Loughborough University

This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (https://dspace.lboro.ac.uk/) under the following Creative Commons Licence conditions.

creative commons

COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

Attribution. You must attribute the work in the manner specified by the author or licensor.

Noncommercial. You may not use this work for commercial purposes.

No Derivative Works. You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full license).

Disclaimer

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/

BLDSC no:- DX 78779

LOUGHBOROUGH
UNIVERSITY OF TECHNOLOGY
LIBRARY

AUTHOR/FILING TITLE

TOUATI-AHMED, D

ACCESSION/COPY NO.

03407802

VOL. NO. | CLASS MARK

~~5 JUL 1991~~ | LOAN COPY

~~5 JUL 1991~~

3 JUL 1992

LB 0034078029

A STUDY OF HYBRID CONJUGATE GRADIENT METHODS

by

Djamal Touati-Ahmed

A thesis submitted in partial fulfilment of the requirements for the award of

DOCTOR OF PHILOSOPHY

of the Loughborough University of Technology.

September 1989.

© by D. Touati-Ahmed 1989.

Loughborough University
of Technology Library
Date Nw 90
Class
Acc
No 04034078 02

y93179 22

To: My wife PINA

and my parents ZINA and EL-HABIB.

Acknowledgements

My sincere thanks are due to Professor C. Storey for his valuable supervision, guidance and comments for the research in this thesis.

I would also like to thank my wife Pina and my parents Zina and El-Habib for their great moral support.

Thanks are also due to Mrs. Wright for her patience while typing this thesis.

Finally I wish to thank the Algerian Ministry of Higher Education for sponsoring this research.

Summary

The main subject of the research in this thesis is the study of conjugate gradient methods for optimization and the development of improved algorithms. After an introductory first chapter, Chapter 2 contains a background of numerical methods for optimization in general and of conjugate gradient-type algorithms in particular. In Chapter 3 we study the convergence properties of conjugate gradient methods and discuss Powell's (1983) counter example that proves that there exist twice continuously differentiable functions with bounded level sets for which the Polak-Ribière method fails to achieve global convergence whereas the Fletcher-Reeves method is shown to be globally convergent, despite the fact that in numerical computations the Polak-Ribière method is far more efficient than that of Fletcher-Reeves. Chapters 4 and 5 deal with the development of a number of new hybrid algorithms, three of which are shown to satisfy the descent property at every iteration and achieve global convergence regardless of whether exact or inexact line searches are used. A new restarting procedure for conjugate gradient methods is also given that ensures a descent property to hold and global convergence for any conjugate gradient method using a non negative update. The application of these hybrid algorithms and that of the new restarting procedure to a wide class of well-known test problems is given and discussed in the final Chapter "Discussions and Conclusions". The results obtained, given in the appendices, show that a considerable improvement is achieved by these hybrids and by methods using the new restarting procedure over the existing conjugate gradient methods and also over quasi-Newton methods.

TABLE OF CONTENTS

	PAGE
CHAPTER 1. INTRODUCTION.	1
CHAPTER 2. BACKGROUND OF NUMERICAL METHODS FOR OPTIMIZATION.	7
1. Introduction.	7
2. Some Basic Results for Constrained Optimization.	9
3. Solving Constrained Optimization Problems Via Unconstrained Optimization.	13
3.1. The Parametric Methods.	14
3.2. The Non-Parametric Methods.	16
3.3. The Lagrangian methods.	17
4. Numerical Methods for Unconstrained Optimization.	18
4.1. Direct Search Methods.	19
4.2. Descent Methods.	22
4.2.1. Convergence Theorems for Descent Methods.	25
4.2.2. Line Search Techniques.	32
4.2.3. The Method of Steepest Descent.	38
4.2.4. The Method of Newton.	41
4.2.5. The Modified Newton Method.	42
4.2.6. The Quasi-Newton Methods.	43
4.2.7. Conjugate Gradient Methods.	47
5. A Study of Conjugate Gradient Methods.	50
5.1. Quadratic Functions and Conjugacy.	50
5.1.1. Basic Properties of Quadratic Functions.	51
5.1.2. Conjugacy.	53
5.1.3. Conjugate Properties of Quadratic Functions.	55
5.2. Conjugate Gradient Methods for Quadratic Functions.	58
5.2.1. Generation of Conjugate Gradients.	59
5.2.2. Convergence Properties.	64
5.3. Conjugate Gradient Methods for General Functions.	66
5.3.1. Conjugate Gradients Without Line Searches.	67
5.3.2. Restarting Procedures.	67

CHAPTER 3. EFFICIENCY AND CONVERGENCE PROPERTIES OF CONJUGATE GRADIENT METHODS. 71

1. Introduction and Preliminaries. 71

2. Descent Property and Global Convergence of the Fletcher-Reeves Method. 73

2.1. The Case of Exact Line Search. 73

2.1.1. Descent Property. 73
2.1.2. Global Convergence. 73

2.2. The Case of Inexact Line Search. 76

2.2.1. Descent Property. 76
2.2.2. Global Convergence. 78

3. The Polak-Ribière Method. 80

3.1. Comparison with the Fletcher-Reeves Methd. 80

3.1.1. Computational Comparison. 80
3.1.2. Theoretical Explanation. 81

3.2. Global Convergence of the Polak-Ribière Method. 90

4. Improvements on the Conjugate Gradient Methods. 95

4.1. A Conjugate Direction Method by D. Le 95
4.2. An Angle Test as a Restart Criterion by D.F. Shanno 98
4.3. Avoiding Negative Updates of the P.R. Method. 101

CHAPTER 4. GLOBALLY CONVERGENT HYBRID CONJUGATE GRADIENT METHODS. 106

1. Introduction. 106

2. A New Hybrid Algorithm: HYBRID 1. 107

2.1. Construction of the Algorithm. 107
2.2. Descent Property and Global Convergence of HYBRID1. 109

2.2.1. Exact Line Search. 109
2.2.2. Inexact Line Search. 112

2.3. Computational Performance of HYBRID 1. 117

3. An Improved Hybrid Algorithm: HYBRID 2. 121

3.1. Definition of the New Algorithm. (HYBRID 2). 124
3.2. Descent Property and Global Convergence of HYBRID2. 126

3.2.1. Exact Line Search. 126
3.2.2. Inexact Line Search. 129

3.3. Computational Performance of HYBRID 2. 141

CHAPTER 5. EFFICIENT HYBRID CONJUGATE GRADIENT TECHNIQUES. 143

1. Introduction. 143

2. An Efficient Hybrid Conjugate Gradient Algorithm HYBRID3. 144

2.1. Construction of the Algorithm. 144
2.2. Descent Property and Global Convergence of HYBRID3. 148

2.2.1. The Case of Exact Line Search. 148
2.2.2. The Case of Inexact Line Search. 151

2.3. Computational Performance of HYBRID 3. 156

3. An Efficient Restarting Procedure. 158

3.1. Defining the Procedure. 158
3.2. Computational Performance of the new Restarting Procedure. 160

CHAPTER 6. DISCUSSIONS AND CONCLUSIONS. 162

1. Results Description. 162

2. Discussions of the Results. 166

3. Comparison of HYBRID 3 with Quasi-Newton Methods 184

APPENDIX 1. PROOFS OF THEOREMS GIVEN IN CHAPTER 2. 187

1. Proof of Theorem 2.2.1. 187

2. Proof of Theorem 2.5.2. 188

APPENDIX 2. EXPLICIT SETS OF RESULTS. 190

APPENDIX 3. SUMMARIZED AND COMPARISON RESULTS. 318

REFERENCES. 340

CHAPTER 1

INTRODUCTION

The research carried out in this thesis, concerns the study of numerical methods for optimization and the development of improved algorithms. This study is mainly concerned with the numerical methods for unconstrained optimization, for the simple reason that, as can be seen in Chapter 2, section 3, constrained problems in optimization can easily be converted into a sequence of unconstrained problems by a method of multipliers or into a sequence of unconstrained saddle points problems involving Lagrangians.

A particular interest is given to conjugate direction methods for optimization in general and to conjugate gradient-type algorithms in particular. This interest was triggered by a particular paper of Powell (1983) in which he had shown that, although in numerical computations, the conjugate gradient algorithm of Polak-Ribière (1969) is far more successful than that of Fletcher-Reeves (1964), if the Polak-Ribière algorithm is used, then even with exact arithmetic and exact line search, there exist twice continuously differentiable functions with bounded level sets, for which the gradient norms are bounded away from zero. In this same paper, Powell (1983) has also shown by a standard method of proof that the conjugate gradient algorithm of Fletcher-Reeves (1964) with exact arithmetic and exact line search is always convergent. This result has further been extended by Al-Baali (1985) to show that even with an inexact line search satisfying some standard conditions, the Fletcher-Reeves algorithm satisfies the descent property and is globally convergent.

After this introductory chapter, Chapter 2 of this thesis gives a thorough study of the theoretical and practical aspects of the existing numerical methods for optimization, providing an extensive literature survey of the subject and an up-to-date insight into these methods and their convergence properties. After the introductory section of Chapter 2, section 2 outlines and discusses the basic theorems and methods related to constrained optimization. In section3, we discuss and present the various techniques of converting constrained optimization problems into unconstrained optimization problems and solving these problems via unconstrained optimization. Section 4 deals with the numerical methods for unconstrained optimization. We discuss in this section both direct search and descent methods, outlining as much as possible, their convergence properties and the line search techniques. It will then be apparent how our interest was directed towards the methods of conjugate gradients, to which we have devoted the following section; section 5. In this latter section we go further in detail into the theoretical aspects of the conjugate gradient methods, from the generation of conjugate gradients in the case of quadratic functions, to the use of restarting (or reinitializing) in the case of general functions.

In Chapter 3, we discuss the computational efficiency and the convergence properties of conjugate gradient methods. This chapter is divided into four sections. After an introductory section containing all the preliminaries needed for the discussion of the following sections, section 2 deals with the descent properties and global convergence of the method of Fletcher Reeves. We prove the theorems given by Powell (1983) for the case of exact arithmetic and exact line search and the theorems given by Al-Baali (1985) for the case of inexact line search. The study then continues in section 3 by considering the method of Polak-Ribière that despite the fact that it is computationally far more efficient than that of Fletcher-Reeves, there exist twice continuously differentiable functions with bounded level sets, for which, if this method is used, the gradient norms remain bounded away from zero, meaning that the method fails to converge globally. In this section we give thorough computational comparisons based on both historical numerical results and on results obtained from the work included in this thesis.

Furthermore, we also attempt in this section to theoretically explain the reasons forthis feature of the Polak-Ribière method. We then conclude Chapter 3 by a fourth section dealing with the recent ideas in the literature to attempt to improve on the conjugate gradient methods and also ideas developed in this thesis in an attempt to combine the desirable computational aspects of the Polak-Ribière method and the useful theoretical aspects of the Fletcher-Reeves method.

We then devote Chapters 4 and 5 of this thesis to the development of a number of new improved hybrid conjugate gradient algorithms. In Chapter 4, we develop two new algorithms and prove theorems for their descent properties and global convergence, whereas in Chapter 5, on top of providing a third globally convergent hybrid algorithm that proves to be even more efficient computationally than Quasi-Newton algorithms on a large class of test problems, we also give some new restarting criteria for conjugate

gradient methods, that provide descent property and global convergence for any conjugate gradient method with a non-negative update, and that prove to be computationally very efficient. The first two hybrids (HYBRID 1 and HYBRID 2) are also reported in Touati-Ahmed and Storey (1986) and the third hybrid (HYBRID 3) and the restarting criteria are reported in Touati-Ahmed and Storey (1987) which then has been communicated by L.C.W. Dixon to J.O.T.A. (Journal of Optimization Theory and Applications) and has been accepted for publication.

The discussions of the results obtained for the various methods dealt with in this thesis and their comparisons, together with our final conclusions are given in Chapter 6.

Chapter 2

Background of Numerical Methods for Optimization

1. INTRODUCTION.

Numerical methods for optimization are designed to find an n-dimensional real vector $x^* \in K$ ($K \subset \mathbb{R}^n$) such that if $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function to be minimized, we have:

$$f(x^*) \leq f(x) \quad \text{for all } x \in K \quad . \tag{2.1.1}$$

A general optimization problem can be written in the following form:

$$\left.\begin{array}{l} \text{Minimize } f(x) \\ \text{Subject to:} \\ \qquad g_i(x) \leq 0 \qquad i = 1,2,\ldots,m \\ \qquad h_j(x) = 0 \qquad j = 1,2,\ldots,p \end{array}\right\} \tag{2.1.2}$$

where f, g_i and h_j are real valued functions from $\mathbb{R}^n \to \mathbb{R}$. The set K of admissible (or feasible) points is written as follows:

$$K = \left\{ x \in \mathbb{R}^n \,\middle|\, \begin{array}{l} g_i(x) \leq 0, \quad i = 1,\ldots,m; \\ h_j(x) = 0, \quad j = 1,\ldots,p \end{array} \right\} \quad . \tag{2.1.3}$$

By putting conditions on f, g_i, h_j and the set K, one can distinguish the existing problem type categories in optimization and develop numerical methods for solving each type of problem accordingly.

The particular case in which the functions f, g_i and h_j used in (2.1.2) are linear is termed "Linear Programming". To solve problems of this type, there exist good algorithms that always

give an exact solution of the problem. For example, the Simplex method of Dantzig [ ] is a highly efficient algorithm. The average number of iterations required to reach an optimum for problems of n variables and m constraints with n much greater than m, is of the order 2m, much less than might be expected from the number of vertices of the constraint set. If a nonlinear programming problem can be arranged so as to be solvable by a modified simplex method, this is commonly the most efficient procedure. In particular, a problem which allows an adequate approximation by piecewise linear functions, of not too many variables, may be computed as a separable programming problem. Also a problem with a quadratic objective function and linear constrains can be solved by a modified simplex method, such as the method of Wolfe.

Unless the problem is convex, any iterative method will, if it converges at all, converge to a local critical point, which is not necessarily the global optimum. This difficulty does not occur with a convex problem, for any local minimum of a convex problem is a global minimum. In general however, it is hard to find a computationally efficient algorithm which will always find the global minimum. There is always a tradeoff between efficiency, and the ability to follow small-scale features of the function. In practice, some assumption of smoothness of the function(s) is made. For further details on linear, quadratic, separable and convex programming the reader should refer to Zoutendijk (1976), Fletcher (1981), Nimirovsky and Yudin (1983), Osborne (1985) and Minoux (1986) amongst others.

For the purpose of this thesis however, we shall distinguish

two main categories namely: Constrained and Unconstrained Optimization methods. Unconstrained optimization, as its name indicates, is the case in which there are no constraints g_i or h_j. In other words the set K of admissible points is equal to $\mathbb{R}^n$. This is a very important case in practice because, as will be seen in Section 3 of this chapter, constrained problems in optimization can be converted into a sequence of minimum/maximum problems by a method of multipliers or into a sequence of saddle point problems using Lagrangians. Some of the recent text books dealing with unconstrained optimization include Wolfe (1978), Fletcher (1980), Gill, Murray and Wright (1981), Dennis and Schnabel (1983) and Minoux (1986). In the remaining part of this chapter, we shall give some basic results for constrained optimization, outline some of the methods with which constrained problems can be converted into unconstrained problems, give a description of some well known numerical methods for unconstrained optimization and give an explicit up-to-date study of conjugate gradient methods on which the research in this thesis is based.

2. SOME BASIC RESULTS FOR CONSTRAINED OPTIMIZATION.

The most important concept in the theory of constrained optimization is the concept of Lagrange multipliers. To illustrate the introduction of this concept, we shall first consider problem (2.1.2) with the equality constraints $h_j(x) = 0$, $j = 1,\ldots,p$ only, that is we shall ignore the inequality constraints $g_i(x) \leq 0$, $i = 1,\ldots,m$ for the time being. In this case the Lagrangian function takes the form:

$$\mathcal{L}(x,\lambda) = f(x) - \sum_{j=1}^{p} \lambda_j h_j(x) \qquad (2.2.1)$$

where λ_j, $j = 1,\ldots,p$ are the Lagrange multipliers of the constraints. The Lagrange multiplier of any constraint measures the rate of change in the objective function, consequent upon changes in that constraint function (see Fletcher (1981)). This information can be valuable in the sense that it indicates how sensitive the objective function is to changes in different constraints.

We now consider the additional complication of having inequality constraints as well as equality constraints. It is important to realize that only the active constraints at x^* (i.e. such that $g_i(x^*) = 0$) can influence the problem, but it is possible and consistent to regard any inactive constraint as having a zero Lagrange multiplier. The Lagrangian function takes therefore the form:

$$\mathcal{L}(x,\lambda,\mu) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) - \sum_{j=1}^{p} \mu_j h_j(x) \qquad (2.2.2)$$

where λ_i and μ_j are Lagrange multipliers of the constraints. For a more detailed study of the concept of Lagrange multipliers in optimization the reader should refer to Rockafeller (1976) and Fletcher (1981).

Having illustrated the introduction of Lagrange multipliers, we shall now discuss the optimality conditions and the existence and unicity of an optimal solution x^* of (2.1.2).

Since equality constraints can be converted into inequality constraints in the following manner:

$$h_j(x) = 0 \quad \Leftrightarrow \quad \begin{cases} h_j(x) \leq 0 \\ \text{and} \\ -h_j(x) \leq 0 \end{cases} \qquad (2.2.3)$$

it is no restriction to consider problem (2.1.2) with only inequality constraints. Therefore, in what follows, instead of (2.1.2), we shall consider the problem:

$$\left.\begin{array}{l} \text{Minimize} \quad f(x) \\ \text{Subject to:} \\ \qquad g_i(x) \leq 0 \qquad i = 1,\ldots,m \end{array}\right\} \quad . \tag{2.2.4}$$

We also make the general assumption that the functions f, g_1, $g_2,\ldots,g_m$ are differentiable. For $x \in K$, let:

$$Z(x) = \left\{ s \in \mathbb{R}^n \mid \exists\, \bar{\alpha} > 0 \text{ s.t. } x + \alpha s \in K \text{ for } 0 \leq \alpha \leq \bar{\alpha} \right\}$$

be the set of feasible directions from x. A first optimality criterion via feasible directions is obtained by the following theorem.

Theorem 2.2.1: Let x^* be a feasible solution of the problem (2.2.4). If x^* is an optimal solution of (2.2.4) then we have:

$$s^T \nabla f(x^*) \geq 0 \qquad \forall\, s \in \overline{Z(x^*)}. \tag{2.2.5}$$

If we strengthen our assumptions to impose the conditions that f is pseudoconvex (i.e. $(y-x)^T \nabla f(x) \geq 0 \Rightarrow f(y) \geq f(x)$), and that $g_1, g_2,\ldots,g_m$ are quasiconvex (i.e. $g_i(\lambda x + (1-\lambda)y) \leq \max\{g_i(x), g_i(y)\}$ for $i = 1,\ldots,m$) then we have:

$$x^* \text{ optimal} \Leftrightarrow s^T \nabla f(x^*) \geq 0 \quad \forall\, s \in \overline{Z(x^*)} \quad .$$

The proof of this theorem is given in Appendix 1. This theorem gives necessary and sufficient conditions for a feasible point x^* to be optimal. Unfortunately these conditions are not very practical

in the sense that it is very difficult in practice to handle the set $Z(x^*)$. Fritz and John (1948) and Kuhn and Tucker (1951) give more practical conditions for a feasible solution to be optimal, using Lagrange multipliers and Lagrangian functions. For a more detailed study of these conditions, we also refer to Powell (1974), Avriel (1976), Fletcher (1981) and Minoux (1986).

In what follows we shall give some results concerning the existence and unicity of an optimal solution of problem (2.2.4), the proofs of which can be found in most text books dealing with constrained optimization.

Let

$$s(f,\alpha) = \left\{x \in \mathbb{R}^n \mid f(x) \leq \alpha\right\}$$

be the level set of the function f at the level α.

Lemma 2.2.1: Suppose f is continuous. If there exists $x^{(0)} \in K$ for which $s(f,f(x^{(0)})) \cap K$ is compact then f has an optimal solution x^* in K.

Lemma 2.2.2: Suppose $f(x^{(k)}) \to \infty$ as $k \to \infty$ for all sequences $\{x^{(k)}\}_{k\in\mathbb{N}}$ from K such that $\|x^{(k)}\| \to \infty$ as $k \to \infty$. Then we have:

$$s(f, f(x^{(0)})) \cap K \quad \text{is bounded.}$$

Theorem 2.2.2: (Existence). Let $f, g_1, \ldots, g_m$ be continuous functions and suppose the following condition is satisfied:

$$\left\{\{x^{(k)}\}_{k\in\mathbb{N}} \subset K;\ \|x^{(k)}\| \to \infty\right\} \Rightarrow f(x^{(k)}) \to \infty.$$

Then, the problem (2.2.4) has an optimal solution.

The proof of this theorem follows from Lemmas 2.2.1 and 2.2.2.

Definition: The Hessian matrix H(x) of f (the matrix of second derivatives of f at x) is said to be uniformly positive definite on a set $E \in \mathbb{R}^n$ if there exists $m > 0$ such that:

$$z^T H(x) z \geq m \|z\|^2 \qquad \forall z \in \mathbb{R}^n \quad \text{and} \quad \forall x \in E.$$

Theorem 2.2.3: (Existence and Unicity).

. Let K be a closed convex set.

Let f be twice continuously differentiable in K, and

Let H(x) be uniformly positive definite on K.

Then:

f has a unique minimum x^* in K.

3. SOLVING CONSTRAINT OPTIMIZATION PROBLEMS VIA UNCONSTRAINED OPTIMIZATION.

Various methods have been suggested in the literature that enable constrained optimization problems to be solved via unconstrained optimization techniques. These methods solve the problem (2.2.4) by reducing the computational process to unconstrained minimization of a compound (or penalty) function, combining in a particular way the objective function f, the constraints $g_1,\ldots,g_m$ and possibly one or more controlling parameters.

Surveying the literature in this subject, one can roughly distinguish three main classes of methods:

The interior point methods: These methods operate in the set

$$\mathbb{R}^o = \left\{ x \in \mathbb{R}^n \mid g_i(x) < 0 ; \quad i = 1,\ldots,m \right\}.$$

The exterior point methods: These methods present an approach to a minimum solution x^* of the problem (2.2.4) from outside the constraints set:

$$R = \left\{ x \in \mathbb{R}^n \mid g_i(x) \leq 0 , \quad i = 1,\ldots,m \right\}.$$

The Lagrangian methods: These methods are based on the Lagrangian function associated with problem (2.2.4) and they operate in the n-dimensional Euclidian vector space E_n.

The interior-point and exterior-point methods can be subdivided into two classes; the parametric methods have one or more controlling parameters in the penalty function to control the convergence towards a minimum solution; the non-parametric methods on the other hand, do not explicitly operate with controlling parameters. Lagrangian methods, however, are always parametric.

3.1 The Parametric Methods:

The parametric interior-point methods are based on penalty functions, which in this case are called "Barrier" functions, of the form:

$$B_r(x) = f(x) - r^{\lambda} \sum_{i=1}^{m} \Phi(g_i(x)) ,$$

where r denotes a positive controlling parameter, Φ is a function of one variable, η say, defined and continuously differentiable in the interval $\{\eta \mid \eta > 0\}$ and such that $\Phi(0^+) = -\infty$, and

where λ denotes the order of the pole of Φ' at $\eta = 0$. For further references on these methods we refer to Murray (1967), Davies (1970) and Lootsma (1972).

The parametric exterior-point methods on the other hand, are based on penalty functions, which in this case are called "Loss" functions, of the form:

$$L_s(x) = f(x) - s^{-\mu} \sum_{i=1}^{m} \psi(g_i(x)),$$

where s denotes a positive controlling parameter and ψ is a continuously differentiable function of one variable, η say, such that

$$\psi(\eta) = 0 \qquad \text{for} \quad \eta \geq 0,$$

$$\psi(\eta) < 0 \qquad \text{for} \quad \eta < 0 .$$

To define μ we need to introduce the function ω such that

$$\omega(\eta) = \psi(\eta) \qquad \text{for } \eta \leq 0.$$

A loss function is then said to be of order μ if the derivative ω' of ω is analytic and if it has a zero of order μ at $\eta = 0$. Simple examples of loss functions are obtained by using $\omega'(\eta) = (-\eta)^{\mu}$ with positive integer μ. These methods were originated by Courant (1943). Further developments came from Ablow and Brigham (1955), Butler and Martin (1962), Fiacco and McCormick (1967a) and Beltrami (1969).

Mixed penalty functions were also considered in the literature so that many of the properties of barrier and loss function techniques can be established simultaneously. Combinations of these methods were first introduced by Fiacco and McCormick (1966). A mixed

penalty function is given by:

$$M_{rs}(x) = f(x) - r^{\lambda} \sum_{i \in I_1} \Phi(g_i(x)) - s^{-\mu} \sum_{i \in I_2} \psi(g_i(x))$$

where $I_1 \cap I_2 = \phi$ and $I_1 \cup I_2 = I = \{1,2,\ldots,m\}$. This partitioning is arbitrary and either I_1 or I_2 may be empty.

3.2 The Non-Parametric Methods.

The non-parametric interior point methods are based on barrier functions of the form:

$$B_t(x) = p\, \Phi(t-f(x)) + \sum_{i=1}^{m} \Phi(g_i(x))$$

where t is a positive controling variable,
Φ is as defined for the parametric case and p is a positive weight factor attached to the term which contains the objective function to control convergence. An interesting development was initiated by Rosenbrock (1960a) and continued by Huard (1964) when he proposed the method of centres. It then has been theoretically and computationally explored by Faure and Huard (1965) and (1966), Bui Trong Lieu and Huard (1966), Huard (1967) and (1968) and Tremoliĕres (1968). The method of centres generates a sequence of points converging to a minimum solution of the problem. Each of these points (or centres) is obtained by unconstrained maximization of a distance function: a particular combination of the objective function and the constraint functions. However, some distance functions may also be regarded as penalty functions without controlling parameters. Starting from this idea, Fiacco and McCormick (1967b) presented a non-parametric version of Sequential Unconstrained

Minimization Techniques (S.U.M.T.), and Fiacco (1967) showed that similar versions can be obtained for a large class of interior-point as well as exterior-point methods.

The non-parametric exterior-point methods on the other hand, are based on loss functions of the form:

$$L_t(x) = p\,\psi(t-f(x)) + \sum_{i=1}^{m} \psi(g_i(x))$$

where ψ is as defined for the parametric case above and p as defined in the non-parametric interior-point case.

The unconstrained penalty functions turn out to be increasingly ill conditioned if the penalty parameter increases. Thus, the solution of the unconstrained optimization problems becomes more difficult. The numerical difficulties encountered, as reported in Murray (1967) and Lootsma (1969), stimulated further research. To overcome these difficulties, the use of some kind of scaling technique to the unconstrained problem seems to be a good remedy. For example, the algorithms presented in Fletcher (1970), Oren and Luenberger (1974) or Shanno and Phua (1978) could be applied. An alternative way is to prevent the occurrence of ill conditioned problems by the introduction of problems that do not suffer from this complication. This approach is done by means of "Augmented Lagrangian Functions".

3.3 The Lagrangian Methods:

Let us consider the classical constrained minimization problem:

$$\left.\begin{array}{l} \text{Minimize} \quad f(x) \\ \text{Subject to:} \\ \qquad h_j(x) = 0 \qquad j = 1,\ldots,p \end{array}\right\} \tag{2.3.1}$$

The Lagrangian function to be considered is the following:

$$\mathcal{L}(x,\lambda) = f(x) - \sum_{j=1}^{p} \lambda_j h_j(x) \quad . \tag{2.3.2}$$

The stationary points of this function are characterized by the n+p nonlinear equations:

$$\left.\begin{array}{l} \nabla f(x) - \sum_{j=1}^{p} \lambda_j \nabla h_j(x) = 0 \\ \\ h_j(x) = 0 \end{array}\right\} \tag{2.3.2}$$

with n+p variables $x_1,x_2,\ldots,x_n$, $\lambda_1,\lambda_2,\ldots,\lambda_p$. This technique of forming the Lagrange function (2.3.2) and solving the system (2.3.3) is called the Lagrange multiplier technique. By this method the dimensionality of the problem is considerably increased: the original problem (2.3.1) is a problem in E_n but the system (2.3.3) is a problem in E_{n+p}. However, more recent methods that reduce the solution of a constrained minimization problem to sequential minimization of a Lagrangian function as a function of x, exist. For further details in this subject the reader should refer to Fletcher (1969), Powell (1969), Lootsma (1972), Rockafeller (1974) and Van Der Hoek (1980).

4. NUMERICAL METHODS FOR UNCONSTRAINED OPTIMIZATION.

A particular interest is given to this and the next sections

of this chapter, for they represent the main area in which the research in this thesis has been undertaken. We shall first give a brief description of "Direct Search" methods, also known in the literature as "Ad-Hoc" methods, which will serve as an introduction to the development of "descent methods" on which the emphasis is to be placed.

4.1 Direct Search Methods.

The early direct search methods that were suggested for minimization are based on rough and ready ideas without much theoretical background. These methods only require the ability to evaluate the function at any given point and can be used for general continuous functions. In general however, they do not give a rapid rate of convergence and hence are inefficient for finding a minimum with high precision. There are nevertheless problems in which these features of an optimization method are not considered to be essential disadvantages.

If the problem is in only two or three dimensions, then it is possible that some sort of repeated bisection in each of the variables can be tried so as to establish a region in which the minimum lies. Then an attempt can be made to contract this region systematically. However, methods based on this sort of sectioning in n-dimensional problems have been suggested. A good review of this type of methods is given in Swann (1972).

Another possibility is to generate points $x^{(k)}$ at random in some fixed region, and select the one that gives the best function value over a large number of trials. Many variations of this

random search idea have been suggested in the literature. See for example, Swann (1972).

One of the most reliable direct search methods is the "Simplex Method". This is not the simplex method for linear programming but its name is derived from the same geometrical concept. In this method we first determine the vertex at which the function value is smallest (largest if a maximum is being sought); then we form a new simplex by reflecting this vertex in the centroid of the other n vertices. The function is then evaluated at this new vertex and the process is repeated. This method was first introduced by Spendley et al. (1962) and then Nelder and Mead (1965) suggested a modified simplex method that allows irregular simplexes. Amongst other modifications of the simplex method, we find those of Box (1965), Ward, Nag and Dixon (1969) and Dambrauskas (1970) and (1972). Versions of this method to deal with constraints have also been suggested. There seems to be a common belief at present that the Nelder and Mead variant of the simplex method is the most reliable direct search method in the presence of noise or error in the objective function and that it is reasonably efficient. Box and Draper (1969) stated that this algorithm is the most efficient of all current sequential techniques. For further details about these methods the reader should refer to Hans-Paul Schwefel (1981).

The "Alternative Variable Methods" form another class of direct search methods that operate as follows:- At iteration k, the k^{th} component of the point $x^{(k)}$ (or the variable x_k) alone is changed in an attempt to reduce the function value, and the other

components are kept fixed. After the n^{th} iteration, when all the variables have been changed, the whole cycle is repeated until convergence occurs. The strategy of Rosenbrock (1960b) of rotating coordinates and that of Hooke and Jeeves (1961) of pattern search belong to this class of methods. Further modifications have been proposed to the alternative variable methods, based on the observation that the points at the beginning and end of a cycle determine a line along which more substantial progress might be made. Thus, if provision is made for searching along this line (introduction to the idea of line searches), a more efficient method could result. The strategy of Davies, Swann and Campey (D.S.C.) that was originally described in Swann (1969), combines Rosenbrock's idea of rotating coordinates with one dimensional search methods, and the strategy of Powell (1964) of conjugate directions generates lines (or search directions) by joining up points in a similar way.

Versions of direct search methods that solve constrained problems are also available. For a full description of direct search methods, the reader should refer to Hans-Paul Schwefel (1981). Hans-Paul Schwefel (1981) together with Storey (1962) and Rosenbrock and Storey (1965) contain various examples of applications of direct search methods to both constrained and unconstrained optimization problems.

Descent methods that are described in the remaining part of this chapter, also generate search directions by joining up points in a similar way as described above, but have a stronger theoretical background and many of them have been developed to be more efficient.

Even when the gradient of the objective function and the solution of the one dimensional search problem are not computationally easily available, there exist very efficient numerical differentiation techniques and very efficient inexact line search procedures to overcome these difficulties. Therefore, the early direct search methods are gradually falling out of use, although in recent years there seems to be a new interest being given to these methods. The books of Hans-Paul Schwefel (1981) and Reklaitis, Ravindran and Ragsdell (1983) are a result of this recent interest.

4.2 Descent Methods.

We shall now consider methods that in addition to using function values also make use of the gradient of the objective function. Descent methods form a class of methods in which the solution of the general unconstrained minimization problem:

$$\text{Min } f(x) , \quad x \in \mathbb{R}^n \tag{2.4.1}$$

is found by solving a sequence of one dimensional problems. The conjugate direction methods are generally the most important descent methods for solving the general unconstrained minimization problem (2.4.1), from the point of view of practical computations. They assume that in the neighbourhood of the minimum, the function can be closely approximated by a positive definite quadratic form, and this is the major assumption made in the development of descent methods in general.

In the application of descent methods, the following situation is typical. We attempt to move from the current point $x^{(k)}$ in such a way as to reduce the value of the objective function f. Each

descent method is characterized by the provision of a descent vector $s^{(k)}$ at each iteration k (Search Direction), and the next point, $x^{(k+1)}$, is found by solving the one dimensional minimum problem, in which the function to be minimized is:

$$\Phi(\alpha) = f(x^{(k)} + \alpha s^{(k)}) \ . \tag{2.4.2}$$

Geometrically, we proceed in the direction $s^{(k)}$ to lower and lower level surfaces until $x^{(k+1)}$ is reached. As we cannot further reduce f in this direction, $s^{(k)}$ must lie in the tangent plane to the level surface through $x^{(k+1)}$. Therefore we have:

$$s^{(k)^T} g^{(k+1)} = 0 \tag{2.4.3}$$

where $g^{(k+1)} = \nabla f(x^{(k+1)})$ denotes the gradient of f at $x^{(k+1)}$. This situation is sketched in Figure 2.4.1 below:

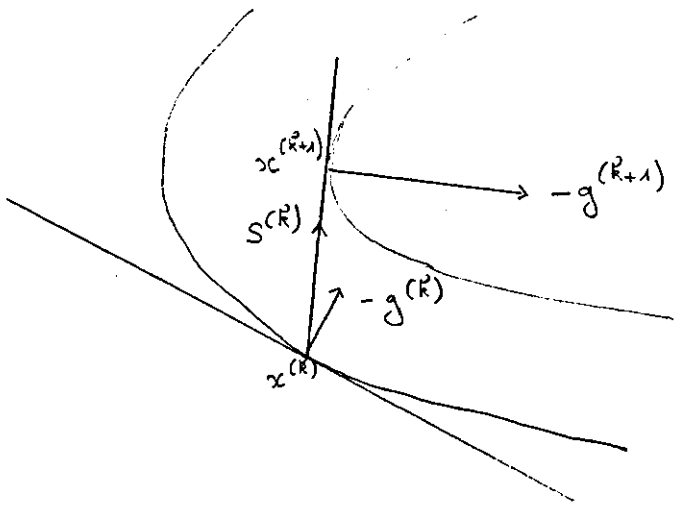

Figure 2.4.1 : The descent vector $s^{(k)}$ lies in the tangent plane to the level surface through $x^{(k+1)}$.

