ECONOMICS OF PLANNING
Vol. 3, No. 3, December 1963
Printed in Norway

The Nonplex Method in its Subconditional Form

MAIN POINTS OF THE THEORY

By
Ragnar Frisch
assisted by
Håvard Alstadheim
University of Oslo

I. INTRODUCTION

In our work at the Institute of Economics, University of Oslo, on advanced methods of national and supra-national economic planning we have to an ever increasing extent been confronted with mathematical programming problems of a strongly nonlinear character. Without a reasonably effective method of handling such problems our progress would have come to an end. We have therefore been forced to take up the computational problem as a special part of our research work.

The nonplex method (as distinct from the multiplex and simplex methods) is developed in an attempt to attack mathematical programming problems which are so strongly non-linear that the admissible region may be non-convex and/or the preference function non-concave, and where the number of variables is very large.

The nonplex method in its pure *basis* form is a method where all the *equational conditions* can be and are solved in such a way that all the variables are expressed as *single-valued* functions of a number of basis variables. This form is discussed in considerable detail – including flow charts for automatic computation – in the Institute memorandum of 4 October 1963 (in Norwegian) "Nonplex-metoden i dens rene basisform" by Ragnar Frisch in cooperation with Kåre Edvardsen and assisted by Håvard Alstadheim. Work is in preparation for the detailed coding of the nonplex method in this form on UNIVAC 1107 (the largest electronic computer in Scandinavia) recently installed at the Norwegian Computing Centre in Oslo. In the 4 October 1963 memorandum we confined ourselves to considering the case where the dependant variables as well as the preference function are quadratic functions of the basis variables. But we made no assumption about the non singularity or the positive – or negative definiteness of the quadratic forms involved. This is a fundamental feature of the work, because it means that we are no longer in

a position to say that a local optimum is also a global one. This, of course, is the real difficulty in the more general types of programming work.

As explained rather fully in the introductory section of the 4 October 1963 memorandum there are several reasons for trying to handle the problem in *its subconditional* form, i. e. without having to proceed by means of a complete reduction to the basis equations. Some of these reasons are: Saving of storing space, the need for handling very general types of equations, say algebraic equations, the need for rapid change of primary coefficients, the desirability of being able to handle problems where it is very laborious or even impossible as a practical proposition to solve the equations explicitly in the form of single valued basis functions etc.

In the 4 October 1963 memorandum it was announced that the nonplex method in its subconditional form would be discussed in a subsequent memorandum. The main points of the theory of this further work is hereby presented.

The presentation in the present article will be made compact and brief, making use of many concepts that have been more fully explained in the 4 October 1963 memorandum and in other earlier memoranda.

It is hoped that the nonplex method in its subconditional form can be coded for UNIVAC 1107 in a not too distant future, provided sufficient financial support materializes.

II. THE VARIABLES AND THE BOUNDS

Let x_i ($i = 1, 2 \ldots N$, or shorter $i =$ all) be a complete list of all the variables to be considered. They may for instance be the values in each year in the planning period of each concrete entity such as consumption, production in each of a number of sectors, investment possibilities in each of a number of investment channels etc. This completely dynamic aspect of the analysis is one of the circumstances which tends to increase the number of variables.

For each such variable is specified a lower and an upper *bound*, i. e.

$$\underline{x}_i \leqq x_i \leqq \overline{x}_i \qquad (i = \text{all}) \tag{2. 1}$$

where $\underline{x}_i$ and $\overline{x}_i$ are given constants, $\underline{x}_i \leqq \overline{x}_i$.

As special cases we may have for some or all i

$$\underline{x}_i = -\infty \quad \text{and/or} \quad \overline{x}_i = +\infty \tag{2. 2}$$

which means that the variation of the variables x_i for which (2. 2) holds, is unbounded in one direction or in both directions.

III. EQUATIONAL CONDITIONS

3a. Generalities

We assume that the variables specified in section 2 are subject to satisfying m independent equations – the *standard equations* – which we write in the implicit form

(3a. 1) $$a_g\,(x_1, x_2 \ldots . x_N) = 0 \qquad (g = 1, 2 \ldots m)$$

In principle we may then choose (usually in many different ways) a set of $n = N - m$ of the variables (n = number of degrees of freedom), say

(3a. 2) $$x_h \qquad (h = u, v \ldots w, \text{ or shorter } h = \text{bas})$$

in such a way that these n variables are functionally independent in the model and further such that all the other $m = N - n$ variables x_j

(3a. 3) $$x_j \qquad (j = 1, 2 \ldots (\text{except } u, v, \ldots w) \ldots N, \text{ or shorter } j = \text{dep})$$

can be expressed as functions of the variables (3a. 2). Let these functions be

(3a. 4) $$x_j = b_j\,(x_u, x_v \ldots x_w) \qquad (j = \text{dep}).$$

The variables (3a. 2) are called the *basis variables* and the x_j in (3a. 4) the dependent variables. The functions (3a. 4) are called the basis functions.

Conventionally we may introduce the further n "basis functions"

(3a. 5) $$b_k\,(x_u, x_v \ldots x_w) = x_k \qquad (k = \text{bas})$$

so that not only the m dependent variables x_j (j = dep) but all the N variables x_i (i = all) may be looked upon as being expressible in terms of the n basis variables (3a. 2), i. e.

