

AD-A127 204

FLEET MOORING LEG DESIGN PROGRAM DOCUMENTATION VOLUME 5 - 1/1

SOURCE LISTINGS: COMPOUND LEG BASIC SOLUTION(U)

PRESEARCH INC ARLINGTON VA DEC 82 FPO-1-82-(38)

UNCLASSIFIED

N82477-81-C-0025

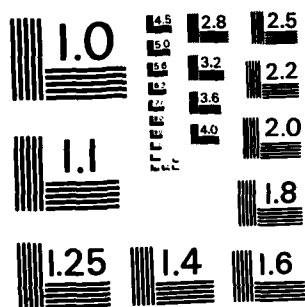
F/G 9/2

NL



1.2

END  
NOV  
1983  
S B 3  
DTIC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

①

# PRESEARCH

AD A127204

DTIC FILE COPY

FLEET MOORING LEG  
DESIGN PROGRAM DOCUMENTATION

Volume 5

SOURCE LISTINGS:  
COMPOUND LEG BASIC SOLUTION

FPO-1-82-(36)

December 1982

*General Distribution*

DTIC  
ELECTE  
APR 25 1983  
S E

**PRESEARCH INCORPORATED**

2361 S. JEFFERSON DAVIS HIGHWAY, ARLINGTON, VA. 22202 (703) 553-2700

This document has been approved  
for public release and sale; its  
distribution is unlimited.

83 04 07 019

PRESEARCH INCORPORATED

①

FLEET MOORING LEG  
DESIGN PROGRAM DOCUMENTATION

Volume 5

SOURCE LISTINGS:  
COMPOUND LEG BASIC SOLUTION

FPO-1-82-(36)

December 1982

*General Distribution*

Performed for  
Ocean Engineering and Construction Project Office  
Chesapeake Division  
Naval Facilities Engineering Command  
Washington, D.C. 20374  
Under  
Contract N62477-81-C-0025

This document has been approved  
for public release and sale; its  
distribution is unlimited.

Presearch Incorporated  
2361 South Jefferson Davis Highway  
Arlington, Virginia 22202

DTIC  
ELECTE  
S D  
APR 25 1983  
E

PRESEARCH INCORPORATED

FLEET MOORING LEG  
DESIGN PROGRAM DOCUMENTATION

Volume 5

SOURCE LISTINGS:  
COMPOUND LEG BASIC SOLUTION

|       | <u>Section</u>                                      | <u>Volume</u> |
|-------|-----------------------------------------------------|---------------|
| I.    | EQUILIBRIUM EQUATIONS REPORT                        | 1             |
| II.   | USER DOCUMENTATION                                  | 2             |
| III.  | SUBROUTINE DESCRIPTIONS                             | 3             |
| IV.   | SOURCE LISTINGS                                     |               |
|       | Query, Preprocessor, and Simple Leg                 | 4             |
|       | Compound Leg Basic Solution                         | 5             |
|       | Compound Leg Reverse Solutions and<br>Postprocessor | 6             |
|       | Table and Graphs                                    | 7             |
| V.    | LINK COMMAND FILES                                  | 8             |
| VI.   | PROGRAM FILE DESCRIPTIONS                           | 8             |
| VII.  | COMMON BLOCKS                                       | 8             |
| VIII. | DIRECTORY LISTINGS                                  | 8             |
| IX.   | IPL TAPE LISTINGS                                   | 8             |

|                     |                                     |
|---------------------|-------------------------------------|
| Accession For       |                                     |
| NTIS GRA&I          | <input checked="" type="checkbox"/> |
| DTIC TAB            | <input type="checkbox"/>            |
| Unannounced         | <input type="checkbox"/>            |
| Justification       | <i>in file</i>                      |
| By _____            |                                     |
| Distribution/ _____ |                                     |
| Availability Codes  |                                     |
| Dist                | Avail and/or<br>Special             |
| <b>A</b>            |                                     |



# IV. SOURCE LISTINGS

COMPOUND LEG BASIC SOLUTION

CSLACK - EDGPT, PAGES 90-164

|        |    |        |     |        |     |         |     |
|--------|----|--------|-----|--------|-----|---------|-----|
| MOORØ1 | 1  | SECV   | 65  | JUNCT  | 127 | CTEN2   | 217 |
| MOORØ2 | 3  | VCAITØ | 67  | EBUOY  | 129 | CTEN3   | 218 |
| BKDAT  | 5  | ESTV   | 68  | SCOIL  | 132 | GRAPH1  | 222 |
| QUERY  | 6  | LENS   | 69  | XSECV  | 135 | GRIN1   | 224 |
| GEINIT | 17 | VEUN   | 70  | SHIFT  | 142 | GRIN2   | 228 |
| RW     | 18 | CALC2  | 71  | SUBVX  | 143 | GROUT1  | 233 |
| ADDEXT | 19 | EPSLV  | 76  | WGTH   | 145 | GROUT2  | 237 |
| ECHO   | 20 | RDBACK | 80  | CERISR | 146 | GRAPHS  | 242 |
| OUTVAR | 25 | ELV1   | 83  | SECVIT | 148 | STICK   | 244 |
| CONVRT | 27 | GCOEFF | 85  | STEFAB | 150 | SINKER  | 247 |
| RWCOM1 | 32 | CTEN1  | 87  | CALC3  | 159 | ANCHOR  | 249 |
| HXØRY  | 34 | CSLACK | 90  | EDGPT  | 162 | ELIZER  | 251 |
| SOLVE  | 36 | CPREPØ | 94  | CSEHP  | 165 | BUOY    | 253 |
| PRSLV  | 37 | CPREP1 | 97  | FTEN   | 170 | ELVPNT  | 255 |
| SECNT  | 44 | SUMSC  | 99  | CSSPR  | 172 | SYMSNK  | 265 |
| NWGT   | 45 | CPREP2 | 100 | CSEPR  | 177 | WELVPT  | 266 |
| UMAP   | 46 | CPREP3 | 102 | CSSXZ1 | 181 | ELVCAT  | 267 |
| CUMAP  | 47 | CSSHØP | 104 | CSSXZ2 | 184 | PLNPNT  | 270 |
| ISORT  | 48 | PHIAB  | 110 | CSEXZ1 | 188 | PLNSNK  | 279 |
| TAUT   | 49 | HSPLIT | 111 | CSEXZ2 | 191 | WPLNPT  | 280 |
| STEF2A | 51 | TRISR  | 113 | CEPSLV | 195 | PLNCAT  | 281 |
| CHS2A  | 54 | SCA7AB | 115 | CRDBAK | 207 | MOORØ4  | 283 |
| SEC1A  | 55 | SRISR  | 117 | COMMAP | 210 | HXCALC  | 285 |
| CHS1A  | 57 | TAN1   | 119 | CONTP  | 211 | HXCCLC1 | 287 |
| CALC1  | 58 | X4CALC | 120 | ENØD   | 212 | HXCCLC2 | 291 |
| SLACK  | 59 | TSPLIT | 122 | LEN.1  | 213 | CSEXHP  | 295 |
| STEF2V | 61 | JTEN   | 125 | ELV2   | 214 | MOORØ5  | 296 |

```

et sys final/12for/cslack for!!
  subroutine CSLACK
  ****
  implicit integer*2 (*)

  integer*2 ileg,ist,ncs,ncb,nwa,nwb,isol,ibrnch,uz(5)
  double precision z(67),cz,cx,d,ta,tb
  common /VGL08/ ileg,ist,ncs,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
  & isol,ibrnch,uz
  double precision za(25),zb(25)
  equivalence (z(1),za(1)),(z(26),zb(1))
  double precision ha,a1a,va,s1a,w1a,c1a,s2a,w2a,c2a,s3a,w3a,
  & xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
  & tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
  equivalence (za(1),ha),(za(2),a1a,va),
  & (za(3),s1a),(za(4),w1a),(za(5),c1a),
  & (za(6),s2a),(za(7),w2a),(za(8),c2a),
  & (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
  & (za(13),x1a),(za(14),x2a),(za(15),x3a),
  & (za(16),y1a),(za(17),y2a),(za(18),y3a),
  & (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
  & (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
  double precision hb,a1b,vb,s1b,w1b,c1b,s2b,w2b,c2b,s3b,w3b,
  & xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
  & tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
  equivalence (zb(1),hb),(zb(2),a1b,vb),
  & (zb(3),s1b),(zb(4),w1b),(zb(5),c1b),
  & (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
  & (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
  & (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
  & (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
  & (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
  & (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
  double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
  & h,phih,rtot,xtot,ztot,do
  equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
  & (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
  & (z(59),tana7),(z(60),tana8),(z(61),l),
  & (z(62),h),(z(63),phih),
  & (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
  integer*2 iuks
  equivalence (uz(3),iuks)

```

```

double precision pi,halfpi,degrad,raddeg,zero,one,half
integer*2 izero,ione,iitwo
common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
& izero,ione,iitwo

integer*2 iscopa,iscopb,itanb,itanb,iitwo
double precision epsy,gamma,se
common /VCMPD/ epsy,gamma,se,iscopa,iscopb,itanb,itanb,iitwo

integer*2 iold
double precision ss0,dien0,ss1,dien1,ss2,dien2,slp0,sa0,smin(2)
common /VEQUAL/ ss0,dien0,ss1,dien1,ss2,dien2,slp0,sa0,smin,
& iold
equivalence (smin(1),samin),(smin(2),sbmin)

double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
& eex0,eez0,eeey0,a0,b0,phia0,phib0
integer*2 icase
common /VSPID/ sa,sb,ca,cb,vc0a,vc0b,
& eex0,eez0,eeey0,a0,b0,phia0,phib0,
& icase

double precision snphih,csphih,snafh,csafh,tnafh,scaf,dsnph
common /VHDIR/ snphih,csphih,snafh,csafh,tnafh,scaf,dsnph

double precision htnafh,hw4,w4h,s4w4h,c3h
common /VHVEC/ htnafh,hw4,w4h,s4w4h,c3h

double precision epsxz,xziru(2),xzbas(2),hbas(2),scra1(10)
common /VCSSXZ/ epsxz,xziru,xzbas,hbas,scra1
double precision xiru,ziru,xbas,zbas,hbasx,hbasz
equivalence (xziru(1),xiru),(xziru(2),ziru),
& (xzbas(1),xbas),(xzbas(2),zbas),
& (hbas(1),hbasx),(hbas(2),hbasz)

integer*2 itant
double precision a,b,snphi,tnafa,tnafb,
& seco7,seco8,ut,st,ykt,zkt,eex,eez,eeey,ybuoy
common /VCSSHP/ a,b,snphi,tnafa,tnafb,
& seco7,seco8,ut,st,ykt,zkt,eex,eez,eeey,ybuoy,itant

integer*2 ivs
double precision v0,v1,v2,f0,f1,f2,f,eps

```

```

common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps,ivs
double precision varray(3),farray(3)
equivalence (v0,varray),(f0,farray)

integer*2 ilh0,ilh1,ilh2,il
double precision lh0,lh1,lh2,ce
common /VSC0IL/ lh0,lh1,lh2,ce,ilh0,ilh1,ilh2,il
integer*2 ilh(3)
double precision lh(3)
equivalence (ilh,ilh0),(ilh,lh0)

double precision xred
integer*2 isidf,nerra,nerrb
common /VSTAB/ xred,isidf,nerra,nerrb
*****
* BEGIN EXECUTABLE CODE
*****
call ovlink('CPREP0 ')
if (ileg eq 3) call CPREP1
goto (1000,2000,3000), iuks

1000 continue
call CPREP2
call CPREP3
if (ileg ne 3) goto 1200
call ovlink('CSSHP ')
goto 1500
1200 continue
call ovlink('CSEHP ',0)
1500 continue
goto 5000

2000 continue
call CPREP2
if (ileg ne 3) goto 2200
call ovlink('CSSPR ',0)
goto 2500
2200 continue
call ovlink('CSEPR ',0)
2500 continue
goto 5000

3000 continue

