LAMPIRAN

Lampiran 1 : Contoh Program Fortran

```
C
C    riser1.for
C    13 Juli, 2006
C

      COMMON/TYPE/ITYPE
      COMMON/INFORM/INFO,INFILE
      COMMON/INPUT1/PI,GRAV,RHO,NELM,NODES,NDOF0,NDOF
      COMMON/INPUT2/X(25),ELL(24),RISLEN
      COMMON/INPUT3/WAIR,WDISP,WMUD,WADD,RISIN,E,TTOP
      COMMON/INPUT4/DEPTH,HLBJ,DIAO,DIAE,CD,CM
      COMMON/INPUT5/ALPHAP,BETAP,PRCNTP,AMP
      COMMON/INPUT6/VCMWL,WAVEH,WAVET
      COMMON/INPUT7/ALPHAC,BETAC
      COMMON/INPUT8/NTIME,DTIME,BTIME
      COMMON/MDKMAT/STIFF(50,50),EMASS(50,50),DAMP(50,50)
      COMMON/FVEC/FDISC(50,50),FC(50),FW(50,121)

COMMON/KINVEC/UST(50),UDYN(50,121),U1DYN(50,121),U2DYN(50,121
)
      COMMON/RELAT/VEL(25,121),CDRAG,REL(25,121)
      COMMON/BND/UTOT(50,50),UMAX(50),UMIN(50)

      CHARACTER*8  INFILE
      CHARACTER*80 INFO

      CALL INPUT
      CALL MATRIX
      IF(ITYPE.EQ.0) THEN
        CALL FORCES
        CALL DISP
        CALL BOUND
        CALL OUTPUT
      ELSE
        CALL OUT2
      END IF

      STOP
      END

*************************************************************
```

```
  SUBROUTINE OUTPUT

    COMMON/INFORM/INFO,INFILE
    COMMON/INPUT1/PI,GRAV,RHO,NELM,NODES,NDOF0,NDOF
    COMMON/INPUT2/X(25),ELL(24),RISLEN
    COMMON/INPUT8/NTIME,DTIME,BTIME

COMMON/KINVEC/UST(50),UDYN(50,121),U1DYN(50,121),U2DYN(50,12)
    COMMON/MDKMAT/STIFF(50,50),EMASS(50,50),DAMP(50,50)
    COMMON/FVEC/FDISC(50,50),FC(50),FW(50,121)
    COMMON/BND/UTOT(50,50),UMAX(50),UMIN(50)

    CHARACTER*8  INFILE
    CHARACTER*80 INFO

  01 FORMAT(5E16.8)

    OPEN(5, FILE='E:\'//INFILE//'.UST', STATUS='NEW')
    WRITE(5,*)X(1),0.0,UST(1)*57.2958
    DO 100 I=2,NODES
      WRITE(5,*)X(I),UST(2*I-2),UST(2*I-1)*57.2958
 100 CONTINUE


C    BOUNDS

    OPEN(4,FILE='E:\'//INFILE//'.X0',STATUS='NEW')
    WRITE(4,*)X(1),0.0,0.0
    DO 120 I=2,NDOF,2
      WRITE(4,*)X((I+2)/2),UMAX(I),UMIN(I)
 120 CONTINUE
    CLOSE(4)

    OPEN(7,FILE='E:\'//INFILE//'.ZER',STATUS='NEW')
    DO 150 I=1,NDOF
      IF(I.LT.4) J1=1
      IF(I.GE.4) J1=I-3
      IF(I.LT.NDOF-3) J2=I+3
      IF(I.GE.NDOF-3) J2=NDOF
      WRITE(7,*)(EMASS(I,J),J=J1,J2)
      WRITE(7,*)(STIFF(I,J),J=J1,J2)
 150 CONTINUE
    WRITE(7,*)
        WRITE(7,*)
    WRITE(7,*)(UST(I),I=1,NDOF)
    WRITE(7,*)
        WRITE(7,*)
        DO 160 IT=1,NTIME
