1970

Parametric Programming on an Analog Computer

Vinod B. Kumar

PARAMETRIC PROGRAMMING ON
AN ANALOG COMPUTER

BY

VINOD B. KUMAR

A thesis submitted
in partial fulfillment of the requirements for the
degree Master of Science, Department of
Mechanical Engineering, South Dakota
State University

January, 1970

PARAMETRIC PROGRAMMING ON

AN ANALOG COMPUTER

This thesis is approved as a creditable and independent investigation by a candidate for the degree, Master of Science, and is acceptable as meeting the thesis requirements for this degree, but without implying that the conclusions reached by the candidate are necessarily the conclusions of the major department.

Thesis Adviser Date

Head, Mechanical Engineering Department Date

ACKNOWLEDGEMENTS

The author wishes to express his deep sense of gratitude to Dr. R. P. Covert for the excellent advice and guidance received, in making this study a rewarding experience; and his sincere appreciation to Prof. J. F. Sandfort for making it possible to execute his graduate studies at this school. Thanks are also due Dr. F. C. Fitchen, Head, Electrical Engineering Department for his cooperation in the use of the analog computer.

VBK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Tables | Page

CHAPTER I

INTRODUCTION

Most activities are undertaken with the intent to optimize something. In business and industrial organizations the goal is frequently to maximize profit or minimize costs. Although this seems straight forward, the complexity of the relations have made simple mathematical solutions impossible. Increasingly, it has been found that major segments of the total management decision structure can be represented by a set of mathematical constraints[1] and a mathematical objective function.[2] These constraints can be solved such that the objective function is optimized through the power of techniques such as linear and nonlinear programming.

Linear programming is advantageously being currently applied in many industrial and business fields, including the following:

1. Product Allocation.
2. Distribution and Shipping.
3. Market Research.

---

[1]Constraints are represented as equalities or inequalities and could be broadly classified as follows:

a. Limitations on the usage of the resources available.
b. Interrelationship between these resources.
c. Conditions required to be fulfilled by the totality of the resources (1).

[2]Objective function is usually formulated in economic terms: Maximizing profit or minimizing cost. Minimizing risk, and partitioning large problems into subproblems are the recent developments, involving more complex objectives (1).

4. Job and Salary Evaluation.
5. Blending of Ingredients.
6. Material handling.
7. Production Planning.
8. Trim Scheduling.
9. Traffic Analysis.
10. Production Scheduling and Inventory Control.
11. Structural Design.

Mathematical models with either nonlinear constraints, nonlinear objective function or both, have been in certain cases approximated and solved (2). Nonlinear problems with convex objective function and linear constraints have been reduced to linear programming problems whose objective function approximates the convex function (3).

A major task in the development of realistic linear programming models is the determination of the accuracy and reliability of the numerical values for the relations and constraints. Further, it may be necessary to examine the behaviour of the solutions when the input data differ from the chosen values, or may even be stochastic in nature. In these cases the problem is considered as a parametric programming problem. The areas in which the parameters can be varied are:

1. Coefficients of restrictions.
2. Coefficients of objective functions.
3. Limitations of constraints.

One of the powers of the solution techniques is that, unlike simultaneous equations, the number of constraining equations and the number of variables do not need to be equal. This raises additional problems as well as raising the possibility of multiple solutions.

Solutions to linear programming problems can be obtained by trial and error, simplex calculations and analog computer simulation.

Trial and Error Method.

If the system of inequalities is treated as a system of equalities the resulting sets of equations can be solved simultaneously. To do this the excess variables must be equated to zero or excess constraints must be ignored.

The set of solutions would include all possible intersections of two or more constraints. The solution which violates restrictions would need to be rejected and the objective function computed for all remaining vertices to obtain the optimum answer. If the number of variables and constraints are large, the number of solutions is larger and the computations very long and tedious. Obviously, parametric programming cannot be easily conducted using this method, because the total computations involved would be prohibitively long.

Simplex Method.

The most successful and best known general procedure for optimal solution of linear sets of inequalities is the simplex method or its modifications; used in conjunction with high speed digital computers.

The simplex method uses the slack variables as its starting basis or solution and then proceeds to find a series of new basis. All variables, including the slack variables, are considered as candidates in the formation of each new basis. The net gain (or reduction in cost penalty) per unit associated with all nonbasis variables are calculated. The variable with the largest net gain per unit used, is then selected for the new basis. For every new variable selected, an old one is deleted. This process of selecting a new variable and deleting an old one is called pivot selection. Once the new basis is selected the entire procedure is repeated, until the objective function is optimized.

Mathematically speaking, the simplex method utilizes the technique of matrix[3] inversion directed by the coefficients of the objective function.

Figure 1.1 illustrates the flow diagram of the simplex method as used on digital computers.

A geometric interpretation of the simplex method is that an objective point starts from the origin of the convex polyhedron, formed by the constraints of the problem. With every pivot selection the objective point is moved from one corner to another adjacent corner in a manner such that at each step the objective function is increased (or decreased in case of minimizing).

---

[3]Matrix formed by the coefficients of the constraint equations.

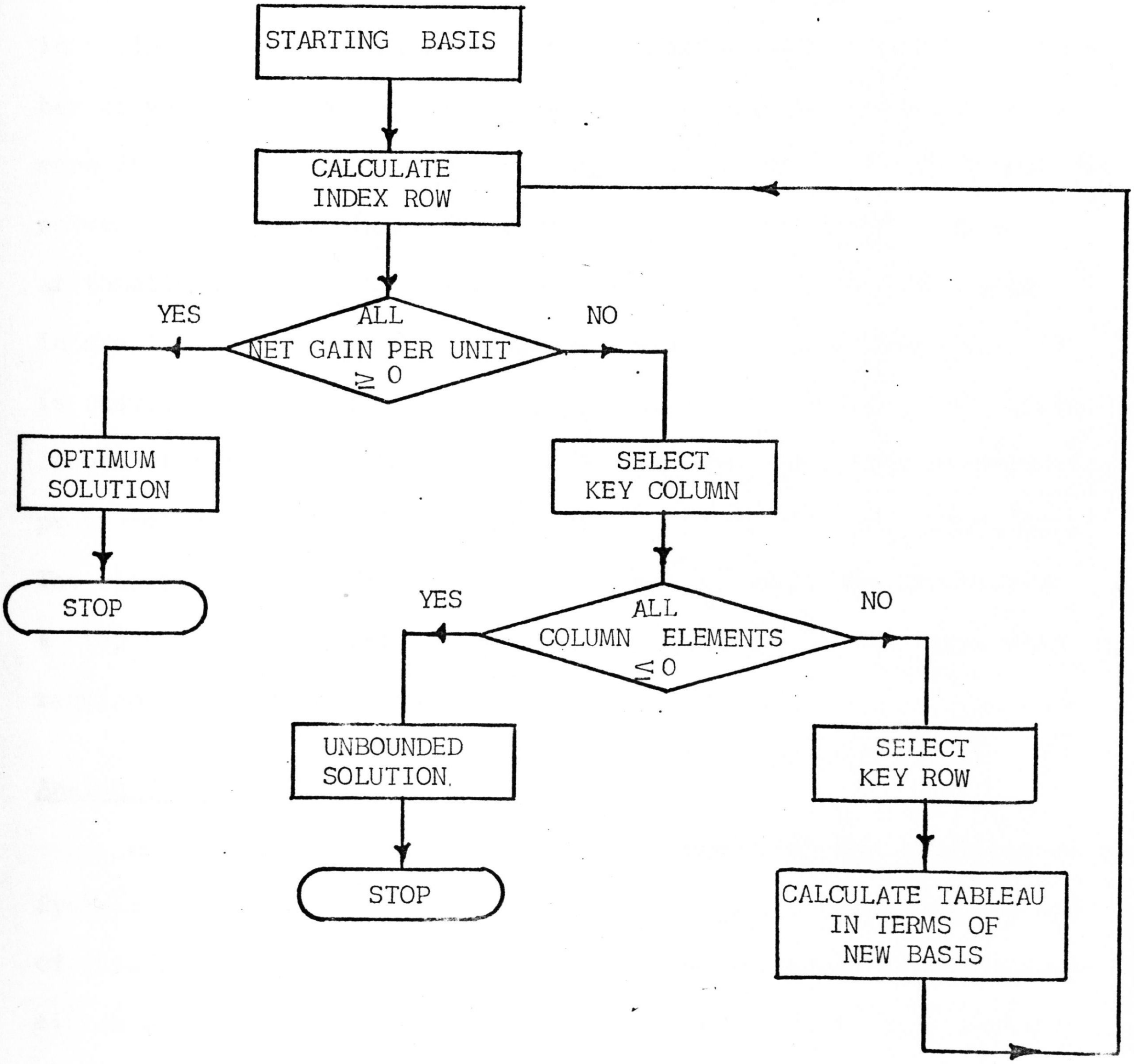


Figure 1.1 Flow Diagram of Simplex Method

The simplex method enables the optimum solution to be attained in a finite number of steps, the exact number depending upon the number of variables and constraints involved in the problem; but never more than twice as many steps as there are constraints. Most problems solved by this method require a great deal of relatively simple arithmetic computation. Hence the digital computer with its high internal computing speed is an excellent aid in their solution. It is unsurpassed in its ability to solve large problems in a reasonable length of time. Unfortunately, procedures for conducting parametric programming and sensitivity analysis are not equally well developed. The digital computer using the simplex method seeks the solution in a step by step numerical calculation, and changes in the input data require complete new solutions.

Analog Computer Simulation.

Analog computers are normally associated with the solutions of dynamic or time variant problems. We shall not be really making use of this common feature, but instead the dynamic error reduction associated with negative feedback. When the error reduction is combined with the concept of the objective function as a driving force, we in fact have an analog steepest ascent procedure.

In the analog computer approach, an objective point is generated in the convex polyhedron formed by the constraints, and forced to travel along the gradient of function to be optimized, until it strikes a barrier. The point then experiences an additional force at right angles to the barrier. As a result the point travels along the varrier to the solution vertex where the objective function takes on an optimum value.

The analog computer solves the problem in a dynamic fashion and each input data can be represented by a potentiometer setting. The high computing speeds available are ideally suited for investigating the effect of parametric changes. The analog computer model makes possible the use of the analyst's judgement and creative reasoning during the investigation, since he can see immediately the results of his decisions and its effect on the validity of the problem solution. Moreover the analog computer model can also be used to find "optimal" combinations of the input data for achieving desired results.

"Only a limited number of references as can be found in the subject area of solving linear programming by using analog computers." Jackson (4) discusses the potential of analog computers in the field of operations research. In addition to the linear programming model, he illustrates the role of analog computation in simulating economic and stochastic models. Pyne (5) gives a general procedure to solve linear programming problems on an analog computer. He also provides a method for calculating the various elements (resistances and voltages) to represent a specific set of constraints and an objective function. However,

---

[4]Hyperplane, corresponding to a constraint of the problem.

he does not develop the model beyond the elementary stage nor demonstrate how the model can be applied to parametric programming. Further he does not use the model to obtain the information found in the index row of the final simplex tableau.

The object of this investigation is to use the general purpose analog computer to simulate a linear programming problem and to extend the general purpose model developed by Pyne, to handle parametric programming, sensitivity analysis, and post-optimum analysis. Moreover, it is also intended to demonstrate how the information provided by the index row of the simplex tableau could be made available from this dynamic analog model.

CHAPTER II

THE ANALOG COMPUTER

A review of the procedure of operation of the analog computer and its components, enables a better visualization of the use of the analog computer in the solution of convex polyhedron sets.

The analog computer consists of a number of electronic units called operational amplifiers used in conjunction with simple resistor-capacitor circuits. In addition, specialized devices like diodes, potentiometers, relays and function generators are used as auxiliary equipment for the simulation of linear and nonlinear programming problems. As the name suggests, solutions to problems by the analog computer are accomplished by analogy. The computer is programmed such that the equations of the problem have the same mathematical form as the circuit equations. Because of parallel logic, the problems are dynamically solved, rather than iterative. In linear programming problems, input voltages will represent physical quantities such as machine hours and sales limitations while the output voltages represent the number of units to be produced. The magnitudes of the voltages are related to the numerical values of the physical quantities, by use of scale factors. The computer operates in a continuous fashion with respect to time.

The DC Amplifier

The DC amplifier is the most fundamental building block of the electronic analog computer. It is essentially a high gain (10^4 - 10^8) direct current amplifier. It can be made to perform the basic mathematical operations such as inversion, summation, multiplication by constants, and integration when used in conjunction with appropriate input and feedback impedances.

The conventional way of representing an amplifier is by a triangle with base at the input end and apex at the output end as shown in Figure 2.1

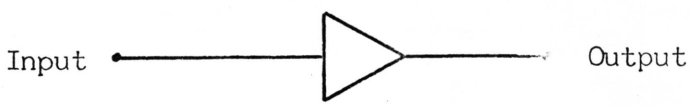
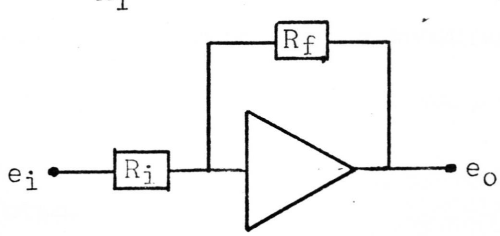


Figure 2.1 Symbol- Amplifier

In the simple circuit shown in Figure 2.2, R_f is called the feedback resistor and R_i is called the input resistor. The output e_o is given by

$$e_o = -\frac{R_f}{R_i} e_i \qquad [2.1]$$



$$e_o = -\frac{R_f}{R_i} e_i$$

Figure 2.2 Operational Amplifier

If R_f is equal to R_i we have a circuit for an inverter and e_o is equal to $- e_i$.

