```
c   NONSMOOTH OPTIMIZATION SUBGRADIENT METHOD WITH VARIABLE METRIC
c   AND STEPSIZE CONTROL USING SUBGRADIENTS FOR THE LINE SEARCH

c  IDENTIFICATION
c  Variable metric algorithm for nonsmooth optimization problems.
c  Fortran 77 subroutine was written by S. Uryasev at the International
c  Institute for Applied System Analysis, A-2361 Laxenburg Austria.
c  Version of December 24 1989.
c
c
c  METHOD
c  Adaptive variable metric algorithm. The algorithm simultaneously runs
two
c  subgradient methods: the first in the main space, and the second
c  with respect to the matrices that modify the space variables.
c
c  REFERENCES
c  1. Uryasev S.P. Adaptive Variable Metric Algorithms for Nonsmooth
c     Optimization Problems, IIASA, Laxenburg, Austria, WP-88-60, 1988.
c  2. Uryasev S.P. New  Variable Metric Algorithms for Nondifferenti-
c     able Optimization Problems. Journal of Optimization Theory and
c     Applications  Vol. 71, No. 2, 359-388, 1991.
c



c     START MODULE (AN EXAMPLE)
c     DESCRIPTIONS
      implicit real*8(a-h,o-z)
      integer s,sfull,smax

      parameter (nd=2,nkd=2)

      dimension x(nd),xpr(nd),gr(nd),grpr(nd),grold(nd),b(nd,nkd),
     1          grb(nkd),grbb(nd),vsold(nkd),vsnew(nkd)

c     METHOD PARAMETERS
      noprin=10
      smax=100
      n=nd
      nk=nkd
      bstep=0.65
      ro=0.1
      rodecr=0.7
      roincr=1.25
      grstop=1.d-10
      dxstop=1.d-5
      flevel= 0.

      ifpr=2
      open(ifpr,file='varmetg.res')

c     INPUT X
      x(1)=10.
```

```
      x(2)=1.

       call varmetg(n,nk,x,xpr,gr,grpr,grold,b,grb,grbb,vsold,vsnew,
     1              flevel,fn,ro,rodecr,roincr,bstep,ifpr,noprin,s,
     2              sfull,smax,grstop,dxstop,lstop)


      close(ifpr)
      stop
      end


c   NONSMOOTH OPTIMIZATION SUBGRADIENT METHOD WITH VARIABLE METRIC
c   AND STEPSIZE CONTROL USING SUBGRADIENTS FOR THE LINE SEARCH
c
       subroutine varmetg(n,nk,x,xpr,gr,grpr,grold,b,grb,grbb,vsold,
     1            vsnew,flevel,fn,ro,rodecr,roincr,bstep,ifpr,noprin,
     2            s,sfull,smax,grstop,dxstop,lstop)
c
c
c  PURPOSE
c  Varmetg finds an approximation of a minimum point of a function
(possibly
c  nonsmooth) on Euclidear n-dimentional space.
c
c
c  REQUIRED SUBROUTINES
c  The user should write subroutines fun and grad
c  according to the following format:
c
c     subroutine fun(x,n,fn)
c     implicit real*8(a-h,o-z)
c     dimension x(n)
c     ...
c     fn = (the value of objective function f(x) at x)
c     return
c     end
c
c     subroutine grad(x,gr,n)
c     implicit real*8(a-h,o-z)
c     dimension x(n),gr(n)
c     ...
c     do 10 j=1,n
c     gr(j) = (the value of j-th component of the objective function
c              f(x) subgradient at the point x)
c  10 continue
c     return
c     end
c
c  where
c  (input)  x      is a double precision input array, argument of the
c                  objective function f(x);
c  (input)  n      is an integer input variable, the number of variables
c                  in the objective function f(x);
```

```
c  (output) fn     is a double precision output variable, the value of
c                  the objective function;
c  (output) gr     is a double precision output array, the subgradient
c                  of the objective function.
c
c  The subroutine varmetg calls subroutines wrivar and regsb.
c  These subroutines are supplied together with varmetg.
c
c  CONTROL
c
c     implicit real*8(a-h,o-z)
c     integer s,sfull,smax
c     dimension x(n),xpr(n),gr(n),grold(n),b(n,nk),grb(nk),grbb(n),
c    1          vsold(nk),vsnew(nk)
c      ...
