

[View Forum Message](#) <> [Reply to Message](#)

(It is written to work with a function of any number of variables (curves or surfaces) because what you minimize is the χ^2 .)

It would be nice to have a back feed from people using it (drop me a line). I do not guarantee for bugs but it seem to work correctly.

[illegible][illegible]

Here it is:

FUNCTION funzione,par
common dati,image,f0

```
;here you define your function and call it f0
```

```
;then you pass the square of the difference between function and data (chi)
return,total((f0-image)^2)
END
```

```
pro simplex
common dati,image,f0
;read your data (curve or surface)
image=findgen(61,69)
openr,unit,'acampio.bin',/get lun
```

```
readu,unit,image
free_lun,unit
```

```
;the following if you want to see your data
```

```
;loadct,14
;window,/free,xsize=61,ysize=69
```

```
;tvsc1,image
```

```
nchiamate=1000      ;maximum number of calls
errmax=0.005        ;minimum difference between neighbour values
par=[,,,] ;initial guess for the parameters
step=par*0.1         ;initial steps
```

```
;the following to see your initial guess
;f=funzione(par)
;window,/free,xsize=61,ysize=69
;tvsc1,image
;window,/free,xsize=500,ysize=350
;surface,image,zrange=[0,50] & surface,f0,zrange=[0,50],color=12,/noerase

;stop
```

```
,*****Simplex Algorithm*****
```

```
A9=1. & B9=0.5 & C9=2.
```

```
buffer=size(par)
n=buffer(1)
n1=n+1
B=fltarr(n,n1)
y=fltarr(n1)
C=fltarr(n,4)
;builds up the initial Simplex points
for ip=0,n1-1 do B(*,ip)=par
for ip=0,n-1 do B(ip,ip)=par(ip)+step(ip)
```

```
;evaluates the function on the initial Simplex points
for ip=0,n1-1 do begin
    par=b(*,ip)
    y(ip)=funzione(par)
endfor
z9=n1
```

```
;determines the best (lowest) and worst (highest) values of the function
L8=min(y,L9)
H8=max(y,H9)
```

```

convergenza=max(abs(B(*,L9)-B(*,H9))/B(*,L9))

;minimization loop
WHILE (Z9 LT nchiamate AND (convergenza GE errmax OR convergenza EQ 0.0)) DO BEGIN

;determines the Simplex centroid exluding the worst point
C(*,1)=0.0
for ip=0,n1-1 do C(*,1)=C(*,1)+B(*,ip)
C(*,1)=(C(*,1)-B(*,H9))/n

C(*,2)=(1+A9)*C(*,1)-A9*B(*,H9) ;reflection
par=C(*,2)
riflesso=funzione(par) ;function evaluation at reflection
Z9=Z9+1

IF (riflesso LT L8) THEN BEGIN
;at reflection it's better -->tries to expand the simplex
C(*,3)=(1-C9)*C(*,1)+C9*C(*,2) ;extension
par=C(*,3)
estensione=funzione(par)
Z9=Z9+1
IF (estensione LT L8) THEN BEGIN
;at extension it's better -->starts from the beginning substituting extension to the worst point

B(*,H9)=par
y(H9)=estensione
ENDIF ELSE BEGIN
;at extension it's not better -->starts from the beginning substituting reflection to the worst
point
B(*,H9)=C(*,2)
y(H9)=riflesso
ENDELSE
ENDIF ELSE BEGIN
;at reflection it's better but...
IF (riflesso LT max(where(y LT H8))) THEN BEGIN
;it's better in respect to at least one point besides the worst
;-->starts from the beginning after having saved the reflection
;max(where(y LT H8)) it's the second worst

B(*,H9)=C(*,2)
y(H9)=riflesso
ENDIF ELSE BEGIN
;it's worse then the second worst (or equal)
;-->tries to shrink the simplex
IF (riflesso LT H8) THEN B(*,H9)=C(*,2) & y(H9)=riflesso
;reflection is between the worst and the second worst

```

```

;-->reflection is substituted to the worst
C(*,3)=(1-B9)*C(*,1)+B9*B(*,H9) ;contraction around the centroid
par=c(*,3)
contrazione=funzione(par)
Z9=Z9+1
IF (contrazione LE H8) THEN BEGIN
    ;contraction helps-->starts from the beginning after substituting it to the worst
    B(*,H9)=c(*,3)
    y(H9)=contrazione
ENDIF ELSE BEGIN
    ;contraction doesn't help-->it shrinks around the best point
    for ip=1,n1-1 do B(*,ip)=0.5*(B(*,ip)+B(*,L9))
    ;evaluates the function at the new simplex (shrunked)
    for ip=0,n1-1 do begin
        par=B(*,ip)
        y(ip)=funzione(par)
    endfor
    Z9=Z9+n1
ENDELSE
ENDELSE
ENDELSE

;determines the best (lowest) and worst (highest) values of the function
L8=min(y,L9)
H8=max(y,H9)
convergenza=max(abs(B(*,L9)-B(*,H9))/B(*,L9))

;output
print, Z9,convergenza,L8
print,B(*,L9),B(n-1,L9)!DtoR

ENDWHILE

;output
print,'number of calls', Z9,'convergence',convergenza
print, 'function value', L8
print, 'parameters',B(*,L9)

stop
END

```
