|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 | C BI-DIRECTIONAL RESPONSE OF MULTI-STOREY BUILDING WITH SLIDING SYSTEMS.  DIMENSION DX1(11),VX1(11),AX1(11),PDX(11),PAX(11)  DIMENSION DX2(11),VX2(11),AX2(11),AAX(11)  DIMENSION DY1(11),VY1(11),AY1(11),PDY(11),PAY(11)  DIMENSION DY2(11),VY2(11),AY2(11),AAY(11)  DIMENSION SKX(11,11),SMX(11,11),CDX(11,11)  DIMENSION SKY(11,11),SMY(11,11),CDY(11,11)  DIMENSION DDX(11),DVX(11),PX1(11),PX2(11),PINCX(11)  DIMENSION DDY(11),DVY(11),PY1(11),PY2(11),PINCY(11)  DIMENSION EFKX(11,11),EFPX(11),EFKY(11,11),EFPY(11)  DIMENSION FAB(2),DFAB(2)  DIMENSION V(11,11),VI(11,11),W(11)  DIMENSION AM(11),AK(11),ZETA(11)  DIMENSION RES(2),PRES(2)  COMMON/RS1/XG(1501),YG(1501)  C  OPEN(1,FILE='Sliding.DAT')  OPEN(2,FILE='SYLM360.TXT')  OPEN(3,FILE='Sliding.3')  OPEN(4,FILE='Sliding.4')  OPEN(5,FILE='Sliding.513')  OPEN(6,FILE='SYLM90.TXT')  READ(1,\*) NDT,DTN,NDIV,NIT  READ(1,\*) NST  READ(1,\*) (AM(I),I=1,NST)  READ(1,\*) (AK(I),I=1,NST)  READ(1,\*) (ZETA(I),I=1,NST)  READ(1,\*) KNOR,N  READ(1,\*)KPF,LXY  READ(2,\*)(XG(I),I=1,NDT)  READ(6,\*)(YG(I),I=1,NDT)  C LXY=1,2,<SINGLE X-COMPONENT>,<Y-COMPONENT>  C KPF=0,1,<PURE-FRICTION>,<PURE-FRICTION WITH RESTORING FORCE>  C PUT TB>10.0 FOR P-F SYSTEM ONLY (KPF IS DISABLED)  DO 101 I=1,NDT  IF(LXY.EQ.1) YG(I)=0.0  IF(LXY.EQ.2) XG(I)=0.0  101 CONTINUE  IF(LXY.GT.0) NIT=1  PI=ATAN(1.0)\*4.0  DO 1050 KN=1,N  READ(1,\*) IJK,TX1,RTYTX,RMBM,TB,ZETAB,AMU  BM=RMBM\*AM(1)  WX1=2.0\*PI/TX1  WY1=WX1/RTYTX  NDOF=NST+1  TM=BM  DO 106 I=1,NST  TM=TM+AM(I)  106 CONTINUE  C KPF=1 NASEEF: IT IS ALREADY READ FROM THE INPUT FILE #1  WB=2.0\*PI/TB  C IF(TB.GE.10.0) KPF=0 NASEEF: IF TB IS MORE THAN 10 SEC PROGRAM BY DEFAULT CONSIDERS IT PURE-FRICTION SYSTEM WITHOUT LINEAR SPRING  IF(KPF.EQ.0) WB=0.0  CKABX=TM\*WB\*WB  CKABY=TM\*WB\*WB  CDABX=2.0\*TM\*ZETAB\*WB  CDABY=2.0\*TM\*ZETAB\*WB  QX=AMU\*981.0\*TM  QY=QX  DT=DTN/NDIV  WRITE(3,1321)TX1  WRITE(3,1322)RTYTX  WRITE(3,1336)RMBM  WRITE(3,1301)TB  WRITE(3,1337)BM  WRITE(3,1304)ZETAB  WRITE(3,1342)CKABX  WRITE(3,1343)CKABY  