The Summing Amplifier

When an operational amplifier uses a feedback resistor and multiple input resistors as shown in Figure 2.3 and 2.3a, it becomes a summer. The network consists of input resistors $R_1, R_2, ----, R_n$, one for each voltage being summed and a single feedback resistor, R_f. The output voltage, e_o is equal to the negative of the algebraic sum of the modified input voltages $e_1, e_2, ----, e_n$:

$$e_o = - \left[\frac{R_f}{R_1} e_1 + \frac{R_f}{R_2} e_2 + ---- + \frac{R_f}{R_n} e_n \right] \qquad [2.2a]$$

or $$e_o = - R_f \sum_{i=1}^{n} \frac{e_i}{R_i} \qquad [2.2b]$$

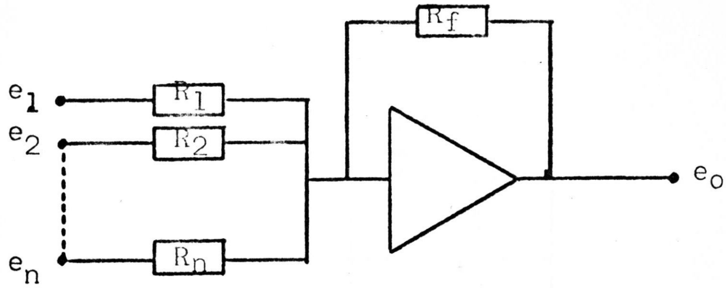
or $$e_o = - \sum_{i=1}^{n} g_i e_i \qquad [2.2c]$$

where $g_i = \frac{R_f}{R_i}$, with dimensions of pure number, is called the gain of the ith input; the summation being taken over all inputs.

The conventional symbol for a summing amplifier with gains $g_1, g_2, ----, g_n$ is shown in Figure 2.3a on page 12.

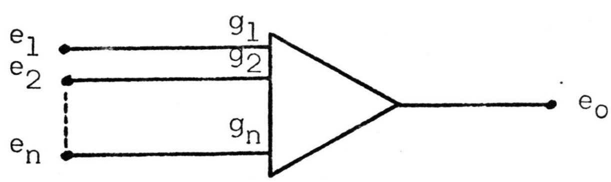
The Summing Integrator

When a capacitor is used as the feedback element for an operational amplifier, as shown in Figure 2.4 and 2.4a, it is called a summing integrator. The circuit consists again of the input resistors



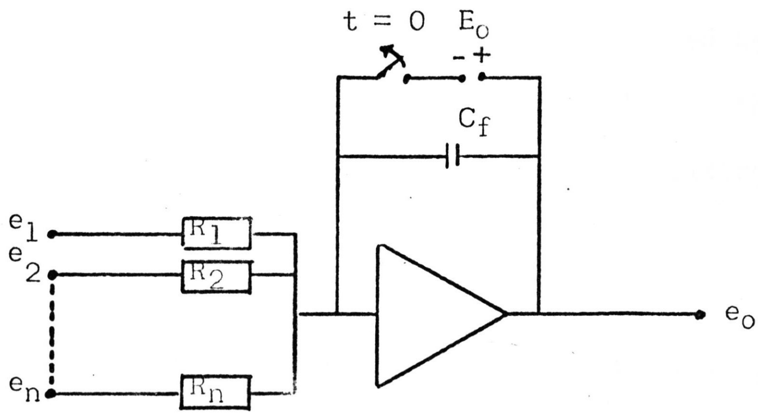
$$e_o = -\left(\frac{R_f}{R_1} e_1 + \frac{R_f}{R_2} e_2 + \text{----} + \frac{R_f}{R_n} e_n \right)$$

Figure 2.3 Summing Amplifier



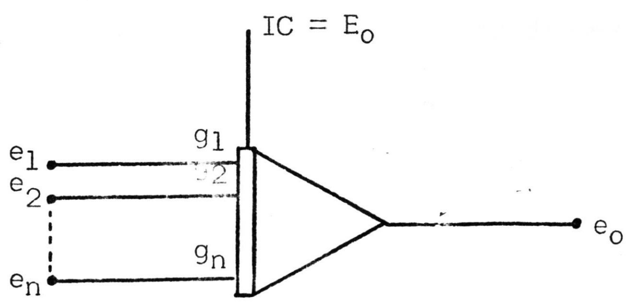
$$e_o = -(g_1 e_1 + g_2 e_2 + \text{----} + g_n e_n)$$

Figure 2.3a Summing Amplifier



$$e_o = -\frac{1}{C_f}\int_0^t \left(\frac{e1}{R_1} + \frac{e2}{R_2} + \text{----} + \frac{en}{R_n} \right) dt + E_o$$

Figure 2.4 Summing Integrator



$$e_o = -\int_0^t (g_1\, e_1 + g_2\, e_2 + \text{----} + g_n\, e_n)\, dt + E_o$$

Figure 2.4a Summing Integrator

R_1, R_2,----,R_n with the feedback resistor replaced by a capacitor. The output voltage e_o is equal to the negative of the integral with respect to time of the algebraic sum of the modified input voltages e_1, e_2,----,e_n:

$$e_o = -\frac{1}{C_f}\int_0^t \left(\frac{e_1}{R_1} + \frac{e_2}{R_2} + \text{----} + \frac{e_n}{R_n}\right) dt + E_o \qquad [2.3a]$$

$$\text{or} \quad e_o = -\int_0^t (g_1 e_1 + g_2 e_2 + \text{----} + g_n e_n)\, dt + E_o \qquad [2.3b]$$

where $g_i = \frac{1}{C_f R_i}$ with dimension of number divided by time are called gains of the integrator.

Specialized Devices

Diodes. The introduction of diodes into analog computer network permits the representation of many types of complex restrictions. Figure 2.5 shows the symbolic notations for diodes.

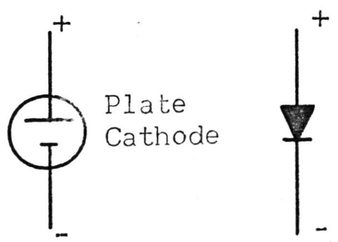


Figure 2.5 Symbol-Diodes

Diodes may be thought of as undirectional devices with current flow allowed in the forward (plate to cathode) direction and blocked in the opposite direction. A diode can be back-baised by placing a voltage source in series, to allow conduction at voltages other than zero.

The use of Diodes in a Switching Circuit. Diodes are frequently used in switching circuits. The switching circuit shown in Figure 2.6 has properties which can be described as follows:

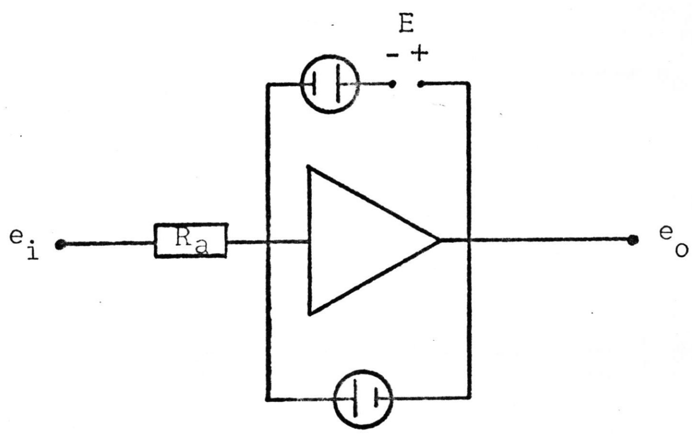
$$e_o = 0 \qquad \text{when} \qquad e_i > 0 \qquad [2.4a]$$

$$e_o = E \qquad \text{when} \qquad e_i < 0 \qquad [2.4b]$$

given $R_a \gg R_{diode, forward}$

When e_i is positive, the lower diode conducts; and since the forward resistance, $R_{diode, forward}$ is negligible compared to the input resistance, R_a, e_o is zero volts. When e_i is negative the upper diode conducts the bais voltage, E.

Positive Output Integrators. If a diode is connected across a summing integrator as shown in Figure 2.7 on page 17, the output of the network is restricted to positive values only. The diode D prevents the output from becoming negative. This is because when $\sum_{i=1}^{n} e_i$ is positive the capacitor, C_f is bypassed, and the forward resistance of the diode being negligible, the output of the network is zero. When $\sum_{i=1}^{n} e_i$ is

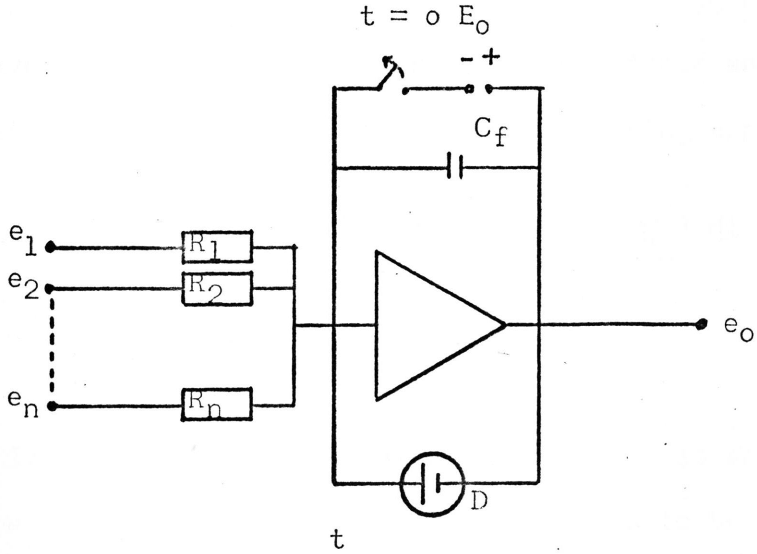


$R_a >> R_{diode,\ forward}$

e_o = 0 volt when $e_i > 0$

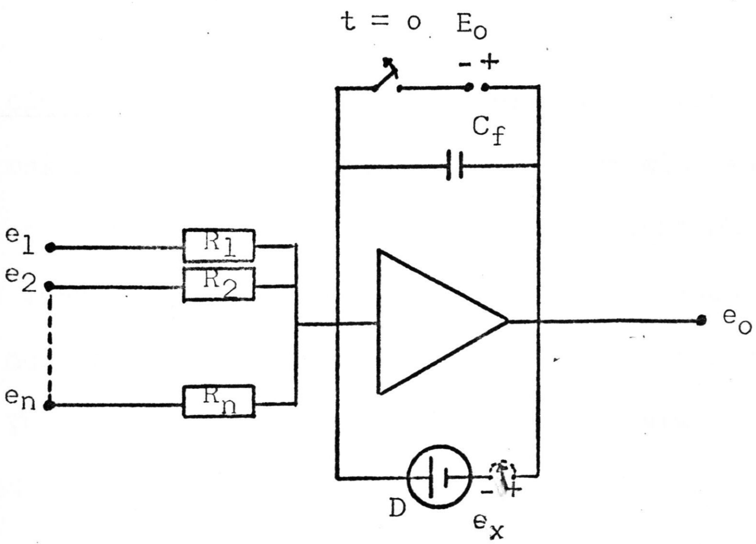
e_o = E volts when $e_i < 0$

Figure 2.6 Switching Circuit



$$e_o = -\frac{1}{C_f}\int_o^t \left(\frac{e_1}{R_1} + \frac{e_2}{R_2} + ---- + \frac{e_n}{R_n}\right) dt + E_o \geq 0$$

Figure 2.7 Positive Output Integrator



$$e_o = -\frac{1}{C_f}\int_o^t \left(\frac{e_1}{R_1} + \frac{e_2}{R_2} + ---- + \frac{e_n}{R_n}\right) dt + E_o \geq e_x$$

Figure 2.7a Baised Positive Output Integrator negative the diode offers a very high resistance and the feedback capacitor conducts, thus integrating and giving a positive output.

$$e_o = -\frac{1}{C_f}\int_0^t \left(\frac{e_1}{R_1} + \frac{e_2}{R_2} + ---- + \frac{e_n}{R_n}\right) dt + E \qquad [2.5]$$

$$e_o \geq 0$$

If the diode is back-baised in a manner as shown in Figure 2.7a on page 17, the output voltage, e_o is forced to be greater or equal to the variable bais voltage, e_x.

$$e_o = -\frac{1}{C_f}\int_0^t \left(\frac{e_1}{R_1} + \frac{e_2}{R_2} + ---- + \frac{e_n}{R_n}\right) dt + E_o \qquad [2.6]$$

$$e_o \geq e_x$$

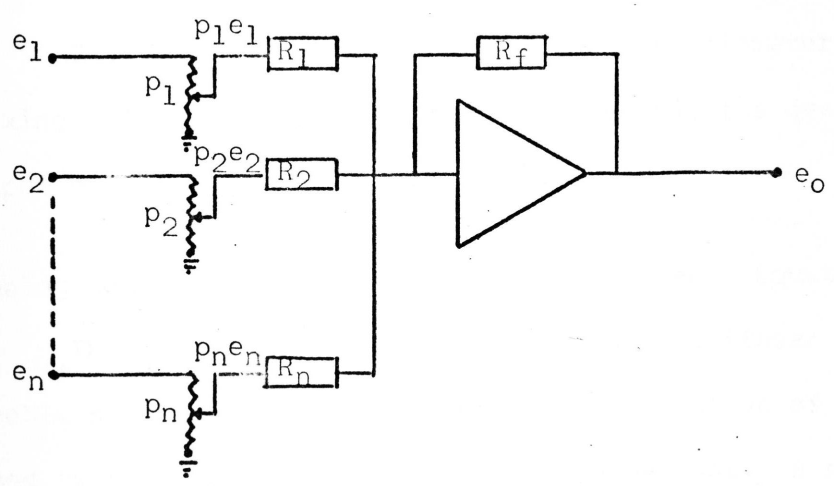
<u>Coefficient Potentiometers</u>. The coefficients appearing in most problem equations are not whole numbers. It is unwieldy and impractical to represent coefficients which are not convenient whole numbers by the use of input gains, g_i (the $\frac{R_f}{R_i}$ ratio) alone, because of the infinitely large number of resistance values that would be required.