At iteration k of a descent method, the following property is satisfied:

$$s^{(k)^T} g^{(k)} < 0 \quad . \tag{2.4.4}$$

This property (2.4.4) is called the descent property and $s^{(k)}$ is said to be a descent (or downhill) vector.

In descent methods, the descent vector $s^{(k)}$ is usually a certain transformation of the gradient of f at $x^{(k)}$: $g^{(k)}$. That is to say that all descent methods use the basic iterative step:

$$x^{(k+1)} = x^{(k)} + \alpha \, Bg^{(k)} \tag{2.4.5}$$

where B is a matrix defining a transformation of the gradient and α is the step length in the direction $s^{(k)} = B.g^{(k)}$. The important question that arises at this stage is how to choose a "good" direction $s^{(k)}$, in other words how to choose a suitable transformation matrix B of the gradient $g^{(k)}$. In contrast to the one-dimensional case, where the only moves are in the positive or negative directions, even in two dimensions there are an infinite number of possible choices. In what follows we shall describe different methods for specifying $s^{(k)}$ and how to choose a suitable step length α to ensure a sufficient decrease in the function value. A descent method is usually named according to the procedure by which it computes the search direction, since the determination of the step length α is common to all descent methods and is usually viewed as a separate procedure (Line Search Procedure).

Before considering specific algorithms however, we shall first consider some necessary and sufficient conditions for the convergence of descent methods.

4.2.1 Convergence Theorems For Descent Methods.

The study of convergence for descent methods can be made considerably easier by considering a model algorithm. This will make it possible to carry out crucial reasoning in establishing the convergence of an algorithm, without becoming submerged in the clutter that is otherwise introduced by the great complexity of the subroutines used in modern algorithms. All descent methods have the basic general form of the model algorithm below:

The Model Algorithm For Descent Methods:-

Step 1 : If the convergence conditions are satisfied the algorithm terminates with the current point $x^{(k)}$ as the solution.

Step 2 : Compute a descent search direction $s^{(k)} \neq 0$.

Step 3 : Compute a step length $\alpha^{(k)} > 0$ satisfying $f(x^{(k)} + \alpha^{(k)} s^{(k)}) < f(x^{(k)})$.

Step 4 : $x^{(k+1)} = x^{(k)} + \alpha^{(k)} s^{(k)}$, $k = k+1$ and go to Step 1.

Because the search direction $s^{(k)}$ computed in Step 2 is a descent direction, it follows that there must exist $\alpha > 0$ such that $f(x^{(k)} + \alpha s^{(k)}) < f(x^{(k)})$. The requirement that $f(x^{(k+1)}) < f(x^{(k)})$ is not sufficient to ensure that the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ converges to a minimum of f. Ideally $\alpha^{(k)}$ is chosen to solve the one-dimensional minimization problem

$$\text{Min } f(x^{(k)} + \alpha s^{(k)}), \quad \alpha \in \mathbb{R}^+ - \{0\} \ , \tag{2.4.6}$$

but in practice however, an exact solution to problem (2.4.6)

is not usually possible and one accepts any value of $\alpha^{(k)}$ that satisfies certain standard conditions. The concept of exact and inexact line searches is investigated in subsection 4.2.2 below.

Let $x^{(1)}$ be an arbitrary starting point for a descent method of the form of the model algorithm above to solve the problem (2.4.1). The level set for this problem is therefore defined as follows:

$$\left\{x \in \mathbb{R}^n \ : f(x) \leq f(x^{(1)})\right\} \quad . \tag{2.4.7}$$

We also introduce:

$$\cos(\theta^{(k)}) = -\frac{g^{(k)^T} s^{(k)}}{\|g^{(k)}\| \cdot \|s^{(k)}\|} \quad , \tag{2.4.8}$$

where $\theta^{(k)}$ is the angle between the negative gradient $-g^{(k)}$ and the search direction $s^{(k)}$. For all descent methods $\cos(\theta^{(k)}) > 0$ holds. We now establish the following convergence theorem.

Theorem 2.4.1 (Zoutendijk (1976)).

Suppose

(1) f is twice continuously differentiable in (2.4.7);

(2) (2.4.7) is bounded;

(3) The Hessian matrix H(x) of second partial derivatives of f is bounded in (2.4.7);

(4) $\alpha^{(k)} = \underset{\alpha>0}{\text{Arg Min}}\ f(x^{(k)} + \alpha s^{(k)})$

(5) $\Sigma_k \cos^2(\theta^{(k)}) = +\infty$.

Then $\nabla f(\bar{x}) = 0$ for at least one point of accumulation $\bar{x}$ of the sequence $\{x^{(k)}\}$ generated by the model algorithm.

Theorem 2.4.2 (Zoutendijk (1976)).

If in Theorem (2.4.1) we replace condition (5) by the following condition:

$$(5')\ \exists\, k' \in \mathbb{N} \text{ and } \exists\, \theta > 0 : \forall\, k \geq k' \quad \cos(\theta^{(k)}) \geq \theta > 0,$$

then $\nabla f(\bar{x}) = 0$ for all points of accumulation $\bar{x}$ of the sequence $\{x^{(k)}\}$ generated by the model algorithm.

Proofs of Theorems 2.4.1 and 2.4.2.

Let us first consider the proof of Theorem 2.4.1. Suppose that for all points of accumulation $\bar{x}$ of the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ generated by the model algorithm above, we have $\nabla f(\bar{x}) \neq 0$. Then there exists a constant ε say, such that:

$$\|\nabla f(x^{(k)})\| \geq \varepsilon > 0 \qquad \text{for all } k \quad . \tag{2.4.9}$$

It is no restriction and no loss of generality to assume that $\|s^{(k)}\| = 1$. Let $\mu^{(k)}$ be the first value of α such that

$$\nabla f(x^{(k)} + \mu^{(k)} s^{(k)})^T s^{(k)} = \frac{1}{2} \nabla f(x^{(k)})^T s^{(k)} \quad . \tag{2.4.10}$$

Hence $0 < \mu^{(k)} < \alpha^{(k)}$. Now, if Ω is an upper bound on $\|\nabla^2 f(x)\|$, where x is any point in the level set (2.4.7), then we have

$$\nabla f(x^{(k)} + \mu^{(k)} s^{(k)})^T s^{(k)} \leq \nabla f(x^{(k)})^T s^{(k)} + \Omega \mu^{(k)} \|s^{(k)}\|^2 \quad . \tag{2.4.11}$$

Therefore using (2.4.10) and the assumption that $\|s^{(k)}\| = 1$, we obtain:

$$\mu^{(k)} \geq - \frac{\nabla f(x^{(k)})^T s^{(k)}}{2\Omega} \quad . \tag{2.4.12}$$

Now, we also have:

$$f(x^{(k)} + \mu^{(k)} s^{(k)}) = f(x^{(k)}) + \mu^{(k)} \nabla f(x^{(k)} + \lambda\mu^{(k)} s^{(k)})^T s^{(k)} \tag{2.4.13}$$

where $0 \leq \lambda \leq 1$.

But $f(x^{(k+1)}) = f(x^{(k)} + \alpha^{(k)} s^{(k)}) < f(x^{(k)} + \mu^{(k)} s^{(k)})$, and therefore we have:

$$f(x^{(k)}) - f(x^{(k+1)}) > f(x^{(k)}) - f(x^{(k)} + \mu^{(k)} s^{(k)})$$

$$= - \mu^{(k)} \nabla f(x^{(k)} + \lambda\mu^{(k)} s^{(k)})^T s^{(k)} \quad .$$

Now, because $\lambda \leq 1 \Rightarrow \lambda\mu^{(k)} \leq \mu^{(k)}$ and therefore:

$$- \nabla f(x^{(k)} + \lambda\mu^{(k)} s^{(k)})^T s^{(k)} \leq \nabla f(x^{(k)} + \mu^{(k)} s^{(k)})^T s^{(k)} \tag{2.4.14}$$

And therefore using (2.4.12) and (2.4.14) we obtain:

$$f(x^{(k)}) - f(x^{(k+1)}) > \frac{1}{2\Omega} \nabla f(x^{(k)})^T s^{(k)} . \nabla f(x^{(k)} + \mu^{(k)} s^{(k)})^T s^{(k)} \quad . \tag{2.4.15}$$

By substituting (2.4.10) and (2.4.8) into (2.4.15) we obtain:

$$f(x^{(k)}) - f(x^{(k+1)}) > \frac{1}{4\Omega} \cos^2(\theta^{(k)}) \; \|\nabla f(x^{(k)})\|^2 \quad ,$$

and finally by using (2.4.9) we get:

$$f(x^{(k)}) - f(x^{(k+1)}) > \frac{\varepsilon^2 \cos^2(\theta^{(k)})}{4\Omega} \quad .$$

Since $\Sigma_k \cos^2(\theta^{(k)}) = \infty$, it follows that $\Sigma_k \left\{ f(x^{(k)}) - f(x^{(k+1)}) \right\} = \infty$ which is impossible because $x^{(k)}$ is in the bounded level set (2.4.7). Hence, for at least one point of accumulation $\bar{x}$ of the sequence $\{x^{(k)}\}_{k \in \mathbb{N}}$ generated by the model algorithm above we have $\nabla f(\bar{x}) = 0$.

Let us now assume that condition (5) of Theorem 2.4.1 is replaced by condition (5') of Theorem 2.4.2 and prove Theorem 2.4.2.

Let $\bar{x}$ be a point of accumulation and $\{x^{(k_\ell)}\}_{\ell \in \mathbb{N}}$, a subsequence of the sequence $\{x^{(k)}\}_{k \in \mathbb{N}}$, converging to $\bar{x}$. Suppose there exists $\varepsilon > 0$ such that (2.4.9) is true for all k. Therefore for a sufficiently large $\ell \in \mathbb{N}$ we have:

$$\|\nabla f(x^{(k_\ell)})\| \geq \frac{1}{2}\varepsilon .$$

Let $k_\ell \geq k'$ (k' sufficiently large). It then follows from the proof of Theorem (2.4.1) that:

$$f(x^{(k_\ell)}) - f(x^{(k_\ell+1)}) \geq \frac{(\nabla f(x^{(k_\ell)})^T s^{(k_\ell)})^2}{4\Omega}$$

$$= \frac{\cos^2(\theta^{(k_\ell)})\|\nabla f(x^{(k_\ell)})\|^2}{4\Omega}$$

$$\geq \frac{\theta^2\varepsilon^2}{16\Omega} .$$

Hence $\Sigma_k\left\{f(x^{(k_\ell)}) - f(x^{(k_\ell+1)})\right\} = \infty$ which is again impossible for the same reason as above.

The conditions for Theorem 2.4.1 to hold are quite reasonable. The second condition prevents infinite solutions and in particular convergence of $f(x^{(k)})$ to a finite value while $x^{(k)} \to \pm\infty$. This will exclude functions such as e^{-x} that are bounded below but strictly decreasing as $\|x\|$ becomes infinite. The third condition, that can only be eligible to hold if the first condition holds, puts a restriction on the curvature of the objective function. It can be expected that functions with unbounded second partial derivatives

will present special problems. We shall not comment on the fourth condition at this stage since a full investigation on how to compute the step lengths $\alpha^{(k)}$ is given in the following subsection 4.2.2. Finally, the fifth condition states that the angle between the negative gradient and the search direction should not go to $\pi/2$ too rapidly. This is strengthened in Theorem 2.4.2. There must be a limit on the closeness of $s^{(k)}$ to orthogonality to the negative gradient. This prevents $s^{(k)}$ to be chosen so that to first order, f is almost constant along $s^{(k)}$ which will only occur if $s^{(k)}$ is almost parallel to the first-order approximation to the contour line $f(x) = f(x^{(k)})$, so that the negative gradient and the search direction are almost orthogonal (i.e. $-\nabla f(x^{(k)})^T s^{(k)} / \|\nabla f(x^{(k)}\| \|s^{(k)}\|$ is close to zero). The stronger requirement (5') implies strong convergence ($\nabla f(\bar{x}) = 0$ for all points of accumulation); condition (5) only implies weak convergence (there exists a point of accumulation $\bar{x}$ with $\nabla f(\bar{x}) = 0$). If there is only one point of accumulation, which is usually the case in practice, then it does not make any difference; if there is more than one point of accumulation, then they obviously all have the same value.

The type of convergence result, that has just been established, is termed "Global Convergence", since there is no restriction on the closeness of $x^{(1)}$ to a stationary point. For further details on the study of global convergence we refer to Ostrowski (1966), Wolfe (1969), Sargent and Sebastian (1973), Polak, Sargent and Sebastian (1974), Zoutendijk (1970) and (1976), Fletcher (1980) and Gill, Murray and Wright (1981).

Even when it can be proved theoretically that a sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ generated by the model algorithm above will converge in the limit to the required minimum, a method will be practicable only if convergence occurs with some rapidity. Global convergence, however, does not give a measure of this rapidity and hence gives practically no idea on how efficient the method is. In what follows we shall briefly discuss some means of characterizing the rate of convergence at which such sequences converge.

Let $\{x^{(k)}\}_{k\in\mathbb{N}}$ be a sequence of points generated by the model algorithm above, converging to x^*. To simplify the discussion, we shall assume that the elements of this sequence are distinct, and that for no value of k does $x^{(k)}$ equal x^*.

An effective technique for judging the rate of convergence is to compare the improvements at two consecutive steps, i.e., to measure the closeness of $x^{(k+1)}$ to x^* relative to that of $x^{(k)}$ to x^*.

A sequence of $\{x^{(k)}\}_{k\in\mathbb{N}}$ is said to converge with order r, if r is the largest number such that

$$0 \leq \overline{\lim_{k\to\infty}} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|^r} < \infty \qquad (2.4.16)$$

r is usually known as the asymptotic rate of convergence. If $r = 1$, the sequence is said to have linear convergence and if $r = 2$, it is said to have quadratic convergence.

If the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ has order of convergence r, the asymptotic error constant is the value γ that satisfies:

$$\gamma = \overline{\lim_{k\to\infty}} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|^r} \quad . \qquad (2.4.17)$$

When $r = 1$, γ must be strictly less than 1 in order for convergence to occur.

If a sequence has linear convergence, the step-wise decrease in $\| x^{(k)} - x^* \|$ varies substantially with the value of the asymptotic error constant. If the latter is zero when $r = 1$, the associated type of convergence is given the name superlinear convergence. Note however, that any value of r greater than 1 implies superlinear convergence. For more detailed studies of rates of convergence we refer to Polak (1971), Wolfe (1978) and Gill, Murray and Wright (1981).

4.2.2 Line Search Techniques.

Until the mid sixties, the prevailing belief was that the step lengths $\alpha^{(k)}$ should be chosen to solve the one-dimensional minimization problem:

$$\underset{\alpha>0}{\text{Min}} \ \Phi(\alpha) \tag{2.4.18}$$

accurately, where Φ is as defined in (2.4.2). However, more careful computational testing has led to a turnabout. In this subsection we shall describe some of the techniques that only impose some weak acceptance criteria on $\alpha^{(k)}$, that lead to methods that perform as well in theory and better in practice.

In the proof of Theorem (2.4.1), we defined $\mu^{(k)}$ as the first value of α such that (2.4.10) holds. $\mu^{(k)}$ can also be defined by

$$\nabla f(x^{(k)} + \mu^{(k)} s^{(k)})^T s^{(k)} = \sigma \nabla f(x^{(k)})^T s^{(k)},$$

where $0 < \sigma < 1$. In this case we have:

$$\mu^{(k)} \geq - \frac{1-\sigma}{\Omega} \cos(\theta^{(k)}) . \| \nabla f(x^{(k)}) \| \quad .$$

It then follows that convergence can still be maintained even if we do not require $\alpha^{(k)}$ to be the solution of the one-dimensional minimization problem (2.4.18) but only require that $\alpha^{(k)} \geq \mu^{(k)}$. Hence, if during the line search an α is found with:

$$\nabla f(x^{(k)} + \alpha s^{(k)})^T s^{(k)} \geq \sigma \nabla f(x^{(k)})^T s^{(k)} \tag{2.4.19}$$

we may stop the search for α.

A frequently used method for determining the step lengths $\alpha^{(k)}$ is to estimate a local minimizer of $\Phi(\alpha)$. Then $\alpha^{(k)}$ satisfies at least approximately the requirement:

$$\alpha^{(k)} = \underset{\alpha>0}{\text{Arg Min}}\ f(x^{(k)} + \alpha s^{(k)}) \quad . \tag{2.4.20}$$

In other words, $\alpha^{(k)}$ satisfies at least approximately the requirement:

$$\Phi'(\alpha) = 0, \tag{2.4.21}$$

where Φ' is the first derivative of Φ.

In general (2.4.21) is a nonlinear equation and so it cannot usually be solved analytically. Therefore a numerical method for estimating a value of α that satisfies (2.4.21) must be used. When $\alpha^{(k)}$ is chosen to satisfy (2.4.20) accurately, the procedure is termed an "Exact Line Search" and when (2.4.20) is only satisfied approximately, the procedure is termed an "Inexact Line Search".

A fundamental requirement for a step length algorithm associated with a descent method involves the change in f at each iteration. If convergence is to be assured, the step length must produce a "Sufficient" or"Satisfactory" decrease in f(x). Although it seems to be common sense to require that:

$$f(x^{k+1}) < f(x^{(k)}) \ , \qquad (2.4.22)$$

it comes as no great surprise that this simple condition does not guarantee that the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ will converge to a minimizer of f. Simple counter examples to prove this statement can be found in most good text books dealing with line searches and step lengths. For example see Gill, Murray and Wright (1981) and Dennis and Schnabel (1983).

The "Sufficient" or "Satisfactory" decrease in f(x) requirement is usually satisfied by several alternative sets of conditions on $\alpha^{(k)}$. Although the process of finding $\alpha^{(k)}$ by estimating a minimizer of Φ produces a satisfactory decrease in the value of f at each iteration of a descent algorithm, the line search is computationally expensive, in that several evaluations of f or ∇f are required; in fact the major part of the computational labour required to implement many descent algorithms is that required to determine $\alpha^{(k)}$. The computational efficiency of the line search therefore usually determines the computational efficiency of the whole descent algorithm. Consequently a great deal of effort has been expended upon the construction of efficient line search procedures. Most of these procedures are based on the principles of Goldstein (1962), (1965) and (1967) and Armijo (1966) from which the well known Goldstein-Armijo principle is derived.

The Goldstein-Armijo principle is as follows:
A sufficient decrease in f(x) is achieved if $\alpha^{(k)}$ satisfies:

$$\alpha^{(k)}\, q_2\, \nabla f(x^{(k)})^T s^{(k)} \leq f(x^{(k+1)}) - f(x^{(k)}) \leq \alpha^{(k)}\, q_1\, \nabla f(x^{(k)})^T s^{(k)} \qquad (2.4.23)$$

where q_1 and q_2 are scalars satisfying $0 < q_1 < q_2 < 1$ and $q_1 = 1 - q_2$. The upper and lower bounds of (2.4.23) ensure that $\alpha^{(k)}$ is neither too large nor too small.

In typical algorithms based on (2.4.23), the trial values of the step length are defined in terms of an initial step $\alpha^{(1)}$ and a scalar ω. The value of $\alpha^{(k)}$ is taken as the first member of the sequence $\{\omega^j \alpha^{(1)}\}$, $j = 0,1,\ldots$, for which (2.4.23) is satisfied for some q_1 and q_2.

The performance of these algorithms depends critically on the choice of $\alpha^{(1)}$, rather than on any merits of condition (2.4.23). Step length algorithms of this type perform well only for descent methods in which an a priori value of $\alpha^{(1)}$ tends to be a good step, so that only the first member of the sequence $\{\omega^j \alpha^{(1)}\}$ usually needs to be computed. We must emphasize however, that condition (2.4.23) alone does not guarantee a good value of $\alpha^{(k)}$. Although this strategy would be "efficient" in that a suitable $\alpha^{(k)}$ would be found with only a single function evaluation per iteration, any descent method that uses such a step length algorithm would be extremely inefficient. It is important, for the sake of the computational labour required, to minimize the number of function evaluations, but it is also essential to consider the performance of a step length algorithm not merely in terms of that, but also in terms of the overall reduction in f achieved at each step.

Obviously there is a tradeoff between the effort expended to determine a good $\alpha^{(k)}$ at each iteration and the resulting benefits for the descent method. The balance varies with the type of algorithm as well as the problem to be solved, and hence some

flexibility is desirable in specifying the conditions to be satisfied by $\alpha^{(k)}$.

Fletcher (1980) suggests that $\alpha^{(k)}$ is such that $x^{(k+1)}$ satisfies the condition:

$$|g^{(k+1)^T} s^{(k)}| \leq -\sigma g^{(k)^T} s^{(k)} , \qquad (2.4.24)$$

together with the Goldstein (1965) requirement that:

$$f(x^{(k+1)}) \leq f(x^{(k)}) + \rho\alpha^{(k)} g^{(k)^T} s^{(k)} \qquad (2.4.25)$$

where $g^{(k)} = \nabla f(x^{(k)})$ is the gradient vector at $x^{(k)}$, and where $\rho \in (0, \frac{1}{2})$, $\sigma \in (0, 1)$ and $\rho < \sigma$.

Condition (2.4.24) ensures that $\alpha^{(k)}$ is not too small. The value of σ determines the accuracy with which $\alpha^{(k)}$ approximates a stationary point of f along $s^{(k)}$, and consequently provides a means of controlling the balance of effort to be expended in computing $\alpha^{(k)}$. The case of exact line search occurs when $\sigma = 0$, condition (2.4.25) requires that $f(x^{(k)} + \alpha s^{(k)})$ lies on or below the line $\ell(\alpha) = f(x^{(k)}) + \rho\alpha g^{(k)^T} s^{(k)}$, and ensures a sufficient decrease when used together with (2.4.24).

An advantage of condition (2.4.24) as an acceptance criterion is that its interpretation in terms of a local minimizer suggests efficient methods for computing a good value of $\alpha^{(k)}$. In particular safeguarded polynomial fitting techniques for univariate minimization converge very rapidly on well behaved functions. A full description of these techniques can be found in Gill, Murray and Wright (1981).

A similar procedure is described in Dennis and Schnabel(1983). They give an algorithm of backtracking line search procedure using

quadratic and cubic interpolation.

One of the most recent investigations of line search procedures, is due to Al-Baali and Fletcher (1986). Their investigation is mainly aimed at developing a line search method that is applicable to the nonlinear least-squares problem in which f(x) is a sum of squares of nonlinear functions;

$$f(x) = \frac{1}{2} \sum_{i=1}^{m} (r_i(x))^2 \quad . \tag{2.4.26}$$

If $r \in \mathbb{R}^m$ denotes the column vector whose elements are the functions $r_i(x)$ and $A = \nabla r^T$ is the Jacobian matrix of r, then it follows that:

$$g(x) = A(x).r(x) \quad . \tag{2.4.27}$$

Thus, to evaluate f(x) requires an evaluation of r(x), and to evaluate g(x) requires in addition an evaluation of the Jacobian matrix.

Their work consisted of a study of various iterative schemes for the line search sub-problem that guarantee finding an acceptable step length α in a finite number of steps. This is achieved by first bracketing an interval of acceptable α values, and then reducing this bracket uniformly by repeated sectioning of the bracket in a systematic way. These schemes for the line search sub-problem include a scheme for finding an acceptable value of α that satisfies (2.4.19) together with the Goldstein (1965) requirement (2.4.25). This is then generalized to find an acceptable value of α that satisfies (2.4.21) and (2.4.25) * that of Fletcher (1980). Further modifications were also suggested with the aim of producing

* as well as

schemes in which the gradient vector is evaluated as infrequently as possible, on the assumption that this is the major cost in using the methods.

Their work shows in particular that substantial gains in efficiency can be obtained in nonlinear least-squares problems by making polynomial approximations to the individual functions r_i, rather than the overall function f of (2.4.26).

For more details on line search techniques, we refer to Wolfe (1978), Fletcher (1980), Gill, Murray and Wright (1981) and Dennis and Schnabel (1983).

Having established some convergence results and described some line search techniques for descent methods, we are now left with the question of how to choose a suitable descent direction of search $s^{(k)}$. As mentioned earlier, a descent method is usually named according to the procedure by which it computes the search direction. Therefore, describing a procedure for determining a search direction is in fact describing a specific descent method. Some of these methods are described below.

4.2.3 The Method of Steepest Descent.

One of the oldest and simplest methods for finding a solution to the general minimization problem (2.4.1) is the method of steepest descent. This method was originally proposed by Cauchy (1847) for the solution of systems of nonlinear equations.

Consider the linear approximation to f based on the Taylor series expansion about $x^{(k)}$:

$$f(x^{(k)} + s) \approx f(x^{(k)}) + g^{(k)^T}.s \qquad (2.4.28)$$

Assuming that a step of unity will be taken along s, it would appear that a good way to achieve a large reduction in f(x) is to choose s so that $g^{(k)^T}s$ is large and negative. Obviously some normalization must be imposed on s, otherwise, for any $\bar{s}$ such that $g^{(k)^T}\bar{s} < 0$, one would simply choose s as an arbitrary large positive multiple of $\bar{s}$. The aim, however, is to choose s so that, amongst all suitably normalized vectors, $g^{(k)^T}s$ is minimum.

Given some norm $\|\cdot\|$, $s^{(k)}$ is thus the solution of the minimization problem:

$$\underset{s\in\mathbb{R}^n}{\text{Min}} \quad \frac{g^{(k)^T}s}{\|s\|} \quad . \qquad (2.4.29)$$

The solution of the problem (2.4.29) depends on the choice of norm. When the norm is defined by a given symmetric positive definite matrix, C say, (i.e. $\|s\|_C = (s^T C\, s)^{\frac{1}{2}}$, the solution of (2.4.29) is:

$$s^{(k)} = - C^{-1}\, g^{(k)} \quad . \qquad (2.4.30)$$

The matrix B defined in (2.4.5) used as a transformation of the gradient is in this case

$$B = - C^{-1} \quad .$$

If the two-norm is used, (i.e. $s = (s^T s)^{\frac{1}{2}}$), the solution of (2.4.29) is just the negative gradient:

$$s^{(k)} = - g^{(k)} \quad . \qquad (2.4.31)$$

In this case we have B = - I, where I is the identity matrix. Because (2.4.31) solves (2.4.29) with respect to the two norm,

the negative gradient direction $s^{(k)}$ in (2.4.31) is termed the direction of steepest descent and the algorithm using this direction at every iteration is called the steepest descent algorithm.

Unless the gradient vanishes, the steepest descent direction is clearly a descent direction, since the vectors $s^{(k)}$ and $g^{(k)}$ are bounded away from orthogonality and the directional derivative is such that

$$g^{(k)^T} s^{(k)} = - g^{(k)^T} g^{(k)} < 0 \quad .$$

Consequently any of the line search techniques described in 4.2.2 above, may be combined with the steepest descent algorithm to yield a method with guaranteed global convergence.

Unfortunately, a proof of global convergence for an algorithm does not ensure that it is an efficient method. Although this method is useful for a large class of well conditioned problems, experience has shown that it can be extremely slow. Kowalik and Osborne (1968) discuss some of the reasons for this inefficiency. This weakness of the method is mainly due to the fact that the search directions generated by the algorithm are not linearly independent. However, the fact that the method only uses successive linear approximations to the objective function is another factor in its inefficiency. The steepest descent method is shown to have at best a linear rate of convergence. Proofs of global and local convergence can be found in Gill, Murray and Wright (1981).

4.2.4 The Method of Newton.

This is another standard method for the solution of the general minimization problem (2.4.1). The method of Newton uses successive quadratic approximations to the objective function based on the Taylor series expansion, as opposed to the linear approximation (2.4.28) used for the method of steepest descent.

The quadratic approximation to f based on the Taylor series expansion about $x^{(k)}$ is as follows:

$$f(x^{(k)} + s) \approx f(x^{(k)}) + g^{(k)^T} s + \frac{1}{2} s^T H^{(k)} s \qquad (2.4.32)$$

where $H^{(k)} = \nabla^2 f(x^{(k)})$ is the matrix of second partial derivatives at $x^{(k)}$ (or Hessian matrix). This quadratic approximation of f(x) is used to form an iteration sequence by forcing $x^{(k+1)}$, the next point in the sequence, to be a point where the gradient of the approximation is zero. Therefore:

$$g^{(k)} + H^{(k)} s^{(k)} = 0 \quad , \qquad (2.4.33)$$

and hence,

$$s^{(k)} = - H^{(k)^{-1}} g^{(k)} \quad . \qquad (2.4.34)$$

Accordingly, this successive quadratic approximation scheme produces the method of Newton for Optimization:

$$x^{(k+1)} = x^{(k)} - H^{(k)^{-1}} g^{(k)} \quad . \qquad (2.4.35)$$

In this case the transformation matrix B is:

$$B = - H^{(k)^{-1}}$$

and the step length $\alpha^{(k)}$ is always taken equal to 1. Note however, that the direction (2.4.34) is not necessarily a descent direction

unless the Hessian matrix is positive definite. To see this more clearly, assume the current point $x^{(k)}$ is not a stationary point (i.e. $g^{(k)} \neq 0$), and examine the projection of the Newton direction (2.4.34) onto the local gradient. By definition of a descent direction, condition (2.4.4) must hold. This leads to:

$$- g^{(k)^T} H^{(k)^{-1}} g^{(k)} < 0 \quad . \tag{2.4.36}$$

Clearly this condition is satisfied if $H^{(k)}$ is positive definite. If a point is encountered where $H^{(k)}$ is negative definite, the direction is an ascent direction; and when the Hessian is indefinite, there is no assurance of ascent or descent directions.

If the Hessian matrix $H^{(k)}$ is positive definite, Newton's method will minimize a quadratic function, from any starting point, in exactly one step. Therefore, we expect good convergence from the method when the quadratic model (2.4.32) is accurate. Mangasarian (1971) has shown, under mild regularity conditions on f(x), that Newton's method exhibits a quadratic rate of convergence.

4.2.5 The Modified Newton's Method.

Experience has shown that Newton's method can be unreliable for non quadratic functions. The Newton step ($\alpha^{(k)} = 1$) will often be large when the starting point is far from the solution x^*, and there is a real possibility of divergence. It is however, possible to rectify the method in a logical and simple way to ensure a descent property to hold and to achieve a sufficient decrease in f(x) at each iteration by combining the method with a line search technique as in the method of steepest descent. That is we form

the sequence of iterates:

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} H^{(k)^{-1}} . g^{(k)} \qquad (2.4.37)$$

by choosing $\alpha^{(k)}$ either by exact line search when possible or inexact line search techniques.

This is a modified Newton's method. This method is reliable and efficient when first and second derivatives are accurate and computed inexpensively. The major difficulty in this method and in Newton's method is the need to calculate and solve the linear equation involving the Hessian matrix $H^{(k)}$ at each iteration.

Experience has also shown that the method of steepest descent tends to be effective far from the minimum but becomes less so as x^* is approached, whereas the method of Newton can be unreliable far from x^* but is very efficient as $x^{(k)}$ approaches the minimum.

4.2.6 Quasi-Newton Methods.

The theory of Quasi-Newton methods is based on the fact that an approximation to the curvature of a nonlinear function can be computed, without explicity forming the Hessian matrix, using only first order information. These methods generate directions of search of the form:

$$s^{(k)} = - A^{(k)} . g^{(k)} \quad , \qquad (2.4.38)$$

where $A^{(k)}$ is an n×n matrix called the metric. Methods that use directions of this form are often called "Variable Metric Methods", because $A^{(k)}$ changes at each iteration. To be precise, a variable metric method is a quasi-Newton method if it is designed so that

the iterates on a quadratic function $f(x) = a + b^T x + \frac{1}{2} x^T C x$ satisfy the condition:

$$\Delta x = C^{-1}.\Delta g \quad . \tag{2.4.39}$$

Unfortunately, the literature is not precise or consistent in the use of these terms (see Avriel (1976)), therefore, we shall use these terms interchangeably. This is appropriate, since both expressions are of equal importance in the design and execution of these methods.

An iteration of a variable metric algorithm for minimizing a differentiable function f takes the following form:

Given an estimate $x^{(k)}$ of the required minimizer and a positive definite matrix $A^{(k)}$, which is an approximation to the inverse of the Hessian matrix $H^{(k)}$, apply the formula:

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} A^{(k)}.g^{(k)} \quad , \tag{2.4.40}$$

where $\alpha^{(k)}$ is a step length chosen by a line search procedure. Then calculate a positive definite matrix $A^{(k+1)}$, depending only on $A^{(k)}$ and the vectors:

$$\left.\begin{aligned} \Delta x^{(k)} &= x^{(k+1)} - x^{(k)} \\ \Delta g^{(k)} &= g^{(k+1)} - g^{(k)} \end{aligned}\right\} \tag{2.4.41}$$

that obeys the quasi-Newton condition:

$$\Delta x^{(k)} = A^{(k+1)} \Delta g^{(k)} \quad . \tag{2.4.42}$$

Various formulae to compute the successive approximations to the inverse of $H^{(k)}$, $A^{(k)}$, that satisfy the quasi-Newton condition (2.4.42), have been proposed in the literature. Two of the most

commonly used formulae are given below:

The Davidon, Fletcher and Powell : (1959) and (1963). (D.F.P)

The D.F.P. method is the first method of this type to be used. Ituses the following approximation matrix at each iteration, starting with the identity matrix I:

$$A^{(k+1)} = A^{(k)} + \frac{\Delta x^{(k)} . \Delta x^{(k)^T}}{\Delta x^{(k)^T} . \Delta g^{(k)}} - \frac{A^{(k)} . \Delta g^{(k)} . \Delta g^{(k)^T} . A^{(k)}}{\Delta g^{(k)^T} . A^{(k)} . \Delta g^{(k)}} \quad . \qquad (2.4.43)$$

The Broyden, Fletcher, Goldfarb and Shanno : (1970). (B.F.G.S)

The B.F.G.S. method uses the following approximation matrix at each iteration, starting with the identity matrix I:

$$A^{(k+1)} = \left[I - \frac{\Delta x^{(k)} . \Delta g^{(k)^T}}{\Delta x^{(k)^T} \Delta g^{(k)}} \right] . A^{(k)} . \left[I - \frac{\Delta g^{(k)} . \Delta x^{(k)^T}}{\Delta x^{(k)^T} \Delta g^{(k)}} \right] +$$

$$+ \frac{\Delta x^{(k)} . \Delta x^{(k)^T}}{\Delta x^{(k)^T} \Delta g^{(k)}} \quad . \qquad (2.4.44)$$

Since a stationary point x^* of f is a strong local minimizer if the Hessian matrix at x^* is positive definite , it would seem desirable for the approximating matrices $\{A^{(k)}\}$ to be positive definite . Furthermore, if $A^{(k)}$ is positive definite, the local quadratic model has a unique local minimum and the search direction $s^{(k)}$ computed from (2.4.38) is a descent direction. Thus it is usual to require that the update formulae possess the property of hereditary positive definiteness, i.e. if $A^{(k)}$ is positive definite, so is $A^{(k+1)}$. The positive definiteness of the Hessian

or inverse Hessian approximation can be lost through rounding errors. However, this can be eliminated by updating the Cholesky factors of the matrix, rather than an explicit representation of the elements of the matrix.

