Scan Range Method for the HP-71B

By Namir Shammas

Introduction

Ever since HP launched programmable calculators, like the HP-65, HP-67, and the HP-25, it included root-seeking programs in its manuals, standard applications, and math applications. When HP released the HP-34C in 1978, it offered a built-in root *Solver*, for the first time. The Solver found a single root for a nonlinear function, given two guesses for (and near) a root. HP refined this Solver in later machines to accept an initial guess for and near a root.

This article presents the *Scan Range Method* and an HP-71B listing for a multi-root method. The method scans a user-specified range and using user-defined step sizes to examine multiple small intervals. The *Scan Range Method* finds roots and inflection points (minima, maxima, and saddle points) in the given range. The method relies on two basic algorithms:

- A root-seeking algorithm that locates a root in a sub-interval where the function values at the ends of that sub-interval change signs.
- An optimum-seeking algorithm that locates minima, maxima, and saddle points in sub-intervals where the slopes change signs at the ends of the sub-interval. These points can also double up as roots.

The Algorithm

Here is the pseudo-code for the Scan Range Method:

```
Initialize mechanism or structure used to report the results
Xa=A
Fa = f(Xa)
Da = d1(Xa)
SFa = sign of Fa
SDa = sign of Da
NumSteps = 0
Repeat
  Increment NumSteps
  Xb = A + NumSteps*StepSize
  Fb=f(Xb)
  Db=d1(Xb)
  SignFb = sign of Fb
  SignDb = sign of Db
  // Xb landed on a root?
  If Fb=0 Then
    Report and/or store Xb, Fb
    If SignDb and SignDa have opposite signs Then
      // Second derivative
      Drv2 = d2(Xb)
      If |Drv2|>=FxToler Then
        If Drv2 < 0 Then
          Report a maximum point
```



```
        Else
          Report a minimum point
        End
      Else
        Report a saddle point
      End
    End
  Else If SignFb and SignFa have opposite signs Then
    X = calculated root in [Xa,Xb] with tolerance Toler
    Report and/or store X, f(x)
  Else If (SignFa and SignFb have same values) AND
          (SignDa and SignDb do not have the same values) THEN
    X = calculated minima/maxima in [Xa,Xb] with tolerance Toler
    Report and/or store X, f(X)
    Drv2 = d2(X)
    If Drv2 < 0 Then
      Report a maximum point
    Else
      Report a minimum point
    End
  End

  If found a root, minimum, or maximum Then
    Increment NumSteps
    Xa = A + NumSteps*StepSize
    Fa = f(Xa)
    Da = d1(Xa)
    SignFa = sign of Fa
    SignDa = sign of Da
  Else
    Xa = Xb
    Fa = Fb
    Da = Db
    SignFa = SignFb
    SignDa = SignDb
  End
Until Xa>=B
Return accumulated information
```

HP-71B Listing

I will present two versions of the Scan Range Method programs. The first one uses the JPCROM and the second one dos not. The JPCROM supports structured constructs such as IF-THEN-ELSE, REPEAT-UNTIL, and so on. These constructs make it a bit easier to write a program with logically nested IF statements.

HP-71B Listing (Using the JPCROM)

