Multivariable Optimization for the HP-41C

by

Namir Shammas

This article presents an HP-41C program that performs multivariable. The program calculates the minimum of a multivariable by performing sequential optimization on each variable, using Newton's method.

Program OPTIM

Usage

XEQ OPTIM (same as pressing A)

A Program prompts you to enter N, tolerance for the optimization for each variable, the tolerance for the norm (calculated for the variables), and the initial guesses for the optimum values of the variables.

Program calculates and displays the value of the minimized function and the values for the variables at the minimum.

Example

Consider the following function:

$f(X) = (X(1) - 2)^2 + (X(2) - 3)^2 + (X(3) + 1)^2$

The following table lists the input values:

Input Variable	Value
Number of variables	3
Tolerance for optimizing each variable	1e-5
Tolerance for the norms	1e-5
Initial value for X(1)	1
Initial value for X(2)	1

Input Variable	Value
Initial value for X(3)	1

Using the above data, calculate the optimum function value and its coordinates. The Steps involved are:

Step	Task	Command/Input	Output
1	Switch to program mode and write the code for the targeted function under the label FX.	[PRGM] LBL "FX" … [RTN] [PRGM]	
2	Start the program.	XEQ "OPTIM"	N?
3	Enter the number of points.	3 [R/S]	TOLER?
4	Enter the tolerance used to optimize each variable.	1 [EEX] [CHS] 5 [R/S]	NORM TOLER?
5	Enter the norm tolerance.	1 [EEX] [CHS] 5 [R/S]	X1?
6	Enter the initial value for variable X(1).	1 [R/S]	X2?
7	Enter the initial value for variable X(2).	1 [R/S]	X3?
8	Enter the initial value for variable X(3).	1 [R/S]	(after showing intermediate values) FX=0.00000
9	View the optimum value for variable X(1)	[R/S]	X1=2.00000
10	View the optimum value for variable	[R/S]	X2=3.00000

Step	Task	Command/Input	Output
	X(2)		
11	View the optimum value for variable X(3)	[R/S]	X3=-1.00000

Algorithm

```
Input N, X(), Toler, NormToler

Calculate Norm2

Do

  For I =1 to N

    Do
      Xt = X(I)
      h = 0.01 *(1 + |Xt|)
      F0 = F(X)
      X(I) = Xt + h
      Fp = F(X)
      X(I) = Xt - h
      Fm = F(X)
      X(I) = Xt
      Diff = (Fp - Fm) / (Fp - 2 * F0 + Fm) * h / 2
      X(I) = X(I) - Diff
    Loop Until |Diff| <= Toler

  Next I

  Norm1 = Norm2
  Calculate Norm2

Loop Until |(Norm2 - Norm1)| <= NormToler

Return X() and F(X)
```

Memory Map

R00 = N
R01 = I
R02 = I used to calculate norm
R03 = Xt

R04 = h
R05 = F0
R06 = Fp
R07 = Fm
R08 = Toler
R09 = NormToler
R10 = Norm1
R11 = Norm2
R12 = available for user
...
R19 = available for user
R21 = X(1)
R22 = X(2)
R23 = X(3)

Source Code

The source code for the HP-41C program appears below. Please note the following:

Text appearing in a pair of double quotes represents characters in the Alpha register.

The |- characters represent the single *append character* for the Alpha register.

The blank lines are intentionally inserted to separate logical blocks of commands:

```
♦LBL "OPTIM"
♦LBL A
"N?"
PROMPT
STO 00                  # store the number of variables
XEQ 00
STO 01
"TOLER?"
CF 22
PROMPT
FC?C22
1E-5
STO 08                  # store the tolerance
"NORM TOLER?"
PROMPT
FC? 22
1E-5
```

```
STO 09              # store the norm tolerance
♦LBL 03             # start input loop
FIX 0
CF 29
"X"
XEQ 08
ARCL X
"|-?"
FIX 5
SF 29
PROMPT
STO IND 01
ISG 01
GTO 03              # end of input loop
XEQ 01
STO 11              # Calculate Norm2

♦LBL 04             # ---------- start main outer loop
XEQ 00
STO 01              # initialize For loop variable

♦LBL 05             # start of "For" loop

♦LBL 06             # initialize innermost loop
RCL IND 01
STO 03              # Xt = X(I)
ABS
1
+
0.01
*
STO 04               h = 0.01 * (1 + |X(I)|)
XEQ "FX"
STO 05              # F0 = F(X)
RCL 03
RCL 04
+
STO IND 01          # X(I) = Xt + h
```

```
XEQ "FX"
STO 06                  # Fp = F(X)
RCL 03
RCL 04
-
STO IND 01              # X(I) = Xt - h
XEQ "FX"
STO 07                  # Fm = F(X)
RCL 03
STO IND 01              # X(I) = Xt
RCL 06
STO Y
RCL 07
ST- Z
+
RCL 05
ST+ X
-
/
RCL 04
*
2
/                       # Calculate Diff
ST- IND 01
ABS
RCL 08
TONE 0
X<=Y?                   # Toler <= |Diff|
GTO 06                  # end of inner most loop

FIX 0
CF 29
XEQ 08
"X"
ARCL X
"|-:"
FIX 5
SF 19
```

```
ARCL IND 01
AVIEW
TONE 5

ISG 01
GTO 05                  # end of For loop

RCL 10
STO 11                  # Norm2 = Norm1
XEQ 01
STO 10                  # Calculate Norm1
RCL 11
-
ABS
RCL 09
X<=Y?                   # NormToler <= |Norm1 - Norm2|
GTO 04                  # ---------- end of main outer loop
BEEP
XEQ "FX"
"FX="
ARCL X
PROMPT                  # display optimum function value
XEQ 00
STO 01
♦LBL 07                 # start loop to display results
FIX 0
CF 29
"X"
XEQ 08
ARCL X
"|-="
FIX 5
SF 29
ARCL IND 01
PROMPT
ISG 01
GTO 07
RTN
```

```
♦LBL 00          # calculate bbb.eee for accessing array X()
21
ENTER
20
RCL 00
+
1E3
/
+
RTN

♦LBL 01          # calculate the norm
XEQ 00
STO 02
0
♦LBL 02          # start loop
RCL IND 02
X^2
+
ISG 02
GTO 02           # end loop
SQRT
RTN

♦LBL 08          # Calculate variable's index
RCL 01
INT
20
-
RTN

♦LBL FX          # function to be minimized
RCL 21
2
-
X^2
```

```
RCL 22
3
-
X^2
+
RCL 23
1
+
X^2
+
RTN
```