NASA-CR-204348

Final Report: NASA-PO H-13048D
NASA Marshall Space Flight Center
Marshall Space Flight Center, AL

Cold-Flow Study of Low Frequency Pressure Instability in Hybrid Rocket Motors

Rhonald M. Jenkins
Auburn University, Alabama

May 8, 1997

Final Report: NASA-PO H-13048D
NASA Marshall Space Flight Center
Marshall Space Flight Center, AL

Cold-Flow Study of Low Frequency Pressure Instability in Hybrid Rocket Motors

Rhonald M. Jenkins
Auburn University, Alabama

Past experience with hybrid rockets has shown that certain motor operating conditions are conducive to the formation of low frequency pressure oscillations, or flow instabilities, within the motor. Both past and present work in the hybrid propulsion community acknowledges deficiencies in the understanding of such behavior, though it seems probable that the answer lies in an interaction between the flow dynamics and the combustion heat release. Knowledge of the fundamental flow dynamics is essential to the basic understanding of the overall stability problem. A first step in this direction was a study conducted at NASA Marshall Space Flight Center (MSFC), centered around a laboratory-scale two dimensional water flow model of a hybrid rocket motor. Principal objectives included: (1) visualization of flow and measurement of flow velocity distributions: (2) assessment of the importance of shear layer instabilities in driving motor pressure oscillations; (3) determination of the interactions between flow induced shear layers with the mainstream flow, the secondary (wall) throughflow, and solid boundaries; (4) investigation of the interactions between wall flow oscillations and the mainstream flow pressure distribution. The investigation proceeded in two portions:

The first portion consisted of experimental flow visualization and measurement within a test bed developed from a ⅓ scale model of an 11 inch (28cm) subscale Solid Rocket Combustion Simulator (SRCS) hybrid motor which has recently undergone (hot-fire) tests at MSFC. The test section was installed in a dual closed-loop water flow facility, originally developed in 1990 to investigate fluid flow through porous materials. Oxidizer, or main stream flow, was simulated by a water stream entering through injector slots located in the forward section of the "motor"; this flow may be injected either axially (along the motor axis) or radially. Fuel grain "burning" (mass injection only) was simulated by a secondary water stream entering the test section through 100μ sintered bronze porous plate material. The test section side walls were constructed of clear acrylic for flow visualization and optical velocity measurements. Flow visualization was accomplished through injection of very small helium bubbles into the mainstream and/or secondary flow streams and recording their movement with a 1000 frames/second video recorder. Qualitative flow field mapping was accomplished with an LDV system. A test section schematic is shown in Figure 1. A complete description of the system setup and results obtained is given in Reference 1.

The second portion of the investigation consisted of the generation of an analytical model

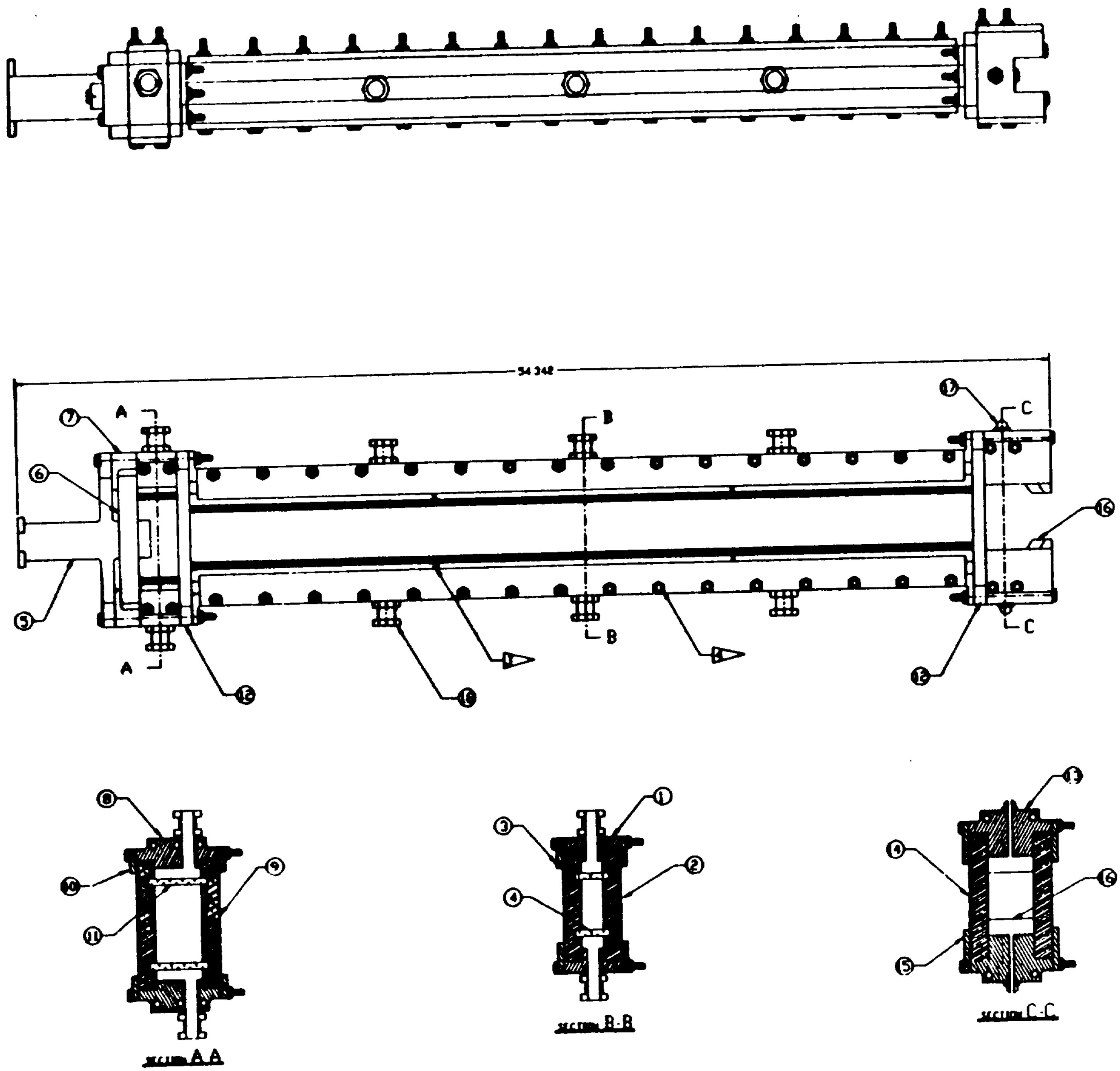

Figure 1: Test Section Schematic

to supplement the experimental data obtained, and is the subject of this report. The model is an extension to hybrid rocket motors of an analysis originally developed by Ciucci and Jenkins for solid rocket motors[2,3,4,5]. The analysis will handle unsteady, two-dimensional (or axisymmetric), incompressible or compressible flows characterized by the presence of one or more jets issuing into a finite volume. Boundary walls may be porous or non-porous. The two-dimensional, unsteady, compressible, full Navier-Stokes equations are employed. The overall computational space is characterized by one or more planes of symmetry, with the analysis restricted to a single plane of symmetry for computational efficiency. Though not applicable to the present application (incompressible flow), the model can easily be extended to incorporate locally imbedded supersonic regions. The principal governing fluid flow equations are as follows:

Conservation of mass:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = \phi_c$$

Conservation of Momentum:

$$\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u^2}{\partial x} + \frac{\partial \rho uv}{\partial y} = - \frac{\partial p}{\partial x} + \frac{\partial \mu}{\partial x} [\frac{4}{3}\frac{\partial u}{\partial x} - \frac{2}{3}\frac{\partial v}{\partial y}] + \frac{\partial \mu}{\partial y} [\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}]$$

$$+ \mu [\frac{4}{3}\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{1}{3}\frac{\partial^2 v}{\partial x \partial y}] + \phi_{xm}$$

$$\frac{\partial \rho v}{\partial t} + \frac{\partial \rho uv}{\partial x} + \frac{\partial \rho v^2}{\partial y} = - \frac{\partial p}{\partial y} + \frac{\partial \mu}{\partial x} [\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}] + \frac{\partial \mu}{\partial y} [\frac{4}{3}\frac{\partial v}{\partial y} - \frac{2}{3}\frac{\partial u}{\partial x}]$$

$$+ \mu [\frac{4}{3}\frac{\partial^2 v}{\partial y^2} + \frac{\partial^2 v}{\partial x^2} + \frac{1}{3}\frac{\partial^2 u}{\partial x \partial y}] + \phi_{ym}$$

Conservation of Energy:

$$\frac{\partial \rho e}{\partial t}+\frac{\partial \rho u e}{\partial x}+\frac{\partial \rho v e}{\partial y} = \frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k\frac{\partial T}{\partial y}\right)-p\left(\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}\right)$$

$$+ 2\mu\left[\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial v}{\partial y}\right)^2+\frac{1}{2}\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right)^2\right] + \Phi_e$$

In the above equations t is time, ρ is density, p is pressure, u and v are velocity components associated with x and y spacial directions, respectively, T is temperature, e is energy, and μ and k are flow viscosity and thermal conductivity, respectively. The Φ terms are source terms whose exact form depends upon whether the computational domain is two-dimensional or axisymetric (2-D in this case). Though a turbulence model can be built in to the analysis, only laminar flows are considered here.

Numerical Technique

The numerical solution of the equations of motion is obtained utilizing an explicit, time-dependent, predictor-corrector finite difference method developed by MacCormack[6]. In order to damp numerical oscillations induced by the gradients in the flow field associated with the developing jet (and wall) flows, a fourth-order damping scheme introduced by Holst[7] and modified by Berman[8] and Kuruvila[9] is utilized. This scheme involves certain free adjustable parameters, C_x and C_y, usually referred to as damping or dissipation coefficients. Generally, C_x, C_y should be such that $0 \leq C_x, C_y \leq 0.5$. General nomenclature for the computational domain is shown in Figure 2.

Computational Grid

A uniform rectangular grid is utilized in the upstream plenum section, extending a distance of 1.5 times the CP height into the CP portion of the domain. For the extended portion of the domain, which encompasses the outflow region, a scheme introduced by Cebeci and Smith is employed[10], whereby grid spacing is increased by a fixed percentage from an initial value. General nomenclature for the mesh generated in this manner is shown in Figure 3.

Initial and Boundary Conditions

At time t = 0 velocity components u and v are everywhere set to $u=v=0$, and pressure and

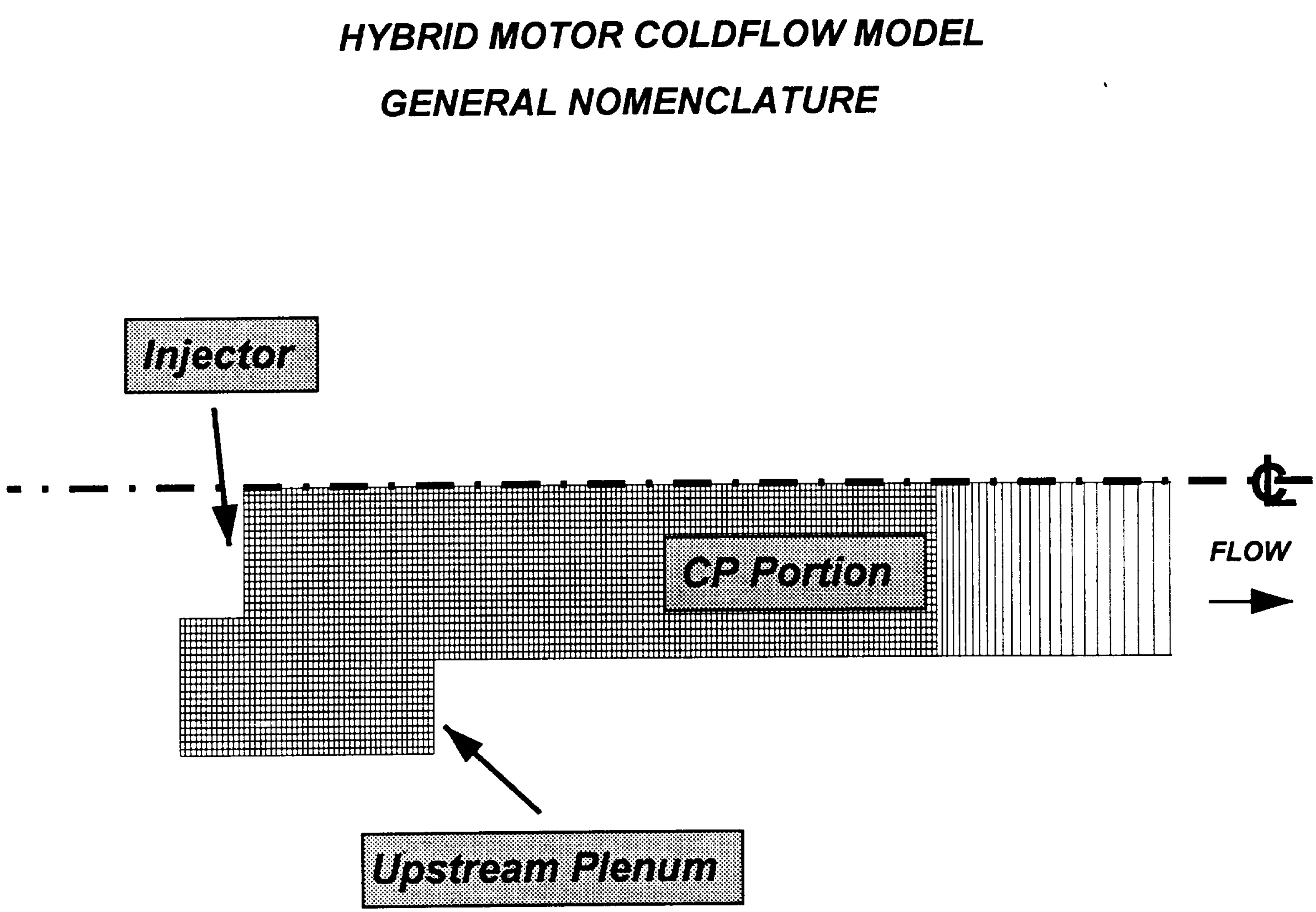

Figure 2: General Nomenclature, Computational Domain

HYBRID MOTOR COLDFLOW MODEL

GRID GENERATION NOMENCLATURE

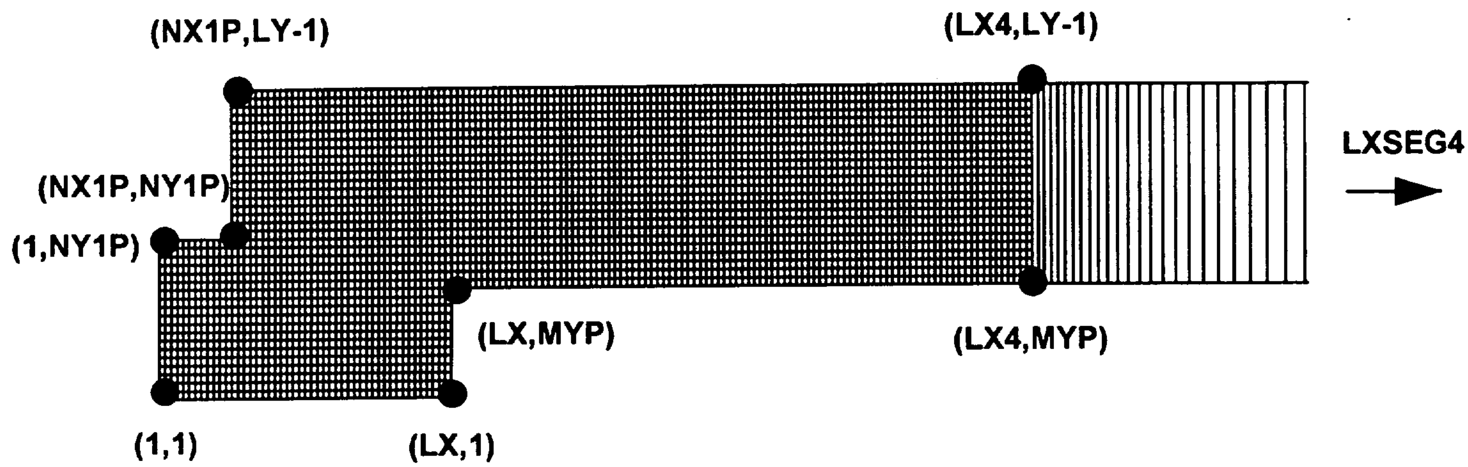

Figure 3: Nomenclature, Computational Grid

temperature are set to ambient values everywhere in the flow field. Density and internal energy are obtained from the equation of state. Boundary conditions must also be specified for all of the dependent variables, u, v, ρ, e, along with corresponding values of p and T. Along all solid walls of the computational domain a "no slip" condition is enforced, that is, $u=v=0$ for all time t. An adiabatic wall condition is imposed at all solid walls, as is the condition that the normal pressure gradient equals zero there. This latter condition is obtained from the momentum equation applied in the direction normal to the wall, as in a boundary layer type flow situation.

The top boundary of the computational domain is an axis of symmetry. Special note should be taken of the fact that, unless the outflow boundary is properly chosen, the downstream boundary conditions are not known. The approach adopted here is to continuously move the downstream boundary as the solution proceeds in time so that the initial disturbances (pressure waves) never reach it. Thus, it is possible to assume unperturbated conditions at the outflow boundary, requiring that both velocity components are zero, $u=v=0$ and that pressure, temperature and density are equal to their corresponding initial, ambient values.

The remaining boundary conditions are for the jet (oxidizer) flow and for the porous wall flow(s). Both the upstream plenum wall and the CP wall are porous, as in the experimental test rig. At time $t=0$, all jet and porous wall flow rates are zero. These flows are then ramped up linearly with time up to specified values, at which point they remain constant. Thus, the flow field variables at the inflow boundary do not initially remain constant with respect to time at the inflow boundary; this period is, however, very short as compared to the overall calculation interval.

Computer Code and Sample Results

The computer code (Appendix A) is written in Fortran-77 and is currently run on a Sun Systems Sparc 10 workstation. Simulation of the (incompressible) flow in the hybrid model test-bed was accomplished through Reynolds number scaling of the measured flowfield velocities to a comparable flow in a gas having the properties of air at "room" temperature. The resulting velocities were well within the incompressible flow regime (Mach number ≤ 0.2). All input units are SI, but provision is made for US units if the user so desires. Required input is documented by annotation within the code listing (Appendix A) and will not be discussed further here, except that geometrical input nomenclature for a typical (spacerless) configuration is shown in Figure 4; the corresponding calculation stations are shown in Figure 5. A sample listing of typical input for the case of a radial injector configuration with a spacer is shown in Appendix B.

Figure 6 illustrates a comparison of experimental results[11] with computed results for axial flow injection through a non-porous wall. Figure 7 provides a similar comparison (same reference) for a radial flow injection pattern. Figures 8 and 9 compare LDV data (from portion 1 of this investigation) and calculated data, respectively, for the case of axial flow injection with no spacer (inhibitor, in the actual motor). Figures 10 and 11 provide the same comparison for a radial injection pattern. Figures 12 - 15 repeat the comparisons for the case where a restrictive spacer is present between the upstream plenum and the CP portion of the model.

