UNITED ARAB REPUBLIC

THE INSTITUTE OF NATIONAL PLANNING

Memo. No. 406

Notes on Interpolation Formulae

By

Dr. S. Hamid

and

Fayza H. Omar

CAIRO

Part I.

The Lagrangian Interpolation Formula.

Section 1. Introduction:

Assume we are given n+1 values of a dependent variable "f" corresponding to n+1 values of independent variable "a". Denote these variables respectively

$$f(a_j)\ ,\quad a_j \qquad \text{for } j=1,2,\ldots,n+1. \qquad \ldots (1)$$

Then it is possible to construct a polynomial of degree n.

$$f(x)=A_0 + A_1\ x + A_2\ x^2 + \ldots + A_n\ x^n. \qquad \ldots (2)$$

Such that:

$$f(x)= f(a_j) \quad \text{at} \quad x = a_j. \qquad \ldots (3)$$

For numerical application, a more convenient form for the polynomial f(x) is needed, which is derived in the following section.

Section 2. The Lagrangian Interpolation Formula:

From equations 2 & 3 we get the following n+1 equations

$$f(a_j) = A_0 + A_1\ a_j + A_2\ a_j^2 + \ldots + A_n\ a_j^n\ .$$

$$j = 1,2, \ldots,n+1. \qquad \ldots (4)$$

The system of n+2 equations given by equation (2) & (4) are consistent if & only if

$$\begin{vmatrix} f(x) & 1 & x & x^2 & \cdots & x^n \\ f(a_1) & 1 & a_1 & a_1^2 & \cdots & a_1^n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f(a_j) & 1 & a_j & a_j^2 & \cdots & a_j^n \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f(a_{n+1}) & 1 & a_{n+1} & a_{n+1}^2 & \cdots & a_{n+1}^n \end{vmatrix} = 0. \qquad \ldots(5)$$

Expanding the above determinant columnwise we have

$$f(x).\ \Delta_o = \sum_{j=1}^{n+1} f(a_j).\ \Delta_j(x).\ (-1)^{j+1}. \qquad \ldots(6)$$

Where

$$\Delta_o = \begin{vmatrix} 1 & a_1 & a_1^2 & \cdots & a_1^n \\ 1 & a_2 & a_2^2 & \cdots & a_n^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_j & a_j^2 & \cdots & a_j^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_{n+1} & a_{n+1}^2 & \cdots & a_{n+1}^n \end{vmatrix} \qquad \ldots(7)$$

and Δ_j (x) is the [illegible] in the determinent given in the right-hand side of equation 6.

For example:

$$\Delta_3 = \begin{vmatrix} 1 & x & x^2 & \ldots & x^n \\ 1 & a_1 & a_1^2 & \ldots & a_1^n \\ 1 & a_2 & a_2^2 & \ldots & a_2^n \\ 1 & a_4 & a_4^2 & \ldots & a_4^n \\ 1 & a_5 & a_5^2 & \ldots & a_5^n \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & a_{n+1} & a_{n+1}^2 & \ldots & a_{n+1}^n \end{vmatrix} \qquad \ldots (8)$$

Notice that Δ_j (x) is a polynomial of degree n, and hence can in general be expressed in the form

$$\Delta_j(x) = C_j(x - c_1)(x - c_2) \ldots (x - c_n). \qquad \ldots (9)$$

From the determinantal definition of Δ_j (x), it is seen that

$$\Delta_j(x) = 0 \qquad \ldots (10)$$

at the n values of x given by

$$x = a_i \qquad i \neq j \qquad \ldots (11)$$

From (9) , (10) & (11) we can put

$$\begin{aligned} c_i &= a_i & i &< j \\ c_i &= a_{i+1} & i &\geqslant j \end{aligned} \qquad \ldots (12)$$

Or in other words

$$\Delta_j(x) = \frac{C_j}{(x-a_j)} \cdot \prod_{j=1}^{n+1} (x - a_j) \qquad \ldots (13)$$

$$= \frac{C_j}{(x-a_j)} \cdot P_n(x) \qquad \ldots (14)$$

where

$$P_n(x) = \prod_{j=1}^{n+1} (x - a_j) \qquad \ldots (15)$$

Consider the determinant Δ_0. By analogy it can be expanded in a similar way as $\Delta_j(x)$ and this can take the following form:

$$\Delta_0 = C_j \prod_{i=1)j(n+1} (a_j - a_i) \qquad \ldots (16)$$

where i = 1) j (n+1 stands for $i = 1, 2, \ldots, j-1, j+1, \ldots, n+1$.
From the above definition of $p_n(x)$ it can be seen that

$$\Delta_0 = C_j \left[\frac{d\,P_n(x)}{dx} \right]_{x = a_j} = C_j\ P_n'(a_j) \qquad \ldots (17)$$

From (14) & (17) we can put

$$L_j(x) = \frac{\Delta_j(x)}{\Delta_0} = \frac{1}{(x-a_j)} \cdot \frac{P_n(x)}{P_n'(a_j)} \qquad \ldots (18)$$

Substituting the above equation in equation (6) defining the function f(x), we obtain the following convenient expression for the function f(x).

$$f(x) = \sum_{j=1}^{n+1} L_j(x)\ f(a_j) \qquad \ldots (19)$$

This formula is the wellknown "Lagrangian Interpolation Formula".(1)

Section 3 : Computation procedure:

The above Lagrangian interpolation formula had been programmed on the I.B.M. 1620 using Fortran Language.

