

```

c +-----+
c + subroutine check : quality control on observations +
c + obs = observed value +
c + int = interpolated value +
c + sdint = sqrt(eo**2+ei**2); eo-obs err, ei-int err +
c + dev = abs(obs-int)/sdint +
c + check : if dev > tol, skip obs +
c +-----+
subroutine check(nobs,co,coi,coired,valana,nskip,lskip,skip,
+ dev,flag,
+ w1,w2,w3,w4,sdevob,sdevfg,toleral,tolera2,valmean,iiiana)
integer nobs
real co(nobs,nobs),coi(nobs,nobs),coired(nobs,nobs)
real valana(1),dev(1)
integer nskip,lskip(1),iiiana(1),flag(1)
logical skip(1)
real w1(1),w2(1),w3(1),w4(1),sdevob,sdevfg,toleral,
x tolera2,valmean

do k=1,nobs
skip(k) = .false.
enddo
nskip = 0
20 dmax = 0.
do kskip=1,nobs
if( .not.skip(kskip) ) then
nskip = nskip + 1
lskip(nskip) = kskip
skip(kskip) = .true.
call redmat(nobs,coi,coired,nskip,lskip,w1,w2,w3,w4)
do k=1,nobs
w1(k) = co(k,kskip)
enddo
w1(kskip) = w1(kskip) - sdevob*sdevob
valmean = 0.
do k=1,nobs
if( .not.skip(k) ) valmean = valmean + valana(k)
enddo
valmean = valmean/float(nobs-nskip)
do k=1,nobs
w2(k) = 0.
do l=1,nobs
w2(k) = w2(k) + w1(l)*coired(k,l)
enddo
enddo
valint = valmean
eps = sdevfg*sdevfg + sdevob*sdevob
do k=1,nobs
valint = valint + w2(k)*( valana(k) - valmean )
eps = eps - w2(k)*w1(k)
enddo
if( eps.lt.0.1 ) eps = 0.1
sdint = sqrt(abs(eps))
dev(kskip) = abs( valana(kskip) - valint )/sqrt( eps )
if( dev(kskip).ge.dmax ) then
dmax = dev(kskip)
kdmax = kskip
endif
nskip = nskip - 1
skip(kskip) = .false.
endif
enddo
if( dmax.ge.tolera2 ) then
nskip = nskip + 1
lskip(nskip) = kdmax
skip(kdmax) = .true.
if( nskip.lt.nobs ) goto 20
endif
if(nskip.eq.0) then
do i=1,nobs
do j=1,i
coired(i,j) = coi(i,j)
if( i.ne.j ) coired(j,i) = coired(i,j)
enddo
enddo
else
call redmat(nobs,coi,coired,nskip,lskip,w1,w2,w3,w4)
endif
valmean = 0.
do k=1,nobs
if( .not.skip(k) ) valmean = valmean + valana(k)
enddo
valmean = valmean/float(nobs-nskip)
do k=1,nobs
if(skip(k))then
flag(k)=2
else
flag(k)=0
if(dev(k).gt.toleral) flag(k)=1
endif
enddo
return
end
c ----- subroutine redmat, only used by check.f -----
subroutine redmat(nobs,covinv,covred,nskip,lskip
x ,w1,w2,w3,w4)
integer nobs
real covinv(nobs,nobs),covred(nobs,nobs)
integer nskip,lskip(1)
real w1(nskip,nskip),w2(1),w3(nskip,nskip),w4(1)

```

```

if( nskip.eq.0 ) then
do i=1,nobs
do j=1,i
covred(i,j) = covinv(i,j)
if( i.ne.j ) covred(j,i) = covred(i,j)
enddo
enddo
else if( nskip.eq.1 ) then
kskip = lskip(1)
c = 1./covinv(kskip,kskip)
do i=1,nobs
do j=1,i
covred(i,j) = covinv(i,j)
& - covinv(i,kskip)*covinv(j,kskip)*c
if( i.ne.j ) covred(j,i) = covred(i,j)
enddo
enddo
else
do k=2,nskip
ks = lskip(k)
kml = k - 1
do l=1,kml
ls = lskip(l)
w1(k,l) = covinv(ls,ks)
w1(l,k) = covinv(ls,ks)
enddo
enddo
do k=1,nskip
ks = lskip(k)
w1(k,k) = covinv(ks,ks)
enddo

call linds(nskip,w1,nskip,w3,nskip,w4)
do i=1,nobs
do j=1,i
covred(i,j) = covinv(i,j)
enddo
enddo

do it=1,nskip
lt = lskip(it)
do is=1,nskip
ls = lskip(is)
do i=1,nobs
do j=1,i
covred(i,j) = covred(i,j)
& - covinv(i,lt)*covinv(j,ls)*w3(is,it)
enddo
enddo
enddo
enddo

do i=2,nobs
iml = i - 1
do j=1,iml
covred(j,i) = covred(i,j)
enddo
enddo
endif

return
end

```