      WRITE(7,*)(UDYN(I,IT),I=1,NDOF)
      WRITE(7,*)(U1DYN(I,IT),I=1,NDOF)
      WRITE(7,*)(U2DYN(I,IT),I=1,NDOF)
 160 CONTINUE
    CLOSE(7)

    OPEN(8,FILE='E:\'//INFILE//'.S0',STATUS='NEW')
    WRITE(8,1)X(1),0.0,0.0
    DO 200 I=1,NELM
      J=2*I-1
      Z=X(I) + ( X(I+1)-X(I) ) / 2.0
      WRITE(8,1)Z,UMAX(J),UMIN(J)
 200 CONTINUE
    WRITE(8,1)X(NELM+1),0.0,0.0
    CLOSE(8)

    RETURN
    END

*****************************************************************

    SUBROUTINE INPUT

    COMMON/TYPE/ITYPE
    COMMON/INFORM/INFO,INFILE
    COMMON/INPUT1/PI,GRAV,RHO,NELM,NODES,NDOF0,NDOF
    COMMON/INPUT2/X(25),ELL(24),RISLEN
    COMMON/INPUT3/WAIR,WDISP,WMUD,WADD,RISIN,E,TTOP
    COMMON/INPUT4/DEPTH,HLBJ,DIAO,DIAE,CD,CM
    COMMON/INPUT5/ALPHAP,BETAP,PRCNTP,AMP
    COMMON/INPUT6/VCMWL,WAVEH,WAVET
    COMMON/INPUT7/ALPHAC,BETAC
    COMMON/INPUT8/NTIME,DTIME,BTIME

    CHARACTER*8 INFILE
    CHARACTER*80 INFO

  10 FORMAT(A)

    PI=3.14159
    GRAV=32.17
    RHO=1.99

    OPEN(1,FILE='FILE.NAM',STATUS='OLD')
    READ(1,10)INFILE
    READ(1,*)ITYPE
    CLOSE(1)
    OPEN(1,FILE='T_0500_1.DAT',STATUS='OLD')
    READ(1,10)INFO

    READ(1,*)NELM
    NODES=NELM+1
    NDOF0=2*NODES
    NDOF=NDOF0-1
    READ(1,*)(X(I),I=1,NODES)
    RISLEN=X(NODES)-X(1)
    DO 100 I=1,NELM
      ELL(I)=X(I+1)-X(I)
 100 CONTINUE

    READ(1,*)
    READ(1,*)WAIR,WDISP,WMUD,WADD,RISIN,E,TTOP
    READ(1,*)DEPTH,HLBJ,DIAO,DIAE,CD,CM
    READ(1,*)BETAP,PRCNTP,AMP
    READ(1,*)VCMWL,WAVEH,WAVET
    READ(1,*)ALPHAC,BETAC
    READ(1,*)NTIME,DTIME,BTIME

    CLOSE(1)

    RETURN
    END

******************************************************************

    SUBROUTINE MATRIX

    COMMON/INPUT1/PI,GRAV,RHO,NELM,NODES,NDOF0,NDOF
    COMMON/INPUT2/X(25),ELL(24),RISLEN
    COMMON/INPUT3/WAIR,WDISP,WMUD,WADD,RISIN,E,TTOP
    COMMON/INPUT4/DEPTH,HLBJ,DIAO,DIAE,CD,CM
    COMMON/INPUT5/ALPHAP,BETAP,PRCNTP,AMP
    COMMON/INPUT7/ALPHAC,BETAC
    COMMON/MDKMAT/STIFF(50,50),EMASS(50,50),DAMP(50,50)
    COMMON/FVEC/FDISC(50,50),FC(50),FW(50,121)

    DO 20 I=1,NDOF0
      DO 10 J=1,NDOF0
        STIFF(I,J)=0.0
        EMASS(I,J)=0.0
        FDISC(I,J)=0.0
  10   CONTINUE
  20 CONTINUE
```

```
DO 30 IELM=1,NELM
    Z=X(IELM)
    CEI=2.0*E*RISIN/ELL(IELM)**3
    IF( X(IELM+1).LE.DEPTH) THEN
      CT0 = ( TTOP - ( WAIR + WMUD ) * ( RISLEN - ( Z - HLBJ ) )
