```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c     program to analyze SB1 RV curve
c     Expects pairs of observational points from program sporb.f
c     using pgplot routines
c     orbit plots
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      program anorb
      character*60 xlabel,ylabel,title,outdev
      character*40 infile
      character*1 dum,ans
      real x(801),y(801),ti(801),xo(2 000),yo(2 000),K
      Pi = 3.141592653589793
c           12345678901234567890123456789012345678901234567
      title='Spectroscopic Binary - e =    , \\gw\\fr =    '
      xlabel='time - Period =     '
      ylabel='Radial Velocity - K =    , Gamma Vel =    '
      write(6,*)' outdev set to /xw for present'
      outdev = '/xw'
      write(6,*)' Enter file with observations'
      read(5,'(a40)')infile
      open(unit=2,file=infile)
1     continue
      rewind 2
      write(6,*)' Enter your guess for the gamma-velocity (q to quit)'
      read(5,*,err=999)V0
      write(6,*)' Enter your guess for the period'
      read(5,*)p
      write(6,*)' Enter your guess for the parameter K'
      read(5,*)K
      write(xlabel(16:19), '(f4.1)')p
      write(ylabel(22:26), '(f4.0)')K
      write(ylabel(41:44), '(f4.0)')V0
c-----read data from infile--skip one header record
      read(2,'(a1)')dum
      nobs = 1
      do 210 i=1,100000
      read(2,*,end=215)xo(nobs),yo(nobs)
c-----rescale the observations to fit K = 1, V0 = 0.0
c     xo(nobs) = xo(nobs)/P
      yo(nobs) = (yo(nobs)-V0)/K
      nobs = nobs+1
210   continue
```

```
215   nobs = nobs-1
c-----fold observations onto guessed period-and normalize by P
      do 310 i=1,nobs
       xxdum = xo(i)
       call pout(xxdum,P)
       xo(i) = xxdum/P
310   continue
      if(nobs.gt.900)then
217     write(6,*)' Increase dimensions for data points'
       stop
      end if
      write(6,*)' Number of observational points is: ',nobs
      write(6,*)' Double observations to cover 2*P? (y/n)'
      read(5,'(a1)')ans
      if((ans.eq.'y').or.(ans.eq.'Y')) then
c-----For purposes of display, double the observations to cover 2*P
       if(nobs.gt.450)go to 217
       do 340 i=nobs+1,2*nobs
        xo(i) = xo(i-nobs)+1.0
        yo(i) = yo(i-nobs)
340      continue
       nobs = 2*nobs
      end if
      write(6,*)' Enter e and omega in degrees -- negative e to quit'
      read(5,*)e,w
      if(e.lt.0.0)stop
      write(title(27:30),'(f4.1)')e
      write(title(43:46),'(f4.0)')w
c-----convert to radians
      w = 0.017453293*w
c-----Call the values of M, x
c-----get 200 additional points at equal intervals of M
      E0 = 0.0
      step = 2.0*Pi/400.0
      do  10 i=1,801
        t = (step)*float(i-1)
        ti(i) = t/(2.0*Pi)
        if(t.gt.2.0*Pi)t = t-2.0*Pi
        if((abs(E0).gt.10.0*step).and.(t.lt.10.0*step))E0 = step
        call kepler(t,E0,e,EE)
        call vfore(EE,e,v)
        x(i) = v
c       write(3,60)i,t,E0,EE,v
c-----   Guess the new E will be the old one plus the same INCREMENT
```

```
c-----   as that of M.
         E0 = EE+(2.0*Pi/400.0)
10     continue
c      close(unit=3)
60     format(i4,4f12.8)
c-----Get the y's.  Use Aitken p. 152 Eq. 1 with mu*a*sin(i) = 1.0
       do 11 i=1,801
        y(i) = (e*cos(w)+cos(x(i)+w))
11     continue
       lastj = 801
20     call DRAWIT(ti,Y,LASTJ,xo,yo,nobs,xlabel,ylabel,TITLE,OUTDEV)
c      pause
       go to 1
999    stop
       end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c     subroutine pout -- subtract multiples of period from x
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
       subroutine pout(x,p)
       fac = x/p
       ifac = fac
       x = x-float(ifac)*p
       return
       end


C**********************************************************************
C
C     SUBROUTINE DRAWIT: TO graph data