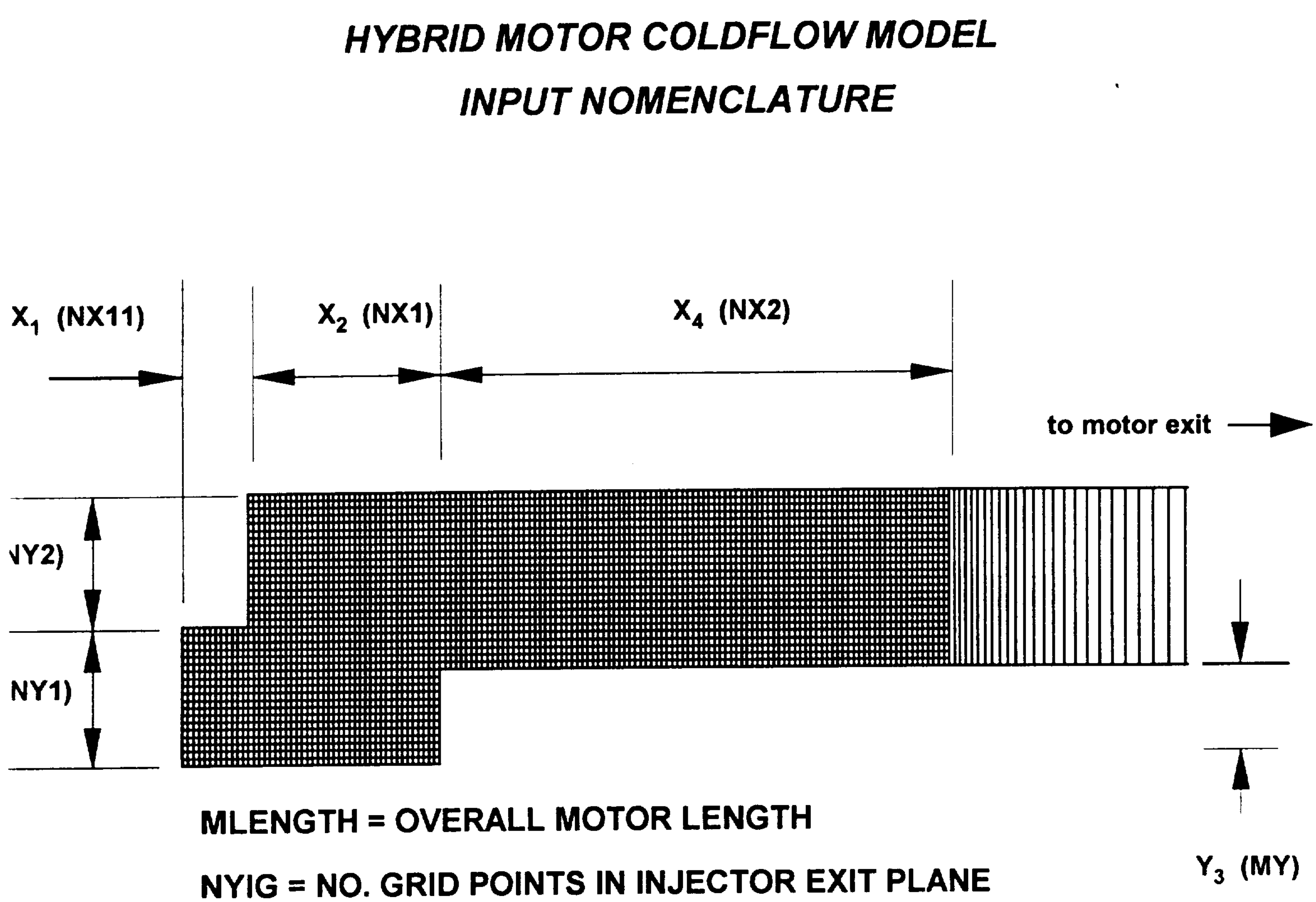

Figure 4: Computer Code (Geometrical) Input Nomenclature

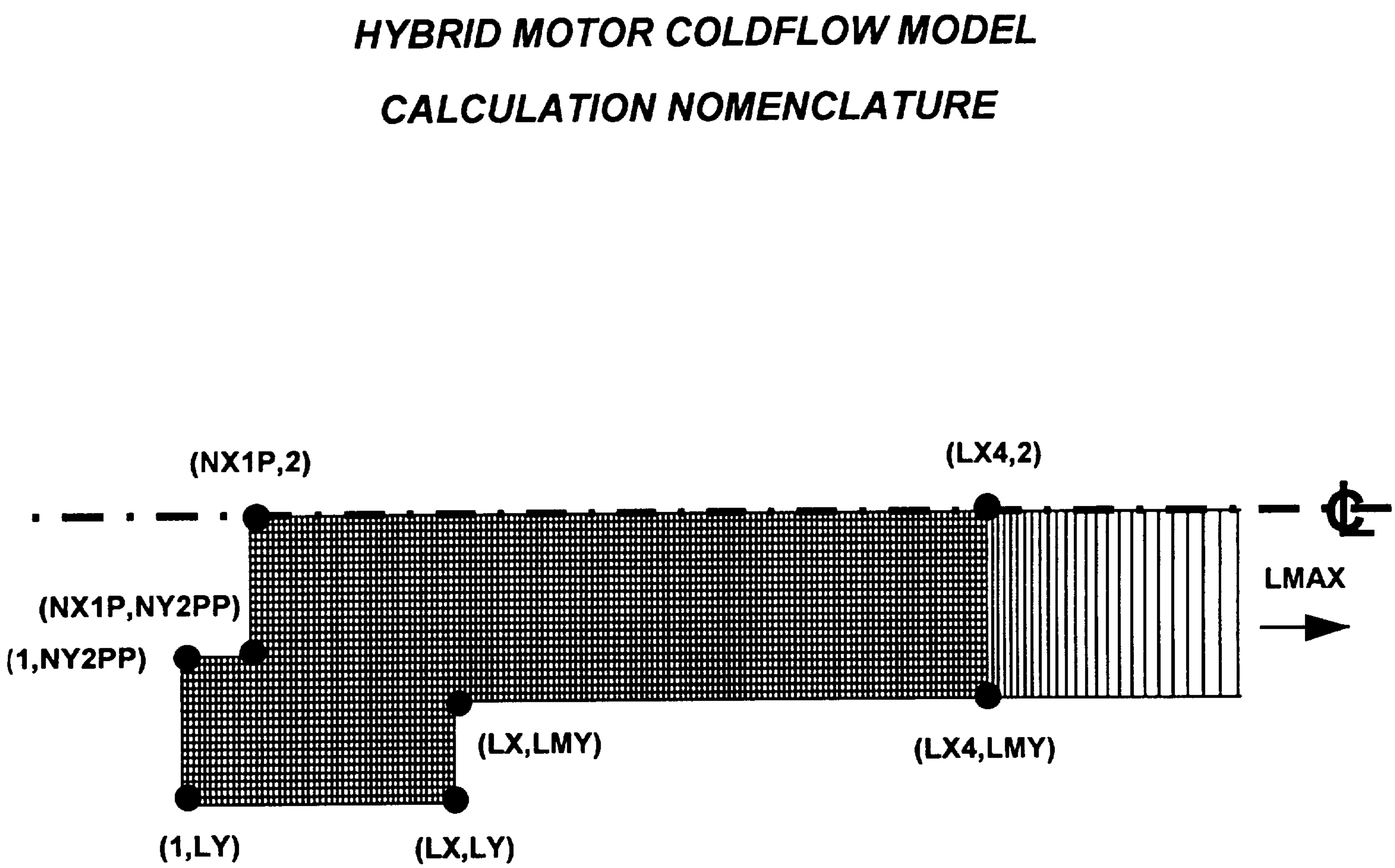

Figure 5: Computer Code (Geometrical) Calculation Nomenclature

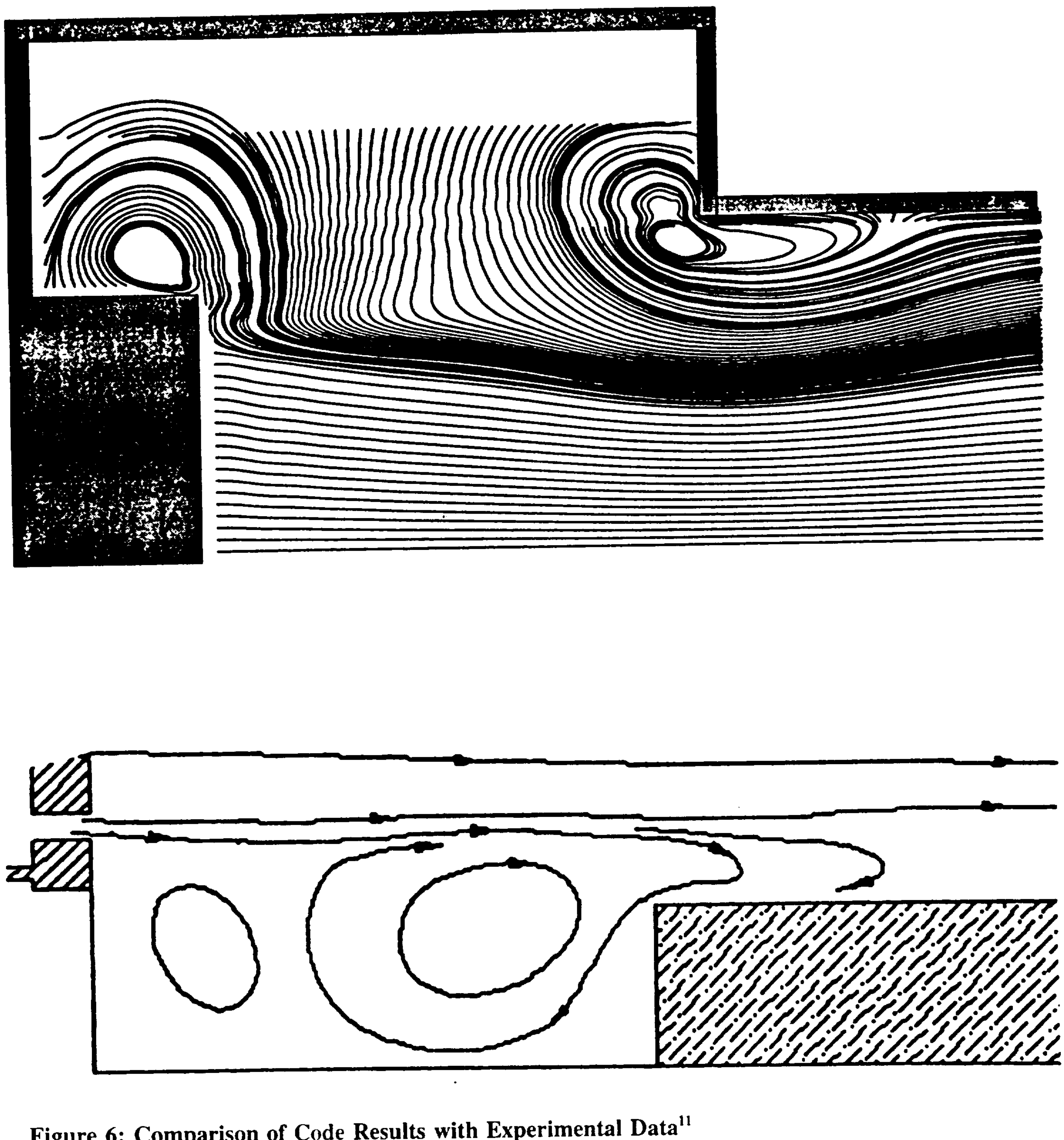

Figure 6: Comparison of Code Results with Experimental Data[11]
Axial Flow Injection (No Wall Blowing)

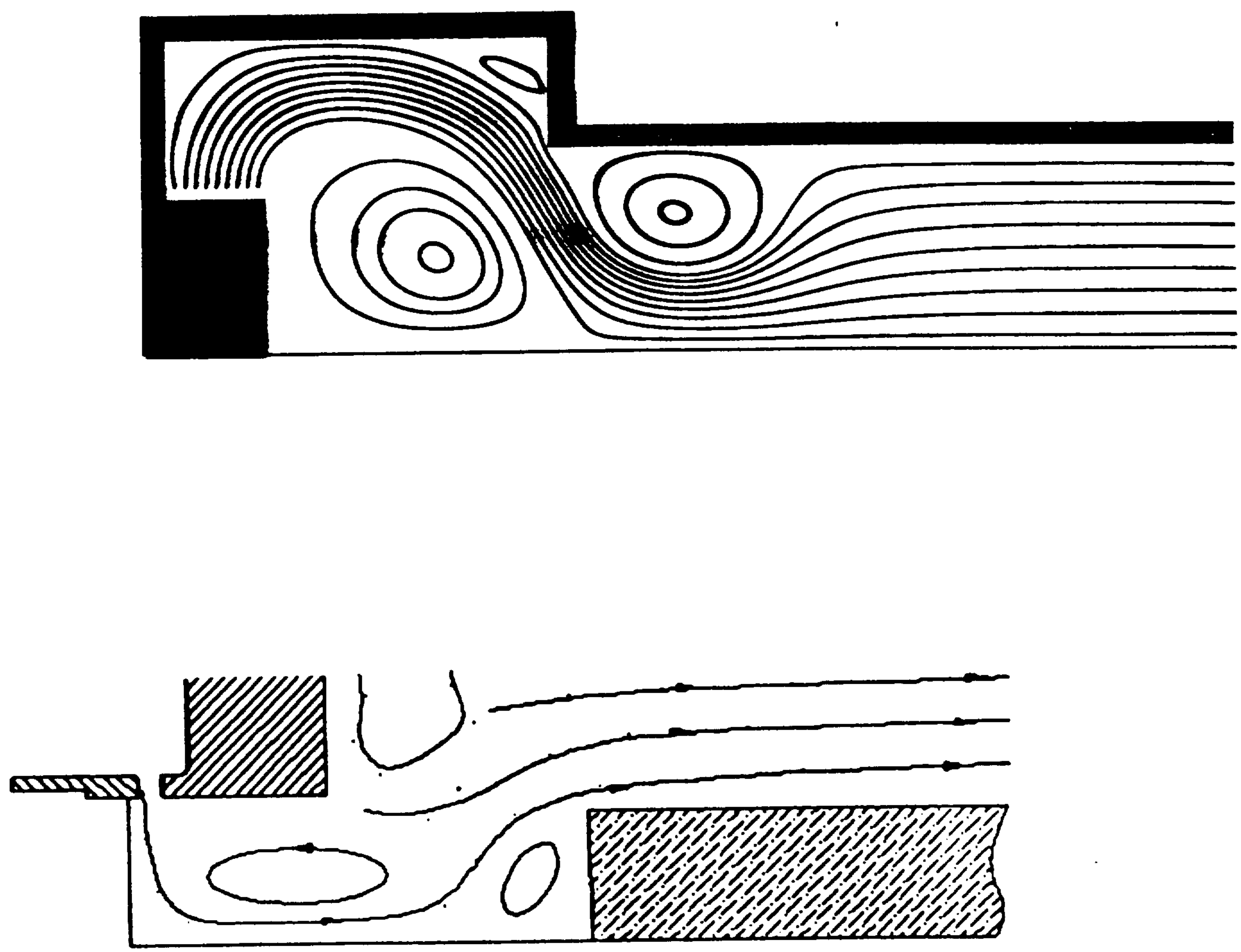

Figure 7: Comparison of Code Results with Experimental Data[11]
Radial Flow Injection (No Wall Blowing)

