A COMPARISON OF THE SPRIET-BARON AND EXTENDED COGGINS ALGORITHMS FOR THE SUBMERGED SEWAGE DISPERSION MODEL

BY

AUDU, UMAR OMESA
M.TECH/SSSE/2000/585

A THESIS SUBMITTED TO THE DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE, FEDERAL UNIVERSITY OF TECHNOLOGY, MINNA, NIGERIA. IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE AWARD OF MASTERS OF TECHNOLOGY (MTECH) DEGREE IN MATHEMATICS.

July, 2004

DEDICATION

TO ALLAH, SUBHANA WATA'ALA

TO MY LATE FATHER AUDU OMESA AND MY MOTHER RABIATU WHO SHARED MY PHYSICAL BEING

TO MY TEACHERS – THE BEGINNING AND END OF MY EDUCATION

TO MY PATIENT WIFE (FATIMA) WHO SHARED THE BURDEN OF MY ADVERSITIES AT THIS PERIOD

TO MY CHILDREN WHO SUFFERED SO MANY CONSTRAINTS IN MANY WAYS

AND TO MY BROTHERS, SISTERS AND FRIENDS.

Finally wish to express my gratitude to Major A.M. Peni whose persistent advice and encouragement I so much cherish. Abubakar Al-sodia Audu (the orator), Iliya Isah Peni, all the foreign brothers, Abdul-rahman Ndanusa, Yusuf Nuhu, Mal Hakimi Danladi, Engr Mahmud A. Manga, Engr (Alh) Aliyu Muazu, Yusuf Kaka, Alh Aminu K. Ahmed, Grace Tela and all my well wishers, too numerous to mention. May Allah reward all of them bountifully (Amen).

Audu U.O.
Minna, June 2004.

ABSTRACT

In this work, we applied the unconstrained non–gradient optimization algorithms of Spriet–Baron and Coggins to solve the Submerged Sewage Dispersion Model and compared the output results of the two algorithms alongside with an analytical solution. The output results show that both methods attain the global minimum at 2.1×10^{-4}. In doing so, the number of iterations for the Spriet–Baron is 184 while that of the Extended Coggins is 12. This shows that the Extended Coggins algorithm is a better algorithm for the Sewage Dispersion model considered in this work as it converges much faster.

CHAPTER ONE

INTRODUCTION TO OPTIMIZATION THEORY

1.1 PREAMBLE

The evolution of optimization theory originates among many others, with economic problems and game theory where optimal strategy was to be described mathematically.

Stephenson (1971), postulates that the activity of man is developed entirely trying to optimize the various situations he finds himself. In the light of this, optimization can be defined as the art for determining the best decision in a given set of circumstances.

Optimization is a field of applied mathematics consisting of a collection of principles and methods used for the solution of quantitative problems in many disciplines: physics, biology, engineering, economics, business and others. Mathematically, the purpose of optimization is to find the best solution to a given problem (which may also include a number of limiting constraints). This mathematical area, optimization, grew from the recognition that problems under consideration in manifestly many fields could be posed theoretically in such a way that a central store of ideas and methods could be used in obtaining solution for all of them.

A typical optimization problem may be described in the following way

Example

There is a system, such as a physical machine, a set of biological organism or a business organization whose behaviour is determined by several specified factors. The operation of the system has a goal as the optimization of the performance of this system. The latter is determined at least in part by the level of the factors over which the operator has control; the performance may also be affected however by other factors over which there is no control. The operator seeks the right levels for the controllable factors that will optimize, as far as possible, the performance of the system.

For example, in the case of a banking system, the operator is the governing body of the central bank; the inputs over which there is control are interest rates and money supply; and the performance of the system is described by economic indicators of the economic and political units in which the banking system operates.

The first step in the application of optimization theory to a practical problem is the identification of relevant theoretical components. This is often the most difficult part of the analysis, requiring a thorough understanding of the operation of the system and the ability to describe the operation of the system in precise mathematical terms. Generally, on the development of optimization technique one begins the construction of such a method, according to Polak (1971) by inventing a conceptual algorithm.

Then one modifies this conceptual process in such a way as to reduce each of its iteration to a finite number of digital computer operations. That is, one reduces it to an implementable algorithm. To achieve this objective, in an effective manner, one has to use an adaptive or closed loop method for truncating at least some of the infinite sub procedures. This approach has the advantage of avoiding a great deal of time put into very precise calculations when one is still quite far from the optimal point that one is trying to find. To make matters worse, the resulting algorithm may fail to converge. Bonday (1984) supported this same view point by saying that it is not always economical to do a thorough linear search. All that is necessary, he said, is to obtain a reduction in the function value. At the first sight, this may seem rather crude. The computation to find the minimum in this direction might be considerable. Again he stated that practical experience with these types of problems shows that it is just not worthwhile. He t herefore presumed that what we lose on the accuracy swing at this stage we make up for on the progress to the minimum via changes in direction roundabouts.

Looking at example one, the main theoretical components are the system, the inputs and outputs, and its rules of operation. The system has a set of possible states at each moment in the life of the system, it is one of these states, and it changes from state to state according to certain rules determined by inputs and outputs. There is a numerical quantity called the performance measure, which the operator seeks to maximize or minimize. It is a mathematical function whose value is determined by the history of the system. The operator is able to influence the value of the performance measure through a schedule of inputs. Finally, the constraints of the system must be identified; these are the restrictions on the inputs that are beyond the control of the operator.

Frankly speaking, the modern large scale digital computer has given a great impetus to computational procedures of solving large class of optimization problems.

1.2 NUMERICAL OPTIMIZATION PROBLEM

Many problems involve finding the best, in some defined respect, of many possible solutions. The best solution might be the one leading to the lowest cost, the largest profit or the shortest route in a journey. Such problems are ones of optimization. Because of their economic importance, their effective computational solution is extremely important.

1.2.1 STATEMENT OF AN OPTIMIZATION PROBLEM

(i) FOR AN UNCONSTRAINED PROBLEM

The mathematical problem is to find a set of values x_i such that $F(x_i)$ is as small (or as large) as possible. Simply put: Find

$$x = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}$$

which minimizes $F(x)$.

(ii) FOR A CONSTRAINED PROBLEM

The mathematical problem is to find a set of values x_i such that $F(x_i)$ is as small (or as large) as possible. Simply put: Find

$$x = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}$$

which minimizes $F(x)$.

Subject to the constraint:

$$g_j(x) \leq 0; \quad j = 1,\ 2,\ \ldots, m$$

$$L_j(x) = 0; \quad j = 1,\ 2,\ \ldots, p$$

Where x is an n-dimensional vector called the design vector, i.e. x_i means the set of all x_i; $i = 0,\ 2,\ \ldots, n$.

The function F or $F(x)$ represents the cost or other value to be optimized and it is called the objective function. And the problem is usually defined so that the cost (objective) is to be minimized. $g_j(x)$ and $l_j(x)$ are, respectively, the inequality and the equality constraints.

The number of variables n and the number of constraints m and/or p need not be related in any way.

In most optimization problems, the objective function F depends on several variable, $x_1, x_2, \ldots, x_n$. These are called the control variables because we can control them, that is, chose their value. Generally, in any optimization problem the objective is to optimize (maximize or minimize) some function f. This function is called the objective function. Optimization theory develops methods for optimal choice of $x_1, x_2, \ldots, x_n$ which maximize (or minimize) the objective function f. that is method for finding optimal values of $x_1, x_2, \ldots, x_n$.

1.2.2 CONSTRAINED/ UNCONSTRAINED VARIABLES

In many problems the variable x_i (i.e choice of values of $x_1, x_2, \ldots, x_n$) are not entirely free but are subject to constraints, that is additional conditions arising from the nature of the problem and the variable.

These constraints can be equality constraints, or both. They take the form
$q_j(x_i) = 0, \;\; j = 1, 2, \ldots, p$ for equality constraints and $g_k(x_i) \geq 0, \;\; k = 1, 2, \ldots, m$ for inequality constraints. Either or both of p and m can be zero, meaning that there are no constraints in that class.

1.2.3 LINEAR PROGRAMMING

The objective function and the constraints may be linear or non-linear. If both are linear, the problem belong to the speciality called linear programming.

A linear programming is defined as the minimization of a linear objective function whose variable satisfy a system of linear inequalities.

Linear programming or linear optimization consists of methods for solving optimization problems in which the objective function F is a linear function of control variables $x_1, x_2, \ldots, x_n$ and the domain of these variables restricted by system of linear inequalities. Problems here can also involve thousands of variables and require the solution of numerous linear equations at each step of an iterative process.

1.2.4 NON-LINEAR PROGRAMMING

Non–linear programming are those in which either the objective function or at least one of the constraint function is non–linear.

1.2.5 MATHEMATICAL PROGRAMMING

Both linear and non–linear programming falls under the specificity, referred to as mathematical programming. Mathematical programming may be described in terms of its mathematical structure and computational procedures or in terms of the broad class of important decision problems which can be formulated as the minimization (maximization) of a function of several variables that are subject to system of side constraints.

1.3 DEFINITION OF TERMS

Definition 1.3.1 – DESIGN VECTOR

This is described by a set of quantities some of which are viewed as variables during the design process.

Definition 1.3.2 – PREASSIGNED PARAMETERS

These are the quantities that are usually fixed at the outset in any engineering system or components.

Definition 1.3.3 – DESIGN OR DECISION VARIABLE

These are the quantities that are treated as variables in the design process in any engineering system. The design variables are collectively represented as a design vector, thus:

$$x = \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix}$$

Definition 1.3.4 – DESIGN CONSTRAINTS

In many practical problems, the design variables cannot be chosen arbitrarily; rather, they have to satisfy certain specified functional and other requirements. The restrictions that must be satisfied in order to produce an acceptable design are collectively called design constraints.

The constraints which represent limitations on the behaviour or performance of the system are termed as behaviour or functional constraints.

The constraints which represent physical limitations on the design variables like availability, fabricability and transportability are known as geometric or side constraints.

Definition 1.3.5 – OBJECTIVE FUNCTION

Note that, the conventional design procedure aims at finding an acceptable or adequate design which merely satisfies the functional and other requirements of the problems. In general, there will be more than one acceptable designs and the purpose of optimization is to choose the best one out of the many acceptable designs available. Thus a criterion has to be chosen for comparing the different alternate acceptable designs and for selecting the best one. The criterion with respect to which the design is optimized when expressed as a function of the design variables is known as the criterion or merit or objective function.

Definition 1.3.6 – OPTIMIZATION TECHNIQUE

The various technique(s) available for the solution of optimization problem(s) are classified under the heading mathematical programming technique (also known as the optimum seeking methods). These techniques are useful in finding the minimum or maximum of a function of several variables under a prescribed set of constraints.

Example of such classification includes the classical methods of differential calculus which can be used to find unconstraint maximum or minimum of a function of several variables.

1.4 CLASSIFICATION OF OPTIMIZATION PROBLEMS

Generally, optimization problems can be classified as follows:

1.4.1 CLASSIFICATION BASED ON THE EXISTENCE OF CONSTRAINTS

As already stated, any optimization problem can be classified as a constrained or unconstrained one depending upon whether the constraints exist or not in the problem.

1.4.2 CLASSIFICATION BASED ON THE NATURE OF DESIGN VARIABLES

Taking into cognisance the nature of design variables encountered, optimization problem can be classified into two broad categories, viz:

Category I

The problem is to find values to a set of design parameters, which make some prescribed function of these parameter minimum subject to certain constraints.

Category II

The objective is to find a set of design parameters, which are all continuous functions of some other parameter, that minimize an objective function subject to the prescribed constraints.

1.4.3 CLASSIFICATION BASED ON THE PHYSICAL STRUCTURE OF THE PROBLEM

Considering the physical structure of the problem, optimization problem can be classified as optimal control and non-optimal control problems.

Two types of variables usually describe an optimal control problem, viz:

(i) The control (design) variables

(ii) The state variables

The control variables govern the evolution of the system from one stage to the next and the state variables describe the behaviour of the system in any stage. Clearly stated, the optimal control problem is a mathematical programming problem involving a number of strategies, where each stage evolves from the stage in a prescribed manner.

Optimal control problem are stated as follows:
Find the set of control or design variables such that the total objective function over the L number of stages is minimized subject to certain constraints on the state and control variable. i.e.

Find x which minimizes

$$F(x) = \sum_{i=0}^{L} f_i(x_i,\ y_i)$$

subject to the constraints

$$q_i(x_i,\ y_i) = y_{i+1};\quad i = 1, 2, \ldots, L$$

$$g_j(x_j) \leq 0;\quad j = 1, 2, \ldots, L$$

and

$$h_k(y_k) \leq 0;\quad k = 1, 2, \ldots, L$$

where
x_i is the i^{th} control variable;
y_i is the i^{th} state variable;
f_i is the contribution of the i^{th} stage to the total objective function ;
g_j h_k and q_i are functions of x_i, y_k; and x_i and y_i respectively.

1.4.4 CLASSIFICATION BASED ON THE NATURE OF EQUATIONS INVOLVED

This classification is based on the nature of the expression for the objective function and the constraints. Here, optimization problems can be classified as:

(i) Linear programming problems

(ii) Non–linear programming problems

(iii) Geometric programming problems

(iv) Quadratic programming problems

This classification is extremely useful from the computational point of view since there are many methods developed solely for the efficient solution of a particular class of problems.

(i) LINEAR PROGRAMMING PROBLEM

If the objective function and all the constraints in equation 1.2.2 (a and b) are linear functions of the design variables, the mathematical programming problem is called a linear programming (LP) problem. A linear progrmming problem is often stated as follows:

Find

$$x = \left\{ \begin{matrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{matrix} \right\}$$

which minimizes

$$F(x) = \sum_{i=1}^{n} c_i x_i$$

subject to the constraints

$$\sum_{k=1}^{n} a_{jk} x_k = b_j; \quad j = 1, 2, \ldots, m$$

and $x_i \geq 0, \quad i = 1, 2, \ldots, n$
where c_i, a_{jk} and b_j are constants.

(ii) NON–LINEAR PROGRAMMING PROBLEM

If any of the function among the objective and constraints function 1.2.1 a and b is non–linear, the problem is called a non–linear programming problem (NLP).

(iii) GEOMETRIC PROGRAMMING PROBLEM

A geometric programming problem (GMP) is one in which the objective function and constraints are expressed as posynomials in x.

Definition

A function h(x) is called a posynomial if h can be expressed as the sum of power terms of the form:

$$c_i;\ x_1^{a/1},\ x_2^{a/2}, \ldots, x_n^{a/n}$$

where c_i and a_{ij} are constraints with $c_i > 0$ and $x_j > 0$

Thus a posynomial fuction can be expressed as

$$H(x) = c_i x_1^{a_{1n}} x_2^{a_{2n}} \ldots x_n^{a_{nn}}$$

Thus the GMP problem can be stated as follows:
Find x which minimizes

$$F(x) = \sum_{i=1}^{N_c} c_i \left[\Pi_{j=1}^{n} x_j^{p_{ij}}\right]; \quad c_i > p, \quad x_j > 0$$

subject to

$$g_j(x) = \sum_{i=1}^{N_j} a_{ij} \left[\Pi_{j=1}^{n} x_k^{a_{ik}}\right] \leq 0; \quad j = 1, 2, \ldots m$$

where N_c and N_j denote the number of posynomial terms in the objective and j^{th} constraint function respectively.

(iv) QUADRATIC PROGRAMMING PROBLEM

A quadratic programming problem is a non-linear programming problem with a quadratic objective function and linear constraints. The problem is formulated as follows:

Find x which minimizes

$$F(x) = c + \sum_{i=1}^{n} q_i x_i + \sum_{i=1}^{n} \sum_{j=1}^{n} Q_{ij} x_j$$

subject to

$$\sum_{i=1}^{L} a_{ij} x_i = b_j; \quad j = 1, 2, \ldots m; \quad x_i \geq 0, \quad i = 1, 2, \ldots, n$$

where c, q_i, Q_{ij} and b_j are constants.

1.4.5 CLASSIFICATION BASED ON THE PERMISSIBLE VALUES OF THE DESIGN VARIABLES

Depending on the values permitted for the design variables, optimization problem can be classified as follows:

(i) Integer programming problems

(ii) Real-valued programming problems

(i) INTEGER PROGRAMMING PROBLEMS

If some or all of the design variables x_1, x_2, ..., x_n of an optimization problem are restricted to take only intergcr (or discrete) values, the problem is called an integer programming problem.