WRITE(3,1345)AMU  WRITE(3,1344)QX  WRITE(3,1344)QY  WRITE(3,1339)NDIV  C COMPUTATION OF SUPERSTRUCTURE MATRICES IN X-DIRECTION  CALL SMS(SMX,AM,NST)  CALL SKS(SKX,VI,AK,NST)  CALL INVD(VI,SMX,V,NST)  CALL EIGEN(VI,SMX,V,W,NST,99)  IF(KNOR.EQ.0) GO TO 102  CALL NORM(W(1),WX1,AK,NST)  CALL SKS(SKX,VI,AK,NST)  CALL INVD(VI,SMX,V,NST)  CALL EIGEN(VI,SMX,V,W,NST,99)  102 CALL INV(V,VI,NST)  CALL SCS(CDX,VI,W,ZETA,NST)  WRITE(3,1351)  WRITE(3,1325)  WRITE(3,1302)((SMX(I,J),J=1,NST),I=1,NST)  WRITE(3,1300)  WRITE(3,1302) (W(I),I=1,NST)  WRITE(3,1329)  WRITE(3,1302)((V(I,J),J=1,NST),I=1,NST)  WRITE(3,1326)  WRITE(3,1302)((CDX(I,J),J=1,NST),I=1,NST)  WRITE(3,1310)  WRITE(3,1302)((SKX(I,J),J=1,NST),I=1,NST)  C COMPUTATION OF SUPERSTRUCTURE MATRICES IN Y-DIRECTION  CALL SMS(SMY,AM,NST)  CALL SKS(SKY,VI,AK,NST)  CALL INVD(VI,SMY,V,NST)  CALL EIGEN(VI,SMY,V,W,NST,99)  IF(KNOR.EQ.0) GO TO 103  CALL NORM(W(1),WY1,AK,NST)  CALL SKS(SKY,VI,AK,NST)  CALL INVD(VI,SMY,V,NST)  CALL EIGEN(VI,SMY,V,W,NST,99)  103 CALL INV(V,VI,NST)  CALL SCS(CDY,VI,W,ZETA,NST)  WRITE(3,1352)  WRITE(3,1325)  WRITE(3,1302)((SMY(I,J),J=1,NST),I=1,NST)  WRITE(3,1300)  WRITE(3,1302) (W(I),I=1,NST)  WRITE(3,1329)  WRITE(3,1302)((V(I,J),J=1,NST),I=1,NST)  WRITE(3,1326)  WRITE(3,1302)((CDY(I,J),J=1,NST),I=1,NST)  WRITE(3,1310)  WRITE(3,1302)((SKY(I,J),J=1,NST),I=1,NST)  C  CALL SMO(SMX,BM,NDOF)  CALL SCO(CDX,CDABX,CX1,NDOF)  CALL SKO(SKX,CKABX,AKX1,NDOF)  CALL SMO(SMY,BM,NDOF)  CALL SCO(CDY,CDABY,CY1,NDOF)  CALL SKO(SKY,CKABY,AKY1,NDOF)  WRITE(3,1328)  WRITE(3,1312)((SMX(I,J),J=1,NDOF),I=1,NDOF)  WRITE(3,1327)  WRITE(3,1312)((CDX(I,J),J=1,NDOF),I=1,NDOF)  WRITE(3,1311)  WRITE(3,1312)((SKX(I,J),J=1,NDOF),I=1,NDOF)  WRITE(3,1328)  WRITE(3,1312)((SMY(I,J),J=1,NDOF),I=1,NDOF)  WRITE(3,1327)  WRITE(3,1312)((CDY(I,J),J=1,NDOF),I=1,NDOF)  WRITE(3,1311)  WRITE(3,1312)((SKY(I,J),J=1,NDOF),I=1,NDOF)  C INITIALIZATION OF VECTORS.  ID=1  FAB(1)=0.0  FAB(2)=0.0  PRES(1)=0.0  PRES(2)=0.0  DO 110 I=1,NDOF  DX1(I)=0.0  VX1(I)=0.00001  C NASEEF: WHY 0.00001 FOR VX1 AND VY1???  