Section 3.1 gives the symbols used in the Fortran program and the way the data must be prapared.

Section 3.2 gives the block diagram.

Section 3.3 gives the program itself in the Fortran Language.

Section 3.4 gives a numerical example.

The program available can handle interpolation by Lagrangian formula provided the number of points at which the function is given is less or equal to 100. If we have more points, equal in value to the integer J1, then we must change the first dimension statment in the source program to the following statment

```
DIMENSION  A(J1), F(J1)
```

and of course we have to compile the program again to get the object program.

---

1) See Kopal, Zdenek
Numerical analysis, 2nd. ed. N.Y., Wiley, 1961.

Section 3.1 : Symbols used in the program and the preparation of data.

The formula is

$$f(x) = \sum_{j=1}^{n+1} L_j(x)\, f(a_j)$$

where

$$L_j(x) = \frac{(x-a_1)\,(x-a_2)\,\ldots\,(x-a_{j-1})(x-a_{j+1})\ldots(x-a_{n+1})}{(a_j-a_1)(a_j-a_2)\ldots(a_j-a_{j-1})(a_j-a_{j+1})\ldots(a_j-a_{n+1})}$$

Symbols in Theory	Corresponding Symbols in Fortran
$n+1$	N1
j	J
a_j	A(J)
$f(a_j)$	F(J)
x	X
$f(x)$	FX

The preparation of data:

The first card must contain the number N1 of values at which the function is given and the code of the process as following:

N1	:	Colomn	1→3	xxx	integer form.
Code	:	Colomn	4→10	xx.xxxx	floating form.

After the first card we have N1 cards each corresponding to one given value of the function.

The data in these N1 cards are as following:

J	: colomn	1→3	xxx	integer form
A(J)	: colomn	4→17	±x.xxxxxxxE±xx	E form
F(J)	: colomn	18→31	±x.xxxxxxxE±xx	E form

After the N1 cards, we have cards, each corresponding to the value of the argument X at which we want to determine our function.

The data in these cards are as following:

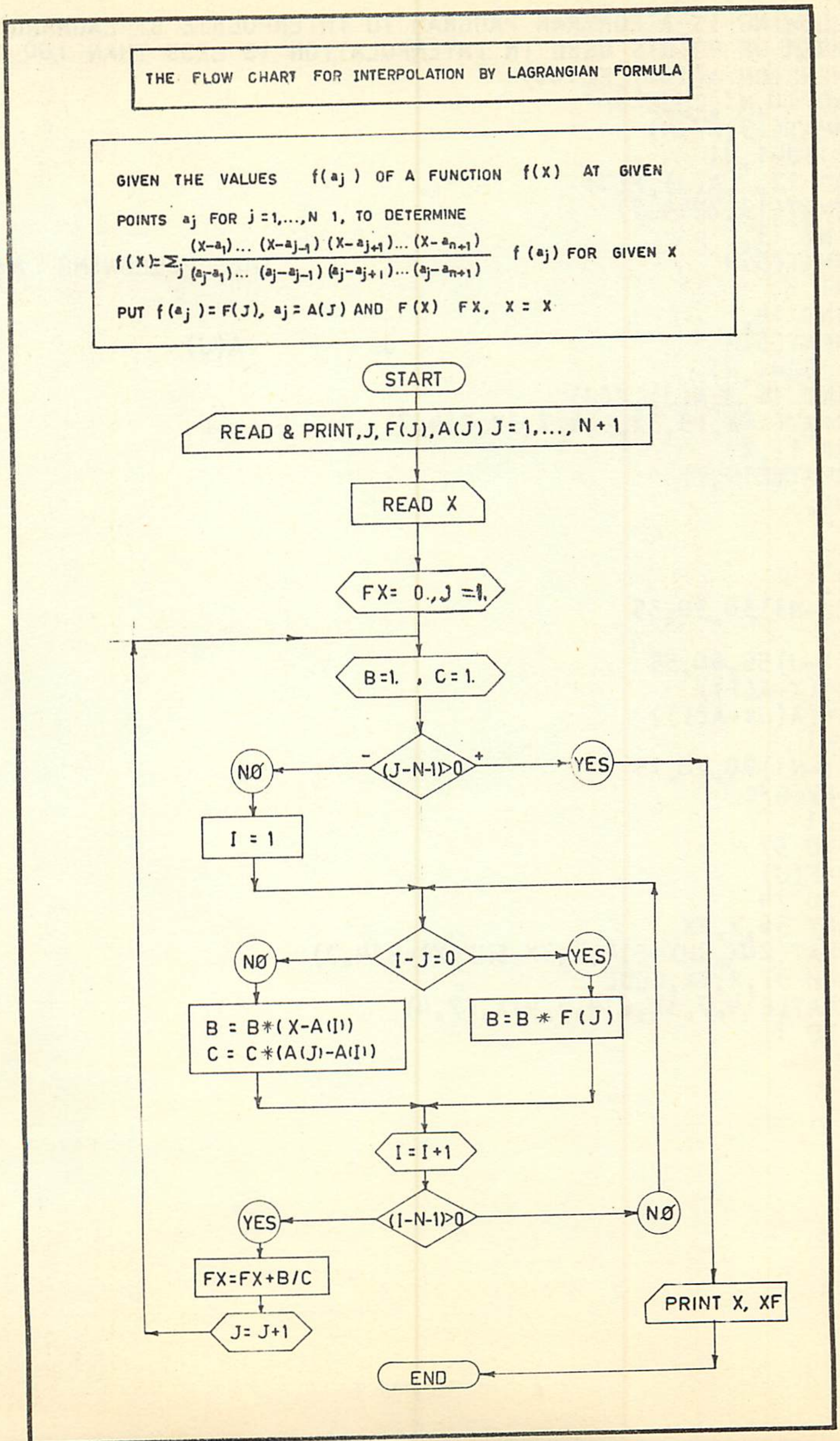
X : colomn 1→14 ±x.xxxxxxxE±xx E form

So if our problem is to interpolation for M1 values, given N1 values of the function, then our data cards will be

1 + N1 + M1 .

Section 3.2

Block diagram
of Lagrangian.



```
C THE FOLLOWING IS A FORTRAN PROGRAM TO INTERPOLATE BY LAGRANGIAN FORMUL
C THE NUMBER OF POINTS USED IN INTERPOLATION IS LESS THAN 100
      DIMENSION A(100),F(100)
      READ 10,N1,CODE
   10 FORMAT(I3,F7.4)
      DO 11J=1,N1
   11 READ 12,J,A(J),F(J)
   12 FORMAT(I3,2E14.7)
      PRINT 13,
   13 FORMAT(59H                                        THE  FOLLOWING  ARE  THE  D
     XATA)
      PRINT 14,
   14 FORMAT(52H                            J          A(J)              F(J))
      DO 15J=1,N1
   15 PRINT 16,J,A(J),F(J)
   16 FORMAT(20X,I3,3X,E14.7,3X,E14.7)
    1 READ 17,X
   17 FORMAT(E14.7)
      FX=0.
      J=1
   57 B=1.
      C=1.
      IF(J-N1)30,30,35
   30 I=1
   70 IF(I-J)55,60,55
   55 B=B*(X-A(I))
      C=C*(A(J)-A(I))
   74 I=I+1
      IF(I-N1)70,70,75
   75 FX=FX+B/C
      J=J+1
      GO TO 57
   60 B=B*F(J)
      GO TO 74
   35 PRINT 36,X,FX
   36 FORMAT(20X,2HX=E14.7,3X,5HF(X)=E14.7)
      PUNCH 37,X,FX,CODE
   37 FORMAT(E14.7,3X,E14.7,41X,F7.4)
      GO TO 1
      END
```

THE FOLLOWING ARE THE DATA

J	A(J)	F(J)
1	1.0000000E+00	0.0000000E-99
2	9.0380000E-01	2.2030000E-01
3	8.0920000E-01	4.2130000E-01
4	7.2870000E-01	5.7930000E-01
5	6.6790000E-01	6.7560000E-01
6	5.8470000E-01	7.6730000E-01
7	4.8290000E-01	8.5650000E-01
8	3.7100000E-01	9.2660000E-01
9	2.4800000E-01	9.7180000E-01
10	7.6500000E-02	9.9450000E-01

X= 5.0000000E-01 F(X)= 8.4171143E-01

Part II.

Aitken's Interpolation Formula.

Section 1. Introduction:

Aitken's Algorithim for the Lagrangian Interpolation Formula:

There is an algorithim, due to Aitken, for computing Lagrangian interpolation formula, which is suitable for digital computers & even desk calculators. Given the function $f(a_j)$ at the n+1 points a_j $(j=1,2,\dots,n+1)$, Aitken's algorithim for computing the lagrangian interpolation formula $f(x)$ whose numerical values at that is satisfied at the point $x=a_j (j=1,\dots, n+1)$ coincides with the values $f(a_j)$, proceeds as following:

The Lagrangian interpolation formula of degree r :

$$f(x \mid a_1, a_2, \dots, a_r, a_t) \qquad n+1 \geqslant t \geqslant r+1 \qquad \dots (1)$$

which numerically is equal to the function $f(x)$ at the points:

$$x=a_1, a_2, \dots, a_r, a_t. \qquad \dots (2)$$

can be obtained from the two Lagrangian interpolation formulae of degree r-1:

$$f(x \mid a_1, a_2, \dots, a_{r-1}, a_r) \qquad \dots (3)$$

$$\& \qquad f(x \mid a_1, a_2, \dots, a_{r-1}, a_t) \qquad \dots (4)$$

that passes respectively through the two following sets of points:

$$a_1, a_2, \dots, a_{r-1}, a_r . \qquad \dots (5)$$

$$\& \qquad a_1, a_2, \dots, a_{r-1}, a_t . \qquad \dots (6)$$