The general circuit equation for the network shown in Figure 2.8 and 2.8a can be expressed as

$$e_o = -\left(p_1 e_1 \frac{R_f}{R_1*} + p_2 e_2 \frac{R_f}{R_2*} + ---- + p_n e_n \frac{R_f}{R_n*}\right) \qquad [2.7a]$$

$$= -\left(p_1 e_1 G_1 + p_2 e_2 G_2 + ---- + p_n e_n G_n\right) \qquad [2.7b]$$

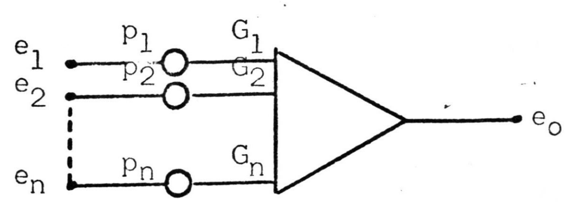
where $p_i \frac{R_f}{R_i*} = \frac{R_f}{R_i}$



$$e_o = -\left(p_1e_1 \frac{R_f}{R_1^*} + p_2e_2 \frac{R_f}{R_2^*} + ---- + p_ne_n\frac{R_f}{R_n^*} \right)$$

where $\frac{R_f}{R_i^*}$ is larger than the desired value.

Figure 2.8 Circuit Showing use of Potentiometer



$$e_o = -(p_1e_1G_1 + p_2e_2G_2 + ---- + p_ne_nG_n)$$

where G_i is larger than the desired value.

Figure 2.8a Circuit Showing use of Potentiometer

Thus by varying the value of the potentiometer setting p_i and fixing R_i* (such that R_f/R_i* is an integer), the desired value of the input gain can be achieved.

Analog Computer Approach to Solve Simultaneous Equations

The method used to solve linear and nonlinear programming problems on the analog computer is the variation of the general form used to solve simultaneous algebraic equations. A review of the analog computer approach to solve simultaneous equations would enable a better understanding of the steepest ascent technique.

A set of algebraic equations can be written in the form

$$\sum_{k=1}^{n} a_{jk} - b_j = 0 \quad (j = 1,2,----,n) \qquad [2.8]$$

The most obvious way to solve these sets of equations on an analog computer would be to use a set of n summers to perform the desired summation of the n equations. The variables $x_1,x_2,----,x_n$ would then be generated implicitly by connecting the outputs to each input as shown in Figures 2.9 and 2.9a. Each summer represents one equation in the form

$$x_i = \frac{b_i}{a_{ii}} - \frac{a_{i1}}{a_{ii}}x_1 \cdots - \frac{a_{i,i-1}}{a_{ii}} x_{i-1} - \frac{a_{i,i+1}}{a_{ii}} x_{i+1} - \cdots - \frac{a_{in}}{a_{ii}} x_n \qquad [2.9]$$

Additional sign changers are required for negative coefficients.

This method is quite restricted and not recommended because of stability considerations. That is, in some sets of simultaneous equations, solutions overload the amplifiers in attempting to reach a finite solution.

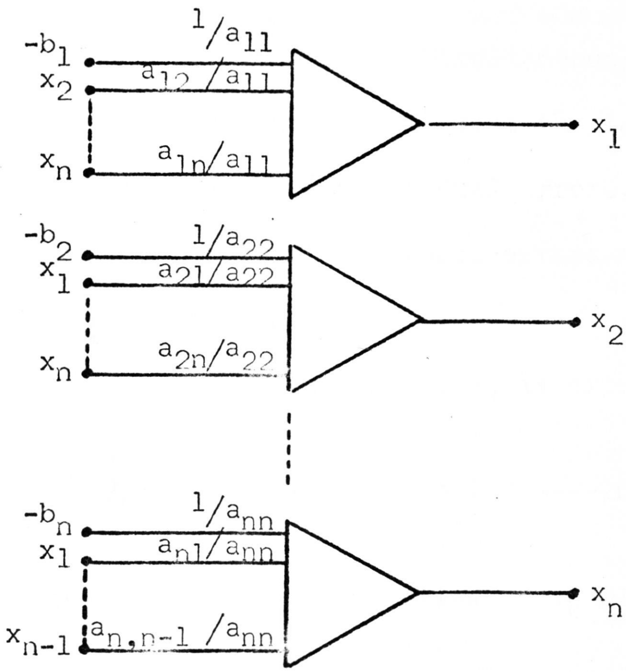


Figure 2.9 General Computer Program to Solve n Simultaneous Equations

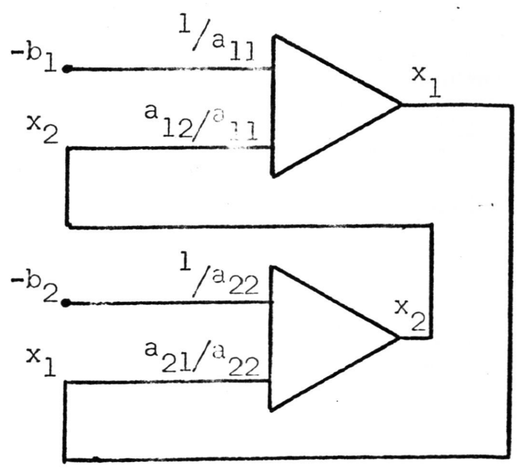


Figure 2.9a Complete Computer Program to Solve Simultaneous Equations
$(a_{11}x_1 + a_{12}x_2 = b_1 \; ; \; a_{21}x_1 + a_{22}x_2 = b_2)$

Perfect Integrator Method for Solving Simultaneous Equations.

The method of perfect integrators uses the technique of minimizing the sum of the squares of the residual errors. The use of perfect integrators ensures that all residual errors due to the computer set up would be driven to zero.

If for the jth equation the error, e_j is expressed as

$$e_j = \sum_{k=1}^{n} a_{jk} x_k - b_j \qquad (j = 1,2,----,n) \qquad [2.10]$$

It can be proved[1] that the derivative of x_k with respect to time is given by

$$\frac{dx_k}{dt} = - K \sum_{j=1}^{n} a_{jk} e_j \qquad (k = 1,2,----,n) \qquad [2.11]$$

$$\text{therefore} \int \frac{dx_k}{dt}\, dt \longrightarrow x_k \quad \text{as} \quad e_j \longrightarrow 0 \qquad (k = 1,2,----,n \quad j = 1,2,----,n) \qquad [2.12]$$

The resulting computer set up is shown in Figures 2.10 and 2.10a. With time the residual errors are driven to zero and the system automatically balances itself, thus outputing the variables x_k $(k = 1,2,----,n)$.

This method can also be verified[2] to be the method of steepest ascent.

---

[1] Jackson, A. S. (4) pp. 341-342 gives the mathematical proof for this expression.

[2] Fifter, S. (6) v.3 pp. 847-854 gives the mathematical proof for this verification.

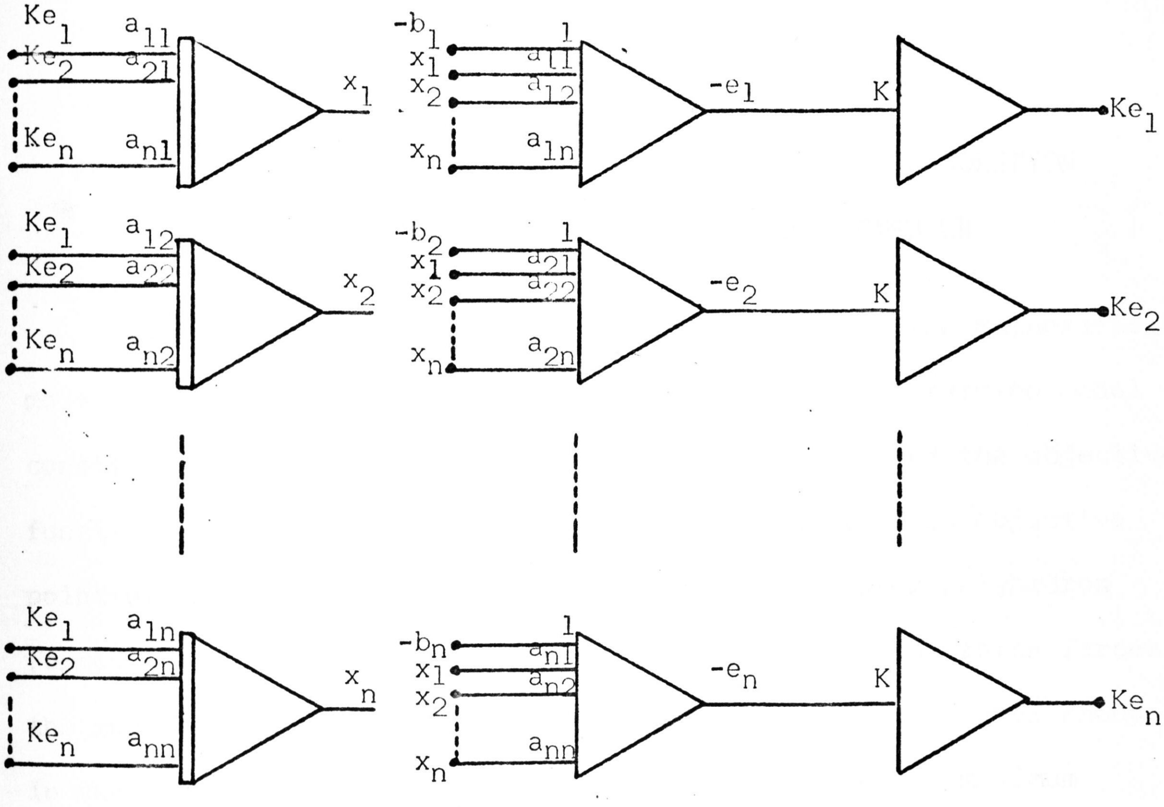


Figure 2.10 General Program for Solving n Simultaneous Equations By Integrator Method

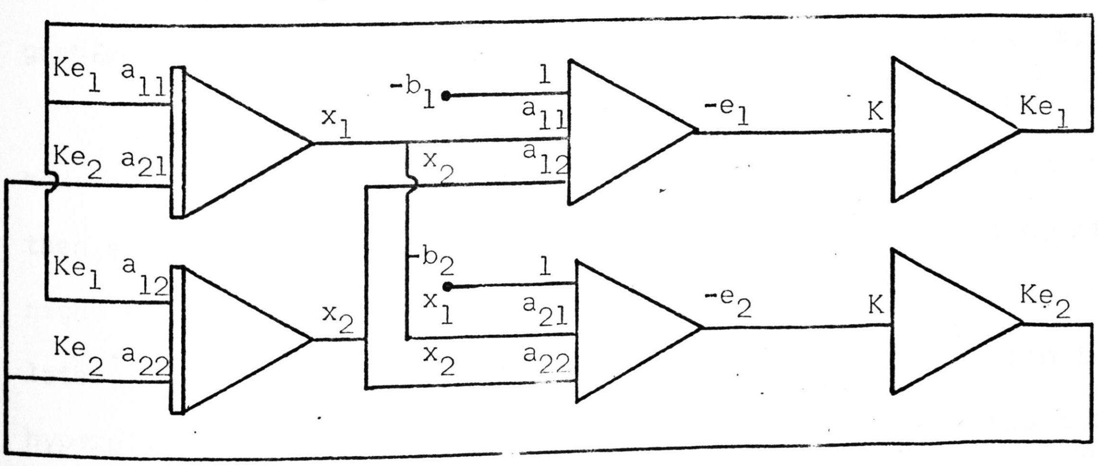


Figure 2.10a Complete Program of Integrator Method for Solving two Simultaneous Equations ($a_{11}x_1 + a_{12}x_2 = b_1$; $a_{21}x_1 + a_{22}x_2 = b_2$)

CHAPTER III

PHYSICAL SYSTEM OF THE LINEAR PROGRAMMING MODEL AND SOLUTION OF ITS MATHEMATICAL MODEL ON THE ANALOG COMPUTER

The analog computer is generally used to solve the mathematical model representing a physical system. The linear programming model consists of two types of equations, the constraints and the objective function. The analog can be considered to consist of an objective point of the function, Z, being generated in the convex polyhedron formed by the constraints of the problem. A maximizing force forces the movement of this objective point such that an incremental change in the displacement of the objective point will produce maximum change in the objective function. The objective function is continuous and single valued within the convex polyhedron. The maximum is thus accomplished by forcing the objective point to travel along the gradient of the objective function until the optimum value is reached.

Whenever the objective point is driven outside the convex polyhedron, at least one restriction is violated. The objective point is then subjected to a strong additional constraint force, acting at right angles to the hyperplane, associated with the restriction violated. The net result is that the objective point moves along the hyperplane in a zig-zag fashion to the solution vertex, where the objective function attains its maximum value.

This method is a variation of the method of steepest ascent in that the objective point travels along the gradient of the function and the optimum solution is directed by the objective function under the influence of additional constraint forces.

The concepts involved in this physical system can be best illustrated through a simple two-dimensional linear programming problem.

maximize $Z = x_1 + 2x_2$

subject to

$R_1 : x_2 \leq 2$

$R_2 : 5x_1 + 3x_2 \leq 15$

$R_3 : 3x_1 + 4x_2 \leq 12$

$R_4 : x_1 \geq 0$

$R_5 : x_2 \geq 0$

A graphical representation of this problem is shown in Figure 3.1.

The convex polyhedron OABCD (ALLOWED SPACE) is seen to be bounded by five hyperplanes[1], R_1, R_2, R_3 and two non-negative restraints, R_4 and R_5. If the objective point is initially started at the origin 0, the point travels along the path indicated in dotted lines, until it strikes the first barrier, AB. The driving force is still applied, the point thus moves along the barrier AB in a zig-zag

---

[1] In two dimensions a hyperplane is a line.

