MARSHALL GRANT
IN-25 CR
102767
18P.

SEMI-ANNUAL REPORT

DEVELOPMENT OF COMPUTERIZED ANALYSIS FOR SOLID PROPELLANT COMBUSTION (ISAP-2)

NAG8-627

by

T. J. Chung and Raymond Hallit

Department of Mechanical Engineering
The University of Alabama in Huntsville

prepared for

NASA/MSFC

(NASA-CR-181399) DEVELOPMENT OF COMPUTERIZED ANALYSIS FOR SOLID PROPELLANT COMBUSTION (ISAP-2) Semiannual Report (Alabama Univ.) 78 p Avail: NTIS HC A05/MF A01 CSCL 21B G3/25

N87-29634

Unclas
0102767

November, 1987

ABSTRACT

This report is an improvement to ISAP-1, "SRB Vorticity-Acoustic Coupled Instability Analysis", September, 1986. Included in this report are the automatic generation of all input data for grid configuration, boundary conditions for coupled acoustic and vortical field calculations, transformation of all dimensions to a parametric form, resulting in flexibility for the user to define the size of the problem (geometric configurations) with reduction in storage (15-65%) and computer time (50-75%). Additional research is required for the following areas: (1) effects of turbulence, (2) nonlinear wave oscillations, and (3) chemistry upon combustion instability.

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

Tables

I. INTRODUCTION

This report represents an improved version of "SRB Vorticity-Acoustic Coupled Instability Analysis - ISAP-1, September, 1986. The many basic changes to the original code include the automatic generation of all the input data for grid structure, boundary conditions, and coupling between the flow field and the acoustic/vortical field. In addition, all the dimensions in the program were transformed into a parametric form. These new parameters will enable the user to control the computer memory storage and the program execution time by specifying different sets of parameters and different geometric configurations. Also presented is the comparison between the results of the original program and those of the new one, along with recommendations to the user and additional research requirements.

As noted in the earlier report (ISAP-1), unstable waves may occur as a result of acoustic and/or vortical (hydrodynamic) oscillations. If these two different types of waves are coupled together, their physical interactions lead to extremely complicated phenomena. Theoretically, there exists an infinite number of frequencies for both acoustic and vortical oscillations. Realistically, however, only a limited number of combined frequencies are excited. Our objective is to determine the combined nature of acoustic and vortical frequencies at which instabilities may arise. This subject is important in rocket motor combustion chambers when the vortical field is coupled with acoustic pressure oscillations. In the past, the acoustic combustion instability was studied independently of the vortical instability induced by vortex motions. This report is intended to combine the two different sources of energy everywhere within the spatial domain and to determine the effect of one upon the other. This can be achieved by calculating the mean flow velocities and vorticities and their fluctuating parts of velocities and vortices, as well as the fluctuating pressure.

To elucidate this coupling mechanism, the acoustic wave equation and the perturbed vortical transport equation are solved, being combined with the results of mean flow calculations from the Navier-Stokes system by means of finite elements. With these data, growth constants are calculated and stability boundaries determined. Contributions to stability and/or instability from various sources such as combustion, convection (flow turning), and viscous damping on propellant surfaces and energy convection, momentum convection, momentum viscous damping, and dissipative energy from the interior domain are separately identified. It is also found that stability boundaries for coupled acoustic and vortical oscillations are somewhat similar to the classical hydrodynamic stability boundaries, but they occur in the form of multiple islands.

II. MESH GENERATION

In the original version of the program, the input data was read externally, and this procedure required extensive preparation for any changes in the input constants and/or the configuration of the field. This process is complex and time consuming. Consequently, a mesh generation routine is added to the program.

The program has two major parts: (1) the flow field calculations and (2) the acoustic and vortical calculations from which the stability integrals and growth constants are derived. Therefore, a complete set of input data is required for each part.

A. Flow Field

The flow field calculations include velocities and pressure with grid configurations coarser than those in ISAP-1. Figure 1a shows the original grid in ISAP-1. It is apparent that the element sizes are smaller near boundaries (4), (5), and (6) in Fig. 2, but these reductions were set arbitrarily. The mesh generation routine has a much better approach. It reduces the grid size logarithmically near the same boundaries. This will enhance the flexibility of the program. The generated arrays are the following:

(a) NENN - the element connectivity matrix; it sets the global nodal values of the nodes of each element.

(b) XX, YY - the coordinates of each global node.

(c) NU, NV, and ND - the global nodes for the Dirichlet boundary condition adjustment for the U, V velocities and pressure, respectively.

(d) UB, VB, and PB - the boundary values associated with NU, NV, and ND, respectively.

(e) UU, VV, PP - the velocities in the x-direction, y-direction, and the pressure, respectively.

B. <u>Acoustic and Vortical Field</u>

This is the smaller grid shown in Fig. 2b,d. The two grids are interconnected through the ICON matrix. Again this connectivity was previously set arbitrarily. While in the mesh generation routine, the new grid is set through ICON, such that the distances between the nodes at each boundary are as equal as possible. The arrays for this field are:

(a) NENL - the element connectivity matrix for the smaller grid.

(b) ICON - the interconnectivity matrix between the flow field and the acoustic/vortical field.

(c) NODE - the adjustment matrix for the boundary conditions in the computation of vortical modes.

(d) FFNX, FFNY - the direction cosines at the boundaries in the x and y directions, respectively.

(e) NBQ - the number of elements at each boundary.

(f) ABN, ANN - the admittance at the burning surface and nozzle, respectively.

(g) NC - the connectivity matrix of the boundary elements.

C. <u>Initial and Flow Constants</u>

For each problem, there are some basic constants that are needed to define it. They deal with the geometry, flow properties, error, time-step, and convergence parameters. Many of these constants, such as REN, GAMMA, DT, ERROR, ITMAX, and NPT, have already been defined in the original report. In order to enhance the flexibility of the program, a few additional constants dealing with the geometry of the problem were utilized in the mesh generation routine. They are the following:

(a) XMIN, XMAX, YMIN, YMAX - the limiting values of the domain.

(b) XO, YO - the coordinates of the corner node (see Fig. 1).

These geometric values must be redefined by the user every time the configuration of the problem changes.

III. PARAMETRIC DIMENSIONS

The original program was bounded by a set of fixed parameters which controls the memory storage requirements. For the flow field, the program used 832 grid points and 767 elements; for the acoustic/vortical field, it used 40 grid points and 27 elements.

In order to make the programs more flexible, parametric dimensions were added. This addition will allow the user to choose the memory storage needed for different executions. This procedure requires the definition of the parameters and the respective arrays.

A. Parameters

A total of 10 parameters is needed to maximize the flexibility of the program. All but two of these parameters are geometric; they will generate both grids. Of the remaining two parameters, one is fixed and the other is floating (not fixed).

(a) Geometric Parameters:

Four parameters are needed for each grid.

1. Flow field:

The four parameters, shown in Fig. 2, are defined as follows:

NELF1 - the number of boundary elements above the corner node in the y-direction.

NELF2 - the number of boundary elements below the corner node in the y-direction.

NETP1 - the number of boundary elements left of the corner node in the x-direction.

NETP2 - the number of boundary elements right of the corner node in the x-direction.

2. Acoustic/vortical field:

The four parameters associated with this field are similar to those of the flow field with their respective positions. NALF1, NALF2, NATP1, and NATP2 are the number of elements on boundaries (5), (1), (6), and (4), respectively (see Fig. 2).

(b) Floating Parameter:

The parameter MI is affected by other parameters and constants and is a direct result of the calculations. In the original code, the value of MI was set at 65. However, this value can be much lower (see Table 3).

From the program, the needed value of MI is augmented for each St (Strouhal number) $\geq .01$. Those values of St $< .01$ will produce oscillations which are negligible relative to the acoustic oscillations and, therefore, are eliminated to avoid unnecessary calculations. As the Reynolds number increases, the vortical oscillations expand further in the field, resulting in higher values of St at additional nodes; therefore, the needed value of MI will increase.

However, in order to limit the memory storage requirements, MI must be minimized. This procedure will be achieved by changing MI relative to the changes in REN and the geometric parameters (see Section V for the recommended values of MI).

(c) <u>Fixed Parameter</u>:

The parameter NBP, which is always equal to 6, represents the 6 boundaries of the domain. It must be noted that the remaining parameters are derived directly from the 10 preset parameters already defined.

B. Arrays

The size of all the different arrays can be set either by a DIMENSION statement or by a COMMON block. When the parameters were added to the program, two other steps had to be taken:

(1) All the COMMON blocks were eliminated from the code. This step required a change in the way the subroutines are called, by including all the variable arrays that were otherwise included in the COMMON statements.

(2) All the arrays must be referenced in the MAIN program. This step was required because the dimensions of all the arrays are generated in the PARAMETER statements in the MAIN program.

IV. COMPARISON OF RESULTS

The computer memory storage and the execution time are the important outcome of this study. Therefore, to compare these effects, the results will be displayed in two ways: (1) by comparing the original code and the revised code and (2) by comparing the original grids' results with those of the reduced ones.

First of all, the basic assumptions and flow constants are presented:

1. The Reynolds number, REN = 10^3, 10^4, 10^5 for different runs.
2. The specific heat ratio, GAMMA = 1.2.
3. The time step for the calculation of the mean flow field, DT = 1.
4. The convergence error for mean flow, ERROR = .001.
5. The maximum number of iterations for flow field calculations, ITMAX = 30.
6. The domain of calculations: XMIN = 0, XMAX = 10, YMIN = 0, YMAX = 1.5.
7. The corner node coordinates: X0 = 2, Y0 = 1.
8. Boundary and initial conditions: With reference to Fig. 1, burning takes place only on boundary (6); therefore, an instantaneous flux normal to this boundary appears at time T = 0. Then, the velocity in the negative y-direction is set at a dimensionless value of v = -.01, only at the nodes of boundary (6), and excluding the end points of this boundary. All other flow variables are set equal to zero

everywhere in the domain. (The value of -.01 was taken from the original code and is a function of the burning rate at the surface of the solid fuel).

9. The normal vectors at the boundaries: At boundaries (1) - (6), the values of the normal vectors are FFNX = 1., 0., 1., 0., 1., 0.; FFNY = 0., 1., 0., 1., 0., 1., respectively.

10. Admittance: Only at the burning surface (boundary 6), ABN = 1.; otherwise, it is equal to zero. Only at the nozzle (boundary 3), AAN = 1.; otherwise, it is equal to zero.

These assumptions and flow constants were kept fixed (except for REN) throughout the calculations. Now, we can proceed to the comparison of the results.

A. Original Grids

It is clear that with the original grids (ISAP-1), the storage required to generate the data in the revised code will exceed the storage needed to read the data from an outside source. Therefore, the total memory requirements for the revised code exceed that of the original code by about 2%.

Concerning the run time requirements, on the other hand, the revised code used about 2% less time than that of the original code, even though it went through 5 additional values of Strouhal numbers, i.e., vortical calculations (see Table 1).

NOTE: The value of MI has been defined earlier, but this value is also dependent on the systems on which these programs are running. The original code gave a value of MI = 15 on the IBM

3084 and a value of MI = 16 on the UNIVAC. This phenomenon is a result of the eigenvalue solver routine. This routine is very sensitive to any small variations in its input data. To eliminate the sensitivity problem, these codes should be transformed to operate in double-precision, while reducing the ERROR value and increasing ITMAX. These new additions will increase the storage requirements and run time by over 50%.

The flow field results, as shown in Fig. 3, prove that the revised code presents the same exact flow field as that of the original code. In the figure, the original code results are displayed in the upper half of each section for (a) the pressure, (b) the velocity in the x-direction, and (c) the velocity in the y-direction. These results were expected because of the linearity of the solution. However, the difference between the values of MI was not expected to be as large.

Even though the revised version of the program made it more flexible and easy to operate, it still has not made any major contribution to reduce the memory requirements nor the necessary run time. Therefore, we will present the other aspects of possible improvements which reduce the memory storage and run time. But, as we will see, they will slightly affect the computational results.

B. Reduced Grids

As a part of the flexibility of the revised code, the user has the option of changing the number of nodes and elements used in the program. Figure 1 shows the different grids used for comparison purposes. The number of nodes and elements are tabulated in Table 2. As seen in Table 2, if the number of elements at the boundaries is reduced by 1/4, the total number of nodes and elements is reduced by about 1/2. As a result of the reduced grids, the required memory storage and run time will definitely be reduced. Then, there is a need to find what percentage reduction is achieved and how it will affect the results.

Table 3 shows the different reductions achieved for required memory and run time at different Reynolds numbers. The storage requirement is reduced by 14% and 66% with coarser grids for the flow field and the flow and A/V fields, respectively, while the execution time is reduced between 56% and 76%. From these values, we can conclude that the flow field calculations affect mostly the run time required, while the acoustic/vortical (A/V) field affects the storage size of the program. On the other hand, the computational results of such reductions are slightly affected. Figures 4-6 show the flow field variables for the reduced grid (the lower part of each section), as comapred to the original grid, for Re = 10^3, 10^4, and 10^5, respectively. The difference between the results becomes more obvious as Re gets larger. Consequently, at high values of the Reynolds number (Re), the coarse mesh does not produce accurate results.

V. RECOMMENDATIONS

Two types of recommendations are necessary to conclude this report: (1) those directed towards the users of the program and (2) those needed for a more realistic modeling of the problem at hand.

A. To the user:

(1) Grids: This program does not contain a turbulent flow model to accommodate for the higher values of Re. Also, there is a limit, in reducing the grids, at which the results would become insignificant. Therefore, a coarse mesh, such as (6, 9, 13, 20), may be used at Re $\leq 10^3$; but, at higher values of Re, a more refined mesh is needed. Recommended values for the parameters: NELF1 $\geq$ 6, NELF2 $\geq$ 9, NETP1 $\geq$ 13, NETP2 $\geq$ 20, 2 $\leq$ NALF1, NALF2, NATP1, NATP2 $\leq$ 4. It is imperative that the last 4 parameters be $\geq$ 2; otherwise, the calculations will become senseless.

(2) The value of MI: As stated earlier, the needed value of MI is calculated within the program. The maximum allowable value of MI is equal to the number of A/V modes: MI_{max} = NT. In order to control the memory storage requirements, MI can be specified smaller than NT. As shown in Table 3, the needed values of MI increase as the Reynolds number increases.

Therefore, it is recommended that the user adjust the value of MI, as needed, to correspond with the values of the Reynolds number and the appropriate geometric parameters. The recommended values of MI are: $MI_{(needed)} < MI \le NT$, where $MI_{(needed)}$ are the values listed in Table 3.

B. Additional Recommendations:

There are many areas where this program can be improved, some of which can be done in the near future, others will be the topics of further study and research.

(1) Double precision, smaller DT, smaller ERROR, and larger ITMAX: these extra steps are needed for accuracy and to avoid the sensitivity in the eigenvalue solver.

(2) Turbulent model: this model will be used for the relatively high values of Re.

(3) Higher order approximations of the pressure and the vorticity.

(4) The study of the effect of compressibility.

(5) The addition of chemistry to the model by adding the species and energy equations.

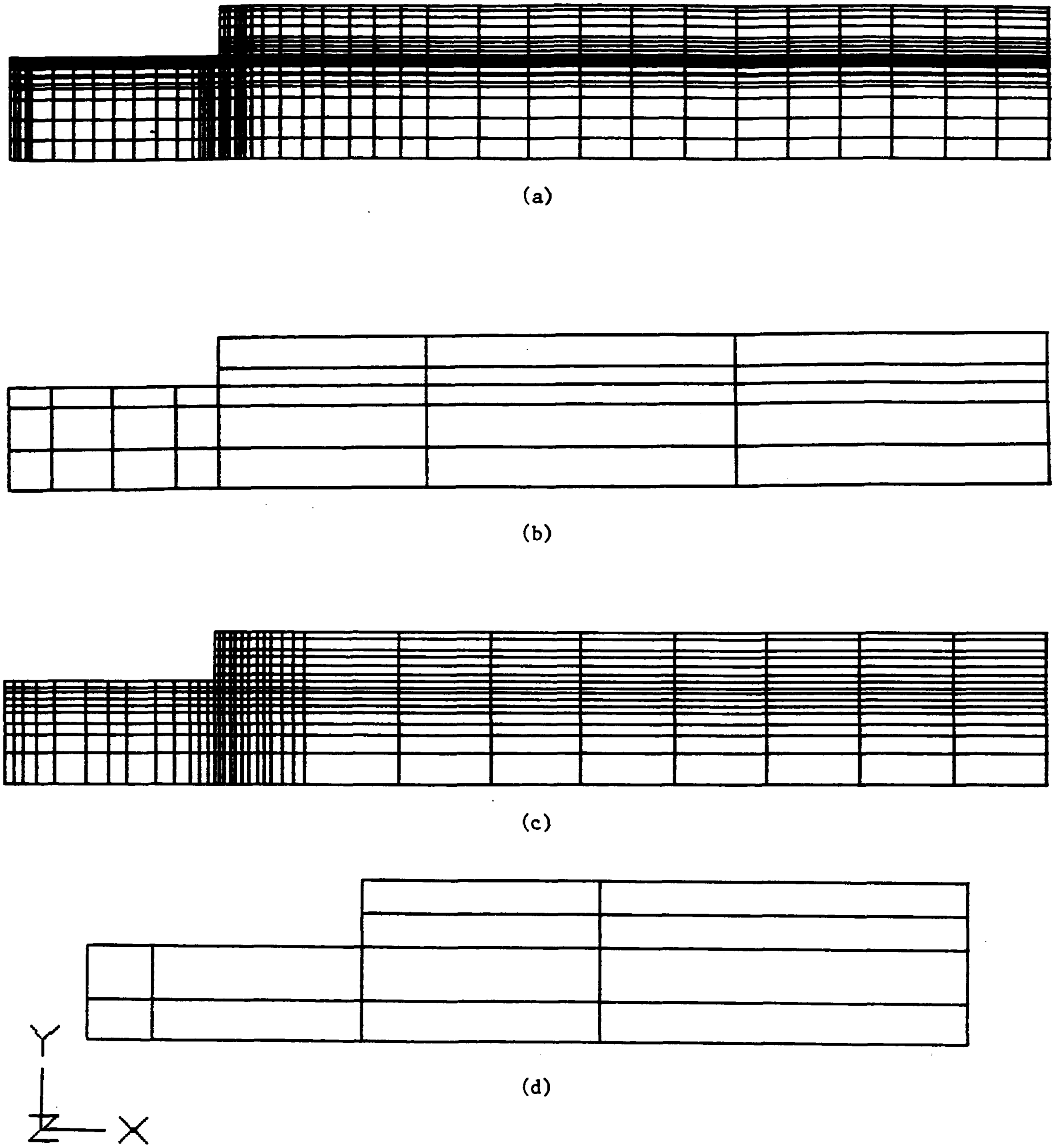


Fig 1. The grids: a) for the original flow calculations, b) for the original A/V calculations, c) for the reduced flow calculations, and d) for the reduced A/V calculations.

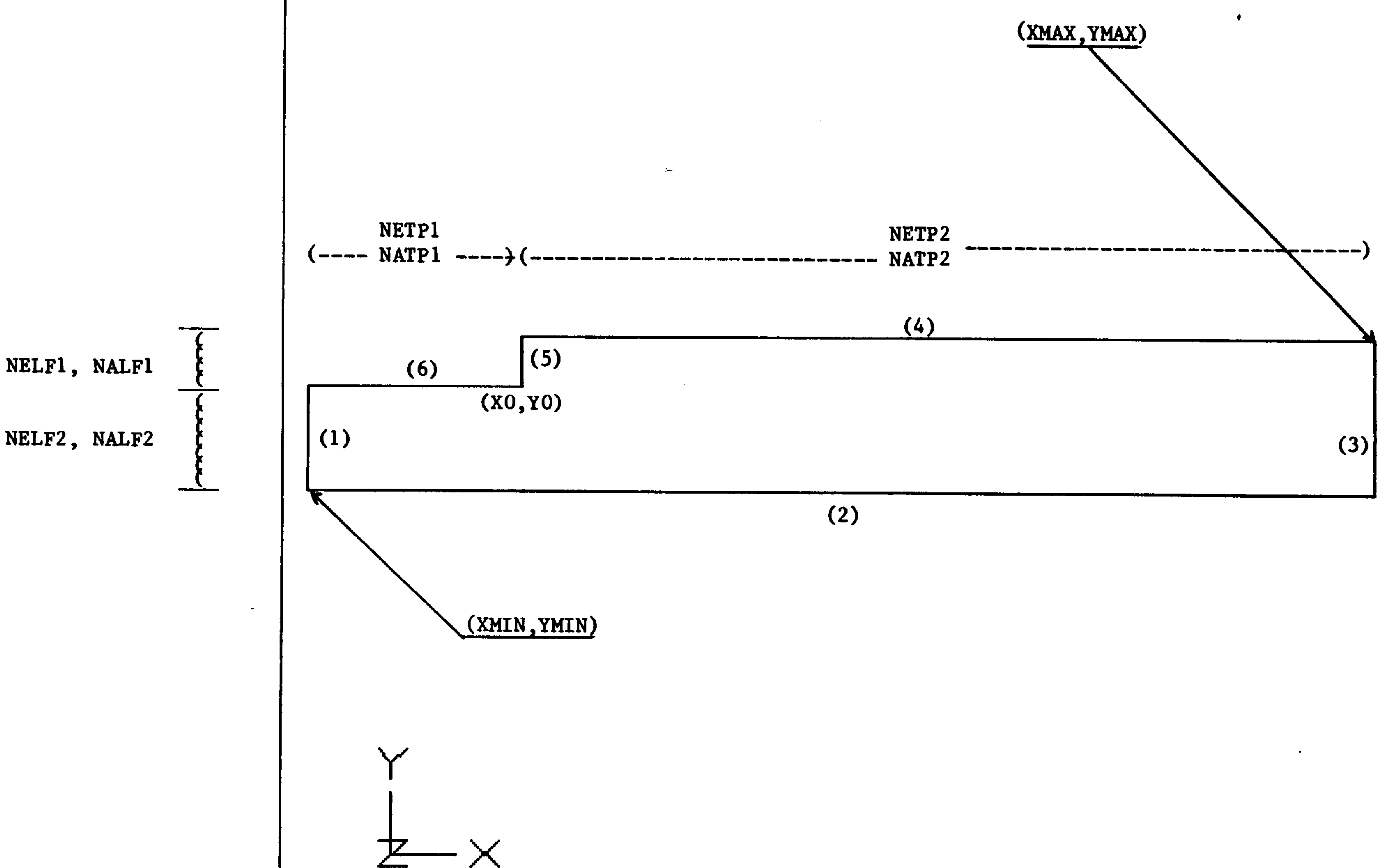
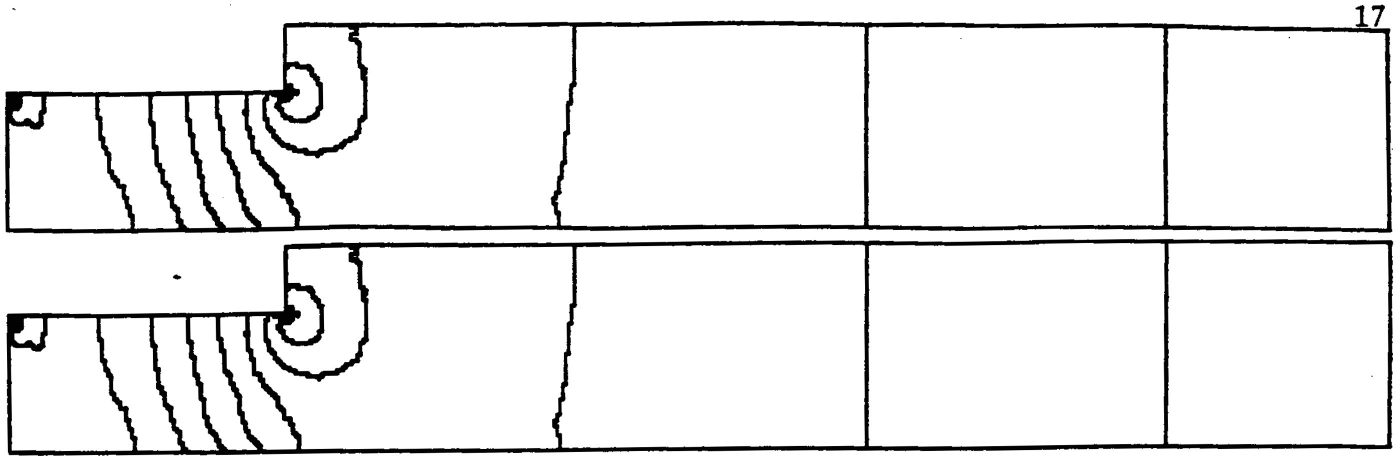
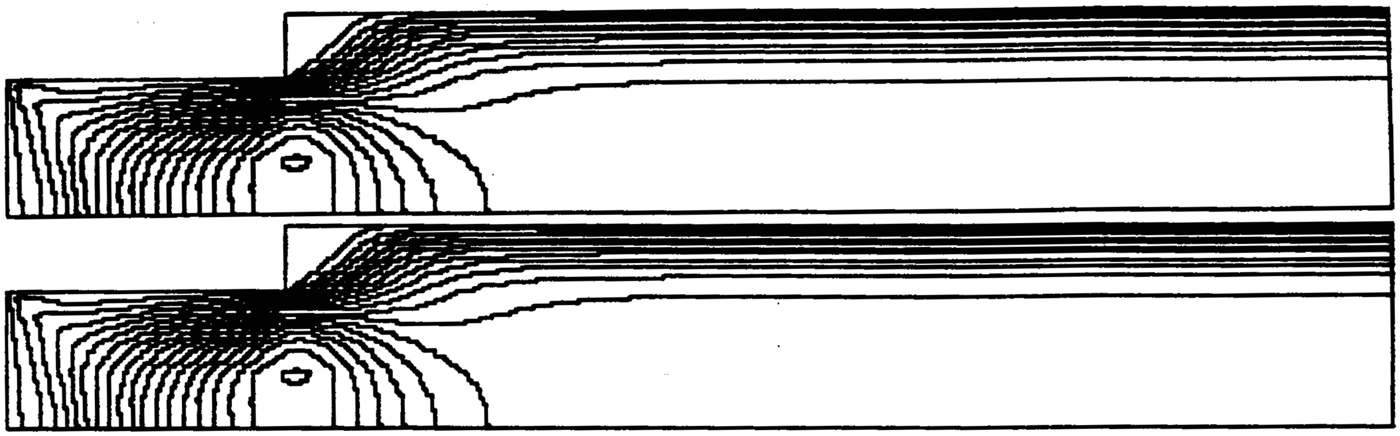