AX1(I)=0.0  DX2(I)=0.0  VX2(I)=0.0  AX2(I)=0.0  AAX(I)=0.0  PDX(I)=0.0  PAX(I)=0.0  DY1(I)=0.0  VY1(I)=0.00001  AY1(I)=0.0  DY2(I)=0.0  VY2(I)=0.0  AY2(I)=0.0  AAY(I)=0.0  PDY(I)=0.0  PAY(I)=0.0  DDX(I)=0.0  DVX(I)=0.0  PX1(I)=0.0  PX2(I)=0.0  PINCX(I)=0.0  DDY(I)=0.0  DVY(I)=0.0  PY1(I)=0.0  PY2(I)=0.0  PINCY(I)=0.0  110 CONTINUE  TIME = 0.0  DO 1020 K=1,NDT-1  DO 112 I=1,NDOF  PX1(I)=-SMX(I,I)\*XG(K)  PX2(I)=-SMX(I,I)\*XG(K+1)  PY1(I)=-SMY(I,I)\*YG(K)  PY2(I)=-SMY(I,I)\*YG(K+1)  112 CONTINUE  DO 116 I=1,NDOF  PINCX(I)=(PX2(I)-PX1(I))/NDIV  PINCY(I)=(PY2(I)-PY1(I))/NDIV  116 CONTINUE  DO 1010 KS=1,NDIV  TIME = TIME + DT  DO 117 I=1,NDOF  PX2(I)=PX1(I)+PINCX(I)  PY2(I)=PY1(I)+PINCY(I)  117 CONTINUE  EQX=-PX2(1)/SMX(1,1)  EQY=-PY2(1)/SMY(1,1)  IF(K+KS-2)1020,118,120  118 CALL ACCNS(DX2,VX2,AX2,SKX,SMX,CDX,PX1,NST)  CALL ACCNS(DY2,VY2,AY2,SKY,SMY,CDY,PY1,NST)  DO 119 I=1,NST  AX1(I)=AX2(I)  AX2(I)=0.0  AY1(I)=AY2(I)  AY2(I)=0.0  119 CONTINUE  120 IF(ID)141,141,121  121 CALL EFFP(VX1,AX1,SMX,CDX,PINCX,EFPX,DT,NST)  CALL EFFK(SKX,SMX,CDX,EFKX,DT,NST)  CALL GAUS(NST,EFKX,EFPX,DDX)  CALL EFFP(VY1,AY1,SMY,CDY,PINCY,EFPY,DT,NST)  CALL EFFK(SKY,SMY,CDY,EFKY,DT,NST)  CALL GAUS(NST,EFKY,EFPY,DDY)  DO 122 I=1,NST  DX2(I)=DX1(I)+DDX(I)  DVX(I)=3.0\*DDX(I)/DT-3.0\*VX1(I)-DT\*AX1(I)\*0.5  VX2(I)=VX1(I)+DVX(I)  DY2(I)=DY1(I)+DDY(I)  DVY(I)=3.0\*DDY(I)/DT-3.0\*VY1(I)-DT\*AY1(I)\*0.5  VY2(I)=VY1(I)+DVY(I)  122 CONTINUE  FAB(1)=CX1\*VX2(NST) + AKX1\*DX2(NST)  FAB(1)=FAB(1) - CKABX\*DX2(NDOF) - BM\*EQX  FAB(2)=CY1\*VY2(NST) + AKY1\*DY2(NST)  FAB(2)=FAB(2) - CKABY\*DY2(NDOF) - BM\*EQY  CALL STAT1(FAB,QX,QY,ID)  IF(ID.EQ.0) GO TO 123  CALL ACCNS(DX2,VX2,AX2,SKX,SMX,CDX,PX2,NST)  CALL ACCNS(DY2,VY2,AY2,SKY,SMY,CDY,PY2,NST)  GO TO 124  123 CALL ACCNO(DX2,VX2,AX2,SKX,SMX,CDX,PX2,NDOF,FAB(1))  CALL ACCNO(DY2,VY2,AY2,SKY,SMY,CDY,PY2,NDOF,FAB(2))  124 CALL PEAK(PX1,PX2,DX1,VX1,AX1,DX2,VX2,AX2,AAX,PDX,PAX,EQX,NDOF)  CALL PEAK(PY1,PY2,DY1,VY1,AY1,DY2,VY2,AY2,AAY,PDY,PAY,EQY,NDOF)  GO TO 1010  141 DFAB(1)=0.0  DFAB(2)=0.0  DO 1005 KIT=1,NIT  CALL