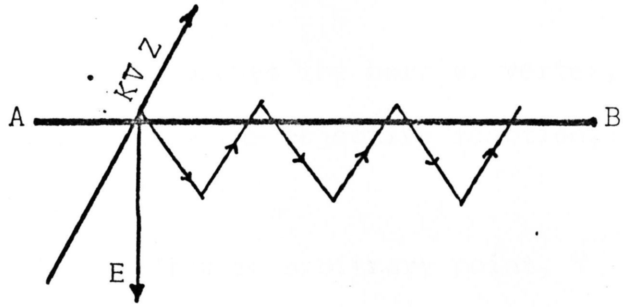


Figure 3.1a Motion of the Objecting Point When it Strikes Barrier (AB)

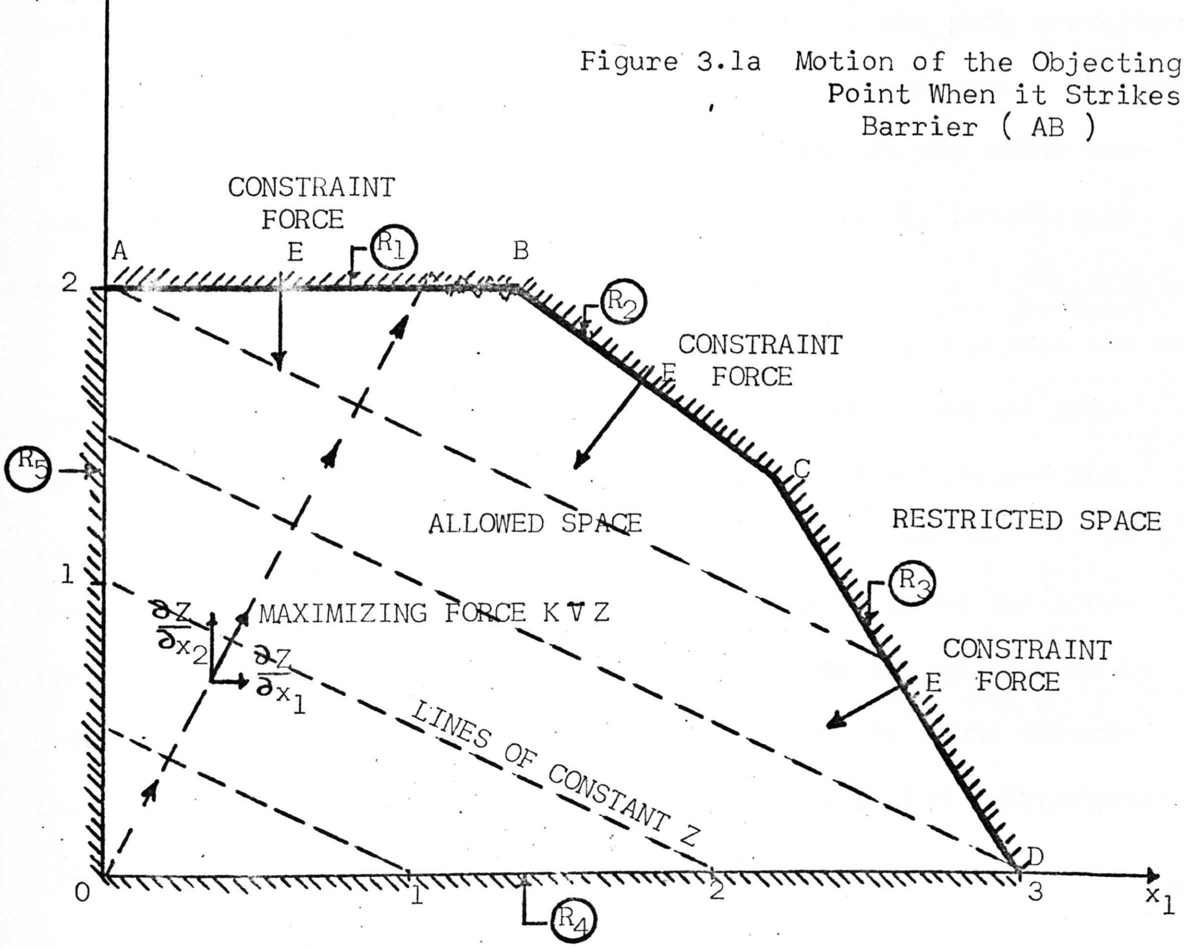


Figure 3.1 Dynamical Analogy of the Linear Programming Problem

fashion and shown in Figure 3.1a, until it reaches the barrier vertex, B. This point corresponds to the point where the objective function, Z can attain the maximum allowed value.

If the objective point is initiated from an arbitrary point, Y inside the polyhedron such as shown in Figure 3.2, the path travelled by the point along the barriers is indicated in dotted lines. Further, it is possible to initialize the objective point at any point outside the polyhedron, because the constraint forces, E, immediately force the point back into the allowed space.

Having obtained an optimum solution sensitivity analysis can be applied. Parametric programming deals with the behaviour of this dynamical system when the slope of the objective function and the slope or position of the constraint hyperplanes are varied. To vary these parameters, it is necessary to change the value of the coefficients. An analog computer, where coefficients are determined by potentiometer settings, and in which the solution is found dynamically from any point in space, should be ideally suited for parametric programming and sensitivity analysis.

Mathematical Model

To develop the mathematical relations involved in the analog computer solution of linear programming problems, the constraints and objective function are defined in terms of variables and constants. The interaction of the forces is then expressed in terms of these variables and constants.

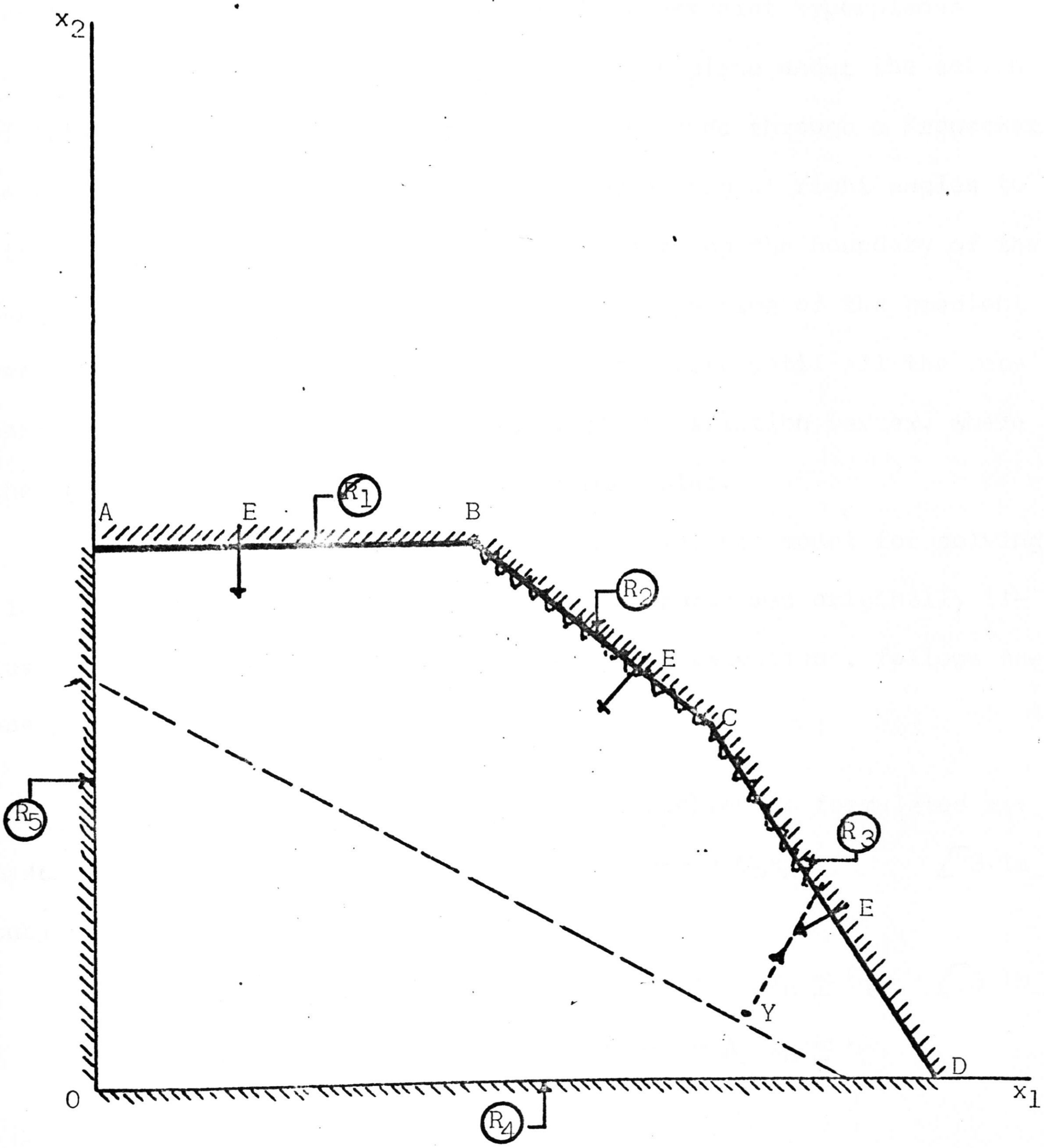


Figure 3.2 Path Followed by the Objective Point When Initiated at Y

The maximizing force is expressed as a gradient of the objective function vector and directed towards the constraint hyperplanes.

When the solution point strikes any hyperplane under the action of this maximizing force, the hyperplane responds through a Kronecker delta to supply a strong constraint vector acting at right angles to itself. The result is that the point moves along the boundary of the convex polyhedron in the direction of the projection of the gradient vector on that hyperplane. This motion continues until all the vectors acting are balanced. This occurs at the solution vertex, where the objective function attains its optimum value.

The concepts of development of a mathematical model for solving linear programming problems on an analog computer was originally illustrated by Pyne (5). This discussion, with extensions, follows the one presented by Pyne.

A general form of linear programming problem is formulated as:

maximize $$Z = C_1x_1 + C_2x_2 + \cdots + C_kx_k + \cdots + C_nx_n \qquad [3.1a]$$

subject to a set of m constraints

$$\begin{array}{l} A_{11}x_1 + A_{12}x_2 + \cdots + A_{1k}x_k + \cdots + A_{1n}x_n \leq b_1 \qquad [3.1b] \\ A_{21}x_1 + A_{22}x_2 + \cdots + A_{2k}x_k + \cdots + A_{2n}x_n \leq b_2 \\ \vdots \\ A_{j1}x_1 + A_{j2}x_2 + \cdots + A_{jk}x_k + \cdots + A_{jn}x_n \leq b_j \\ \vdots \\ A_{m1}x_1 + A_{m2}x_2 + \cdots + A_{mk}x_k + \cdots + A_{mn}x_n \leq b_m \end{array}$$

plus a set of non-negative constraints

$$x_1 \geq 0, \quad x_2 \geq 0, \quad ----, \quad x_k \geq 0, \quad ----, \quad x_n \geq 0$$

This set of equations can be expressed as:

$$\text{maximize} \quad Z = \frac{x_1}{c_1} + \frac{x_2}{c_2} + ---- + \frac{x_k}{c_k} + ---- + \frac{x_n}{c_n} \qquad [3.2a]$$

subject to [3.2b]

$$\frac{x_1}{b_1/A_{11}} + \frac{x_2}{b_1/A_{12}} + ---- + \frac{x_k}{b_1/A_{1k}} + ---- + \frac{x_n}{b_1/A_{1n}} \leq 1$$

$$\frac{x_1}{b_2/A_{21}} + \frac{x_2}{b_2/A_{22}} + ---- + \frac{x_k}{b_2/A_{2k}} + ---- + \frac{x_n}{b_2/A_{2n}} \leq 1$$

$$\vdots \qquad\qquad\qquad\qquad \vdots$$

$$\frac{x_1}{b_j/A_{j1}} + \frac{x_2}{b_j/A_{j2}} + ---- + \frac{x_k}{b_j/A_{jk}} + ---- + \frac{x_n}{b_j/A_{jn}} \leq 1$$

$$\vdots \qquad\qquad\qquad\qquad \vdots$$

$$\frac{x_1}{b_m/A_{m1}} + \frac{x_2}{b_m/A_{m2}} + ---- + \frac{x_k}{b_m/A_{mk}} + ---- + \frac{x_n}{b_m/A_{mn}} \leq 1$$

This set of equations can be further simplified as:

maximize $$Z = \sum_{k=1}^{n} \frac{x_k}{c_k} \qquad (x_k \geq 0) \qquad [3.3a]$$

subject to $$\sum_{k=1}^{n} \frac{x_k}{a_{1k}} \leq 1 \qquad [3.3b]$$

$$\sum_{k=1}^{n} \frac{x_k}{a_{2k}} \leq 1$$

$$\vdots$$

$$\sum_{k=1}^{n} \frac{x_k}{a_{jk}} \leq 1$$

$$\vdots$$

$$\sum_{k=1}^{n} \frac{x_k}{a_{mk}} \leq 1$$

where $$c_k = \frac{1}{C_k} \qquad (k = 1,2,----,n)$$

$$a_{jk} = \frac{b_j}{A_{jk}} \qquad (j = 1,2,----,m \quad k = 1,2,----,n)$$

Denominator coefficients are used as they then directly represent the intercept of a hyperplane with one of the n axes.

The variables, $x_1, x_2, ----, x_k, ----, x_n$ may be considered as co-ordinates of a solution point in the n - dimensional Euclidean space. The coefficient, a_{jk}, then represents the intercept of the jth hyperplane with the kth co-ordinate axis and the equation [3.3b] represents a convex polyhedron (hypervolume) in the n - space bounded by m hyperplanes.

The objective function is continuous and single valued within the convex polyhedron. A gradient vector is defined as ∇Z. If $i_1, i_2, ----, i_k, ----, i_n$ are unit vectors in the direction of the n co-ordinate axes, then

$$\nabla Z = \sum_{k=1}^{n} \frac{\partial Z}{\partial x_k} i_k = \sum_{k=1}^{n} \left(\frac{1}{c_k}\right) i_k \qquad [3.4]$$

The vector ∇Z is a constant acting normally to the hyperplanes of equal values of the objective function, and pointed in the direction of the steepest ascent, regardless of its location.

In order to represent the constraint forces associated with each of the m restrictions, we define m vectors, E_j $(j = 1,2,----,m)$ acting normally to the hyperplanes and directed towards the convex polyhedron. Thus each hyperplane acts as a barrier to the maximizing force vector.

