KAPPA2)
      PRINT 61,I,R,GR(I),KAPPA(I)
   60 CONTINUE
   61 FORMAT(1H ,I3,F5.1,2E16.5)
      PRINT 64
   64 FORMAT(1H0*ANGULARLY AVERAGED QUANTITIES FOR THE END OF THE LAST C
     1YCLE WHICH HAVE BEEN COMPUTED AT CELL CENTERS OF MASS AND FOR AVER
     2AGED R.R*)
      PRINT 68
   68 FORMAT(1H * I  NO. PART.     RNEW             MASS             DENS
     1     SPECIFIC IE            PRES            RDOT*)
      DO 80 I=1,NO4M2
      RNEWR(I)=0.
      MASSR(I)=0.
      PRESR(I)=0.
      GRR=0.
      RDOTR=0.
      NNTH=NTH(I)
      DO 73 J=1,NNTH
      IF(MASS(I,J).EQ.0.) GO TO 73
      XNEW=XNWAV(I,J)
      YNEW=YNWAV(I,J)
      RNEW=SQRT(XNEW*XNEW+YNEW*YNEW)
      RNEWR(I)=RNEWR(I)+MASS(I,J)*RNEW
      MASSR(I)=MASSR(I)+MASS(I,J)
      PRESR(I)=PRESR(I)+MASS(I,J)*IE(I,J)
      X=CENTER+XNEW
      Y=CENTER+YNEW
      IX=X
      IY=Y
      XX=IX-X
      YY=IY-Y
```

```
      RAD3=RNEW*RNEW*RNEW
      IF(RNEW.LT.RCC) RAD3=RCC*RCC*RCC
      GX=(YY+1.)*((XX+1.)*(RHO(IX+1,IY)-RHO(IX-1,IY))-XX*(RHO(IX+2,IY)-R
     1HO(IX,IY)))-YY*((XX+1.)*(RHO(IX+1,IY+1)-RHO(IX-1,IY+1))-XX*(RHO(IX
     2+2,IY+1)-RHO(IX,IY+1)))-GMC*XNEW/RAD3
      GY=(XX+1.)*((YY+1.)*(RHO(IX,IY+1)-RHO(IX,IY-1))-YY*(RHO(IX,IY+2)-R
     1HO(IX,IY)))-XX*((YY+1.)*(RHO(IX+1,IY+1)-RHO(IX+1,IY-1))-YY*(RHO(IX
     2+1,IY+2)-RHO(IX+1,IY)))-GMC*YNEW/RAD3
      GRR=GRR+MASS(I,J)*(XNEW*GX+YNEW*GY)/RNEW
      RDOTR=RDOTR+MASS(I,J)*RDOT(I,J)
   73 CONTINUE
      IF(MASSR(I).EQ.0.) GO TO 75
      NPARTR=MASSR(I)/DMG
      RNEWR(I)=RNEWR(I)/MASSR(I)
      DENSR(I)=MASSR(I)/(NNTH*AREA(I)*DT2)
      IER(I)=PRESR(I)/(MASSR(I)*DT2)
      PRESR(I)=GM1*PRESR(I)/(NNTH*AREA(I)*DT4)
      GR(I)=GRR/(MASSR(I)*DT2)
      RDOTR=RDOTR/(MASSR(I)*DT)
      MASSR(I)=MASSR(I)/DT2
      GO TO 78
   75 NPARTR=0
      RNEWR(I)=0.
      DENSR(I)=0.
      IER(I)=0.
      PRESR(I)=0.
      GR(I)=0.
      RDOTR=0.
   78 PRINT 81,I,NPARTR,RNEWR(I),MASSR(I),DENSR(I),IER(I),PRESR(I),
     1    RDOTR
   80 CONTINUE
   81 FORMAT(1H ,I3,I6,6E16.5)
      PRINT 83
   83 FORMAT(1H0* I          R2MID              R2NEW             TDOT      RNEW
     1.THDT02.THDT02     DP/M          GR(CNTR MASS)  Q(GR(CELL CNTR))*)
      DO 100 I=1,NO4M2
      R2MDR=0.
      R2NWR=0.
      PTHR=0.
```

F-20

```
      NNTH=NTH(I)
      DO 85 J=1,NNTH
      IF(MASS(I,J).EQ.0.) GO TO 85
      R2MID=R2MDAV(I,J)
      R2NEW=R2NWAV(I,J)
      R2MDR=R2MDR+MASS(I,J)*R2MID
      R2NWR=R2NWR+MASS(I,J)*R2NEW
      PTHR=PTHR+MASS(I,J)*R2MID*TDOT(I,J)
   85 CONTINUE
      IF(MASSR(I).EQ.0.) GO TO 90
      R2MDR=R2MDR/(MASSR(I)*DT2)
      R2NWR=R2NWR/(MASSR(I)*DT2)
      TDOTR=PTHR/(MASSR(I)*R2MDR*DT3)
      THDT02=PTHR/(MASSR(I)*R2NWR*DT3)
      RTDOT2=RNEWR(I)*THDT02*THDT02
      IF(I.EQ.1) GO TO 86
      IF(I.EQ.NO4M2) GO TO 87
      PIP=(PRESR(I)+PRESR(I+1))/2.
      IF(MASSR(I+1).EQ.0.) PIP=0.
      PIM=(PRESR(I)+PRESR(I-1))/2.
      IF(MASSR(I-1).EQ.0.) PIM=0.
      GO TO 88
   86 PIP=(PRESR(I)+PRESR(I+1))/2.
      IF(MASSR(I+1).EQ.0.) PIP=0.
      PIM=PRESR(1)
      GO TO 88
   87 PIP=0.
      PIM=(PRES(I)+PRES(I-1))/2.
      IF(MASSR(I-1).EQ.0.) PIM=0.
   88 DPOM=(PIM-PIP)/MASSR(I)
      IF(IER(I).LT.0.) IER(I)=0.
      QR=KAPPA(I)*SQRT(IER(I))/(3.36*DENSR(I))
      GO TO 95
   90 R2MDR=0.
      R2NWR=0.
      TDOTR=0.
      RTDOT2=0.
      DPOM=0.
      QR=0.
   95 PRINT 101,I,R2MDR,R2NWR,TDOTR,RTDOT2,DPOM,GR(I),QR
  100 CONTINUE
```

```
101 FORMAT(1H ,I3,7E16.5)
120 DO 122 I=1,N04M1
    NNTH=NTH(I)
    DO 122 J=1,NNTH
    IF(MASS(I,J).EQ.0.)  GO TO 121
    PRES(I,J)=GM1*IE(I,J)*MASS(I,J)/AREA(I)
    IF(IE(I,J).LT.0.) PRES(I,J)=0.
    GO TO 122
121 PRES(I,J)=0.
122 CONTINUE
    IETEST=0
    NHIVEL=0
    SUMPEF=0.
    SUMPES=0.
    DO 200 I=1,N04M2
    ARCIP=I*DTH(I)
    ARCIM=(I-1)*DTH(I)
    IPTEST=0
    IF(I.EQ.1) IPTEST=1
    IF(I.EQ.2) IPTEST=1
    IF(I.EQ.4) IPTEST=1
    IF(I.EQ.8) IPTEST=1
    IF(I.EQ.16) IPTEST=1
    IMTEST=0
    IF(I.EQ.2) IMTEST=1
    IF(I.EQ.3) IMTEST=1
    IF(I.EQ.5) IMTEST=1
    IF(I.EQ.9) IMTEST=1
    IF(I.EQ.17) IMTEST=1
    NNTH=NTH(I)
    DO 200 J=1,NNTH
    IF(MASS(I,J).EQ.0.)  GO TO 200
    XNEW=XNWAV(I,J)
    YNEW=YNWAV(I,J)
    RNEW=SQRT(XNEW*XNEW+YNEW*YNEW)
    X=CENTER+XNEW
    Y=CENTER+YNEW
    IX=X
```

```
      IY=Y
      XX=IX-X
      YY=IY-Y
      RAD3=RNEW*RNEW*RNEW
      IF(RNEW.LT.RCC) RAD3=RCC*RCC*RCC
      GX=(YY+1.)*((XX+1.)*(RHO(IX+1,IY)-RHO(IX-1,IY))-XX*(RHO(IX+2,IY)-R
     1HO(IX,IY)))-YY*((XX+1.)*(RHO(IX+1,IY+1)-RHO(IX-1,IY+1))-XX*(RHO(IX
     2+2,IY+1)-RHO(IX,IY+1)))-GMC*XNEW/RAD3
      GY=(XX+1.)*((YY+1.)*(RHO(IX,IY+1)-RHO(IX,IY-1))-YY*(RHO(IX,IY+2)-R
     1HO(IX,IY)))-XX*((YY+1.)*(RHO(IX+1,IY+1)-RHO(IX+1,IY-1))-YY*(RHO(IX
     2+1,IY+2)-RHO(IX+1,IY)))-GMC*YNEW/RAD3
      DX=-XX
      DY=-YY
      D11=1-DY-DX+DX*DY
      D12=DY*(1-DX)
      D21=DX*(1-DY)
      D22=DX*DY
      SUMPES=SUMPES+MASS(I,J)*(D11*RHO(IX,IY)+D12*RHO(IX,IY+1)
     1    +D21*RHO(IX+1,IY)+D22*RHO(IX+1,IY+1))
      EGC=GMCP1-GMCP2*RNEW*RNEW
      IF(RNEW.GT.RCC) EGC=GMC/RNEW
      SUMPEF=SUMPEF+MASS(I,J)*EGC
      IF(IPTEST.EQ.1) GO TO 130
      PIP=(PRES(I,J)+PRES(I+1,J))/2.
      IF(MASS(I+1,J).EQ.0.) PIP=0.
      DEIP=-ARCIP*PIP*(RDOT(I,J)+RDOT(I+1,J))/2.
      GO TO 140
  130 PIP1=(PRES(I,J)+PRES(I+1,2*J-1))/2.
      IF(MASS(I+1,2*J-1).EQ.0.) PIP1=0.
      PIP2=(PRES(I,J)+PRES(I+1,2*J))/2.
      IF(MASS(I+1,2*J).EQ.0.) PIP2=0.
      PIP=(PIP1+PIP2)/2.
      DEIP=-.5*ARCIP*PIP1*(RDOT(I,J)+RDOT(I+1,2*J-1))/2.
     1     -.5*ARCIP*PIP2*(RDOT(I,J)+RDOT(I+1,2*J))/2.
  140 IF(I.NE.1) GO TO 145
      JP2=J+2
      IF(JP2.GT.4) JP2=JP2-4
      PIM=(PRES(1,J)+PRES(1,JP2))/2.
      IF(MASS(1,JP2).EQ.0.) PIM=0.
      DEIM=0.
```

```
      GO TO 160
  145 JJ=J
      IF(IMTEST.EQ.1)  JJ=(J+1)/2
      PIM=(PRES(I,J)+PRES(I-1,JJ))/2.
      IF(MASS(I-1,JJ).EQ.0.)  PIM=0.
      DEIM=ARCIM*PIM*(RDOT(I,J)+RDOT(I-1,JJ))/2.
  160 JP1=J+1
      IF(JP1.GT.NNTH) JP1=1
      PJP=(PRES(I,J)+PRES(I,JP1))/2.
      IF(MASS(I,JP1).EQ.0.) PJP=0.
      DEJP=-PJP*(I-.5)*(TDOT(I,J)+TDOT(I,JP1))/2.
      JM1=J-1
      IF(JM1.LT.1) JM1=NNTH
      PJM=(PRES(I,J)+PRES(I,JM1))/2.
      IF(MASS(I,JM1).EQ.0.) PJM=0.
      DEJM=PJM*(I-.5)*(TDOT(I,J)+TDOT(I,JM1))/2.
      GR=(XNEW*GX+YNEW*GY)/RNEW
      FR=GR+(PIM-PIP)/MASS(I,J)
      GTH=(XNEW*GY-YNEW*GX)/RNEW
      FTH=GTH+(PJM-PJP)/MASS(I,J)
      R2MID=R2MDAV(I,J)
      R2NEW=R2NWAV(I,J)
      PTH=R2MID*TDOT(I,J)
      TORQ=RNEW*FTH
      PTHDT=PTH+TORQ
      THDT02=.5*(PTH+PTHDT)/R2NEW
      RDDOT(I,J)=FR+RNEW*THDT02*THDT02
      RDOTDT=RDOT(I,J)+RDDOT(I,J)
      RDOT34=RDOT(I,J)+.75*RDDOT(I,J)
      R2DT=R2NEW+RNEW*RDOT34+.25*RDOT34*RDOT34
      IF(R2DT.EQ.0.) R2DT=1.
      TDOTDT=PTHDT/R2DT
      IE(I,J)=IE(I,J)+.5*(RDOT(I,J)*RDOT(I,J)+R2MID*TDOT(I,J)*TDOT(I,J)
     1    -RDOTDT*RDOTDT-R2DT*TDOTDT*TDOTDT)+GR*(RDOT(I,J)+RDOTDT)/2.
     2    +GTH*RNEW*THDT02+(DEIP+DEIM+DEJP+DEJM)/MASS(I,J)
      IF(IE(I,J).LT.0.) IETEST=1
      IF(MASS(I,J).LT.5.0*DMG) GO TO 185
      ABRDOT=ABS(RDOTDT)
```

```
      ABTDOT=ABS(TDOTDT)
      IF(ABRDOT.GE..375.OR.ABTDOT.GE..375*DTH(I)) NHIVEL=NHIVEL+1
      IF(ABRDOT.LT.1.0.AND.ABTDOT.LT.DTH(I)) GO TO 185
      CY=CYY
      IPLOTG=1
      IPRNTG=1
      PRINT 165,RDOTDT,TDOTDT,I,J
  165 FORMAT(1H0*RDOT=*E16.8*  AND TDOT=*E16.8*  FOR I=*I3*  AND J=*I3)
  185 PTHETA(I,J)=TDOTDT
  200 CONTINUE
      IF(NHIVEL.GT.0) PRINT 205,NHIVEL
  205 FORMAT(1H0*NUMBER OF CELLS WITH .GT. 5 PARTICLES AND WITH SCALED V
     1ELOCITY COMPONENTS .GT. .375.(CELL DIMENSION) =   *I5)
      IF(IETEST.EQ.1.AND.CYY-CYY/5*5.EQ.0) CALL NEGIE
      PTH=SUMPTH/DT3
      TM=SUMMAS/DT2
      INE=SUMIE/DT4
      KE=SUMKE/DT4
      PESNEW=-SUMPES/DT4
      PEFNEW=-SUMPEF/DT4
      IF(CYY.EQ.1) PESOLD=PESNEW
      IF(CYY.EQ.1) PEFOLD=PEFNEW
      PES=(PESOLD+PESNEW)/2.
      PEF=(PEFOLD+PEFNEW)/2.
      PESOLD=PESNEW
      PEFOLD=PEFNEW
      PE=PES+PEF
      TE=PE+INE+KE
      PRINT 232
  232 FORMAT(1H0 *THE VALUES ON THE NEXT TWO LINES ARE FOR THE END OF TH
     1E LAST CYCLE*)
      PRINT 234,PES,PEF,PTH,TM
  234 FORMAT(1H *PES=*E14.7*  PEF=*E14.7*    PTH=*E14.7*    TM=*E14.7)
      PRINT 236,PE,INE,KE,TE
  236 FORMAT(1H *PE=*E14.7*    IE=*E14.7*     KE=*E14.7*    TE=*E14.7)
      ICYPLT=CYY-CYMIN
      SAVPTH(ICYPLT)=PTH
      SAVPE(ICYPLT)=PE
```

```
      SAVIE(ICYPLT)=INE
      SAVKE(ICYPLT)=KE
      SAVTE(ICYPLT)=TE
      DO 245 I=1,NO4M1
      NNTH=NTH(I)
      DO 245 J=1,NNTH
      IF(MASS(I,J).EQ.0.) GO TO 240
      TDOT(I,J)=PTHETA(I,J)
  240 PRAD(I,J)=0.
      PTHETA(I,J)=0.
      ETOTAL(I,J)=0.
      MASS(I,J)=0.
      XNWAV(I,J)=0.
      YNWAV(I,J)=0.
      R2MDAV(I,J)=0.
      R2NWAV(I,J)=0.
  245 CONTINUE
      DO 248 I=1,NO2
      DO 248 J=1,NO2
      RHO(I,J)=0.
  248 CONTINUE
      PRINT 250
  250 FORMAT(1H0*LAST PARTICLE OF 1ST 10 SETS OF NBSG GAS PARTICLES*)
      PRINT 260
  260 FORMAT(1H *NUMBER    X-CENTER            Y-CENTER          R
     1   THETA             RDOT                TDOT*)
      NOUT=0
      NPART=1
      NS2=0
  270 READ(JTG) RADIUS, THETA
      DO 350 IS=1,NBSG
      ROLD=RADIUS(IS)
      IF(ROLD.GT.RMAX) GO TO 340
      IROLD=ROLD+1
      TOLD=THETA(IS)
      JTOLD=TOLD/DTH(IROLD)+1
      RDOT34=RDOT(IROLD,JTOLD)+.75*RDDOT(IROLD,JTOLD)
      RMID=ROLD+.5*RDOT34
```

```
      RDOTT=RDOT(IROLD,JTOLD)+RDDOT(IROLD,JTOLD)
      RNEW=ROLD+RDOTT
      IF(RNEW.GT.RMAX) GO TO 340
      TDOTT=TDOT(IROLD,JTOLD)
      TNEW=TOLD+TDOTT
      IF(RNEW) 273,274,275
  273 RNEW=-RNEW
      TNEW=TNEW+PI
      GO TO 275
  274 RNEW=1.0E-8
  275 IRNEW=RNEW+1
      IF(TNEW.LT.0.) TNEW=TNEW+TWOPI
      IF(TNEW.GE.TWOPI)TNEW=TNEW-TWOPI
      JTNEW=TNEW/DTH(IRNEW)+1
      DPR=DMG*RDOTT
      R2MID=RMID*RMID
      DPTH=DMG*R2MID*TDOTT
      DE=DMG*(IE(IROLD,JTOLD)+.5*(RDOTT*RDOTT+R2MID*TDOTT*TDOTT))
      PRAD(IRNEW,JTNEW)=PRAD(IRNEW,JTNEW)+DPR
      PTHETA(IRNEW,JTNEW)=PTHETA(IRNEW,JTNEW)+DPTH
      ETOTAL(IRNEW,JTNEW)=ETOTAL(IRNEW,JTNEW)+DE
      MASS(IRNEW,JTNEW)=MASS(IRNEW,JTNEW)+DMG
      R2MDAV(IRNEW,JTNEW)=R2MDAV(IRNEW,JTNEW)+R2MID
      R2NWAV(IRNEW,JTNEW)=R2NWAV(IRNEW,JTNEW)+RNEW*RNEW
      XNEW=RNEW*COS(TNEW)
      YNEW=RNEW*SIN(TNEW)
      IXNEW=CENTER+XNEW+.5
      JYNEW=CENTER+YNEW+.5
      RHO(IXNEW,JYNEW)=RHO(IXNEW,JYNEW)+.5*DMG
      XNWAV(IRNEW,JTNEW)=XNWAV(IRNEW,JTNEW)+XNEW
      YNWAV(IRNEW,JTNEW)=YNWAV(IRNEW,JTNEW)+YNEW
      GO TO 345
  340 NOUT=NOUT+1
      RNEW=999.
      TNEW=999.
      XNEW=999.
      YNEW=999.
      TDOTT=999.
      RDOTT=999.
```

```
345 RADIUS(IS)=RNEW
    THETA(IS)=TNEW
350 CONTINUE
    WRITE(JSG) RADIUS,THETA
    IF(NPART.LE.10)PRINT 360,NPART,XNEW,YNEW,RNEW,TNEW,RDOTT,TDOTT
360 FORMAT(1H ,I6,6E16.8)
    NPART=NPART+1
    IF(IPLOTG.EQ.0) GO TO 370
    DO 365 IS=1,NBSG
    R=RADIUS(IS)
    TH=THETA(IS)
    XPLOT(IS)=R*COS(TH)+CENTER
    YPLOT(IS)=R*SIN(TH)+CENTER
365 CONTINUE
    Q=0
    IF(NS2.EQ.NMKG) Q=1
    CALL DDIPLT(Q,ING,NBSG,XPLOT,YPLOT,XMING,XMAXG,YMING,YMAXG,
   1     2,XPG,1,YPG,13,ITAPXG)
370 NS2=NS2+NBSG
    IF(NS2.LT.NBRG) GO TO 270
    REWIND JTG
    REWIND JSG
    JSAVE=JSG
    JSG=JTG
    JTG=JSAVE
    PRINT 373,NOUT
373 FORMAT(1H0*NUMBER OF PARTICLES OUTSIDE OF MESH=*I6)
    SUMMAS=0.
    SUMPTH=0.
    SUMKE=0.
    SUMIE=0.
    DO 380 I=1,NO4M1
    NNTH=NTH(I)
    DO 380 J=1,NNTH
    IF(MASS(I,J).EQ.0.) GO TO 376
    NUMBER=MASS(I,J)/DMG
    XNWAV(I,J)=XNWAV(I,J)/NUMBER
    YNWAV(I,J)=YNWAV(I,J)/NUMBER
    R2MDAV(I,J)=R2MDAV(I,J)/NUMBER
```

```
      R2NWAV(I,J)=R2NWAV(I,J)/NUMBER
      SUMMAS=SUMMAS+MASS(I,J)
      RDOT(I,J)=PRAD(I,J)/MASS(I,J)
      SUMPTH=SUMPTH+PTHETA(I,J)
      IF(R2MDAV(I,J).EQ.0.) R2MDAV(I,J)=1.0E-8
      TDOT(I,J)=PTHETA(I,J)/(MASS(I,J)*R2MDAV(I,J))
      KE=.5*MASS(I,J)*(RDOT(I,J)*RDOT(I,J)
     1     +R2MDAV(I,J)*TDOT(I,J)*TDOT(I,J))
      SUMKE=SUMKE+KE
      IE(I,J)=ETOTAL(I,J)-KE
      SUMIE=SUMIE+IE(I,J)
      IE(I,J)=IE(I,J)/MASS(I,J)
      GO TO 380
  376 RDOT(I,J)=0.
      TDOT(I,J)=0.
      IE(I,J)=0.
  380 CONTINUE
      WRITE(8) RDOT,TDOT,IE,MASS,XNWAV,YNWAV,R2MDAV,R2NWAV
      REWIND 8
      IF(IPRNTG. NE.1) GO TO 485
      PRINT 390
  390 FORMAT(1H0*J,  (RDOT(I,J),I=1,15)*)
  400 FORMAT(1H ,I4,15F8.4)
      DO 410 J=1,64
      PRINT 400,J,(RDOT(I,J),I=1,15)
  410 CONTINUE
      PRINT 420
  420 FORMAT(1H0*J,  (TDOT(I,J),I=1,15)*)
      DO 430 J=1,64
      PRINT 400,J,(TDOT(I,J),I=1,15)
  430 CONTINUE
      PRINT 440
  440 FORMAT(1H0*J,  (IE(I,J),I=1,15)*)
  450 FORMAT(1H ,I4,15E8.1)
      DO 460 J=1,64
      PRINT 450,J,(IE(I,J),I=1,15)
  460 CONTINUE
      PRINT 470
  470 FORMAT(1H0*J,  (MASS(I,J),I=1,15)*)
```