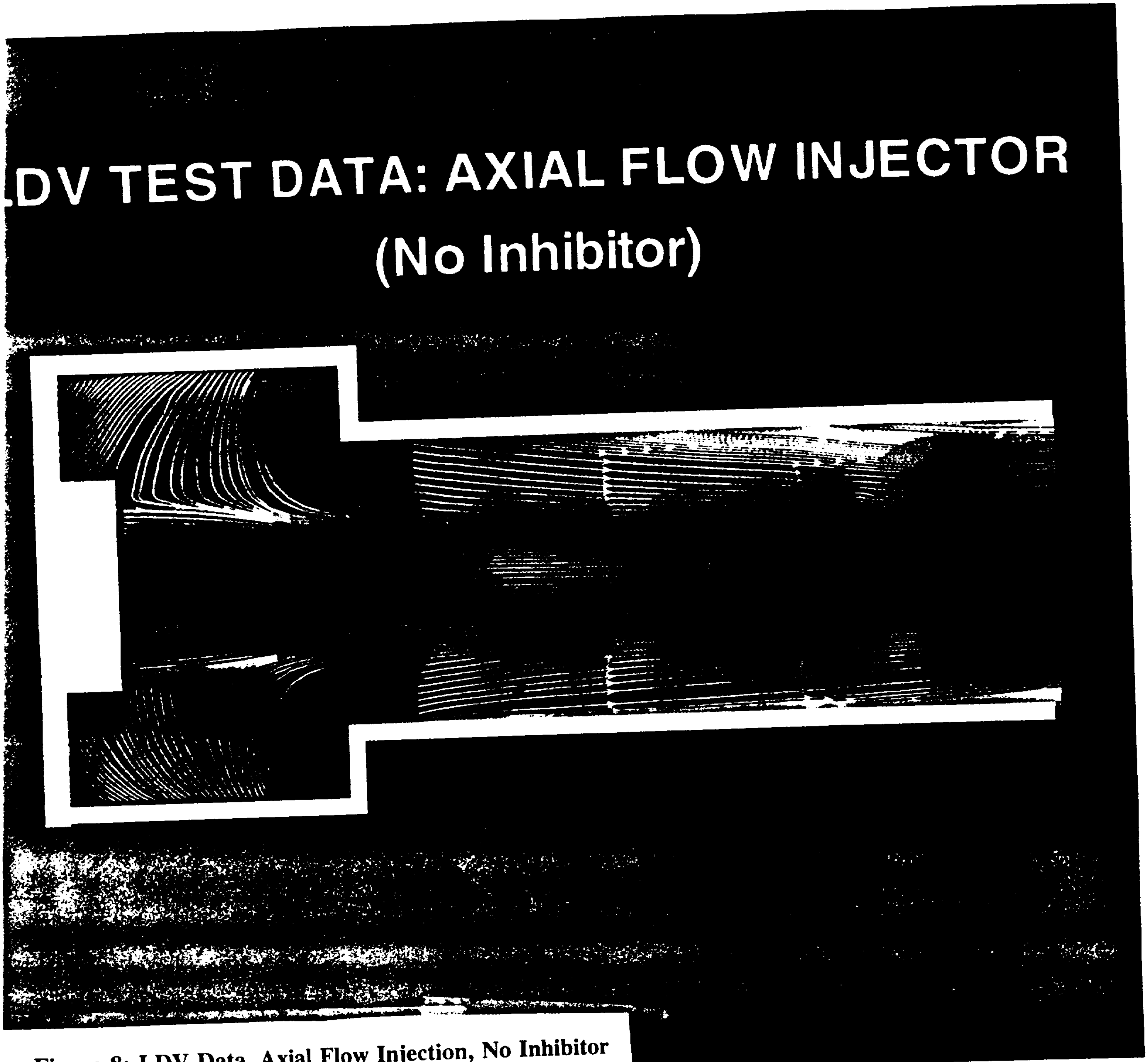

Figure 8: LDV Data, Axial Flow Injection, No Inhibitor

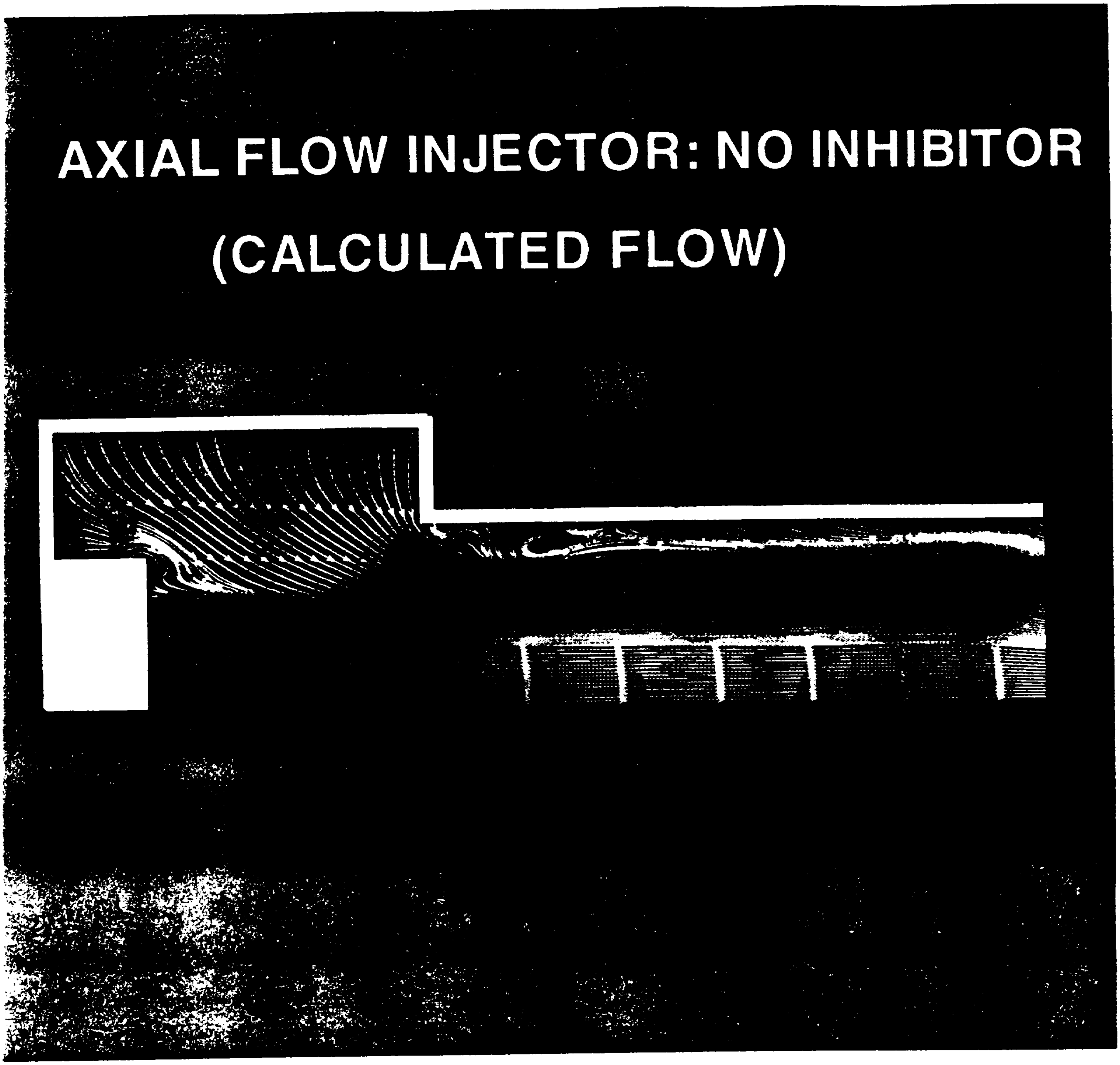

Figure 9: Calculated Data, Axial Flow Injection, No Inhibitor

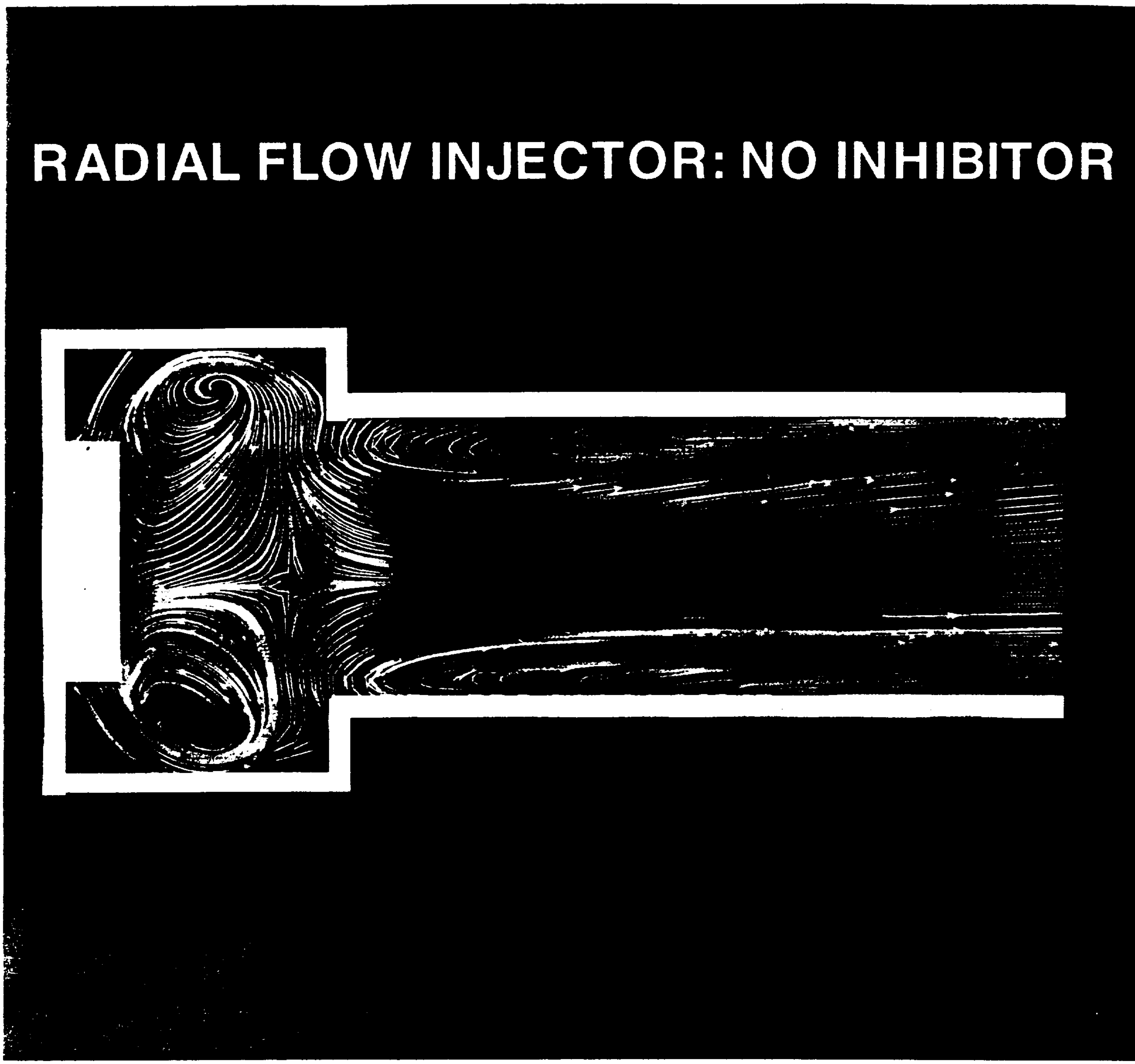

Figure 10: LDV Data, Radial Flow Injection, No Inhibitor

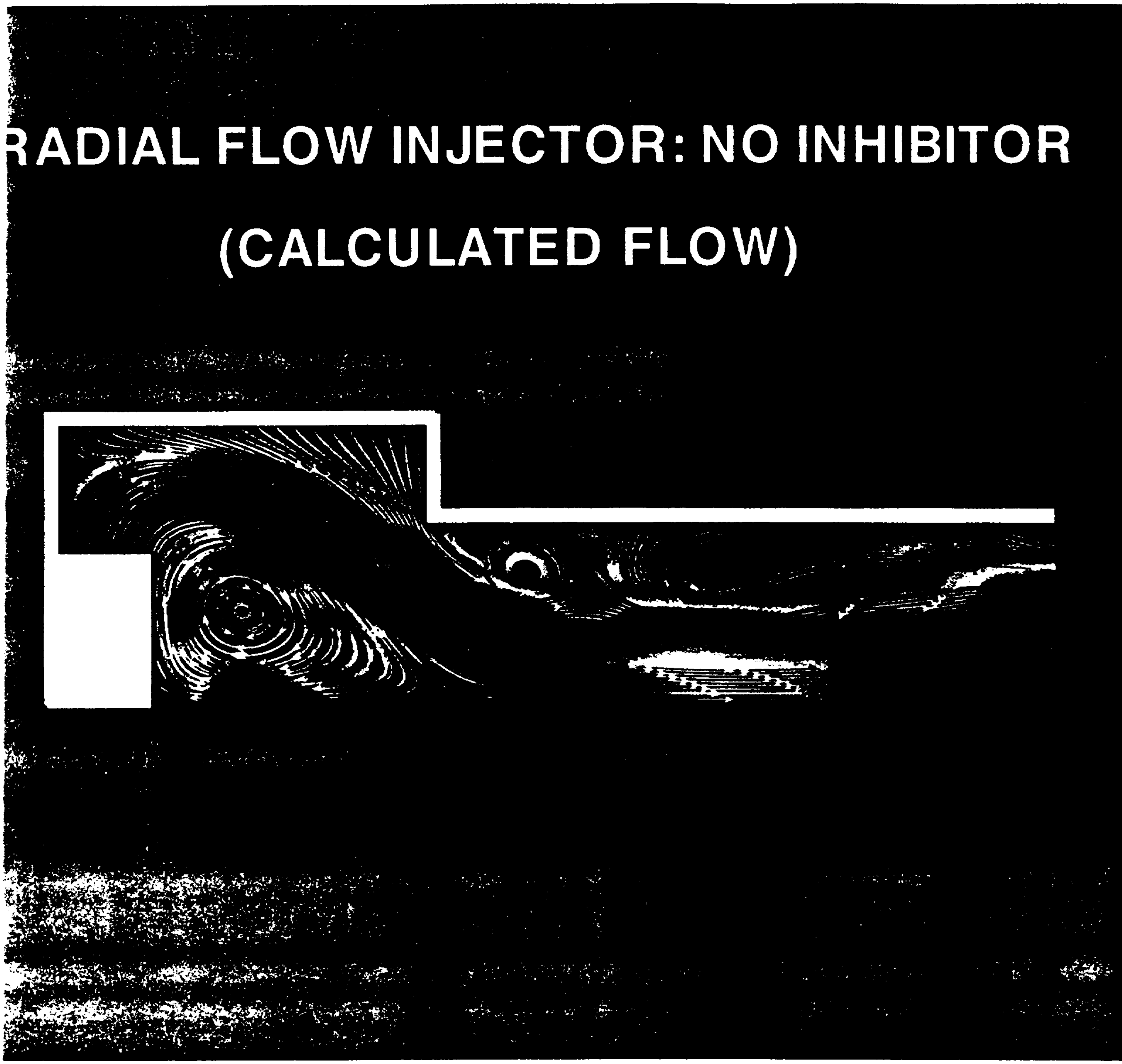

Figure 11: Calculated Data, Radial Flow Injection, No Inhibitor

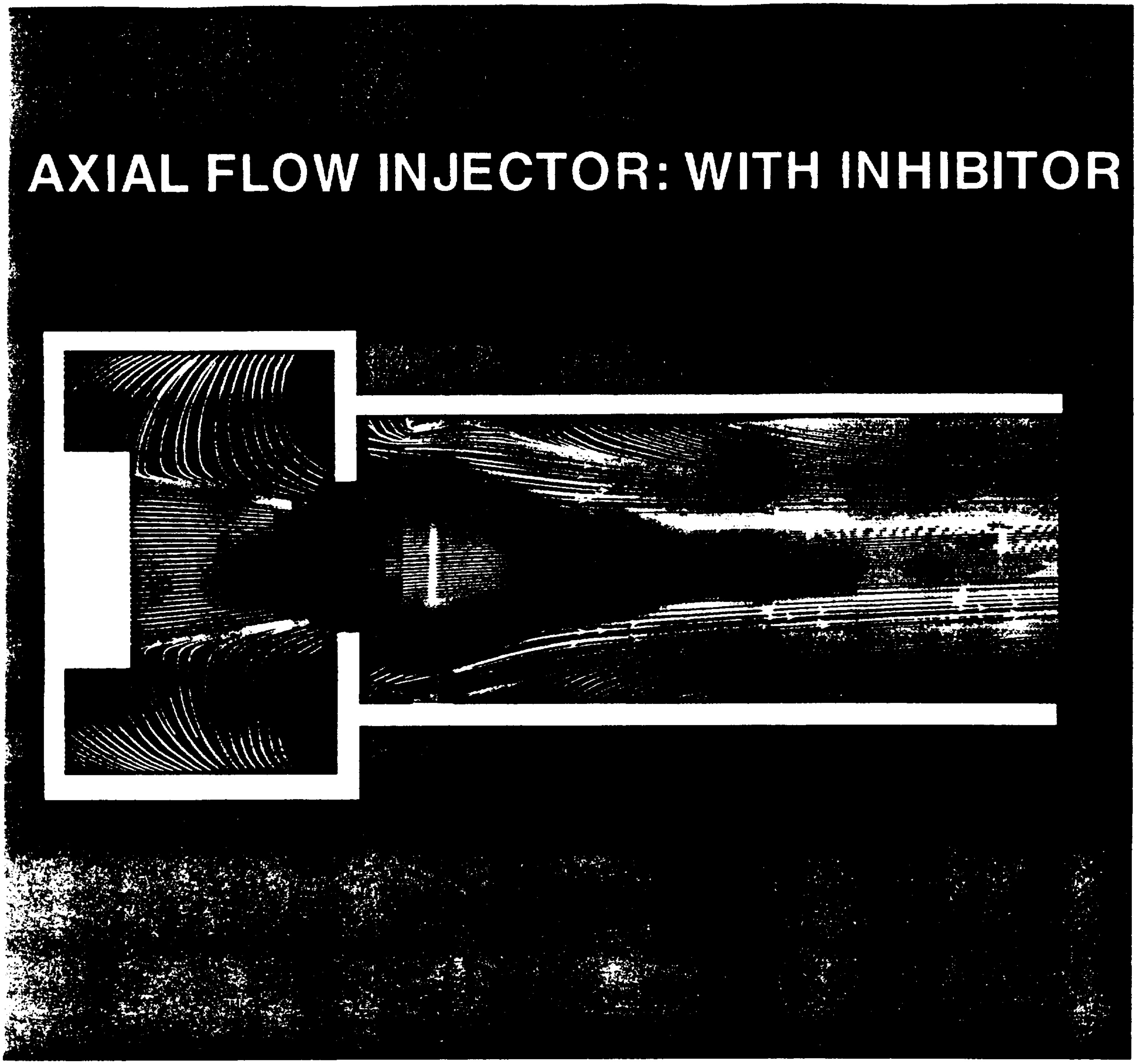

Figure 12: LDV Data, Axial Flow Injection, Inhibitor Present

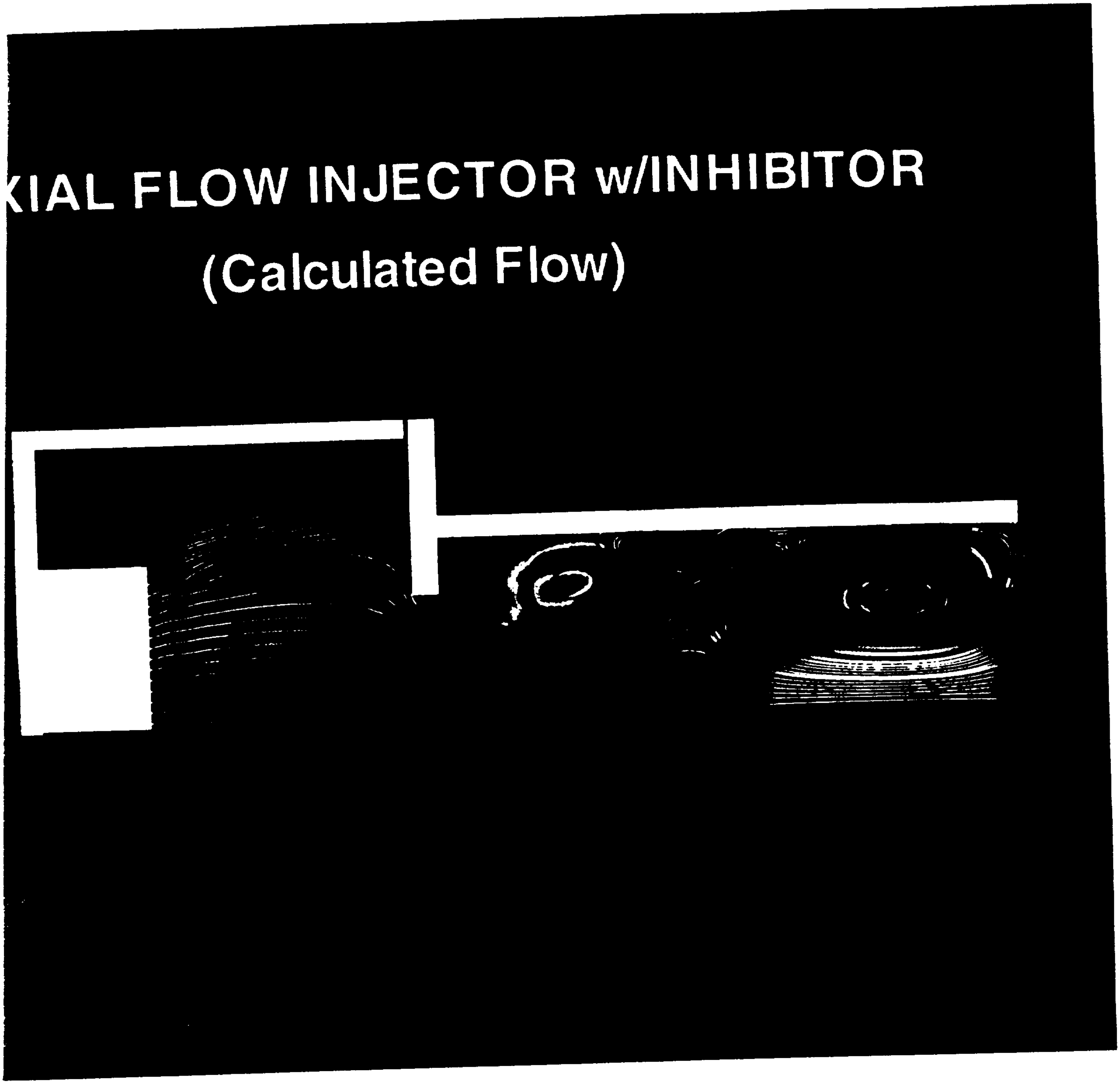

Figure 13: Calculated Data, Axial Flow Injection, Inhibitor Present

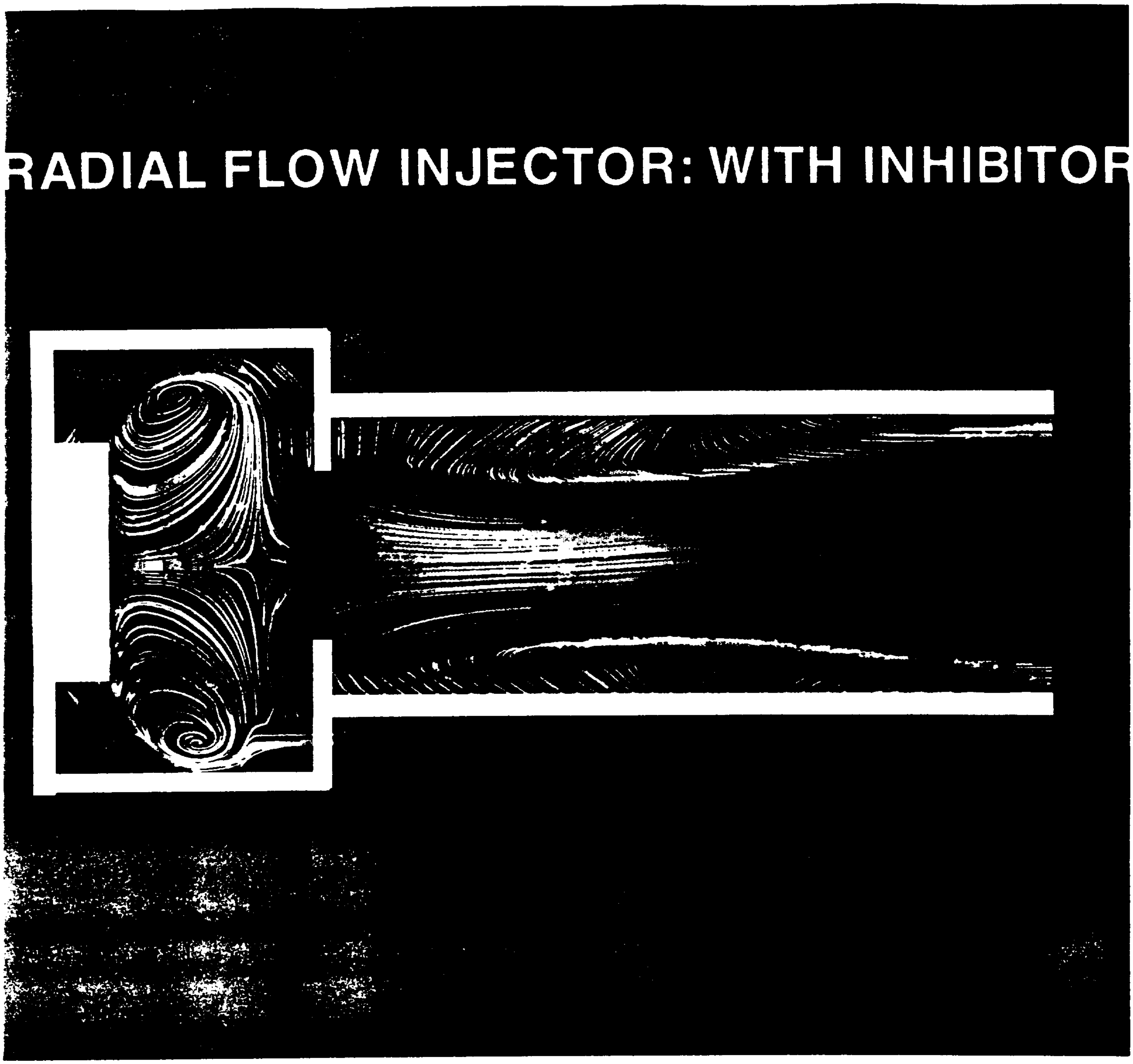

Figure 14: LDV Data, Radial Flow Injection, Inhibitor Present

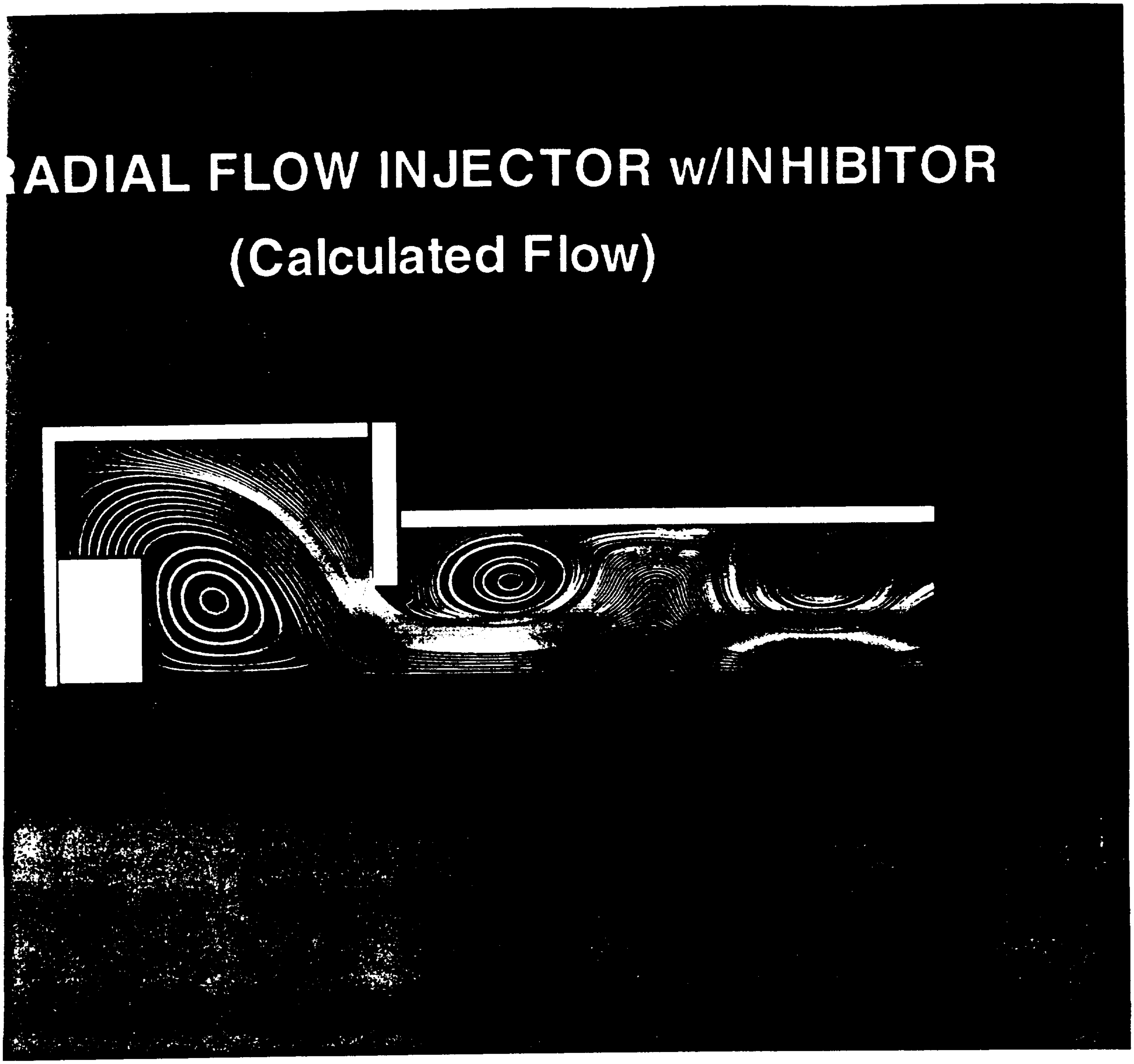

Figure 15: Calculated Data, Radial Flow Injection, Inhibitor Present

References

1. "Cold-Flow Study of Hybrid Rocket Motor Flow Dynamics," R. Jenkins, W. Cruit, A. Smith, AIAA 96-2843, presented at 32nd Joint Propulsion Conference, Orlando, FL, July 1996.

2. "Numerical Analysis of Unsteady Multiple Jet Plume Interactions," R. Jenkins, W. Foster, L. Wirth, *Journal of Applied Mathematics and Computation*, Vol.79, pp. 239-247, 1996.

3. "Analysis of Ignition and Flame Spreading in Solid Rocket Motor Star Slots," A. Ciucci, R. M. Jenkins, and W.A. Foster, Jr., accepted for publication in *AIAA Journal of Propulsion and Power.*

4. "Unsteady Jet Plume Interaction in Multi-Port SRM Ignition Systems," R.M. Jenkins, W.A. Foster, Jr., and L.S. Wirth, AIAA95-0874, presented at 33rd Aerospace Sciences Meeting and Exhibit, Reno, NV, January 9-12, 1995.

5. "Numerical Analysis of an Unsteady Confined Supersonic Jet," A. Ciucci, R.M. Jenkins, and W.A. Foster, Jr., *Journal of Applied Mathematics and Computation*, Vol. 57, pp. 245-254, 1993.

6. "The Effect of Viscosity in Hypervelocity Impact Cratering," R.W. MacCormack, AIAA Paper 69-354, 1969.

7. "Numerical Solution of Axisymmetric Boattail Fields with Plume Simulators," T.L. Holst, T. L., AIAA Paper 77-224, 1977.

8. "Supersonic Flow over a Rearward Facing Step with Transverse Nonreacting Hydrogen Injection," H.A. Berman and J.D. Anderson, *AIAA Journal*, Vol.21, No.12, Dec. 1983, pp.1707-1713.

9. "A Study of the Effects of Numerical Dissipation on the Calculation of Supersonic Separated Flows," K. Kuruvila and J.D. Anderson, AIAA Paper 85-0301, 1985.

10. <u>Analysis of Turbulent Boundary Layers</u>, T. Cebeci and A.M.O. Smith, Academic Press, New York, 1974.

11. "Visualization of Recirculation Zones in Hybrid Rocket Motors," J. Jones and R. Frederick, Jr., AIAA Paper 96-2842, 1996.