To compute the components of E_j (5), it should be first noted that the linearly independent vectors, given by equation [3.5], are parallel to the jth hyperplane; i.e., they lie on the hyperplane

$$\sum_{k=1}^{n} \frac{x_k}{a_{jk}} = 1$$

$$\begin{aligned} u_1 &= (a_{j1}i_1 - a_{j2}i_2) \\ u_2 &= (a_{j2}i_2 - a_{j3}i_3) \\ &\vdots \\ u_j &= (a_{jn-1}i_{n-1} - a_{jn}i_n) \end{aligned} \qquad [3.5]$$

The vector E_j has the property that the inner product with each vector of the set, equation [3.5], vanishes and also that it is directed away from the jth hyperplane toward the allowed space. Therefore it can be expressed in terms of its n components as

$$E_j = -\left(\frac{i_1}{a_{j1}} + \frac{i_2}{a_{j2}} + ---- + \frac{i_k}{a_{jk}} + ---- + \frac{i_n}{a_{jn}}\right) \qquad [3.6]$$

Each of the m constraints can be defined in terms of Kronecker delta notation δ_1, δ_2, ----, δ_m

$$\delta_j = 0 \qquad \text{when} \quad \sum_{k=1}^{n} \frac{x_k}{a_{jk}} \leq 1 \qquad [3.7]$$

$$\delta_j = 1 \qquad \text{when} \quad \sum_{k=1}^{n} \frac{x_k}{a_{jk}} > 1$$

The vector of representing the forces acting on the objective point can then be expressed as follows

$$f = K\nabla Z + \sum_{j=1}^{m} \delta_j E_j \qquad [3.8]$$

where K is a constant

Expressing f in terms of components we have

$$f = f_1 i_1 + f_2 i_2 + ---- + f_k i_k + ---- + f_n i_n \qquad [3.9]$$

The kth component of f can then be expressed as

$$f_k = \frac{K}{c_k} - \sum_{j=1}^{m} \frac{\delta_j}{a_{jk}} \qquad [3.10]$$

The magnitude of the vector f or its kth component depend on the co-ordinates of the objective point because of the presence of the Kronecker deltas in equation [3.8] and [3.10]. As long as the objective point is inside the convex polyhedron it is subjected only to the maximizing vector $K\nabla Z$. Whenever it travels outside the convex polyhedron its motion is governed by both, the maximizing vector, $K\nabla Z$ and the constraint vector, E_j.

The position of the objective point can be denoted by the vector

$$r = x_1 i_1 + x_2 i_2 + ---- + x_k i_k + ---- + x_n i_n \qquad [3.11]$$

and its velocity in the n-dimensional space is then the vector

$$v = \frac{dr}{dt} \qquad [3.12]$$

where t is the time

The velocity of the objective point can also be expressed as

$$\frac{dZ}{dt} = \gamma f \qquad [3.13]$$

where γ is a constant scalar

Equation [3.13] describes the motion of the objective function as the solution is approached. The solution is the value which r finally attains. From equation [3.10] the kth component of equation [3.13] can be written as

$$\frac{dx_k}{dt} = \gamma \left(\frac{K}{c_k} - \sum_{j=1}^{m} \frac{\delta_j}{a_{jk}} \right) \qquad [3.14]$$

As stated earlier, the objective point travels through the convex polyhedron with a velocity $\gamma K \nabla Z$ until it strikes a constraint hyperplane such as the jth hyperplane. The motion of the vectors expressed as the two parts of equation [3.8], the gradient and the constraint vector acting normally to the jth hyperplane. If the constraint vector, E_j is always greater than the normal component of $K \nabla Z$, the point is driven back into the polyhedron. The vector, E_j then vanishes, and the gradient forces the point again out of the polyhedron. In this way the objective point travels along the boundary of the convex polyhedron in a zig-zag fashion until the optimal vertex is reached.

Solution of Mathematical Model on Analog Computer

Solution of the mathematical model on the analog computer is obtained by simultaneously solving the set of differential equations and the attendant algebraic expressions of the model. The variables $x_1, x_2, ----, x_k, ----, x_n$ are represented on the analog computer by corresponding voltages $e_1, e_2, ----, e_k, ----e_n$. These voltages are proportional to the corresponding original variables on a convenient scale. The given set of relations between the original variables are expressed by an analogous set of relations between the voltages.

The analog computer network required to solve the linear programming model could be considered to be a modification of the 'Integrator Method' approach used to solve simultaneous equations on the analog computer as explained in Chapter II.

In case of the linear programming model the number of summing amplifiers, (m) may be equal to, greater or less than the number of integrators, (n) which are limited to positive output only. Moreover, the solution point is directed by the objective function, and associated with each of the m constraint equations is a switch, to supply a constraint voltage when the equation is violated. However, the networks of both the integrator method and analog computer approach for solving linear programming problems use the method of error reduction by negative feedback and apply the method of steepest ascent to attain the final solution.

To solve the mathematical equations of the linear programming problem on the analog computer, it is best to divide the analog computer circuitry into the following four types of network systems

1. A set of n 'positive output' summing integrators[2] to solve the n differential equations and output the n variables, $x_1, x_2, ----, x_k, ----x_n$.
2. A set of m summing amplifiers[3] to equate the m restrictions,
   $$\sum_{k=1}^{n} x_k/a_{jk} \leq 1 \quad (j = 1,2,----,n).$$
3. A set of m switches[4], one for each of the m restrictions, to apply the constraint force, E_j $(j = 1,2,----,m)$ when the objective point reaches the constraint hyperplane.
4. A set of invertors[5] as required by negative coefficients.

These network systems are interconnected in a manner as shown in Figure 3.3.

---

[2]Figures 2.7 and 2.7a on page 17 describe this circuit and its properties.
[3]Figures 2.3 and 2.3a on page 12 describe this circuit and its properties.
[4]Figure 2.6 on page 16 describes this circuit and its properties.
[5]Figure 2.2 on page 10 describes this circuit and its properties.

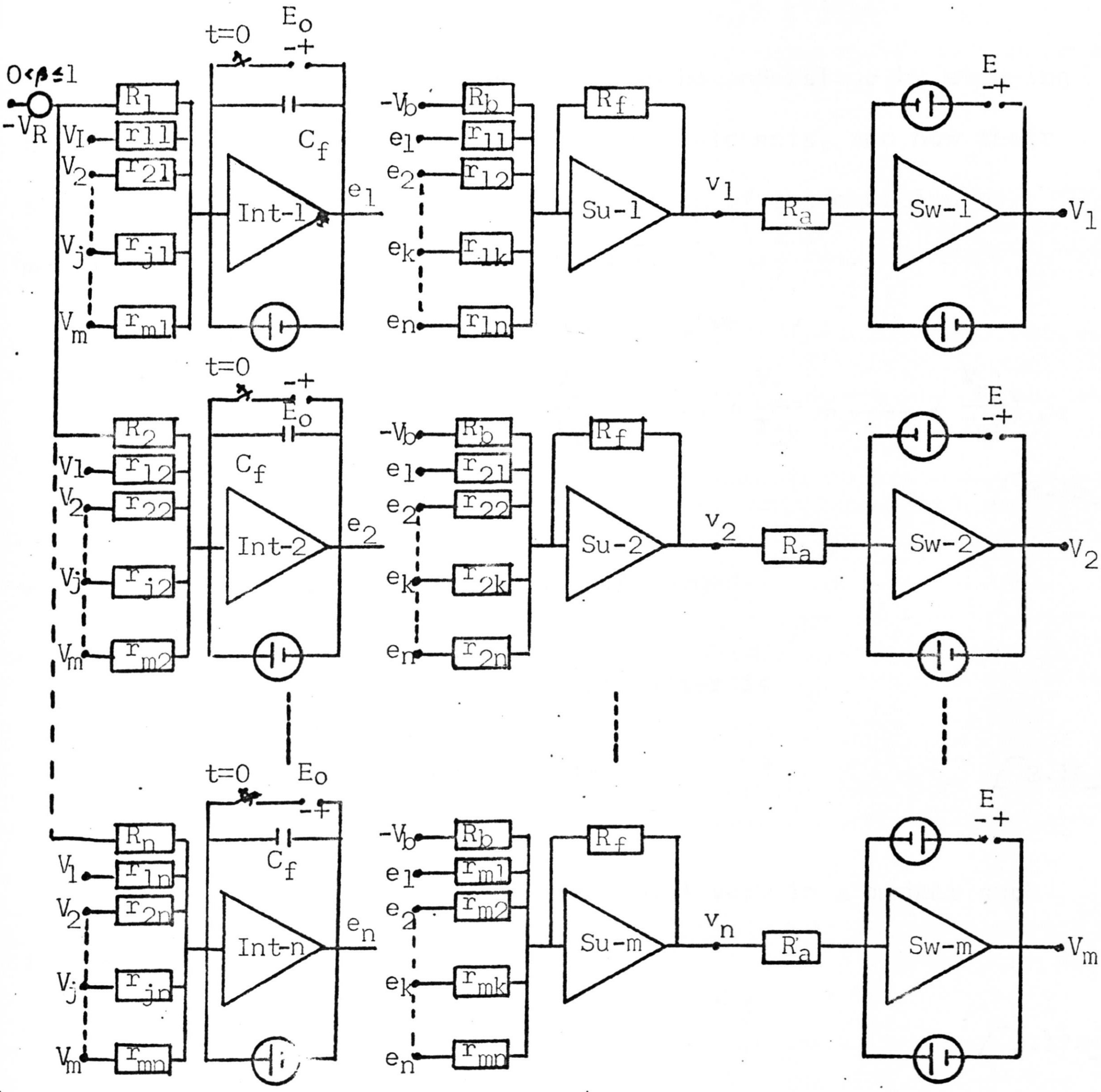


Figure 3.3 General Analog Computer Program in Terms of Input Resistances

The logic of the circuitry can best be understood by studying the output voltages of the various network elements, and how their simultaneous operation leads to the solution of the equation of motion as described by equation [3.14].

The output of the kth integrator is given by

$$e_k = -\frac{1}{C_f}\int_0^t \left(-\frac{\beta V_R}{R_k} + \frac{V_1}{r_{1k}} + \frac{V_2}{r_{2k}} + \text{----} + \frac{V_j}{r_{jk}} + \text{----} + \frac{V_m}{r_{mk}}\right) dt \quad \text{Volts} \qquad [3.15]$$

when the capacitor, C_f is initially discharged.

The output of the jth summing amplifier is

$$v_j = -R_f \left(-\frac{V_b}{R_b} + \sum_{k=1}^{n} \frac{e_k}{r_{jk}}\right) \text{ Volts} \qquad [3.16]$$

The output, V_j of the jth switch will vary in a manner such that

$$V_j = 0 \qquad \text{when } v_j > 0 \qquad [3.17a]$$

$$V_j = E \text{ volts} \qquad \text{when } v_j < 0 \qquad [3.17b]$$

Using Kronecker delta notation, we have

$$\delta_j = 0 \quad \text{when} \quad \sum_{k=1}^{n} \frac{e_k}{r_{jk}} \leq \frac{V_b}{R_b} \qquad [3.18a]$$

$$\delta_j = 1 \quad \text{when} \quad \sum_{k=1}^{n} \frac{e_k}{r_{jk}} > \frac{V_b}{R_b} \qquad [3.18b]$$

The output of the jth switch can then be described as

$$V_j = \delta_j E \text{ volts} \qquad [3.19]$$

If equation [3.15] is differentiated with respect to time, we have

$$\frac{de_k}{dt} = \frac{1}{C_f}\left(\frac{\beta V_R}{R_k} - \frac{V_1}{r_{ik}} - \frac{V_2}{r_{2k}}, - \ldots, - \frac{V_j}{r_{jk}}, - \ldots, - \frac{V_m}{r_{mk}}\right)$$

Volts per second [3.20]

Introducing the symbolism described by equations [3.17], [3.18] and [2.19], the equation describing the computation performed by the circuitry of Figure 3.3 can be written as

$$\frac{de_k}{dt} = \frac{1}{C_f}\left(\frac{\beta V_R}{R_k} - E \sum_{j=1}^{m} \frac{\delta_j}{r_{jk}}\right) \text{ Volts per second} \qquad [3.21]$$

whereas equation [3.14] which describes the desired motion of the objective point of function, Z is

$$\frac{dx_k}{dt} = \gamma\left(\frac{K}{c_k} - \sum_{j=1}^{m} \frac{\delta_j}{a_{jk}}\right) \qquad [3.14 \text{ repeated}]$$

Equations [3.14] and [3.21] are identical in form. The circuitry of Figure 3.3 can thus perform the computations of equation [3.14] by proper selection of the computing resistors and supply voltage values.

When the circuit is initially started with the capacitors, C_f discharged, the e_k ($k = 1,2,----,n$) voltages are zero and hence the outputs of the m summing amplifiers, v_j ($j = 1,2,----,m$) are positive ($v_j = \frac{V_b R_f}{R_b}$) and the V_j ($j = 1,2,----,m$) voltages are zero.

The result is that the e_k voltages start increasing with time in the positive direction until the v_j voltages are equal to or less than zero. As soon as an e_k voltage increases to a value such as to cause any of the v_j voltages to become negative, a voltage E is applied by the corresponding switch; thereby reducing the value of the e_k voltages. In this manner the circuit voltages balance with time, giving positive output voltages, e_k corresponding to each integrator, while satisfying the conditions

$$\sum_{k=1}^{n} \frac{e_k}{r_{jk}} \leq \frac{V_b}{R_b} \quad (j = 1,2,----,n)$$

for each of the m amplifiers.

Machine Variables and Scale Factors

The analog computer will establish mathematical relations not between the original variables $x_1, x_2, ----, x_k, ----, x_n$, but between the voltages $e_1, e_2, ----, e_k, ----, e_n$, simulating these variables. It is therefore necessary to establish a set of relations describing the transformation of the given problem variables and parameters to machine voltages and settings, respectively.