(3a. 6) $$x_i = b_i\,(x_u, x_v \ldots x_w) \qquad (i = \text{all})$$

The case where the m basis equations are *single-valued* is a *special* case.

The set-up (3a. 1)–(3a. 6) was the one used in the 4 October 1963 memorandum.

3b. Subequations and subfunctions

We now generalize the set-up in the sense that we do *not* require that we are to use *all* the m equations (3a. 1) for the construction of basis functions. We leave the possibility open that we may *temporarily neglect* in (3a. 1) a certain number of the equations – to be called the subequations – and use only the remaining ones for the construction of basis functions. These basis functions we call *sub-basis* functions or shorter, the *subfunctions*. In principle the subfunctions may depend on a larger

number of basis variables than n, namely as many more basis variables as we have seggregated subequations from (3a. 1).

In principle we put no restriction on the way in which we should be permitted to segregate subequations. We may seggregate only a few (perhaps none, which would lead to a complete basisform as we used it in the 4 October 1963 memorandum) or we may proceed very far and at the limit perhaps let all the equations in (3a.1) be subequations.

In practice we will select for subequations those which are mathematically the most *difficult to handle,* in particular those whose inclusion will make it difficult, perhaps next to impossible to arrive at manageable forms of the basis equations. The more equations we seggregate as subequations, the more machine time we must be prepared to use at subsequent stages, but in turn the more mechanized and simple may the individual operations in the successive rounds be.

3c. Absolute value deviations or square deviations?

The main idea of the subequations is that we aim at assuring the fulfilment of these equations by adding a suitable *penalty term* in the search function which we are going to maximize, cf. (4. 7), and to consider all the variables that occur in this search function as *independent* variables. To insure generality let us formally assume that all the variables x_i (i = all) may occur in the penalty term. If they do not, this is simply a special case.

The most obvious and straightforwarded way to construct the penalty term is simply to take the sum of the absolute values, or the sum of the squares, of the neglected functions a_g $(x_1, x_2 \ldots x_N)$ in (3a. 1) and let this sum (multiplied by a suitable scalar penalty coefficient) constitute the penalty term in the search function.

Sometimes, however, it may be better to proceed in a little more sophisticated way, namely to introduce a number of *subsearch* condition functions

$$C_\sigma (x_1, x_2 \ldots x_N) \qquad (\sigma = \text{sub}) \tag{3c. 1}$$

which have the property that their being equal to zero is a necessary and sufficient condition for the fulfilment of the subequations. The letter C in (3c. 1) may be thought of as standing for (equational) *conditions.* The penalty term in question will then be an arbitrarily chosen constant penalty coefficient multiplied by a sum of the absolute values, or of the squares, of the subsearch condition function (3c. 1), or possibly some

other positive definite functions of the subsearch condition functions (3c.1).

The choice between absolute values and squares, or possibly some transformation which will still make the sum positive definite and equal to zero when and only when all the C_σ are zero, is a practical and conventional question. If we use squares, the sum will be continuous and with continuous derivatives provided the functions C_σ themselves have these properties.

Continuous derivatives are unquestionably an advantage but the squaring does consume some machine time since it has to be done over and over again in the several rounds of the programming algorithm. I have also an intuitive feeling that even in principle the sum of absolute values is better than the sum of squares.

I cannot substantiate this feeling theoretically, but I can point to some empirical evidence. See for instance the numerical example in section 3e. At this writing my preference is therefore for the sum of absolute values even though this will entail discontinuities in the derivatives.

The discontinuities which will occur when we work with absolute values is after all not very serious, they can be handled by *choosing* between the forward and the backward derivatives, an operation which the machine can do very quickly whether it concerns a total or a partial derivative.

Fig. (3c. 2) illustrates the four main cases in a function y of the single variable x. In the two upper cases the signs of the two derivatives – the forward and backward – are the same, in the two lower cases the signs are opposite.

The forward derivative with respect to a single variable is defined by

$$\left(\frac{dy}{dx}\right)^{+} = \operatorname*{Lim}_{\triangle x \to +0} \frac{y(x+\triangle x) - y(x)}{\triangle x} \tag{3c. 3}$$

The backward derivative is defined by

$$\left(\frac{dy}{dx}\right)^{-} = \operatorname*{Lim}_{\triangle x \to -0} \frac{y(x+\triangle x) - y(x)}{\triangle x} \tag{3c. 4}$$

In (3c. 3) $\triangle x$ tends towards zero through *positive* values, while in (3c. 4) $\triangle x$ tends towards zero through *negative* values.

