Subject: Simplex Algorithm
Posted by Pierpa on Tue, 28 Mar 1995 10:34:18 GMT
View Forum Message <> Reply to Message

Two months ago I promised to post the simplex algorithm as soon as I had time to improve it. Since I see that I don't find any time to do that, I'll send it as it is. I just put some comments in english.
It's not in the form of a function to call, it's a program and you have to write your function inside it. It would be nice if someone could find the time to make it a function.
The program originated from a basic program written by Caceci M.S. and Cacheris W.P. (Byte, 5,340-362 (1984)). The original version was full of goto statements and I restructured it with if statements. Furthermore I tried to avoid index looping as much as possible.
(It is written to work with a function of any number of variables (curves or surfaces) because what you minimize is the chi^2.)

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

<--<>--<>--<>--<>--<>--<>--<>--<>--<>--<>--<>--<>-- <>--<>--<>--<>-->

Stefano Polizzi |
Dipartimento di Chimica Fisica | tel: +39-41-5298618
Universita' di Venezia | FAX: +39-41-5298594
Dorsoduro 2137 | e-mail:polizzi@unive.it
I-30123 VENEZIA (Italy) |
<--<>--<>--<>--<>--<>--<>--<>--<>--<>--<>--<>--<>-- <>--<>--<>--<>-->

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

;tvscl,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
;tvscl,image
;window,/free,xsize=500,ysize=350
;surface,image,zrange=[0,50] & surface,f0,zrange=[0,50],color=12,/noerase

;stop

;*****************************Simplex Alghorithm*************************************

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 (shrinked)
            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
```