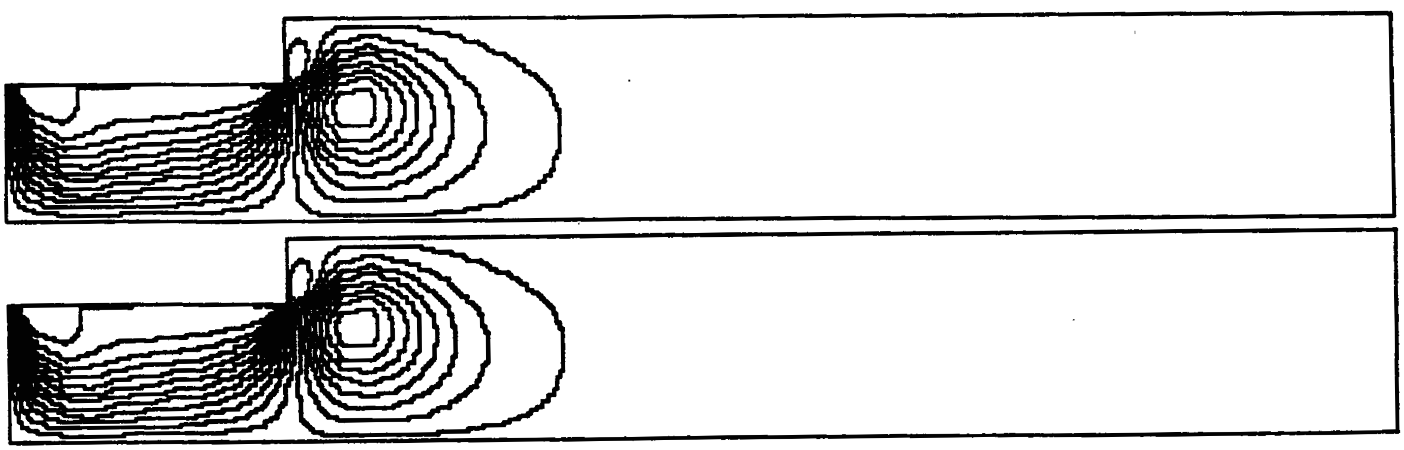
Fig 2. Parameters, boundaries, and domain of calculations.



(a)

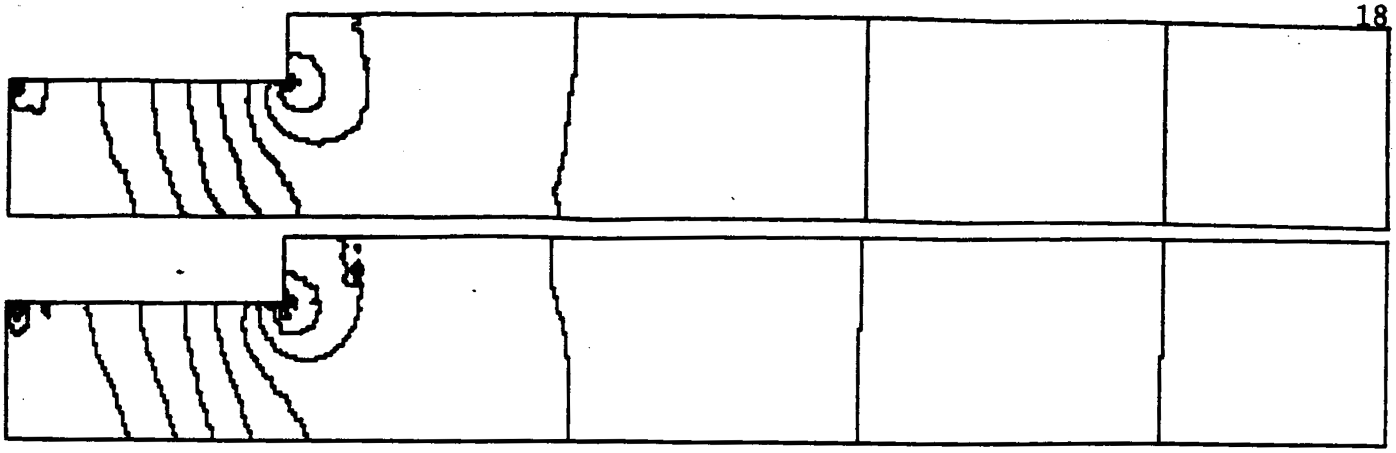


(b)

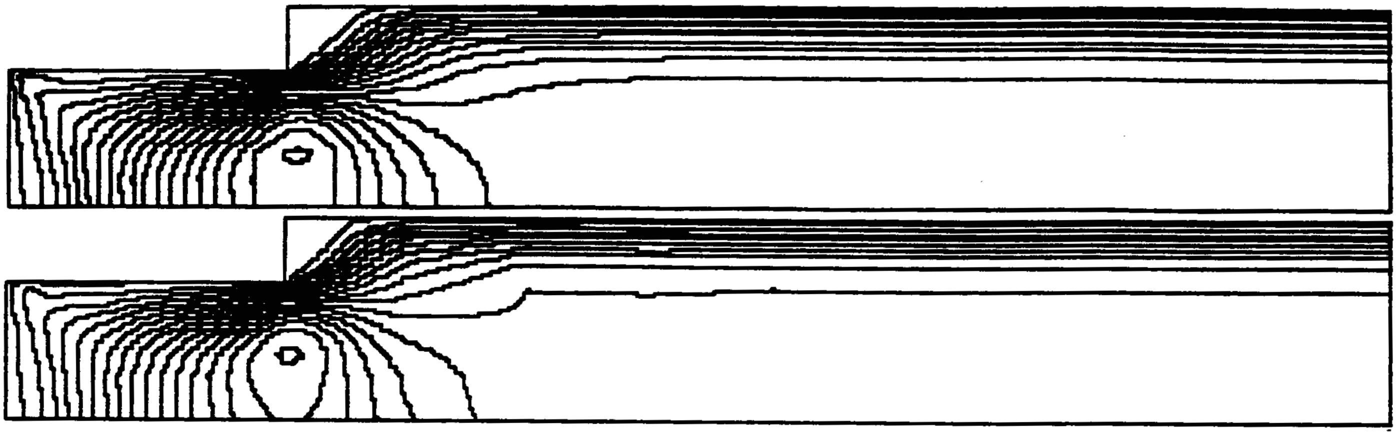


(c)

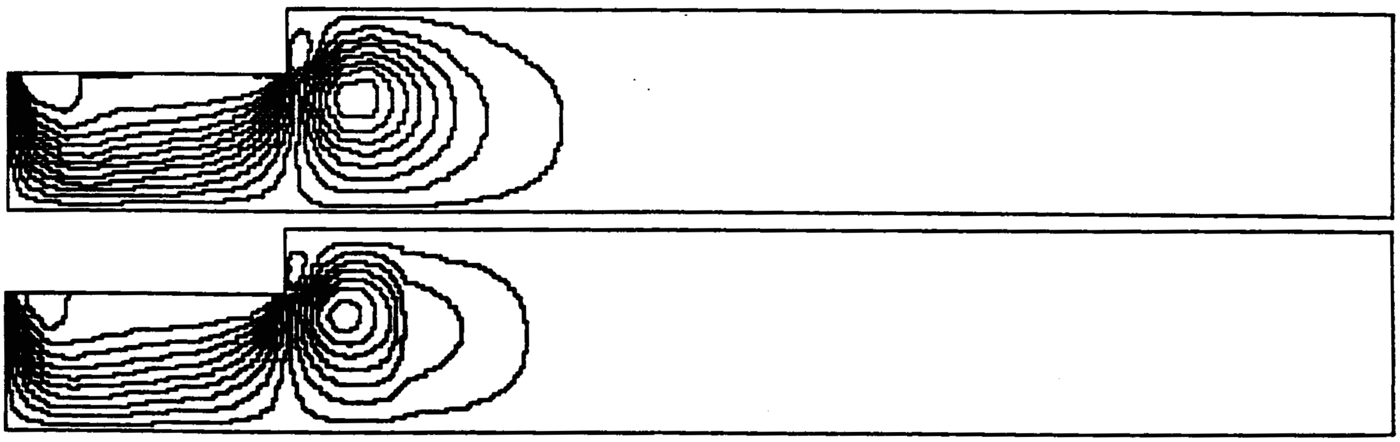
Fif 3. Comparison between the flow field results of the original code (upper part) and those of the revised code (lower part), for (a) the pressure, (b) the U-velocity, and (c) the V-velocity. (Re=1000.)



(a)

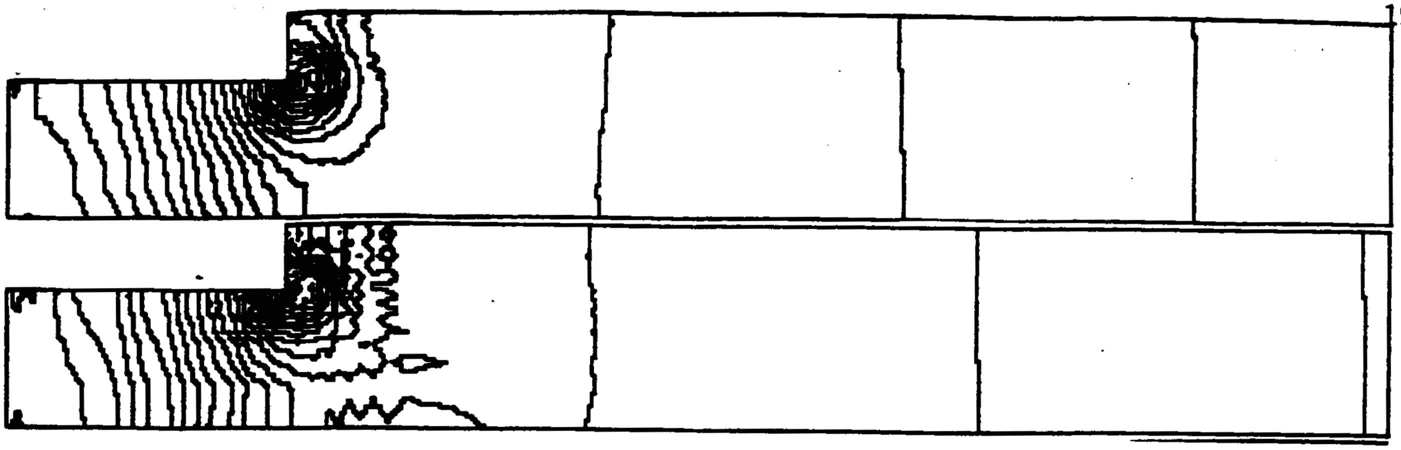


(b)

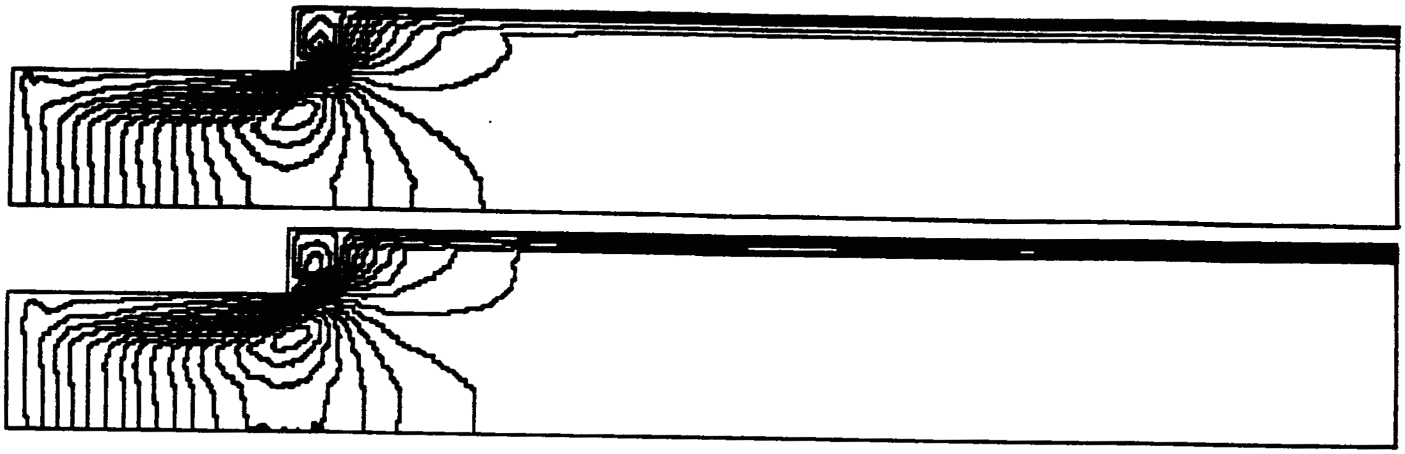


(c)

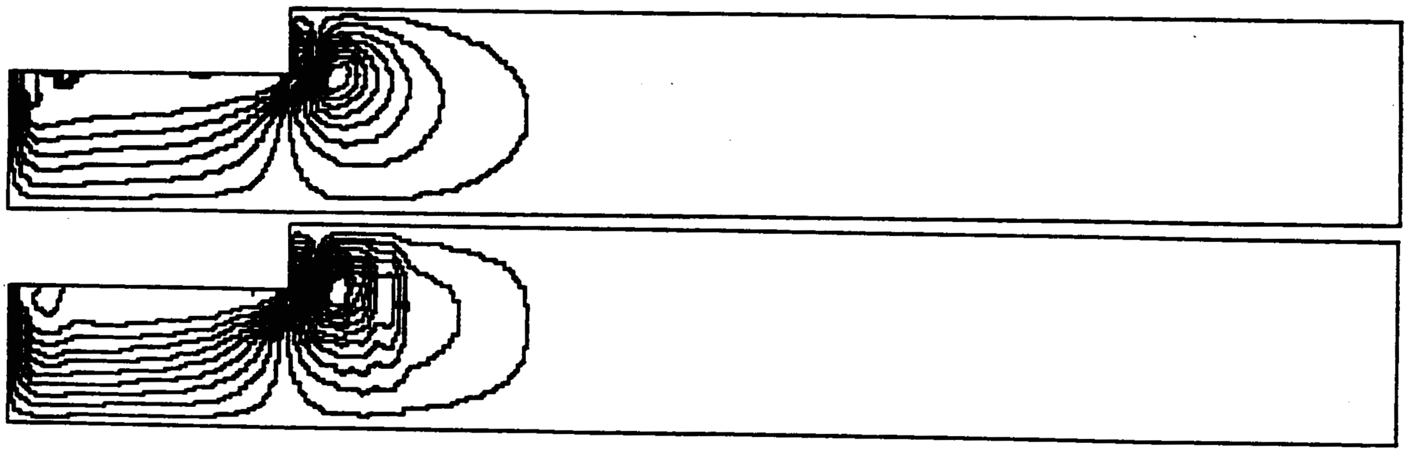
Fig 4. Comparison of the flow field variables between the original mesh (upper part) and the reduced mesh (lower part) : (a) the pressure, (b) the U-velocity, (c) the V-velocity; Re= 1000.



(a)

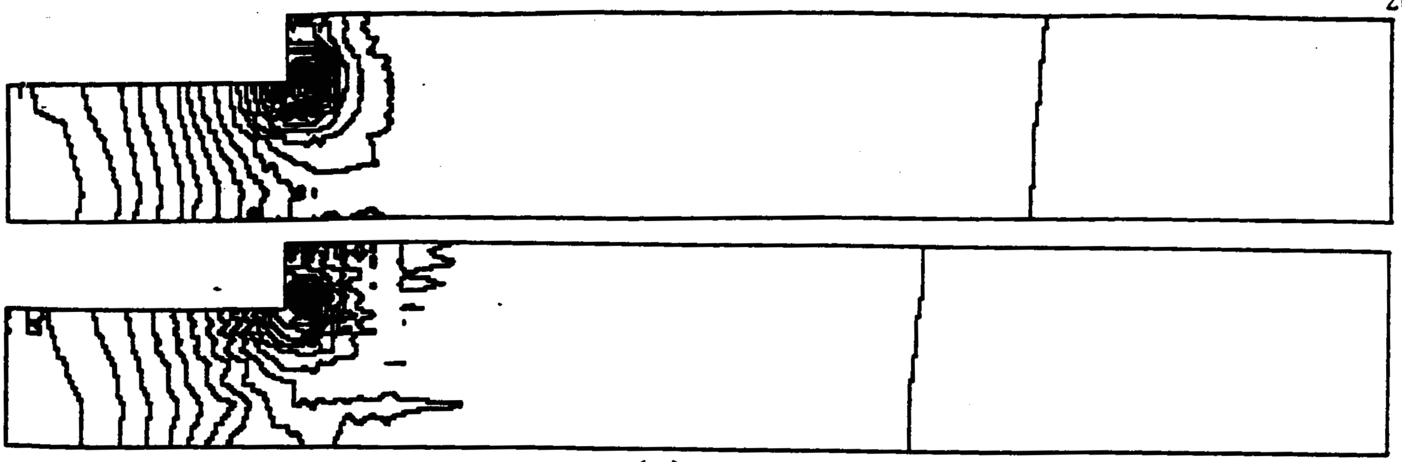


(b)

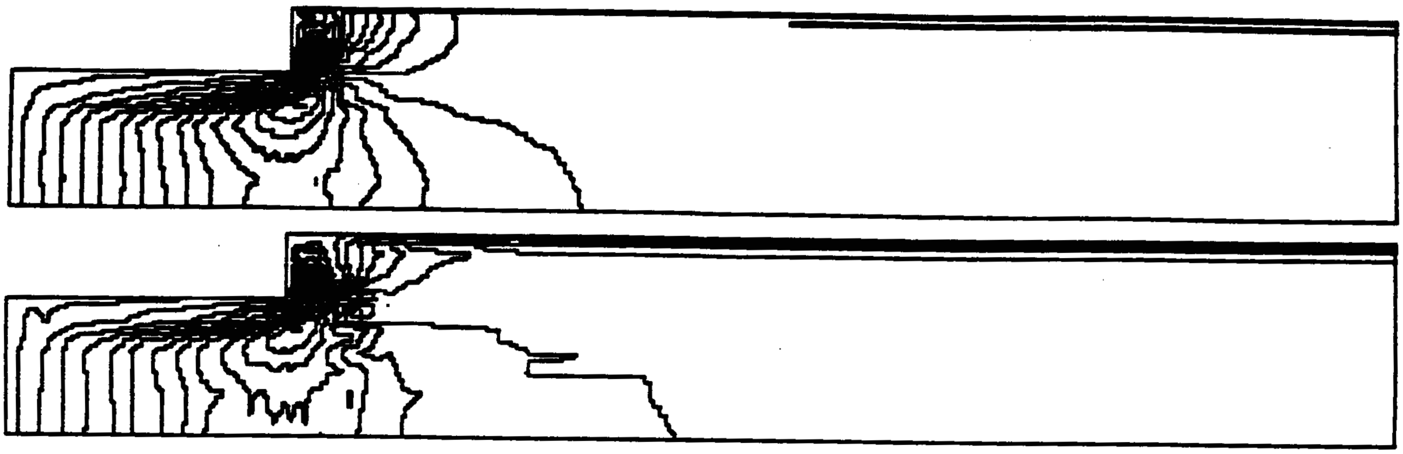


(c)

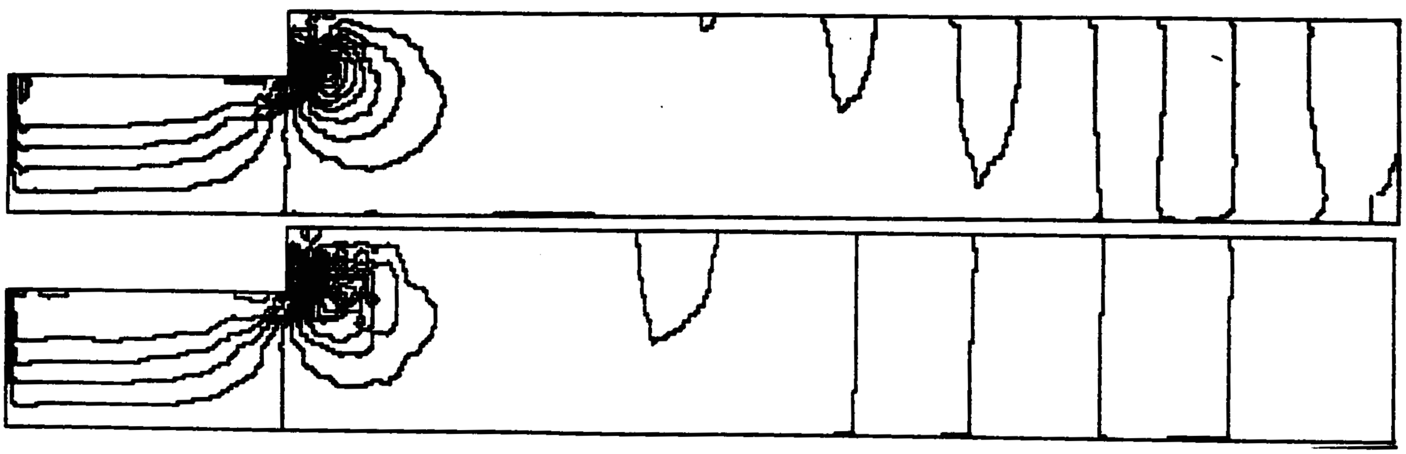
Fig 5. Comparison of the flow field variables between the original mesh (upper part) and the reduced mesh (lower part) : (a) the pressure, (b) the U-velocity, (c) the V-velocity; Re= 10000.