APPENDIX A

Computer Code Listing

```
C     PROGRAM TO CALCULATE FLOW DYNAMICS IN A HYBRID ROCKET MOTOR
C     VERSION 1: COLDFLOW ANALYSIS, AXIAL OR RADIAL INJECTOR W/INHIBITOR
C                as of 05/25/97

      REAL MU,K,LREF,MUREF,KREF,MW
      REAL LPR,MACHREF,MACH,MLENGTH
      REAL MUT,KT,MUEFF,KEFF,KET,KETB
      DIMENSION YR(100),ROLD(250,100)
      COMMON/D2/TIME(10000),DTM(10000)
      COMMON/GEOM1/X(250),Y(100)
      COMMON/R1/MU(250,100),K(250,100),BR(250,100),HC(250,100)
      COMMON/R2/MUT(250,100),KT(250,100)
      COMMON/PRED1/U(250,100),V(250,100),RHO(250,100),P(250,100),
     &T(250,100),E(250,100),KET(250,100),EPS(250,100),SRK(250,100)
      COMMON/PRED2/DRUDT(250,100),DRVDT(250,100),DRDT(250,100),
     &DREDT(250,100),DRKEDT(250,100),DREPDT(250,100)
      COMMON/COR1/UB(250,100),VB(250,100),RHOB(250,100),PB(250,100),
     &TB(250,100),EB(250,100),KETB(250,100),EPSB(250,100),
     &SRKB(250,100)
      COMMON/ALO/MACH(250,100)
      COMMON/S2/LPR,NPR
      COMMON/D3/IUNIT,ISTOP,COUR,CXD,CYD
      COMMON/D3BIS/MLENGTH
      COMMON/D4/GAMMA,R,MW,TIME2,TIME1
      COMMON/D5/PAMB,TAMB,FLOW1,FLOW2,FLOW3
      COMMON/D5BIS/TM,KCHK,KSHK,GM,EXP1,EXP2
      COMMON/D6/X1,X2,Y1,Y2,X4,Y3,B,NX11,NX1,NX2,NY1,NY2,MY
      COMMON/D6BIS/NY1P,NX1P,LX,LX4,LY,LMY,Y34,X12,X123
      COMMON/D6TRI/NXSTIP,NYSTIP,NY2PP,NYIG,MY4
      COMMON/STEPS/DX11,DX1,DX2,DY1,DY2,DY3
      COMMON/SEGM1/NXSEG4,LXSEG4,AKGR
      COMMON/D7/LREF,VREF,TREF,RHOREF,RREF,KREF,MUREF
      COMMON/ANEW1/CPREF,PRAREF,DHYD
      COMMON/FLOW/MAXCAL,PFAC
      COMMON/HYB/YEX
      COMMON/RESTART/NSAVE,KSAVE
      COMMON/INITIAL/UINJ0
      COMMON/CONFIG/SPACER,NFIG,NSPACE,ISPACE,JSPACE

      F2(UINJ0,FLOW,TIME1,TM)=UINJ0+(FLOW-UINJ0)*TM/TIME1

C     open main input file (maindat)

      OPEN(UNIT=1,FILE='maindat',STATUS='UNKNOWN')

C     write input data to file (dataout)

      OPEN(UNIT=2,FILE='dataout',STATUS='UNKNOWN')

      OPEN(UNIT=9,FILE='screen',STATUS='UNKNOWN')

C     open grid output file (gout)

      OPEN(UNIT=10,FILE='gout',STATUS='UNKNOWN')
      OPEN(UNIT=11,FILE='gosul',STATUS='UNKNOWN')

C     open velocity output files
```



```
      OPEN(UNIT=70,FILE='va01',STATUS='UNKNOWN')
      OPEN(UNIT=71,FILE='va02',STATUS='UNKNOWN')
      OPEN(UNIT=72,FILE='va03',STATUS='UNKNOWN')
      OPEN(UNIT=73,FILE='va04',STATUS='UNKNOWN')
      OPEN(UNIT=74,FILE='va05',STATUS='UNKNOWN')
      OPEN(UNIT=75,FILE='va06',STATUS='UNKNOWN')
      OPEN(UNIT=76,FILE='va07',STATUS='UNKNOWN')
      OPEN(UNIT=77,FILE='va08',STATUS='UNKNOWN')
      OPEN(UNIT=78,FILE='va09',STATUS='UNKNOWN')
      OPEN(UNIT=79,FILE='va10',STATUS='UNKNOWN')
      OPEN(UNIT=80,FILE='va11',STATUS='UNKNOWN')
      OPEN(UNIT=81,FILE='va12',STATUS='UNKNOWN')
      OPEN(UNIT=82,FILE='va13',STATUS='UNKNOWN')
      OPEN(UNIT=83,FILE='va14',STATUS='UNKNOWN')
      OPEN(UNIT=84,FILE='va15',STATUS='UNKNOWN')
      OPEN(UNIT=85,FILE='va16',STATUS='UNKNOWN')
      OPEN(UNIT=86,FILE='va17',STATUS='UNKNOWN')
      OPEN(UNIT=87,FILE='va18',STATUS='UNKNOWN')
      OPEN(UNIT=88,FILE='va19',STATUS='UNKNOWN')
      OPEN(UNIT=89,FILE='va20',STATUS='UNKNOWN')
      OPEN(UNIT=90,FILE='va21',STATUS='UNKNOWN')
      OPEN(UNIT=91,FILE='va22',STATUS='UNKNOWN')
      OPEN(UNIT=92,FILE='va23',STATUS='UNKNOWN')
      OPEN(UNIT=93,FILE='va24',STATUS='UNKNOWN')
      OPEN(UNIT=94,FILE='va25',STATUS='UNKNOWN')
      OPEN(UNIT=95,FILE='va26',STATUS='UNKNOWN')
      OPEN(UNIT=96,FILE='va27',STATUS='UNKNOWN')
      OPEN(UNIT=97,FILE='va28',STATUS='UNKNOWN')
      OPEN(UNIT=98,FILE='va29',STATUS='UNKNOWN')
      OPEN(UNIT=99,FILE='va30',STATUS='UNKNOWN')
      OPEN(UNIT=151,FILE='va31',STATUS='UNKNOWN')
      OPEN(UNIT=152,FILE='va32',STATUS='UNKNOWN')
      OPEN(UNIT=153,FILE='va33',STATUS='UNKNOWN')
      OPEN(UNIT=154,FILE='va34',STATUS='UNKNOWN')
      OPEN(UNIT=155,FILE='va35',STATUS='UNKNOWN')
      OPEN(UNIT=156,FILE='va36',STATUS='UNKNOWN')
      OPEN(UNIT=157,FILE='va37',STATUS='UNKNOWN')
      OPEN(UNIT=158,FILE='va38',STATUS='UNKNOWN')
      OPEN(UNIT=159,FILE='va39',STATUS='UNKNOWN')

C     open restart files

      OPEN(UNIT=12,FILE='readcal',STATUS='UNKNOWN')
      OPEN(UNIT=41,FILE='ritcall',STATUS='UNKNOWN')
      OPEN(UNIT=42,FILE='ritcal2',STATUS='UNKNOWN')


      OPEN(UNIT=50,FILE='crap',STATUS='UNKNOWN')

      OPEN(UNIT=51,FILE='crap2',STATUS='UNKNOWN')

      OPEN(UNIT=52,FILE='crap3',STATUS='UNKNOWN')

C     call data input/output subroutine

      ISTOP=0

      CALL RWDATA

      IF(ISTOP.EQ.1) GO TO 1
```

```
      GO TO 2
    1 WRITE(*,*) 'WRONG INPUT DATA FOR: IUNIT. EXECUTION ABORTED.'
      GO TO 999

    2 IF(NSAVE.EQ.1) THEN
       READ(12,*)NSTART
       WRITE(*,*)NSTART
      ELSE
       NSTART=1
      ENDIF

      UINJ0=0.0
      PMAX=PFAC*PAMB
      ASSREF=SQRT(GAMMA*RREF*TREF)
      MACHREF=VREF/ASSREF
      REYREF=RHOREF*VREF*LREF/MUREF
      CPREF=GAMMA*RREF/(GAMMA-1.)
      PRAREF=MUREF*CPREF/KREF
      SMU=110.5
      SIGT=0.91
      CMUREF=0.09
      C1=1.44
      C2REF=1.92
      SIGK=1.0
      SIGE=1.3
      EXP2=(GAMMA+1.)/(2.*(GAMMA-1.))
      WRITE(2,*)
      WRITE(2,*)
      WRITE(2,6)
      WRITE(2,7)
      WRITE(2,8) MACHREF,REYREF,PRAREF,CPREF
    6 FORMAT(1X,'DIMENSIONLESS REFERENCE PARAMETERS',/)
    7 FORMAT(2X,'MACH NUM',3X,'REYN NUM',3X,'PRAN NUM',6X,'Cp',/)
    8 FORMAT(2X,F7.4,2X,E11.4,3X,F7.4,4X,F8.3,///)
      WRITE(2,9)
      WRITE(2,19) SIGT,CMUREF,C1,C2REF,SIGK,SIGE
    9 FORMAT(2X,'SIGMAT',5X,'CMU',7X,'C1',6X,'C2',6X,'SIGMAK',3X,
     &'SIGMAEPS',/)
   19 FORMAT(2X,6(F6.3,3X),///)

C     call subroutine to determine the required number of axial grid
C     points in CP portion of motor

      CALL CPLEN(X1,X2,X4,MLENGTH,NX11,AKGR,NXSEG4)

C     call grid generation subroutine

      CALL GRID

      MYP=MY+1
      DO 10 J = 2,LY
      JI=LY-(J-1)
      YR(J)=Y1+Y2-Y(JI)
   10 CONTINUE
      YR(1)=-DY2

C     non-dimensionalize coordinates

      DO 11 J = 1,LY
      Y(J)=YR(J)/LREF
```



```
   11 CONTINUE
      DO 12 I = 1,LXSEG4
      X(I)=X(I)/LREF
   12 CONTINUE
      DX11=DX11/LREF
      DX1=DX1/LREF
      DX2=DX2/LREF
      DY1=DY1/LREF
      DY2=DY2/LREF
      DY3=DY3/LREF
      LMY=LY-MY
      NY2PP=NY2+2

C     call subroutine to initialize variables

C     NOTE: LX ...... number of x-grid points in upstream plenum
C           LX4 ..... total no. of x-grid points in headend section
C                     (includes part of CP)
C           LMAX .... total number of x-grid points in motor

      LMAX=LXSEG4
      CALL INITDAT(LMAX,NSTART)

C     set wall boundary conditions

      CALL VELBC(LMAX)
      CALL DWBC(LMAX)

C     open time iteration loop

      IF(NSTART.EQ.1) THEN
       TIME(NSTART)=0.
       DTM(NSTART)=0.0
       N=1
       N1=2
       NTIME=0
      ELSE
       N=NSTART
       N1=NSTART+1
       NTIME=NSTART-1
       READ(12,703)TIME(NSTART),DTM(NSTART)
      ENDIF

      GMA=GAMMA*MACHREF**2
      RGMA=1./GMA
      RREYN=1./REYREF
      R43=4./3.
      R23=2./3.
      R13=1./3.
      RPRREMA=1./(PRAREF*REYREF*MACHREF**2)/(GAMMA-1.)
      LX4M=LX4-1
      LMXM=LMAX-1
      LYM=LY-1

C     begin time iteration

      DO 1111 N = N1,MAXCAL
      NTIME=NTIME+1

C     calculate time step size
```

```
      DT=10.
      DO 15 I = 2,LMXM
      DO 16 J = 2,LYM
      IF((I.LE.NX1P).AND.(J.LE.NY2PP)) GO TO 16
      IF((I.GE.LX).AND.(J.GE.LMY)) GO TO 16
      A=SQRT(GAMMA*R*T(I,J)*TREF)
      A=A/VREF
      DX=DX11
      IF(I.GE.LX4) DX=X(I+1)-X(I)
      DY=DY2
      TEMX=DX/(ABS(U(I,J))+A)
      TEMY=DY/(ABS(V(I,J))+A)
      IF(TEMX.LT.DT) DT=TEMX
      IF(TEMY.LT.DT) DT=TEMY
   16 CONTINUE
   15 CONTINUE
      DT=DT*COUR
      DTM(N)=DT*LREF/VREF
      NM1=N-1
      TIME(N)=TIME(NM1)+DTM(N)
      WRITE(*,*) 'NTIME =',NTIME,'     TSTEP=',DTM(N),
     &'     TIME=',TIME(N)
      IF(TIME(N).GT.TIME2) GO TO 886
      IF(DTM(N).LT.1.E-9) GO TO 999
      TM=TIME(N)

C     set boundary conditions at injector exit

      CV=R/(GAMMA-1.)
      YEX=Y(NYIG)
      NXIGN=3+NYIG
      IF(NFIG.NE.0) GOTO 203

C     axial injector configuration

      DO 30 J = 2,NYIG
      IF(N.EQ.N1.AND.N1.NE.2) THEN
       READ(12,700)U(NX1P,J),V(NX1P,J),RHO(NX1P,J),P(NX1P,J),
     & T(NX1P,J),E(NX1P,J)
       READ(12,701)KET(NX1P,J),EPS(NX1P,J),SRK(NX1P,J)
       READ(12,700)UB(NX1P,J),VB(NX1P,J),RHOB(NX1P,J),PB(NX1P,J),
     & TB(NX1P,J),EB(NX1P,J)
       READ(12,701)KETB(NX1P,J),EPSB(NX1P,J),SRKB(NX1P,J)
       READ(12,702)MUT(NX1P,J),KT(NX1P,J),MU(NX1P,J),K(NX1P,J)

      ELSE
       IF(TM.GE.TIME1) GO TO 21
       GO TO 22
   21  UINJ=FLOW1
       GO TO 23
   22  UINJ=F2(UINJ0,FLOW1,TIME1,TM)
   23  U(NX1P,J)=UINJ/3.28/VREF
       V(NX1P,J)=0.0
       T(NX1P,J)=TAMB/TREF
       P(NX1P,J)=P(NX1P+1,J)
       RHO(NX1P,J)=P(NX1P,J)*(RHOREF*VREF**2)/(R*TAMB)/RHOREF
       E(NX1P,J)=CV*T(NX1P,J)*TREF/VREF**2
       SRK(NX1P,J)=SQRT(KET(NX1P,J))
       UB(NX1P,J)=UINJ/3.28/VREF
```



```
       VB(NX1P,J)=0.0
       TB(NX1P,J)=TAMB/TREF
       PB(NX1P,J)=PB(NX1P+1,J)
       RHOB(NX1P,J)=PB(NX1P,J)*(RHOREF*VREF**2)/(R*TAMB)/RHOREF
       EB(NX1P,J)=CV*TB(NX1P,J)*TREF/VREF**2
       KETB(NX1P,J)=KET(NX1P,J)
       EPSB(NX1P,J)=EPS(NX1P,J)
       SRKB(NX1P,J)=SRK(NX1P,J)
      ENDIF
   30 CONTINUE
      GOTO 204

C     radial injector configuration

  203 CONTINUE

      DO 231 I = 2,NX1P
      IF(I.LE.2.OR.I.GE.NXIGN) GOTO 231
      IF(TM.GE.TIME1) GO TO 221
      GO TO 222
  221 UINJ=FLOW1
      GO TO 223
  222 UINJ=F2(UINJ0,FLOW1,TIME1,TM)
  223 U(I,NY2PP)=0.0
      V(I,NY2PP)=UINJ/3.28/VREF
       T(I,NY2PP)=TAMB/TREF
       P(I,NY2PP)=P(I,NY2PP+1)
       RHO(I,NY2PP)=P(I,NY2PP)*(RHOREF*VREF**2)/(R*TAMB)/RHOREF
       E(I,NY2PP)=CV*T(I,NY2PP)*TREF/VREF**2
       SRK(I,NY2PP)=SQRT(KET(I,NY2PP))
       UB(I,NY2PP)=0.0
       VB(I,NY2PP)=UINJ/3.28/VREF
       TB(I,NY2PP)=TAMB/TREF
       PB(I,NY2PP)=PB(I,NY2PP+1)
       RHOB(I,NY2PP)=PB(I,NY2PP)*(RHOREF*VREF**2)/(R*TAMB)/RHOREF
       EB(I,NY2PP)=CV*TB(I,NY2PP)*TREF/VREF**2
       KETB(I,NY2PP)=KET(I,NY2PP)
       EPSB(I,NY2PP)=EPS(I,NY2PP)
       SRKB(I,NY2PP)=SRK(I,NY2PP)
  231 CONTINUE

C     set boundary conditions along upstream plenum bottom

  204 CONTINUE
      DO 31 I = 2,LX-1

      IF(N.EQ.N1.AND.N1.NE.2) THEN
       READ(12,700)U(I,LY),V(I,LY),RHO(I,LY),P(I,LY),T(I,LY),E(I,LY)
       READ(12,701)KET(I,LY),EPS(I,LY),SRK(I,LY)
       READ(12,700)UB(I,LY),VB(I,LY),RHOB(I,LY),PB(I,LY),TB(I,LY),
     & EB(I,LY)
       READ(12,701)KETB(I,LY),EPSB(I,LY),SRKB(I,LY)
       READ(12,702)MUT(I,LY),KT(I,LY),MU(I,LY),K(I,LY)

      ELSE
       U(I,LY)=0.
       IF(TM.GE.TIME1) GO TO 24
       GO TO 25
   24  VPLEN=FLOW2
       GO TO 26
```

```
  25  VPLEN=F2(UINJ0,FLOW2,TIME1,TM)
  26  V(I,LY)=-VPLEN/3.28/VREF
      T(I,LY)=TAMB/TREF
      P(I,LY)=P(I,LY-1)
      RHO(I,LY)=P(I,LY)*(RHOREF*VREF**2)/(R*TAMB)/RHOREF
      E(I,LY)=CV*T(I,LY)*TREF/VREF**2
      KET(I,LY)=0.
      EPS(I,LY)=0.
      SRK(I,LY)=SQRT(KET(I,LY))
      UB(I,LY)=0.
      VB(I,LY)=-VPLEN/3.28/VREF
      TB(I,LY)=TAMB/TREF
      PB(I,LY)=PB(I,LY-1)
      RHOB(I,LY)=PB(I,LY)*(RHOREF*VREF**2)/(R*TAMB)/RHOREF
      EB(I,LY)=CV*TB(I,LY)*TREF/VREF**2
      KETB(I,LY)=KET(I,LY)
      EPSB(I,LY)=EPS(I,LY)
      SRKB(I,LY)=SRK(I,LY)
     ENDIF
  31 CONTINUE

C     set boundary conditions along CP bottom

     DO 32 I = ISPACE+1,LX4
     IF(N.EQ.N1.AND.N1.NE.2) THEN
      READ(12,700)U(I,LMY),V(I,LMY),RHO(I,LMY),P(I,LMY),T(I,LMY),
    & E(I,LMY)
      READ(12,701)KET(I,LMY),EPS(I,LMY),SRK(I,LMY)
      READ(12,700)UB(I,LMY),VB(I,LMY),RHOB(I,LMY),PB(I,LMY),TB(I,LMY),
    & EB(I,LMY)
      READ(12,701)KETB(I,LMY),EPSB(I,LMY),SRKB(I,LMY)
      READ(12,702)MUT(I,LMY),KT(I,LMY),MU(I,LMY),K(I,LMY)
     ELSE
      U(I,LMY)=0.
      IF(TM.GE.TIME1) GO TO 27
      GO TO 28
  27  VCP=FLOW3
      GO TO 29
  28  VCP=F2(UINJ0,FLOW3,TIME1,TM)
  29  V(I,LMY)=-VCP/3.28/VREF
      T(I,LMY)=TAMB/TREF
      P(I,LMY)=P(I,LMY-1)
      RHO(I,LMY)=P(I,LMY)*(RHOREF*VREF**2)/(R*TAMB)/RHOREF
      E(I,LMY)=CV*T(I,LMY)*TREF/VREF**2
      KET(I,LMY)=0.
      EPS(I,LMY)=0.
      SRK(I,LMY)=SQRT(KET(I,LMY))
      UB(I,LMY)=0.
      VB(I,LMY)=-VCP/3.28/VREF
      TB(I,LMY)=TAMB/TREF
      PB(I,LMY)=PB(I,LMY-1)
      RHOB(I,LMY)=PB(I,LMY)*(RHOREF*VREF**2)/(R*TAMB)/RHOREF
      EB(I,LMY)=CV*TB(I,LMY)*TREF/VREF**2
      KETB(I,LMY)=KET(I,LMY)
      EPSB(I,LMY)=EPS(I,LMY)
      SRKB(I,LMY)=SRK(I,LMY)
     ENDIF
  32 CONTINUE

C     set other boundary conditions
```



```
      DO 50 I = 1,LMAX
      DO 51 J = 2,LY
      IF((I.LT.NX1P).AND.(J.LT.NY2PP)) GO TO 51
      IF((I.GT.LX).AND.(J.GT.LMY)) GO TO 51
      MU(I,J)=(T(I,J)**1.5)*((1.+SMU/TREF)/(T(I,J)+SMU/TREF))
      K(I,J)=MU(I,J)
   51 CONTINUE
   50 CONTINUE
      DO 53 I = 1,LMAX
      DO 54 J = 2,LY
      IF((I.LT.NX1P).AND.(J.LT.NY2PP)) GO TO 54
      IF((I.GT.LX).AND.(J.GT.LMY)) GO TO 54
      REYT=RHO(I,J)*KET(I,J)**2/(MU(I,J)*EPS(I,J))*REYREF**2
      CMU=CMUREF*EXP(-2.5/(1.+REYT/50.))
      MUT(I,J)=CMU*RHO(I,J)*KET(I,J)**2/EPS(I,J)*REYREF**2
      KT(I,J)=MUT(I,J)*PRAREF/SIGT
      MUT(I,J)=0.
      KT(I,J)=0.
   54 CONTINUE
   53 CONTINUE
      DO 56 J = 2,NYIG
      REYT=RHO(NX1P,J)*KET(NX1P,J)**2/(MU(NX1P,J)*EPS(NX1P,J))*
     &REYREF**2
      CMU=CMUREF*EXP(-2.5/(1.+REYT/50.))
      MUT(NX1P,J)=0.
      KT(NX1P,J)=0.
   56 CONTINUE
      DO 57 I = 2,LX-1
      REYT=RHO(I,LY)*KET(I,LY)**2/(MU(I,LY)*EPS(I,LY))*
     &REYREF**2
      CMU=CMUREF*EXP(-2.5/(1.+REYT/50.))
      MUT(I,LY)=0.
      KT(I,LY)=0.
   57 CONTINUE

C     set up CL symmetry condition

      DO 60 I = NX1P,LMAX
      U(I,1)=U(I,3)
      V(I,1)=-V(I,3)
      V(I,2)=0.
      RHO(I,1)=RHO(I,3)
      P(I,1)=P(I,3)
      T(I,1)=T(I,3)
      E(I,1)=E(I,3)
      KET(I,1)=KET(I,3)
      EPS(I,1)=EPS(I,3)
      SRK(I,1)=SRK(I,3)
      MU(I,1)=MU(I,3)
      K(I,1)=K(I,3)
      MUT(I,1)=MUT(I,3)
      KT(I,1)=KT(I,3)
   60 CONTINUE

C     PREDICTOR STEP

      DO 300 I = 2,LMXM
      DX=DX11
      IF(I.GE.LX4) DX=X(I+1)-X(I)
```

```
      DO 310 J = 2,LYM
      IF((I.LE.NX1P).AND.(J.LE.NY2PP)) GO TO 310
      IF((I.GE.LX).AND.(J.GE.LMY)) GO TO 310
      IF(I.GE.LX.AND.I.LE.ISPACE) THEN
       IF(J.GE.(LY-JSPACE+1)) GOTO 310
      ELSE
       CONTINUE
      ENDIF
      DY=DY2
C     set up derivatives

      DRUDX=(RHO(I+1,J)*U(I+1,J)-RHO(I,J)*U(I,J))/DX
      DRVDY=(RHO(I,J+1)*V(I,J+1)-RHO(I,J)*V(I,J))/DY
      DRUUDX=(RHO(I+1,J)*U(I+1,J)**2-RHO(I,J)*U(I,J)**2)/DX
      DRUVDY=(RHO(I,J+1)*U(I,J+1)*V(I,J+1)-RHO(I,J)*U(I,J)*V(I,J))/DY
      DPDX=(RHO(I+1,J)*T(I+1,J)-RHO(I,J)*T(I,J))/DX
      DMUDX=(MU(I+1,J)-MU(I,J))/DX