```
      DO 480 J=1,64
      PRINT 450,J,(MASS(I,J),I=1,15)
  480 CONTINUE
  485 IF(IPLOTG.NE.1) GO TO 600
C PLOT VELOCITY VECTORS
      XNO2=NO2
      A=0.
      CALL DDIPLT(0,ING,1,A,A,0,XNO2,0,XNO2,2,XPG,1,YPG,13,ITAPXG)
      XYSCAL=10./NO2
      VSCALE=.001/DT
      DO 500 I=1,NO4M2
      NNTH=NTH(I)
      DO 500 J=1,NNTH
      IF(MASS(I,J).EQ.0.) GO TO 500
      XNEW=XNWAV(I,J)
      YNEW=YNWAV(I,J)
      RNEW=SQRT(XNEW*XNEW+YNEW*YNEW)
      VX=-YNEW*TDOT(I,J)+XNEW*RDOT(I,J)/RNEW
      VY=XNEW*TDOT(I,J)+YNEW*RDOT(I,J)/RNEW
      XMID=XNEW-.5*VX
      YMID=YNEW-.5*VY
      X=XMID+CENTER
      Y=YMID+CENTER
      XA=XYSCAL*X
      YA=XYSCAL*Y
      XB=XYSCAL*(X+VSCALE*VX)
      YB=XYSCAL*(Y+VSCALE*VY)
      CALL PARROW(XA,YA,XB,YB,1)
  500 CONTINUE
      CALL DDIPLT(1,ING,1,A,A,0,XNO2,0,XNO2,2,XPG,1,YPG,13,ITAPXG)
C MAKE A CONTOUR PLOT OF THE DENSITY
      CALL DDIPLT(0,ING,5,XPR,YPR,XMIN4,XMAX4,YMIN4,YMAX4,3,XYL(1),
     1    2,PLM(1),14,ITAPXG)
      DX=0.
      DY=1.
      DO 515 J=2,NO2
      K=0
      DX=DX+RCOS
      DY=DY+RSIN
      DU=0.
      DV=1.
```

```
      DO 510 I=2,NO2
      K=K+1
      DU=DU+RCOS
      DV=DV+RSIN
      XC=I-CENTER
      YC=J-CENTER
      IR=SQRT(XC*XC+YC*YC)+1
      IF(IR.GT.NO4M2) GO TO 506
      TH=ATAN2(XC,YC)
      IF(TH.LT.0.) TH=TH+TWOPI
      JT1=TH/DTH(IR)+1
      Z1=MASS(IR,JT1)/AREA(IR)
      TH=ATAN2(YC,XC)
      IF(TH.LT.0.) TH=TH+TWOPI
      JT2=TH/DTH(IR)+1
      Z2=MASS(IR,JT2)/AREA(IR)
      GO TO 508
  506 Z1=0.
      Z2=0.
  508 XPLOT1(K)=J+DU
      YPLOT1(K)=SPHIM*Z1+DV
      XPLOT2(K)=I+DX
  510 YPLOT2(K)=SPHIM*Z2+DY
      CALL DDIPLT(0,ING,K,XPLOT2,YPLOT2,XMIN4,XMAX4,YMIN4,YMAX4,
     1    3,XYL(1),2,PLM(1),14,ITAPXG)
      CALL DDIPLT(0,ING,K,XPLOT1,YPLOT1,XMIN4,XMAX4,YMIN4,YMAX4,
     1    3,XYL(1),2,PLM(1),14,ITAPXG)
  515 CONTINUE
      K=0
      DO 520 I=2,NO2
      XI=I
      K=K+1
      XPLOT2(K)=XI+RCOS
      YPLOT2(K)=1.+RSIN
  520 CONTINUE
      CALL DDIPLT(1,ING,K,XPLOT2,YPLOT2,XMIN4,XMAX4,YMIN4,YMAX4,
     1    3,XYL(1),2,PLM(1),13,ITAPXG)
C MAKE A CONTOUR PLOT OF THE TEMPERATURE
      CALL DDIPLT(0,ING,5,XPR,YPR,XMIN4,XMAX4,YMIN4,YMAX4,3,XYL(1),
     1    2,PLT(1),14,ITAPXG)
      DX=0.
```

```
      DY=1.
      DO 540 J=2,NO2
      K=0
      DX=DX+RCOS
      DY=DY+RSIN
      DU=0.
      DV=1.
      DO 530 I=2,NO2
      K=K+1
      DU=DU+RCOS
      DV=DV+RSIN
      XC=I-CENTER
      YC=J-CENTER
      IR=SQRT(XC*XC+YC*YC)+1
      IF(IR.GT.NO4M2) GO TO 524
      TH=ATAN2(XC,YC)
      IF(TH.LT.0.) TH=TH+TWOPI
      JT1=TH/DTH(IR)+1
      Z1=IE(IR,JT1)
      TH=ATAN2(YC,XC)
      IF(TH.LT.0.) TH=TH+TWOPI
      JT2=TH/DTH(IR)+1
      Z2=IE(IR,JT2)
      GO TO 526
  524 Z1=0.
      Z2=0.
  526 XPLOT1(K)=J+DU
      YPLOT1(K)=SPHIT*Z1+DV
      XPLOT2(K)=I+DX
  530 YPLOT2(K)=SPHIT*Z2+DY
      CALL DDIPLT(0,ING,K,XPLOT2,YPLOT2,XMIN4,XMAX4,YMIN4,YMAX4,
     1     3,XYL(1),2,PLT(1),14,ITAPXG)
      CALL DDIPLT(0,ING,K,XPLOT1,YPLOT1,XMIN4,XMAX4,YMIN4,YMAX4,
     1     3,XYL(1),2,PLT(1),14,ITAPXG)
  540 CONTINUE
      K=0
      DO 545 I=2,NO2
      XI=I
      K=K+1
      XPLOT2(K)=XI+RCOS
```

```
      YPLOT2(K)=1.+RSIN
  545 CONTINUE
      CALL DDIPLT(1,ING,K,XPLOT2,YPLOT2,XMIN4,XMAX4,YMIN4,YMAX4,
     1     3,XYL(1),2,PLT(1),13,ITAPXG)
C MAKE A CONTOUR PLOT OF THE PRESSURE
      CALL DDIPLT(0,ING,5,XPR,YPR,XMIN4,XMAX4,YMIN4,YMAX4,3,XYL(1),
     1     2,PLP(1),14,ITAPXG)
      DX=0.
      DY=1.
      DO 560 J=2,NO2
      K=0
      DX=DX+RCOS
      DY=DY+RSIN
      DU=0.
      DV=1.
      DO 550 I=2,NO2
      K=K+1
      DU=DU+RCOS
      DV=DV+RSIN
      XC=I-CENTER
      YC=J-CENTER
      IR=SQRT(XC*XC+YC*YC)+1
      IF(IR.GT.NO4M2) GO TO 547
      TH=ATAN2(XC,YC)
      IF(TH.LT.0.) TH=TH+TWOPI
      JT1=TH/DTH(IR)+1
      Z1=IE(IR,JT1)*MASS(IR,JT1)/AREA(IR)
      TH=ATAN2(YC,XC)
      IF(TH.LT.0.) TH=TH+TWOPI
      JT2=TH/DTH(IR)+1
      Z2=IE(IR,JT2)*MASS(IR,JT2)/AREA(IR)
      GO TO 549
  547 Z1=0.
      Z2=0.
  549 XPLOT1(K)=J+DU
      YPLOT1(K)=SPHIP*Z1+DV
```

```
      XPLOT2(K)=I+DX
  550 YPLOT2(K)=SPHIP*Z2+DY
      CALL DDIPLT(0,ING,K,XPLOT2,YPLOT2,XMIN4,XMAX4,YMIN4,YMAX4,
     1     3,XYL(1),2,PLP(1),14,ITAPXG)
      CALL DDIPLT(0,ING,K,XPLOT1,YPLOT1,XMIN4,XMAX4,YMIN4,YMAX4,
     1     3,XYL(1),2,PLP(1),14,ITAPXG)
  560 CONTINUE
      K=0
      DO 570 I=2,NO2
      XI=I
      K=K+1
      XPLOT2(K)=XI+RCOS
      YPLOT2(K)=1.+RSIN
  570 CONTINUE
      CALL DDIPLT(1,ING,K,XPLOT2,YPLOT2,XMIN4,XMAX4,YMIN4,YMAX4,
     1     3,XYL(1),2,PLP(1),13,ITAPXG)
  600 IF(CYY.LT.CY) RETURN
C IF(CYY.GE.CY) WRITE INFORMATION TO BE SAVED ON TAPE1
      NS2=0
  610 READ(JTG) RADIUS,THETA
      WRITE(1) RADIUS,THETA
      NS2=NS2+NBSG
      IF(NS2.LT.NBRG) GO TO 610
      WRITE(1) RDOT,TDOT,IE,MASS,XNWAV,YNWAV,R2MDAV,R2NWAV,SUMMAS,
     1     SUMPTH,SUMIE,SUMKE,PESOLD,PEFOLD,CYY,DT,DT2,DT3,DT4,DMG,GMC,
     2     GMCP1,GMCP2,SPHIM,SPHIT,SPHIP,PLM(2),PLT(2),PLP(2),EMAX,EMIN,
     3     PTHMAX,PTHMIN,SPHI,PL(2)
      REWIND 1
      RETURN
      END
```

```
      SUBROUTINE NEGIE
      COMMON/ALLCOM/I2A,ITEST,N,NO2,CY,CYY,RHO(32,32)
      COMMON/GASCOM/NTH(15),DTH(15),AREA(15),NPLOTG,NPRNTG,NRING,NO4,
     1    NO4M1,NO4M2,RCC,GMC,GMCP1,GMCP2,DMG,DT,DT2,DT3,DT4,NRTCEL,
     2    RMAX,CENTER,SUMPTH,SUMMAS,SUMIE,SUMKE,PESOLD,PEFOLD,XMG,
     3    SAVPTH(200),SAVPE(200),SAVIE(200),SAVKE(200),SAVTE(200),
     4    MASK1,NBSG,NBRG,JTG,JSG,ING(2),XMING,XMAXG,YMING,YMAXG,XPG(2),
     5    YPG,ITAPXG,GM1,NMKG,XYL(3),XPR(5),YPR(5),XMIN4,XMAX4,YMIN4,
     6    YMAX4,PLM(2),RCOS,RSIN,SPHIM,PLT(2),SPHIT,PLP(2),SPHIP,TWOPI,
     7    PTHMIN,PTHMAX,EMIN,EMAX,CYMIN,PI,PL(2),SPHI,MTEST,ITAPE
      COMMON/NEGCOM/IE(15,64),MASS(15,64),IECOR(15,64)
      REAL IE,IECOR,IE1,IE2,IE3,IE4,IE5,IE6,IE7,IE8,IE9,MASS
      DO 5 I=1,NO4M1
      NNTH=NTH(I)
      DO 5 J=1,NNTH
      IECOR(I,J)=0.
      IE(I,J)=MASS(I,J)*IE(I,J)
    5 CONTINUE
      IE(1,8)=0.
      IE(1,9)=0.
      IECOR(1,8)=0.
      IECOR(1,9)=0.
      DO 80 I=1,NO4M2
      NNTH=NTH(I)
      IRTEST=1
      IF(I.EQ.1) IRTEST=2
      IF(I.EQ.2) IRTEST=3
      IF(I.EQ.4) IRTEST=4
      IF(I.EQ.8) IRTEST=4
      IF(I.EQ.16) IRTEST=4
      IF(I.EQ.3) IRTEST=5
      IF(I.EQ.5) IRTEST=5
      IF(I.EQ.9) IRTEST=5
      IF(I.EQ.17) IRTEST=5
      DO 80 J=1,NNTH
```

```
      IF(IE(I,J).GE.0.) GO TO 80
      GO TO (10,20,30,40,50) IRTEST
   10 I1=I+1
      J1=J
      I2=I+1
      J2=J+1
      I3=I
      J3=J+1
      I4=I-1
      J4=J+1
      I5=I-1
      J5=J
      I6=I-1
      J6=J-1
      I7=I
      J7=J-1
      I8=I+1
      J8=J-1
      I9=1
      J9=9
      IF(J.NE.1) GO TO 15
      J6=NNTH
      J7=NNTH
      J8=NNTH
      GO TO 60
   15 IF(J.NE.NNTH) GO TO 60
      J2=1
      J3=1
      J4=1
      GO TO 60
   20 I1=1
      J1=J+1
      I2=1
      J2=J+2
      I3=1
      J3=J+3
      I4=2
```

```
      J4=2*J-2
      I5=2
      J5=2*J-1
      I6=2
      J6=2*J
      I7=2
      J7=2*J+1
      I8=1
      J8=8
      I9=1
      J9=9
      IF(J1.GT.4) J1=J1-4
      IF(J2.GT.4) J2=J2-4
      IF(J3.GT.4) J3=J3-4
      IF(J7.GT.8) J7=J7-8
      IF(J4.LT.1) J4=J4+8
      GO TO 60
   30 I1=3
      J1=2*J
      I2=3
      J2=2*J+1
      I3=2
      J3=J+1
      I4=1
      J4=J/2+1
      I5=1
      J5=J/2
      I6=2
      J6=J-1
      I7=3
      J7=2*J-2
      I8=3
      J8=2*J-1
      I9=1
      J9=9
      IF(J.NE.1) GO TO 35
      J5=4
```

```
      J6=8
      J7=16
      GO TO 60
   35 IF(J.NE.8) GO TO 60
      J2=1
      J3=1
      J4=1
      GO TO 60
   40 I1=I+1
      J1=2*J
      I2=I+1
      J2=2*J+1
      I3=I
      J3=J+1
      I4=I-1
      J4=J+1
      I5=I-1
      J5=J
      I6=I-1
      J6=J-1
      I7=I
      J7=J-1
      I8=I+1
      J8=2*J-2
      I9=I+1
      J9=2*J-1
      IF(J.NE.1) GO TO 45
      J6=NTH(I6)
      J7=NTH(I7)
      J8=NTH(I8)
      GO TO 60
   45 IF(J.NE.NNTH) GO TO 60
      J2=1
      J3=1
      J4=1
      GO TO 60
```

```
 50 I1=I+1
    J1=J
    I2=I+1
    J2=J+1
    I3=I
    J3=J+1
    I4=I-1
    J4=J/2+1
    I5=I-1
    J5=J/2
    I6=I
    J6=J-1
    I7=I+1
    J7=J-1
    I8=1
    J8=8
    I9=1
    J9=9
    IF(J.NE.1) GO TO 55
    J5=NTH(I5)
    J6=NTH(I6)
    J7=NTH(I7)
    GO TO 60
 55 IF(J.NE.NNTH) GO TO 60
    J2=1
    J3=1
    J4=1
 60 IE1=IE(I1,J1)
    IF(IE1.LT.0.) IE1=0.
    IE2=IE(I2,J2)
    IF(IE2.LT.0.) IE2=0.
    IE3=IE(I3,J3)
    IF(IE3.LT.0.) IE3=0.
    IE4=IE(I4,J4)
    IF(IE4.LT.0.) IE4=0.
    IE5=IE(I5,J5)
```

```fortran
      IF(IE5.LT.0.) IE5=0.
      IE6=IE(I6,J6)
      IF(IE6.LT.0.) IE6=0.
      IE7=IE(I7,J7)
      IF(IE7.LT.0.) IE7=0.
      IE8=IE(I8,J8)
      IF(IE8.LT.0.) IE8=0.
      IE9=IE(I9,J9)
      IF(IE9.LT.0.) IE9=0.
      SUM=IE1+IE2+IE3+IE4+IE5+IE6+IE7+IE8+IE9
      IF(SUM.LT.-IE(I,J)) GO TO 75
      QUOTNT=IE(I,J)/SUM
      IECOR(I1,J1)=IECOR(I1,J1)+QUOTNT*IE1
      IECOR(I2,J2)=IECOR(I2,J2)+QUOTNT*IE2
      IECOR(I3,J3)=IECOR(I3,J3)+QUOTNT*IE3
      IECOR(I4,J4)=IECOR(I4,J4)+QUOTNT*IE4
      IECOR(I5,J5)=IECOR(I5,J5)+QUOTNT*IE5
      IECOR(I6,J6)=IECOR(I6,J6)+QUOTNT*IE6
      IECOR(I7,J7)=IECOR(I7,J7)+QUOTNT*IE7

      IECOR(I8,J8)=IECOR(I8,J8)+QUOTNT*IE8
      IECOR(I9,J9)=IECOR(I9,J9)+QUOTNT*IE9
   75 IECOR(I,J)=IECOR(I,J)-IE(I,J)
   80 CONTINUE
      DO 100 I=1,NO4M2
      NNTH=NTH(I)
      DO 100 J=1,NNTH
      IF(MASS(I,J).EQ.0.) GO TO 100
      IE(I,J)=(IE(I,J)+IECOR(I,J))/MASS(I,J)
      IF(IE(I,J).LT.0.) IE(I,J)=0.
      IECOR(I,J)=0.
  100 CONTINUE
      RETURN
      END
```

```
      OVERLAY(IFILE,4,0)
      PROGRAM GASPLT
      COMMON/GASCOM/NTH(15),DTH(15),AREA(15),NPLOTG,NPRNTG,NRING,NO4,
     1    NO4M1,NO4M2,RCC,GMC,GMCP1,GMCP2,DMG,DT,DT2,DT3,DT4,NRTCEL,
     2    RMAX,CENTER,SUMPTH,SUMMAS,SUMIE,SUMKE,PESOLD,PEFOLD,XMG,
     3    SAVPTH(200),SAVPE(200),SAVIE(200),SAVKE(200),SAVTE(200),
     4    MASK1,NBSG,NBRG,JTG,JSG,ING(2),XMING,XMAXG,YMING,YMAXG,XPG(2),
     5    YPG,ITAPXG,GM1,NMKG,XYL(3),XPR(5),YPR(5),XMIN4,XMAX4,YMIN4,
     6    YMAX4,PLM(2),RCOS,RSIN,SPHIM,PLT(2),SPHIT,PLP(2),SPHIP,TWOPI,
     7    PTHMIN,PTHMAX,EMIN,EMAX,CYMIN,PI,PL(2),SPHI,MTEST,ITAPE
      COMMON/ALLCOM/I2A,ITEST,N,NO2,CY,CYY,RHO(32,32)
      DIMENSION XDATA(200),LABLPTH(2),LABLEN(8)
      DIMENSION ENDCYY(2),ENDPTH(2),ENDPE(2),ENDIE(2),ENDKE(2),ENDTE(2)
      INTEGER CY
      CALL PSEUDO
      CYMAX=CY
      NCYY=CY-CYMIN
      DO 20 I=1,NCYY
   20 XDATA(I)=CYMIN+I
      LABLCYY=10HCYCLES
      LABLPTH(1)=10HANGULAR MO
      LABLPTH(2)=10HMENTUM
      LABLEN(1)=10HENERGY***C
      LABLEN(2)=10HIRCLE-POTE
      LABLEN(3)=10HNTIAL,SQUA
      LABLEN(4)=10HRE-INTERNA
      LABLEN(5)=10HL,DIAMOND-
      LABLEN(6)=10HKINETIC,TR
      LABLEN(7)=10HIANGLE-TOT
      LABLEN(8)=10HAL
      ENDCYY(1)=CYMIN+1
      ENDCYY(2)=CY
      ENDPTH(1)=SAVPTH(1)
      ENDPTH(2)=SAVPTH(NCYY)
```