  *          + WDISP * ( DEPTH -  Z ) ) / ( 30.0 * ELL(IELM) )
      CTP = WAIR + WMUD - WDISP
      CMASS = 1.0/GRAV * ( WAIR + WMUD + WADD )  *
ELL(IELM)/420.0
      CALL MATGEN(CEI,CT0,CTP,CMASS,IELM)
    ELSE
      CT0 = ( TTOP - ( WAIR + WMUD ) * ( RISLEN - ( Z - HLBJ ) ) )
  *          / ( 30.0 *ELL(IELM) )
      CTP = WAIR + WMUD
      CMASS = 1.0/GRAV * ( WAIR + WMUD ) * ELL(IELM)/420.0
      CALL MATGEN(CEI,CT0,CTP,CMASS,IELM)
    ENDIF
 30 CONTINUE
   IDOF=NDOF0-1
   ALPHAP=BETAP * STIFF(IDOF,IDOF)
   STIFF(IDOF,IDOF)=STIFF(IDOF,IDOF) + ALPHAP

   DO 50 I=1,NDOF0
    DO 40 J=1,NDOF0
      DAMP(I,J)=ALPHAC*EMASS(I,J) + BETAC*STIFF(I,J)
 40   CONTINUE
 50 CONTINUE

   RETURN
   END

****************************************************************

   SUBROUTINE MATGEN(CEI,CT0,CTP,CMASS,IELM)

   COMMON/INPUT2/X(25),ELL(24),RISLEN
   COMMON/MDKMAT/STIFF(50,50),EMASS(50,50),DAMP(50,50)
   COMMON/FVEC/FDISC(50,50),FC(50),FW(50,121)

   DIMENSION ELKEI(4,4),ELKT0(4,4),ELKTP(4,4),ELK(4,4),ELM(4,4),
  *   FMAT(4,4)
   ELKEI(1,1)=CEI * 6.0
   ELKEI(1,2)=CEI * 3.0*ELL(IELM)
   ELKEI(1,3)=CEI * (-6.0)
   ELKEI(1,4)=CEI * 3.0*ELL(IELM)
   ELKEI(2,2)=CEI * 2.0*ELL(IELM)**2
   ELKEI(2,3)=CEI * (-3.0)*ELL(IELM)
   ELKEI(2,4)=CEI * ELL(IELM)**2
```

```
      ELKEI(3,3)=CEI * 6.0
      ELKEI(3,4)=CEI * (-3.0)*ELL(IELM)
      ELKEI(4,4)=CEI * 2.0*ELL(IELM)**2

      ELKT0(1,1)=CT0 * 36.0
      ELKT0(1,2)=CT0 * 3.0*ELL(IELM)
      ELKT0(1,3)=CT0 * (-36.0)
      ELKT0(1,4)=CT0 * 3.0*ELL(IELM)
      ELKT0(2,2)=CT0 * 4.0*ELL(IELM)**2
      ELKT0(2,3)=CT0 * (-3.0)*ELL(IELM)
      ELKT0(2,4)=CT0 * (-1.0)*ELL(IELM)**2
      ELKT0(3,3)=CT0 * 36.0
      ELKT0(3,4)=CT0 * (-3.0)*ELL(IELM)
      ELKT0(4,4)=CT0 * 4.0*ELL(IELM)**2

      ELKTP(1,1)= CTP * 3.0/5.0
      ELKTP(1,2)= CTP * ELL(IELM)/10.0
      ELKTP(1,3)= CTP * (-3.0)/5.0
      ELKTP(1,4)= CTP * 0.0
      ELKTP(2,2)= CTP * ELL(IELM)**2/30.0
      ELKTP(2,3)= CTP * (-ELL(IELM))/10.0
      ELKTP(2,4)= CTP * (-ELL(IELM)**2)/60.0
      ELKTP(3,3)= CTP * 3.0/5.0
      ELKTP(3,4)= CTP * 0.0
      ELKTP(4,4)= CTP * ELL(IELM)**2/10.0

      ELM(1,1)=CMASS * 156.0
      ELM(1,2)=CMASS * 22.0*ELL(IELM)
      ELM(1,3)=CMASS * 54.0
      ELM(1,4)=CMASS * (-13.0)*ELL(IELM)
      ELM(2,2)=CMASS * 4.0*ELL(IELM)**2
      ELM(2,3)=CMASS * 13.0*ELL(IELM)