EFFP(VX1,AX1,SMX,CDX,PINCX,EFPX,DT,NDOF)  CALL EFFK(SKX,SMX,CDX,EFKX,DT,NDOF)  CALL EFFP(VY1,AY1,SMY,CDY,PINCY,EFPY,DT,NDOF)  CALL EFFK(SKY,SMY,CDY,EFKY,DT,NDOF)  EFPX(NDOF)=EFPX(NDOF) - DFAB(1)  EFPY(NDOF)=EFPY(NDOF) - DFAB(2)  CALL GAUS(NDOF,EFKX,EFPX,DDX)  CALL GAUS(NDOF,EFKY,EFPY,DDY)  DO 142 I=1,NDOF  DX2(I)=DX1(I)+DDX(I)  DVX(I)=3.0\*DDX(I)/DT-3.0\*VX1(I)-DT\*AX1(I)\*0.5  VX2(I)=VX1(I)+DVX(I)  DY2(I)=DY1(I)+DDY(I)  DVY(I)=3.0\*DDY(I)/DT-3.0\*VY1(I)-DT\*AY1(I)\*0.5  VY2(I)=VY1(I)+DVY(I)  142 CONTINUE  RVL=SQRT(VX2(NDOF)\*VX2(NDOF) + VY2(NDOF)\*VY2(NDOF))  IF(RVL.LE.1.0E-4) GO TO 1005  DFAB(1)=QX\*VX2(NDOF)/RVL - FAB(1)  DFAB(2)=QY\*VY2(NDOF)/RVL - FAB(2)  1005 CONTINUE  IF(NIT.EQ.1) GO TO 143  FAB(1)=FAB(1)+DFAB(1)  FAB(2)=FAB(2)+DFAB(2)  143 CALL STAT0(DDX(NDOF),DDY(NDOF),FAB,ID)  IF(ID)145,145,144  144 VX2(NDOF)=0.0  AX2(NDOF)=0.0  VY2(NDOF)=0.0  AY2(NDOF)=0.0  CALL ACCNS(DX2,VX2,AX2,SKX,SMX,CDX,PX2,NST)  CALL ACCNS(DY2,VY2,AY2,SKY,SMY,CDY,PY2,NST)  GO TO 146  145 CALL ACCNO(DX2,VX2,AX2,SKX,SMX,CDX,PX2,NDOF,FAB(1))  CALL ACCNO(DY2,VY2,AY2,SKY,SMY,CDY,PY2,NDOF,FAB(2))  146 CALL PEAK(PX1,PX2,DX1,VX1,AX1,DX2,VX2,AX2,AAX,PDX,PAX,EQX,NDOF)  CALL PEAK(PY1,PY2,DY1,VY1,AY1,DY2,VY2,AY2,AAY,PDY,PAY,EQY,NDOF)  1010 CONTINUE  C WRITE(\*,1315)  C WRITE(\*,1316)K,TIME,ID,(FAB(I),I=1,2)  WRITE(4,1346)TIME,AAX(1),AAY(1),DX2(NDOF),DY2(NDOF),  + FAB(1),FAB(2)  RES(1)=SQRT(AAX(1)\*AAX(1) + AAY(1)\*AAY(1))  RES(2)=SQRT(DX2(NDOF)\*DX2(NDOF) + DY2(NDOF)\*DY2(NDOF))  IF(PRES(1).LE.RES(1)) PRES(1)=RES(1)  IF(PRES(2).LE.RES(2)) PRES(2)=RES(2)  1020 CONTINUE  WRITE(5,1356) IJK,PRES(1),PRES(2)  C WRITE(5,1356) IJK,PAX(1),PAY(1),PDX(NDOF),PDY(NDOF)  WRITE(3,1350)  DO 152 I=1,NDOF  WRITE(3,1349) PDX(I),PAX(I),PDY(I),PAY(I)  152 CONTINUE  1050 CONTINUE  C  1300 FORMAT(/,10X,'FIXED BASE NATURAL FREQUENCIES ')  1301 FORMAT(3X,'PERIOD OF ISOLATED STRUCTURE = ',E13.5)  1302 FORMAT(5X,5E13.6)  1303 FORMAT(/10X,'STIFFNESS MARTRIX')  1304 FORMAT(3X,'DAMPING RATIO OF SLIDING SYSTEM = ',E13.5)  1305 FORMAT(/10X,'NATURAL FREQUENCIES')  1306 FORMAT(/10X,'MODAL