      DUDX=(U(I+1,J)-U(I,J))/DX
      DVDY=(V(I,J+1)-V(I,J))/DY
      DMUDY=(MU(I,J+1)-MU(I,J))/DY
      DUDY=(U(I,J+1)-U(I,J))/DY
      DVDX=(V(I+1,J)-V(I,J))/DX
      DRUVDX=(RHO(I+1,J)*U(I+1,J)*V(I+1,J)-RHO(I,J)*U(I,J)*V(I,J))/DX
      DRVVDY=(RHO(I,J+1)*V(I,J+1)**2-RHO(I,J)*V(I,J)**2)/DY
      DPDY=(RHO(I,J+1)*T(I,J+1)-RHO(I,J)*T(I,J))/DY
      DREUDX=(RHO(I+1,J)*E(I+1,J)*U(I+1,J)-RHO(I,J)*E(I,J)*U(I,J))/DX
      DREVDY=(RHO(I,J+1)*E(I,J+1)*V(I,J+1)-RHO(I,J)*E(I,J)*V(I,J))/DY
      DKDX=(K(I+1,J)-K(I,J))/DX
      DTDX=(T(I+1,J)-T(I,J))/DX
      DKDY=(K(I,J+1)-K(I,J))/DY
      DTDY=(T(I,J+1)-T(I,J))/DY
      DRUKEDX=(RHO(I+1,J)*U(I+1,J)*KET(I+1,J)-RHO(I,J)*U(I,J)*KET(I,J))
     &/DX
      DRVKEDY=(RHO(I,J+1)*V(I,J+1)*KET(I,J+1)-RHO(I,J)*V(I,J)*KET(I,J))
     &/DY
      DKEDX=(KET(I+1,J)-KET(I,J))/DX
      DKEDY=(KET(I,J+1)-KET(I,J))/DY
      DMUTDX=(MUT(I+1,J)-MUT(I,J))/DX
      DMUTDY=(MUT(I,J+1)-MUT(I,J))/DY
      DSRKDX=(SRK(I+1,J)-SRK(I,J))/DX
      DSRKDY=(SRK(I,J+1)-SRK(I,J))/DY
      DRUEPDX=(RHO(I+1,J)*U(I+1,J)*EPS(I+1,J)-RHO(I,J)*U(I,J)*EPS(I,J))
     &/DX
      DRVEPDY=(RHO(I,J+1)*V(I,J+1)*EPS(I,J+1)-RHO(I,J)*V(I,J)*EPS(I,J))
     &/DY
      DEPDX=(EPS(I+1,J)-EPS(I,J))/DX
      DEPDY=(EPS(I,J+1)-EPS(I,J))/DY
      DKTDX=(KT(I+1,J)-KT(I,J))/DX
      DKTDY=(KT(I,J+1)-KT(I,J))/DY
      MUEFF=MU(I,J)+MUT(I,J)
      KEFF=K(I,J)+KT(I,J)
      DMEFDX=DMUDX+DMUTDX
      DMEFDY=DMUDY+DMUTDY
      DKEFDX=DKDX+DKTDX
      DKEFDY=DKDY+DKTDY
      REYT=RHO(I,J)*KET(I,J)**2/(MU(I,J)*EPS(I,J))*REYREF**2
      C2=C2REF*(1.-0.3*EXP(-REYT**2))
      P1=MUT(I,J)*(4./3.*(DUDX**2+DVDY**2-DUDX*DVDY)+(DUDY+DVDX)**2)
      P2=-2./3.*RHO(I,J)*KET(I,J)*(DUDX+DVDY)
      DUUDYY=(U(I,J+1)-2.*U(I,J)+U(I,J-1))/DY**2
```



```
      DVVDYY=(V(I,J+1)-2.*V(I,J)+V(I,J-1))/DY**2
      DTTDYY=(T(I,J+1)-2.*T(I,J)+T(I,J-1))/DY**2
      DKKEDYY=(KET(I,J+1)-2.*KET(I,J)+KET(I,J-1))/DY**2
      DEEPDYY=(EPS(I,J+1)-2.*EPS(I,J)+EPS(I,J-1))/DY**2
      IF(I.LT.LX4) THEN
       DUUDXX=(U(I+1,J)-2.*U(I,J)+U(I-1,J))/DX**2
       DVVDXX=(V(I+1,J)-2.*V(I,J)+V(I-1,J))/DX**2
       DTTDXX=(T(I+1,J)-2.*T(I,J)+T(I-1,J))/DX**2
       DKKEDXX=(KET(I+1,J)-2.*KET(I,J)+KET(I-1,J))/DX**2
       DEEPDXX=(EPS(I+1,J)-2.*EPS(I,J)+EPS(I-1,J))/DX**2
       DUUDXY=(U(I+1,J+1)+U(I-1,J-1)-U(I+1,J-1)-U(I-1,J+1))/(4.*DX*DY)
       DVVDXY=(V(I+1,J+1)+V(I-1,J-1)-V(I+1,J-1)-V(I-1,J+1))/(4.*DX*DY)
      ELSE
       DXP=X(I+1)-X(I)
       DXM=X(I)-X(I-1)
       DUUDXX=((U(I+1,J)-U(I,J))/DXP-(U(I,J)-U(I-1,J))/DXM)/DXP
       DVVDXX=((V(I+1,J)-V(I,J))/DXP-(V(I,J)-V(I-1,J))/DXM)/DXP
       DTTDXX=((T(I+1,J)-T(I,J))/DXP-(T(I,J)-T(I-1,J))/DXM)/DXP
       DKKEDXX=((KET(I+1,J)-KET(I,J))/DXP-(KET(I,J)-KET(I-1,J))/DXM)/DXP
       DEEPDXX=((EPS(I+1,J)-EPS(I,J))/DXP-(EPS(I,J)-EPS(I-1,J))/DXM)/DXP
       DUUDXY=(U(I+1,J+1)+U(I-1,J-1)-U(I+1,J-1)-U(I-1,J+1))/
     &(4.*DY*(DXP+DXM))
       DVVDXY=(V(I+1,J+1)+V(I-1,J-1)-V(I+1,J-1)-V(I-1,J+1))/
     &(4.*DY*(DXP+DXM))
      ENDIF

C     source terms

  318 CPORT=1.
      SC=0.
      SXM=0.
      SRM=0.
      SE=-R23*RREYN*MUEFF*(DUDX+DVDY)**2
      SKET=0.
      SEPS=0.

C     continuity

      DRDT(I,J)=-DRUDX-DRVDY+SC

C     x-momentum

      DRUDT(I,J)=-DRUUDX-DRUVDY-DPDX*RGMA+RREYN*DMEFDX*
     &(R43*DUDX-R23*CPORT*DVDY)+RREYN*DMEFDY*(DUDY+CPORT*DVDX)+
     &RREYN*MUEFF*(R43*DUUDXX+DUUDYY+R13*CPORT*DVVDXY)+SXM

C     r- momentum

      DRVDT(I,J)=-DRUVDX-DRVVDY+1./CPORT*(-DPDY*RGMA+
     &RREYN*DMEFDX*(DUDY+CPORT*DVDX)+RREYN*DMEFDY*(R43*CPORT*
     &DVDY-R23*DUDX)+RREYN*MUEFF*(R43*CPORT*DVVDYY+CPORT*
     &DVVDXX+R13*DUUDXY)+SRM)

C     energy

      DREDT(I,J)=-DREUDX-CPORT*DREVDY+RPRREMA*
     &(DKEFDX*DTDX+KEFF*DTTDXX+DKEFDY*DTDY+KEFF*DTTDYY)-
     &RHO(I,J)*T(I,J)*RGMA*(DUDX+CPORT*DVDY)+2.*RREYN*MUEFF*(DUDX**2+
     &CPORT*DVDY**2+0.5*(CPORT*DVDX**2+DUDY**2+2.*CPORT*DVDX*DUDY))+SE
```

```
C     k-equation

      DRKEDT(I,J)=-DRUKEDX-DRVKEDY+RREYN*(DMUDX*DKEDX+
     &DMUDY*DKEDY+MU(I,J)*(DKKEDXX+DKKEDYY)+1./SIGK*(DMUTDX*DKEDX+
     &DMUTDY*DKEDY+MUT(I,J)*(DKKEDXX+DKKEDYY)))+
     &RREYN*P1+P2-RREYN*RHO(I,J)*EPS(I,J)-
     &2.*RREYN*MU(I,J)*(DSRKDX**2+DSRKDY**2)+SKET

C     epsilon-equation

      DREPDT(I,J)=-DRUEPDX-DRVEPDY+RREYN*(DMUDX*DEPDX+
     &DMUDY*DEPDY+MU(I,J)*(DEEPDXX+DEEPDYY)+1./SIGE*(DMUTDX*DEPDX+
     &DMUTDY*DEPDY+MUT(I,J)*(DEEPDXX+DEEPDYY)))+
     &C1*EPS(I,J)/KET(I,J)*(RREYN*P1+P2)-
     &C2*RREYN*RHO(I,J)/KET(I,J)*EPS(I,J)**2-
     &2.*RREYN*MU(I,J)*MUT(I,J)/RHO(I,J)*(DUUDXX**2+DUUDYY**2+
     &DVVDXX**2+DVVDYY**2+2.*DUUDXY**2+2.*DVVDXY**2)+SEPS

C     advance variables in time

      ROUB=RHO(I,J)*U(I,J)+DRUDT(I,J)*DT
      ROVB=RHO(I,J)*V(I,J)+DRVDT(I,J)*DT
      ROEB=RHO(I,J)*E(I,J)+DREDT(I,J)*DT
      ROKE=RHO(I,J)*KET(I,J)+DRKEDT(I,J)*DT
      ROEP=RHO(I,J)*EPS(I,J)+DREPDT(I,J)*DT
      RHOB(I,J)=RHO(I,J)+DRDT(I,J)*DT

C     fourth order damping terms (new bar values)

      TXD=ABS(T(I+1,J)-2.*T(I,J)+T(I-1,J))/(T(I+1,J)+2.*T(I,J)+T(I-1,J))
      TYD=ABS(T(I,J+1)-2.*T(I,J)+T(I,J-1))/(T(I,J+1)+2.*T(I,J)+T(I,J-1))
      PXD=ABS(P(I+1,J)-2.*P(I,J)+P(I-1,J))/(P(I+1,J)+2.*P(I,J)+P(I-1,J))
      PYD=ABS(P(I,J+1)-2.*P(I,J)+P(I,J-1))/(P(I,J+1)+2.*P(I,J)+P(I,J-1))
      WX=CXD*(TXD+PXD)
      WY=CYD*(TYD+PYD)
      ROUB=ROUB+WX*(RHO(I+1,J)*U(I+1,J)-2.*RHO(I,J)*U(I,J)+
     &RHO(I-1,J)*U(I-1,J))+WY*(RHO(I,J+1)*U(I,J+1)-2.*RHO(I,J)*U(I,J)+
     &RHO(I,J-1)*U(I,J-1))
      ROVB=ROVB+WX*(RHO(I+1,J)*V(I+1,J)-2.*RHO(I,J)*V(I,J)+
     &RHO(I-1,J)*V(I-1,J))+WY*(RHO(I,J+1)*V(I,J+1)-2.*RHO(I,J)*V(I,J)+
     &RHO(I,J-1)*V(I,J-1))
      ROEB=ROEB+WX*(RHO(I+1,J)*E(I+1,J)-2.*RHO(I,J)*E(I,J)+
     &RHO(I-1,J)*E(I-1,J))+WY*(RHO(I,J+1)*E(I,J+1)-2.*RHO(I,J)*E(I,J)+
     &RHO(I,J-1)*E(I,J-1))
      ROKE=ROKE+WX*(RHO(I+1,J)*KET(I+1,J)-2.*RHO(I,J)*
     &KET(I,J)+RHO(I-1,J)*KET(I-1,J))+WY*(RHO(I,J+1)*KET(I,J+1)-
     &2.*RHO(I,J)*KET(I,J)+RHO(I,J-1)*KET(I,J-1))
      ROEP=ROEP+WX*(RHO(I+1,J)*EPS(I+1,J)-2.*RHO(I,J)*
     &EPS(I,J)+RHO(I-1,J)*EPS(I-1,J))+WY*(RHO(I,J+1)*EPS(I,J+1)-
     &2.*RHO(I,J)*EPS(I,J)+RHO(I,J-1)*EPS(I,J-1))
      RHOB(I,J)=RHOB(I,J)+WX*(RHO(I+1,J)-2.*RHO(I,J)+RHO(I-1,J))+
     &WY*(RHO(I,J+1)-2.*RHO(I,J)+RHO(I,J-1))

C     solve for bar values

      UB(I,J)=ROUB/RHOB(I,J)
      VB(I,J)=ROVB/RHOB(I,J)
      EB(I,J)=ROEB/RHOB(I,J)
      TB(I,J)=EB(I,J)*GAMMA*(GAMMA-1.)*MACHREF**2
      PB(I,J)=RHOB(I,J)*TB(I,J)*RGMA
```



```
      KETB(I,J)=0.
      EPSB(I,J)=0.
      SRKB(I,J)=SQRT(KETB(I,J))
  310 CONTINUE
  300 CONTINUE

C     set wall BC for corrector step

      DX=DX11
      NX1PP=NX1P+1
      NYIGP=NYIG+1
      NY2P=NY2+1

C     injector face (except port opening)

      DO 350 J = NYIGP,NY2P
      TB(NX1P,J)=TB(NX1PP,J)
       PB(NX1P,J)=PB(NX1PP,J)
       RHOB(NX1P,J)=PB(NX1P,J)*(RHOREF*VREF**2)/(R*TB(NX1P,J)*
     &TAMB)/RHOREF
       EB(NX1P,J)=CV*TB(NX1P,J)*TREF/VREF**2
  350 CONTINUE

C     injector wall (bottom)

      NY2P3=NY2PP+1
      DO 352 I = 2,NX11
      TB(I,NY2PP)=TB(I,NY2P3)
       PB(I,NY2PP)=PB(I,NY2P3)
       RHOB(I,NY2PP)=PB(I,NY2PP)*(RHOREF*VREF**2)/(R*TB(I,NY2PP)*
     &TAMB)/RHOREF
       EB(I,NY2PP)=CV*TB(I,NY2PP)*TREF/VREF**2
  352 CONTINUE

C     injector corner

      RHOB(NX1P,NY2PP)=0.5*(RHOB(NX1PP,NY2PP)+RHOB(NX1P,NY2P3))
      PB(NX1P,NY2PP)=0.5*(PB(NX1PP,NY2PP)+PB(NX1P,NY2P3))
      TB(NX1P,NY2PP)=PB(NX1P,NY2PP)*GMA/RHOB(NX1P,NY2PP)
      EB(NX1P,NY2PP)=CV*TB(NX1P,NY2PP)*TREF/VREF**2
      LYM=LY-1

C     forward vertical wall (upstream plenum)

      DO 355 J = NY2PP,LY
      TB(1,J)=TB(2,J)
       PB(1,J)=PB(2,J)
       RHOB(1,J)=PB(1,J)*(RHOREF*VREF**2)/(R*TB(1,J)*
     &TAMB)/RHOREF
       EB(1,J)=CV*TB(1,J)*TREF/VREF**2
  355 CONTINUE
      LXM=LX-1

C     upstream plenum aft-end vertical wall

      LMYP=LMY+1
      DO 360 J = LY-JSPACE+2,LY
      TB(LX,J)=TB(LXM,J)
       PB(LX,J)=PB(LXM,J)
       RHOB(LX,J)=PB(LX,J)*(RHOREF*VREF**2)/(R*TB(LX,J)*
```

```
     &TAMB)/RHOREF
      EB(LX,J)=CV*TB(LX,J)*TREF/VREF**2
  360 CONTINUE

C     CP wall

      LXP=LX+1
      LMYM=LMY-1

C     vertical wall/spacer corner (upstream)

      RHOB(LX,LY-JSPACE+1)=0.5*(RHOB(LXM,LY-JSPACE+1)+
     &RHOB(LX,LY-JSPACE))
      PB(LX,LY-JSPACE+1)=0.5*(PB(LXM,LY-JSPACE+1)+PB(LX,LY-JSPACE))
      TB(LX,LY-JSPACE+1)=PB(LX,LY-JSPACE+1)*GMA/RHOB(LX,LY-JSPACE+1)
      EB(LX,LY-JSPACE+1)=CV*TB(LX,LY-JSPACE+1)*TREF/VREF**2

C     spacer top

      DO 356 I = LX+1,ISPACE-1
      TB(I,LY-JSPACE+1)=TB(I,LY-JSPACE)
      PB(I,LY-JSPACE+1)=PB(I,LY-JSPACE)
      RHOB(I,LY-JSPACE+1)=PB(I,LY-JSPACE+1)*(RHOREF*VREF**2)/
     &(R*TB(I,LY-JSPACE+1)*TAMB)/RHOREF
      EB(I,LY-JSPACE+1)=CV*TB(I,LY-JSPACE+1)*TREF/VREF**2
  356 CONTINUE

C     vertical wall/spacer corner (downstream)

      RHOB(ISPACE,LY-JSPACE+1)=0.5*(RHOB(ISPACE+1,LY-JSPACE+1)+
     &RHOB(ISPACE,LY-JSPACE))
      PB(ISPACE,LY-JSPACE+1)=0.5*(PB(ISPACE+1,LY-JSPACE+1)+
     &PB(ISPACE,LY-JSPACE))
      TB(ISPACE,LY-JSPACE+1)=PB(ISPACE,LY-JSPACE+1)*
     &GMA/RHOB(ISPACE,LY-JSPACE+1)
      EB(ISPACE,LY-JSPACE+1)=CV*TB(ISPACE,LY-JSPACE+1)*TREF/VREF**2

C     spacer vertical wall (downstream)

      DO 357 J = LY-JSPACE+2,LMY
      TB(ISPACE,J)=TB(ISPACE+1,J)
      PB(ISPACE,J)=PB(ISPACE+1,J)
      RHOB(ISPACE,J)=PB(ISPACE,J)*(RHOREF*VREF**2)/(R*TB(ISPACE,J)*
     &TAMB)/RHOREF
      EB(ISPACE,J)=CV*TB(ISPACE,J)*TREF/VREF**2
  357 CONTINUE

C      CORRECTOR STEP

C     set up symmetry condition at CL

      DO 405 I = NX1P,LMAX
      UB(I,1)=UB(I,3)
      VB(I,1)=-VB(I,3)
      VB(I,2)=0.
      RHOB(I,1)=RHOB(I,3)
      PB(I,1)=PB(I,3)
      TB(I,1)=TB(I,3)
      EB(I,1)=EB(I,3)
      KETB(I,1)=KETB(I,3)
```



```
      EPSB(I,1)=EPSB(I,3)
      SRKB(I,1)=SRKB(I,3)
  405 CONTINUE

      DO 430 I = 2,LMXM
      DX=DX11
      IF(I.GT.LX4) DX=X(I)-X(I-1)
      DO 440 J = 2,LYM
      IF((I.LE.NX1P).AND.(J.LE.NY2PP)) GO TO 440
      IF((I.GE.LX).AND.(J.GE.LMY)) GO TO 440
      IF(I.GE.LX.AND.I.LE.ISPACE) THEN
       IF(J.GE.(LY-JSPACE+1)) GOTO 440
      ELSE
       CONTINUE
      ENDIF
      DY=DY2

C     set up derivatives

      CRUDX=(RHOB(I,J)*UB(I,J)-RHOB(I-1,J)*UB(I-1,J))/DX
      CRVDY=(RHOB(I,J)*VB(I,J)-RHOB(I,J-1)*VB(I,J-1))/DY
      CRUUDX=(RHOB(I,J)*UB(I,J)**2-RHOB(I-1,J)*UB(I-1,J)**2)/DX
      CRUVDY=(RHOB(I,J)*UB(I,J)*VB(I,J)-RHOB(I,J-1)*UB(I,J-1)*
     &VB(I,J-1))/DY
      CPDX=(RHOB(I,J)*TB(I,J)-RHOB(I-1,J)*TB(I-1,J))/DX
      CMUDX=(MU(I,J)-MU(I-1,J))/DX
      CUDX=(UB(I,J)-UB(I-1,J))/DX
      CVDY=(VB(I,J)-VB(I,J-1))/DY
      CMUDY=(MU(I,J)-MU(I,J-1))/DY
      CUDY=(UB(I,J)-UB(I,J-1))/DY
      CVDX=(VB(I,J)-VB(I-1,J))/DX
      CRUVDX=(RHOB(I,J)*UB(I,J)*VB(I,J)-RHOB(I-1,J)*UB(I-1,J)*
     &VB(I-1,J))/DX
      CRVVDY=(RHOB(I,J)*VB(I,J)**2-RHOB(I,J-1)*VB(I,J-1)**2)/DY
      CPDY=(RHOB(I,J)*TB(I,J)-RHOB(I,J-1)*TB(I,J-1))/DY
      CREUDX=(RHOB(I,J)*EB(I,J)*UB(I,J)-RHOB(I-1,J)*EB(I-1,J)*
     &UB(I-1,J))/DX
      CREVDY=(RHOB(I,J)*EB(I,J)*VB(I,J)-RHOB(I,J-1)*EB(I,J-1)*
     &VB(I,J-1))/DY
      CKDX=(K(I,J)-K(I-1,J))/DX
      CTDX=(TB(I,J)-TB(I-1,J))/DX
      CKDY=(K(I,J)-K(I,J-1))/DY
      CTDY=(TB(I,J)-TB(I,J-1))/DY
      CRUKEDX=(RHOB(I,J)*UB(I,J)*KETB(I,J)-RHOB(I-1,J)*UB(I-1,J)*
     &KETB(I-1,J))/DX
      CRVKEDY=(RHOB(I,J)*VB(I,J)*KETB(I,J)-RHOB(I,J-1)*VB(I,J-1)*
     &KETB(I,J-1))/DY
      CKEDX=(KETB(I,J)-KETB(I-1,J))/DX
      CKEDY=(KETB(I,J)-KETB(I,J-1))/DY
      CMUTDX=(MUT(I,J)-MUT(I-1,J))/DX
      CMUTDY=(MUT(I,J)-MUT(I,J-1))/DY
      CSRKDX=(SRKB(I,J)-SRKB(I-1,J))/DX
      CSRKDY=(SRKB(I,J)-SRKB(I,J-1))/DY
      CRUEPDX=(RHOB(I,J)*UB(I,J)*EPSB(I,J)-RHOB(I-1,J)*UB(I-1,J)*
     &EPSB(I-1,J))/DX
      CRVEPDY=(RHOB(I,J)*VB(I,J)*EPSB(I,J)-RHOB(I,J-1)*VB(I,J-1)*
     &EPSB(I,J-1))/DY
      CEPDX=(EPSB(I,J)-EPSB(I-1,J))/DX
      CEPDY=(EPSB(I,J)-EPSB(I,J-1))/DY
      CKTDX=(KT(I,J)-KT(I-1,J))/DX
```

```
      CKTDY=(KT(I,J)-KT(I,J-1))/DY
      MUEFF=MU(I,J)+MUT(I,J)
      KEFF=K(I,J)+KT(I,J)
      CMEFDX=CMUDX+CMUTDX
      CMEFDY=CMUDY+CMUTDY
      CKEFDX=CKDX+CKTDX
      CKEFDY=CKDY+CKTDY
      REYT=RHOB(I,J)*KETB(I,J)**2/(MU(I,J)*EPSB(I,J))*REYREF**2
      C2=C2REF*(1.-0.3*EXP(-REYT**2))
      P1=MUT(I,J)*(4./3.*(CUDX**2+CVDY**2-CUDX*CVDY)+(CUDY+CVDX)**2)
      P2=-2./3.*RHOB(I,J)*KETB(I,J)*(CUDX+CVDY)
      CUUDYY=(UB(I,J+1)-2.*UB(I,J)+UB(I,J-1))/DY**2
      CVVDYY=(VB(I,J+1)-2.*VB(I,J)+VB(I,J-1))/DY**2
      CTTDYY=(TB(I,J+1)-2.*TB(I,J)+TB(I,J-1))/DY**2
      CKKEDYY=(KETB(I,J+1)-2.*KETB(I,J)+KETB(I,J-1))/DY**2
      CEEPDYY=(EPSB(I,J+1)-2.*EPSB(I,J)+EPSB(I,J-1))/DY**2
      IF(I.LT.LX4) THEN
       CUUDXX=(UB(I+1,J)-2.*UB(I,J)+UB(I-1,J))/DX**2
       CVVDXX=(VB(I+1,J)-2.*VB(I,J)+VB(I-1,J))/DX**2
       CTTDXX=(TB(I+1,J)-2.*TB(I,J)+TB(I-1,J))/DX**2
       CKKEDXX=(KETB(I+1,J)-2.*KETB(I,J)+KETB(I-1,J))/DX**2
       CEEPDXX=(EPSB(I+1,J)-2.*EPSB(I,J)+EPSB(I-1,J))/DX**2
       CUUDXY=(UB(I+1,J+1)+UB(I-1,J-1)-UB(I+1,J-1)-UB(I-1,J+1))/
     &(4.*DX*DY)
       CVVDXY=(VB(I+1,J+1)+VB(I-1,J-1)-VB(I+1,J-1)-VB(I-1,J+1))/
     &(4.*DX*DY)
      ELSE
       DXP=X(I+1)-X(I)
       DXM=X(I)-X(I-1)
       CUUDXX=((UB(I+1,J)-UB(I,J))/DXP-(UB(I,J)-UB(I-1,J))/DXM)/DXM
       CVVDXX=((VB(I+1,J)-VB(I,J))/DXP-(VB(I,J)-VB(I-1,J))/DXM)/DXM
       CTTDXX=((TB(I+1,J)-TB(I,J))/DXP-(TB(I,J)-TB(I-1,J))/DXM)/DXM
       CKKEDXX=((KETB(I+1,J)-KETB(I,J))/DXP-(KETB(I,J)-KETB(I-1,J))/
     &DXM)/DXM
       CEEPDXX=((EPSB(I+1,J)-EPSB(I,J))/DXP-(EPSB(I,J)-EPSB(I-1,J))/
     &DXM)/DXM
       CUUDXY=(UB(I+1,J+1)+UB(I-1,J-1)-UB(I+1,J-1)-UB(I-1,J+1))/
     &(4.*DY*(DXP+DXM))
       CVVDXY=(VB(I+1,J+1)+VB(I-1,J-1)-VB(I+1,J-1)-VB(I-1,J+1))/
     &(4.*DY*(DXP+DXM))
      ENDIF
C     source terms
  455 CPORT=1.
      SC=0.
      SXM=0.
      SRM=0.
      SE=-R23*RREYN*MUEFF*(CUDX+CVDY)**2
      SKET=0.
      SEPS=0.
C     continuity
      CRDT=-CRUDX-CRVDY+SC
C     x-momentum
      CRUDT=-CRUUDX-CRUVDY-CPDX*RGMA+RREYN*CMEFDX*
     &(R43*CUDX-R23*CPORT*CVDY)+RREYN*CMEFDY*(CUDY+CPORT*CVDX)+
```



```
     &RREYN*MUEFF*(R43*CUUDXX+CUUDYY+R13*CPORT*CVVDXY)+SXM
C     r-momentum
      CRVDT=-CRUVDX-CRVVDY+1./CPORT*(-CPDY*RGMA+
     &RREYN*CMEFDX*(CUDY+CPORT*CVDX)+RREYN*CMEFDY*(R43*CPORT*
     &CVDY-R23*CUDX)+RREYN*MUEFF*(R43*CPORT*CVVDYY+CPORT*
     &CVVDXX+R13*CUUDXY)+SRM)
C     energy
      CREDT=-CREUDX-CPORT*CREVDY+RPRREMA*
     &(CKEFDX*CTDX+KEFF*CTTDXX+CKEFDY*CTDY+KEFF*CTTDYY)-
     &RHOB(I,J)*TB(I,J)*RGMA*(CUDX+CPORT*CVDY)+2.*RREYN*MUEFF*(CUDX**2
     &+CPORT*CVDY**2+0.5*(CPORT*CVDX**2+CUDY**2+2.*CPORT*CVDX*CUDY))
     &+SE
C     k-equation
      CRKEDT=-CRUKEDX-CRVKEDY+RREYN*(CMUDX*CKEDX+
     &CMUDY*CKEDY+MU(I,J)*(CKKEDXX+CKKEDYY)+1./SIGK*(CMUTDX*CKEDX+
     &CMUTDY*CKEDY+MUT(I,J)*(CKKEDXX+CKKEDYY)))+
     &RREYN*P1+P2-RREYN*RHOB(I,J)*EPSB(I,J)-