C     PROFILES
C**********************************************************************
       SUBROUTINE DRAWIT(X,Y,LASTJ,xo,yo,nobs,xlabel,ylabel,TITLE,OUTDEV)
       LOGICAL PLTOBS,hardcopy
       REAL X(801),Y(801),xo(2 000),yo(2 000)
       character*60 xlabel,ylabel
       CHARACTER*60 TITLE,OUTDEV
       CHARACTER*20 JUNK
c-----default option is no hardcopy
       hardcopy = .false.
c-----get max and min value of y, and x for plot scaling
       rmaxx = x(1)
       rminx = x(1)
```

```
      rmaxy = y(1)
      rminy = y(1)
      do 8 kk=2,lastj
        if(x(kk).lt.rminx)rminx = x(kk)
        if(x(kk).gt.rmaxx)rmaxx = x(kk)
        if(y(kk).lt.rminy)rminy = y(kk)
        if(y(kk).gt.rmaxy)rmaxy = y(kk)
8     continue
c-----Make sure the observations will fit on the plot
      do 800 kk = 1,nobs
        if(yo(kk).gt.rmaxx)rmaxy = yo(kk)
        if(yo(kk).lt.rminy)rminy = yo(kk)
800   continue
      rhiy = rmaxy+0.1*rmaxy
      rhix = rmaxx+0.1*rmaxx
      if(rminy.lt.0.0)then
        rloy = rminy - 0.1*abs(rminy)
      else
        rloy = rminy-0.05*abs(rmaxy)
      end if
      if(rminx.lt.0.0)then
        rlox = rminx - 0.1*abs(rminx)
      else
        rlox = rminx - 0.05*abs(rmaxx)
      end if
c     re-entry point if hardcopy desired
11    continue
      CALL PGBEGIN(0,OUTDEV,1,1)
      call pgscf(2)
      CALL PGSCH(1.7)
c       CALL PGENV(0.0,327.0,23.75,32.2,0,0)
      call pgenv(rlox,rhix,rloy,rhiy,0,0)
c       call pgenv(0.0,1100.0,0.0,1100.0,0,0)
      CALL PGLABEL(xlabel,ylabel,TITLE)
      CALL PGLINE(LASTJ,X,Y)
c       call pgtext(3.5,1.3,'S II Ratio')
       call pgpoint(nobs,xo,yo,2)
c       call pgpoint(1,0.59,2.34,10)
20    FORMAT(A20)
      CALL PGEND
c-----branch around this block if hardcopy has already been made
      if(hardcopy)go to 45
      write(6,*)' Hard copy? (y/n)'
      read(5,'(a1)')ans
```

```
      if(ans.eq.'y'.or.ans.eq.'Y')then
        hardcopy = .true.
        write(6,*)' Enter name of output file with /ps'
        read(5,'(a60)')outdev
        go to 11
      end if
45    continue
      RETURN
      END
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c     subroutine kepler
c     to solve kepler's equation with input m, E0, and e
c     EE is eccentric anomaly, E0 its first approximation
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine kepler(m,E0,e,EE)
      real m, m0
      if(m.eq.0)then
        EE = 0
        return
      end if
      kount =  0
10    M0 = E0 - e*sin(E0)
      DM = M - M0
      DE = DM/(1.0 -e*cos(E0))
      EE = E0 + DE
c     write(6,*)DM,DE,EE
c-----check for 0.1% accuracy
      cm = EE - e*sin(EE)
      test = abs((cm-m)/m)
c     write(6,*)' test = ',test
      if(test.gt.0.001)then
c       write(6,*)' E0 = ',E0, ' kount = ',kount
        E0 = EE
        kount = kount + 1
        if(kount.gt.10)then
          write(6,*)' 10 iterations no convergence'
c         write(6,*)' M = ',M
c         write(6,*)' E0 = ',E0
c         write(6,*)' cm = ',cm
          pause
        end if
        go to 10
```

```
      end if
      return
      end
ccccccccccccccccccccccccccccccccccccccccc
c
c     subroutine vfore
c     v from EE for orbit problems
c
ccccccccccccccccccccccccccccccccccccccccc
      subroutine vfore(EE,e,vc)
      logical pstap
      Pi = 3.141592653589793
      pstap = .false.
c-----reflect about 180=2Pi
      if((EE.gt.Pi).and.(EE.lt.2.0*Pi)) then
        pstap = .true.
        EEp = 2.0*Pi-EE
      else
        EEp = EE
      end if

      vc = acos((cos(EEp)-e)/(1-e*cos(EEp)))
c       vt = 2.0*atan(sqrt((1.0+e)/(1.0-e))*tan(EE/2.0))
c       y = tan(x)
c       arc = atan(y)
      if(pstap)then
        vc = 2.0*Pi - vc
c         vt = 2.0*Pi - vt
      end if
c       write(6,*)' vc = ',vc
c       write(6,*)' vt = ',vt
      return
      end
```