Defining a set of scale factors S_k such that

$$e_k = S_k x_k \qquad (k = 1,2,----,n) \qquad [3.22]$$

Substituting for x_k in equations [3.7] and [3.14], we have

$$\delta_j = 0 \quad \text{when} \quad \sum_{k=1}^{n} \frac{e_k}{S_k a_{jk}} \leq 1 \qquad [3.23a]$$

$$\delta_j = 1 \quad \text{when} \quad \sum_{k=1}^{n} \frac{e_k}{S_k a_{jk}} > 1 \qquad [3.23b]$$

and $$\frac{1}{S_k}\left(\frac{de_k}{dt}\right) = \gamma\left(\frac{K}{c_k} - \sum_{j=1}^{m} \frac{\delta_j}{a_{jk}}\right) \qquad [3.24]$$

multiplying both sides of equation [3.23a] and [3.23b] by

$\frac{V_b}{R_b}$ we have

$$\delta_j = 0 \quad \text{when} \quad \sum_{k=1}^{n} \frac{e_k V_b}{S_k a_{jt} R_b} \leq \frac{V_b}{R_b} \qquad [3.25a]$$

$$\delta_j = 1 \quad \text{when} \quad \sum_{k=1}^{n} \frac{e_k V_b}{S_k a_{jk} R_b} > \frac{V_b}{R_b} \qquad [3.25b]$$

where as the circuit equations as given by equation [3.18] is

$$\delta_j = 0 \quad \text{when} \quad \sum_{k=1}^{n} \frac{e_k}{r_{jk}} \leq \frac{V_b}{R_b} \qquad [3.18a \text{ repeated}]$$

$$\delta_j = 1 \quad \text{when} \quad \sum_{k=1}^{n} \frac{e_k}{r_{jk}} > \frac{V_b}{R_b} \qquad [3.18b \text{ repeated}]$$

Comparing equations [3.18] and [3.25] we have the relation describing the transformation of the matrix variables, a_{jk} to input resistances, r_{jk}.

$$r_{jk} = a_{jk} S_k \frac{R_b}{V_R} \qquad [3.26]$$

substituting for a_{jk} from equation [3.26] into equation [3.24]

$$\frac{de_k}{dt} = \frac{\gamma S_k^2 R_b}{EV_b}\left(\frac{KEV_b}{R_b S_k c_k} - \sum_{j=1}^{m} \frac{\delta_j}{r_{jk}}\right) \qquad [3.27]$$

The circuit as given by equation [3.21] is

$$\frac{de_k}{dt} = \frac{1}{C_f}\left(\frac{\beta V_R}{R_k} - E\sum_{j=1}^{m} \frac{\delta_j}{r_{jk}}\right) \qquad [3.21 \text{ repeated}]$$

for the two equations to be identical we must have

$$R = US_k c_k \qquad [3.28]$$

where $U = \frac{\beta V_R R_b}{KEV_b}$

$$\text{and } \gamma = \frac{EV_b}{C_f R_b S_k^2} \qquad [3.29]$$

The analogy is complete through equations [3.26], [3.28] and [3.29].

The input resistance, R_a, of the switches is not critical, but should be much greater than the forward resistance of the diodes. A convenient value is 1 megohm or larger.

Although the value of the constraint voltage, E is not critical, it should be equal to the computer reference voltage, V_R, or greater, to assure that the constraint voltage will always be greater than the component of the driving force normal to any hyperplane. This avoids any possible break-through of the constraints by the objective point.

Equation [3.16] shows that if the e_k ($k = 1,2,----,n$) voltages representing variables are all zero, the output of each summing amplifier will start at E_{max}, if R_f and R_b are chosen in the ratio

$$\frac{R_f}{R_b} = \frac{E_{max}}{V_b} \qquad [3.30]$$

For parametric programming it is preferable to have V_b approximately equal to half the value of V_R, so that the constraints values, b_j ($j = 1,2,----,m$) can be varied over a wide range in either directions.

A convenient method exists for transforming the variables of the given problem to machine variables.

The original equations [3.1a] and [3.1b] are normalized and expressed in the form given by equations [3.3a] and [3.3b]. The procedure is illustrated by Tables 3.1 and 3.2.

For each column of Table 3.2, the largest coefficient, a_{jk} is selected and the corresponding scale factors S_k ($k = 1,2,----,n$) are calculated using the relation

$$S_k = \frac{V_b R_f}{R_b(\max a_{jk})} \qquad [3.31]$$

	x_1	x_2	......	x_n	
R_1	A_{11}	A_{12}	.....	A_{1n}	b_1
R_2	A_{21}	A_{22}	.....	A_{2n}	b_2
⋮	⋮	⋮		⋮	⋮
R_m	A_{m1}	A_{m2}	.....	A_{mn}	b_m
Z	C_1	C_2		C_n	

TABLE 3.1 Transformation Table 1

	x_1	x_2	....	x_n
R_1	$a_{11}=b_1/A_{11}$	$a_{12}=b_1/A_{12}$	.....	$a_{1n}=b_1/A_{1n}$
R_2	$a_{21}=b_2/A_{21}$	$a_{22}=b_2/A_{22}$	.....	$a_{2n}=b_2/A_{2n}$
⋮	⋮	⋮		⋮
R_m	$a_{m1}=b_m/A_{m1}$	$a_{m2}=b_m/A_{m2}$	.....	$a_{mn}=b_m/A_{mn}$
Z	$c_1=1/c_1$	$c_2=1/c_2$	....	$c_n=1/c_n$

TABLE 3.2 Transformation Table 2

$e_1=S_1x_1$	$e_2=S_2x_2$	.....	$e_n=S_nx_n$
S_1a_{11}	S_2a_{12}	......	S_na_{1n}
S_1a_{12}	S_2a_{22}	......	S_na_{2n}
⋮	⋮		⋮
S_1a_{m1}	S_2a_{m2}	......	S_na_{mn}
S_1c_1	S_2c_2	....	S_nc_n

TABLE 3.3 Transformation Table 3

	e_1	$e_2 \cdots\cdots$	e_n
V_1	$r_{11}=S_1 a_{11} R_b/V_b$	$r_{12}=S_2 a_{12} R_b/V_b \cdot\cdot$	$r_{1n}=S_n a_{1n} R_b/V_b$
V_2	$r_{21}=S_1 a_{21} R_b/V_b$	$r_{22}=S_2 a_{22} R_b/V_b \cdot\cdot$	$r_{2n}=S_n a_{2n} R_b/V_b$
V_m	$r_{m1}=S_1 a_{m1} R_b/V_b$	$r_{m2}=S_2 a_{m2} R_b/V_b \cdot\cdot$	$r_{mn}=S_n a_{mn} R_b/V_b$
Z^*	$R_1=US_1c_1$	$R_2=US_2c_2 \cdots\cdots$	$R_n=US_nc_n$

TABLE 3.4 Transformation Table 4

1. Input resistances r_{jk} represent the matrix coefficients A_{jk} $(j=1,2,----,m \quad k=1,2,----,m)$.
2. The resistances R_k represent the objective function coefficients C_k $(k=1,2,----,n)$.
3. The input voltages minus V_b represent the constraints b_j $(j=1,2,----,m)$.

Each column of Table 3.2 is then multiplied by its corresponding scale factor S_k, to obtain the array as shown in Table 3.3. The array shown in Table 3.4 represents the computer input resistances r_{jk} ($j = 1,2,----,m$ $k = 1,2,----,n$) and R_k ($k = 1,2,----,n$); and is obtained by multiplying each a - element of the array (Table 3.3) by $\frac{R_b}{V_b}$ and each of the C - element of the objective function by a constant, U, so that the resulting resistances, R_k, will be convenient in size.

The values given in the array of Table 3.4 are then the fixed input resistive values for the program shown in Figure 3.3 on page 38.

Extension of the Basic Program to Conduct Parametric Programming

For conducting parametric programming it is necessary to be able to vary the slope of the objective function (parameters C_k), the slope of the constraint hyperplanes (parameters A_{jk}) and the position of the hyperplanes (parameters b_j). This can be accomplished by representing these parameters by potentiometer settings rather than fixed input resistors, throughout the basic model (Figure 3.3 on page 38).

If feedback impedances of one megohm and one microfarad be used for each of the m 'summing amplifiers' and n 'positive output integrators' respectively, the various input gains (g_{jk} and g_k) would be then equal to the reciprocal of the value of the input resistors (r_{jk} and R_k)

$$g_k = \frac{1}{R_k} \qquad (k=1,2,----,n) \qquad [3.32a]$$

$$g_{jk} = \frac{1}{r_{jk}} \qquad (j=1,2,----,m \quad k=1,2,----,n) \qquad [3.32b]$$

Table 3.5 on page 49 gives the array of the various input gains required to represent the given problem, and Figure 3.4 on page 50 illustrates the circuitry.

Each of the input gains, g_k and g_{jk} are then obtained by choosing a suitable higher fixed input gain G_k and G_{jk} and using it in conjunction with a potentiometer with a setting p_k and p_{jk}, respectively given by the relations.

$$g_k = p_k G_k \qquad (k=1,2,----,n) \qquad [3.33a]$$

$$g_{jk} = p_{jk} G_{jk} \qquad (j=1,2,----,m \quad k=1,2,----,m) \qquad [3.33b]$$

It is preferable to choose a value of G_k and G_{jk} such that p_k p_{jk} will have a value near 0.5. This will enable the various parameters to be varied through a wide range of values.

Table 3.6 gives the array of the various fixed input gains $[G_k]$ and $[G_{jk}]$ used in conjunction with potentiometer setting (p_k) and (p_{jk}), respectively. The supply voltage V_b, representing the constraint value b_j (j=1,2,----,m) are available via potentiometers in almost all analog computers.

Moreover, if each of the n positive output integrators are back-baised[6] by a variable voltage supply V_{Fk} (k=1,2,----,m), equal to

[6]Figure 2.7a on page 17 describes this circuit and its properties.

	e_1	e_2	e_n
V_1	$g_{11}=1/r_{11}$	$g_{12}=1/r_{12}$ ...	$g_{1n}=1/r_{1k}$
V_2	$g_{21}=1/r_{21}$	$g_{22}=1/r_{22}$ ...	$g_{2n}=1/r_{2k}$
⋮	⋮		⋮
V_m	$g_{m1}=1/r_{m1}$	$g_{m2}=1/r_{m2}$ ...	$g_{mn}=1/r_{mn}$
Z^*	$g_1=1/R_1$	$g_2=1/R_2$	$g_n=1/R_n$

TABLE 3.5 Transformation Table 5

	e_1	e_2	e_n
V_1	$g_{11}=(p_{11}),G_{11}$	$g_{12}=(p_{12}),G_{12}$ ..	$g_{1n}=(p_{1n}),G_{1n}$
V_2	$g_{21}=(p_{21}),G_{21}$	$g_{22}=(p_{22}),G_{22}$ ..	$g_{2n}=(p_{2n}),G_{2n}$
⋮	⋮		⋮
V_m	$g_{m1}=(p_{m1}),G_{m1}$	$g_{m2}=(p_{m2}),G_{m2}$ ..	$g_{mn}=(p_{mn}),G_{mn}$
Z^*	$g_1=(p_1),G_1$	$g_2=(p_2),G_2$ . . .	$g_n=(p_n),G_n$

TABLE 3.6 Transformation Table 6

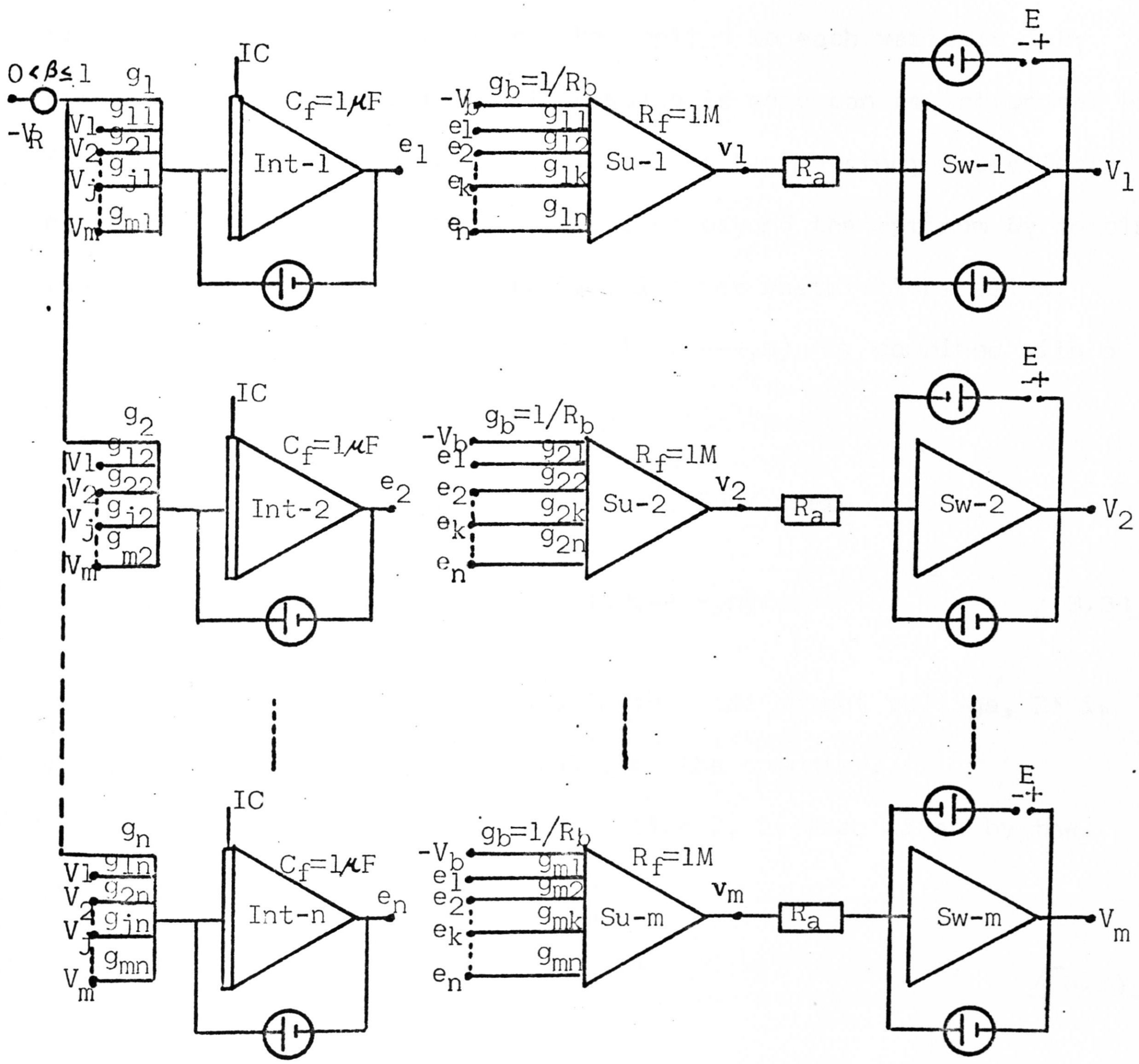


Figure 3.4 General Analog Computer Program in Terms of Input Gains

or greater than restrictions can be applied to each variable. By extension, variables whose optimum value is zero can be introduced by forcing a positive value. It is also possible to study the effect of increasing the value of the variables beyond the optimum by forcing the variable beyond this point, if all other restrictions allow.