```
      ENDPE(1)=SAVPE(1)
      ENDPE(2)=SAVPE(NCYY)
      ENDIE(1)=SAVIE(1)
      ENDIE(2)=SAVIE(NCYY)
      ENDKE(1)=SAVKE(1)
      ENDKE(2)=SAVKE(NCYY)
      ENDTE(1)=SAVTE(1)
      ENDTE(2)=SAVTE(NCYY)
      IF(NCYY.LT.2) GO TO 30
      CALL DDIPLT(0,ING,2,ENDCYY,ENDPTH,CYMIN,CYMAX,PTHMIN,PTHMAX,
     1     1,LABLCYY,2,LABLPTH,1)
   30 CALL DDIPLT(1,ING,NCYY,XDATA,SAVPTH,CYMIN,CYMAX,PTHMIN,PTHMAX,
     1     1,LABLCYY,2,LABLPTH,0)
      IF(NCYY.LT.2) GO TO 40
      CALL DDIPLT(0,ING,2,ENDCYY,ENDPE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,1)
      CALL DDIPLT(0,ING,2,ENDCYY,ENDIE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,2)
      CALL DDIPLT(0,ING,2,ENDCYY,ENDKE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,3)
      CALL DDIPLT(0,ING,2,ENDCYY,ENDTE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,4)
   40 CALL DDIPLT(0,ING,NCYY,XDATA,SAVPE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,0)
      CALL DDIPLT(0,ING,NCYY,XDATA,SAVIE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,0)
      CALL DDIPLT(0,ING,NCYY,XDATA,SAVKE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,0)
      CALL DDIPLT(1,ING,NCYY,XDATA,SAVTE,CYMIN,CYMAX,EMIN,EMAX,
     1     1,LABLCYY,8,LABLEN,0)
      RETURN
      END
```

APPENDIX G

Comparison of Computer Plots Produced by the Polar Coordinate Rotating Gas Simulator of Appendix F and an Earlier Rectangular Coordinate Code.

The initial conditions for the two codes are identical. The galaxy's mass consists of a 5% dynamic gaseous component of initial maximum radius of 10 cells and a 95% constant stellar component which is a uniform density sphere of radius 6 cells. The stellar component is represented