Before I present the listing for the Scan Range Method, I will present a neo-BASIC unnumbered listing that uses variables with multiple names. The names of these variables begin with a $ character and are easier to read. I use a utility program to convert the neo-BASIC unnumbered listing into one that runs on the HP-71B. Here is the neo-BASIC unnumbered and indented listing:


```
DESTROY ALL
RADIANS
REM USER-DEFINED FUNCTION
DEF FNF(X) = EXP(X) - 3*X^2
REM
REM CALCULATE FIRST DERIVATIVE
DEF FNF1(X,F0)
H = 0.001*(1+ABS(X))
FNF1 = (FNF(X+H)-F0)/H
END DEF
REM
REM CALCULATE SECOND DERIVATIVE
DEF FNF2(X)
H = 0.001*(1+ABS(X))
F0=FNF(X)
F1=FNF(X+H)
F2=FNF(X-H)
FNF2 = (F1-2*F0+F2)/H^2
END DEF
REM
REM CALCULATE ROOT USING NEWTON'S METHOD
DEF FNN(X,T0)
REPEAT
  H = 0.001*(1+ABS(X))
  F0=FNF(X)
  D=H*F0/(FNF(X+H)-F0)
  X=X-D
UNTIL ABS(D)<T0
FNN=X
END DEF
REM
REM CALCULATE OPTIMUM USING NEWTON'S METHOD
DEF FNO(X,T0)
REPEAT
  H = 0.001*(1+ABS(X))
  F0=FNF(X)
  F1=FNF(X+H)
  F2=FNF(X-H)
  D1=(F1-F2)/2/H
  D2=(F1-2*F0+F2)/H^2
  D=D1/D2
  X=X-D
UNTIL ABS(D)<T0
FNO=X
END DEF
REM
REM
REM ------------ START -------------
REM
INPUT "ENTER A? ";A
INPUT "ENTER B? ";B
INPUT "ENTER STEP SIZE? ","0.1";$STEPSIZE
```



```
INPUT "ENTER TOLERANCE? ","1E-8";$TOLER
INPUT "ENTER FX TOLERANCE? ","1E-4";$FTOLER
$XA=A
$FA = FNF($XA)
$DA = FNF1($XA,$FA)
$SFA = SGN($FA)
$SDA = SGN($DA)
$NUMSTEPS = 0
REPEAT
  $NUMSTEPS = $NUMSTEPS + 1
  $BMOVEONEEXTRASTEP = 0
  $XB = A + $NUMSTEPS * $STEPSIZE
  $FB = FNF($XB)
  $DB = FNF1($XB, $FB)
  $SFB = SGN($FB)
  $SDB = SGN($DB)
  DISP "X @ ";$XB @ WAIT 1
  REM LANDED ON A ROOT??
  IF ABS($FB) = 0 THEN
    DISP "X=";$XB@ PAUSE
    DISP "FX=0"@ PAUSE
    $DRV2 = FNF2($XB)
    IF $SDA * $SDB < 0 THEN
      IF ABS($DRV2) > $FTOLER THEN
        IF $DRV2 > 0 THEN
          DISP "ROOT & MINIMUM" @ PAUSE
        ELSE
          DISP "ROOT & MAXIMUM" @ PAUSE
        END IF
      ELSE
        DISP "ROOT & SADDLE POINT"@ PAUSE
      END IF
    END IF
  END IF IF
  $BMOVEONEEXTRASTEP = 1
  REM ----------------
  REM ----------------
  REM FOUND A RANGE THAT CONTAINS A ROOT?
  IF $SFB * $SFA < 0 THEN
    X = FNN(($XA + $XB) / 2, $TOLER)
    DISP "X="; X @ PAUSE
    DISP "FX="; FNF(X)@ PAUSE
    $BMOVEONEEXTRASTEP = 1
  END IF
  REM ----------------
  REM ----------------
  REM LOCATED A RANGE THAT HAS A MINIMUM/MAXIMUM/ROOT?
  IF $SFA * $SFB > 0 AND $SDA * $SDB < 0 THEN
    X = FNO(($XA + $XB) / 2, $TOLER)
    $DRV2 = FNF2(X)
    REM FOUND A ROOT
    DISP "X="; X @ PAUSE
```



```
    $FX=FNF(X)
    DISP "FX="; $FX @ PAUSE
    $BMOVEONEEXTRASTEP = 1
    IF ABS($FX) < $FTOLER THEN
      DISP "ROOT/";
    ELSE
      DISP "";
    END IF
    IF $DRV2 > 0 THEN
      DISP "MINIMUM" @ PAUSE
    ELSE
      DISP "MAXIMUM" @ PAUSE
    END IF
  END IF
  REM ---------------
  REM ---------------
  IF $BMOVEONEEXTRASTEP = 1 THEN
    $NUMSTEPS = $NUMSTEPS + 1
    $XA = A + $NUMSTEPS * $STEPSIZE
    $FA = FNF($XA)
    $DA = FNF1($XA, $FA)
    $SFA = SGN($FA)
    $SDA = SGN($DA)
  ELSE
    $XA = $XB
    $FA = $FB
    $DA = $DB
    $SFA = $SFB
    $SDA = $SDB
  END IF
UNTIL $XA >= B
DISP "DONE!"
END
```