      ELM(2,4)=CMASS * (-3.0)*ELL(IELM)**2
      ELM(3,3)=CMASS * 156.0
      ELM(3,4)=CMASS * (-22.0)*ELL(IELM)
      ELM(4,4)=CMASS * 4.0*ELL(IELM)**2

      FMAT(1,1)=  0.35*ELL(IELM)
      FMAT(1,3)=  0.15*ELL(IELM)
      FMAT(2,1)=  ELL(IELM)**2/20.0
      FMAT(2,3)=  ELL(IELM)**2/30.0
      FMAT(3,1)=  0.15*ELL(IELM)
      FMAT(3,3)=  0.35*ELL(IELM)
      FMAT(4,1)= -1.0*ELL(IELM)**2/30.0
C     FMAT(4,1)= -1.0*ELL(IELM)**2/7.5
      FMAT(4,3)= -1.0*ELL(IELM)**2/20.0

      DO 20 I=1,4
      DO 10 J=1,4
        ELKEI(J,I)=ELKEI(I,J)
        ELKT0(J,I)=ELKT0(I,J)
        ELKTP(J,I)=ELKTP(I,J)
        ELK(I,J)=ELKEI(I,J)+ELKT0(I,J)+ELKTP(I,J)
        ELM(J,I)=ELM(I,J)
10      CONTINUE
      FMAT(I,2)=0.0
      FMAT(I,4)=0.0
20 CONTINUE

    ICNT=2*IELM-1
    II=0
    DO 60 I=ICNT,ICNT+3
      II=II+1
      JJ=0
      DO 50 J=ICNT,ICNT+3
        JJ=JJ+1
        STIFF(I,J)=STIFF(I,J) + ELK(II,JJ)
        EMASS(I,J)=EMASS(I,J) + ELM(II,JJ)
        FDISC(I,J)=FDISC(I,J) + FMAT(II,JJ)
50      CONTINUE
60 CONTINUE

    RETURN
    END

******************************************************************

    SUBROUTINE FORCES

    COMMON/INPUT1/PI,GRAV,RHO,NELM,NODES,NDOF0,NDOF
    COMMON/INPUT2/X(25),ELL(24),RISLEN
    COMMON/INPUT4/DEPTH,HLBJ,DIAO,DIAE,CD,CM
    COMMON/INPUT5/ALPHAP,BETAP,PRCNTP,AMP
    COMMON/INPUT6/VCMWL,WAVEH,WAVET
    COMMON/INPUT8/NTIME,DT,BTIME
    COMMON/MDKMAT/STIFF(50,50),EMASS(50,50),DAMP(50,50)
    COMMON/FVEC/FDISC(50,50),FC(50),FW(50,121)
    COMMON/RELAT/VEL(25,121),CDRAG,REL(25,121)

    DIMENSION F(25),FI(25,121)

    CDRAG= 0.5 * RHO * DIAE * CD
    CINERT= 0.25 * PI * RHO * CM * DIAE**2
    WAVEL=GRAV * WAVET**2 / (2.0 * PI)
    IDOF=NDOF0-1
    OFFSET=PRCNTP * DEPTH
```

```
   SLOPE=VCMWL / ( DEPTH - HLBJ )

   DO 10 I=1,NODES
     Z=X(I)
     F(I)= CDRAG * ( Z * SLOPE )**2
     EXPO= EXP(2.0 * PI * (Z-DEPTH) / WAVEL)
     IF (Z.GT.DEPTH) EXPO=0.0
     CV=PI * WAVEH / WAVET * EXPO
     CI=2.0 * WAVEH * (PI/WAVET)**2 * EXPO
     IF (Z.GT.DEPTH) F(I)=0.0
     DO 5 IT=1,NTIME
       TIME=(IT-1)*DT
       VEL(I,IT)=CV * SIN(2.0*PI * TIME/WAVET)
       FI(I,IT)= CINERT*CI * COS(2.0*PI * TIME/WAVET)
05   CONTINUE
10 CONTINUE

   DO 20 I=1,NDOF0
     FC(I)=0.0
     DO 15 IT=1,NTIME
       FW(I,IT)=0.0
15   CONTINUE
     JCNT=-1.0