by the application of an analytically computed constant central force. The gaseous component has an initial density distribution which varies as $[1 - (r/r_o)^2]^{1/2}$, where r and r_o are the radius and maximum radius respectively. The gaseous component has an initial cellular specific internal energy distribution which corresponds to a constant times the minimum velocity dispersion required to satisfy the Toomre stability criterion for a particulate system. Initially, the cells have zero radial velocity and radially balancing angular velocities. Both runs were made on a 32 x 32 mesh (or polar equivalent) while later runs will be made on a 64 x 64 mesh.

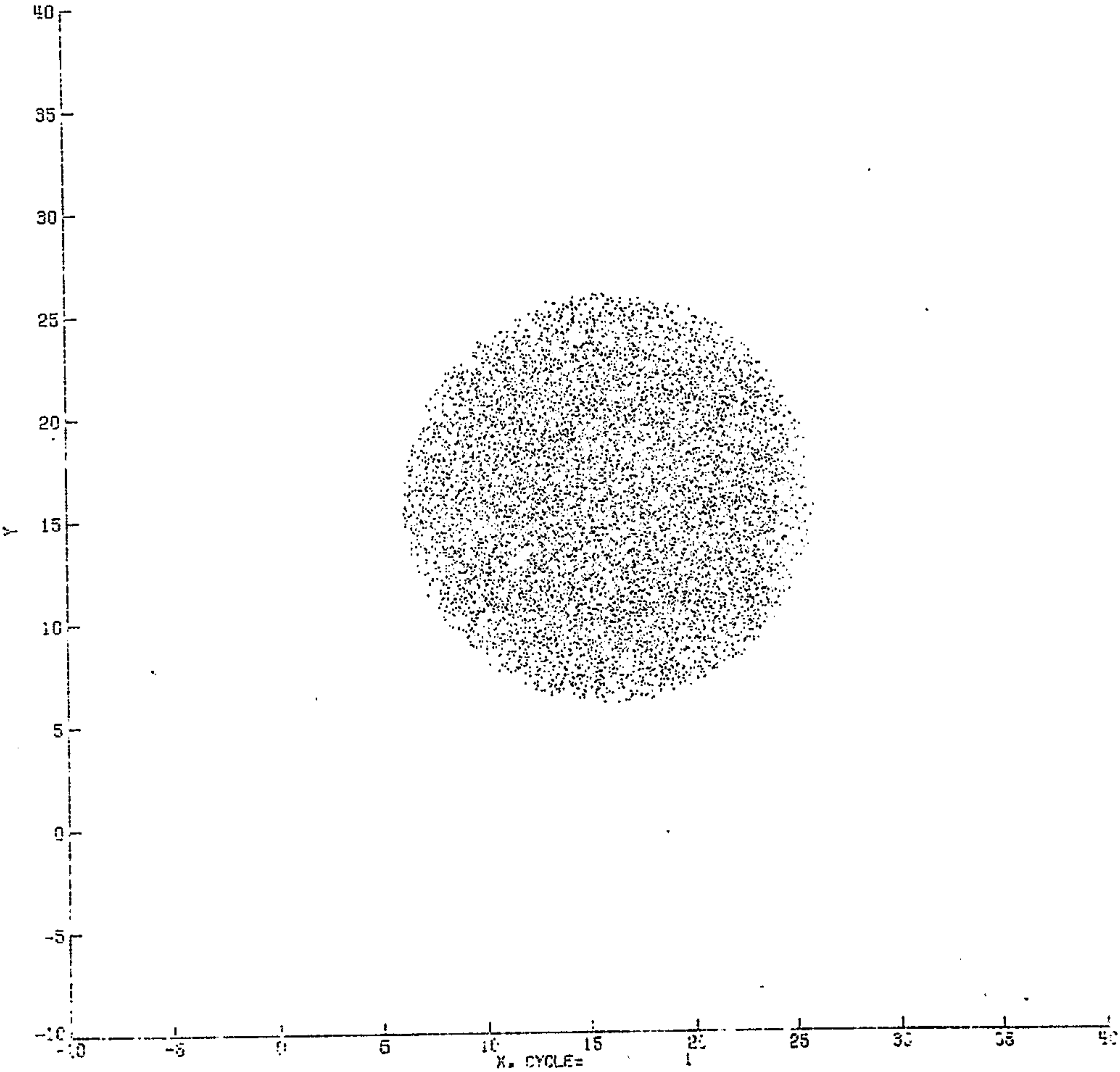
The superiority of the polar coordinate code is most readily apparent from the last two plots of each run, which show the variation of total angular momentum and energy with time. Although the rectangular coordinate code conserves total energy, it rapidly looses angular momentum. This loss of angular momentum is accompanied by a non-physical decrease in kinetic energy and a non-physical increase in internal energy of equal magnitude. The later causes a non-physical increase in pressure which along with some rectangular grid effects drives the particles rapidly outward. (Because of the rapid non-physical rise in internal energy, the scales of the temperature

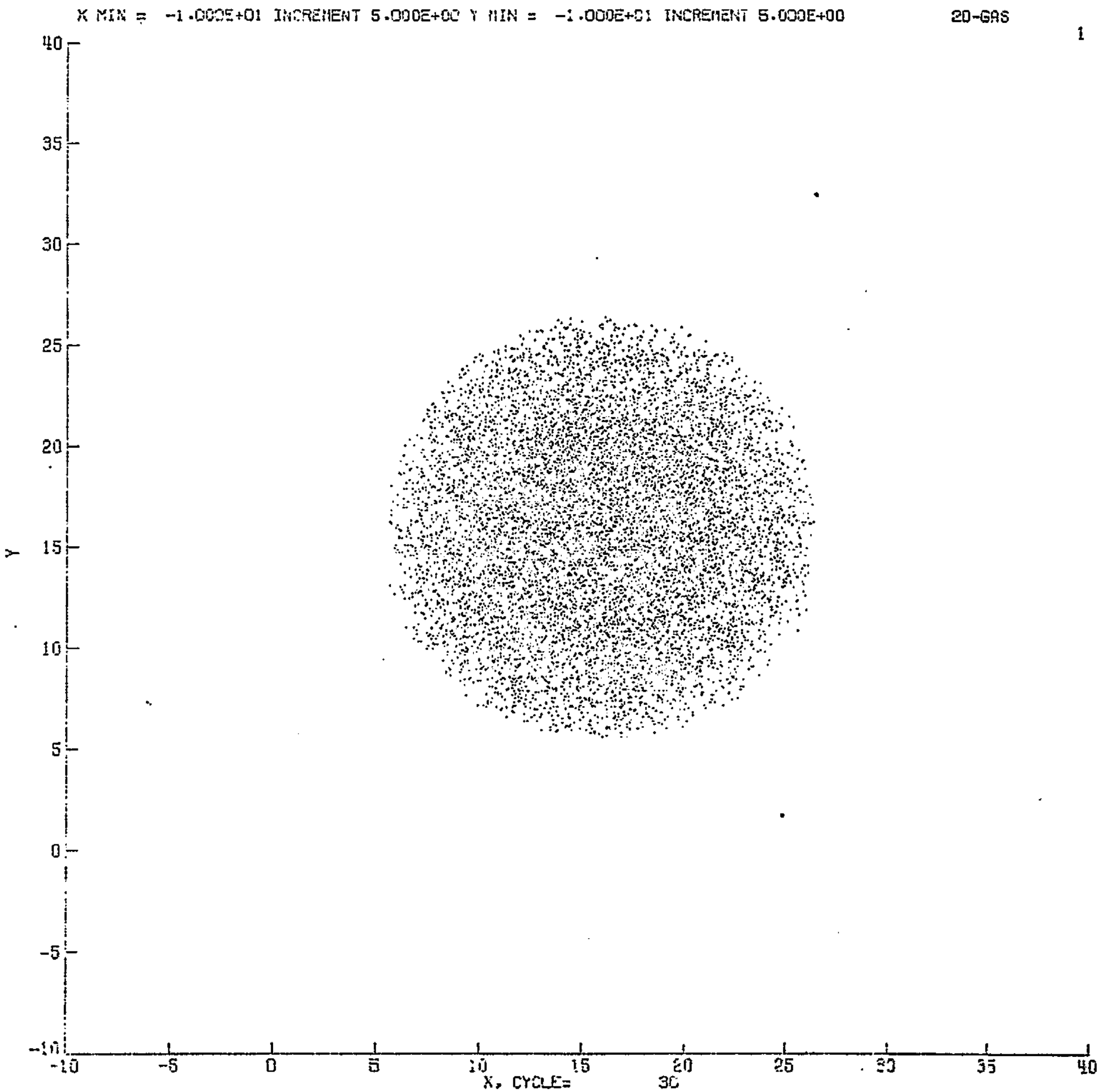
*Toomre, Alar: On the Gravitational Stability of a Disk of Stars. Astrophys. J., vol. 139, no. 4, May 15, 1964, pp. 1217-1238.