COLUMN MARTRIX')  1310 FORMAT(/,10X,'FIXED BASE STIFFNESS MATRIX')  1311 FORMAT(/,10X,'ASSEMBLED STIFFNESS MATRIX')  1312 FORMAT(5X,6E13.5)  1313 FORMAT(3X,4E13.5)  1315 FORMAT(6X,'NDT TIME FABX ',8X,'FABY')  1316 FORMAT(5X,I10,F8.3,I10,5X,2E15.6)  1319 FORMAT(3X,I2,2F9.2,2X,2F10.2)  1321 FORMAT(3X,'FUNDAMENTAL S.S. PERIOD (X) = ',E13.5)  1322 FORMAT(3X,'Y TO X FUNDAMNETAL PERIOD RATIO = ',F13.5)  1325 FORMAT(/,10X,'FIXED BASE MASS MATRIX')  1326 FORMAT(/,10X,'FIXED BASE DAMPING MATRIX')  1327 FORMAT(/,10X,'ASSEMBLED DAMPING MATRIX')  1328 FORMAT(/,10X,'ASSEMBLED MASS MATRIX')  1329 FORMAT(/,10X,'FIXED BASE MODE-SHAPE MATRIX')  1336 FORMAT(3X,'RATION OF BASE MASS TO FLOOR = ',E13.5)  1337 FORMAT(3X,'MASS OF BASE RAFT = ',E13.5)  1339 FORMAT(3X,'NDIV VALUE = ',I13)  1342 FORMAT(3X,'SLIDING SYSTEM STIFFNESS (X) = ',E13.5)  1343 FORMAT(3X,'SLIDING SYSTEM STIFFNESS (Y) = ',E13.5)  1344 FORMAT(3X,'LIMITING FRICTIONAL FORCE (X/Y) = ',E13.5)  1345 FORMAT(3X,'COEFFICIENT OF FRICTION = ',E13.5)  1346 FORMAT(F6.3,1X,7E13.5)  1349 FORMAT(3X,4E13.5)  1350 FORMAT(/,5X,'PEAK DISP PEAK ACCN ( X/Y)')  1351 FORMAT(/,3X,'SUPERSTRUCTURE PROPERTIES = ',15(' X'))  1352 FORMAT(/,3X,'SUPERSTRUCTURE PROPERTIES = ',15(' Y'))  1353 FORMAT(I3,6X,6F6.2,F6.2,F6.2)  1354 FORMAT(6F10.4)  1356 FORMAT(I10,6E13.5)  1362 FORMAT(3X,6F7.2,4F8.2,2F5.2)  1367 FORMAT(I4,F10.2,6E14.6)  STOP  END    SUBROUTINE SMS(SM,AM,NST)  DIMENSION SM(11,11),AM(11)  DO 1 I=1,NST+1  DO 1 J=1,NST+1  SM(I,J)=0.0  1 CONTINUE  DO 2 I=1,NST  SM(I,I)=SM(I,I)+AM(I)  2 CONTINUE  RETURN  END    SUBROUTINE SKS(SK,VI,AK,NST)  DIMENSION SK(11,11),VI(11,11),AK(11)  DO 1 I=1,NST+1  DO 1 J=1,NST+1  SK(I,J)=0.0  1 CONTINUE  SK(1,1)=SK(1,1) + AK(1)  SK(1,2)=SK(1,2) - AK(1)  SK(2,1)=SK(2,1) - AK(1)  DO 2 I=2,NST  II=I-1  SK(I,I)=AK(I)+AK(II)  IF(I.EQ.NST) GO TO 2  SK(I,I+1)=SK(I,I+1)-AK(I)  SK(I+1,I)=SK(I+1,I)-AK(I)  2 CONTINUE  DO 3 I=1,NST  DO 3 J=1,NST  VI(I,J)=SK(I,J)  3 CONTINUE  RETURN  END    SUBROUTINE SCS(CD,VI,W,ZETA,NST)  DIMENSION CD(11,11),VI(11,11),W(11),ZETA(11),XX(11,11)  DO 1 