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c     program to do x-y plots of orbit for spectroscopic bianry
c     User enters inclination and omega.  Option for hardcopy.
c     using pgplot routines
c     orbit plots
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      program plotit
```

```
      character*60 xlabel,ylabel,title,outdev
c     character*40 infile
      real x(801),y(801),ti(801)
c     read(2,'(a60)')title
c     read(2,'(a60)')xlabel
c     read(2,'(a60)')ylabel
      Pi = 3.141592653589793
c           12345678901234567890123456789012345678901234567
      title='Spectroscopic Binary - e =    , \\gw\\fr =    '
      xlabel='time'
      ylabel='Radial Velocity'
      write(6,*)' outdev set to /xw for present'
      outdev = '/xw'
1     write(6,*)' Enter eccentricity - negative to quit'
      read(5,*)e
      if(e.lt.0.0)stop
c     open(unit=3,file='outfil')
      write(6,*)' Enter little omega in degrees'
      read(5,*)w
      write(title(27:30),'(f4.1)')e
      write(title(43:46),'(f4.0)')w
c-----convert to radians
      w = 0.017453293*w
c-----Call the values of M, x
c-----get 200 additional points at equal intervals of M
      E0 = 0.0
      step = 2.0*Pi/400.0
      do  10 i=1,801
        t = (step)*float(i-1)
        ti(i) = t/(2.0*Pi)
        if(t.gt.2.0*Pi)t = t-2.0*Pi
        if((abs(E0).gt.10.0*step).and.(t.lt.10.0*step))E0 = step
        call kepler(t,E0,e,EE)
        call vfore(EE,e,v)
        x(i) = v
c       write(3,60)i,t,E0,EE,v
c-----   Guess the new E will be the old one plus the same INCREMENT
c-----   as that of M.
        E0 = EE+(2.0*Pi/400.0)
10    continue
c     close(unit=3)
60    format(i4,4f12.8)
c-----Get the y's.  Use Aitken p. 152 Eq. 1 with mu*a*sin(i) = 1.0
      do 11 i=1,801
```

```
       y(i) = (e*cos(w)+cos(x(i)+w))
11    continue
      lastj = 801
20    call DRAWIT(ti,Y,LASTJ,xlabel,ylabel,TITLE,OUTDEV)
c     pause
      go to 1
      end


C*********************************************************************
C
C     SUBROUTINE DRAWIT: TO graph data
C     PROFILES
C*********************************************************************
      SUBROUTINE DRAWIT(X,Y,LASTJ,xlabel,ylabel,TITLE,OUTDEV)
      LOGICAL PLTOBS,hardcopy
      REAL X(801),Y(801)
      character*60 xlabel,ylabel
      CHARACTER*60 TITLE,OUTDEV
      CHARACTER*20 JUNK
      character*1 ans
c-----default option is no hardcopy
      hardcopy = .false.
c-----get max and min value of y, and x for plot scaling
      rmaxx = x(1)
      rminx = x(1)
      rmaxy = y(1)
      rminy = y(1)
      do 8 kk=2,lastj
        if(x(kk).lt.rminx)rminx = x(kk)
        if(x(kk).gt.rmaxx)rmaxx = x(kk)
        if(y(kk).lt.rminy)rminy = y(kk)
        if(y(kk).gt.rmaxy)rmaxy = y(kk)
8     continue
      rhiy = rmaxy+0.1*rmaxy
      rhix = rmaxx+0.1*rmaxx
      if(rminy.lt.0.0)then
        rloy = rminy - 0.1*abs(rminy)
      else
        rloy = rminy-0.05*abs(rmaxy)
      end if
      if(rminx.lt.0.0)then
        rlox = rminx - 0.1*abs(rminx)
      else
```