and pressure plots from the rectangular coordinate code are 11 times smaller than those scales for the polar coordinate code). The last two plots of the polar code show that it conserves angular momentum exactly and that for the first 60 cycles (about one half of one rotation) no heating is evident. (The printouts show a slight heating.) The slight outward drift of particles near the maximum radius in the polar code is due partly to the physical effects of the pressure gradient and partly to a slight instability in the radial acceleration, which is currently being investigated.

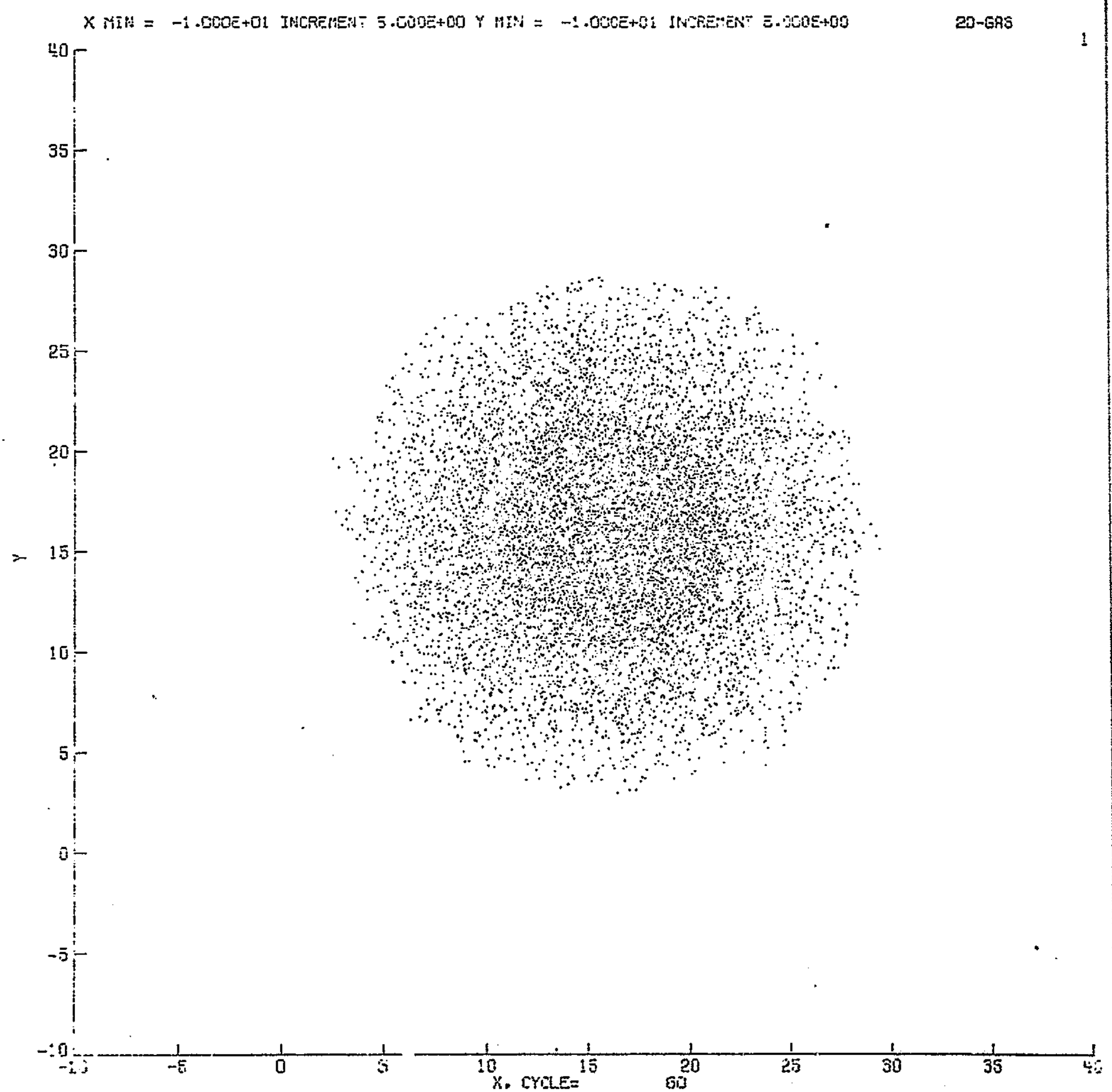
Cycle No.	Plot Type	Page No. Rect. Coord.	Page No. Polar Coord.
1	particle x-y position	G-3	G-20
30	particle x-y position	G-4	G-21
60	particle x-y position	G-5	G-22
1	cell velocity over x-y plane	G-6	G-23
30	cell velocity over x-y plane	G-7	G-24
60	cell velocity over x-y plane	G-8	G-25
1	cell density over x-y plane	G-9	G-26
30	cell density over x-y plane	G-10	G-27
60	cell density over x-y plane	G-11	G-28
1	cell temperature over x-y plane	G-12	G-29
30	cell temperature over x-y plane	G-13	G-30
60	cell temperature over x-y plane	G-14	G-31
1	cell pressure over x-y plane	G-15	G-32
30	cell pressure over x-y plane	G-16	G-33
60	cell pressure over x-y plane	G-17	G-34
1-60	total angular momentum vs. cycle (time)	G-18	G-35
1-60	total potential, internal, kinetic, and total energies vs. cycle (time)	G-19	G-36

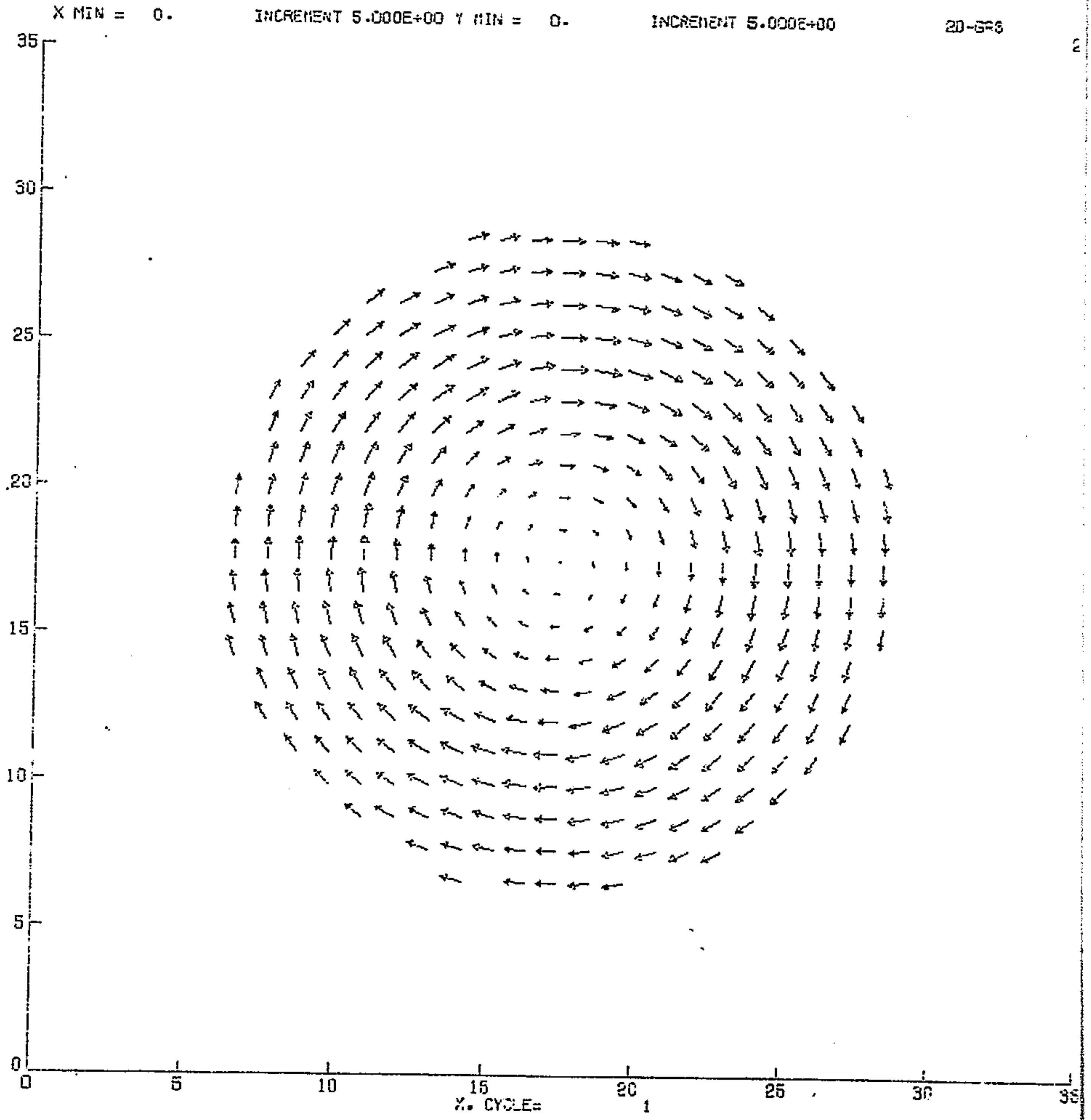
X MIN = -1.000E+01 INCREMENT 5.000E+00 Y MIN = -1.000E+01 INCREMENT 5.000E+00 2D-GAS 1

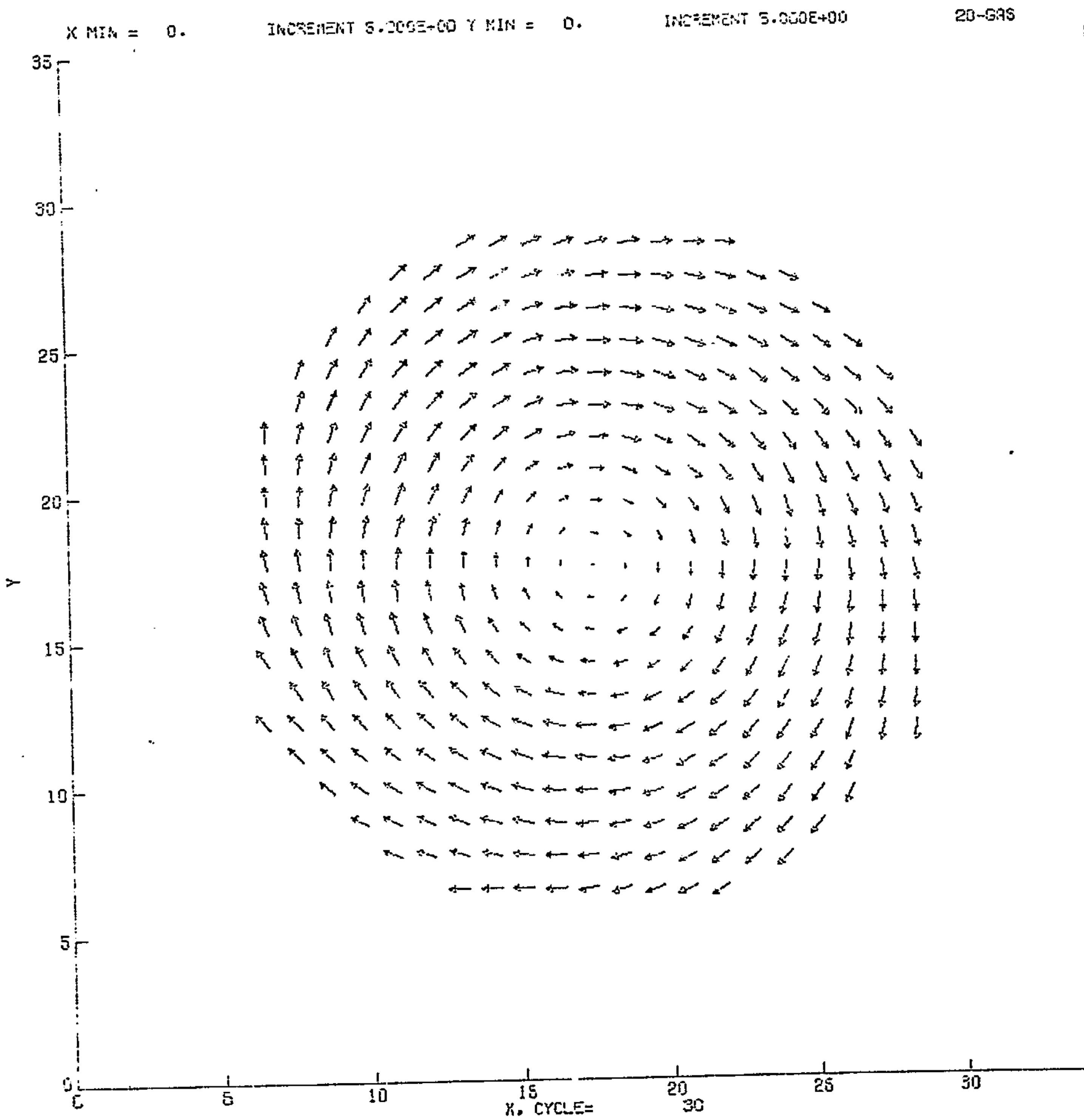




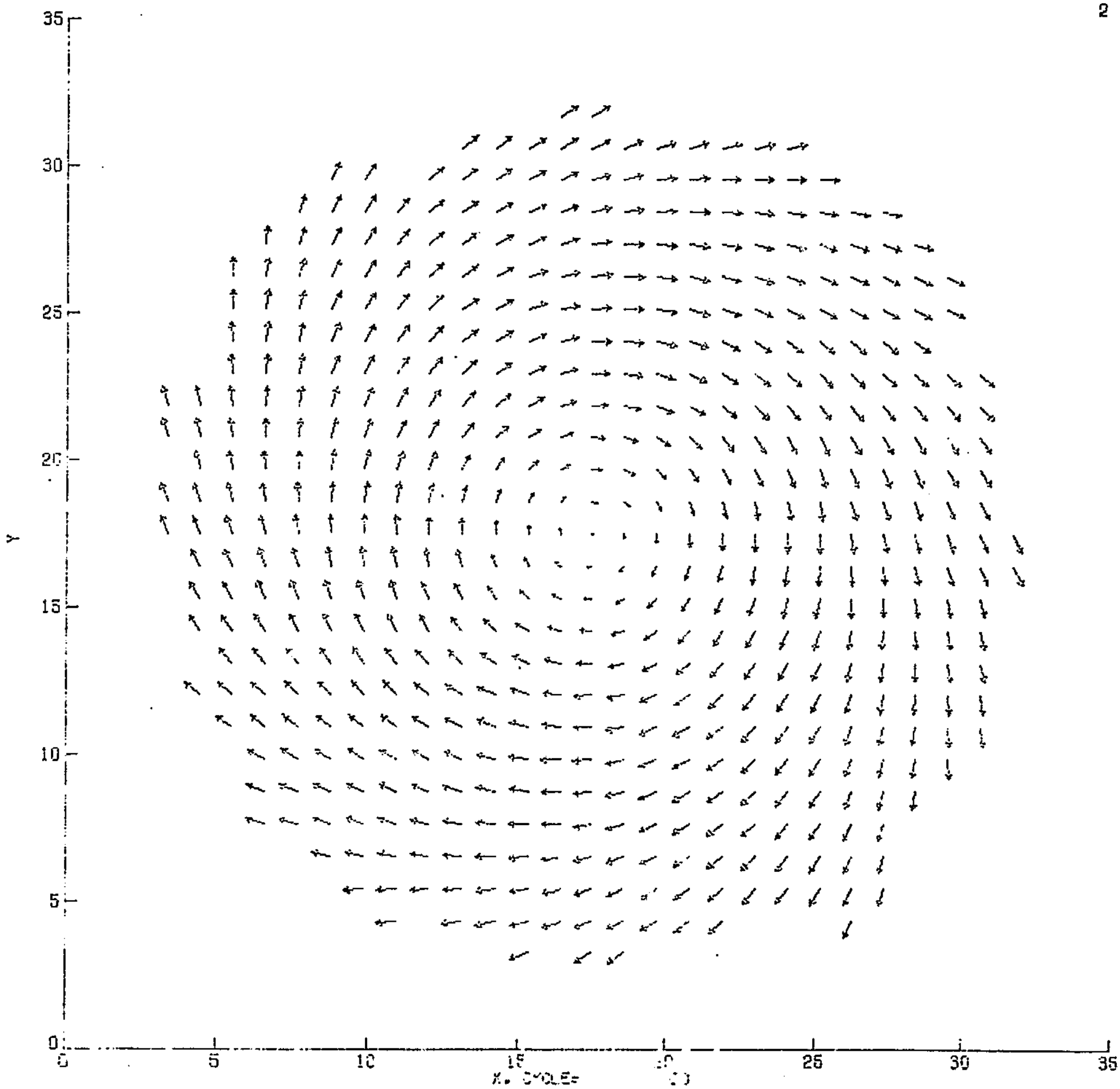
X MIN = -1.000E+01 INCREMENT 5.000E+00 Y MIN = -1.000E+01 INCREMENT 5.000E+00
2D-GAS
1
Y
X, CYCLE= 30


X MIN = -1.000E+01 INCREMENT 5.000E+00 Y MIN = -1.000E+01 INCREMENT 5.000E+00
2D-GAS
1
40
35
30
25
20
15
10
5
0
-5