(a)



(b)



(c)

Fig 6. Comparison of the flow field variables between the original mesh (upper part) and the reduced mesh (lower part) : (a) the pressure, (b) the U-velocity, (c) the V-velocity; Re= 100000.

Table 1. Comparison between the storage and runtime of the original code and the revised one, at Re= 10^3. (original grids)

	Memory Requirement Bytes	Fraction of the original	Run-time (UNIVAC) min	Fraction of the original	MI (needed)
ORIGINAL	3023004	1	226	1	16
REVISED	3109452	1.02	223	.98	21

Table 2. The reduced grids.

		Parameters	No. of Nodes	No. of Elements
Flow Field	Original	(8,13,17,26)	832	767
	Reduced	(6,9,13,20)	466	417
A/V Field	Original	(2,3,4,3)	40	27
	Reduced	(2,2,2,2)	21	12

Table 3. Storage and run-time comparison between the original grids and the reduced ones. (MI was set at 65 as an initial assumption).

	Re	Memory Requirement Bytes	Fraction of the Original	Run-time (UNIVAC) min	Fraction of the Original	MI (needed)
Original GRID	10^3	3109452	1.	223	1.	21
	10^4	3109452	1.	240	1.	30
	10^5	3109452	1.	242	1.	33
Flow field reduced (only)	10^3	2682808	.86	80	.358	19
	10^4	2682808	.86	100	.420	27
	10^5	2682808	.86	105	.434	33
Flow & A/V fields reduced	10^3	1077816	.34	52	.233	7
	10^4	1077816	.34	64	.267	12
	10^5	1077816	.34	65	.270	14

APPENDIX

Listing of the Revised Code.

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  1   C
 11             PROGRAM ISAP1
  1   C         VIRTUAL A,A2,B,B2,C,CZZ2,D,SI,SR
  1   C     VIRTUAL PAA,PAH,PAT,UUI,UUR,VVI,VVR
  1   C
  1   C         PPROGRAM FOR SRB VORTICALLY-COUPLED COMBUSTION INSTABILITY
  1   C         ANALYSIS BY FINITE ELEMENTS
  1   C
  7         PARAMETER (NELF1=6,NELF2=9,NETP1=13,NETP2=20)
  1   C     PARAMETER (NELF1=8,NELF2=13,NETP1=17,NETP2=26)
  1   C     PARAMETER (NALF1=2,NALF2=3,NATP1=4,NATP2=3)
  7         PARAMETER (NALF1=2,NALF2=2,NATP1=2,NATP2=2)
  1   C
  7         PARAMETER (MI=65,NC1=20,NBP=6,
  6        &NERGT=NELF1+NELF2,NEBOT=NETP1+NETP2,
  6        &NEL=NETP2*NERGT+NETP1*NELF2,NBO=2*NERGT+7,
  6        &NARGT=NALF1+NALF2,NABOT=NATP1+NATP2,
  6        &NPT1A=(NALF2+1)*NATP1,NPT2A=(NATP2+1)*(NARGT+1),
  6        &NL=NATP2*NARGT+NATP1*NALF2,NT=NPT1A+NPT2A,MII=4*NT,
  6        &NT1=NT-NALF1-NALF2-NATP2-2,NTT=3*NT1+NT,
  6        &NPT1=(NELF2+1)*NETP1,NPT2=(NETP2+1)*(NERGT+1),
  6        &NGPT=NPT1+NPT2,NDP=NERGT+1,
  6        &NUX=NELF1+NELF2+NETP1+NETP2+3,NVY=NUX+NDP+NEBOT+1)
  1   C
  7         DIMENSION NENN(NEL,4),NENL(NL,4),ICON(NT),NC(NBP,NC1,2),FRE(NT),
  6        &PRESS(NT,NT),FVE(MI),UUR(NT,MI),UUI(NT,MI),VVR(NT,MI),
  6        &VVI(NT,MI)
  7         DIMENSION NBN1(NELF2+1),NBN2(NELF1+NETP1+1),NBN3(NETP2+1),
  6        &NBN4(NERGT+1),NBN5(NEBOT+1)
  1   C
  7         DIMENSION NODE(NTT),NBQ(NBP),FFNX(NBP),FFNY(NBP),ABN(NBP)
  7         DIMENSION ANN(NBP)
  7         DIMENSION XT(NT),YT(NT),UT(NT),VT(NT)
 11             DIMENSION NU(NUX),NV(NVY),ND(NDP),UB(NUX),VB(NVY),PB(NDP)
 11             DIMENSION XX(NGPT),YY(NGPT),UU(NGPT),VV(NGPT),PP(NGPT)
 11             DIMENSION PFR(NT),PFV(MI),PAA(NT,MI),PAH(NT,MI),PAT(NT,MI)
  7         DIMENSION US(NGPT),VS(NGPT),DPP(NGPT),DELU(NGPT),DELV(NGPT),
  6        *A(NGPT,NBO),B(NGPT,NBO),FU(NGPT),FV(NGPT),TARR1(NT),
  6        *ITER(NT),SR(MII,NTT),SI(MII,NTT),FV1(MII),ITER2(NTT)
  7         COMPLEX A1(NT,NT),B1(NT,NT),CZZ1(NT,NT),EIGAV(NT),EIGBV(NT),
  6        *EIGV(NT),A2(MII,MII),B2(MII,MII),C(NTT,NTT),D(NTT,NTT),
  6        *CZZ2(NTT,NTT),EIGAV2(NTT),EIGBV2(NTT),EIGV2(NTT)
  1   C
  7         DATA XMIN,XMAX,YMIN,YMAX/0.,10.,0.,1.5/
  7         DATA XO,YO/2.,1./
  1   C
  7         REN=1000.
  7         GAMMA=1.2
  7         DT=1.
  7         ERROR=.001
  7         ITMAX=30
  7         NPT=4
  1   C
  1   C
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

   1  C         CT=SPEED OF SOUND(IN/SEC)/CHARACTERISTIC LENGTH(IN)
   1  C
  11            CT=40000.0/24.0
   1  C
   1  C         INPUT DATA
   1  C
   7        CALL DINPUT(NELF1,NELF2,NETP1,NETP2,NALF1,NALF2,
   6       &NATP1,NATP2,NERGT,NEBOT,NARGT,NABOT,NPT1A,NPT1,
   6       &NEL,NGPT,NPT,NBW1,NBWT,NUX,NVY,NDP,NBP,NTT,NODE,
   6       &NENN,NENL,ICON,NC,NBN1,NBN2,NBN3,NBN4,NBN5,NT1,NC1,
   6       *                  XMIN,XMAX,YMIN,YMAX,XO,YO,
   6       *                 NBQ,XX,YY,UU,VV,PP,NU,NV,ND,UB,VB,PB,FFNX,
   6       *                 FFNY,ABN,ANN,REN,GAMMA,DT,ERROR,ITMAX,NL,NT)
   1  C
   1  C         COMPUTATION OF MEAN FLOW FIELDS
   1  C
  11            CALL VELOT(NEL,NGPT,NPT,NBW1,NBWT,XX,YY,UU,VV,PP,REN,DT,ERROR,
   6       * NENN,ICON,XT,YT,UT,VT,NBO,US,VS,DPP,DELU,DELV,FU,FV,
   6       *          A,B,NUX,NVY,NDP,NU,NV,ND,UB,VB,PB,NT,ITMAX)
   1  C
   1  C         COMPUTATION OF ACOUSTIC MODES
   1  C
  11            CALL EIPRE(NL,NT,NPT,NENL,FRE,PRESS,
   6       * XT,YT,TARR1,A1,B1,CZZ1,EIGAV,EIGBV,EIGV,ITER)
   1  C
   1  C      IF(NELF1.GT.1)GOTO 100
   1  C
   1  C         COMPUTATION OF VORTICAL MODES
   1  C
  11            CALL EIVOR(NL,NT,NPT,REN,NTT,NODE,MVP,MI,MII,
   6       * SR,SI,FV1,A2,B2,C,D,CZZ2,EIGAV2,EIGBV2,EIGV2,ITER2,
   6       * NENL,FVE,UUR,UUI,VVR,VVI,XT,YT,UT,VT)
   1  C
  11            WRITE(6,1000)
   1  C
   1  C         DO LOOP FOR EACH ACOUSTIC MODE
   1  C
  11            DO 1 IK=2,NT
   1  C
  11            FR=CT*FRE(IK)
  11            PFR(IK)=FR
   1  C
   1  C         DO LOOP FOR EACH VORTICAL MODE
   1  C
  11            DO 1 IG=1,MVP
   1  C
   1  C         STABILITY INTEGRAL AT THE BOUNDARY
   1  C
  11            CALL SURFCE(IK,IG,NBP,FFNX,FFNY,ABN,ANN,NC1,
   6       *            NC,FRE,PRESS,FVE,UUR,UUI,VVR,VVI,
   6       *             NT,AAA,AAB,AHB,AHC,REN,GAMMA,NBQ,MI,
   6       *              XT,YT,UT,VT)
   1  C
   1  C         STABILITY INTEGRAL IN THE VOLUME
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
START
 COL   ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  1   C
 11            CALL VOLUME(IK,IG,NPT,NL,NT,EN,AAD,AAE,AAF,
  6        *           NENL,FRE,PRESS,UUR,UUI,VVR,VVI,
  6        *                AHE,AHF,AHG,REN,GAMMA,MI,XT,YT,UT,VT)
  1   C
  1   C        DIMENSIONALIZATIONS OF GROWTH CONSTANTS
  1   C
 11            FV2 =FVE(IG)
 11            PFV(IG)=FV2
  9          CTOEN=CT/EN
 11            AAA=CTOEN*AAA
 11            AAB=CTOEN*AAB
 11            AAC=CTOEN*AAC
 11            AAD=CTOEN*AAD
 11            AAE=CTOEN*AAE
 11            AAF=CTOEN*AAF
  1   C
 11            AHB=CTOEN*AHB
 11            AHC=CTOEN*AHC
 11            AHE=CTOEN*AHE
 11            AHF=CTOEN*AHF
 11            AHG=CTOEN*AHG
  1   C
 11            AA=AAA+AAB+AAC+AAD+AAE+AAF
 11            AH=AHB+AHC+AHE+AHF+AHG
 11            AT=AA+AH
  1   C
 11            PAA(IK,IG)=AA
 11            PAH(IK,IG)=AH
 11            PAT(IK,IG)=AT
  1   C
 11            WRITE(6,1001) FR,FV2
 11            WRITE(6,1002) AT,AA,AH
 11            WRITE(6,1012)
 11            WRITE(6,1003) AAA,AAB,AAC,AAD,AAE,AAF
 11            WRITE(6,1004) AHB,AHC,AHE,AHF,AHG
  1   C
  1   1        CONTINUE
  1   C
 11            WRITE(6,1006)
 11            DO 100 IK=1,NT
 11            DO 100 IG=1,MVP
 11            WRITE(6,1005) PFR(IK),PFV(IG),PAA(IK,IG),PAH(IK,IG),
  6        *              PAT(IK,IG)
  1   100      CONTINUE
  1   C
  1   C
 11            STOP
  1   1000     FORMAT(/////,5X,'STABILITY INTEGRALS',//)
  1   1001     FORMAT(//,10X,'ACOUSTIC FREQ =',E10.5,'HZ',5X,
  6        *        'STRAUHAL NO. =',E10.5,/)
  1   1002     FORMAT(5X,'TOTAL ALP =',E15.5,3X,'ACOUSTIC ALP =',
  6        *        E15.5,3X,'VORTICAL ALP =',E15.5)
  1   1012     FORMAT(20X,'A',14X,'B',14X,'C',14X,'D',14X,'E',
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
  6        *          14X,'F',14X,'G')
  1   1003      FORMAT(6X,'ALPA',6E15.5)
  1   1004      FORMAT(6X,'ALPH',15X,2E15.5,15X,3E15.5)
  1   1005      FORMAT(5X,5E12.3)
  1   1006      FORMAT(/////)
 11             END
  1   C
  1   C
  1   C
  7         SUBROUTINE DINPUT(NELF1,NELF2,NETP1,NETP2,NALF1,NALF2,
  6        &NATP1,NATP2,NERGT,NEBOT,NARGT,NABOT,NPT1A,NPT1,
  6        &NEL,NGPT,NPT,NBW1,NBWT,NUX,NVY,NDP,NBP,NTT,NODE,
  6        &NENN,NENL,ICON,NC,NBN1,NBN2,NBN3,NBN4,NBN5,NT1,NC1,
  6        *                 XMIN,XMAX,YMIN,YMAX,XO,YO,
  6        *                NBQ,XX,YY,UU,VV,PP,NU,NV,ND,UB,VB,PB,FFNX,
  6        *                FFNY,ABN,ANN,REN,GAMMA,DT,ERROR,ITMAX,NL,NT)
  1   C
  1   C                GENERATION OF INPUT DATA
  1   C
  1   C          NEL               : NUMBER OF FINITE ELEMENTS
  1   C          NGPT              : NUMBER OF GLOBAL NODES
  1   C          NPT               : NUMBER OF LOCAL NODES IN EACH FINITE
  1   C                              ELEMENT
  1   C          NBW1              : HALF BAND WIDTH + 1 OF THE STIFFNESS
  1   C                              MATRIX
  1   C          NBWT              : TOTAL BAND WIDTH OF THE STIFFNESS
  1   C                              MATRIX
  1   C          NUX               : NUMBER OF BOUNDARY CONDITIONS OF
  1   C                              U-VELOCITY
  1   C          NVY               : NUMBER OF BOUNDARY CONDITIONS OF
  1   C                              V-VELOCITY
  1   C          NDP               : NUMBER OF BOUNDARY CONDITIONS OF
  1   C                              PRESSURE
  1   C          NTT               : NUMBER OF TITAL EQUATIONS FOR THE
  1   C                              COMPUTATION OF VORTICAL MODE
  1   C          NODE(NTT)         : ADJUSTMENT MATRIX FOR THE BOUNDARY
  1   C                              CONDITIONS OF THE COMPUTATION OF VIRTICAL
  1   C                              MODES
  1   C          NBP               : NUMBER OF BOUNDARY FACES OF DOMAIN
  1   C          NBQ(NBP)          : BOUNDARY ELEMENT MATRIX
  1   C          XX(NGPT)          : X-COORDINATE
  1   C          YY(NGPT)          : Y-COORDINATE
  1   C          UU(NGPT)          : INITIAL CONDITIONS OF U-VELOCITY
  1   C          VV(NGPT)          : INITIAL CONDITIONS OF V-VELOCITY
  1   C          PP(NGPT)          : INITIAL CONDITIONS OF PRESSURE
  1   C          NU(NUX)           : GLOBAL NODES OF BOUNDARY CONDITIONS OF
  1   C                              U-VELOCITY
  1   C          NV(NVY)           : GLOBAL NODES OF BOUNDARY CONDITIONS OF
  1   C                              V-VELOCITY
  1   C          ND(NDP)           : GLOBAL NODES OF BOUNDARY CONDITIONS OF
  1   C                              PRESSURE
  1   C          FFNX(NBP)         : X-DIRECTION NORMAL VECTORS AT EACH BOUNDARY
  1   C          FFNY(NBP)         : Y-DIRECTION NORMAL VRCTORS AT EACH BOUNDARY
  1   C          ABN(NBP)          : ADMITTANCE AT THE BURNING SURFACE
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  1  C          ANN(NBP)          : ADMITTANCE AT THE NOZZLE
  1  C          REN               : REYNOLDS NUMBER
  1  C          GAMMA             : SPECIFIC HEAT RATIO
  1  C          DT                : TIME STEP FOR THE COMPUTATION OF MEAN FLOW
  1  C          ERROR             : CONVERGENCE ERROR CRITERIA OF THE
  1  C                              COMPUTATION OF MEAN FLOW
  1  C          ITMAX             : MAXIMUM NUMBER OF ITERATIONS OF THE
  1  C                              COMPUTATION OF MEAN FLOW
  1  C
  7        DIMENSION NENN(NEL,4),NENL(NL,4),ICON(NT),NC(NBP,NC1,2)
 11            DIMENSION XX(NGPT),YY(NGPT),NU(NUX),NV(NVY),ND(NDP),
  6       *              UB(NUX),VB(NVY),PB(NDP)
 11            DIMENSION UU(NGPT),VV(NGPT),PP(NGPT)
 11            DIMENSION NODE(NTT),NBQ(NBP),FFNX(NBP),FFNY(NBP),ABN(NBP),
  6       *              ANN(NBP)
  7        DIMENSION NBN1(NELF2+1),NBN2(NELF1+NETP1+1),NBN3(NETP2+1),
  6       &NBN4(NERGT+1),NBN5(NEBOT+1)
  1  C
  1  C
  1  C          CONSTANTS FOR BAND MATRIX
  1  C
  7        NBW=NERGT+3
 11            NBW1=NBW+1
 11            NBWT=NBW*2+1
  1  C
  7        DO 10 J=1,NETP1
  7        DO 10 I=1,NELF2
  7        K=(J-1)*NELF2+I
  7        NNZ=0
  7        IF(J.EQ.NETP1)NNZ=NELF1
  7        NENN(K,1)=(J-1)*(NELF2+1)+I+1
  7        NENN(K,2)=J*(NELF2+1)+I+1+NNZ
  7        NENN(K,3)=NENN(K,2)-1
  7        NENN(K,4)=NENN(K,1)-1
  1  10    CONTINUE
  1  C
  7        NEL1=NETP1*NELF2
  7        DO 20 J=1,NETP2
  7        DO 20 I=1,NERGT
  7        K=(J-1)*NERGT+I+NEL1
  7        NENN(K,1)=(J-1)*(NERGT+1)+I+1+NPT1
  7        NENN(K,2)=J*(NERGT+1)+I+1+NPT1
  7        NENN(K,3)=NENN(K,2)-1
  7        NENN(K,4)=NENN(K,1)-1
  1  20    CONTINUE
  1  C
  7        DO 40 I=1,NETP1
  7        K=(I-1)*(NELF2+1)
  7        YY(K+1)=YO
  7        DO 40 J=1,NELF2-1
  7        YY(K+J+1)=(YO-YMIN)*ALOG10(10.*(1.-FLOAT(J)/FLOAT(NELF2+1)))
  7        YY(K+J+2)=YMIN
  1  40    CONTINUE
  1  C
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  7           DY11=(YMAX-YO)/FLOAT(NELF1)
  7           DO 50 I=1,NETP2+1
  7           K=(I-1)*(NERGT+1)
  7           YY(NPT1+K+1)=YMAX
  7           DO 45 I1=1,NELF1-1
  7           YY(NPT1+K+I1+1)=YMAX-DY11*FLOAT(I1)
  1    45     CONTINUE
  7           DO 50 J=1,NELF2+1
  7           YY(NPT1+K+NELF1+J)=YY(J)
  1    50     CONTINUE
  1    C
  7           DO 60 I=1,NELF2+1
  7           XX(I)=XMIN
  1    60     CONTINUE
  1    C
  7           NO=NELF2+1
  1    C
 11               DX12=(XO-XMIN)*.5
 11               NTP22=INT(.5*FLOAT(NETP1-1)+.1)
 11               PINV22=1./FLOAT(NTP22)
 11               PINV33=1./FLOAT(NETP1-NTP22)
  1    C
 11               DO 75 J=2,NETP1
 11               N1=NO
  7           IF(J.EQ.NTP22+1)GOTO 72
 11               IF(J.GT.NTP22)GOTO 73
 11               DUMMY=1.-PINV22*FLOAT(J-1)
 11               DX22=DX12*(1.-ALOG10(10.*DUMMY))
 11               XXOO=XMIN
 11               GOTO 74
  1    72     XXOO=XMIN
  7           DX22=DX12
  7           GOTO 74
  1    73         DUMMY=PINV33*FLOAT(J-1-NTP22)
 11               DX22=DX12*ALOG10(10.*DUMMY)
 11               XXOO=XMIN+DX12
  1    74         DO 75 I=1,NELF2+1
 11               NO=NO+1
 11               XX(N1+I)=XXOO+DX22
  1    75         CONTINUE
  1    C
 11               DX33=XO-XMIN
 11               PINV44=1./FLOAT(NETP1+NTP22)
 11               DO 77 J=1,NETP1
 11               N1=NO
 11               DUMMY=1.-PINV44*FLOAT(J-1)
 11               DX34=DX33*(1.-ALOG10(10.*DUMMY))
 11               DO 77 I=1,NERGT+1
 11               NO=NO+1
 11               XX(N1+I)=XO+DX34
  1    77         CONTINUE
  1    C
 11               NTP44=NETP2-NETP1
 11               DX44=(XMAX-XX(NO))/FLOAT(NTP44+1)
```