Each of the variables e_k $(k = 1,2,----,n)$ is combined with a corresponding gain of

g_{ek} $(k = 1,2,----,n)$ in a summer[7], Su-Z where

$$g_{ek} = \frac{C_k}{S_k L} = p_{ek} G_{ek} \qquad (k = 1,2,----,n) \qquad [3.34]$$

where L is any constant, chosen such that the output voltage, Z^* is within the maximum permissible value of the computer.

The value of the objective function Z, is then given by the relation

$$Z = Z^* L \qquad [3.35]$$

Figure 3.5 gives the complete analog computer program to conduct parametric programming.

---

[7]Figure 2.3a on page 12 describes this circuit and its properties.

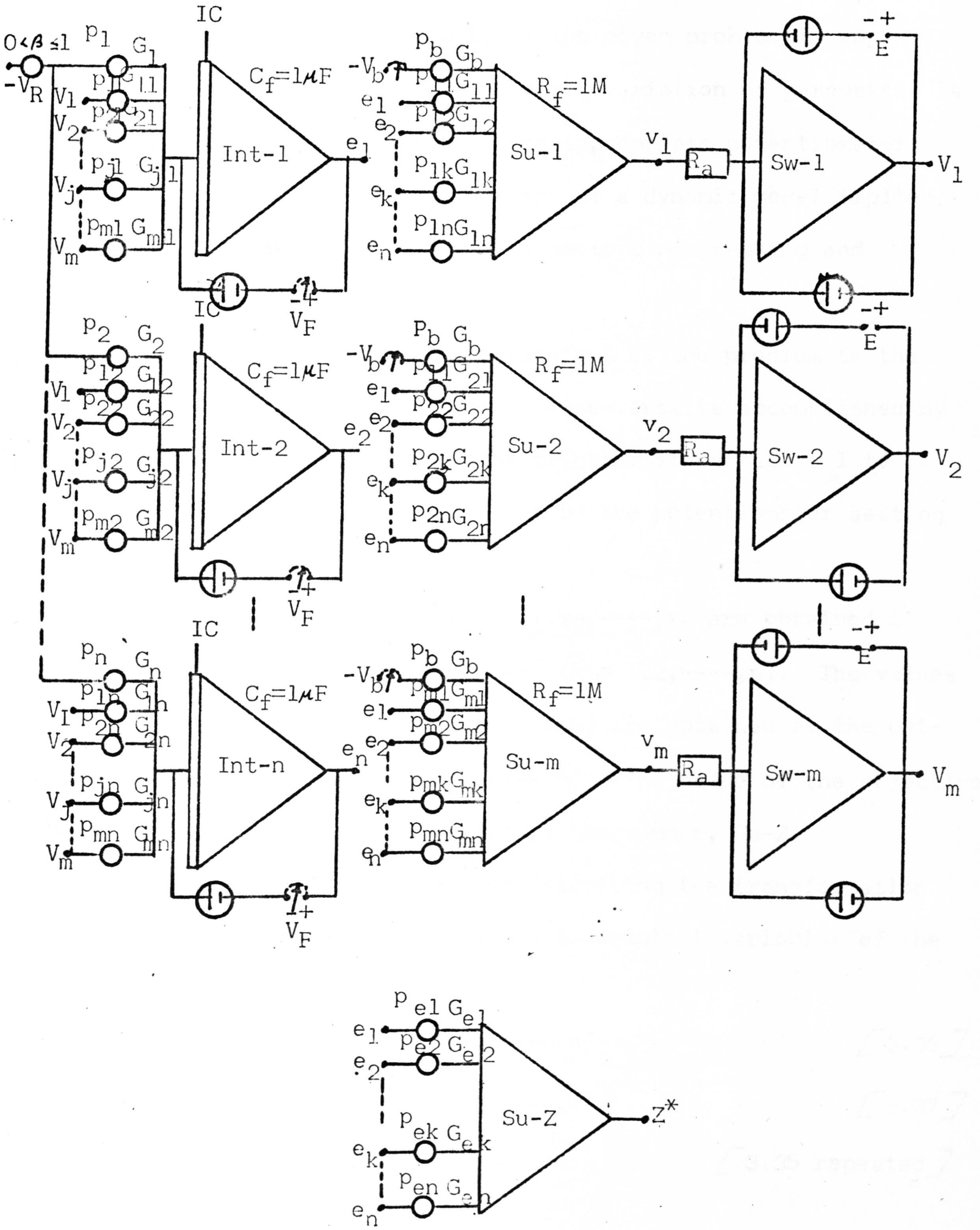


Figure 3.5 General Analog Computer Program for Conducting Parametric Programming

Each parameter C_k, A_{jk} and b_j of the given problem is represented by a potentiometer setting. Hence, variation of parameters is accomplished by the manipulation of the appropriate potentiometer. The ability to manipulate the parameters of a dynamic model implies that it would be ideally suited for parametric programming and sensitivity analysis.

The transformation from the parameters of the problem to the corresponding potentiometer setting or vice-versa is accomplished by a set of relations illustrated by the transformation tables 1 to 6. The speed of the solution is influenced by the potentiometer setting β.

The values of the n variables $x_1, x_2, ----, x_n$ are obtained at the output of the n integrators, Int-k $(k = 1,2,----,n)$. The values of the slack variables, W_j $(j = 1,2,----,m)$ are obtained at the output of the m summers, Su_j $(j = 1,2,----,m)$. The value of the objective function, Z is obtained at the output of the summer, Su-Z.

The following are the equations describing the transformation of the output from the machine voltages to original variables of the given problem.

$$x_k = \frac{e_k}{S_k} \qquad (k = 1,2,----,n) \qquad [3.36]$$

$$W_j = \frac{v_j b_j R_b}{V_b} \qquad (j = 1,2,----,m) \qquad [3.37]$$

$$Z = Z^* L \qquad [3.35 \text{ repeated}]$$

Additional Computational Aids

1. If the voltage - V_b of the jth summer, Su-j is increased by one volt, the corresponding jth restriction, b_j is relaxed by b_j/V_b units.

2. Any variable x_k not appearing in the optimum solution can be introduced into the solution by back-baising the corresponding integrator, Int-k with the variable voltage, V_F.

3. To shift the hyperplanes parallel to itself inward, lower the voltage applied to G_b of the corresponding summer from a value V_b to the required lower voltage; and to shift it outward, increase the voltage applied.

4. To minimize instead of maximize the objective function, apply $+V_R$ instead of $-V_R$ to the integrators.

5. To interchange the 'allowed' and restricted regions for any constraint hyperplane, reverse input and output terminals of the switching amplifier.

6. Negative coefficient among the a's must be treated by introducing an invertor in the corresponding paths. Since each such coefficient enters the circuit twice, with input voltages, e_k and V_j, from the output of the integrator and switch respectively, two sign changers are required for each negative coefficient.

7. A negative coefficient in the objective function is handled by using an invertor at the appropriate integrator input.

8. If the integrating capacitors are all started from a discharged condition, the objective point will start its motion from the origin. Whenever it is desirable to start the objective point from any other initial condition, it can be done by introducing appropriate initial voltages on the integrating capacitors.

CHAPTER IV

TESTING OF THE MODEL

The model illustrated in Chapter III was tested by solving a three dimensional numerical linear programming problem by both the simplex method on the digital computer and the analog computer approach on the analog computer.

Considering the linear programming problem:

$$\begin{array}{ll} \text{maximize} & Z = 10x_1 + 15x_2 + 8x_3 \\ \text{subject to} & R_1 : \cdot 5x_1 + x_2 + x_3 \leq 100 \\ & R_2 : \quad x_1 + 3x_2 \qquad \leq 150 \\ & R_3 : \; 2x_1 + 3x_2 + \cdot 5x_3 \leq 200 \\ & x_1 \geq 0, \quad x_2 \geq 0, \quad x_3 \geq 0 \end{array} \qquad [4.17]$$

The solution to this problem as provided by the digital computer using the simplex method is illustrated by the final simplex tableau shown in Table 4.1.

The positive index numbers under the body 0, 1·57 and 0 indicate the algebraic decrease that would occur in the objective function, Z, if one more unit of x_1, x_2 and x_3 respectively, were introduced in the solution.

TABLE 4.1

FINAL SIMPLEX TABLEAU OF PROBLEM [4.1]

		10·0	15·0	8·0	0·0	0·0	0·0
Basis		x_1	x_2	x_3	W_1	W_2	W_3
x_3	57.143	0·0	0·286	1·0	1·143	0·0	-0·286
W_2	64·268	0·0	1·571	0·0	0·286	1·0	-0·571
x_1	85·714	1·0	1·429	0·0	-0·286	0·0	0·571
Index Row		0·0	1·57	0·0	6·29	0·0	3·43

- - - - - Body- - - - - - - -Identity- - - -

Optimum Solution is $Z = 1314 \cdot 285$

The positive index numbers under the identity, 6·29, 0 and 3·43 represent the algebraic increase possible in the objective function, Z if the restricting constants, 100, 150 and 200 respectively, were relaxed by one unit.

This optimum solution (x_1 = 86·714, x_2 = 0, x_3 = 57·143 and Z = 1314·285) was accomplished in the seventh iteration.

To represent the problem given by equation [4.1] in an analog computer model, it is necessary to determine the various coefficient elements of the program. This can best be accomplished by following the procedure illustrated by Tables 3.1 to 3.6.

These calculations for the given problem are shown in Tables 4.2 through 4.7.

The analog computer program to simulate the given problem is given in Figure 4.1 on page 63, incorporating the values of the coefficient elements as given by Table 4.7 on page 64.

The program was run on the 'EASE BERKELEY' analog computer and eight sets of readings were tabulated as given in Table 4.8 on page 64, in terms of machine parameters and variables; and Table 4.9 on page 65, in terms of variables and parameters of the given problem, to demonstrate the ability of the analog computer solution to do parametric programming.

TABLES ILLUSTRATING THE TRANSFORMATION OF THE COEFFICIENTS OF THE PROBLEM TO MACHINE SETTINGS

	x_1	x_2	x_3	b
R_1	·5	1	1	100
R_2	1	3		150
R_3	2	3	·5	200
Z	10	15	8	

TABLE 4.2 Transformation Table - 1

	x_1	x_2	x_3
R_1	200	100	100
R_2	150	50	
R_3	100	66·66	400
Z	$1/10$	$1/15$	$1/8$

TABLE 4.3 Transformation Table - 2

Choosing the feedback impedances R_f = 1 meghohm and C_f = 1 microfarad and V_b = 50 volts and having E_{max} = 100 volts.

Input resistances R_b are given by equation [3.30]

$$\frac{R_b}{R_f} = \frac{E_{max}}{V_b}$$

$$R_b = \frac{100x1}{50} = 2M$$

Scale factors for x_1, x_2 and x_3 are determined from equation [3.31]

$$Sx_1 = \frac{V_b R_f}{R_b(\max a_{j1})} \qquad Sx_2 = \frac{V_b R_f}{R_b(\max a_{j2})} \qquad Sx_3 = \frac{V_b R_f}{(R_b(\max a_{j3})}$$

$$Sx_1 = \frac{50x1}{2x200} \qquad Sx_2 = \frac{50x1}{2x100} \qquad Sx_3 = \frac{50x1}{2x400}$$

$$Sx_1 = \frac{1}{8} \qquad Sx_2 = \frac{1}{4} \qquad Sx_3 = \frac{1}{16}$$

multiplying each column of Table 4.3 by the corresponding scale factors, we have

$e_1 = \frac{1}{8}x_1$	$e_2 = \frac{1}{4}x_2$	$e_3 = \frac{1}{16}x_3$
25	25	6.25
18.75	12.5	
12.5	16.66	25
$\frac{1}{80}$	$\frac{1}{60}$	$\frac{1}{128}$

TABLE 4.4 Transformation Table - 3

To obtain the various input resistive values, each a - element is multiplied by $\frac{R_b}{V_b} = \frac{1}{25}$ and each c - element by a convenient factor, $U = \frac{1}{6}$.

	$\frac{1}{8}x_1$	$\frac{1}{4}x_2$	$\frac{1}{16}x_3$
V_1	1M	1M	25M
V_2	·75M	·5M	
V_3	·5M	·66M	1M
Z^*	·75M	1	·469M

TABLE 4.5 Transformation Table - 4

Since the feedback impedances, R_f and C_f are chosen to be equal to unity, the input gains are given by the reciprocals of their corresponding input resistances as shown in Table 4.6.

	$e_1 = \frac{1}{8}x_1$	$e_2 = \frac{1}{4}x_2$	$e_3 = \frac{1}{16}x_3$
V_1	1	1	4
V_2	1·33	2	
V_3	2	1·5	1
Z^*	1·33	1	2·13

TABLE 4.6 Transformation Table - 5

Each of the input gains of Table 4.6 are then represented by a potentiometer in conjunction with a higher input gain using equations [3.33a] and [3.33b], as shown in Table 4.7.

	$e_1 = \frac{1}{8}x_1$	$e_2 = \frac{1}{4}x_2$	$e_3 = \frac{1}{16}x_3$
V_1	(·5), 2	(·5), 2	(·4), 10
V_2	(·66), 2	(·5), 4	
V_3	(·4), 5	(·75), 2	(·5), 2
Z^*	(·66), 2	(·5), 2	(·53), 4

TABLE 4.7 Transformation Table - 6

The input gains of the summer, Su-Z are calculated using equation [3.34].