In the case of a function y of several variables $x_1, x_2 \ldots$ etc. where all the partial derivatives $\left(\frac{\partial y}{\partial x_i}\right)$ are *continuous*, the *gradient* is simply defined in the traditional way as the vector whose components are

$$\left(\frac{\partial y}{\partial x_i}\right) = \text{component No. i of the gradient vector.} \tag{3c. 5}$$

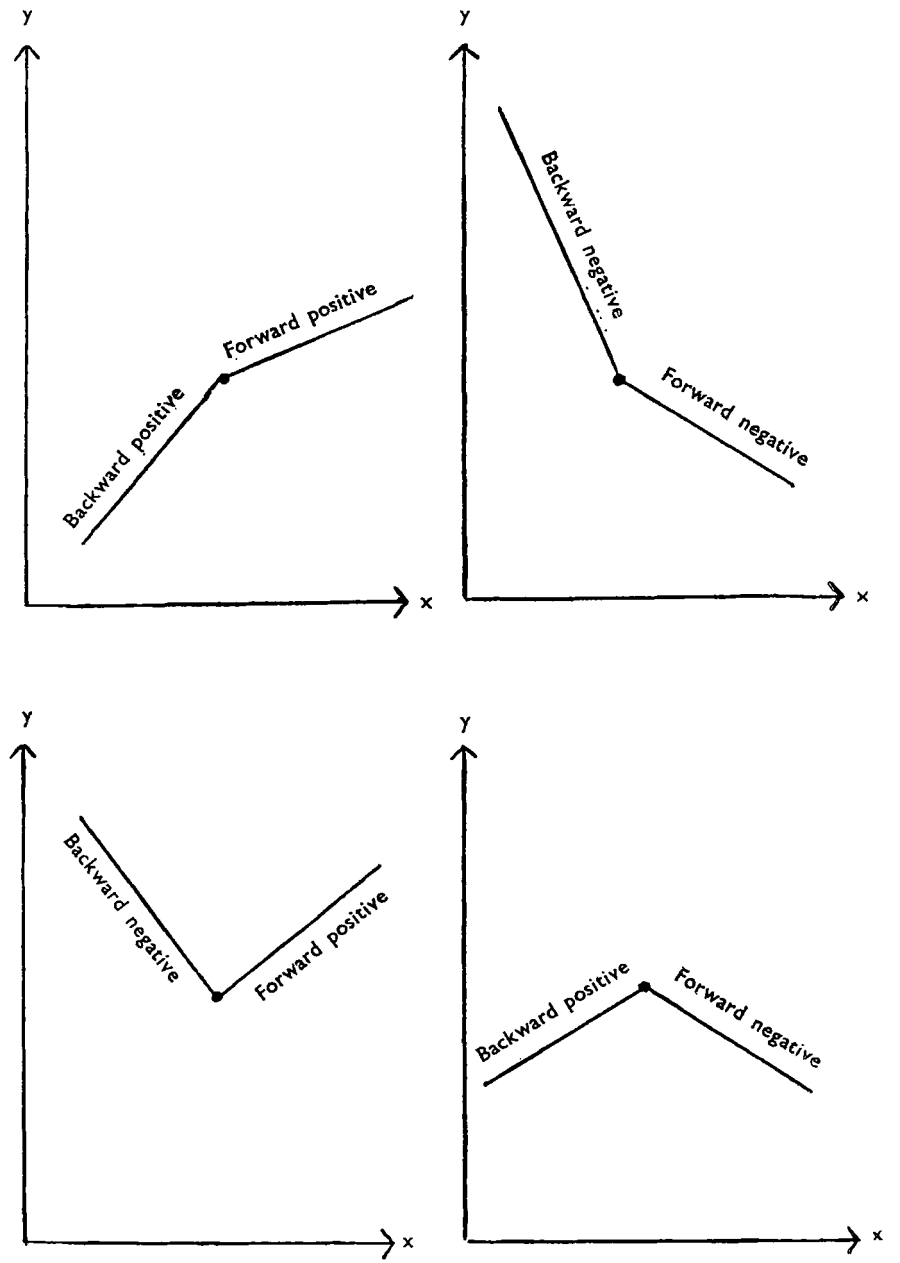

Fig. (3c. 2). Four cases of the forward and the backward derivative (tangent).

If we move in the direction (3c. 5) i. e. if we move from the point x_i $(i = 1, 2 \ldots)$ to the neighbouring point $x_i + \triangle x_i$ $(i = 1, 2 \ldots)$ where

$$\triangle x_i = \lambda \left(\frac{\partial y}{\partial x_i}\right) \qquad \lambda \text{ pos} \tag{3c. 6}$$

we are in the continuous case sure to *increase* the value of y, if at least one of the gradient components (3c. 5) is different from zero, and the positive factor λ is small enough. Indeed, in this case, we have to the first approximation

$$\triangle y = \Sigma_i \left(\frac{\partial y}{\partial x_i}\right) \triangle x_i, \text{ and hence}$$

$$\triangle y = \lambda \Sigma_i \left(\frac{\partial y}{\partial x_i}\right)^2 \tag{3c. 7}$$

And this expression is effectively positive if at least one of the gradient components is effectively different from zero.

To reach a similar definition in the case where the forward and backward derivative may be different, we must for each of the independent variables make an appropriate choice between these two derivatives.

By a simple graphical inspection of fig. (3c. 2) and an inspection of the somewhat more elaborate figure which we would get by also taking account of different alternative combinations of the *steepness* of the slopes, the following rule will determine the gradient components d_i which will lead to a positive increase – and the *most* positive one in the absolute value – of y (if a positive increase is at all possible, which it will be in all the four cases in fig. (3c. 2) except the last one which satisfies the first order condition for maximum):

(3c. 8) *Rule for optimal determination of the gradient component* d_i.