Here is the numbered BASIC listing:


```
10 REM Program scans a range of values to search for roots, minima,
20 REM and maximum of a function
30 REM Copyright (c) 2012 by Namir Shammas
40 REM Version 1.0.0D Deluxe Version Requires JPCROM
50 REM Last Update:
60 REM =============== VARIABLE NAMES TABLE ================
70 REM B0 <==> BMOVEONEEXTRASTEP
80 REM D0 <==> DA
90 REM D1 <==> DB
100 REM D2 <==> DRV2
110 REM F0 <==> FA
120 REM F1 <==> FB
130 REM F2 <==> FTOLER
140 REM F3 <==> FX
150 REM N0 <==> NUMSTEPS
160 REM S0 <==> SDA
170 REM S1 <==> SDB
180 REM S2 <==> SFA
190 REM S3 <==> SFB
200 REM S4 <==> STEPSIZE
220 REM T0 <==> TOLER
230 REM X0 <==> XA
240 REM X1 <==> XB
250 REM ======================================================
250 DESTROY ALL
260 RADIANS
270 REM USER-DEFINED FUNCTION
280 DEF FNF(X) = EXP(X) - 3*X^2
290 REM
300 REM CALCULATE FIRST DERIVATIVE
310 DEF FNF1(X,F0)
320 H = 0.001*(1+ABS(X))
330 FNF1 = (FNF(X+H)-F0)/H
340 END DEF
350 REM
360 REM CALCULATE SECOND DERIVATIVE
370 DEF FNF2(X)
380 H = 0.001*(1+ABS(X))
390 F0=FNF(X)
400 F1=FNF(X+H)
410 F2=FNF(X-H)
420 FNF2 = (F1-2*F0+F2)/H^2
430 END DEF
440 REM
450 REM CALCULATE ROOT USING NEWTON'S METHOD
460 DEF FNN(X,T0)
470 REPEAT
480 H = 0.001*(1+ABS(X))
490 F0=FNF(X)
500 D=H*F0/(FNF(X+H)-F0)
510 X=X-D
520 UNTIL ABS(D)<T0
```

```
530 FNN=X
540 END DEF
550 REM
560 REM CALCULATE OPTIMUM USING NEWTON'S METHOD
570 DEF FNO(X,T0)
580 REPEAT
590 H = 0.001*(1+ABS(X))
600 F0=FNF(X)
610 F1=FNF(X+H)
620 F2=FNF(X-H)
630 D1=(F1-F2)/2/H
640 D2=(F1-2*F0+F2)/H^2
650 D=D1/D2
660 X=X-D
670 UNTIL ABS(D)<T0
680 FNO=X
690 END DEF
700 REM
710 REM
720 REM ------------ START -------------
730 REM
740 INPUT "ENTER A? ";A
750 INPUT "ENTER B? ";B
760 INPUT "ENTER STEP SIZE? ","0.1";S4
770 INPUT "ENTER TOLERANCE? ","1E-8";T0
780 INPUT "ENTER FX TOLERANCE? ","1E-4";F2
790 X0=A
800 F0 = FNF(X0)
810 D0 = FNF1(X0,F0)
820 S2 = SGN(F0)
830 S0 = SGN(D0)
840 N0 = 0
850 REPEAT
860 N0 = N0 + 1
870 B0 = 0
880 X1 = A + N0 * S4
890 F1 = FNF(X1)
900 D1 = FNF1(X1, F1)
910 S3 = SGN(F1)
920 S1 = SGN(D1)
925 DISP "X @ ";X1 @ WAIT 1
930 REM LANDED ON A ROOT??
940 IF ABS(F1) = 0 THEN
950 DISP "X=";X1@ PAUSE
960 DISP "FX=0"@ PAUSE
970 D2 = FNF2(X1)
980 IF S0 * S1 < 0 THEN
990 IF ABS(D2) > F2 THEN
1000 IF D2 > 0 THEN
1010 DISP "ROOT & MINIMUM" @ PAUSE
1020 ELSE
1030 DISP "ROOT & MAXIMUM" @ PAUSE
```



```
1040 END IF
1050 ELSE
1060 DISP "ROOT & SADDLE POINT"@ PAUSE
1070 END IF
1080 END IF
1090 B0 = 1
1100 END IF
1110 REM ----------------
1120 REM ----------------
1130 REM FOUND A RANGE THAT CONTAINS A ROOT?
1140 IF S3 * S2 < 0 THEN
1150 X = FNN((X0 + X1) / 2, T0)
1160 DISP "X="; X @ PAUSE
1170 DISP "FX="; FNF(X)@ PAUSE
1180 B0 = 1
1190 END IF
1200 REM ----------------
1210 REM ----------------
1220 REM LOCATED A RANGE THAT HAS A MINIMUM/MAXIMUM/ROOT?
1230 IF S2 * S3 > 0 AND S0 * S1 < 0 THEN
1240 X = FNO((X0 + X1) / 2, T0)
1250 D2 = FNF2(X)
1260 REM FOUND A ROOT?
1270 DISP "X="; X @ PAUSE
1280 F3=FNF(X)
1290 DISP "FX="; F3 @ PAUSE
1300 B0 = 1
1310 IF ABS(F3) < F2 THEN
1320 DISP "ROOT/";
1330 ELSE
1340 DISP "";
1350 END IF
1360 IF D2 > 0 THEN
1370 DISP "MINIMUM" @ PAUSE
1380 ELSE
1390 DISP "MAXIMUM" @ PAUSE
1400 END IF
1410 END IF
1420 REM ----------------
1430 REM ----------------
1440 IF B0 = 1 THEN
1450 N0 = N0 + 1
1460 X0 = A + N0 * S4
1470 F0 = FNF(X0)
1480 D0 = FNF1(X0, F0)
1490 S2 = SGN(F0)
1500 S0 = SGN(D0)
1510 ELSE
1520 X0 = X1
1530 F0 = F1
1540 D0 = D1
1550 S2 = S3
```



```
1560 S0 = S1
1570 END IF
1580 UNTIL X0 >= B
1590 DISP "DONE!"
1600 END
```

