UNITED ARAB REPUBLIC

THE INSTITUTE OF NATIONAL PLANNING

التخطيط القومي

Memo. No. 542

APPLICATION OF LINEAR PROGRAMMING TO CURVE FITTING AND THEORY OF APPROXIMATION

BY

M.S. EL-HADDAD

Under the supervision of

Dr. ROSHDI AMER

January 1965

Operations Research Center

CAIRO

Acknowledgement

I wish to express my deep thanks to DR. ROSHDI AMER the supervisor of this thesis to whom I am indebted for useful critical comments, improvements and in collecting better examples from the real fields of application.

My great thanks are to DR. SALAH HAMID the Director of the Operations Research Center for his help and encourgements.

Finally my thanks are due to Miss. Elen Zaki & Miss. Nadia Amer and Mrs. Malaka Mourad for expert typing help and drawing the accompanied figures.

Contents.

Introduction

One of the most frequent mathematical problems that ecounter research workers is the problem of determing a suitable function which is required to fit a physical or a mathematical relation between two variables; the relation may be obtained either by the results of some measurements or through complicated calculations. This is the well known "curve fitting problem".

This classical methods for treating this problem are either by interpolating some of the measured points, which is liable to give doubtful results, or by approximating the whole set of points using the principle of the least error squares.

A much better principle for approximation is to minimize the maximum possible error, which is known as "Tchebycheff approximation", whose only disadvantage is that the determination of the approximating function will envolve much more complicated calculations. In the age of electronic computers, this difficulty should not prevent us from obtaining the better results of the "Tchebycheff approximation."

A closely related problem to that of the "curve fitting" is the problem of obtaining a solution for an over-determinate system of equations. Again, this can be done according to either the least squar method or by using the principle of

"Tchebycheff approximation"

In this research work the solution of both problems (the "curve fitting problem" and the over-determinate system of equations) is studied. A theoretical treatment of the least square method for solving both problems is given. The technique of transforming these problems to a linear programming problem is exposed in details, together with the necessary computer programs.

It is hoped that this research will provide a useful tool to research workers who are in need of the techniques of curve fitting or Tchebycheff approximation.

Cairo 28.10.64

Roshdi Amer

CHAPTER 1.

Curve Fitting

1.0. Structure of curve fitting problem and its meaning

Given a set of values for a pair of quantities (x_i, y_i) concerning a given phenomenon and supposing that the relation between these quantities can be written in the form.

$$y = f(x) ,$$

it is required to determine the function f(x) on the basis of the set of values of the pair of quantities (x_i, y_i). These values may either be exact or may show random errors due to the fact that they are measured statistical value concerning the given phenomenon which shows random fluctuation. However, the points here can be represented in an (x,y) corrdinate system and we have to indicate a curve which lies optimally between these points. This procedure is called curve fitting. It is clear that we need some principle for optimum choice. The most popular principle of curve fitting is the so-called method of least squares. Another principle which had been carried out by Tchebycheff, can be applied by using the approach of linear programming which we shall try to do in details.

Choice of the proper method of approxiamtion

When fitting a curve to a cluster of sample points, i.e. in case of statistical data, the T-approximation can not be used at all. The only approximation to be used is the least squares approxiamtion since the distribution of the given statistical data is always Gaussian. However, on fitting a curve, the fitting of a polynomial is advantageous from the point of view of numerical work required and therefore other types of curves are used only in the case of good theoretical reasons.

Extrapolation

The fitted curve yields a good approxiamtion of the theoretical curve only within the domain of the given data. Actually, the extrapolation is of doutful value in approximating by using the least squares method while it leads to erronous results when using the T-approximation.

In the following section, we shall briefly explain the method of least squares.

1.1. Principle of least squares

According to this principle and all other principles, the form of the function f(x) (i.e. polynomial, exponential function etc.) is assumed to be known and the problem consists in determining the values of the unkown parameters; i.e.

$$f(x) = f(x; \xi_0, \xi_1, \dots, \xi_n)$$

These parameters are to be chosen in such a way that

$$S = \sum_{i=1}^{n} \left[y_i - f(x_i ; \xi_0, \xi_1, \dots, \xi_n) \right]^2$$

should assume a minimum value. In other words, the sum of the squares of the vertical distances (i.e. residuals) between the observed values y_i and the theoretical values $f(x_i)$ must be as small as possible

The solution of this problem implies the following system of equations.

$$\frac{\partial s}{\partial \xi_k} = 0 \quad , \qquad (k=o,1,2, \dots, n)$$

Concerning the parameters ξ_k.

This principle can be extended to the multi-variate case, then, in the expression of s, x should be regarded as a vector variable.

Now, if the points lie approximately on a straight line, the most natural choice of the type of the curve is that of a straight line, but on the basis of our data, the fitting of a straight line sometimes gives clear unsatisfactory results. By using a polynomial of a higher degree, a close fit can always be attained. Thus if we know the real form of the curve to be fitted, then fitting of polynomials can be regarded in many cases as an approximating method.

Fitting A Polynomial

When fitting a polynomial of degree n we choose the numbers $\xi_0, \xi_1, \ldots, \xi_n$ in such a way that

$$S = \sum_{i=1}^{n} \left[y_i - (\xi_0 + \xi_1 x_i + \xi_2 x_i^2 + \ldots + \xi_n x_i^n) \right]^2$$

takes its minimal value. This method leads to the equations

$$\sum_{i=1}^{n} y_i = \xi_0 N + \xi_1 \sum_i x_i + \xi_2 \sum_i x_i^2 + \ldots + \xi_n \sum_i x_i^n$$

$$\sum_i x_i y_i = \xi_0 \sum_i x_i + \xi_1 \sum_i x_i^2 + \xi_2 \sum_i x_i^3 + \ldots + \xi_n \sum_i x_i^{n+1}$$

$$\vdots$$

$$\sum_i x_i^n y_i = \xi_0 \sum_i x_i^n + \xi_1 \sum_i x_i^{n+1} + \xi_2 \sum_i x_i^{n+2} + \ldots + \xi_n \sum_i x_i^{2n}$$

which can easily be solved for the parameters $\xi_o, \xi_1, \ldots, \xi_n$.
The sums $\sum_i x_i^k$, $\sum_i x_i^L y_i$, $(k=1,2,\ldots,2n;\ L=o,1,\ldots,n)$ are to be computed from the given data.

An Extension to Fitting A Polynomial

The method of fitting polynomials can be extended to a more general case; if a curve of the type

$$y = \xi_o\, g_o(x) + \xi_1 g_1(x) + \ \ldots\ + \xi_n\, g_n(x)$$

is fitted, where $g_j(x)$ $(j=o,1,2,\ldots,n)$ are arbitrarly chosen functions of x. Later on we shall deal with the different forms of $g_j(x)$. The method of least squares leads to the equations.

$$\sum_i y_i g_o(x_i) = \xi_o \sum_i (g_o(x_i))^2 + \xi_1 \sum_i g_1(x_i) g_o(x_i) + \xi_n \sum_i g_n(x_i) g_o(x_i)$$

$$\sum_i y_i g_1(x_i) = \xi_o \sum_i g_o(x_i) g_1(x_i) + \xi_1 \sum_i (g_1(x_i))^2 + \ldots + \xi_n \sum_i g_n(x_1) g_1(x_i)$$

$$\vdots$$

$$\sum_i y_i g_n(x_i) = \xi_o \sum_i g_o(x_i) g_n(x_i) + \xi_1 \sum_i g_1(x_i) g_n(x_i) + \ldots + \xi_n \sum_i (g_n(x_i))^2$$

In this case the numerical work has slightly increased as compared to the case of polynomials by the need of substituting in the functions $g_o(x)$, $g_1(x)$, ..., $g_n(x)$.

Least squares and least q^{th} Approximation

As is well known, Guass uses the principle $\sum_i h_i^2 = \text{Min.}$ in order to find the required function with fairly small error h_i. Sometimes instead of Gauss's principle, it is used to minimize $\sum_i h_i^q$ where q is a large positve integer, that is " the principle of least q^{th} power". The Tchebycheff's approximation is justified by the following fact "T-Approximation is a limiting case of the least q^{th} approximation if q tends to infinity".

Proof:

Let $\underline{h} = \max_{1 \leq i \leq N} |h_i|$

To minimize $\sum_i h_i^q$ is the same as

" " $\underline{h}^q \sum_i \left(\frac{h_i}{\underline{h}} \right)^q$

" " $\left[\underline{h}^q \sum_i \left(\frac{h_i}{\underline{h}} \right)^q \right]^{1/q}$

" " $\underline{h} \left[\left(\frac{h_1}{\underline{h}} \right)^q + \left(\frac{h_2}{\underline{h}} \right)^q + \dots + \left(\frac{h_i}{\underline{h}} \right)^q + \dots + \left(\frac{h_N}{\underline{h}} \right)^q \right]^{\frac{1}{q}}$

which tends to $\underline{h}$ as q tends to infinity. This proves the fact.

1.2. Approximation by Linear Programming

Let the values of a function be $f(x_i)$ at $x = x_i$ $(i=1,2, \ldots, N)$ and let n functions $g_j(x)$ be given $(j=1,2,\ldots,n)$

We should like to approximate the N values $f(x_i)$ by a linear combination of $g_j(x)$, i.e. by an expression

$$P_n(x) = \xi_o \; g_o(x) + \xi_1 \, g_1(x) + \ldots + \xi_n \, g_n(x)$$

choosing $\xi_o, \xi_1, \ldots, \xi_j, \ldots, \xi_n$ in such a way that the largest deviation between $P_n(x_i)$ and $f(x_i)$ is as small as possible. Formally, we wish to find ξ_j $(j=o,1,\ldots,n)$ such that the max. of $|P_n(x_i) - f(x_i)|$ is minimized.

Methods of representation:

For transforming this problem to a linear programming one, the trial was as follows, let

$$\underset{1 \leqslant i \leqslant N}{\text{Max.}} \left| P_n(x_i) - f(x_i) \right| = H$$

Then, for all i, we have

$$\left| P_n(x_i) - f(x_i) \right| \leqslant H \qquad (i=1,2, \ldots, N)$$

i.e. $-H \leqslant P_n(x_i) - f(x_i) \leqslant H$

and it is required to minimize H.

Because $P_n(x)$ is a linear expression in ξ_j, this is a linear programming problem. However, we must note that, it is not required that the values of ξ_j & H be non-negative.

Thus the linear programming problem can be written in the form :-

Find H, ξ_j subject to the following constraints

$$H + P_n(x_i) - f(x_i) \geqslant 0 \quad (i=1,2,\ldots,N)$$

& $$H - P_n(x_i) + f(x_i) \geqslant 0 \quad (i=1,2,\ldots,N)$$

and minimize

$$Z = H$$

Or more simply, find the (n+2) parameters $\xi_0, \xi_1, \ldots, \xi_j, \ldots, \xi_n$ & H subjected to the 2N inequalities

$$y_i = -f(x_i)+H+\xi_0 g_0(x_i)+\xi_1 g_1(x_i)+\ldots+\xi_n g_n(x_i) \quad (i=1,2,\ldots,N)$$

$$y_i' = f(x_i)+ H -\xi_0 g_0(x_i)-\xi_1 g_1(x_i)-\ldots-\xi_n g_n(x_i) \quad (i=1,2\ldots,N)$$

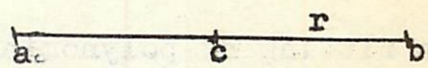
Thus our problem have been transformed as we have seen above to a linear programming problem with 2N inequalities and 2N+(n+2) variables out of them (n+2) are non-restricted ones and the others i.e. 2N are non-negative.

The Different Forms of $g_j(x)$

For the function $g_j(x)$, it is always assumed to have one of the following forms in the given interval on which f(x) is defined

1. Tchebycheff Polynomial of Degree j:

This polynomial is defined in the following form:- let(a,b) be the interval on the real axis on which $f(x_i)$ is defined. Let C be the mid-point of the interval (a,b) i.e. $\frac{a+b}{2}$ and let r be the length of ac or cb i.e. $\frac{b-a}{2}$,



Then the Tchebycheff's polynomial of degree J is defined by

$$g_j(x) = \text{Cos} \left(j \cos^{-1}\left(\frac{x-c}{r}\right)\right) = \cos j\phi$$

where $\cos\phi = \frac{x-c}{r} = X$

Some properties of Tchebycheff's polynomial

It is easy to see that Tchebycheff polynomial has the following properties

$$g_0(x) = 1$$

$$g_1(x) = X = \frac{x-c}{r} \quad \text{e.t.c.}$$

Also the following recurrence formula holds true

$$g_{j+1}(x) = 2\,X\,g_j(x) - g_{j-1}(x)$$

2. Any given polynomial of degree j

A very important special case of this form is the following simple form.

$$g_j(x) = x^j$$

In this case we have

$$P_n(x) = \xi_0 + \xi_1 x + \dots + \xi_n x^n \quad \text{i.e.}$$

the fitting of polynomials.