-10
Y
-10
-5
0
5
10
15
20
25
30
35
40
X, CYCLE= 60


X MIN = 0.
INCREMENT 5.000E+00 Y MIN = 0.
INCREMENT 5.000E+00
2D-GAS
Y
X. CYCLE= 1
0
5
10
15
20
25
30
35

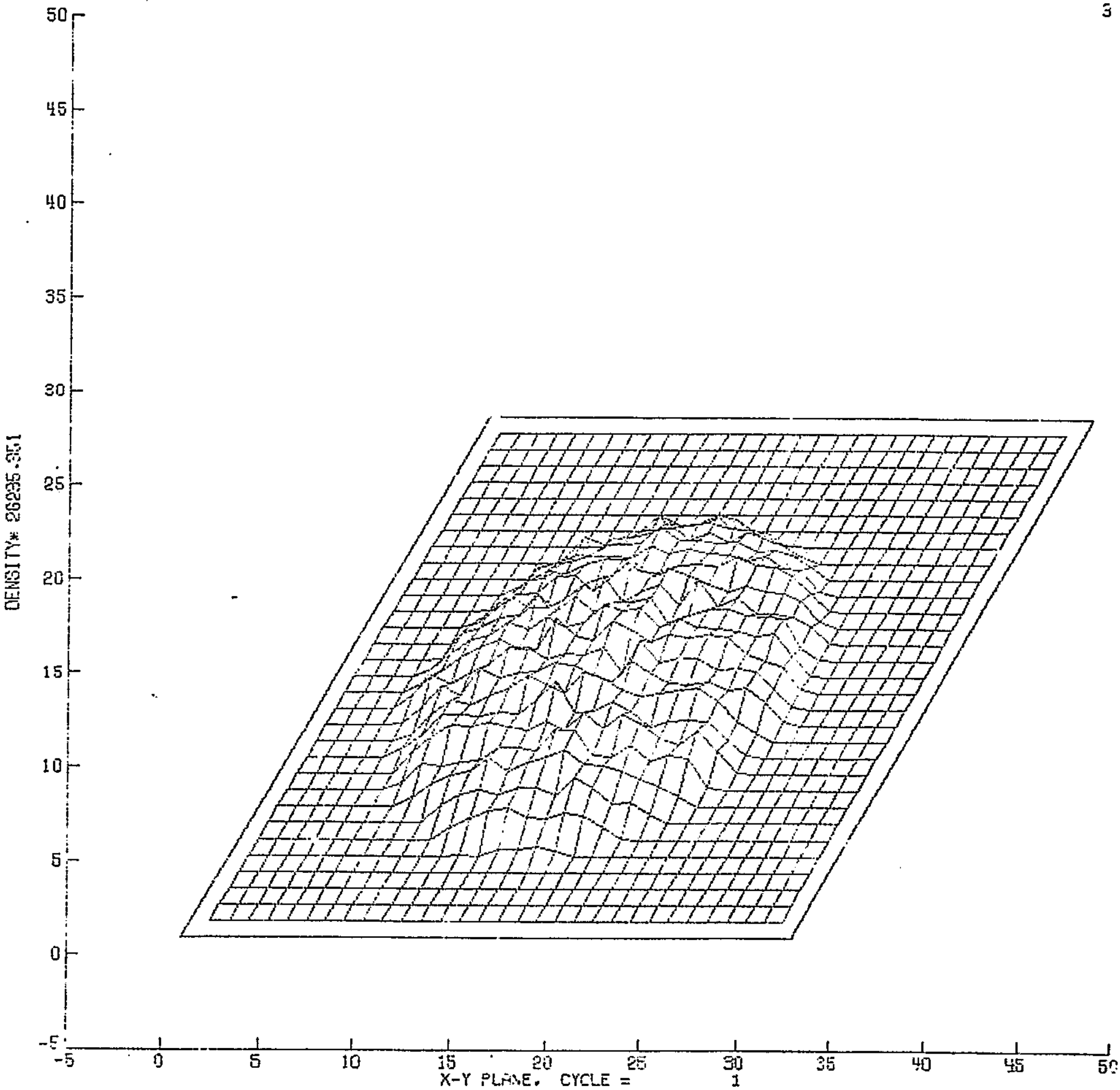

X MIN = 0.
INCREMENT 5.000E+00 Y MIN = 0.
INCREMENT 5.000E+00
2D-GAS
35
30
25
20
15
10
5
0
Y
0
5
10
15
20
25
30
X, CYCLE= 30

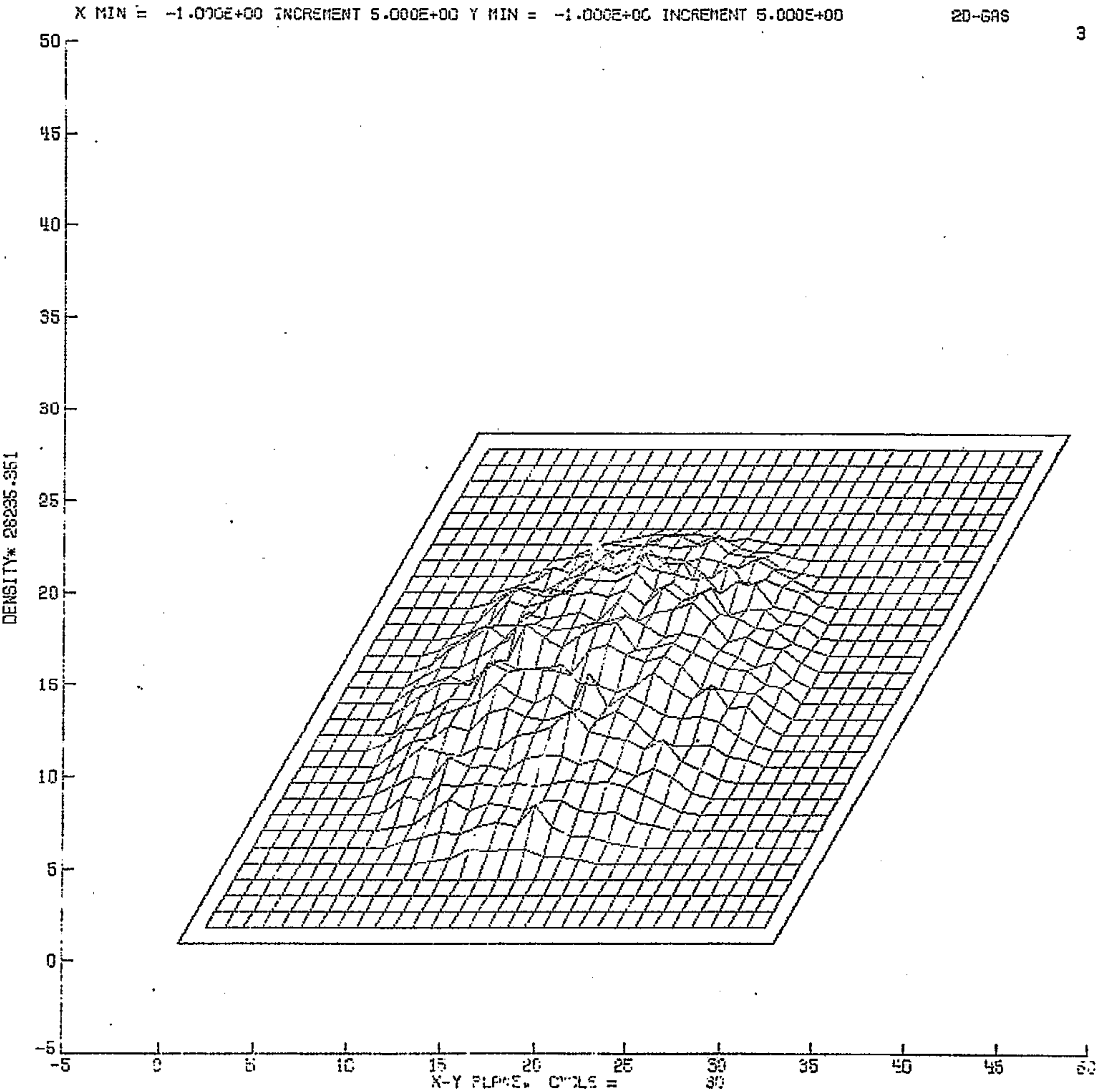
X MIN = 0. INCREMENT 5.000E+00 Y MIN = 0. INCREMENT 5.000E+00 20-GAS



X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00 2D-GAS

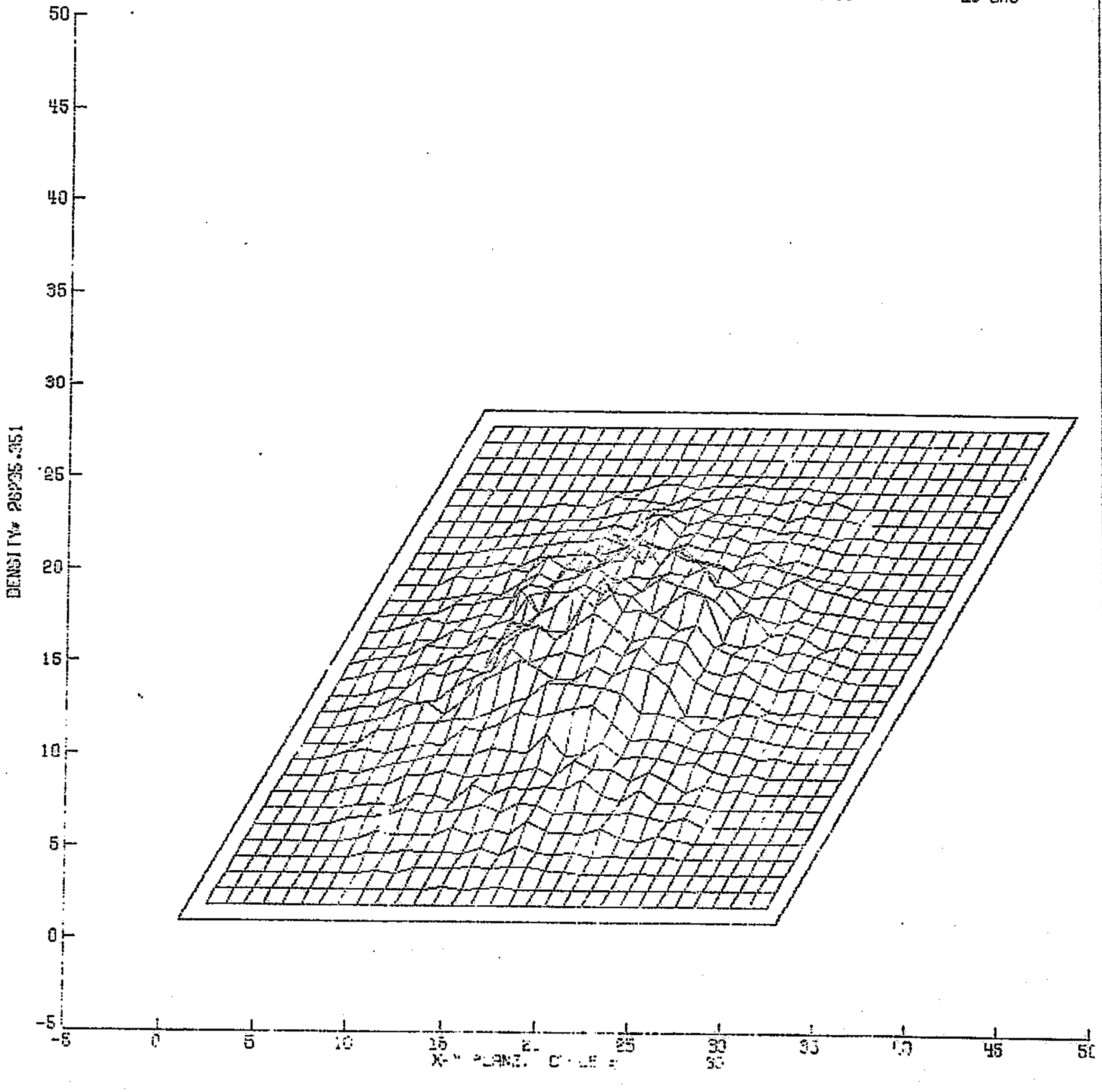
3

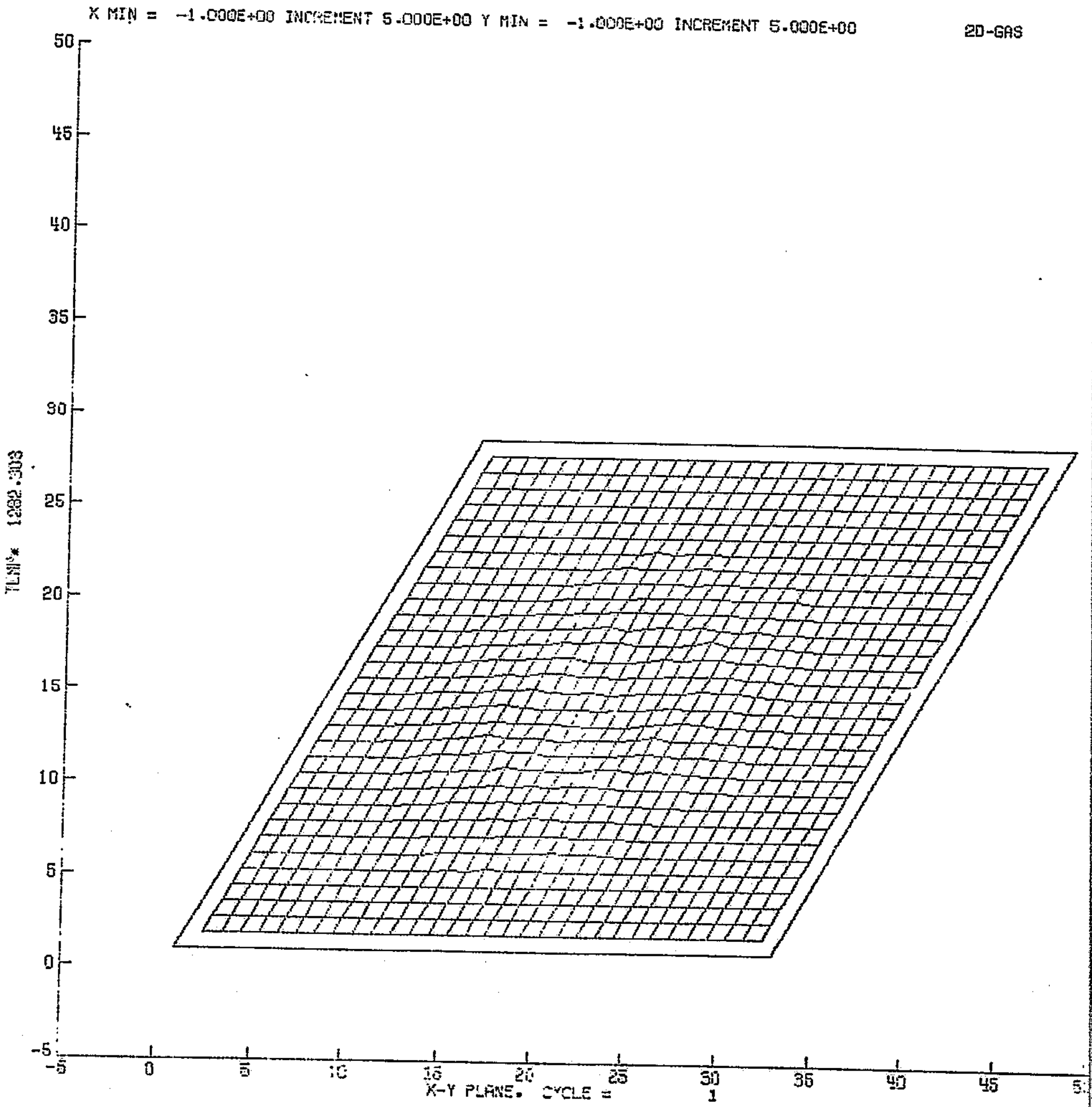


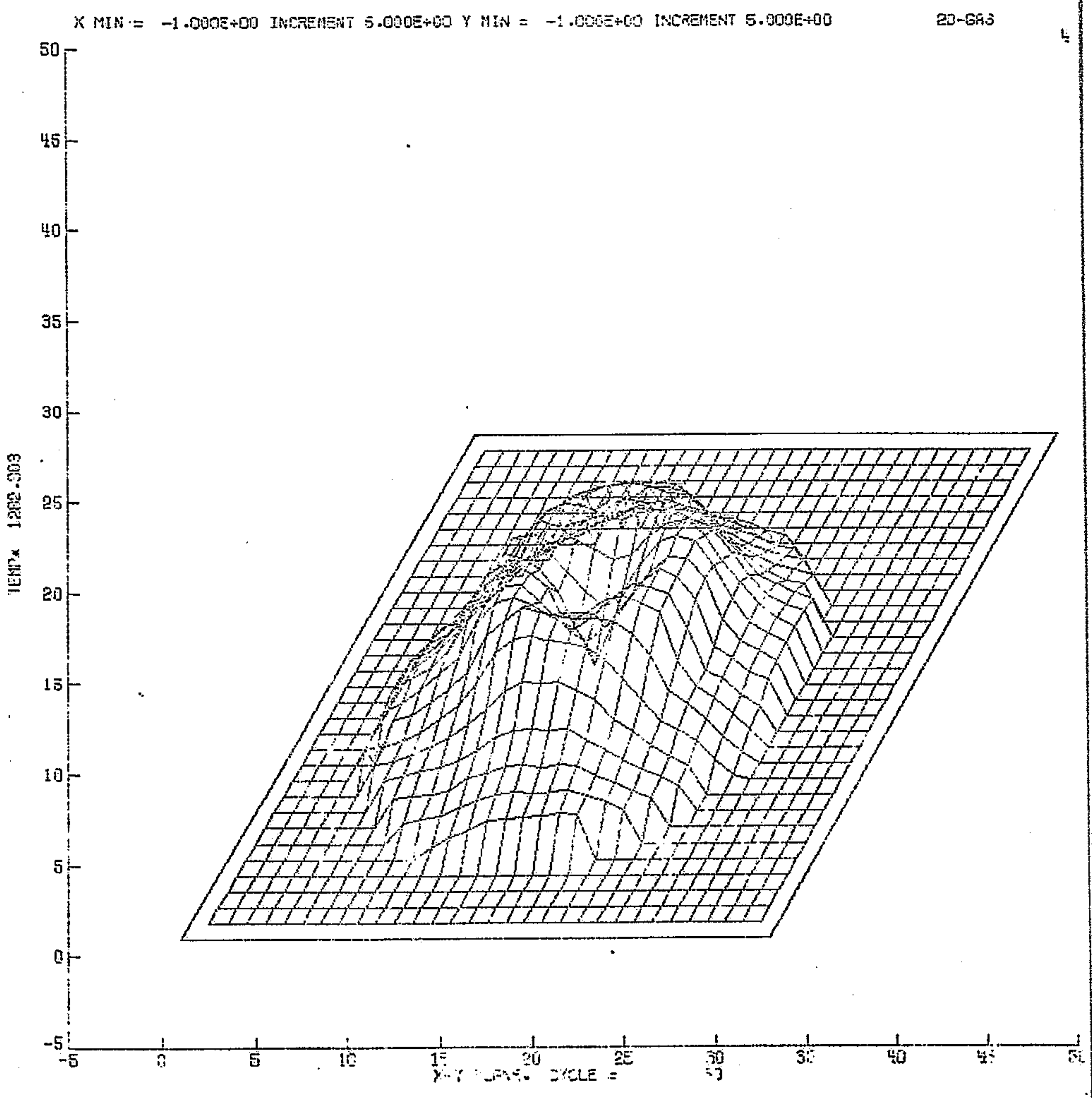

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
3
DENSITY* 26235.351
X-Y PLANE, CYCLE = 30

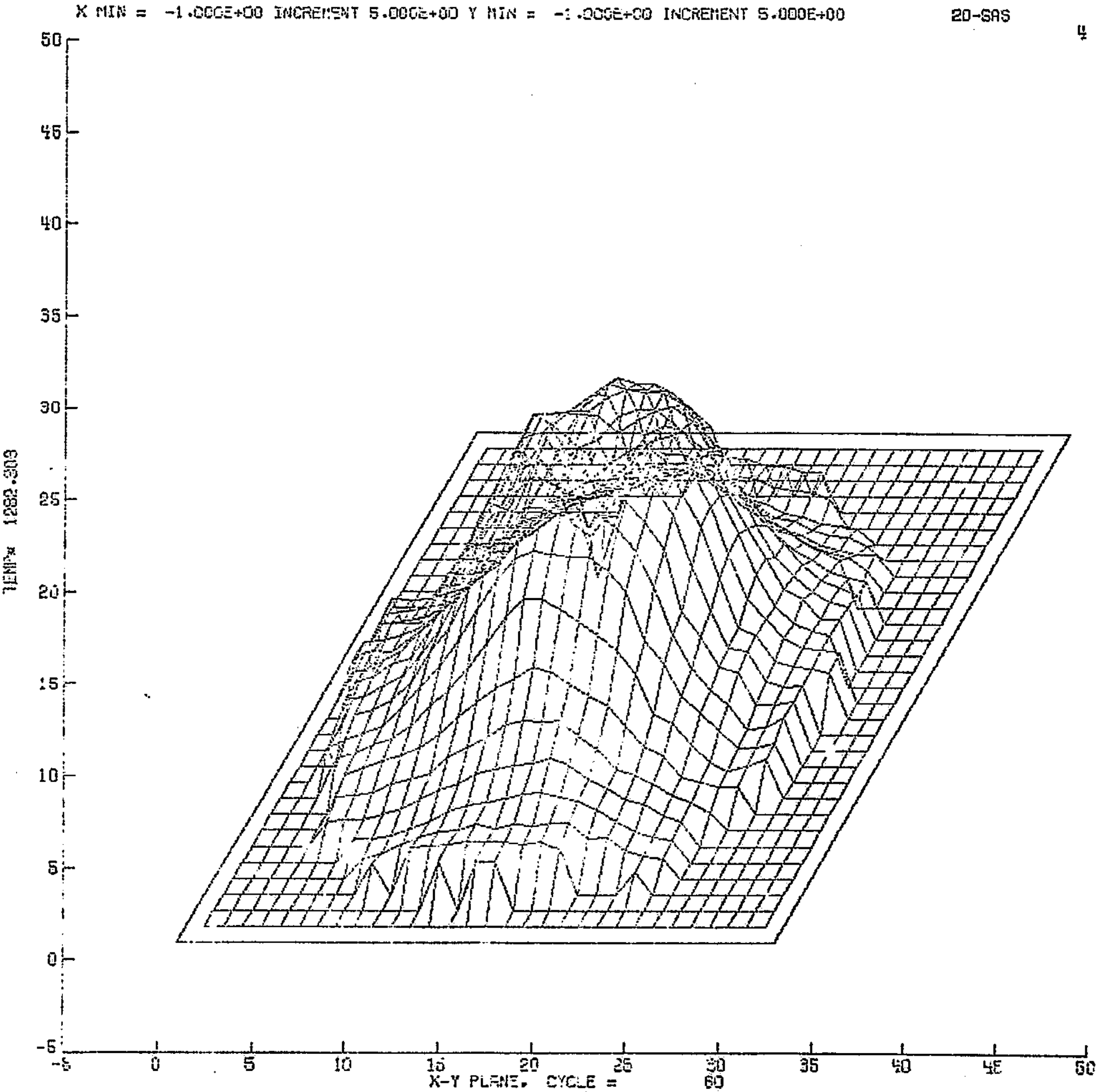
C-3

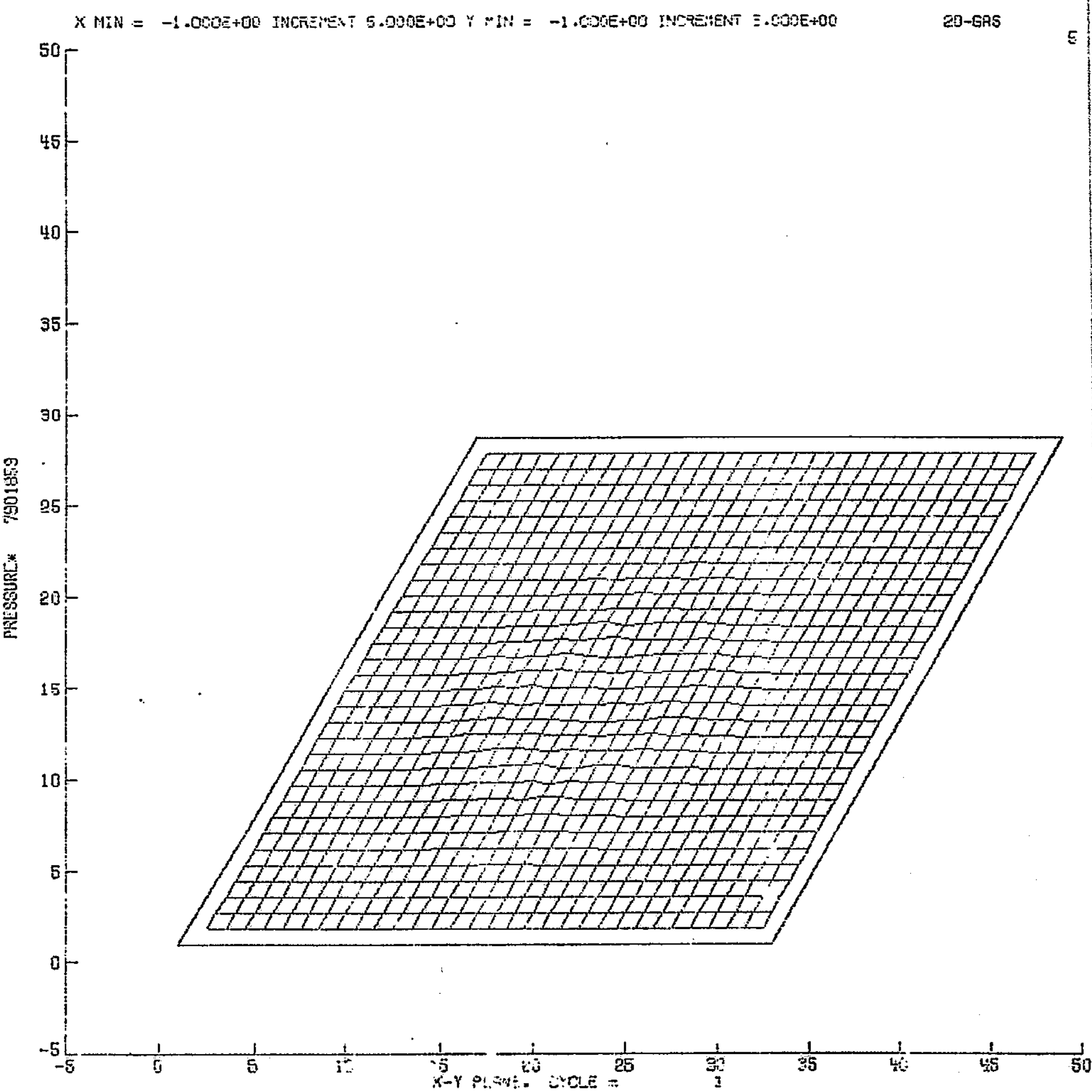
X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00 20-GAS




X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
TEMP* 1282.303
X-Y PLANE. CYCLE = 1
50
45
40
35
30
25
20
15
10
5
0
-5
-5
0
5
10
15
20
25
30
35
40
45

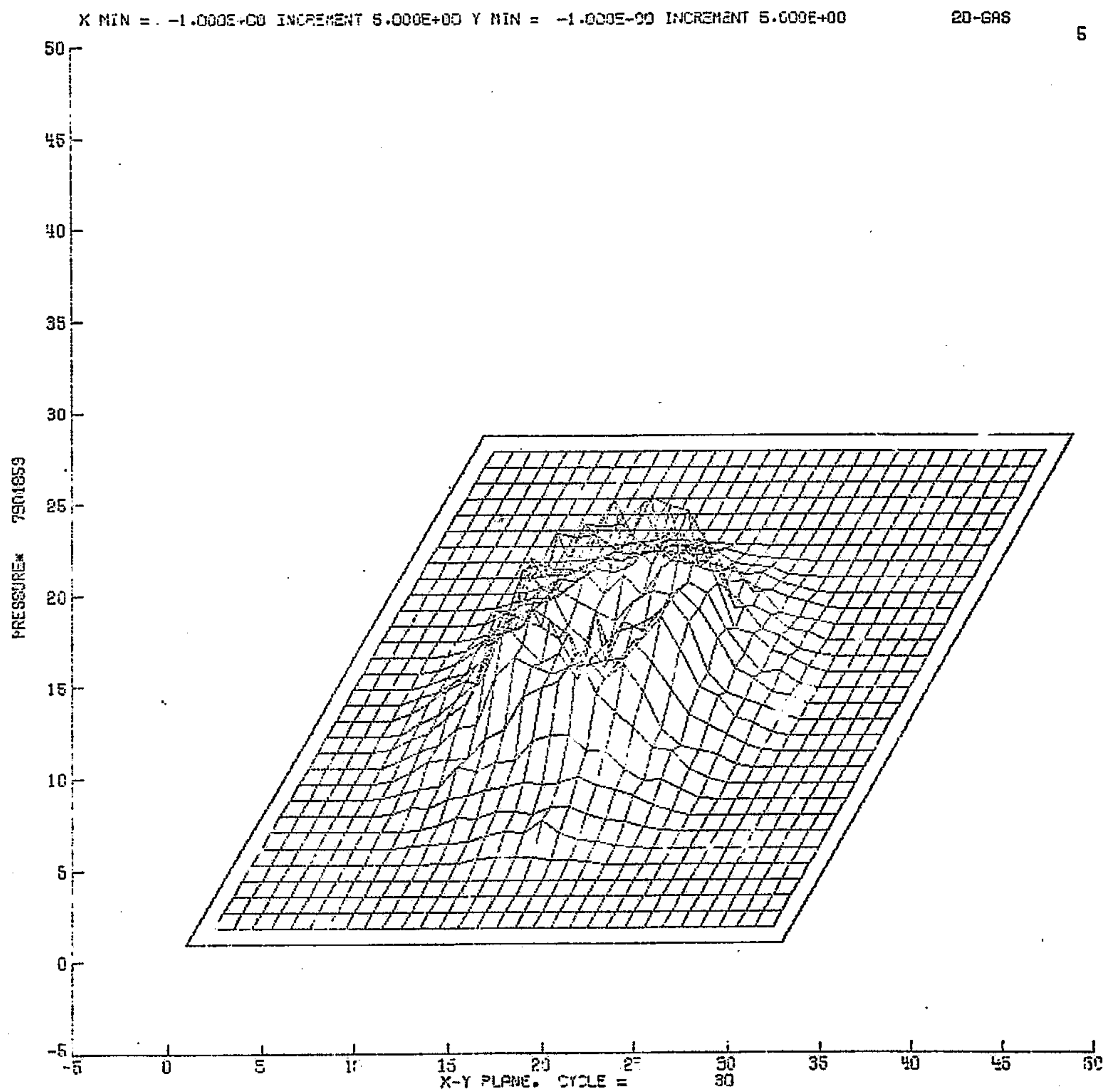

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
TEMP* 1282.303
CYCLE =
-5
0
5
10
15
20
25
30
35
40
45
50

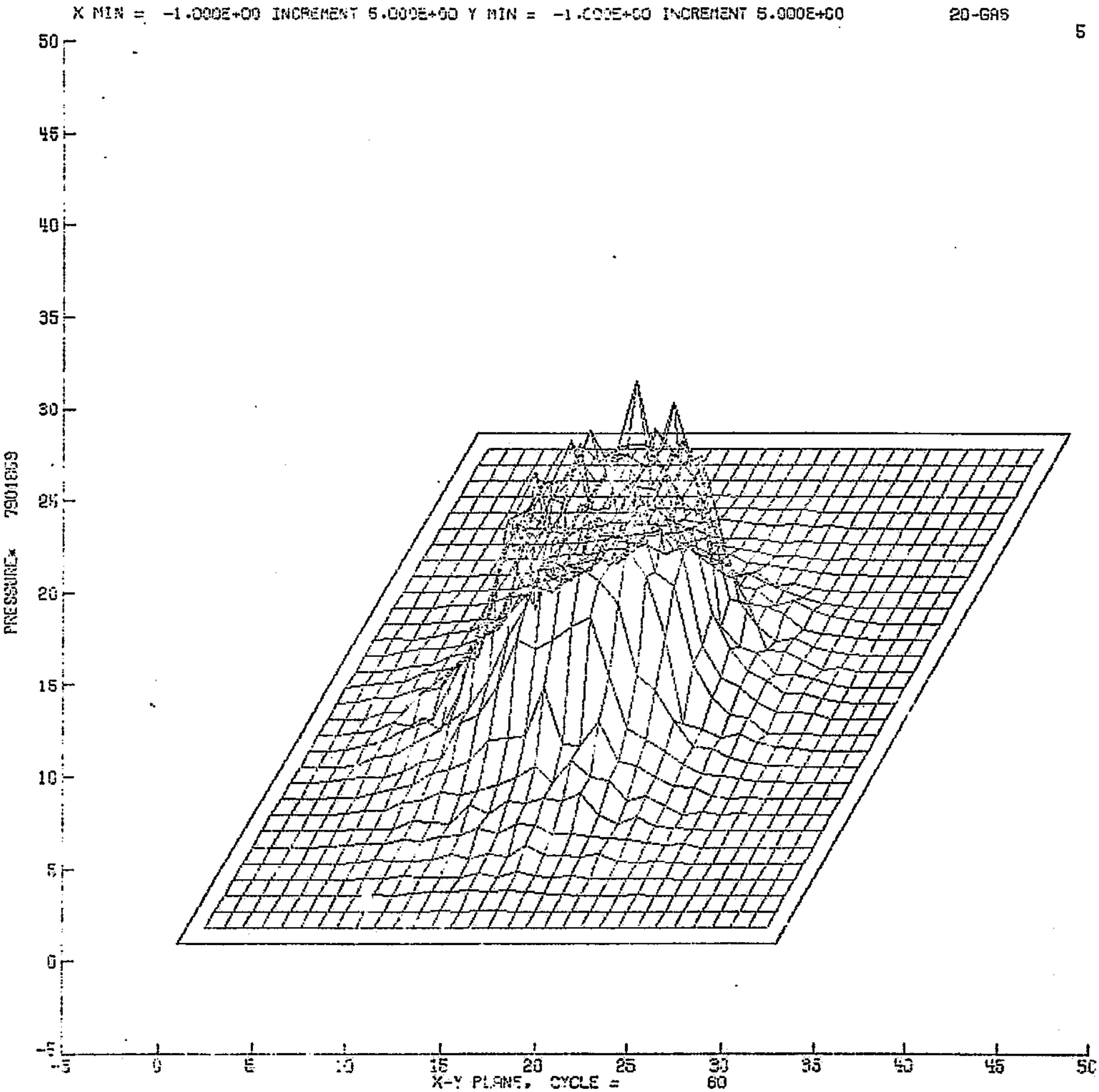

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
20-SAS
TEMP* 1282.303
X-Y PLANE, CYCLE = 60



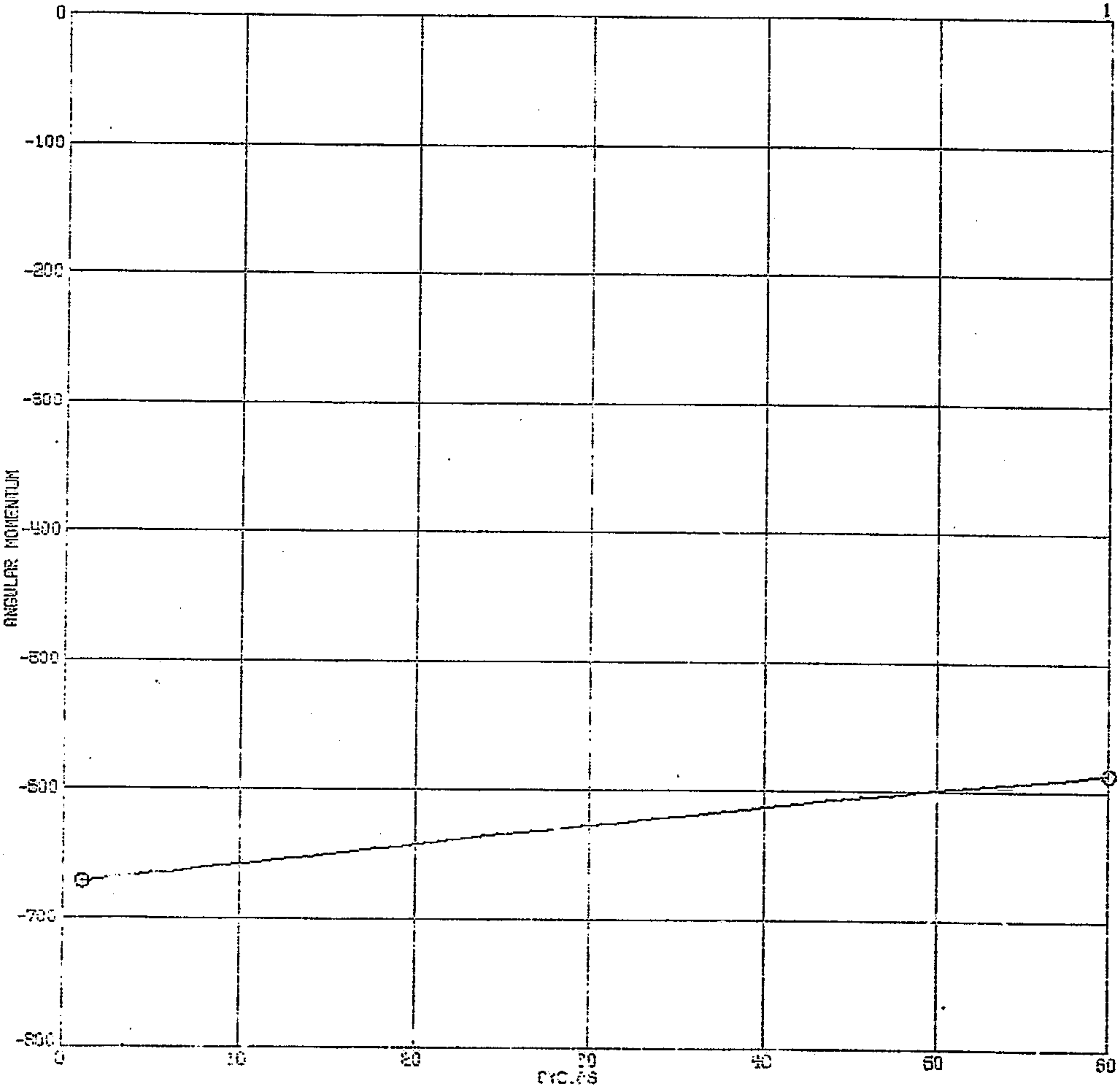
X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
5
PRESSURE* 7901853
X-Y PLANE. CYCLE = 1

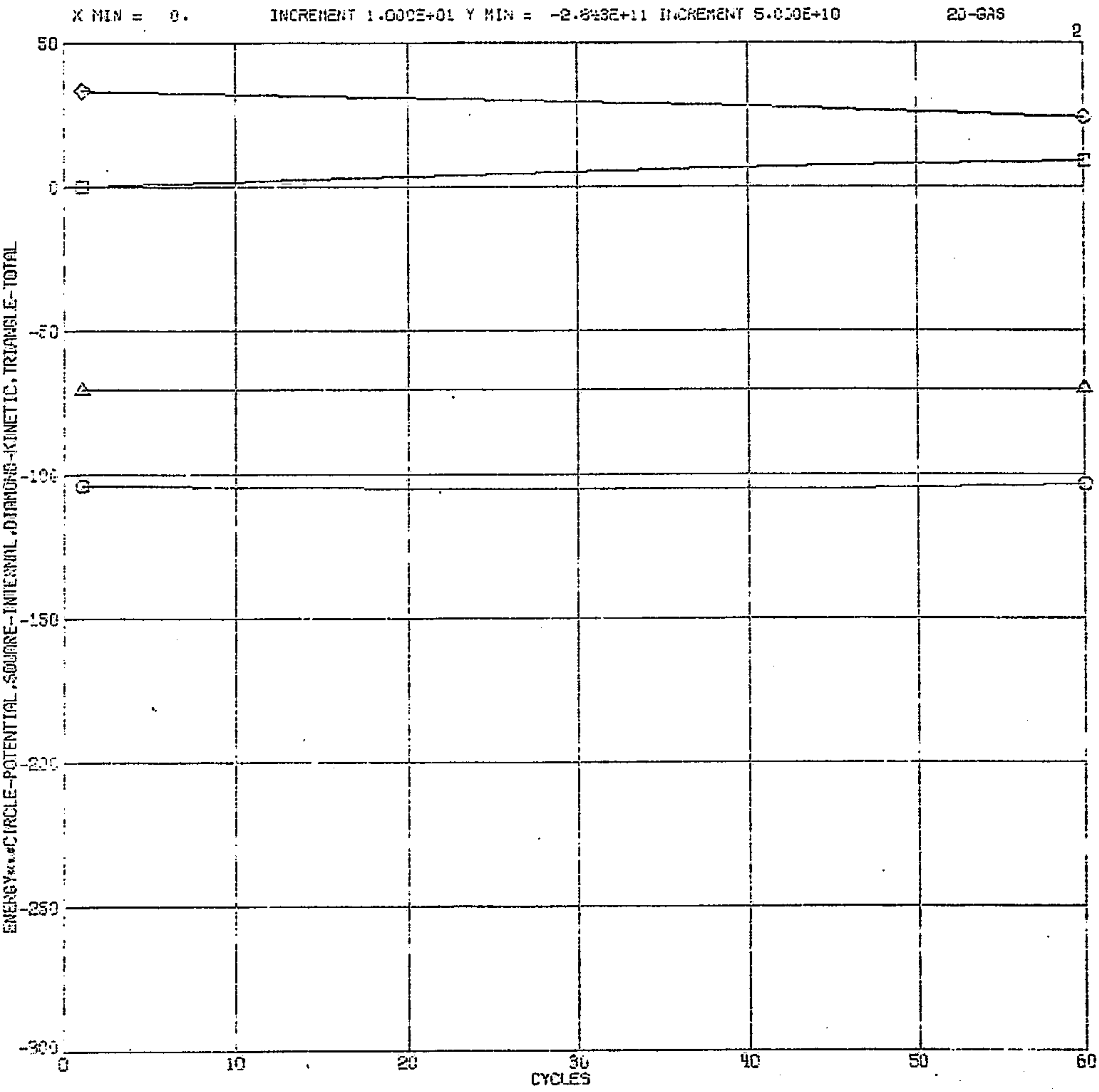
G-15

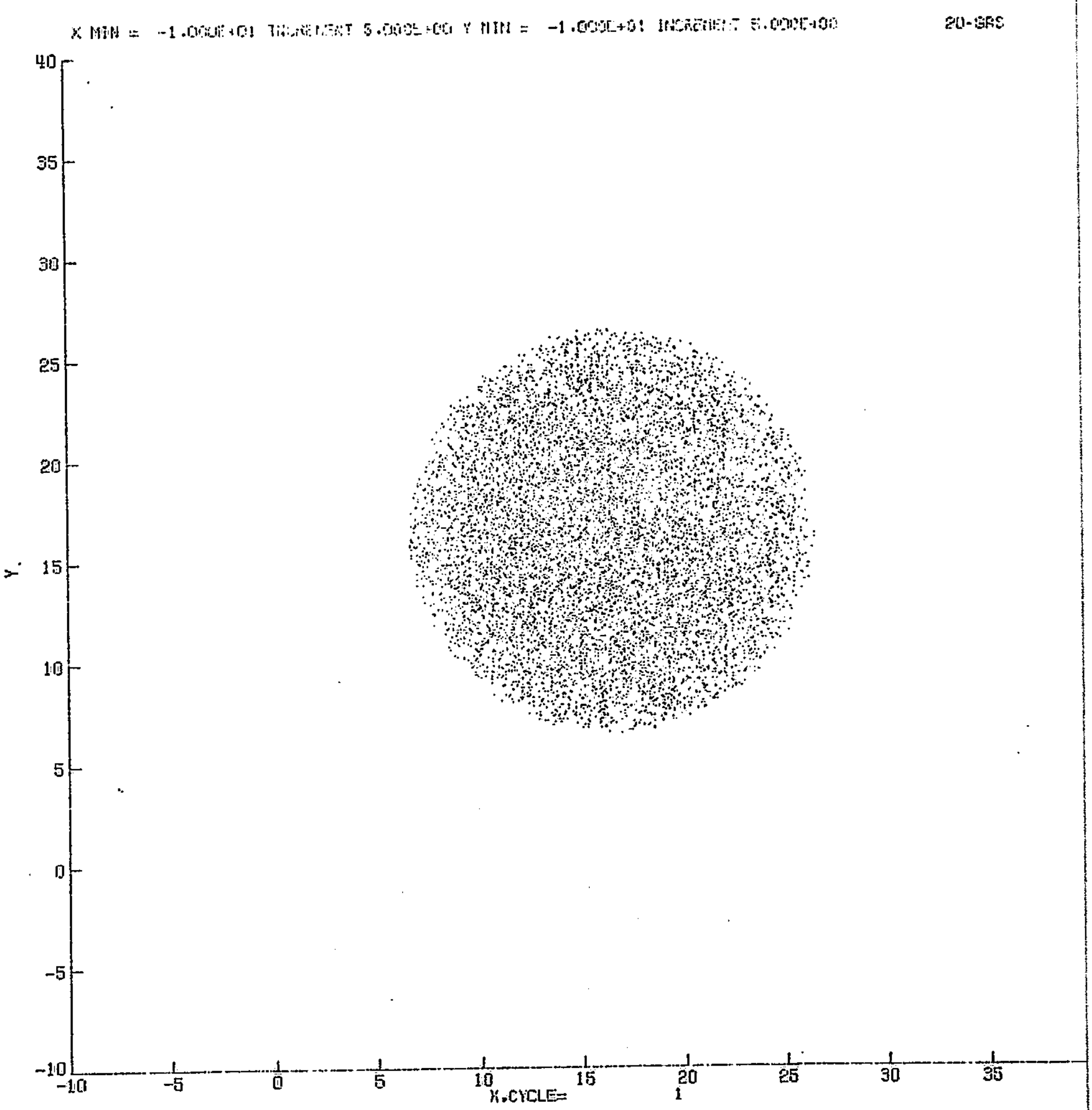

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
PRESSURE* 7901853
X-Y PLANE. CYCLE = 30
50
45
40
35
30
25
20
15
10
5
0
-5
0
5
15