HP-71B Listing

I created the regular BASIC listing from the one that uses the JPCROM. I edited the IF statements and replaced the REPEAT-UNTIL loops with simple IF statements. Here is the regular version of the HP-71B BASIC listing:


```
10 REM Program scans a range of values to search for roots, minima,
20 REM and maximum of a function
30 REM Copyright (c) 2012 by Namir Shammas
40 REM Version 1.0.0R Regular Version
50 REM Last Update:
60 REM =============== VARIABLE NAMES TABLE ================
70 REM B0 <==> BMOVEONEEXTRASTEP
80 REM D0 <==> DA
90 REM D1 <==> DB
100 REM D2 <==> DRV2
110 REM F0 <==> FA
120 REM F1 <==> FB
130 REM F2 <==> FTOLER
140 REM F3 <==> FX
150 REM N0 <==> NUMSTEPS
160 REM S0 <==> SDA
170 REM S1 <==> SDB
180 REM S2 <==> SFA
190 REM S3 <==> SFB
200 REM S4 <==> STEPSIZE
220 REM T0 <==> TOLER
230 REM X0 <==> XA
240 REM X1 <==> XB
250 REM ================================================250
DESTROY ALL
260 RADIANS
270 REM USER-DEFINED FUNCTION
280 DEF FNF(X) = EXP(X) - 3*X^2
290 REM
300 REM CALCULATE FIRST DERIVATIVE
310 DEF FNF1(X,F0)
320 H = 0.001*(1+ABS(X))
330 FNF1 = (FNF(X+H)-F0)/H
340 END DEF
350 REM
360 REM CALCULATE SECOND DERIVATIVE
370 DEF FNF2(X)
380 H = 0.001*(1+ABS(X))
390 F0=FNF(X)
400 F1=FNF(X+H)
410 F2=FNF(X-H)
420 FNF2 = (F1-2*F0+F2)/H^2
430 END DEF
440 REM
450 REM CALCULATE ROOT USING NEWTON'S METHOD
460 DEF FNN(X,T0)
470 REM REPEAT
480 H = 0.001*(1+ABS(X))
490 F0=FNF(X)
500 D=H*F0/(FNF(X+H)-F0)
510 X=X-D
520 IF ABS(D)>=T0 THEN 470
```

```
530 FNN=X
540 END DEF
550 REM
560 REM CALCULATE OPTIMUM USING NEWTON'S METHOD
570 DEF FNO(X,T0)
580 REM REPEAT
590 H = 0.001*(1+ABS(X))
600 F0=FNF(X)
610 F1=FNF(X+H)
620 F2=FNF(X-H)
630 D1=(F1-F2)/2/H
640 D2=(F1-2*F0+F2)/H^2
650 D=D1/D2
660 X=X-D
670 IF ABS(D)>=T0 THEN 580
680 FNO=X
690 END DEF
700 REM
710 REM
720 REM ------------ START -------------
730 REM
740 INPUT "ENTER A? ";A
750 INPUT "ENTER B? ";B
760 INPUT "ENTER STEP SIZE? ","0.1";S4
770 INPUT "ENTER TOLERANCE? ","1E-8";T0
780 INPUT "ENTER FX TOLERANCE? ","1E-4";F2
790 X0=A
800 F0 = FNF(X0)
810 D0 = FNF1(X0,F0)
820 S2 = SGN(F0)
830 S0 = SGN(D0)
840 N0 = 0
850 REM REPEAT
860 N0 = N0 + 1
870 B0 = 0
880 X1 = A + N0 * S4
890 F1 = FNF(X1)
900 D1 = FNF1(X1, F1)
910 S3 = SGN(F1)
920 S1 = SGN(D1)