Our questionnow is, which form of $g_j(x)$ is to be used in fitting a curve or approximating a function? The answer of this question is, of course, depending on the type of curve to be fitted.

Construction of the equations of the linear programming problem.

After choosing the functions $g_j(x)$, the equations of the linear programming problem can be written in the form of the following table.

Table (1)

	1	H	ξ_o	ξ_1	ξ_j	ξ_n
$y_1=$	$-f(x_1)$	1	$g_o(x_1)$	$g_1(x_1)$	$g_j(x_1)$	$g_n(x_1)$
$y_2=$	$-f(x_2)$	1	$g_o(x_2)$	$g_1(x_2)$	$g_j(x_2)$	$g_n(x_2)$
$y_N=$	$-f(x_N)$	1	$g_1(x_N)$	$g_1(x_N)$	$g_j(x_N)$	$g_n(x_N)$
$y_1'=$	$f(x_1)$	1	$-g_o(x_1)$	$-g_1(x_1)$	$-g_j(x_1)$	$-g_n(x_1)$
$y_2'=$	$f(x_2)$	1	$-g_o(x_2)$	$-g_1(x_2)$	$-g_j(x_2)$	$-g_n(x_2)$
$y_N'=$	$f(x_N)$	1	$-g_o(x_N)$	$-g_1(x_N)$	$-g_j(x_N)$	$-g_n(x_N)$

For the sake of simplicity and in general words Table 1 can be written in the following tabular from.

Table (2)

	$1=x_1$	x_2	x_3	----	x_s	---	x_{n+3}
y_1 =	a_{11}	a_{12}	a_{13}	- - -	a_{1s}	---	$a_{1,n+3}$
y_2 =	a_{21}	a_{22}	a_{23}	- - -	a_{2s}	--	$a_{2,n+3}$
y_r =	a_{r1}	a_{r2}	a_{r3}	- - -	a_{rs}	--	$a_{r,n+3}$
y_N =	a_{N1}	a_{N2}	a_{N3}	- - -	a_{Ns}	--	$a_{N,n+3}$
y_{N+1}=	$a_{N+1,1}$	$a_{N+1,2}$	$a_{N+1,3}$	- - -	$a_{N+1,s}$	- -	$a_{N+1,n+3}$
y_{2N} =	$a_{2N,1}$	$a_{2N,2}$	$a_{2N,3}$	- - -	$a_{2N,s}$	- -	$a_{2N,n+3}$

Now, due to the fact that, the objective function z is expressen in terms of H and ξ_j (j=o,1,2,...,n) which are not necessarily required to be non-negative, then it is of high importance to say that this linear programming problem, after expressing it in a suitable form, can be easily solved by the well-known methods* of linear programming. This suitable form can be reached by expressing the objective function as well as the dependent variables in terms of other variables which are supposed to be non-negative. Thus for this purpose, it is required to change the (n+2) indpendent variables H and ξ_j's by some other (n+2) of the dependent one's y_i's. This operation can be done in a number of (n+2) steps called "pivoting steps" In each step one of the dependent variables is exchanged with one of the independent variables.

After each step a so called "current solution" has to be read from the new table. The current solution is obtained by putting a zero value for the independent variables and then the values of the objective function as well as those of the depedent variables have to be read from the column of the constant term in the new table.

---

* By these methods we mean the multiplex method and the simplex method in its different forms. These methods are supposed to be known to the mathematical programers.

<u>The pivoting step:</u>

The transformation of table (2.) in such a way that the dependent variable (assume this variable to be y_r) is exchanged with the independent variable (assume this variable to be x_s) is called a "pivoting operation". The row r is called the <u>pivot row</u> and the column S is called the <u>pivot column</u>. The element in the intersection of the pivot row with the pivot column is called the "<u>pivot element</u>" or simply the pivot. The choice of this pivot element is necessary and sufficient to know what is required to be changed.

Carring the pivoting operation on table (2.), it will have the form of table (3.).

Now, it is necessary to know some rules or principles for calculating the new elements of table (3.), that is the coefficients of the new independent variables. For this purpose, since

$$y_r \sum_{j=1}^{n+3} a_{rj}\, x_j \quad \ldots\ldots\ldots \quad (1)$$

and assumed* the element a_{rs} to be non-zero i.e. $a_{rs} \neq 0$, then

$$x_s = \sum_{j=1}^{s-1} - \frac{a_{rj}}{a_{rs}}\, x_j + \frac{1}{a_{rs}}\, y_r + \sum_{j=s+1}^{n+3} - \frac{a_{rj}}{a_{rs}}\, x_j \quad ..(2)$$

---

* This assumption is necessary, otherwise we can not devide by a_{rs}.

Table (3).

	x_1	x_2	x_3	y_r	x_{n+3}
$y_1 =$	$\underline{a}_{11}$	$\underline{a}_{12}$	$\underline{a}_{13}$	$\underline{a}_{1s}$	$\underline{a}_{1,n+3}$
$y_2 =$	$\underline{a}_{21}$	$\underline{a}_{22}$	$\underline{a}_{23}$	$\underline{a}_{2s}$	$\underline{a}_{2,n+3}$
$X_s =$	$\underline{a}_{r1}$	$\underline{a}_{r2}$	$\underline{a}_{r3}$	$\underline{a}_{rs}$	$\underline{a}_{r,n+3}$
$y_N =$	$\underline{a}_{N1}$	$\underline{a}_{N2}$	$\underline{a}_{N3}$	$\underline{a}_{Ns}$	$\underline{a}_{N,n+3}$
$y_{N+1} =$	$\underline{a}_{N+1,1}$	$\underline{a}_{N+1,2}$	$\underline{a}_{N+1,3}$	$\underline{a}_{N+1,s}$	$\underline{a}_{N+1,n+3}$
$y_{2N} =$	$\underline{a}_{2N,1}$	$\underline{a}_{2N,2}$	$\underline{a}_{2N,3}$	$\underline{a}_{2N,s}$	$\underline{a}_{2N,n+3}$

From equation (2) and table (3.), it follows that

1. The element $\underline{a}_{rs}$ in table (3.) which corresponds to the pivot element a_{rs} in table (2.) is given by

$$\underline{a}_{rs} = 1 / a_{rs}$$

2. The new elements $\underline{a}_{rj}$ $(j=1,2, \ldots,s-1,s+1,\ldots,n+3)$ in table (3.) which corresponds to the elements a_{rj} in the pivot row in table (2.) are given by

$$\underline{a}_{rj} = - a_{rj} / a_{rs}$$

Now, for the rest of the elements of table (3), we have

$$y_i = \sum_{j=1}^{n+3} a_{ij}\, x_j \quad , \qquad (i=1,2,\ldots,2N) \;\ldots.\; (3)$$

Then, substituting for x_s from equation (2) in equation (3), we get for $i \neq r$

$$y_i = \sum_{j=1}^{s-1} \left(a_{ij} - \frac{a_{is}a_{rj}}{a_{rs}}\right)x_j + \frac{a_{is}}{a_{rs}}\, y_r + \sum_{j=s+1}^{n+3} \left(a_{ij} - \frac{a_{is}a_{rj}}{a_{rs}}\right)x_j$$

which means that

3. The elements $\underline{a}_{is}$ $(i=1,2,\ldots,r-1,r+1,\ldots,2N)$ in table (3.) which corresponds to the old elements a_{is} of the pivot column in table (2.) are given by

$$\underline{a}_{is} = a_{is} / a_{rs} \qquad \text{(for all i, } i \neq r)$$

4. The elements $\underline{a}_{ij}$ $(i \neq r,\ j \neq s)$ which corresponds to all the elements a_{ij} in table (2.) exept those either in the pivot

column or in the pivot row are given by

$$\underline{a}_{ij} = a_{ij} - \frac{a_{is}\,a_{rj}}{a_{rs}}$$

Thus all new elements in table (3) can be constructed according to the following 4 rules

Rule 1:

New pivot element - 1/old pivot element

i.e. $\underline{a}_{rs} = 1/a_{rs}$

4	3	4
2	1	2
4	3	4

Rule 2.

The new elements in the pivot row=

=- old corresponding element/old pivot element

i.e. $\underline{a}_{rj} = - a_{rj}/a_{rs} \quad (j \neq s)$

Rule 3.

The new element in the pivot column =

= old corresponding element/old pivot element

i.e. $\underline{a}_{is} = a_{is} / a_{rs} \quad (i \neq r)$

Rule 4.

The new element (neither in the pivot row not in the pivot column) = old corresponding element +

(old element of the pivot column in the same row)

x (new element of the pivot row " " " column)

i.e. $\underline{a}_{ij} = a_{ij} - a_{is}\, a_{rj}/a_{rs} = a_{ij} + a_{is}\, \underline{a}_{rj}$

This means that, assuming the element a_{rs} to be a pivot element (of course $a_{rs} \neq o$), we can exchange the dependent variable y_r and the independent variable x_s, and by appling the previous 4 rules, table (3.) can be constructed.

Since our aim is to exchange the (n+2) variables x_j ($j=2,3, \ldots, n+3$) with some (n+2) variables from y_i ($i=1,2,\ldots,2N$) then the question now is how can we choose the (n+2) variables from the y_i's? In other words, for a fixed j, that is a fixed pivot column, which row i has to be chosen as a pivot row? Actually, we can choose any row i such that $a_{ij} \neq o$. A suitable choice is to choose the row for which

$$a_{rs} = \max_{1 \leqslant i \leqslant 2N} |a_{is}|$$

If there is more than one element in the column j having this property, we can select any one of them.

Assuming that, we had chosen the pivot element a_{rs} and consequently, we have constructed table (3.), then it is of high importance to put in consideration that it is not allowed in the next pivoting operations - to exchange any x_j ($j=2,3, \ldots, n+3$ & $j \neq s$) with the dependent variable in the $r^{\underline{th}}$ row, since it is again x_s which is one of the unbounded variables.

Table (4)

	$1=x_1$	y_{r1}	y_{rs}	$y_{r,n+2}$	
x_2	A_{11}	A_{12}	A_{13}	$A_{1,n+3}$	
y_1	A_{21}	A_{22}	A_{23}	$A_{2,n+3}$	$\geqslant 0$
y_2	A_{31}	A_{32}	A_{33}	$A_{3,n+3}$	$\geqslant 0$
$\vdots$					
y_{r1}					
y_{r2}					
$\vdots$					
y_{rn+2}					
$\vdots$					
y_{2N}	$A_{2N-n-2,1}$	$A_{2N-n-2,2}$	$A_{2N-n-2,3}$	$A_{2N-n-2,n+3}$	$\geqslant 0$
x_3	$A_{2N-n-1,1}$	$A_{2N-n-1,2}$	$A_{2N-n-1,3}$	$A_{2N-n-1,n+3}$	
x_4	$A_{2N-n,1}$	$A_{2N-n,2}$	$A_{2N-n,3}$	$A_{2N-n,n+3}$	
x_{n+3}	$A_{2N,1}$	$A_{2N,2}$	$A_{2N,3}$	$A_{2N,n+3}$	

In this table it is assumed that $n+2 < N$

Now, repeating this pivoting operation and exchanging other variables and so on till we reach that position where all x_j ($j=2,3,\ldots,n+3$) are exchanged with some $(n+2)$ variables from y_i ($i=1,2, \ldots , 2N$) (say $y_{r1}, y_{r2} , \ldots , y_{rn+2}$) and remembering that $n+2 \leqslant N$, we should get a new table generally has the form of table (4.) but after some arrangements. Table (4.) is a table of the linear programming problem. The following is the summary of what we have done to obtain table (4.)