If a twice continuously differentiable function f has bounded second derivatives and the level set (2.4.7) is bounded, global convergence can be proved for a quasi-Newton method if every $A^{(k)}$ is positive definite with a bounded condition number, and if one of the line search procedures described in 4.2.2 is used to choose $\alpha^{(k)}$. The restrictions on $A^{(k)}$ ensure that the search direction remains sufficiently bounded away from orthogonality with the negative gradient. The matrices $\{A^{(k)}\}$ generated by an implementation that updates the Cholesky factors can be made to satisfy these requirements by including a procedure that explicitly bounds the condition number.

In considering whether quasi-Newton methods converge superlinearly, we note that the following property holds for a superlinearly convergent sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$:

$$\lim_{k\to\infty} \frac{\|x^{(k+1)} - x^{(k)}\|}{\|x^{(k)} - x^*\|} = 0 \quad . \tag{2.4.45}$$

If $\alpha^{(k)} = 1$ for all k, and the sequence $\{x^{(k)}\}_{k\in\mathbb{N}}$ converges locally and linearly, convergence is superlinear if and only if the sequence $\{A^{(k)}\}$ satisfies:

$$\lim_{k\to\infty} \frac{\|(A^{(k)} - \bar{H}(x^*))(x^{(k+1)} - x^{(k)})\|}{\|x^{(k+1)} - x^{(k)}\|} = 0 \quad . \tag{2.4.46}$$

One of the most interesting properties of these updates is that it is not necessary for $A^{(k)}$ to converge to $H^{-1}(x^*)$ in order for (2.4.46) to be satisfied, and therefore the iterates with $\alpha^{(k)} = 1$ for all k can converge superlinearly even if the Hessian approximation does not converge to the true Hessian. The property (2.4.46) can be verified for both the D.F.P. and the B.F.G.S. updates under suitable assumptions. However, if step lengths other than unity are allowed, the sequence $\{\alpha^{(k)}\}_{k\in\mathbb{N}}$ must converge to unity at a sufficiently fast rate if superlinear convergence is to be verified. This is the reason for the frequent advice in the literature to initiate a step length procedure for a quasi-Newton method with a full Newton step of unity.

Most of the work on convergence of quasi-Newton methods is due to Broyden (1967) and (1970), Broyden, Dennis and Moré (1973), Powell (1971), (1975) and (1976a). The characterization (2.4.45) of superlinear convergence is due to Dennis and Moré (1974).

4.2.7 Conjugate Gradient Methods.

We have seen above that the weakness of the steepest descent method is due to the fact that the search directions generated by the method are not linearly independent and that the major difficulty in the Newton and modified Newton methods is the need to compute and solve the linear equation involving the inverse of the Hessian matrix $H^{(k)}$ at each iteration k. This suggests that a good method will have the characteristics that any t consecutive search directions ($t \leq n$) are linearly independent, and the curvature information (provided by the Hessian matrix) will be

built up as the iterations proceed, using the observed behaviour of $f(x^{(k)})$ and $g^{(k)}$. The family of conjugate direction methods combines these two characteristics and provides the most useful general-purpose minimization methods currently available. It has been recently shown that the search directions generated by quasi-Newton methods are mutually conjugate and therefore this class of methods belong to the family of conjugate direction methods. Although quasi-Newton methods are very efficient in terms of their superlinear convergence their main drawback is the need to store an n×n matrix for the approximation of the inverse of the Hessian and the fair amount of housekeeping labour needed to compute these approximations.

In contrast to most methods for unconstrained optimization, conjugate gradient methods form a class of methods that generate directions of search without storing any matrix. They are essential in circumstances when methods based on matrix factorization are not viable because the relevant matrices are too large or too dense.

Conjugate gradient methods were originally introduced by Hestenes and Stiefel (1952) for the solution of systems of linear equations and are therefore based on the assumption that in the neighbourhood of a nondegenerate minimum point, a function behaves like a quadratic function.

The search directions generated by conjugate gradient algorithms are of the form:

$$s^{(k)} = \begin{cases} -g^{(k)} & \text{if } k = 1 \\ -g^{(k)} + \beta^{(k)} s^{(k-1)} & \text{if } k > 1 \end{cases} . \qquad (2.4.47)$$

Various formulae in $\beta^{(k)}$ that ensure conjugacy of successive search directions have been suggested in the literature. Some of the better known of these formulae are given below:

Hestenes and Stiefel (1952):

$$\beta^{(k)} = \frac{(g^{(k+1)} - g^{(k)})^T g^{(k+1)}}{(g^{(k+1)} - g^{(k)})^T s^{(k)}} \quad . \tag{2.4.48}$$

Fletcher and Reeves (1964):

$$\beta^{(k)} = \frac{g^{(k+1)^T} g^{(k+1)}}{g^{(k)^T} . g^{(k)}} \quad . \tag{2.4.49}$$

Polak and Ribière (1969):

$$\beta^{(k)} = \frac{g^{(k+1)^T} (g^{k+1)} - g^{(k)})}{g^{(k)^T} . g^{(k)}} \quad . \tag{2.4.50}$$

Dixon (1972) attributes the following formula to Myers:

$$\beta^{(k)} = - \frac{g^{(k+1)^T} g^{(k+1)}}{s^{(k)^T} . g^{(k)}} \quad . \tag{2.4.51}$$

For a wide class of functions, the conjugate gradient method with exact line search and exact arithmetic is n-step superlinearly convergent, i.e. :

$$\lim_{j \to \infty} \frac{\| x^{(nj+n)} - x^* \|}{\| x^{(nj)} - x^* \|} = 0 \quad . \tag{2.4.52}$$

However, for nonquadratic functions, the rate of convergence of the algorithm is usually only linear, unless the iterative procedure is occasionally "restarted" with a steepest descent direction. Fletcher and Reeves (1964) restart every (n+1) iterations, where n is the number of variables. In any case, the term "n-step superlinearly convergent" has little meaning when n is very large because of the large number of iterations required for the asymptotic convergence theory to hold.

A full description of the method, including historical development and recent advances of the conjugate gradient methods is given in the following section.

5. A STUDY OF CONJUGATE GRADIENT METHODS.

Conjugate gradient methods were originally proposed by Hestenes and Stiefel (1952) for the solution of systems of linear equations. Because the problem of finding the unconstrained minimizer of a quadratic function is equivalent to the problem of solving a system of linear equations, these methods have been used for this purpose also. Their application to nonquadratic functions has first been introduced by Fletcher and Reeves (1964) and is based on the fundamental assumption that in the neighbourhood of the required minimum the function is expected to behave like a quadratic function and therefore, can be approximated by one. Hence we first need to study some conjugate properties of quadratic functions.

5.1 Quadratic Functions and Conjugacy.

5.1.1 Basic Properties of Quadratic Functions.

Let n be a fixed integer and let f be the quadratic function:

$$f(x) = a + b^T x + \frac{1}{2} x^T Cx \ , \qquad (2.5.1)$$

where C is a real symmetric n×n matrix, b is a fixed n-dimensional vector and a is a scalar.

The gradient of f at x is the vector:

$$g(x) = Cx + b \ . \qquad (2.5.2)$$

A stationary point of f is a point x such that g(x) = 0. Thus x is a stationary point of the quadratic function f if and only if x is a solution of the linear system of equations:

$$Cx = - b \ . \qquad (2.5.3)$$

The system (2.5.3) may or may not, have a solution. However, if C is non-singular there is exactly one solution, namely:

$$x^* = - C^{-1} b \ , \qquad (2.5.4)$$

and x^* is the unique stationary point of f. If C is singular and x^* is a solution of (2.5.3), then every solution x of (2.5.3) is expressible in the form $x = x^* + z$, where z is a null vector of C, that is, a vector z such that Cz = 0. In other words, if x^* is a stationary point of f, then every stationary point of f differs from x^* by a null vector z of C.

Because f is quadratic we have the identity:

$$f(x+s) = f(x) + s^T(Cx+b) + \frac{1}{2} s^T Cs \ . \qquad (2.5.5)$$

This follows from the fact that the quadratic approximation to f based on the Taylor series expansion (2.4.32) is exact for a quadratic function. Rewritten (2.5.5) states that:

$$f(x+s) = f(x) + s^T g(x) + \frac{1}{2} s^T H(x)\, s. \qquad (2.5.6)$$

The matrix $H(x) = C$ is the Hessian matrix of f. If x^* is a stationary point, then, by replacing x by x^* and s by $x - x^*$ in (2.5.6) we obtain the formula:

$$f(x) = f(x^*) + \frac{1}{2} (x-x^*)^T C(x-x^*) \qquad (2.5.7)$$

for f relative to a stationary point of f. Geometrically this function states that, when C is nonsingular, the stationary point $x^* = - C^{-1}b$ of f is the centre of the quadratic surface

$$f(x) = \gamma \quad , \qquad (2.5.8)$$

where γ is a constant.

A minimum point x^* of f is a stationary point of f and by (2.5.7) a stationary point x^* of f is a minimum point of f if and only if C is nonnegative, i.e. if and only if the inequality $s^T C\, s \geq 0$ holds for every vector s. If $s^T C\, s > 0$ whenever $s \neq 0$, i.e. if C is positive definite, then $x^* = - C^{-1}b$ is the unique minimum point of f. f is said to be a positive definite quadratic function if C is positive definite. The level surfaces (2.5.8) for a positive definite quadratic function f are ellipsoids having $x^* = - C^{-1}b$ as their common centre. The problem of minimizing f is therefore equivalent to the geometrical problem of finding the centre of an ellipsoid. This fact leads us to a geometrical

description of the methods of conjugate directions and conjugate gradients, which is the main topic in this section.

It is important to note that the level surfaces

$$f(x) = \gamma \qquad \text{for} \quad \gamma > f(x^*)$$

are similar ellipsoids. That is to say that if x_α and x_β are the points in which a ray emanating from the common centre x^* cuts the level surfaces $f(x) = \alpha$ and $f(x) = \beta$, respectively, then the ratio

$$\frac{\|x_\beta - x^*\|}{\|x_\alpha - x^*\|} = r$$

of the distances of x_α and x_β from x^* is independent of the choice of this ray. This situation is illustrated in Figure 2.5.1 below:

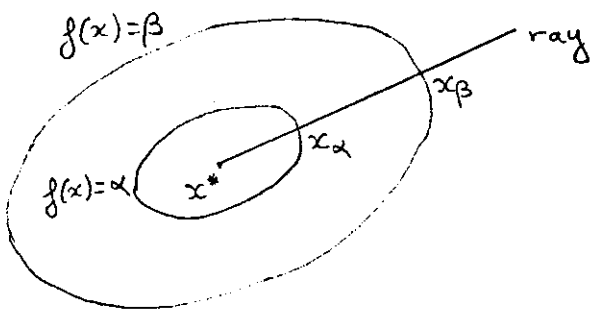

Figure 2.5.1

To ensure the existence of the minimum point $x^* = -C^{-1}b$, we shall assume that the matrix C is positive definite.

5.1.2 Conjugacy.

Let n be a fixed integer and let C be an n×n positive definite matrix.

Two vectors x and y from $\mathbb{R}^n$ are said to be conjugate with respect to the matrix C, if $x^T Cy = 0$. Note that if C is taken to be the identity matrix I, the definition of conjugacy becomes that of orthogonality. Conjugacy is therefore a generalization of the concept of orthogonality by letting the scalar product be

$$[x,y] = x^T Cy \quad . \tag{2.5.9}$$

Note also that if x and y are conjugate with respect to the matrix C then the vectors x and z = Cy are orthogonal.

A system of vectors $\{z^{(1)},z^{(2)},\ldots,z^{(m)}\}$ from $\mathbb{R}^n$ is said to be a system of conjugate vectors with respect to the matrix C, if

$$z^{(i)^T} C\, z^{(j)} = 0 \qquad \text{for all } i \neq j \quad . \tag{2.5.10}$$

Because C is a positive definite matrix, the system $\{r^{(1)},r^{(2)},\ldots,r^{(n)}\}$ of its eigenvectors is a system of conjugate vectors. In this case we have:

$$r^{(i)^T} C\, r^{(j)} = r^{(i)^T} (\lambda_j r^{(j)}) = \lambda_j r^{(i)^T} r^{(j)} = 0 \cdot$$

$$\text{for } i \neq j \tag{2.5.11}$$

where λ_j is the j^{th} eigenvalue of C.

A system of conjugate vectors with respect to the matrix C can be constructed from any free system of vectors $x^{(1)},\ldots,x^{(n)}$ by using the orthogonalization method of Gram-Schmidt using the scalar product (2.5.9). (See Dahlquist and Björk (1974)). This is done as follows:

Put $z^{(1)} = x^{(1)}$. Then for i = 2,3,...n successively put

$$z^{(i)} = x^{(i)} + \sum_{j=1}^{i-1} \beta_{ij}\, z^{(j)} \quad , \tag{2.5.12}$$

where the coefficients β_{ij} are chosen to make $z^{(i)^T} C\, z^{(k)} = 0$ for all $k = 1,2,\ldots,i-1$. This means that β_{ij} must satisfy the equations:

$$x^{(i)^T} C\, z^{(k)} + \sum_{j=1}^{i-1} \beta_{ij}\, z^{(j)^T} C\, z^{(k)} = 0 \quad . \tag{2.5.13}$$

In principle (2.5.13) defines i-1 simultaneous equations for the i-1 unknown β_{ij}. But if we use the fact that $z^{(i)},\ldots,z^{(i-1)}$ are mutually conjugate with respect to the matrix C, we find that they reduce to the i-1 simple equations:

$$\beta_{ij} = - x^{(i)^T} C\, z^{(j)} / z^{(j)^T} C\, z^{(j)}, \quad j = 1,\ldots,i-1 \quad . \tag{2.5.14}$$

The denominator in (2.5.14) cannot vanish if C is positive definite, since the $x^{(j)}$ are assumed to be linearly independent (free system of vectors). Hence $z^{(j)}$ cannot vanish.

5.1.3 Conjugate Properties of Quadratic Functions.

We come now to a description of the properties of the quadratic function (2.5.1), upon which our algorithms of conjugate gradients are based.

Theorem 2.5.1:

The minimum points of the quadratic function of (2.5.1), on parallel lines, lie on an (n-1)-plane π_{n-1} through the minimum point x^* of f. The (n-1)-plane π_{n-1} is defined by the equation

$$s^T(Cx+b) = 0 \quad , \tag{2.5.15}$$

where s is a direction vector for these parallel lines. The vector Cs is normal to π_{n-1} .

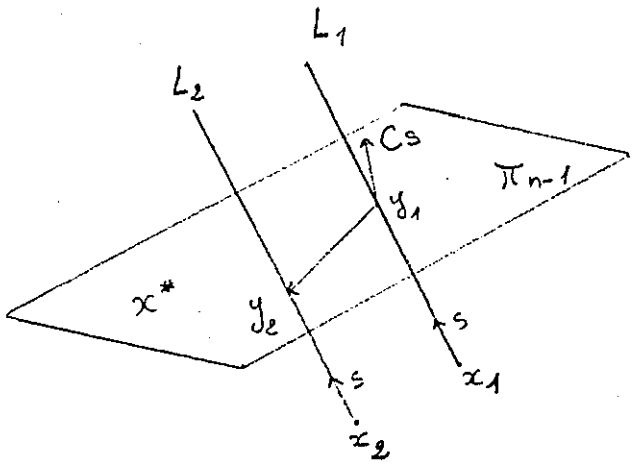

Figure 2.5.2.

Proof:

This result is shown schematically in Figure 2.5.2 above. In this figure, the points y_1 and y_2 are, respectively, the minimum points of f on the two parallel lines L_1 and L_2. The direction of these lines can be represented by a nonnull vector s.

At the minimum point y_1 of f on L_1 the gradient $g(y_1) = Cy_1 + b$ of f is orthogonal to L_1 and hence also to s. The point y_1 therefore satisfies equation (2.5.15). Similarly y_2 satisfies this equation. Equation (2.5.15) represents an (n-1)-plane π_{n-1} whose normal is Cs. Because $Cx^* = - b$, the minimum point x^* of f lies in π_{n-1}. Since π_{n-1} is uniquely determined by the direction vector s, it follows that the minimum points of f on all lines with direction s must lie in the (n-1)-plane π_{n-1}.

Note that because Cs is orthogonal to π_{n-1}, it is orthogonal to every vector q in π_{n-1}. In other words the vector s is conjugate to every vector q in π_{n-1} with respect to the positive definite matrix C and hence the relation

$$s^T C\, q = 0 \tag{2.5.16}$$

holds for every vector q in π_{n-1}. Referring to Figure 2.5.2, note also that the line L_1 is given parametrically in the form $x = x_1 + \alpha s$, where x_1 is a point on L_1 and α a parameter ranging from $-\infty$ to $+\infty$.

By (2.5.5), along the line $x = x_1 + \alpha s$, we have:

$$f(x_1+\alpha s) = f(x_1) + \alpha s^T(Cx_1+b) + \frac{\alpha^2}{2} s^T C\, s.$$

And because $g(x_1) = Cx_1 + b$, this becomes:

$$f(x_1+\alpha s) = f(x_1) + \alpha s^T g(x_1) + \frac{\alpha^2}{2} s^T C\, s \quad . \tag{2.5.17}$$

This function of α has a minimum value when $\alpha = - s^T g(x_1)/s^T C\, s$.

Theorem 2.5.2:

Let $x^{(1)} \in \mathbb{R}^n$ and let $\{z^{(1)},z^{(2)},\ldots,z^{(n)}\}$ be a system of conjugate vectors with respect to the positive definite matrix C of the quadratic function (2.5.1). Let Z be the affine set:

$$Z = \left\{x = x^{(1)} + z \mid z \in [z^{(1)},\ldots,z^{(n)}]\right\} \tag{2.5.18}$$

where $[z^{(1)},\ldots,z^{(n)}]$ is the subspace of the linear combinations of $z^{(1)},z^{(2)},\ldots,z^{(n)}$. Then:

The minimum point x^* of the quadratic function (2.5.1) over

the set Z can be calculated by minimizing f over the points $x^{(i)} \in Z$ such that

$$z^{(i)^T} C(x^{(i)} - x^{(1)}) = 0$$

successively. In other words:

$$f(x^{(i)} + \alpha_i^* z^{(i)}) = \min_{\alpha_i} f(x^{(i)} + \alpha_i z^{(i)}) \Rightarrow f(x^*) = \min_{x \in Z} f(x)$$
$$i = 1,2,\ldots,n$$

where $x^* = x^{(1)} + \alpha_1^* z^{(1)} + \alpha_2^* z^{(2)} + \ldots + \alpha_n^* z^{(n)}$.

The proof of this theorem is given in Appendix 1.

5.2 Conjugate Gradient Methods for Quadratic Functions.

We now proceed to develop specific computational methods for minimizing the positive definite quadratic function f of (2.5.1). These methods consist of minimizing f successively along lines. If these lines are mutually conjugate, the method is called a conjugate direction method for finding the minimum point $x^* = - C^{-1}b$ of f. By virtue of Theorem 2.5.2, a conjugate direction method terminates in $m \leq n$ steps (if no roundoff errors occur). This fact also follows from Theorem 2.5.1, as can be seen from the following geometrical description of a conjugate direction method.

(1) Select a point $x^{(1)}$ and a Line L_1 through $x^{(1)}$ in a direction $s^{(1)}$.

(2) Find the minimum point $x^{(2)}$ of f on L_1.

(3) Construct the (n-1)-plane π_{n-1} through $x^{(2)}$ which is conjugate to $s^{(1)}$.

By Theorem (2.5.1) the minimum x^* of f is in π_{n-1}. Consequently, our next search can be limited to π_{n-1} so that we have reduced the dimensionality of our space of search by one. We now repeat the process, restricting ourselves to the (n-1)-plane π_{n-1}.

(1) Select a line L_2 in π_{n-1} through $x^{(2)}$ in a direction $s^{(2)}$.

(2) Find the minimum point $x^{(3)}$ of f on L_2.

(3) Construct the (n-2)-plane π_{n-2} through $x^{(3)}$ which is conjugate to $s^{(2)}$.

By Theorem (2.5.1) again with π_{n-1} playing the role of the Euclidean space E_n, the minimum point x^* of f is in π_{n-2} so that we can limit our search to π_{n-2}. Again the dimension of our space of search has been reduced by one. Proceeding in this manner we reduce the dimensionality of our space of search by one at each step.

At the n^{th} step our space of search is the line π_1 through x^* so that the minimum point $x^{(n+1)}$ of f on π_1 coincides with the minimum point x^*. Of course, on rare occasions we have $x^{(m+1)} = x^*$ at an m^{th} step ($m < n$), in which case we can terminate in $m < n$ steps.

A conjugate gradient method is a conjugate direction method characterized by the construction in step (3) of each iteration k of the (n-k)-plane π_{n-k}, through the minimum point $x^{(k+1)}$ of f on the line L_k, which is conjugate to the direction $s^{(k)}$ of the line L_k.

5.2.1 Generation of Conjugate Gradients.

Because the function f in (2.5.1) is a positive definite quadratic function, at any point $x^{(k)}$, the gradient:

$$g^{(k)} = g(x^{(k)}) = C\, x^{(k)} + b$$

is available. Initially, at the chosen starting point $x^{(1)}$, only one direction is known, that of the gradient $g^{(1)}$. This is chosen as the first direction of the conjugate set, i.e.

$$s^{(1)} = - g^{(1)} \quad . \tag{2.5.19}$$

The function f is then minimized along the line $L_1 = \{x \mid x = x^{(1)} + \alpha s^{(1)}\}$. The minimum point along L_1 is say $x^{(2)} = x^{(1)} + \alpha_1^* s^{(1)}$, where α_1^* is found by an exact line search. This will imply that

$$s^{(1)^T} g^{(2)} = 0 \quad . \tag{2.5.20}$$

The property (2.5.20) follows from the fact that the line L_1 is tangent to the level surface $f(x) = f(x^{(2)})$ at the point $x^{(2)}$ and therefore $g^{(2)}$ is orthogonal to L_1, hence to $s^{(1)}$.

At $x^{(2)}$, two directions are available : $g^{(2)}$ and $s^{(1)}$, and a direction $s^{(2)}$ is sought in the plane defined by these two vectors, i.e.

$$s^{(2)} = - g^{(2)} + \beta s^{(1)} \tag{2.5.21}$$

for some scalar β, which will be termed the modifying constant or the update. For $s^{(2)}$ to be conjugate to $s^{(1)}$ with respect to the matrix C, we require:

$$s^{(2)^T} C s^{(1)} = 0 \quad . \tag{2.5.22}$$

The requirement (2.5.22) is equivalent to:

$$s^{(2)^T} C(\alpha_1^* s^{(1)}) = 0 \quad ,$$

and because $C(\alpha_1^* s^{(1)}) = g^{(2)} - g^{(1)}$, the conjugacy condition (2.5.22) becomes:

$$(-g^{(2)} + \beta s^{(1)})^T (g^{(2)} - g^{(1)}) = 0$$

and therefore

$$\beta = \frac{g^{(2)^T}(g^{(2)} - g^{(1)})}{s^{(1)^T}(g^{(2)} - g^{(1)})} \quad . \tag{2.5.23}$$

A line search is now made in the direction $s^{(2)}$ so that $g^{(3)^T} s^{(2)} = 0$, and by simple algebra this implies that:

$$g^{(3)^T} s^{(2)} = g^{(3)^T} s^{(1)} \quad \text{and} \quad g^{(3)^T} g^{(2)} = g^{(3)^T} g^{(1)} = 0 \quad .$$

We now continue the process by induction. Assume that at $x^{(k+1)}$, $k+1$ directions are available, namely $g^{(k+1)}$, $s^{(1)},\ldots,s^{(k)}$, that have the following properties:

$$s^{(1)},\ldots,s^{(k)} \quad \text{are conjugate with respect to } C, \tag{2.5.24a}$$

$$g^{(k+1)^T} s^{(i)} = 0 \qquad i = 1,\ldots,k \ , \tag{2.5.24b}$$

$$g^{(k+1)^T} g^{(i)} = 0 \qquad i = 1,\ldots,k \quad . \tag{2.5.24c}$$

We can now use the orthogonalization process of Gram-Schmidt defined by (2.5.12), (2.5.13) and (2.5.14) above, in which the vectors x and z are replaced by $-g$ and s respectively, to obtain:

$$s^{(k+1)} = -g^{(k+1)} + \sum_{j=1}^{k} \beta_j s^{(j)} \quad . \tag{2.5.25}$$

The k conjugacy conditions to be considered here are:

$$s^{(k+1)^T} C s^{(i)} = 0 \qquad i = 1,\ldots,k \ ,$$

but because these conditions are equivalent to:

$$\alpha_i^* \, s^{(k+1)^T} C \, s^{(i)} = 0 \qquad i = 1,\ldots,k \ ,$$

we shall use the latter for reasons that will become clearer as this analysis goes on. Equations (2.5.13) will then become:

$$g^{(k+1)^T} C(\alpha_i^* \, s^{(i)}) = \sum_{j=1}^{k} \beta_{ij} \, s^{(j)^T} C(\alpha_i^* \, s^{(i)}) \ ; \quad i = 1,\ldots,k \ .$$

And when we use the fact that $s^{(1)},\ldots,s^{(k)}$ are conjugate (2.5.24a), we obtain

$$\beta_{ij} = \beta_i = \frac{g^{(k+1)^T} C(\alpha_i^* \, s^{(i)})}{s^{(i)^T} C(\alpha_i^* \, s^{(i)})} \qquad i = 1,\ldots,k \ . \tag{2.5.26}$$

We now observe that:

$$g^{(i+1)} - g^{(i)} = b + C(x^{(i)} + \alpha_i^* \, s^{(i)}) - b - C \, x^{(i)} = \alpha_i^* \, C \, s^{(i)} \ . \tag{2.5.27}$$

Therefore, by substituting (2.5.27) into (2.5.26) we obtain:

$$\beta_i = \frac{g^{(k+1)^T} (g^{(i+1)} - g^{(i)})}{s^{(i)^T} (g^{(i+1)} - g^{(i)})} \qquad i = 1,\ldots,k \ .$$

Finally by using the orthogonality conditions (2.5.24b) and (2.5.24c) above we get:

$$\beta_i = 0 \quad \text{for} \quad i < k$$

and hence (2.5.25) becomes

$$s^{(k+1)} = - g^{(k+1)} + \beta^{(k)} s^{(k)} \tag{2.5.28}$$

where

$$\beta^{(k)} = \frac{g^{(k+1)^T}(g^{(k+1)} - g^{(k)})}{s^{(k)^T}(g^{(k+1)} - g^{(k)})} \tag{2.5.29}$$

which is the standard form of Hestenes and Stiefel (1952) of the conjugate gradient algorithm. Further simplifications of (2.5.29) have been suggested in the literature based on the facts that:

(1) $g^{(k+1)^T} s^{(k)} = 0$ as the line search along $s^{(k)}$ is exact,

(2) $g^{(k)^T} s^{(k-1)} = 0$ as the line search along $s^{(k-1)}$ has been exact, and

(3) $g^{(k+1)^T} g^{(k)} = 0$ as the function is quadratic, the search directions are conjugate and the line searches are exact.

By using (1) and (3) we obtain the formula:

$$\beta^{(k)} = - \frac{g^{(k+1)^T} g^{(k+1)}}{s^{(k)^T} g^{(k)}} \tag{2.5.30}$$

that Dixon (1972) attributes to Myers. If we note that

$$s^{(k)} = - g^{(k)} + \beta^{(k-1)} s^{(k-1)} , \tag{2.5.31}$$

the denominator of (2.5.30) becomes:

$$(- g^{(k)} + \beta^{(k-1)} s^{(k-1)})^T g^{(k)} = - g^{(k)^T} g^{(k)} + \beta^{(k-1)} s^{(k-1)^T} g^{(k)},$$

and by using (2) we obtain the formula:

$$\beta^{(k)} = \frac{g^{(k+1)^T} g^{(k+1)}}{g^{(k)^T} g^{(k)}} \qquad (2.5.32)$$

of Fletcher and Reeves (1964). If we now consider (2.5.29) again and use (1), (2) and (2.5.31) we obtain the formula:

$$\beta^{(k)} = \frac{g^{(k+1)^T} (g^{(k+1)} - g^{(k)})}{g^{(k)^T} g^{(k)}} \qquad (2.5.33)$$

of Polak and Ribiere (1969).

These formulae for the update $\beta^{(k)}$, generate the same set of conjugate directions on quadratic functions. However, when, as is often the case, these formulae are applied on non-quadratic functions, different directions result as * is no longer correct.

One of the major advantages in conjugate gradient methods comes from the observation (2.5.27); because the matrix C of second derivatives of f is constant, one can find out curvature information by observing the way the first derivatives change and therefore avoid the storage of matrices and any computation involving matrices.

5.2.2 Convergence Properties.

We shall now discuss the convergence properties of conjugate gradient methods when the objective function is a positive definite quadratic function.

It is well known that when applied on a positive definite quadratic function, the conjugate gradient method using any of the formulae above to compute the update $\beta^{(k)}$ will terminate in

* (2.5.3)

at most n steps from any starting point $x^{(1)}$, provided the starting direction is the steepest descent direction $s^{(1)} = - g^{(1)}$. This result follows from Theorem 2.5.2 above. It can also be shown that the number of steps required for termination to occur, is equal to the number of distinct eigenvalues of the matrix C. For a proof of this result see, for example, Hestenes (1980).

Crowder and Wolfe (1971) have shown that if the wrong starting direction is used, then the rate of convergence is at best linear. This result has been extended by Powell (1976), to show that if the objective function is a positive definite quadratic function and if the initial search direction is an arbitrary descent direction, then if termination does not occur within n+1 steps, the rate of convergence is only linear. Thus superlinear convergence never occurs even when f is quadratic. Furthermore, Powell (1976) also found that it can happen that linear convergence is obtained and every sequence of $(\ell+1)$ consecutive search directions is mutually conjugate, where ℓ is a positive integer less than (n-1). However, he also found conditions to improve on $x^{(1)}$ and $s^{(1)}$ that are necessary and sufficient for termination to occur in a finite number of steps. This result of Powell's also shows that if $s^{(1)}$ is fixed and if the components of $x^{(1)}$ are chosen at random, for instance from the uniform distribution in [-1, 1], then the probability of obtaining termination by chance, when $n \geq 3$ and there are no degeneracies, is zero. Thus a linear rate of convergence is usual when the conjugate gradient algorithm is applied on a general convex quadratic function and when both $s^{(1)}$ and $x^{(1)}$ are arbitrary.

5.3 Conjugate Gradient Methods for General Functions.

The conjugate gradient algorithm for minimizing a quadratic function can be combined with Newton's method for minimizing a non-quadratic function f to obtain an effective method for finding the minimum x^* of f. We assume here of course that the Hessian matrix of f is positive definite. The only significant modification to the method is that the step lengths $\alpha^{(k)}$ are no longer easily provided by an exact line search but they are usually computed by an inexact line search procedure. The finite termination property of the method on quadratic functions suggests that the definition (2.5.28) should be abandoned after a cycle of linear searches, and that $s^{(k)}$ should then be set to the steepest descent direction $-g^{(k)}$. This strategy is known as restarting; it is also known as "resetting" or "reinitialization". The combined method proceeds as follows:

(1) Select a starting point $x^{(1)}$.

(2) Construct the Newton approximation:

$$F(s) = f(x^{(1)}+s) = f(x^{(1)}) + g^{(1)^T}s + \frac{1}{2}s^T H(x^{(1)})s \quad .$$

(3) With $s^{(1)} = 0$ as the initial point of F, use a conjugate gradient routine to obtain the minimum point $s^{(n+1)}$ of F.

(4) Repeat computations (2) and (3) with $x^{(1)}$ replaced by $x^{(1)} + s^{(n+1)}$. Terminate if $|g^{(k)}|$ is so small that $x^{(k)} = x^{(1)} + s^{(k(n+1))}$ is an acceptable estimate of the minimum point x^* of f.

The approximation however, is not done explicitly, and since it

is hardly possible to obtain an exact step length without major complications, inexact line search procedures are used instead.

5.3.1 Conjugate Gradients Without Exact Line Searches.

When the criterion (2.4.20) of exact line search is relaxed then the equation following (2.5.27) does not give:

$$\beta_i = 0, \qquad \text{for} \quad i < k ,$$

and the correct calculation of β_i would involve the storage of all $g^{(i)}$. Dixon (1975) gives a modification of the strategy for generating conjugate gradients, that makes the accurate line search (2.4.20) unnecessary. His modification is based on estimating the differences

$$z^{(k)} = x^{(k)} - x^{(k)^*} \quad \text{and} \quad w^{(k)} = g^{(k)} - g^{(k)^*}$$

using the properties of quadratic functions, where $x^{(k)}$ and $g^{(k)}$ denote the variables and gradients calculated with an inexact line search, and $x^{(k)^*}$ and $g^{(k)^*}$ denote the variables and gradients that would have been calculated if the line searches were exact. In particular the vector $w^{(k)}$ is used to ensure that the direction $(x^{(k+1)} - x^{(k)})$ is parallel to $(x^{(k+1)^*} - x^{(k)^*})$. Further details of the method can be found in Dixon (1975).

5.3.2 Restarting Procedures.

It is usually suggested that, when using a conjugate gradient method to minimize non-quadratic functions, the procedure should be restarted at regular intervals (say after n steps for an

n-dimensional problem) with a steepest descent step. An apparent advantage of this strategy is that if the iterates progress from a non-quadratic region into a neighbourhood of the solution in which f(x) is closely approximated by a quadratic, accumulated errors from previous iterations will not interfere with the expected good behaviour of the algorithm.