(ii) REAL–VALUED PROGRAMMING PROBLEMS

If all the design variables are permitted to take any real value, the optimization problem is called a real–valued programming problem.

1.4.6 CLASSIFICATION BASED ON THE DETERMINISTIC NATURE OF THE VARIABLES INVOLVED

Based on the deterministic nature of the variables involved, optimization problem can be classified as deterministic and stochastic programming problems.

This is an optimization problem in which some or all of the parameters (design variables and/or preassigned parameters) are probabilistic, stochastic or deterministic as the case may be.

1.4.7 CLASSIFICATION BASED ON THE SEPARABILITY OF THE FUNCTIONS

Based on the separability of the functions (objective and constraints), optimization problem can be classified as

(i) Separable programming problem

(ii) Non–separable programming problem

(i) SEPARABLE PROGRAMMING PROBLEM

A function F(x) is said to be separable if it can be expressed as the sum of n single variable function $f_1(x)$, $f_2(x)$, ..., $f_n(x)$, i.e.

$$F(x) = \sum_{i=1}^{n} f_i(x_i)$$

A separable programming problem is one in which the objective function and the constraints are separable and can be expressed in standard form as

Find x which minimizes

$$F(x) = \sum_{i=1}^{n} f_i(x_i)$$

subject to

$$G_j(x) = \sum_{i=1}^{n} g_{ij}(x_i) < b_j; \quad j = 1, 2, \ldots m$$

where b_j's are constants.

(ii) NON–SEPARABLE PROGRAMMING PROBLEM

A non–separable programming problem is one in which the objective function and/or the constraints are non–separable.

1.4.8 CLASSIFICATION BASED ON THE NUMBER OF THE OBJECTIVE FUNCTIONS

Depending on the number of objective functions to be minimized, optimization problems can be classified as single and multi–objective programming problems.

A multi–objective programming problem can be stated as follows:

Find x which minimizes

$$F_1(x), \; F_2(x), \; \ldots, \; F_k(x)$$

subject to

$$g_j(x) \leq 0, \; j = 1, 2, \ldots m$$

where F_1, F_2, ... K_k denote the objective functions to be minimized simultaneously.

1.4.9 GENERAL APPRAISAL OF OPTIMIZATION THEORY

As noted earlier on, the first step in the application of optimization theory to a practical problem is the identification of relevant theoretical components, that is:

(a) a thourough understanding of the operation of the system; i.e. conceptual algorithm.

(b) the ability to describe the operation of the system in precise mathematical terms i.e. Implementation algorithm.

The next step is the choice of an appropriate method to be used. However, the method used for solving most optimization problem are often grouped as gradient and non-gradient methods. The gradient method requires function and derivative evaluation while the non-gradient method requires function evaluation only. These are further elaborated as follow:

Most methods for solving constrained optimization problem employ the first and sometimes the second partial derivatives of the objective function. The choice of such method is clear because for example, first and second derivatives of a function define its gradient and curvature and thereby determine the existence and location of the extremum which solves the problem under consideration.

However, in practical optimization problem, it frequently occurs that the evaluation of the function and constraints involve a lengthy and complicated calculation and as a consequence it is difficult or even impossible to derive explicit expression for the required derivatives by means of finite difference approximation. However, the use of this approach can introduce truncation and or cancellation errors which may nullify the theory underlying the chosen algorithm and lead the search astray so that it converges to the solution only very slowly.

An alternate approach to the use of finite difference is to employ an optimization procedure which does not call for derivative values. Such non-gradient methods are termed DIRECT SEARCH METHOD. The direct search strategies for generating a sequence of improving approximation to the solution are based simply on comparison of function values, and generally though not always, methods are heuristic in nature having little or no mathematical basis. By their nature, they make only very limited assumption about the function and generally no more than continuity so as a result they have a very wide field of applications. Thus not only can they be used in problems for which differentiation is difficult but also for those cases where it may be appropriate; derivatives are discontinuous, or when the function values are subject to errors. These are situations in which gradient-based methods can prove ineffective or inefficient. Most of the direct methods are little affected by such difficulties.

Furthermore, because of their lack of assumption about the function, they can prove more reliable and stable than the gradient-based methods, or most of them, because of their lack of a basis, and hence assumed inefficiency, one should not ignore them from practical point of view.

1.5 AIM AND OBJECTIVE OF THE STUDY

The aims and objectives of the study are:

1. To review the direct search technique of Spriet and Baron for the submerged sewage dispersion model.

2. To review the Coggins optimization algorithm for the submerged sewage dispersion model.

3. To find the most efficient line search algorithm in attaining the minimum for the submerged sewage dispersion model.

CHAPTER TWO

SUBMERGED SEWAGE DISPERSION MODEL

2.1 THE SPRIET–BARON MODEL

2.1.1 INTRODUCTION

Most urban communities located on a sea shore utilise or consider utilising a deceptively simple system of disposal of their sewage water after a rough preliminary treatment (sedimentation), the liquid is pumped to a linear diffusor enclosed on the sea floor, at several kilometers from the shore under a submergence of some 50 meters. The diffusor itself is a Sparger pipe, 2 to 4 meters in diameter, and pierced with equidistant side holes of 5 to 10 centimeters diameter. When the sea current is naught, the buoyant Jets formed at the side holes unite near the diffusor into a linear vertical buoyant plume whose behaviour was studied in great detail for the case of Laminer flow [1] and for that of turbulent flow [1, 2]. It has been shown for instance, that the maximum density difference between sea water and the plume decreases assymptotically (when the distance to the diffusor, y, decreases) like

$$y^{-3/5}F_0^{4/5}$$

for Laminar flow and like

$$y^{-1}F_0^{2/3}$$

for turbulent flow.

As the submergence is finite, these plumes are eventually deflected into horizontal buoyant plumes either at the sea surface or at the level of a thermodine if the flux of density difference per unit length of diffusor, F_0, is small enough.

The structure of these horizontal buoyant plumes has not yet been thoroughly investigated, and therefore prevailing design methods of marine sewage disposal system [6] take only the dispersion in vertical plumes into account. The Spriet Baron model gives the main results for the case of linear Laminar horizontal buoyant plumes.

2.1.2 THE CONSERVATION EQUATION

When the Bousinesq hypothesis (which allows one to study the effects of Buoyancy) pertaining to natural convection in a quasi incompressible (partly constant density) fluid applies, the momentum and energy equations respectively assume the following form for bi-dimensional flow [0x is horizontal, 0y is vertical]:

$$\frac{\partial(\Theta,\ \Psi)}{\partial(x,\ y)} = -\frac{\partial\Theta}{\partial x} + \frac{1}{G_r^{1/2}}\Delta^2\Psi \tag{2.1}$$

$$\frac{\partial(\Theta,\ \Psi)}{\partial(x,\ y)} = \frac{1}{P_r G_r^{1/2}}\Delta\Theta \tag{2.2}$$

where
Ψ = Stream function
Θ = reduced density difference
G_r = Grashof number defined as the ratio of buoyant to to viscous forces given as

$$G_r = \frac{g\rho_0^2\beta(T_1 - T_0)L^3}{\mu^2}$$

where
β = Temperature coefficient of volume expansion
$T_1 - T_0$ = is a characteristics temperature difference of the system.
L = characteristic dimension
ρ = mass density
μ = absolute or dynamic viscosity

g = gravity

P_r = Prandtl number which is the ratio of diffusivity of momentum to the diffusivity of heat.

$$P_r = \frac{\mu c_p}{k}$$

c_p = specific heat at constant presure

k = the thermal conductivity

The plumes considered by Spriet–Baron model are infact Prandtl boundary layers along the 0x axis. To find their assymptotic solution for $G_r \to \infty$ in the vicinity of y = 0 (inner solution), one has to stretch y and Ψ as follows:

$$y = yG_r^{3/10} \qquad (2.3(i))$$

$$\Psi = \Psi G_r^{3/10} \qquad (2.3(ii))$$

The fundamental term in the inner solution satisfies then

$$\frac{\partial\Psi}{\partial Y}\frac{\partial^3\Psi}{\partial x\partial Y^2} - \frac{\partial\Psi}{\partial x}\frac{\partial^3\Psi}{\partial Y^3} = -\frac{\partial\Theta}{\partial x} + \frac{\partial^4\Psi}{\partial Y^4} \qquad (2.4)$$

$$\frac{\partial\Psi}{\partial Y}\frac{\partial\Theta}{\partial x} - \frac{\partial\Psi}{\partial x}\frac{\partial\Theta}{\partial Y} = \frac{1}{P_r}\frac{\partial^2\Theta}{\partial Y^2} \qquad (2.5)$$

The inner solution will be valid to

$$O(G_r^{-1/10})$$

However, the The Spriet–Baron model in investigating the behaviour of horizontal buoyant plumes consider those plumes formed at the sea surface, on the sea floor and the case where the plume is submerged at the level of a thermocline.

2.1.3 SUPERFICIAL HORIZONTAL BUOYANT PLUMES

When the plume is formed at the sea surface the solution of (2.4) and (2.5) must satisfy the following boundary conditions:

$$Y = 0; \quad \Psi = 0; \tag{2.6}$$

(which means that the surface is a stream line.)

$$\frac{\partial^2 \Psi}{\partial Y^2} = 0 \tag{2.7}$$

(which implies that no shear stress at the surface).

$$\frac{\partial \Theta}{\partial Y} = 0 \tag{2.8}$$

(which implies that there is no heat flux to the atmosphere).

$$Y = \infty, \quad \frac{\partial \Psi}{\partial Y} = 0 \tag{2.9}$$

(this means that there is no velocity in the x direction far from the surface). $\Theta = 0$ (no effect on specific mass far from the surface) (2.10)

The problem admits the following similarity solution:

$$\Psi = \sqrt{x} f(\eta), \quad \Theta = \frac{1}{\sqrt{x}} g(\eta) \tag{2.11}$$

where the similarity variable is

$$\eta = \frac{Y}{\sqrt{x}} \tag{2.12}$$

The function f and g satisfy the system

$$f''' = \frac{1}{2} f f'' - \frac{1}{2} \eta g \tag{2.13}$$

$$g' = -\frac{P_r}{2} f g \tag{2.14}$$

and the boundary conditions

$$\eta = 0, \; f = f'' = 0 \tag{2.15}$$

$$\eta = \infty, \; f' = 0 \tag{2.16}$$

Moreover, to completely determine the solution the enthalply flux is normalized:

$$\int_{-\infty}^{\infty} U \Theta dY = \int_{-\infty}^{\infty} f' g d\eta = 1 \tag{2.17}$$

This problem was solved with an optimization scheme [5]. For numerical integration a 4th order Runge-Kutta Gill method was used. It is possible to check that the numerical solution is correct for large values of Prandtl (the assymptotic solution is easily found); for instance

$$\lim_{P_r \to \infty} g(\eta) = g(0) \exp\left[\frac{-P_r f'(0)}{4} \eta^2\right] \tag{2.18}$$

and

$$\lim_{P_r \to \infty} \frac{g(0)\sqrt{f'(0)}}{\sqrt{P_r}} = \frac{1}{2\sqrt{\pi}} \tag{2.19}$$

It is worth remarking that the dilution along the surface is by no means negligible. $\Theta(0)$ varies like $x^{-1/2} F_0^{5/6}$, while the superficial velocity is independent of x (and proportional to $F_0^{1/3}$).

2.1.4 HORIZONTAL BUOYANT PLUMES ON THE SEA FLOOR

If instead of urban sewage water one would pump some dense industrial effluent to the distributor, the boundary layer would now spread on the sea floor. The velocity field and density difference field are again given by (2.4), (2.5), (2.6), (2.8), (2.9) and (2.10) instead of (2.7).

$$Y = 0, \; \frac{\partial \Psi}{\partial Y} = 0 \tag{2.20}$$

(which means that the velocity is zero on the sea floor). The similarity solution (2.11), (2.12) still applies

$$\Psi = \sqrt{x} f(\eta)$$

and

$$\Theta = \frac{1}{\sqrt{x}} g(\eta)$$

where f and g are given by (2.13) and (2.14) but under the boundary condition

$$\eta = 0, \; f = 0, \; f' = 0 \tag{2.21}$$

$$\eta = \infty, \; f' = 0 \tag{2.22}$$

and with the conservation equation for the enthalpy flux

$$\int_{-\infty}^{\infty} f' g d\eta \tag{2.23}$$

The assymptotic solution for $P_r \to \infty$ is such that

$$\lim_{P_r \to \infty} g(\eta) = g(0) \exp\left[-\frac{P_r f''(0)}{12} \eta^3 \right] \tag{2.24}$$

and

$$\lim_{P_r \to \infty} \frac{g(0) [f''(0)]^{1/3}}{[P_r]^{2/3}} = \frac{3}{2(12)^{2/3}} \Gamma(2/3) \tag{2.25}$$

2.1.5 HORIZONTAL BUOYANT PLUMES SUBMERGED AT THE LEVEL OF A THERMOCLINE

When the sea density increases with depth one says that the sea is stably stratified. A vertical density profile then typically displays two or more plateaux some 10 to 100 metres deep separated by transition zones of only 1 meter depth, the thermoclines. If a vertical plume reaching a thermocline has lost enough buoyancy underway it will be deflected horizontally and feed a so called submerged sewage field. To model this field, suppose that a known flow of liquid of density equal to the mean density between those of the adjacent plateaus is injected at the level of the (infinitely thin) thermoclines. The resulting plume will be symmetric with respect to 0x and the equation describing this free shear boundary layer are again (2.4), (2.5) with the boundary condition:

$$Y = 0, \quad Psi = 0 \tag{2.26}$$

0x is a streamline;

$$\frac{\partial^2 \Psi}{\partial Y^2} = 0; \tag{2.27}$$

the horizontal velocity profile is symmetric with respect to 0x.

$$\Theta = 0; \tag{2.28}$$

by symmetry.

$$\frac{\partial \Psi}{\partial Y} = 0; \tag{2.29}$$

far from the plume the velocity is purely vertical.

$$\Theta = 1; \tag{2.30}$$

density is given.

Adapting Schichtings [7] solution for the linear isothermal Jet and look for a Blasius– Howarth [8] expressio'fsn of the

$$\Psi = X^{1/3} \sum_{i=0}^{\infty} X^{4i/3} f_i(\eta) \tag{2.31}$$

$$\Theta = \sum_{i=0}^{\infty} X^{4i/3} g_i(\eta) \tag{2.32}$$

where the similarity variable is:

$$\eta = \frac{Y}{X^{2/3}} \tag{2.33}$$

the fundamental terms of these expansion are

$$f_0 = 6\alpha \tanh \alpha\eta \tag{2.34}$$

$$g_0 = \frac{\int_0^\eta \left[\cosh^{2P_r} \alpha\eta\right]^{-1} d\eta}{\int_0^\infty \left[\cosh^{2P_r} \alpha\eta\right]^{-1} d\eta} \tag{2.35}$$

where α is related to the momentum flux by

$$M = 2\rho \int_0^\infty U^2 dY = 48\rho\alpha^3 \tag{2.36}$$

For the practical case of disposal of urban sewage in the sea water, the density difference is essentially due to the concentration difference in sodium chloride. The interesting pollutants might be present in minute concentrations and would then diffuse through this plume, but without disturbing its density or its velocity field, if concentration of such a pollutant is C, a solution of the following form exists

$$C = X^{-1/3} \sum_{i=0}^{\infty} X^{45/3} h_i(\eta) \tag{2.37}$$

satisfying

$$\frac{\partial C}{\partial X}\frac{\partial \Psi}{\partial Y} - \frac{\partial C}{\partial Y}\frac{\partial \Psi}{\partial X} = \frac{1}{S_c}\frac{\partial^2 C}{\partial Y^2} \tag{2.38}$$

$$Y = 0, \ \frac{\partial C}{\partial Y} = 0 \tag{2.39}$$

by symmetry.

$$Y = \infty, \ C = 0 \tag{2.40}$$

the dilution far from the plume is complete.