```
START
 COL   ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

 11              DO 79 J=1,NTP44+1
 11              N1=NO
 11              DO 79 I=1,NERGT+1
 11              NO=NO+1
 11              XX(N1+I)=XX(N1)+DX44
  1   79         CONTINUE
  1   C
 11              DO 80 I=1,NGPT
 11              PP(I)=0.
 11              UU(I)=0.
 11              VV(I)=0.
  1   80         CONTINUE
 11              DO 90 I=2,NETP1
 11              J=(I-1)*(NELF2+1)
 11              VV(J+1)=-0.01
  1   90         CONTINUE
  1   C
  7         DO 101 I=1,NELF2+1
  7         NBN1(I)=I
  1   101   CONTINUE
  7         DO 102 I=1,NELF1+1
  7         NBN2(I)=NPT1+I
  1   102   CONTINUE
  7         DO 103 I=1,NETP1
  7         J=I+NELF1+1
  7         NBN2(J)=NPT1+1-I*(NELF2+1)
  1   103   CONTINUE
  7         DO 104 I=1,NETP2+1
  7         NBN3(I)=NGPT+1-I*(NERGT+1)
  1   104   CONTINUE
  7         DO 105 I=1,NERGT+1
  7         NBN4(I)=NGPT-I+1
  1   105   CONTINUE
  7         NN=NELF2+1
  7         NBN5(1)=NN
  7         DO 106 I=2,NEBOT+1
  7         IF(I.GT.NETP1)NN=NERGT+1
  7         NBN5(I)=NBN5(I-1)+NN
  1   106   CONTINUE
  1   C
  7         N2=0
  7         N3=0
  7         DO 110 I=1,NELF2+1
  7         N2=N2+1
  7         N3=N3+1
  7         NU(I)=NBN1(I)
  7         NV(I)=NBN1(I)
  1   110   CONTINUE
  1   C
  7         N4=N2
  7         DO 111 I=1,NEBOT+1
  7         N2=N2+1
  7         NV(N4+I)=NBN5(I)
  1   111   CONTINUE
```

```
   1 C
   7       N4=N2
   7       DO 112 I=1,NERGT+1
   7       N2=N2+1
   7       NV(N4+I)=NBN4(I)
   7       ND(I)=NBN4(I)
   1  112  CONTINUE
   1 C
   7       N4=N2
   7       N5=N3
   7       DO 113 I=1,NETP2+1
   7       N2=N2+1
   7       N3=N3+1
   7       NV(N4+I)=NBN3(I)
   7       NU(N5+I)=NBN3(I)
   1  113  CONTINUE
   1 C
   7       N6=NELF1+NETP1+1
   7       DO 114 I=1,N6
   7       NV(N2+I)=NBN2(I)
   7       NU(N3+I)=NBN2(I)
   1  114  CONTINUE
   1 C
   7       DO 116 I=1,NUX
   7       J=NU(I)
   7       UB(I)=UU(J)
   1  116  CONTINUE
   1 C
   7       DO 117 I=1,NVY
   7       J=NV(I)
   7       VB(I)=VV(J)
   1  117  CONTINUE
   1 C
   7       DO 118 I=1,NDP
   7       J=ND(I)
   7       PB(I)=PP(J)
   1  118  CONTINUE
   1 C
   7       DO 120 J=1,NATP1
   7       DO 120 I=1,NALF2
   7       K=(J-1)*NALF2+I
   7       NNZ=0
   7       IF(J.EQ.NATP1)NNZ=NALF1
   7       NENL(K,1)=(J-1)*(NALF2+1)+I+1
   7       NENL(K,2)=J*(NALF2+1)+I+1+NNZ
   7       NENL(K,3)=NENL(K,2)-1
   7       NENL(K,4)=NENL(K,1)-1
   1  120   CONTINUE
   1 C
   7       NAL1=NATP1*NALF2
   7       DO 121 J=1,NATP2
   7       DO 121 I=1,NARGT
   7       K=(J-1)*NARGT+I+NAL1
   7       NENL(K,1)=(J-1)*(NARGT+1)+I+1+NPT1A
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  7          NENL(K,2)=J*(NARGT+1)+I+1+NPT1A
  7          NENL(K,3)=NENL(K,2)-1
  7          NENL(K,4)=NENL(K,1)-1
  1    121    CONTINUE
  1    C
  1    C :::::::::::::::::::::::::::::::::::::::::::::::::
  7          YDY=(YO-YMIN)/FLOAT(NALF2)
  7          ICON(1)=1
  7          ICON(NALF2+1)=NELF2+1
  7          DO 525 I=2,NALF2
  7          YDY1=(NALF2+1-I)*YDY
  7          DO 520 J=2,NELF2
  7          YINC1=YDY1-YY(J)+YMIN
  7          YINC2=YDY1-YY(J+1)+YMIN
  7          IF(YINC2.LT.O.)GOTO 520
  7          IF(ABS(YINC2).GE.ABS(YINC1))GOTO 521
  7          IF(ABS(YINC2).LT.ABS(YINC1))GOTO 522
  1    520   CONTINUE
  1    521   ICON(I)=J
  7          GOTO 525
  1    522   ICON(I)=J+1
  1    525   CONTINUE
  7          IF(NALF2.LE.2)GOTO 526
  7          DO 526 I=3,NALF2
  7          IF(ICON(I).EQ.ICON(I+1))ICON(I)=ICON(I)-1
  1    526   CONTINUE
  1    C
  7          XMID=(XO-XMIN)/2.
  7          XDX=(XO-XMIN)/FLOAT(NATP1)
  7          DO 535 I=2,NATP1
  7          XDX1=(I-1)*XDX
  7          DO 530 J=1,NETP1-1
  7          JJ1=J*(NELF2+1)+1
  7          JJ2=(J+1)*(NELF2+1)+1
  7          XINC1=XX(JJ1)-XDX1-XMIN
  7          XINC2=XX(JJ2)-XDX1-XMIN
  7          IF(XINC2.LT.O.)GOTO 530
  7          IF(ABS(XINC2).GE.ABS(XINC1))GOTO 531
  7          IF(ABS(XINC2).LT.ABS(XINC1))GOTO 532
  1    530   CONTINUE
  1    531   JJ=JJ1
  7          GOTO 533
  1    532   JJ=JJ2
  1    533   IF(NATP1.LE.2)GOTO 534
  7          LL=-1
  7          IF(XX(JJ).GT.XMID)LL=1
  7          JK=(I-2)*(NALF2+1)+1
  7          IF(ICON(JK).EQ.JJ)JJ=JJ+LL*(NELF2+1)
  1    534   ICON((I-1)*(NALF2+1)+1)=JJ
  7          DO 535 K=2,NALF2+1
  7          KK=(I-1)*(NALF2+1)+K
  7          ICON(KK)=ICON(K)+JJ-1
  1    535   CONTINUE
  1    C
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  7          ICON(NPT1A+1)=NPT1+1
  7          ICON(NPT1A+NALF1+1)=NPT1+NELF1+1
  7          YDY2=(YMAX-YO)/FLOAT(NALF1)
  7          DO 545 I=2,NALF1
  7          YDY3=(NALF1+1-I)*YDY2
  7          DO 540 J=1,NELF1-1
  7          YINC3=YDY3-YY(NPT1+J)+YO
  7          YINC4=YDY3-YY(NPT1+J+1)+YO
  7          IF(YINC4.LT.O.)GOTO 540
  7          IF(ABS(YINC4).GE.ABS(YINC3))GOTO 541
  7          IF(ABS(YINC4).LT.ABS(YINC3))GOTO 542
  1    540   CONTINUE
  1    541   ICON(NPT1A+I)=NPT1+J
  7          GOTO 545
  1    542   ICON(NPT1A+I)=NPT1+J+1
  1    545   CONTINUE
  7          IF(NALF1.LE.2)GOTO 546
  7          DO 546 I=2,NALF1
  7          J=NPT1A+I
  7          IF(ICON(J).EQ.ICON(J+1))ICON(J)=ICON(J)-1
  1    546   CONTINUE
  7          DO 547 I=2,NALF2+1
  7          ICON(NPT1A+NALF1+I)=NPT1+NELF1+ICON(I)
  1    547   CONTINUE
  1    C
  7          XDX2=(XMAX-XO)/FLOAT(NATP2)
  7          DO 555 I=2,NATP2
  7          XDX3=(I-1)*XDX2
  7          DO 550 J=2,NETP2
  7          JJ3=NPT1+J*(NERGT+1)+1
  7          JJ4=NPT1+(J+1)*(NERGT+1)+1
  7          XINC3=XX(JJ3)-XDX3-XO
  7          XINC4=XX(JJ4)-XDX3-XO
  7          IF(XINC4.LT.O.)GOTO 550
  7          IF(ABS(XINC4).GE.ABS(XINC3))GOTO 551
  7          IF(ABS(XINC4).LT.ABS(XINC3))GOTO 552
  1    550   CONTINUE
  1    551   JJ=JJ3
  7          GOTO 553
  1    552   JJ=JJ4
  1    553   IF(NATP2.LE.2)GOTO 554
  7          JK=NPT1A+(I-1)*(NARGT+1)
  7          IF(ICON(JK).EQ.JJ)JJ=JJ-(NERGT+1)
  1    554   ICON(NPT1A+(I-1)*(NARGT+1)+1)=JJ
  7          DO 555 K=2,NARGT+1
  7          KK=NPT1A+(I-1)*(NARGT+1)+K
  7          ICON(KK)=ICON(NPT1A+K)+JJ-NPT1-1
  1    555   CONTINUE
  1    C
  7          ICON(NT)=NGPT
  7          ICON(NT-NARGT)=NGPT-NERGT
  7          DO 556 I=2,NARGT
  7          J=NT-(NARGT+1)+I
  7          JDD=ICON(NPT1A+I)-ICON(NPT1A+I-1)
```

```
START
 COL   ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  7          ICON(J)=ICON(J-1)+JDD
  1    556   CONTINUE
  1    C
  1    C ::::::::::::::::::::::::::::::::::::::::::::::::
  7          J1=0
  7          DO 320 J=2,NATP1
  7          DO 320 I=1,NALF2+1
  7          J1=J1+1
  7          NODE(J1)=(J-1)*(NALF2+1)+I
  1    320   CONTINUE
  7          DO 321 I=1,NALF2
  7          J1=J1+1
  7          NODE(J1)=NPT1A+NALF1+1+I
  1    321   CONTINUE
  7          DO 322 J=2,NATP2+1
  7          DO 322 I=2,NARGT+1
  7          J1=J1+1
  7          J2=0
  7          IF(I.EQ.2)J2=1
  7          NODE(J1)=NODE(J1-1)+1+J2
  1    322   CONTINUE
  7          DO 323 I=1,2
  7          DO 323 J=1,NT1
  7          K=I*NT1+J
  7          NODE(K)=NODE(J)+I*NT
  1    323   CONTINUE
  7          DO 324 I=1,NT
  7          J=3*NT1+I
  7          NODE(J)=NODE(J-1)+1
  1    324   CONTINUE
  1    C
  1    C
  7          DO 350 I=1,6
  7          FFNX(I)=0.
  7          FFNY(I)=0.
  7          ABN(I)=0.
  7          ANN(I)=0.
  1    350   CONTINUE
  7          NBQ(1)=NALF2
  7          NBQ(2)=NABOT
  7          NBQ(3)=NARGT
  7          NBQ(4)=NATP2
  7          NBQ(5)=NALF1
  7          NBQ(6)=NATP1
  7          DO 360 I=1,6,2
  7          FFNX(I)=1.
  7          FFNY(I+1)=1.
  1    360   CONTINUE
  7          ABN(6)=1.
  7          ANN(3)=1.
  1    C
  1    C
  7          DO 371 J=1,NBQ(1)
  7          NC(1,J,1)=J
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
  7          NC(1,J,2)=J+1
  1    371   CONTINUE
  1    C
  7          K=0
  7          NBDQ2=NALF2+1
  7          DO 372 J=1,NBQ(2)
  7          K=K+NBDQ2
  7          IF(J.GE.NATP1)NBDQ2=NARGT+1
  7          NC(2,J,1)=K
  7          NC(2,J,2)=K+NBDQ2
  1    372   CONTINUE
  1    C
  7          DO 373 J=1,NBQ(3)
  7          NC(3,J,1)=NT+1-J
  7          NC(3,J,2)=NT-J
  1    373   CONTINUE
  1    C
  7          DO 374 J=1,NBQ(4)
  7          NC(4,J,1)=NT-J*(NARGT+1)+1
  7          NC(4,J,2)=NC(4,J,1)-(NARGT+1)
  1    374   CONTINUE
  1    C
  7          DO 375 J=1,NBQ(5)
  7          NC(5,J,1)=NPT1A+J
  7          NC(5,J,2)=NPT1A+J+1
  1    375   CONTINUE
  1    C
  7          DO 376 J=1,NBQ(6)
  7          NBDQ6=0
  7          IF(J.EQ.1)NBDQ6=NALF1
  7          NC(6,J,1)=NPT1A+1-(J-1)*(NALF2+1)+NBDQ6
  7          NC(6,J,2)=NPT1A+1-J*(NALF2+1)
  1    376   CONTINUE
  1    C         PRINT INPUT DATA
  1    C
 11              WRITE(6,2000)
 11              WRITE(6,2001) REN,GAMMA
 11              WRITE(6,2002) DT,ERROR,ITMAX
 11              WRITE(6,2003) NEL,NGPT,NPT,NBW,NBW1,NBWT
  1    C
 11              WRITE(6,2004)
 11              DO 1  I=1,NEL
 11              WRITE(6,1003) (NENN(I,J),J=1,NPT)
  1    1         CONTINUE
  1    C
 11              WRITE(6,2005)
 11              WRITE(6,2006)
 11              DO 11 I=1,NGPT
 11              WRITE(6,1004) I,XX(I),YY(I),UU(I),VV(I),PP(I)
  1    11        CONTINUE
  1    C
 11              WRITE(6,2007)
 11              DO 12 I=1,NUX
 11              WRITE(6,1005) NU(I),UB(I)
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  1   12       CONTINUE
 11            WRITE(6,2008)
 11            DO 13 I=1,NVY
 11            WRITE(6,1005) NV(I),VB(I)
  1   13       CONTINUE
 11            WRITE(6,2009)
 11            DO 14 I=1,NDP
 11            WRITE(6,1005) ND(I),PB(I)
  1   14       CONTINUE
  1   C
 11            WRITE(6,2010)
 11            DO 15 I=1,NTT
 11            WRITE(6,1006) I,NODE(I)
  1   15       CONTINUE
  1   C
 11            WRITE(6,2011)
 11            WRITE(6,2012)
 11            DO 16 I=1,NBP
 11            WRITE(6,1007) I,NBQ(I),FFNX(I),FFNY(I),ABN(I),ANN(I)
  1   16       CONTINUE
  1   C
 11            WRITE(6,2013)
 11            DO 17 I=1,NBP
 11            NB=NBQ(I)
 11            DO 17 J=1,NB
 11            WRITE(6,1008) (NC(I,J,K),K=1,2)
  1   17       CONTINUE
  1   C
  1   1003     FORMAT(4I5)
  1   1004     FORMAT(I5,5F10.5)
  1   1005     FORMAT(I5,F10.5)
  1   1006     FORMAT(2I5)
  1   1007     FORMAT(2I5,4F10.5)
  1   1008     FORMAT(2I5)
 11            RETURN
  1   2000     FORMAT(///,10X,'INPUT DATA',///)
  1   2001     FORMAT(5X,'REN=',E10.5,5X,'GAMMA=',F10.5)
  1   2002     FORMAT(5X,'DT=',F10.5,2X,'ERROR=',F10.5,2X,'ITMAX=',I5)
  1   2003     FORMAT(5X,'NEL=',I5,2X,'NGPT=',I5,2X,'NPT=',I5,2X,
  6        *           'NBW=',I5,2X,'NBW1=',I5,2X,'NBWT=',I5)
  1   2004     FORMAT(//,10X,'ELEMENT CONNECTIVITY MATRIX',/)
  1   2005     FORMAT(//,10X,'COORDINATE VALUE AT EACH GLOBAL NODE',/)
  1   2006     FORMAT(1X,'NO',7X,'XX',7X,'YY',7X,'UU',7X,'VV',7X,'PP',/)
  1   2007     FORMAT(//,10X,'BOUNDARY CONDITIONS FOR MEAN FLOW FIELDS',/)
  1   2008     FORMAT(/)
  1   2009     FORMAT(/)
  1   2010     FORMAT(//,10X,'ADJUSTMENT FOR BOUNDARY CONDITIONS',/)
  1   2011     FORMAT(//,10X,'NORMAL VELCTORS AND ADMITTANCES',/)
  1   2012     FORMAT(1X,'NO',3X,'NEQ',6X,'FFNX',6X,'FFNY',7X,'ABN',
  6        *           7X,'ANN',/)
  1   2013     FORMAT(//,10X,'BOUNDARY ELEMENT CONNECTIVITY MATRIX',/)
 11            END
  1   C
  1   C
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  1   C
 11            SUBROUTINE VELOT(NEL,NGPT,NPT,NBW1,NBWT,XX,YY,
  6        *                    UU,VV,PP,REN,DT,ERROR,
  6        * NENN,ICON,XT,YT,UT,VT,NBO,US,VS,DPP,DELU,DELV,FU,FV,
  6        *         A,B,NUX,NVY,NDP,NU,NV,ND,UB,VB,PB,NT,ITMAX)
  1   C
  1   C        VIRTUAL
  1   C
  1   C        SUBROUTINE FOR MEAN VELOCITY CALCULATIONS
  1   C
  1   C        A(NGPT,NBWT)     : GLOBAL BAND STIFFNESS MATRIX
  1   C        B(NGPT,NBWT)     : GLOBAL BAND STIFFNESS MATRIX
  1   C        FN(NGPT)         : GLOBAL FORCE MATRIX
  1   C        FV(NGPT)         : GLOBAL FORCE MATRIX
  1   C        US(NGPT)         : ESTIMATED U-VELOCITY
  1   C        VS(NGPT)         : ESTIMATED V-VELOCITY
  1   C        DPP(NGPT)        : PRESSURE CORRECTIONS
  1   C        DELU(NGPT)       : ACCELERATIONS IN X-DIRECTION
  1   C        DELV(NGPT)       : ACCELERATIONS IN Y-DIRECTION
  1   C        ST(4)            : GAUSSIAN POINTS OF GAUSSIAN QUADRTURE
  1   C        WS(4)            : WEIGHTING FUNCTIONS OF GAUSSIAN QUADRATURE
  1   C        ANM(NPT,NPT)     : LOCAL STIFFNESS MATRIX
  1   C        BNM(NPT,NPT)     : LOCAL STIFFNESS MATRIX
  1   C        CNM(NPT,NPT)     : LOCAL STIFFNESS MATRIX
  1   C        FNU(NPT)         : LOCAL FORCE MATRIX
  1   C        FNV(NPT)         : LOCAL FORCE MATRIX
  1   C        GN(NPT)          : LOCAL FORCE MATRIX
  1   C        HNU(NPT)         : LOCAL FORCE MATRIX
  1   C        HNV(NPT)         : LOCAL FORCE MATRIX
  1   C        SUM              : CONVERGENCE ERROR IN EACH ITERATION
  1   C        ITER             : COUNTER OF ITERATION
  1   C
 11            DIMENSION XX(NGPT),YY(NGPT),UU(NGPT),VV(NGPT),PP(NGPT)
 11            DIMENSION UB(NUX),VB(NVY),PB(NDP),WS(4),ST(4)
 11            DIMENSION NU(NUX),NV(NVY),ND(NDP)
  7        DIMENSION XT(NT),YT(NT),UT(NT),VT(NT)
 11            DIMENSION US(NGPT),VS(NGPT),DPP(NGPT),DELU(NGPT),DELV(NGPT)
 11            DIMENSION ANM(4,4),BNM(4,4),CNM(4,4),FNU(4),FNV(4),GN(4),
  6        *             HNU(4),HNV(4),A(NGPT,NBO),B(NGPT,NBO),
  6        *             FU(NGPT),FV(NGPT),NENN(NEL,4),ICON(NT)
  1   C
  1   C        CALL THE VALUES FOR GAUSSIAN QUADRATURE INTEGRATIONS
  1   C
 11            CALL GAUSS(4,WS,ST)
  1   C
 11            ITER=0
  1   C
  1   2000     CONTINUE
  1   C
 11            ITER=ITER+1
  1   C
  1   C        ESTIMATED VELOCITIES
  1   C
 11            DO 100 I=1,NGPT
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

 11            FU(I)=0.0
 11            FV(I)=0.0
 11            DO 100 J=1,NBWT
 11            A(I,J)=0.0
 11            B(I,J)=0.0
  1   100      CONTINUE
  1   C
 11            DO 200 I=1,NEL
 11            CALL ELEUV(I,NPT,NGPT,XX,YY,UU,VV,PP,REN,DT,NENN,
  6        *             ANM,FNU,FNV,WS,ST,NEL)
 11            DO 200 J=1,NPT
 11            JJ=NENN(I,J)
 11            FU(JJ)=FU(JJ)+FNU(J)
 11            FV(JJ)=FV(JJ)+FNV(J)
 11            DO 200 K=1,NPT
 11            KK=NENN(I,K)
 11            KKJJ=KK-JJ+NBW1
 11            A(JJ,KKJJ)=A(JJ,KKJJ)+ANM(J,K)
 11            B(JJ,KKJJ)=A(JJ,KKJJ)
  1   200      CONTINUE
  1   C
 11            CALL ADJUST(NUX,NGPT,NU,A,UB,FU,NBW1,NBWT)
 11            CALL ADJUST(NVY,NGPT,NV,B,VB,FV,NBW1,NBWT)
  1   C
 11            CALL GAUSU(A,FU,US,NGPT,NBW1,NBWT)
 11            CALL GAUSU(B,FV,VS,NGPT,NBW1,NBWT)
  1   C
  1   C        PRESSURE CORRECTIONS
  1   C
 11            DO 300 I=1,NGPT
 11            FU(I)=0.0
 11            DO 300 J=1,NBWT
 11            A(I,J)=0.0
  1   300      CONTINUE
  1   C
 11            DO 400 I=1,NEL
 11            CALL ELEPR(I,NPT,NGPT,XX,YY,US,VS,DT,BNM,GN,
  6        *             WS,ST,NEL,NENN)
 11            DO 400 J=1,NPT
 11            JJ=NENN(I,J)
 11            FU(JJ)=FU(JJ)+GN(J)
 11            DO 400 K=1,NPT
 11            KK=NENN(I,K)
 11            KKJJ=KK-JJ+NBW1
 11            A(JJ,KKJJ)=A(JJ,KKJJ)+BNM(J,K)
  1   400      CONTINUE
  1   C
 11            CALL ADJUST(NDP,NGPT,ND,A,PB,FU,NBW1,NBWT)
  1   C
 11            CALL GAUSU(A,FU,DPP,NGPT,NBW1,NBWT)
  1   C
  1   C        ACCELERATIONS
  1   C
 11            DO 500 I=1,NGPT
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