```

93

```
if (ileg ne 3) goto 3200
call ovlint('CSSXZ1 ')
call ovlint('CSSXZ2 ')
goto 3500
3200 continue
call ovlint('CSEXZ1 ')
call ovlint('CSEXZ2 ')
3500 continue
5000 continue
return
end
*
```

93

```

ei sys f1ncl/12for/cprep0 for11
subroutine CPREP0
*****
implicit integer*2 (i)
implicit double precision (a-z)

integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
double precision z(67),cz,cx,d,ta,tb
common /VCL08/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
& isol,ibrnch,uz
double precision za(25),zb(25)
equivalence (z(1),za(1)),(z(26),zb(1))
double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
& xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
equivalence (za(1),ha),(za(2),ala,va),
& (za(3),sla),(za(4),wla),(za(5),cla),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),x1a),(za(14),x2a),(za(15),x3a),
& (za(16),y1a),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
& xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
equivalence (zb(1),hb),(zb(2),alb,vb),
& (zb(3),slb),(zb(4),wlb),(zb(5),clb),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
& (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
& (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phih,rtot,xtot,ztot,do
equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phih),
& (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
integer*2 nc(2)
equivalence (nca,nc)

```

```

integer*2 uz1,uz2
equivalence (uz(1),uz1),(uz(2),uz2)

double precision pi,halfpi,degrad,raddeg,zero,one,half
integer*2 izero,ione,itywo
common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
& izero,ione,itywo

double precision delyk,twod,halfd,dsq
common /VANCH/ delyk,twod,halfd,dsq

integer*2 iscopa,iscopb,itanb,itanb,ityis
double precision epsy,gamma,se
common /VCHPD/ epsy,gamma,se,iscopa,iscopb,itanb,itanb,ityis

integer*2 itold
double precision ss0,dten0,ss1,dten1,ss2,dten2,slp0,so0,smin(2)
common /VEQUAL/ ss0,dten0,ss1,dten1,ss2,dten2,slp0,so0,smin,
& itold
equivalence (smin(1),samin),(smin(2),sbmin)

integer*2 il,i,j,n
*****
itanb=2*ncb+17
iscopa=3*nce
iscopb=3*ncb+25
se=z(iscopa)+z(iscopb)

yk=-cz*halfd
do 20 il=1,2
smn=(do-yk)-s4
n=nc(il)
do 10 j=1,n
if (i.eq.1) goto 10
j=25*(il-1)+3*(i-1)
smn=smn-z(j)
10 continue
smin(1)=dmax1(smn,zero)
yk=-yk
20 continue

slp0=dmax1(dmin1(slp,z(iscopa)-samin),sbmin-z(iscopb))

```

96

sa0-z(,scopa)

return  
end

\*

96

```

ei sys final/i2for/cprepl for##
  subroutine CPREP1
*****
  implicit double precision (a-z)

  integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
  double precision z(67),cz,cx,d,ta,tb
  common /VCL08/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
& isol,ibrnch,uz
  double precision za(25),zb(25)
  equivalence (z(1),za(1)),(z(26),zb(1))
  double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
& xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
  equivalence (za(1),ha),(za(2),ala,va),
& (za(3),sla),(za(4),wla),(za(5),cla),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),x1a),(za(14),x2a),(za(15),x3a),
& (za(16),y1a),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
  double precision hb,a1b,vb,s1b,w1b,c1b,s2b,w2b,c2b,s3b,w3b,
& xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
  equivalence (zb(1),hb),(zb(2),a1b,vb),
& (zb(3),s1b),(zb(4),w1b),(zb(5),c1b),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
& (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
& (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
  double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phih,ctot,xtot,ztot,do
  equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phih),
& (z(64),ctot),(z(65),xtot),(z(66),ztot),(z(67),do)

  double precision pi,halfpi,degrad,raddeg,zero,one,half
  integer*2 izero,ione,ltwo

```

```

common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
&  izero,ione,iiwo

double precision delyk,twod,halfd,dsq
common /VANCH/ delyk,twod,halfd,dsq

double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
&  eex0,eez0,eey0,a0,b0,phia0,phib0
integer*2 icase
common /VSPID/ sa,sb,ca,cb,vc0a,vc0b,
&  eex0,eez0,eey0,a0,b0,phia0,phib0,
&  icase

equivalence (czsq1,eez0),(ddsq,eta,eex0),(a0sq,eey0),
&  (a0mbb,eex0sq,b0sq,phib0),(a0sq,phia0)
*****
call SUMSC(nca,za,sa,ca)
call SUMSC(ncb,zb,sb,cb)
call VCRIT0(nca,za,vc0a)
call VCRIT0(ncb,zb,vc0b)
czsq1=cz*c2+one
ddsq=dsq*czsq1
icase=1
if ((sa-sb)**2 gt ddsq) goto 100
icase=2
a0sq=sa*sa
a0mbb=a0sq-sb*sb
eta=dsqrt(((4.0d0*a0sq*ddsq-(ddsq+a0mbb)**2)/(cx*cx+czsq1)))
eez0=-(a0mbb+cz*cx*eta)/(twod*czsq1)
eex0=eta/twod
eey0=cx*eex0+cz*eez0
eex0sq=eex0*eex0
a0sq=eex0sq+(eez0-halfd)**2
b0sq=eex0sq+(eez0+halfd)**2
a0=dsqrt(a0sq)
b0=dsqrt(b0sq)
call PHIAB(a0,b0,a0sq-b0sq,dsq,twod,phia0,phib0)
100 continue
return
end

```

\*

```
ei sys final/12for/sumsc for!!  
  subroutine SUMSC(nc,z,s,c)  
*****  
  implicit double precision (a-z)  
  
  integer*2 nc  
  double precision z(25),s,c  
*****  
  s=z(3)  
  c=s*z(4)  
  if (nc eq 1) goto 100  
  s=s+z(6)  
  c=c+z(5)+z(6)*z(7)  
  if (nc eq 2) goto 100  
  s=s+z(9)  
  c=c+z(8)+z(9)*z(10)  
100 continue  
  return  
  end  
*
```

```

ei sys final/i2for/cprep2 for**
  subroutine CPREP2
  ****
  implicit double precision (a-z)

  integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
  double precision z(67),cz,cx,d,ta,tb
  common /VGL08/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
    & isol,ibrnch,uz
  double precision za(25),zb(25)
  equivalence (z(1),za(1)),(z(26),zb(1))
  double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
    & xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
    & tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
  equivalence (za(1),ha),(za(2),ala,va),
    & (za(3),sla),(za(4),wla),(za(5),cla),
    & (za(6),s2a),(za(7),w2a),(za(8),c2a),
    & (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
    & (za(13),x1a),(za(14),x2a),(za(15),x3a),
    & (za(16),y1a),(za(17),y2a),(za(18),y3a),
    & (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
    & (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
  double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
    & xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
    & tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
  equivalence (zb(1),hb),(zb(2),alb,vb),
    & (zb(3),slb),(zb(4),wlb),(zb(5),clb),
    & (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
    & (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
    & (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
    & (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
    & (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
    & (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
  double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
    & h,phih,rtoi,xtoi,ztoi,do
  equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
    & (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
    & (z(59),tana7),(z(60),tana8),(z(61),l),
    & (z(62),h),(z(63),phih),
    & (z(64),rtoi),(z(65),xtoi),(z(66),ztoi),(z(67),do)

  double precision pi,halfpi,deggrad,raddeg,zero,one,half
  integer*2 izero,ione,itwo

```

101

```
common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,  
& izero,ione,itwo  
  
double precision inaf,phif  
common /VOFLR/ inaf,phif  
  
double precision snphih,csphih,snafh,csafh,inafh,scafh,dsnph  
common /VHDIR/ snphih,csphih,snafh,csafh,inafh,scafh,dsnph  
*****  
csphih=dcos(phih)  
snphih=dsin(phih)  
inafh=dcos(phih-phif)*inaf  
scafh=SECNT(inafh)  
snafh=inafh/scafh  
csafh=one/scafh  
dsnph=d*snphih  
return  
end  
*
```

101

```

er sys final/r2for/cprep3 for!!
subroutine CPREP3
*****
implicit double precision (a-z)

integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
double precision z(67),cz,cx,d,ta,tb
common /VGL0B/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
& isol,ibrnch,uz
double precision za(25),zb(25)
equivalence (z(1),za(1)),(z(26),zb(1))
double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
& xa,ya,xla,x2a,x3a,yla,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
equivalence (za(1),ha),(za(2),ala,va),
& (za(3),sla),(za(4),wla),(za(5),cla),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),xla),(za(14),x2a),(za(15),x3a),
& (za(16),yla),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
& xb,yb,xlb,x2b,x3b,ylb,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
equivalence (zb(1),hb),(zb(2),alb,vb),
& (zb(3),slb),(zb(4),wlb),(zb(5),clb),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
& (zb(13),xlb),(zb(14),x2b),(zb(15),x3b),
& (zb(16),ylb),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phih,rtot,xtot,ztot,do
equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phih),
& (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

double precision snphih,csphih,snafh,csafh,tanafh,scafh
common /VHDIR/ snphih,csphih,snafh,csafh,tanafh,scafh

```

```
double precision h1nafh,hw1,w1h,s1w1h,c3h
common /VHVEC/ h1nafh,hw1,w1h,s1w1h,c3h
*****
h1nafh=h1nafh
hw1=h/w1
w1h=w1/h
s1w1h=s1*w1h
c3h=c3/h
return
end
*
```

```

e1 sys final/12for/csshp for!!
subroutine CSSHP
*****
implicit integer*2 (*)
implicit double precision (a-z)

integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
double precision z(67),cz,cx,d,ra,rb
common /VGL08/ ileg,ist,nca,ncb,z,cz,cx,d,ra,rb,nwa,nwb,
& isol,ibrnch,uz
double precision za(25),zb(25)
equivalence (z(1),za(1)),(z(26),zb(1))
double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
& xa,ya,xla,x2a,x3a,yla,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
equivalence (za(1),ha),(za(2),ala,va),
& (za(3),sla),(za(4),wla),(za(5),cla),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),xla),(za(14),x2a),(za(15),x3a),
& (za(16),yla),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
& xb,yb,xlb,x2b,x3b,ylb,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
equivalence (zb(1),hb),(zb(2),alb,vb),
& (zb(3),slb),(zb(4),wlb),(zb(5),clb),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
& (zb(13),xlb),(zb(14),x2b),(zb(15),x3b),
& (zb(16),ylb),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phih,rtot,xtot,ztot,do
equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phih),
& (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

double precision pi,halfpi,degrad,raddeg,zero,one,half