It is easy to show that the fundamental term in (2.37) is

$$h_0 = \frac{h_0(0)}{\cosh^{2S_c} \alpha\eta} \tag{2.41}$$

2.1.6 OPTIMIZATION TECHNIQUE

Boundary value problems can be solved using an optimization scheme. The expression

$$f = \alpha_1 \left(\int_{-\infty}^{\infty} f' g d\eta - 1 \right)^2 + \alpha_2 f'^2(\infty) \tag{2.42}$$

for instance is a suitable objective function for the solution of (2.13), (2.14), (2.16), (2.17). For the case of analog. Integration in a hybrid configuration, machine noise disturbes the correct evaluation of the criterion function if the partial derivative cannot be determined analytically, the numerical evaluation of the derivative is jeopardized by noise. A good and fast direct search technique is preferable. The method chosen here is modified rotating coordinate technique. The algorithm has been provided for an efficient line search for determining the minimum point in a given direction.

Line Search

The line search is a combination of direct search and curve fitting in such a way that under fairly general conditions, convergence to the minimum is guaranteed [9].

Let $\underline{X}_k$ be the present point, $\underline{d}_k$ the direction of the search and α_k a given step. Following function evaluation are done:

$$f(\underline{X}_k + \alpha_k \underline{d}_k), \quad f(\underline{X}_k + 2\alpha_k \underline{d}_k), \quad f(\underline{X}_k + 4\alpha_k \underline{d}_k)$$

till three points $\underline{X}_1 = \underline{X}_k + \alpha_1 \underline{d}_k$
$\underline{X}_2 = \underline{X}_k + \alpha_2 \underline{d}_k$
$\underline{X}_3 = \underline{X}_k + \alpha_3 \underline{d}_k$
are obtained which satisfy the condition

$$f(\underline{X}_1) > f(\underline{X}_2) < f(\underline{X}_3)$$

If the function $f(\underline{X})$ is strictly unimodal in the given direction the coordinate α_m of the minimum point $\underline{X}_1 + \alpha_m \underline{d}_k$ will be the interval (α_1, α_2). Then a curve fitting procedure is started which does not require derivatives.

A quadratic

$$q(\alpha) = \sum_{i=1}^{3} f(\underline{X}_i) \frac{\Pi_{j \neq i}(\alpha - \alpha_j)}{\Pi_{j \neq i}(\alpha_i - \alpha_j)} \tag{2.43}$$

is passed through the three points and the coordinate of the extremum

$$\alpha_e = n \frac{1}{2} \frac{(\alpha_1^2 - \alpha_3^2) f(\underline{X}_1) + (\alpha_3^2 - \alpha_1^2) f(\underline{X}_2) + (\alpha_1^2 - \alpha_2^2) f(\underline{X}_3)}{(\alpha_1 - \alpha_3) f(\underline{X}_1) + (\alpha_3 - \alpha_1) f(\underline{X}_2) + (\alpha_1 - \alpha_2) f(\underline{X}_3)} \tag{2.44}$$

is warranted to be a minimum and contained in the interval (α_1, α_3); $f(\underline{X}_k + \alpha_e \underline{d}_k)$ is evaluated. If $\alpha_e < \alpha_2$ a new point $\underline{X}_1 = \underline{X}_k + \alpha_e \underline{d}_k$ is introduced reducing (α_1, α_2) to (α_e, α_3).

If $\alpha_e > \alpha_3$, $\underline{X}_3 = \underline{X}_k + \alpha_m \underline{d}_k$ is calculated and (α_1, α_3) reduce to (α_1, α_e). A new quadratic fit is performed on the reduced interval . If $\alpha_1 = \alpha_2$, the interval (α_2, α_i) - α_i is the coordinate of $\underline{X}_i$ being the argument of $f_i = \max\{f(\underline{X}_1), f(\underline{X}_2)\}$ - is divided to obtain a new point $\underline{X}_n$ in such a way that the new interval is smaller than the preceeding one. It can be proved by the Global convergence theorem [9] that this algorithm converges to the solution if the objective function is continuous and unimodal in α. The order of convergence is known to be about 1.3 [9]. in practice the search procedure has to be terminated before it has converged. For these problems α_m is determined to within a fixed percentage of its true value. A constant c, $0 < c < 1$ is selected ($c = 0.01$) and α is found so as to satisfy $|\alpha - \bar{\alpha}| \leq c|\bar{\alpha}|$ where $\bar{\alpha}$ is the lower bound α_1 on the true minimizing value of the parameter if α_1 is different from zero or equal to the termination value for the complete algorithm if α equals zero.

2.1.7 OPTIMIZATION ALGORITHM

In a simple coordinate descent method the coordinate directions $(\underline{e}_1, \underline{e}_2, \ldots, \underline{e}_n)$ are cyclically used to provide the directions for individual line searches. If the objective function has continuous partial derivatives this method is globally convergent [9], and the convergence rate is affected by relation of the coordinates. However if the first partial derivative are not continuous objective functions and the coordinate directions can be found so that the algorithm will not find the minimum. By rotating the coordinate system after n line searches an attempt is made to solve the problem

. If at the same time one axis is oriented towards the direction of the valley, locally estimated in a way analogous to the method used in the parallel tangent algorithm it has been found by some trial objective functions that the convergence rate is improved. An efficient method for obtaining a new orthonormal set is that of Powel [11], which requires $O(n^2)$ multiplications instead of $O(n^3)$.

The final algorithm is the following:

Given $\underline{X}_0$ and the current set of orthonormal set of orthogonal directions $D = (\underline{d}_1, \underline{d}_2, \ldots, \underline{d}_n)$ a set of β_j's are computed using n line searches.

$$\beta_j = \min_{\beta} f(\underline{X}_j, \beta \underline{d}_j)$$

with $\underline{X}_{j+1} = \underline{X}_j + \beta_j \underline{d}_j$ for $j = 1, 2, \ldots, n-1$.

The orders of the directions $\underline{d}_j$ is changed yielding $D' = (\underline{d}'_0, \underline{d}'_1, \ldots, \underline{d}'_{n-1})$ so that the first k directions have β – values different from zero $(\beta_0, \beta_1, \ldots, \beta_k, 0, 0, \ldots, 0)$. Then a new set of directions is computed.

1. set $j = k$

$\tau = (\beta_k)^2$

$\underline{\sigma} = \beta_k \underline{d}'_k$

2. if $j = 0$ terminate the process otherwise compute

$$d_j^n = \frac{(\tau \underline{d}'_{j-1} - \beta_{j-1}\underline{\sigma})}{[\tau(\tau + \beta_{j-1}^2)]^{1/2}} \tag{2.45}$$

3. set $j = j - 1$
$\tau = \tau + (\beta_j)^2$
$\underline{\sigma} = \underline{\sigma} + \beta_j \underline{d}'_j$ go to 2.

4. The remaining vectors are obtained as follows:

$$\underline{d}_0^n = \frac{\underline{\sigma}}{\sqrt{\tau}}; \quad \underline{\sigma} = \sum_{j=0}^{k} \beta_j \underline{d}'_j, \quad \tau = \sum_{j=0}^{k} (\beta_j)^2 \tag{2.46}$$

$\underline{d}_k^n = \underline{d}'_k$ for $j = k+1,\ k+2,\ \ldots,\ n-1$

We now have a new set $D^n = (\underline{d}_0^n,\ \underline{d}_1^n,\ \ldots,\ \underline{d}_{n-1}^n)$ to repeat the procedure.

To minimize the number of objective function evaluations a suitable step for the line search is necessary. If the step is too small; the initial value has to be doubled too many times. If the step is too large, too many curve fittings have to be performed. Therefore the step is adjusted during the optimization. For every coordinate relaxation (n line searches)

$$a = \frac{1}{n} \sum_{j=0}^{n-1} \beta_j$$

is computed. The series $\{a_k\}$ converges at least linearly for the quadratic case [9]. The convergence rate is dependent of the special objective function under study but experimentally it has been found that if a fraction of a (say a/8) is used as step for the next coordinate search an improvement in overall computation time is observed for the different objective function encountered in the problem.

Conclusion

The classical methods of boundary layer theory allows us to accurately model linear Laminar horizontal buoyant plumes. Using the modern developments of the theory (method of assymptotic expansion) we could even produce still better solutions of the non-linear problems considered. However, for any reasonable design the unit flow f_0 is likely to be so large that the flow would be turbulent rather than Laminar.

2.2 REVIEW OF THE EXTENDED COGGINS OPTIMIZATION TECHNIQUE

2.2.1 INTRODUCTION

Coggins algorithm as a one variable search method algorithm for obtaining the optimum value of an objective function with one variable [5]. It is not a rampantly used iterative procedure because of its limitations are being its restriction on one variable cost function. Even though it was developed solely to be used on objective function with a single variable, however, an attempt was made to [5] construct a more generalised algorithm based on the formulation of the coggins' one variable method.

The constrained optimization problem is

max (or min) z = F(x) where $X = (X^{(1)}, X^{(2)}, \ldots, X^{(n)})$

Here, unimodality is assumed while for a multimodal function multiple starting points should be used .

In the next section, consideration is made of an objective function with two variables and subsequently generalised for n variables.

2.2.2 THE ALGORITHM

The algorithm to find the optimum value of a function with two variables is listed in the steps below.

Step (1)

The objective function is evaluated using the initial value $X_0^{(1)}$, $X_0^{(2)}$.

Step (2)

The values of $X^{(1)}$ and $X^{(2)}$ are incremented

$$X^{(1)} = X^{(1)} + \Delta X^{(1)} \tag{2.47(i)}$$

$$X^{(2)} = X^{(2)} + \Delta X^{(2)} \tag{2.47(ii)}$$

The new value of $X^{(1)}$ and $X^{(2)}$ are used to evaluate the function. If there is function improvement then

$$\Delta X^{(1)} = 2 \star \Delta X^{(1)}, \;\; \Delta X^{(2)} = 2 \star \Delta X^{(2)} \tag{2.48}$$

else

$$\Delta X^{(1)} = -\Delta X^{(1)}, \;\; \Delta X^{(2)} = -\Delta X^{(2)}$$

Step (3)

After the first step, if there is function improvement then

$$\Delta X^{(1)} = 2 \star \Delta X^{(1)}, \;\; \Delta X^{(2)} = 2 \star \Delta X^{(2)} \tag{2.49}$$

else

$$\Delta X^{(1)} = \frac{\Delta X^{(1)}}{2}, \quad \Delta X^{(2)} = \frac{\Delta X^{(2)}}{2}$$

Step (4)

When a local optimum is obtained

$$\left((X_k^{(1)}, X_k^{(2)}), (X_{k-1}^{(1)}, X_{k-1}^{(2)}), (X_{k-2}^{(1)}, X_{k-2}^{(2)})\right)$$

Straddling the optimum. Then the additional point $X_{k+1}^{(1)}, \ X_{k+1}^{(2)}$ is located

$$X_{k+1}^{(1)} = X_{k-1}^{(1)} + \frac{\Delta X^{(1)}}{2} \tag{2.50(i)}$$

$$X_{k+1}^{(2)} = X_{k-1}^{(2)} + \frac{\Delta X^{(2)}}{2} \tag{2.50(i)}$$

The best three points

$$\left((X_1^{(1)}, X_1^{(2)}), (X_2^{(1)}, X_2^{(2)}), (X_3^{(1)}, X_3^{(2)})\right)$$

are obtained

Step (5)

A quadratic equation, f, is then curve fitted to the three retained points, the optimum location $X^{(*1)}, \ X^{(*2)}$ is located by setting dF = 0.

$$dF = \frac{\partial F}{\partial X^{(1)}} dX^{(1)} + \frac{\partial F}{\partial X^{(2)}} dX^{(2)} = 0 \tag{2.51}$$

$$\frac{\partial F}{\partial X^{(1)}} = 0 \text{ and } \frac{\partial F}{\partial X^{(2)}} = 0 \tag{2.52}$$

$$X^{\star(1)} = \frac{1}{2}\left\{\left(X_2^{2(1)} - X_3^{2(1)}\right) F\left(X_1^{(1)}, X_1^{(2)}\right) + \left(X_3^{2(1)} - X_1^{2(1)}\right) F\left(X_2^{(1)}, X_2^{(2)}\right)\right.$$
$$\left.+ \left(X_1^{2(1)} - X_2^{2(1)}\right) F\left(X_3^{(1)}, X_3^{(2)}\right)\right\} / \left\{\left(X_2^{(1)} - X_3^{(1)}\right) F\left(X_1^{(1)}, X_1^{(2)}\right) +\right.$$
$$\left.\left(X_3^{(1)} - X_1^{(1)}\right) F\left(X_2^{(1)}, X_2^{(2)}\right) + \left(X_1^{(1)} - X_2^{(1)}\right) F\left(X_3^{(1)}, X_3^{(2)}\right)\right\} \tag{2.53}$$

$$X^{\star(2)} = \frac{1}{2}\left\{\left(X_2^{2(2)} - X_3^{2(2)}\right) F\left(X_1^{(1)}, X_1^{(2)}\right) + \left(X_3^{2(2)} - X_1^{2(2)}\right) F\left(X_2^{(1)}, X_2^{(2)}\right) +\right.$$
$$\left.\left(X_1^{2(2)} - X_2^{2(2)}\right) F\left(X_3^{(1)}, X_3^{(2)}\right)\right\} / \left\{\left(X_2^{(2)} - X_3^{(2)}\right) F\left(X_1^{(1)}, X_1^{(2)}\right) +\right.$$
$$\left.\left(X_3^{(2)} - X_1^{(2)}\right) F\left(X_2^{(1)}, X_2^{(2)}\right) + \left(X_1^{(2)} - X_2^{(2)}\right) F\left(X_3^{(1)}, X_3^{(2)}\right)\right\} \tag{2.54}$$

Step (6)

The value of the objective function at $X^{(1)} = X^{\star(1)}$ and $X^{(2)} = X^{\star(2)}$ is compared with the best previous point subject to a convergence limit.

$$|X^{\star(1)} - X_j^{(1)}(\text{best})| \leq \text{ limit} \tag{2.55(i)}$$

$$|X^{\star(2)} - X_j^{(2)}(\text{best})| \leq \text{ limit} \tag{2.55(ii)}$$

If the inequalities (2.55) are satisfied, the procedure stops else the worst points are replaced by $X^{\star(1)}$, $X^{\star(2)}$, and a new quadratic surface is fitted and local optimum obtained.

This continues until equations (2.55) are satisfied.

2.3 COMPARISON OF OPTIMIZATION TECHNIQUES

Coggins method and Spriet–Baron optimization technique falls under the classification of non–gradient based methods of solving optimization problems. These methods are generally termed Direct search methods.

2.3.1 DIRECT SEARCH METHODS

The direct search strategies for generating a sequence of improving approximations to the solution are based simply on comparison of function values and generally, though not always, methods are heuristic in nature, having little or no mathematical basis. By their nature they make only very limited assumptions about the function, and generally no more than continuity so as a result they have a very wide field of application. Thus not only can they be used in problems for which differentiation is difficult,

but also for those cases where it may be appropriate, derivatives are discontinuous, or when the function values are subject to errors. These are situation in which gradient based methods can prove ineffective or inefficient. Most of the direct search methods are little affected by such difficulties, and because of their lack of assumptions about the function they can prove more reliable and stable than the gradient based methods.

2.3.2 COGGINS/SPRIET–BARON OPTIMIZATION TECHNIQUE

Coggins method is used to solve an unconstrained optimization problem that employs a direct search technique. Similarly, Coggins algorithm as a one – variable search method is is an algorithm for obtaining optimum value of an objective function with one variable [5].Even though it was developed solely to be used on objective function with a single variable, however, Sasindro and Reju [5] have generalised the algorithmto that of multi–variable based on the formalism of the one variable method.

The unconstrained optimization problem is as follows:

Maximize (or Minimize) Z = F(X) where $X = (X^{(1)},\ X^{(2)},\ \ldots,\ X^{(n)})$. Here Unimodality is assumed.

In the Spriet–Baron model, the expression

$$f = \alpha_1 \left(\int_{-\infty}^{\infty} f' g d\eta - 1 \right)^2 + \alpha_2 f'^2(\infty) \tag{2.56}$$

is a suitable objective function for the solution of (2.13), (2.14), (2.15), (2.16), (2.17). The method chosen by Spriet–Baron is a modified rotating coordinate technique. The algorithm has been provided of an efficient line search for determining the minimum point for a given direction.

The line search employed by Spriet–Baron is a combination of direct search and curve fitting in such a way that under fairly general conditions, convergence to the minimum is guaranteed (see 2.1.6)

2.3.3 COGGINS/SPRIET–BARON OPTIMIZATION ALGORITHM

Step 1

For Coggins: the objective function is evaluated using the initial value $\underline{X}_0^{(1)}$, $\underline{X}_0^{(2)}$.