 11            FU(I)=0.0
 11            FV(I)=0.0
 11            DO 500 J=1,NBWT
 11            A(I,J)=0.0
 11            B(I,J)=0.0
  1   500      CONTINUE
  1   C
 11            DO 600 I=1,NEL
 11            CALL ELEAC(I,NPT,NGPT,XX,YY,DPP,CNM,HNU,HNV,
  6                      WS,ST,NEL,NENN)
 11            DO 600 J=1,NPT
 11            JJ=NENN(I,J)
 11            FU(JJ)=FU(JJ)+HNU(J)
 11            FV(JJ)=FV(JJ)+HNV(J)
 11            DO 600 K=1,NPT
 11            KK=NENN(I,K)
 11            KKJJ=KK-JJ+NBW1
 11            A(JJ,KKJJ)=A(JJ,KKJJ)+CNM(J,K)
 11            B(JJ,KKJJ)=A(JJ,KKJJ)
  1   600      CONTINUE
  1   C
 11            CALL GAUSU(A,FU,DELU,NGPT,NBW1,NBWT)
 11            CALL GAUSU(B,FV,DELV,NGPT,NBW1,NBWT)
  1   C
  1   C        CORRECTIONS OF PRESSURE AND VELOCITIES
  1   C
 11            DO 700 I=1,NGPT
 11            PP(I)=PP(I)+DPP(I)
 11            UU(I)=US(I)+DELU(I)*DT
 11            VV(I)=VS(I)+DELV(I)*DT
  1   700      CONTINUE
  1   C
  1   C        CALCULATOINS OF CONVERGENCE ERROR
  1   C
 11            SUM1=0.0
 11            SUM2=0.0
 11            DO 800 I=1,NGPT
 11            SUM1=SUM1+(DELU(I)*DT)**2
 11            SUM2=SUM2+UU(I)**2
  1   800      CONTINUE
 11            SUM=SQRT(SUM1/SUM2)
  1   C
 11            IF(ITER.GT.ITMAX) GO TO 3000
 11            IF(SUM.GT.ERROR) GO TO 2000
  1   C
  1   3000     CONTINUE
  1   C
 11            DO 900 I=1,NGPT
 11            UU(I)=US(I)
 11            VV(I)=VS(I)
  1   900      CONTINUE
  1   C
 11            DO 910 I=1,NT
 11            K=ICON(I)
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

 11             XT(I)=XX(K)
 11             YT(I)=YY(K)
 11             UT(I)=UU(K)
 11             VT(I)=VV(K)
  1   910       CONTINUE
  1   C
  1   C         OUTPUT FOR CONVERGED MEAN VELOCITY FIELDS
  1   C
 11             WRITE(6,5000) ITER,SUM
 11             WRITE(6,5010)
 11             DO 1200 I=1,NGPT
 11             WRITE(6,5020) I, XX(I),YY(I),UU(I),VV(I),PP(I)
  1   1200      CONTINUE
  1   C
 11             RETURN
  1   5000      FORMAT(/////,10X,'ITER=',2X,I5,5X,'ERROR=',2X,E15.8,//)
  1   5010      FORMAT(8X,'NO',4X,'XX',8X,'YY',14X,'UU',13X,'VV',
  6        *             13X,'PP')
  1   5020      FORMAT(5X,I5,2F10.5,3E15.5)
 11             END
  1   C
  1   C
 11             SUBROUTINE EIPRE(NL,NT,NPT,NENL,FRE,PRESS,
  6        * XT,YT,TARR1,A1,B1,CZZ1,EIGAV,EIGBV,EIGV,ITER)
  1   C         VIRTUAL
  1   C
  1   C         SUBROUTINE FOR ACOUSTIC MODES
  1   C
  1   C         FRE(NGPT)        : ACOUSTIC FREQUENCIES
  1   C         PRESS(NGPT,NGPT): ACOUSTIC MODES
  1   C         A1(NGPT,NGPT)     : LEFT-HAND-SIDE GLOBAL EIGENMATRIX
  1   C         B1(NGPT,NGPT)     : RIGHT-HAND-SIDE GLOBAL EIGENMATRIX
  1   C         CZZ1(NGPT,NGPT) : SOLUTIONS OF EIGENVECTORS
  1   C         EIGAV(NGPT)      : SOLUTIONS OF EIGENVECTORS
  1   C         EIGBV(NGPT)      : SOLUTIONS OF EIGENVECTORS
  1   C         EIGV(NGPT)       : EIGAV(NGPT)/EIGBV(NGPT)
  1   C         ANM(NPT,NPT)      : LEFT-HAND-SIDE LOCAL EIGENMATRIX
  1   C         BNM(NPT,NPT)      : RIGHT-HAND-SIDE LOCAL EIGENMATRIX
  1   C
  7          DIMENSION XT(NT),YT(NT)
 11             DIMENSION TARR1(NT),WS(4),ST(4),ANM(4,4),BNM(4,4)
  6        *,    NENL(NL,4),FRE(NT),PRESS(NT,NT)
 11             COMPLEX    A1(NT,NT),B1(NT,NT),CZZ1(NT,NT),EIGAV(NT),
  6        *               EIGBV(NT),EIGV(NT)
 11             INTEGER    ITER(NT)
  1   C
 11             CALL GAUSS(4,WS,ST)
  1   C
  1   C         INITIALIZATION OF ALL GLOBAL MATRICES
  1   C
 11             DO 190 I=1,NT
 11             DO 190 J=1,NT
 11             A1(I,J)=(0.0,0.0)
 11             B1(I,J)=(0.0,0.0)
```

```
     PROJECT: CTJC197            MEMBER: MAIN2             DATE: 87/09/24           MAIN2
     GROUP:   STB1               LEVEL:  01.00             TIME: 12:02
     TYPE:    FORT               USERID: CTJC197           PAGE: 18 OF 50
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  1   190       CONTINUE
  1   C
  1   C         ASSEMBLY OF GLOBAL MATRICES
  1   C
 11             DO 200 I=1,NL
 11             CALL ELEMP(I,NPT,ANM,BNM,WS,ST,NL,NT,NENL,
  6        *              XT,YT)
  1   C
 11             DO 200 J=1,NPT
 11             JJ=NENL(I,J)
 11             DO 200 K=1,NPT
 11             KK=NENL(I,K)
 11             A1(JJ,KK)=A1(JJ,KK)+CMPLX(ANM(J,K),0.0)
 11             B1(JJ,KK)=B1(JJ,KK)+CMPLX(BNM(J,K),0.0)
  1   200       CONTINUE
  1   C
  1   C         APPLY EIGENVALUE SUBROUTINE IN IMSL
  1   C
 11             CALL CONVRT(NT,A1,NT,B1,NT,CZZ1,NT)
 11             CALL SOLVE(NT,A1,NT,B1,NT,CZZ1,NT,ITER,EIGAV,EIGBV)
  1   C
  1   C         OBTAIN EIVENVALUES AND EIGENVECTORS
  1   C
 11             DO 220 IEG=1,NT
 11             IF(EIGBV(IEG).EQ.0.0) GO TO 220
 11             EIGV(IEG)=EIGAV(IEG)/EIGBV(IEG)
 11             FRE(IEG)=SQRT(ABS(REAL(EIGV(IEG))))
 11             DO 221 IEF=1,NT
 11             PRESS(IEF,IEG)=REAL(CZZ1(IEF,IEG))
  1   221       CONTINUE
  1   220       CONTINUE
  1   C
  1   C         SORTING PROCESS
  1   C
 11             NN=NT-1
 11             DO 4000 K=1,NN
 11             JJ=NT-K
 11             DO 410 L=1,JJ
 11             IF(FRE(L).LE.FRE(L+1)) GO TO 410
 11             TEMP=FRE(L)
 11             FRE(L)=FRE(L+1)
 11             FRE(L+1)=TEMP
 11             DO 420 NP=1,NT
 11             TARR1(NP)=PRESS(NP,L)
 11             PRESS(NP,L)=PRESS(NP,L+1)
 11             PRESS(NP,L+1)=TARR1(NP)
  1   420       CONTINUE
  1   410       CONTINUE
  1   4000      CONTINUE
  1   C
  1   C         OUTPUT FOR ACOUSTIC FREQUENCIES AND THEIR MODES
  1   C
 11             WRITE(6,1000)
 11             DO 500 I=1,NT
```

```
 11             WRITE(6,1001) I,FRE(I)
  1  500        CONTINUE
 11             WRITE(6,1002)
 11             DO 501 I=1,NT
 11             WRITE(6,1003) I-1,FRE(I)
 11             WRITE(6,1013)
 11             DO 502 J=1,NT
 11             WRITE(6,1004) J,XT(J),YT(J),PRESS(J,I)
  1  502        CONTINUE
 11             WRITE(6,1005)
  1  501        CONTINUE
  1  C
 11             RETURN
  1  1000       FORMAT(//,10X,'ACOUSTIC FREQUENCIES',/)
  1  1001       FORMAT(15X,I5,E15.5)
  1  1002       FORMAT(//,10X,'LOWEST TWENTY ACOUSTIC MODES',/)
  1  1003       FORMAT(10X,I5,'-TH ACOUSTIC MODE',5X,'FRE = ',E15.5)
  1  1013       FORMAT(8X,'NO',4X,'XX',8X,'YY',14X,'PRESS')
  1  1004       FORMAT(5X,I5,2F10.5,E15.5)
  1  1005       FORMAT(//)
 11             END
  1  C
  1  C
  1  C
 11             SUBROUTINE EIVOR(NL,NT,NPT,REN,NTT,NODE,MVP,MI,MII,'
  6        * SR,SI,FV1,A2,B2,C,D,CZZ2,EIGAV2,EIGBV2,EIGV2,ITER2,
  6        * NENL,FVE,UUR,UUI,VVR,VVI,XT,YT,UT,VT)
  1  C          VIRTUAL
  1  C
  1  C          SUBROUTINE FOR VORTICAL MODES
  1  C
  1  C          MVP              : NUMBER OF USEFUL VORTICAL EIGENMODES
  1  C          FVE(MVP)         : VORTICAL  FREQUENCIES
  1  C          UUR(NT,MVP)      : REAL PART OF VORTICAL DISTURBANCES OF
  1  C                             U-VELOCITY
  1  C          VVR(NT,MVP)      : REAL PART OF VORTICAL DISTURBANCES OF
  1  C                             V-VELOCITY
  1  C          UUI(NT,MVP)      : IMAGINARY PART OF VORTICAL DISTURBANCES OF
  1  C                             U-VELOCITY
  1  C          VVI(NT,MVP)      : IMAGINARY PART OF VIRTICAL DISTURBANCES OF
  1  C                             V-VELOCITY
  1  C          C(NTT,NTT)       : LEFT-HAND-SIDE GLOBAL EIGENMATRIX
  1  C          D(NTT,NTT)       : RIGHT-HAND-SIDE GLOBAL EIGENMATRIX
  1  C          CZZ2(NTT,NTT)    : SOLUTIONS OF EIGENVECTORS
  1  C          EIGAV2(NTT)       : SOLUTIONS OF EIGENVALUES
  1  C          EIGBV2(NTT)       : SOLUTIONS OF EIGENVALUES
  1  C          EIGV2(NTT)        : EIGAV2(NTT)/EIGBV2(NTT)
  1  C          SR(4*NT,4*NT)    : REAL PART OF CZZ2(NTT,NTT)
  1  C          SI(4*NT,4*NT)    : IMAGINARY PART OF CZZ2(NTT,NTT)
  1  C          ANM(4,4,4,4)     : LEFT-HAND-SIDE OF LOCAL EIGENMATRIX
  1  C          BNM(4,4,4,4)     : RIGHT-HAND-SIDE OF LOCAL EIGENMATRIX
  1  C
  7        DIMENSION XT(NT),YT(NT),UT(NT),VT(NT)
 11             DIMENSION SR(MII,NTT),SI(MII,NTT),NODE(NTT)
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

 11            DIMENSION ANM(4,4,4,4),BNM(4,4,4,4),FV1(MII),NENL(NL,4),
  6        * FVE(MI),UUR(NT,MI),UUI(NT,MI),VVR(NT,MI),VVI(NT,MI)
 11            COMPLEX A2(MII,MII),B2(MII,MII),C(NTT,NTT),D(NTT,NTT),
  6        *           CZZ2(NTT,NTT),EIGAV2(NTT),EIGBV2(NTT),EIGV2(NTT)
 11            INTEGER   ITER2(NTT)
  1   C
 11            CALL GAUSS(4,WS,ST)
  1   C
  1   C        INITIALIZATION OF GLOBAL MATRICES
  1   C
  7        NT4=NT*4
  7        NT2=NT*2
 11            DO 1 I=1,NT4
 11            DO 1 J=1,NT4
 11            A2(I,J)=(0.0,0.0)
 11            B2(I,J)=(0.0,0.0)
  1   1        CONTINUE
  1   C
 11            DO 111 I=1,NTT
 11            DO 111 J=1,NT
 11            C(I,J)=(0.0,0.0)
 11            D(I,J)=(0.0,0.0)
  1   111      CONTINUE
  1   C
 11            DO 112 I=1,NT4
 11            DO 112 J=1,NTT
 11            SR(I,J)=0.0
 11            SI(I,J)=0.0
  1   112      CONTINUE
  1   C
  1   C        ASSEMBLY OF GLOBAL MATRICES
  1   C
 11            DO 2 I=1,NL
 11            CALL ELEVR(I,NPT,ANM,BNM,WS,ST,REN,NL,NT,NENL,
  6        *           XT,YT,UT,VT)
 11            DO 2 L=1,4
  7        LNT=(L-1)*NT
 11            DO 2 J=1,NPT
 11            JJ=NENL(I,J)+LNT
 11            DO 2 LL=1,4
  7        LLNT=(LL-1)*NT
 11            DO 2 K=1,NPT
 11            KK=NENL(I,K)+LLNT
 11            A2(JJ,KK)=A2(JJ,KK)+CMPLX(ANM(L,LL,J,K),0.0)
 11            B2(JJ,KK)=B2(JJ,KK)+CMPLX(BNM(L,LL,J,K),0.0)
  1   2        CONTINUE
  1   C
  1   C        APPLY BOUNDARY CONDITIONS
  1   C
 11            DO 20 I=1,NTT
 11            DO 20 J=1,NTT
 11            C(I,J)=A2(NODE(I),NODE(J))
 11            D(I,J)=B2(NODE(I),NODE(J))
  1   20       CONTINUE
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  1  C
 11           CALL CONVRT(NTT,C,NTT,D,NTT,CZZ2,NTT)
 11           CALL SOLVE(NTT,C,NTT,D,NTT,CZZ2,NTT,ITER2,EIGAV2,EIGBV2)
  1  C
  1  C        OBTAIN EIGENVALUES AND CORRESPONDING EIGENMODES
  1  C
 11           DO 3 IEG=1,NTT
 11           IF(EIGBV2(IEG) .EQ. 0.0) GO TO 3
 11           EIGV2(IEG)=EIGAV2(IEG)/EIGBV2(IEG)
 11           FV1(IEG)=AIMAG(EIGV2(IEG))
 11           DO 4 IEF=1,NTT
 11           SR(NODE(IEF),IEG)=REAL(CZZ2(IEF,IEG))
 11           SI(NODE(IEF),IEG)=AIMAG(CZZ2(IEF,IEG))
  1  4        CONTINUE
  1  3        CONTINUE
  1  C
  1  C        SORTING PROCESS
  1  C
 11           NN=NTT-1
 11           DO 5 K=1,NN
 11           JJ=NTT-K
 11           DO 6 L=1,JJ
 11           IF(FV1(L).LT.FV1(L+1)) GO TO 6
 11           TEMP=FV1(L)
 11           FV1(L)=FV1(L+1)
 11           FV1(L+1)=TEMP
 11           DO 7 NP=1,NT4
 11           TR=SR(NP,L)
 11           TI=SI(NP,L)
 11           SR(NP,L)=SR(NP,L+1)
 11           SI(NP,L)=SI(NP,L+1)
 11           SR(NP,L+1)=TR
 11           SI(NP,L+1)=TI
  1  7        CONTINUE
  1  6        CONTINUE
  1  5        CONTINUE
  1  C
 11           IP=0
 11           DO 8 I=1,NTT
 11           IF(FV1(I).LT.0.01) GO TO 8
 11           IP=IP+1
 11           FVE(IP)=FV1(I)
 11           DO 9 J=1,NT
 11           UUR(J,IP)=SR(NT2+J,I)
 11           UUI(J,IP)=SI(NT2+J,I)
 11           VVR(J,IP)=-SR(NT+J,I)
 11           VVI(J,IP)=-SI(NT+J,I)
  1  9        CONTINUE
  1  8        CONTINUE
  1  C
 11           MVP=IP
  1  C
  1  C        OUTPUT FOR VORTICAL MODES
  1  C
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

 11             WRITE(6,999)
 11             DO 10 I=1,20
 11             WRITE(6,1000) I,FVE(I)
 11             WRITE(6,1010)
 11             DO 11 J=1,NT
 11             WRITE(6,1001) J,XT(J),YT(J),UUR(J,I),UUI(J,I),
  6       *                   VVR(J,I),VVI(J,I)
  1   11        CONTINUE
  1   10        CONTINUE
  1   C
 11             RETURN
  1   999       FORMAT(/////,10X,'LOWEST TWENTY VORTICAL MODES',/)
  1   1000      FORMAT(/////,I5,'-TH VORTICAL MODE',5X,
  6       *            'STROUHAL NO. =',E10.5,/)
  1   1010      FORMAT(3X,'NO',5X,'XT',8X,'YT',9X,'UUR',12X,'UUI',
  6       *            12X,'VVR',12X,'VVI')
  1   1001      FORMAT(I5,2F9.5,4E14.5)
 11             END
  1   C
  1   C
  1   C
 11             SUBROUTINE SURFCE(IK,IG,NBP,FFNX,FFNY,ABN,ANN,NC1,
  6       *             NC,FRE,PRESS,FVE,UUR,UUI,VVR,VVI,
  6       *                        NT,AAA,AAB,
  6       *                        AHB,AHC,REN,GAMMA,NBQ,MI,
  6       *                XT,YT,UT,VT)
  1   C     VIRTUAL
  1   C
  1   C
  1   C         STABILITY INTEGRALS AT THE SURFACES
  1   C
  1   C         IK                : COUNTER OF ACOUSTIC MODES
  1   C         IG                : COUNTER OF VIRTICAL MODES
  1   C         AAA               : (A)-TERM OF ACOUSTIC GROWTH CONSTANT
  1   C         AAB               : (B)-TERM OF ACOUSTIC GROWTH CONSTANT
  1   C         AHB               : (B)-TERM OF VORTICALLY COUPLED ACOUSTIC
  1   C                             GROWTH CONSTANT
  1   C         AHC               : (C)-TERM OF VORTICALLY COUPLED ACOUSTIC
  1   C                             GROWTH CONSTANT
  1   C         X(2)              : LOCAL BOUNDARY X-COORDINATES
  1   C         Y(2)              : LOCAL BOUNDARY Y-COORDINATES
  1   C         U(2)              : LOCAL BOUNDARY MEAN VELOCITY IN X-DIRECTION
  1   C         V(2)              : LOCAL BOUNDARY MEAN VELOCITY IN Y-DIRECTION
  1   C         P(2)              : LOCAL BOUNDARY ACOUSTIC MODES
  1   C         UR(2)             : LOCAL BOUNDARY VALUE OF REAL PART OF VORTICAL
  1   C                             DISTURBANCES IN X-DIRECTION
  1   C         VR(2)             : LOCAL BOUNDARY VALUE OF REAL PART OF VORTICAL
  1   C                             DISTURBANCES IN Y-DIRECTION
  1   C         UI(2)             : LOCAL BOUNDARY VALUE OF IMAGINARY PART OF
  1   C                             VORTICAL DISTURBANCES IN X-DIRECTION
  1   C         VI(2)             : LOCAL BOUNDARY VALUE OF IMAGINARY PART OF
  1   C                             VORTICAL DISTURBANCES IN Y-DIRECTION
  1   C         FI(2)             : INTERPOLATION FUNCTIONS IN BOUNDARY ELEMENT
  1   C         DS(2)             : FIRST DERIVATIVES OF INTERPOLATION FUNCTIONS
```

```
START
 COL   ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  1  C                          IN BOUNDARY ELEMENT
  1  C
  7        DIMENSION XT(NT),YT(NT),UT(NT),VT(NT)
 11            DIMENSION FFNX(NBP),FFNY(NBP),ABN(NBP),ANN(NBP),NBQ(NBP)
 11            DIMENSION U(2),V(2),
  6       *             WS(4),ST(4),X(2),Y(2),P(2),UI(2),VI(2),FI(2),DS(2)
 11            DIMENSION UR(2),VR(2),NC(NBP,NC1,2),FRE(NT),PRESS(NT,NT),
  6       * FVE(MI),UUR(NT,MI),UUI(NT,MI),VVR(NT,MI),VVI(NT,MI)
  1  C
 11            CALL GAUSS(4,WS,ST)
  1  C
 11            FR=FRE(IK)
  7        FRINV=1./FR
  7        FR2INV=FRINV*FRINV
  7        RFRINV=FRINV/REN
 11            FV3=FVE(IG)
  1  C
  1  C         INITIALIZATIONS
  1  C
 11            AAA=0.0
 11            AAB=0.0
 11            AHB=0.0
 11            AHC=0.0
  1  C
  1  C         REPEAT INTEGRATION IN EACH BOUNDARY AREA
  1  C
 11            DO 1 IB=1,NBP
  1  C
 11            IF(IB.EQ.2) GO TO 1
  1  C
  1  C         NECESSARY INFORMATION IN EACH BOUNDARY AREA
  1  C
 11            FNX=FFNX(IB)
 11            FNY=FFNY(IB)
 11            NB = NBQ(IB)
 11            AB = ABN(IB)
 11            AN = ANN(IB)
  1  C
 11            DO 2 IC=1,NB
  1  C
 11            DO 10 N=1,2
 11            NN=NC(IB,IC,N)
 11            X(N)=XT(NN)
 11            Y(N)=YT(NN)
 11            U(N)=UT(NN)
 11            V(N)=VT(NN)
 11            P(N)=PRESS(NN,IK)
 11            UR(N)=UUR(NN,IG)
 11            VR(N)=VVR(NN,IG)
 11            UI(N)=UUI(NN,IG)
 11            VI(N)=VVI(NN,IG)
  1  10        CONTINUE
  1  C
 11            IF(FNX .EQ. 0.0) DTA=0.5*ABS(X(2)-X(1))
```