     &2.*RREYN*MU(I,J)*(CSRKDX**2+CSRKDY**2)+SKET
C     epsilon equation
      CREPDT=-CRUEPDX-CRVEPDY+RREYN*(CMUDX*CEPDX+
     &CMUDY*CEPDY+MU(I,J)*(CEEPDXX+CEEPDYY)+1./SIGE*(CMUTDX*CEPDX+
     &CMUTDY*CEPDY+MUT(I,J)*(CEEPDXX+CEEPDYY)))+
     &C1*EPSB(I,J)/KETB(I,J)*(RREYN*P1+P2)-
     &C2*RREYN*RHOB(I,J)/KETB(I,J)*EPSB(I,J)**2-
     &2.*RREYN*MU(I,J)*MUT(I,J)/RHOB(I,J)*(CUUDXX**2+CUUDYY**2+
     &CVVDXX**2+CVVDYY**2+2.*CUUDXY**2+2.*CVVDXY**2)+SEPS
C     advance variables in time
      ROLD(I,J)=RHO(I,J)
      RHOU=RHO(I,J)*U(I,J)+(0.5*(DRUDT(I,J)+CRUDT))*DT
      RHOV=RHO(I,J)*V(I,J)+(0.5*(DRVDT(I,J)+CRVDT))*DT
      RHOE=RHO(I,J)*E(I,J)+(0.5*(DREDT(I,J)+CREDT))*DT
      RHOKET=RHO(I,J)*KET(I,J)+(0.5*(DRKEDT(I,J)+CRKEDT))*DT
      RHOEPS=RHO(I,J)*EPS(I,J)+(0.5*(DREPDT(I,J)+CREPDT))*DT
      RHO(I,J)=RHO(I,J)+(0.5*(DRDT(I,J)+CRDT))*DT
C     fourth order damping terms (redefine variables)
      TXD=ABS(TB(I+1,J)-2.*TB(I,J)+TB(I-1,J))/(TB(I+1,J)+2.*TB(I,J)+
     &TB(I-1,J))
      TYD=ABS(TB(I,J+1)-2.*TB(I,J)+TB(I,J-1))/(TB(I,J+1)+2.*TB(I,J)+
     &TB(I,J-1))
      PXD=ABS(PB(I+1,J)-2.*PB(I,J)+PB(I-1,J))/(PB(I+1,J)+2.*PB(I,J)+
     &PB(I-1,J))
      PYD=ABS(PB(I,J+1)-2.*PB(I,J)+PB(I,J-1))/(PB(I,J+1)+2.*PB(I,J)+
     &PB(I,J-1))
      WX=CXD*(TXD+PXD)
      WY=CYD*(TYD+PYD)
      RHOU=RHOU+WX*(RHOB(I+1,J)*UB(I+1,J)-2.*RHOB(I,J)*UB(I,J)+
     &RHOB(I-1,J)*UB(I-1,J))+WY*(RHOB(I,J+1)*UB(I,J+1)-2.*RHOB(I,J)*
     &UB(I,J)+RHOB(I,J-1)*UB(I,J-1))
```

```
      RHOV=RHOV+WX*(RHOB(I+1,J)*VB(I+1,J)-2.*RHOB(I,J)*VB(I,J)+
     &RHOB(I-1,J)*VB(I-1,J))+WY*(RHOB(I,J+1)*VB(I,J+1)-2.*RHOB(I,J)*
     &VB(I,J)+RHOB(I,J-1)*VB(I,J-1))
      RHOE=RHOE+WX*(RHOB(I+1,J)*EB(I+1,J)-2.*RHOB(I,J)*EB(I,J)+
     &RHOB(I-1,J)*EB(I-1,J))+WY*(RHOB(I,J+1)*EB(I,J+1)-2.*RHOB(I,J)*
     &EB(I,J)+RHOB(I,J-1)*EB(I,J-1))
      RHOKET=RHOKET+WX*(RHOB(I+1,J)*KETB(I+1,J)-2.*RHOB(I,J)*KETB(I,J)
     &+RHOB(I-1,J)*KETB(I-1,J))+WY*(RHOB(I,J+1)*KETB(I,J+1)-
     &2.*RHOB(I,J)*KETB(I,J)+RHOB(I,J-1)*KETB(I,J-1))
      RHOEPS=RHOEPS+WX*(RHOB(I+1,J)*EPSB(I+1,J)-2.*RHOB(I,J)*EPSB(I,J)
     &+RHOB(I-1,J)*EPSB(I-1,J))+WY*(RHOB(I,J+1)*EPSB(I,J+1)-
     &2.*RHOB(I,J)*EPSB(I,J)+RHOB(I,J-1)*EPSB(I,J-1))
      RHO(I,J)=RHO(I,J)+WX*(RHOB(I+1,J)-2.*RHOB(I,J)+RHOB(I-1,J))+
     &WY*(RHOB(I,J+1)-2.*RHOB(I,J)+RHOB(I,J-1))

C     solve for primary variables

      U(I,J)=RHOU/RHO(I,J)
      V(I,J)=RHOV/RHO(I,J)
      E(I,J)=RHOE/RHO(I,J)
      T(I,J)=E(I,J)*GAMMA*(GAMMA-1.)*MACHREF**2
      P(I,J)=RHO(I,J)*T(I,J)*RGMA
      KET(I,J)=0.
      EPS(I,J)=0.
      SRK(I,J)=SQRT(KET(I,J))
  440 CONTINUE
  430 CONTINUE

C     set wall BC for predictor step

      DX=DX11

C     injector face (except port opening)

      DO 461 J = NYIGP,NY2P
      T(NX1P,J)=T(NX1PP,J)
       P(NX1P,J)=P(NX1PP,J)
       RHO(NX1P,J)=P(NX1P,J)*(RHOREF*VREF**2)/(R*T(NX1P,J)*
     &TAMB)/RHOREF
      E(NX1P,J)=CV*T(NX1P,J)*TREF/VREF**2
  461 CONTINUE

C     injector wall (bottom)

      DO 462 I = 2,NX11
      T(I,NY2PP)=T(I,NY2P3)
       P(I,NY2PP)=P(I,NY2P3)
       RHO(I,NY2PP)=P(I,NY2PP)*(RHOREF*VREF**2)/(R*T(I,NY2PP)*
     &TAMB)/RHOREF
      E(I,NY2PP)=CV*T(I,NY2PP)*TREF/VREF**2
  462 CONTINUE

C     injector corner

      RHO(NX1P,NY2PP)=0.5*(RHO(NX1PP,NY2PP)+RHO(NX1P,NY2P3))
      P(NX1P,NY2PP)=0.5*(P(NX1PP,NY2PP)+P(NX1P,NY2P3))
      T(NX1P,NY2PP)=P(NX1P,NY2PP)*GMA/RHO(NX1P,NY2PP)
      E(NX1P,NY2PP)=CV*T(NX1P,NY2PP)*TREF/VREF**2

C     upstream plenum headend vertical wall
```



```
      DO 465 J = NY2PP,LY
      T(1,J)=T(2,J)
       P(1,J)=P(2,J)
       RHO(1,J)=P(1,J)*(RHOREF*VREF**2)/(R*T(1,J)*
     &TAMB)/RHOREF
      E(1,J)=CV*T(1,J)*TREF/VREF**2
  465 CONTINUE

C     upstream plenum aft-end vertical wall

      DO 470 J = LY-JSPACE+2,LY
      T(LX,J)=T(LXM,J)
       P(LX,J)=P(LXM,J)
       RHO(LX,J)=P(LX,J)*(RHOREF*VREF**2)/(R*T(LX,J)*
     &TAMB)/RHOREF
      E(LX,J)=CV*T(LX,J)*TREF/VREF**2
  470 CONTINUE

C     vertical wall/spacer corner (upstream)

      RHO(LX,LY-JSPACE+1)=0.5*(RHO(LXM,LY-JSPACE+1)+
     &RHO(LX,LY-JSPACE))
      P(LX,LY-JSPACE+1)=0.5*(P(LXM,LY-JSPACE+1)+P(LX,LY-JSPACE))
      T(LX,LY-JSPACE+1)=P(LX,LY-JSPACE+1)*GMA/RHO(LX,LY-JSPACE+1)
      E(LX,LY-JSPACE+1)=CV*T(LX,LY-JSPACE+1)*TREF/VREF**2

C     spacer top

      DO 456 I = LX+1,ISPACE-1
      T(I,LY-JSPACE+1)=T(I,LY-JSPACE)
      P(I,LY-JSPACE+1)=P(I,LY-JSPACE)
      RHO(I,LY-JSPACE+1)=P(I,LY-JSPACE+1)*(RHOREF*VREF**2)/
     &(R*T(I,LY-JSPACE+1)*TAMB)/RHOREF
      E(I,LY-JSPACE+1)=CV*T(I,LY-JSPACE+1)*TREF/VREF**2
  456 CONTINUE

C     vertical wall/spacer corner (downstream)

      RHO(ISPACE,LY-JSPACE+1)=0.5*(RHO(ISPACE+1,LY-JSPACE+1)+
     &RHO(ISPACE,LY-JSPACE))
      P(ISPACE,LY-JSPACE+1)=0.5*(P(ISPACE+1,LY-JSPACE+1)+
     &P(ISPACE,LY-JSPACE))
      T(ISPACE,LY-JSPACE+1)=P(ISPACE,LY-JSPACE+1)*
     &GMA/RHO(ISPACE,LY-JSPACE+1)
      E(ISPACE,LY-JSPACE+1)=CV*T(ISPACE,LY-JSPACE+1)*TREF/VREF**2

C     spacer vertical wall (downstream)

      DO 457 J = LY-JSPACE+2,LMY
      T(ISPACE,J)=T(ISPACE+1,J)
      P(ISPACE,J)=P(ISPACE+1,J)
      RHO(ISPACE,J)=P(ISPACE,J)*(RHOREF*VREF**2)/(R*T(ISPACE,J)*
     &TAMB)/RHOREF
      E(ISPACE,J)=CV*T(ISPACE,J)*TREF/VREF**2
  457 CONTINUE

      DO 610 I = 1,LX4+6
      DO 611 J = 1,LY
      IF(NTIME.EQ.200) WRITE(70,*)X(I),Y(J),U(I,J),V(I,J)
```

```
      IF(NTIME.EQ.400) WRITE(71,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.600) WRITE(72,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.800) WRITE(73,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.1000) WRITE(74,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.1200) WRITE(75,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.1400) WRITE(76,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.1600) WRITE(77,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.1800) WRITE(78,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.2000) WRITE(79,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.2200) WRITE(80,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.2400) WRITE(81,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.2600) WRITE(82,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.2800) WRITE(83,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.3000) WRITE(84,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.3200) WRITE(85,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.3400) WRITE(86,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.3600) WRITE(87,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.3800) WRITE(88,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.4000) WRITE(89,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.4200) WRITE(90,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.4400) WRITE(91,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.4600) WRITE(92,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.4800) WRITE(93,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.5000) WRITE(94,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.5200) WRITE(95,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.5400) WRITE(96,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.5600) WRITE(97,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.5800) WRITE(98,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.6000) WRITE(99,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.6200) WRITE(151,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.6400) WRITE(152,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.6600) WRITE(153,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.6800) WRITE(154,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.7000) WRITE(155,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.7200) WRITE(156,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.7400) WRITE(157,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.7600) WRITE(158,*)X(I),Y(J),U(I,J),V(I,J)
      IF(NTIME.EQ.7800) WRITE(159,*)X(I),Y(J),U(I,J),V(I,J)
  611 CONTINUE
  610 CONTINUE

 1111 CONTINUE
      WRITE(*,*) ' SPECIFIED DO LOOP LIMIT EXCEEDED'
      GO TO 887
  886 WRITE(*,*) ' SPECIFIED TIME LIMIT EXCEEDED'
      WRITE(9,*) ' SPECIFIED TIME LIMIT EXCEEDED'
  887 CONTINUE
  700 FORMAT(6E16.8)
  701 FORMAT(3E16.8)
  702 FORMAT(4E16.8)
  703 FORMAT(2E16.8)
  704 FORMAT(E16.8)
  999 CLOSE(UNIT=1)
      CLOSE(UNIT=2)
      CLOSE(UNIT=3)
      CLOSE(UNIT=4)
      CLOSE(UNIT=9)
      CLOSE(UNIT=10)
      CLOSE(UNIT=11)
      CLOSE(UNIT=12)
```



```
      CLOSE(UNIT=41)
      CLOSE(UNIT=42)
      CLOSE(UNIT=70)
      CLOSE(UNIT=71)
      CLOSE(UNIT=72)
      CLOSE(UNIT=73)
      CLOSE(UNIT=74)
      CLOSE(UNIT=75)
      CLOSE(UNIT=76)
      CLOSE(UNIT=77)
      CLOSE(UNIT=78)
      CLOSE(UNIT=79)
      CLOSE(UNIT=80)
      CLOSE(UNIT=81)
      CLOSE(UNIT=82)
      CLOSE(UNIT=83)
      CLOSE(UNIT=84)
      CLOSE(UNIT=85)
      CLOSE(UNIT=86)
      CLOSE(UNIT=87)
      CLOSE(UNIT=88)
      CLOSE(UNIT=89)
      CLOSE(UNIT=90)
      CLOSE(UNIT=91)
      CLOSE(UNIT=92)
      CLOSE(UNIT=93)
      CLOSE(UNIT=94)
      CLOSE(UNIT=95)
      CLOSE(UNIT=96)
      CLOSE(UNIT=97)
      CLOSE(UNIT=98)
      CLOSE(UNIT=99)
      CLOSE(UNIT=151)
      CLOSE(UNIT=152)
      CLOSE(UNIT=153)
      CLOSE(UNIT=154)
      CLOSE(UNIT=155)
      CLOSE(UNIT=156)
      CLOSE(UNIT=157)
      CLOSE(UNIT=158)
      CLOSE(UNIT=159)
      CLOSE(UNIT=50)
      CLOSE(UNIT=51)
      CLOSE(UNIT=52)

      STOP
      END

C **************************************************************

C     SUBROUTINE RWDATA - READ AND WRITE DATA

C **************************************************************

      SUBROUTINE RWDATA
      REAL LREF,MUREF,KREF,MW,LPR,MLENGTH
      COMMON/D3/IUNIT,ISTOP,COUR,CXD,CYD
      COMMON/D3BIS/MLENGTH
      COMMON/D4/GAMMA,R,MW,TIME2,TIME1
      COMMON/D5/PAMB,TAMB,FLOW1,FLOW2,FLOW3
```

```
      COMMON/D6/X1,X2,Y1,Y2,X4,Y3,B,NX11,NX1,NX2,NY1,NY2,MY
      COMMON/D6BIS/NY1P,NX1P,LX,LX4,LY,LMY,Y34,X12,X123
      COMMON/D6TRI/NXSTIP,NYSTIP,NY2PP,NYIG,MY4
      COMMON/D7/LREF,VREF,TREF,RHOREF,RREF,KREF,MUREF
      COMMON/SEGM1/NXSEG4,LXSEG4,AKGR
      COMMON/S2/LPR,NPR
      COMMON/FLOW/MAXCAL,PFAC
      COMMON/RESTART/NSAVE,KSAVE
      COMMON/CONFIG/SPACER,NFIG,NSPACE,ISPACE,JSPACE

C     READ INPUT DATA

C     IUNIT ......... system of units:  1 --> MKS
C                                       2 --> ENGLISH
C                                       3 --> CGS
C     COUR .......... Courant No. (maximum = 0.3)
C     CXD,CYD ....... damping factors for x,y directions (0.1 - 0.45)
C     NSAVE ......... flag for calculation restart: 0 if no
C                                                   1 if yes
C     KSAVE ......... interval at which restart files are saved
C     MLENGTH ....... overall motor length
C     FLOW1 ......... injector flow velocity
C     FLOW2 ......... upstream plenum wall injection velocity
C     FLOW3 ......... CP wall injection flow velocity
C     PFAC .......... presure intensification factor for porous wall flow
C     GAMMA ......... ratio of gas specific heats
C     R ............. gas constant
C     MW ............ gas molecular weight
C     TIME1 ......... rampup time for injector flow
C     TIME2 ......... total run time
C     PAMB .......... ambient pressure
C     TAMB .......... ambient temperature

C *** note: refer to sketch below for grid input parameters

C
C
C        ^       ****************************************************
C        Y2      *                                                  *
C        ^  ******                                                  *
C        |  *                                                       *
C        |  *    .                                                  *
C        |  *    .                                                  *
C        |  *    .                                                  *
C        |  *    .                                           ********
C        |  *    .                                           *      |
C        Y1 *    .                                           *      |
C        |  *    .                                           *      Y3
C        |  *    .                                           *      |
C        |  *    .                                           *      |
C        |  *                                                *      |
C        |  *                                                *      |
C        |  **************************************************
C
C           < X1><---------------------X2-----------------------><--X4->
C
C     X1,X2,X4,Y1,Y2,Y3 .... see sketch
C     B ............. slot width
C     NX11 .......... No. of grid points in interval X1
```



```
C     NX1 ........... No. of grid points in interval X2
C     NX2 ........... No. of grid points in interval X4
C     NY1 ........... No. of grid points in interval Y1
C     NY2 ........... No. of grid points in interval Y2
C     MY ............ No. of grid points in interval Y3
C
C     AKGR .......... grid spacing factor, applied to NXSEG4 only (set to 1
C
C *** note: all ref values are max values taken at injector nozzle exit
C     LREF .......... reference length
C     VREF .......... reference velocity
C     TREF .......... reference temperature
C     RHOREF ........ reference density
C     RREF .......... reference gas constant
C     KREF .......... reference gas thermal conductivity
C     MUREF ......... reference gas viscosity
C     NYIG .......... No. of grid points in injector exit
C     NFIG .......... injector configuration
C                        0 if axial
C                        1 if radial
C     SPACER ........ inhibitor height, measured from CP bottom
C     MAXCAL ........ maximum number of time steps

      READ(1,*) IUNIT,COUR,CXD,CYD,NSAVE,KSAVE
      READ(1,*) MLENGTH,FLOW1,FLOW2,FLOW3
      READ(1,*) GAMMA,R,MW,TIME1,TIME2
      READ(1,*) PAMB,TAMB
      READ(1,*) X1,X2,Y1,Y2,X4,Y3,B,NX11,NX1,NX2,NY1,NY2,MY
      READ(1,*) AKGR
      READ(1,*) TREF,RHOREF,RREF,KREF,MUREF
      READ(1,*) NYIG,NFIG,SPACER,MAXCAL

      LREF=MLENGTH
      VREF=FLOW1/3.28

      WRITE(2,*) IUNIT,COUR,CXD,CYD
      WRITE(2,*) MLENGTH, FLOW1, FLOW2,FLOW3,PFAC
      WRITE(2,*) GAMMA,R,MW,TIME1,TIME2
      WRITE(2,*) PAMB,TAMB
      WRITE(2,*) X1,X2,Y1,Y2,X4,Y3,B,NX11,NX1,NX2,NY1,NY2,MY
      WRITE(2,*) AKGR
      WRITE(2,*) LREF,VREF,TREF,RHOREF,RREF,KREF,MUREF
      WRITE(2,*) NYIG,NFIG,SPACER,MAXCAL
      WRITE(2,*)
      WRITE(2,*)
      WRITE(2,*)
      WRITE(2,*)
      IF(IUNIT.EQ.1) GO TO 10
      IF(IUNIT.EQ.2) GO TO 20
      IF(IUNIT.EQ.3) GO TO 30
      ISTOP=1
      GO TO 999
   10 WRITE(2,110)