Fletcher (1980) reckons that for some large problems with certain types of symmetry it might be appropriate to restart more frequently than on every n iterations. His analysis suggests that on large problems for which the conjugate gradient methods are suitable, restarting with the steepest descent direction every n iterations is an irrelevant consideration. What might be preferable is to restart every m << n iterations where m is somehow determined by the symmetry. Unfortunately it is not obvious how to choose an m in the algorithm.

Restarting with the steepest descent direction is also based on the questionable assumption that the reduction in f(x) along the restart direction will be greater than that obtained if the usual formula were used. However, Powell (1977) shows by example that this is not always true, and in fact the reduction at the restart iteration is often poor compared with the reduction that would have occurred without restarting. Although it would seem useful if a cycle of n iterations could commence with the last direction of the previous cycle, this idea must be applied with some care.

If, when applied to a quadratic function a conjugate gradient algorithm takes an arbitrary initial search direction $s^{(1)}$, the

required conjugacy relations may not hold because $g^{(1)}$ is no longer a linear combination of the search directions. However, one can use the strategy of Beale (1972) to derive conjugate directions without having to start with the steepest descent direction. Therefore to ensure that the successive directions are conjugate, the following recurrence relation must be used:

$$s^{(k)} = - g^{(k)} + \beta^{(k-1)} s^{(k-1)} + \gamma^{(k)} s^{(1)} \qquad (2.5.34)$$

where $\beta^{(k-1)}$ is given by (2.5.29) and where $\gamma^{(k)}$ is:

$$\gamma^{(k)} = \frac{g^{(k)^T}(g^{(2)} - g^{(1)})}{s^{(1)^T}(g^{(2)} - g^{(1)})} \quad .$$

This strategy of Beale (1972) has been tried by McGuire and Wolfe (1973) who found their numerical results rather disappointing. Powell (1977) however, has turned their difficulties to an advantage because a restart is needed when these difficulties occur, and obtains an efficient restarting procedure that takes account of the objective function automatically and that does not abandon the second derivative information that is found by the previous searches. Powell's restart is based on the fact that (2.5.34) can be extended to non-quadratic problems by computing a cycle of n directions: for k = 1,2,...,n.

$$s^{(k)} = - g^{(k)} + \beta^{(k-1)} s^{(k-1)} + \gamma^{(k)} s^{(i)} \quad , \qquad (2.5.35)$$

where

$$\gamma^{(k)} = \frac{g^{(k)}(g^{i+1)} - g^{(i)})}{s^{(i)^T}(g^{(i+1)} - g^{(i)})} \quad . \qquad (2.5.36)$$

The direction $s^{(i)}$ is known as the restart direction and is the last direction of the previous cycle along which a linear search was made.

When minimizing a non quadratic function, $s^{(k)}$ as defined in (2.5.35) may not be a descent direction, even if exact line searches are made; this implies that the method may generate a poor direction of descent. Steps must therefore be taken to replace this direction, if necessary, by the usual conjugate gradient direction. A typical requirement for a search direction to be suitable is that $s^{(k)}$ should be "sufficiently downhill"; for example, we may impose a condition such as:

$$- g^{(k)^T} s^{(k)} \geq \rho \| s^{(k)} \| . \| g^{(k)} \| \quad , \tag{2.5.37}$$

for some positive value ρ and where $\| \cdot \|$ denotes the Euclidean norm. If this requirement is not satisfied, a new cycle commences with $s^{(k-1)}$ as the restart direction and with $s^{(k)}$ computed from (2.4.35).

If an inexact line search is used, condition (2.5.37) should be checked at trial step lengths with the restart direction. This will ensure that, if the direction computed from the three term formula (2.5.35) is not to be used, the restart direction will be a descent direction.

Chapter 3

Efficiency and Convergence Properties of Conjugate Gradient Methods

1. INTRODUCTION AND PRELIMINARIES.

Conjugate gradient methods form a class of methods that generate directions of search without storing any matrix. They aim to solve the unconstrained minimization problem:

$$\text{Minimize } f(x), \quad x \in \mathbb{R}^n \tag{3.1.1}$$

by a sequence of line searches:

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} s^{(k)} , \tag{3.1.2}$$

starting from a user supplied estimate $x^{(1)}$ of the minimizer x^*. If the line search is exact, the stepsize $\alpha^{(k)}$ taken by the algorithm in the direction $s^{(k)}$, is defined by:

$$\alpha^{(k)} = \text{Arg} \min_{\alpha} f(x^{(k)} + \alpha s^{(k)}) . \tag{3.1.3}$$

In practice however, an exact line search is not usually possible and one accepts any value of $\alpha^{(k)}$ that satisfies certain standard conditions. Fletcher (1980) suggests that $\alpha^{(k)}$ is such that $x^{(k+1)}$ satisfies the condition:

$$|g^{(k+1)^T} s^{(k)}| \leq -\sigma g^{(k)^T} s^{(k)} , \tag{3.1.4}$$

together with the Goldstein (1965) requirement:

$$f(x^{k+1)}) \leq f(x^{(k)}) + \rho\alpha^{(k)} g^{(k)^T} s^{(k)} , \tag{3.1.5}$$

where $g^{(k)} = \nabla f(x^{(k)})$ is the gradient vector at $x^{(k)}$, and where $\rho \in (0, \frac{1}{2})$, $\sigma \in (0, 1)$ and $\rho < \sigma$.

The search direction $s^{(k)}$ is usually defined so that:

$$g^{(k)^T} s^{(k)} < 0 \qquad (3.1.6)$$

holds for all k such that $g^{(k)} \neq 0$, which ensures that f(x) can be decreased in the line search. Hence condition (3.1.4) ensures that the modulus of the slope is reduced by a factor σ or less in the line search (see Fletcher (1980)). Specifically, the search direction $s^{(k)}$, for conjugate gradient methods is defined as follows:

$$\left.\begin{aligned} s^{(1)} &= - g^{(1)} \\ s^{(k+1)} &= - g^{(k+1)} + \beta^{(k)} s^{(k)}, \quad k > 1 \end{aligned}\right\} \qquad (3.1.7)$$

Various formulae for $\beta^{(k)}$ have been suggested in the literature, but for the purpose of this thesis, attention will be focussed on the Fletcher-Reeves and the Polak-Ribière versions of conjugate gradient algorithms, for which $\beta^{(k)}$ is defined respectively as follows:

- The Fletcher-Reeves Update (1964):

$$\beta^{(k)} = \| g^{(k+1)} \|^2 / \| g^{(k)} \|^2 \quad . \qquad (3.1.8)$$

- The Polak-Ribière Update (1969):

$$\beta^{(k)} = g^{(k+1)^T} (g^{(k+1)} - g^{(k)}) / \| g^{(k)} \|^2 \qquad (3.1.9)$$

where $\| \cdot \|$ denotes the Euclidean norm.

In what follows we shall assume that the level set:

$$\{x : f(x) \leq f(x^{(1)})\} \qquad (3.1.10)$$

is bounded. This assumption will ensure that $\alpha^{(k)}$ is well defined for all k.

2. DESCENT PROPERTY AND GLOBAL CONVERGENCE OF THE FLETCHER-REEVES METHOD.

2.1 The Case of Exact Line Search.

2.1.1 Descent Property.

It is clear that $s^{(1)^T}.g^{(1)} = -g^{(1)^T}.g^{(1)} < 0$, so the descent property (3.1.6) holds on the first iteration for any conjugate gradient algorithm. Moreover, the line search being exact we have:

$$g^{(k+1)^T} s^{(k)} = 0 \quad \text{for} \quad k \geq 1 \tag{3.2.1}$$

and therefore from (3.1.7) and (3.2.1) it follows that:

$$g^{(k+1)^T} s^{(k+1)} = g^{(k+1)^T}(-g^{(k+1)} + \beta^{(k)} s^{(k)})$$

$$= -\| g^{(k+1)} \|^2 < 0 \quad . \tag{3.2.2}$$

This shows that a descent property holds on all iterations for any conjugate gradient formula and in particular for that of Fletcher-Reeves.

2.1.2 Global Convergence.

Powell (1983) shows that if the level set (3.1.10) is bounded, if $\alpha^{(k)}$ is defined so that (3.2.1) holds for all k, and if f(x) is twice continuously differentiable, then the Fletcher-Reeves method achieves the limit:

$$\lim_{k\to\infty} \inf \| g^{(k)} \| = 0 \quad . \tag{3.2.3}$$

His method of proof is of interest to what will follow in the next two chapters and is therefore given below.

From the definition of $s^{(k)}$ and exact line search, we deduce

the equation:

$$\|s^{(k)}\|^2 = \|g^{(k)}\|^2 + \beta^{(k-1)^2}\|s^{(k-1)}\|^2$$

$$= \|g^{(k)}\|^2 + \beta^{(k-1)^2}\left[\|g^{(k-1)}\|^2 + \beta^{(k-2)^2}\|s^{(k-2)}\|^2\right]$$

$$= \text{-------}$$

$$= \sum_{\ell=1}^{k}\left\{\prod_{j=\ell}^{k-1}\beta^{(j)^2}\right\}\|g^{(\ell)}\|^2 = \sum_{\ell=1}^{k}\frac{\|g^{(k)}\|^4}{\|g^{(\ell)}\|^2}, \qquad (3.2.4)$$

where the product is defined to be one if $\ell = k$, and where the last line depends on the value (3.1.8). Further, we recall that $\|g^{(k)}\|$ is bounded above because $x^{(k)}$ is in the level set (3.1.10). Therefore if (3.2.3) is not true, then there exists a constant ε, say, such that

$$\|g^{(k)}\| \geq \varepsilon > 0 \quad \text{for all } k \qquad (3.2.5)$$

and hence (3.2.4) gives:

$$\|s^{(k)}\|^2 \leq ck$$

where c is a positive constant. From the definition of $\alpha^{(k-1)}$ and $s^{(k)}$, it follows that:

$$\cos\theta^{(k)} = -s^{(k)^T}g^{(k)} / (\|s^{(k)}\|.\|g^{(k)}\|) \qquad (3.2.6)$$

can be written

$$\cos\theta^{(k)} = \|g^{(k)}\| / \|s^{(k)}\|,$$

and because $\|g^{(k)}\|$ is also bounded above in the level set (3.1.10), we have:

$$\cos^2\theta^{(k)} \geq c_1/k$$

which clearly implies that the series $\sum_k \cos^2\theta^{(k)}$ is divergent. On the other hand, by letting Ω be an upper bound on the induced

matrix norms $\| \nabla^2 f(x) \|$, for x in the level set (3.1.10), we obtain by a simple Taylor expansion about $x^{(k)}$ that:

$$f(x) \leq f(x^{(k)}) + (x-x^{(k)})^T g^{(k)} + \frac{1}{2}\Omega \| x-x^{(k)} \|^2 \qquad (3.2.7)$$

for any x in the level set (3.1.10). Because $x^{(k+1)}$ is in (3.1.10), relation (3.2.7) holds when x is replaced by $x^{(k+1)}$. Therefore we obtain:

$$f(x^{(k+1)}) \leq f(x^{(k)}) + (x^{(k+1)} - x^{(k)})^T g^{(k)} + \frac{1}{2}\Omega \| x^{(k+1)} - x^{(k)} \|^2 .$$

From (3.1.2) we have $x^{(k+1)} - x^{(k)} = \alpha^{(k)} s^{(k)}$ where $\alpha^{(k)}$ is such that (3.1.3) is satisfied. It then follows that:

$$f(x^{(k+1)}) \leq \min_{\alpha} \left[f(x^{(k)}) + \alpha s^{(k)T} g^{(k)} + \frac{1}{2}\Omega \alpha^2 \| s^{(k)} \|^2 \right] . \qquad (3.2.8)$$

The minimum value of the right hand side of (3.2.8) with respect to α, is obtained by computing the derivative with respect to α of the expression between squared brackets and making it equal to zero, i.e.

$$s^{(k)T} g^{(k)} + \Omega \alpha \| s^{(k)} \|^2 = 0 .$$

It therefore follows that the minimum value is obtained when

$$\alpha^{(k)} = - \frac{s^{(k)T} g^{(k)}}{\Omega \| s^{(k)} \|^2} . \qquad (3.2.9)$$

By substituting (3.2.9) into (3.2.8) and by using (3.2.6) we obtain:

$$f(x^{(k+1)}) \leq f(x^{(k)}) + \left[- \frac{s^{(k)T} g^{(k)}}{\Omega \| s^{(k)} \|^2} s^{(k)T} g^{(k)} + \frac{1}{2} \frac{\Omega}{\Omega^2} \left(\frac{s^{(k)T} g^{(k)}}{\| s^{(k)} \|^2} \right)^2 \| s^{(k)} \|^2 \right]$$

$$= f(x^{(k)}) - \frac{1}{2\Omega}\left[\frac{s^{(k)^T} g^{(k)}}{\|s^{(k)}\| . \|g^{(k)}\|}\right]^2 . \|g^{(k)}\|^2$$

$$= f(x^{(k)}) - \frac{1}{2\Omega} \cos^2 \theta^{(k)} . \|g^{(k)}\|^2 \quad .$$

Thus, because $f(x)$ is bounded below in the level set (3.1.10), the series $\sum_k \cos^2 \theta^{(k)} . \|g^{(k)}\|^2$ is convergent; and because of (3.2.5), this contradicts the fact that $\sum_k \cos^2 \theta^{(k)}$ is divergent established above. It therefore follows that (3.2.5) is false and the limit (3.2.3) must be achieved.

2.2 The Case of Inexact Line Search.

Al-Baali (1985) extends Powell's result to show that even for an inexact line search, the descent property (3.1.6) holds for all k and global convergence is achieved for the Fletcher-Reeves method. Again his methods of proof are of importance to what will follow in the next two chapters and are therefore given below.

2.2.1 Descent Property.

Theorem 3.2.1: (Al-Baali (1985)).

If an $\alpha^{(k)}$ is calculated which satisfies (3.1.4) with $\sigma \in (0, \frac{1}{2}]$ for all k such that $g^{(k)} \neq 0$, then the descent property for the Fletcher-Reeves method holds for all such k.

Proof:

The method of proof is to show by induction that the inequalities:

$$-\sum_{j=0}^{k-1}\sigma^j \leq \frac{g^{(k)^T} s^{(k)}}{\|g^{(k)}\|^2} \leq -2+\sum_{j=0}^{k-1}\sigma^j \qquad (3.2.10)$$

hold for all k such that $g^{(k)} \neq 0$. It will then follow inductively that the descent property holds. For k = 1, (3.2.10) is clearly satisfied. Now assume that it is true for any $k \geq 1$. Since

$$\sum_{j=0}^{k-1}\sigma^j < \sum_{j=0}^{\infty}\sigma^j = \frac{1}{1-\sigma} \quad , \qquad (3.2.11)$$

it is clear that the right hand side of (3.2.10) is negative for any σ in $(0, \frac{1}{2}]$ and hence the descent property (3.1.6) holds on iteration k. From the definition of $s^{(k+1)}$, (3.1.7) and that of $\beta^{(k)}$ (3.1.8) it follows that

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\|g^{(k+1)}\|^2} = -1 + \frac{g^{(k+1)^T} s^{(k)}}{\|g^{(k)}\|^2}$$

and therefore by using (3.1.4) and (3.1.6) we obtain:

$$-1+\sigma\frac{g^{(k)^T} s^{(k)}}{\|g^{(k)}\|^2} \leq \frac{g^{(k+1)^T} s^{(k+1)}}{\|g^{(k+1)}\|^2} \leq -1-\sigma\frac{g^{(k)^T} s^{(k)}}{\|g^{(k)}\|^2} \quad .$$

We now use the induction assumption (3.2.10) to obtain:

$$-1-\sigma\sum_{j=0}^{k-1}\sigma^j \leq \frac{g^{(k+1)^T} s^{(k+1)}}{\|g^{(k+1)}\|^2} \leq -1+\sigma\sum_{j=0}^{k-1}\sigma^j \quad . \qquad (3.2.12)$$

On the other hand we have:

$$\sigma\sum_{j=0}^{k-1}\sigma^j = \frac{\sigma-\sigma^{k+1}}{1-\sigma}$$

$$= \frac{-1+\sigma}{1-\sigma} + \frac{1-\sigma^{k+1}}{1-\sigma}$$

$$= -1+\sum_{j=0}^{k}\sigma^j \quad . \qquad (3.2.13)$$

Finally, by substituting (3.2.13) into (3.2.12) we obtain:

$$-\sum_{j=0}^{k}\sigma^j \leq \frac{g^{(k+1)^T} s^{(k+1)}}{\|g^{(k+1)}\|^2} \leq -2+\sum_{j=0}^{k}\sigma^j$$

which is (3.2.10) with k replaced by k+1. Thus the induction is complete.

2.2.2 Global Convergence.

As a consequence of the descent property that has just been shown above, Al-Baali (1985) also proves a theorem for the global convergence of the Fletcher-Reeves method with an inexact line search satisfying (3.1.4) and (3.1.5).

Theorem 3.2.2: (Al-Baali (1985)).

If the level set (3.1.10) is bounded, if f(x) is twice continuously differentiable, and if an inexact line search satisfying (3.1.4) and (3.1.5) with $\rho < \sigma < \frac{1}{2}$ is used, then the Fletcher-Reeves method achieves the limit (3.2.3).

Proof:

From the definition of $s^{(k)}$ in (3.1.7) we have:

$$\|s^{(k)}\|^2 = \|g^{(k)}\|^2 - 2\beta^{(k-1)}\, g^{(k)^T} s^{(k-1)} + \beta^{(k-1)^2}\|s^{(k-1)}\|^2 . \tag{3.2.14}$$

On the other hand from (3.1.4), (3.2.10) and (3.2.11) it follows that:

$$|g^{(k)^T} s^{(k-1)}| \leq -\sigma g^{(k-1)^T} s^{(k-1)} \leq \frac{\sigma}{1-\sigma}\|g^{(k-1)}\|^2 ,$$

which can be substituted into (3.2.14) to give:

$$\| s^{(k)} \|^2 \leq \| g^{(k)} \|^2 + \frac{2\sigma}{1-\sigma} \beta^{(k-1)} \| g^{(k-1)} \|^2 + \beta^{(k-1)^2} \| s^{(k-1)} \|^2 \quad ,$$

and by using the special form of the Fletcher-Reeves beta in (3.1.8) and a similar induction argument as that of (3.2.4) in Powell's proof, we obtain:

$$\| s^{(k)} \|^2 \leq \left[\frac{1+\sigma}{1-\sigma}\right] \| g^{(k)} \|^4 \sum_{\ell=1}^{k} \| g^{(\ell)} \|^{-2} \quad . \tag{3.2.15}$$

Further we recall again that $\| g^{(k)} \|$ is bounded above in the level set (3.1.10); therefore if (3.2.3) is not true, then there exists a constant ϵ, say, such that (3.2.5) is true. Hence (3.2.15) gives:

$$\| s^{(k)} \| \leq c_1 k \quad , \tag{3.2.16}$$

where c_1 is a positive constant. From (3.2.10) and (3.2.11) it follows that:

$$\cos \theta^{(k)} \geq \left[\frac{1-2\sigma}{1-\sigma}\right] \frac{\| g^{(k)} \|}{\| s^{(k)} \|} \quad , \tag{3.2.17}$$

where $\cos \theta^{(k)}$ is as defined in (3.2.6). Because $\sigma < \frac{1}{2}$ we obtain from (3.2.16) and (3.2.17) that

$$\sum_k \cos^2 \theta^{(k)} \geq \left[\frac{1-2\sigma}{1-\sigma}\right]^2 \sum_k \| g^{(k)} \|^2 / \| s^{(k)} \|^2$$

$$\geq c_2 \sum_k 1/k \tag{3.2.18}$$

where c_2 is a positive constant. Hence the series $\sum_k \cos^2 \theta^{(k)}$ diverges. However, it is possible to contradict this result using the line search conditions. If Ω is an upper bound on $\| \nabla^2 f(x) \|$, where x is any point in the level set (3.1.10), we have

$$g^{(k+1)^T} s^{(k)} \leq g^{(k)^T} s^{(k)} + \Omega \alpha^{(k)} \| s^{(k)} \|^2 \quad .$$

Then by using (3.1.4) we obtain:

$$\alpha^{(k)} \geq -\frac{1-\sigma}{\Omega\|s^{(k)}\|^2}\, g^{(k)^T} s^{(k)} \quad ,$$

which can be substituted into (3.1.5), and by using (3.1.6) and (3.2.6) it follows that:

$$f(x^{k+1}) \leq f(x^{(k)}) - c_3 \|g^{(k)}\|^2 \cos^2 \theta^{(k)},$$

where $c_3 = \rho(1-\sigma)/\Omega > 0$. Thus, since $f(x)$ is bounded, $\sum_k \cos^2 \theta^{(k)} \|g^{(k)}\|^2$ is convergent. Because $\|g^{(k)}\|$ is bounded below this contradicts (3.2.18). Since this contradiction arises from (3.2.5) it follows that (3.2.5) is false and hence (3.2.3) is true.

3. THE POLAK-RIBIÈRE METHOD.

3.1 Comparison with the Fletcher-Reeves Method.

3.1.1 Computational Comparison.

In numerical computations, it is usually found that the method of Polak-Ribière is generally far more successful than that of Fletcher-Reeves.

Powell (1977) mentions a particular minimization problem of 165 variables that gives the energy of a model of an atomic system and the least value gives the structure of the nuclei when a gas condenses. This problem was first solved using the Fletcher-Reeves algorithm which showed some slow progress for several iterations and the required minimum was only obtained by forcing the algorithm to restart more often than every n iterations. Then using the same starting point, the Polak-Ribière method only took twelve iterations

to find the required minimum to six decimals accuracy without any restarts.

In this thesis too, the two methods are tested and compared. We used seven different test problems each of which is considered with 26 different numbers of variables, resulting in a total of 182 cases. These test problems are described in Chapter 7. The results obtained by the two methods on each individual case are given in Appendix 2, Tables (2.1.i) and (2.2.i) i = 1,2,...,7 whereas the overall results with the comparison of the figures for the two methods are given in Appendix 3, Tables (3.1) and (3.2). These results confirm the conjecture that the Polak-Ribière method is numerically far more efficient than that of Fletcher-Reeves. Out of the 182 cases considered, only on nine occasions was the Fletcher-Reeves method slightly better than that of Polak-Ribière; in all the other cases, the Polak-Ribière method proved to be far more efficient. The overall comparison results shown in Appendix 3, Table (3.2), show that the Polak-Ribière algorithm obtains the required minima of all the 182 cases in 47%, 44% and 45% less than the Fletcher-Reeves algorithm, in terms of number of iterations (NI), number of function evaluations (NF) and index of computational labour (NC) respectively, the index of computational labour being defined as NC = NF + n.NG, where n is the number of variables and NG the number of gradient evaluations.

3.1.2 Theoretical Explanations.

Powell (1977) also gives a remark that explains the theoretical reasons for this inefficiency of the Fletcher-Reeves method and how this

inefficiency is automatically avoided when the Polak-Ribière method is used instead. His argument is based on Figure 3.3.1 below:

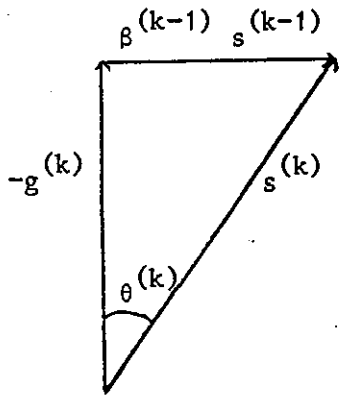

Figure 3.3.1

The definition of $\theta^{(k)}$.

that gives the equation:

$$\| s^{(k)} \| = \sec \theta^{(k)} . \| g^{(k)} \| . \tag{3.3.1}$$

Then, the identity:

$$\beta^{(k)} \| s^{(k)} \| = \tan \theta^{(k+1)} . \| g^{(k+1)} \| \tag{3.3.2}$$

follows if k is replaced by k+1 in the Figure. By eliminating $\| s^{(k)} \|$ from (3.3.1) and (3.3.2) and substituting the formula (3.1.8) for $\beta^{(k)}$ of Fletcher-Reeves we obtain:

$$\begin{aligned} \tan \theta^{(k+1)} &= \sec \theta^{(k)} . \| g^{(k+1)} \| / \| g^{(k)} \| \\ &> \tan \theta^{(k)} \| g^{(k+1)} \| / \| g^{(k)} \| . \end{aligned} \tag{3.3.3}$$

Powell (1977) then notes that if $\theta^{(k)}$ is close to $\pi/2$, the iteration may take a very small step, and $(g^{(k+1)} - g^{(k)})$ is therefore small too. Thus the ratio $\| g^{(k+1)} \| / \| g^{(k)} \|$ is close to 1. It follows from (3.3.3) that $\theta^{(k+1)}$ is also close to $\pi/2$ leading to a slow progress again on the next iteration. Suppose

that the early iterations of the algorithm have made $\theta^{(k)} > 0$, but a region in the space of the variables has been reached where the objective function is the quadratic function:

$$f(x) = x_1^2 + x_2^2 \quad . \tag{3.3.4}$$

In this case the line search along $s^{(k)}$ makes $\| g^{(k+1)} \| \; / \| g^{(k)} \|$ equal to $\sin \theta^{(k)}$. Therefore from the first line of (3.3.3), we obtain $\theta^{(k+1)}$ equals $\theta^{(k)}$. Thus the angle between the search direction and the negative gradient remains constant for all iterations, which is highly inefficient if $\theta^{(k)}$ is close to $\pi/2$. This kind of inefficiency however, is corrected by a steepest descent direction. Alternatively, if the Polak-Ribière formula for $\beta^{(k)}$, (3.1.9) is used instead, Powell uses the equations (3.3.1) and (3.3.2) to show that the behaviour just described does not occur. The definition of $\beta^{(k)}$ (3.1.9) provides the bound:

$$\beta^{(k)} \leq \| g^{(k+1)} \| \; . \; \| g^{(k+1)} - g^{(k)} \| \; / \| g^{(k)} \|^2 \quad . \tag{3.3.5}$$

So, the elimination of $\| s^{(k)} \|$ from (3.3.1) and (3.3.2) gives the inequality

$$\tan \theta^{(k+1)} \leq \sec \theta^{(k)} \; . \; \| g^{(k+1)} - g^{(k)} \| \; / \| g^{(k)} \| \; . \tag{3.3.6}$$

It then follows that if $\theta^{(k)}$ is close to $\pi/2$ and if this causes the step from $x^{(k)}$ to $x^{(k+1)}$ to be so small that the change $(g^{(k+1)} - g^{(k)})$ is much less than $\| g^{(k)} \|$, then $\tan \theta^{(k+1)}$ is much less than $\sec \theta^{(k)}$. Therefore, the search direction $s^{(k+1)}$ is automatically turned towards the steepest descent direction.

In this thesis too, we consider a similar quadratic function

to that of (3.3.4) to investigate the effect of the use of an inexact line search on the loss of conjugacy of the two methods. We consider the quadratic function:

$$f(x) = a\, x_1^2 + x_2^2 \quad , \quad a \geq 1 \quad . \tag{3.3.7}$$

We exclude the cases $a < 0$ and $0 \leq a < 1$ to avoid infinite solutions when $a < 0$ and to be able to make a as large as possible in order to make the objective function as ill-conditioned as possible due to the difference between the magnitudes of the eigenvalues of the Hessian matrix of f.

Starting from an initial estimate $x^{(1)} = (x_1^{(1)}, x_2^{(1)})$, of the minimizer $x^* = (0, 0)^T$ we obtain:

$$f(x^{(1)}) = a\, x_1^{(1)^2} + x_2^{(1)^2}$$

$$g^{(1)} = 2(a\, x_1^{(1)}, x_2^{(1)})^T \quad \text{and} \quad H = 2 \begin{bmatrix} a & 0 \\ 0 & 1 \end{bmatrix}$$

$$s^{(1)} = -\, g^{(1)} = -\, 2(a\, x_1^{(1)}, x_2^{(1)})^T \quad .$$

We shall now consider the cases of exact and inexact line searches separately.

a) Exact Line Search:

Since f is a two-dimensional quadratic function, an exact line search will achieve the required minimum in 2 steps if $a \neq 1$ and in one step if $a = 1$. In any case if we let:

$$K_i = a^i\, x_1^{(1)^2} + x_2^{(1)^2}, \quad \text{for} \quad i = 1,2,\ldots,$$

$$K = K_2^2 / K_3 \quad \text{and}$$

$$A = K_4 K_2 / K_3^2 \quad ,$$

we obtain:

a1. First Iteration:

$$\Phi(\alpha) = f(x^{(1)} + \alpha s^{(1)}) = 4K_3 \alpha^2 - 4K_2\alpha + K_1$$

$$\alpha^* = K_2/2K_3$$

$$x^{(2)} = \frac{(a-1)x_1^{(1)} x_2^{(1)}}{K_3} \begin{bmatrix} -x_2^{(1)} \\ a^2 x_1^{(1)} \end{bmatrix}$$

$$f(x^{(2)}) = \Phi(\alpha^*) = K_1 - K .$$

Note that $K_1 = f(x^{(1)})$ and that if $a = 1$ then $\alpha^* = \frac{1}{2}$, $x^{(2)} = (0.\ 0)^T$, $K_1 = f(x^{(1)})$ and $f(x^{(2)}) = 0$. If $a > 1$ then $0 < \alpha^* < \frac{1}{2}$ and $0 < K < f(x^{(1)})$. K then represents the reduction in the function value. We now suppose that $a > 1$ and proceed with the next iteration.

a2. Second Iteration:

$$g^{(2)} = \frac{2(a-1)a\, x_1^{(1)} x_2^{(1)}}{K_3} \begin{bmatrix} -x_2^{(1)} \\ a\, x_1^{(1)} \end{bmatrix}$$

$$\beta^{(1)} = \beta^{(1)}(F.R) = \beta^{(1)}(P.R) = \frac{a^2 x_1^{(1)2} x_2^{(1)2} (a-1)^2}{K_3^2} .$$

Note that $\lim_{a\to\infty} \beta^{(1)} = \lim_{a\to\infty} \frac{const.}{a^2} = 0$. Therefore when a is very large and the function is ill-conditioned, the conjugate gradient direction is almost the steepest descent direction. $\beta^{(1)}$ however, can also be written as follows:

$$\beta^{(1)} = A - 1 \qquad (3.3.8)$$

and $s^{(2)}$ is the following:

$$s^{(2)} = - g^{(2)} + \beta^{(1)} s^{(1)}$$

$$= - 2a(a-1)\, x_1^{(1)}\, x_2^{(1)}\, \frac{K_2}{K_3^2} \begin{bmatrix} - x_2^{(1)} \\ a^2\, x_1^{(1)} \end{bmatrix}.$$

The iteration then continues to give $x^{(3)} = (0, 0)^T$ which is the required minimum. But because we are interested in the value of $\beta^{(1)}$ and the direction $s^{(2)}$ it is not necessary to continue beyond this point since there will not be another update.