The procedure to obtain table (4) is as follows

1- In the pivot operations 1,2,..., n+2 repectively, suppose that we have to exchange $x_2, x_3, \ldots, x_{n+3}$ respectively with the n+2 variables y_i $(i=r_1, r_2, \ldots, r_{n+2})$ respectively

2- For the first pivot operation, we have to exchange x_2 with the variable y_i such that

$$a_{i,2} = \underset{1 \leqslant i \leqslant 2N}{\text{Max.}} |a_{i2}|$$

Let this fixed i be r_1

3- For the next pivot operation, we have to exchange x_3 with the variable y_i such that

$$a_{i,3} = \underset{1 \leqslant i \leqslant 2N}{\text{Max.}} |a_{i,3}| \qquad (i \neq r_1)$$

let this fixed i be r_2

4- Assuming now that, we have exchanged x_j $(j=2,3,\ldots,t)$ with y_i $(i=r_1,r_2,\ldots,r_{t-1})$ respectively, then we can exchange x_{t+1} with y_i such that

$$a_{i,t+1} = \underset{1 \leqslant i \leqslant 2N}{\text{Max.}} |a_{i,t+1}| \qquad (i \neq r_1,r_2,\ldots,r_{t-1})$$

let this fixed i be r_t.

5- repeat step 4 till we reach table (4).

Table (4) can be written expressed in our original symbols, H, ξ_j $(j=o,1,\ldots,n)$ as follows.

Table (5)

	$1=x_1$	y_{r1}	y_{r2}	y_{1-n+2}	
x_2	A_{11}	A_{12}	A_{13}	$A_{1,n+3}$	
y_1	A_{21}	A_{22}	A_{23}	$A_{2,n+3}$	$\geqq 0$
y_2	A_{31}	A_{32}	A_{33}	$A_{3,n+3}$	$\geqq 0$
$\vdots$					
y_{r1}					
y_{r2}					
$\vdots$					
y_{rn+2}					
$\vdots$					
y_{2N}	$A_{2N-n-2,1}$	$A_{2N-n-2,2}$	$A_{2N-n-2,3}$	$A_{2N-n-2,n+3}$	$\geqq 0$
ξ_o	$A_{2N-n-1,1}$	$A_{2N-n-1,2}$	$A_{2N-n-1,3}$	$A_{2N-n-1,n+3}$	
ξ_1	$A_{2N-n,1}$	$A_{2N-n,2}$	$A_{2N-n,3}$	$A_{2N-n,n+3}$	
$\vdots$					
ξ_n	$A_{2N,1}$	$A_{2N,2}$	$A_{2N,3}$	$A_{2N,n+3}$	

Table (5) is a table prepared to linear programming.

A flow chart and its FORTRAN program have been done (see chapter 3.) for transforming the curve fitting problem to the linear programming problem indicated by table (5.). This linear programming problem can be solved by any of the well-known methods f solving such problems. A flow chart and the corresponding FORTRAN program for the simplex algorithm compact for inequalities had been done in Memo. 485 by Dr. Roshdi Amer "faculty of engineering-Ein Shams university". We have carried out a slight change in that program to neutrialize the effect of adding the last (n+1) equations in table (5.) and to obtain the required results. The modified FORTRAN program can be seen in shapter (3.). By the help of this last program, we shall be able to obtain the coefficients of the required polynomial of degree n and the value of the maximum absolute error. We have done also a third FORTRAN program to calculate the errors from the approximating function at all the given points.

The best approximating polynomial of degree n in the given N points can be acheived through the optimal solution obtained from these mentioned programs. The method of organizing the work with these programs depends on the type of the problem. In the following, the two possible types, that can be encountered by any application, are to be treated.

Type 1.

If the number of the given points (N) is not too large compared to the degree of the required polynomial (n), we have to take all the N points (x_i, y_i) in consideration and to apply consequently the three programs. The optimal approximating polynomial will be achieved through the solution obtained by using these programs according to the following steps:-

1) For the first program, the required data are only the given N points and the degree of the required polynomial. Its result will be the previously mentioned table (5.) i.e. a table for a linear programming programming problem . This linear programming problem will be taken as an input data for the second FORTRAN program.

2) The results of program II are the maximum absolute error as well as the coefficients of the required approximating polynomial.

3) Taking these results of program II as an input data for the last program, the purpose of this last program is to provide us with the errors at different points from the derived polynomial.

4) If it is found that the solution obtained untill now is satisfactory(*), then the process is terminated, otherwise step 1) is repeated. Thus by the help of these 3 programs,

---

*See the iterative routine to obtain the optimal solution in type 2.

we have already obtained the best approximating polynomial and the errors at the different points from this polynomial.

Type 2

If the number of the given points (N) is too large compared to the degree of the best approximating polynomial (n), we can not take all the given N points in consideration, in applying the three programs. To obtain the optimal solution for the given curve fitting problem, let us introduce the following

Definition

A reference is a set $\{x_i\}$ of (n+2) distinct abscissas among the given N abscissas.

The iterative routine to obtain the optimal solution

We choose any reference $\{x_i\}$. Appling the three programs to the points of reference as input data, the best approximating function for these points of reference and the errors at all the given N points (i.e. h_i for $i=1,2,\ldots,N$) from the approximating function can be obtained. The fact that the errors at the points of reference are all equal in magnitude and alternating in signs must be observed when arranging the points of reference according to the order of their abscissas. Let the value of these equal errors be H. Now, it turns out that either

$$h_i \leqslant H \qquad \text{(for all } i=1,2,\ldots,N\text{)}$$

or

$$h_i > H \qquad \text{(for some } i=r,s,\ldots,t\text{)}$$

In the first case, it can be proved that the approximating function at hand is a function of the best fit i.e. we have reached the optimal solution.

In the second case, we choose a new reference $\{\bar{x}_i\}$ among all given points. Let the abscissas of this new reference be denoted by $x_{u1} < x_{u2} < \dots < x_{u(n+2)}$. The choice of the new reference must sotisfy the following

a) The new reference points errors must again be alternating in signs if their abscissas are orderd i.e.

$$\text{sign } h_{ui} = \text{sign } h_{u(i+2n)} = - \text{sign } h_{u(i+2n+1)}$$

where n is any integer (+ve or - ve)

b) We prefere the point whose abscissa x_u than that whose abscissa x_v if the absolute value of the error at the first is greater than that at the second point i.e. if

$$|h_u| > |h_v|$$

It is lear that in this case, where $|h_i| > H$ for $(i=r,s,\dots,t)$, at least one point (x_i, y_i) $(i=r,s,\dots,t)$ must be exchanged with acorresponding point from the points of the first reference.

The new reference chosed by the previous 2 conditions is called <u>the optimal reference</u>.

To go from any reference to the optimal reference through computations on the electronic computer, this needs a somewhat complicated program which can be done if it is required to automize the whole process.

Now, due to the fact that, the objective function z is expressen in terms of H and ξ_j (j=o,1,2,...,n) which are not necessarily required to be non-negative, then it is of high importance to say that this linear programming problem, after expressing it in a suitable form, can be easily solved by the well-known methods* of linear programming. This suitable form can be reached by expressing the objective function as well as the dependent variables in terms of other variables which are supposed to be non-negative. Thus for this purpose, it is required to change the (n+2) indpendent variables H and ξ_j's by some other (n+2) of the dependent one's y_i's. This operation can be done in a number of (n+2) steps called "pivoting steps" In each step one of the dependent variables is exchanged with one of the independent variables.

After each step a so called "current solution" has to be read from the new table. The current solution is obtained by putting a zero value for the independent variables and then the values of the objective function as well as those of the depedent variables have to be read from the column of the constant term in the new table.

---

* By these methods we mean the multiplex method and the simplex method in its different forms. These methods are supposed to be known to the mathematical programmers.

The pivoting step:

The transformation of table (2.) in such a way that the dependent variable (assume this variable to be y_r) is exchanged with the independent variable (assume this variable to be x_s) is called a "pivoting operation". The row r is called the pivot row and the column S is called the pivot column. The element in the intersection of the pivot row with the pivot column is called the "pivot element" or simply the pivot. The choice of this pivot element is necessary and sufficient to know what is required to be changed.

Carring the pivoting operation on table (2.), it will have the form of table (3.).

Now, it is necessary to know some rules or principles for calculating the new elements of table (3.), that is the coefficients of the new independent variables. For this purpose, since

$$y_r \sum_{j=1}^{n+3} a_{rj} \; x_j \qquad \ldots\ldots\ldots \quad (1)$$

and assumed* the element a_{rs} to be non-zero i.e. $a_{rs} \neq 0$, then

$$x_s = \sum_{j=1}^{s-1} - \frac{a_{rj}}{a_{rs}} x_j + \frac{1}{a_{rs}} y_r + \sum_{j=s+1}^{n+3} - \frac{a_{rj}}{a_{rs}} x_j \ldots (2)$$

---

* This assumption is necessary, otherwise we can not devide by a_{rs}.

To clarify the meaning of the choice of the new reference, let us discuss the following problem.

Suppose we have N = 15 points and it is required to find the best approximating polynomial of degree n= 2 i.e. ($n \ll N$).

To solve this problem; we choose first any reference i.e. any set of 4 points and then we apply the three programs to determine the error curve. Let the error curve be as shown in fig.(1.) below.

Now, if the point of the first reference is the point indicated by the arrows i.e. O_1 then, chosen the new reference by the points indicated by the dotted arrows i.e. O_2, it follows that, the chosen points satisfy the previous two conditions. Thus it is the optimal reference .

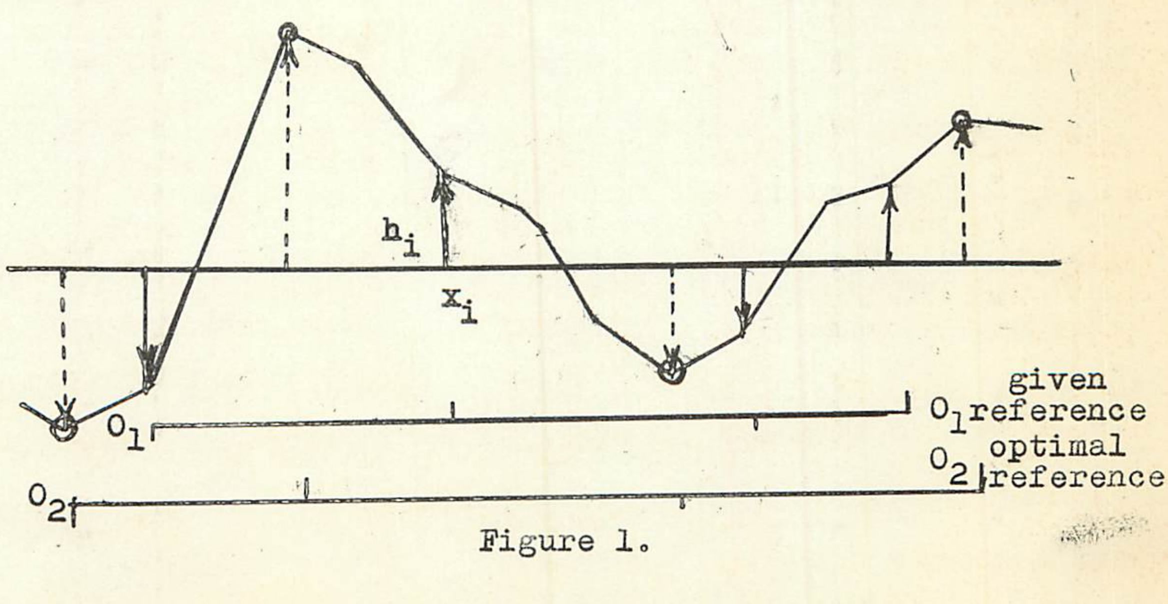


Figure 1.

Appling now, the three programs to this new reference to determine

1- The approximating polynomial through this new reference and the maximum absolute error.

2- The errors at all the given points,

We can test whether we have reached the optimal solution or not. If we reached such optimal position we stop the routine, and if not we continue this iterative method.

The following fig (2.) is a diagram for the iterative method used to obtain the optimal solution.

The convergence of this method to the required result is ensured by the exchange theorem which is mentioned in the appendix.