30
35
40
45
50

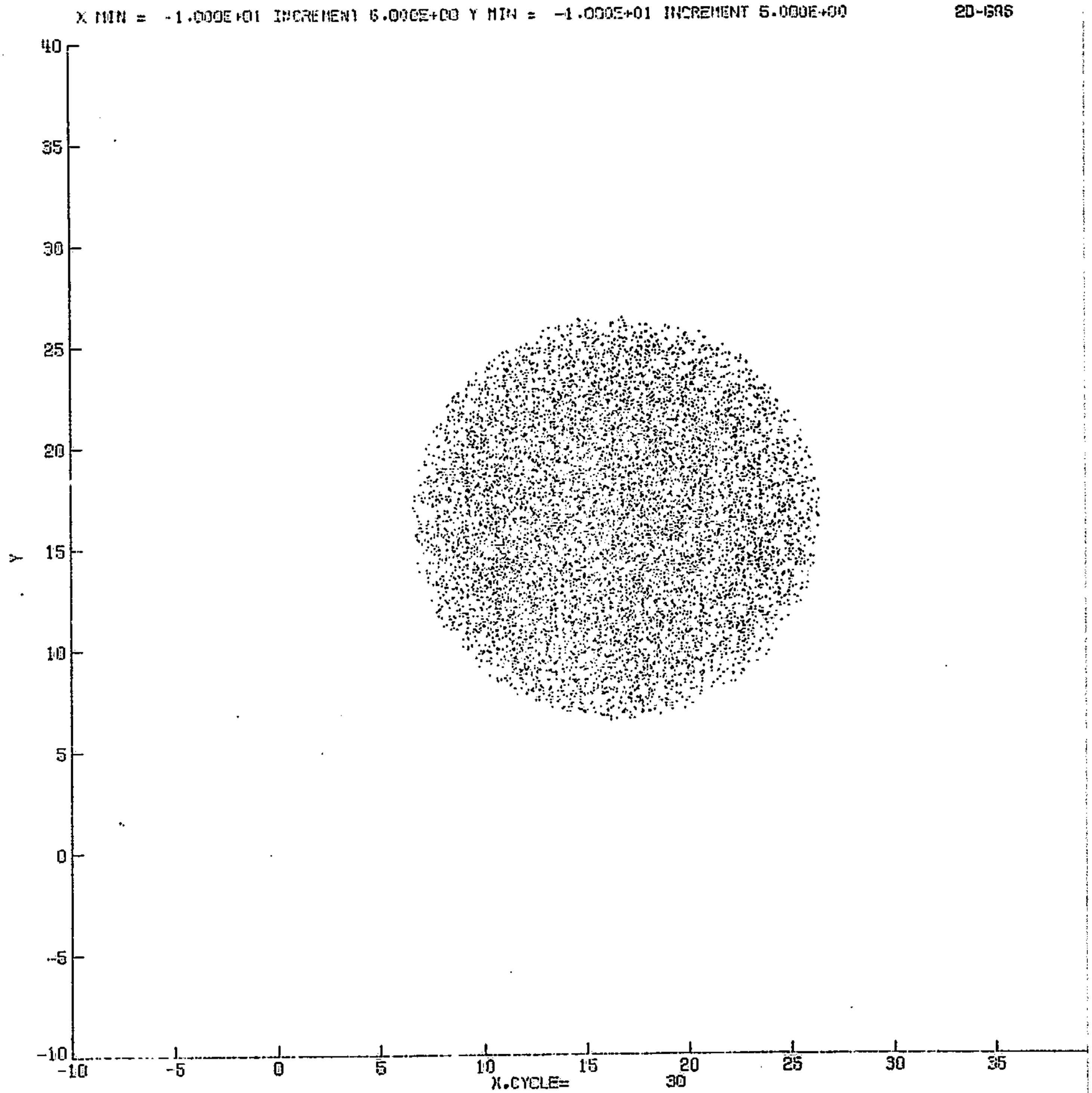

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
5
PRESSURE* 7901659
X-Y PLANE, CYCLE = 60
50
45
40
35
30
25
20
15
10
5
0
-5
0
5
10
15
20
25
30
35
40
45
50

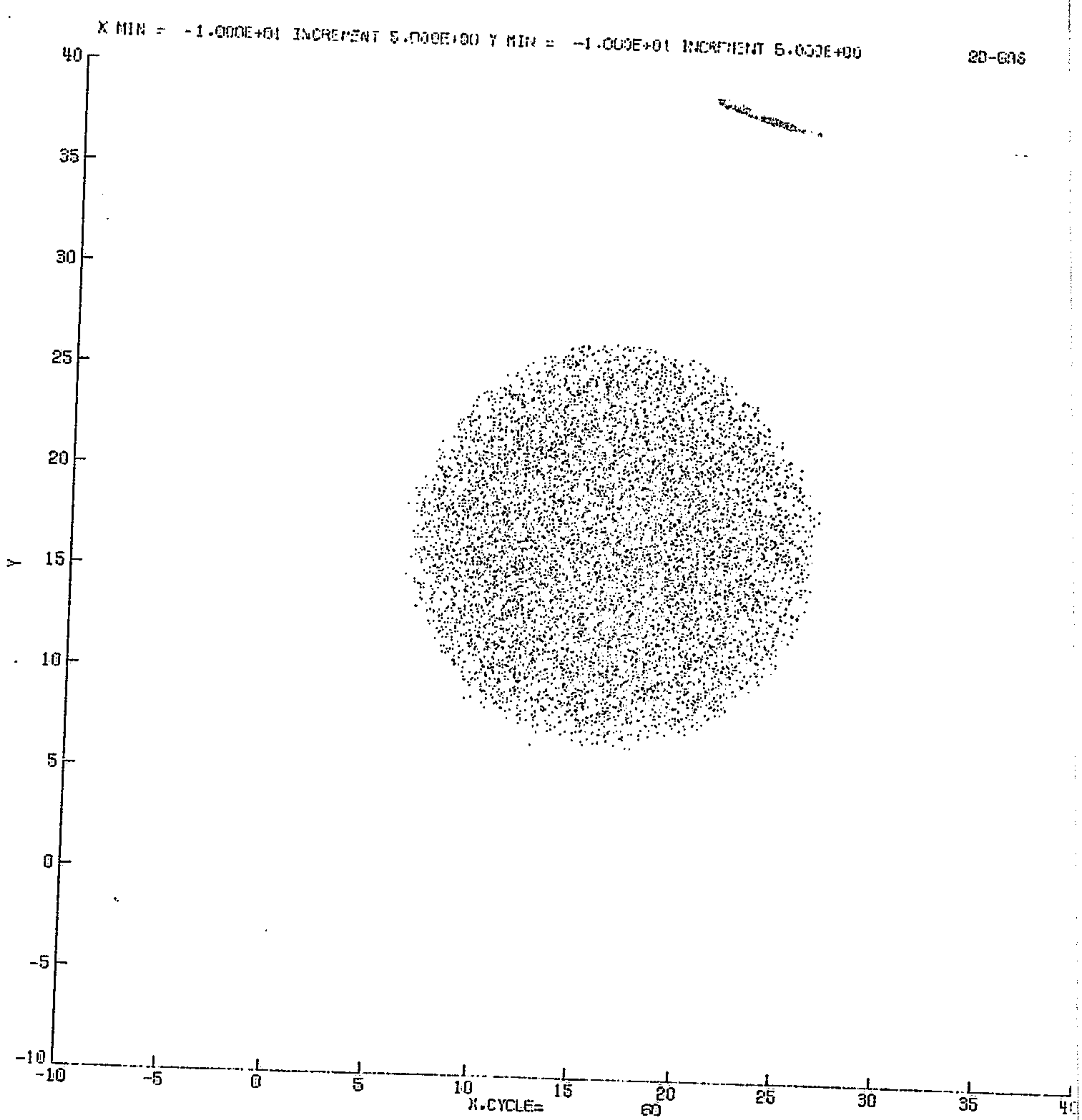
X MIN = 0. INCREMENT 1.000E+01 Y MIN = -7.885E+08 INCREMENT 1.000E+08 2D-GAS



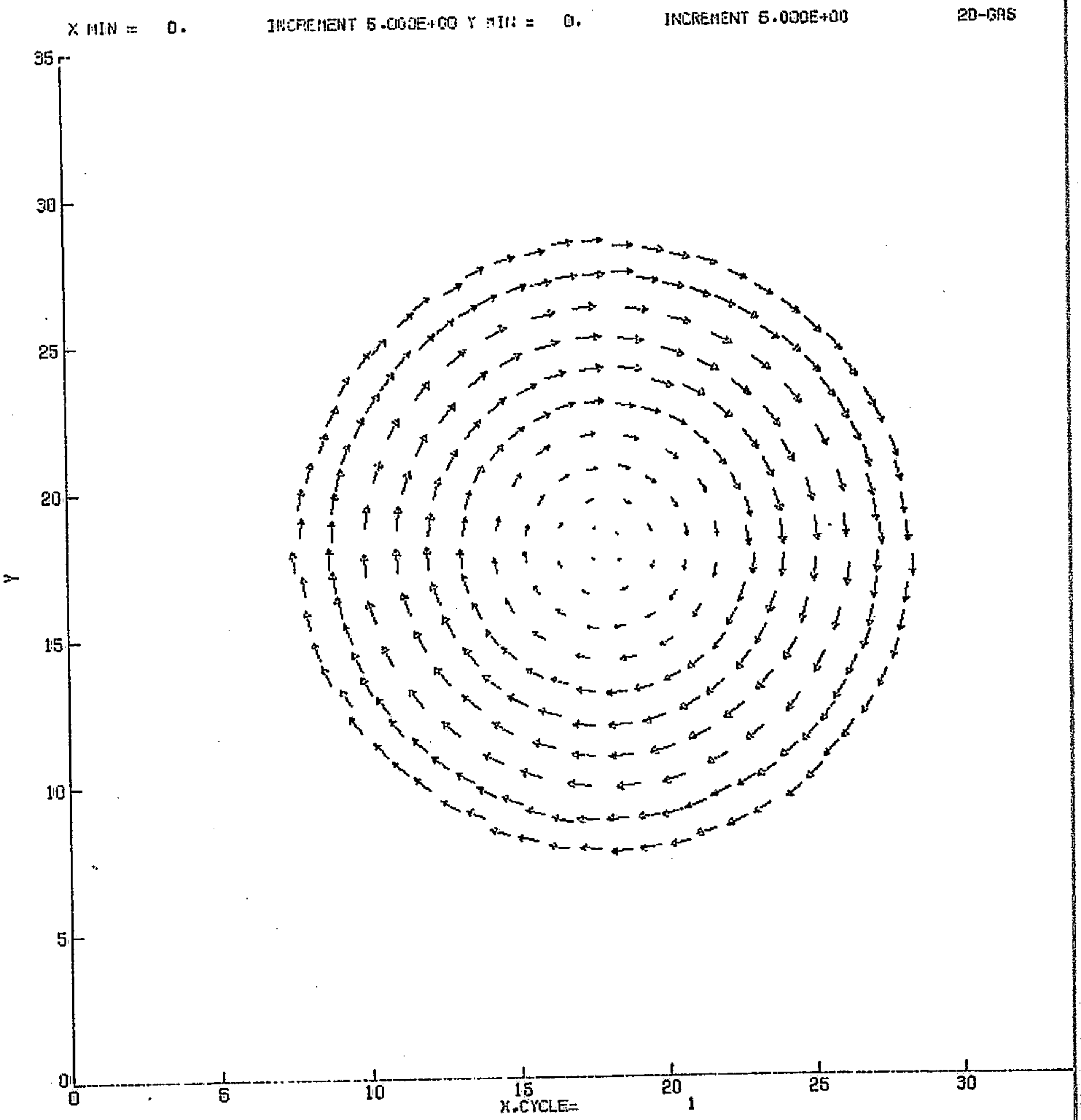

X MIN = 0. INCREMENT 1.000E+01 Y MIN = -2.843E+11 INCREMENT 5.000E+10
2D-GAS
2
ENERGY***CIRCLE-POTENTIAL,SQUARE-INTERNAL,DIAMOND-KINETIC,TRIANGLE-TOTAL
CYCLES
50
0
-50
-100
-150
-200
-250
-300
0
10
20
30
40
50
60


X MIN = -1.000E+01 INCREMENT 5.000E+00 Y MIN = -1.000E+01 INCREMENT 5.000E+00
20-GRS
40
35
30
25
20
15
10
5
0
-5
-10
Y
-10
-5
0
5
10
15
20
25
30
35
X.CYCLE= 1

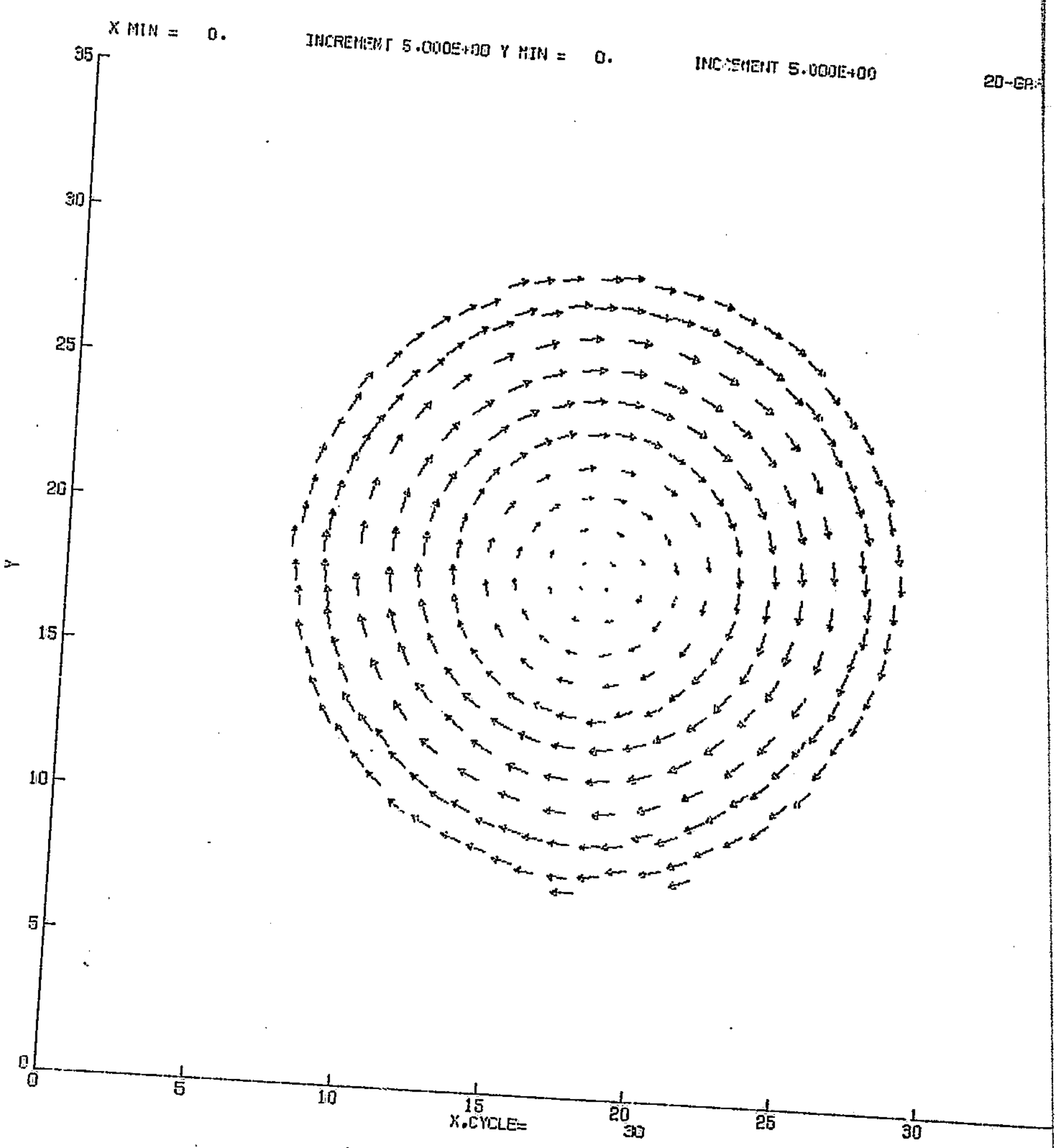

X MIN = -1.000E+01 INCREMENT 5.000E+00 Y MIN = -1.000E+01 INCREMENT 5.000E+00
2D-GAS
Y
X.CYCLE= 30

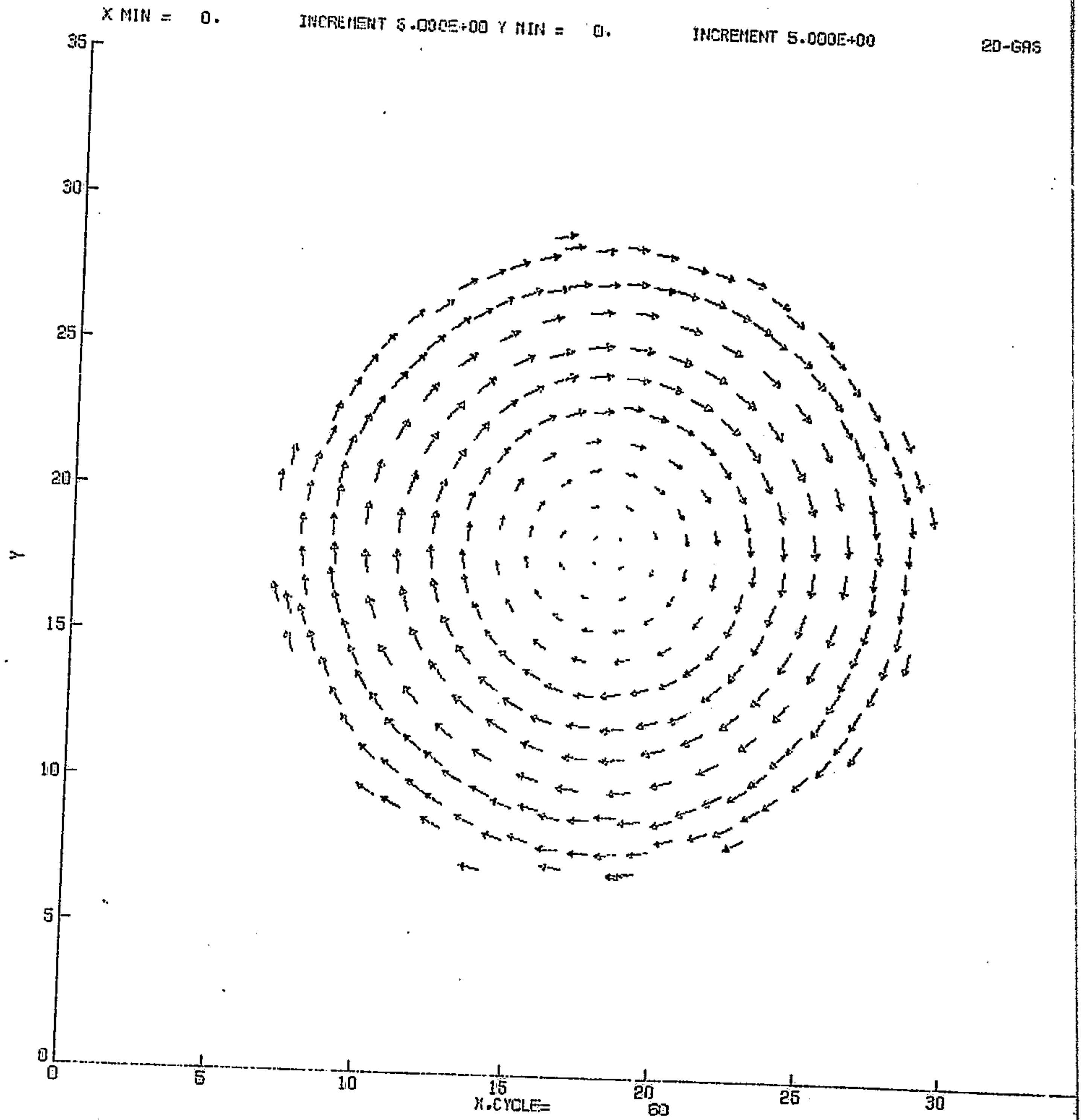

X MIN = -1.000E+01 INCREMENT 5.000E+00 Y MIN = -1.000E+01 INCREMENT 5.000E+00
20-608
Y
X.CYCLE= 60
40
35
30
25
20
15
10
5
0
-5
-10

C-22

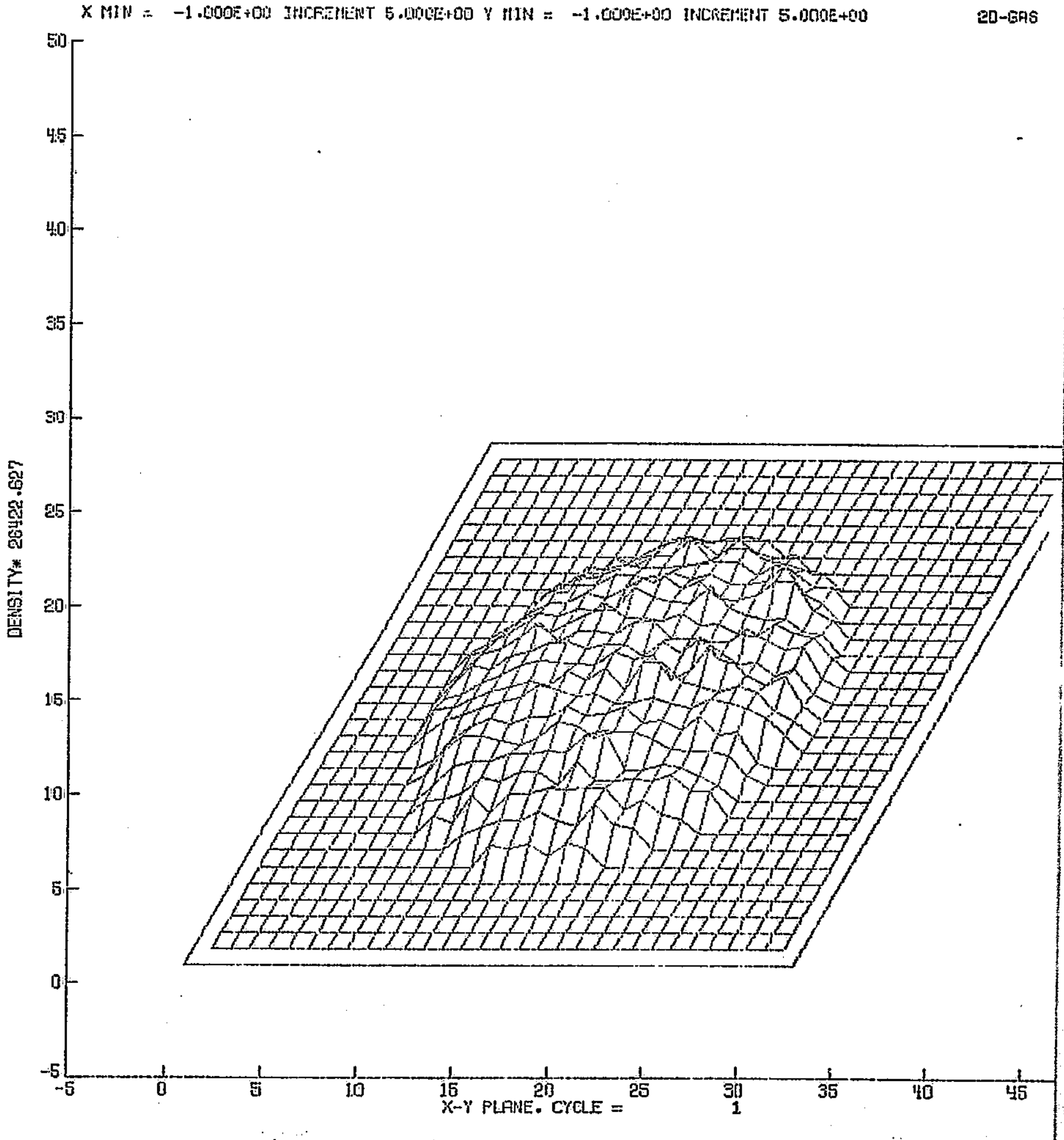


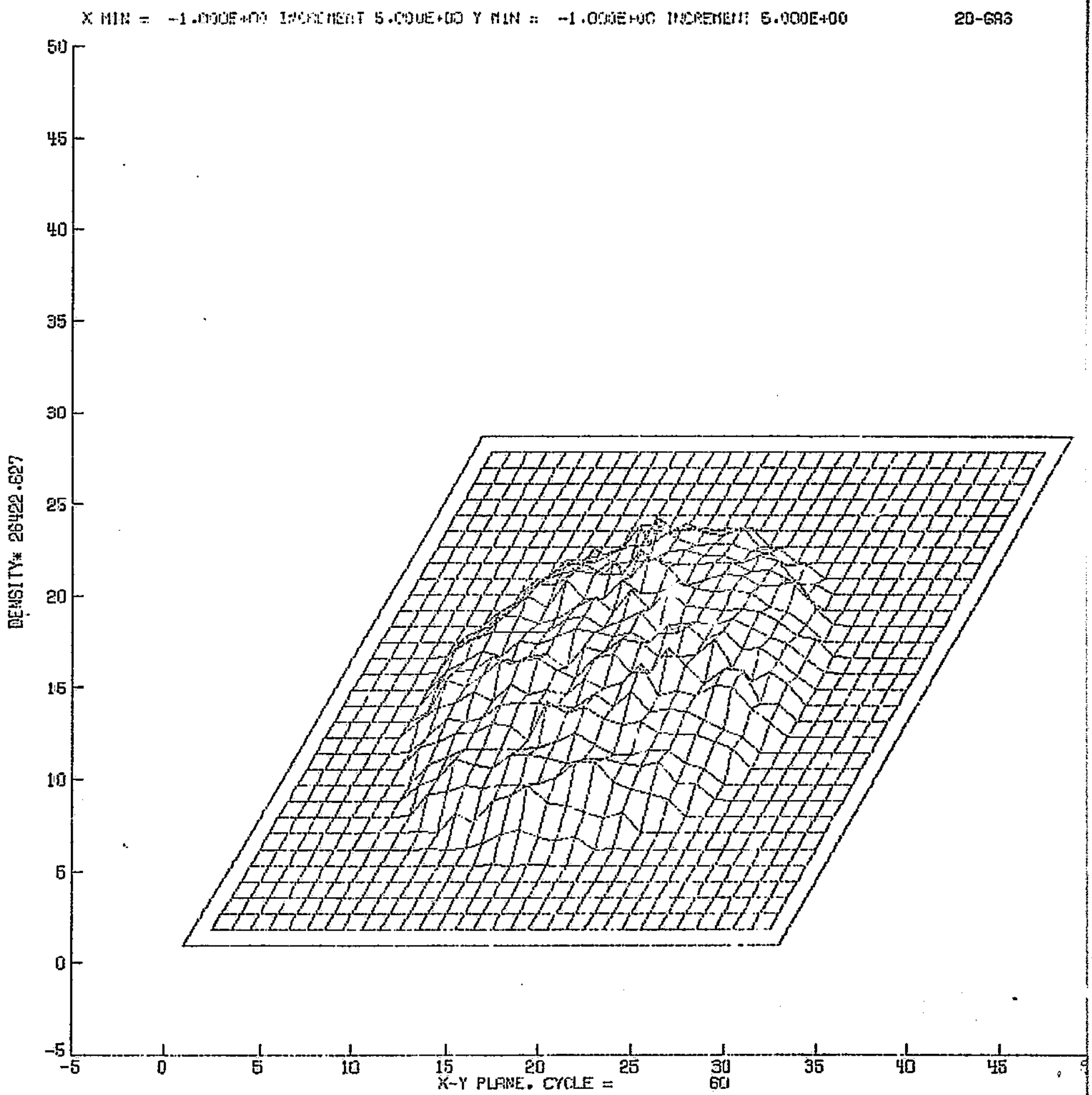
X MIN = 0.
INCREMENT 5.000E+00 Y MIN = 0.
INCREMENT 5.000E+00
2D-GAS
35
30
25
20
15
10
5
0
Y
0
5
10
15
20
25
30
X.CYCLE= 1


X MIN = 0.
INCREMENT 5.000E+00 Y MIN = 0.
INCREMENT 5.000E+00
20-GR
35
30
25
20
15
10
5
0
Y
0
5
10
15
20
25
30
X.CYCLE= 30

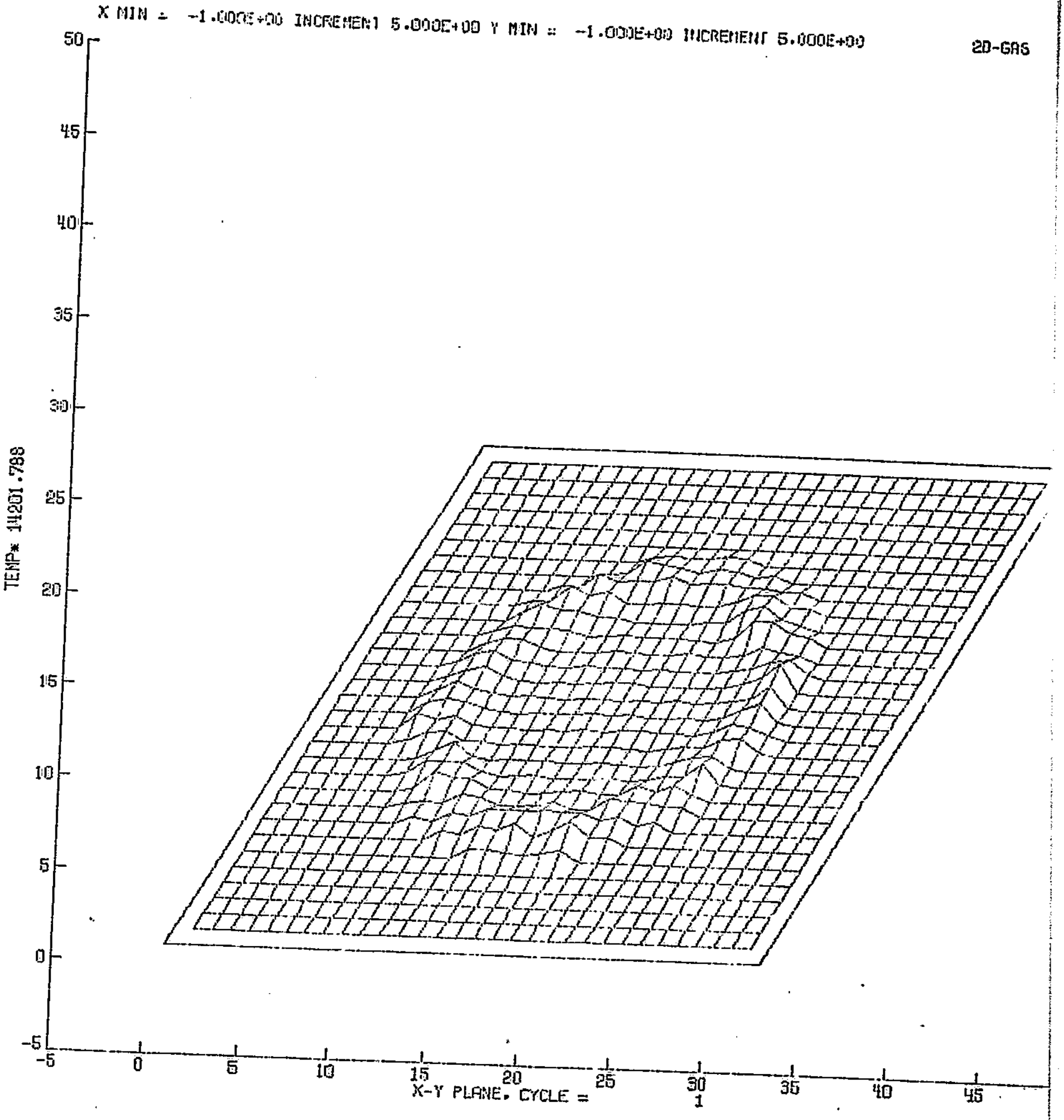

X MIN = 0.
INCREMENT 5.000E+00 Y MIN = 0.
INCREMENT 5.000E+00
2D-GAS
Y
35
30
25
20
15
10
5
0
0
5
10
15
20
25
30
X.CYCLE= 60

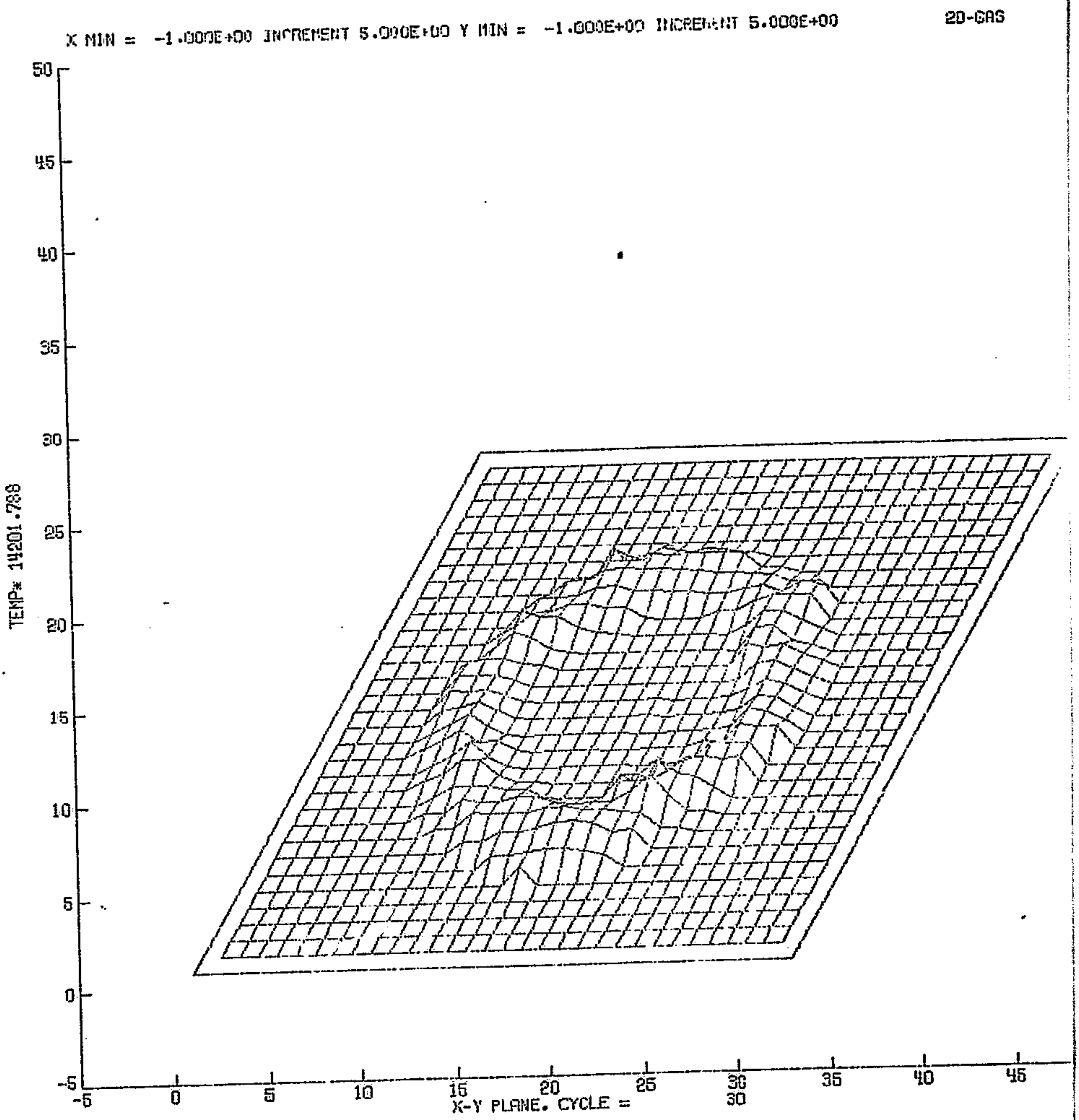
G-25

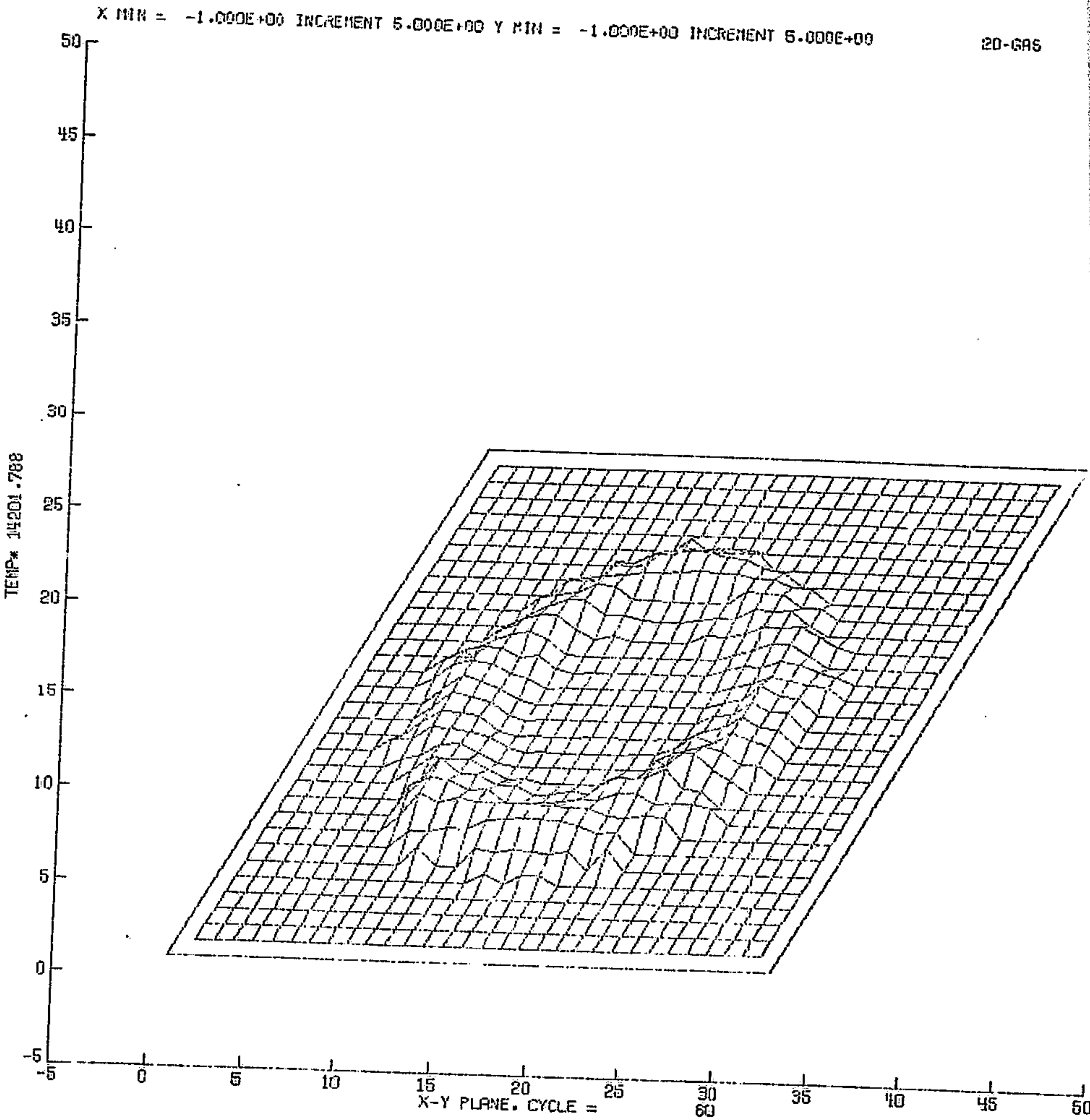

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
