Appendix I: Levenberg-Marquart Method of Non-linear Least Squares

```
c------Title of program: Non-linear Least Squares INTEGER ma,nca,ndata,ia(1)

REAL*8 x(103),y(103),alamda,chisq,alpha(1,1),a(1),covar(1,1), & diff,u(103)

1      Format (i1)
2      Format (F10.8)

3      Format (F4.2)
4      Format(F13.10)
5      Format(F12.10)

6      Format(F4.1)
ma=1

ndata=103
nca=1

Open (Unit=20,File='Ne_input_1to_infinity.txt') Read(20,1)(ia(j),j=1,ma)
Read(20,2)(a(j),j=1,ma) Read(20,3)(x(j),j=1,ndata) Read(20,4)(y(j),j=1,ndata)
Read(20,5)(u(j),j=1,ndata)
Read(20,6)alamda Close (Unit=20)

Do 11 j=1,100

call mrqmin(x,y,u,ndata,a,ia,ma,covar,alpha,nca,chisq,alamda, & diff)

Open (Unit=20,File='Ne_output_1to_infinity.txt')
     Write(20,*)j,'                         '
     Write(20,*)chisq
     Write(20,*)(a(l),l=1,1)
     Write(20,*)alamda
     Write(20,*)diff
     Write(20,*)'---------------'
 11  Continue
     Close (Unit=20)
     Stop
     End
c------mrqmin-- does one iteration

SUBROUTINE mrqmin(x,y,u,ndata,a,ia,ma,covar,alpha,nca,chisq,
&      alamda,diff)
INTEGER ma,nca,ndata,ia(1),MMAX

REAL*8 alamda,chisq,a(1),alpha(1,1),covar(1,1),x(136),

&      y(136),u(136),diff PARAMETER (MMAX=2) INTEGER j,k,l,mfit
REAL*8 ochisq,atry(MMAX),beta(MMAX),da(MMAX) SAVE ochisq,atry,beta,da,mfit
if(alamda.lt.0)then

mfit=0
Do 11 j=1,ma
if (ia(j).ne.0) mfit=mfit+1
```

```
11         Continue alamda=0.001

Call mrqcof(x,y,u,ndata,a,ia,ma,alpha,beta,nca,chisq) ochisq=chisq
```

```
Do          12 j=1,ma atry(j)=a(j)

12       Continue endif

Do 14 j=1,mfit

Do 13 k=1,mfit covar(j,k)=alpha(j,k)

13       Continue covar(j,j)=alpha(j,j)*(1.+alamda) da(j)=beta(j)
14    Continue

call gaussj(covar,mfit,nca,da) j=0

Do 15 l=1,ma if(ia(l).ne.0) then
j=j+1

atry(l)=a(l)+da(j) endif
15    Continue

call mrqcof(x,y,u,ndata,atry,ia,ma,covar,da,nca,chisq) diff=ochisq-chisq

if(chisq.lt.ochisq)then

alamda=0.1*alamda

ochisq=chisq Do 17 j=1,mfit

Do 16 k=1,mfit alpha(j,k)=covar(j,k)

16         Continue beta(j)=da(j)

17       Continue

Do 18 l=1,ma a(l)=atry(l)

18       Continue else

alamda=10.*alamda chisq=ochisq

endif return End

c------mrqcof--calculates the Hessian(alpha) and merit function(chisq)
SUBROUTINE mrqcof(x,y,u,ndata,a,ia,ma,alpha,beta,nalp,chisq) INTEGER
ma,nalp,ndata,ia(1),MMAX

REAL*8 chisq,a(1),alpha(1,1),beta(1),x(136),y(136),u(136) PARAMETER
(MMAX=1)

INTEGER mfit,i,j,k,l,m REAL*8 dy,ymod,dyda(MMAX) mfit=0

Do 11 j=1,ma
if (ia(j).ne.0) mfit=mfit+1

11    Continue
```

```
Do 13 j=1,mfit Do 12 k=1,j

alpha(j,k)=0.