925 DISP "X @ ";X1 @ WAIT 1
930 REM LANDED ON A ROOT??
940 IF ABS(F1) # 0 THEN 1140
950 DISP "X=";X1@ PAUSE
960 DISP "FX=0"@ PAUSE
970 D2 = FNF2(X1)
980 IF S0 * S1 >= 0 THEN 1080
990 IF ABS(D2) <= F2 THEN 1060
1000 IF D2 <= 0 THEN 1030
1010 DISP "ROOT & MINIMUM" @ PAUSE
1020 GOTO 1090
1030 DISP "ROOT & MAXIMUM" @ PAUSE
```



```
1040 GOTO 1090
1060 DISP "ROOT & SADDLE POINT"@ PAUSE
1090 B0 = 1
1100 REM END IF
1110 REM ----------------
1120 REM ----------------
1130 REM FOUND A RANGE THAT CONTAINS A ROOT?
1140 IF S3 * S2 >= 0 THEN 1230
1150 X = FNN((X0 + X1) / 2, T0)
1160 DISP "X="; X @ PAUSE
1170 DISP "FX="; FNF(X)@ PAUSE
1180 B0 = 1
1190 END IF
1200 REM ----------------
1210 REM ----------------
1220 REM LOCATED A RANGE THAT HAS A MINIMUM/MAXIMUM/ROOT?
1230 IF NOT(S2 * S3 > 0 AND S0 * S1 < 0) THEN 1440
1240 X = FNO((X0 + X1) / 2, T0)
1250 D2 = FNF2(X)
1260 REM FOUND A ROOT?
1270 DISP "X="; X @ PAUSE
1280 F3=FNF(X)
1290 DISP "FX="; F3 @ PAUSE
1300 B0 = 1
1310 IF ABS(F3) >= F2 THEN 1360
1320 DISP "ROOT/";
1330 GOTO 1360
1340 DISP "";
1360 IF D2 > 0 THEN DISP "MINIMUM" ESLE DISP "MAXIMUM"
1370 PAUSE
1420 REM ----------------
1430 REM ----------------
1440 IF B0 = 0 THEN 1520
1450 N0 = N0 + 1
1460 X0 = A + N0 * S4
1470 F0 = FNF(X0)
1480 D0 = FNF1(X0, F0)
1490 S2 = SGN(F0)
1500 S0 = SGN(D0)
1510 GOTO 1580
1520 X0 = X1
1530 F0 = F1
1540 D0 = D1
1550 S2 = S3
1560 S0 = S1
1570 END IF
1580 IF X0 < B THEN 850
1590 DISP "DONE!"
1600 END
```


Sample Session

Here is a sample session that finds the roots and inflection points for the function:

f(x) = exp(x) – 3*x^2

The sample session searches for the roots and inflection points using the following input data:

- The range of [-1, 4].
- The search step size of 0.1.
- The tolerance value of 1e-8.
- The function tolerance value of 1e-4

Display	*Input/Command*
	[RUN]
ENTER A?	-1[ENDLINE]
ENTER B?	4[ENDLINE]
ENTER STEP SIZE?	0.1[ENDLINE]
ENTER TOLERANCE?	1E-8[ENDLINE]
ENTER FX TOLERANCE?	1E-4[ENDLINE]
X=-.458962267536	[CONT]
FX= .000000000003	[CONT]
X= .204481512364	[CONT]
FX= 1.10145070666	[CONT]
MAXIMUM	[CONT]
X= .910007572493	[CONT]
FX=-.00000000001	[CONT]
X= 2.83314410892	[CONT]
FX=-7.0812935823	[CONT]
MINIMUM	[CONT]
X= 3.73307902865	[CONT]
FX= .0000000002	[CONT]
DONE!	