I=1,NST+1  DO 1 J=1,NST+1  CD(I,J)=0.0  1 CONTINUE  DO 2 I=1,NST  CD(I,I)=2.0\*ZETA(I)\*W(I)  2 CONTINUE  DO 3 I=1,NST  DO 3 J=1,NST  XX(I,J)=0.0  DO 3 K=1,NST  XX(I,J)=XX(I,J) + CD(I,K)\*VI(K,J)  3 CONTINUE  DO 4 I=1,NST  DO 4 J=1,NST  CD(I,J)=0.0  DO 4 K=1,NST  CD(I,J)=CD(I,J) + VI(K,I)\*XX(K,J)  4 CONTINUE  RETURN  END    SUBROUTINE SMO(SM,BM,NDOF)  DIMENSION SM(11,11)  NST=NDOF-1  DO 1 I=1,NST  SM(I,NST+1)=SM(I,I)  1 CONTINUE  SM(NDOF,NDOF)=BM  RETURN  END    SUBROUTINE SCO(CD,CDAB,C1,NDOF)  DIMENSION CD(11,11)  NST=NDOF-1  C1=CD(NST,NST)+CD(NST,NST-1)  CD(NDOF,NDOF)=CDAB  CD(NDOF,NST)=-C1  RETURN  END    SUBROUTINE SKO(SK,CKAB,AK1,NDOF)  DIMENSION SK(11,11)  NST=NDOF-1  AK1=SK(NST,NST)+SK(NST,NST-1)  SK(NDOF,NDOF)=CKAB  SK(NST,NDOF)=0.0  SK(NDOF,NST)=-AK1  RETURN  END    SUBROUTINE ACCNS(D,V,A,SK,SM,CD,P,NST)  DIMENSION D(11),V(11),A(11)  DIMENSION SK(11,11),SM(11,11),CD(11,11)  DIMENSION FK(11),FC(11),EP1(11),P(11)  DO 3 I=1,NST  FC(I)=0.0  FK(I)=0.0  DO 2 J=1,NST  FC(I)=FC(I)+CD(I,J)\*V(J)  FK(I)=FK(I)+SK(I,J)\*D(J)  2 CONTINUE  EP1(I)=P(I)-FC(I)-FK(I)  3 CONTINUE  DO 4 I=1,NST  A(I)=EP1(I)/SM(I,I)  4 CONTINUE  RETURN  END  SUBROUTINE EFFP(V1,A1,SM,CD,PINC,EFP,T,NDOF)  DIMENSION EFP(11),PINC(11),V1(11),A1(11)  DIMENSION SM(11,11),CD(11,11)  DO 1 I=1,NDOF  1 EFP(I) =0.0  DO 2 I=1,NDOF  EFP(I)=EFP(I)+PINC(I)  DO 3 J=1,NDOF  X1=SM(I,J)\*(6.0\*V1(J)/T + 3.0\*A1(J))  X2=CD(I,J)\*(3.0\*V1(J) + 0.5\*T\*A1(J))  EFP(I)=EFP(I) + X1 + X2  3 CONTINUE  2 CONTINUE  C WRITE(8,11)  C WRITE(8,12)(EFP(I),I=1,3)  C 11 FORMAT(10X,'EFFECTIVE FORCE')  RETURN  END    SUBROUTINE EFFK(SK,SM,CD,EFK,T,NDOF)  DIMENSION EFK(11,11)  DIMENSION SK(11,11),SM(11,11),CD(11,11)  DO 1 I=1,NDOF  DO 1 J=1,NDOF  1 EFK(I,J)=SK(I,J)  DO 2 I=1,NDOF  DO 2 J=1,NDOF  EFK(I,J)=EFK(I,J) + CD(I,J)\*3.0/T + SM(I,J)\*6.0/(T\*T)  2 CONTINUE  C WRITE(8,11)  C WRITE(8,12)((EFK(I,J),J=1,NDOF),I=1,NDOF)  C 11 FORMAT(10X,'EFFECTIVE STIFFNESS MATRIX')  C 12 FORMAT(5X,3E13.6)  RETURN  END    SUBROUTINE GAUS(N,A,F,X)  DIMENSION A(11,11),F(11),X(11)  C CHOOSE THE K TH UNKNOWN