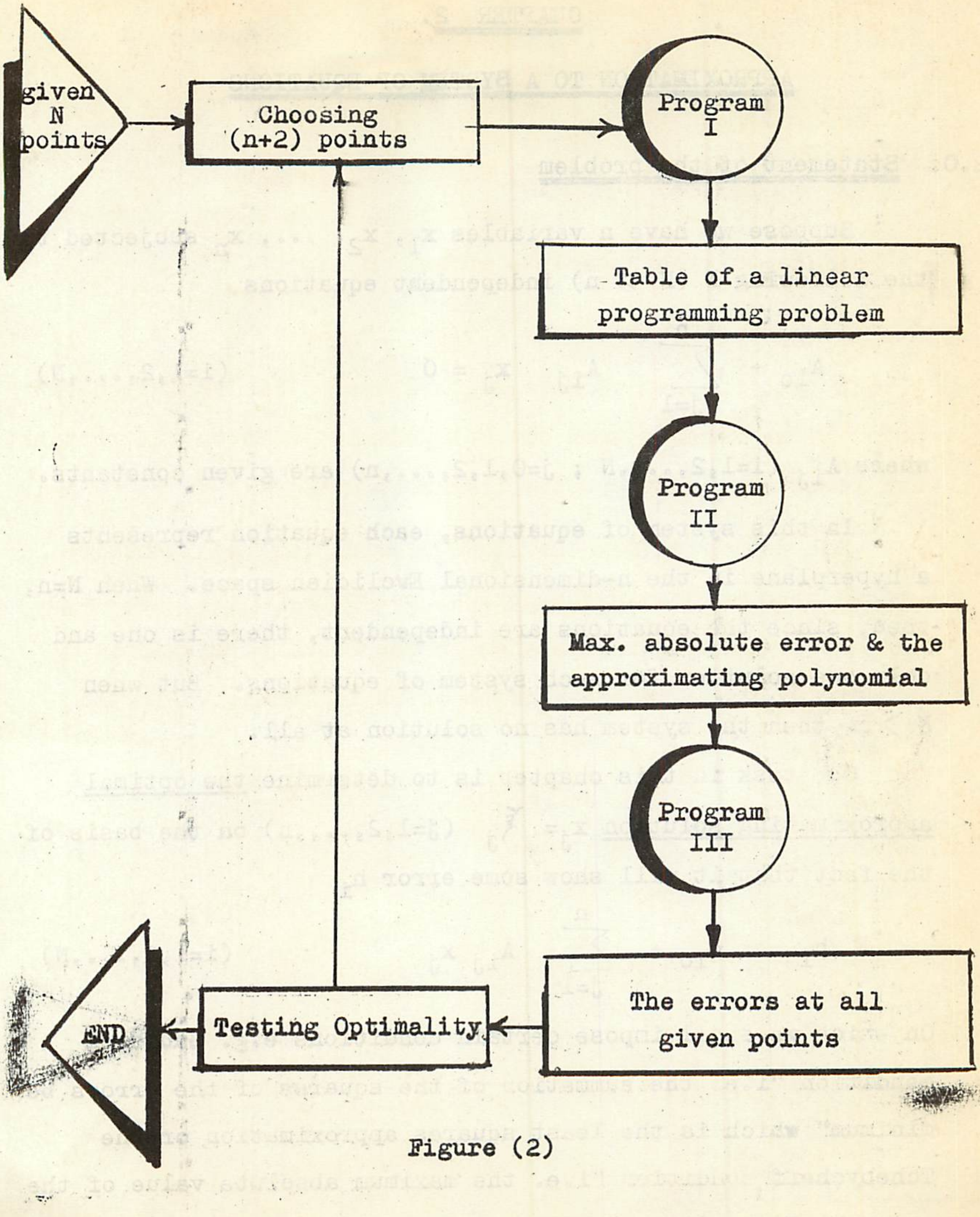

given
N
points
Choosing
(n+2) points
Program
I
Table of a linear
programming problem
Program
II
Max. absolute error & the
approximating polynomial
Program
III
The errors at all
given points
Testing Optimality
END

Figure (2)

CHAPTER 2.

APPROXIMATION TO A SYSTEM OF EQUATIONS

2.0. Statement of the problem

Suppose we have n variables $x_1, x_2, \ldots, x_n$ subjected to the following N ($N \geqslant n$) independent equations

$$A_{io} + \sum_{j=1}^{n} A_{ij} \quad x_j = 0 \qquad (i=1,2,\ldots,N)$$

where A_{ij} ($i=1,2,\ldots,N$; $j=0,1,2,\ldots,n$) are given constants.

In this system of equations, each equation represents a hyperplane in the n-dimensional Evclidian space. When N=n, then, since the equations are independent, there is one and only one solution for such system of equations. But when $N > n$, then the system has no solution at all.

Our task in this chapter is to determine the optimal approximating solution $x_j = \xi_j$ ($j=1,2,\ldots,n$) on the basis of the fact that it will show some error h_i

$$h_i = A_{io} + \sum_{j=1}^{n} A_{ij} x_j \qquad (i=1,2,\ldots,N)$$

On which we shall impose certain conditions e.g. Gaussion condition "i.e. the summation of the squares of the errors be minimum" which is the least squares approximation or the Tchebycheff condition "i.e. the maximum absolute value of the error h_i be a minimum" which is used either in T-approximation or the linear programming approach.

2.1 Least squares approximation

The approximation problem in this case can be summarized as follows:-

Given the following system of independent equations

$$A_{io} + \sum_{j=1}^{n} A_{ij} \; x_j = 0 \qquad (i=1,2,\ldots,N)$$

where $N > n$; A_{ij} $(i=1,2,\ldots,N; j=0,1,2,\ldots,n)$ are given constants, it is required to determine the optimal approximating solution ξ_j $(j=1,2,\ldots,n)$ such that the errors h_i

$$h_i = A_{io} + \sum_{j=1}^{n} A_{ij} \; x_j \qquad (i=1,2,\ldots,N)$$

satisfy the following Gaussian condition

$$S = \sum_{i=1}^{N} h_i^2 = \text{Min.}$$

For the solution of such a problem, we have to solve the following first order partial defferential conditions.

$$\frac{\partial S}{\partial \xi_j} = 0 \qquad (j=1,2,\ldots,n)$$

which leads to the system

$$\sum_{i=1}^{N} \left[A_{io} + \sum_{j=1}^{n} A_{ij} \; \xi_j \right] A_{ij} = 0 \qquad (j=1,2,\ldots,n)$$

i.e.
$$\sum_{i=1}^{N} A_{io} \; A_{ij} + \sum_{i=1}^{N} A_{ij} \sum_{j=1}^{n} A_{ij} \; \xi_j = 0 \quad (j=1,\ldots,n)$$

$$\sum_{i=1}^{N} A_{io} \; A_{ij} + \sum_{k=1}^{n} \left(\sum_{i=1}^{N} A_{ik} \; A_{ij} \right) \xi_k = 0$$

$$\therefore \; a_{oj} + \sum_{k=1}^{n} a_{kj} \; \xi_k = 0 \qquad (j=1,2,\ldots,n)$$

where

$$a_{oj} = \sum_{i=1}^{N} A_{io} A_{ij} \quad , \quad a_{kj} = \sum_{i=1}^{N} A_{ik} A_{ij}$$

From the last system of equations, it follows that we have n variables in n-independent equations.

Therefore, it can be solved for the approximating values

$\xi_1, \; \xi_2, \; \ldots, \; \xi_n.$

In case of hand calculations, the complications arising are those of calculating the values of a_{kj} for

$$(k=o,1,2,\ldots,n; \qquad j=1,2,\ldots,n)$$

2.2. T-approximation by Linear Programming

It is wanted to caculate the optimal approximating solution i.e. the values of $\xi_1, \xi_2, \ldots, \xi_n$ of the variables $x_1, x_2, \ldots, x_n$ which are subjected to the following N ($N > n$) constraints

$$a_{io} + \sum_{j=1}^{n} a_{ij} x_j = o \qquad (i=1,2,\ldots,N)$$

such that the maximum absolute value of the error h_i

$$h_i = a_{io} + \sum_{j=1}^{n} a_{ij} \xi_j \qquad (i=1,2,\ldots,N)$$

is as small as possible, i.e.

$$\underset{1 \leqslant i \leqslant N}{\text{Max.}} \left| h_i \right| = \text{Min.}$$

The trial to transform this problem to a linear programming one is similar to the previous attempt done in the method of representation (see I.2.) i.e.

Let

$$\underset{1 \leqslant i \leqslant N}{\text{Max.}} \left| h_i \right| = H$$

$$\therefore \left| a_{io} + \sum_{j=1}^{n} a_{ij} \xi_j \right| \leqslant H \qquad (i=1,2,\ldots,N)$$

$$-H \leqslant a_{io} + \sum_{j=1}^{n} a_{ij} \xi_j \leqslant H \qquad (i=1,2,\ldots,N)$$

$$\therefore \; y_i = a_{io} + H + \sum_{j=1}^{n} a_{ij} \xi_j \geqslant o \qquad (i=1,2,\ldots,N)$$

$$\& \quad y_i = -a_{io} + H - \sum_{j=1}^{n} a_{ij} \xi_j \geqslant o \qquad (i=1,2,\ldots,N)$$

Because the inequalities are linear expressions in H and ξ_j (j=1,2,...,n), this is a linear programming problem as previously mentioned, However we must remember again that the values of H and ξ_j are not required to be non-negative. This linear programming problem can be written in the form.

Find the (n+1) variables H and ξ_j (j=1,2,...,n) subjected to the 2N inequalities

$$y_i = a_{io} + H + a_{i1}\xi_1 + a_{i2}\xi_2 + \dots + a_{in}\xi_n \quad (i=1,2,..,N)$$

$$\& \quad y_{i+N} = -a_{io} + H - a_{i1}\xi_1 - a_{i2}\xi_2 - \dots - a_{in}\xi_n \quad (i=1,2,..,N)$$

and minimize the objective function

$$z = H$$

Aga n, we should like to emphasize that the approximation problem has be transformed to a linear programming problem with 2N const s in 2N+(n+1) variables, Out of this number of variables (n+1) are on-restricted and the rest (i.e. 2N) are non-negative.

This las form of the problem can be tabulated in a tabular form simil to that of table (2.) and the algorithm can be continued as previously mentioned in case of curve fitting.

About the programs

We have generalized the FORTRAN program of the curve fitting problem to be suitable for solving the approximation problem. On solving the problem (either curve fitting or approximation) on the electronic computer, it is of high importance to adjust the switchs according to the associated problem, since the difference is contained in the positions of the switchs.

Ageometrical interpretation of Tchebycheff's approximation:

To pick out the idea of Tchebycheff's approximation let us interprete geometrically the following example.