c       call varmetg(n,nk,x,xpr,gr,grpr,grold,b,grb,grbb,vsold,vsnew,
c    1               flevel,fn,ro,rodecr,roincr,bstep,ifpr,noprin,s,
c    2               sfull,smax,grstop,dxstop,lstop)
c      ...
c
c  where
c  (input)  n      is an integer input variable, the number of variables
c                  in the objective function f(x) (n is number of rows in
c                  the matrix b);
c  (input)  nk     is an integer input variable, the number of columns in
c                  the matrix b (if RAM is enough for n*n numbers than
c                  nk=n, otherwise 1 < nk < n);
c  (input-
c   output) x      is a double precision input array containing the
c                  initial approximation of the solution vector; after
c                  finishing the work this vector contains
c                  the best appriximation of the solution vector;
c  (auxil)  xpr    is a double precision auxiliary array containing
c                  the approximation of the solution vector at the
c                  previous point;
c  (output) gr     is a double precision output array, the subgradient
c                  of the objective function;
c  (auxil)  grold  is a double precision auxiliary array containing
c                  the subgradient of the objective function at the
c                  previous point;
c  (auxil)  b      is a double precision auxiliary array for metric
c                  change;
c  (auxil)  grb    is a double precision auxiliary array;
c  (auxil)  grbb   is a double precision auxiliary array;
c  (auxil)  vsold  is a double precision auxiliary array;
c  (auxil)  vsnew  is a double precision auxiliary array;
c  (input)  flevel is a double precision input variable, algorithm
c                  stops if a value of objective function is less than
c                  flevel;
c  (output) fn     is a double precision output variable, the value of
c                  objective function;
c  (input)  ro     is a double precision input variable, initial step-
c                  size with respect to x ( ro > 0 );
c  (input)  rodecr is a double precision input variable, a decrease
```

```
c                  coefficient of the stepsize ro ( 0 < rodecr < 1, re-
c                  commendable value  rodecr=0.7 );
c  (input)  roincr is a double precision input variable, an increase
c                  coefficient of the stepsize ro ( roincr > 1, recom-
c                  mendable value roincr=1.25 );
c  (input)  bstep  is a double precision input variable, a stepsize
c                  with respect to the matrix b ( 0 < bstep < 1, recom-
c                  mendable value bstep=0.65 );
c  (input)  ifpr   is an integer input variable, a channel number for
c                  print file;
c  (input)  noprin is an integer input variable, a print interval (each
c                  noprin iteration vectors x,gr are written in the file
c                  with channel number ifpr);
c  (output) s      is an integer output variable, the algorithm itera-
c                  tion number (amount of subgradient calulation);
c  (output) sfull  is an integer output variable, the total amount of
c                  objective function calculation;
c  (input)  smax   is an integer input variable, a maximal amount of the
c                  algorithm iteration (algorithm stops if s is equal
c                  smax );
c  (input)  grstop is a double precision input variable, algorithm stops
c                  if a subgradient norm is less than grstop;
c  (input)  dxstop is a double precision input variable, algorithm
c                  stops if a distance between two successive approxi-
c                  mations is less than dxstop;
c  (output) lstop  is an integer output variable, the return code of the
c                  algorithm:
c                  lstop=0 if algorithm  stops when a value of the objec-
c                          tive function is less than flevel,
c                  lstop=1 if algorithm  stops when a subgradient norm
c                          is less than grstop,
c                  lstop=2 if algorithm  stops when a distance between
c                          two successive approximations is less than
c                          dxstop,
c                  lstop=3 if algorithm  stops when s=smax;


c
c     DESCRIPTIONS
      implicit real*8(a-h,o-z)
      integer s,sfull,smax

      dimension x(n),xpr(n),gr(n),grpr(n),grold(n),b(n,nk),grb(nk),
     1          grbb(n),vsold(nk),vsnew(nk)

c     INITIALIZATION
      nouv=1
      lstep=0
      induv=1
      sfull=1
      s=1
      ropr=ro
      lbadst=0
      nbadst=3
```

```
      grbnrm=1.
c     FUNCTION CALCULATION
      call fun(x,n,fn)
c     FUNCTION VALUE STOP
      if (flevel.ge.fn) then
c        WRITE
         call wrivar(n,x,gr,fn,s,sfull,ro,ifpr,noprin)
         lstop=0
         return
      end if
c     SUBGRADIENT CALCULATION
      call grad(x,gr,n)

c     WRITE
      call wrivar(n,x,gr,fn,s,sfull,ro,ifpr,noprin)

c     CALCULATION OF SUBGRADIENT NORM
      grnorm=0.