     DO 30 J=1,NDOF0,2
       JCNT=JCNT+1
       FC(I)=FC(I) + FDISC(I,J) * F(J-JCNT)
       DO 25 IT=1,NTIME
         FW(I,IT)=FW(I,IT) + FDISC(I,J) *  FI(J-JCNT,IT)
25     CONTINUE
30   CONTINUE
20 CONTINUE
   FC(IDOF)= FC(IDOF) + ALPHAP*OFFSET
   DO 40 IT=2,NTIME
     TIME=(IT-2)*DT
     OFFSET=AMP * SIN(2.0*PI * TIME/WAVET)
     FW(IDOF,IT)=FW(IDOF,IT) + ALPHAP*OFFSET
40 CONTINUE

   RETURN
   END

*************************************************************

   SUBROUTINE DISP

   COMMON/INPUT1/PI,GRAV,RHO,NELM,NODES,NDOF0,NDOF
   COMMON/INPUT2/X(25),ELL(24),RISLEN
   COMMON/INPUT3/WAIR,WDISP,WMUD,WADD,RISIN,E,TTOP
```

```
      COMMON/INPUT4/DEPTH,HLBJ,DIAO,DIAE,CD,CM
      COMMON/INPUT5/ALPHAP,BETAP,PRCNTP,AMP
      COMMON/INPUT6/VCMWL,WAVEH,WAVET
      COMMON/INPUT7/ALPHAC,BETAC
      COMMON/INPUT8/NTIME,DTIME,BTIME
      COMMON/MDKMAT/STIFF(50,50),EMASS(50,50),DAMP(50,50)
      COMMON/FVEC/FDISC(50,50),FC(50),FW(50,121)

COMMON/KINVEC/UST(50),UDYN(50,121),U1DYN(50,121),U2DYN(50,121
)
      COMMON/RELAT/VEL(25,121),CDRAG,REL(25,121)

      DIMENSION A(50,50),F(50),INDX(50)

      N=NDOF
      DO 20 I=1,NDOF
        DO 10 J=1,NDOF
          A(I,J)=STIFF(I+1,J+1)
          EMASS(I,J)=EMASS(I+1,J+1)
          DAMP(I,J)=DAMP(I+1,J+1)
          STIFF(I,J)=STIFF(I+1,J+1)
          FDISC(I,J)=FDISC(I+1,J+1)
          UDYN(I,1)=0.0
          U1DYN(I,1)=0.0
          U2DYN(I,1)=0.0
  10    CONTINUE
        F(I)=FC(I+1)
        DO 15 IT=1,NTIME
          FW(I,IT)=FW(I+1,IT)
  15    CONTINUE
  20 CONTINUE

      CALL LUDCMP(A,N,50,INDX,D0)
      CALL LUBKSB(A,N,50,INDX,F)
      CALL NEWMAR
      DO 30 I=1,NDOF
        UST(I)=F(I)
  30 CONTINUE

      RETURN
      END
```

```
    SUBROUTINE BOUND

    COMMON/INPUT1/PI,GRAV,RHO,NELM,NODES,NDOF0,NDOF
    COMMON/INPUT2/X(25),ELL(24),RISLEN
    COMMON/INPUT3/WAIR,WDISP,WMUD,WADD,RISIN,E,TTOP
    COMMON/INPUT4/DEPTH,HLBJ,DIAO,DIAE,CD,CM
    COMMON/INPUT8/NTIME,DTIME,BTIME

COMMON/KINVEC/UST(50),UDYN(50,121),U1DYN(50,121),U2DYN(50,121
)
    COMMON/BND/UTOT(50,50),UMAX(50),UMIN(50)

    DO 10 IT=NTIME-BTIME,NTIME
      JT=IT+1 - (NTIME-BTIME)
      DO 20 I=2,NDOF,2
        UTOT(I,JT)=UST(I)+UDYN(I,IT)
  20   CONTINUE
      J=0
      DO 25 I=1,NDOF-2,2
        J=J+1
        THETA0=UST(I+2)+UDYN(I+2,IT)
        THETA1=UST(I)+UDYN(I,IT)