      WRITE(2,111)
      WRITE(2,112)
      WRITE(2,113) PAMB,TAMB
  110 FORMAT(1X,'INITIAL CONDITIONS',/)
  111 FORMAT(2X,'PRESSURE',4X,'TEMPERATURE')
  112 FORMAT(3X,'(KPa)',10X,'(K)',/)
  113 FORMAT(1X,F8.2,6X,F8.3,//)
```

```
      WRITE(2,120)
      WRITE(2,121)
      WRITE(2,122)
      WRITE(2,123) LREF,VREF,TREF,RHOREF,RREF,KREF,MUREF
  120 FORMAT(1X,'REFERENCE VALUES',/)
  121 FORMAT(2X,'REF LENGHT',3X,'REF VEL',3X,'REF TEMP',3X,'REF DENS',
     &3X,'REF GAS CONST',3X,'REF THER COND',3X,'REF VISC')
  122 FORMAT(6X,'(m)',6X,'(m/sec)',5X,'(K)',6X,'(Kg/m^3)',6X,'(J/Kg K)'
     &,8X,'(W/m K)',6X,'(Kg/m sec)',/)
  123 FORMAT(2X,F7.3,4X,F9.3,2X,F9.3,4X,F6.3,6X,F8.3,9X,F6.4,5X,F12.8,
     &//)
      WRITE(2,130)
      WRITE(2,131)
      WRITE(2,132)
      WRITE(2,133) GAMMA,R,MW
  130 FORMAT(1X,'COMBUSTION GAS PROPERTIES',/)
  131 FORMAT(2X,'GAMMA',5X,'GAS CONST',5X,'MOL WEIGHT')
  132 FORMAT(12X,'(J/Kg K)',/)
  133 FORMAT(1X,F6.4,5X,F8.3,7X,F7.3,//)
      WRITE(2,140)
      WRITE(2,141)
      WRITE(2,142)
      WRITE(2,143) X1,X2,X4,Y1,Y2,Y3,B
  140 FORMAT(1X,'SLOT DIMENSIONS',/)
  141 FORMAT(4X,'X1',8X,'X2',8X,'X4',8X,'Y1',8X,'Y2',8X,'Y3',
     &8X,'B')
  142 FORMAT(3X,7('(m)',7X),/)
  143 FORMAT(1X,7(F6.3,4X),//)
      WRITE(2,150)
      WRITE(2,151)
      WRITE(2,152) NX1,NX2,NY1,NY2,MY
  150 FORMAT(1X,'NUMBER OF INTERVALS',/)
  151 FORMAT(2X,'NX1',4X,'NX2',4X,'NY1',4X,'NY2',5X,'MY',/)
  152 FORMAT(2X,5(I3,4X),//)
      PAMB=PAMB*1000.
      PBRREF=PBRREF*1000.
      GO TO 999
   20 WRITE(2,210)
      WRITE(2,211)
      WRITE(2,212)
      WRITE(2,213) PAMB,TAMB
  210 FORMAT(1X,'INITIAL CONDITIONS',/)
  211 FORMAT(2X,'PRESSURE',4X,'TEMPERATURE')
  212 FORMAT(3X,'(psi)',10X,'(R)',/)
  213 FORMAT(1X,F7.3,7X,F8.3,//)
      WRITE(2,220)
      WRITE(2,221)
      WRITE(2,222)
      WRITE(2,223) LREF,VREF,TREF,RHOREF,RREF,KREF,MUREF
  220 FORMAT(1X,'REFERENCE VALUES',/)
  221 FORMAT(2X,'REF LENGHT',3X,'REF VEL',3X,'REF TEMP',3X,'REF DENS',
     &3X,'REF GAS CONST',3X,'REF THER COND',5X,'REF VISC')
  222 FORMAT(5X,'(in)',5X,'(ft/sec)',5X,'(R)',5X,'(lb/in^3)',3X,
     &'(ft lbf/lb R)',4X,'(lbf/sec R)',4X,'(lbf hr/ft^2)',/)
  223 FORMAT(2X,F8.3,3X,F9.3,2X,F9.3,4X,F6.3,5X,F8.3,9X,F6.4,9X,E12.5,
     &//)
      WRITE(2,230)
      WRITE(2,231)
      WRITE(2,232)
      WRITE(2,233) GAMMA,R,MW
```



```
  230 FORMAT(1X,'COMBUSTION GAS PROPERTIES',/)
  231 FORMAT(2X,'GAMMA',5X,'GAS CONST',5X,'MOL WEIGHT')
  232 FORMAT(11X,'(ft lbf/lb R)',/)
  233 FORMAT(1X,F6.4,5X,F8.3,7X,F7.3,//)
      WRITE(2,240)
      WRITE(2,241)
      WRITE(2,242)
      WRITE(2,243) X1,X2,X4,Y1,Y2,Y3,B
  240 FORMAT(1X,'SLOT DIMENSIONS',/)
  241 FORMAT(3X,'X1',8X,'X2',8X,'X4',8X,'Y1',8X,'Y2',8X,'Y3',
     &8X,'B')
  242 FORMAT(2X,7('(in)',6X),/)
  243 FORMAT(1X,7(F7.3,3X),//)
      WRITE(2,250)
      WRITE(2,251)
      WRITE(2,252) NX1,NX2,NY1,NY2,MY
  250 FORMAT(1X,'NUMBER OF INTERVALS',/)
  251 FORMAT(2X,'NX1',4X,'NX2',4X,'NY1',4X,'NY2',5X,'MY',/)
  252 FORMAT(2X,5(I3,4X),//)
      CALL BRITCONV
      GO TO 999
   30 WRITE(2,310)
      WRITE(2,311)
      WRITE(2,312)
      WRITE(2,313) PAMB,TAMB
  310 FORMAT(1X,'INITIAL CONDITIONS',/)
  311 FORMAT(2X,'PRESSURE',4X,'TEMPERATURE')
  312 FORMAT(3X,'(atm)',10X,'(k)',/)
  313 FORMAT(1X,F7.3,7X,F8.3,//)
      WRITE(2,320)
      WRITE(2,321)
      WRITE(2,322)
      WRITE(2,323) LREF,VREF,TREF,RHOREF,RREF,KREF,MUREF
  320 FORMAT(1X,'REFERENCE VALUES',/)
  321 FORMAT(2X,'REF LENGHT',3X,'REF VEL',3X,'REF TEMP',3X,'REF DENS',
     &3X,'REF GAS CONST',3X,'REF THER COND',3X,'REF VISC')
  322 FORMAT(5X,'(cm)',5X,'(cm/sec)',5X,'(K)',6X,'(g/cm^3)',5X,
     &'(cal/g K)',5X,'(cal/cm sec K)',2X,'(g/cm sec)',/)
  323 FORMAT(3X,F7.2,4X,F9.2,1X,F9.3,2X,F7.2,6X,F8.4,10X,F6.4,5X,F12.8,
     &//)
      WRITE(2,330)
      WRITE(2,331)
      WRITE(2,332)
      WRITE(2,333) GAMMA,R,MW
  330 FORMAT(1X,'COMBUSTION GAS PROPERTIES',/)
  331 FORMAT(2X,'GAMMA',5X,'GAS CONST',5X,'MOL WEIGHT')
  332 FORMAT(9X,'(cm^3 atm/g K)',/)
  333 FORMAT(1X,F6.4,4X,F8.3,7X,F7.3,//)
      WRITE(2,340)
      WRITE(2,341)
      WRITE(2,342)
      WRITE(2,343) X1,X2,X4,Y1,Y2,Y3,B
  340 FORMAT(1X,'SLOT DIMENSIONS',/)
  341 FORMAT(3X,'X1',8X,'X2',8X,'X4',8X,'Y1',8X,'Y2',8X,'Y3',
     &8X,'B')
  342 FORMAT(2X,7('(cm)',6X),/)
  343 FORMAT(1X,7(F6.2,4X),//)
      WRITE(2,350)
      WRITE(2,351)
      WRITE(2,352) NX1,NX2,NY1,NY2,MY
```

```
  350 FORMAT(1X,'NUMBER OF INTERVALS',/)
  351 FORMAT(2X,'NX1',4X,'NX2',4X,'NY1',4X,'NY2',5X,'MY',/)
  352 FORMAT(2X,5(I3,4X),//)
      CALL CGSCONV
  999 RETURN
      END
C
C ********************************************************************
C
C     SUBROUTINE BRITCONV - PERFORM UNIT TRANSFORMATION FROM BRITISH
C     SYSTEM TO SI SYSTEM
C
C ********************************************************************
C
      SUBROUTINE BRITCONV
      REAL LREF,MUREF,KREF,MW
      REAL LCONV,MCONV,MUCONV,KCONV,LPR
      COMMON/D3/IUNIT,ISTOP,COUR,CXD,CYD
      COMMON/D3BIS/MLENGTH
      COMMON/D4/GAMMA,R,MW,TIME2,TIME1
      COMMON/D5/PAMB,TAMB,FLOW1,FLOW2,FLOW3
      COMMON/D6/X1,X2,Y1,Y2,X4,Y3,B,NX11,NX1,NX2,NY1,NY2,MY
      COMMON/D6BIS/NY1P,NX1P,LX,LX4,LY,LMY,Y34,X12,X123
      COMMON/D6TRI/NXSTIP,NYSTIP,NY2PP,NYIG,MY4
      COMMON/D7/LREF,VREF,TREF,RHOREF,RREF,KREF,MUREF
      COMMON/S2/LPR,NPR
      COMMON/SEGM1/NXSEG4,LXSEG4,AKGR
C
      LCONV=1./39.37008
      MCONV=1./2.2046226
      TCONV=1./1.8
      PCONV=1./0.000145037
      RHOCONV=1./0.000036127257
      RU=8314.34
      CPCONV=1./0.185862535
      VCONV=1./3.280839895
      ECONV=1./0.737562149
      MUCONV=1./5.8015095*1000000.
      KCONV=1./0.124893857
      R=RU/MW
      PAMB=PAMB*PCONV
      TAMB=TAMB*TCONV
      X1=X1*LCONV
      X2=X2*LCONV
      Y1=Y1*LCONV
      Y2=Y2*LCONV
      X4=X4*LCONV
      Y3=Y3*LCONV
      B=B*LCONV
      LREF=LREF*LCONV
      VREF=VREF*VCONV
      TREF=TREF*TCONV
      RHOREF=RHOREF*RHOCONV
      RREF=R
      KREF=KREF*KCONV
      MUREF=MUREF*MUCONV
      RETURN
      END
C
C ********************************************************************
```



```
C
C     SUBROUTINE CGSCONV - PERFORM UNIT CONVERSION FROM CGS (modified)
C     SYSTEM TO SI SYSTEM
C
C ********************************************************************
C
      SUBROUTINE CGSCONV
      REAL LREF,MUREF,KREF,MW
      REAL LCONV,MCONV,MUCONV,KCONV,LPR
      COMMON/D3/IUNIT,ISTOP,COUR,CXD,CYD
      COMMON/D3BIS/MLENGTH
      COMMON/D4/GAMMA,R,MW,TIME2,TIME1
      COMMON/D5/PAMB,TAMB,FLOW1,FLOW2,FLOW3
      COMMON/D6/X1,X2,Y1,Y2,X4,Y3,B,NX11,NX1,NX2,NY1,NY2,MY
      COMMON/D6BIS/NY1P,NX1P,LX,LX4,LY,LMY,Y34,X12,X123
      COMMON/D6TRI/NXSTIP,NYSTIP,NY2PP,NYIG,MY4
      COMMON/D7/LREF,VREF,TREF,RHOREF,RREF,KREF,MUREF
      COMMON/S2/LPR,NPR
      COMMON/SEGM1/NXSEG4,LXSEG4,AKGR
      LCONV=0.01
      MCONV=0.001
      PCONV=101325.
      RHOCONV=1000.
      RU=8314.34
      CPCONV=1./2.3884589*10000.
      VCONV=0.01
      ECONV=0.0000001
      MUCONV=0.1
      KCONV=1./0.0023884589
      R=RU/MW
      PAMB=PAMB*PCONV
      X1=X1*LCONV
      X2=X2*LCONV
      Y1=Y1*LCONV
      Y2=Y2*LCONV
      X4=X4*LCONV
      Y3=Y3*LCONV
      B=B*LCONV
      LREF=LREF*LCONV
      VREF=VREF*VCONV
      RHOREF=RHOREF*RHOCONV
      RREF=R
      KREF=KREF*KCONV
      MUREF=MUREF*MUCONV
      RETURN
      END
C ********************************************************************
C
C     SUBROUTINE GRID - GENERATE A 2-D GRID FOR THE HYBRID MOTOR GEOMETRY
C
C ********************************************************************
C

      SUBROUTINE GRID
      COMMON/GEOM1/X(250),Y(100)
      COMMON/D6/X1,X2,Y1,Y2,X4,Y3,B,NX11,NX1,NX2,NY1,NY2,MY
      COMMON/D6BIS/NY1P,NX1P,LX,LX4,LY,LMY,Y34,X12,X123
      COMMON/STEPS/DX11,DX1,DX2,DY1,DY2,DY3
      COMMON/SEGM1/NXSEG4,LXSEG4,AKGR
      COMMON/CONFIG/SPACER,NFIG,NSPACE,ISPACE,JSPACE
```

```
C     THE GRID IS EVERYWHERE RECTANGULAR, BUT IS BROKEN UP INTO DIFFERENT
C     REGIONS, EACH WITH ITS ON GRID SPACING.  "REGIONS" ARE ASSOCIATED
C     WITH THE X AND Y DIRECTIONS, NOT AREAS.  WITHIN A  REGION, SPACINGS
C     ARE CONSTANT, EXCEPT FOR THE CP REGION DOWNSTREAM OF THE HEADEND
C     SECTION.  THERE ARE 4 X "REGIONS" AND 3 Y "REGIONS", WHICH CAN
C     BE GENERALLY DESCRIBED AS FOLLOWS:
C     X-REGIONS:  1 ........ from upstream plenum upstream wall to
C                            injector exit plane
C                 2 ........ from injector exit plane to upstream plenum
C                            downstream wall
C                 3 ........ portion of CP from upstream plenum downstream
C                            wall to headend exit plane
C                 4 ........ portion of CP from headend exit plane
C                            to motor exit plane
C     Y-REGIONS:  1 ........ from upstream plenum bottom wall to
C                            CP port bottom wall
C                 2 ........ from CP port bottom wall to injector
C                            bottom plane
C                 3 ........ across injector exit to centerline

C     CALCULATE STEP SIZES

      DX11=X1/NX11
      DX1=X2/NX1
      DX2=X4/NX2
      Y13=Y1-Y3
      DY1=Y3/MY
      MYD=NY1-MY
      DY2=Y2/NY2
      DY3=Y13/MYD

C     CALCULATE NUMBER OF Y-GRID POINTS IN SPACER
C     (NUMBER OF X-GRID POINTS = 4)

      NSPACE=SPACER/DY3

C     CALCULATE X AND Y COORDINATES FOR X-REGION 1, Y REGIONS 1-2

      X(1)=0.
      Y(1)=0.
      NY1P=NY1+1
      NX1P=NX11+1
      DO 10 I = 2,NX1P
      X(I)=X(1)+DX11*(I-1)
   10 CONTINUE
      MYP=MY+1
      MYPP=MYP+1
      DO 15 J = 2,MYP
      Y(J)=Y(1)+(J-1)*DY1
   15 CONTINUE
      DO 18 J = MYPP,NY1P
      Y(J)=Y3+(J-MYP)*DY3
   18 CONTINUE

C     CALCULATE X AND Y COORDINATES IN X-REGIONS 2-3, Y REGION 3

      NY1PP=NY1P+1
      NX1PP=NX1P+1
```



```
      LX=NX11+NX1+1
      DO 20 I = NX1PP,LX
      X(I)=X1+(I-NX1P)*DX1
   20 CONTINUE
      LY=NY1+NY2+2
      DO 30 J = NY1PP,LY
      Y(J)=Y1+(J-NY1P)*DY2
   30 CONTINUE
      DO 35 I = 1,NX2
      IH=LX+I
      X(IH)=X(LX)+I*DX2
   35 CONTINUE

C     CALCULATE X COORDINATES FOR REGION 4

      LX4=LX+NX2
      LXSEG4=LX4+NXSEG4
      NXSE4P=NXSEG4+1
      DO 50 I = 2,NXSE4P
      IH=LX4+(I-1)
      IM=I-1
      X(IH)=X(LX4)+DX11*(AKGR**IM-1.)/(AKGR-1.)
   50 CONTINUE

C *****************************
C     PLOT GRID
C *****************************

C     PLOT BOUNDARY  -

      JSPACE=MYP+NSPACE
      ISPACE=LX+3
      LY=LY-1
      WRITE(10,*) X(1),Y(1)
      WRITE(10,*) X(1),Y(NY1P)
      WRITE(10,*) X(NX1P),Y(NY1P)
      WRITE(10,*) X(NX1P),Y(LY)
      WRITE(10,*) X(LX4),Y(LY)
      WRITE(10,*) X(LX4),Y(MYP)
      WRITE(10,*) X(ISPACE),Y(MYP)
      WRITE(10,*) X(ISPACE),Y(JSPACE)
      WRITE(10,*) X(LX),Y(JSPACE)
      WRITE(10,*) X(LX),Y(1)
      WRITE(10,*) X(1),Y(1)
      WRITE(10,*) X(1),Y(NY1P)

C      WRITE(51,*) X(1),Y(1)
C      WRITE(51,*) X(1),Y(NY1P)
C      WRITE(51,*) X(NX1P),Y(NY1P)
C      WRITE(51,*) X(NX1P),Y(LY)
C      WRITE(51,*) X(LX4),Y(LY)
C      WRITE(51,*) X(LX4),Y(MYP)
C      WRITE(51,*) X(ISPACE),Y(MYP)
C      WRITE(51,*) X(ISPACE),Y(JSPACE)
C      WRITE(51,*) X(LX),Y(JSPACE)
C      WRITE(51,*) X(LX),Y(1)
C      WRITE(51,*) X(1),Y(1)
C      WRITE(51,*) X(1),Y(NY1P)

C
```

```
C  -  PLOT INTERNAL POINTS  -  VERTICAL LINES  -
C
      WRITE(10,*) X(2),Y(NY1P)
      DO 110 J = 2,NX11
      WRITE(10,*) X(J),Y(1)
      JP=J+1
      WRITE(10,*) X(JP),Y(1)
      WRITE(10,*) X(JP),Y(NY1P)
  110 CONTINUE
      WRITE(10,*) X(NX1P),Y(LY)
      WRITE(10,*) X(NX1PP),Y(LY)
      WRITE(10,*) X(NX1PP),Y(1)
      LXM=LX-1
      DO 115 J = NX1PP,LXM
      JP=J+1
      WRITE(10,*) X(J),Y(LY)
      WRITE(10,*) X(JP),Y(LY)
      WRITE(10,*) X(JP),Y(1)
  115 CONTINUE
      LX4M=LX4-1
      LXP=LX+1
      ISPACEM=ISPACE-1
      ISPACEP=ISPACE+1
      WRITE(10,*) X(LX),Y(LY)
      WRITE(10,*) X(LXP),Y(LY)
      DO 125 I = LXP,ISPACEM
      WRITE(10,*) X(I),Y(JSPACE)
      IP=I+1
      WRITE(10,*) X(IP),Y(JSPACE)
      WRITE(10,*) X(IP),Y(LY)
  125 CONTINUE

      WRITE(10,*) X(ISPACE),Y(LY)
      WRITE(10,*) X(ISPACEP),Y(LY)
      DO 126 I = ISPACEP,LX4M
      WRITE(10,*) X(I),Y(MYP)
      IP=I+1
      WRITE(10,*) X(IP),Y(MYP)
      WRITE(10,*) X(IP),Y(LY)
  126 CONTINUE

C     PLOT INTERNAL POINTS  -  HORIZONTAL LINES  -

      WRITE(10,*) X(LX),Y(LY)
      WRITE(10,*) X(LX),Y(2)
      DO 130 I = 2,MY
      WRITE(10,*) X(1),Y(I)
      IP=I+1
      WRITE(10,*) X(1),Y(IP)
      WRITE(10,*) X(LX),Y(IP)
  130 CONTINUE

      MYPP=MYP+1
      WRITE(10,*) X(LX),Y(MYP)
      WRITE(10,*) X(LX),Y(MYPP)
      DO 135 I = MYPP,NY1
      WRITE(10,*) X(1),Y(I)
      IP=I+1
      WRITE(10,*) X(1),Y(IP)
      WRITE(10,*) X(LX),Y(IP)
```



```
  135 CONTINUE

      WRITE(10,*) X(LX),Y(NY1P)
      WRITE(10,*) X(LX),Y(NY1P+1)
      DO 136 I = NY1P+1,JSPACE-1
      WRITE(10,*) X(NX1P),Y(I)
      IP=I+1
      WRITE(10,*) X(NX1P),Y(IP)
      WRITE(10,*) X(LX),Y(IP)
  136 CONTINUE

      WRITE(10,*) X(ISPACE), Y(JSPACE)
      WRITE(10,*) X(ISPACE),Y(MYP)
      WRITE(10,*) X(LX4),Y(MYP)
      WRITE(10,*) X(LX4),Y(MYPP)

      DO 137 I = MYPP,JSPACE-1
      WRITE(10,*) X(ISPACE),Y(I)
      IP=I+1
      WRITE(10,*) X(ISPACE),Y(IP)
      WRITE(10,*) X(LX4),Y(IP)
  137 CONTINUE

      WRITE(10,*) X(LX4),Y(JSPACE+1)
      DO 138 I = JSPACE+1,LY-1
      WRITE(10,*) X(NX1P),Y(I)
      IP=I+1
      WRITE(10,*) X(NX1P),Y(IP)
      WRITE(10,*) X(LX4),Y(IP)
  138 CONTINUE
      WRITE(10,*) X(LX4),Y(NY1PP)
      LYM=LY-1
      DO 140 J = NY1PP,LYM
      WRITE(10,*) X(NX1P),Y(J)
      JP=J+1
      WRITE(10,*) X(NX1P),Y(JP)
      WRITE(10,*) X(LX4),Y(JP)
  140 CONTINUE
      WRITE(10,*) X(LXSEG4),Y(LY)
      WRITE(10,*) X(LXSEG4),Y(MYP)
      WRITE(10,*) X(LX4),Y(MYP)
      LXSM=LXSEG4-1
      DO 150 J = LX4,LXSM
      WRITE(10,*) X(J),Y(LY)
      JP=J+1
      WRITE(10,*) X(JP),Y(LY)
      WRITE(10,*) X(JP),Y(MYP)
  150 CONTINUE
      WRITE(10,*) X(LXSEG4),Y(MYPP)
      DO 200 I = MYPP,LYM
      WRITE(10,*) X(LX4),Y(I)
      IH=I+1
      WRITE(10,*) X(LX4),Y(IH)
      WRITE(10,*) X(LXSEG4),Y(IH)
  200 CONTINUE
      LY=LY+1
  250 RETURN
      END

C ********************************************************************
```

```
C     SUBROUTINE CPLEN - CALCULATE NUMBER OF GRID POINTS REQUIRED FOR
C     PORTION OF MOTOR DOWNSTREAM OF HEAD-END PLENUM

C *******************************************************************

      SUBROUTINE CPLEN(X1,X2,X4,MLENGTH,NX11,AKGR,NXSEG4)
      REAL MLENGTH
      SLOT=X1+X2+X4
      DX11=X1/NX11
      DO 10 I=1,200
      X=SLOT+DX11*(AKGR**I-1.)/(AKGR-1.)
      IF(X.GE.MLENGTH) GOTO 30
   10 CONTINUE
   30 NXSEG4=I