      do 10 i=1,n
         grnorm=grnorm+gr(i)**2
   10 continue
      grnorm=dsqrt(grnorm)

c     SUBGRADIENT NORM STOP
      if (grnorm.le.grstop) then
         lstop=1
         return
      end if
c     SUBGRADIENT NORMALIZATION
      do 20 i=1,n
         gr(i)=gr(i)/grnorm
   20 continue

c     BEGINNING OF MAIN CYCLE
      do 9999 s=1,smax

c     GROLD MEMORY
      do 30 i=1,n
         grold(i)=gr(i)
   30 continue

c     CALCULATION OF GRB,GRBB
      if (s.eq.1) then
         do 40 i=1,n
           grbb(i)=gr(i)
   40    continue
      else
         do 50 i=1,nk
            grb(i)=0.
            do 60 j=1,n
               grb(i)=grb(i)+b(j,i)*gr(j)
   60       continue
   50    continue
         do 70 i=1,n
```

```
            grbb(i)=0.
            do 80 j=1,nk
               grbb(i)=grbb(i)+b(i,j)*grb(j)
   80       continue
   70    continue
c        CALCULATION OF GRB NORM
         grbnrm=0.
         do 90 i=1,nk
            grbnrm=grbnrm+grb(i)**2
   90    continue
         grbnrm=dsqrt(grbnrm)
         if ( grbnrm.le.1.d-20) grbnrm=1.d-20
      end if

c     LINE SEARCH
  100 continue
c          XPR,FNPR MEMORY
           do 110 i=1,n
              xpr(i)=x(i)
  110      continue
           fnpr=fn

c          GRPR MEMORY
           do 120 i=1,n
              grpr(i)=gr(i)
  120      continue

           shnorm=0.
           do 130 i=1,n
c             MOTION IN SPACE X
              shift=grbb(i)*ro/grbnrm
              x(i)=x(i)-shift
c             CALCULATION OF SHIFT NORM
              shnorm=shnorm+shift**2
  130      continue
           shnorm=dsqrt(shnorm)

c          SHIFT NORM STOP
           if (shnorm.le.dxstop) then
c             WRITE
              call wrivar(n,x,gr,fn,s,sfull,ro,ifpr,noprin)
              lstop=2
              return
           end if

c          FUNCTION  CALCULATION
           call fun(x,n,fn)
c          FUNCTION VALUE STOP
           if (flevel.ge.fn) then
c             WRITE
             call wrivar(n,x,gr,fn,s,sfull,ro,ifpr,noprin)
             lstop=0
             return
           end if
```

```
c          SUBGRADIENT CALCULATION
           call grad(x,gr,n)

c          INNER PRODUCT SKPR CALCULATION
           skpr=0.
           do 140 i=1,n
              skpr=skpr+grbb(i)*gr(i)
  140      continue

c          INCREASE OF SFULL
           sfull=sfull+1

c          RO CONTROL
           if (skpr.gt.0.) then
               lstep=lstep+1
               if (s.eq.1) ro=ro*1.5
               if (lstep.gt.nouv .and. s.gt.1) ro=ro*roincr
      go to 100
           end if
      if (lstep.eq.0) ro=ro*rodecr

c     INCREASE OF LBADST
      lbadst=lbadst+1

c     RO CONTROL
      if (lstep.gt.0 .and. lbadst.le.nbadst) ro=dmax1(ropr,ro)
      if (lstep.gt.0) then
          ropr=ro
          lbadst=0
      end if

c     CALCULATION OF SUBGRADIENT NORM
      grnorm=0.
      do 150 i=1,n
         grnorm=grnorm+gr(i)**2
  150 continue
      grnorm=dsqrt(grnorm)

c     SUBGRADIENT NORM STOP
      if (grnorm.le.grstop) then
c        WRITE
         call wrivar(n,x,gr,fn,s,sfull,ro,ifpr,noprin)
         lstop=1
         return
      end if

c     SUBGRADIENT NORMALIZATION
      do 160 i=1,n
         gr(i)=gr(i)/grnorm
  160 continue

c     MATRIX B COMPUTATION
      call regsb(n,nk,s,gr,grold,b,bstep,vsold,vsnew)
```

```
      if (fnpr.lt.fn) then
c        GR,X,FN RESET
         do 170 i=1,n
            x(i)=xpr(i)
  170    continue
         fn=fnpr

         if (lstep.gt.0) then
c           CALCULATION OF GRPR NORM
            grnorm=0.