12       Continue beta(j)=0.

13    Continue chisq=0.
```

```
Do 16 i=1,ndata

call funcs(x(i),u(i),a,ymod,dyda,ma) dy=y(i)-ymod
j=0

Do 15 l=1,ma if(ia(l).ne.0) then

j=j+1

k=0

Do                 14 m=1,l if(ia(m).ne.0) then
k=k+1

alpha(j,k)=alpha(j,k)+dyda(m)*dyda(m) endif

14             Continue beta(j)=beta(j)+dy*dyda(l)
endif

15         Continue chisq=chisq+dy*dy
16      Continue

Do 18 j=2,mfit Do 17 k=1,j-1

alpha(k,j)=alpha(j,k)

17         Continue
18      Continue

return End

c-------gaussj--Subroutine for Gauss-Jordan elim SUBROUTINE
gaussj(covar,mfit,nca,da) INTEGER mfit,nca,NMAX

REAL*8 covar(1,1),da(1) PARAMETER (NMAX=1)

INTEGER i,icol,irow,j,k,l,ll,indxc(1),indxr(1),ipiv(NMAX) REAL*8
big,dum,pivinv

Do        11 j=1,mfit ipiv(j)=0
11      Continue

Do 22 i=1,mfit big=0.

Do 13 j=1,mfit if(ipiv(j).ne.1)then

Do 12 k=1,mfit if(ipiv(k).eq.0) then

if (abs(covar(j,k)).ge.big)then
big=abs(covar(j,k))

irow=j

icol=k endif
```

```
endif 12 Continue

endif

13          Continue ipiv(icol)=ipiv(icol)+1 if (irow.ne.icol) then

Do 14 l=1,mfit dum=covar(irow,l) covar(irow,l)=covar(icol,l)
covar(icol,l)=dum
```

```
14            Continue dum=da(irow) da(irow)=da(icol) da(icol)=dum

endif indxr(i)=irow indxc(i)=icol

if(covar(icol,icol).eq.0.) pause 'singular matrix in gaussj'
pivinv=1./covar(icol,icol)

covar(icol,icol)=1. Do 16 l=1,mfit

covar(icol,l)=covar(icol,l)*pivinv

16          Continue da(icol)=da(icol)*pivinv Do 21 ll=1,mfit
if(ll.ne.icol)then
dum=covar(ll,icol)

covar(ll,icol)=0. Do 18 l=1,mfit

covar(ll,l)=covar(ll,l)-covar(icol,l)*dum 18 Continue

da(ll)=da(ll)-da(icol)*dum endif
21          Continue

22      Continue

Do 24 l=mfit,1,-1 if(indxr(l).ne.indxc(1))then

Do 23 k=1,mfit dum=covar(k,indxr(l))

covar(k,indxr(l))=covar(k,indxc(l))
covar(k,indxc(l))=dum
23            Continue

endif 24 Continue

Return END

c-------funcs--subroutine for supplying function & derivatives
SUBROUTINE funcs(x,u,a,y,dyda,ma)
INTEGER ma

REAL*8 x,y,u,a(1),dyda(1),denom1,denom2,n1,n2,n3,d1,d2,d3,d4 REAL*8
d5,num,denom,k

INTEGER i i=1 k=0.920497

denom1=((x**2)-((2.75/a(i))**2))**3 denom2=(2.75**6)*((1/k)-
(((a(i)**2)-1)/(a(i)**2))**3) y=u-0.14016687/(denom1+denom2)
n1=(a(i)**2)*(x**4)*(2.75**2)

n2=2*(x**2)*(2.75**4) n3=(-2+(a(i)**2))*(2.75**6)
d1=(a(i)**4)*(2.75**6) d2=(a(i)**4)*(x**6)
d3=3*(a(i)**2)*(x**4)*(2.75**2) d4=3*(x**2)*(2.75**4) d5=(3-
(3*(a(i)**2))+(a(i)**4))*(2.75**6)
```

```
num=-0.14016687*(k**2)*(-6*(a(i)**3)*(n1-n2-n3)) denom=(d1+k*(d2-
d3+d4-d5))**2
```

```
dyda(i)=num/denom
Return

END
```