```

```

integer*2 izero,ione,itywo
common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
& izero,ione,itywo

double precision tnafe,phif
common /VOFLR/ tnafe,phif

double precision delyk,twod,halfd,dsq
common /VANCH/ delyk,twod,halfd,dsq

integer*2 iscopa,iscopb,itanb,itanb,ie
double precision epsy,gamma,se
common /VCMPD/ epsy,gamma,se,iscopa,iscopb,itanb,itanb,ie

double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
& eex0,eez0,eev0,a0,b0,phia0,phib0
integer*2 icase
common /VSPID/ sa,sb,ca,cb,vc0a,vc0b,
& eex0,eez0,eev0,a0,b0,phia0,phib0,
& icase

double precision snphi,csphi,snaf,csaf,tnaf,scaf
common /VHDIR/ snphi,csphi,snaf,csaf,tnaf,scaf

double precision htnaf,hw4,w4h,s4w4h,c3h
common /VHVEC/ htnaf,hw4,w4h,s4w4h,c3h

integer*2 itan1
double precision a,b,snphi,tnaf,tnafb,
& seca7,seca8,ut,st,ykt,zkt,eex,eez,eev,ybuoy
common /VCSSHP/ a,b,snphi,tnaf,tnafb,
& seca7,seca8,ut,st,ykt,zkt,eex,eez,eev,ybuoy,itan1

integer*2 ivs
double precision v0,v1,v2,f0,f1,f2,f,eps
common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps,ivs

double precision xred
integer*2 isidf,nerna,nernb
common /VSTAB/ xred,isidf,nerna,nernb

integer*2 isb,nw,intest,nern
*****

```

```

epsy=do*1 0d-10
ibrnch=1
ieb=0

nw=0
if (nwa eq 0 and nwb eq 0) goto 500
nw=1
if (nwa eq 1 and nwb eq 1) goto 4000
if (nwb eq 1) ibrnch=2
goto 3000

*****
* Determine the number of branches under tension
* when junction lies on ocean floor
*****
500 continue
*****
* If branch lengths differ by more than distance between anchors,
* then one branch is under tension
*****
if (icase eq 1) goto 2000
*****
* If load is directed outside angle formed by branch extensions,
* then one branch is under tension
*****
if (phih lt phia0 or phih gt phib0) goto 2000

*****
* Both branches are under tension if junction lies on ocean floor
* Assume this to be the case, with branches forming a triangle
* Calculate buoy elevation when junction is just lifted off floor
*****
phia=phia0
phib=phib0
call HSPLIT
iana7=(c3+ha*inafa+hb*inafb)/h
call TRISR
ybuoy=eev0+y4
if (ybuoy lt do) goto 1200
*****
* Computed buoy elevation is no less than water level,
* therefore junction lies on ocean floor
* Calculate riser displacements and angles directly

```

```

*****
call SRISR(eey0,one)
call TSPLIT
call JUNCT(inafa,inafb,0)
eex=eex0
eez=eez0
isol=1
goto 5000
*****
* Computed buoy elevation is less than water level,
* therefore junction lies above ocean floor
* Test each branch for plane solution before searching for solution
* in three dimensions, shorter branch is first to be tested
*****
1200 continue
    if (sa gt sb) ibrnch=2
    goto 3000

*****
* One branch is under tension if junction lies on ocean floor
* This is true for one for two reasons, as indicated by index 'icase'
*   1 if branch lengths differ by more than distance between anchors
*   2 if load is directed outside angle formed by branch extensions
* Assume that junction lies on ocean floor
*****
* Determine index 'ibrnch' of branch under tension 1 for A, 2 for B
*****
2000 continue
    if (icase ne 1) goto 2010
    if (sa gt sb) ibrnch=2
    goto 2015
2010 continue
    if (phih gt phib) ibrnch=2
2015 continue
*****
* Set parameters for branch under tension
* Calculate buoy elevation when junction is just lifted off ocean floor
*****
call EBUOY
    if (ybuoy lt do) goto 2200
*****
* Computed buoy elevation is no less than water level,
* therefore junction lies on ocean floor

```

```

* Calculate riser displacements and angles directly
*****
  call SRISR(eey,one)
  call JTEN(ia)
  ib=zero
  if (ibrnch eq 1) goto 2110
  ib=ia
  ia=zero
2110 continue
  gamma=pi
  eex=s1*csafh
  eez=z1+eex*snphih
  eex=eex*csphih
  isol=2
  goto 5000
*****
* Computed buoy elevation is less than water level,
* therefore junction lies above ocean floor
* Test each branch for plane solution before searching for solution
* in three dimensions, beginning with branch currently assumed to be
* under tension
*****
2200 continue
  ieb=1

*****
* Possibility of junction on floor has been eliminated
* For each branch, find solution in vertical plane of loading force,
* and compute straight-line distances for other branch
*****
3000 continue
  ntest=1
3010 continue
  if (ieb eq 1) goto 3100
  call EBUOY
  if (nw eq 0 and ybuoy gt do) goto 3500
3100 continue
  if (ibrnch eq 2) goto 3150
  call SC0IL(nca,za,vc0a,ca,nwa,ncb,zb,vc0b,sb,nerr)
  goto 3200
3150 continue
  call SC0IL(ncb,zb,vc0b,cb,nwb,nca,za,vc0a,sa,nerr)
3200 continue

```

```

      if (nerr eq 0 and coil ge zero) goto 3600
*****
* No possibility of plane solution in current tension branch
* Either depth is insufficient (from subroutine EBUOY,
* or other branch is too short (from subroutine SCOLL)
*****
3500 continue
      if (intest eq 2 or nw eq 1) goto 4000
      ibrnch=3-ibrnch
      ntest=2
      ieb=0
      goto 3010
*****
* Solution in plane of loading force is consistent with length of
* other branch
*****
3600 continue
      isol=3
      goto 5000

*****
* Solution must be three-dimensional with junction above ocean floor,
* solve by iteration over sides a,b of horizontal triangle
*****
4000 continue
      call STEFAB
      call JUNCT(zol(tlanb),zbl(tlanb),1)
      eex=xo*dcos(phia)
      eez=halfd+xo*dsin(phia)
      isol=4

*****
* Final computations for all solution types
*****
5000 continue
      gamma=dexp(gamma*frct)
      x101=eex+x4*csphih
      z101=eez+x4*snphih
      r101=csphih*x101+snphih*z101
      return
      end

```

\*

```

ei sys final/12for/phiab for##
  subroutine PHIAB(a,b,aambb,dsq,twod,phia,phib)
*****
  implicit double precision (a-z)

  double precision a,b,aambb,dsq,twod,phia,phib

  double precision pi,halfpi,degrad,raddeg,zero,one,half
  integer*2 izero,ione,iitwo
  common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
& izero,ione,iitwo
*****
  phia=acos((dsq+aambb)/(a*twod))-halfpi
  phib=halfpi-acos((dsq-aambb)/(b*twod))
  return
end
*

```

```

el sys final/12for/hsplit for!!
  subroutine HSPLIT
*****
  implicit double precision (a-z)

  integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
  double precision z(67),cz,cx,d,ta,tb
  common /VGL08/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
& isol,ibrnch,uz
  double precision za(25),zb(25)
  equivalence (z(1),za(1)),(z(26),zb(1))
  double precision ha,a1a,va,s1a,w1a,c1a,s2a,w2a,c2a,s3a,w3a,
& xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
  equivalence (za(1),ha),(za(2),a1a,va),
& (za(3),s1a),(za(4),w1a),(za(5),c1a),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),x1a),(za(14),x2a),(za(15),x3a),
& (za(16),y1a),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
  double precision hb,a1b,vb,s1b,w1b,c1b,s2b,w2b,c2b,s3b,w3b,
& xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
  equivalence (zb(1),hb),(zb(2),a1b,vb),
& (zb(3),s1b),(zb(4),w1b),(zb(5),c1b),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
& (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
& (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
  double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phih,rtot,xtot,ztot,do
  equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phih),
& (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

  double precision tnaf,phif
  common /VOFLR/ tnaf,phif

```

```

double precision a,b,snphi,tnafa,tnafb,
&   seca7,seca8,ut,st,ykt,zkt,eex,eez,eev,ybuoy
common /VCSSHP/ a,b,snphi,tnafa,tnafb,
&   seca7,seca8,ut,st,ykt,zkt,eex,eez,eev,ybuoy
*****
snphi=dsin(phi b-phi a)
ha=h*dsin(phi b-phi h)/snphi
hb=h*dsin(phi h-phi a)/snphi
tnafa=dcos(phi a-phi f)*tnaf
tnafb=dcos(phi b-phi f)*tnaf
return
end

```

\*

```

et sys final/12for/trisr for**
subroutine TRISR
*****
implicit double precision (a-z)

integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
double precision z(67),cz,cx,d,ta,tb
common /VGL08/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
& isol,ibrnch,uz
double precision za(25),zb(25)
equivalence (z(1),za(1)),(z(26),zb(1))
double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
& xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
equivalence (za(1),ha),(za(2),ala,va),
& (za(3),sla),(za(4),wla),(za(5),cla),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),x1a),(za(14),x2a),(za(15),x3a),
& (za(16),y1a),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
double precision hb,alb,vb,s1b,w1b,c1b,s2b,w2b,c2b,s3b,w3b,
& xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
equivalence (zb(1),hb),(zb(2),alb,vb),
& (zb(3),s1b),(zb(4),w1b),(zb(5),c1b),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
& (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
& (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phih,rtot,x1ot,z1ot,do
equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phih),
& (z(64),rtot),(z(65),x1ot),(z(66),z1ot),(z(67),do)

double precision hinafh,hw4,w4h,s4w4h,c3h
common /VHVEC/ hinafh,hw4,w4h,s4w4h,c3h

```

```

double precision a,b,snphi,inafa,inafb,
&   seca7,seca8,ut,st,ykt,zkt,eex,eez,eeey,ybuoy
common /VCSSH/ a,b,snphi,inafa,inafb,
&   seca7,seca8,ut,st,ykt,zkt,eex,eez,eeey,ybuoy
*****
      ina8=ina7+s4w4h
      call SCA7A8
      y4=hw4*(seca8-seca7)
      return
end
*
```

```

e1 sys final/i2for/sca7a8 for!!
  subroutine SCA7A8
  *****
  implicit double precision (a-z)

  integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
  double precision z(67),cz,cx,d,ta,tb
  common /VCL08/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
  & isol,ibrnch,uz
  double precision za(25),zb(25)
  equivalence (z(1),za(1)),(z(26),zb(1))
  double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
  & xa,ya,xla,x2a,x3a,yla,y2a,y3a,
  & tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
  equivalence (za(1),ha),(za(2),ala,va),
  & (za(3),sla),(za(4),wla),(za(5),cla),
  & (za(6),s2a),(za(7),w2a),(za(8),c2a),
  & (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
  & (za(13),xla),(za(14),x2a),(za(15),x3a),
  & (za(16),yla),(za(17),y2a),(za(18),y3a),
  & (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
  & (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
  double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
  & xb,yb,xlb,x2b,x3b,ylb,y2b,y3b,
  & tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
  equivalence (zb(1),hb),(zb(2),alb,vb),
  & (zb(3),slb),(zb(4),wlb),(zb(5),clb),
  & (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
  & (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
  & (zb(13),xlb),(zb(14),x2b),(zb(15),x3b),
  & (zb(16),ylb),(zb(17),y2b),(zb(18),y3b),
  & (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
  & (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
  double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
  & h,phih,rtot,xtot,ztot,do
  equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
  & (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
  & (z(59),tana7),(z(60),tana8),(z(61),l),
  & (z(62),h),(z(63),phih),
  & (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

  double precision a,b,snphi,tanaa,tanab,
  & seca7,seca8,ut,st,ykt,zkt,eez,eez,eez,ybuoy

```

116

```
common /VCSSHP/ a,b,snphi,inafa,inafb,  
& seca7,seca8,ui,si,ykl,zkl,eez,eez,eez,ybuoy  
*****  
seca7=SECNT(ina7)  
seca8=SECNT(ina8)  
return  
end  
*
```