If $\left(\frac{\partial y}{\partial x_i}\right)^+$ is non positive and $\left(\frac{\partial y}{\partial x_i}\right)^-$ non negative – the case illustrated in the bottom right diagram in fig. (3c. 2) – put $d_i = 0$

In all other cases put d_i equal to that one of the two numbers $\left(\frac{\partial y}{\partial x_i}\right)^+$ and $\left(\frac{\partial y}{\partial x_i}\right)^-$ which has the *largest absolute value*. If the two absolute values are equal, decide the choice by random drawing.

In the case of strictly continuous partial derivatives the rule (3c. 8) yields for d_i the traditional gradient component $\left(\frac{\partial y}{\partial x_i}\right)$.

Whatever the d_i, determined by (3c. 8) the sum

(3c. 9) $$\triangle y = \sum_i d_i \triangle x_i$$

where

(3c. 10) $$\triangle x_i = \lambda d_i \qquad \lambda \text{ pos}$$

will always be effectively positive if at least one of the d_i is different from zero, the sum (3c. 9) is indeed equal to

(3c. 11) $$\triangle y = \lambda \sum_i d_i^2$$

Neither in the strictly continuous case (3c. 7) nor in the possibly discontinuous case (3c. 11) will $\triangle y$ necessarily be exactly the change in y which is produced by the $\triangle x_i$, but in both cases $\triangle y$ may be taken as an approximation. And as such (3c. 11) is just as plausible as (3c. 7) when the d_i are determined by the optimal rule (3c. 8).

3d. *Step functions and functions with discontinuous partial derivatives*

A step function is a function which is such that the ordinate itself in certain points makes a discontinuous jump. This is an even more serious type of discontinuity than that of discontinuous jumps in derivatives. Fig. (3d. 1) illustrates the case where both types of discontinuities occur.

Fig. (3d. 1) illustrates a case when we are in the point x_0 and *have*

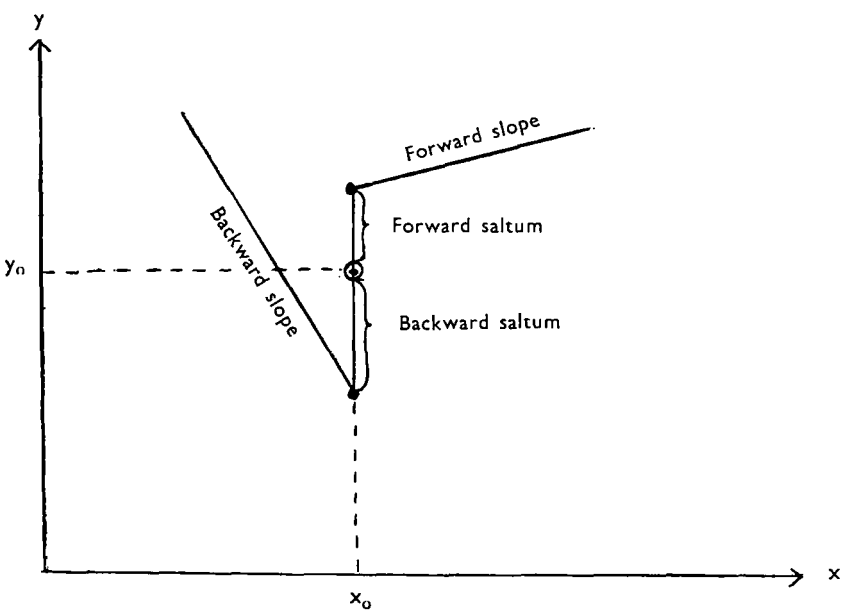

Fig. (3d. 1). Example of both types of discontinuities.

arrived in this point by a process which in some way or other has defined the value y_0 which the ordinate is to have in this point (a value which, of course, must lie between the lower and the upper bound for the step function in this point x_0).

In this case a forward move means that we first have to add to the ordinate the forward saltum and then continue along the forward slope. And a backward move means that we first have to subtract the backward saltum and then continue along the backward slope.

In this case the *sign* of the increment in the function cannot be indicated simply by specifying in what *direction* we are to move (as it could when the only kind of discontinuity present was a jump in the derivative). We must now also specify *how far* we are to go. For instance if we move forward only a small distance, this will obviously be more "profitable" than to move backward a small distance. But if we move backward a sufficiently long distance the "overhead cost" represented by the backward saltum will be more than covered, so that a sufficiently long move backward will be more "profitable" than an equally long move forward.

An analysis of such cases cannot be made simply by considering *continuous changes in the independent variable*, but would require some sort of "Quantum theory" of the changes in the independent variable.

In several variables the analogon of (3d. 1) means that we would have to compare various alternative beams of variation that do not start from the very point where we find ourselves, but start from some neighbouring point.

Possibly even such cases might be handled by the nonplex method in its subconditional form, but at best this analysis would be extremely complex. I shall therefore in the sequel disregard the case of stepfunctions and confine myself to the study of search functions which are continuous but whose partial derivatives may be discontinuous, and discontinuous in such a way that the situation can be handled simply by distinguishing – for each variable taken separately – between the forward and the backward partial derivative.