That of Spriet–Baron: Given $\underline{X}_0$ and the current set of orthogonal directions

$$D = (\underline{d}_0,\ \underline{d}_1,\ \ldots,\ \underline{d}_{n-1})$$

a set of β_j's are computed using n line searches. $\beta_j = \min_\beta f(\underline{X}_j,\ \beta\underline{d}_j)$ with $\underline{X}_{j+1} = \underline{X}_j + \beta_j\underline{d}_j$ for $j = 0,\ 1,\ \ldots,\ n-1$.

The order of the directions $\underline{d}_j$ is changed yielding

$$D' = (\underline{d}'_0,\ \underline{d}'_1,\ \ldots,\ \underline{d}'_{n-1})$$

so that the first k directions have β – values different from zero $(\beta_0,\ \beta_1,\ \ldots,\ \beta_k,\ 0,\ 0,\ \ldots,\ 0)$.

Step 2

For Coggins, the values of $X^{(1)}$ and $X^{(2)}$ are incremented

$$X^{(1)} = X^{(1)} + \Delta X^{(1)} \qquad (2.57(i))$$

$$X^{(2)} = X^{(2)} + \Delta X^{(2)} \qquad (2.57(i))$$

But that of Spriet–Baron, a new set of directions is computed: set

$$j = k,\quad \tau = (\beta_k)^2,\quad \underline{\delta} = \beta_k\underline{d}_k^2 \qquad (2.58)$$

The new value of $X^{(1)}$, $X^{(2)}$ in (2.57) are used to evaluate the function. if there is function improvement then

$$\Delta X^{(1)} = 2 \star \Delta X^{(1)},\quad \Delta X^{(2)} = 2 \star \Delta X^{(2)} \qquad (2.59)$$

else

$$\Delta X^{(1)} = -\Delta X^{(1)},\quad \Delta X^{(2)} = -\Delta X^{(2)}$$

but for (2.59):

if $j = 0$ terminate the process, otherwise compute

$$d_j^n = \frac{(\tau \underline{d}'_{j-1} - \beta_{j-1}\underline{\delta})}{[\tau(\tau + \beta_{j-1}^2)]^{1/2}} \tag{2.60}$$

Step 3

After the first step in (2.3.2) if there is function improvement then

$$\Delta X^{(1)} = 2 \star \Delta X^{(1)}, \quad \Delta X^{(2)} = 2 \star \Delta X^{(2)} \tag{2.61}$$

else

$$\Delta X^{(1)} = \frac{\Delta X^{(1)}}{2}, \quad \Delta X^{(2)} = \frac{\Delta X^{(2)}}{2}$$

However after computing (2.60) for Spriet–Baron, set

$$j = j - 1, \quad \tau = \tau + (\beta_j)^2, \quad \underline{\delta} = \underline{\delta} + \beta_j \underline{d}'_j \tag{2.62}$$

Step 4

For Coggins, when a local optimum is obtained

$$\left((X_k^{(1)}, X_k^{(2)}), (X_{k-1}^{(1)}, X_{k-1}^{(2)}), (X_{k-2}^{(1)}, X_{k-2}^{(2)})\right)$$

straddling the optimum. Then an additional point $X_{k+1}^{(1)}, X_{k+1}^{(2)}$ is located.

$$X_{k+1}^{(1)} = X_{k-1}^{(1)} + \frac{\Delta X^{(1)}}{2}, \quad X_{k+1}^{(2)} = X_{k-1}^{(2)} + \frac{\Delta X^{(2}}{2} \tag{2.64}$$

The best three points

$$\left((X_1^{(1)}. X_1^{(2)}), (X_2^{(1)}, X_2^{(2)}), (X_3^{(1)}, X_3^{(2)})\right)$$

are obtained.

In the case of Spriet–Baron, the remaining vectors are obtained as follow:

$$\underline{d}_0^n = \frac{\underline{\delta}}{\sqrt{\tau}}; \quad \underline{\delta} = \sum_{j=0}^{k} \beta_j \underline{d}_j'; \quad \tau = \sum_{j=0}^{k} (\beta_j)^2 \tag{2.65}$$

$\underline{d}_k^n = \underline{d}_k'$ for $j = k+1,\ k+2,\ \ldots,\ n-1$

We now have a new set

$$D^n = (\underline{d}_0^n,\ \underline{d}_1^n,\ \ldots,\ \underline{d}_{n-1}^n)$$

to repeat the procedure.

Continue the procedure until the best 3 points are located, see 2.1.6.

Step 5

For Coggins, a quadratic equation f is then curve fitted to the three retained points. The optimum location $X^{*(1)},\ X^{*(2)}$ is located by setting dF = 0.

$$dF = \frac{\partial F}{\partial X^{(1)}} dX^{(1)} + \frac{\partial F}{\partial X^{(2)}} dX^{(2)} = 0 \tag{2.66}$$

$$\frac{\partial F}{\partial X^{(1)}} = 0, \quad \frac{\partial F}{\partial X^{(2)}} = 0 \tag{2.67}$$

$$X^{*(1)} = \frac{1}{2} \Big\{ \left(X_2^{2(1)} - X_3^{2(1)}\right) F\left(X_1^{(1)},\ X_1^{(2)}\right) + \left(X_3^{2(1)} - X_1^{2(1)}\right) F\left(X_2^{(1)},\ X_2^{(2)}\right) + \left(X_1^{2(1)} - X_2^{2(1)}\right) F\left(X_3^{(1)},\ X_3^{(2)}\right) \Big\} / \Big\{ \left(X_2^{(1)} - X_3^{(1)}\right) F\left(X_1^{(1)},\ X_1^{(2)}\right) + \left(X_3^{(1)} - X_1^{(1)}\right) F\left(X_2^{(1)},\ X_2^{(2)}\right) + \left(X_1^{(1)} - X_2^{(1)}\right) F\left(X_3^{(1)},\ X_3^{(2)}\right) \Big\} \tag{2.68}$$

$$X^{*(2)} = \frac{1}{2}\left\{\left(X_2^{2(2)} - X_3^{2(2)}\right) F\left(X_1^{(1)}, X_1^{(2)}\right) + \left(X_3^{2(2)} - X_1^{2(2)}\right) F\left(X_2^{(1)}, X_2^{(2)}\right) + \left(X_1^{2(2)} - X_2^{2(2)}\right) F\left(X_3^{(1)}, X_3^{(2)}\right)\right\} / \left\{\left(X_2^{(2)} - X_3^{(2)}\right) F\left(X_1^{(1)}, X_1^{(2)}\right) + \left(X_3^{(2)} - X_1^{(2)}\right) F\left(X_2^{(1)}, X_2^{(2)}\right) + \left(X_1^{(2)} - X_2^{(2)}\right) F\left(X_3^{(1)}, X_3^{(2)}\right)\right\} \tag{2.69}$$

However, for the Spriet–Baron model, if the function f(x) is strictly unimodal in the given direction the coordinate α_m of the minimum point $\underline{\alpha}_1 + \alpha_m \underline{d}_k$ will be in the interval $\alpha_1,\ \alpha_3$. Then a curve fitting procedure is started which does not require derivatives.

A quadratic

$$q(\alpha) = \sum_{i=0}^{2} f(x) \frac{\Pi_{j \neq i}(\alpha - \alpha_j)}{\Pi_{j \neq i}(\alpha_i - \alpha_j)} \tag{2.70}$$

is passed through the three points and the coordinate of the extremum.

$$\alpha_e = \frac{1}{2}\left[\frac{(\alpha_2^2 - \alpha_3^2)F(\underline{X}_1) + (\alpha_3^2 - \alpha_1^2)F(\underline{X}_2) + (\alpha_1^2 - \alpha_2^2)F(\underline{X}_3)}{(\alpha_2 - \alpha_3)F(\underline{X}_1) + (\alpha_3 - \alpha_1)F(\underline{X}_2) + (\alpha_1 - \alpha_2)F(\underline{X}_3)}\right] \tag{2.71}$$

is warranted to be a minimum and contained in the interval $((\alpha_1,\ \alpha_3)$; $F(\underline{X}_k + \alpha_e \underline{d}_k)$ is evaluated.

Step 6

For Coggins, the value of the objective function at $X^{(1)} = X^{(*1)}$ and $X^{(2)} = X^{(*2)}$ is compared with the best previous point subject to a convergence limit

$$|X^{(*1)} - X_j^{(1)}(\text{best})| \leq \text{ limit}, \quad |X^{(*2)} - X_j^{(2)}(\text{best})| \leq \text{ limit} \tag{2.72}$$

If the inequality (2.72) is satisfied the procedure stops, else the worst points are replaced by $X^{(*1)}$, $X^{(*2)}$ and a new quadratic surface is fitted and local optimum obtained. This continues until (2.72) is satisfied.

At this level, for the Spriet–Baron, to minimize the number of objective function evaluations a suitable step for the line search is necessary. If the step is too small, the initial value has to be doubled too many times. If the step is too large, too many curve fittings have to be performed. Therefore the step is adjusted during the optimization.

For every coordinate relaxation (n line searches)

$$a = \frac{1}{n}\sum_{j=0}^{n-1}\beta_j$$

is computed.

The series $\{a_k\}$ converges at least linearly for the quadratic case.

2.3.4 REMARK

The Spriet–Baron model as outlined above and that of Coggins (extended) optimization algorithm when compared seem to be very similar. However, absolute resemblance in the methodology used is not guaranteed. But with little modification, the Coggins extended method can be used to solve the integral functional as used by the Spriet–Baron model as we shall examine latter.

CHAPTER THREE
SOLUTION OF THE SUBMERGED SEWAGE DISPERSION MODEL

3.1 STATEMENT/DERIVATION OF THE OBJECTIVE CRITERION

From equation (2.42), the expression given as:

$$f = \alpha_1 \left(\int_{-\infty}^{\infty} f' g d\eta - 1 \right)^2 + \alpha_2 f'^2(\infty)$$

according to Spriet–Baron [11] a suitable objective function for the solution of (2.13), (2.14), (2.16) and (2.17).

To simplify this expression (2.42) we adopt Schlictings [7] solution for the linear isothermal Jet, thus

$$f(\eta) = 6\alpha \tanh \alpha\eta \tag{3.1}$$

$$g(\eta) = \frac{\int_0^\eta \left[\cosh^{2P_r} \alpha\eta\right]^{-1} d\eta}{\int_0^\infty \left[\cosh^{2P_r} \alpha\eta\right]^{-1} d\eta} \tag{3.2}$$

where $\alpha = 0.099$; (see appendix 1c).

Differentiating (3.1) and taking the square of both sides gives

$$f'^2(\eta) = \left(6\alpha \operatorname{sech}^2 \alpha\eta\right)^2 = 36\alpha^2 \left(\operatorname{sech}(\alpha\eta)\right)^4$$

Integrating:

$$\int_{-\infty}^{\infty} f'(\eta) d\eta = 6\alpha \int_{-\infty}^{\infty} \operatorname{sech}^2 \alpha\eta d\eta = 6\alpha \tanh \alpha\eta \tag{3.4}$$

Also integrating (3.2) with

$$\int_0^\infty \left[\cosh^{2P_r} \alpha\eta\right]^{-1} d\eta = 1$$

yields

$$\int_{-\infty}^{\infty} g(\eta)d\eta = \text{sech}^{2P_r}\alpha\eta \tag{3.5}$$

Combining (3.4) and (3.5) gives:

$$\int_{-\infty}^{\infty} f'gd\eta = (6\alpha\tanh\alpha\eta)(\text{sech}^{2P_r}\alpha\eta)$$

squaring both sides yields:

$$\left(\int_{-\infty}^{\infty} f'gd\eta\right)^2 = \left[(6\alpha\tanh\alpha\eta)(\text{sech}^{2P_r})\right]^2$$

Expanding the expression (2.42) and substituting accordingly we get

$$f = \alpha_1\left(\int_{-\infty}^{\infty} f'gd\eta - 1\right)^2 + \alpha_2 f'^2(\infty)$$

$$= \alpha_1\left\{\left[(6\alpha\tanh\alpha\eta)(\text{sech}^{2P_r}\alpha\eta)\right]^2 - 2(6\alpha\tanh\alpha\eta)(\text{sech}^{2P_r}) + 1\right\}$$

$$+\alpha_2\left[36\alpha^2\left(\text{sech}^2\alpha\eta\right)^2\right]$$

$$= \alpha_1\left[36\alpha^2\tanh^2\alpha\eta\left(\text{sech}^2\alpha\eta\right)^{P_r} - 2\alpha\tanh\alpha\eta)\left(\text{sech}^2\alpha\eta\right)^{P_r} + 1\right]$$

$$+\alpha_2\left[36\alpha^2\left(\text{sech}^2\alpha\eta\right)^2\right] \tag{3.6}$$

Where
$\alpha_1 = \alpha_2 = 0.21(\times 10^{-3}|K)$
$P_r = 6.4748$
$\eta = YX^{-2/3}$
$\alpha = 0.099$

Note:

(i) η is the similarity variable since η is used in dimensionless analysis and we intend to consider f and g as only functions of η, we set $\eta = 0.1,\ 0.2,\ 0.3,\ \ldots$.

(ii) $\alpha_1 = \alpha_2$ is the thermal expansion coefficient whose value according to Howatson et al [10] is $0.21(\times 10^{-3}|K)$

Substituting these values in equation (3.6), gives:

$$f = 2.1\times10^{-4}\left[0.352836\tanh^2\alpha\eta\left[\left(\mathrm{sech}^2\alpha\eta\right)^2\right]^{P_r} - 1.188\tanh\alpha\eta\left(\mathrm{sech}^2\alpha\eta\right)^{P_r} + 1\right]$$

$$+2.1\times10^{-4}\left[0.352836\left(\mathrm{sech}^2\alpha\eta\right)^2\right]$$

$$= 7.4\times10^{-5}\tanh^2\alpha\eta\left[\left(\mathrm{sech}^2\alpha\eta\right)^2\right]^{P_r} - 2.5\times10^{-4}\tanh\alpha\eta\left(\mathrm{sech}^2\alpha\eta\right)^{P_r}$$

$$+2.1\times10^{-4} + 7.4\times10^{-5}\left(\mathrm{sech}^2\alpha\eta\right)^2$$

The simplified objective function is given as:

$$f(\eta) = 7.4\times10^{-5}\tanh^2\alpha\eta\left[(\mathrm{sech}\alpha\eta)^4\right]^{P_r} - 2.5\times10^{-4}\tanh\alpha\eta\left(\mathrm{sech}^2\alpha\eta\right)^{P_r}$$

$$+2.1\times10^{-4} + 7.4\times10^{-5}(\mathrm{sech}\alpha\eta)^4$$

The objective function can now be stated as:

Find

$$\eta = \begin{Bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_n \end{Bmatrix}$$

which minimizes

$$7.4\times10^{-5}\tanh^2\alpha\eta\left[(\mathrm{sech}\alpha\eta)^4\right]^{P_r} - 2.5\times10^{-4}\tanh\alpha\eta\left(\mathrm{sech}^2\alpha\eta\right)^{P_r}$$

$$+2.1\times10^{-4} + 7.4\times10^{-5}(\mathrm{sech}\alpha\eta)^4$$

3.2 ANALYTICAL SOLUTION OF THE OBJECTIVE FUNCTION

Since our intent is to adopt the line search method which is a combination of direct search and curve fitting to attain the minimum, we decided to use hypothetical values to solve the objective function (3.6) with a view to serve as a basis for further comparison with the optimization algorithm method of Spriet–Baron and Extended Coggins optimization algorithm.

Problem

Minimize

$$f(\eta) = 7.4\times10^{-5}\tanh^2\alpha\eta\left[(\mathrm{sech}\alpha\eta)^4\right]^{P_r} - 2.5\times10^{-4}\tanh\alpha\eta\left(\mathrm{sech}^2\alpha\eta\right)^{P_r} + 2.1\times10^{-4} + 7.4\times10^{-5}(\mathrm{sech}\alpha\eta)^4 \qquad (3.7)$$

Solution

$\alpha = 0.099,\ P_r = 6.4748,\ \eta = 0.1,\ 0.2,\ 0.3, \ldots$

Iteration 1

$$f(0.1) = 7.4\times10^{-5}[\tanh(0.0099)]^2\left[(\mathrm{sech}(0.0099))^4\right]^{P_r} - 2.5\times10^{-4}\tanh(0.0099)\left[(\mathrm{sech}(0.0099))^2\right]^{P_r} + 2.1\times10^{-4} + 7.4\times10^{-5}(\mathrm{sech}(0.0099))^4$$

$$= 0.000281519390149 = 2.81519390149\times10^{-4}$$

Iteration 2

$$f(0.2) = 7.4\times10^{-5}[\tanh(0.0198)]^2\left[(\mathrm{sech}(0.0198))^4\right]^{P_r}$$

$$-2.5 \times 10^{-4} \tanh(0.0198) \left[(\text{sech}(0.0198))^2\right]^{P_r}$$

$$+2.1 \times 10^{-4} + 7.4 \times 10^{-5} (\text{sech}(0.0198))^4$$

$$= 0.000279034054484 = 2.79034054484 \times 10^{-4}$$

Subsequent iteration using math cad code shows the result as outlined in table 4.1.