```
 11            IF(FNY .EQ. 0.0) DTA=0.5*ABS(Y(2)-Y(1))
  7        DTINV5=-.5/DTA
  1  C
 11            DO 100 K=1,4
 11            XI=ST(K)
 11            ACOF=WS(K)
  1  C
  1  C         INTERPOLATION FUNCTIONS AT THE BOUNDARY ELEMENT
  1  C
 11            FI(1)=0.5*(1.0-XI)
 11            FI(2)=0.5*(1.0+XI)
  1  C
 11            DS(1)=DTINV5
 11            DS(2)=-DS(1)
  1  C
 11            YP=0.0
 11            SU=0.0
 11            SV=0.0
 11            PN=0.0
 11            PS=0.0
 11            SUI=0.0
 11            SUIS=0.0
 11            SVIS=0.0
  1  C
 11            DO 110 N=1,2
 11            YP=YP+FI(N)*Y(N)
 11            SU=SU+FI(N)*U(N)
 11            SV=SV+FI(N)*V(N)
 11            PN=PN+FI(N)*P(N)
 11            PS=PS+DS(N)*P(N)
 11            SUI=SUI+FI(N)*UI(N)
 11            SVI=SVI+FI(N)*VI(N)
 11            SUIS=SUIS+DS(N)*UI(N)
 11            SVIS=SVIS+DS(N)*VI(N)
  1  110       CONTINUE
  1  C
 11            C=3.14159*YP*ACOF*DTA
  1  C
 11            AAA=AAA+C*((AB-AN)*PN*PN+(GAMMA+1.0)*(SU*FNX+SV*FNY)*PN*PN)
 11            AAB=AAB-C*(SU*FNX+SV*FNY)*PS*PS*FR2INV
 11            AHB=AHB+C*GAMMA*(2.0*(SU*SUI*PS*FNX+SV*SVI*PS*FNY)
  6       *                  +(SU*SVI+SV*SUI)*PS*(FNX+FNY))*FRINV
 11            AHC=AHC+C*(SVIS*PS*FNX+SUIS*PS*FNY)*RFRINV
  1  100       CONTINUE
  1  C
  1  2         CONTINUE
  1  1         CONTINUE
  1  C
 11            RETURN
 11            END
  1  C
  1  C
  1  C
 11            SUBROUTINE VOLUME(IK,IG,NPT,NL,NT,EN,AAD,AAE,AAF,
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  6        *            NENL,FRE,PRESS,UUR,UUI,VVR,VVI,
  6        *                   AHE,AHF,AHG,REN,GAMMA,MI,XT,YT,UT,VT)
  1   C          VIRTUAL
  1   C
  1   C          STABILITY INTEGRALS IN THE VOLUME
  1   C
  1   C          IK                : COUNTER OF ACOUSTIC MODES
  1   C          IG                : COUNTER OF VORTICAL MODES
  1   C          EN                : DENOMINATOR OF EACH STABILITY INTEGRAL TERM
  1   C          AAD               : (D)-TERM OF ACOUSTIC GROWTH CONSTANT
  1   C          AAE               : (E)-TERM OF ACOUSTIC GROWTH CONSTANT
  1   C          AAF               : (F)-TERM OF ACOUSTIC GROWTH CONSTANT
  1   C          AHE               : (E)-TERM OF VORTICALLY COUPLED ACOUSTIC
  1   C                              GROWTH CONSTANT
  1   C          AHF               : (F)-TERM OF VORTICALLY COUPLED ACOUSTIC
  1   C                              GROWTH CONSTANT
  1   C          AHG               : (G)-TERM OF VORTICALLY COUPLED ACOUSTIC
  1   C                              GROWTH CONSTANT
  1   C          X(NPT)            : LOCAL X-COORDINATES
  1   C          Y(NPT)            : LOCAL Y-COORDINATES
  1   C          U(NPT)            : LOCAL VALUES OF MEAN VELOCITIES IN
  1   C                              X-DIRECTION
  1   C          V(NPT)            : LOCAL VALUES OF MEAN VELOCITIES IN
  1   C                              Y-DIRECTION
  1   C          P(NPT)            : LOCAL VALUES OF ACOUSTIC MODES
  1   C          UR(NPT)           : LOCAL VALUES OF REAL PART OF VORTICAL
  1   C                              DISTURBANCES IN X-DIRECTION
  1   C          VR(NPT)           : LOCAL VALUES OF REAL PART OF VORTICAL
  1   C                              DISTURBANCES IN Y-DIRECTION
  1   C          UI(NPT)           : LOCAL VALUES OF IMAGINARY PART OF
  1   C                              VORTICAL DISTURBANCES IN X-DIRECTION
  1   C          VI(NPT)           : LOCAL VALUES OF IMAGINARY PART OF
  1   C                              VORTICAL DISTURBANCES IN Y-DIRECTION
  1   C          FI(NPT)           : INTERPOLATION FUNCTIONS
  1   C          DX(NPT)           : FIRST DERIVATIVES OF INIERPOLATION
  1   C                              FUNCTIONS IN X-DIRECTION
  1   C          DY(NPT)           : FIRST DERIVATIVES OF INTERPOLATION
  1   C                              FUNCTIONS IN Y-DIRECTION
  1   C          DXX(NPT)          : SECOND DERIVATIVES OF INTERPOLATION
  1   C                              FUNCTIONS IN X-DIRECTION
  1   C          DXY(NPT)          : SECOND DERIVATIVES OF INTERPOLATION
  1   C                              FUNCTIONS IN X- AND Y-DIRECTION
  1   C          DYY(NPT)          : SECOND DERIVATIVES OF INTERPOALTION
  1   C                              FUNCTIONS IN Y-DIRECTION
  1   C
  7         DIMENSION XT(NT),YT(NT),UT(NT),VT(NT)
 11             DIMENSION X(4),Y(4),U(4),V(4),P(4),UR(4),VR(4),UI(4),VI(4),
  6        *             FI(4),DX(4),DY(4),AA(4),AB(4),DXX(4),DXY(4),
  6        *             DYY(4),WS(4),ST(4),NENL(NL,4),FRE(NT),PRESS(NT,NT),
  6        * UUR(NT,MI),UUI(NT,MI),VVR(NT,MI),VVI(NT,MI)
  1   C
 11             CALL GAUSS(4,WS,ST)
  1   C
 11             FR=FRE(IK)
```

```
START
 COL   ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  7          RENINV=1./REN
  7          FRINV=1./FR
  7          FR2INV=FRINV*FRINV
  7          RFRINV=FRINV*RENINV
  7          RF2INV=RENINV*FR2INV
  7          RN3INV=RENINV/3.
  1    C
  1    C         INITAILIZATIONS
  1    C
 11              EN=0.0
 11              AAD=0.0
 11              AAE=0.0
 11              AAF=0.0
 11              AHE=0.0
 11              AHF=0.0
 11              AHG=0.0
  1    C
  1    C         VOLUME INTEGRATIONS
  1    C
 11              DO 1 MMM=1,NL
  1    C
 11              DO 10 N=1,NPT
 11              NN=NENL(MMM,N)
 11              X(N)=XT(NN)
 11              Y(N)=YT(NN)
 11              U(N)=UT(NN)
 11              V(N)=VT(NN)
 11              P(N)=PRESS(NN,IK)
 11              UR(N)=UUR(NN,IG)
 11              VR(N)=VVR(NN,IG)
 11              UI(N)=UUI(NN,IG)
 11              VI(N)=VVI(NN,IG)
  1    10        CONTINUE
  1    C
 11              DO 20 K=1,4
 11              DO 20 L=1,4
 11              XI=ST(K)
 11              ETA=ST(L)
 11              ACOF=WS(K)*WS(L)
  1    C
 11              CALL INTER(XI,ETA,NPT,X,Y,DX,DY,FI,DT[illegible],AA,AB)
  1    C
 11              CALL QUADR(AA,AB,X,Y,DX,DY,DXX,DXY,DYY,NPT)
  1    C
 11              YP=0.0
 11              PRN=0.0
 11              PRX=0.0
 11              PRY=0.0
 11              PRXX=0.0
 11              PRXY=0.0
 11              PRYY=0.0
 11              UNN=0.0
 11              UNX=0.0
 11              UNY=0.0
```

49

```
START
 COL   ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  11           VNN=0.0
  11           VNX=0.0
  11           VNY=0.0
  11           URN=0.0
  11           URX=0.0
  11           URY=0.0
  11           VRN=0.0
  11           VRX=0.0
  11           VRY=0.0
  11           UIN=0.0
  11           UIX=0.0
  11           UIY=0.0
  11           VIN=0.0
  11           VIX=0.0
  11           VIY=0.0
   1   C
  11           DO 30 N=1,NPT
  11           YP=YP+FI(N)*Y(N)
  11           PRN=PRN+FI(N)*P(N)
  11           PRX=PRX+DX(N)*P(N)
  11           PRY=PRY+DY(N)*P(N)
  11           PRXX=PRXX+DXX(N)*P(N)
  11           PRXY=PRXY+DXY(N)*P(N)
  11           PRYY=PRYY+DYY(N)*P(N)
  11           UNN=UNN+FI(N)*U(N)
  11           UNX=UNX+DX(N)*U(N)
  11           UNY=UNY+DY(N)*U(N)
  11           VNN=VNN+FI(N)*V(N)
  11           VNX=VNX+DX(N)*V(N)
  11           VNY=VNY+DY(N)*V(N)
  11           URN=URN+FI(N)*UR(N)
  11           URX=URX+DX(N)*UR(N)
  11           URY=URY+DY(N)*UR(N)
  11           VRN=VRN+FI(N)*VR(N)
  11           VRX=VRX+DX(N)*VR(N)
  11           VRY=VRY+DY(N)*VR(N)
  11           UIN=UIN+FI(N)*UI(N)
  11           UIX=UIX+DX(N)*UI(N)
  11           UIY=UIY+DY(N)*UI(N)
  11           VIN=VIN+FI(N)*VI(N)
  11           VIX=VIX+DX(N)*VI(N)
  11           VIY=VIY+DY(N)*VI(N)
   1   30      CONTINUE
   1   C
  11           C=3.14159*YP*ACOF*DTA
   1   C
  11           EN=EN+C*2.0*PRN*PRN
  11           AAD=AAD-C*(2.0*GAMMA+1.0)*PRN*(UNN*PRX+VNN*PRY)
  11           AAE=AAE+C*(-(UNN*PRX+VNN*PRY)*PRN
   6        *            +2.0*(UNN*PRX*PRXX+VNN*PRY*PRYY
   6        *                 +(UNN*PRX+VNN*PRY)*PRXY)*FR2INV)
  11           AAF=AAF-C*((PRXX*PRXX+2.0*PRXY*PRXY+PRYY*PRYY)*RF2INV
   6        *            +FR*FR*PRN*PRN*RN3INV)
  11           AHE=AHE+2.0*C*GAMMA*(UNN*UIN*PRXX+VNN*VIN*PRYY
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  6        *             +(UNN*VIN+VNN*UIN)*PRXY)*FRINV
 11              AHF=AHF-C*GAMMA*(UIX*PRXX+VIX*PRYY+(UIY+VIX)*PRXY)*RFRINV
 11              AHG=AHG+2.0*C*GAMMA*(GAMMA-1.0)
  6        *             *(2.0*(UNX*URX+VNY*VRY)
  6        *              +(UNY+VNX)*(URY+VRX))*RENINV
  1   20         CONTINUE
  1   C
  1   1          CONTINUE
  1   C
 11              RETURN
 11              END
  1   C
  1   C
  1   C
 11              SUBROUTINE ELEUV(MMM,NPT,NGPT,XX,YY,UU,VV,PP,
  6        *                       REN,DT,NENN,ANM,FNU,FNV,WS,ST,NEL)
  1   C
  1   C          LOCAL MATRICES FOR VELOCITIES
  1   C
 11              DIMENSION XX(NGPT),YY(NGPT),UU(NGPT),VV(NGPT),PP(NGPT)
 11              DIMENSION ANM(4,4),FNU(4),FNV(4),WS(4),ST(4)
 11              DIMENSION X(4),Y(4),FI(4),DX(4),DY(4),AA(4),AB(4),
  6        *               U(4),V(4),P(4),NENN(NEL,4)
  1   C
  7          RENINV=1./REN
  7          DTINV=1./DT
  1   C
 11              DO 1 N=1,NPT
 11              NN=NENN(MMM,N)
 11              X(N)=XX(NN)
 11              Y(N)=YY(NN)
 11              U(N)=UU(NN)
 11              V(N)=VV(NN)
 11              P(N)=PP(NN)
  1   1          CONTINUE
  1   C
 11              DO 2 N=1,NPT
 11              FNU(N)=0.0
 11              FNV(N)=0.0
 11              DO 2 M=1,NPT
 11              ANM(N,M)=0.0
  1   2          CONTINUE
  1   C
 11              DO 300 K=1,4
 11              DO 300 L=1,4
 11              XI=ST(K)
 11              ETA=ST(L)
 11              ACOF=WS(K)*WS(L)
  1   C
 11              CALL INTER(XI,ETA,NPT,X,Y,DX,DY,FI,DTA,AA,AB)
  1   C
 11              SSU=0.0
 11              SSV=0.0
 11              SXP=0.0
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

 11             SYP=0.0
 11             YP=0.0
  1   C
 11             DO 301 N=1,NPT
 11             SSU=SSU+FI(N)*U(N)
 11             SSV=SSV+FI(N)*V(N)
 11             SXP=SXP+DX(N)*P(N)
 11             SYP=SYP+DY(N)*P(N)
 11             YP =YP +FI(N)*Y(N)
  1   301       CONTINUE
  1   C
 11             C=ACOF*DTA*YP
  1   C
 11             DO 500 N=1,NPT
  7         CFIN=C*FI(N)
 11             FNU(N)=FNU(N)+CFIN*(SSU*DTINV-SXP)
 11             FNV(N)=FNV(N)+CFIN*(SSV*DTINV-SYP)
 11             DO 500 M=1,NPT
 11             ANM(N,M)=ANM(N,M)+CFIN*(FI(M)*DTINV+SSU*DX(M)+SSV*DY(M))
  6        *                      +C*(DX(N)*DX(M)+DY(N)*DY(M))*RENINV
  1   500       CONTINUE
  1   300       CONTINUE
  1   C
 11             RETURN
 11             END
  1   C
  1   C
 11             SUBROUTINE ELEPR(MMM,NPT,NGPT,XX,YY,US,VS,
  6        *                      DT,BNM,GN,WS,ST,NEL,NENN)
  1   C
  1   C         LOCAL MATRICES FOR PRESSURE CORRECTIONS
  1   C
 11             DIMENSION XX(NGPT),YY(NGPT),US(NGPT),VS(NGPT)
 11             DIMENSION BNM(4,4),GN(4),WS(4),ST(4),NENN(NEL,4)
 11             DIMENSION X(4),Y(4),FI(4),DX(4),DY(4),AA(4),AB(4),U(4),V(4)
  1   C
  7         DTINV=1./DT
 11             DO 1 N=1,NPT
 11             NN=NENN(MMM,N)
 11             X(N)=XX(NN)
 11             Y(N)=YY(NN)
 11             U(N)=US(NN)
 11             V(N)=VS(NN)
  1   1         CONTINUE
  1   C
 11             DO 2 N=1,NPT
 11             GN(N)=0.0
 11             DO 2 M=1,NPT
 11             BNM(N,M)=0.0
  1   2         CONTINUE
  1   C
 11             DO 300 K=1,4
 11             DO 300 L=1,4
 11             XI=ST(K)
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

 11              ETA=ST(L)
 11              ACOF=WS(K)*WS(L)
  1   C
 11              CALL INTER(XI,ETA,NPT,X,Y,DX,DY,FI,DTA,AA,AB)
  1   C
 11              SXU=0.0
 11              SYV=0.0
 11              YP =0.0
  1   C
 11              DO 301 N=1,NPT
 11              SXU=SXU+DX(N)*U(N)
 11              SYV=SYV+DY(N)*V(N)
 11              YP =YP +FI(N)*Y(N)
  1   301        CONTINUE
  1   C
 11              C=ACOF*DTA*YP
  1   C
 11              DO 500 N=1,NPT
 11              GN(N)=GN(N)-C*FI(N)*(SXU+SYV)*DTINV
 11              DO 500 M=1,NPT
 11              BNM(N,M)=BNM(N,M)+C*(DX(N)*DX(M)+DY(N)*DY(M))
  1   500        CONTINUE
  1   300        CONTINUE
  1   C
 11              RETURN
 11              END
  1   C
  1   C
 11              SUBROUTINE ELEAC(MMM,NPT,NGPT,XX,YY,DPP,CNM,
  6        *                      HNU,HNV,WS,ST,NEL,NENN)
  1   C
  1   C          LOCAL MATRICES FOR ACCELERATIONS
  1   C
 11              DIMENSION XX(NGPT),YY(NGPT),DPP(NGPT),NENN(NEL,4)
 11              DIMENSION CNM(4,4),HNU(4),HNV(4),WS(4),ST(4)
 11              DIMENSION X(4),Y(4),DP(4),FI(4),DX(4),DY(4),AA(4),AB(4)
  1   C
 11              DO 1 N=1,NPT
 11              NN=NENN(MMM,N)
 11              X(N)=XX(NN)
 11              Y(N)=YY(NN)
 11              DP(N)=DPP(NN)
  1   1          CONTINUE
  1   C
 11              DO 2 N=1,NPT
 11              HNU(N)=0.0
 11              HNV(N)=0.0
 11              DO 2 M=1,NPT
 11              CNM(N,M)=0.0
  1   2          CONTINUE
  1   C
 11              DO 300 K=1,4
 11              DO 300 L=1,4
 11              XI=ST(K)
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  11              ETA=ST(L)
  11              ACOF=WS(K)*WS(L)
   1   C
  11              CALL INTER(XI,ETA,NPT,X,Y,DX,DY,FI,DTA,AA,AB)
   1   C
  11              SDPX=0.0
  11              SDPY=0.0
  11              YP  =0.0
   1   C
  11              DO 310 N=1,NPT
  11              SDPX=SDPX+DX(N)*DP(N)
  11              SDPY=SDPY+DY(N)*DP(N)
  11              YP  =YP  +FI(N)*Y(N)
   1   310        CONTINUE
   1   C
  11              C=ACOF*DTA*YP
   1   C
  11              DO 500 N=1,NPT
   7          CFIN=C*FI(N)
  11              HNU(N)=HNU(N)-CFIN*SDPX
  11              HNV(N)=HNV(N)-CFIN*SDPY
  11              DO 500 M=1,NPT
  11              CNM(N,M)=CNM(N,M)+CFIN*FI(M)
   1   500        CONTINUE
   1   300        CONTINUE
   1   C
  11              RETURN
  11              END
   1   C
   1   C
  11              SUBROUTINE ELEMP(MMM,NPT,ANM,BNM,WS,ST,NL,NT,NENL,
   6         *              XT,YT)
   1   C
   7          DIMENSION XT(NT),YT(NT)
  11              DIMENSION ANM(4,4),BNM(4,4),WS(4),ST(4),NENL(NL,4)
  11              DIMENSION X(4),Y(4),FI(4),DX(4),DY(4),AA(4),AB(4)
   1   C
  11              DO 100 N=1,NPT
  11              NN=NENL(MMM,N)
  11              X(N)=XT(NN)
  11              Y(N)=YT(NN)
   1   100        CONTINUE
   1   C
  11              DO 110 I=1,NPT
  11              DO 110 J=1,NPT
  11              ANM(I,J)=0.0
  11              BNM(I,J)=0.0
   1   110        CONTINUE
   1   C
  11              DO 300 K=1,4
  11              DO 300 L=1,4
  11              XI=ST(K)
  11              ETA=ST(L)
  11              ACOF=WS(K)*WS(L)
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  1   C
 11               CALL INTER(XI,ETA,NPT,X,Y,DX,DY,FI,DTA,AA,AB)
  1   C
 11               YP=0.0
 11               DO 310 N=1,NPT
 11               YP=YP+FI(N)*Y(N)
  1   310         CONTINUE
  1   C
 11               C=•COF*DTA*YP
  1   C
 11               DO 500 N=1,NPT
  7           CFIN=C*FI(N)
 11               DO 500 M=1,NPT
 11               ANM(N,M)=ANM(N,M)+C*(DX(N)*DX(M)+DY(N)*DY(M))
 11               BNM(N,M)=BNM(N,M)+CFIN*FI(M)
  1   500         CONTINUE
  1   300         CONTINUE
  1   C
 11               RETURN
 11               END
  1   C
  1   C
  1   C
 11               SUBROUTINE ELEVR(MMM,NPT,ANM,BNM,WS,ST,REN,NL,NT,NENL,
  6          *                XT,YT,UT,VT)
  1   C
  1   C           LOCAL ELEMENT FOR VORTICAL EIGENMODES
  1   C
  7           DIMENSION XT(NT),YT(NT),UT(NT),VT(NT)
 11               DIMENSION X(4),Y(4),U(4),V(4),ANM(4,4,4,4),BNM(4,4,4,4),
  6          *              WS(4),ST(4),PI(4,4),DPX(4,4),DPY(4,4),
  6          *              DPXX(4,4),DPXY(4,4),DPYY(4,4),NENL(NL,4)
  1   C
  7           RENINV=1./REN
  1   C
 11               DO 100 N=1,NPT
 11               NN=NENL(MMM,N)
 11               X(N)=XT(NN)
 11               Y(N)=YT(NN)
 11               U(N)=UT(NN)
 11               V(N)=VT(NN)
  1   100         CONTINUE
  1   C
 11               DO 110 L=1,4
 11               DO 110 N=1,NPT
 11               DO 110 LL=1,4
 11               DO 110 M=1,NPT
 11               ANM(L,LL,N,M)=0.0
 11               BNM(L,LL,N,M)=0.0
  1   110         CONTINUE
  1   C
 11               DA=ABS(X(2)-X(1))*.5
 11               DB=ABS(Y(4)-Y(1))*.5
  1   C
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