            do 180 i=1,n
               grnorm=grnorm+grpr(i)**2
  180       continue
            grnorm=dsqrt(grnorm)

c           SUBGRADIENT NORM STOP
           if (grnorm.le.grstop) then
c             WRITE
              call wrivar(n,x,gr,fn,s,sfull,ro,ifpr,noprin)
              lstop=1
              return
           end if

c           GRPR NORMALIZATION
            do 190 i=1,n
              grpr(i)=grpr(i)/grnorm
  190       continue
         end if

         do 200 i=1,n
            gr(i)=grpr(i)
  200    continue
      end if

c     WRITE
      call wrivar(n,x,gr,fn,s,sfull,ro,ifpr,noprin)

c     LSTEP RESET
      lstep=0

c     END OF MAIN CYCLE
 9999 continue
      lstop=3
      return
      end

c     MATRIX B CALCULATION
      subroutine regsb(n,nk,s,gr,grold,b,bstep,vsold,vsnew)

c  IDENTIFICATION
c  Update matrix b using the gradient method with respect to b.
c  Fortran 77 subroutine written by S. Uryasev at the International
c  Institute for Applied System Analysis, A-2361 Laxenburg Austria.
c  Version of December 24 1989.
```

```
c
c  PURPOSE
c  Update matrix b using the subgradient method.
c
c  CONTROL
c
c     implicit real*8(a-h,o-z)
c     integer s
c     dimension gr(n),grold(n),b(n,nk),vsold(nk),vsnew(nk)
c     ...
c     call regsb(n,nk,s,gr,grold,b,bstep,vsold,vsnew)
c     ...
c
c  where
c  (input)  n      is an integer input variable, the number of rows in
c                  the matrix b;
c  (input)  nk     is an integer input variable, the number of columns in
c                  the matrix b (if RAM is enough for n*n numbers than
c                  nk=n, otherwise 1 < nk < n);
c  (input)  s      is an integer input variable, the algorithm itera-
c                  tion number;
c  (input)  gr     is a double precision input array, the subgradient
c                  of the objective function;
c  (input)  grold  is a double precision input array containing
c                  the subgradient of the objective function at the
c                  previous point;
c  (input-
c   output) b      is a double precision input-output array for metric
c                  change;
c  (input)  bstep  is a double precision input variable, a stepsize
c                  with respect to matrix b ( 0 < bstep < 1 );
c  (auxil)  vsold  is a double precision auxiliary array;
c  (auxil)  vsnew  is a double precision auxiliary array;

      implicit real*8(a-h,o-z)
      integer s
      dimension gr(n),grold(n),b(n,nk),vsold(nk),vsnew(nk)

c     ASSIGNING OF INITIAL B
      if (s.eq.1) then
         do 180 i=1,n
            do 190 j=1,nk
               if (i.eq.j) then
                   b(i,j)=1.
               else
                   b(i,j)=0.
               end if
  190       continue
  180    continue
      end if
c     CALCULATION OF VSOLD,VSNEW
      do 200 j=1,nk
         vsold(j)=0.
         vsnew(j)=0.
```

```
         do 210 i=1,n
            vsold(j)=vsold(j)+b(i,j)*grold(i)
            vsnew(j)=vsnew(j)+b(i,j)*gr(i)
  210    continue
  200 continue
c     B CALCULATION
      do 220 i=1,n
         do 230 j=1,nk
            b(i,j)=b(i,j)+bstep*(gr(i)*vsold(j)+grold(i)*vsnew(j))
  230    continue
  220 continue

      return
      end


      subroutine wrivar(n,x,gr,fn,s,sfull,ro,ifpr,noprin)
c
c  IDENTIFICATION
c  The subroutine for algorithm trajectory writing.
c  Fortran 77 subroutine written by S. Uryasev at the International
c  Institute for Applied System Analysis, A-2361 Laxenburg Austria.
c  Version of December 24 1989.
c
c  PURPOSE
c  To write the algorithm trajectory in a file with channel number ifpr.
c
c  CONTROL
c
c     implicit real*8(a-h,o-z)
c     integer s,sfull
c     dimension x(n),gr(n)
c     ...
c     call wrivar(n,x,gr,fn,s,sfull,ro,ifpr,noprin)
c     ...