TO BE  C ELIMINATED FROM ALL THE SUCCEEDING EQUATION  DO 10 K=1,(N-1)  C ELIMINATE THE KTH UNKNOWN FROM ALL  C (K+1) TO N EQUATIONS  DO 20 I=(K+1),N  QUOT=A(I,K)/A(K,K)  C MODIFY THE LOAD VECTOR BY  F(I)=F(I)-QUOT\*F(K)  C MODIFY THE I TH EQUATION ELEMENTS  DO 30 J=(K+1),N  A(I,J)=A(I,J)-QUOT\*A(K,J)  30 CONTINUE  20 CONTINUE  10 CONTINUE  X(N)=F(N)/A(N,N)  DO 40 I=(N-1),1,-1  SUM=0.0  DO 50 J=(I+1),N  SUM=SUM+A(I,J)\*X(J)  50 CONTINUE  X(I)=(F(I)-SUM)/A(I,I)  40 CONTINUE  C WRITE(\*,11)  C WRITE(\*,12)(X(I),I=1,NDOF)  C 11 FORMAT(10X,'DELTA DISPLACEMENT')  C 12 FORMAT(3X,2E13.6)  RETURN  END  SUBROUTINE ACCNO(D,V,A,SK,SM,CD,P,NDOF,F)  DIMENSION D(11),V(11),A(11)  DIMENSION SK(11,11),SM(11,11),CD(11,11)  DIMENSION FK(11),FC(11),EP1(11),P(11)  DO 3 I=1,NDOF  FC(I)=0.0  FK(I)=0.0  DO 2 J=1,NDOF  FC(I)=FC(I)+CD(I,J)\*V(J)  FK(I)=FK(I)+SK(I,J)\*D(J)  2 CONTINUE  EP1(I)=P(I)-FC(I)-FK(I)  3 CONTINUE  EP1(NDOF)=EP1(NDOF)-F  A(NDOF)=EP1(NDOF)/SM(NDOF,NDOF)  DO 4 I=1,NDOF-1  A(I)=(EP1(I) - SM(I,I)\*A(NDOF))/SM(I,I)  4 CONTINUE  RETURN  END  SUBROUTINE PEAK(P1,P2,D1,V1,A1,D2,V2,A2,AA,PD,PA,EQ,N)  DIMENSION D1(11),A1(11),PD(11),V1(11)  DIMENSION D2(11),A2(11),PA(11),V2(11)  DIMENSION P1(11),P2(11),AA(11)  AA(N)=A2(N)+EQ  DO 1 I=1,N-1  AA(I)=A2(I)+A2(N)+EQ  1 CONTINUE  DO 10 I=1,N  IF(ABS(PD(I))-ABS(D1(I)))9,10,10  9 PD(I)=D1(I)  10 CONTINUE  DO 14 I=1,N  IF(ABS(PA(I))-ABS(AA(I)))13,14,14  13 PA(I)=AA(I)  14 CONTINUE  DO 17 I=1,N  A1(I)=A2(I)  V1(I)=V2(I)  D1(I)=D2(I)  P1(I)=P2(I)  17 CONTINUE  RETURN  END  SUBROUTINE EIGEN(XK,XM,V,W,N,IT)  C STODOLA METHOD FOR MODE SHAPE AND FREQUENCY.  DIMENSION XK(11,11),S(11,11),X(100,11),X1(11),X2(11)  DIMENSION XX(11,11),XM(11,11),V(11,11),W(11)  DO 1 I=1,N  DO 1 J=1,N  S(I,J)=0.0  S(I,I)=1.0  1 CONTINUE  DO 16 IE=1,N  IF(IE-1) 16,6,2  2 DO 4 I=1,N  DO 4 J=1,N  XX(I,J)= 0.0  DO 3 K=1,N  3 XX(I,J)=XX(I,J)+X1(I)\*X1(K)\*XM(K,J)  4 S(I,J)=S(I,J)-XX(I,J)/UM  DO 5 I=1,N  DO 5 J=1,N  XK(I,J)=0.0  DO 5 K=1,N  5 XK(I,J)=XK(I,J)+X(I,K)\*S(K,J)  6 DO 7 I=1,N  7 X(1,I)=1.0  DO 11 K=1,IT  DO 9 I=1,N  