        RO=( THETA0-THETA1 ) / ELL(J)
        UTOT(I,JT)=0.5*DIAO*E*RO / (1000.0 * 144.0)
  25   CONTINUE
  10 CONTINUE

    DO 30 I=1,NDOF-1
      UMAX(I)=UTOT(I,1)
      UMIN(I)=UTOT(I,1)
      DO 40 JT=1,BTIME+1
        IF(UTOT(I,JT).GT.UMAX(I)) UMAX(I)=UTOT(I,JT)
        IF(UTOT(I,JT).LT.UMIN(I)) UMIN(I)=UTOT(I,JT)
  40   CONTINUE
  30 CONTINUE

    RETURN
    END

*************************************************************************

    SUBROUTINE NEWMAR

    COMMON/INPUT1/PI,GRAV,RHO,NELM,NODES,NDOF0,N
    COMMON/INPUT2/X(25),ELL(24),RISLEN
    COMMON/INPUT4/DEPTH,HLBJ,DIAO,DIAE,CD,CM
    COMMON/INPUT8/NT,DT,BTIME
    COMMON/MDKMAT/K0(50,50),MASS(50,50),C(50,50)
```

```
      COMMON/FVEC/FDISC(50,50),FC(50),FOR(50,121)
      COMMON/KINVEC/UST(50),U(50,121),U1(50,121),U2(50,121)
      COMMON/RELAT/VEL(25,121),CDRAG,REL(25,121)

      REAL K(50,50),K0,MASS
      DIMENSION INDX(50),FD(50),R1(50),R2(50),RHAT(50),U1NEW(50),
     *   U1OLD(50),FDR(50)
      DATA DELTA,ALPHA/0.5,0.25/

      A0= 1.0 / (ALPHA * DT**2)
      A1= DELTA / (ALPHA * DT)
      A2= 1.0 / (ALPHA * DT)
      A3= 1.0 / (2.0 * ALPHA) - 1.0
      A4= DELTA / ALPHA - 1.0
      A5= DT / 2.0 * (DELTA / ALPHA - 2.0)
      A6= DT * (1.0 - DELTA)
      A7= DELTA * DT

      DO 20 I=1,N
       DO 10 J=1,N
        K(I,J)= K0(I,J) + A0*MASS(I,J) + A1*C(I,J)
    10  CONTINUE
    20 CONTINUE
      CALL LUDCMP(K,N,50,INDX,D)

      DO 100 IT=2,NT
       JTEST=0
       IFLAG=0
       JT=IT-1

       REL(1,IT)=VEL(1,IT)
       DO 540 I=2,NODES
         Z=X(I)
         REL(I,IT)=VEL(I,IT)-U1(2*I-2,JT)
         IF(Z.GT.DEPTH) REL(I,IT)=0.0
         FDR(I)=CDRAG*REL(I,IT)*ABS(REL(I,IT))
   540   CONTINUE

        DO 545 I=1,NDOF0
         FD(I)=0.0
         LCNT=-1
         DO 546 J=2,N,2
          LCNT=LCNT+1
          FD(I)=FD(I)+FDISC(I,J)*FDR(J-LCNT)
   546    CONTINUE
   545   CONTINUE

        DO 560 I=1,N
```

```
          FOR(I,IT)=FOR(I,IT)+FD(I)
 560      CONTINUE

      DO 30 I=1,N
        R1(I)=A0*U(I,JT) + A2*U1(I,JT) + A3*U2(I,JT)
        R2(I)=A1*U(I,JT) + A4*U1(I,JT) + A5*U2(I,JT)
        U1NEW(I)=U1(I,JT)
  30   CONTINUE

 300    CONTINUE

      DO 50 I=1,N
        RHAT(I)=0.0
        DO 40 II=1,N
          RHAT(I)= RHAT(I) + MASS(I,II)*R1(II) + C(I,II)*R2(II)
  40     CONTINUE
        RHAT(I)= RHAT(I) + FOR(I,IT)