3e. *An example : Minimizing the sum of the absolute values of the deviations from the arithmetic average*

Consider the N variables $x_1, x_2 \ldots x_N$ and assume that we want to minimize

$$C = \sum_{j=1}^{N} |x_j - a| \tag{3e. 1}$$

where

$$a = \frac{x_1 + x_2 + \ldots + x_N}{N} \tag{3e. 2}$$

The exact solution of this problem is, of course, obvious. It simply consists in putting all the variables equal. This common magnitude of all the variables may be arbitrary, which means that the optimal solution has one degree of freedom.

We have chosen the example in this simple way in order that we may be able to evaluate precisely the success of the technique based on forward and backward partial derivatives.

I shall first give the explicit expressions for the partial derivatives of the function C defined by (3e. 1). They are

$$\text{(3e. 3)} \qquad \left(\frac{\partial C}{\partial x_i}\right)^{+} = \begin{cases} 2\left(1-\frac{\mu}{N}\right) & \text{if } x_i > a \\ -\frac{2\mu}{N} & \text{if } x_i < a \\ 2\left(1-\frac{\mu+1}{N}\right) & \text{if } x_i = a \end{cases}$$

$$\text{(3e. 4)} \qquad \left(\frac{\partial C}{\partial x_i}\right)^{-} = \begin{cases} \frac{2\gamma}{N} & \text{if } x_i > a \\ -2\left(1-\frac{\gamma}{N}\right) & \text{if } x_i < a \\ -2\left(1-\frac{\gamma+1}{N}\right) & \text{if } x_i = a \end{cases}$$

where

$$\text{(3e. 5)} \qquad \begin{cases} \mu \text{ is the number of variables that are } \textit{larger} \text{ than } a \\ \gamma \text{ is the number of variables that are } \textit{smaller} \text{ than } a \\ N-(\mu+\gamma) \text{ is the number of variables that are } \textit{equal} \text{ to } a. \end{cases}$$

To prove the formulae (3e. 3)–(3e. 4) we note that for any function $y(x_1, x_2 \ldots x_N)$ with continuous partial derivatives $\frac{\partial y}{\partial x_i}$ we have

$$\text{(3e. 6)} \qquad \frac{\partial |y(x_i, x_2 \ldots x_N)|}{\partial x_i} = \begin{cases} \operatorname{sgn} y \cdot \frac{\partial y}{\partial x_i} \text{ if } y \neq 0 \\ \left\{ \begin{array}{l} \text{forward } + \left|\frac{\partial f}{\partial x_i}\right| \\ \text{backward } - \left|\frac{\partial f}{\partial x_i}\right| \end{array} \right\} \text{ if } y = 0 \end{cases}$$

where

$$\text{(3e. 7)} \qquad \operatorname{sgn} y = \begin{cases} +1 \text{ if } y > 0 \\ -1 \text{ if } y < 0 \\ 0 \text{ if } y = 0 \end{cases}$$

Letting

$$y = x_j - a \tag{3e. 8}$$

and hence

$$\frac{\partial y}{\partial x_i} = e_{ji} - \frac{1}{N} \qquad (i=\text{all}) \tag{3e. 9}$$

where e_{ji} are the unit numbers

$$e_{ji} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ otherwise} \end{cases} \tag{3e. 10}$$

we see that

$$\frac{\partial \mid x_j - a \mid}{\partial x_i} = \begin{cases} \operatorname{sgn}(x_j - a) \cdot (e_{ji} - \frac{1}{N}) \text{ if } x_j \neq a \\ \left\{ \begin{array}{l} \text{forward } + \left| e_{ji} - \frac{1}{N} \right| \\ \text{backward } - \left| e_{ji} - \frac{1}{N} \right| \end{array} \right\} \text{if } x_j = a \end{cases} \tag{3e. 11}$$

and therefore

$$\left(\frac{\partial \Sigma_j \mid x_j - a \mid}{\partial x_i}\right)^+ = \sum_{j \mid x_j \neq a} \operatorname{sgn}(x_j - a) \cdot (e_{ji} - \frac{1}{N}) + \sum_{j \mid x_j = a} \left| e_{ji} - \frac{1}{N} \right| \tag{3e. 12}$$

$$\left(\frac{\partial \Sigma_j \mid x_j - a \mid}{\partial x_i}\right)^- = \sum_{j \mid x_j \neq a} \operatorname{sgn}(x_j - a) \cdot (e_{ji} - \frac{1}{N}) - \sum_{j \mid x_j = a} \left| e_{ji} - \frac{1}{N} \right| \tag{3e. 13}$$

Incidentally this shows that in any point $(x_1, x_2 \ldots x_N)$ where there are *no* variables exactly on the average a the partial derivative with respect to any x_i is continuous, i. e. the forward and the backward partial derivatives equal.

Splitting the first sum in (3e. 12) and (3e. 13) into those j for which $x_j > a$ and those j for which $x_j < a$ we get

$$\left(\frac{\partial \Sigma_j \mid x_j - a \mid}{\partial x_i}\right)^+ = \sum_{j \mid x_j > a} (e_{ji} - \frac{1}{N}) - \sum_{j \mid x_j < a} (e_{ji} - \frac{1}{N}) + \sum_{j \mid x_j = a} \left| e_{ji} - \frac{1}{N} \right| \tag{3e. 14}$$

$$\left(\frac{\partial \Sigma_j \mid x_j - a \mid}{\partial x_i}\right)^- = \sum_{j \mid x_j > a} (e_{ji} - \frac{1}{N}) - \sum_{j \mid x_j < a} (e_{ji} - \frac{1}{N}) - \sum_{j \mid x_j = a} \left| e_{ji} - \frac{1}{N} \right| \tag{3e. 15}$$

By considering the three alternative cases where the affix i occurs in the first, second or third term in (3e. 14)–(3e. 15) we get (3e. 3)–(3e. 4).