We shall now consider the case of inexact line searches to see which of the betas makes the quantity $s^{(2)}\, Hs^{(1)}$ closer to zero and therefore suffers less the effect of inaccuracy in the value of α^*.

b) Inexact Line Search:

Suppose that the line search performed in the first iteration gives $\alpha^*(r) = r\alpha^*$, $r > 0$. Therefore we obtain:

$$x^{(2)} = x^{(1)} + r\alpha^* s^{(1)} = \frac{(a-1)x_1^{(1)}\, x_2^{(1)}}{K_3} \begin{bmatrix} - x_2^{(1)} \frac{(ra-1)}{(a-1)} \\ a^2 x_1^{(1)} \frac{(a-r)}{(a-1)} \end{bmatrix} + \frac{1-r}{K_3} \begin{bmatrix} a^3\, x_1^{(1)^3} \\ x_2^{(1)^3} \end{bmatrix}$$

$$f(x^{(2)}) = K_1 - (2r-r^2)K = f(x^{(1)}) - (2r-r^2)K \quad .$$

A necessary condition for the descent property to hold is that $f(x^{(2)}) < f(x^{(1)})$, and because $r > 0$ and $K > 0$ we obtain that r must be such that $0 < r < 2$. Further, we can exclude the case

$r > 1$ because in the commonly used backtracking line search algorithms we tend to reduce the step lengths $\alpha^{(k)}$ and therefore r is more likely to be such that $0 < r \lesssim 1$. On the other hand we have:

$$g^{(2)} = \frac{2(a-1)a\, x_1^{(1)} x_2^{(1)}}{K_3} \begin{bmatrix} -x_2^{(1)} \dfrac{(ra-1)}{(a-1)} \\ a\, x_1^{(1)} \dfrac{(a-r)}{(a-1)} \end{bmatrix} + 2\frac{(1-r)}{K_3} \begin{bmatrix} a^4 x_1^{(1)3} \\ x_2^{(1)3} \end{bmatrix} .$$

Note the similarity between the errors that occur in the computation of $x^{(2)}$ and $g^{(2)}$ with inexact line search with respect to the computation of the two vectors with an exact line search. We also have:

$$\| g^{(1)} \|^2 = 4K_2$$

$$\| g^{(2)} \|^2 = 4K_2[Ar^2 - 2r + 1]$$

$$g^{(2)T} g^{(1)} = 4K_2 (1-r) \quad .$$

We therefore obtain:

$$\beta^{(1)} (F.R) = Ar^2 - (2r-1) .$$

and

$$\beta^{(1)} (P.R) = Ar^2 - r$$

where $\beta^{(1)}(F.R)$ and $\beta^{(1)}$ (P.R) denote the formulae (3.1.8) and (3.1.9) respectively. Note that if $r = 1$ then $\beta(F.R) = \beta(P.R) = A-1$, which is the beta obtained with an exact line search. To simplify the analysis without loss of generality we shall assume that the starting point $x^{(1)}$ is such that $x_1^{(1)} = x_2^{(1)} = c$. Thus we obtain:

$$K_i = c^2(a^i+1), \quad K = c^2(a^2+1)^2/(a^3+1), \quad A = \frac{(a^4+1)(a^2+1)}{(a^3+1)^2}$$

$$s^{(1)} = -2c\begin{bmatrix} a \\ 1 \end{bmatrix}, \quad g^{(2)} = \frac{2c^3}{K_3}\left\{a\begin{bmatrix} 1-ra \\ a(a-r) \end{bmatrix} + (1-r)\begin{bmatrix} a^4 \\ 1 \end{bmatrix}\right\}$$

$$s^{(2)} = -g^{(2)} + \beta^{(1)}s^{(1)} = -\frac{2ac^3}{K_3}\begin{bmatrix} 1-ra \\ a^2-ra \end{bmatrix} - \frac{2(1-r)c^3}{K_3}\begin{bmatrix} a^4 \\ 1 \end{bmatrix} - 2c\beta^{(1)}\begin{bmatrix} a \\ 1 \end{bmatrix}.$$

In order to compare the two betas in terms of loss of conjugacy, we shall calculate the quantity $s^{(2)}\,Hs^{(1)}$ for both betas and see which of the two makes this quantity closer to zero, where H is the Hessian matrix of f. After some arithmetical manipulations we obtain the following:

If $\beta^{(1)} = \beta(F.R) = Ar^2 - (2r-1)$ then

$$s^{(2)^T}Hs^{(1)} = 0 \Leftrightarrow f^{(F.R)}(r) = (a^4+1)(a^2+1)r^2 - [3(a^3+1)^2 + a^2(a-1)^2]r + 2(a^3+1)^2 = 0$$

and if $\beta^{(1)} = \beta(P.R) = Ar^2 - r$ then

$$s^{(2)^T}H\,s^{(1)} = 0 \Leftrightarrow f^{(P.R)}(r) = r^2 - r = 0\ .$$

A study of the variations of $f^{(F.R)}$ and $f^{(P.R)}$ show that $f^{(F.R)}(r) = 0$ for $r = 1$ and for $r = 2\,\frac{(a^3+1)^2}{(a^4+1)(a^2+1)}$ and $f^{(P.R)}(r) = 0$ for $r = 0$ and for $r = 1$. This study also allows us to draw the following graphs of both $f^{(F.R)}$ and $f^{(P.R)}$. Note that $f^{(P.R)}(r)$ does not depend on the value of a whereas $f^{(F.R)}$ does. The graph for $f^{(F.R)}$ is drawn for $a = 1$. For $a > 1$ however, $f^{(F.R)}(r) = 0$ for $r = 1$

and for r = t where t > 2. We have

$$f^{(F.R)}_{a=1}(r) = r^2 - 3r + 2 \quad , \qquad t = 2 \quad .$$

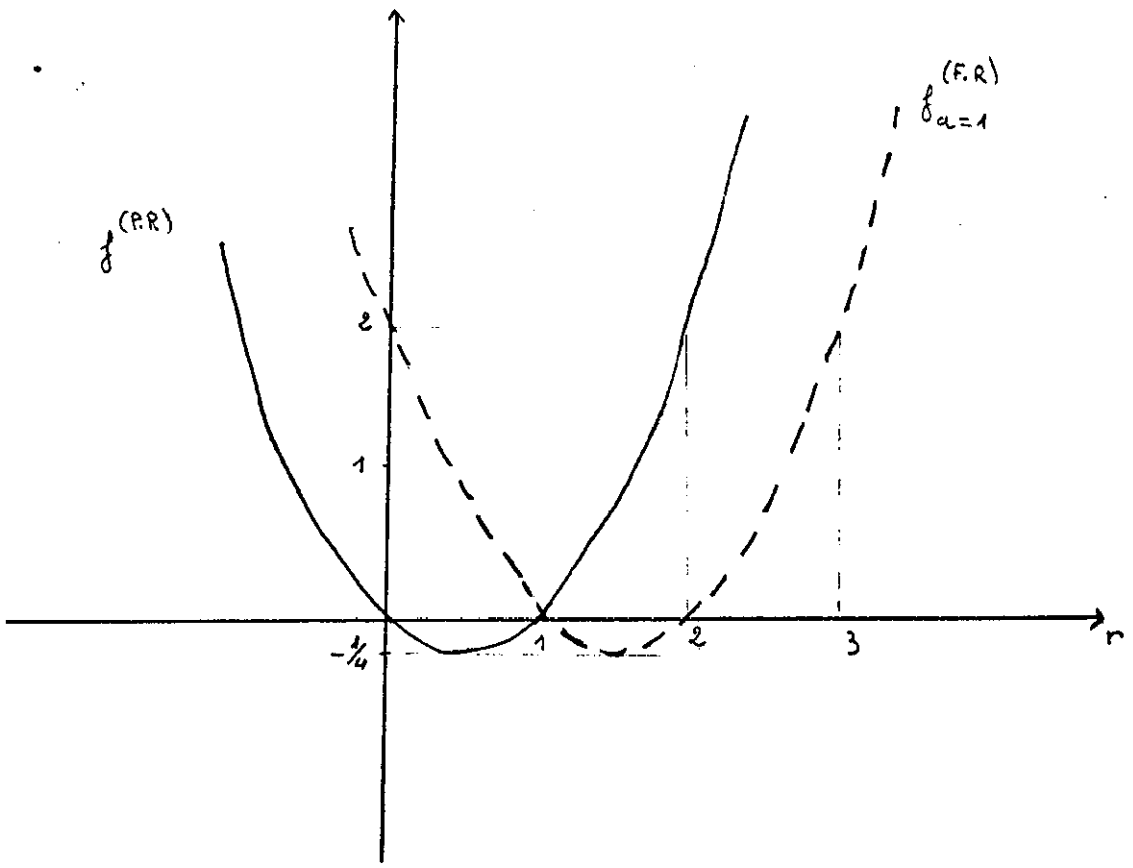

Figure 3.3.2

The graphs representing the variations of $f^{(F.R)}_{a=1}(r)$ and $f^{(P.R)}(r)$.

We can clearly see from the Figure that for r between 0 and 1 which is the interval of interest $f^{(P.R)}(r)$ is closer to zero than $f^{(F.R)}_{a=1}(r)$ is, and therefore the quantity $s^{(2)}_{Hs}{}^{(1)}$ is closer to zero when β(P.R) is used. When a > 1 however, $f^{(F.R)}(r)$ will

approach 0 as a gets larger but we also know that as a gets larger the two betas will tend towards zero and the method will behave as a steepest descent method. We can therefore conclude that for $a \geq 1$ but sufficiently small (not extremely large) the Polak-Ribière method will suffer less the loss of conjugacy due to inexact line searches than the Fletcher-Reeves method would do, and the method of Polak-Ribière is therefore practically preferable.

3.2 GLOBAL CONVERGENCE OF THE POLAK-RIBIÈRE METHOD.

Although in numerical computations the Polak-Ribière method is generally found to be far more successful than that of Fletcher-Reeves (see the discussion in Sub-Section 3.1.1 and the results in Appendix 2) and although this computational out-performance can somehow be explained theoretically as seen in Sub-Section 3.1.2 above, it has not been possible to establish for the Polak-Ribière method, the global convergence results obtained for the Fletcher-Reeves method by Powell (1983) and by Al-Baali (1985), the proofs of which are given in Section 2 of this Chapter.

However, when some additional conditions are imposed such as "f is a convex function" or "the step-lengths $\| x^{(k+1)} - x^{(k)} \|$ tend to zero", the global convergence of the Polak-Ribière method with exact line search may be obtained. Powell (1977) notes that inequality (3.3.6) is sufficiently powerful to prove a global convergence theorem, for the Polak-Ribière method, which, in contrast to a similar theorem given by Polak (1971) does not require f to satisfy any convexity conditions. This theorem of Powell's is proved below.

Theorem 3.3.1: (Powell (1977)).

If the level set (3.1.10) is bounded, if f is continuously differentiable and if the step-lengths $\| x^{(k+1)} - x^{(k)} \|$ tend to zero, then the Polak-Ribière method without restarts achieves the limit (3.2.3).

Proof:

If (3.2.3) is not true there exists a positive constant ε and an integer ℓ such that the bound

$$\| g^{(k)} \| \geq \varepsilon \tag{3.3.9}$$

holds for all $k \geq \ell$. Since $g(x)$ is continuous and since the step-lengths $\| x^{(k+1)} - x^{(k)} \|$ tend to zero, there exists an integer $m \geq \ell$ such that the inequality

$$\| g^{(k+1)} - g^{(k)} \| \geq \frac{1}{2} \varepsilon \quad , \quad k \geq m \tag{3.3.10}$$

is satisfied. Therefore, because the relation

$$\sec \theta^{(k)} \leq 1 + \tan \theta^{(k)} \tag{3.3.11}$$

is true for all $\theta^{(k)}$ in $[0, \pi/2]$, expression (3.3.6) implies the bound

$$\tan \theta^{(k+1)} \leq \frac{1}{2} (1 + \tan \theta^{(k)}), \quad k \geq m , \tag{3.3.12}$$

which is applied recursively to give the inequality

$$\tan \theta^{(k+1)} \leq \frac{1}{2} + \frac{1}{4} + \ldots + \left(\frac{1}{2}\right)^{k+1-m} (1 + \tan \theta^{(m)}). \tag{3.3.13}$$

Therefore the angle between $s^{(k)}$ and the steepest descent direction $- g^{(k)}$ is bounded away from orthogonality. This condition implies that $\| g^{(k)} \|$ tends to zero (Zoutendijk, 1970),

contradicts the hypothesis (3.3.9) and therefore (3.3.9) is false and (3.2.3) must be true.

Before 1983, it was not known whether either of the Fletcher-Reeves or the Polak-Ribière methods provide the limit (3.2.3) for a twice continuously differentiable function with bounded level set from an arbitrary starting point. But, because of condition (3.3.6) and Theorem *, it is straightforward to show for the Polak-Ribière method that if the sequence $\{x^{(k)}, \quad k = 1,2,\ldots\}$ converges to a limit x^* say, then $\nabla f(x^*) = 0$. Therefore, one would imagine that establishing (3.2.3) would be easier for the Polak-Ribière method than for that of Fletcher-Reeves. In 1983, however, Powell came out with the surprising negative result. Not only has he shown by a standard method of proof that the limit (3.2.3) is always achieved by the Fletcher-Reeves method (a result that has further been extended by the Al-Baali (1985) for the case of an inexact line search), but he also found that if the Polak-Ribière method is used, then even with exact arithmetic and an exact line search, there exists a twice continuously differentiable function with bounded level set for which the gradient norms $\{\| g^{(k)} \| , \quad k = 1,2,\ldots.\}$ are bounded away from zero. This counter example of Powell's triggered off the main ideas that lead to the research undertaken in this thesis, and is therefore briefly described below.

It is well known that the limit (3.2.3) is obtained if the directional derivatives:

$$\rho^{(k)} = \frac{s^{(k)^T} g^{(k)}}{\| s^{(k)} \| . \| g^{(k)} \|} < 0, \quad k = 1,2,3,..... \qquad (3.3.14)$$

* 3-3-1

are bounded away from zero. Using the arguments developed earlier for the global convergence of the Fletcher-Reeves method from equation (3.2.6) onwards, we deduce that the limit (3.2.3) fails only if $\sum_k \rho^{(k)}$ is finite (Zoutendijk, 1970). Therefore noting that the definitions of $\alpha^{(k-1)}$ and $s^{(k)}$ imply the value $\rho^{(k)} = -\|g^{(k)}\| / \|s^{(k)}\|$, the gradient norms $\{\|g^{(k)}\|, k = 1,2,\ldots\}$ are bounded away from zero only if the sequence $\{s^{(k)}, k = 1,2,\ldots\}$ is divergent. In this case, due to the existence of an upper bound on $\{\|g^{(k)}\|$ for $x^{(k)}$ in (3.1.10)$\}$, the direction $s^{(k)}$ tends to be parallel to $s^{(k-1)}$. Therefore to seek counter examples to the limit (3.2.3), it is suitable to consider cases where the points $\{x^{(k)}, k = 1,2,\ldots\}$ tend to lie on a straight line, which we take as the first co-ordinate direction in $\mathbb{R}^n$.

For many finite sequences of distinct points $\{x^{(k)}, k = 1,2,\ldots,\ell\}$ one can find gradients $\{g^{(k)}, k = 1,2,\ldots,\ell-1\}$ such that the points can be generated by the conjugate gradient method. Specifically, we let $g^{(1)} = x^{(1)} - x^{(2)}$, and, for $k \geq 2$, because of the definitions of $\alpha^{(k-1)}$ and $s^{(k)}$, $g^{(k)}$ has to be a multiple of the vector in the space spanned by $(x^{(k+1)} - x^{(k)})$ and $(x^{(k)} - x^{(k-1)})$ that is orthogonal to $(x^{(k)} - x^{(k-1)})$. To determine the sign and length of $g^{(k)}$, we note that the value (3.1.9) implies the conjugacy condition

$$(x^{(k+1)} - x^{(k)})^T (g^{(k)} - g^{(k-1)}) = 0, \quad k \geq 2. \tag{3.3.15}$$

Thus, starting with $g^{(1)} = x^{(1)} - x^{(2)}$, the gradients $g^{(k)}$ for $k = 2,3,\ldots$ can usually be found recursively, but the descent conditions $\{g^{(k)T}(x^{(k+1)} - x^{(k)}) < 0, \quad k = 2,3,\ldots,\ell-1\}$ may not hold.

Therefore not all sequences are admissible. Further restrictions on the sequences occur if one lets $\ell \to \infty$, in particular from the aim of keeping the gradient norms bounded away from zero.

To simplify the analysis Powell (1983) imposes the conditions:

$$\left.\begin{aligned} (x^{(k+m)})_1 &= (x^{(k)})_1 \\ \hat{x}^{(k+m)} &= \theta\, x^{(k)} \end{aligned}\right\} \qquad k = 1,2,\ldots \tag{3.3.16}$$

where m is a small positive integer, $(x)_1$ denotes the first component of x, $\hat{x}$ is the vector in $\mathbb{R}^{n-1}$ whose components are the last (n-1) components of x and θ is a constant from [0, 1]. Thus the distance from $x^{(k)}$ to the first co-ordinate direction tends to zero as $k \to \infty$. Having chosen m and n, there are only a finite number of parameters in the sequence $\{x^{(k)},\ k = 1,2,\ldots\}$. One can then express the conditions of consistency with the conjugate gradient method as inequality constraints on the parameters and investigate whether the inequalities have a solution. Powell (1983) then reports some interesting cases for n = 2 and n = 3. For n = 3 he finds that the gradients can stay bounded away from zero if one gives up the second derivative continuity of the objective function. Then, by letting m = 8, he shows that one can preserve the second derivative continuity if one modifies (3.1.3) by allowing $\alpha^{(k)}$ to be any local minimum of $\phi(\alpha) = f(x^{(k)} + \alpha s^{(k)})$ that satisfies $\phi(\alpha^{(k)}) < \phi(0)$. This is an important case because in practice one can usually accept any local minimum that sufficiently reduces the objective function. Finally by letting n = 3 and m = 6 Powell (1983) finds that the gradients $\{g^{(k)},\ k = 1,2,\ldots.\}$ can remain bounded away from zero when all the conditions that have been stated are satisfied.

Powell's two-variable examples are relevant not only to the conjugate gradient methods but also to all variable metric algorithms in Broyden's linear family that make exact line searches (see Fletcher (1980), for example). The reason is that condition (3.3.15) and $s^{(k)^T} g^{(k)} < 0$ define the direction $(x^{(k+1)} - x^{(k)})$ when there are only two variables. Therefore the D.F.P. and B.F.G.S. algorithms may fail to converge, if the condition on the step length $\alpha^{(k)}$ is only that it be a local minimum of the function $\phi(\alpha) \quad \alpha \geq 0$ that satisfies $\phi(\alpha^{(k)}) < \phi(0)$. An important consequence of this remark is that if a proof of convergence of one of these algorithms for general twice continuously differentiable functions could be found, then the proof would depend on line search conditions that are stronger than one usually assumes. Thomson (1977) gives other examples of non convergence for the D.F.P algorithm but he allows the objective function to have first derivative discontinuities.

4. IMPROVEMENTS ON THE CONJUGATE GRADIENT METHOD.

4.1 A Conjugate Direction Method by D. Le (1985).

D. Le (1985) describes a new unconstrained optimization procedure that employs conjugate directions and requiring only three n-dimensional vectors. In common with conjugate gradient methods, this method generates a sequence of search directions $s^{(k)}$ that are linear combinations of $- g^{(k)}$ and $s^{(k-1)}$. This method can be described as follows:

Given a starting point $x^{(1)}$, let $s^{(1)} = - g^{(1)} / \| g^{(1)} \|$,

the vector norms being Euclidean.

For $k \geq 1$, let $x^{(k+1)} = x^{(k)} + \alpha^{(k)} s^{(k)}$

where $\alpha^{(k)}$ is the minimizer of the function $\phi(\alpha) = f(x^{(k)} + \alpha s^{(k)})$.

If $g^{(k+1)} = 0$, stop; otherwise let

$$\omega^{(k+1)} = x^{(k+1)} + \beta^{(k+1)}(-g^{(k+1)}/\|g^{(k+1)}\|) ,$$

$$z^{(k+1)} = \omega^{(k+1)} + \gamma^{(k+1)} s^{(k)} , \text{ and}$$

$$s^{(k+1)} = (z^{(k+1)} - x^{(k+1)})/\|z^{(k+1)} - x^{(k+1)}\| ,$$

where $\gamma^{(k+1)}$ is the minimizer of the function $\phi(\gamma) = f(\omega^{(k+1)} + \gamma s^{(k)})$, and $\beta^{(k+1)}$ is determined by the following procedure:

Let $\lambda \equiv \beta$, $v \equiv x$ and $u \equiv -g/\|g\|$

(1) Let $\lambda_o = 0$, $\lambda_1 = \lambda_o$, $\lambda_2 = \sigma$ and $\xi = \psi(0)$

where $\psi(\lambda) = f(v+\lambda u)$.

(2) Fit a parabola $p(\lambda)$ to $\psi(\lambda)$ using $\psi(0)$, $\psi'(0)$ and $\psi(\lambda_2)$.

(3) Let $\hat{\lambda}$ be the point such that $p(\hat{\lambda}) = p(0) = \psi(0)$ and $\hat{\lambda} \neq 0$. Care should be taken to avoid overflow in solving $p(\lambda) = \psi(0)$ for $\hat{\lambda}$. If two solutions exist that are both non-zero, the larger value is taken for $\hat{\lambda}$.

(4) If $\lambda_1 < \hat{\lambda} \leq 4\sigma$ and $p(\lambda)$ is convex, then accept $\hat{\lambda}$ as the required solution for $\beta^{(k)}$; otherwise let $\lambda_o = \lambda_1$, $\lambda_1 = \lambda_2$, $\sigma = 4\sigma$, $\lambda_2 = \sigma$ and fit a parabola $p(\lambda)$ through $\psi(\lambda_o)$, $\psi(\lambda_1)$, $\psi(\lambda_2)$ and repeat step (3).

The value σ used in this procedure is the initial stepsize in the

line search. D. Le takes $\sigma = 0.05$

Apart from its use in $\omega^{(k+1)}$ above, $\beta^{(k+1)}$ also plays the important role of setting the initial stepsize for subsequent line searches to determine $\gamma^{(k+1)}$, $\alpha^{(k+1)}$ and $\beta^{(k+2)}$. The chosen value of $\beta^{(k+1)}$ is usually larger than that required to minimize $f(x)$ along the steepest descent direction $- g^{(k+1)}$ and the motivation for such a choice is that it could help to speed up convergence by cutting corners over curved ridges. D. Le (1985) has experimentally examined several alternatives on the choice of $\beta^{(k)}$ and has found that both very small and very large values of $\beta^{(k)}$ are undesirable. Small values of $\beta^{(k)}$ tend to produce search directions $s^{(k)}$ close to the steepest descent direction $- g^{(k)}$ which usually results in slower speed of convergence. Moreover, as $\beta^{(k)}$ is used as initial stepsize in subsequent line searches, a small value of $\beta^{(k)}$ means more steps and thus more function evaluations required to bracket the local minimum. Although large values of $\beta^{(k)}$ can sometimes produce spectacular results by side-stepping ridges, it usually requires more refining steps at the final stage of each line search and there is also a high probability of overstepping the local minimum to an unwanted nearby one. D. Le (1985) has also tested this method for computational efficiency and stability on a large set of test functions and compared them with numerical data of other major methods. Although the comparison showed that the method possesses strong superiority over other existing conjugate gradient methods on all problems and can out-perform or is at least as efficient as quasi-Newton methods on many tested problems, our feeling is that because the comparison approach adopted relies

on the literature for numerical results of other methods, it (the approach) presents the major drawback that the results reported for different methods as compiled from different authors' work are not fairly comparable due to such factors as the difference in computers and compiler systems, single or double precision, different line search methods, different stopping criteria, etc.

4.2 An Angle Test as a Restart Criterion for C.G. Algorithms by Shanno (1985).

Shanno (1985) gives an angle test procedure for determining when to restart conjugate gradient methods with a steepest descent direction. The test is based on guaranteeing that the cosine of the angle between the search direction and the negative gradient is within a constant multiple of the cosine of the angle between the Fletcher-Reeves search direction and the negative gradient. This guarantees convergence for the Fletcher-Reeves method is known to converge. This procedure is of importance to what will follow in Chapter 5 and is therefore described below.

Under the assumption that exact line searches are performed, for all conjugate gradient methods we have

$$\cos^2(\theta^{(k)}) = \| g^{(k)} \|^2 / \| s^{(k)} \|^2 \qquad (3.4.1)$$

and

$$\| s^{(k)} \|^2 = \| g^{(k)} \|^2 + \beta^{(k-1)^2} \| s^{(k-1)} \|^2 \quad . \qquad (3.4.2)$$

For $\beta^{(k-1)}$ taken to be the Fletcher-Reeves update, from (3.2.4) we obtain

$$\| s^{(k)} \|^2 = \sum_{\ell=1}^{k} \| g^{(k)} \|^4 / \| g^{(\ell)} \|^2 \qquad (3.4.3)$$

and because $\| g^{(k)} \|$ is bounded above and if $\| g^{(k)} \|$ is also bounded away from zero we obtain:

$$\| s^{(k)} \| \leq ck \tag{3.4.4}$$

and hence from (3.4.1), $\sum_k \cos^2(\theta^{(k)})$ is divergent. (See Powell's proof for the convergence of the Fletcher-Reeves method in Section 2). It is immediately obvious that if $\sum_k \cos^2(\theta^{(k)})$ is divergent then for any $\tau > 0$, $\tau \sum_k \cos^2(\theta^{(k)})$ is divergent. Now, by substituting (3.4.3) into (3.4.1), we obtain:

$$\cos^2(\theta^{(k)}) = \frac{1}{\| g^{(k)} \|^2 \sum_{\ell=1}^{k} \| g^{(\ell)} \|^{-2}} \quad . \tag{3.4.5}$$

Thus for any $\tau > 0$, we define $\gamma^{(k)}$ by

$$\gamma^{(k)^2} = \frac{\tau}{\| g^{(k)} \|^2 \sum_{\ell=1}^{k} \| g^{(\ell)} \|^{-2}} \quad . \tag{3.4.6}$$

Shanno's proposed algorithm is then, to calculate a trial $\hat{s}^{(k)}$ by any desired conjugate gradient formula, then calculate:

$$\cos^2(\hat{\theta}^{(k)}) = (g^{(k)^T} \hat{s}^{(k)})^2 / \| g^{(k)} \|^2 . \| \hat{s}^{(k)} \|^2 \quad . \tag{3.4.7}$$

The final search direction $s^{(k)}$ is chosen by:

$$s^{(k)} = \begin{cases} \hat{s}^{(k)} & \text{if} \quad \cos^2(\hat{\theta}^{(k)}) \geq \gamma^{(k)^2} \\ -g^{(k)} & \text{if} \quad \cos^2(\hat{\theta}^{(k)}) < \gamma^{(k)^2} \end{cases} \quad . \tag{3.4.8}$$

Thus $\gamma^{(k)}$ is used to determine when to fully restart a conjugate gradient search in the direction of the negative gradient.

This algorithm can easily be shown to converge to a stationary point. By (3.4.6), $\tau \leq 1$ implies $\gamma^{(k)^2} \leq 1$. If $s^{(k)} = \hat{s}^{(k)}$, by (3.4.8) $\cos^2(\theta^{(k)}) = \cos^2(\hat{\theta}^{(k)}) \geq \gamma^{(k)^2}$. If $s^{(k)} = -g^{(k)}$, $\cos^2(\theta^{(k)}) = 1$. Hence $\cos^2(\theta^{(k)}) \geq \gamma^{(k)^2}$ for all k and therefore $\sum_k \cos^2(\theta^{(k)})$ diverges by the divergence of $\sum_k \gamma^{(k)^2}$. Thus by Zoutendijk's theorem (1970), the sequence $x^{(k)}$ converges.

An important feature of choosing $s^{(k)}$ by (3.4.8) is that for τ appropriately chosen, once the iterative sequence approaches the minimum sufficiently closely, restarts will not occur. This follows directly from the fact that once the minimum has been approached sufficiently closely, terms of higher order than quadratic have negligible influence on the search direction. As at least two trial points are evaluated using quadratic interpolation along each search direction, all conjugate gradient algorithms reduce to essentially the Fletcher-Reeves algorithm. Thus asymptotically, the rate of convergence is not affected.

The identity (3.4.1) is true for all conjugate gradient methods provided exact line searches are performed. However this is hardly ever done in practice. When $s^{(k)}$ is chosen by (3.1.7) for example, we have:

$$(s^{(k)^T} g^{(k)})^2 = (g^{(k)^T} g^{(k)})^2 - 2\beta^{(k-1)} s^{(k-1)} + \beta^{(k)^2} (g^{(k)^T} s^{(k-1)})^2 . \tag{3.4.9}$$

With inexact line searches, $(g^{(k)^T} s^{(k-1)})$ may be substantially different from zero. Thus setting $\tau = 1$ could appear too restrictive. Shanno used the values $\tau = 0.1$, $\tau = 0.01$ and $\tau = 0$, and from his results noted that τ in the range $0.01 \leq \tau \leq 0.1$ should be adequate in most cases.

Shanno (1985) suggests that because the counter-example of Powell (1983) showing cycling of the Polak-Ribière method casted doubts upon whether a global convergence proof for quasi-Newton methods will ever be found, a test similar to (3.4.6) could be implemented for these methods as well, since in a region surrounding the optimum where f(x) behaves essentially as a quadratic, conjugate gradient and quasi-Newton methods produce the same sequence of points.

4.3 Avoiding Negative Updates of the Polak-Ribière Method.

Powell (1985) gives an alternative explanation of the computational superiority of the Polak-Ribière method over that of Fletcher-Reeves, to that discussed in Section 3.1.2 of this Chapter. He supposes that the k^{th} iteration of a conjugate gradient method has made a change $\| x^{(k+1)} - x^{(k)} \|$ that is much smaller than the step that would have been taken by the steepest descent algorithm and investigates whether the next iteration can be as bad. A relatively small value of $\| x^{(k+1)} - x^{(k)} \|$ occurs only if the angle $\theta^{(k)}$ between $s^{(k)}$ and $- g^{(k)}$ is close to $\pi/2$. Therefore, because the exact line search along $s^{(k-1)}$ and the definition of $s^{(k)}$ imply the value

$$\cos \theta^{(k)} = \| g^{(k)} \| / \| s^{(k)} \|$$

we must have $\| s^{(k)} \| >> \| g^{(k)} \|$, and we wish to avoid $\| s^{(k+1)} \| >> \| g^{(k+1)} \|$. Now, the relatively small value of $\| x^{(k+1)} - x^{(k)} \|$ gives $\| g^{(k+1)} \| \approx \| g^{(k)} \|$, so the Fletcher-Reeves and the Polak-Ribière formulae yield $\beta^{(k)} \approx 1$ and $|\beta^{(k)}| << 1$ respectively. Further using the line search along $s^{(k)}$ being exact and the definition of $s^{(k+1)}$, we deduce the relation:

$$\| s^{(k+1)} \| = \left\{ \| g^{(k+1)} \|^2 + \beta^{(k)^2} \| s^{(k)} \|^2 \right\}^{\frac{1}{2}} . \qquad (3.4.10)$$

It then follows that the Fletcher-Reeves formula gives $\|s^{(k+1)}\| \approx \|s^{(k)}\|$, but the Polak-Ribière formula gives $\| s^{(k+1)} \| << \|s^{(k)}\|$ as required. The crucial difference between the two formulae for $\beta^{(k)}$ is that, if the term $\beta^{(k)} s^{(k)}$ dominates the term $- g^{(k+1)}$ in the equation defining $s^{(k+1)}$, (3.1.7), then in the Fletcher-Reeves methods, the directions $s^{(k+1)}$ and $s^{(k)}$ are nearly the same, but these directions can be almost opposite in the Polak-Ribière method due to $\beta^{(k)} < 0$. Powell (1985) therefore suggests to avoid the use of negative updates when using the Polak-Ribière algorithm by using the following value instead:

$$\beta^{(k)} = \max \left\{ 0, \; g^{(k+1)^T} (g^{(k+1)} - g^{(k)}) / \| g^{(k)} \|^2 \right\} , \qquad (3.4.11)$$

that he reckons may be more useful than those of Fletcher-Reeves and Polak-Ribière.

In this thesis we have implemented this formula (3.4.11) in a conjugate gradient routine and tested the resulting method on all the test problems discussed herein. The results given in Appendices 2 and 3 show that this method does indeed out-perform both the Fletcher-Reeves and the Polak-Ribière method on a large number of test problems. The overall comparison results shown in Appendix 3, Table (3.3), show that this algorithm obtains the required minima of all the 182 cases considered in 52%, 50% and 52% less than the Fletcher-Reeves algorithm and in 11%, 10% and 12% less than the Polak-Ribière algorithm, in terms of number of iterations (NI), number of function evaluations (NF) and index of computational labour (NC) respectively, the index of computational labour being defined as

NC = NF + n.NG, where n is the number of variables and NG the number of gradient evaluations. This idea being computationally attractive, we thought of trying to use both the Polak-Ribière and the Fletcher-Reeves formulae for $\beta^{(k)}$ on one algorithm in an attempt to try and approach the method to fit the convergence proofs of the Fletcher-Reeves method. As a first attempt, we tried to replace each negative update of the Polak-Ribière algorithm by a Fletcher-Reeves one which is always positive. This resulted in the following hybrid algorithm: ORIG1 :

Use a conjugate gradient algorithm where the update $\beta^{(k)}$ is defined as follows:

$$\beta^{(k)} = \begin{cases} \beta^{(k)}(P.R) & \text{if} \quad \beta^{(k)}(P.R) > 0 \\ \beta^{(k)}(F.R) & \text{if} \quad \beta^{(k)}(P.R) \leq 0 \end{cases} \tag{3.4.12}$$

where $\beta^{(k)}(F.R)$ and $\beta^{(k)}(P.R)$ denote the updates of the Fletcher-Reeves and Polak-Ribière methods respectively. This hybrid algorithm ORIG1 was also tested on all the test problems discussed in this thesis and proved to be computationally preferable to both the Fletcher-Reeves and Polak-Ribière algorithm. The overall comparison results shown in Appendix 3, Table (3.4) show that ORIG1 obtains the required minima in 49%, 47% and 49% less than the Fletcher-Reeves and 4%, 4% and 6% less than the Polak-Ribière algorithm in terms of the NI, NF and NC respectively where NI, NF and NC are as defined above. It was however, slightly less efficient than the algorithm using formula (3.4.11) to compute $\beta^{(k)}$.

As a second attempt, it was found that including the Hestenes and Stiefel update in the hybrid was also useful, although using the Hestenes-Stiefel $\beta^{(k)}$ on its own has failed on many occasions to converge and therefore its results are not included here. The new hybrid algorithm ORIG2 using the three formulae together in an conjugate gradient algorithm computes its update as follows:

1°/ If $\beta^{(k)}(H.S) > 0$, $\beta^{(k)} = \beta^{(k)}(H.S)$, return

otherwise go to step 2°

2°/ If $\beta^{(k)}(P.R) > 0$, $\beta^{(k)} = \beta^{(k)}(P.R)$, return

otherwise go to step 3°

3°/ $\beta^{(k)} = \beta^{(k)}(F.R)$.

Here $\beta^{(k)}(F.R)$ and $\beta^{(k)}(P.R)$ are as defined above and $\beta^{(k)}(H.S)$ denotes the update $\beta^{(k)}$ of Hestenes and Stiefel. This hybrid algorithm ORIG2 was tested on all the test problems discussed in this thesis and proved to be computationally preferable to all the Fletcher-Reeves, Polak-Ribière, the algorithm using Powell's suggestion of using formula (3.4.11) and to ORIG1. The overall comparison results given in Appendix 3, Table (3.5) show the following improvement percentages over the Fletcher-Reeves and the Polak-Ribière respectively: 52%, 50%, 52% and 10%, 10%, 12% in turns of NI, NF and NC respectively.

These encouraging computational results lead to thought on how to combine the attractive computational performance of the Polak-Ribière method and the desirable theoretical features of the Fletcher-Reeves in order to come out with a hybrid algorithm

that has the two properties. The investigation of this idea and the resulting hybrid algorithms make the subject of the following two chapters.

Chapter 4

Globally Convergent Hybrid Conjugate Gradient Methods

1. INTRODUCTION.

For general twice continuously differentiable functions with bounded level sets (3.1.10), the Fletcher-Reeves method is shown to achieve the limit (3.2.3), when either exact line searches (Powell (1983)) or inexact line searches satisfying certain standard conditions (Al-Baali (1985)) are used, without any convexity assumptions, (see Chapter 3 - Section 2).

Although in numerical computations, the Polak-Ribière method is generally far more successful than that of Fletcher-Reeves, as can be seen from the results tables reported in Appendices 2 and 3, and as can theoretically be explained as seen in Chapter 3, subsection 3.1.2, it has not been possible to establish these global convergence results for the Polak-Ribière method. Furthermore, Powell (1983) also shows that if the Polak-Ribière is used, then even with exact line searches and exact arithmetic, there exist twice continuously differentiable functions with bounded level sets (3.1.10) for which the gradient norms $\{\| g^{(k)} \|, k = 1,2,...\}$ are bounded away from zero.

This has consequently led to thoughts on how to combine the desirable computational aspects of the Polak-Ribière method and the useful theoretical features of the Fletcher-Reeves method in an attempt to construct some efficient hybrid algorithms that are globally convergent.

The purpose of this chapter is then to show how the proofs of Powell (1983) and the Al-Baali (1985) can be used to guarantee global convergence of conjugate gradient algorithms, that are to be constructed in this thesis, which are hybrids between the Polak-Ribière and the Fletcher-Reeves updates.

2. A NEW HYBRID ALGORITHM: HYBRID 1.

2.1 Construction of the Algorithm.

The idea of avoiding the use of negative updates of the Polak-Ribière method and replacing them with updates of the Fletcher-Reeves, that resulted in the hybrid algorithm ORIG1 discussed in Chapter 3 subsection 4.3, has proved to be computationally efficient but does not ensure the global convergence of the obtained algorithm. However, we find that if the Polak-Ribière $\beta^{(k)}$ are restricted to remain non-negative and at the same time less than or equal to the Fletcher-Reeves $\beta^{(k)}$, then the convergence proofs given by Powell (1983) and by Al-Baali (1985) both apply to the Polak-Ribière method also, but these restrictions are unfortunately not always satisfied.

