Lagrangian Interpolation for the HP67

by

Namir Shammas

This article presents an HP-67 program for Langrangian Interpolation of N points, where 1 < N < 9.

Usage

A	**Program prompts you to enter N and the N points**
B	**Program prompts you to enter the interpolated value of x**

Program calculates and displays the value of the interpolated Y.

Example

Consider the data in the following table:

X	Y
1	1
5	25
10	100

Using the above data, calculate Y for X = 4. The Steps involved are:

Step	Task	Command/Input	Output
1	Start the program.	3 [A]	1.0000
2	Enter the first data point.	1 [ENTER]1 [R/S]	2.0000
3	Enter the second data point.	25 [ENTER]5 [R/S]	3.0000
4	Enter the third data point.	100 [ENTER]10[R/S]	4.0000

Step	Task	Command/Input	Output
5	Start the interpolation.	4 [B]	16.0000

Algorithm

```
INPUT N, array X(1..N), Y(1..N), and Xint
Yint = 0
FOR I = 1 TO N
  Product = Y(I)
  FOR J = 1 to N
    IF I <> J THEN
      Product = Product * (Xint - X(J)) / (X(I) - X(J))
    ENDIF
  NEXT J
  Yint = Yint + Product
NEXT I
Show Yint
```

Memory Map

```
RA = Xint
RB = Yint
RC =
RD = X(I)
RE = X(J)

R0 = N
R1 = I
R2 = J
R3 = Product
R4 = X(1)
R5 = X(2)
R6 = X(3)
R7 = X(4)
R8 = X(5)
R9 = X(6)

SR0 = X(7)
SR1 = X(8)
SR2 = Y(1)
```

```
SR3 = Y(2)
SR4 = Y(3)
SR5 = Y(4)
SR6 = Y(5)
SR7 = Y(6)
SR8 = Y(7)
SR9 = Y(8)
```

Source Code

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

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

```
LBL A
STO 0       # store N, the number of points
1
STO 1
LBL 3       # Start loop for data input
-x-         # Blink X reg to prompt for next point
R/S         # Get Y /^ X
RCL 1
GSB 1       # Get index for X(I)
ST I
RDN
STO (i)
RDN
RCL 1
GSB 2       # Get index for Y(I)
ST I
RDN
STO (i)
1
STO + 1
RCL 0
RCL 1
X<=Y?
GTO 3       # end of loop
RTN

LBL 1       # Get Index for X(I)
3
+
RTN
```

```
LBL 2      # Get Index for Y(I)
11
+
RTN

LBL B
STO A      # Store Xint
0
STO B      # Yint = 0
1
STO 1      # I = 1
LBL 4      # Start outer loop
RCL 1
GSB 2      # Get index for Y(I)
ST I
RCL (i)
STO 3      # Product = Yint

RCL 1
GSB 1      # Get index for X(I)
ST I
RCL (i)
STO D      # Store X(I) in register D
1
STO 2      # J = 1
LBL 5      # start inner loop
RCL 1
RCL 2
X=Y?       # Skip calculation step?
GTO 6
RCL 2
GSB 1      # Get index for X(J)
ST I
RCL (i)
STO E      # Store X(I) in register E
RCL A
RCL E
-
RCL D
RCL E
-
/
STO* 3
LBL 6      # Jump here when I = J
1
STO+ 2     # J = J + 1
```

```
RCL 0
RCL 2
X<=Y?
GTO 5       # end of inner loop
RCL 3
RCL B
+
STO B       # Yint = Yint + Product
1
STO+ 1      # I = I + 1
RCL 0
RCL 1
X<=Y?
GTO 4       # end of outer loop
RCL B       # return Yint
RTN
```