```
      program trinve
*************************************************************************
*     this program inverts transmission data of a
*     substrate. On input it wants to know precisely the substrate optical
*     constant, and the thickness.
*     The program samples segments of one period of Fabry-Perod oscillations
*     and fits the plan-parallel-substrate formula to this.
*************************************************************************
      parameter(num=100000,nn=1000)
      real x(num),y(num),ylo,yhi,dsub,p
     * ,xmn(nn),xmx(nn),ymn(nn),ymx(nn),xav(nn),ymin,ymax,phi
     * ,a,b,c,dp,dm,pimp,pi
     * ,xx(nn),yy(nn),sig(nn),ysel
      complex epssub(nn)
      integer i,i1,i2,i3,i4,imx,ihi(num),ilo(num),nmx,nperio,n,itl,ntl
      character*40 flin,flout
      pi=4.*atan(1.)
      write(*,*) 'give inputfilename '
      read(*,'(a40)') flin
      open(14,file=flin)
      write(*,*) 'give outputfilename '
      read(*,'(a40)') flout
      write(*,*) 'how many measurement points per period ? '
      read(*,*) nperio
      write(*,*) 'substrate thickness in cm ? '
      read(*,*) dsub
      open(23,file=flout)
      open(28,file='ppqq.out')
      do 100 i=1,num
       read(14,*,END=101) x(i),y(i)
       y(i)=1/y(i)
100   continue
101   imx=i-1
      close(14)
      n=1
      i1=1
      write(*,*) 'give no of points for memory of nperiod (typically nperiod
     * itselve ) '
      read(*,*) nqmx
150   ylo=1e8
      yhi=-1e8
      do 200 i=i1,i1+0.75*nperio
*      find nth maximum
c       write(*,*) yhi,ylo,y(i)
       if (y(i).gt.yhi) then
        ihi(n)=i
        yhi=y(i)
       endif
200   continue
      i1=ihi(n)
      do 205 i=i1,i1+1.25*nperio
*      find nth minimum
       if (y(i).lt.ylo) then
        ilo(n)=i
        ylo=y(i)
       endif
205   continue
      i1=ilo(n)
```

```
      if (.not.(n.eq.1)) then
       nq=n-1
       if (n.gt.nqmx) nq=nqmx
       nperio=(ihi(n)-ihi(n-nq))/nq
       p=nq/(2*dsub*(x(ihi(n))-x(ihi(n-nq))))
      else
       p=1/(2*dsub*(x(1+nperio)-x(1)))
      endif
      write(*,*) 'n,x,di,period',n,x(ihi(n)),ihi(n)-ihi(n-1),nperio
      if (nperio.eq.0) then
       write(*,*) 'nperio = 0, endless loop, give new value '
       read(*,*) nperio
      endif
      i2=i1+1.25*nperio
      if (i2.gt.imx) goto 151
      xmx(n)=x(ihi(n))
      ymx(n)=y(ihi(n))
      xmn(n)=x(ilo(n))
      ymn(n)=y(ilo(n))
      epssub(n)=p**2
      n=n+1
      goto 150
151   nmx=n
      do 300 n=2,nmx-1
c       p=real(csqrt(epssub(n-1)))
c       i1=ilo(n-1)
c       i2=ihi(n)
c       i3=ilo(n)
c       i4=ihi(n+1)
c       ysel=ymn(n)+0.5*(ymx(n)-ymn(n))
cc select the points around nth maximum for which y > ysel
c       write(*,*) 'k ',x(ihi(n)),' nsubs ',p
c       itl=0
c       write(*,*) 'pts around ',xmx(n),' with ',ymx(n),' > y > ',ysel
c       do 310 i=i1,i3
c        if (y(i).gt.ysel) then
c         itl=itl+1
c         xx(itl)=x(i)
c         yy(itl)=y(i)
c         sig(itl)=1.
c         write(*,*) itl,i,xx(itl),yy(itl)
c        endif
c310    continue
c       ntl=itl
c       phi=0
cc fit a sinus to this set of points:
c       write(*,*) 'old xmx,ymx is : ',xmx(n),ymx(n)
c       call fitsin(xx,yy,sig,ntl,dsub,p,xmx(n),ymx(n),ymn(n),phi)
c       write(*,*) 'new xmx,ynx is : ',xmx(n),ymx(n)
cc select the points around nth minimum for which y < ysel
c       itl=0
c       write(*,*) 'pts around ',xmn(n),' with ',ymn(n),' < y < ',ysel
c       do 320 i=i2,i4
c        if (y(i).lt.ysel) then
c         itl=itl+1
c         xx(itl)=x(i)
c         yy(itl)=y(i)
c         sig(itl)=1.