Determine the nearest point to the following 3 straight lines

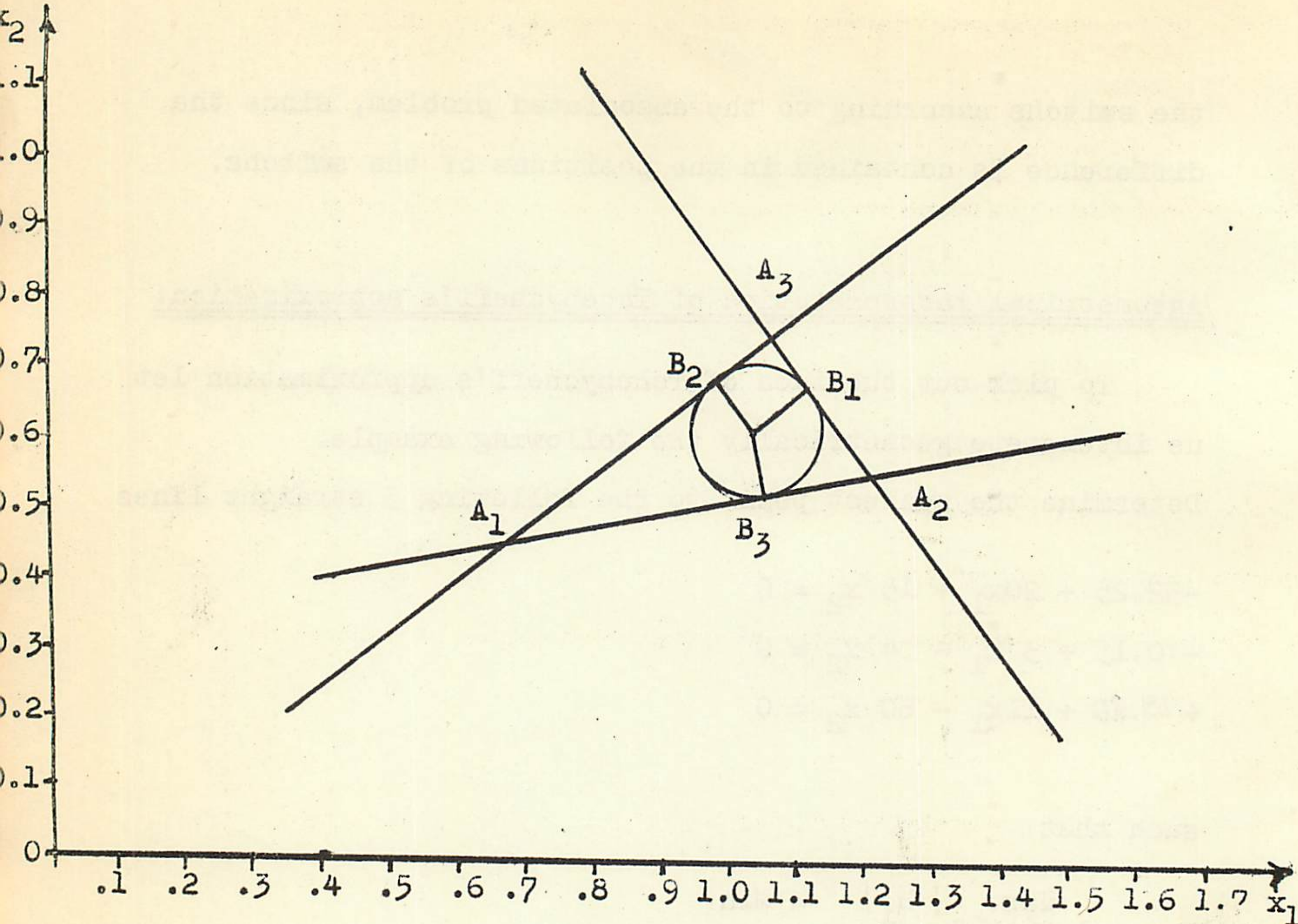
$$-32.25 + 20x_1 + 15\,x_2 = 0$$
$$-\,0.15 + 3\,x_1 - .4\,x_2 = 0$$
$$+19.80 + 11x_1 - 60\,x_2 = 0$$

such that

$$\underset{1 \leqslant i \leqslant 3}{\text{Max.}} \, |h_i| = \text{Min.}$$

where h_i's are the respective distances from the given st. lines.

Here we have 3 equations in 2 variables and each equation represents in the 2-dimensional Euclidian space a straight line. Plotting these 3 straight lines we get figure (1).



Let the points of intersection of these 3 straight lines be A_1, A_2, A_3.

Solving this problem by the programs done for that purpose, the results shows that

maximum $|h_i|$ = 0.0950

& ξ_1 = 1.0308

& ξ_2 = 0.6162

i.e. the best approximating point to the given problem is the point (1.0308, 0.6162)

Now let us calculate the length of the perpendiculars from the point $O \equiv (1.0308, 0.6162)$ to the 3 sides A_2A_3, A_3A_1, A_1A_2 of the triangle (i.e. OB_1, OB_2, OB_3)

$$OB_1 = \frac{20(1.0308)+15(0.6162)-32.25}{25} = 0.096$$

$$OB_2 = \frac{3(1.0308)-4(0.6162)-0.15}{5} = 0.096$$

$$OB_3 = \frac{11(1.0308)-60(0.6162)+1980}{61} = 0.096$$

Again, this means that for 3 equations in 2 variables, the optimum approximating solution shows that it is the point from which the perpendiculars to the 3 given equations are all equal to the maximum approximating error. More precisely it is the centre of the inscribed circle. Thus, we can reach the following conclusion:-

In a 2- dimensional Euclidian space, if we have more than 3 equations in 2 variables, the optimal point must be the centre of the inscribed circle (i.e. the circle which touches 3 of the straight lines) such that each one of the rest of the straight lines must intersect the circle in 2 real points, otherwise the errors for those lines will be more than the maximum absolute error (i.e. the radias of the circle).

CHAPTER 3.

3.0 FIELDS OF APPLICATION

This research can be applied in a broad variaty to problems from great many fields such as engineering, physical, statistical and economical fields.

We shall not mention here many problems from these fields of application, however I have the great pleasure to mention that, I have already solved during the preparation of this thesis some problems which were been given to the Operations Research Center from the Faculty of Engineering and the Atomic Energy Establishment. The problems of the fuclty ofengineering are curve fitting problems and that of the atomic energy establishment are approximating one's. The two types of problems can be expressed as follows :-

In the first type of problems, in a research for the analysis of the non-linear distortion in a semi-conductor diode detectors, it is wanted to fit a relation between the direct current I_{dc} and the applied voltage under certain biasing and series resistance conditions.

The solution of this problem is given later on & indicated by problem number (1).

In the second type of problems, in a research in nuclear physics, it is wanted to find the constants $|A|^2$ and $|B|^2$ which satisfy the following system of equations

$$\left| A\,F(I) + B\,G(I) \right|^2 = EX\,(I) \qquad (I=1,2,\ldots,N)$$

where

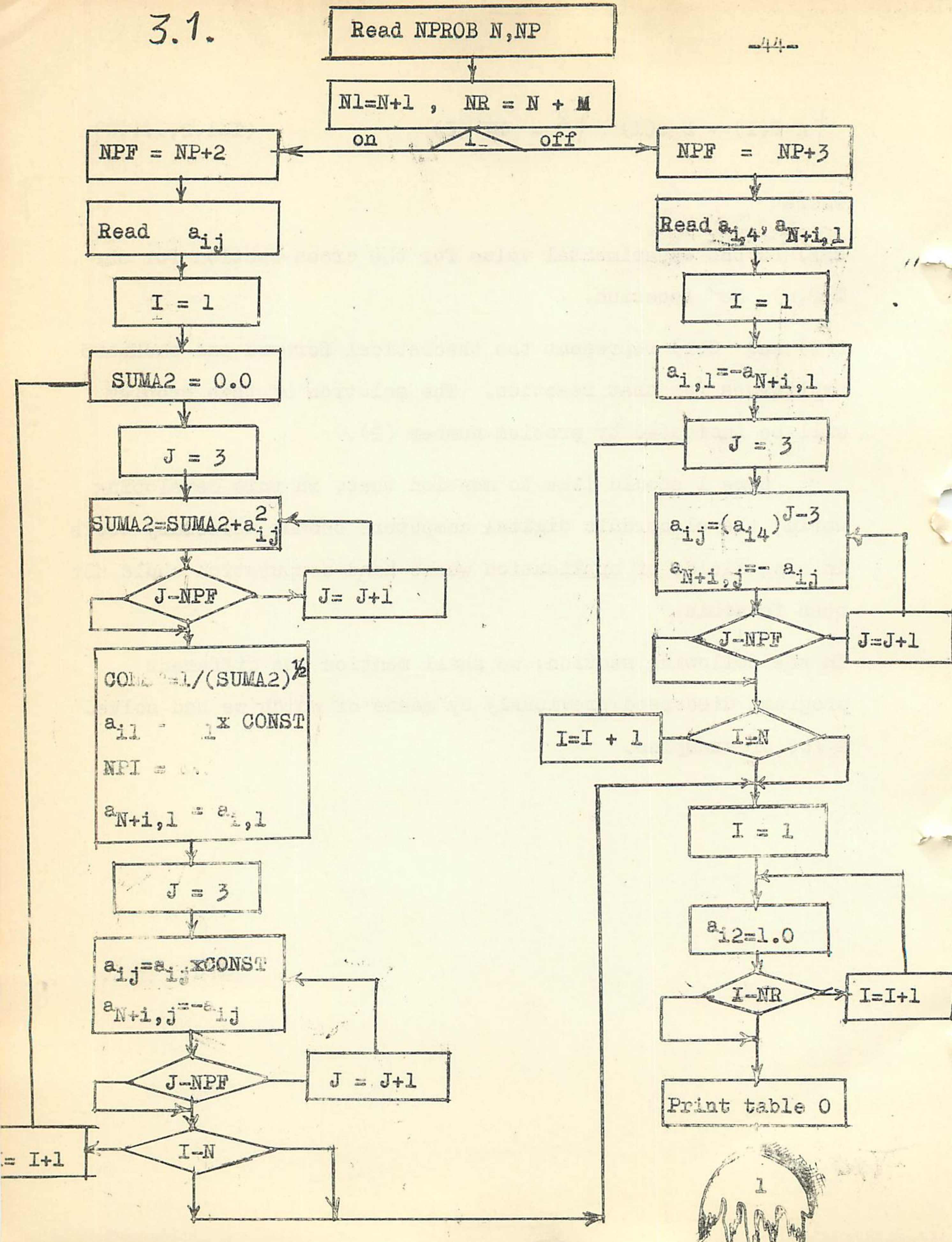
E(I) is the experimental value for the cross-section for the $T(P,n)\ He^3$ reaction.

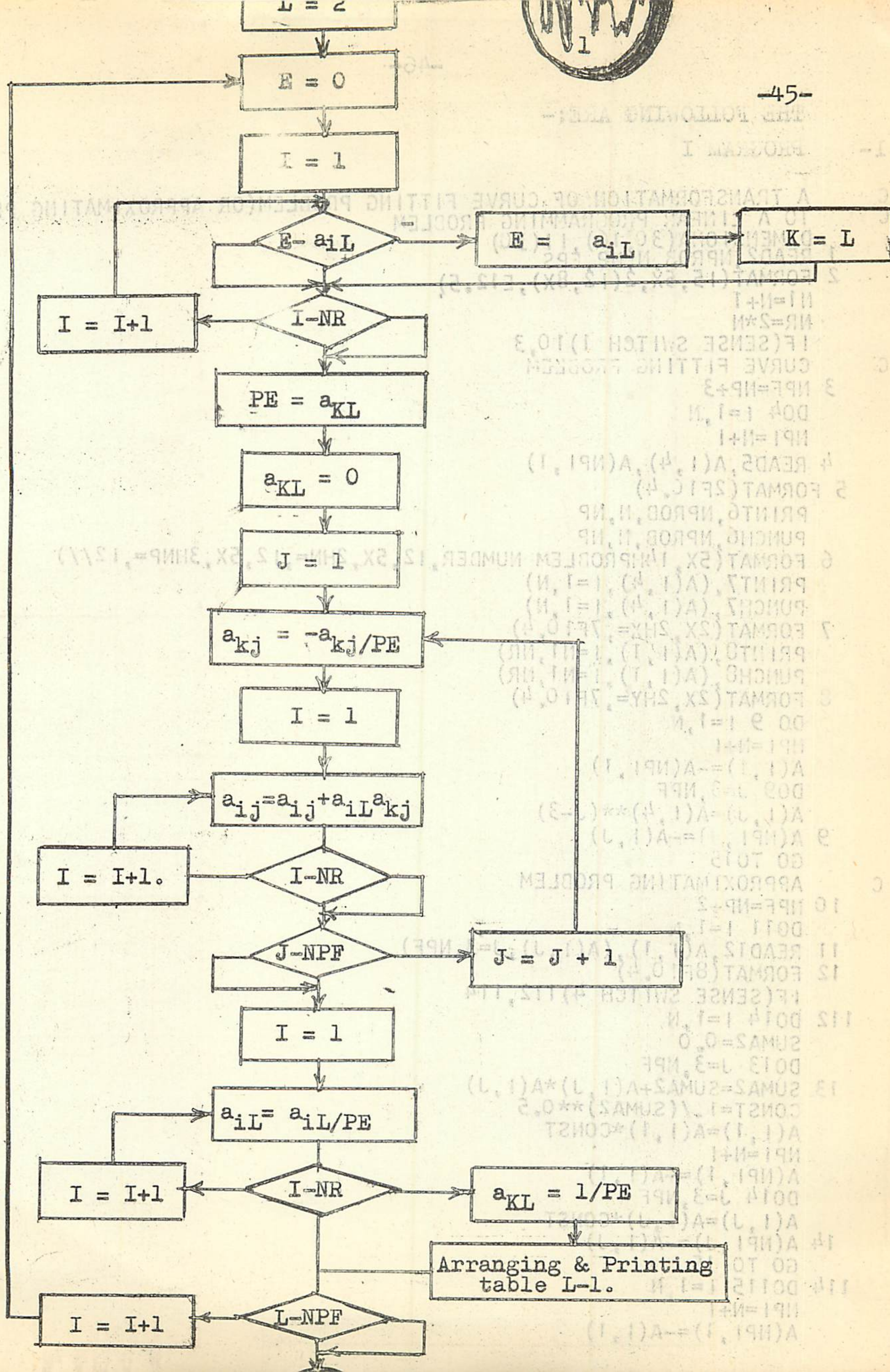
F(I) and G(I) represent the theoretical forward and backward amplitudes for that reaction. The solution of this problem will be indicated by problem number (2).