  50   CONTINUE

      CALL LUBKSB(K,N,50,INDX,RHAT)

      DO 60 I=1,N
        U(I,IT)=RHAT(I)
  60   CONTINUE

      DO 70 I=1,N
        U2(I,IT)=A0*(U(I,IT)-U(I,JT))-A2*U1(I,JT)-A3*U2(I,JT)
        U1(I,IT)=U1(I,JT)+A6*U2(I,JT)+A7*U2(I,IT)
  70   CONTINUE

C      RELATIVE MOTION
      JTEST=JTEST+1
      ITEST=0
      DO 150 I=2,N,2
        ERR=ABS(U1(I,IT)-U1NEW(I))
        IF(U1(I,IT).EQ.0.0) THEN
          IF(ERR.GT.0.0005)ITEST=1
        ELSE IF(U1(I,IT).NE.0.0) THEN
          ERR2=ERR/ABS(U1(I,IT))
          IF(ERR2.GT.0.05)ITEST=1
        END IF
 150    CONTINUE

      IF(JTEST.GE.10)THEN
        WRITE(5,*)'EXCEEDED 10 ITERATIONS.  TIME =',IT
        GOTO 99
      END IF
```

```
        IF(ITEST.NE.0.OR.IFLAG.EQ.0) THEN
          IFLAG=IFLAG+1
          DO 200 I=1,N
            FOR(I,IT)=FOR(I,IT)-FD(I)
200         CONTINUE

          U1OLD(1)=0
          DO 210 I=2,NODES
            U1OLD(I)=U1(2*I-2,IT)
210         CONTINUE
          DO 240 I=1,NODES
            Z=X(I)
            REL(I,IT)=VEL(I,IT)-U1OLD(I)
            IF(Z.GT.DEPTH) REL(I,IT)=0.0
            FDR(I)=CDRAG*REL(I,IT)*ABS(REL(I,IT))
240         CONTINUE

          DO 245 I=1,NDOF0
            FD(I)=0.0
            LCNT=-1
            DO 246 J=2,N,2
              LCNT=LCNT+1
              FD(I)=FD(I)+FDISC(I,J)*FDR(J-LCNT)
246           CONTINUE
245         CONTINUE

          DO 260 I=1,N
            FOR(I,IT)=FOR(I,IT)+FD(I)
            U1NEW(I)=U1(I,IT)
260         CONTINUE

          GOTO 300

        END IF

 99     CONTINUE
100 CONTINUE

      RETURN
      END
**************************************************************

      SUBROUTINE OUT2

      COMMON/INFORM/INFO,INFILE
      COMMON/INPUT1/PI,GRAV,RHO,NELM,NODES,NDOF0,NDOF
      COMMON/MDKMAT/STIFF(50,50),EMASS(50,50),DAMP(50,50)
      CHARACTER*8  INFILE
      CHARACTER*80 INFO
      OPEN(2,FILE='E:\'//INFILE//'.MK',STATUS='NEW')
   05 FORMAT(5E16.8)

      WRITE(2,*)INFO
      WRITE(2,*)NDOF
      DO 10 I=1,NDOF
        WRITE(2,5)(EMASS(I+1,J+1),J=1,NDOF)
   10 CONTINUE
      DO 20 I=1,NDOF
        WRITE(2,5)(STIFF(I+1,J+1),J=1,NDOF)
   20 CONTINUE
      WRITE(5,*)'EXECUTE RISERFRE.FOR TO COMPUTE NATURAL
FREQUENCIES'
      WRITE(5,*)'AND ESTIMATE OF 3% PROPORTIONAL DAMPING
CONSIDERING'
      WRITE(5,*)'THE FIRST FOUR MODES OF VIBRATION.'