3.3 SOLUTION OF THE OBJECTIVE FUNCTION USING SPRIET–BARON OPTIMIZATION ALGORITHM

The objective function to be minimized is:

$$f(\eta) = 7.4\times10^{-5} \tanh^2 \alpha\eta \left[(\text{sech}\alpha\eta)^4\right]^{P_r} - 2.5\times10^{-4} \tanh \alpha\eta \left(\text{sech}^2 \alpha\eta\right)^{P_r}$$

$$+2.1 \times 10^{-4} + 7.4 \times 10^{-5} (\text{sech}\alpha\eta)^4$$

The algorithm is as follow:

Let X_k be the present point.
d_k the direction of search
α_k a given step

We shall evaluate:

$$f(X_k + \alpha_k d_k), \quad f(X_k + 2\alpha_k d_k), \quad f(X_k + 4\alpha_k d_k) \ldots$$

We define:
$X_k = (0, -1)$
$d_k = (1, 2)$
$\alpha_k = 0.1$
then,
$\eta_1 = X_k^{(1)} + n\alpha_k d_k^{(1)} = 0 + 1(0.1)1 = 0.1$
Similarly
$\eta_2 = X_k^{(2)} + n\alpha_k d_k^{(2)} = (-1) + 1(0.1)2 = -0.8$

Re–writing the objective function:

$$f(\eta_1,\, \eta_2) = f_{nm21} = 7.4 \times 10^{-5} \tanh^2 \alpha\eta_1 \left[(\mathrm{sech}\alpha\eta_2)^4\right]^{P_r}$$
$$-2.5 \times 10^{-4} \tanh \alpha\eta_1 \left(\mathrm{sech}^2\alpha\eta_2\right)^{P_r}$$
$$+2.1 \times 10^{-4} + 7.4 \times 10^{-5}\,(\mathrm{sech}\alpha\eta_2)^4 \tag{3.8}$$

Iteration 1

$X_k = (0,\, -1)$, $\alpha_k = 0.1$ $d_k = (1,\, 2)$, $\alpha = 0.099$, $P_r = 6.4748$, $n = 1$
$\eta_1 = X_k^{(1)} + n\alpha_k d_k^{(1)} = 0 + (1)(0.1)(1) = 0.1$ $\eta_2 = X_k^{(2)} + n\alpha_k d_k^{(2)} = -1 + (1)(0.1)(2) = -0.8$

$$f_{nm21} = 7.4 \times 10^{-5} \tanh^2((0.099)(0.1)) \left[(\mathrm{sech}((0.099)(-0.8)))^4\right]^{6.4748}$$
$$-2.5 \times 10^{-4} \tanh((0.099)(0.1)) \left(\mathrm{sech}^2((0.099)(-0.8))\right)^{6.4748}$$
$$+2.1 \times 10^{-4} + 7.4 \times 10^{-5}\,(\mathrm{sech}((0.099)(-0.8)))^4$$
$$= 7.4 \times 10^{-5} \tanh^2(0.0099) \left[(\mathrm{sech}(-0.0792))^4\right]^{6.4748}$$
$$-2.5 \times 10^{-4} \tanh(0.0099) \left(\mathrm{sech}^2(-0.0792)\right)^{6.4748}$$
$$+2.1 \times 10^{-4} + 7.4 \times 10^{-5}\,(\mathrm{sech}(-0.0792))^4$$
$$= 0.000280708574959 = 2.80708574959 \times 10^{-4}$$

Iteration 2

$X_k = (0,\, -1)$, $\alpha_k = 0.1$ $d_k = (1,\, 2)$, $\alpha = 0.099$, $P_r = 6.4748$, $n = 2$
$\eta_1 = X_k^{(1)} + n\alpha_k d_k^{(1)} = 0 + (2)(0.1)(1) = 0.2$ $\eta_2 = X_k^{(2)} + n\alpha_k d_k^{(2)} = -1 + (2)(0.1)(2) = -0.6$

$$f_{nm22} = 7.4 \times 10^{-5} \tanh^2((0.099)(0.2)) \left[(\mathrm{sech}((0.099)(-0.6)))^4\right]^{6.4748}$$
$$-2.5 \times 10^{-4} \tanh((0.099)(0.2)) \left(\mathrm{sech}^2((0.099)(-0.6))\right)^{6.4748}$$
$$+2.1 \times 10^{-4} + 7.4 \times 10^{-5}\,(\mathrm{sech}((0.099)(-0.6)))^4$$
$$= 7.4 \times 10^{-5} \tanh^2(0.0198) \left[(\mathrm{sech}(-0.0594))^4\right]^{6.4748}$$

$$-2.5 \times 10^{-4} \tanh(0.0198) \left(\mathrm{sech}^2(-0.0594)\right)^{6.4748}$$

$$+2.1 \times 10^{-4} + 7.4 \times 10^{-5} (\mathrm{sech}(-0.0594))^4$$

$$= 0.00027860024412 = 2.7860024412 \times 10^{-4}$$

Iteration 3

$X_k = (0,\ -1)$, $\alpha_k = 0.1$ $d_k = (1,\ 2)$, $\alpha = 0.099$, $P_r = 6.4748$, $n = 4$
$\eta_1 = X_k^{(1)} + n\alpha_k d_k^{(1)} = 0 + (4)(0.1)(1) = 0.4$ $\eta_2 = X_k^{(2)} + n\alpha_k d_k^{(2)} = -1 + (4)(0.1)(2) = -0.2$

$$f_{nm22} = 7.4 \times 10^{-5} \tanh^2((0.099)(0.4)) \left[(\mathrm{sech}((0.099)(-0.2)))^4\right]^{6.4748}$$

$$-2.5 \times 10^{-4} \tanh((0.099)(0.4)) \left(\mathrm{sech}^2((0.099)(-0.2))\right)^{6.4748}$$

$$+2.1 \times 10^{-4} + 7.4 \times 10^{-5} (\mathrm{sech}((0.099)(-0.2)))^4$$

$$= 7.4 \times 10^{-5} \tanh^2(0.0396) \left[(\mathrm{sech}(-0.0198))^4\right]^{6.4748}$$

$$-2.5 \times 10^{-4} \tanh(0.0396) \left(\mathrm{sech}^2(-0.0198)\right)^{6.4748}$$

$$+2.1 \times 10^{-4} + 7.4 \times 10^{-5} (\mathrm{sech}(-0.0198))^4$$

$$= 0.000274187595294 = 2.74187595294 \times 10^{-4}$$

As a result of the tedious nature of generating the values manually, we decided to use the aid of computer to generate the subsequent values. The math cad simulation procedure is as follows:
$n = 1,\ 2,\ 4,\ 6,\ 8,\ 10,\ \ldots$
$X_k = (0,\ -1)$, $d_k = (1,\ 2)$, $\alpha = 0.099$, $\alpha_k = 0.1$, $P_r = 6.4748$
$G = X_k^{(1)} + n\alpha_k d_k^{(1)} = 0 + n(0.1)1$
$A = X_k^{(2)} + n\alpha_k d_k^{(2)} = -1 + n(0.1)2$
which gives:

$$f(G,\ A) = f_{nm2} = 7.4 \times 10^{-5} \tanh^2(\alpha G) \left[(\mathrm{sech}(\alpha A))^4\right]^{P_r}$$

$$-2.5 \times 10^{-4} \tanh(\alpha G) \left(\mathrm{sech}^2(\alpha A)\right)^{P_r}$$

$$+2.1 \times 10^{-4} + 7.4 \times 10^{-5} (\mathrm{sech}(\alpha A))^4$$

3.4 SOLUTION OF THE OBJECTIVE FUNCTION USING THE EXTENDED COGGINS ALGORITHM

Minimize

$$f(\eta) = 7.4\times10^{-5}\tanh^2\alpha\eta\left[(\text{sech}\alpha\eta)^4\right]^{P_r} - 2.5\times10^{-4}\tanh\alpha\eta\left(\text{sech}^2\alpha\eta\right)^{P_r}$$

$$+2.1\times10^{-4} + 7.4\times10^{-5}(\text{sech}\alpha\eta)^4 \qquad (3.9)$$

The algorithm assumes the following: Let X_k be the present point
ΔP be a step length
where $X_k = (0,\ -1)$
$\Delta P = 0.1$
$P = 2^r,\ \ r = 0.\ 1,\ 2,\ 3,\ \ldots$

Iteration 1 (direct substitution)

$X_1 = 0,\ X_2 = -1,\ \alpha = 0.099,\ P_r = 6.4748$

$$\begin{aligned} f(X_1,\ X_2) = f_{nm13} &= 7.4\times10^{-5}\tanh^2(\alpha X_1)\left[(\text{sech}(\alpha X_2))^4\right]^{P_r} \\ &\quad -2.5\times10^{-4}\tanh(\alpha X_1)\left(\text{sech}^2(\alpha X_2)\right)^{P_r} \\ &\quad +2.1\times10^{-4} + 7.4\times10^{-5}(\text{sech}(\alpha X_2))^4 \\ &= 7.4\times10^{-5}\tanh^2(0)\left[(\text{sech}(-0.099))^4\right]^{6.4748} \\ &\quad -2.5\times10^{-4}\tanh(0)\left(\text{sech}^2(-0.099)\right)^{6.4748} \\ &\quad +2.1\times10^{-4} + 7.4\times10^{-5}(\text{sech}(-0.099))^4 \\ &= 0.000282565893840 = 2.8256589384\times10^{-4} \end{aligned}$$

Iteration 2

$X_1 = 0 + 0.1 = 0.1,\ X_2 = -1 + 0.1 = -0.9,\ \alpha = 0.099,\ P_r = 6.4748$

$$f_{nm23} = 7.4\times10^{-5}\tanh^2(0.0099)\left[(\text{sech}(-0.0891))^4\right]^{6.4748}$$

$$-2.5 \times 10^{-4} \tanh(0.0099) \left(\operatorname{sech}^2(-0.0891)\right)^{6.4748}$$

$$+2.1 \times 10^{-4} + 7.4 \times 10^{-5} \left(\operatorname{sech}(-0.0891)\right)^4$$

$$= 0.000280491329455 = 2.80491329455 \times 10^{-4}$$

Iteration 3

$X_1^{(3)} = 0 + 2(0.1) = 0.2$, $X_2^{(3)} = -1 + 2(0.1) = -0.8$, $\alpha = 0.099$, $P_r = 6.4748$

$$f_{nm33} = 7.4 \times 10^{-5} \tanh^2(0.0198) \left[(\operatorname{sech}(-0.0792))^4\right]^{6.4748}$$

$$-2.5 \times 10^{-4} \tanh(0.0198) \left(\operatorname{sech}^2(-0.0792)\right)^{6.4748}$$

$$+2.1 \times 10^{-4} + 7.4 \times 10^{-5} \left(\operatorname{sech}(-0.0792)\right)^4$$

$$= 0.0002783525794 59 = 2.78352579459 \times 10^{-4}$$

To hasten this process of iteration, the use of mathcad code is employed with the following assumptions:
Let $\Delta p = 0.1$, $p = 2^r$, $p = 0 + \Delta p \dot{p}$, $q = -1 + \Delta p \dot{p}$
with $\alpha = 0.099$, $P_r = 6.4748$
(see appendix 4)
Then

$$f(p, q) = 7.4 \times 10^{-5} \tanh^2(\alpha p) \left[(\operatorname{sech}(\alpha q))^4\right]^{P_r}$$

$$-2.5 \times 10^{-4} \tanh(\alpha p) \left(\operatorname{sech}^2(\alpha q)\right)^{P_r}$$

$$+2.1 \times 10^{-4} + 7.4 \times 10^{-5} \left(\operatorname{sech}(\alpha q)\right)^4$$

(see Table 4.3)

CHAPTER FOUR

COMPUTATIONAL/SIMULATION ANALYSIS FOR THE SUBMERGED SEWAGE DISPERSION MODEL

4.1 COMPUTATIONAL RESULTS USING ANALYTICAL SOLUTION

Using the values:
$\eta = 0.1,\ 0.2,\ \ldots,\ \alpha = 0.099$, and $P_r = 6.4748$ in (3.8) and using the mathcad code we obtain the following results for the various values of η as presented in tables 4.1, 4.1a, 4.1b. and 4.1d.

From the table 4.1 we obtain the graphical illustrations in figures 4.1.

Table 4.1: Computational results using analytical method

S/N	f(η)	S/N	f(η)	S/N	f(η)
1	0.000281519390149	62	0.000232882586313	123	0.000216409581041
2	0.000279034054484	63	0.000232846658131	124	0.000216205033558
3	0.000276553530118	64	0.000232791168895	125	0.000216005995912
4	0.000274087033998	65	0.000232716111712	126	0.000215812388078
5	0.000271643407108	66	0.000232621613087	127	0.000215624125804
6	0.000269231064864	67	0.000232507921874	128	0.000215441121162
7	0.000266857953922	68	0.000232375398149	129	0.000215263283056
8	0.000264531515546	69	0.000232224502104	130	0.000215090517692
9	0.000262258655509	70	0.000232055783073	131	0.000214922729017
10	0.000260045720427	71	0.000231869868777	132	0.000214759819129
11	0.000257898480296	72	0.000231667454867	133	0.000214601688646
12	0.000255822116950	73	0.000231449294830	134	0.000214448237058
13	0.000253821218052	74	0.000231216190329	135	0.000214299363038
14	0.000251899776210	75	0.000230968982018	136	0.000214154964739
15	0.000250061192766	76	0.000230708540866	137	0.000214014940060
16	0.000248308285766	77	0.000230435760033	138	0.000213879186888
17	0.000246643301653	78	0.000230151547310	139	0.000213747603320
18	0.000245067930193	79	0.000229856818130	140	0.000213620087866
19	0.000243583322198	80	0.000229552489175	141	0.000213496539630
20	0.000242190109605	81	0.000229239472550	142	0.000213376858474
21	0.000240888427543	82	0.000228918670546	143	0.000213260945166
22	0.000239677938016	83	0.000228590970956	144	0.000213148701513
23	0.000238557854912	84	0.000228257242939	145	0.000213040030478
24	0.000237526970061	85	0.000227918333411	146	0.000212934836281
25	0.000236583680117	86	0.000227575063930	147	0.000212833024496
26	0.000235726014074	87	0.000227228228066	148	0.000212734502130
27	0.000234951661255	88	0.000226878589211	149	0.000212639177689
28	0.000234257999637	89	0.000226526878810	150	0.000212546961241
29	0.000233642124415	90	0.000226173794978	151	0.000212457764465
30	0.000233100876688	91	0.000225820001477	152	0.000212371500697
31	0.000232630872189	92	0.000225466127021	153	0.000212288084960
32	0.000232228529988	93	0.000225112764876	154	0.000212207433995
33	0.000231890101055	94	0.000224760472734	155	0.000212129466280
34	0.000231611696630	95	0.000224409772822	156	0.000212054102047
35	0.000231389316276	96	0.000224061152227	157	0.000211981263294
36	0.000231218875538	97	0.000223715063411	158	0.000211910873787
37	0.000231096233079	98	0.000223371924885	159	0.000211842859062
38	0.000231017217186	99	0.000223032122024	160	0.000211777146422
39	0.000230977651510	100	0.000222696008002	161	0.000211713664933
40	0.000230973379917	101	0.000222363904826	162	0.000211652345408
41	0.000231000290310	102	0.000222036104447	163	0.000211593120400
42	0.000231054337283	103	0.000221712869938	164	0.000211535924185
43	0.000231131563485	104	0.000221394436715	165	0.000211480692742
44	0.000231228119559	105	0.000221081013800	166	0.000211427363735
45	0.000231340282542	106	0.000220772785094	167	0.000211375876494
46	0.000231464472625	107	0.000220469910671	168	0.000211326171984
47	0.000231597268179	108	0.000220172528068	169	0.000211278192788
48	0.000231735418982	109	0.000219880753562	170	0.000211231883079
49	0.000231875857584	110	0.000219594683443	171	0.000211187188592
50	0.000232015708796	111	0.000219314395252	172	0.000211144056595
51	0.000232152297266	112	0.000219039949000	173	0.000211102435863
52	0.000232283153184	113	0.000218771388350	174	0.000211062276648
53	0.000232406016121	114	0.000218508741767	175	0.000211023530651
54	0.000232518837066	115	0.000218252023623	176	0.000210986150987
55	0.000232619778731	116	0.000218001235267	177	0.000210950092161
56	0.000232707214209	117	0.000217756366044	178	0.000210915310034
57	0.000232779724086	118	0.000217517394282	179	0.000210881761791
58	0.000232836092129	119	0.000217284288217	180	0.000210849405916
59	0.000232875299664	120	0.000217057006885	181	0.000210818202155
60	0.000232896518792	121	0.000216835500967	182	0.000210788111491
61	0.000232899104571	122	0.000216619713582	183	0.000210759096110