In the case where the affix i occurs in the third term we note in particular that we have

$$\sum_{j \mid x_j = a} \left| e_{ji} - \frac{1}{N} \right| = 1 - \frac{1}{N} + \sum_{j \mid \substack{x_i = a \\ j \neq i}} \left| -\frac{1}{N} \right| \qquad (\text{if } x_i = a) \tag{3e. 16}$$

The summation in the right member of (3e. 16) contains N–$(\mu+\gamma)$–1 terms, hence

$$\sum_{j \mid x_j = a} \left| e_{ji} - \frac{1}{N} \right| = 2\left(1 - \frac{1}{N}\right) - \frac{\mu+\gamma}{N} \qquad \text{(if } x_i = a\text{)} \tag{3e. 17}$$

Since in the case $x_i = a$ the first two terms in (3e. 14)–(3e. 15) yield $-\frac{\mu}{N} + \frac{\gamma}{N}$ we get the last line in (3e. 3)–(3e. 4).

The derivation of the first and second line in (3e. 3)–(3e .4) is straightforward.

There are several plausibility tests we may apply to (3e. 3)–(3e. 4) for instance:

I. Under a partial variation of a variable x_i which is *above* the average, C will be increasing with this variable both before and after the special point we are considering. Vice versa for a variable x_i which is *below* the average.

II. Under a partial variation of a variable x_i which is *on* the average C will increase whether we move this variable to the left or to the right of the special point we are considering.

III. If $N = 1$, in which case C is constantly equal to zero, the first two lines in (3e. 3)–(3e. 4) do not apply, and in the last line we get 0 since now $\mu = \gamma = 0$.

If we apply the optimum rule (3c. 8) to (3e. 3)–(3e. 4) we get the following very simple and very plausible determination of the direction numbers d_i for the function $(-C)$, i. e. the direction numbers to be applied if we want to minimize C:

$$\begin{cases} \text{For any } x_i \text{ which is } \textit{above} \text{ the average, i.e. } x_i > a, \text{ put } d_i = -2\left(1 - \frac{\mu}{N}\right) \\ \text{For any } x_i \text{ which is } \textit{below} \text{ the average, i.e. } x_i < a, \text{ put } d_i = +2\left(1 - \frac{\gamma}{N}\right) \\ \text{For any } x_i \text{ which is } \textit{on} \text{ the average, i. e. } x_i = a, \text{ put } d_i = 0 \end{cases} \tag{3e. 18}$$

This determines the gradient-*direction* in which we ought to move if we want to minimize C, cf. (3c. 10). But *how far* should we move in this direction?

Without attempting to use any complicated procedure we may adopt the time honoured NEWTON formula for locating approximately a zero

of a given function when we start from any point in the vicinity of such a zero. This formula in the case of the function C amounts to putting

$$C + \triangle C = 0 \tag{3e. 19}$$

In order to apply this to the present case we must determine

$$\triangle C = \sum_i \frac{\partial C}{\partial x_i} \triangle x_i \tag{3e. 20}$$

Separating here the summation over i according to the three cases in (3e. 18) we get

$$\triangle C = \sum_{i|x_i > a} \frac{\partial C}{\partial x_i} \triangle x_i + \sum_{i|x_i < a} \frac{\partial C}{\partial x_i} \triangle x_i + \sum_{i|x_i = a} \frac{\partial C}{\partial x_i} \triangle x_i \tag{3e. 21}$$

In the first term here $\triangle x_i$ is equal to $-2\left(1-\frac{\mu}{N}\right)\lambda$ according to (3e. 18), cf. (3c. 10). Since this is a negative quantity we must use the backward formula (3e. 4) so that the first term in (3e. 21) becomes

$$\sum_{i|x_i > a} \frac{2\gamma}{N} \cdot \left(-2\lambda\left(1-\frac{\mu}{N}\right)\right) = -4\frac{\gamma}{N}\lambda \sum_{i|x_i > a} \left(1-\frac{\mu}{N}\right)$$

The last summation here gives $\mu\left(1-\frac{\mu}{N}\right)$. The first term in (3e. 21) therefore reduces to $-4\frac{\mu\gamma}{N}\left(1-\frac{\mu}{N}\right)\lambda$.

Similarly the second term in (3e. 21) reduces to $-4\frac{\mu\gamma}{N}\left(1-\frac{\gamma}{N}\right)\lambda$.

The third term in (3e. 21) is zero by the last line in (3e. 18). Collecting the terms we get

$$\triangle C = -4\frac{\mu\gamma}{N}\left(2-\frac{\mu+\gamma}{N}\right)\lambda \tag{3e. 22}$$

This expression is zero when and only when all the variables lie exactly on the average i. e. when and only when $\mu = \gamma = 0$. In all other cases it is effectively negative. This follows from the fact that we always have $0 \leqq \mu \leqq N-1$ and $0 \leqq \gamma \leqq N-1$.