116

```

e1 sys final/i2for/sr1sr for!!
subroutine SR1SR(ey,lfact)
*****
implicit double precision (a-z)

double precision ey,lfact

integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
double precision z(67),cz,cx,d,fa,fb
common /VGL08/ ileg,ist,nca,ncb,z,cz,cx,d,fa,fb,nwa,nwb,
& isol,ibrnch,uz
double precision za(25),zb(25)
equivalence (z(1),za(1)),(z(26),zb(1))
double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
& xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
equivalence (za(1),ha),(za(2),ala,va),
& (za(3),sla),(za(4),wla),(za(5),cla),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),x1a),(za(14),x2a),(za(15),x3a),
& (za(16),y1a),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
double precision hb,alb,vb,s1b,w1b,clb,s2b,w2b,c2b,s3b,w3b,
& xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
equivalence (zb(1),hb),(zb(2),alb,vb),
& (zb(3),s1b),(zb(4),w1b),(zb(5),clb),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
& (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
& (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phih,ctot,xtot,ztot,do
equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phih),
& (z(64),ctot),(z(65),xtot),(z(66),ztot),(z(67),do)

```

```

double precision pi,halfpi,degrad,raddeg,zero,one,half
integer*2 izero,ione,itiwo
common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
& izero,ione,itiwo

double precision snphih,csphih,snafh,csafh,tnafh,scafh
common /VHDIR/ snphih,csphih,snafh,csafh,tnafh,scafh

double precision hinafh,hw4,w4h,s4w4h,c3h
common /VHVEC/ hinafh,hw4,w4h,s4w4h,c3h
*****
y4=do-ey
l=LENS(y4,csafh,snafh,s4,w4,h)*lfac1
if (l gt zero) goto 20
l=zero
iana7=TANI(s4w4h,y4*w4h)
goto 50
20 continue
iana7=tnafh
50 continue
iana8=iana7+(s4-1)*w4h
call SCA7A8
call X4CALC
x4=x4+1*csafh
return
end

```

\*

```

er sys final/12for/tant for11
function TANI(d1,ds)
*****
* Computes tangent of the algebraically smaller of two angles,
* given the differences between their tangents and secants
*****
implicit double precision (a-z)

double precision tant,d1,ds

double precision pi,halfpi,degrad,raddeg,zero,one,half
integer*2 izero,ione,i1wo
common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
& izero,ione,i1wo
*****
* tant = tangent of smaller angle
* d1 = tangent of larger angle - tangent of smaller angle
* ds = secant of larger angle - secant of smaller angle
*****
tant=half*(ds*dsqrt(one+4.0d0/(d1*d1-ds*ds))-d1)
return
end
*

```

```

er sys final/12for/x4calc for!!
subroutine X4CALC
*****
implicit double precision (a-z)

integer*2 ilcg,ist,nca,ncb,nwa,nwb,isol,ibrnch,iuz(5)
double precision z(67),cz,cx,d,ta,tb
common /VGL0B/ ilcg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
& isol,ibrnch,iuz
double precision za(25),zb(25)
equivalence (z(1),za(1)),(z(26),zb(1))
double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
& xa,ya,xla,x2a,x3a,yla,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
equivalence (za(1),ha),(za(2),ala,va),
& (za(3),sla),(za(4),wla),(za(5),cla),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),xla),(za(14),x2a),(za(15),x3a),
& (za(16),yla),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
& xb,yb,xlb,x2b,x3b,ylb,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
equivalence (zb(1),hb),(zb(2),alb,vb),
& (zb(3),slb),(zb(4),wlb),(zb(5),clb),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
& (zb(13),xlb),(zb(14),x2b),(zb(15),x3b),
& (zb(16),ylb),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
double precision co1,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phih,ctot,xtot,ztot,do
equivalence (z(51),co1),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phih),
& (z(64),ctot),(z(65),xtot),(z(66),ztot),(z(67),do)

double precision hinafh,hw4,w4h,s4w4h,c3h
common /VHVEC/ hinafh,hw4,w4h,s4w4h,c3h

```

```
double precision a,b,snphi,tnafa,tnafb,  
& seca7,seca8,ui,si,ykl,zkl,eex,eez,eev,ybuoy  
common /VCSSHP/ a,b,snphi,tnafa,tnafb,  
& seca7,seca8,ui,si,ykl,zkl,eex,eez,eev,ybuoy  
*****  
x4=hw4*dlog((tna8+seca8)/(tna7+seca7))  
return  
end  
*
```

```

er sys final/12for/1split for
subroutine TSPLIT
*****
implicit double precision (a-z)

integer*2 ilcg,ist,ncd,ncb,nwd,nwb,isol,ibrnch,iuz(5)
double precision z(67),cz,cx,d,ta,lb
common /VGLOBAL/ ilcg,ist,ncd,ncb,z,cz,cx,d,ta,lb,nwd,nwb,
& isol,ibrnch,iuz
double precision za(25),zb(25)
equivalence (z(1),za(1)),(z(26),zb(1))
double precision ha,a1a,va,s1a,w1a,c1a,s2a,w2a,c2a,s3a,w3a,
& xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
equivalence (za(1),ha),(za(2),a1a,va),
& (za(3),s1a),(za(4),w1a),(za(5),c1a),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),x1a),(za(14),x2a),(za(15),x3a),
& (za(16),y1a),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
double precision hb,a1b,vb,s1b,w1b,c1b,s2b,w2b,c2b,s3b,w3b,
& xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
equivalence (zb(1),hb),(zb(2),a1b,vb),
& (zb(3),s1b),(zb(4),w1b),(zb(5),c1b),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
& (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
& (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
double precision ca,sla,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phia,rtot,xtot,ztot,do
equivalence (z(51),cot1),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phia),
& (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
double precision b,sinb,cosb,tanb,secb
equivalence (z(25),b),(z(26),sinb),(z(27),cosb),(z(28),tanb),
& (z(29),secb)

```

```

integer*2 uz1,uz2,uz3,uz4,uz5
equivalence (uz(1),uz1),(uz(2),uz2),(uz(3),uz3),(uz(4),uz4),
& (uz(5),uz5)

double precision pi,halfpi,degrad,raddeg,zero,one,half
integer*2 izero,ione,itywo
common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
& izero,ione,itywo

double precision delyk,twod,halfd,dsq
common /VANCH/ delyk,twod,halfd,dsq

double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
& eex0,eez0,eev0,a0,b0,phia0,phib0
integer*2 icase
common /VSPID/ sa,sb,ca,cb,vc0a,vc0b,
& eex0,eez0,eev0,a0,b0,phia0,phib0,
& icase

double precision snphih,csphih,snafh,csafh,tnafh,scafh
common /VHDIR/ snphih,csphih,snafh,csafh,tnafh,scafh

equivalence (czsq1,ddsq,phibb,eex),(fx,phihh,eez),
& (fz,dd,snphih,eev),(phiaa,gamma)
*****
czsq1=cz*c2+one
fx=czsq1*csphih-cx*c2*snphih+cx*tnafh
fz=snphih+c2*tnafh
if (fx ne zero) goto 20
if (fz li zero) goto 10
phihh=halfpi
goto 15
10 continue
phihh=-halfpi
15 continue
goto 50
20 continue
phihh=datan(dsqr1(cx*cx+czsq1))*fz/fx
50 continue
ddsq=dsq*czsq1
dd=dsqr1(ddsq)
call PHIB(sa,sb,sa*sa-sb*sb,ddsq,dd+dd,phiaa,phibb)
snphih=dsin(phibb-phiaa)

```

124

```
call JTEN(tenj)
ia=tenj*dsin(phibb-phihh)/snphh
ib=tenj*dsin(phihh-phiao)/snphh
return
end
```

\*

124

```

ei sys final/i2for/jten for##
  subroutine JTEN(ienj)
*****
  implicit double precision (a-z)

  double precision ienj

  integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
  double precision z(67),cz,cx,d,ta,tb
  common /VGL08/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
& isol,ibrnch,uz
  double precision za(25),zb(25)
  equivalence (z(1),za(1)),(z(26),zb(1))
  double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
& xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
  equivalence (za(1),ha),(za(2),ala,va),
& (za(3),sla),(za(4),wla),(za(5),cla),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),x1a),(za(14),x2a),(za(15),x3a),
& (za(16),y1a),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
  double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
& xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
  equivalence (zb(1),hb),(zb(2),alb,vb),
& (zb(3),slb),(zb(4),wlb),(zb(5),clb),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
& (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
& (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
  double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phih,rtot,xtot,ztot,do
  equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phih),
& (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)
  double precision b,sinb,cosb,tanb,secb

```

```

equivalence (z(25),b),(z(26),sinb),(z(27),cosb),(z(28),tanb),
& (z(29),secb)
integer*2 uz1,uz2,uz3,uz4,uz5
equivalence (uz(1),uz1),(uz(2),uz2),(uz(3),uz3),(uz(4),uz4),
& (uz(5),uz5)

double precision snph1h,csph1h,snafh,csafh,tnafh,scaf1h
common /VHDIR/ snph1h,csph1h,snafh,csafh,tnafh,scaf1h
*****
tenj=h*SECNT(tana7)*dcos(datan(tana7)-datan(tnafh))
& -(c3+)*w4)*snafh
return
end
*
```

```

et sys final/i2for/junct for##
  subroutine JUNCT(tna,tnb,index)
  *****
  implicit double precision (a-z)

  integer*2 index
  double precision tna,tnb

  integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
  double precision z(67),cz,cx,d,ta,tb
  common /VGL0B/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
& isol,ibrnch,uz
  double precision za(25),zb(25)
  equivalence (z(1),za(1)),(z(26),zb(1))
  double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
& xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
  equivalence (za(1),ha),(za(2),ala,va),
& (za(3),sla),(za(4),wla),(za(5),cla),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),x1a),(za(14),x2a),(za(15),x3a),
& (za(16),y1a),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
  double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
& xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
  equivalence (zb(1),hb),(zb(2),alb,vb),
& (zb(3),slb),(zb(4),wlb),(zb(5),clb),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
& (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
& (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
  double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phih,rtot,xtot,ztot,do
  equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phih),
& (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

```

```

double precision pi,halfpi,degrad,raddeg,zero,one,half
integer*2 izero,ione,iitwo
common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
& izero,ione,iitwo

integer*2 iscopa,iscopb,itana,itamb,ie
double precision epsy,gamma,se
common /VCHPD/ epsy,gamma,se,iscopa,iscopb,itana,itamb,ie

double precision a,b,snphi,tnafa,tnafb,
& seca7,seca8,ut,st,ykt,zkt,eex,eez,eeey,ybuoy
common /VCSSHP/ a,b,snphi,tnafa,tnafb,
& seca7,seca8,ut,st,ykt,zkt,eex,eez,eeey,ybuoy

equivalence (csphi,gamma),(sca,eex),(scb,eez)
*****
sca=SECNT(tna)
scb=SECNT(tnb)
csphi=dsqrt(one-snphi*snphi)
if (phib-phia gt halfpi) csphi=-csphi
gamma=pi-dacos((tna*tnb+csphi)/(sca*scb))
if (index ne ione) goto 100
ia=ha*sca
ib=hb*scb
100 continue
return
end
*
```

```

ei sys final/12for/ebuoy for**
subroutine EBUOY
*****
* Assigns values to parameters ut,st,ykt,zkt for branch assumed to be
* under tension Assumes junction to be just off ocean floor
* Computes junction y-coord, vertical riser displacement, buoy y-coord
*
* ibrnch = index of tension branch 1 for A, 2 for B
* ut = unit factor for tension branch 1 for A, -1 for B
* st = length of tension branch
* halhb = load on tension branch
* ykt = y-coord of tension branch anchor
* zkt = z-coord of tension branch anchor
* eev = y-coord of junction, in general, of point on floor
*       directly beneath junction
* ybuoy = y-coord of buoy
*****
implicit double precision (a-z)

integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
double precision z(67),cz,cx,d,ta,tb
common /VGL08/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
& isol,ibrnch,uz
double precision za(25),zb(25)
equivalence (z(1),za(1)),(z(26),zb(1))
double precision ha,da,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
& xa,ya,xla,x2a,x3a,yla,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
equivalence (za(1),ha),(za(2),da,va),
& (za(3),sla),(za(4),wla),(za(5),cla),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),xla),(za(14),x2a),(za(15),x3a),
& (za(16),yla),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
double precision hb,alb,vb,s1b,w1b,c1b,s2b,w2b,c2b,s3b,w3b,
& xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
equivalence (zb(1),hb),(zb(2),alb,vb),
& (zb(3),s1b),(zb(4),w1b),(zb(5),c1b),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),