S/N	f(η)		S/N	f(η)		S/N	f(η)
184	0.000210731119373		245	0.000210070192958		306	0.000210006407513
185	0.000210704145785		246	0.000210067508710		307	0.000210006159860
186	0.000210678140968		247	0.000210064926341		308	0.000210005921758
187	0.000210653071628		248	0.000210062442029		309	0.000210005692839
188	0.000210628905529		249	0.000210060052090		310	0.000210005472751
189	0.000210605611466		250	0.000210057752981		311	0.000210005261154
190	0.000210583159231		251	0.000210055541285		312	0.000210005057721
191	0.000210561519593		252	0.000210053413714		313	0.000210004862139
192	0.000210540664266		253	0.000210051367100		314	0.000210004674105
193	0.000210520565884		254	0.000210049398395		315	0.000210004493329
194	0.000210501197977		255	0.000210047504662		316	0.000210004319532
195	0.000210482534939		256	0.000210045683072		317	0.000210004152444
196	0.000210464552012		257	0.000210043930905		318	0.000210003991808
197	0.000210447225252		258	0.000210042245537		319	0.000210003837375
198	0.000210430531513		259	0.000210040624447		320	0.000210003688906
199	0.000210414448419		260	0.000210039065204		321	0.000210003546171
200	0.000210398954342		261	0.000210037565470		322	0.000210003408951
201	0.000210384028378		262	0.000210036122991		323	0.000210003277031
202	0.000210369650329		263	0.000210034735602		324	0.000210003150208
203	0.000210355800677		264	0.000210033401215		325	0.000210003028285
204	0.000210342460567		265	0.000210032117822		326	0.000210002911074
205	0.000210329611784		266	0.000210030883488		327	0.000210002798393
206	0.000210317236733		267	0.000210029696355		328	0.000210002690067
207	0.000210305318422		268	0.000210028554629		329	0.000210002585928
208	0.000210293840439		269	0.000210027456587		330	0.000210002485815
209	0.000210282786940		270	0.000210026400570		331	0.000210002389572
210	0.000210272142623		271	0.000210025384981		332	0.000210002297051
211	0.000210261892717		272	0.000210024408283		333	0.000210002208106
212	0.000210252022964		273	0.000210023468997		334	0.000210002122602
213	0.000210242519598		274	0.000210022565699		335	0.000210002040404
214	0.000210233369335		275	0.000210021697019		336	0.000210001961385
215	0.000210224559354		276	0.000210020861640		337	0.000210001885422
216	0.000210216077282		277	0.000210020058292		338	0.000210001812398
217	0.000210207911180		278	0.000210019285756		339	0.000210001742199
218	0.000210200049529		279	0.000210018542856		340	0.000210001674716
219	0.000210192481215		280	0.000210017828462		341	0.000210001609843
220	0.000210185195517		281	0.000210017141487		342	0.000210001547481
221	0.000210178182092		282	0.000210016480885		343	0.000210001487532
222	0.000210171430964		283	0.000210015845649		344	0.000210001429902
223	0.000210164932511		284	0.000210015234809		345	0.000210001374503
224	0.000210158677455		285	0.000210014647435		346	0.000210001321248
225	0.000210152656846		286	0.000210014082629		347	0.000210001270055
226	0.000210146862055		287	0.000210013539528		348	0.000210001220842
227	0.000210141284761		288	0.000210013017304		349	0.000210001173535
228	0.000210135916942		289	0.000210012515157		350	0.000210001128060
229	0.000210130750862		290	0.000210012032319		351	0.000210001084344
230	0.000210125779066		291	0.000210011568051		352	0.000210001042322
231	0.000210120994364		292	0.000210011121642		353	0.000210001001926
232	0.000210116389828		293	0.000210010692409		354	0.000210000963095
233	0.000210111958779		294	0.000210010279693		355	0.000210000925767
234	0.000210107694780		295	0.000210009882863		356	0.000210000889885
235	0.000210103591626		296	0.000210009501308		357	0.000210000855392
236	0.000210099643339		297	0.000210009134443		358	0.000210000822235
237	0.000210095844158		298	0.000210008781706		359	0.000210000790363
238	0.000210092188529		299	0.000210008442554		360	0.000210000759725
239	0.000210088671103		300	0.000210008116466		361	0.000210000730274
240	0.000210085286726		301	0.000210007802941		362	0.000210000701963
241	0.000210082030430		302	0.000210007501496		363	0.000210000674749
242	0.000210078897431		303	0.000210007211669		364	0.000210000648590
243	0.000210075883120		304	0.000210006933013		365	0.000210000623444
244	0.000210072983055		305	0.000210006665098		366	0.000210000599272

S/N	f(η)		S/N	f(η)		S/N	f(η)
367	0.000210000576037		428	0.000210000051549		489	0.000210000004607
368	0.000210000553702		429	0.000210000049548		490	0.000210000004428
369	0.000210000532232		430	0.000210000047625		491	0.000210000004256
370	0.000210000511595		431	0.000210000045777		492	0.000210000004091
371	0.000210000491757		432	0.000210000044000		493	0.000210000003932
372	0.000210000472688		433	0.000210000042292		494	0.000210000003779
373	0.000210000454357		434	0.000210000040651		495	0.000210000003633
374	0.000210000436738		435	0.000210000039073		496	0.000210000003492
375	0.000210000419801		436	0.000210000037557		497	0.000210000003356
376	0.000210000403520		437	0.000210000036099		498	0.000210000003226
377	0.000210000387871		438	0.000210000034698		499	0.000210000003100
378	0.000210000372828		439	0.000210000033351		500	0.000210000002980
379	0.000210000358369		440	0.000210000032057		501	0.000210000002864
380	0.000210000344469		441	0.000210000030812		502	0.000210000002753
381	0.000210000331109		442	0.000210000029616		503	0.000210000002646
382	0.000210000318266		443	0.000210000028467		504	0.000210000002544
383	0.000210000305922		444	0.000210000027362		505	0.000210000002445
384	0.000210000294056		445	0.000210000026300		506	0.000210000002350
385	0.000210000282650		446	0.000210000025279		507	0.000210000002259
386	0.000210000271686		447	0.000210000024298		508	0.000210000002171
387	0.000210000261147		448	0.000210000023355		509	0.000210000002087
388	0.000210000251017		449	0.000210000022448		510	0.000210000002006
389	0.000210000241280		450	0.000210000021577		511	0.000210000001928
390	0.000210000231920		451	0.000210000020739		512	0.000210000001853
391	0.000210000222923		452	0.000210000019934		513	0.000210000001781
392	0.000210000214275		453	0.000210000019161		514	0.000210000001712
393	0.000210000205963		454	0.000210000018417		515	0.000210000001645
394	0.000210000197972		455	0.000210000017702		516	0.000210000001582
395	0.000210000190292		456	0.000210000017015		517	0.000210000001520
396	0.000210000182909		457	0.000210000016354		518	0.000210000001461
397	0.000210000175813		458	0.000210000015720		519	0.000210000001404
398	0.000210000168992		459	0.000210000015109		520	0.000210000001350
399	0.000210000162436		460	0.000210000014523		521	0.000210000001297
400	0.000210000156133		461	0.000210000013959		522	0.000210000001247
401	0.000210000150076		462	0.000210000013417		523	0.000210000001199
402	0.000210000144253		463	0.000210000012896		524	0.000210000001152
403	0.000210000138656		464	0.000210000012396		525	0.000210000001107
404	0.000210000133276		465	0.000210000011915		526	0.000210000001064
405	0.000210000128105		466	0.000210000011452		527	0.000210000001023
406	0.000210000123134		467	0.000210000011008		528	0.000210000000983
407	0.000210000118356		468	0.000210000010580		529	0.000210000000945
408	0.000210000113764		469	0.000210000010170		530	0.000210000000909
409	0.000210000109349		470	0.000210000009775		531	0.000210000000873
410	0.000210000105106		471	0.000210000009395		532	0.000210000000839
411	0.000210000101028		472	0.000210000009031		533	0.000210000000807
412	0.000210000097107		473	0.000210000008680		534	0.000210000000775
413	0.000210000093339		474	0.000210000008343		535	0.000210000000745
414	0.000210000089717		475	0.000210000008019		536	0.000210000000716
415	0.000210000086236		476	0.000210000007708		537	0.000210000000689
416	0.000210000082889		477	0.000210000007409		538	0.000210000000662
417	0.000210000079673		478	0.000210000007121		539	0.000210000000636
418	0.000210000076581		479	0.000210000006845		540	0.000210000000611
419	0.000210000073609		480	0.000210000006579		541	0.000210000000588
420	0.000210000070752		481	0.000210000006323		542	0.000210000000565
421	0.000210000068007		482	0.000210000006078		543	0.000210000000543
422	0.000210000065367		483	0.000210000005842		544	0.000210000000522
423	0.000210000062831		484	0.000210000005615		545	0.000210000000502
424	0.000210000060392		485	0.000210000005397		546	0.000210000000482
425	0.000210000058048		486	0.000210000005188		547	0.000210000000463
426	0.000210000055796		487	0.000210000004986		548	0.000210000000445
427	0.000210000053630		488	0.000210000004793		549	0.000210000000428

S/N	$f(\eta)$		S/N	$f(\eta)$		S/N	$f(\eta)$
550	0.000210000000412		611	0.000210000000037		672	0.000210000000003
551	0.000210000000396		612	0.000210000000035		673	0.000210000000003
552	0.000210000000380		613	0.000210000000034		674	0.000210000000003
553	0.000210000000365		614	0.000210000000033		675	0.000210000000003
554	0.000210000000351		615	0.000210000000031		676	0.000210000000003
555	0.000210000000338		616	0.000210000000030		677	0.000210000000003
556	0.000210000000324		617	0.000210000000029		678	0.000210000000003
557	0.000210000000312		618	0.000210000000028		679	0.000210000000002
558	0.000210000000300		619	0.000210000000027		680	0.000210000000002
559	0.000210000000288		620	0.000210000000026		681	0.000210000000002
560	0.000210000000277		621	0.000210000000025		682	0.000210000000002
561	0.000210000000266		622	0.000210000000024		683	0.000210000000002
562	0.000210000000256		623	0.000210000000023		684	0.000210000000002
563	0.000210000000246		624	0.000210000000022		685	0.000210000000002
564	0.000210000000236		625	0.000210000000021		686	0.000210000000002
565	0.000210000000227		626	0.000210000000020		687	0.000210000000002
566	0.000210000000218		627	0.000210000000020		688	0.000210000000002
567	0.000210000000210		628	0.000210000000019		689	0.000210000000002
568	0.000210000000202		629	0.000210000000018		690	0.000210000000002
569	0.000210000000194		630	0.000210000000017		691	0.000210000000002
570	0.000210000000186		631	0.000210000000017		692	0.000210000000001
571	0.000210000000179		632	0.000210000000016		693	0.000210000000001
572	0.000210000000172		633	0.000210000000015		694	0.000210000000001
573	0.000210000000166		634	0.000210000000015		695	0.000210000000001
574	0.000210000000159		635	0.000210000000014		696	0.000210000000001
575	0.000210000000153		636	0.000210000000014		697	0.000210000000001
576	0.000210000000147		637	0.000210000000013		698	0.000210000000001
577	0.000210000000141		638	0.000210000000013		699	0.000210000000001
578	0.000210000000136		639	0.000210000000012		700	0.000210000000001
579	0.000210000000131		640	0.000210000000012		701	0.000210000000001
580	0.000210000000125		641	0.000210000000011		702	0.000210000000001
581	0.000210000000121		642	0.000210000000011		703	0.000210000000001
582	0.000210000000116		643	0.000210000000010		704	0.000210000000001
583	0.000210000000111		644	0.000210000000010		705	0.000210000000001
584	0.000210000000107		645	0.000210000000010		706	0.000210000000001
585	0.000210000000103		646	0.000210000000009		707	0.000210000000001
586	0.000210000000099		647	0.000210000000009		708	0.000210000000001
587	0.000210000000095		648	0.000210000000008		709	0.000210000000001
588	0.000210000000091		649	0.000210000000008		710	0.000210000000001
589	0.000210000000088		650	0.000210000000008		711	0.000210000000001
590	0.000210000000084		651	0.000210000000008		712	0.000210000000001
591	0.000210000000081		652	0.000210000000007		713	0.000210000000001
592	0.000210000000078		653	0.000210000000007		714	0.000210000000001
593	0.000210000000075		654	0.000210000000007		715	0.000210000000001
594	0.000210000000072		655	0.000210000000006		716	0.000210000000001
595	0.000210000000069		656	0.000210000000006		717	0.000210000000001
596	0.000210000000067		657	0.000210000000006		718	0.000210000000001
597	0.000210000000064		658	0.000210000000006		719	0.000210000000001
598	0.000210000000062		659	0.000210000000005		720	0.000210000000000
599	0.000210000000059		660	0.000210000000005		721	0.000210000000000
600	0.000210000000057		661	0.000210000000005		722	0.000210000000000
601	0.000210000000055		662	0.000210000000005			
602	0.000210000000052		663	0.000210000000005			
603	0.000210000000050		664	0.000210000000005			
604	0.000210000000048		665	0.000210000000004			
605	0.000210000000047		666	0.000210000000004			
606	0.000210000000045		667	0.000210000000004			
607	0.000210000000043		668	0.000210000000004			
608	0.000210000000041		669	0.000210000000004			
609	0.000210000000040		670	0.000210000000004			
610	0.000210000000038		671	0.000210000000003			