Remark 1:

Let $\beta^{(k)}(F.R)$ and $\beta^{(k)}(P.R)$ denote the betas satisfying (3.1.8) and (3.1.9) respectively. We then have

$$\beta^{(k)}(P.R) = \beta^{(k)}(F.R) - g^{(k+1)^T} g^{(k)} / \|g^{(k)}\|^2 \quad . \qquad (4.2.1)$$

If $\beta^{(k)}(P.R)$ were always non-negative, we would have

$$\beta^{(k)}(P.R) \geq 0 \Rightarrow \beta^{(k)}(F.R) \geq g^{(k+1)^T} g^{(k)} / \|g^{(k)}\|^2$$

$$\Rightarrow \|g^{(k+1)}\|^2 \geq g^{(k+1)^T} g^{(k)}$$

and if $\beta^{(k)}(P.R)$ were always less than or equal to $\beta^{(k)}(F.R)$ we would have

$$\beta^{(k)}(P.R) \leq \beta^{(k)}(F.R) \Rightarrow - g^{(k+1)^T} g^{(k)} / \|g^{(k)}\|^2 \leq 0$$

$$\Rightarrow g^{(k+1)^T} g^{(k)} \geq 0 \quad . \qquad (4.2.2)$$

Therefore the restrictions that we would like to impose on the Polak-Ribière beta will hold if

$$0 \leq g^{(k+1)^T} g^{(k)} \leq \|g^{(k+1)}\|^2 \quad . \qquad (4.2.3)$$

As a consequence of this remark, we suggest the use of a hybrid conjugate gradient algorithm, Hybrid 1, using formula (3.1.9) whenever condition (4.2.3) is satisfied and formula (3.1.8) otherwise. In other words we define $\beta^{(k)}$ in (3.1.7) as follows:

$$\beta^{(k)} = \left\{ \begin{array}{ll} g^{(k+1)^T}(g^{k+1)} - g^{(k)}) / \|g^{(k)}\|^2 & \text{if (4.2.3) is true} \\ \|g^{(k+1)}\|^2 / \|g^{(k)}\|^2 & \text{otherwise} \end{array} \right\} \qquad (4.2.4)$$

It is also possible to use resetting as in straight-forward Fletcher-Reeves and Polak-Ribière methods, but this does not affect the validity of the proofs that follow.

2.2 Descent Property and Global Convergence of Hybrid 1.

2.2.1 Exact Line Search.

We have already shown (3.2.2) that if the line search is exact then a descent property holds on all iterations for both Fletcher-Reeves and Polak-Ribière methods and therefore it is obvious that it also holds for the proposed hybrid method. Let us now assume that f(x) is twice continuously differentiable, that the level set (3.1.10) is bounded and that an exact line search is performed at each iteration of the algorithm using formula (4.2.4). We ask whether these conditions are sufficient to provide the limit (3.2.3) for the proposed hybrid method.

Theorem 4.2.1:

If f(x) is twice continuously differentiable, if the level set (3.1.10) is bounded and if an exact line search is performed at each iteration, then the limit (3.2.3) is achieved by Hybrid 1.

Proof:

The method of proof is the same as that given by Powell (1983) for the global convergence of the Fletcher-Reeves method.

It is well known that the limit (3.2.3) is obtained if the directional derivatives

$$d^{(k)} = s^{(k)^T} g^{(k)} / (\|s^{(k)}\| \cdot \|g^{(k)}\|) < 0 \qquad (4.2.5)$$

are bounded away from zero. Specifically, by letting all vector norms be Euclidean, and by letting Ω be an upper bound on the

induced matrix norms $\{\|\nabla^2 f(x)\|$, for x in the level set (3.1.10)$\}$, we obtain the relation:

$$f(x) \leq f(x^{(k)}) + (x-x^{(k)})^T g^{(k)} + \frac{1}{2}\Omega\|x-x^{(k)}\|^2 \qquad (4.2.6)$$

for any x in the level set (3.1.10). Because $x^{(k+1)}$ is in the level set (3.1.10) (descent property), relation (4.2.6) is also true when we replace x by $x^{(k+1)}$. Therefore

$$f(x^{(k+1)}) \leq f(x^{(k)}) + (x^{(k+1)} - x^{(k)})^T g^{(k)} + \frac{1}{2}\Omega\|x^{(k+1)} - x^{(k)}\|^2 . \qquad (4.2.7)$$

But $x^{(k+1)} = x^{(k)} + \alpha^{(k)} s^{(k)} \Rightarrow x^{(k+1)} - x^{(k)} = \alpha^{(k)} s^{(k)}$

where $\alpha^{(k)}$ is such that (3.1.3) is satisfied. It follows that:

$$f(x^{(k+1)}) \leq \underset{\alpha}{\text{Min}} \left[f(x^{(k)}) + \alpha s^{(k)^T} g^{(k)} + \frac{1}{2}\Omega\alpha^2\|s^{(k)}\|^2\right] . \qquad (4.2.8)$$

The minimum value of the right hand side of (4.2.8) with respect to α is obtained when

$$\alpha = \alpha^{(k)} = -\frac{s^{(k)^T} g^{(k)}}{\Omega\|s^{(k)}\|^2} . \qquad (4.2.9)$$

By substituting (4.2.9) into (4.2.8) and after some simple arithmetic considerations we get

$$f(x^{(k+1)}) \leq f(x^{(k)}) - d^{(k)^2} \frac{\|g^{(k)}\|^2}{2\Omega} . \qquad (4.2.10)$$

Thus, because f(x) is bounded below $\sum_k d^{(k)^2}\|g^{(k)}\|^2$ is a convergent series. Hence condition (3.2.3) fails only if

$$\sum_k d^{(k)^2} \qquad (4.2.11)$$

is convergent (Zoutendijk (1970)).

On the other hand, from the definition of $s^{(k)}$ and exact line search, we deduce the equation

$$\| s^{(k)} \|^2 = \| g^{(k)} \|^2 + \beta^{(k-1)^2} \| s^{(k-1)} \|^2$$

$$= \sum_{\ell=1}^{k} \left\{ \prod_{j=\ell}^{k-1} \beta^{(j)^2} \right\} \| g^{(\ell)} \|^2 \quad .$$

Since every beta calculated by Hybrid 1 is less than or equal to $\| g^{(k)} \|^2 / \| g^{(k-1)} \|^2$, it follows that:

$$\| s^{(k)} \|^2 \leq \sum_{\ell=1}^{k} \| g^{(k)} \|^4 / \| g^{(\ell)} \|^2 \quad . \tag{4.2.12}$$

Further we recall that $\| g^{(k)} \|$ is bounded above because $x^{(k)}$ is in the level set (3.1.10). Therefore if (3.2.3) is not true then:

$$\| s^{(k)} \|^2 \leq k \, c \tag{4.2.13}$$

where c is a positive constant. From the definition of $\alpha^{(k-1)}$ and $s^{(k)}$ it follows that $d^{(k)} = - \| g^{(k)} \| / \| s^{(k)} \|$ and therefore the sum (4.2.11)

$$\sum_k d^{(k)^2} = \sum_k \| g^{(k)} \|^2 / \| s^{(k)} \|^2 \tag{4.2.14}$$

would be divergent, which would contradict the fact that $\sum_k d^{(k)^2} \| g^{(k)} \|^2$ is convergent established above. Since this contradiction arises from the assumption that (3.2.3) is not true then the limit (3.2.3) must be achieved and hence Hybrid 1 using $\beta^{(k)}$ to satisfy (4.2.4) is globally convergent when exact line searches are performed.

2.2.2 Inexact Line Search.

In what follows, Hybrid 1 is shown to have the descent property and to be globally convergent when an inexact line search is performed. The methods of proof to be followed here are the same as those given by Al-Baali (1985).

Theorem 4.2.2:

If an $\alpha^{(k)}$ is calculated which satisfies (3.1.4) with $\sigma \in (0, \frac{1}{2}]$ for all k such that $g^{(k)} \neq 0$, then the descent property (3.1.6) for Hybrid 1, holds for all such k.

Proof:

The method of proof is to show by induction that the inequalities:

$$-\sum_{j=0}^{k-1} \sigma^j \leq \frac{g^{(k)^T} s^{(k)}}{\| g^{(k)} \|^2} \leq -2 + \sum_{j=0}^{k-1} \sigma^j \qquad (4.2.15)$$

hold for all k such that $g^{(k)} \neq 0$. It will then follow inductively that the descent property (3.1.6) holds for all such k.

For k = 1, (4.2.15) is clearly satisfied. Now assume that (4.2.15) is true for any $k \geq 1$. Since

$$\sum_{j=0}^{k-1} \sigma^j < \sum_{j=0}^{\infty} \sigma^j = \frac{1}{1-\sigma} \quad , \qquad (4.2.16)$$

it is clear that the right hand side of (4.2.15) is negative for any σ in $(0, \frac{1}{2}]$ and hence the descent property (3.1.6) is satisfied on iteration k. It follows from the definition of $s^{(k+1)}$ (3.1.7) that:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} = - 1 + \beta^{(k)} \frac{g^{(k+1)^T} s^{(k)}}{\| g^{(k+1)} \|^2} . \qquad (4.2.17)$$

If $\beta^{(k)}$ takes the value (3.1.8), the proof is given by Al-Baali (1985). We shall then concentrate here on the case where $\beta^{(k)}$ takes the value (3.1.9). In this case (4.2.17) becomes:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} = - 1 + \frac{(\| g^{(k+1)} \|^2 - g^{(k+1)^T} g^{(k)})}{\| g^{(k)} \|^2} \cdot \frac{g^{(k+1)^T} s^{(k)}}{\| g^{(k+1)} \|^2}$$

$$= - 1 + \frac{g^{(k+1)^T} s^{(k)}}{\| g^{(k)} \|^2} \left(1 - \frac{g^{(k+1)^T} g^{(k)}}{\| g^{(k+1)} \|^2} \right) .$$

Since we are using $\beta^{(k)}$ that satisfies (3.1.9), by definition of the algorithm we have $0 \leq g^{(k+1)^T} g^{(k)} \leq \| g^{(k+1)} \|^2$. It then follows that:

$$0 \leq a = 1 - \frac{g^{(k+1)^T} g^{(k)}}{\| g^{(k+1)} \|^2} \leq 1 \quad . \qquad (4.2.18)$$

Therefore:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} = - 1 + a \frac{g^{(k+1)^T} s^{(k)}}{\| g^{(k)} \|^2} \qquad (4.2.19)$$

with $0 \leq a \leq 1$. Using (3.1.4), (3.1.6) and (4.2.19) we get:

$$- 1 + a\sigma \frac{g^{(k)^T} s^{(k)}}{\| g^{(k)} \|^2} \leq \frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \leq - 1 - a\sigma \frac{g^{(k)^T} s^{(k)}}{\| g^{(k)} \|^2} . \qquad (4.2.20)$$

Then using the induction hypothesis (4.2.15), the fact that $0 \leq a \leq 1$ and (4.2.20)

$$-\sum_{j=0}^{k}\sigma^j = -1-\sigma\sum_{j=0}^{k-1}\sigma^j \leq -1-a\sigma\sum_{j=0}^{k-1}\sigma^j \leq \frac{g^{(k+1)^T}s^{(k+1)}}{\|g^{(k+1)}\|^2}$$

$$\leq -1+a\sigma\sum_{j=0}^{k-1}\sigma^j \leq -1+\sigma\sum_{j=0}^{k-1}\sigma^j = -2+\sum_{j=0}^{k}\sigma^j$$

$$\Rightarrow -\sum_{j=0}^{k}\sigma^j \leq \frac{g^{(k+1)^T}s^{(k+1)}}{\|g^{(k+1)}\|^2} \leq -2+\sum_{j=0}^{k}\sigma^j$$

which is (4.2.15) with k replaced by k+1. Thus the induction is complete.

A consequence of this descent property is the following global convergent result.

Theorem 4.2.3:

If the set (3.1.10) is bounded, if f(x) is twice continuously differentiable, and if $\alpha^{(k)}$ is any value satisfying (3.1.4) and (3.1.5) with $\rho < \sigma < \frac{1}{2}$, then equation (3.2.3) holds for Hybrid 1.

Proof:

It is shown in Theorem 4.2.2 that the descent property (3.1.6) holds for $\sigma \in (0, \frac{1}{2}]$, so from (3.1.4), (4.2.15) and (4.2.16), it follows that:

$$|g^{(k)^T}s^{(k-1)}| \leq -\sigma g^{(k-1)^T}s^{(k-1)} \leq \frac{\sigma}{1-\sigma}\|g^{(k-1)}\|^2 \quad . \qquad (4.2.21)$$

Then from the definition of $s^{(k)}$ we have:

$$\|s^{(k)}\|^2 = \|g^{(k)}\|^2 - 2\beta^{(k-1)}\,g^{(k)^T}s^{(k-1)} + \beta^{(k-1)^2}\|s^{(k-1)}\|^2 \quad .$$

Using (4.2.21) we obtain:

$$\| s^{(k)} \|^2 \leq \| g^{(k)} \|^2 + \frac{2\sigma}{1-\sigma} \beta^{(k-1)} \| g^{(k-1)} \|^2 + \beta^{(k-1)^2} \| s^{(k-1)} \|^2 \quad .$$

Since $\beta^{(k-1)} \leq \dfrac{\| g^{(k)} \|^2}{\| g^{(k-1)} \|^2}$ it follows that:

$$\| s^{(k)} \|^2 \leq \| g^{(k)} \|^2 + \frac{2\sigma}{1-\sigma} \| g^{(k)} \|^2 + \beta^{(k-1)^2} \| s^{(k-1)} \|^2$$

$$\leq \left(\frac{1+\sigma}{1-\sigma}\right) \| g^{(k)} \|^2 + \beta^{(k-1)^2} \| s^{(k-1)} \|^2$$

and it follows by induction that

$$\| s^{(k)} \|^2 \leq \left(\frac{1+\sigma}{1-\sigma}\right) \| g^{(k)} \|^4 \sum_{\ell=1}^{k} \| g^{(\ell)} \|^{-2} \quad . \tag{4.2.22}$$

Now, if (3.2.3) is not true, then there exists a constant ε say, such that

$$\| g^{(k)} \| \geq \varepsilon \geq 0 \qquad \text{for all } k \quad . \tag{4.2.23}$$

But $g^{(k)}$ is also bounded above in the level set (3.1.10) and so from (4.2.22) we have:

$$\| s^{(k)} \|^2 < c_1 \, k \tag{4.2.24}$$

where c_1 is a positive constant. From (4.2.15) and (4.2.16), it follows that:

$$\cos(\theta^{(k)}) \geq \left(\frac{1-2\sigma}{1-\sigma}\right) \frac{\| g^{(k)} \|}{\| s^{(k)} \|} \tag{4.2.25}$$

where

$$\cos(\theta^{(k)}) = - g^{(k)^T} s^{(k)} / (\| g^{(k)} \| \cdot \| s^{(k)} \|) \qquad (4.2.26)$$

and $\theta^{(k)}$ is the angle between $s^{(k)}$ and the steepest descent direction $- g^{(k)}$. Because $\sigma < \frac{1}{2}$, from (4.2.24) and (4.2.25) we obtain:

$$\sum_k \cos^2(\theta^{(k)}) \geq \left(\frac{1-2\sigma}{1-\sigma}\right)^2 \sum_k \| g^{(k)} \|^2 / \| s^{(k)} \|^2$$

$$\geq c_2 \sum_k k^{-1} \qquad (4.2.27)$$

where c_2 is a positive constant. Hence the series $\sum_k \cos^2 \theta^{(k)}$ diverges. However it is possible to contradict this result using the line search conditions. If Ω is an upper bound on $\| \nabla^2 f(x) \|$, where x is any point in the level set (3.1.10), then we have:

$$g^{(k+1)^T} s^{(k)} \leq g^{(k)^T} s^{(k)} + \Omega \alpha^{(k)} \| s^{(k)} \|^2 .$$

Thus by using (3.1.4) we obtain:

$$\alpha^{(k)} \geq - \frac{1-\sigma}{\Omega} \frac{g^{(k)^T} s^{(k)}}{\| s^{(k)} \|^2}$$

which can be substituted into (3.1.5), and by using (3.1.6) and (4.2.26) it follows that:

$$f(x^{(k+1)}) \leq f(x^{(k)}) - c_3 \| g^{(k)} \|^2 \cos^2(\theta^{(k)}) ,$$

where $c_3 = \rho(1-\sigma)/ \Omega > 0$. Thus since f(x) is bounded in the level set (3.1.10), $\sum_k \| g^{(k)} \|^2 \cos^2(\theta^{(k)})$ is convergent. And because

$\| g^{(k)} \|$ is bounded below, this contradicts (4.2.27). Since this contradiction arises from (4.2.23), it follows that (4.2.23) is false and therefore (3.2.3) is true.

2.3 Computational Performance of Hybrid 1.

This globally convergent hybrid algorithm was tested on several test problems, namely the extended Rosenbrock, Wood, Miele & Cantrell, Powell, Dixon, Beale and Engwall's test problems, each of which were considered with 26 different numbers of variables resulting in a total of 182 cases. These test problems are described in the final Chapter entitled "Discussion and Conclusions". The program used for testing the method is the N.A.G. routine EØ4DBF, in which the computation of the update $\beta^{(k)}$ was modified to compute $\beta^{(k)}$ as defined in (4.2.4). The results obtained by Hybrid 1 on each individual case are given in Appendix 2, Tables (2.6.i) for i = 1,2,...,7 whereas the overall results with the comparison of the figures obtained by Hybrid 1 against those obtained by both the Fletcher-Reeves and Polak-Ribière methods are given in Appendix 3, Table (3.6). The individual results reported in Appendix 2 show that in many cases Hybrid 1 was a significant improvement on both straight-forward Fletcher-Reeves and Polak-Ribière algorithms, for instance on the 40, 180, 200, 480, 500 variable Wood test problems, the 4, 100, 280 variable Miele & Cantrell test functions, the 4, 20, 260, 340, 420, 480 variable Powell test functions and 40, 80, 120, 140, 200, 320, 440, 480 and 500 variable Dixon test functions. There have also been cases however, where both the Fletcher-Reeves and the Polak-

Ribière algorithms performed much better than Hybrid 1. For instance on the 60, 160, 260, 360, 460 variable Rosenbrock test functions and the 180, 220, 240, 260, 280, 300, 320 variable Beale test functions. As we can see in Table (2.6.1) of Appendix 2, this inefficiency occurred especially when Hybrid 1 was tested on the extended Rosenbrock and Beale test problems.

In the following remark, we attempt to explain some of the reasons for this inefficiency and we also explain how the features that cause it, are likely to occur when Hybrid 1 is tested on problems such as the two-variable Rosenbrock test function.

Remark 2:

Suppose that at the k^{th} iteration we face the situation sketched in Figure 4.2.1 below. Explicitly, we have $\beta^{(k-1)}(F.R)$ and $\beta^{(k-1)}(P.R)$ both positive and $\beta^{(k-1)}(P.R)$ is greater than $\beta^{(k-1)}(F.R)$. In this case (4.2.3) is obviously not satisfied and therefore by definition of $\beta^{(k-1)}$ in (4.2.4), Hybrid 1 will choose to use $\beta^{(k-1)}(F.R)$ instead of $\beta^{(k-1)}(P.R)$. However, as can be seen in Figure 4.2.1, searching along $s^{(k)}(P.R)$ obtained by using $\beta^{(k-1)}(P.R)$ would lead to a better approximation $x^{(k+1)}$ than the approximation expected to be obtained by Hybrid 1 which searches along $s^{(k)}(F.R)$ obtained by using $\beta^{(k-1)}(F.R)$. Therefore, Hybrid 1 could prove to be quite inefficient if this situation occurs frequently.

Figure 4.2.2 shows the contours of the two-variable Rosenbrock test function. We can see that when the solution path is descending the valley away from the origin (0, 0) towards the optimal solution

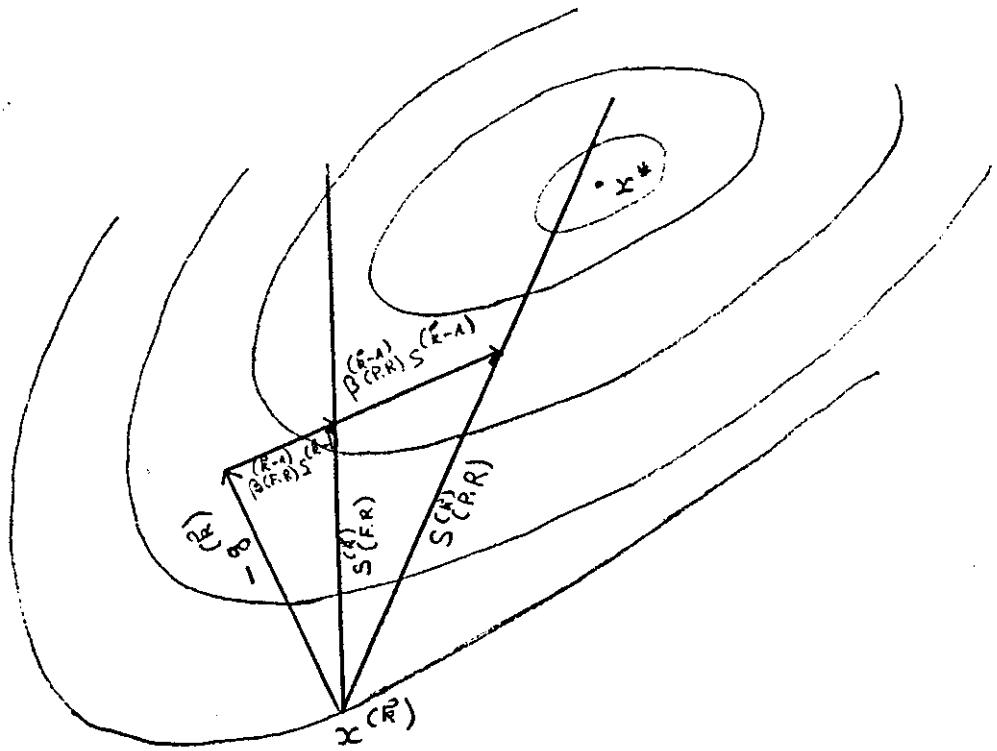

Figure 4.2.1 : A situation where the Polak-Ribière direction of search leads to a better approximation to the minimum than the Fletcher-Reeves direction of search. Because $\beta^{(k-1)}(F.R)$ is less than $\beta^{(k-1)}(P.R)$, by definition (4.2.4) Hybrid 1 chooses to use the Fletcher-Reeves direction. This shows a possible inefficiency of Hybrid 1 and indicates how to improve it, as will be seen in Section 3 below.

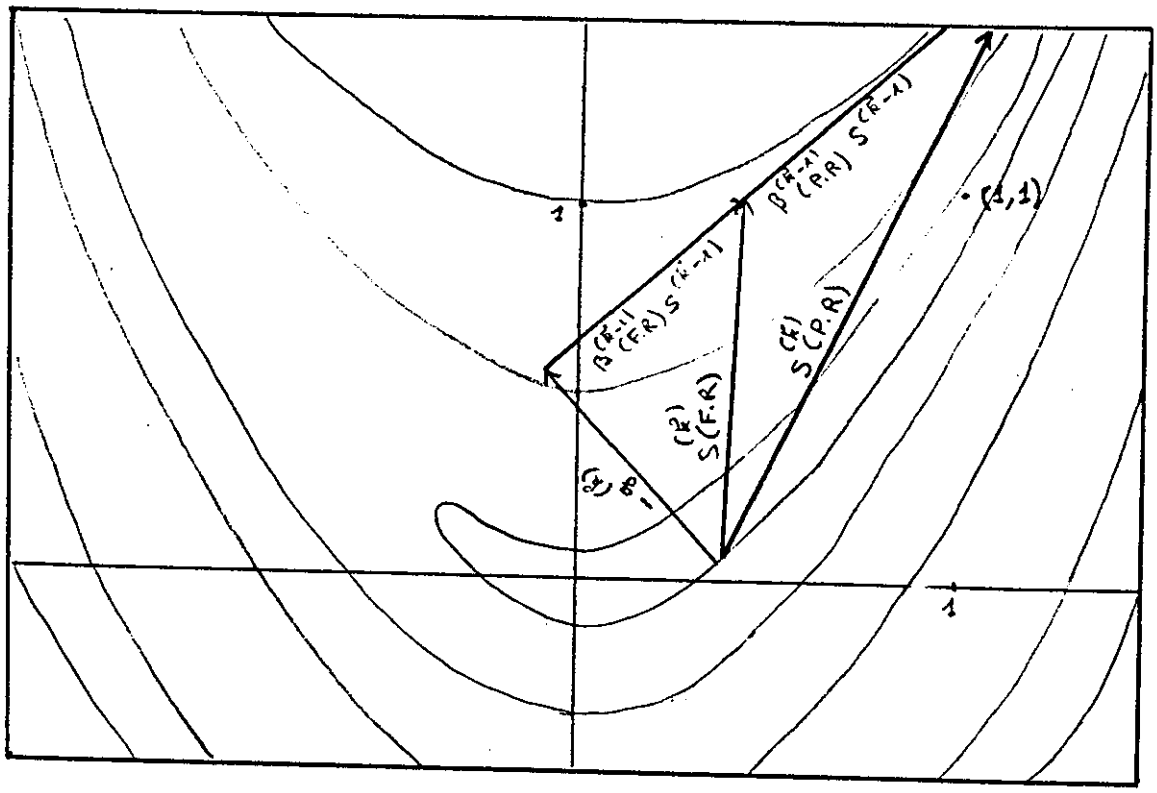

Figure 4.2.2 : Contours of the 2-variable Rosenbrock's test function. The situation shown in Figure 4.2.1 is very likely to occur as shown here.

(1, 1), the situation shown in Figure 4.2.1 is most likely to occur on every iteration thereafter whenever $\beta(P.R) > \beta(F.R)$. This is perhaps the reason for which Hybrid 1 performed quite poorly on the extended Rosenbrock's test problems.

3. AN IMPROVED HYBRID ALGORITHM: HYBRID 2.

To improve the situation described above, one could think of allowing certain values of $\beta(P.R)$, that do not satisfy (4.2.3), to be used as long as the angle between the negative gradient $-g^{(k)}$ and the current search direction $s^{(k)}$ is not too close to $\frac{\pi}{2}$.

Shanno (1985) gives an angle test to determine when conjugate gradient algorithms should be restarted with a steepest descent direction. His procedure is of interest to what will follow and is therefore briefly described below.

His argument is based on the fact that if a series $\sum_k t(k)$ is divergent then for any $\tau > 0$, the series $\tau \sum_k t(k)$ is also divergent. Using the fact that the Fletcher-Reeves algorithm is globally convergent when exact line searches are performed we have:

$$\sum_k \cos^2(\theta^{(k)}) = \sum_k \frac{1}{\|g^{(k)}\|^2 \sum_{\ell=1}^{k} \|g^{(\ell)}\|^{-2}} \qquad (4.3.1)$$

is a divergent series. It is immediately obvious that if $\sum_k \cos^2(\theta^{(k)})$ is divergent, then for any $\tau > 0$, $\tau \sum_k \cos^2(\theta^{(k)})$ is divergent. Thus for any $\tau > 0$, we define $\gamma^{(k)}$ by:

$$\gamma^{(k)^2} = \frac{\tau}{\|g^{(k)}\|^2 \sum_{\ell=1}^{k} \|g^{(\ell)}\|^{-2}} \qquad (4.3.2)$$

Shanno's proposed algorithm is then to calculate a trial direction $\hat{s}^{(k)}$ by any desired conjugate gradient formula, then calculate

$$\cos^2(\hat{\theta}^{(k)}) = \left(g^{(k)^T}\ \hat{s}^{(k)}\right)^2 / \left(\|g^{(k)}\|^2 . \|\hat{s}^{(k)}\|^2\right) \quad . \qquad (4.3.3)$$

The final search vector $s^{(k)}$ is chosen by:

$$s^{(k)} = \begin{cases} \hat{s}^{(k)} & \text{if} \quad \cos^2(\hat{\theta}^{(k)}) \geq \gamma^{(k)^2} \\ -g^{(k)} & \text{if} \quad \cos^2(\hat{\theta}^{(k)}) < \gamma^{(k)^2} \end{cases} \quad . \qquad (4.3.4)$$

This test ensures that the angle between $-g^{(k)}$ and $s^{(k)}$ is small enough to be bounded away from $\frac{\pi}{2}$.

A possible hybrid algorithm would be to use this test for every $\beta(P.R)$ that is greater than $\beta(F.R)$. $\beta(P.R)$ is then used if $\cos^2(\theta^{(k)}) \geq \gamma^{(k)^2}$ and $\beta(F.R)$ is used instead otherwise. In other words one could use the following algorithm to compute $\beta^{(k-1)}$ in (3.1.7) at each iteration k.

<u>Step 1:</u> if $\beta^{(k-1)}(P.R) < 0$ then $\beta^{(k-1)} = \beta^{(k-1)}(F.R)$, return. Otherwise go to step 2.

<u>Step 2:</u> if (4.2.3) holds then $\beta^{(k-1)} = \beta^{(k-1)}(P.R)$, return. Otherwise go to step 3.

<u>Step 3:</u> if $\cos^2(\theta^{(k)}(P.R)) \geq \gamma^{(k)^2}$ then $\beta^{(k-1)} = \beta^{(k-1)}(P.R)$, return. Otherwise $\beta^{(k-1)} = \beta^{(k-1)}(F.R)$, return.

(4.3.5)

This hybrid algorithm (Angle Test Hybrid) was tested on the same test functions used to test Hybrid 1 and showed a considerable improvement on Fletcher-Reeves, Polak-Ribière and Hybrid 1 algorithms. Three different values of τ were tried, namely $\tau = 0.1$, $\tau = 0.01$ and $\tau = 0.001$ and the second value provided better results especially for high dimensional cases. The great improvement of the Angle Test Hybrid over Hybrid 1 on the Rosenbrock test problem confirms the fact that the situation described in Remark 2 did cause some inefficiency of Hybrid 1 and that this test does indeed help to avoid it. The individual results for this hybrid algorithm are given in Tables (2.i.j) for i = 8,9 and 10 and j = 1,2,3,...,7 Appendix 2, whereas the overall results together with the comparison of the figures obtained by the Angle Test Hybrids for different values of τ against those obtained by the straight-forward Fletcher-Reeves and Polak-Ribière methods are given in Appendix 3, Tables (3.8), (3.9) and (3.10).

When exact line searches are performed at each iteration, the hybrid algorithm using (4.3.5) to compute the update β, is shown in subsection 3.2.1 below to have a descent property and to be globally convergent. However, when inexact line searches are used throughout the algorithm, the angle test used in Step 3 of (4.3.5) does not seem to ensure a descent property to hold with which it would be possible to prove global convergence. As it will become clear in the proof of the descent property in subsection 3.2.2 to follow, we find that by imposing the additional condition in Step 3 of (4.3.5) that:

$$\beta^{(k-1)}(P.R) < \frac{1}{2\sigma} \frac{\| g^{(k)} \|^2}{\| g^{(k-1)} \|^2} \qquad (4.3.6)$$

we ensure a descent property to hold and hence the limit (3.2.3) is achieved, where σ is the factor in the line search. Condition (4.3.6) is not needed when exact line searches are used for the simple reason that it is always satisfied. (When exact line searches are used then $\sigma = 0$ and (4.3.6) becomes $\beta^{(k-1)}(P.R) < +\infty$ which is true since $g^{(k)}$ is bounded above in the level set (3.1.10)).

3.1 Definition of the New Hybrid Algorithm: Hybrid 2.

We suggest the use of a new Hybrid Conjugate Gradient Method (Hybrid 2) where the update $\beta^{(k-1)}$ at each iteration k is computed by the following algorithm:

Step 1: if $\beta^{(k-1)}(P.R) < 0$ then $\beta^{(k-1)} = \beta^{(k-1)}(F.R)$, return.
If not go to step 2.

Step 2: if (4.2.3) is true then $\beta^{(k-1)} = \beta^{(k-1)}(P.R)$, return.
If not go to step 3.

Step 3: if $\cos^2(\theta^{(k)}(P.R)) \geq \gamma^{(k)^2}$ and $\beta^{(k-1)}(P.R) < \frac{1}{2\sigma} \frac{\| g^{(k)} \|^2}{\| g^{(k-1)} \|^2}$
then $\beta^{(k-1)} = \beta^{(k-1)}(P.R)$, return.
Otherwise $\beta^{(k-1)} = \beta^{(k-1)}(F.R)$, return. (4.3.7)

This conjugate gradient method Hybrid 2, using algorithm (4.3.7) to compute $\beta^{(k-1)}$ in (3.1.7) produces a sequence of updates $\{\beta^{(\ell)}, \ell = 1,2,\ldots\}$ whose elements $\beta^{(\ell)}$ fall into two categories:

Category 1:

$$0 \leq \beta^{(\ell)} \leq \beta^{(\ell)}(F.R) = \frac{\|g^{(\ell+1)}\|^2}{\|g^{(\ell)}\|^2} \quad , \tag{4.3.8}$$

Category 2:

$$\beta^{(\ell)} > \beta^{(\ell)}(F.R) = \frac{\|g^{(\ell+1)}\|^2}{\|g^{(\ell)}\|^2} \quad , \tag{4.3.9}$$

with

$$\cos^2 \theta^{(\ell+1)} \geq \gamma^{(\ell+1)^2} \tag{4.3.10}$$

and (4.3.6) holds when inexact line searches are used.

This hybrid algorithm Hybrid 2 needs the storage of an extra vector compared with straight-forward Fletcher-Reeves or Polak-Ribière algorithms. This is to store the trial direction of search vector to compute $\cos^2(\theta^{(k)})$ in Step 3 of (4.3.7). As for computer time used for the extra operations we only need to compute $\cos^2(\theta^{(k)})$ and $\gamma^{(k)^2}$ at iterations where a beta from category 2 needs to be tested for use.