      CLOSE(5)
      RETURN
      END

******************************************************************

      SUBROUTINE LUDCMP(A,N,NP,INDX,D)

C     LU decomposition of stiffness matrix

      PARAMETER (NMAX=100,TINY=1.0E-20)
      DIMENSION A(NP,NP),INDX(NP),VV(NMAX)
      D=1.
      DO 12 I=1,N
       AAMAX=0.
       DO 11 J=1,N
        IF (ABS(A(I,J)).GT.AAMAX) AAMAX=ABS(A(I,J))
11      CONTINUE
       IF (AAMAX.EQ.0.) PAUSE 'Singular matrix.'
       VV(I)=1./AAMAX
12     CONTINUE
      DO 19 J=1,N
       IF (J.GT.1) THEN
        DO 14 I=1,J-1
         SUM=A(I,J)
         IF (I.GT.1)THEN
          DO 13 K=1,I-1
           SUM=SUM-A(I,K)*A(K,J)
13         CONTINUE
          A(I,J)=SUM
          ENDIF
14        CONTINUE
        ENDIF
        AAMAX=0.
        DO 16 I=J,N
          SUM=A(I,J)
          IF (J.GT.1)THEN
            DO 15 K=1,J-1
              SUM=SUM-A(I,K)*A(K,J)
15            CONTINUE
            A(I,J)=SUM
          ENDIF
          DUM=VV(I)*ABS(SUM)
          IF (DUM.GE.AAMAX) THEN
            IMAX=I
            AAMAX=DUM
          ENDIF
16        CONTINUE
        IF (J.NE.IMAX)THEN
          DO 17 K=1,N
            DUM=A(IMAX,K)
            A(IMAX,K)=A(J,K)
            A(J,K)=DUM
17          CONTINUE
          D=-D
          VV(IMAX)=VV(J)
        ENDIF
        INDX(J)=IMAX
        IF(J.NE.N)THEN
          IF(A(J,J).EQ.0.)A(J,J)=TINY
          DUM=1./A(J,J)
          DO 18 I=J+1,N
            A(I,J)=A(I,J)*DUM
18          CONTINUE
        ENDIF
19      CONTINUE
      IF(A(N,N).EQ.0.)A(N,N)=TINY
      RETURN
      END
```

****

```
      SUBROUTINE LUBKSB(A,N,NP,INDX,B)

C     solves set of N linear equations where AxX=B.
      DIMENSION A(NP,NP),INDX(NP),B(NP)
      II=0
      DO 12 I=1,N
```

```
       LL=INDX(I)
       SUM=B(LL)
       B(LL)=B(I)
       IF (II.NE.0)THEN
        DO 11 J=II,I-1
         SUM=SUM-A(I,J)*B(J)
11       CONTINUE
       ELSE IF (SUM.NE.0.) THEN
        II=I
       ENDIF
       B(I)=SUM
12     CONTINUE
      DO 14 I=N,1,-1
       SUM=B(I)
       IF(I.LT.N)THEN
        DO 13 J=I+1,N
         SUM=SUM-A(I,J)*B(J)
13       CONTINUE
       ENDIF
       B(I)=SUM/A(I,I)
14     CONTINUE
      RETURN
      END
```