Inserting (3e. 22) into (3e. 19) we get the following determination of λ

$$\lambda = \frac{N}{4\mu\gamma\left(2-\frac{\mu+\gamma}{N}\right)} C \tag{3e. 23}$$

And inserting this value of λ into (3c. 10), where now the d_i are determined by (3e. 18) we find the increments $\triangle x_i$ that we ought to attribute to the x_i, if we find ourselves in any given point $(x_1, x_2 \ldots x_N)$.

This gives rise to a sequence of rounds that we may hope will converge towards a point where all the x_i are equal, and hence equal to the average a.

A numerical example in three variables x_1, x_2, x_3 is given in the left part of tab. (3e. 24), where we start from the arbitrarily given point $(-3, -2, +8)$ which gives $a = 1$. It will be seen that the convergence is fairly rapid. The average remains constant in this example, but this is not a general phenomenon. Examples may be construed where the average is not constant. The behaviour of a is not particularly interesting since we know that the average represents one degree of freedom in the solution and we have only asked for *some* point $(x_1, x_2 \ldots x_N)$ where all the variables are equal.

It is interesting to note that in this algorithm the number of multiplications and/or divisions involved in each round is *independent of the number of variables N*. The only way in which the different variables intervene is through the *sorting process* where they are classified in the three categories $x_i > a$, $x_i < a$, $x_i = a$. A sorting process of this kind is very quickly done on the machine.

Tab. (3e. 24) *COMPARISON of the C-method and the Γ-method*

Round No.	By the absolute sum method $C = \sum_j \lvert x_j - a \rvert = \min!$						By the square sum method $\Gamma = \sum_j (x_j - a)^2 = \min!$					
	x_1	x_2	x_3	a	C	Γ	x_1	x_2	x_3	a	C	Γ
0	−3	−2	+8	1	14	74	−3	−2	8	1	14.00	74
1	$\frac{1}{2}$	$\frac{3}{2}$	1	1	1	$\frac{1}{2}$	−1	$-\frac{1}{2}$	$\frac{9}{2}$	1	7.00	$\frac{37}{2}$
2	$\frac{5}{4}$	$\frac{3}{4}$	1	1	$\frac{1}{2}$	$\frac{1}{8}$	0	$\frac{1}{4}$	$\frac{11}{4}$	1	3.50	$\frac{37}{8}$
3	$\frac{7}{8}$	$\frac{9}{8}$	1	1	$\frac{1}{4}$	$\frac{1}{32}$	$\frac{1}{2}$	$\frac{5}{8}$	$\frac{15}{8}$	1	1.75	$\frac{37}{32}$
4	$\frac{17}{16}$	$\frac{15}{16}$	1	1	$\frac{1}{8}$	$\frac{1}{128}$	$\frac{3}{4}$	$\frac{13}{16}$	$\frac{23}{16}$	1	0.88	$\frac{37}{128}$
5	$\frac{31}{32}$	$\frac{33}{32}$	1	1	$\frac{1}{16}$	$\frac{1}{512}$	$\frac{7}{8}$	$\frac{29}{32}$	$\frac{39}{32}$	1	0.44	$\frac{37}{512}$

Now let us compare this algorithm with the algorithm we would get by minimizing the function

(3e. 25) $$\Gamma = \sum_j (x_j - a)^2$$

This function has continuous partial derivatives. If we handle it in accordance with the same principle as we used when handling C, we get the sequence of rounds described in the right part of tab. (3e. 24).

It is interesting to note that in this case the convergence is *very much slower*. This is apparent whether we judge the *spread* of the values (x_1, x_2, x_3) in the two algorithms by using the C values as a means of comparison (compare the C column to the left with that to the right) or by using the Γ values as a means of comparison (compare the Γ column to the left with that to the right).

Another weak point in the algorithm obtained by minimizing Γ is that here each round involves a number of multiplications and/or divisions which *is of the order N*.

From any algorithm used for minimizing a function y we may deduce another algorithm by minimizing some function

(3e. 26) $$\Omega(y)$$

of the original y, where $\Omega(y)$ is a positive and effectively increasing function of y. The nature of this function may be characterized by its flexibility

(3e. 27) $$El.\,\Omega = \frac{d\Omega}{dy}\cdot\frac{y}{\Omega}$$

If Ω is taken as the function to be minimized instead of y, and we handle Ω by the same NEWTONIAN method as we handled y, the changes that emerge in the various variables x_i from round to round will be different. We get

(3e. 28) $$\underset{\Omega}{\triangle} x_i = \frac{1}{El.\,\Omega}\cdot\underset{y}{\triangle} x_i$$

where $\underset{\Omega}{\triangle}$ and $\underset{y}{\triangle}$ denote the increments that emerge in the Ω algorithm and in the y algorithm respectively.