Conditions (4.3.6) and (4.3.10) seem intuitively to say the same thing and should therefore be equivalent but it has not been possible to prove this. Satisfying condition (4.3.10) is saying that the angle $\theta^{(k)}$ between the negative gradient $-g^{(k)}$ and the current search direction $s^{(k)}$ is bounded away from orthogonality, i.e.,

$$\cos^2(\theta^{(k)}) \geq \gamma^{(k)^2} \Rightarrow \theta^{(k)} \leq r^{(k)}$$

where $r^{(k)}$ is some angle such that $r^{(k)} < \frac{\pi}{2}$. Then:

$$\theta^{(k)} \leq r^{(k)} \Rightarrow \beta^{(k-1)}\, s^{(k-1)} \leq B(k)$$

$$\Rightarrow \beta^{(k-1)} \leq C(k)$$

where B(k) and C(k) are quantities depending on $g^{(k)}$ and hence on k. It would be useful to prove the equivalence of these two conditions for some suitable values of σ and τ or at least to prove that (4.3.6) implies (4.3.10) because then one can use only condition (4.3.6) with no extra storage or extra computation. An algorithm using (4.3.7) with only condition (4.3.6) in Step 3 was used on the same test functions. The results for this hybrid algorithm (Beta Test Hybrid) are discussed in the final Chapter "Discussion and Conclusions" and are given in Appendices 2 and 3, Tables (2.11.i) i = 1,2,3...,7 and Table(3.11).

3.2 Descent Property and Global Convergence of Hybrid 2.

3.2.1 Exact Line Searches.

(a) Descent property:

This follows from the fact that a descent property holds on all iterations for both Fletcher-Reeves and Polak-Ribière methods when the line search is exact, which has already been shown (3.2.2).

(b) Global convergence:

Since exact line searches are performed condition (4.3.6) is always satisfied and need not be used. Therefore algorithm (4.3.5) to compute the update β at each iteration is used instead of algorithm (4.3.7).

Referring to Theorem 4.2.1, where it was shown that Hybrid 1 was globally convergent when exact line searches are performed, all we need, to show that Hybrid 2 is globally convergent when exact line searches are performed, is to establish a similar inequality to (4.2.12). The following theorem gives this inequality.

Theorem 4.3.1:

If an exact line search is performed at each iteration and if $\beta^{(k)}$ is computed from algorithm (4.3.5) at each iteration and if τ is such that $0 < \tau \leq 1$, then the inequality:

$$\| s^{(k)} \|^2 \leq \frac{1}{\tau} \sum_{\ell=1}^{k} \| g^{(k)} \|^4 / \| g^{(\ell)} \|^2 \qquad (4.3.11)$$

holds for all k.

Proof:

We prove this theorem by induction.

For k = 1 we have:

$$\| s^{(1)} \|^2 = \| g^{(1)} \|^2 \leq \frac{1}{\tau} \| g^{(1)} \|^2 = \frac{1}{\tau} \sum_{\ell=1}^{1} \| g^{(1)} \|^4 / \| g^{(\ell)} \|^2 \quad .$$

Now suppose that at iteration (k-1) we have:

$$\| s^{(k-1)} \|^2 \leq \frac{1}{\tau} \sum_{\ell=1}^{k} \| g^{(k-1)} \|^4 / \| g^{(\ell)} \|^2 \quad .$$

At iteration k, Hybrid 2 could either use a beta from category 1 or a beta from category 2.

If $\beta^{(k-1)}$ is calculated from category 1, then

$$\beta^{(k-1)} \leq \frac{\|g^{(k)}\|^2}{\|g^{(k-1)}\|^2} .$$

Therefore:

$$\|s^{(k)}\|^2 = \|g^{(k)}\|^2 + \beta^{(k-1)^2} \|s^{(k-1)}\|^2$$

$$\leq \|g^{(k)}\|^2 + \frac{\|g^{(k)}\|^4}{\|g^{(k-1)}\|^4} \left[\frac{1}{\tau} \|g^{(k-1)}\|^4 \sum_{\ell=1}^{k-1} \|g^{(\ell)}\|^{-2}\right]$$

Because $0 < \tau \leq 1$ we then have:

$$\|s^{(k)}\|^2 \leq \frac{\|g^{(k)}\|^2}{\tau} + \frac{1}{\tau} \|g^{(k)}\|^4 \sum_{\ell=1}^{k-1} \|g^{(\ell)}\|^{-2}$$

$$= \frac{1}{\tau} \|g^{(k)}\|^4 \sum_{\ell=1}^{k} \|g^{(\ell)}\|^{-2} .$$

If $\beta^{(k-1)}$ is calculated from category 2, then by definition of Hybrid 2 we have:

$$\cos^2(\theta^{(k)}) = \frac{\|g^{(k)}\|^2}{\|s^{(k)}\|^2} \geq \frac{\tau}{\|g^{(k)}\|^2 \sum_{\ell=1}^{k} \|g^{(\ell)}\|^{-2}}$$

$$\Rightarrow \|s^{(k)}\|^2 \leq \frac{1}{\tau} \|g^{(k)}\|^4 \sum_{\ell=1}^{k} \|g^{(\ell)}\|^{-2} .$$

Therefore no matter what beta is used at any stage of the algorithm Hybrid 2, inequality (4.3.11) holds for every k. The global convergence proof follows in the same manner as that following inequality (4.2.11).

3.2.2 Inexact Line Searches.

(a) Descent property:

When inexact line searches are performed with Hybrid 2, algorithm (4.3.7) is used to compute beta. So, every time a beta is chosen from category 2, we have:

$$\beta^{(k-1)} = \frac{\|g^{(k)}\|^2 - g^{(k)^T} g^{(k-1)}}{\|g^{(k-1)}\|^2} < \frac{1}{2\sigma} \frac{\|g^{(k)}\|^2}{\|g^{(k-1)}\|^2}$$

$$\Rightarrow - \frac{g^{(k)^T} g^{(k-1)}}{\|g^{(k)}\|^2} < \frac{1 - 2\sigma}{2\sigma} \qquad (4.3.12)$$

and we also have:

$$\beta^{(k-1)} > \frac{\|g^{(k)}\|^2}{\|g^{(k-1)}\|^2} \Rightarrow - \frac{g^{(k)^T} g^{(k-1)}}{\|g^{(k)}\|^2} > 0 \quad .$$

Therefore we have the relation:

$$0 < - \frac{g^{(k)^T} g^{(k-1)}}{\|g^{(k)}\|^2} < \frac{1 - 2\sigma}{2\sigma} \quad . \qquad (4.3.13)$$

Theorem 4.3.2:

The descent property (3.1.6) holds at every iteration of Hybrid 2 when an inexact line search satisfying (3.1.4) with $\sigma < \frac{1}{2}$ is performed.

Proof:

The method of proof is to show by induction that the inequalities:

$$-1-2\sigma-c(k) \leq \frac{g^{(k)^T} s^{(k)}}{\|g^{(k)}\|^2} \leq -1+2\sigma+c(k) \qquad (4.3.14)$$

hold for all k such that $g^{(k)} \neq 0$, where

$$0 \leq c(k) < 1-2\sigma \quad .$$

It will then follow inductively that the descent property (3.1.6) holds.

For k = 1 we have

$$-1-2\sigma-c(k) \leq -1 \leq -1+2\sigma+c(k)$$

holds for any value of c(k) such that $0 \leq c(k) < 1-2\sigma$. This shows that (4.3.14) is satisfied for k = 1. Now assume that (4.3.14) is true for any $k \geq 1$. Since $0 \leq c(k) < 1-2\sigma$, it is clear that the right hand side of (4.3.14) is negative for any $\sigma < \frac{1}{2}$ and hence the descent property (3.1.6) is satisfied on iteration k. From the definition of $s^{(k+1)}$ (3.1.7) it follows that:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\|g^{(k+1)}\|^2} = -1 + \beta^{(k)} \frac{g^{(k+1)^T} s^{(k)}}{\|g^{(k+1)}\|^2} \quad . \qquad (4.3.15)$$

$\beta^{(k)}$ could either be computed from Category 1 or from Category 2. We shall consider each case separately showing that for both cases (4.3.14) holds.

<u>$\beta^{(k)}$ from Category 1:</u>

If $\beta^{(k)}$ is computed from Category 1, then we have:

$$0 \leq \beta^{(k)} \leq \frac{\|g^{(k+1)}\|^2}{\|g^{(k)}\|^2} \quad . \qquad (4.3.16)$$

Using (3.1.4) we get:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \leq -1 - \sigma\beta^{(k)} \frac{g^{(k)^T} s^{(k)}}{\| g^{(k+1)} \|^2} . \qquad (4.3.17)$$

Then substituting (4.3.16) into (4.3.17) and using the induction assumption that $g^{(k)^T} s^{(k)} < 0$, we get:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \leq -1 - \sigma \frac{g^{(k)^T} s^{(k)}}{\| g^{(k)} \|^2} . \qquad (4.3.18)$$

Using the induction assumption again that

$$\frac{g^{(k)^T} s^{(k)}}{\| g^{(k)} \|^2} \geq -1 - 2\sigma - c(k)$$

with

$$0 \leq c(k) < 1 - 2\sigma$$

we get

$$\frac{g^{(k)^T} s^{(k)}}{\| g^{(k)} \|^2} \geq -2 . \qquad (4.3.19)$$

Substituting (4.3.19) into (4.3.18), we get:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \leq -1 + 2\sigma .$$

Therefore for any $c(k+1)$ such that $0 \leq c(k+1) < 1 - 2\sigma$ we have

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \leq -1 + 2\sigma + c(k+1) . \qquad (4.3.20)$$

On the other hand, substituting (3.1.4) into (4.3.15) also gives

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \geq -1 + \sigma\beta^{(k)} \frac{g^{(k)^T} s^{(k)}}{\| g^{(k+1)} \|^2} \quad . \tag{4.3.21}$$

Then substituting (4.3.16) into (4.3.21) and using the induction assumption that $g^{(k)^T} s^{(k)} < 0$, we get:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \geq -1 + \sigma \frac{g^{(k)^T} s^{(k)}}{\| g^{(k)} \|^2} \quad . \tag{4.3.22}$$

Using the induction assumption again that

$$\frac{g^{(k)^T} s^{(k)}}{\| g^{(k)} \|^2} \geq -2 \qquad \text{(see (4.3.19)}$$

and substituting (4.3.19) into (4.3.22), we get:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \geq -1 - 2\sigma \ .$$

Therefore for any $c(k+1)$ such that $0 \leq c(k+1) < 1 - 2\sigma$, we have:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \geq -1 - 2\sigma - c(k+1) \quad . \tag{4.3.23}$$

Finally combining (4.3.20) and (4.3.23) we have:

$$-1 - 2\sigma - c(k+1) \leq \frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \leq -1 + 2\sigma + c(k+1) \tag{4.3.24}$$

with $\qquad 0 \leq c(k+1) < 1 - 2\sigma$

which is (4.3.14) where k is replaced by k+1.

$\beta^{(k)}$ from Category 2:

If $\beta^{(k)}$ is computed from Category 2, then (4.3.13) holds.

So substituting (3.1.4) into (4.3.15) we have:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \leq -1 - \sigma\beta^{(k)} \frac{g^{(k)^T} s^{(k)}}{\| g^{(k+1)} \|^2} . \qquad (4.3.25)$$

Since $\beta^{(k)}$ is from Category 2 then

$$\beta^{(k)} = \frac{\| g^{(k+1)} \|^2 - g^{(k+1)^T} g^{(k)}}{\| g^{(k)} \|^2} .$$

By substituting this into (4.3.25), we get:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \leq -1 - \sigma \left[\frac{\| g^{(k+1)} \|^2 - g^{(k+1)^T} g^{(k)}}{\| g^{(k)} \|^2} \right] \cdot \frac{g^{(k)^T} s^{(k)}}{\| g^{(k+1)} \|^2}$$

$$= -1 - \sigma \left[1 - \frac{g^{(k+1)^T} g^{(k)}}{\| g^{(k+1)} \|^2} \right] \cdot \frac{g^{(k)^T} s^{(k)}}{\| g^{(k)} \|^2} .$$

Then using (4.3.19) and the left-hand side of (4.3.13), we get:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \leq -1 + 2\sigma - 2\sigma \frac{g^{(k+1)^T} g^{(k)}}{\| g^{(k+1)} \|^2} . \qquad (4.3.26)$$

But we have from (4.3.12) that:

$$-\frac{g^{(k+1)^T} g^{(k)}}{\| g^{(k+1)} \|^2} < \frac{1 - 2\sigma}{2\sigma} \Rightarrow -2\sigma \frac{g^{(k+1)^T} g^{(k)}}{\| g^{(k+1)} \|^2} < 1 - 2\sigma$$

we then let

$$c(k+1) = -2\sigma \frac{g^{(k+1)^T} g^{(k)}}{\| g^{(k+1)} \|^2}$$

and we have:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \leq -1 + 2\sigma + c(k+1) \qquad (4.3.27)$$

with $\quad 0 \leq c(k+1) < 1 - 2\sigma \quad .$

On the other hand, by substituting (3.1.4) into (4.3.15) we also get:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \geq -1 + \sigma\beta^{(k)} \frac{g^{(k)^T} s^{(k)}}{\| g^{(k+1)} \|^2} \quad . \qquad (4.3.28)$$

Then by substituting $\beta^{(k)} = \beta(P.R)$ into (4.3.28) we get:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \geq -1 + \sigma \left[\frac{\| g^{(k+1)} \|^2 - g^{(k+1)^T} g^{(k)}}{\| g^{(k)} \|^2} \right] \cdot \frac{g^{(k)^T} s^{(k)}}{\| g^{(k+1)} \|^2}$$

$$= -1 + \sigma \left[1 - \frac{g^{(k+1)^T} g^{(k)}}{\| g^{(k+1)} \|^2} \right] \frac{g^{(k)^T} s^{(k)}}{\| g^{(k)} \|^2} \quad .$$

Then using (4.3.19) and the left-hand side of (4.3.13) again we get:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \geq -1 - 2\sigma + 2\sigma \frac{g^{(k+1)^T} g^{(k)}}{\| g^{(k+1)} \|^2} \quad .$$

And again if we let

$$c(k+1) = -2\sigma \frac{g^{(k+1)^T} g^{(k)}}{\| g^{(k+1)} \|^2}$$

we get:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \geq -1 - 2\sigma - c(k+1) \qquad (4.3.29)$$

with $\quad 0 \leq c(k+1) < 1 - 2\sigma \quad .$

Combining (4.3.27) and (4.3.29) we get:

$$- 1 - 2\sigma - c(k+1) \leq \frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \leq - 1 + 2\sigma + c(k+1) \tag{4.3.30}$$

with $\quad 0 \leq c(k+1) < 1 - 2\sigma$

which is (4.3.14) with k replaced by k+1.

Therefore no matter what category beta is calculated from we have:

$$- 1 - 2\sigma - c(k+1) \leq \frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \leq - 1 + 2\sigma + c(k+1)$$

with $\quad 0 \leq c(k+1) < 1 - 2\sigma \quad .$

The induction is thus completed showing that the descent property (3.1.6) holds on every iteration of Hybrid 2 when inexact line searches are performed.

(b) Global convergence:

We now proceed to show that Hybrid 2, using algorithm (4.3.7) to compute the update $\beta^{(k-1)}$ at each iteration k, is globally convergent when an inexact line search satisfying (3.1.4) with $\sigma < \frac{1}{2}$ is performed at each iteration. This is done by showing that:

$$\sum_k \cos^2(\theta^{(k)}) = + \infty \tag{4.3.31}$$

holds. Before proving that (4.3.31) holds we need to establish an inequality similar to (4.2.22) for $\| s^{(k)} \|^2$, $k = 1,2,\ldots$.
The following Lemma gives this inequality.

Lemma 4.3.1:

If an inexact line search satisfying (3.1.4) with $\sigma < \frac{1}{2}$ is performed at each iteration k, if $\beta^{(k-1)}$ is computed from algorithm (4.3.7) at each iteration k and if τ in $\gamma^{(k)^2}$ is such that $0 < \tau \leq 1$, then the inequality:

$$\| s^{(k)} \|^2 < \frac{4}{\tau} \| g^{(k)} \|^4 \sum_{\ell=1}^{k} \| g^{(\ell)} \|^{-2} \qquad (4.3.32)$$

holds at each iteration k.

Proof:

We prove this Lemma by induction. For k = 1 inequality (4.3.32) is obviously true. We now assume that it holds for any $k \geq 1$ and we shall show that it then holds for k+1.

On iteration k+1, $\beta^{(k)}$ could either be computed from Category 1 of from Category 2. We shall show that in both cases (4.3.32) holds when k is replaced by k+1.

$\beta^{(k)}$ from Category 1:

From the definition of $s^{(k+1)}$ in (3.1.7) we have:

$$\| s^{(k+1)} \|^2 = \| g^{(k+1)} \|^2 - 2\beta^{(k)} \, g^{(k+1)^T} s^{(k)} + \beta^{(k)^2} \| s^{(k)} \|^2 .$$

Using (3.1.4) we obtain:

$$\| s^{(k+1)} \|^2 \leq \| g^{(k+1)} \|^2 - 2\sigma\beta^{(k)} \, g^{(k)^T} s^{(k)} + \beta^{(k)^2} \| s^{(k)} \|^2 .$$

Then using (4.3.19), (4.3.8) and the induction assumption (4.3.32) we get:

$$\|s^{(k+1)}\|^2 \leq \|g^{(k+1)}\|^2 + 4\sigma\|g^{(k+1)}\|^2 + \frac{4}{\tau}\|g^{(k+1)}\|^4 \sum_{\ell=1}^{k} \|g^{(\ell)}\|^{-2}$$

$$= \|g^{(k+1)}\|^4 \left[(1 + 4\sigma)\frac{1}{\|g^{(k+1)}\|^2} + \frac{4}{\tau}\sum_{\ell=1}^{k} \|g^{(\ell)}\|^{-2}\right] .$$

Because $\sigma < \frac{1}{2}$ then $1 + 4\sigma < 3$ and because $0 < \tau \leq 1$ then $\frac{4}{\tau} \geq 4$.

Therefore $\frac{4}{\tau} > 1 + 4\sigma$ and so we have:

$$\|s^{(k+1)}\|^2 < \frac{4}{\tau}\|g^{(k+1)}\|^4 \sum_{\ell=1}^{k+1} \|g^{(\ell)}\|^{-2}$$

which is (4.3.32) with k replaced by k+1.

$\beta^{(k)}$ from Category 2:

Since $\beta^{(k)}$ is from Category 2 then (4.3.10) holds. We then have:

$$\cos^2(\theta^{(k+1)}) \geq \frac{\tau}{\|g^{(k+1)}\|^2 \sum_{\ell=1}^{k+1} \|g^{(\ell)}\|^{-2}} .$$

But $$\cos^2(\theta^{k+1}) = \frac{\left(g^{(k+1)^T} s^{(k+1)}\right)^2}{\|g^{(k+1)}\|^2 \cdot \|s^{(k+1)}\|^2} .$$

Therefore we have:

$$\frac{\left(g^{(k+1)^T} s^{(k+1)}\right)^2}{\|g^{(k+1)}\|^2 \cdot \|s^{(k+1)}\|^2} \geq \frac{\tau}{\|g^{(k+1)}\|^2 \sum_{\ell=1}^{k+1} \|g^{(\ell)}\|^{-2}} . \qquad (4.3.33)$$

Using Theorem 4.3.2, we have:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\| g^{(k+1)} \|^2} \geq -1 - 2\sigma - c(k+1) > -2$$

$$\Rightarrow 0 > g^{(k+1)^T} s^{(k+1)} > -2 \| g^{(k+1)} \|^2$$

$$\Rightarrow (g^{(k+1)^T} s^{(k+1)})^2 \leq 4 \| g^{(k+1)} \|^4 \quad . \tag{4.3.34}$$

Substituting (4.3.34) into (4.3.33) we obtain:

$$\frac{4 \| g^{(k+1)} \|^2}{\| s^{(k+1)} \|^2} \geq \frac{\tau}{\| g^{(k+1)} \|^2 \sum_{\ell=1}^{k+1} \| g^{(\ell)} \|^{-2}} \quad ,$$

from which we finally obtain:

$$\| s^{(k+1)} \|^2 \leq \frac{4}{\tau} \| g^{(k+1)} \|^4 \sum_{\ell=1}^{k+1} \| g^{(\ell)} \|^{-2}$$

which is (4.3.32) with k replaced by k+1. So no matter what category the update $\beta^{(k)}$ is computed from, inequality (4.3.32) holds for all iterations of Hybrid 2.

A consequence of this Lemma is the following global convergence result.

Theorem 4.3.3:

If the set (3.1.10) is bounded, if f(x) is twice continuously differentiable, and if $\alpha^{(k)}$ is any value satisfying (3.1.4) and (3.1.5) with $\rho < \sigma < \frac{1}{2}$, then equation (3.2.3) holds for Hybrid 2.

Proof:

We prove this theorem by contradiction. Suppose that (3.2.3) is not true, then there exists a positive constant, ϵ say, such that:

$$\| g^{(k)} \| \geq \epsilon > 0 \qquad \text{for all } k \quad . \tag{4.3.35}$$

But $g^{(k)}$ is bounded above in the level set (3.1.10). Hence from (4.3.32) in Lemma 4.3.1, we have:

$$\| s^{(k)} \|^2 \leq c_1 k \tag{4.3.36}$$

where c_1 is a positive constant.

no matter what category $\beta^{(k-1)}$ is computed from,

using the descent property of Theorem 4.3.2 we have:

$$\frac{g^{(k)^T} s^{(k)}}{\| g^{(k)} \|^2} \leq -1 + 2\sigma + c(k) \quad . \tag{4.3.37}$$

But we know that for all k we have $0 < c(k) < 1 - 2\sigma$. Therefore there exists a positive constant μ such that:

$$0 < c(k) \leq \mu < 1 - 2\sigma \quad . \tag{4.3.38}$$

It then follows that (4.3.37) still holds when c(k) is replaced by μ, i.e.

$$\frac{g^{(k)^T} s^{(k)}}{\| g^{(k)} \|^2} \leq -1 + 2\sigma + \mu < 0 \qquad (4.3.39)$$

$$\Rightarrow \cos^2(\theta^{(k)}) \geq (-1 + 2\sigma + \mu)^2 \frac{\| g^{(k)} \|^2}{\| s^{(k)} \|^2} \quad . \qquad (4.3.40)$$

Therefore if $\beta^{(k)}$ is computed from either category we have

$$\cos^2(\theta^{(k)}) \geq c_2 \frac{\| g^{(k)} \|^2}{\| s^{(k)} \|^2} \qquad (4.3.41)$$

where $c_2 = (-1 + 2\sigma + \mu)^2 > 0$.

From (4.3.36) and (4.3.41) it follows that:

$$\cos^2(\theta^{(k)}) \geq c_2 \sum_k \| g^{(k)} \|^2 / \| s^{(k)} \|^2$$

$$\geq c_3 \sum_k k^{-1} \qquad (4.3.42)$$

where c_3 is a positive constant. Hence the series $\sum_k \cos^2(\theta^{(k)})$ diverges. However it is possible to contradict this result using the line search conditions. If Ω is an upper bound on $\| \nabla^2 f(x) \|$, where x is any point in the level set (3.1.10), then we have:

$$g^{(k+1)^T} s^{(k)} \leq g^{(k)^T} s^{(k)} + \Omega \alpha^{(k)} \| s^{(k)} \|^2 \quad .$$

Thus by using (3.1.4) we obtain:

$$\alpha^{(k)} \geq -\frac{1 - \sigma}{\Omega} \frac{g^{(k)^T} s^{(k)}}{\| s^{(k)} \|^2}$$

which can be substituted in (3.1.5), and by using (3.1.6) and the definition of $\cos^2(\theta^{(k)})$, it follows that:

$$f(x^{(k+1)}) \leq f(x^{(k)}) - c_4 \| g^{(k)} \|^2 \cos^2(\theta^{(k)})$$

where $c_4 = \rho(1-\sigma)/\Omega > 0$.

Thus since $f(x)$ is bounded, $\sum_k \| g^{(k)} \|^2 \cos^2(\theta^{(k)})$ is convergent and because $\|g^{(k)}\|$ is also bounded below this contradicts (4.3.42). Since this contradiction arises from (4.3.35), it follows that (4.3.35) is false and therefore (3.2.3) is true, and the global convergence of Hybrid 2 is established.

3.3 Computational Performance of Hybrid 2.

Here also, the N.A.G. routine EØ4DBF, in which the computation of the update $\beta^{(k)}$ was modified to compute $\beta^{(k)}$ as defined in (4.3.7), was used to test Hybrid 2 on the same test problems on which Hybrid 1 and all the other algorithms studied herein were tested. Three different values of τ, namely $\tau = 10^{-4}$, $\tau = 10^{-6}$ and $\tau = 10^{-8}$, were tried in an attempt to find the best possible value for this scalar. The choice of these three values came from the fact that it was noted that the computational efficiency of the method deteriorated as τ got larger than 10^{-4}. The results obtained for each individual case and for each value of τ are given in Appendix 2, Tables (2.i.j) for i = 12, 13 and 14 and j = 1,2,3,...,7, whereas the overall results together with the figures obtained by Hybrid 2 for each of the three values of τ mentioned above compared against the figures obtained by both

the Fletcher-Reeves and the Polak-Ribière methods are given in Appendix 3, Tables (3.12), (3.13) and (3.14). It can be seen from these tables that in terms of smallest total for the index of computational labour (denoted NC), Hybrid 2 seems to provide better results as τ decreases although the difference between $\tau = 10^{-4}$ and $\tau = 10^{-6}$ was very small. The value $\tau = 10^{-8}$ however, provides the best results compared to all the methods described so far herein. The percentages of improvement achieved by Hybrid 2 with $\tau = 10^{-8}$ over the methods of Fletcher-Reeves and Polak-Ribière were as follows:

48% improvement in terms of NC and 38% improvement in terms of CPU time needed to solve all problems considered over the method of Fletcher-Reeves and,

8% improvement in terms of NC and 7% improvement in terms of CPU time needed to solve all problems considered, over the method of Polak-Ribière.

A more detailed discussion of the results obtained by the three values of τ is given in the final Chapter "Discussion and Conclusions".

Chapter 5

More Efficient Hybrid Conjugate Gradient Techniques

1. INTRODUCTION.

Whereas Hybrid 1 had the drawback of being too restrictive in choosing to use the Polak-Ribière update according to its switching criterion from one update to the other, which resulted in an eventual computational inefficiency of the algorithm, Hybrid 2 on the other hand, suffers the drawback of the need to store an extra n-dimensional vector for the trial direction of search $\hat{s}^{(k)}$ at each iteration and the need for some extra computations that resulted in slightly higher CPU time at each iteration. The purpose of this chapter is to try to overcome these difficulties encountered with these two hybrid algorithms, in an attempt to produce an even more efficient hybrid conjugate gradient method that not only out-performs the existing conjugate gradient methods but also out-performs Quasi-Newton methods that are currently the most efficient methods for solving medium size unconstrained optimization problems.

It can be seen from the results obtained for the Beta Test Hybrid method that is described in Chapter 4, that this method is the only one out of the Hybrids discussed so far that did not use any extra CPU time to achieve its improvement over the methods of Fletcher-Reeves and Polak-Ribière. Moreover, the Beta Test Hybrid method was as efficient as the Angle Test Hybrid method and was only slightly less efficient than Hybrid 2. However,

condition (4.3.6), used as a switching criterion for the Beta Test Hybrid algorithm, cannot be used directly to ensure the method achieves the limit (3.2.3) when exact line searches are performed and to ensure a descent property holds on all iterations with which it would be possible to prove global convergence of the algorithm when inexact line searches are performed.

The idea to be developed in this chapter is to attempt to impose some weak enough conditions not to destroy the desirable features of computational efficiency and CPU time savings achieved by the Beta Test Hybrid method, in order to make a proof of global convergence possible and possibly to achieve a better computational efficiency for the resulting algorithm. A careful analysis aimed at seeking these conditions that are to be imposed, has led to the development of a very efficient hybrid algorithm that is not only globally convergent in both cases when exact and inexact line searches are performed throughout the algorithm but is also computationally far more efficient than all the conjugate gradient methods described herein. In fact this hybrid conjugate gradient algorithm is also computationally more efficient than Quasi-Newton algorithms on the problems tested.

2. AN EFFICIENT HYBRID CONJUGATE GRADIENT ALGORITHM : HYBRID 3.

2.1 Construction of the Algorithm.

As it was felt in Chapter 4 that the switching criterion of Hybrid 1 (4.2.3) was too restrictive in choosing to use the Polak-Ribière update, the switching criterion of the Beta Test

Hybrid algorithm (4.3.6) is perhaps too permissive in its choice of using the Polak-Ribière update and may therefore allow the use of large values of the Polak-Ribière update that would make the angle between the negative gradient and the search direction too close to $\pi/2$, which is undesirable because such a situation would provide a fairly small reduction in the function value. Therefore, as a first modification to the Beta Test Hybrid algorithm, we shall introduce a positive parameter μ in (4.3.6) instead of σ, such that $\mu > \sigma$, in order to be able to control the closeness of the angle in question to $\pi/2$ by having more flexibility in choosing the best value of this parameter μ, that leads to better computational results. Thus, instead of using (4.3.6) as a switching criterion, we shall use the following condition:

$$\beta^{(k)}(P.R) \leq \frac{1}{2\mu} \frac{\|g^{(k+1)}\|^2}{\|g^{(k)}\|^2} , \qquad \mu > \sigma \quad . \tag{5.2.1}$$

This modification should make the algorithm more efficient computationally for it uses the best of the values of the Polak-Ribière updates and replaces the larger ones by a Fletcher-Reeves update, that is in this case necessarily smaller, to bound the angle away from orthogonality. However, with condition (5.2.1) alone, a proof of global convergence of the algorithm still seems far from being easy to obtain if possible at all. We shall now seek some further conditions that will make a global convergence proof obtainable.

Let us put ourselves in the situation where we wish to prove global convergence for the Beta Test Hybrid algorithm described

in Chapter 4 with (4.3.6) replaced by (5.2.1) with exact line searches, and investigate what the problem is that makes this proof not obtainable. Following the proofs given for Hybrid 1 and Hybrid 2 we have:

From the definition of $s^{(k)}$ and exact line search, we deduce the equation:

$$\| s^{(k)} \|^2 = \| g^{(k)} \|^2 + \beta^{(k-1)^2} \| s^{(k-1)} \|^2 \quad . \tag{5.2.2}$$

If we also impose the condition that $\sigma < \mu < \frac{1}{2}$ then both the Fletcher-Reeves and the Polak-Ribière updates satisfy:

$$0 \leq \beta^{(k-1)} \leq \frac{1}{2\mu} \frac{\| g^{(k)} \|^2}{\| g^{(k-1)} \|^2} \quad . \tag{5.2.3}$$

It then follows that:

$$\| s^{(k)} \|^2 \leq \left(\frac{1}{2\mu}\right)^{2k} \| g^{(k)} \|^4 \sum_{\ell=1}^{k} \| g^{(\ell)} \|^{-2} \quad . \tag{5.2.4}$$

At this point we would like to establish an inequality similar to that of (4.2.22). The problem in (5.2.4) is that the coefficient $\left(\frac{1}{2\mu}\right)^{2k}$ is k dependent as opposed to the coefficient $\left(\frac{1+\sigma}{1-\sigma}\right)$ in (4.2.22) which is constant for all k. However, we note that one could control the decrease in $\| g^{(k)} \|^2$ by the decrease of $(2\mu)^k$. In other words one could impose the condition that $\| g^{(k)} \|^2$ although not necessarily monotonically decreasing, should not exceed some prescribed multiple of $(2\mu)^k$. This is in fact a quite reasonable condition to impose because all it means is that the solution path generated by the algorithm should not move too

far away from the stationary point once a certain level of closeness is achieved by the current estimate. If at any iteration it is found that $\| g^{(k)} \|^2$ is greater than $(2\mu)^k$ the algorithm should be restarted with a steepest descent direction to restore conjugacy that might have been lost as a consequence of this feature. The condition to be imposed is then the following:

$$\lambda \| g^{(k+1)} \|^2 \leq (2\mu)^{k+1} \quad , \qquad \frac{1}{2} > \mu > \sigma \tag{5.2.5}$$

for some $\lambda > 0$.

Consequently, we suggest the use of the following algorithm to compute $\beta^{(k)}$ in (3.1.7) at each iteration

$$\left.\begin{array}{ll} \text{Step 1 :} & \text{If (5.2.5) holds go to Step 2.} \\ & \text{Otherwise } \beta^{(k)} = 0, \text{ return.} \\ \text{Step 2 :} & \text{If } \beta^{(k)}(\text{P.R}) < 0, \text{ then } \beta^{(k)} = \beta^{(k)}(\text{F.R}), \text{ return.} \\ & \text{Otherwise go to Step 3.} \\ \text{Step 3 :} & \text{If (5.2.1) holds, then } \beta^{(k)} = \beta^{(k)}(\text{P.R}), \text{ return.} \\ & \text{Otherwise } \beta^{(k)} = \beta^{(k)}(\text{F.R}), \text{ return.} \end{array}\right\} \tag{5.2.6}$$

Note that we have missed out Step 2 of algorithm (4.3.7) because the set of numbers that satisfy (5.2.1) includes those that satisfy (4.2.3) and it is therefore not necessary to check whether (4.2.3) is satisfied. This is so, of course, because $\mu < \frac{1}{2}$. The case $\mu \geq \frac{1}{2}$ however, is covered in Hybrid 1, described in Chapter 4, which did not prove to be very efficient computationally.

In what follows, we shall show that when this Hybrid algorithm

Hybrid 3, is used to find the least value of a general twice continuously differentiable function f(x) with bounded level set (3.1.10), a descent property holds on all iterations and the limit (3.2.3) is achieved. Proofs are given for both cases when an exact line search and when an inexact line search satisfying (3.1.4) and (3.1.5) are used.

2.2 Descent Property and Global Convergence of Hybrid 3.

2.2.1 The Case of Exact Line Search.

(a) Descent Property.

This follows from the fact that a descent property holds on all iterations for both the Fletcher-Reeves and the Polak-Ribière methods when the line search is exact, which has already been shown in (3.2.2) and from the fact that a steepest descent direction, by definition obviously satisfies the descent property (3.1.6).

(b) Global Convergence.

Let us now assume that f(x) is twice continuously differentiable, that the level set (3.1.10) is bounded and that an exact line search is performed at each iteration of the algorithm using (5.2.6) to compute its update $\beta^{(k)}$ and show that these conditions are sufficient to provide the limit (3.2.3) for Hybrid 3.

Theorem 5.2.1:

If f(x) is twice continuously differentiable, if the level set (3.1.10) is bounded and if $\alpha^{(k)}$ satisfies (3.1.3) on all iterations, then the limit (3.2.3) is achieved by Hybrid 3.

Proof:

The method of proof to be followed here is similar to that given by Powell (1983) for the global convergence of the Fletcher-Reeves method.