```

```
c         write(*,*) itl,i,xx(itl),yy(itl)
c        endif
c320    continue
c       ntl=itl
c       phi=0
cc fit a sinus to this set of points: y=ymx-alfa*(x-xmx)**2
c       write(*,*) 'old xmn,ymn is : ',xmn(n),ymn(n)
c       call fitsin(xx,yy,sig,ntl,dsub,p,xmx(n),ymx(n),ymn(n),phi)
c       write(*,*) 'new xmn,ynn is : ',xmn(n),ymn(n)
c
       xav(n)=xmn(n-1)
       ymin=ymn(n-1)
       ymax=(ymx(n)+ymx(n-1))/2
c       write(28,*) xav(n),(ymax-ymin)/2,(ymin+ymax)/2
       p=sqrt(ymax-ymin)+sqrt(1+ymax-ymin)
       a=ymax+ymin
       b=(1+p)**2/(2*p)
       c=(1-p)**2/(2*p)
       dp=a/b**2+sqrt(a**2/b**4-c**2/b**2)
c       dm=a/b**2-sqrt(a**2/b**4-c**2/b**2)
       pimp=0.5*alog(dp)/(xav(n)*2*pi*dsub)
       epssub(n)=cmplx(p**2,2*pimp*p)
       write(23,*) xav(n),real(epssub(n)),aimag(epssub(n))
300    continue
      close(23)
      close(28)
      end
*************************************************************************

*************************************************************************
      subroutine fitsin(xx,yy,sig,nfil,dsubs,p,xav,ymx,ymn,phi)
      parameter(nfys=50,ma=6,nca=6)
      integer nfil,i,lista(ma),mfit,flag
      real pi,xx(nfys),yy(nfys),work(nfys),tol(nfys)
     *  ,a(ma),dyda(ma),covar(nca,nca),alpha(nca,nca)
     *  ,beta(ma),ochisq,chisq,alamda
     *  ,alamol,chilim
     *  ,p,dsubs,dfilm,w,x,y,xav,ymx,ymn,phi
      external sinus
      pi=4.*atan(1.)
******Fit the double-layer expression to the data
c     initialize fitparameters:
      mfit=2
      lista(1)=1
      lista(2)=4
      a(1)=(ymx+ymn)/2
      a(2)=(ymx-ymn)/2
      a(3)=4*pi*dsubs*p
      a(4)=phi
*     stop if the curve fits better than 0.00001 (absolute) per point:
      chilim=0.00001
      chilim=nfil*(chilim**2)
      alamda=-1
      flag=0
      alamol=alamda
      write(*,'(6a8,1h )') 'yav,yamp,freq,phi'
        write(*,*) a(1),a(2),a(3),a(4)
      do 45 i=1,1000
```

```
        call mrqmin(xx,yy,sig,nfil,a,dyda,ma,lista,mfit,
     *      covar,alpha,beta,nca,chisq,ochisq,alamda,sinus)
        write(*,'(6(e9.3,1h ))') alamda,chisq,a(1),a(2),a(3),a(4)
       if (alamda.gt.alamol) then
        flag=flag+1
       else
        flag=0
       endif
       alamol=alamda
       if (chisq.lt.chilim) then
        write(*,*) 'Iteration interrupted. Reason: '
        write(*,*) chisq,' = chisq dropped below limit = ',chilim
        goto 46
       endif
       if (flag.gt.6) then
        write(*,*) 'Iteration interrupted. Reason: '
        write(*,*) alamda,' = alamda increased six times '
        goto 46
       endif
45     continue
       write(*,*) 'Iteration interrupted. Reason: '
       write(*,*) 'number of iterations equals ', i
46     alamda=0
       call mrqmin(xx,yy,sig,nfil,a,dyda,ma,lista,mfit,
     *    covar,alpha,beta,nca,chisq,ochisq,alamda,sinus)
******fit is finished
       ymx=a(1)+a(2)
       ymn=a(1)-a(2)
       phi=a(4)
       do 100 i=1,nfil
        x=xx(i)
        call sinus(x,a,y,dyda,ma)
        write(28,*) x,yy(i),y
100    continue
       xav=0
       do 33 i=1,nfil
        xav=xav+xx(i)