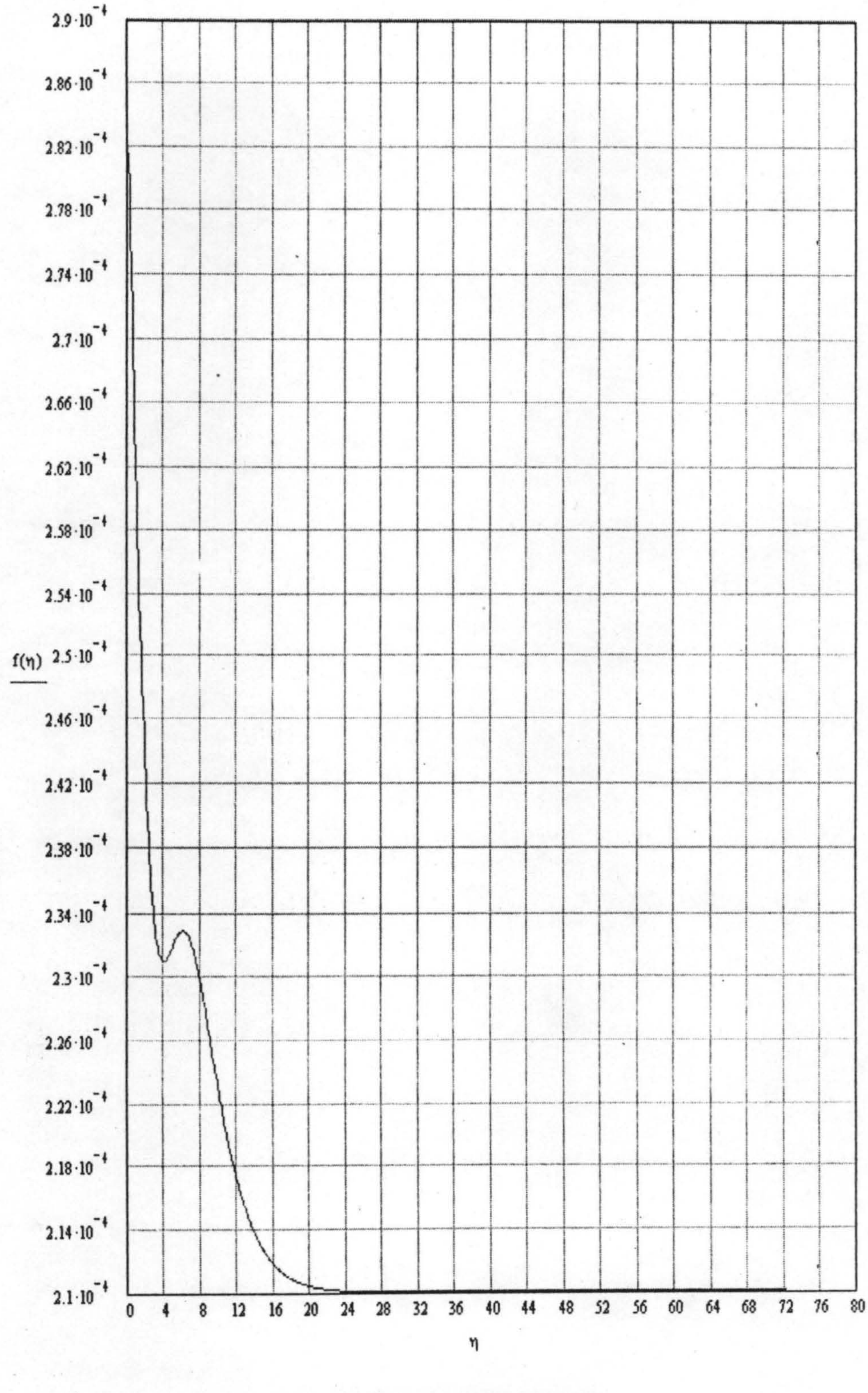

Fig 4.1a: Graphical illustration of analytic results

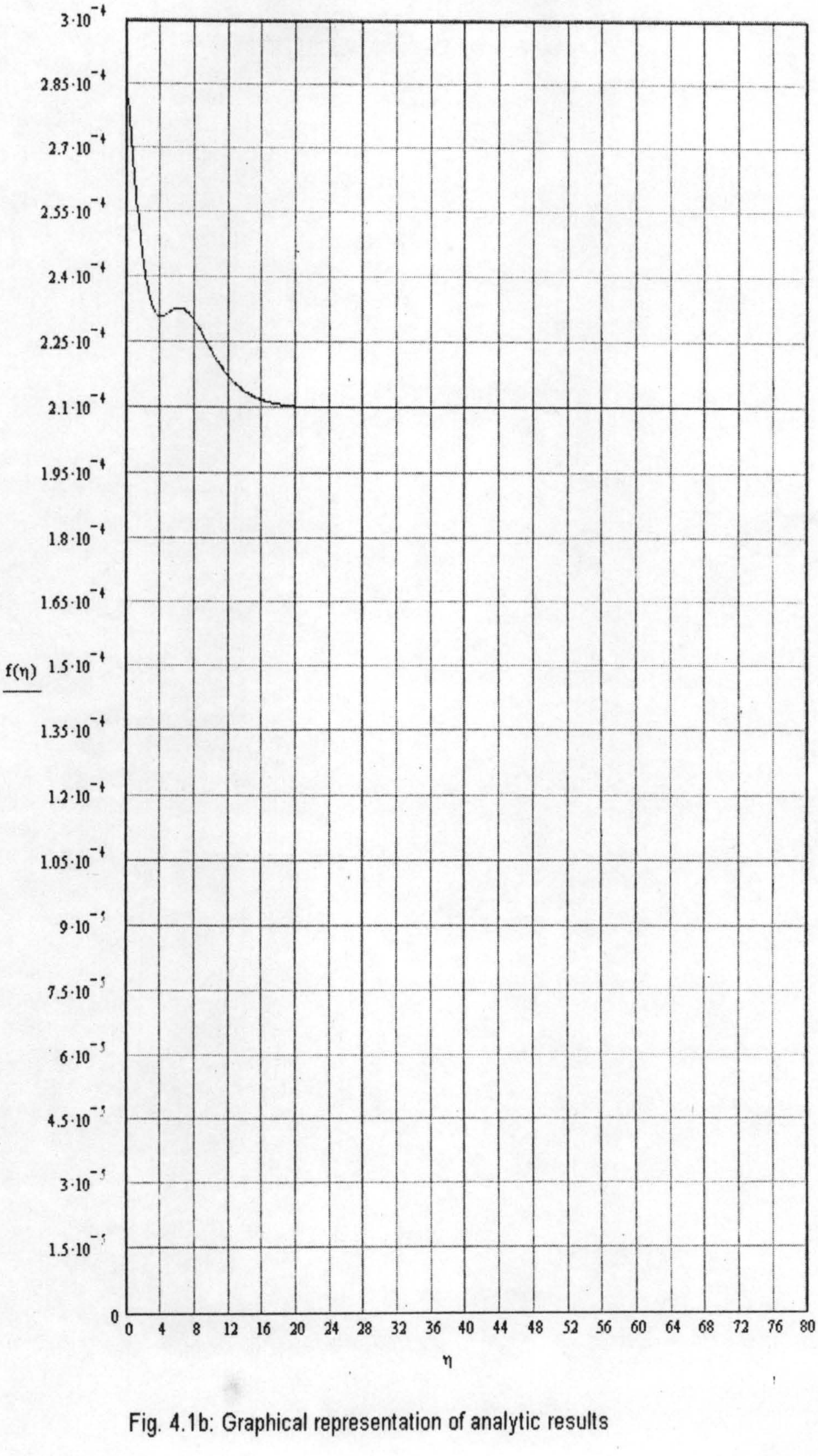

Fig. 4.1b: Graphical representation of analytic results

4.2 COMPUTATIONAL RESULTS USING SPRIET–BARON ALGORITHM

By using the initial value of $X_k = (0,\ -1)$, the step size of $\alpha_k = 0.1$, the direction of search $d_k = (1,\ 2)$, $\alpha = 0.099$ and $P_r = 6.4748$. in (3.9), with $G = 0+n(0.1)1$ and $A = -1+n(0.1)2$ and also using the mathcad code we obtained the result presented in table 4.2; and from table 4.2 we obtained the graphical illustration in figures 4.2.

Table 4.2: Computational results using Spriet-Baron Algorithm					
1	0.000280708574959			48	0.000210915345384
2	0.000278670024412			49	0.000210788132768
3	0.000274187595294			50	0.000210678153738
4	0.000269406212570			51	0.000210583166874
5	0.000264608869074			52	0.000210501202540
6	0.000260045720427			53	0.000210430534232
7	0.000255912933620			54	0.000210369651945
8	0.000252340486260			55	0.000210317237692
9	0.000249388942996			56	0.000210272143190
10	0.000247053978331			57	0.000210233369671
11	0.000245276815172			58	0.000210200049727
12	0.000243958586062			59	0.000210171431080
13	0.000242976636140			60	0.000210146862124
14	0.000242200848233			61	0.000210125779106
15	0.000241508216826			62	0.000210107694803
16	0.000240794224642			63	0.000210092188543
17	0.000239980119060			64	0.000210078897440
18	0.000239015868794			65	0.000210067508715
19	0.000237879248897			66	0.000210057752984
20	0.000236571998090			67	0.000210049398397
21	0.000235114226696			68	0.000210042245538
22	0.000233538225245			69	0.000210036122992
23	0.000231882602413			70	0.000210030883489
24	0.000230187365234			71	0.000210026400570
25	0.000228490235749			72	0.000210022565699
26	0.000226824237640			73	0.000210019285756
27	0.000225216410730			74	0.000210016480885
28	0.000223687419738			75	0.000210014082629
29	0.000222251799259			76	0.000210012032319
30	0.000220918597093			77	0.000210010279693
31	0.000219692221547			78	0.000210008781706
32	0.000218573349171			79	0.000210007501496
33	0.000217559797284			80	0.000210006407513
34	0.000216647305269			81	0.000210005472751
35	0.000215830198380			82	0.000210004674105
36	0.000215101928176			83	0.000210003991808
37	0.000214455496285			84	0.000210003408951
38	0.000213883774963			85	0.000210002911074
39	0.000213379740593			86	0.000210002485815
40	0.000212936636438			87	0.000210002122602
41	0.000212548079623			88	0.000210001812398
42	0.000212208125357			89	0.000210001547481
43	0.000211911299190			90	0.000210001321248
44	0.000211652606029			91	0.000210001128060
45	0.000211427522758			92	0.000210000963095
46	0.000211231979744			93	0.000210000822235
47	0.000211062335200			94	0.000210000701963

95	0.000210000599272			141	0.000210000000412
96	0.000210000511595			142	0.000210000000351
97	0.000210000436738			143	0.000210000000300
98	0.000210000372828			144	0.000210000000256
99	0.000210000318266			145	0.000210000000218
100	0.000210000271686			146	0.000210000000186
101	0.000210000231920			147	0.000210000000159
102	0.000210000197972			148	0.000210000000136
103	0.000210000168992			149	0.000210000000116
104	0.000210000144253			150	0.000210000000099
105	0.000210000123134			151	0.000210000000084
106	0.000210000105106			152	0.000210000000072
107	0.000210000089717			153	0.000210000000062
108	0.000210000076581			154	0.000210000000052
109	0.000210000065367			155	0.000210000000045
110	0.000210000055796			156	0.000210000000038
111	0.000210000047625			157	0.000210000000033
112	0.000210000040651			158	0.000210000000028
113	0.000210000034698			159	0.000210000000024
114	0.000210000029616			160	0.000210000000020
115	0.000210000025279			161	0.000210000000017
116	0.000210000021577			162	0.000210000000015
117	0.000210000018417			163	0.000210000000013
118	0.000210000015720			164	0.000210000000011
119	0.000210000013417			165	0.000210000000009
120	0.000210000011452			166	0.000210000000008
121	0.000210000009775			167	0.000210000000007
122	0.000210000008343			168	0.000210000000006
123	0.000210000007121			169	0.000210000000005
124	0.000210000006078			170	0.000210000000004
125	0.000210000005188			171	0.000210000000004
126	0.000210000004428			172	0.000210000000003
127	0.000210000003779			173	0.000210000000003
128	0.000210000003226			174	0.000210000000002
129	0.000210000002753			175	0.000210000000002
130	0.000210000002350			176	0.000210000000002
131	0.000210000002006			177	0.000210000000001
132	0.000210000001712			178	0.000210000000001
133	0.000210000001461			179	0.000210000000001
134	0.000210000001247			180	0.000210000000001
135	0.000210000001064			181	0.000210000000001
136	0.000210000000909			182	0.000210000000001
137	0.000210000000775			183	0.000210000000001
138	0.000210000000662			184	0.000210000000000
139	0.000210000000565			185	0.000210000000000
140	0.000210000000482			186	0.000210000000000