Here I should like to mention that, in this developing world, the electronic digital computers become necessary tools in such fields of application where hand computation could not been feasible.

In the following section, we shall mention the different programs discussed previously by means of which we had solved a lot of examples.

3.1.


Read NPROB N,NP
N1=N+1 , NR = N + M
on
I
off
NPF = NP+2
Read a_{ij}
I = 1
SUMA2 = 0.0
J = 3
SUMA2=SUMA2+a_{ij}^2
J-NPF
J= J+1
CONST=1/(SUMA2)½
a_{i1} = a_{i1} x CONST
$a_{N+i,1}$ = $a_{i,1}$
J = 3
a_{ij}=a_{ij}xCONST
$a_{N+i,j}$=-a_{ij}
J-NPF
J = J+1
I-N
I= I+1
NPF = NP+3
Read $a_{i,4}$, $a_{N+i,1}$
I = 1
$a_{i,1}$=-$a_{N+i,1}$
J = 3
a_{ij}=$(a_{i4})^{J-3}$
$a_{N+i,j}$=- a_{ij}
J-NPF
J=J+1
I=I + 1
I-N
I = 1
a_{i2}=1.0
I-NR
I=I+1
Print table 0
1


L = 2
E = 0
I = 1
E - a_{iL}
E = | a_{iL} |
K = L
I = I+1
I-NR
PE = a_{KL}
a_{KL} = 0
J = 1
a_{kj} = $-a_{kj}$/PE
I = 1
a_{ij}=a_{ij}+$a_{iL}a_{kj}$
I = I+1.
I-NR
J-NPF
J = J + 1
I = 1
a_{iL}= a_{iL}/PE
I = I+1
I-NR
a_{KL} = 1/PE
Arranging & Printing table L-1.
I = I+1
L-NPF

THE FOLLOWING ARE:-

1- PROGRAM I

```
C     A TRANSFORMATION OF CURVE FITTING PROBLEM(OR APPROXIMATING PROB.)
C     TO A LINEAR PROGRAMMING PROBLEM
      DIMENSIONA(30,16),II(30)
    1 READ2,NPROB,N,NP,EPS
    2 FORMAT(I5,5X,2(I2,8X),E12.5)
      N1=N+1
      NR=2*N
      IF(SENSE SWITCH 1)10,3
C     CURVE FITTING PROBLEM
    3 NPF=NP+3
      DO4 I=1,N
      NPI=N+I
    4 READ5,A(I,4),A(NPI,1)
    5 FORMAT(2F10.4)
      PRINT6,NPROB,N,NP
      PUNCH6,NPROB,N,NP
    6 FORMAT(5X,14HPROBLEM NUMBER,I2,5X,2HN=,I2,5X,3HNP=,I2//)
      PRINT7,(A(I,4),I=1,N)
      PUNCH7,(A(I,4),I=1,N)
    7 FORMAT(2X,2HX=,7F10.4)
      PRINT8,(A(I,1),I=N1,NR)
      PUNCH8,(A(I,1),I=N1,NR)
    8 FORMAT(2X,2HY=,7F10.4)
      DO 9 I=1,N
      NPI=N+I
      A(I,1)=-A(NPI,1)
      DO9 J=3,NPF
      A(I,J)=A(I,4)**(J-3)
    9 A(NPI,J)=-A(I,J)
      GO TO15
C     APPROXIMATING PROBLEM
   10 NPF=NP+2
      DO11 I=1,N
   11 READ12,A(I,1),(A(I,J),J=3,NPF)
   12 FORMAT(8F10.4)
      IF(SENSE SWITCH 4)112,114
  112 DO14 I=1,N
      SUMA2=0.0
      DO13 J=3,NPF
   13 SUMA2=SUMA2+A(I,J)*A(I,J)
      CONST=1./(SUMA2)**0.5
      A(I,1)=A(I,1)*CONST
      NPI=N+I
      A(NPI,1)=-A(I,1)
      DO14 J=3,NPF
      A(I,J)=A(I,J)*CONST
   14 A(NPI,J)=-A(I,J)
      GO TO 15
  114 DO115 I=1,N
      NPI=N+I
      A(NPI,1)=-A(I,1)
```

```
      DO115 J=3,NPF
  115 A(NPI,J)=-A(I,J)
   15 DO16 I=1,NR
   16 A(I,2)=1.0
      PRINT17,(L,L=1,NPF)
   17 FORMAT(36X,7HTABLE 0//2X,1HX,4X,I2,6(8X,I2)//)
      DO18 I=1,NR
   18 PRINT19,I,(A(I,J),J=1,NPF)
   19 FORMAT(1HY,I2,1H=,7F9.3)
C     PIVOT ALGORITHM
      DO20 K=1,NR
   20 II(K)=0
      DO47 L=2,NPF
      E=0.0
      DO23 K=1,NR
      IF(II(K))21,21,23
   21 IF(E-ABSF(A(K,L)))22,23,23
   22 E=ABSF(A(K,L))
      IP=K
      JP=L
   23 CONTINUE
      PE=A(IP,JP)
      A(IP,JP)=0.0
      DO24 J=1,NPF
      A(IP,J)=-A(IP,J)/PE
      DO24 I=1,NR
   24 A(I,J)=A(I,J)+A(I,JP)*A(IP,J)
      DO25 I=1,NR
   25 A(I,JP)=A(I,JP)/PE
      A(IP,JP)=1.0/PE
      IF(SENSE SWITCH 2)26,27
   26 PRINT29,IP,JP
   27 IF(SENSE SWITCH 3)28,30
   28 PUNCH29,IP,JP
   29 FORMAT(//5X,3HIP=,I2,5X,3HJP=,I2//)
C     ARRANGMENT OF THE TABLES
   30 IF(L-2)31,31,33
   31 J1=L-1
      II(IP)=1
      DO32 J=1,NPF
      A1=A(J1,J)
      A(J1,J)=A(IP,J)
   32 A(IP,J)=A1
      GO TO 35
   33 K1=NR-(NPF-L)
      II(K1)=1
      DO34 J=1,NPF
      A1=A(K1,J)
      A(K1,J)=A(IP,J)
   34 A(IP,J)=A1
```

```
C     PRINTING OF THE RESULTS
   35 J1=L-1
      IF(L-NPF)36,48,48
   36 IF(SENSE SWITCH 2)37,38
   37 PRINT40,J1,(J,J=1,NPF)
   38 IF(SENSE SWITCH 3)39,41
   39 PUNCH40,J1,(J,J=1,NPF)
   40 FORMAT(/36X,5HTABLE,I2//3X,I2,7(8X,I2)//)
   41 IF(SENSE SWITCH 4)42,47
   42 DO47 I=1,NR
      IF(SENSE SWITCH 2)43,44
   43 PRINT46,(A(I,J),J=1,NPF)
   44 IF(SENSE SWITCH 3)45,47
   45 PUNCH46,(A(I,J),J=1,NPF)
   46 FORMAT(8F10.4)
   47 CONTINUE
   48 M=NR-(NPF-1)
      N=NPF-1
      PRINT49,NPROB,M,N,EPS
      PUNCH49,NPROB,M,N,EPS
   49 FORMAT(//19X,41HA TABLE OF THE LINEAR PROGRAMMING PROBLEM///14HPRO
     XBLEM NUMBER,I2,5X,2HM=,I2,5X,2HN=,I2,5X,E12.5)
      DO50 I=1,NR
      PUNCH51,(A(I,J),J=1,NPF)
   50 PRINT51,(A(I,J),J=1,NPF)
   51 FORMAT(8F10.4)
      GO TO 1
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START
END OF PASS 1
```