33     continue
       xav=xav/nfil
       return
       END
**********************************************************************


**********************************************************************
       subroutine sinus(x,a,y,dyda,na)
       real klad
       dimension a(na),dyda(na)
       y=a(1)+a(2)*cos(x*a(3)+a(4))
       y=a(1)+a(2)*klad
       dyda(1)=1
       dyda(2)=klad
       klad=-a(2)*sin(x*a(3)+a(4))
       dyda(3)=klad*x
       dyda(4)=klad
       return
       end
**********************************************************************
```

```
**********************************************************************
      SUBROUTINE MRQMIN(X,Y,SIG,NDATA,A,DYDA,MA,LISTA,MFIT,
     *    COVAR,ALPHA,beta,NCA,CHISQ,ochisq,ALAMDA,myfunc)
      PARAMETER (MMAX=20)
      REAL X(NDATA),Y(NDATA),SIG(NDATA),A(MA),DYDA(MA),
     *  COVAR(NCA,NCA),ALPHA(NCA,NCA),ATRY(MMAX),BETA(ma),DA(MMAX)
      INTEGER LISTA(MA)
      integer kk,j,ihit,k
      external myfunc
      IF(ALAMDA.LT.0.)THEN
        KK=MFIT+1
        DO 12 J=1,MA
          IHIT=0
          DO 11 K=1,MFIT
            IF(LISTA(K).EQ.J)IHIT=IHIT+1
11        CONTINUE
          IF (IHIT.EQ.0) THEN
            LISTA(KK)=J
            KK=KK+1
          ELSE IF (IHIT.GT.1) THEN
            PAUSE 'Improper permutation in LISTA'
          ENDIF
12      CONTINUE
        IF (KK.NE.(MA+1)) PAUSE 'Improper permutation in LISTA'
        ALAMDA=0.001
        CALL MRQCOF(X,Y,SIG,NDATA,A,MA,LISTA,MFIT,ALPHA,BETA,NCA,
     *       CHISQ,dyda,myfunc)
        OCHISQ=CHISQ
        DO 13 J=1,MA
          ATRY(J)=A(J)
13      CONTINUE
      ENDIF
      DO 15 J=1,MFIT
        DO 14 K=1,MFIT
          COVAR(J,K)=ALPHA(J,K)
14      CONTINUE
        COVAR(J,J)=ALPHA(J,J)*(1.+ALAMDA)
        DA(J)=BETA(J)
15    CONTINUE
      CALL GAUSSJ(COVAR,MFIT,NCA,DA,1,1)
      IF(ALAMDA.EQ.0.)THEN
        CALL COVSRT(COVAR,NCA,MA,LISTA,MFIT)
        RETURN
      ENDIF
      do 36 j=1,ma
       atry(j)=a(j)
36    continue
      DO 16 J=1,MFIT
        ATRY(LISTA(J))=A(LISTA(J))+DA(J)
16    CONTINUE
      CALL MRQCOF(X,Y,SIG,NDATA,ATRY,MA,LISTA,MFIT,COVAR,DA,NCA,
     *  CHISQ,dyda,myfunc)
      IF(CHISQ.LT.OCHISQ)THEN
        ALAMDA=0.1*ALAMDA
        OCHISQ=CHISQ
        DO 18 J=1,MFIT
          DO 17 K=1,MFIT
```

```
            ALPHA(J,K)=COVAR(J,K)
17        CONTINUE
          BETA(J)=DA(J)
          A(LISTA(J))=ATRY(LISTA(J))
18      CONTINUE
      ELSE
        ALAMDA=10.*ALAMDA
        CHISQ=OCHISQ
      ENDIF
      RETURN
      END
*****************************************************************


*****************************************************************
      SUBROUTINE MRQCOF(X,Y,SIG,NDATA,A,MA,LISTA,MFIT
     * ,ALPHA,BETA,nalp,CHISQ,dyda,myfunc)
      REAL X(NDATA),Y(NDATA),SIG(NDATA),ALPHA(NALP,NALP),BETA(MA),
     *    A(MA),DYDA(MA)
      INTEGER LISTA(MFIT)
      integer j,k,i
      real ymod,sig2i,dy,wt
      external myfunc
      DO 12 J=1,MFIT
        DO 11 K=1,J
          ALPHA(J,K)=0.
11      CONTINUE
        BETA(J)=0.
12    CONTINUE
      CHISQ=0.