```
        rlox = rminx - 0.05*abs(rmaxx)
      end if
c     re-entry point if hardcopy desired
11    continue
      CALL PGBEGIN(0,OUTDEV,1,1)
      call pgscf(2)
      CALL PGSCH(1.7)
c       CALL PGENV(0.0,327.0,23.75,32.2,0,0)
      call pgenv(rlox,rhix,rloy,rhiy,0,0)
c       call pgenv(0.0,1100.0,0.0,1100.0,0,0)
      CALL PGLABEL(xlabel,ylabel,TITLE)
      CALL PGLINE(LASTJ,X,Y)
c       call pgtext(3.5,1.3,'S II Ratio')
c       call pgpoint(lastj,x,y,1)
c       call pgpoint(1,0.59,2.34,10)
20    FORMAT(A20)
      CALL PGEND
c-----branch around this block if hardcopy has already been made
      if(hardcopy)go to 45
      write(6,*)' Hard copy? (y/n)'
      read(5,'(a1)')ans
      if(ans.eq.'y'.or.ans.eq.'Y')then
        hardcopy = .true.
        write(6,*)' Enter name of output file with /ps'
        read(5,'(a60)')outdev
        go to 11
      end if
45    continue
      RETURN
      END
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c     subroutine kepler
c     to solve kepler's equation with input m, E0, and e
c     EE is eccentric anomaly, E0 its first approximation
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine kepler(m,E0,e,EE)
      real m, m0
      if(m.eq.0)then
        EE = 0
        return
      end if
      kount =  0
```

```
10    M0 = E0 - e*sin(E0)
      DM = M - M0
      DE = DM/(1.0 -e*cos(E0))
      EE = E0 + DE
c     write(6,*)DM,DE,EE
c-----check for 0.1% accuracy
      cm = EE - e*sin(EE)
      test = abs((cm-m)/m)
c     write(6,*)' test = ',test
      if(test.gt.0.001)then
        write(6,*)' E0 = ',E0, ' kount = ',kount
       E0 = EE
       kount = kount + 1
       if(kount.gt.10)then
         write(6,*)' 10 iterations no convergence'
         write(6,*)' M = ',M
         write(6,*)' E0 = ',E0
         write(6,*)' cm = ',cm
         pause
       end if
       go to 10
      end if
      return
      end
ccccccccccccccccccccccccccccccccccccccccc
c
c     subroutine vfore
c     v from EE for orbit problems
c
ccccccccccccccccccccccccccccccccccccccccc
      subroutine vfore(EE,e,vc)
      logical pstap
      Pi = 3.141592653589793
      pstap = .false.
c-----reflect about 180=2Pi
      if((EE.gt.Pi).and.(EE.lt.2.0*Pi)) then
        pstap = .true.
        EEp = 2.0*Pi-EE
      else
        EEp = EE
      end if

      vc = acos((cos(EEp)-e)/(1-e*cos(EEp)))
c     vt = 2.0*atan(sqrt((1.0+e)/(1.0-e))*tan(EE/2.0))
```

```
c      y = tan(x)
c      arc = atan(y)
       if(pstap)then
         vc = 2.0*Pi - vc
c          vt = 2.0*Pi - vt
       end if
c      write(6,*)' vc = ',vc
c      write(6,*)' vt = ',vt
       return
       end
```