```

```

& (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
& (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phih,rioi,xioi,zioi,do
equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phih),
& (z(64),rioi),(z(65),xioi),(z(66),zioi),(z(67),do)

double precision pi,halfpi,degrad,raddeg,zero,one,half
integer*2 izero,ione,ilwo
common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
& izero,ione,ilwo

double precision delyk,twod,halfd,dsq
common /VANCH/ delyk,twod,halfd,dsq

double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
& eex0,eez0,eev0,a0,b0,phia0,phib0
integer*2 icase
common /VSPID/ sa,sb,ca,cb,vc0a,vc0b,
& eex0,eez0,eev0,a0,b0,phia0,phib0,
& icase

double precision snphih,csphih,snafh,csafh,tnafh,scafh
common /VHDIR/ snphih,csphih,snafh,csafh,tnafh,scafh

double precision htnafh,hw4,w4h,s4w4h,c3h
common /VHVEC/ htnafh,hw4,w4h,s4w4h,c3h

integer*2 iscopa,iscopb,itanb,itanb,ie
double precision epsy,gamma,se
common /VCPD/ epsy,gamma,se,iscopa,iscopb,itanb,itanb,ie

integer*2 itant
double precision a,b,snphi,tnafa,tnafb,
& seca7,seca8,ut,st,ykt,zkt,eex,eez,eev,ybuov
common /VCSSHP/ a,b,snphi,tnafa,tnafb,
& seca7,seca8,ut,st,ykt,zkt,eex,eez,eev,ybuov,itant
*****

```

11

131

```
if (ibrnch ne 11) goto 20
ui=one
si=sa
ha=h
itan1=itana
goto 50
20 continue
ui=-one
si=sb
hb=h
itan1=itanb
50 continue
zki=ui*halfd
yki=c2*zki
eey=yki+si*snafh
tana7=tanafh+c3h
call TRISR
ybuoy=eey+y4
return
end
```

\*

131

```

ei sys final/12for/scoil for!!
  subroutine SCOIL(ncf,zf,vc0f,cf,nwf,ncc,zc,vc0c,sc,nerr)
*****
  implicit integer*2 (n)
  implicit double precision (a-z)

  integer*2 ncf,nwf,ncc,nerr
  double precision zf(25),vc0f(6),cf,zc(25),vc0c(6),sc

  integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
  double precision z(67),cz,cx,d,ta,tb
  common /VGL08/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
& isol,ibrnch,uz
  double precision za(25),zb(25)
  equivalence (z(1),za(1)),(z(26),zb(1))
  double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
& xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
  equivalence (za(1),ha),(za(2),ala,va),
& (za(3),sla),(za(4),wla),(za(5),cla),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),x1a),(za(14),x2a),(za(15),x3a),
& (za(16),y1a),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
  double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
& xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
  equivalence (zb(1),hb),(zb(2),alb,vb),
& (zb(3),slb),(zb(4),wlb),(zb(5),clb),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
& (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
& (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
  double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phih,rtot,xtot,ztot,do
  equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phih),

```

```

& (z(64),r1ot1),(z(65),x1ot1),(z(66),z1ot1),(z(67),do)

double precision pi,halfpi,degrad,raddeg,zero,one,half
integer*2 izero,ione,itywo
common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
& izero,ione,itywo

integer*2 iscopa,iscopb,itanb,itanb,ie
double precision epsy,gamma,se
common /VCMPO/ epsy,gamma,se,iscopa,iscopb,itanb,itanb,ie

double precision snphih,csphih,snafh,csafh,tnafh,scafh
common /VHDIR/ snphih,csphih,snafh,csafh,tnafh,scafh

double precision h1nafh,hw4,w4h,s4w4h,c3h
common /VHVEC/ h1nafh,hw4,w4h,s4w4h,c3h

integer*2 itant
double precision a,b,snphi,tnafa,tnafb,
& seco7,seco8,ut,st,ykt,zkt,eeex,eez,eev,ybuoy
common /VCSHP/ a,b,snphi,tnafa,tnafb,
& seco7,seco8,ut,st,ykt,zkt,eeex,eez,eev,ybuoy,itant

integer*2 ivs
double precision v0,v1,v2,f0,f1,f2,f,eps
common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps,ivs

integer*2 ilh0,ilh1,ilh2,il
double precision lh0,ilh1,ilh2,ce
common /VSCOIL/ lh0,ilh1,ilh2,ce,ilh0,ilh1,ilh2,il
integer*2 ilh(3)
double precision lh(3)
equivalence (ilh,ilh0),(ilh,ilh0)

equivalence (kte,coil),(kcesq,tona,gamma)
*****
* write(10,*) 'SCOIL',ibrnch,ybuoy
f-do
ivs=0
call XSECV(nc1,z1,vc01,st,ct,nwt,ncc,zc,vc0c,
& snafh,csafh,tnafh,scafh,4,nerr)
if (nerr ne 0) goto 5000

```

```
2000 continue
      call X4CALC
      kte=zi(11)
      eex=kte*csphi*h
      eez=zk1+kte*snphi*h
      kcesq=eex*eex+(zk1+eez)**2
      coil=sc-dsqrt(kcesq+(yk1+yk1+kte*inafh)**2)-lh2
      tanaJ=zi(11ant)
      ta=h*SECNT(tanaJ)
      tb=ce
      if (lbrnch eq 1) goto 2110
      ta=ta
      ta=ce
2110 continue
      gamma=datan(tanaJ)+halfpi
5000 continue
      return
      end
```

\*

```

e1 sys final/12for/xsecv for!!
  subroutine XSECV(nci,zl,vc0l,slp,cl,nwl,ncc,zc,vc0c,
    & sinaf,cosaf,tanaf,secaf,iflyp,nernf)
  *****
  implicit integer*2 (*)
  implicit double precision (a-z)

  integer*2 nci,nwl,ncc,iflyp,nernf
  double precision zl(25),vc0l(6),slp,cl,zc(25),vc0c(6),
    & sinaf,cosaf,tanaf,secaf

  integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,iuz(5)
  double precision z(67),cz,cx,d,ta,tb
  common /VCL08/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
    & isol,ibrnch,iuz
  double precision za(25),zb(25)
  equivalence (z(1),za(1)),(z(26),zb(1))
  double precision ha,sla,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
    & xa,ya,xla,x2a,x3a,yla,y2a,y3a,
    & tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
  equivalence (za(1),ha),(za(2),sla,va),
    & (za(3),sla),(za(4),wla),(za(5),cla),
    & (za(6),s2a),(za(7),w2a),(za(8),c2a),
    & (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
    & (za(13),xla),(za(14),x2a),(za(15),x3a),
    & (za(16),yla),(za(17),y2a),(za(18),y3a),
    & (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
    & (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
  double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
    & xb,yb,xlb,x2b,x3b,ylb,y2b,y3b,
    & tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
  equivalence (zb(1),hb),(zb(2),alb,vb),
    & (zb(3),slb),(zb(4),wlb),(zb(5),clb),
    & (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
    & (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
    & (zb(13),xlb),(zb(14),x2b),(zb(15),x3b),
    & (zb(16),ylb),(zb(17),y2b),(zb(18),y3b),
    & (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
    & (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
  double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
    & h,phih,rtot,xtot,ztot,do
  equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
    & (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),

```

```

& (z(59),tana7),(z(60),tana8),(z(61),1),
& (z(62),h),(z(63),phih),
& (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

double precision pi,halfpi,degrad,raddeg,zero,one,half
integer*2 izero,ione,ityo
common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
& izero,ione,ityo

double precision snphi,csphi,snaf,csaf,tnaf,scaf
common /VHDIR/ snphi,csphi,snaf,csaf,tnaf,scaf

double precision htnaf,hw4,w4h,s4w4h,c3h
common /VHVEC/ htnaf,hw4,w4h,s4w4h,c3h

integer*2 itant
double precision a,b,snphi,tnaf,tnafb,
& sec7,sec8,ut,stykt,zkt,eex,eez,eev,ybuoy
common /VCSHP/ a,b,snphi,tnaf,tnafb,
& sec7,sec8,ut,stykt,zkt,eex,eez,eev,ybuoy,itant

integer*2 ivs
double precision v0,v1,v2,f0,f1,f2,f,eps
common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps,ivs
double precision varray(3),farray(3)
equivalence (v0,varray),(f0,farray)

double precision fred
integer*2 isidf,nerrn,nerrb
common /VSTAB/ fred,isidf,nerrn,nerrb

integer*2 ilh0,ilh1,ilh2,il
double precision lh0,lh1,lh2,ce
common /VSC0IL/ lh0,lh1,lh2,ce,ilh0,ilh1,ilh2,il
integer*2 ilh(3)
double precision lh(3)
equivalence (ilh,ilh0),(ilh,lh0)

integer*2 nerr,init,ij,n,ivint,ndiv,isin
equivalence (ht,ta),(htanaf,tb),(htenth,gamma),(dv,eex),
& (vbase,eez),(fbase,eev),(absfb,rtot),(flim,xtot),(len,ztot),
& (crt,coi)

```

```

equivalence (len,delv,v0sav),(c1ht,delv0,vlow,vlsav),
& (delv1,vhigh,f0sav),(rat,hterm,flsav),(ratmax,rat0),
& (is,in,ivini)
*****
nerrf=1
ht=zt(1)
* write(10,*) 'XSECV',f,ht,iftyp
htanof=ht*ianof
small=1 0d-10
eps=f*small
if (ivs eq 0) goto 1100
call SHIFT(0,1)
call SHIFT(1,2)
goto 5500
1100 continue
ten=1 0d1
htenth=ht/ten
dv=ct*1 0d-3

n=2*nct
do 1200 i=2,n
if (vc01(i) lt vc01(i-1)) goto 1200
j=i-1
goto 1210
1200 continue
j=n
1210 continue
vbase=vc01(j)+htanof
v0=vbase
call SUBVX(nct,zt,vc01,ncc,zc,vc0c,sinaf,cosaf,ianof,secaf,
& htanof,0,iftyp,nerrf)
*****
* if (f0 ne zero) goto 1250
* call SHIFT(2,0)
* nerrf=0
* goto 6000
*1250 continue
*****
fbase=f0
absfb=dabs(fbase)
fred=absfb
flim=-f