c
c  where
c  (input)  n      is an integer input variable, the dimension of the
c                  vector x;
c  (input)  x      is a double precision input array containing the
c                  vector x to be written in the file;
c  (input)  gr     is a double precision input array containing the
c                  vector gr to be written in the file;
c  (input)  fn     is a double precision input variable, the value of
c                  objective function to be written in the file;
c  (input)  s      is an integer input variable, the algorithm itera-
c                  tion number to be written in the file;
c  (input)  sfull  is an integer input variable, the algorithm full
c                  iteration number to be written in the file;
c  (input)  ro     is a double precision input variable, initial step-
c                  size to be written in the file;
c  (input)  ifpr   is an integer input variable, a channel number for
c                  print file;
c  (input)  noprin is an integer input variable, a print interval (each
```

```
c                  noprin iteration vectors x,gr are written in the file
c                  with channel number ifpr);
c

      implicit real*8(a-h,o-z)
      integer s,sfull
      dimension x(n),gr(n)

      linlen=8

      ncycle=(n-1)/linlen+1
      if ((mod(s,noprin).eq.0) .or. (sfull.eq.1)) then
         do 10 ic=1,ncycle
            ism=(ic-1)*linlen+1
            igr=ic*linlen
            if (igr.gt.n) igr=n
            write(ifpr,9000)(i,i=ism,igr)
            write(ifpr,9010)(x(i),i=ism,igr)
            write(ifpr,9020)(gr(i),i=ism,igr)
   10    continue
      end if
      write(ifpr,9030)s,fn,sfull,ro
 9000 format(/,3x,10i14)
 9010 format('  x',10d14.6)
 9020 format(' gr',10d14.6)
 9030 format('   s=',i5,'    fn=',d15.9,'    sfull=',i5,
     1'   ro=',d14.6)
      return
      end


c     GRAD CALCULATION
      subroutine grad(x,gr,n)

      implicit real*8(a-h,o-z)
      dimension x(n),gr(n)

      ds=dsign(1.,x(2)-x(1))
      ds1=dsign(1.,x(1)-1.)

      gr(1)=-100.0*ds+ds1
      gr(2)=100.0*ds

      return
      end

c     FN CALCULATION
      subroutine fun(x,n,fn)

      implicit real*8(a-h,o-z)
      dimension x(n)

      fn=100.0*dabs(x(2)-x(1))+dabs(x(1)-1.)
```

```
      return
      end


################################################################################
##
THE TEST RUN
################################################################################
####


               1             2
 x   .100000D+02   .100000D+01
gr   .101000D+03  -.100000D+03
  s=    1    fn= .909000000D+03    sfull=    1   ro=   .100000D+00
  s=    1    fn= .162723473D+03    sfull=   10   ro=   .256289D+01
  s=    2    fn= .545295975D+02    sfull=   12   ro=   .256289D+01
  s=    3    fn= .191032852D+02    sfull=   14   ro=   .256289D+01
  s=    4    fn= .801436723D+01    sfull=   16   ro=   .256289D+01
  s=    5    fn= .220897029D+01    sfull=   18   ro=   .256289D+01
  s=    6    fn= .183874144D+00    sfull=   20   ro=   .256289D+01
  s=    7    fn= .183874144D+00    sfull=   21   ro=   .179402D+01
  s=    8    fn= .183874144D+00    sfull=   22   ro=   .125582D+01
  s=    9    fn= .183874144D+00    sfull=   23   ro=   .879071D+00

               1             2
 x   .104734D+01   .104598D+01
gr   .710616D+00  -.703580D+00
  s=   10    fn= .183874144D+00    sfull=   24   ro=   .615350D+00
  s=   11    fn= .183874144D+00    sfull=   25   ro=   .430745D+00
  s=   12    fn= .183874144D+00    sfull=   26   ro=   .301521D+00
  s=   13    fn= .183874144D+00    sfull=   27   ro=   .211065D+00
  s=   14    fn= .183874144D+00    sfull=   28   ro=   .147746D+00
  s=   15    fn= .143392258D+00    sfull=   29   ro=   .103422D+00
