ADAS Subroutine fmin

```
c  To get d1mach, mail netlib
c       send d1mach from core
      double precision function fmin(ax,bx,f,tol)
C----------------------------------------------------------------------
c
c      PURPOSE: Find an approximation x to the point where f(x)
c               attains a minimum on the interval (ax,bx).
c
c  input..
c
c  ax    left endpoint of initial interval
c  bx    right endpoint of initial interval
c  f     function subprogram which evaluates  f(x)  for any  x
c        in the interval  (ax,bx)
c  tol   desired length of the interval of uncertainty of the final
c        result (.ge.0.)
c
c  output..
c
c  fmin  abcissa approximating the point where  f  attains a
c        minimum
c
c      the method used is a combination of  golden  section  search  and
c  successive parabolic interpolation.  convergence is never much slower
c  than  that  for  a  fibonacci search.  if  f  has a continuous second
c  derivative which is positive at the minimum (which is not  at  ax  or
c  bx),  then  convergence  is  superlinear, and usually of the order of
c  about  1.324....
c      the function  f  is never evaluated at two points closer together
c  than  eps*abs(fmin)+(tol/3), where eps is  approximately  the  square
c  root  of  the  relative  machine  precision.  if  f  is a unimodal
c  function and the computed values of  f  are  always  unimodal  when
c  separated  by  at least  eps*abs(x)+(tol/3), then  fmin  approximates
c  the abcissa of the global minimum of  f  on the interval  ax,bx  with
c  an error less than  3*eps*abs(fmin)+tol.  if  f  is  not  unimodal,
c  then fmin may approximate a local, but perhaps non-global, minimum to
c  the same accuracy.
c      this function subprogram is a slightly modified  version  of  the
c  algol  60 procedure  localmin  given in richard brent, algorithms for
c  minimization without derivatives, prentice-hall, inc. (1973).
c
C PUT INTO ADAS BY:
C    WILLIAM OSBORN, TESSELLA SUPPORT SERVICES PLC.
C
C DATE:    25TH APRIL 1996
C
C VERSION: 1.1 DATE: 25-04-96
C MODIFIED: WILLIAM OSBORN
C     - FOUND AT WWW.NETLIB.ORG
C
C VERSION: 1.2 DATE: 20-09-99
C MODIFIED: RICHARD MARTIN
```

```
C CHANGED FROM fmin.f TO fmin.for
C
C VERSION: 1.3 DATE: 16-05-07
C MODIFIED: Allan Whiteford
C     - Updated comments as part of subroutine documentation
C             procedure.
C
C-----------------------------------------------------------------------
      double precision ax,bx,f,tol
      double precision  a,b,c,d,e,eps,xm,p,q,r,tol1,t2,u,v,w,fu,fv,fw,
     2    fx,x,tol3
c      double precision  dabs,dsqrt,d1mach
      double precision  dabs,dsqrt,dpmpar
c
c  c is the squared inverse of the golden ratio
      c=0.5d0*(3.0d0-dsqrt(5.0d0))
c
c  eps is approximately the square root of the relative machine
c  precision.
c
C W.R.O. CHANGED FOLLOWING LINE FROM 10 eps=d1mach(4)

   10 eps=dpmpar(1)
      tol1=eps+1.0d0
      eps=dsqrt(eps)
c
      a=ax
      b=bx
      v=a+c*(b-a)
      w=v
      x=v
      e=0.0d0
      fx=f(x)
      fv=fx
      fw=fx
      tol3=tol/3.0d0
c
c  main loop starts here
c
   20 xm=0.5d0*(a+b)
      tol1=eps*dabs(x)+tol3
      t2=2.0d0*tol1
c
c  check stopping criterion
c
      if (dabs(x-xm).le.(t2-0.5d0*(b-a))) go to 190
      p=0.0d0
      q=0.0d0
      r=0.0d0
      if (dabs(e).le.tol1) go to 50
c
c  fit parabola
c
```

```
      r=(x-w)*(fx-fv)
      q=(x-v)*(fx-fw)
      p=(x-v)*q-(x-w)*r
      q=2.0d0*(q-r)
      if (q.le.0.0d0) go to 30
      p=-p
      go to 40
   30 q=-q
   40 r=e
      e=d
   50 if ((dabs(p).ge.dabs(0.5d0*q*r)).or.(p.le.q*(a-x))
     2          .or.(p.ge.q*(b-x))) go to 60
c
c  a parabolic-interpolation step
c
      d=p/q
      u=x+d
c
c  f must not be evaluated too close to ax or bx
c
      if (((u-a).ge.t2).and.((b-u).ge.t2)) go to 90
      d=tol1
      if (x.ge.xm) d=-d
      go to 90
c
c  a golden-section step
c
   60 if (x.ge.xm) go to 70
      e=b-x
      go to 80
   70 e=a-x
   80 d=c*e
c
c  f must not be evaluated too close to x
c
   90 if (dabs(d).lt.tol1) go to 100
      u=x+d
      go to 120
  100 if (d.le.0.0d0) go to 110
      u=x+tol1
      go to 120
  110 u=x-tol1
  120 fu=f(u)
c
c  update  a, b, v, w, and x
c
      if (fx.gt.fu) go to 140
      if (u.ge.x) go to 130
      a=u
      go to 140
  130 b=u
  140 if (fu.gt.fx) go to 170
      if (u.ge.x) go to 150
```

```
      b=x
      go to 160
  150 a=x
  160 v=w
      fv=fw
      w=x
      fw=fx
      x=u
      fx=fu
      go to 20
  170 if ((fu.gt.fw).and.(w.ne.x)) go to 180
      v=w
      fv=fw
      w=u
      fw=fu
      go to 20
  180 if ((fu.gt.fv).and.(v.ne.x).and.(v.ne.w)) go to 20
      v=u
      fv=fu
      go to 20
c
c  end of main loop
c
  190 fmin=x
      return
      end
      DOUBLE PRECISION    AX,          BX,          TOL
```