```

```

      if (lftyp ne 1) flim=flim+slp
      if (lftyp eq 4) flim=flim+ykl+s4
*****
*      if (lftyp ne 2) goto 1500
*      len=LENS(f,cosaf,sinaf,slp,cl/slp,hl)
*      if (len gt zero) goto 1320
*      cthi=cl/hi
*      v0=hi*TANI(cthi,f*cthi/slp)+cl
*      goto 1350
*1320 continue
*      v0=cl*(one-len/slp)+hianaf
*1350 continue
*      v0=dmax1(v0,vbase)
*      call SUBVX(nc1,z1,vc01,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
*      & hianaf,0,2,nerr)
*      vl=v0+ten*dv
*      call SUBVX(nc1,z1,vc01,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
*      & hianaf,1,2,nerr)
*      goto 5500
*1500 continue
*****
      if (fbase*flim ge zero) goto 3000

*      write(10,*) '2'
      delvl=hienth
2100 continue
      vl=cl+hianaf+delvl
      call SUBVX(nc1,z1,vc01,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
      & hianaf,1,lftyp,nerr)
      if (f0*fl lt zero) goto 4000
      call SHIFT(0,1)
      delvl=delvl*ten
      goto 2100

3000 continue
*      write(10,*) '3'
      nit=0
      delvl=dv
      n0max=cl/(slp*small)
      vl=vbase+delvl
      goto 3150
3100 continue
      rat=(v0-vbase)/(f0-fbase)

```

```

        if (dabs(rat) gt ratmax) goto 3500
        vl=vbase-fbase*rat
3150 continue
        call SUBVX(ncr,zt,vc0t,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
& htranaf,l,iftyp,nerr)
        fred-dmin1(fred,dabs(f1))
        if (f0*f1 lt zero) goto 4000
        if (dabs(f1) ge absfb) goto 3500
        delv1=vl-v0
        if (nit le 1) goto 3160
        if (dabs(one-delv*delv1/delv0**2) lt one/ten) goto 5500
3160 continue
        call SHIFT(0,1)
        delv=delv0
        delv0=delv1
        nit=nit+1
        goto 3100

3500 continue
* write(10,*) '7'
        if (nit eq 0 and iftyp eq 1 and nwt eq 0) goto 6000
        vlow=vbase
        hterm=htrnth
        do 3600 ivint=1,4
            ndiv=7-ivint
            vhigh=ct+htranaf+hterm
            delv=vhigh-vlow
            n=1
            do 3550 i=1,ndiv
                vl=vlow+half*delv
                do 3540 j=1,n
                    call SUBVX(ncr,zt,vc0t,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
& htranaf,l,iftyp,nerr)
                    if (f0*f1 lt zero) goto 4000
                    fred-dmin1(fred,dabs(f1))
                    vl=vl+delv
3540 continue
                    n=n+n
                    delv=half*delv
3550 continue
*****
* vl=vhigh
* call SUBVX(ncr,zt,vc0t,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,

```

```

*      & hianaf,1,iftyp,nerr)
*      if (f0*f1 lt zero) goto 4000
*      fred=dmin1(fred,dabs(f1))
*****
vlow=vhigh
hterm=hterm*ten
3600 continue
goto 6000

4000 continue
write(10,*) '4'
nit=1
4100 continue
if (iftyp ne 4) goto 4200
if (dabs(v0-v1) gt dv or dabs(rat0) gt dv/sip) goto 4200
j=ih1-ih0
if (j*j ne 1) goto 4200
j=(3-j)/2
n=j*ih1
f=zero
is=3*ncc
do 4120 i=1,n
f=f+zc(is)
is=is-3
4120 continue
v0sav=v0
vlsav=v1
f0sav=f0
flsav=f1
f0=ih0-f
fl=ih1-f
call SECVIT(nct,zf,vc0t,ncc,zc,vc0c,snafh,csafh,tnafh,
& scafh,hinafh,3,nerr)
if (nerr ne 0) goto 4140
call SRISR(ykt+z1(12),zero)
ce=h*(tana7-z1(12)/tani)-c3
in=2*(ncc-j)
if (ce lt vc0c(in+1) or ce gt vc0c(in)) goto 4140
nerrf=0
goto 6000
4140 continue
f=do
v0=v0sav

```

```

v1=v1sav
f0=f0sav
f1=f1sav
ilh(j)=n
j=3-j
ilh(j)=n+1
4200 continue
v2=half*(v0+v1)
if (nit gt 3) v2=v1-f1*rat0
call SUBVX(nci,zi,vc0i,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
& htanaf,2,iftyp,nerrf)
j=1
if (f2*f0 gt zero) i=0
call SHIFT(i,2)
rat1=(v1-v0)/(f1-f0)
if (dabs(f0)+dabs(f1) lt eps*ten) goto 5500
if (nit ge 5 and dabs(rat0/(rat0-rat1)) gt ten and
& (iftyp ne 4 or ilh0 eq ilh1)) goto 5500
rat0=rat1
nit=nit+1
goto 4100

5500 continue
call SECVIT(nci,zi,vc0i,ncc,zc,vc0c,sinaf,cosaf,tanaf,
& secaf,htanaf,iftyp,nerrf)

6000 continue
if (fbase lt zero) fred= -fred
return
end

```

\*

```

e1 sys final/i2for/shift for!!
  subroutine SHIFT(i,j)
  *****
  implicit integer*2 (i)
  implicit double precision (a-z)

  integer*2 i,j

  integer*2 ivs
  double precision v0,v1,v2,f0,f1,f2,f,eps
  common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps,ivs
  double precision varray(3),farray(3)
  equivalence (v0,varray),(f0,farray)

  integer*2 ih0,ih1,ih2,il
  double precision ih0,ih1,ih2,ce
  common /VSC0IL/ ih0,ih1,ih2,ce,ih0,ih1,ih2,il
  integer*2 ilh(3)
  double precision ih(3)
  equivalence (ih,ilh0),(ih,ilh1)

  integer*2 ix,jx
  *****
  ix=i+1
  jx=j+1

  varray(ix)=varray(jx)
  farray(ix)=farray(jx)
  ih(ix)=ih(jx)
  ilh(ix)=ilh(jx)

  return
end
  
```

\*

```

et sys final/12for/subvx for 11
  subroutine SUBVX(ncf,zf,vc0f,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
    & htanaf,index,iftyp,nerr)
*****
  implicit double precision (a-z)

  integer*2 ncf,ncc,index,iftyp,nerr
  double precision zf(25),vc0f(6),zc(25),vc0c(6),sinaf,cosaf,tanaf,
    & secaf,htanaf

  double precision pi,halfpi,degrad,raddeg,zero,one,half
  integer*2 izero,ione,itywo
  common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
    & izero,ione,itywo

  double precision v0,v1,v2,f0,f1,f2,f,eps
  common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps
  double precision varray(3),farray(3)
  equivalence (v0,varray),(f0,farray)

  integer*2 ilh0,ilh1,ilh2,il
  double precision lh0,lh1,lh2,ce
  common /VSC0IL/ lh0,lh1,lh2,ce,ilh0,ilh1,ilh2,il
  integer*2 ilh(3)
  double precision lh(3)
  equivalence (ilh,ilh0),(ilh,lh0)

  integer*2 ix
*****
  ix=index+1
  zf(2)=varray(ix)
  call CALC2(ncf,zf,vc0f,sinaf,cosaf,tanaf,secaf,htanaf,ione,nerr)
  goto (110,120,130,130),iftyp
110 continue
  fval=zf(11)
  goto 200
120 continue
  fval=zf(12)
  goto 200
130 continue
  fval=zf(12)-zf(11)*tanaf
  lh(ix)=fval
  if (iftyp eq 3) goto 200

```

11

```
ce=WCTH(fval,ncc,zc,vc0c)
call CERISR(ce,zt,fval)
ilh(ix)=11
200 continue
farray(ix)=fval-f
return
end
*
```

```

ei sys final/12for/wgth for!!
function WOTH(lenh,nc,z,vc0)
*****
implicit double precision (a-z)

integer*2 nc
double precision wgth,lenh,z(25),vc0(6)

double precision pi,halfpi,degrad,raddeg,zero,one,half
integer*2 izero,ione,itywo
common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
& izero,ione,itywo

integer*2 ilh0,ilh1,ilh2,il
double precision lh0,lh1,lh2,ce
common /VSC0IL/ lh0,lh1,lh2,ce,ilh0,ilh1,ilh2,il
integer*2 ilh(3)
double precision lh(3)
equivalence (ilh,ilh0),(ilh,lh0)

integer*2 i,is,in
*****
lenl=zero
is=3*nc
in=2*nc
do 100 i=1,nc
lenl=lenl+z(is)
if (i.lt.nc and lenh.gt.lenl) goto 20
wgth=vc0(in)+(lenh-lenl+z(is))*z(is+1)
il=i
goto 110
20 continue
is=is-3
in=in-2
100 continue
110 continue
return
end
*

```

```

et sys final/i2for/cerisr for i1
subroutine CERISR(ce,z1,ybb)
*****
implicit double precision (a-z)

double precision ce,z1(25),ybb

integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
double precision z(67),cz,cx,d,ta,tb
common /VGL0B/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
& isol,ibrnch,uz
double precision za(25),zb(25)
equivalence (z(1),za(1)),(z(26),zb(1))
double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
& xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
equivalence (za(1),ha),(za(2),ala,va),
& (za(3),sla),(za(4),wla),(za(5),cla),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),x1a),(za(14),x2a),(za(15),x3a),
& (za(16),y1a),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
& xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
equivalence (zb(1),hb),(zb(2),alb,vb),
& (zb(3),slb),(zb(4),wlb),(zb(5),clb),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
& (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
& (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phih,rtot,xtot,ztot,do
equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phih),
& (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

```

```

integer*2 itant
double precision a,b,snphi,tnafa,tnafb,
&  seca7,seca8,ut,st,ykt,zkt,eex,eez,eeey,ybuoy
common /VCSHP/ a,b,snphi,tnafa,tnafb,
&  seca7,seca8,ut,st,ykt,zkt,eex,eez,eeey,ybuoy,itant
*****
tana7=zt(itant)*(ce+c3)/h
call TRISR
ybb=ykt+zt(12)+y4
return
end

```

\*

```

PROGRAM final/12for/secvit for!!
  subroutine SECVIT(ncf,zf,vc0f,ncc,zc,vc0c,
    & sinaf,cosaf,tanaf,secaf,htanaf,iftyp,ifail)
*****
    implicit integer*2 (*)
    implicit double precision (a-z)

    integer*2 ncf,ncc,iftyp,ifail
    double precision zf(25),vc0f(6),zc(25),vc0c(6),
    & sinaf,cosaf,tanaf,secaf,htanaf

    double precision pi,halfpi,degrad,raddeg,zero,one,half
    integer*2 izero,ione,ilwo
    common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
    & izero,ione,ilwo

    integer*2 ivs
    double precision v0,v1,v2,f0,f1,f2,f,eps
    common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps,ivs
    double precision varray(3),farray(3)
    equivalence (v0,varray),(f0,farray)

    integer*2 nit,nerr,nr

    equivalence (nerr,nr)
*****
    write(10,*) 'SECVIT',v0,f0,v1,f1,f2,f,iftyp
    *
    ifail=0
    nit=1

1000 continue
    v2=v1-f1*(v1-v0)/(f1-f0)
1010 continue
    call SUBVX(ncf,zf,vc0f,ncc,zc,vc0c,sinaf,cosaf,tanaf,secaf,
    & htanaf,2,iftyp,nerr)
    *
    write(10,*) 'CALC2',nit,v2,f2,nerr
    if (nerr eq 0) goto 1200
    nr=nerr-nerr/3
    v2=half*(v1+vc0f(nr)+htanaf)
    goto 1010

1200 continue