 11         DO 300 K=1,4
 11         DO 300 L=1,4
 11         XI=ST(K)
 11         ETA=ST(L)
 11         ACOF=WS(K)*WS(L)
  1   C
 11         CALL HERMT(XI,ETA,DA,DB,PI,DPX,DPY,DPXX,DPXY,DPYY)
  1   C
 11         SU=0.0
 11         SV=0.0
 11         SUX=0.0
 11         SUY=0.0
 11         SVX=0.0
 11         SVY=0.0
 11         SUXX=0.0
 11         SUXY=0.0
 11         SUYY=0.0
 11         SVXX=0.0
 11         SVXY=0.0
 11         SVYY=0.0
 11         YP=0.0
  1   C
 11         DO 310 N=1,NPT
 11         SU=SU+PI(1,N)*U(N)
 11         SV=SV+PI(1,N)*V(N)
 11         SUX=SUX+DPX(1,N)*U(N)
 11         SUY=SUY+DPY(1,N)*U(N)
 11         SVX=SVX+DPX(1,N)*V(N)
 11         SVY=SVY+DPY(1,N)*V(N)
 11         SUXX=SUXX+DPXX(1,N)*U(N)
 11         SUXY=SUXY+DPXY(1,N)*U(N)
 11         SUYY=SUYY+DPYY(1,N)*U(N)
 11         SVXX=SVXX+DPXX(1,N)*V(N)
 11         SVXY=SVXY+DPXY(1,N)*V(N)
 11         SVYY=SVYY+DPYY(1,N)*V(N)
 11         YP=YP+PI(1,N)*Y(N)
  1   310   CONTINUE
  1   C
 11         C=ACOF*DA*DB*YP
  1   C
 11         DO 500 IL=1,4
 11         DO 500 N=1,NPT
 11         DO 500 JL=1,4
 11         DO 500 M=1,NPT
 11         ANM(IL,JL,N,M)=ANM(IL,JL,N,M)+C*(
  6        *       PI(IL,N)*(SVXX*DPY(JL,M)-SVXY*DPX(JL,M)
  6        *                  -SUXY*DPY(JL,M)+SUYY*DPX(JL,M))
  6        *     +(SU*DPX(IL,N)+SV*DPY(IL,N))*(DPXX(JL,M)+DPYY(JL,M))
  6        *     -(DPXX(IL,N)*DPXX(JL,M)+2.0*DPXY(IL,N)*DPXY(JL,M)
  6        *      +DPYY(IL,N)*DPYY(JL,M))*RENINV)
 11         BNM(IL,JL,N,M)=BNM(IL,JL,N,M)+C*(
  6        *      DPX(IL,N)*DPX(JL,M)+DPY(IL,N)*DPY(JL,M))
  1   500   CONTINUE
  1   300   CONTINUE
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  1   C
 11             RETURN
 11             END
  1   C
  1   C
  1   C
 11             SUBROUTINE ADJUST(NB,NT,NBX,A,UNB,F,NBW1,NBWT)
  1   C
  1   C         ADJUSTMENT OF BOUNDARY CONDITIONS FOR MEAN VELOCITY FIELDS
  1   C
 11             DIMENSION NBX(NB),UNB(NB)
 11             DIMENSION A(NT,NBWT),F(NT)
  1   C
 11             DO 1 I=1,NB
 11             ND=NBX(I)
 11             DO 2 J=1,NBWT
  1   2         A(ND,J)=0.0
 11             DO 3 J=1,NT
 11             NX1=(ND-1)+NBWT
 11             IF(NX1.LE.0.OR.NX1.GT.NBWT) GO TO 3
 11             F(J)=F(J)-A(J,NX1)*UNB(I)
 11             A(J,NX1)=0.0
  1   3         CONTINUE
 11             A(ND,NBW1)=1.0
 11             F(ND)=UNB(I)
  1   1         CONTINUE
  1   C
 11             RETURN
 11             END
  1   C
  1   C
 11             SUBROUTINE GAUSU(A,B,X,N,NBW1,NBWT)
  1   C
  1   C         SOLVER OF LINEAR EQUATIONS WITH BAND MATRIX
  1   C
 11             DIMENSION A(N,NBWT),B(N),X(N)
  1   C
 11             NM1=N-1
  1   C
 11             DO 5 I=1,NM1
 11             IP1=I+1
 11             DO 6 K=IP1,N
 11             II=I-K+NBW1
 11             IF(II.LE.0 .OR. II.GT.NBWT) GO TO 6
 11             FACT=-A(K,II)/A(I,NBW1)
 11             IF(FACT.EQ.0.0) GO TO 6
 11             A(K,II)=0.0
 11             DO 7 J=IP1,N
 11             JI=J-I+NBW1
 11             IF(JI.LE.0 .OR. JI.GT.NBWT) GO TO 7
 11             JJ=J-K+NBW1
 11             IF(JJ.LE.0 .OR. JJ.GT.NBWT) GO TO 7
 11             A(K,JJ)=A(K,JJ)+FACT*A(I,JI)
  1   7         CONTINUE
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8
 11              B(K)=B(K)+FACT*B(I)
  1   6          CONTINUE
  1   5          CONTINUE
  1   C
  1   C
 11              X(N)=B(N)/A(N,NBW1)
 11              NP1=N+1
 11              DO 2 K=1,NM1
 11              SUM=0.0
 11              NMK=N-K
 11              DO 3 J=1,K
 11              JJ=(NP1-J)-NMK+NBW1
 11              IF(JJ.LE.0 .OR.JJ.GT.NBWT) GO TO 3
 11              SUM=SUM+A(NMK,JJ)*X(NP1-J)
  1   3          CONTINUE
 11              X(NMK)=(B(NMK)-SUM)/A(NMK,NBW1)
  1   2          CONTINUE
  1   C
 11              RETURN
 11              END
  1   C
  1   C
  1   C
  7         SUBROUTINE GAUSS(NGG,W,ST)
  1   C
  7         DIMENSION W(NGG),ST(NGG)
  1   C
  7         IF(NGG.EQ.1) GO TO 10
  7         IF(NGG.EQ.2) GO TO 20
  7         IF(NGG.EQ.3) GO TO 30
  7         IF(NGG.EQ.4) GO TO 40
  7         IF(NGG.EQ.5) GO TO 50
  7         IF(NGG.EQ.6) GO TO 60
  1   C
  2    10   W(1)=2.0
  7         ST(1)=0.0
  7         GO TO 70
  1   C
  2    20   W(1)   =  1.0
  7         W(2)   =  W(1)
  7         ST(1)  = -0.577350269
  7         ST(2)  = -ST(1)
  7         GO TO 70
  1   C
  2    30   W(1)   =  0.5555555555
  7         W(2)   =  0.8888888888
  7         W(3)   =  W(1)
  7         ST(1)  = -0.7745966692
  7         ST(2)  =  0.0
  7         ST(3)  = -ST(1)
  7         GO TO 70
  1   C
  2    40   W(1)   =  0.3478548451
  7         W(2)   =  0.6521451548
```

```
  7            W(3)    =  W(2)
  7            W(4)    =  W(1)
  7            ST(1)   = -0.8611363115
  7            ST(2)   = -0.3399810435
  7            ST(3)   = -ST(2)
  7            ST(4)   = -ST(1)
  7            GO TO 70
  1   C
  2    50      W(1)    =  0.2369268850
  7            W(2)    =  0.478628674
  7            W(3)    =  0.5688888888
  7            W(4)    =  W(2)
  7            W(5)    =  W(1)
  7            ST(1)   = -0.9061798459
  7            ST(2)   = -0.5384693101
  7            ST(3)   =  0.0
  7            ST(4)   = -ST(2)
  7            ST(5)   = -ST(1)
  7            GO TO 70
  1   C
  2    60      W(1)    =  0.1713244923
  7            W(2)    =  0.3607615730
  7            W(3)    =  0.4679139345
  7            W(4)    =  W(3)
  7            W(5)    =  W(2)
  7            W(6)    =  W(1)
  7            ST(1)   = -0.9324695142
  7            ST(2)   = -0.6612093864
  7            ST(3)   = -0.2386191860
  7            ST(4)   = -ST(3)
  7            ST(5)   = -ST(2)
  7            ST(6)   = -ST(1)
  1   C
  2    70      CONTINUE
  1   C
  7            RETURN
  7            END
  1   C
  1   C
  1   C
 11                SUBROUTINE INTER(XX,YY,NPT,X,Y,DX,DY,FT,DTA,AA,AB)
  1   C
  1   C            LINEAR INTERPOLATION FUNCTIONS AND THEIR FIRST DERIVATIVES
  1   C
 11                DIMENSION X(NPT),Y(NPT),DX(NPT),DY(NPT),FT(NPT)
 11                DIMENSION AA(NPT),AB(NPT),DTJ(2,2)
  1   C
  1   C            FT(1)=0.25*(1.0-XX)*(1.0-YY)
  1   C            FT(2)=0.25*(1.0+XX)*(1.0-YY)
  1   C            FT(3)=0.25*(1.0+XX)*(1.0+YY)
  1   C            FT(4)=0.25*(1.0-XX)*(1.0+YY)
  1   C
  1   C            AA(1)=0.25*(-1.0+YY)
  1   C            AA(2)=0.25*( 1.0-YY)
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

   1  C          AA(3)=0.25*( 1.0+YY)
   1  C          AA(4)=0.25*(-1.0-YY)
   1  C
   1  C          AB(1)=0.25*(-1.0+XX)
   1  C          AB(2)=0.25*(-1.0-XX)
   1  C          AB(3)=0.25*( 1.0+XX)
   1  C          AB(4)=0.25*( 1.0-XX)
   1  C
   7        XXP1=XX+1.
   7        XXM1=XX-1.
   7        YYP1=YY+1.
   7        YYM1=YY-1.
   1  C
   7        AA(1)=.25*YYM1
   7        AA(2)=-AA(1)
   7        AA(3)=.25*YYP1
   7        AA(4)=-AA(3)
   1  C
   7        AB(1)=.25*XXM1
   7        AB(3)=.25*XXP1
   7        AB(2)=-AB(3)
   7        AB(4)=-AB(1)
   1  C
   7        FT(1)=AB(1)*YYM1
   7        FT(2)=AB(2)*YYM1
   7        FT(3)=AB(3)*YYP1
   7        FT(4)=AB(4)*YYP1
   1  C
  11             DO 4 I=1,2
  11             DO 4 J=1,2
   1  4          DTJ(I,J)=0.0
   1  C
  11             DO 5 N=1,NPT
  11             DTJ(1,1)=DTJ(1,1)+AA(N)*X(N)
  11             DTJ(1,2)=DTJ(1,2)+AA(N)*Y(N)
  11             DTJ(2,1)=DTJ(2,1)+AB(N)*X(N)
   1  5          DTJ(2,2)=DTJ(2,2)+AB(N)*Y(N)
   1  C
  11             DTA=DTJ(1,1)*DTJ(2,2)-DTJ(1,2)*DTJ(2,1)
   7        DTAINV=1./DTA
   1  C
  11             DO 7 N=1,NPT
  11             DX(N)=( DTJ(2,2)*AA(N)-DTJ(1,2)*AB(N))*DTAINV
  11             DY(N)=(-DTJ(2,1)*AA(N)+DTJ(1,1)*AB(N))*DTAINV
   1  7          CONTINUE
   1  C
  11             RETURN
  11             END
   1  C
   1  C
   1  C
  11             SUBROUTINE QUADR(AA,AB,X,Y,DX,DY,DXX,DXY,DYY,NPT)
   1  C
   1  C          SECOND DERIVATIVES OF INTERPOLATION FUNCTIONS
```

```
START
 COL' ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  1  C
 11          DIMENSION AA(NPT),AB(NPT),X(NPT),Y(NPT),DX(NPT),DY(NPT),
  6       *            DXX(NPT),DXY(NPT),DYY(NPT)
 11          DIMENSION DUMXX(4),DUMXY(4),DUMYY(4),AXX(4),AXY(4),AYY(4)
  1  C
 11          AXX(1)=0.0
 11          AXX(2)=0.0
 11          AXX(3)=0.0
 11          AXX(4)=0.0
 11          AXY(1)=0.25
 11          AXY(2)=-0.25
 11          AXY(3)=0.25
 11          AXY(4)=-0.25
 11          AYY(1)=0.0
 11          AYY(2)=0.0
 11          AYY(3)=0.0
 11          AYY(4)=0.0
  1  C
 11          DXX1=0.0
 11          DXY1=0.0
 11          DYX1=0.0
 11          DYY1=0.0
 11          DXXXX=0.0
 11          DXYXX=0.0
 11          DYYXX=0.0
 11          DXXYY=0.0
 11          DXYYY=0.0
 11          DYYYY=0.0
  1  C
 11          DO 5 N=1,NPT
 11          DXX1=DXX1+AA(N)*X(N)
 11          DXY1=DXY1+AA(N)*Y(N)
 11          DYX1=DYX1+AB(N)*X(N)
 11          DYY1=DYY1+AB(N)*Y(N)
 11          DXXXX=DXXXX+AXX(N)*X(N)
 11          DXYXX=DXYXX+AXY(N)*X(N)
 11          DYYXX=DYYXX+AYY(N)*X(N)
 11          DXXYY=DXXYY+AXX(N)*Y(N)
 11          DXYYY=DXYYY+AXY(N)*Y(N)
 11          DYYYY=DYYYY+AYY(N)*Y(N)
  2   5    CONTINUE
  1  C
 11          DDD=DXX1*DXX1*DYY1*DYY1*(DXX1*DYY1+DXY1*DYX1)
  6       *     +2.*DXX1*DYX1*DYX1*DYY1*DXY1*DXY1
  6       *     +2.*DYX1*DYX1*DXY1*DYY1*DXX1*DXY1
  6       *     -DYX1*DYX1*DXY1*DXY1*(DXX1*DYY1+DYX1*DXY1)
  6       *     -2.*DXX1*DYX1*DXX1*DXY1*DYY1*DYY1
  6       *     -2.*DXX1*DXX1*DXY1*DYY1*DYX1*DYY1
 13            DDDINV=1./DDD
  1  C
 11          DO 8 N=1,NPT
 11          DUMXX(N)=AXX(N)-DXXXX*DX(N)-DXXYY*DY(N)
 11          DUMXY(N)=AXY(N)-DXYXX*DX(N)-DXYYY*DY(N)
 11          DUMYY(N)=AYY(N)-DYYXX*DX(N)-DYYYY*DY(N)
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  2    8   CONTINUE
  1   C
 11          D11=(DXX1*DYY1+DXY1*DYX1)*DYY1*DYY1-2.*DYX1*DYY1*DXY1*DYY1
 11          D12=2.*DYX1*DYY1*DYY1*DYY1-2.*DXX1*DXY1*DYY1*DYY1
 11          D13=2.*DXX1*DXY1*DXY1*DYY1-DXY1*DXY1*(DXX1*DYY1+DYX1*DXY1)
 11          D21=DXY1*DXY1*DYX1*DYX1-DXX1*DXX1*DYY1*DYY1
 11          D22=DXX1*DXX1*DYY1*DYY1-DXY1*DXY1*DYX1*DYX1
 11          D23=DXY1*DXY1*DXX1*DYX1-DXY1*DXY1*DYX1*DYX1
 11          D31=2.*DXX1*DXY1*DXY1*DYY1-DXY1*DXY1*(DXX1*DYY1+DYX1*DXY1)
 11          D32=DXY1*DXY1*DXX1*DYX1-DXX1*DXX1*DXY1*DYY1
 11          D33=DXX1*DXX1*(DXX1*DYY1+DYX1*DXX1)-2.*DXX1*DXY1*DXX1*DYX1
  1   C
  1   C
 11          DO 9 N=1,NPT
 11          DXX(N)=(D11*DUMXX(N)+D12*DUMXY(N)+D13*DUMYY(N))*DDDINV
 11          DXY(N)=(D21*DUMXX(N)+D22*DUMXY(N)+D23*DUMYY(N))*DDDINV
 11          DYY(N)=(D31*DUMXX(N)+D32*DUMXY(N)+D33*DUMYY(N))*DDDINV
  2    9   CONTINUE
  1   C
 11          RETURN
 11          END
 11          SUBROUTINE HERMT(XI,ETA,DA,DB,PI,DPX,DPY,DPXX,DPXY,DPYY)
  1   C
  1   C      HERMITE POLYNOMIAL INTERPOLATION FUNCTIONS
  1   C
 11          DIMENSION PI(4,4),DPX(4,4),DPY(4,4),DPXX(4,4),DPXY(4,4),
  6       *            DPYY(4,4)
  1   C
  7        DAINV4=.25/DA
  7        DBINV4=.25/DB
  7        DA2IN4=DAINV4/DA
  7        DB2IN4=DBINV4/DB
  1   C
 11          F1X=(2.-3.*XI+XI*XI*XI)*.25
 11          F2X=(2.+3.*XI-XI*XI*XI)*.25
 11          G1X=DA*(1.-XI-XI*XI+XI*XI*XI)*.25
 11          G2X=DA*(-1.-XI+XI*XI+XI*XI*XI)*.25
  1   C
 11          F1Y=(2.-3.*ETA+ETA*ETA*ETA)*.25
 11          F2Y=(2.+3.*ETA-ETA*ETA*ETA)*.25
 11          G1Y=DB*(1.-ETA-ETA*ETA+ETA*ETA*ETA)*.25
 11          G2Y=DB*(-1.-ETA+ETA*ETA+ETA*ETA*ETA)*.25
  1   C
 11          DF1X=(-3.+3.*XI*XI)*DAINV4
 11          DF2X=( 3.-3.*XI*XI)*DAINV4
 11          DG1X=(-1.-2.*XI+3.*XI*XI)*.25
 11          DG2X=(-1.+2.*XI+3.*XI*XI)*.25
  1   C
 11          DF1Y=(-3.+3.*ETA*ETA)*DBINV4
 11          DF2Y=( 3.-3.*ETA*ETA)*DBINV4
 11          DG1Y=(-1.-2.*ETA+3.*ETA*ETA)*.25
 11          DG2Y=(-1.+2.*ETA+3.*ETA*ETA)*.25
  1   C
 11          DDF1X=6.*XI*DA2IN4
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

 11             DDF2X=-6.*XI*DA2IN4
 11             DDG1X=(-2.+6.*XI)*DAINV4
 11             DDG2X=( 2.+6.*XI)*DAINV4
  1   C
 11             DDF1Y=6.*ETA*DB2IN4
 11             DDF2Y=-6.*ETA*DB2IN4
 11             DDG1Y=(-2.+6.*ETA)*DBINV4
 11             DDG2Y=( 2.+6.*ETA)*DBINV4
  1   C
 11             PI(1,1)=F1X*F1Y
 11             PI(1,2)=F2X*F1Y
 11             PI(1,3)=F2X*F2Y
 11             PI(1,4)=F1X*F2Y
 11             PI(2,1)=G1X*F1Y
 11             PI(2,2)=G2X*F1Y
 11             PI(2,3)=G2X*F2Y
 11             PI(2,4)=G1X*F2Y
 11             PI(3,1)=F1X*G1Y
 11             PI(3,2)=F2X*G1Y
 11             PI(3,3)=F2X*G2Y
 11             PI(3,4)=F1X*G2Y
 11             PI(4,1)=G1X*G1Y
 11             PI(4,2)=G2X*G1Y
 11             PI(4,3)=G2X*G2Y
 11             PI(4,4)=G1X*G2Y
  1   C
 11             DPX(1,1)=DF1X*F1Y
 11             DPX(1,2)=DF2X*F1Y
 11             DPX(1,3)=DF2X*F2Y
 11             DPX(1,4)=DF1X*F2Y
 11             DPX(2,1)=DG1X*F1Y
 11             DPX(2,2)=DG2X*F1Y
 11             DPX(2,3)=DG2X*F2Y
 11             DPX(2,4)=DG1X*F2Y
 11             DPX(3,1)=DF1X*G1Y
 11             DPX(3,2)=DF2X*G1Y
 11             DPX(3,3)=DF2X*G2Y
 11             DPX(3,4)=DF1X*G2Y
 11             DPX(4,1)=DG1X*G1Y
 11             DPX(4,2)=DG2X*G1Y
 11             DPX(4,3)=DG2X*G2Y
 11             DPX(4,4)=DG1X*G2Y
  1   C
 11             DPY(1,1)=F1X*DF1Y
 11             DPY(1,2)=F2X*DF1Y
 11             DPY(1,3)=F2X*DF2Y
 11             DPY(1,4)=F1X*DF2Y
 11             DPY(2,1)=G1X*DF1Y
 11             DPY(2,2)=G2X*DF1Y
 11             DPY(2,3)=G2X*DF2Y
 11             DPY(2,4)=G1X*DF2Y
 11             DPY(3,1)=F1X*DG1Y
 11             DPY(3,2)=F2X*DG1Y
 11             DPY(3,3)=F2X*DG2Y
```

```
START
 COL   ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

 11              DPY(3,4)=F1X*DG2Y
 11              DPY(4,1)=G1X*DG1Y
 11              DPY(4,2)=G2X*DG1Y
 11              DPY(4,3)=G2X*DG2Y
 11              DPY(4,4)=G1X*DG2Y
  1    C
 11              DPXX(1,1)=DDF1X*F1Y
 11              DPXX(1,2)=DDF2X*F1Y
 11              DPXX(1,3)=DDF2X*F2Y
 11              DPXX(1,4)=DDF1X*F2Y
 11              DPXX(2,1)=DDG1X*F1Y
 11              DPXX(2,2)=DDG2X*F1Y
 11              DPXX(2,3)=DDG2X*F2Y
 11              DPXX(2,4)=DDG1X*F2Y
 11              DPXX(3,1)=DDF1X*G1Y
 11              DPXX(3,2)=DDF2X*G1Y
 11              DPXX(3,3)=DDF2X*G2Y
 11              DPXX(3,4)=DDF1X*G2Y
 11              DPXX(4,1)=DDG1X*G1Y
 11              DPXX(4,2)=DDG2X*G1Y
 11              DPXX(4,3)=DDG2X*G2Y
 11              DPXX(4,4)=DDG1X*G2Y
  1    C
 11              DPYY(1,1)=F1X*DDF1Y
 11              DPYY(1,2)=F2X*DDF1Y
 11              DPYY(1,3)=F2X*DDF2Y
 11              DPYY(1,4)=F1X*DDF2Y
 11              DPYY(2,1)=G1X*DDF1Y
 11              DPYY(2,2)=G2X*DDF1Y
 11              DPYY(2,3)=G2X*DDF2Y
 11              DPYY(2,4)=G1X*DDF2Y
 11              DPYY(3,1)=F1X*DDG1Y
 11              DPYY(3,2)=F2X*DDG1Y
 11              DPYY(3,3)=F2X*DDG2Y
 11              DPYY(3,4)=F1X*DDG2Y
 11              DPYY(4,1)=G1X*DDG1Y
 11              DPYY(4,2)=G2X*DDG1Y
 11              DPYY(4,3)=G2X*DDG2Y
 11              DPYY(4,4)=G1X*DDG2Y
  1    C
 11              DPXY(1,1)=DF1X*DF1Y
 11              DPXY(1,2)=DF2X*DF1Y
 11              DPXY(1,3)=DF2X*DF2Y
 11              DPXY(1,4)=DF1X*DF2Y
 11              DPXY(2,1)=DG1X*DF1Y
 11              DPXY(2,2)=DG2X*DF1Y
 11              DPXY(2,3)=DG2X*DF2Y
 11              DPXY(2,4)=DG1X*DF2Y
 11              DPXY(3,1)=DF1X*DG1Y
 11              DPXY(3,2)=DF2X*DG1Y
 11              DPXY(3,3)=DF2X*DG2Y
 11              DPXY(3,4)=DF1X*DG2Y
 11              DPXY(4,1)=DG1X*DG1Y
 11              DPXY(4,2)=DG2X*DG1Y
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

 11             DPXY(4,3)=DG2X*DG2Y
 11             DPXY(4,4)=DG1X*DG2Y
  1   C
 11             RETURN
 11             END
  7         SUBROUTINE CONVRT                                                 530970
  6        $( N, A, NA, B, NB, X, NX )                                        530980
  1   C                                                                       530990
  7         COMPLEX A(NA,N)                                                   531000
  7         COMPLEX B(NB,N)                                                   531010
  7         COMPLEX W                                                         531020
  7         COMPLEX X(NX,N)                                                   531030
  7         COMPLEX Y                                                         531040
  7         COMPLEX Z                                                         531050
  1   C                                                                       531060
  7         REAL C                                                            531070
  7         REAL D                                                            531080
  1   C                                                                       531090
  7         INTEGER I                                                         531100
  7         INTEGER II                                                        531110
  7         INTEGER IMJ                                                       531120
  7         INTEGER IM1                                                       531130
  7         INTEGER IP1                                                       531140
  7         INTEGER J                                                         531150
  7         INTEGER JM2                                                       531160
  7         INTEGER JP1                                                       531170
  7         INTEGER K                                                         531180
  7         INTEGER N                                                         531190
  7         INTEGER NA                                                        531200
  7         INTEGER NB                                                        531210
  7         INTEGER NM1                                                       531220
  7         INTEGER NM2                                                       531230
  7         INTEGER NX                                                        531240
  1   C                                                                       531250
  7         LUWT = 6                                                          531260
  7         NM1 = N - 1                                                       531270
  7         DO  80 I=1,NM1                                                    531280
  7         D = 0.0                                                           531290
  7         IP1 = I + 1                                                       531300
  7         DO  10 K=IP1,N                                                    531310
  7         Y = B(K,I)                                                        531320
  7         C = ABS(REAL(Y)) + ABS(AIMAG(Y))                                  531330
  7         IF( C.LE.D )     GO TO   9                                        531340
  7         D = C                                                             531350
  7         II = K                                                            531360
  5       9 CONTINUE                                                          531370
  4      10 CONTINUE                                                          531380
  7         IF( D.EQ.0.0 )     GO TO  78                                      531390
  7         Y = B(I,I)                                                        531400
  7         IF( D.LE.ABS(REAL(Y)) + ABS(AIMAG(Y)) )     GO TO  40             531410
  7         DO 20 J=1,N                                                       531420
  7         Y = A(I,J)                                                        531430
  7         A(I,J) = A(II,J)                                                  531440
  7         A(II,J) = Y                                                       531450
```