      RETURN
      END

C *******************************************************************

C     SUBROUTINE INITDAT - INITIALIZE VARIABLES TO START PREDICTOR AND
C                          CORRECTOR STEP

C *******************************************************************

      SUBROUTINE INITDAT(LMAX,NSTART)
      REAL LPR
      REAL KET,KETB,MU,K,MUT,KT
      COMMON/PRED1/U(250,100),V(250,100),RHO(250,100),P(250,100),
     &T(250,100),E(250,100),KET(250,100),EPS(250,100),SRK(250,100)
      COMMON/COR1/UB(250,100),VB(250,100),RHOB(250,100),PB(250,100),
     &TB(250,100),EB(250,100),KETB(250,100),EPSB(250,100),
     &SRKB(250,100)
      COMMON/GEOM1/X(250),Y(100)
      COMMON/S2/LPR,NPR
      COMMON/D4/GAMMA,R,MW,TIME2,TIME1
      COMMON/D5/PAMB,TAMB,FLOW1,FLOW2,FLOW3
      COMMON/D6/X1,X2,Y1,Y2,X4,Y3,B,NX11,NX1,NX2,NY1,NY2,MY
      COMMON/D6BIS/NY1P,NX1P,LX,LX4,LY,LMY,Y34,X12,X123
      COMMON/D6TRI/NXSTIP,NYSTIP,NY2PP,NYIG,MY4
      COMMON/D7/LREF,VREF,TREF,RHOREF,RREF,KREF,MUREF
      COMMON/SEGM1/NXSEG4,LXSEG4,AKGR
      COMMON/HYB/YEX
      COMMON/INITIAL/UINJ0
      COMMON/R1/MU(250,100),K(250,100),BR(250,100),HC(250,100)
      COMMON/R2/MUT(250,100),KT(250,100)
      COMMON/CONFIG/SPACER,NFIG,NSPACE,ISPACE,JSPACE

      SMU=110.5
      RHOAMB=PAMB/(R*TAMB)
      CV=R/(GAMMA-1.)
      REYREF=RHOREF*VREF*LREF/MUREF
      NY2P=NY2+1
      IF(NSTART.NE.1)GOTO 500

C     upstream plenum from injector bottom to slot bottom

      DO 10 I = 1,LX
      DO 11 J = NY2PP,LY
```



```
      U(I,J)=0.
      V(I,J)=0.
      RHO(I,J)=RHOAMB/RHOREF
      P(I,J)=PAMB/(RHOREF*VREF**2)
      T(I,J)=TAMB/TREF
      E(I,J)=CV*TAMB/VREF**2
      KET(I,J)=0.
      EPS(I,J)=0.
      SRK(I,J)=SQRT(KET(I,J))
      UB(I,J)=0.
      VB(I,J)=0.
      RHOB(I,J)=RHOAMB/RHOREF
      PB(I,J)=PAMB/(RHOREF*VREF**2)
      TB(I,J)=TAMB/TREF
      EB(I,J)=CV*TAMB/VREF**2
      KETB(I,J)=KET(I,J)
      EPSB(I,J)=EPS(I,J)
      SRKB(I,J)=SRK(I,J)
      MUT(I,J)=0.
      KT(I,J)=0.
      MU(I,J)=(T(I,J)**1.5)*((1.+SMU/TREF)/(T(I,J)+SMU/TREF))
      K(I,J)=MU(I,J)
   11 CONTINUE
   10 CONTINUE

C     remainder of flowfield (including walls)

      NY1PP=NY1P+1
      NX1PP=NX1P+1
      DO 20 I = NX1P,LMAX
      DO 21 J = 1,LMY

      IF(I.GT.LX.AND.I.LT.ISPACE) THEN
       IF(J.GT.(LY-JSPACE+1).AND.J.LE.LMY) GOTO 21
      ELSE
       GOTO 23
      ENDIF

   23 U(I,J)=0.
      V(I,J)=0.
      RHO(I,J)=RHOAMB/RHOREF
      P(I,J)=PAMB/(RHOREF*VREF**2)
      T(I,J)=TAMB/TREF
      E(I,J)=CV*TAMB/VREF**2
      KET(I,J)=0.
      EPS(I,J)=0.
      SRK(I,J)=SQRT(KET(I,J))
      UB(I,J)=0.
      VB(I,J)=0.
      RHOB(I,J)=RHOAMB/RHOREF
      PB(I,J)=PAMB/(RHOREF*VREF**2)
      TB(I,J)=TAMB/TREF
      EB(I,J)=CV*TAMB/VREF**2
      KETB(I,J)=KET(I,J)
      EPSB(I,J)=EPS(I,J)
      SRKB(I,J)=SRK(I,J)
      MUT(I,J)=0.
      KT(I,J)=0.
      MU(I,J)=(T(I,J)**1.5)*((1.+SMU/TREF)/(T(I,J)+SMU/TREF))
      K(I,J)=MU(I,J)
```

```
   21 CONTINUE
   20 CONTINUE
      IF(NFIG.NE.0) GOTO 60

C     injector port (axial configuration)

      DO 30 J=2,NYIG
      U(NX1P,J)=UINJ0/3.28/VREF
      V(NX1P,J)=0.0
      T(NX1P,J)=TAMB/TREF
      RHO(NX1P,J)=PAMB/(R*TAMB)/RHOREF
      P(NX1P,J)=PAMB/(RHOREF*VREF**2)
      E(NX1P,J)=CV*T(NX1P,J)*TREF/VREF**2
C       KET(NX1P,J)=0.01*(U(NX1P,J)**2+V(NX1P,J)**2)
C       EPS(NX1P,J)=1.5*REYREF*(KET(NX1P,J)**1.5)/YEX
      KET(NX1P,J)=0.
      EPS(NX1P,J)=0.
      SRK(NX1P,J)=SQRT(KET(NX1P,J))
      UB(NX1P,J)=UINJ0/3.28/VREF
      VB(NX1P,J)=0.0
      TB(NX1P,J)=TAMB/TREF
      RHOB(NX1P,J)=PAMB/(R*TAMB)/RHOREF
      PB(NX1P,J)=PAMB/(RHOREF*VREF**2)
      EB(NX1P,J)=CV*TB(NX1P,J)*TREF/VREF**2
      KETB(NX1P,J)=KET(NX1P,J)
      EPSB(NX1P,J)=EPS(NX1P,J)
      SRKB(NX1P,J)=SRK(NX1P,J)
      MUT(NX1P,J)=0.
      KT(NX1P,J)=0.
      MU(NX1P,J)=(T(NX1P,J)**1.5)*((1.+SMU/TREF)/(T(NX1P,J)+SMU/TREF))
      K(NX1P,J)=MU(NX1P,J)
   30 CONTINUE
      GOTO 65

C     injector port (radial configuration)

   60 CONTINUE
      NXIGN=3+NYIG
      DO 31 I = 2,NX1P
      IF(I.LE.2.OR.I.GE.NXIGN) GOTO 31
      U(I,NY2PP)=0.0
      V(I,NY2PP)=UINJ0/3.28/VREF
      T(I,NY2PP)=TAMB/TREF
      RHO(I,NY2PP)=PAMB/(R*TAMB)/RHOREF
      P(I,NY2PP)=PAMB/(RHOREF*VREF**2)
      E(I,NY2PP)=CV*T(I,NY2PP)*TREF/VREF**2
C       KET(I,NY2PP)=0.01*(U(I,NY2PP)**2+V(I,NY2PP)**2)
C       EPS(I,NY2PP)=1.5*REYREF*(KET(I,NY2PP)**1.5)/YEX
      KET(I,NY2PP)=0.
      EPS(I,NY2PP)=0.
      SRK(I,NY2PP)=SQRT(KET(I,NY2PP))
      UB(I,NY2PP)=0.0
      VB(I,NY2PP)=UINJ0/3.28/VREF
      TB(I,NY2PP)=TAMB/TREF
      RHOB(I,NY2PP)=PAMB/(R*TAMB)/RHOREF
      PB(I,NY2PP)=PAMB/(RHOREF*VREF**2)
      EB(I,NY2PP)=CV*TB(I,NY2PP)*TREF/VREF**2
      KETB(I,NY2PP)=KET(I,NY2PP)
      EPSB(I,NY2PP)=EPS(I,NY2PP)
      SRKB(I,NY2PP)=SRK(I,NY2PP)
```



```
      MUT(I,NY2PP)=0.
      KT(I,NY2PP)=0.
      MU(I,NY2PP)=(T(I,NY2PP)**1.5)*((1.+SMU/TREF)/(T(I,NY2PP)+
     &SMU/TREF))
      K(I,NY2PP)=MU(I,NY2PP)
   31 CONTINUE

   65 CONTINUE

C     upstream plenum bottom (porous wall portion)

      DO 40 I=2, LX-1
      U(I,LY)=0.
      V(I,LY)=-FLOW2/3.28/VREF
      T(I,LY)=TAMB/TREF
      RHO(I,LY)=PAMB/(R*TAMB)/RHOREF
      P(I,LY)=PAMB/(RHOREF*VREF**2)
      E(I,LY)=CV*T(I,LY)*TREF/VREF**2
      EPS(I,LY)=1.5*REYREF*(KET(I,LY)**1.5)/YEX
      KET(I,LY)=0.
      EPS(I,LY)=0.
      SRK(I,LY)=SQRT(KET(I,LY))
      UB(I,LY)=0.
      VB(I,LY)=-FLOW2/3.28/VREF
      TB(I,LY)=TAMB/TREF
      RHOB(I,LY)=PAMB/(R*TAMB)/RHOREF
      PB(I,LY)=PAMB/(RHOREF*VREF**2)
      EB(I,LY)=CV*TB(I,LY)*TREF/VREF**2
      KETB(I,LY)=KET(I,LY)
      EPSB(I,LY)=EPS(I,LY)
      SRKB(I,LY)=SRK(I,LY)
      MUT(I,LY)=0.
      KT(I,LY)=0.
      MU(I,LY)=(T(I,LY)**1.5)*((1.+SMU/TREF)/(T(I,LY)+SMU/TREF))
      K(I,LY)=MU(I,LY)
   40 CONTINUE

C     CP wall (porous wall portion)

      DO 50 I=ISPACE+1,LX4
      U(I,LMY)=0.
      V(I,LMY)=-FLOW3/3.28/VREF
      T(I,LMY)=TAMB/TREF


      RHO(I,LMY)=PAMB/(R*TAMB)/RHOREF
      P(I,LMY)=PAMB/(RHOREF*VREF**2)
      E(I,LMY)=CV*T(I,LMY)*TREF/VREF**2
      KET(I,LMY)=0.
      EPS(I,LMY)=0.
      SRK(I,LMY)=SQRT(KET(I,LMY))
      UB(I,LMY)=0.
      VB(I,LMY)=-FLOW3/3.28/VREF
      TB(I,LMY)=TAMB/TREF
      RHOB(I,LMY)=PAMB/(R*TAMB)/RHOREF
      PB(I,LMY)=PAMB/(RHOREF*VREF**2)
      EB(I,LMY)=CV*TB(I,LMY)*TREF/VREF**2
      KETB(I,LMY)=KET(I,LMY)
      EPSB(I,LMY)=EPS(I,LMY)
```

```
      SRKB(I,LMY)=SRK(I,LMY)
      MUT(I,LMY)=0.
      KT(I,LMY)=0.
      MU(I,LMY)=(T(I,LMY)**1.5)*((1.+SMU/TREF)/(T(I,LMY)+SMU/TREF))
      K(I,LMY)=MU(I,LMY)
   50 CONTINUE
      GOTO 900

  500 CONTINUE

      DO 101 I = 1,LX
      DO 111 J = NY2PP,LY
      READ(12,700)U(I,J),V(I,J),RHO(I,J),P(I,J),T(I,J),E(I,J)
      READ(12,701)KET(I,J),EPS(I,J),SRK(I,J)
      READ(12,700)UB(I,J),VB(I,J),RHOB(I,J),PB(I,J),TB(I,J),EB(I,J)
      READ(12,701)KETB(I,J),EPSB(I,J),SRKB(I,J)
      READ(12,702)MUT(I,J),KT(I,J),MU(I,J),K(I,J)
  111 CONTINUE
  101 CONTINUE

      NY1PP=NY1P+1
      NX1PP=NX1P+1
      DO 201 I = NX1P,LMAX
      DO 211 J = 1,LMY
      READ(12,700)U(I,J),V(I,J),RHO(I,J),P(I,J),T(I,J),E(I,J)
      READ(12,701)KET(I,J),EPS(I,J),SRK(I,J)
      READ(12,700)UB(I,J),VB(I,J),RHOB(I,J),PB(I,J),TB(I,J),EB(I,J)
      READ(12,701)KETB(I,J),EPSB(I,J),SRKB(I,J)
      READ(12,702)MUT(I,J),KT(I,J),MU(I,J),K(I,J)
  211 CONTINUE
  201 CONTINUE

      DO 301 J=2,NYIG
      READ(12,700)U(NX1P,J),V(NX1P,J),RHO(NX1P,J),P(NX1P,J),
     &T(NX1P,J),E(NX1P,J)
      READ(12,701)KET(NX1P,J),EPS(NX1P,J),SRK(NX1P,J)
      READ(12,700)UB(NX1P,J),VB(NX1P,J),RHOB(NX1P,J),
     &PB(NX1P,J),TB(NX1P,J),EB(NX1P,J)
      READ(12,701)KETB(NX1P,J),EPSB(NX1P,J),SRKB(NX1P,J)
      READ(12,702)MUT(NX1P,J),KT(NX1P,J),MU(NX1P,J),K(NX1P,J)

  301 CONTINUE

      DO 401 I=2, LX-1
      READ(12,700)U(I,LY),V(I,LY),RHO(I,LY),P(I,LY),T(I,LY),E(I,LY)
      READ(12,701)KET(I,LY),EPS(I,LY),SRK(I,LY)
      READ(12,700)UB(I,LY),VB(I,LY),RHOB(I,LY),PB(I,LY),TB(I,LY),
     &EB(I,LY)
      READ(12,701)KETB(I,LY),EPSB(I,LY),SRKB(I,LY)
      READ(12,702)MUT(I,LY),KT(I,LY),MU(I,LY),K(I,LY)
  401 CONTINUE

      DO 501 I=LX+1,LX4
      READ(12,700)U(I,LMY),V(I,LMY),RHO(I,LMY),P(I,LMY),
     &T(I,LMY),E(I,LMY)
      READ(12,701)KET(I,LMY),EPS(I,LMY),SRK(I,LMY)
      READ(12,700)UB(I,LMY),VB(I,LMY),RHOB(I,LMY),PB(I,LMY),
     &TB(I,LMY),EB(I,LMY)
      READ(12,701)KETB(I,LMY),EPSB(I,LMY),SRKB(I,LMY)
      READ(12,702)MUT(I,LMY),KT(I,LMY),MU(I,LMY),K(I,LMY)
```



```
  501 CONTINUE
  700 FORMAT(6E16.8)
  701 FORMAT(3E16.8)
  702 FORMAT(4E16.8)

  900 RETURN
      END

C ******************************************************************

C     SUBROUTINE DWBC - SET DOWNSTREAM BOUNDARY CONDITIONS

C ******************************************************************

      SUBROUTINE DWBC(LMAX)
      REAL KET,KETB,MUT,KT
      COMMON/PRED1/U(250,100),V(250,100),RHO(250,100),P(250,100),
     &T(250,100),E(250,100),KET(250,100),EPS(250,100),SRK(250,100)
      COMMON/COR1/UB(250,100),VB(250,100),RHOB(250,100),PB(250,100),
     &TB(250,100),EB(250,100),KETB(250,100),EPSB(250,100),
     &SRKB(250,100)
      COMMON/GEOM1/X(250),Y(100)
      COMMON/D4/GAMMA,R,MW,TIME2,TIME1
      COMMON/D5/PAMB,TAMB,FLOW1,FLOW2,FLOW3
      COMMON/D6BIS/NY1P,NX1P,LX,LX4,LY,LMY,Y34,X12,X123
      COMMON/D7/LREF,VREF,TREF,RHOREF,RREF,KREF,MUREF
      COMMON/R2/MUT(250,100),KT(250,100)
      COMMON/CONFIG/SPACER,NFIG,NSPACE,ISPACE,JSPACE

      RHOAMB=PAMB/(R*TAMB)
      CV=R/(GAMMA-1.)
      REYREF=RHOREF*VREF*LREF/MUREF
      DO 10 J = 1,LMY
      U(LMAX,J)=0.
      V(LMAX,J)=0.
      T(LMAX,J)=TAMB/TREF
      RHO(LMAX,J)=RHOAMB/RHOREF
      P(LMAX,J)=PAMB/(RHOREF*VREF**2)
      E(LMAX,J)=CV*TAMB/VREF**2
      KET(LMAX,J)=0.
      EPS(LMAX,J)=0.
      SRK(LMAX,J)=SQRT(KET(LMAX,J))
      UB(LMAX,J)=0.
      VB(LMAX,J)=0.
      TB(LMAX,J)=TAMB/TREF
      RHOB(LMAX,J)=RHOAMB/RHOREF
      PB(LMAX,J)=PAMB/(RHOREF*VREF**2)
      EB(LMAX,J)=CV*TAMB/VREF**2
      KETB(LMAX,J)=KET(LMAX,J)
      EPSB(LMAX,J)=EPS(LMAX,J)
      SRKB(LMAX,J)=SRK(LMAX,J)
      MUT(LMAX,J)=0.
      KT(LMAX,J)=0.
   10 CONTINUE
      RETURN
      END

C ******************************************************************

C     SUBROUTINE VELBC - SET VELOCITY BOUNDARY CONDITIONS AT THE WALL
```

```
C ******************************************************************

      SUBROUTINE VELBC(LMAX)
      REAL LREF
      REAL MUT,KT,KET,KETB
      COMMON/PRED1/U(250,100),V(250,100),RHO(250,100),P(250,100),
     &T(250,100),E(250,100),KET(250,100),EPS(250,100),SRK(250,100)
      COMMON/COR1/UB(250,100),VB(250,100),RHOB(250,100),PB(250,100),
     &TB(250,100),EB(250,100),KETB(250,100),EPSB(250,100),
     &SRKB(250,100)
      COMMON/R2/MUT(250,100),KT(250,100)
      COMMON/GEOM1/X(250),Y(100)
      COMMON/D4/GAMMA,R,MW,TIME2,TIME1
      COMMON/D5/PAMB,TAMB,FLOW1,FLOW2,FLOW3
      COMMON/D6/X1,X2,Y1,Y2,X4,Y3,B,NX11,NX1,NX2,NY1,NY2,MY
      COMMON/D6BIS/NY1P,NX1P,LX,LX4,LY,LMY,Y34,X12,X123
      COMMON/D6TRI/NXSTIP,NYSTIP,NY2PP,NYIG,MY4
      COMMON/D7/LREF,VREF,TREF,RHOREF,RREF,KREF,MUREF
      COMMON/SEGM1/NXSEG4,LXSEG4,AKGR
      COMMON/CONFIG/SPACER,NFIG,NSPACE,ISPACE,JSPACE

C     upstream vertical wall (plenum)

      DO 10 J = NY2PP,LY
      U(1,J)=0.
      V(1,J)=0.
      KET(1,J)=0.
      EPS(1,J)=0.
      SRK(1,J)=SQRT(KET(1,J))
      UB(1,J)=0.
      VB(1,J)=0.
      KETB(1,J)=KET(1,J)
      EPSB(1,J)=EPS(1,J)
      SRKB(1,J)=SRK(1,J)
      MUT(1,J)=0.
      KT(1,J)=0.
   10 CONTINUE
      LYM=LY-1

C     downstream vertical wall (plenum)

      DO 30 J = LY,LY-JSPACE+1,-1
      U(LX,J)=0.
      V(LX,J)=0.
      KET(LX,J)=0.
      EPS(LX,J)=0.
      SRK(LX,J)=SQRT(KET(LX,J))
      UB(LX,J)=0.
      VB(LX,J)=0.
      KETB(LX,J)=KET(LX,J)
      EPSB(LX,J)=EPS(LX,J)
      SRKB(LX,J)=SRK(LX,J)
      MUT(LX,J)=0.
      KT(LX,J)=0.
   30 CONTINUE

C     spacer top

      DO 300 I = LX+1,ISPACE
```



```
      U(I,LY-JSPACE+1)=0.
      V(I,LY-JSPACE+1)=0.
      KET(I,LY-JSPACE+1)=0.
      EPS(I,LY-JSPACE+1)=0.
      SRK(I,LY-JSPACE+1)=SQRT(KET(I,LY-JSPACE+1))
      UB(I,LY-JSPACE+1)=0.
      VB(I,LY-JSPACE+1)=0.
      KETB(I,LY-JSPACE+1)=KET(I,LY-JSPACE+1)
      EPSB(I,LY-JSPACE+1)=EPS(I,LY-JSPACE+1)
      SRKB(I,LY-JSPACE+1)=SRK(I,LY-JSPACE+1)
      MUT(I,LY-JSPACE+1)=0.
      KT(I,LY-JSPACE+1)=0.
  300 CONTINUE

C     downstream spacer wall

      DO 310 J = LY-JSPACE+2,LMY
      U(ISPACE,J)=0.
      V(ISPACE,J)=0.
      KET(ISPACE,J)=0.
      EPS(ISPACE,J)=0.
      SRK(ISPACE,J)=SQRT(KET(ISPACE,J))
      UB(ISPACE,J)=0.
      VB(ISPACE,J)=0.
      KETB(ISPACE,J)=KET(ISPACE,J)
      EPSB(ISPACE,J)=EPS(ISPACE,J)
      SRKB(ISPACE,J)=SRK(ISPACE,J)
      MUT(ISPACE,J)=0.
      KT(ISPACE,J)=0.
  310 CONTINUE

      IF(NFIG.NE.0) GOTO 100

C     injector lower wall

      DO 50 I = 2,NX1P
      U(I,NY2PP)=0.
      V(I,NY2PP)=0.
      KET(I,NY2PP)=0.
      EPS(I,NY2PP)=0.
      SRK(I,NY2PP)=SQRT(KET(I,NY2PP))
      UB(I,NY2PP)=0.
      VB(I,NY2PP)=0.
      KETB(I,NY2PP)=KET(I,NY2PP)
      EPSB(I,NY2PP)=EPS(I,NY2PP)
      SRKB(I,NY2PP)=SRK(I,NY2PP)
      MUT(I,NY2PP)=0.
      KT(I,NY2PP)=0.
   50 CONTINUE

C     injector face (except for port)

      NYIGP=NYIG+1
      DO 60 J = NYIGP,NY2PP
      U(NX1P,J)=0.
      V(NX1P,J)=0.
      KET(NX1P,J)=0.
      EPS(NX1P,J)=0.
      SRK(NX1P,J)=SQRT(KET(NX1P,J))
      UB(NX1P,J)=0.
```

```
      VB(NX1P,J)=0.
      KETB(NX1P,J)=KET(NX1P,J)
      EPSB(NX1P,J)=EPS(NX1P,J)
      SRKB(NX1P,J)=SRK(NX1P,J)
      MUT(NX1P,J)=0.
      KT(NX1P,J)=0.
   60 CONTINUE
      GOTO 200

  100 CONTINUE

C     injector face

      DO 160 J = 1,NY2PP
      U(NX1P,J)=0.
      V(NX1P,J)=0.
      KET(NX1P,J)=0.
      EPS(NX1P,J)=0.
      SRK(NX1P,J)=SQRT(KET(NX1P,J))
      UB(NX1P,J)=0.
      VB(NX1P,J)=0.
      KETB(NX1P,J)=KET(NX1P,J)
      EPSB(NX1P,J)=EPS(NX1P,J)
      SRKB(NX1P,J)=SRK(NX1P,J)
      MUT(NX1P,J)=0.
      KT(NX1P,J)=0.
  160 CONTINUE

C     injector lower wall (except for port)

      NXIGN=3+NYIG
      DO 150 I = 2,NX1P
      IF(I.GT.2.AND.I.LT.NXIGN) GOTO 150
      U(I,NY2PP)=0.
      V(I,NY2PP)=0.
      KET(I,NY2PP)=0.
      EPS(I,NY2PP)=0.
      SRK(I,NY2PP)=SQRT(KET(I,NY2PP))
      UB(I,NY2PP)=0.
      VB(I,NY2PP)=0.
      KETB(I,NY2PP)=KET(I,NY2PP)
      EPSB(I,NY2PP)=EPS(I,NY2PP)
      SRKB(I,NY2PP)=SRK(I,NY2PP)
      MUT(I,NY2PP)=0.
      KT(I,NY2PP)=0.
  150 CONTINUE
  200 RETURN
      END
```

APPENDIX B

Typical Code Input

```
0.2 0.35 0.35 0 18000
.2700 100. 20.0 6.0
.4 287. 29. 0.0001 0.007
01.353447 298.
.0127 0.0381 0.020108 0.01905 0.1016 0.0137584
.047244 12 36 96 19 18 13
.05
00. 1.177 287. 0.0261 0.0000185
.6 1 0.01016 7810
```