```

149

```
if (dabs(f2) lt eps) goto 2000
if (nit gt 50) goto 1900
call SHIFT(0,1)
call SHIFT(1,2)
nit=nit+1
goto 1000
```

```
1900 continue
      ifail=1
2000 continue
      return
      end
```

\*

149

```

et sys final/12for/stefab for++
  subroutine STEFAB
  ****
  implicit integer*2 (n)
  implicit double precision (a-z)

  integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
  double precision z(67),cz,cx,d,ta,tb
  common /VGL0B/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
  & isol,ibrnch,uz
  double precision za(25),zb(25)
  equivalence (z(1),za(1)),(z(26),zb(1))
  double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
  & xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
  & tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
  equivalence (za(1),ha),(za(2),ala,va),
  & (za(3),sla),(za(4),wla),(za(5),cla),
  & (za(6),s2a),(za(7),w2a),(za(8),c2a),
  & (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
  & (za(13),x1a),(za(14),x2a),(za(15),x3a),
  & (za(16),y1a),(za(17),y2a),(za(18),y3a),
  & (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
  & (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
  double precision hb,alb,vb,s1b,w1b,c1b,s2b,w2b,c2b,s3b,w3b,
  & xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
  & tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
  equivalence (zb(1),hb),(zb(2),alb,vb),
  & (zb(3),s1b),(zb(4),w1b),(zb(5),c1b),
  & (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
  & (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
  & (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
  & (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
  & (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
  & (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
  double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
  & h,phih,r1ot,xtot,z1ot,da
  equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
  & (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
  & (z(59),tana7),(z(60),tana8),(z(61),l),
  & (z(62),h),(z(63),phih),
  & (z(64),r1ot),(z(65),xtot),(z(66),z1ot),(z(67),da)
  integer*2 uz1,uz2,uz3,uz4,uz5
  equivalence (uz(1),uz1),(uz(2),uz2),(uz(3),uz3),(uz(4),uz4),

```

```
& (uz(5),uz5)
```

```
double precision pi,halfpi,degrad,raddeg,zero,one,half
integer*2 izero,ione,itwo
common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
& izero,ione,itwo
```

```
double precision delyk,twod,halfd,dsq
common /VANCH/ delyk,twod,halfd,dsq
```

```
double precision snphih,csphih,snafh,csafh,tnafh,scafh,dsnph
common /VHDIR/ snphih,csphih,snafh,csafh,tnafh,scafh,dsnph
```

```
double precision epsy,gamma,se
integer*2 ia,ib,ie
common /VCMPD/ epsy,gamma,se,ia,ib,ie
```

```
double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
& eex0,eez0,eeey0,a0,b0,phia0,phib0
integer*2 icase
common /VSP1D/ sa,sb,ca,cb,vc0a,vc0b,
& eex0,eez0,eeey0,a0,b0,phia0,phib0,
& icase
```

```
double precision a,b,snphi,tnafa,tnafb,
& seca7,seca8,ut,st,ykt,zkt,eex,eez,eeey,ybuoy
common /VCSSHP/ a,b,snphi,tnafa,tnafb,
& seca7,seca8,ut,st,ykt,zkt,eex,eez,eeey,ybuoy
```

```
integer*2 ivs
double precision v0,v1,v2,f0,f1,f2,f,eps
common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps,ivs
```

```
double precision xred
integer*2 isidf,nernr,nernb
common /VSTAB/ xred,isidf,nernr,nernb
```

```
integer*2 nit,ipoint,ifail,isyq,ism1
```

```
double precision jac(2,2)
equivalence (jac(1,1),j11),(jac(1,2),j12),(jac(2,1),j21),
& (jac(2,2),j22)
```

```

double precision zz(7)
equivalence
& (zz(1),asav,delamx,delanw,anew),
& (zz(2),aold,delo,chnge,anedge),
& (zz(3),bold,delb,chnge,bnedge),(zz(4),y1,dcoeff),(zz(5),y2),
& (zz(6),ysqo,y1x,aedge),(zz(7),y2x,bedge)
double precision delx(2)
equivalence (delo,delx)

equivalence (cosmx,scafz,halfdd,sl,bsav,delmx,delbm,delbnw,bnew),
& (csphn,hddcsp,xmid,templ,ysqsav,del),(temp,fact,ra1)

funl(arga,argb,argd)=arga*arga+dsq-argb*argb+argd*(snphn+snphn)
& *arga
*****
* Set constant terms
*****
epsysq=epsy*epsy
zp9=0.9d0

*****
* Set iteration switch for subroutine XSECV to zero
*****
ivs=0

*****
* Compute upper bounds for a,b, they might not be least upper bounds
*****
amx=one
bm=one
if (lnafh.le.zero) goto 120
cosmx=dmax1(csafh,one/SECNT(cz))
if (cz.lt.zero) amx=cosmx
if (cz.gt.zero) bm=cosmx
120 continue
amx=amx*sa
bm=bm*sb

*****
* Compute initial guess for (a,b) and assign to anew,bnew
* Set (a,b) equal to nearest asymptotic point
*****
scafz=SECNT(cz)

```

```

csphn=csafh*(snphn+cz*tnafh)/scafh
haldd=haldd*scafh
hddcsp=haldd*csphn
temp=hddcsp*hddcsp-haldd*haldd
s1=dmin1(dsqr1(sa*sa+temp)+hddcsp,dsqr1(sb*sb+temp)-hddcsp)

do 300 i=1,7
  zz(i)=z(i)
300 continue
ha=h
sla=s1
wla=(ca+cb)/s1
cla=c3
s2a=s4
w2a=w4
call VCRIT0(i,wo,za,vc0a)

f=do
call XSECV(2,za,vc0a,sla+s2a,vc0a(1),nwo,ncb,zb,vc0b,
& snafh,csafh,tnafh,scafh,2,ifa(1))

xmld=za(1)
temp=xmld*xmld+haldd*haldd
templ=xmld+dsnph
anew=dsqr1(temp-templ)
bnew=dsqr1(temp+templ)
a=half*(anew+bnew-dsnph)
b=a+dsnph

do 500 i=1,7
  z(i)=zz(i)
500 continue
call VCRIT0(nco,za,vc0a)

```

```

*****
* Beginning of code for Steffensen iteration
*****
* Test new point (a,b) for validity via subroutine CALC3 and adjust
* if necessary, while generating error vector (y1,y2)
* Point (a,b) lies within hyperbolic region, but value of a or b
* may be too large. Point (aold,bold) has passed test with CALC3
*****
n1=0

```

```

      1ysq=0
1000 continue
*      write(10,*)
*      write(10,*) 'ITER',nit
      isw1=0
1010 continue
      isidf=1
      call CALC3(a,b,y1,y2)
      ysq=y1*y1+y2*y2
      if (nerrr+nerrb eq 0 and (nit eq 0 or isw1 eq 1
& or ysq lt ysqo*1.1d0)) goto 1200
      if (nit eq 0 or isw1 eq 1) goto 1100
      a=half*(a+aold)
      b=half*(b+bold)
      goto 1010
1100 continue
      temp=dmax1(xred+xred, half*(half*d*(one-dsnph)-a))
      a=a+temp
      b=b+temp
      goto 1010

1200 continue
      if (nit eq 0) goto 2400
      if (isw1 eq 0 or ysq lt ysqsav) goto 1250
      a=asav
      b=bsav
      ysq=ysqsav
      isw1=0
      goto 1300
1250 continue
      if (isw1 eq 0 and ysq ge zp9*ysqo1 1ysq=1ysq+1

1300 continue
      if (1ysq le 4) goto 1400
      asav=a
      bsav=b
      ysqsav=ysq
      a=half*(a+b-dsnph)
      b=a+dsnph
      isw1=1
      1ysq=0
      goto 1010

```

```

*****
* Finished if error vector is sufficiently small or current test point
* is sufficiently close to previous test point
*****
1400 continue
      if (ysq lt epsysq) goto 5000
      if (ysq lt epsysq*1 0d8 and
          & dabs(one-a/aold)+dabs(one-b/bold) lt 1 0d-8) goto 5000

*****
* Compute deltas for Jacobian matrix estimate
*****
      if (nit eq 1 or iswl eq 1) goto 1500
      dela=(j11*y1+j12*y2)*dsqrt(j21*j21+j22*j22)/de1j
      delb=(j21*y1+j22*y2)*dsqrt(j11*j11+j12*j12)/de1j
      goto 2000
1500 continue
      dela=dsqrt(half*ysq)
      delb=dela

*****
* Adjust deltas as necessary
*****
2000 continue
* write(10,*) 'init del',dela,delb
      rat=dmax1(one,epsy/dabs(dela),epsy/dabs(delb))
      dela=rat*dela
      delb=rat*delb
      delmx=dmin1(dabs(a-b+d),dabs(b-a+d),a+b-d)
      delamx=dmin1(amx-a,delmx)
      delbmxdmin1(bmx-b,delmx)
      rat=dmin1(one,delamx/dabs(dela),delbmxdabs(delb))
      if (rat ne one) rat=0 1d0*rat
      dela=dela*rat
      delb=delb*rat

2005 continue
      delonw=dela
      if (fun1(a+delonw,b,d) gt zero) goto 2110
      call EDCPT(a,b,a+delonw,b,d,aedge,bedge)
      delonw=aedge-a
2110 continue
      if (fun1(b,a+delonw,-d) gt zero) goto 2120

```

```

call EDGPT(b,a,b,a+delanw,-d,bedge,aedge)
delanw=aedge-a
2120 continue
delbnw=delb
if (fun1(a,b+delbnw,d) gt zero) goto 2130
call EDGPT(a,b,a,b+delbnw,d,aedge,bedge)
delbnw=bedge-b
2130 continue
if (fun1(b+delbnw,a,-d) gt zero) goto 2140
call EDGPT(b,a,b+delbnw,a,-d,bedge,aedge)
delbnw=bedge-b
2140 continue

rat=one
if (dela eq delanw and delb eq delbnw) goto 2160
rat=dmin1(dabs(delanw/dela),dabs(delbnw/delb))*0.1d0
2160 continue
dela=rat*dela
delb=rat*delb
* write(10,*) 'fin del',dela,delb,rat

* if (fun1(a+dela,b,d) gt zero and fun1(b,a+dela,-d) gt zero
* & and fun1(a,b+delb,d) gt zero and fun1(b+delb,a,-d) gt zero)
* & goto 2190
* dela=half*dela
* delb=half*delb
* goto 2005
*2190 continue

*****
* Estimate Jacobian matrix
*****
2200 continue
ipoint=1
2205 continue
del=delx(ipoint)
isidf=1
if (del lt zero) isidf=2
if (ipoint eq 2) isidf=3-isidf
call CALC3(a+(2-ipoint)*dela,b+(ipoint-1)*delb,y1x,y2x)
if (nerra+nerrb eq zero) goto 2220
dela=half*dela
delb=half*delb