2. PROGRAM II.

```
C     SIMPLE ALGORITHM
C     COMPACT FOR INEQUALITIES
      DIMENSIONA(30,16),IXROW(30),IXCOL(16),COEF(16)
  120 READ1,NPROB,M,N,EPS
    1 FORMAT(14HPROBLEM NUMBER,I2,5X,2HM=,I2,5X,2HN=,I2,5X,E12.5)
      PRINT3,NPROB,M,N
      IF(SENSE SWITCH 1) 2,4
    2 PUNCH3,NPROB,M,N
    3 FORMAT(//14HPROBLEM NUMBER,I2,5X,2HM=,I2,5X,2HN=,I2)
    4 M1=M+1
      M2=M+2
      N1=N+1
      NR=M+N
      MN1=M1+N
      DO5 I=1,NR
    5 READ 6,(A(I,J),J=1,N1)
    6 FORMAT(8F10.4)
      DO 7 I=1,M1
    7 IXROW(I)=I
      DO201 I=M2,NR
  201 IXROW(I)=M2-I
      DO 8 J=2,N1
    8 IXCOL(J)=M+J
      NCOL=N1
      LOT=-1
      GO TO 70
    9 IZ =1
      NSTEP=0
C     INTRODUCE ARTIFICIAL VARIBLE
      C =0.0
      DO 11 I=2,M1
      IF(A(I,1)-C)10,11,11
   10 IZ=I
      C =A(I,1)
   11 CONTINUE
      IF(C)12,34,34
   12 NCOL=N1+1
      IXCOL(NCOL)=IXROW(IZ)
      IXROW(IZ)=MN1+1
      DO13 I=1,NR
   13 A(I,NCOL)=0.0
      A(IZ,NCOL)=-1.0
      DO 14 J=1,NCOL
      C=-A(IZ,J)
      A(IZ,J)=0.0
      DO 14 I=2,NR
      A(I,J)=A(I,J)+C
   14 CONTINUE
C     PHASE 1
      IPHASE=1
```

```
   20 LJP=-1
      GO TO 40
   21 IF(C)22,125,125
   22 LIP=-1
      GO TO 50
   23 LPS=-1
      GO TO 60
   24 IF(IP-IZ)20,30,20
  125 PRINT 225,A(IZ,1)
  225 FORMAT(///21HARTIFICIAL VARIABLE =,E12.5///)
   25 IF(A(IZ,1)-EPS)325,325,90
  325 A(IZ,1)=0.0
   27 C=0.0
      DO 29 J=2,NCOL
      IF(A(IZ,J)-C)29,29,28
   28 JP=J
      C=A(IZ,J)
   29 CONTINUE
      IP=IZ
      LPS=0
      GO TO 60
   30 IF(JP-N1)31,31,33
   31 IXCOL(JP)=IXCOL(NCOL)
      DO32 I=1,NR
   32 A(I,JP)=A(I,NCOL)
   33 NCOL=N1
      IZ=1
      NSTEP=0
C     PHASE 2
   34 IPHASE=2
   35 LJP=[illegible]
      GO TO 40
   36 IF(C)37,91,91
   37 LIP=1
      GO TO 50
   38 IF(IP)92,92,39
   39 LPS=1
      GO TO 60
C     PIVOT COLUMN
   40 JP =-1
      C =1.0
      DO 42 J=2,NCOL
      IF(A(IZ,J)-C)41,42,42
   41 JP=J
      C = A(IZ,J)
   42 CONTINUE
      IF(LJP)21,43,36
   43 IFT=43
      GO TO 115
C     PIVOT ROW
   50 Q = 1.0E+90
      IP=-1
```

```
      DO 53 I=2,M1
      IF(A(I,JP))51,53,53
   51 IF(Q+A(I,1)/A(I,JP))53,53,52
   52 Q =-A(I,1)/A(I,JP)
      IP=I
   53 CONTINUE
      IF(LIP)23,54,38
   54 IFT=54
      GO TO 115
C     PIVOT STEP
   60 LOT=0
      GO TO 70
   61 PIVINV=1.0/A(IP,JP)
      A(IP,JP)=0.0
      DO 62 J=1,NCOL
      C=-A(IP,J)*PIVINV
      A(IP,J)=C
      DO62 I=1,NR
      A(I,J)=A(I,J)+C*A(I,JP)
   62 CONTINUE
      DO63 I=1,NR
   63 A(I,JP)=A(I,JP)*PIVINV
      A(IP,JP)=PIVINV
      IXR=IXROW(IP)
      IXROW(IP)=IXCOL(JP)
      IXCOL(JP)=IXR
      NSTEP=NSTEP+1
      IF(LPS)24,30,35
C     OUTPUT
   70 IF(LOT)74,71,71
   71 PRINT 72,IPHASE,NSTEP
   72 FORMAT(/////5HPHASE,I2,50X,4HSTEP,I3/)
      IF(SENSE SWITCH 1)73,74
   73 PUNCH 72,IPHASE,NSTEP
      PUNCH 76,(IXCOL(J),J=2,NCOL)
      GO TO 77
   74 IF(SENSE SWITCH 2)75,77
   75 PRINT 76,(IXCOL(J),J=2,NCOL)
   76 FORMAT(10X,1H1,11X,1HX,I2,11X,1HX,I2,11X,1HX,I2,11X,1HX,I2)
   77 DO81 I=1,NR
      IF(SENSE SWITCH 1)78,79
   78 PUNCH 82,IXROW(I),(A(I,J),J=1,NCOL)
      GO TO 81
   79 IF(SENSE SWITCH 2)80,81
   80 IF(SENSE SWITCH 3)180,181
  180 PRINT182,IXROW(I),A(I,1)
      GO TO 81
  181 PRINT 82,IXROW(I),(A(I,J),J=1,NCOL)
   81 CONTINUE
   82 FORMAT(1HX,I2,2H =,(E12.5,2X,E12.5,2X,E12.5,2X,E12.5,2X,E12.5))
```

```
  182 FORMAT(1HX,I2,2H =,E12.5)
      IF(LOT) 9,83,93
   83 PRINT 84,IP,JP
   84 FORMAT(/4HIP =,I2,6X,4HJP =,I2)
      IF(SENSE SWITCH 1)85,86
   85 PUNCH 84,IP,JP
   86 GO TO 61
C     PROGRAM TERMINATION
   90 ISOL=-1
      LOT=1
      GO TO 70
   91 ISOL=0
      LOT=1
      GO TO 70
   92 ISOL=1
      LOT=1
      GO TO 70
   93 IF(ISOL) 96,94,100
   94 IF(C)95,104,107
   95 IFT=95
      GO TO 115
C     SYSTEM INCONSISTENT
   96 PRINT 97,A(IZ,1)
   97 FORMAT(//42HSYSTEM INCONSISTENT, ARTIFICIAL VARIABLE = ,E12.5)
      IF(SENSE SWITCH 1)98,99
   98 PUNCH 97,A(IZ,1)
   99 GO TO 110
C     UNBOUND SOLOTION
  100 IX=IXCOL(JP)
      PRINT 101,IX
  101 FORMAT(//19HUNBOUND SOLUTION, X,I2,10H =INFINITY)
      IF(SENSE SWITCH 1)102,103
  102 PUNCH 101,IX
  103 GO TO 110
C     OPTIMAL SOLUTION NON-UNIQUE
  104 PRINT 105
  105 FORMAT(//27HOPTIMAL SOLUTION NON-UNIQUE/)
      GO TO 110
C     OPTIMAL SOLUTION UNIQUE
  107 PRINT 108
  108 FORMAT(//23HUNIQUE OPTIMAL SOLUTION/)
C     SOLUTION
  110 DO200 I=M2,NR
      J=I-M2
  200 COEF(J)=A(I,1)
      PRINT301,A(1,1)
      PUNCH301,A(1,1)
  301 FORMAT(/20X,39HRESULTS OF THE APPROXIMATING POLYNOMIAL//20HMAX. AB
     XSOLUTE ERROR=,E12.5//25X30HCOEFFICIENTS OF THE POLYNOMIAL)
      NP1=N-1
```

```
          DO202 M=1,NP1
          J=M-1
          PRINT203,J,COEF(J)
    202   PUNCH203,J,COEF(J)
    203   FORMAT(1HA,I2,3H = ,F10.4)
          GO TO 120
C         FAULTS
    115   PRINT 116,IFT
    116   FORMAT(///5HFAULT,I2)
          END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START
```

3- PROGRAM III.

```
C     ERROR FUNCTION FOR CURVE FITTING
      DIMENSIONX(25),F(25),PN(25),ERROR(25),COEF(24)
    1 READ2,NPROB,N,NP
    2 FORMAT(I5,5X,I2,8X,I2)
      DO3 I=1,N
    3 READ4,X(I),F(I)
    4 FORMAT(2F10.4)
      NP1=NP+1
      DO5 M=1,NP1
      J=M-1
    5 READ6,COEF(J)
    6 FORMAT(6X,F10.4)
      DO8 I=1,N
      Y=COEF(NP)
      DO7 J=1,NP
      K=NP-J
    7 Y=Y*X(I)+COEF(K)
      PN(I)=Y
    8 ERROR(I)=PN(I)-F(I)
       IF(SENSE SWITCH 1)9,11
   9 PRINT16,NPROB
      DO10 I=1,N
   10 PRINT14,I,X(I),ERROR(I)
   11 IF(SENSE SWITCH 2)12,15
   12 PUNCH16,NPROB
      DO13 I=1,N
   13 PUNCH14,I,X(I),ERROR(I)
   14 FORMAT(I5,5X,F10.4,13X,F10.4)
   16 FORMAT(5X,14HPROBLEM NUMBER,I2//23X,34HA TABLE FOR ERRORS AT GIVEN
     X POINTS//12X,5HPOINT,15X,5HERROR/)
   15 GO TO 1
      END
TURN SW 1 ON FOR SYMBOL TABLE, PRESS START
```

.2 SOME EXAMPLES FROM REAL FIELDS OF APPLICATIØN

PROBLEM NUMBER 1 N= 6 NP= 3

X=	.8245	1.1405	1.4184	1.8234	2.4417	2.9506
Y=	7.4908	12.0674	22.9974	39.2608	74.3811	113.6420

TABLE 0

X	1	2	3	4	5	6
Y 1=	-7.490	1.000	1.000	.824	.679	.560
Y 2=	-12.067	1.000	1.000	1.040	1.082	1.126
Y 3=	-22.997	1.000	1.000	1.418	2.011	2.853
Y 4=	-39.260	1.000	1.000	1.823	3.324	6.062
Y 5=	-74.381	1.000	1.000	2.441	5.961	14.557
Y 6=	-113.642	1.000	1.000	2.950	8.706	25.688
Y 7=	7.490	1.000	-1.000	-.824	-.679	-.560
Y 8=	12.067	1.000	-1.000	-1.040	-1.082	-1.126
Y 9=	22.997	1.000	-1.000	-1.418	-2.011	-2.853
Y10=	39.260	1.000	-1.000	-1.823	-3.324	-6.062
Y11=	74.381	1.000	-1.000	-2.441	-5.961	-14.557
Y12=	113.642	1.000	-1.000	-2.950	-8.706	-25.688

A TABLE OF THE LINEAR PROGRAMMING PROBLEM

PROBLEM NUMBER 1 M= 7 N= 5 1.00000E-05

0.0000	.5000	.5000	0.0000	0.0000	0.0000
-.0313	1.4404	.8303	.1942	-.8303	-.6347
-.0350	1.5224	1.3375	.2018	-1.3375	-.7242
0.0000	.9999	.9999	0.0000	-.9999	0.0000
0.0000	1.0000	1.0000	-1.0000	0.0000	0.0000
0.0000	.9999	.9999	0.0000	0.0000	-.9999
.0350	-.5224	-.3375	-.2018	1.3375	.7242
.0313	-.4404	.1695	-.1942	.8304	.6347
.0254	-5.2075	-9.0323	-3.0098	8.5323	8.7173
.6017	11.7743	16.7347	6.5339	-16.7347	-18.3082
9.2457	-6.8289	-8.9298	-4.1731	8.9298	11.0020
1.2202	1.1452	1.4364	.8199	-1.4364	-1.9651

Taking the previous results of the first program as input data for the second program, the results will be as follows.

PROBLEM NUMBER 1 M= 7 N= 5

PHASE 1 STEP 0

IP = 3 JP = 2

PHASE 2 STEP 0

IP = 5 JP = 4

PHASE 2 STEP 1

UNIQUE OPTIMAL SOLUTION

RESULTS OF THE APPROXIMATING POLYNOMIAL

MAX. ABSOLUTE ERROR= 1.01496E-02

COEFFICIENTS OF THE POLYNOMIAL

A 0 = -.1414
A 1 = .9733
A 2 = 9.0223
A 3 = 1.2600

PROBLEM NUMBER 2 N=7 NP=2

TABLE 0

X	1	2	3	4
Y 1=	2.854	1.000	.366	.930
Y 2=	2.790	1.000	.361	.932
Y 3=	2.608	1.000	.347	.937
Y 4=	2.368	1.000	.328	.944
Y 5=	2.264	1.000	.307	.951
Y 6=	2.300	1.000	.289	.957
Y 7=	2.582	1.000	.272	.962
Y 8=	-2.854	1.000	-.366	-.930
Y 9=	-2.790	1.000	-.361	-.932
Y10=	-2.608	1.000	-.347	-.937
Y11=	-2.368	1.000	-.328	-.944
Y12=	-2.264	1.000	-.307	-.951
Y13=	-2.300	1.000	-.289	-.957
Y14=	-2.582	1.000	-.272	-.962

A TABLE OF THE LINEAR PROGRAMMING PROBLEM

PROBLEM NUMBER 2 M=11 N= 3 1.00000E-05

0.0000	.5000	.4999	0.0000
-.0489	1.0001	.0588	-.0589
-.1930	1.0004	.2037	-.2041
-.3781	1.0006	.4103	-.4109
-.4225	1.0005	.6264	-.6270
-.3320	1.0003	.8236	-.8240
0.0000	1.0000	.9999	-.9999
.3320	-.0003	.1763	.8240
.0489	-.0001	.9411	.0589
.1930	-.0004	.7962	.2041
.3781	-.0006	.5896	.4109
.4225	-.0005	.3735	.6270
-3.4521	.1601	-9.4810	9.3209
-1.7078	.4743	3.2003	-3.6746

Taking the linear programming table as input data for the second program, the results will be as follows:-

PROBLEM NUMBER 2 M=11 N= 3

PHASE 1 STEP 0

IP = 5 JP = 2

PHASE 2 STEP 0

UNIQUE OPTIMAL SOLUTION

RESULTS OF THE APPROXIMATING POLYNOMIAL

MAX. ABSOLUTE ERROR= 2.11144E-01

COEFFICIENTS OF THE POLYNOMIAL

A 0 = -3.3844
A 1 = -1.5075

PROBLEM NUMBER 3 N = 3 NP = 2

TABLE 0

X	1	2	3	4
Y 1=	-1.290	1.000	.800	.600
Y 2=	-.030	1.000	.600	-.800
Y 3=	.324	1.000	.180	-.983
Y 4=	1.290	1.000	-.800	-.600
Y 5=	.030	1.000	-.600	.800
Y 6=	-.324	1.000	-.180	.983

A TABLE OF THE LINEAR PROGRAMMING PROBLEM

PROBLEM NUMBER 3 M= 3 N= 3 1.00000E-05

0.0000	.5000	.5000	0.0000
.2237	1.1704	.7245	-.8950
-.2237	-.1704	.2754	.8950
0.0000	.9999	.9999	-.9999
1.0500	.1000	-.7000	.6000
.7500	.7000	.1000	-.8000

PHASE 1 STEP 0

IP = 4 JP = 4

PHASE 1 STEP 1

IP = 3 JP = 3

PHASE 2 STEP 0

The last table is the result of the first program which is the data for the following second program.

UNIQUE OPTIMAL SOLUTION

RESULTS OF THE APPROXIMATING POLYNOMIAL

MAX. ABSOLUTE ERROR= 9.55656E-02

COEFFICIENTS OF THE POLYNOMIAL

A 0 = 1.0308
A 1 = .6162

Appendix
Tcheby cheff Approximation*

We are going to state some basic definitions by means of which we shall deduce some basic theorems for the method of Tchebycheff approximation. First of all I should like to mention the following property.