DENSITY* 26422.627
X-Y PLANE. CYCLE = 1
50
45
40
35
30
25
20
15
10
5
0
-5
-5
0
5
10
15
20
25
30
35
40
45

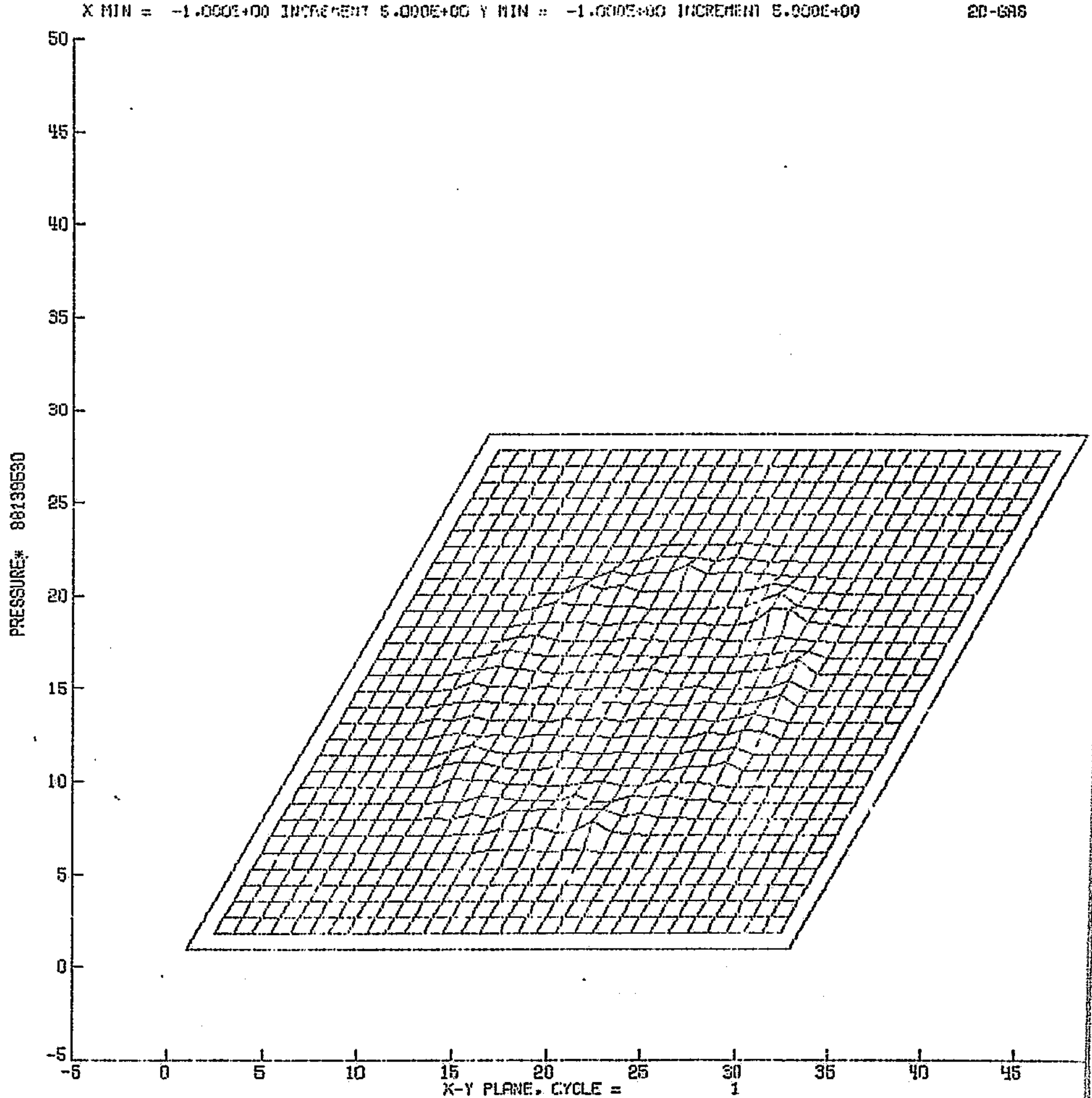


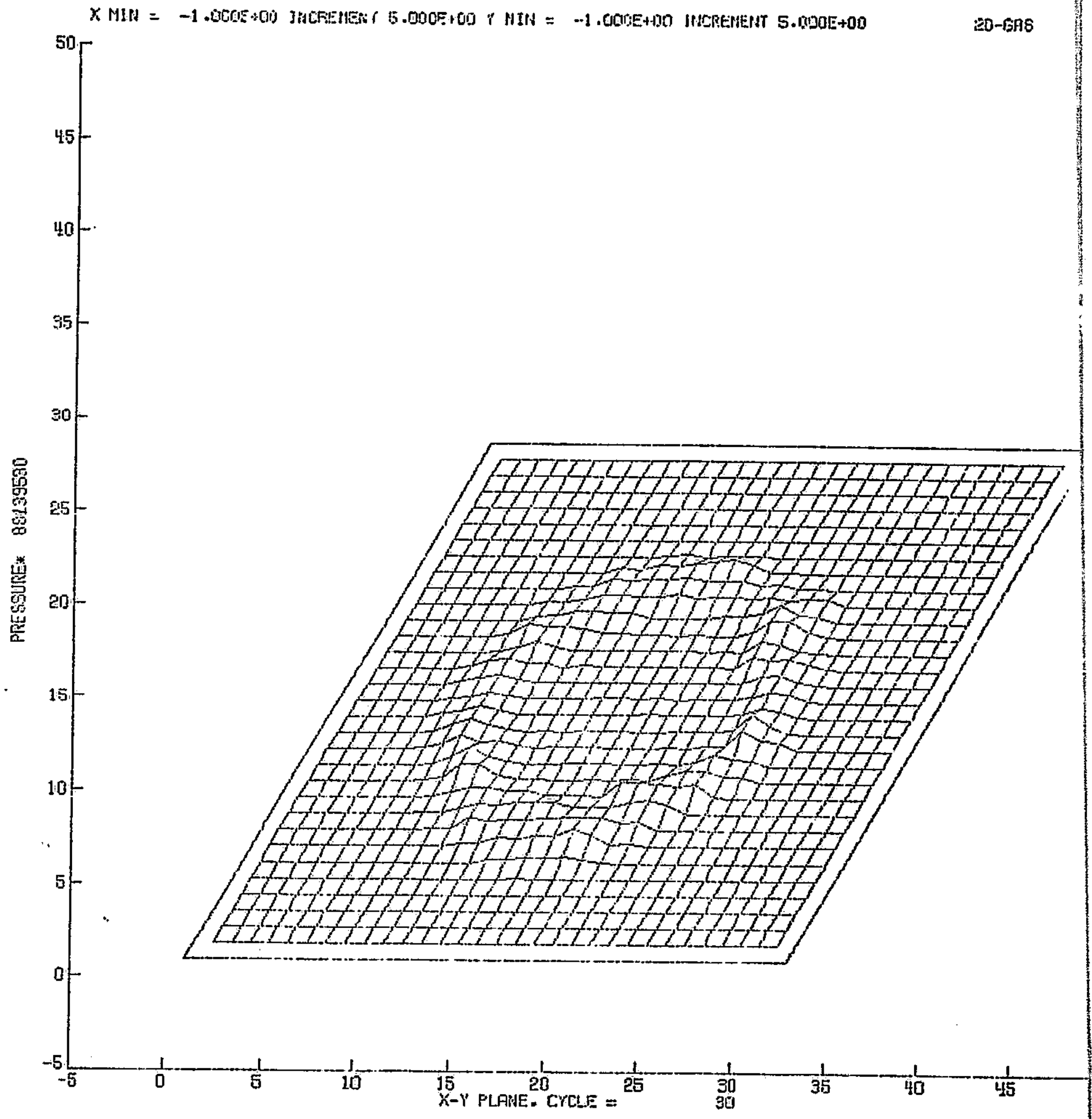
X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
DENSITY* 26422.627
X-Y PLANE, CYCLE = 60

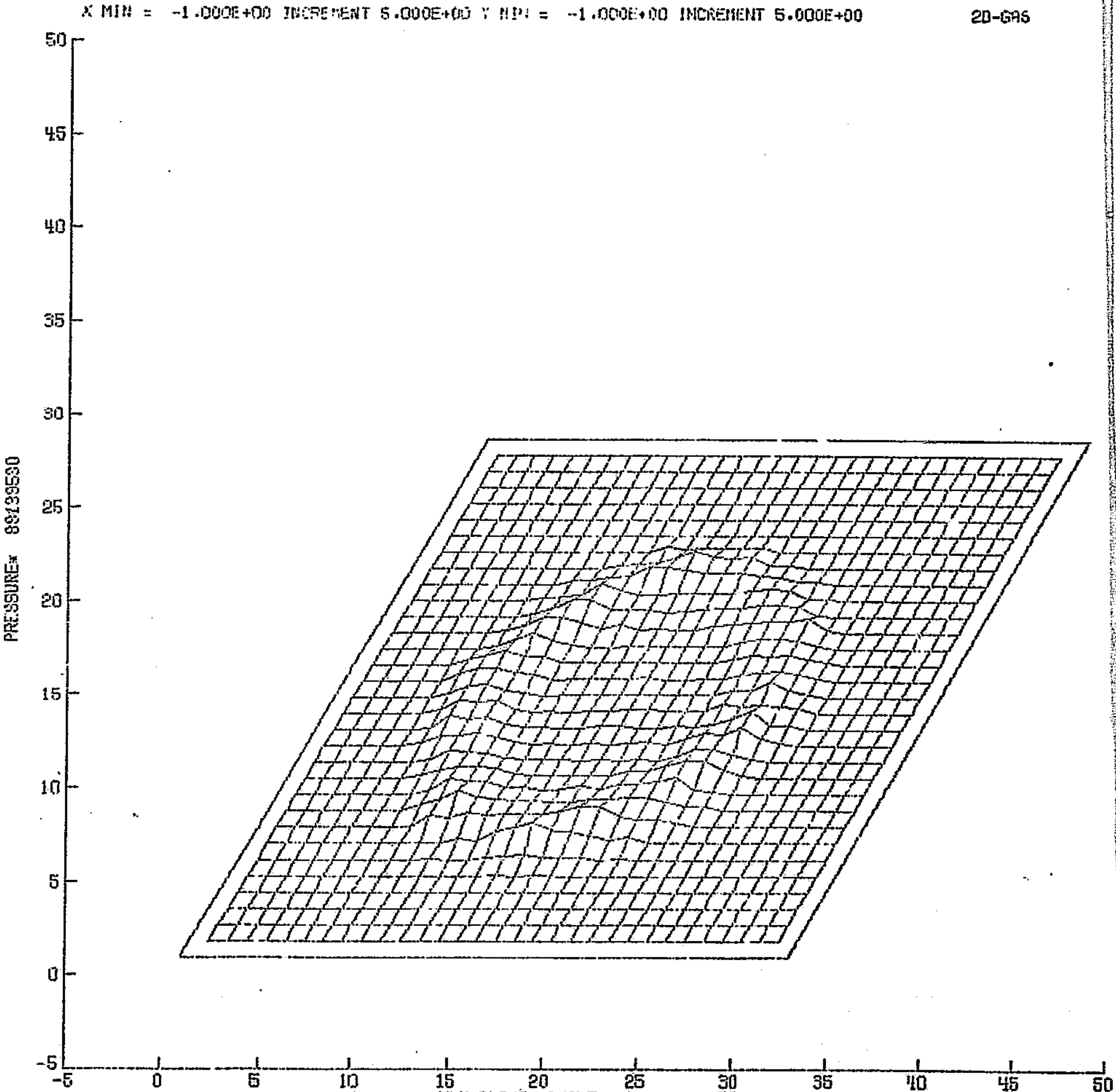

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
TEMP* 14201.788
50
45
40
35
30
25
20
15
10
5
0
-5
-5
0
5
10
15
20
25
30
35
40
45
X-Y PLANE, CYCLE = 1


X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
TEMP* 14201.738
X-Y PLANE. CYCLE = 30

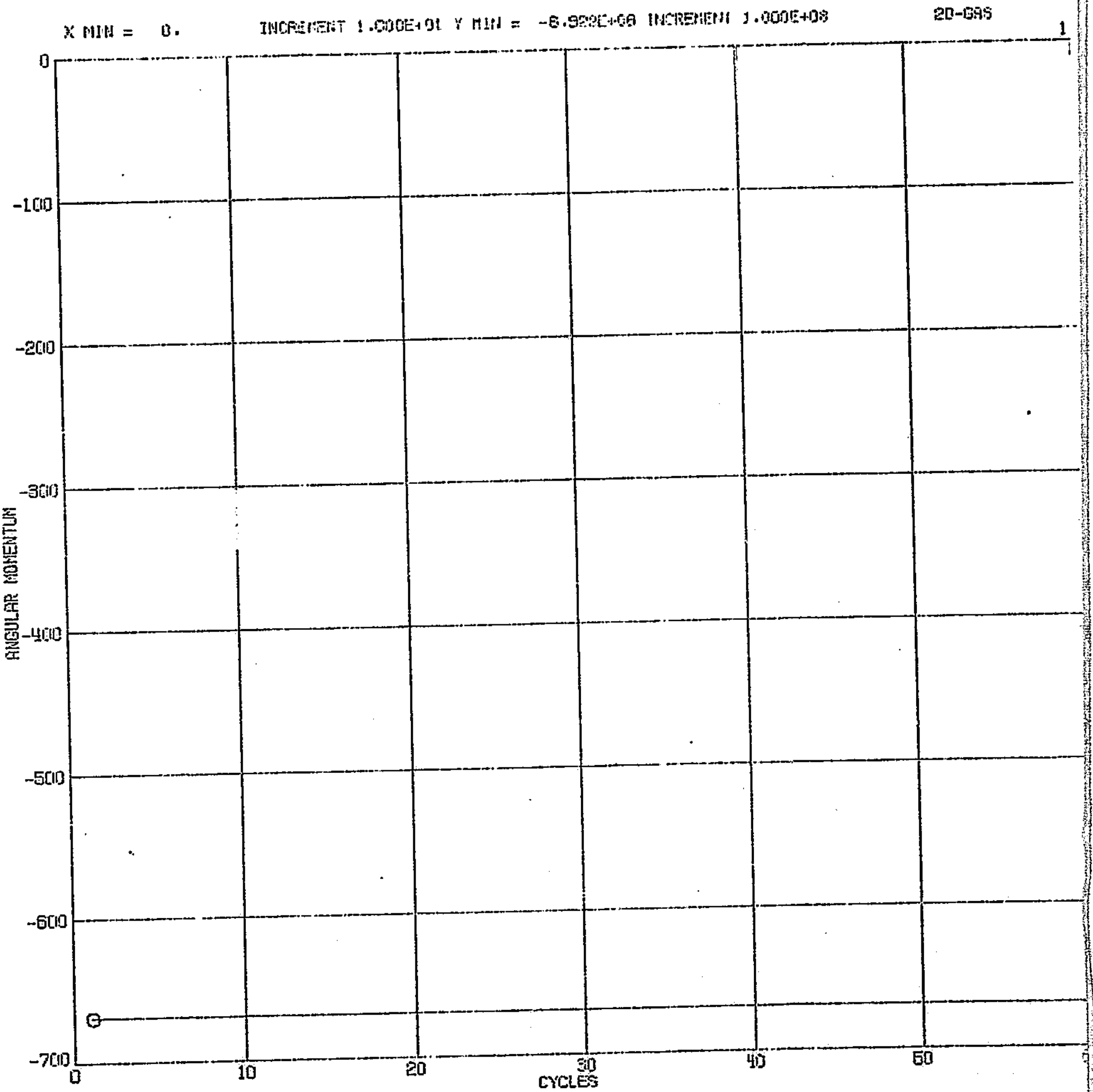

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
20-GAS
TEMP* 14201.788
50
45
40
35
30
25
20
15
10
5
0
-5
-5
0
5
10
15
20
25
30
35
40
45
50
X-Y PLANE. CYCLE = 60


X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
PRESSURE* 88139530
X-Y PLANE, CYCLE = 1
50
45
40
35
30
25
20
15
10
5
0
-5
-5
0
5
10
15
20
25
30
35
40
45

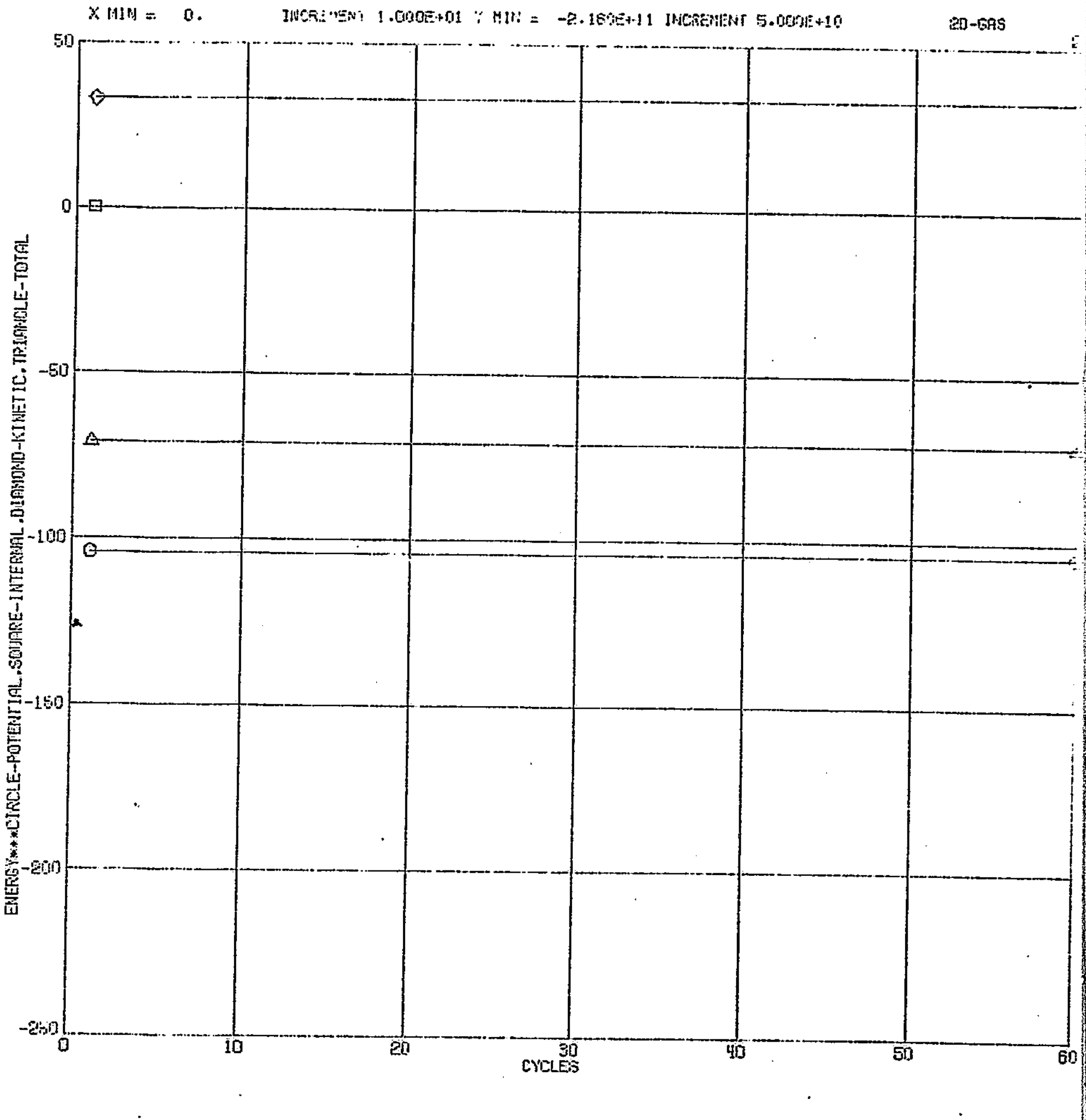

X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
PRESSURE* 88139530
X-Y PLANE. CYCLE = 30
-5
0
5
10
15
20
25
30
35
40
45
50



X MIN = -1.000E+00 INCREMENT 5.000E+00 Y MIN = -1.000E+00 INCREMENT 5.000E+00
2D-GAS
PRESSURE* 89139530
X-Y PLANE, CYCLE = 60
50
45
40
35
30
25
20
15
10
5
0
-5
-5
0
5
10
15
20
25
30
35
40
45
50



X MIN = 0.
INCREMENT 1.000E+01 Y MIN = -6.922E+08 INCREMENT 1.000E+08
2D-GAS
1
0
-100
-200
-300
-400
-500
-600
-700
ANGULAR MOMENTUM
0
10
20
30
40
50
CYCLES


X MIN = 0. INCREMENT 1.000E+01 Y MIN = -2.160E+11 INCREMENT 5.000E+10
2D-GAS
ENERGY***CIRCLE-POTENTIAL,SQUARE-INTERNAL,DIAMOND-KINETIC,TRIANGLE-TOTAL
CYCLES
50
0
-50
-100
-150
-200
-250
0
10
20
30
40
50
60