```

```

      goto 2200
2220 continue
      jac(1,ipoint)=(y1x-y1)/del
      jac(2,ipoint)=(y2x-y2)/del
      ipoint=ipoint+1
      if (ipoint.le.2) goto 2205
*      write(10,*) 'jac',j11,j12
*      write(10,*) '      ',j21,j22

*****
* Invert Jacobian matrix and compute new point (anew,bnew)
*****
      detj=j11*j22-j12*j21
      temp=j11
      j11=j22/detj
      j22=temp/detj
      j12=-j12/detj
      j21=-j21/detj
      anew=a-(j11*y1+j12*y2)
      bnew=b-(j21*y1+j22*y2)
*      write(10,*) 'inv',j11,j12
*      write(10,*) '      ',j21,j22

*****
* Adjust new point (anew,bnew) as necessary
*****
2400 continue
*      write(10,*) 'init new pt',anew,bnew
      fact=one
      if (dabs(anew-bnew) lt d) goto 2500
      fact=zp9
      dcoeff=d
      if (anew gt bnew) dcoeff=-d
      chnga=anew-a
      chngb=bnew-b
      anew=(chnga*(b-dcoeff)-chngb*a)/(chnga-chngb)
      bnew=anew+dcoeff
2500 continue
*      write(10,*) anew,bnew
      if (anew+bnew gt d) goto 2600
      fact=zp9
      chnga=anew-a
      chngb=bnew-b

```

```

      anew=(chnga*(d-b)+chngb*a)/(chnga+chngb)
      bnew=d-anew
2600  continue
      * write(10,*) anew,bnew
      if (funl(anew,bnew,d) gt zero) goto 2800
      fact=zp9
      call EDCPT(a,b,anew,bnew,d,anedge,bnedge)
      anew=anedge
      bnew=bnedge
2800  continue
      * write(10,*) anew,bnew
      if (funl(bnew,anew,-d) gt zero) goto 2900
      fact=zp9
      call EDCPT(b,a,bnew,anew,-d,bnedge,anedge)
      anew=anedge
      bnew=bnedge
2900  continue
      * write(10,*) anew,bnew
      anew=a+fact*(anew-a)
      bnew=b+fact*(bnew-b)
      * write(10,*) 'fin new pt ',anew,bnew,fact
      * if (funl(anew,bnew,d) gt zero and funl(bnew,anew,-d) gt zero)
      * & goto 3000
      * anew=half*(a+anew)
      * bnew=half*(b+bnew)
      * goto 2600
3000  continue

*****
* Shift values and return to beginning of Steffensen iteration
*****
      aold=a
      bold=b
      ysqa=ysq
      a=anew
      b=bnew
      nit=nit+1
      goto 1000

5000  continue
      return
      end

```

```

e1 sys final/i2for/calc3 for##
  subroutine CALC3(a,b,y1,y2)
*****
  implicit integer*2 (*)
  implicit double precision (a-z)

  double precision a,b,y1,y2

  integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
  double precision z(67),cz,cx,d,ta,tb
  common /VGL08/ ileg,ist,nca,ncb,z,cz,cx,d,ta,tb,nwa,nwb,
& isol,ibrnch,uz
  double precision za(25),zb(25)
  equivalence (z(1),za(1)),(z(26),zb(1))
  double precision ha,ola,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
& xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
  equivalence (za(1),ha),(za(2),ola,va),
& (za(3),sla),(za(4),wla),(za(5),cla),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),x1a),(za(14),x2a),(za(15),x3a),
& (za(16),y1a),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
  double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
& xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
  equivalence (zb(1),hb),(zb(2),alb,vb),
& (zb(3),slb),(zb(4),wlb),(zb(5),clb),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
& (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
& (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
  double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phih,rtot,xtot,ztot,do
  equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phih),
& (z(64),rtot),(z(65),xtot),(z(66),ztot),(z(67),do)

```

```

double precision pi,halfpi,degrad,raddeg,zero,one,half
integer*2 izero,ione,itiwo
common /VCONST/ pi,halfpi,degrad,raddeg,zero,one,half,
& izero,ione,itiwo

double precision delyk,twod,halfd,dsq
common /VANCH/ delyk,twod,halfd,dsq

double precision sa,sb,ca,cb,vc0a(6),vc0b(6),
& eex0,eez0,eeey0,a0,b0,phia0,phib0
integer*2 icas
common /VSPID/ sa,sb,ca,cb,vc0a,vc0b,
& eex0,eez0,eeey0,a0,b0,phia0,phib0,
& icas

integer*2 iscopa,iscopb,itanb,itanb,ie
double precision epsy,gamma,se
common /VCPMD/ epsy,gamma,se,iscopa,iscopb,itanb,itanb,ie

double precision qa,qb,snphi,tnafa,tnafb,
& seca7,seca8,ut,st,ykt,zkt,eex,eez,eeey,ybuoy
common /VCSSHP/ qa,qb,snphi,tnafa,tnafb,
& seca7,seca8,ut,st,ykt,zkt,eex,eez,eeey,ybuoy

double precision v0,v1,v2,f0,f1,f2,f,eps
common /VSEC/ v0,v1,v2,f0,f1,f2,f,eps

double precision xred
integer*2 isidf,nerra,nerrb
common /VSTAB/ xred,isidf,nerra,nerrb

integer*2 nsid

equivalence (sca,scb,iana7)
*****
* write(10,*) 'CALC3',a,b
nerra=0
nerrb=0
call PHIAB(a,b,a*a-b*b,dsq,twod,phia,phib)
call HSPLIT
if (ha gt zero and hb gt zero) goto 900
nerra=1

```

```

      nerrb=1
      goto 5000
900  continue
      nsid=1
1000 continue
      goto (1100,1200),nsidf
1100 continue
      sca=SECNT(inafa)
      f=a
      call XSECV(nca,za,vc0a,sa,ca,nwa,ncb,zb,vc0b,
& inafa/sca,one/sca,inafa,sca,1,nerra)
      if (nerra eq 0) goto 2000
      goto 5000
1200 continue
      scb=SECNT(inafb)
      f=b
      call XSECV(ncb,zb,vc0b,sb,cb,nwb,nca,za,vc0a,
& inafb/scb,one/scb,inafb,scb,1,nerrb)
      if (nerrb eq 0) goto 2000
      goto 5000
2000 continue
      if (nsid eq 2) goto 2200
      nsid=2
      isidf=3-isidf
      goto 1000
2200 continue
      iana7=(c3+ha*za((1:iana)+hb*zbl(1:ianb)))/h
      call TRISR
      y1=half*(ya+yb)+y4-do
      y2=yb-ya-delyh
      call X4CALC
5000 continue
*   write(10,*) 'END CALC3',nerra,nerrb,y1,y2
*   write(10,*) va,ya,vb,yb
*   write(10,*)
      return
      end
*
```

162

```

et sys final/12for/edgpt for##
  subroutine EDGPT(a,b,aa,bb,d,x,y)
*****
  implicit double precision (a-z)

  double precision a,b,aa,bb,d,x,y

  integer*2 ileg,ist,nca,ncb,nwa,nwb,isol,ibrnch,uz(5)
  double precision z(67),cz,cx,qd,ta,tb
  common /VGL08/ ileg,ist,nca,ncb,z,cz,cx,qd,ta,tb,nwa,nwb,
& isol,ibrnch,uz
  double precision za(25),zb(25)
  equivalence (z(1),za(1)),(z(26),zb(1))
  double precision ha,ala,va,sla,wla,cla,s2a,w2a,c2a,s3a,w3a,
& xa,ya,x1a,x2a,x3a,y1a,y2a,y3a,
& tana2a,tana3a,tana4a,tana5a,tana6a,la,phia
  equivalence (za(1),ha),(za(2),ala,va),
& (za(3),sla),(za(4),wla),(za(5),cla),
& (za(6),s2a),(za(7),w2a),(za(8),c2a),
& (za(9),s3a),(za(10),w3a),(za(11),xa),(za(12),ya),
& (za(13),x1a),(za(14),x2a),(za(15),x3a),
& (za(16),y1a),(za(17),y2a),(za(18),y3a),
& (za(19),tana2a),(za(20),tana3a),(za(21),tana4a),
& (za(22),tana5a),(za(23),tana6a),(za(24),la),(za(25),phia)
  double precision hb,alb,vb,slb,wlb,clb,s2b,w2b,c2b,s3b,w3b,
& xb,yb,x1b,x2b,x3b,y1b,y2b,y3b,
& tana2b,tana3b,tana4b,tana5b,tana6b,lb,phib
  equivalence (zb(1),hb),(zb(2),alb,vb),
& (zb(3),slb),(zb(4),wlb),(zb(5),clb),
& (zb(6),s2b),(zb(7),w2b),(zb(8),c2b),
& (zb(9),s3b),(zb(10),w3b),(zb(11),xb),(zb(12),yb),
& (zb(13),x1b),(zb(14),x2b),(zb(15),x3b),
& (zb(16),y1b),(zb(17),y2b),(zb(18),y3b),
& (zb(19),tana2b),(zb(20),tana3b),(zb(21),tana4b),
& (zb(22),tana5b),(zb(23),tana6b),(zb(24),lb),(zb(25),phib)
  double precision coil,slp,frct,c3,s4,w4,x4,y4,tana7,tana8,l,
& h,phih,rtoi,xtoi,ztoi,do
  equivalence (z(51),coil),(z(52),slp),(z(53),frct),(z(54),c3),
& (z(55),s4),(z(56),w4),(z(57),x4),(z(58),y4),
& (z(59),tana7),(z(60),tana8),(z(61),l),
& (z(62),h),(z(63),phih),
& (z(64),rtoi),(z(65),xtoi),(z(66),ztoi),(z(67),do)

```

```

double precision pi, halfpi, degrad, raddeg, zero, one, half
integer*2 1zero, 1one, 1two
common /VCONST/ pi, halfpi, degrad, raddeg, zero, one, half,
& 1zero, 1one, 1two

```

```

double precision delyk, twod, halfd, dsq
common /VANCH/ delyk, twod, halfd, dsq

```

```

double precision snphih, csphih, snafh, csafh, tnafh, scafh
common /VDIR/ snphih, csphih, snafh, csafh, tnafh, scafh

```

```

integer*2 1scopb, 1scopb, 1tana, 1tanb, 1e
double precision epsy, gamma, se
common /VCPD/ epsy, gamma, se, 1scopb, 1scopb, 1tana, 1tanb, 1e

```

```

integer*2 1tani
double precision qa, qb, snphi, tnafa, tnafb,
& seca7, seca8, ut, st, ykt, zkt, eex, eez, eey, ybuoy
common /VCSSHP/ qa, qb, snphi, tnafa, tnafb,
& seca7, seca8, ut, st, ykt, zkt, eex, eez, eey, ybuoy, 1tani

```

```

equivalence (dela, ta), (delb, root1, tb), (delasq, coeff2, temp, gamma),
& (cp, coeff1, eex), (coeff0, discr, eez), (slope, eey), (dsnph, rtor)
*****
dsnph=dsnphih
if (a ne aa) goto 100
x=a
y=dsqrt(a*(a+dsnph+dsnph)+dsq)
goto 1000
100 continue
dela=aa-a
delasq=dela*dela
delb=bb-b
cp=a*bb-aa*b
coeff0=dsq*delasq-cp*cp
coeff1=cp*delb+dsnph*delasq
slope=delb/dela
if (debs1one-debs1slope) gt 10d-6) goto 200
x=-coeff0/(coeff1+coeff1)
goto 500
200 continue
coeff2=delasq-delb*delb
discr=dsqrt(coeff1*coeff1-coeff2*coeff0)

```

```
root1=(-coeff1-discr)/coeff2
x=(-coeff1+discr)/coeff2
if (root1 lt x) goto 220
temp=x
x=root1
root1=temp
220 continue
if ((a lt aa and root1 gt a) or (aa lt a and x ge a))
& x=root1
500 continue
if ((x-a)*(x-aa) ge zero) x=aa
y=b+(x-a)*slope
1000 continue
return
end
```

\*

END

DATE  
FILMED

5-83

DTI