X2(I)=0.0  DO 8 J=1,N  8 X2(I)=X2(I)+XK(I,J)\*X(K,J)  9 X(K+1,I)=X2(I)/X2(1)  11 CONTINUE  W(IE)=SQRT(1.0/X2(1))  DO 12 I=1,N  X1(I)=X(IT,I)  DO 12 J=1,N  12 X(I,J)=XK(I,J)  UM=0.0  DO 14 I=1,N  X2(I) = 0.0  DO 13 J = 1,N  13 X2(I)=X2(I)+XM(I,J)\*X1(J)  14 UM=UM+X1(I)\*X2(I)  DO 15 I=1,N  15 V(I,IE)=X1(I)/SQRT(UM)  16 CONTINUE  RETURN  END    SUBROUTINE INVD(XK,XM,V,N)  C GENERATION OF FLEXIBILITY & DYNAMIC MATRIX.  DIMENSION XK(11,11),V(11,11),XM(11,11)  DO 1 I=1,N  DO 1 J=1,N  V(I,J)=0.0  V(I,I)=1.0  1 CONTINUE  DO 6 I=1,N  TEMP=XK(I,I)  DO 2 J=1,N  XK(I,J)=XK(I,J)/TEMP  V(I,J)=V(I,J)/TEMP  2 CONTINUE  DO 5 J=1,N  IF(I-J) 3,5,3  3 RT=XK(J,I)  DO 4 K=1,N  XK(J,K)=XK(J,K)-XK(I,K)\*RT  4 V(J,K)=V(J,K)-V(I,K)\*RT  5 CONTINUE  6 CONTINUE  DO 7 I=1,N  DO 7 J=1,N  XK(I,J)=0.0  DO 7 K=1,N  7 XK(I,J)=XK(I,J)+V(I,K)\*XM(K,J)  RETURN  END      SUBROUTINE INV(XX,B,N)  C TO INVERT A REAL MATRIX.  DIMENSION A(11,11),B(11,11),XX(11,11)  DO 1 I=1,N  DO 1 J=1,N  A(I,J)=XX(I,J)  B(I,J)=0.0  B(I,I)=1.0  1 CONTINUE  DO 6 I=1,N  TEMP=A(I,I)  DO 2 J=1,N  A(I,J)=A(I,J)/TEMP  B(I,J)=B(I,J)/TEMP  2 CONTINUE  DO 5 J=1,N  IF(I-J) 3,5,3  3 RT=A(J,I)  DO 4 K=1,N  A(J,K)=A(J,K)-A(I,K)\*RT  4 B(J,K)=B(J,K)-B(I,K)\*RT  5 CONTINUE  6 CONTINUE  RETURN  END  SUBROUTINE NORM(W1,WS1,AK,NST)  DIMENSION AK(11)  RT=WS1/W1  DO 11 I=1,NST  AK(I)=AK(I)\*RT\*RT  11 CONTINUE  RETURN  END  SUBROUTINE STAT1(FAB,QX,QY,ID)  DIMENSION FAB(2)  Q1=FAB(1)  Q2=FAB(2)  RATIO=SQRT((Q1/QX)\*\*2 + (Q2/QY)\*\*2)  IF(RATIO-1.0)11,12,12  11 FAB(1) = Q1  FAB(2) = Q2  ID=1  GO TO 99  12 FAB(1) = Q1/RATIO  FAB(2) = Q2/RATIO  ID=0  99 RETURN  END    SUBROUTINE STAT0(DY3,DY4,FAB,ID)  DIMENSION FAB(2)  Q1 = FAB(1)  Q2 = FAB(2)  WD = Q1\*DY3 + Q2\*DY4  IF(WD)11,99,99  11 ID=1  99 RETURN  END | # Defitions  Most of them are default  Cleary define what is “FAB”, “RES” and “PRES”  # What is COMMON?  File #3: Stores properties of structure  File #4: Stores responses.  File #5: ????  NIT: Maximum number of iteration for fixed point iteration scheme used.  “AMU” is the coefficient of friction |