  s=   16    fn= .136059952D+00    sfull=   31   ro=   .103422D+00
  s=   17    fn= .575391675D-01    sfull=   33   ro=   .103422D+00
  s=   18    fn= .575391675D-01    sfull=   34   ro=   .723953D-01
  s=   19    fn= .571824912D-01    sfull=   35   ro=   .506767D-01

               1             2
 x   .999683D+00   .100024D+01
gr  -.710616D+00   .703580D+00
  s=   20    fn= .559306508D-01    sfull=   37   ro=   .103422D+00
  s=   21    fn= .559306508D-01    sfull=   38   ro=   .723953D-01
  s=   22    fn= .559306508D-01    sfull=   39   ro=   .506767D-01
  s=   23    fn= .363386714D-01    sfull=   41   ro=   .103422D+00
  s=   24    fn= .363386714D-01    sfull=   42   ro=   .723953D-01
  s=   25    fn= .363386714D-01    sfull=   43   ro=   .506767D-01
  s=   26    fn= .189796495D-01    sfull=   45   ro=   .103422D+00
  s=   27    fn= .189796495D-01    sfull=   46   ro=   .723953D-01
  s=   28    fn= .189796495D-01    sfull=   47   ro=   .506767D-01
  s=   29    fn= .189796495D-01    sfull=   48   ro=   .354737D-01
```

```
                1             2
 x   .100006D+01   .999871D+00
gr   .710616D+00  -.703580D+00
  s=   30    fn= .189796495D-01    sfull=   49   ro=   .248316D-01
  s=   31    fn= .189796495D-01    sfull=   50   ro=   .173821D-01
  s=   32    fn= .189796495D-01    sfull=   51   ro=   .121675D-01
  s=   33    fn= .176156421D-01    sfull=   53   ro=   .121675D-01
  s=   34    fn= .512740182D-02    sfull=   54   ro=   .851723D-02
  s=   35    fn= .512740182D-02    sfull=   55   ro=   .596206D-02
  s=   36    fn= .444511761D-02    sfull=   57   ro=   .121675D-01
  s=   37    fn= .444511761D-02    sfull=   58   ro=   .851723D-02
  s=   38    fn= .444511761D-02    sfull=   59   ro=   .596206D-02
  s=   39    fn= .200125851D-02    sfull=   60   ro=   .417344D-02

                1             2
 x   .999975D+00   .999956D+00
gr   .703545D+00  -.710651D+00
  s=   40    fn= .200125851D-02    sfull=   61   ro=   .292141D-02
  s=   41    fn= .200125851D-02    sfull=   62   ro=   .204499D-02
  s=   42    fn= .194032776D-02    sfull=   64   ro=   .204499D-02
  s=   43    fn= .183937718D-02    sfull=   65   ro=   .143149D-02
  s=   44    fn= .104594552D-02    sfull=   66   ro=   .100204D-02
  s=   45    fn= .717753307D-03    sfull=   68   ro=   .204499D-02
  s=   46    fn= .717753307D-03    sfull=   69   ro=   .143149D-02
  s=   47    fn= .688033786D-03    sfull=   70   ro=   .100204D-02
  s=   48    fn= .642039275D-03    sfull=   71   ro=   .701431D-03
  s=   49    fn= .642039275D-03    sfull=   72   ro=   .491002D-03

                1             2
 x   .999619D+00   .999619D+00
gr  -.710616D+00   .703580D+00
  s=   50    fn= .426179828D-03    sfull=   73   ro=   .343701D-03
  s=   51    fn= .320117628D-03    sfull=   75   ro=   .343701D-03
  s=   52    fn= .320117628D-03    sfull=   76   ro=   .240591D-03
  s=   53    fn= .144816023D-03    sfull=   77   ro=   .168414D-03
  s=   54    fn= .144816023D-03    sfull=   78   ro=   .117889D-03
  s=   55    fn= .144816023D-03    sfull=   79   ro=   .825226D-04
  s=   56    fn= .104966055D-03    sfull=   80   ro=   .577658D-04
  s=   57    fn= .875869178D-04    sfull=   82   ro=   .577658D-04
  s=   58    fn= .290542586D-04    sfull=   84   ro=   .577658D-04
  s=   59    fn= .290542586D-04    sfull=   85   ro=   .404361D-04

                1             2
 x   .100000D+01   .100000D+01
gr   .710616D+00  -.703580D+00
  s=   60    fn= .119560181D-04    sfull=   86   ro=   .283053D-04
  s=   61    fn= .119560181D-04    sfull=   87   ro=   .198137D-04
  s=   62    fn= .119560181D-04    sfull=   88   ro=   .138696D-04
  s=   63    fn= .815103827D-05    sfull=   89   ro=   .970870D-05
  s=   64    fn= .815103827D-05    sfull=   89   ro=   .970870D-05
```