Choosing the value L = 100

$$g_{e1} = \frac{10}{\frac{1}{8}x100} = \cdot 8 = (\cdot 8),\ 1$$

$$g_{e2} = \frac{15}{\frac{1}{4}x100} = \cdot 6 = (\cdot 6),\ 1$$

$$g_{e3} = \frac{8}{\frac{1}{16}x100} = 1\cdot 28 = (\cdot 512),\ 2\cdot 5$$

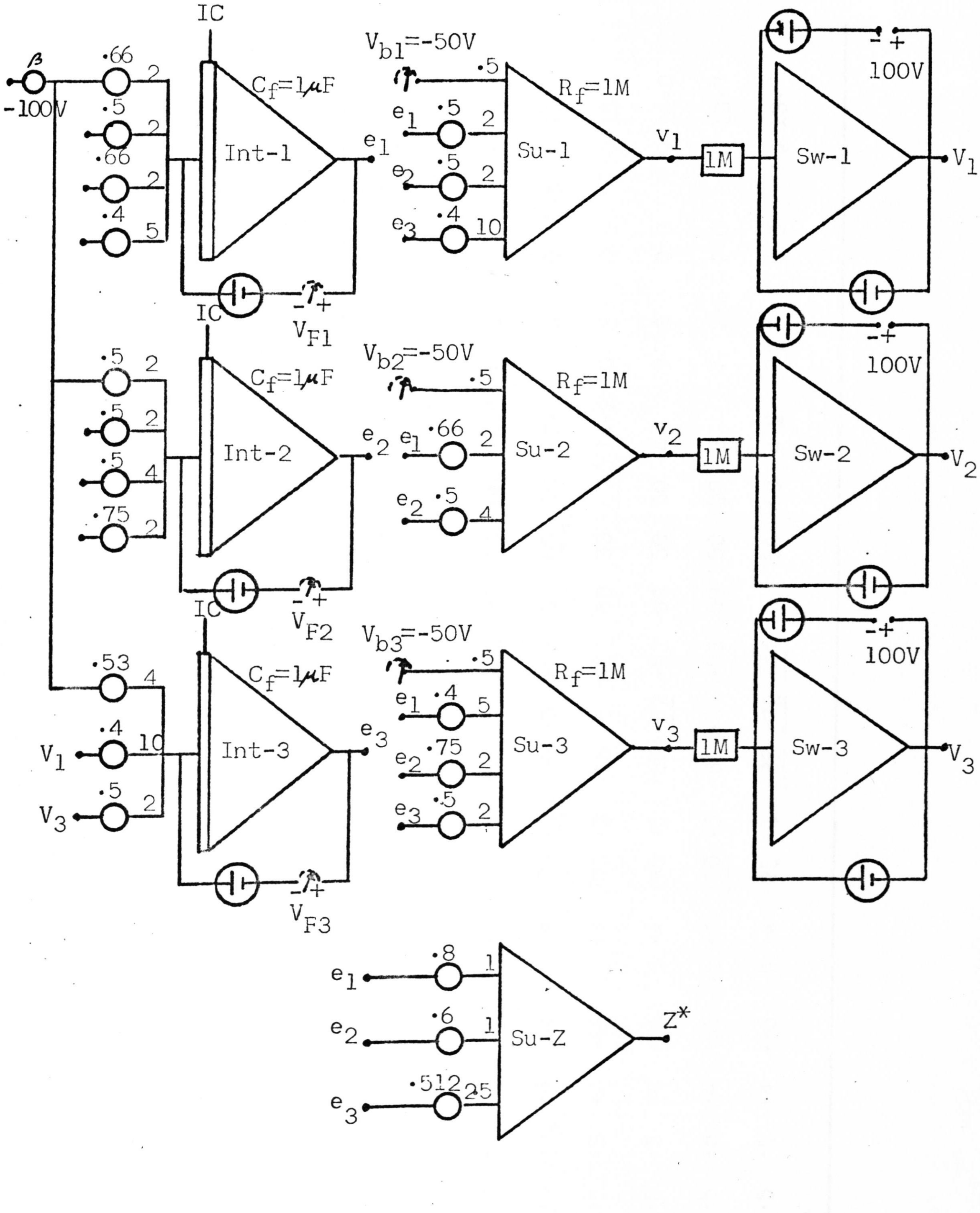


Figure 4.1 Analog Computer Program for Parametric Programming of Problem [4.1]

TABLE 4.8

READINGS IN TERMS OF MACHINE VARIABLES

	Matrix Coefficients								Objective Function Coefficients			Restrictions			Output Variables						
No.	p_{11}	p_{12}	p_{13}	p_{21}	p_{22}	p_{31}	p_{32}	p_{33}	p_1	p_2	p_3	V_{b1}	V_{b2}	V_{b3}	e_1	e_2	e_3	v_1	v_2	v_3	Z^*
1	.5	.5	.4	.66	.5	.4	.75	.5	.66	.5	.53	50	50	50	10.8	0	3.6	0	10.5	0	13.1
2	.5	.5	.4	.66	.5	.4	.75	.5	.66	.5	.53	50	50	50	7	5	3.25	0	5	0	12.8
3	.5	.5	.4	.66	.5	.4	.75	.5	.66	.5	.53	55	50	50	10.5	0	4.1	0	10.1	0	13.75
4	.5	.5	.4	.66	.5	.4	.75	.5	.66	.5	.53	50	60	50	10.8	0	3.7	0	10.5	0	13.1
5	.5	.5	.4	.66	.5	.4	.75	.5	.66	.5	.53	50	50	62.5	14.2	0	3	0	5	0	15.1
6	.5	.5	.4	.66	.5	.4	.75	.5	.66	.72	.53	50	50	50	4	9.7	3	0	0	0	14.0
7	.5	.5	.4	.66	.5	.4	.50	.5	.66	.5	.53	50	50	50	8	7	2.8	0	0	0	13.7
8	.5	.5	.4	.66	.5	.4	.75	.5	.88	.72	.44	50	50	50	4.2	9.2	3	0	0	0	13.7

TABLE 4.9

READINGS IN TERMS OF VARIABLES OF THE PROBLEM

	Matrix Coefficients								Objective Function Coefficients			Restrictions			Output			Variables			
No.	A_{11}	A_{12}	A_{13}	A_{21}	A_{22}	A_{31}	A_{32}	A_{33}	C_1	C_2	C_3	b_1	b_2	b_3	x_1	x_2	x_3	W_1	W_2	W_3	Z
1	.5	1	1	1	3	2	3	.5	10	15	8	100	150	200	86	0	57	0	63	0	1310
2	.5	1	1	1	3	2	3	.5	10	15	8	100	150	200	56	20	52	0	30	0	1280
3	.5	1	1	1	3	2	3	.5	10	15	8	110	150	200	84	0	72	0	62	0	1375
4	.5	1	1	1	3	2	3	.5	10	15	8	100	180	200	86	0	57	0	63	0	1310
5	.5	1	1	1	3	2	3	.5	10	15	8	100	150	250	114	0	48	0	30	0	1510
6	.5	1	1	1	3	2	3	.5	10	18	8	100	150	200	32	39	48	0	0	0	1400
7	.5	1	1	1	3	2	2	.5	10	15	8	100	150	200	64	28	45	0	0	0	1370
8	.5	1	1	1	3	2	3	.5	11	18	7	100	150	200	34	37	48	0	0	0	1370

An explanation of the readings in Tables 4.8 and 4.9 will enable a better understanding of the procedure of perturbating the model.

Row 1 gives the solution of the original problem.

Row 2: The variable x_2, not in the optimum solution, was introduced into the solution by means of the back-baised supply voltage, V_{F2}. This solution indicates that the objective function, Z reduces by 1·5 units for every x_2 unit that enters the solution.

Row 3: The restriction R_1 was relaxed by 10 units, using the variable supply V_{b1}. The objective function increased by about 65 units, indicating an increase in objective function by 6·5 units for every additional unit of b_1 available.

Row 4: The restriction R_2 was relaxed by 30 units, using the supply V_{b2}. The objective function value remained the same, indicating zero increase in objective function for every additional unit of b_2 available.

Row 5: The restriction R_3 was relaxed by 50 units, using supply V_{b3}. The objective function increased by about 100 units, indicating an increase in objective function by 4 units for every additional unit of b_3 available.

The above set of 5 readings agree with those provided by the simplex tableau given by Table 4.1 on page 57.

Row 6: The objective function coefficient, C_2 was changed to 18 by manipulating potentiometer p_2. Reading 6 gives the solution with this new parameter.

Row 7: The matrix coefficient A_{32} was changed from a value of 3 to 2, by manipulating potentiometers p_{32}. Reading 7 gives the solution with this new parameter.

Row 8: Objective function coefficients, C_1, C_2, and C_3 were all simultaneously changed and the solution obtained. Thus illustrating that any combination of changes can be represented on this model.

In sensitivity analysis it is often required to determine the range of any given parameter for which the problem, as originally stated, remains optimal. This can be accomplished on this model by merely varying the appropriate potentiometer until the basis of the solution changes. The sensitivity of the objective function of the given problem was checked with regard to implicit parameters in the neighborhood of the optimal vertex, and found in terms of potentiometer readings as follows:

$$.6 \leq p_1 \leq 1; \quad -0 \leq p_2 \leq .57; \quad .2 \leq p_3 \leq 1$$

which in terms of actual variables is

$$9 \leq C_1 \leq \infty; \quad -\infty \leq C_2 \leq 17; \quad 3 \leq C_3 \leq \infty$$

with C_k restricted by the appropriate bounds, the basis remains optimal but the value of the objective function changes.

CHAPTER V

SUMMARY AND CONCLUSIONS

A linear programming model can be effectively simulated on the analog computer by considering the dynamics of an objective point driven by a maximizing force along the gradient of the objective function hyperplane within the convex polyhedron, formed by the restrictions of the problem. Each restriction is represented as a barrier or hyperplane, with which is associated a constraint force. When the objective point is initiated anywhere within the polyhedron, it travels through the convex polyhedron along the gradient of the objective function until it strikes a hyperplane. It then travels along the boundary of the convex polyhedron and finally settles down at a vertex, the n co-ordinates of which represent the values of the n variables which constitute the optimum solution.

On the analog computer model it is possible to represent each parameter of the given problem by a potentiometer reading. Using this dynamic model the analog computer with its fast operating time immediately seeks new solutions to the problem as the parameters are varied, without the necessity of returning to the initial conditions.

The analog computer is thus a powerful tool for conducting parametric programming and sensitivity analysis.

In addition to solving a programming problem the model (figure 3.5 page 52) can be used to provide additional information regarding the given problem.

1. The effect of relaxing any restriction, b_j on the optimum solution, can be studied by increasing the corresponding variable voltage supply - V_b of the jth summer, Su-j. With an increase in one volt of - V_b, the restriction b_j is relaxed by b_j/V_b units.
2. The effect of introducing a variable x_k, not in the optimum solution or increasing its value beyond that given by the optimum solution, can be studied by introducing the back-baising voltage, V_F of the corresponding integrator, Int-k. By introducing one additional volt of V_F, $1/S_k$ units of x_k are forced into the solution.
3. The slope of the objective function hyperplane can be varied by manipulating the potentiometers, p_k $(k = 1,2,----,n)$.
4. The slope of the constraint hyperplanes can be varied by manipulating the potentiometers, p_{jk} $(j = 1,2,----,m \quad k = 1,2,----,n)$.
5. The values of the slack variables, W_j $(j = 1,2,----,m)$ are obtained at the output, v_j of the m summers, Su-j $(j = 1,2,----,m)$ and is given by

$$W_j = \frac{v_j b_j R_b}{V_b}$$

6. The objective point can be initiated anywhere within or outside the convex polyhedron by introducing appropriate initial voltages on the integrating capacitors, C_f of the integrators, Int-k $(k = 1,2,----,n)$. When the objective point is initiated outside

the polyhedron, it is immediately driven inside by the constraint voltages of the hyperplanes, it then proceeds along the boundary of the convex polyhedron to the optimal vertex.

7. A break-through by the objective point at a hyperplane-barrier occurs when the component of βV_R ($0 < \beta \leq 1$) normal to any hyperplane, is greater than the constraint voltage, E. A condition $E > V_R$ would therefore prevent any possible break-through at a hyperplane-barrier.

8. A case of degeneracy can be recognized if the output Z^*, of the summer Su-Z remains constant while the variables $x_1, x_2, ----, x_n$ continue to change values slowly.

9. Most optimum solutions were reached within one minute and accuracy obtained was within $\pm$ 5 percent. This is more than the accuracy of the input data generally available for economic studies.

10. The various voltages at the output of the summer, Su-j ($j = 1,2,----,m$) closest to zero correspond to those of the restrictions which actually determine the solution vertex. It is then possible to obtain solutions to the desired accuracy by solving these corresponding n simultaneous equations.

Mathematical programming problems with nonlinear restrictions can be simulated on the analog computer by using more elaborate equipments like function generators. However, using the procedure described in this thesis, any particular set of initial conditions will always produce the same solution, which might not be the optimum solution. Hence, there is a need to apply the analog computer approach to study nonlinear programming problems and related search techniques.

Another class of problems in the field of operations research is statistical linear programming, which involves the concepts of probability. That is, it involves phenomena that, although describable on a long term basis with various degrees of certainty, cannot be predicted on an instantaneous basis. Variables involved in such a system are called as stochastic variables.

Four situations can occur:

1. Objective function coefficients, C_k may be stochastic.
2. Matrix elements, A_{jk} may be stochastic.
3. Constraint constants, b_j may be stochastic.
4. Combination of these values may be stochastic.

Ideally, we would like to symbolize and analyse a system where some or all the values are subject to statistical fluctuation; given the probability distribution of the fluctuation of these parameters.

An analog computer approach to this problem needs to be investigated because statistical quantities can be represented on an analog computer by a time-varying voltage.

BIBLIOGRAPHY

1. Maynard, H. S., Industrial Engineering Handbook, Second Ed. McGraw Hill Book Co.

2. Kuhn, H. W. and Tucker, A. W., Nonlinear programming in "Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability", University of California Press, Berkeley, Calif., 1951.

3. Charles, A. and Lemke, C. E., "Minimization of Nonlinear Separable Functions", Graduate School of Industrial Administration, Carnegie Institute of Technology, Pittsburg, Pa., 1954.

4. Jackson, A. S., Analog Computation, McGraw Hill Book Co., 1960.

5. Pyne, I. B., "Linear Programming on an Analog Computer," Trans, AIEE, Part I, p. 142, 1956.

6. Fifter, S., Analogue Computation, McGraw Hill Book Co., 1961.