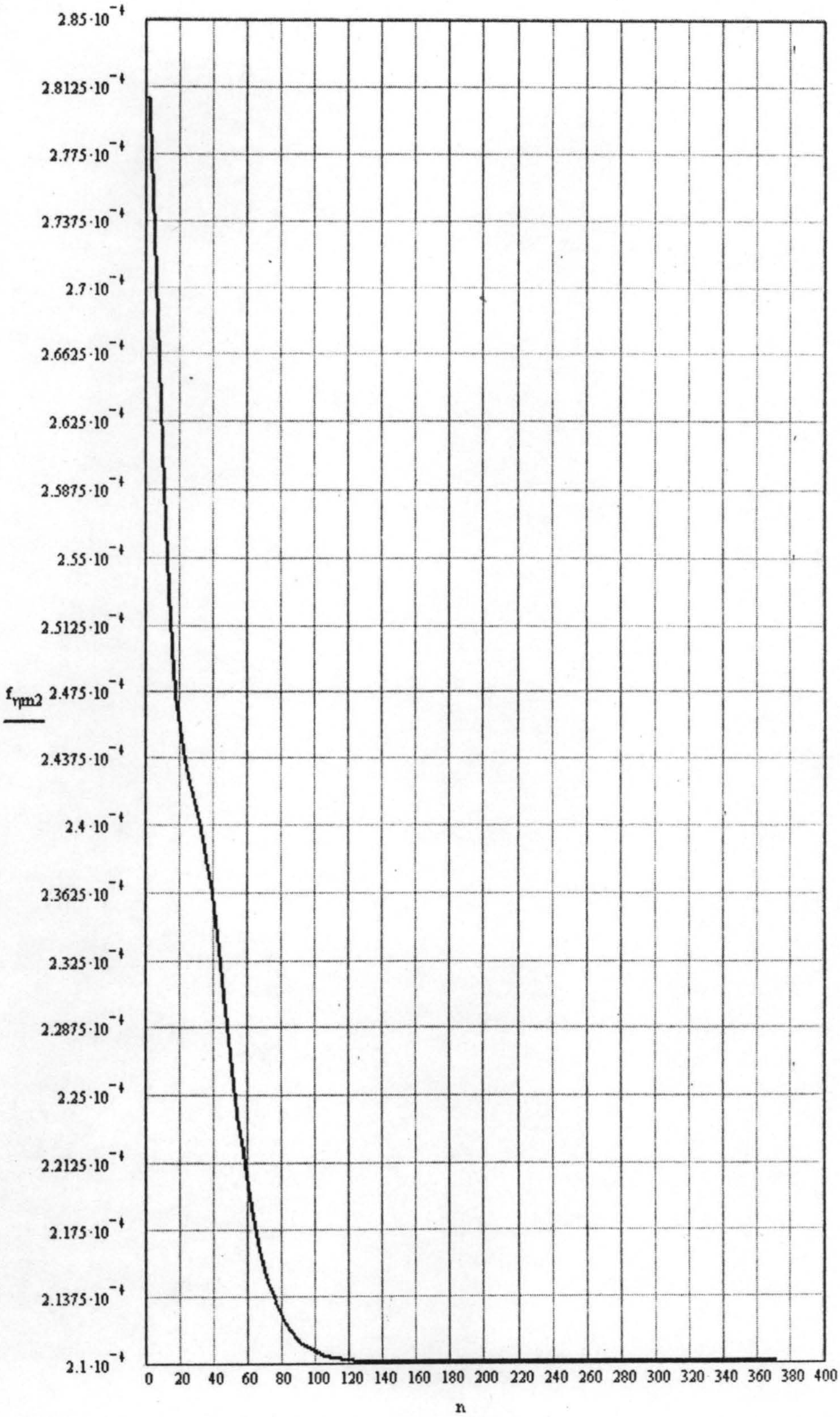

Fig. 4.2a: Graphical illustration of Spriet-Baron results

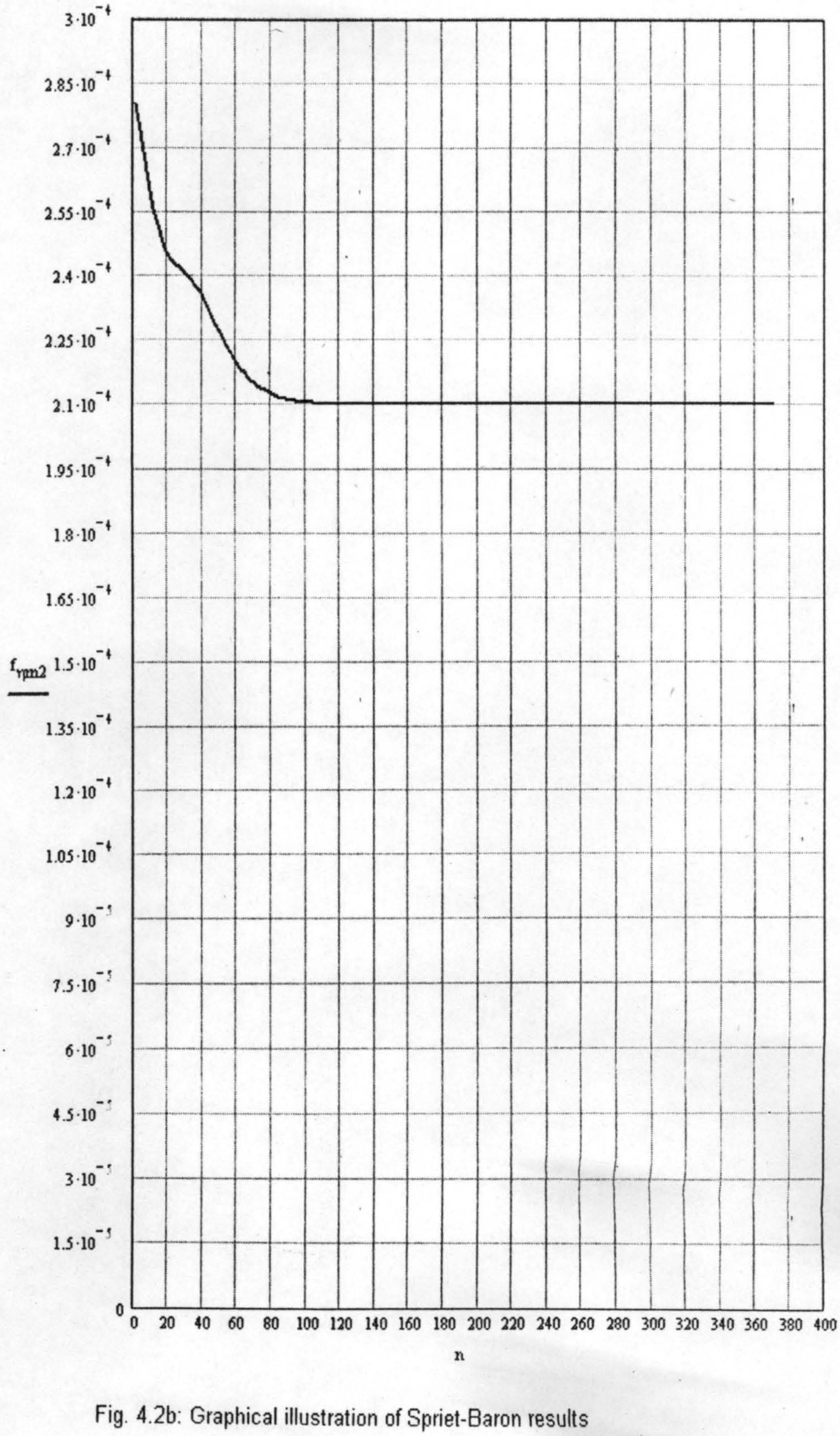

Fig. 4.2b: Graphical illustration of Spriet-Baron results

4.3 COMPUTATIONAL RESULTS USING EXTENDED COGGINS ALGORITHM

By using the initial valueS $(0, -1)$, the step length $\Delta P = 0.1$, $P = 2^r$ where $r = 0, 1, 2, \ldots$, $\alpha = 0.099$ and $P_r = 6.4748$. in (3.7), with $p = 0 + \Delta Pp$ and $q = -1 + \Delta Pp$ and using the mathcad code we obtained the result presented in tables 4.3; and the corresponding figures 4.3.

Table 4.3: Computational results using Extended Coggins Algorithm

1	0.000282565893840
2	0.000280491329455
3	0.000278352579459
4	0.000273919224579
5	0.000264693289241
6	0.000246838921457
7	0.000224617029471
8	0.000229625680436
9	0.000217512666704
10	0.000210067508710
11	0.000210000002753
12	0.000210000000000
13	0.000210000000000
14	0.000210000000000
15	0.000210000000000
16	0.000210000000000

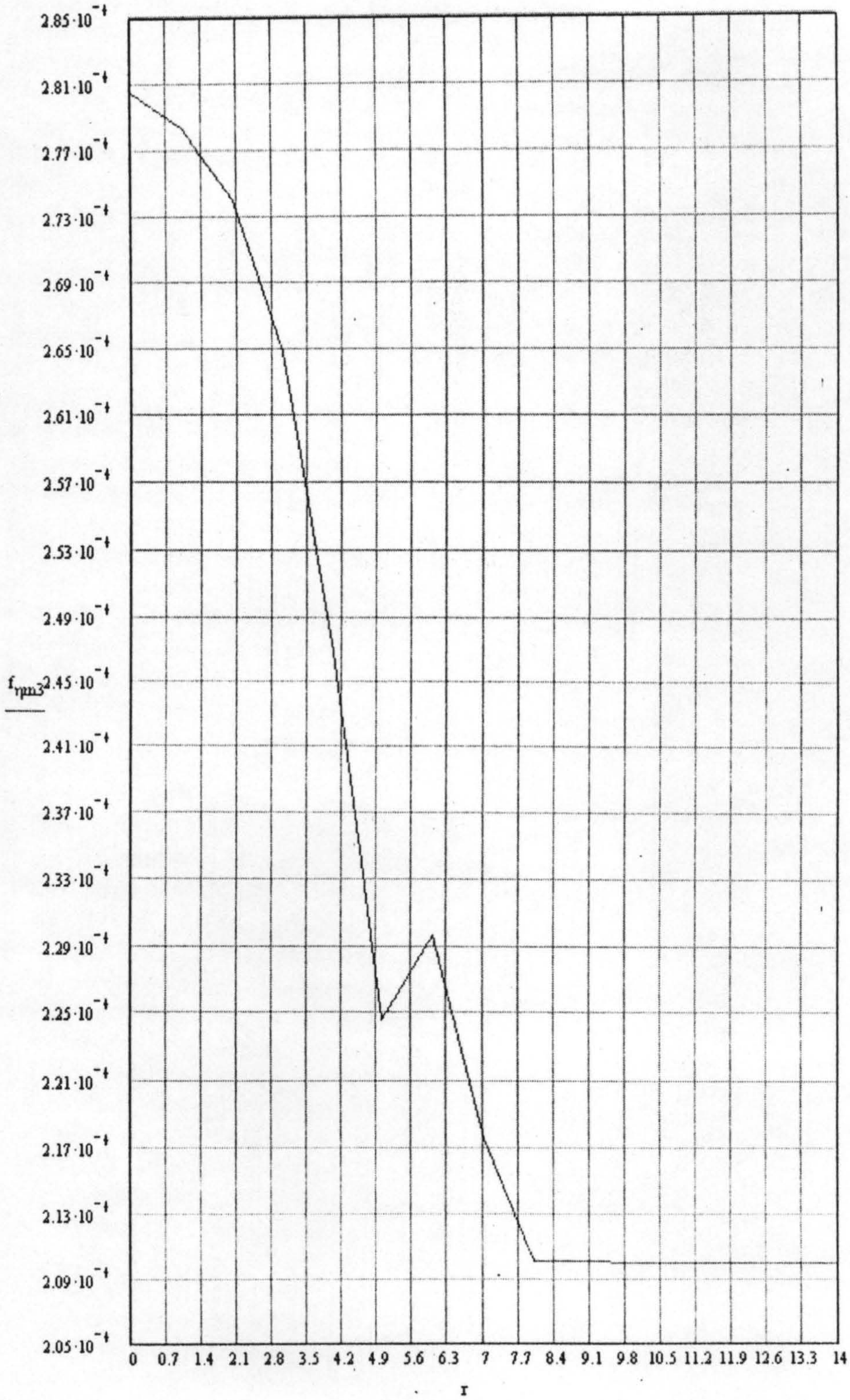

Fig. 4.3a: Graphical illustration of Extended Coggins results

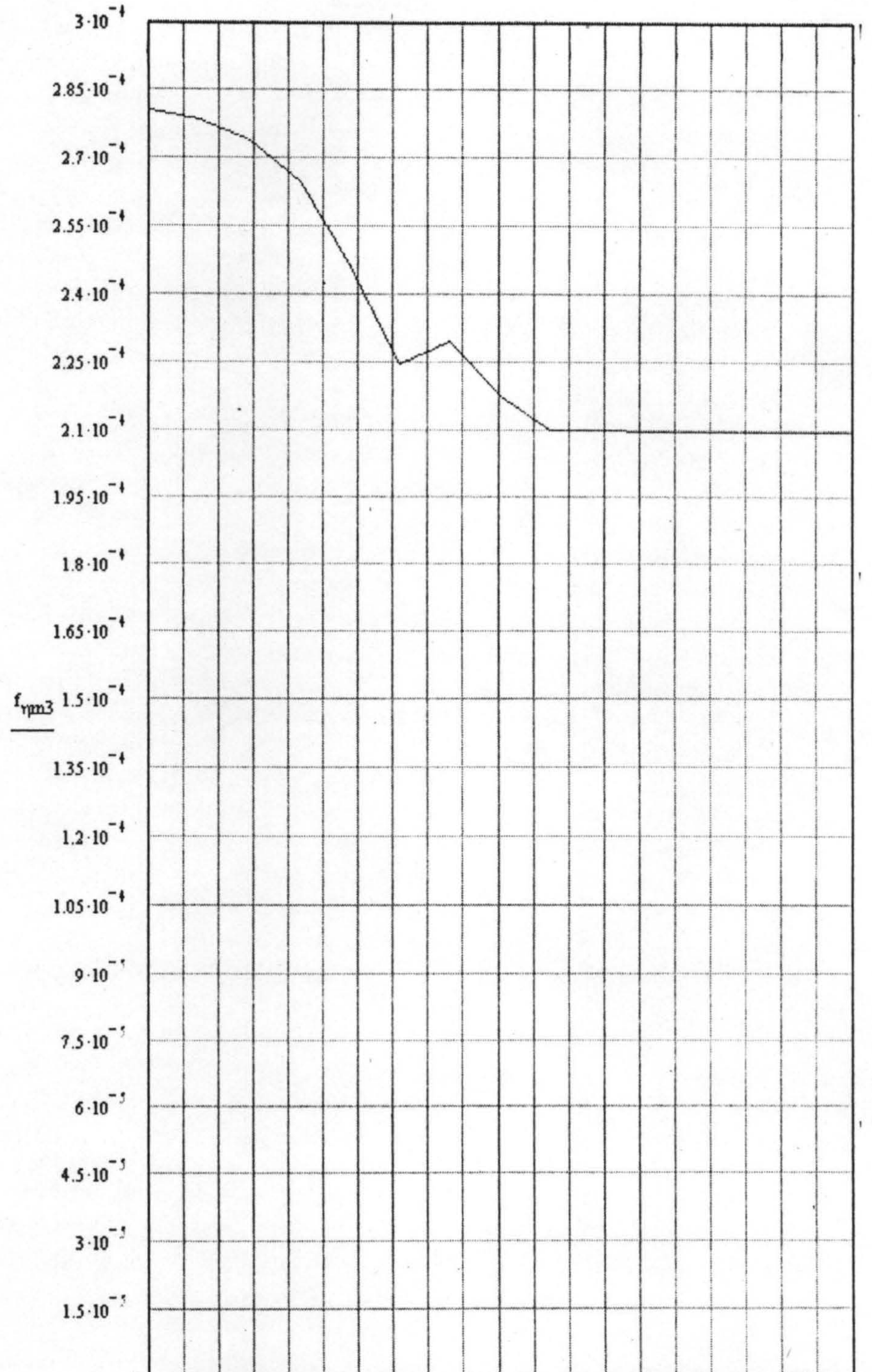

Fig. 4.3b: Graphical illustration of Extended Coggins results

4.4 ANALYSIS OF THE COMPUTATIONAL RESULTS

The table below gives a summary of the various optimization algorithms employed in this study to analyse the Submerged Sewage Dispersion Model.

Algorithm	No of iterations	Global minimum
Analytical	721	2.1×10^{-4}
Spriet-Baron	184	2.1×10^{-4}
Extended Coggins	12	2.1×10^{-4}

Table 4.4.1

The following observations arise from the above tabular presentation:

Remarks

1. In the analytical simulation, the objective function is considered as a function of the variable.

   The optimal point was located after 721 iterations.

   The optimal point of $\eta = 72.1$

   The minimum value of $f(\eta) = 2.1 \times 10^{-4}$

2. For the Spriet-Baron algorithm, the objective function is considered as a function with two variables.

   The optimal point was located after 184 iterations.

   The minimum value of $f(\eta) = 2.1 \times 10^{-4}$

3. For the extended Coggins algorithm, the objective function is considered as a function of two variables.

   The optimal point was located after 12 iterations.

   The minimum value of $f(\eta) = 2.1 \times 10^{-4}$

All the algorithms attain global minimum with different number of iterations. A comparison of the results from the table above shows that the extended Coggins optimization algorithm is a better algorithm for the solution of the problem under study.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 CONCLUSION

The non-gradient method considered so far avals to one the fundamental issues in the design of line search which is a combination of direct search and curve fitting in such a way that under fairly general conditions, convergence to the minimum is guaranteed.

A comparison of the efficiencies of the line search methods considered in locating the optimal value of the function (table 4.4.1) shows that though each method succeeded in approximating the location of the minimum at $X^* = 2, 1 \times 10^{-4}$, the number of iteration shows a great difference. While the Spriet–Baron optimization algorithm requires 184 iterations before attaining the global minimum; the Extended Coggins algorithm attains the global minimum with just 12 iterations.

5.2 RECOMMENDATION

Going by the above presentation, the Extended Coggins optimization method has as an iterative method proves to be better than the analytical and Spriet–Baron methods. This is because it does not consume (occupy) much of computer space and at the same time produces better results with fewer iterations; making it a time-saving non-gradient method.

REFERENCES

1. Gebharl B.P. & Schorr A.W – Steady Laminar Natural Convection Plumes above a Horizontal Line Heat Source; Int. J. Heat and Mass Transfer, 13, 161 – 171, 1970.

2. Howatson A.M. Lund P.G. & Todd J.D. – Engineering Tables and Statistics; Chapman and Hall, 1977

3. Kor R.C.J. & Brookes N.H. – Fluid Mechanics of Waste Water Disposal in the Occean; Annual Review of Fluid Mechanics – Annual Review Inc. Palo Alto Cal.. USA, 7, 187, 1975

4. Luenberger D. – Introduction to Linear and Non–Linear Programming; Addison & Wesley, 1973.

5. Rao S.S. – Optimization Theory and Application; Wiley & Sons, 1984.

6. Reju S.A. – Lecture Notes on Research Oriented Course in Computational Mathematics and Optimization (ROC–CMAI); National Mathematical Centre, Abuja; July 2001.

7. Rosenhead L. – Laminar Boundary Layers; Oxford Clarendon Press, 254 –260, 1963.

8. Ruester J.L. & Mize J.H. – Optimization Techniques with FORTRAN; Mc–Graw Hill book co. 1978.

9. Spriet J. & Baron G. – Modelling Dispersion in Submerged Sewage Field; Optimization Techniques J. cee (Ed) pp 229, 1975.

10. Subramanian S, Reju S.a. & Ibiejugba M.A. – An Extended Coggins Optimization Technique; Nig J Math Appl, Vol 9, 204 – 258, 1996.

11. Walsh G.R. – Methods of Optimization; Wiley & Sons, 1975.

APPENDIX I

1(a) Momentum = Mass x Velocity
Momentum = Area x Dynamic velocity
if area = $10m^2$
$M = 10m^2 \times (1.00 \times 10^{-3}) Kgm^{-1}s^{-1}$
$= 10m^2 \times 0.001 Kgm^{-1}s^{-1}$
$= 0.01 Kgms^{-1}$

1(b)

$$P_r = \frac{C_p \mu}{k}$$

$C_p = 3930\ J/g\ K$; $\mu = 1.005\ Kgm^{-1}$; $K = 0.61\ w/mK$
Given these values, we have that: $P_r = 6.4748$

1(c) According to Schlictings [7], we can determine α from the expression:

$$\alpha = 0.2753 \left(\frac{M}{\rho}\right)^{2/3}$$

M = Momentum
ρ = density
Howatsn et al [15] gave the values as follows:
$M = 0.01\ Kg\ m.\ s^{-1}$; $\rho = 1025\ Kg\ m^{-1}$; which gives the value:
$\alpha = 0.099$

APPENDIX II
NOTATION / SELECTIVE NUMENCLATURE

u = horizontal velocity component

X = horizontal distance

y = vertical distance

F_0 = density difference flux per unit length of diffusion

G_r = GRASHOF number

P_0 = mass flux of pollutant per unit length of diffusion

P_r = PRANDTL number

α = thermal expansion coefficient

Ψ = stream function

θ = reduced density difference

ρ specific mass

Γ = Gamma function

K = thermal conductivity of fluid

η = similarity variable

C_p = specific heat at constant pressure of the fluid

f = dimensionless stream function