If $\{x_i\}$ is a set of (n+1) distinct abscissas, and $f(x_i)$ is the corresponding functional values, then, there exists an umbiguously determinate function $P_n(x)$ such that

$P_n(x_i) = f(x_i)$, $(i = 1,2,\ldots,n+1)$

This property is equivalent to the statement that the determinant of the linear system

$$\xi_0 \; g_0(x_i) + \xi_1 \, g_1(x_i) + \ldots + \xi_n \, g_n(x_i) = f(x_i)$$

does not vanish.

Definition 1.

A reference is a set $\{x_i\}$ of (n+2) distinct abscissas x_i whose corresponding values $P_n(x_i)$ of any function $P_n(x)$ are related by the following linear relation.

$$(2) \qquad \sum_{i=1}^{n+2} \lambda_i \, P_n(x_i) = 0$$

---

* This part is an extract of a paper by E. Stiefel. see (1)

where λ_i may be obtained as co-factors of the last column in the determinant

$$\left| g_o(x_i) \quad , \quad g_1(x_i) \, , \, \ldots , \, g_n(x_i), \, P_n(x_i) \right| \qquad (i=1,2,..,n+2)$$

Assuming now that interpolation by function $P_n(x)$ of type (1) is possible and unique, it follows that

(3) $\lambda_i \neq 0$ $\qquad$ $(i=1,2,\ldots, n+2)$

Relation (2) is of high importance for solving the T- problem. It is called "the characteristic relation"

Definition 2

Let $P_n(x)$ be any function of class (1) and let $h_i = P_n(x_i) - f(x_i)$ be its errors of approximation at the reference points x_i, then $P_n(x_i)$ is called a reference function" with respect to the reference $\{ x_i \}$ if

(4) Sign h_i = sign λ_i $\qquad$ or if sign h_i = - sign λ_i

Thus we have the following

Theorem 1.

The errors (taken at the reference points) of all the reference function corresponding to a fixed reference build a bound set of numbers.

Proof :

From the characteristic relation (2), it follows that $\sum_{i=1}^{n+2} \lambda_i (f (x_i) + h_i) = 0$ or

(5) $$\sum_{i=1}^{n+2} \lambda_i h_i = - \sum_{i=1}^{n+2} \lambda_i f(x_i)$$

taking (4) in consideration we have

$$\sum_{i=1}^{n+2} |\lambda_i| \; |h_i| = \pm \sum_{i=1}^{n+2} \lambda_i \; f(x_i)$$

Therefore, the errors h_i are bounded.

Definition 3.

A levelled reference function withe respect to a given reference $\{ x_i \}$ is a reference function characterized by the property that its error h_i has the same absolute value at each reference point. This common absolute value of the approximation error h_i is called the reference deviation" corresponding to the given reference x_i .

To compute this levelled reference function, since
$h_i = h \text{ sign } \lambda_i$
then it follows from (5) that

(6) $$h = - \frac{\sum_i \lambda_i \; f(x_i)}{\sum_i |\lambda_i|}$$

and then the required function can be determined by its value $f(x_i) + h \operatorname{sign} \lambda_i$ at the reference points and can be computed by interpolation.

Theorem 2.

The reference deviation is a weighted mean of the errors $|h_i|$ of any reference function (taken at the reference point). The weights are non-Vanishing positive numbers.

Proof :

Let $P_n(x)$ be any function of class (1) with approximation errors $h_i = P_n(x_i) - f(x_i)$.

Then it follows from (6) and (2) that

$$(7) \quad h = - \frac{\sum_i \lambda_i \left[P_n(x_i) - h_i \right]}{\sum_i |\lambda_i|} = \frac{\sum_i \lambda_i \, h_i}{\sum_i |\lambda_i|}$$

and therefore for a reference function, this is equivalent to

$$|h| = \frac{\sum_i |\lambda_i| \, |h_i|}{\sum_i |\lambda_i|}$$

For any approximating function $P_n(x)$ having not necessarily the property of a reference function it follows from (7)

$$(8) \quad |h| \leqslant \text{Max.} \, |h_i|$$

The last statement follows from (3).

Thus theorem 2, implies that the inequalities

(9) $\text{Min.} \, |h_i| \leq h \leq \text{Max} \, |h_i|$

holds true for any reference function $P_n(x)$ on the form (1).

The Exchange Method

Assume now that only tabulated values $f(x_i)$ of the function at the distinct abscissas $x_1 < x_2 < \ldots < x_i < \ldots < x_N$ are given, and it is required an approximation function $P_n(x)$ of type (1) with $n < N-1$, such that the max. absolute error.

$$|h_i| = |P_n(x_i) - f(x_i)|$$

is as small as possible . i.e.

$$\text{Max.} \, |h_i| = \text{Min.}$$

Definition

The smallest value of max $|h_i|$ of the approximation function is called T - deviation. By a reference is now understood a subset $\{x_i\}$ of $n+2$ abscissas among the given N abscissas.

The solution of this direct T-problem is based on the following fact.

<u>Exchange Theorem</u>+

Let a reference $\{x_i\}$ and a corresponding reference function $P_n(x)$ be given, Furthermore let $\bar{x}_i$ be any abscissa among the N abscissas not co-onsiding with a reference point. Then there is an abscissa x_ρ out of the reference $\{x_i\}$ such that $P_n(x)$ is again a reference function with respect to the reference built by the remaining points of reference (after excluding x_ρ) and the new point $\bar{x}_i$.

This means that we can exchange x_ρ and $\bar{x}_i$ without loosing the poperty of a function to be a reference function. <u>Thus to solve the T - problem, we use the following iterative routine</u>

A reference $\{x_i\}$ is choosen and the corresponding levelled reference function $P_n(x)$ is constructed. Its error h_i would have certainly the property

$$m = \text{Max} \; |h_i| \geqslant |h| \qquad (i=1,2, \ldots, N)$$

where $|h|$ is the reference deviation of $P_n(x)$. Hence either $m > |h|$ or $m = |h|$

In the second case, we stop the routine because it turns out that $P_n(x)$ is already a function of the best fit.

\+ The proof of this theorem can be seen in (2)

Now, in the first case where $m > |h|$, a point $\bar{x}_i$ is selected where the error assumes its maximal value m. Using the exchange theorem a new reference $\{\bar{x}_i\}$ can be found including the point $\bar{x}_i$ and having the property that $P_n(x)$ is again a reference function.

Among the errors $\bar{h}_i$ of $P_n(x)$ at the new reference points, (n+1) are equal to $|h|$ and one is equal to m. Constructing now the levelled reference function $\bar{P}_n(x)$ with respect to the new reference $\{\bar{x}_i\}$ and supposing that its reference deviation is $|\bar{h}|$, it follows (from theorem 2) that $|\bar{h}|$ is a weighted mean of $|h|$ and m with positive non-vanishing weights. Thus, since $m > |h|$, it follows that

10) $|\bar{h}| > |h|$

Now a new reference is constructed and soon. Alternatively, the approximation function and the reference are changed. Since the reference deriation is allways raised monatonically as indicated from (10), the same reference can never appear twice during the routine, and since there is a finite number of references in the whole set of abscissa, then after a finite number of steps, the procedure must come to an end. At the end, a new reference $\{x_i\}$ and a corresponding levelled reference function $P_n(x)$ with reference deviation H and error H_i is faced and certainly

(11) $\quad \text{Max.} \left| H_i \right| = H \qquad (i=1,2,\ldots,N)$

because otherwize the algorithm could be carried on.

It turns out now that

Theorem 3.

This last function $P_n(x)$ is a function of the best fit.

Proof

Let $Q_n(x)$ be any function of class (1) with errors h_i $(i=1,2,\ldots,N)$ and h_σ at the points of our last reference, then from (9) and (11) we get.

(12) $\quad \text{Max.} \left| h_i \right| \geqslant \text{Max.} \left| h_\sigma \right| \geqslant H = \text{Max.} \left| H_i \right|$

and this proves the above theorem.

H is nothing else than the T - deviation.

Theorem 4.

The function of the best fit is uniquely determined. It is a levelled reference function characterized by the property to have a maximal approxmation error equals to its reference deviation.

Proof

Let us suppose that we have any function $\bar{P}_n(x)$ having errors $\bar{H}_i$ satisfying

$\text{Max.} \left| \bar{H}_i \right| \leqslant H$

Then, in particular at the abscissas of our last reference

Max. $\quad |\bar{H}_\sigma| \leqslant H.$

Therefore it follows from (8) that

Max. $\quad |\bar{H}_\sigma| = H$

$\therefore \quad |\bar{H}_\sigma| \leqslant H$

Taking in consideration now the previous levelled reference function $P_n(x)$ with respect to the reference at hand, whose errors $|H_\sigma|$ are all equal to the reference deviation $|H|$ it follows that

$|\bar{H}_\sigma| \leqslant |H_\sigma|$

$$(H_\sigma - \bar{H}_\sigma) \begin{cases} \geqslant 0 & \text{if} \quad H_\sigma \geqslant 0 \\ \leqslant 0 & \text{if} \quad H_\sigma \leqslant 0 \end{cases}$$

One can realize, that the expression (sign λ_σ)($H_\sigma - \bar{H}_\sigma$) have all the same sign by appling (4) to the reference function $P_n(x)$, and consequently the same statement holds true for the expression $\lambda_\sigma(H_\sigma - \bar{H}_\sigma)$. But, since The characteristic relation implies

$$\sum_\sigma \lambda_\sigma (H_\sigma - \bar{H}_\sigma) = 0,$$

it follows from (3) that

$(H_\sigma - \bar{H}_\sigma) = 0 \qquad$ for any σ

The uniqueners of the inter polation finally shows that

$P_n(x) = P_n(x)$

i.e. the equality holds if $P_n(x)$ coinsides with the levelled reference function of the given reference.

This theorem is a special case of de la Vallée-Poussin theorem. Furthermore, the following basic inequality holds true for any reference function $P_n(x)$ with error h_i

(13) $\text{Min.} \; |h_\nu| \leqslant H \leqslant \text{Max.} \; |h_\nu|$

where h_ν are the errors at the reference points.
The left side of (13) follows from (9) and the right side follows from (12)

It is of high importance to say in such a position that the exchange method is a maximizing method because the lower bound of the T-deviation (13) is raised during each step monotonically.

List of Refrences

(1) STIEFEL, On numerical approximation, University of Wisconsin Press, 1959

(2) E. Stiefel, Ueber diskrate und linear Tschebyscheff Approximation to appear in Zeitochrift fur numeusche Mathematike, Springer, Heidelberg)

(3) A.A. Goldstain, W. Chenery, On the method of descent in convex domain and its application to the minimal approx of overdetermined astronontics, math. pre-print No. 1,1956)

(4) A.A. Goldstein, w. Chenery, Note on paper by Zuhoritsku Concerning the Tcheycheff problem for linear eg^s (Convair astronautics, mathe pre-print 8, 1957)

(5) DORFMAN, R,P.A. SAMUELSON, and R.M. SOLOW, Linear Programming and Economic Analysis,

(6) Gass, SAUL I., Linear Programming : Methods and applications, New York, N.Y. 1958

(7) KOOPMANS, TJALLING C. (edit), Activity Analysis of Production and Allocation, Monograph No. 13 of the cowles commission, New York, N.Y., 1951.

(8) Vajda, S., The Theory of Games and Linear Programming, London, 1956.

(9) Vajda, Readings in Linear Programming, London, 1958.

(10) FRISCH, RAGNAR A. K., "The Multiplex Method for Linear Programming sanklya, 1957, 18,329-362.

(11) KELLEY, JAMESE., Jr., "An Application of Linear Programming to Curve Fitting", Journal of the Society of industrial and applied mathematics, 1958 - 6, 15-22.

(12) KENDALL, M.G. and STUART, S., The advanced theory of Statistics, part II, Charles Griffin & Company limited, London, 1955

(13) AMER., R. Basic Principles in Linear Programming, Memo No. 483 I.N.P. Sept. 1964

(14) AMER, R. The Simplex Algorithm in a compact Form, Memo. No. 484, INP Sept. 1964

(15) AMER, R. A New Two-Phase Method For Linear Programming with Inequality Constraints, Memo. No. 485, I.N.P. Sept. 1964.