```
START
 COL   ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  4      20 CONTINUE                                                          531460
  7         DO 30  J=I,N                                                      531470
  7         Y = B(I,J)                                                        531480
  7         B(I,J) = B(II,J)                                                  531490
  7         B(II,J) = Y                                                       531500
  4      30 CONTINUE                                                          531510
  4      40 CONTINUE                                                          531520
  7         DO  70 J=IP1,N                                                    531530
  7         Y = B(J,I)/B(I,I)                                                 531540
  7         IF( REAL(Y).EQ.0.0  .AND.  AIMAG(Y).EQ.0.0 )     GO TO  68        531550
  7         DO  50 K=1,N                                                      531560
  7         A(J,K) = A(J,K) - Y*A(I,K)                                        531570
  4      50 CONTINUE                                                          531580
  7         DO  60 K=IP1,N                                                    531590
  7         B(J,K) = B(J,K) - Y*B(I,K)                                        531600
  4      60 CONTINUE                                                          531610
  4      68 CONTINUE                                                          531620
  4      70 CONTINUE                                                          531630
  7         B(IP1,I) = CMPLX(0.0,0.0)                                         531640
  4      78 CONTINUE                                                          531650
  4      80 CONTINUE                                                          531660
  1   C                                                                       53167
  7         DO 100 I=1,N                                                      531680
  7         DO  90 J=1,N                                                      531690
  7         X(I,J) = CMPLX(0.0,0.0)                                           531700
  4      90 CONTINUE                                                          531710
  7         X(I,I) = CMPLX(1.0,0.0)                                           531720
  3     100 CONTINUE                                                          531730
  1   C                                                                       531740
  7         NM2 = N - 2                                                       531750
  7         IF( NM2.LT.1 )      GO TO 270                                     531760
  7         DO 260 J=1,NM2                                                    531770
  7         JM2 = NM1 - J                                                     531780
  7         JP1 = J + 1                                                       531790
  7         DO 250 II=1,JM2                                                   531800
  7         I = N + 1 - II                                                    531810
  7         IM1 = I - 1                                                       531820
  7         IMJ = I - J                                                       531830
  1   C                                                                       531840
  7         W = A(I,J)                                                        531850
  7         Z = A(IM1,J)                                                      531860
  7         IF( ABS(REAL(W)) + ABS(AIMAG(W)).LE.                              531870
  6        $    ABS(REAL(Z)) + ABS(AIMAG(Z)) )     GO TO 140                  531880
  7         DO 120 K=J,N                                                      531890
  7         Y = A(I,K)                                                        531900
  7         A(I,K) = A(IM1,K)                                                 531910
  7         A(IM1,K) = Y                                                      531920
  3     120 CONTINUE                                                          531930
  7         DO 130 K=IM1,N                                                    531940
  7         Y = B(I,K)                                                        531950
  7         B(I,K) = B(IM1,K)                                                 531960
  7         B(IM1,K) = Y                                                      531970
  3     130 CONTINUE                                                          531980
  3     140 CONTINUE                                                          531990
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  1   C                                                                       532000
  7         Z = A(I,J)                                                        532010
  7         IF( REAL(Z).EQ.0.0  .AND.  AIMAG(Z).EQ.0.0 )      GO TO 170       532020
  7         Y = Z/A(IM1,J)                                                    532030
  7         DO 150 K=JP1,N                                                    532040
  7         A(I,K) = A(I,K) - Y*A(IM1,K)                                      532050
  3     150 CONTINUE                                                          53767
  7         DO 160 K=IM1,N                                                    532070
  7         B(I,K) = B(I,K) - Y*B(IM1,K)                                      532080
  3     160 CONTINUE                                                          532090
  3     170 CONTINUE                                                          532100
  1   C                                                                       532110
  7         W = B(I,IM1)                                                      532120
  7         Z = B(I,I)                                                        532130
  7         IF( ABS(REAL(W)) + ABS(AIMAG(W)).LE.                              532140
  6        $    ABS(REAL(Z)) + ABS(AIMAG(Z)) )     GO TO 210                  532150
  7         DO 180 K=1,I                                                      532160
  7         Y = B(K,I)                                                        532170
  7         B(K,I) = B(K,IM1)                                                 532180
  7         B(K,IM1) = Y                                                      532190
  3     180 CONTINUE                                                          532200
  7         DO 190 K=1,N                                                      532210
  7         Y = A(K,I)                                                        532220
  7         A(K,I) = A(K,IM1)                                                 532230
  7         A(K,IM1) = Y                                                      532240
  3     190 CONTINUE                                                          532250
  7         DO 200 K=IMJ,N                                                    532260
  7         Y = X(K,I)                                                        532270
  7         X(K,I) = X(K,IM1)                                                 532280
  7         X(K,IM1) = Y                                                      532290
  3     200 CONTINUE                                                          532300
  3     210 CONTINUE                                                          532310
  1   C                                                                       532320
  7         Z = B(I,IM1)                                                      532330
  7         IF( REAL(Z).EQ.0.0  .AND.  AIMAG(Z).EQ.0.0 )      GO TO 249       532340
  7         Y = Z/B(I,I)                                                      532350
  7         DO 220 K=1,IM1                                                    532360
  7         B(K,IM1) = B(K,IM1) - Y*B(K,I)                                    532370
  3     220 CONTINUE                                                          532380
  7         B(I,IM1) = CMPLX(0.0,0.0)                                         538320
  7         DO 230 K=1,N                                                      532400
  7         A(K,IM1) = A(K,IM1) - Y*A(K,I)                                    532410
  3     230 CONTINUE                                                          532420
  7         DO 240 K=IMJ,N                                                    532430
  7         X(K,IM1) = X(K,IM1) - Y*X(K,I)                                    532440
  3     240 CONTINUE                                                          532450
  3     249 CONTINUE                                                          532460
  1   C                                                                       532470
  3     250 CONTINUE                                                          532480
  7         A(JP1+1,J) = CMPLX(0.0,0.0)                                       532490
  3     260 CONTINUE                                                          532500
  3     270 CONTINUE                                                          532510
  1   C                                                                       532520
  7         RETURN                                                            532530
```

```
      PROJECT: CTJC197             MEMBER: MAIN2              DATE: 87/09/24
      GROUP:   STB1                LEVEL:  01.00              TIME: 12:02
      TYPE:    FORT                USERID: CTJC197            PAGE: 45 OF 50
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  7         END                                                         532540
  7         SUBROUTINE SOLVE                                            532550
  6        $( N, A, NA, B, NB, X, NX, ITER, EIGA, EIGB )                532560
  1   C                                                                 532570
  7         COMPLEX S                                                   532580
  7         COMPLEX W                                                   532590
  7         COMPLEX Y                                                   532600
  7         COMPLEX Z                                                   532610
  7         COMPLEX A(NA,N)                                             532620
  7         COMPLEX B(NB,N)                                             532630
  7         COMPLEX X(NX,N)                                             532640
  7         COMPLEX EIGA(N)                                             532650
  7         COMPLEX EIGB(N)                                             532660
  7         COMPLEX ANNM1                                               53267
  7         COMPLEX ALFM                                                532680
  7         COMPLEX BETM                                                532690
  7         COMPLEX D                                                   532700
  7         COMPLEX SL                                                  532710
  7         COMPLEX DEN                                                 532720
  7         COMPLEX NUM                                                 532730
  7         COMPLEX ANM1M1                                              532740
  1   C                                                                 532750
  7         REAL DO                                                     532760
  7         REAL D1                                                     532770
  7         REAL D2                                                     532780
  7         REAL EO                                                     532790
  7         REAL E1                                                     532800
  7         REAL C                                                      532810
  7         REAL EPSA                                                   532820
  7         REAL EPSB                                                   532830
  7         REAL SS                                                     532840
  7         REAL R                                                      532850
  7         REAL ANORM                                                  532860
  7         REAL BNORM                                                  532870
  7         REAL ANI                                                    532880
  7         REAL BNI                                                    532890
  1   C                                                                 532900
  7         INTEGER ITER(N)                                             532910
  1   C                                                                 532920
  7         NN = N                                                      532930
  7         ANORM = 0.0                                                 532940
  7         BNORM = 0.0                                                 532950
  7         DO 30 I=1,N                                                 532960
  7         ANI = 0.0                                                   532970
  7         IF( I.EQ.1 )      GO TO 10                                  532980
  7         Y = A(I,I-1)                                                532990
  7         ANI = ANI + ABS(REAL(Y)) + ABS(AIMAG(Y))                    533000
  4      10 CONTINUE                                                    533010
  7         BNI = 0.0                                                   533020
  7         DO 20 J=I,N                                                 533030
  7         ANI = ANI + ABS(REAL(A(I,J))) + ABS(AIMAG(A(I,J)))          533040
  7         BNI = BNI + ABS(REAL(B(I,J))) + ABS(AIMAG(B(I,J)))          533050
  4      20 CONTINUE                                                    533060
  7         IF( ANI.GT.ANORM )      ANORM = ANI                         533070
```

```
      PROJECT: CTJC197             MEMBER: MAIN2                DATE: 87/09/24
      GROUP:   STB1                LEVEL:  01.00                TIME: 12:02
      TYPE:    FORT                USERID: CTJC197              PAGE: 46 OF 50
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  7         IF( BNI.GT.BNORM )     BNORM = BNI
  4      30 CONTINUE
  1   C
  7         IF( ANORM.EQ.0.0 )     ANORM = 1.0
  7         IF( BNORM.EQ.0.0 )     BNORM = 1.0
  7         EPSA = ANORM
  7         EPSB = BNORM
  4      40 CONTINUE
  7         EPSA = EPSA/2.0
  7         EPSB = EPSB/2.0
  7         C = ANORM + EPSA
  7         IF( C.GT.ANORM )     GO TO 40
  7         IF( N.LE.1 )     GO TO 320
  4      50 CONTINUE
  7         ITS = 0
  7         NM1 = NN - 1
  4      60 CONTINUE
  7         D2 = ABS(REAL(A(NN,NN))) + ABS(AIMAG(A(NN,NN)))
  7         DO 70 LB=2,NN
  7         L = NN + 2 - LB
  7         SS = D2
  7         Y = A(L-1,L-1)
  7         D2 = ABS(REAL(Y)) + ABS(AIMAG(Y))
  7         SS = SS + D2
  7         Y = A(L,L-1)
  7         R = SS + ABS(REAL(Y)) + ABS(AIMAG(Y))
  7         IF( R.EQ.SS )     GO TO 80
  4      70 CONTINUE
  7         L = 1
  4      80 CONTINUE
  7         IF( L.EQ.NN )     GO TO 320
  7         IF( ITS.LT.30 )     GO TO 90
  7         ITER(NN) = -1
  7         IF( ABS(REAL(A(NN,NM1))) + ABS(AIMAG(A(NN,NM1))).GT.
  6        $0.8*ABS(REAL(ANNM1)) +     ABS(AIMAG(ANNM1))     )     GO TO 999
  4      90 CONTINUE
  7         IF( ITS.EQ.10  .OR.  ITS.EQ.20 )     GO TO 110
  1   C
  7         ANNM1 = A(NN,NM1)
  7         ANM1M1 = A(NM1,NM1)
  7         S = A(NN,NN)*B(NM1,NM1) - ANNM1*B(NM1,NN)
  7         W = ANNM1*B(NN,NN)*
  6        $    (A(NM1,NN)*B(NM1,NM1) - ANM1M1*B(NM1,NN))
  7         Y = (ANM1M1*B(NN,NN) - S)/2.0
  7         Z = CSQRT(Y*Y + W)
  7         IF( REAL(Z).EQ.0.0  .AND.  AIMAG(Z).EQ.0.0 )     GO TO 100
  7         DO = REAL(Y/Z)
  7         IF( DO.LT.0.0 )     Z = -Z
  3     100 CONTINUE
  7         DEN = (Y + Z)*B(NM1,NM1)*B(NN,NN)
  7         IF(  REAL(DEN).EQ.0.0   .AND.
  6        $    AIMAG(DEN).EQ.0.0 )
  6        $DEN = CMPLX(EPSA,0.0)
  7         NUM = (Y + Z)*S - W
```

```
START
 COL   ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  7          GO TO 120                                                          533620
  3      110 CONTINUE                                                           533630
  7          Y = A(NM1,NN-2)                                                    533640
  7          NUM = CMPLX(ABS(REAL(ANNM1)) + ABS(AIMAG(ANNM1)),                  533650
  6         $            ABS(REAL(Y))     + ABS(AIMAG(Y))     )                 533660
  7          DEN = CMPLX(1.0,0.0)                                               53367
  3      120 CONTINUE                                                           533680
  7          IF( NN.EQ.L+1 )      GO TO 140                                     533690
  7          D1 = ABS(REAL(A(NN,NN))) + ABS(AIMAG(A(NN,NN)))                    533700
  7          D2 = ABS(REAL(A(NM1,NM1))) + ABS(AIMAG(A(NM1,NM1)))                533710
  7          E1 = ABS(REAL(ANNM1)) + ABS(AIMAG(ANNM1))                          533720
  7          NL = NN - (L + 1)                                                  533730
  7          DO 130 MB=1,NL                                                     533740
  7          M = NN - MB                                                        533750
  7          EO = E1                                                            533760
  7          Y = A(M,M-1)                                                       533770
  7          E1 = ABS(REAL(Y)) + ABS(AIMAG(Y))                                  533780
  7          DO = D1                                                            533790
  7          D1 = D2                                                            533800
  7          Y = A(M-1,M-1)                                                     533810
  7          D2 = ABS(REAL(Y)) + ABS(AIMAG(Y))                                  533820
  7          Y = A(M,M)*DEN - B(M,M)*NUM                                        533830
  7          DO = (DO + D1 + D2)*( ABS(REAL(Y)) + ABS(AIMAG(Y)) )               533840
  7          EO = EO*E1*( ABS(REAL(DEN)) + ABS(AIMAG(DEN)) ) + DO               533850
  7          IF( EO.EQ.DO )      GO TO 150                                      533860
  3      130 CONTINUE                                                           533870
  3      140 CONTINUE                                                           533880
  7          M = L                                                              533890
  3      150 CONTINUE                                                           533900
  7          ITS = ITS + 1                                                      533910
  7          W = A(M,M)*DEN - B(M,M)*NUM                                        533920
  7          Z = A(M+1,M)*DEN                                                   533930
  7          D1 = ABS(REAL(Z)) + ABS(AIMAG(Z))                                  533940
  7          D2 = ABS(REAL(W)) + ABS(AIMAG(W))                                  533950
  7          LOR1 = 1                                                           533960
  7          NNORN = N                                                          533970
  7          DO 310 I=M,NM1                                                     533980
  7          J = I + 1                                                          533990
  7          IF( I.EQ.M )      GO TO 170                                        534000
  7          W = A(I,I-1)                                                       534010
  7          Z = A(J,I-1)                                                       534020
  7          D1 = ABS(REAL(Z)) + ABS(AIMAG(Z))                                  534030
  7          D2 = ABS(REAL(W)) + ABS(AIMAG(W))                                  534040
  7          IF( D1.EQ.0.0 )      GO TO 60                                      534050
  3      170 CONTINUE                                                           534060
  7          IF( D2.GT.D1 )      GO TO 190                                      534070
  7          DO 180 K=I,NNORN                                                   534080
  7          Y = A(I,K)                                                         534090
  7          A(I,K) = A(J,K)                                                    534100
  7          A(J,K) = Y                                                         534110
  7          Y = B(I,K)                                                         534120
  7          B(I,K) = B(J,K)                                                    534130
  7          B(J,K) = Y                                                         534140
  3      180 CONTINUE                                                           534150
```

```
START
 COL   ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  7          IF( I.GT.M )      A(I,I-1) = A(J,I-1)                             534160
  7          IF( D2.EQ.0.0 )      GO TO 220                                    534170
  7          Y = CMPLX( REAL(W)/D1, AIMAG(W)/D1 )/                             534180
  6         $    CMPLX( REAL(Z)/D1, AIMAG(Z)/D1 )                              534190
  7          GO TO 200                                                         534200
  3      190 CONTINUE                                                          534210
  7          Y = CMPLX( REAL(Z)/D2, AIMAG(Z)/D2 )/                             534220
  6         $    CMPLX( REAL(W)/D2, AIMAG(W)/D2 )                              534230
  3      200 CONTINUE                                                          534240
  7          DO 210 K=I,NNORN                                                  534250
  7          A(J,K) = A(J,K) - Y*A(I,K)                                        534260
  7          B(J,K) = B(J,K) - Y*B(I,K)                                        534270
  3      210 CONTINUE                                                          534280
  3      220 CONTINUE                                                          534290
  7          IF( I.GT.M )      A(J,I-1) = CMPLX(0.0,0.0)                       534300
  7          Z = B(J,I)                                                        534310
  7          W = B(J,J)                                                        534320
  7          D1 = ABS(REAL(Z)) + ABS(AIMAG(Z))                                 534330
  7          D2 = ABS(REAL(W)) + ABS(AIMAG(W))                                 534340
  7          IF( D1.EQ.0.0 )      GO TO 60                                     534350
  7          IF( D2.GT.D1 )      GO TO 270                                     534360
  7          DO 230 K=LOR1,J                                                   534370
  7          Y = A(K,J)                                                        534380
  7          A(K,J) = A(K,I)                                                   534390
  7          A(K,I) = Y                                                        534400
  7          Y = B(K,J)                                                        534410
  7          B(K,J) = B(K,I)                                                   534420
  7          B(K,I) = Y                                                        534430
  3      230 CONTINUE                                                          534440
  7          IF( I.EQ.NM1 )      GO TO 240                                     534450
  7          Y = A(J+1,J)                                                      534460
  7          A(J+1,J) = A(J+1,I)                                               534470
  7          A(J+1,I) = Y                                                      534480
  3      240 CONTINUE                                                          534490
  7          DO 250 K=1,N                                                      534500
  7          Y = X(K,J)                                                        534510
  7          X(K,J) = X(K,I)                                                   534520
  7          X(K,I) = Y                                                        534530
  3      250 CONTINUE                                                          534540
  7          B(J,I) = CMPLX(0.0,0.0)                                           534550
  7          IF( D2.EQ.0.0 )      GO TO 310                                    534560
  7          Z = CMPLX( REAL(W)/D1, AIMAG(W)/D1 )/                             534570
  6         $    CMPLX( REAL(Z)/D1, AIMAG(Z)/D1 )                              534580
  7          GO TO 280                                                         534590
  3      270 CONTINUE                                                          534600
  7          Z = CMPLX( REAL(Z)/D2, AIMAG(Z)/D2 )/                             534610
  6         $    CMPLX( REAL(W)/D2, AIMAG(W)/D2 )                              534620
  3      280 CONTINUE                                                          534630
  7          DO 290 K=LOR1,J                                                   534640
  7          A(K,I) = A(K,I) - Z*A(K,J)                                        534650
  7          B(K,I) = B(K,I) - Z*B(K,J)                                        534660
  3      290 CONTINUE                                                          53467
  7          B(J,I) = CMPLX(0.0,0.0)                                           534680
  7          IF( I.LT.NM1 )      A(I+2,I) = A(I+2,I) - Z*A(I+2,J)              534690
```

```
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

 7          DO 300 K=1,N
 7          X(K,I) = X(K,I) - Z*X(K,J)
 3      300 CONTINUE
 3      310 CONTINUE
 7          GO TO 60
 3      320 CONTINUE
 7          EIGA(NN) = A(NN,NN)
 7          EIGB(NN) = B(NN,NN)
 7          IF( NN.EQ.1 )      GO TO 330
 7          ITER(NN) = ITS
 7          NN = NM1
 7          IF( NN.GT.1 )      GO TO 50
 7          ITER(1) = 0
 7          GO TO 320
 3      330 CONTINUE
 7          M = N
 3      340 CONTINUE
 7          ALFM = A(M,M)
 7          BETM = B(M,M)
 7          B(M,M) = CMPLX(1.0,0.0)
 7          L = M - 1
 7          IF( L.EQ.0 )      GO TO 370
 3      350 CONTINUE
 7          L1 = L + 1
 7          SL = CMPLX(0.0,0.0)
 7          DO 360 J=L1,M
 7          SL = SL + B(J,M)*(BETM*A(L,J) - ALFM*B(L,J))
 3      360 CONTINUE
 7          Y = BETM*A(L,L) - ALFM*B(L,L)
 7          IF(  REAL(Y).EQ.0.0    .AND.
 6         $     AIMAG(Y).EQ.0.0 )
 6         $Y = CMPLX( (EPSA+EPSB)/2.0, 0.0 )
 7          B(L,M) = -SL/Y
 7          L = L - 1
 3      370 CONTINUE
 7          IF( L.GT.0 )      GO TO 350
 7          M = M - 1
 7          IF( M.GT.0 )      GO TO 340
 7          M = N
 3      380 CONTINUE
 7          DO 400 I=1,N
 7          S = CMPLX(0.0,0.0)
 7          DO 390 J=1,M
 7          S = S + X(I,J)*B(J,M)
 3      390 CONTINUE
 7          X(I,M) = S
 3      400 CONTINUE
 7          M = M - 1
 7          IF( M.GT.0 )      GO TO 380
 7          M = N
 3      410 CONTINUE
 7          SS = 0.0
 7          DO 420 I=1,N
 7          R = ABS(REAL(X(I,M))) + ABS(AIMAG(X(I,M)))
```

```
      PROJECT: CTJC197             MEMBER: MAIN2              DATE: 87/09/24
      GROUP:   STB1                LEVEL:  01.00              TIME: 12:02
      TYPE:    FORT                USERID: CTJC197            PAGE: 50 OF 50
START
 COL  ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8

  7         IF( R.LT.SS )     GO TO 418                                  535240
  7         SS = R                                                       535250
  7         D = X(I,M)                                                   535260
  3     418 CONTINUE                                                     535270
  3     420 CONTINUE                                                     535280
  7         IF( SS.EQ.0.0 )     GO TO 440                                535290
  7         DO 430 I=1,N                                                 535300
  7         X(I,M) = X(I,M)/D                                            535310
  3     430 CONTINUE                                                     535320
  3     440 CONTINUE                                                     535330
  7         M = M - 1                                                    535340
  7         IF( M.GT.0 )     GO TO 410                                   535350
  3     999 CONTINUE                                                     535360
  7         RETURN                                                       535370
  7         END                                                          535380
```