If we choose the nature of the function Ω in a particular way *adapted to the nature of the individual functions* that ought to be zero – in this case the functions $\left(x_i - \frac{\Sigma_j x_j}{N}\right)$ – the convergence may be speeded up. In the right part of tab. (3e. 24) we might for instance try

(3e. 29) $$\Omega(y) = \sqrt{y}$$

That is to say we may put up the minimalization of $\sqrt{\Sigma_j (x_j - a)^2}$ instead of the minimalization of $\Sigma_j (x_j - a)^2$. Prima facie one might think that this would not make much difference, but (3e. 28) tells us that there *is* a difference, and this difference is all the more important the smaller the flexibility of Ω. In the case (3e. 29) we get

(3e. 30) $$El.\,\Omega = \frac{1}{2}$$

This means that the increments downward in each x_i column in the right part of tab. (3e. 24) should be just doubled. Looking at the passage from line 0 to line 1 this means for instance that instead of an x_1 change from -3 to -1, which is an increase of 2, we will have an increase of 4. In other words we will move from -3 to $+1$. In the x_2 column, instead of moving from -2 to $-\frac{1}{2}$, we will move from -2 to $+1$, and in the x_3 column, instead of moving from $+8$ to $+\frac{9}{2}$ we will move from $+8$ to $+1$. In other words already the very first round will give us an exact solution. But this extraordinary result is, of course, only due to our having chosen a transformation of the function $\Gamma = \Sigma_j (x_j - a)^2$ which is particularly adapted to this special example considered, namely the squarerooting of Γ.

If we want a general rule that is to be applied regardless of the particular nature of the individual functions that it is desired to bring down to zero, I would at this writing rather be inclined to construct the penalty term by taking the absolute values of the functions than by taking squares.

If we are not only concerned with minimizing C but also concerned with some entirely different purposes, cf. (4. 7) we will have to consider a *compromise* direction of move and comprise length of move in each round.

IV. THE PREFERENCE FUNCTION AND THE SEARCH FUNCTION

In a general programming problem there will be specified a certain *preference function* whose maximalization is the main problem of the analysis. The various bounds and equations we have discussed so far may be looked upon as *sideconditions* only in the maximalization of the preference function.

The preference function defined as a function of all the variables in the model, we call the *gross* form of the preference function. If in this gross form of the preference function we introduce the basis expressions for all the variables – whether we have decided to use a complete system of basis equations or only a subconditional system of basis equations – we get the net form of the preference function.

The preference function we shall denote

(4. 1) $$F(x_1, x_2 \ldots x_N)$$

This function will form the first part of our *search function*, i. e. of the

total function which we want to maximize. In (4. 1) those variables x_i which we have chosen to consider as dependent (not basis) variables will now have to be looked upon as functions of the basis variables.

A second part will be constituted by a penalty term derived by considering the misplaced variables, i. e. the variables that are either below the lower bound or above the upper bound in (2. 1). We may consider the misplacements either in the exact or in the *threshold* sense. The latter concept is derived by defining for each variable a certain threshold which indicates that we are in fact not too particular about the exactness with which the bounds are fulfilled. A precise definition of the threshold concept is given in previous memoranda. The structure of the penalty now considered will be built on the function

$$M = \sum_{i \mid \text{mis}} \varepsilon_i (x_i - x_i^*) \tag{4. 2}$$

where

$$x_i^* = \begin{cases} \bar{x}_i \text{ if } x_i > \bar{x}_i \\ \underline{x}_i \text{ if } x_i < \underline{x}_i \end{cases} \tag{4. 3}$$

and

$$\varepsilon_i = \begin{cases} -1 \text{ if } x_i > \bar{x}_i \\ +1 \text{ if } x_i < \underline{x}_i \end{cases} \tag{4. 4}$$

and 'mis' indicates a summation over all the misplaced variables. If M is multiplied by some conveniently chosen scalar penalty coefficient μ we get the second term in the search function.

Sometimes it may be useful to include a third and *special* penalty term which concerns those variables that are thresholdly bound attained, i. e. that lie within threshold nearness of one of their bounds. For these variables we may consider the sum

$$B = \sum_{i \mid \text{bat}} \in_i (x_i - x_i^*) \tag{4. 5}$$

where the summation runs over i for all the variables x_i that lie within threshold nearness of the bound x_i^* (bat = bound attained). This function B multiplied by some scalar penalty coefficient β constitutes the special penalty term. Its purpose is to give a special treatment to the variables that have already come thresholdly in their admissible interval. We might for instance want to attribute some importance to their not becoming too inadmissible in the further moves. The term (4. 5) might have been included with (4. 2), but then we would not have had a possibility of distinguishing between the values of the penalty coefficients for the two sets of variables.

Finally there is the penalty term built up on the functions (3c. 1). Tentatively we might – for the reasons explained in section 3c – aggregate them in their absolute value forms, i. e. put

$$C = -\sum_{\sigma \mid \text{sub}} |C_\sigma| \tag{4. 6}$$

where σ runs through the subconditional functions. The minus sign in (4. 6) simply stems from our convention that the search function is a function to be maximized (not minimized). The function C defined by (4. 6) will have to be multiplied by some conveniently chosen penalty coefficient γ.

The three scalar penalty coefficients μ, β, γ will have to be entered in the machine as parametres specifying the algorithm.

Collecting the terms we get a complete search function of the form

$$\Omega = F + \mu M + \beta B + \gamma C \tag{4. 7}$$

and it is this which is to be maximized in a sequence of rounds, analogous to those in tab. (3e. 24), but much more complex.