```
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc c
c      program to do x-y plots -- Modified to take out period
c      using pgplot routines   -- for Spectroscopic Binary
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
       program ploto
       character*60 xlabel,ylabel,title,outdev
       character*1 junk
       character*40 infile
       real x(10 000),y(10 000)
       write(6,*)' Enter name of file containing data'
       READ(5,'(a40)')infile
       open(unit=2,file=infile,status='old')
c      read(2,'(a60)')title
c      read(2,'(a60)')xlabel
c      read(2,'(a60)')ylabel
c-----or read period estimate
       write(6,*)' Enter an estimate for the period'
       read(5,*)period
88     continue
       rewind 2
c-----skip comment line-
       read(2,'(a1)')junk
       title = 'Radial Velocity Plot with Period         '
       write(title(37:45), '(f8.5)')period
       xlabel='Time'
       ylabel='Radial Velocity'
       write(6,*)' outdev set to /xw for present'
       outdev = '/xw'
c      write(6,*)' Enter integer for plotting, e.g.'
```

```
c     write(6,*)' a 5 would mean plot every 5th point'
c     read(5,*)nthplot
      nthplot=1
      m = 1
      do 10 kk=1,100000
           read(2,*,end=11)xxdum,yydum
        if((kk/nthplot)*nthplot.eq.kk)then
          call pout(xxdum,period)
             x(m) = xxdum
          y(m) = yydum
          m = m+1
        end if
10    continue
11    continue
      lastj = m-1
20    call DRAWIT(X,Y,LASTJ,xlabel,ylabel,TITLE,OUTDEV)
      write(6,*)' Enter another guess at the period, or'
      write(6,*)' negative number to quit'
      read(5,*)period
      if(period.lt.0)stop
      go to 88
      end
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
c
c     subroutine pout -- subtract multiples of period from x
c
ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc
      subroutine pout(x,p)
      fac = x/p
      ifac = fac
      x = x-float(ifac)*p
      return
      end

C**********************************************************************
C
C     SUBROUTINE DRAWIT: TO graph data
C     PROFILES
C**********************************************************************
      SUBROUTINE DRAWIT(X,Y,LASTJ,xlabel,ylabel,TITLE,OUTDEV)
      LOGICAL PLTOBS,hardcopy
      REAL X(10 000),Y(10 000)
      character*60 xlabel,ylabel
      CHARACTER*60 TITLE,OUTDEV
```

```
      CHARACTER*20 JUNK
c-----default option is no hardcopy
      hardcopy = .false.
c-----get max and min value of y, and x for plot scaling
      rmaxx = x(1)
      rminx = x(1)
      rmaxy = y(1)
      rminy = y(1)
      do 8 kk=2,lastj
        if(x(kk).lt.rminx)rminx = x(kk)
        if(x(kk).gt.rmaxx)rmaxx = x(kk)
        if(y(kk).lt.rminy)rminy = y(kk)
        if(y(kk).gt.rmaxy)rmaxy = y(kk)
8     continue
      rhiy = rmaxy+0.1*abs(rmaxy-rminy)
      rhix = rmaxx+0.1*abs(rmaxx-rminx)
      rloy = rminy - 0.1*abs(rminy-rmaxy)
      rlox = rminx - 0.1*abs(rmaxx-rminx)
      write(6,*)' min and maxx',rminx,rmaxx
c     re-entry point if hardcopy desired
11    continue
      CALL PGBEGIN(0,OUTDEV,1,1)
      call pgscf(2)
      CALL PGSCH(1.7)
      call pgenv(rlox,rhix,rloy,rhiy,0,0)
c       call pgenv(0.0,1100.0,0.0,1100.0,0,0)
      CALL PGLABEL(xlabel,ylabel,TITLE)
c     CALL PGLINE(LASTJ,X,Y)
c     call pgtext(3.5,1.3,'S II Ratio')
       call pgpoint(lastj,x,y,12)
20    FORMAT(A20)
      CALL PGEND
c-----branch around this block if hardcopy has already been made
      if(hardcopy)go to 45
      write(6,*)' Hard copy? (y/n)'
      read(5,'(a1)')ans
      if(ans.eq.'y'.or.ans.eq.'Y')then
        hardcopy = .true.
        write(6,*)' Enter name of output file with /ps'
        read(5,'(a60)')outdev
        go to 11
      end if
45    continue
      RETURN
```

END