      DO 15 I=1,NDATA
        CALL myfunc(X(I),A,YMOD,DYDA,MA)
        SIG2I=1./(SIG(I)*SIG(I))
        DY=Y(I)-YMOD
        DO 14 J=1,MFIT
          WT=DYDA(LISTA(J))*SIG2I
          DO 13 K=1,J
            ALPHA(J,K)=ALPHA(J,K)+WT*DYDA(LISTA(K))
13        CONTINUE
          BETA(J)=BETA(J)+DY*WT
14      CONTINUE
        CHISQ=CHISQ+DY*DY*SIG2I
15    CONTINUE
      DO 17 J=2,MFIT
        DO 16 K=1,J-1
          ALPHA(K,J)=ALPHA(J,K)
16      CONTINUE
17    CONTINUE
      RETURN
      END
*****************************************************************


*****************************************************************
      SUBROUTINE COVSRT(COVAR,NCVM,MA,LISTA,MFIT)
      real COVAR(NCVM,NCVM)
      integer LISTA(MFIT)
      integer j,i
```

```
      real swap
      DO 12 J=1,MA-1
        DO 11 I=J+1,MA
          COVAR(I,J)=0.
11      CONTINUE
12    CONTINUE
      DO 14 I=1,MFIT-1
        DO 13 J=I+1,MFIT
          IF(LISTA(J).GT.LISTA(I)) THEN
            COVAR(LISTA(J),LISTA(I))=COVAR(I,J)
          ELSE
            COVAR(LISTA(I),LISTA(J))=COVAR(I,J)
          ENDIF
13      CONTINUE
14    CONTINUE
      SWAP=COVAR(1,1)
      DO 15 J=1,MA
        COVAR(1,J)=COVAR(J,J)
        COVAR(J,J)=0.
15    CONTINUE
      COVAR(LISTA(1),LISTA(1))=SWAP
      DO 16 J=2,MFIT
        COVAR(LISTA(J),LISTA(J))=COVAR(1,J)
16    CONTINUE
      DO 18 J=2,MA
        DO 17 I=1,J-1
          COVAR(I,J)=COVAR(J,I)
17      CONTINUE
18    CONTINUE
      RETURN
      END
*****************************************************************************


*****************************************************************************
      SUBROUTINE GAUSSJ(A,N,NP,B,M,MP)
      PARAMETER (NMAX=20)
      real A(NP,NP),B(NP,MP)
      integer IPIV(NMAX),INDXR(NMAX),INDXC(NMAX)
      integer j,i,k,irow,icol,l,ll
      real big,dum,pivinv
      DO 11 J=1,N
        IPIV(J)=0
11    CONTINUE
      DO 22 I=1,N
        BIG=0.
        DO 13 J=1,N
          IF(IPIV(J).NE.1)THEN
            DO 12 K=1,N
              IF (IPIV(K).EQ.0) THEN
                IF (ABS(A(J,K)).GE.BIG)THEN
                  BIG=ABS(A(J,K))
                  IROW=J
                  ICOL=K
                ENDIF
              ELSE IF (IPIV(K).GT.1) THEN
                PAUSE 'Singular matrix'
              ENDIF
```

```
12          CONTINUE
          ENDIF
13      CONTINUE
        IPIV(ICOL)=IPIV(ICOL)+1
        IF (IROW.NE.ICOL) THEN
          DO 14 L=1,N
            DUM=A(IROW,L)
            A(IROW,L)=A(ICOL,L)
            A(ICOL,L)=DUM
14        CONTINUE
          DO 15 L=1,M
            DUM=B(IROW,L)
            B(IROW,L)=B(ICOL,L)
            B(ICOL,L)=DUM
15        CONTINUE
        ENDIF
        INDXR(I)=IROW
        INDXC(I)=ICOL
        IF (A(ICOL,ICOL).EQ.0.) PAUSE 'Singular matrix.'
        PIVINV=1./A(ICOL,ICOL)
        A(ICOL,ICOL)=1.
        DO 16 L=1,N
          A(ICOL,L)=A(ICOL,L)*PIVINV
16      CONTINUE
        DO 17 L=1,M
          B(ICOL,L)=B(ICOL,L)*PIVINV
17      CONTINUE
        DO 21 LL=1,N
          IF(LL.NE.ICOL)THEN
            DUM=A(LL,ICOL)
            A(LL,ICOL)=0.
            DO 18 L=1,N
              A(LL,L)=A(LL,L)-A(ICOL,L)*DUM
18          CONTINUE
            DO 19 L=1,M
              B(LL,L)=B(LL,L)-B(ICOL,L)*DUM
19          CONTINUE
          ENDIF
21      CONTINUE
22    CONTINUE
      DO 24 L=N,1,-1
        IF(INDXR(L).NE.INDXC(L))THEN
          DO 23 K=1,N
            DUM=A(K,INDXR(L))
            A(K,INDXR(L))=A(K,INDXC(L))
            A(K,INDXC(L))=DUM
23        CONTINUE
        ENDIF
24    CONTINUE
      RETURN
      END
*****************************************************************************
```