From the definition of $s^{(k)}$ and exact line search, we deduce the equation:

$$\| s^{(k)} \|^2 = \| g^{(k)} \|^2 + \beta^{(k-1)^2} \| s^{(k-1)} \|^2 \quad .$$

Because $\mu < \frac{1}{2}$, then also $\beta^{(k-1)}(F.R) \leq \frac{1}{2\mu} \frac{\| g^{(k)} \|^2}{\| g^{(k-1)} \|^2}$ and therefore all betas computed from (5.2.6) satisfy:

$$0 \leq \beta^{(k-1)} \leq \frac{1}{2\mu} \frac{\| g^{(k)} \|^2}{\| g^{(k-1)} \|^2} \quad . \tag{5.2.7}$$

It therefore follows that:

$$\| s^{(k)} \|^2 \leq \left(\frac{1}{2\mu} \right)^{2k} \| g^{(k)} \|^4 \sum_{\ell=1}^{k} \| g^{(\ell)} \|^{-2} \quad . \tag{5.2.8}$$

Note that condition (5.2.5) holds on all iterations where a restart has not been made and therefore for these iterations we have:

$$\| s^{(k)} \|^2 \leq \frac{1}{\lambda^2} \sum_{\ell=1}^{k} \| g^{(\ell)} \|^{-2} \quad . \tag{5.2.9}$$

For those iterations where a restart has been made, we have:

$$\| s^{(k)} \|^2 = \| g^{(k)} \|^2 \quad . \tag{5.2.10}$$

Therefore, if (3.2.3) is not true, then there exists a constant ε, say, such that:

$$\| g^{(k)} \| \geq \varepsilon > 0 \qquad \text{for all } k \tag{5.2.11}$$

and hence (5.2.9) gives:

$$\| s^{(k)} \| \leq ck \ ,$$

where c is a positive constant. From the definition of $\alpha^{(k-1)}$ and $s^{(k)}$, it follows that the directional derivatives

$$d^{(k)} = s^{(k)^T} g^{(k)} / (\| s^{(k)} \| . \| g^{(k)} \|) \tag{5.2.12}$$

can be written:

$$d^{(k)} = - \| g^{(k)} \| / \| s^{(k)} \| \ ,$$

and because $\| g^{(k)} \|$ is also bounded above in the level set (3.1.10) we have:

$$d^{(k)^2} \begin{cases} \geq c_1/k & \text{if (5.2.9) is true} \\ = 1 & \text{if (5.2.10) is true} \end{cases}$$

which clearly implies that the series $\sum_k d^{(k)^2}$ is divergent. On the other hand, following Powell's proof for the global convergence of the Fletcher-Reeves method given in Chapter 3, Subsection 2.1, by letting Ω be an upper bound on the induced matrix norms $\{ \| \nabla^2 f(x) \| \ ; \ x \text{ in the level set (3.1.10)} \}$, we obtain:

$$f(x) \leq f(x^{(k)}) + (x - x^{(k)})^T g^{(k)} + \frac{1}{2} \Omega \| x - x^{(k)} \|^2 \tag{5.2.13}$$

for any x in the level set (3.1.10), which after some considerations of rates of change of first derivatives gives the inequality:

$$f(x^{(k+1)}) \leq f(x^{(k)}) - d^{(k)^2} \frac{\| g^{(k)} \|^2}{2\Omega} \quad . \tag{5.2.14}$$

The details of these considerations can be found in Subsection 2.1 of Chapter 3. Thus, because f(x) is bounded below in the level set (3.1.10), the series:

$$\sum_k d^{(k)^2} \| g^{(k)} \|^2 \tag{5.2.15}$$

is convergent. But, because of (5.2.11), this contradicts the fact that $\sum_k d^{(k)^2}$ is divergent established above. Because this contradiction arises from the assumption that (5.2.11) is true, it therefore follows that (5.2.11) is false and the limit (3.2.3) must then be achieved. The proof is thus complete.

2.2.2 The Case of Inexact Line Search.

In what follows, Hybrid 3 is shown to have a descent property and to be globally convergent when an inexact line search satisfying (3.1.4) and (3.1.5) is performed. The methods of proof to be followed here are similar to those given by the Al-Baali (1985) for the Fletcher-Reeves method which are reported in Subsection 2.2 of Chapter 3.

(a) Descent Property.

Theorem 5.2.2:

If an $\alpha^{(k)}$ is calculated that satisfies (3.1.4) with $\sigma \in (0, \frac{1}{2})$ for all k such that $g^{(k)} \neq 0$, then the descent property (3.1.6) holds for all such k when Hybrid 3 is used.

Proof:

The method of proof is to show by induction that the inequalities:

$$-\sum_{j=0}^{k-1}\left(\frac{\sigma}{2\mu}\right)^j \leq \frac{g^{(k)^T} s^{(k)}}{\|g^{(k)}\|^2} \leq -2 + \sum_{j=0}^{k-1}\left(\frac{\sigma}{2\mu}\right)^j \qquad (5.2.16)$$

hold for all k such that $g^{(k)} \neq 0$. It will then follow that the descent property (3.1.6) holds for all such k.

(5.2.16) is clearly satisfied for k = 1. We now assume that it is true for any $k \geq 1$. Since $\mu > \sigma$ we have $\sigma/2\mu < \frac{1}{2}$. Therefore since

$$\sum_{j=0}^{k-1}\left(\frac{\sigma}{2\mu}\right)^j < \sum_{j=0}^{\infty}\left(\frac{\sigma}{2\mu}\right)^j = \frac{1}{1-\frac{\sigma}{2\mu}} , \qquad (5.2.17)$$

it is clear that the right-hand side of (5.2.16) is negative and hence the descent property (3.1.6) is satisfied on iteration k. It follows from the definition of $s^{(k+1)}$ (3.1.7) that:

$$\frac{g^{(k+1)^T} s^{(k+1)}}{\|g^{(k+1)}\|^2} = -1 + \beta^{(k)} \frac{g^{(k+1)^T} s^{(k)}}{\|g^{(k+1)}\|^2} . \qquad (5.2.18)$$

Now, using (3.1.4) and (5.2.18) we obtain:

$$-1 + \sigma\beta^{(k)} \frac{g^{(k)^T} s^{(k)}}{\|g^{(k+1)}\|^2} \leq \frac{g^{(k+1)^T} s^{(k+1)}}{\|g^{(k+1)}\|^2} \leq -1 - \sigma\beta^{(k)} \frac{g^{(k)^T} s^{(k)}}{\|g^{(k+1)}\|^2} . \qquad (5.2.19)$$

Then using (5.2.7) and (5.2.19) we obtain:

$$-1 + \left(\frac{\sigma}{2\mu}\right) \frac{g^{(k)^T} s^{(k)}}{\|g^{(k)}\|^2} \leq \frac{g^{(k+1)^T} s^{(k+1)}}{\|g^{(k+1)}\|^2} \leq -1 - \left(\frac{\sigma}{2\mu}\right) \frac{g^{(k)^T} s^{(k)}}{\|g^{(k)}\|^2} ,$$

which by using the induction hypothesis (5.2.16) becomes:

$$-1-\left(\frac{\sigma}{2\mu}\right)\sum_{j=0}^{k-1}\left(\frac{\sigma}{2\mu}\right)^{j}\leq\frac{g^{(k+1)^{T}}s^{(k+1)}}{\|g^{(k+1)}\|^{2}}\leq-1+\left(\frac{\sigma}{2\mu}\right)\sum_{j=0}^{k-1}\left(\frac{\sigma}{2\mu}\right)^{j}$$

which again by using the argument developed in (3.2.13) replacing σ by $\frac{\sigma}{2\mu}$ we obtain:

$$-\sum_{j=0}^{k}\left(\frac{\sigma}{2\mu}\right)^{j}\leq\frac{g^{(k+1)^{T}}s^{(k+1)}}{\|g^{(k+1)}\|^{2}}\leq-2+\sum_{j=0}^{k}\left(\frac{\sigma}{2\mu}\right)^{j}$$

which is (5.2.16) with k replaced by k+1. Thus the induction is complete.

A consequence of this descent property is the following global convergence result.

(b) Global Convergence.

Theorem 5.2.3:

If the level set (3.1.10) is bounded, if f(x) is twice continuously differentiable, if $\alpha^{(k)}$ is any value satisfying (3.1.4) and (3.1.5) with $\rho<\sigma<\frac{1}{2}$ and if $\beta^{(k)}$ is computed from (5.2.6) at each iteration, then the limit (3.2.3) is achieved.

Proof:

From the definition of $s^{(k)}$ in (3.1.7) we obtain:

$$\|s^{(k)}\|^{2}=\|g^{(k)}\|^{2}-2\beta^{(k-1)}\,g^{(k)^{T}}s^{(k-1)}+\beta^{(k-1)^{2}}\|s^{(k-1)}\|^{2}\quad.\tag{5.2.20}$$

It is shown in Theorem 5.2.2 that the descent property (3.1.6)

holds for $\sigma \in (0, \frac{1}{2})$. Therefore from (3.1.4), (5.2.16) and (5.2.17), it follows that:

$$|g^{(k)^T} s^{(k-1)}| \leq -\sigma\, g^{(k-1)^T} s^{(k-1)} \leq \frac{\sigma}{1 - \frac{\sigma}{2\mu}} \|g^{(k-1)}\|^2$$

which can be substituted into (5.2.20) to give:

$$\|s^{(k)}\|^2 \leq \|g^{(k)}\|^2 + \frac{2\sigma}{1 - \frac{\sigma}{2\mu}} \beta^{(k-1)} \|g^{(k-1)}\|^2 + \beta^{(k-1)^2} \|s^{(k-1)}\|^2 ,$$

and finally using (5.2.7) we obtain:

$$\|s^{(k)}\|^2 \leq \|g^{(k)}\|^2 + \frac{\frac{\sigma}{\mu}}{1 - \frac{\sigma}{2\mu}} \|g^{(k)}\|^2 + \beta^{(k-1)^2} \|s^{(k-1)}\|^2$$

$$= \left(\frac{1 + \frac{\sigma}{2\mu}}{1 - \frac{\sigma}{2\mu}}\right) \|g^{(k)}\|^2 + \beta^{(k-1)^2} \|s^{(k-1)}\|^2 .$$

It then follows by induction that:

$$\|s^{(k)}\|^2 \leq \left(\frac{1 + \frac{\sigma}{2\mu}}{1 - \frac{\sigma}{2\mu}}\right) \left(\frac{1}{2\mu}\right)^{2k} \|g^{(k)}\|^4 \sum_{\ell=1}^{k} \|g^{(\ell)}\|^{-2} .$$

Because (5.2.5) holds on all iterations where a restart has not been made, we therefore obtain for these iterations:

$$\|s^{(k)}\|^2 \leq \left(\frac{1 + \frac{\sigma}{2\mu}}{1 - \frac{\sigma}{2\mu}}\right) \cdot \frac{1}{\lambda^2} \cdot \sum_{\ell=1}^{k} \|g^{(\ell)}\|^{-2} . \qquad (5.2.21)$$

For those iterations where a restart has been made on the other hand we have:

$$\|s^{(k)}\|^2 = \|g^{(k)}\|^2 . \qquad (5.2.22)$$

Now, if (3.2.3) is not true, then (5.2.11) is true, and therefore (5.2.21) gives:

$$\| s^{(k)} \|^2 \leq c_2 k$$

where c_2 is a positive constant. From (5.2.16) and (5.2.17) it follows that:

$$\cos^2(\theta^{(k)}) \geq \left(\frac{1 + \frac{\sigma}{\mu}}{1 - \frac{\sigma}{2\mu}} \right)^2 \frac{\| g^{(k)} \|^2}{\| s^{(k)} \|^2} \quad , \tag{5.2.23}$$

where $\theta^{(k)}$ is the angle between the steepest descent direction $-g^{(k)}$ and the search direction $s^{(k)}$ and where $\cos(\theta^{(k)})$ is as defined in (3.2.6). Therefore because $\| g^{(k)} \|$ is also bounded above in the level set (3.1.10) we obtain from (5.2.23):

$$\cos^2(\theta^{(k)}) \begin{cases} \geq c_3/k & \text{if (5.2.21) is true} \\ = 1 & \text{if (5.2.22) is true} \end{cases}$$

which clearly implies that the series $\sum_k \cos^2(\theta^{(k)})$ diverges. However, it is possible to contradict this result using the line search conditions. If Ω is an upper bound on $\| \nabla^2 f(x) \|$ for x in the level set (3.1.10), we have:

$$g^{(k+1)^T} s^{(k)} \leq g^{(k)^T} s^{(k)} + \Omega \ \alpha^{(k)} \| s^{(k)} \|^2 \quad .$$

Thus, by using (3.1.4) we obtain:

$$\alpha^{(k)} \geq - \frac{1 - \sigma}{\Omega} \cdot \frac{g^{(k)^T} s^{(k)}}{\| s^{(k)} \|^2} \ ,$$

which can be substituted in (3.1.5), and by using (3.1.6) and

(3.2.6) it follows that:

$$f(x^{(k+1)}) \leq f(x^{(k)}) - K \frac{\left(g^{(k)^T} s^{(k)}\right)^2}{\| s^{(k)} \|^2}$$

$$= f(x^{(k)}) - K \| g^{(k)} \|^2 \cos^2(\theta^{(k)})$$

where $K = \rho(1-\sigma)/\Omega > 0$. Thus since $f(x)$ is bounded in the level set (3.1.10), $\sum_k \| g^{(k)} \|^2 \cos^2(\theta^{(k)})$ is convergent, and because $\| g^{(k)} \|$ is bounded below, this contradicts the fact that $\sum_k \cos^2(\theta^{(k)})$ is divergent established above. Since this contradiction arises from the assumption that (3.2.3) is not true, then the limit (3.2.3) must be achieved and Hybrid 3 is globally convergent when an inexact line search is used satisfying (3.1.4) and (3.1.5).

It is important to note that in contrast to the proofs given in Chapter 4 for Hybrid 1 and Hybrid 2, the proofs given in this chapter for the descent property and global convergence of Hybrid 3 do not use the fact that the quantity $\beta^{(k)}(P.R)$ used in (5.2.6) actually takes the value (3.1.9) of Polak-Ribière. Therefore, one can use any desired conjugate gradient formula instead of (3.1.9) in (5.2.6) without affecting the validity of these proofs.

2.3 Computational Performance of Hybrid 3.

Here again, the N.A.G. routine EØ4DBF, in which the computation of the update $\beta^{(k)}$ was modified to compute $\beta^{(k)}$ as defined in (5.2.6), was used to test Hybrid 3 on the same test problems on which all the methods described so far were tested. These test problems are described in the final Chapter entitled "Discussion and Conclusions".

Testing the method on any of these problems involves running the program 26 times (once for every different number of variables), each run producing a set of results containing the number of variables, the number of iterations, the number of function evaluations, the index of computational labour, the CPU time needed for the run and the function value at the minimum. These extensive sets of results for Hybrid 3 are given in Appendix 2, Tables (2.15.i) for i = 1,2,3,...,7, whereas the overall results with the comparison of the figures obtained by Hybrid 3 against those obtained by both the Fletcher-Reeves and Polak-Ribière methods are given in Appendix 3, Table (3.15).

Hybrid 3 uses two scalars λ and μ. Although different values of these scalars work better on different types of problems, it was found by trying different values of these scalars that Hybrid 3 achieves its best overall results when $\lambda = 10^{-8}$ and $\mu = 0.1$. The overall total figures obtained with these values of these two scalars, represent 25%, 29% and 24% of the overall total figures obtained by the Fletcher-Reeves method, in terms of: number of iterations (NI), number of function evaluations (NF) and index of computational labour (NC) respectively, and represent 47%, 51% and 44% of the overall total figures obtained by the Polak-Ribière method respectively. This means that Hybrid 3 achives spectacular improvements over both the Fletcher-Reeves and the Polak-Ribière of 75%, 71% and 76%, and of 53%, 49% and 56% respectively, in terms of NI, NF and NC respectively. Furthermore, because of these great improvements of Hybrid 3 over the Fletcher-Reeves and the Polak-Ribière methods we carried on the testing to see

whether Hybrid 3 can outperform Quasi-Newton methods. We used the N.A.G. routine EØ4KBF to obtain the results by the Quasi-Newton described in Gill and Murray (1972). This routine was run on all the test problems considered with up to 100 variables which reduces the number of problems from 182 cases to 63. This is to avoid the computational inefficiency of Quasi-Newton methods that could occur for problems with large dimensions. The results obtained by EØ4KBF on each case separately are given in Appendix 2, Tables (2.18.i) for i = 1,2,...,7 whereas the overall results and the comparison of the figures obtained by H3 and EØ4KBF are given in Appendix 3, Table (3.21). These results show that Hybrid 3 does indeed outperform this Quasi-Newton method by 33% in terms of the index of computational labour which is a great achievement. Further discussions of these results are given in the final Chapter "Discussion and Conclusions".

3. AN EFFICIENT RESTARTING PROCEDURE.

3.1 Defining the Procedure.

When minimizing a non quadratic function, it is usually suggested that a conjugate gradient method should be restarted at regular intervals. Fletcher and Reeves (1964) for example, suggest a restart after every n+1 iterations where n is the dimension of f. Thus when a region is entered where the objective function is very nearly quadratic, accumulated errors from previous iterations due to effects of inaccurate line searches and loss of conjugacy, will not interfere with the expected good behaviour of the algorithm.

However, when n is very large, restarting at regular intervals has very little effect. Various restarting procedures for conjugate gradient methods that take account of the objective function being minimized have been suggested in the literature amongst which are the procedures of Powell (1977) and Shanno (1985). In what follows we describe a different procedure that proved to be computationally very efficient.

First we note that condition (5.2.5) used in Hybrid 3 puts an iteration dependent upper bound on how far the solution path can move away from the stationary point that it is approaching. Therefore if one restarts the algorithm when this condition is no longer satisfied, this should re-establish the normal behaviour of the algorithm. Moreover, when $\beta^{(k)}$ takes the value in (3.1.9), condition (5.2.1) in Hybrid 3 implies:

$$- g^{(k+1)^T} g^{(k)} \leq \left(\frac{1 - 2\mu}{2\mu}\right) \| g^{(k+1)} \|^2 \quad . \tag{5.3.1}$$

And when $\beta^{(k)}(P.R) \geq \beta^{(k)}(F.R)$ we obtain:

$$0 \leq - g^{(k+1)^T} g^{(k)} \leq \left(\frac{1 - 2\mu}{2\mu}\right) \| g^{(k+1)} \|^2 \quad . \tag{5.3.2}$$

By substituting (5.2.5) into (5.3.2) we obtain:

$$0 \leq - g^{(k+1)^T} g^{(k)} \leq \lambda(1 - 2\mu)(2\mu)^k \quad ,$$

and therefore (5.2.1) together with (5.2.5) put a similar upper bound on the loss of orthogonality between successive gradients due to inexact line searches. Consequently it seems logical to impose these two conditions as restarting criteria for conjugate

gradient methods. These criteria not only provide computational improvements of conjugate gradient methods as will be seen in Subsection 3.2 of this chapter, but they also make a descent property hold and a global convergence proof possible for any conjugate gradient method with a non-negative update β, for both cases when either an exact line search or an inexact line search is used.

3.2 Computational Performance of the New Restarting Procedure.

The program used to test these criteria is again the N.A.G. routine EØ4DBF where a modification of its original restart every n+1 iterations, was made to implement these new criteria. We used these criteria with the Fletcher-Reeves method and with that of Polak-Ribière and tested them on all the test problems discussed in this thesis. The extensive results for each of the 182 cases considered, separately are given in Appendix 2, Tables (2.16.i) $i = 1,2,\ldots,7$ for the Fletcher-Reeves method and Tables (2.17.i) $i = 1.2,\ldots,7$ for the Polak-Ribière method, whereas the overall results with the comparison of the figures obtained by the Fletcher-Reeves method with new restarts against those obtained by both straight-forward Fletcher-Reeves and Polak-Ribière methods that restart every n+1 iterations are given in Appendix 3, Table (3.16) and the overall results with the comparison of the figures obtained by the Polak-Ribière method with new restarts against those obtained by both straight-forward Fletcher-Reeves and Polak-Ribière methods that restart every n+1 iterations are given in the same appendix: Appendix 3, Table (3.17). These results show that when these

restart criteria are used for the Fletcher-Reeves method they better the method by 72%, 68% and 63% in terms of NI, NF and NC respectively and they better the Polak-Ribière method by 47%, 43% and 51% in terms of NI, NF and NC, whereas when they are used for the Polak-Ribière method, they better the method by 55%, 49% and 57% in terms of NI, NF and NC respectively, and they better the Fletcher-Reeves method by 76%, 72% and 76% in terms of NI, NF and NC respectively. These results are further discussed in the "Discussion and Conclusions" Chapter.

Chapter 6

Discussion and Conclusions

1. RESULTS DESCRIPTION.

The program used to test all the conjugate gradient methods described in this thesis is the N.A.G. routine EØ4DBF in which the computation of $\beta^{(k)}$ had been modified accordingly. For Hybrid 3 and when using the new restarting criteria of Chapter 5, further modifications in EØ4DBF which originally restarts every n+1 iterations, were made to allow for the setting of $\beta^{(k)}$ to zero when required by the conditions of Hybrid 3 or according to the new restarting criteria. As to testing a variable metric method for comparison with the new hybrid conjugate gradient methods, we used the N.A.G. routine EØ4KBF which contains the quasi-Newton method as described by Gill and Murray (1972).

The test functions used for comparison purposes are the following well known test functions:

Problem 1 : "The Extended Rosenbrock Test Function".

$$F = \sum_{i=1}^{n/2} \left[100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2 \right] .$$

$$n = 2, 20, 40, 60, \ldots\ldots, 440, 460, 480, 500.$$

$$x^{(o)} = (-1.2, 1, -1.2, 1, \ldots., -1.2, 1)^T.$$

Problem 2 : "The Extended Wood Test Function".

$$F = \sum_{i=1}^{n/4} \Big[100(x_{4i-3}^2 - x_{4i-2})^2 + (x_{4i-3} - 1)^2 + 90(x_{4i-1}^2 - x_{4i})^2 + (1 - x_{4i-1})^2 + 10.1\Big\{(x_{4i-2} - 1)^2 + (x_{4i} - 1)^2\Big\} + 19.8(x_{4i-2} - 1)(x_{4i} - 1)\Big].$$

$$n = 4, 20, 40, 60, \ldots, 440, 460, 480, 500.$$

$$x^{(o)} = (-3, -1, -3, -1, \ldots, -3, -1, -3, -1)^T.$$

Problem 3 : "The Extended Miele and Cantrell Test Function".

$$F = \sum_{i=1}^{n/4} \Big[(\exp(x_{4i-3}) - x_{4i-2})^2 + 100(x_{4i-2} - x_{4i-1})^6 + \Big\{\tan(x_{4i-1} - x_{4i})\Big\}^4 + x_{4i-3}^8\Big] .$$

$$n = 4, 20, 40, 60, \ldots, 440, 460, 480, 500.$$

$$x^{(o)} = (1, 2, 2, 2, 1, 2, 2, 2, \ldots, 1, 2, 2, 2)^T.$$

Problem 4 : "The Extended Powell Test Function".

$$F = \sum_{i=1}^{n/4} \Big[(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4\Big] .$$

$$n = 4, 20, 40, 60, \ldots, 440, 460, 480, 500.$$

$$x^{(o)} = (3, -1, 0, 1, 3, -1, 0, 1, \ldots, 3, -1, 0, 1)^T.$$

Problem 5 : "The Extended Dixon Test Function".

$$F = \sum_{i=1}^{n/10} \left[(1 - x_{10i-9})^2 + (1 - x_{10i})^2 + \sum_{j=10i-9}^{10i-1} (x_j^2 - x_{j+1})^2 \right] .$$

$$n = 10, 20, 40, 60, \ldots, 440, 460, 480, 500.$$

$$x^{(o)} = (-2, -2, \ldots, -2)^T.$$

Problem 6 : "The Extended Beale Test Function".

$$F = \sum_{i=1}^{n/2} \left[\left\{ 1.5 - x_{2i-1}(1 - x_{2i}) \right\}^2 + \left\{ 2.25 - x_{2i-1}(1 - x_{2i}^2) \right\}^2 + \left\{ 2.625 - x_{2i-1}(1 - x_{2i}^3) \right\}^2 \right] .$$

$$n = 2, 20, 40, 60, \ldots, 440, 460, 480, 500.$$

$$x^{(o)} = (1, 0.8, 1, 0.8, \ldots, 1, 0.8)^T.$$

Problem 7 : "The Extended Engrall Test Function".

$$F = \sum_{i=1}^{n/2} \left[x_{2i-1}^4 + x_{2i}^4 + 2x_{2i-1}^2 x_{2i}^2 - 4x_{2i-1} + 3 \right] .$$

$$n = 2, 20, 40, 60, \ldots, 440, 460, 480, 500.$$

$$x^{(o)} = (0.5, 2, 0.5, 2, \ldots, 0.5, 2)^T.$$

Testing a method on any of these problems involves running the corresponding version of the program 26 times, each run producing a set of results containing the number of variables (NV), the number of iterations (NI), the number of function

evaluations (NF), the index of computational labour (NC), the CPU time needed to reach the minimum and the function value at the minimum (FV). These extensive sets of results for each method, each problem and each number of variables considered are given in Appendix 2. A summarized version of these results, that is more practical for comparison, is given in Appendix 3. These latter results are obtained by considering for each problem, the totals of the 26 values for each of the numbers of iterations, function evaluations, indices of computational labour and cpu times achieved by each method, computing the ratios of these totals for each of the new methods over those of the existing ones and finally by considering the grand totals (of the totals, for each problem) and by computing the overall ratios in the same manner as above. In this way, it is possible to compare the performance of each of the new methods with those of the existing ones on each problem separately, and overall.

Let the Rosenbrock, Wood, Miele and Cantrell, Powell, Dixon, Beale and Engrall test functions be problems 1, 2, 3, 4, 5, 6 and 7 respectively and let the F.R, the P.R, the Powell hybrid, ORIG1, ORIG2, Hybrid 1, the Shanno, the ATH (τ = 0.1), ATH (τ = 0.01), ATH (τ = 0.001), the BTH, Hybrid 2 ($\tau = 10^{-4}$), Hybrid 2 ($\tau = 10^{-6}$), Hybrid 2 ($\tau = 10^{-8}$), Hybrid 3, F.R with new restart, P.R with new restart and the Quasi-Newton method be methods 1, 2, 3,...., 16, 17 and 18 respectively. The tables in Appendix 2 containing the extensive sets of results are labelled Tables (2-i-j) where 2 denotes the Appendix number, i denotes the method (i = 1,2,...,18) and j denotes the problem (j = 1,...,7), whereas the tables in

Appendix 3 containing the summarized results are labelled Tables (3-i) where 3 denotes the Appendix number and i = 1,...,17 for the 17 first tables denotes the corresponding method. For the Quasi-Newton method (method 18) a different setting of the tables is used because for this method the test problems are used up to 160 variables instead of 500 variables. Using the 7 test problems up to 160 variables, Table (3-18) contains the summarized results of the Fletcher-Reeves, Table (3-19) contains those of the P.R method and their comparison with the results of F.R, Table (3-20) contains the summarized results of the Quasi-Newton method and their comparison with those of both the F.R and P.R methods whereas Table (3-21) contains the results of Hybrid 3 compared to the F.R, the P.R and the Q.N. methods.

2. DISCUSSIONS OF THE RESULTS.

Let us now consider the computational performances of all the conjugate gradient methods dealt with in this thesis, on each problem separately and overall.

First however, we shall describe and compare the performances of the Fletcher-Reeves and Polak-Ribière methods on the seven extended test problems described above. Then all the other methods will be compared with these two methods.

The results obtained for the Fletcher-Reeves and Polak-Ribière methods do indeed confirm the well-known fact that in numerical computations, the Polak-Ribière algorithm is far more successful than that of the Fletcher-Reeves. Our results show that, out of the 182 cases considered (7 test problems each with

26 different numbers of variables) only in 17 cases was the Fletcher-Reeves performance as good as or slightly better than that of the Polak-Ribière. On all the other cases the Polak-Ribière method proved to be far more successful. The greatest improvements of the Polak-Ribière method over that of Fletcher-Reeves were obtained especially on Problems 1, 5 and 6 where the respective improvements were of 72%, 77% and 88% in terms of the NI, 65%, 75% and 79% in terms of the NF, 66%, 79% and 81% in terms of the NC, and, 66%, 79% and 83% in terms of cpu time. The lowest improvement however, was obtained when the methods were tested on the extended Powell test problem. Problem 4 where for the same CPU time the Polak-Ribière method achieved 14%. 13% and 11% improvements over the method of Fletcher-Reeves in terms of the NI, NF and NC respectively. Considering all problems resulting in 182 cases, the Polak-Ribière method achieved overall improvements over the method of Fletcher-Reeves of 47% 44%, 45% and 45% in terms of NI, NF, NC and CPU respectively. These comparison results are given in Appendix 3, Table 3-2.

Another method considered in this thesis was the method extracted from Powell's idea of avoiding negative betas in the Polak-Ribière method. Although this method has failed to better the straight-forward Polak-Ribière method on problems 3, 6 and 7 it did nevertheless achieve some improvements on the other problems. Its best improvement was on problem 2 where it bettered the original method by 13%, 13%, 17% and 24% in terms of the NU, NF, NC and cpu time respectively. However, its overall improvement was of 11%, 10%, 12% and 18% in terms NI,

NF, NC and cpu time respectively. The results of this method compared with those of the Fletcher-Reeves and those of the Polak-Ribière methods are given in Appendix 3, Table 3-3.

As to the method suggested by Shanno that uses an angle test to decide when a conjugate gradient algorithm should be restarted, this method using the Polak-Ribière update to compute $\beta^{(k)}$ achieved better results than Powell's algorithm above but this improvement was achieved at the expense of some extra cpu time. The overall improvement achieved by Shanno's algorithm over the straight-forward Polak-Ribière method are as follows: 13%. 12%, 13% and 3% in terms of NI, NF, NC and cpu time. Therefore it used 15% extra cpu time as compared to Powell's algorithm, to achieve extra 2%, 2% and 1% improvements in terms of NI, NF and NC respectively over the Polak-Ribière method. The results of this method compared with those of the methods of Fletcher-Reeves and Polak-Ribière are given in Appendix 3, Table 3-7.

Having discussed the results of the existing conjugate gradient methods and of those constructed from ideas from outside this thesis, we shall now turn to the discussion of the hybrid conjugate gradient methods that we have constructed in this thesis. We shall consider the performances of these hybrids on each problem separately and overall.

The Results Obtained for Problem 1.

ORIG1 Method : Only in the case of 2 variables, was the performance of the Fletcher-Reeves method slightly better than

that of ORIG1 hybrid method. On all the other 25 cases considered for Problem 1, ORIG1 performed much better than the Fletcher-Reeves method. When compared with the results achieved by the Polak-Ribière method too, only on a few cases (NI = 120 and NI = 420 for example) was the performance of the P.R method better than that of ORIG1 and in most other cases ORIG1 has out-performed the P.R method. The case NV = 2 is interesting. We can see that for one extra function evaluation needed by ORIG1 as compared with the P.R method, ORIG1 obtains an approximation to the function value at the minimum FV = 0, which is 8 decimals better than that obtained by the P.R method. The P.R method obtains FV = 0.5894×10^{-19} whereas ORIG1 obtains FV = 0.1831×10^{-27}. Considering all the 26 cases of Problem 1, ORIG1 betters the performance of the F.R method by 74%. 67%, 68% and 70% in terms of NI, NF, NC and CPU respectively and betters that of the P.R method by 7%, 7%, 7% and 12% in terms of NI, NF, NC and CPU respectively. These improvements are greater than the improvements achieved by both Powell's method and Shanno's method. The results obtained by ORIG1 hybrid method on Problem 1 are given in Appendix 2, Table 2-4-1 and the comparison of these results with the F.R and P.R methods is given in Appendix 3, Table 3-4.

ORIG2 Method : Here again, only on a few cases was the method of F.R or that of P.R slightly better than ORIG2. On all the other cases ORIG2 out-performed both methods. When considering all the cases of Problem 1, ORIG2 performed more or less as well as ORIG1. It bettered the performance of the F.R method by

74%, 67%, 68% and 69% in terms of NI, NF, NC and CPU respectively and bettered the performance of the P.R method by 5%, 7%, 5% and 6% in terms of NI, NF, NC and CPU respectively. The results obtained by ORIG2 on Problem 1 are given in Appendix 2, Table 2-5-1 and the comparison of these results with those of the F.R and P.R methods is given in Appendix 3, Table 3-5.

Hybrid 1 Method : On Problem 1, only on a few cases (NV = 400, 440 or 480 for example) was the performance of Hybrid 1 better than that of the F.R method, but it failed to better it overall (see Tables 2-1-1 and 2-6-1 in Appendix 2). Since the P.R method, as was seen above, performed far better than the F.R method, the performance of Hybrid 1 was much worse than that of the P.R method (see Tables 2-2-1 and 2-6-1 in Appendix 2). We have attempted to explain, in Remark 2 of Chapter 4, some of the reasons for which such inefficiency may occur when using Hybrid 1 and have pointed out in particular that one of the disadvantages of Hybrid 1 was that the angle $\theta^{(k)}$ between the negative gradient and the search direction may tend to be too small to get enough decrease in the function value. We also showed in Figure 4-2-2 of Chapter 4 that this undesirable feature is very likely to occur when Hybrid 1 is used to solve the Rosenbrock problem, and suggested the use of Shanno's angle test to allow for more P.R updates to be used. The implementation of Shanno's angle test within Hybrid 1 has resulted in what we called the "Angle Test Hybrid Method" (ATH), and has indeed helped the situation just described a great deal. The results obtained by Hybrid 1 on Problem 1 with its 26 cases,

are given in Appendix 2, Table 2-6-1 and the comparison results of Hybrid 1 with the methods of both F.R and P.R are given in Appendix 3, Table 3-6.

The Angle Test Hybrid (ATH) Method : We can see from Tables 2-8-1, 2-9-1 and 2-10-1 that the use of this angle test does improve the situation described above a great deal. Indeed the Angle Test Hybrid with $\tau = 0.1$ (ATH1), with $\tau = 0.01$ (ATH2) and with $\tau = 0.001$ (ATH3) was respectively 7%, 8% and 7% better than the P.R method in terms of the index of computational labour. These improvements over the P.R method are however achieved at the expense of approximately 8%, 7% and 6½% extra cpu time respectively. As to their improvements over the F.R method, ATH1, ATH2 and ATH3 were 68%, 68% and 68% better respectively in 63%, 64% and 64% less CPU time. Other values of τ were tried in an attempt to find the best value of τ in the sense the value that provides the greatest improvement in the index of computational labour. The results of these attempts are not given here because it was found that the best value of τ was $\tau = 0.01$ used in ATH2. The results obtained by ATH1, ATH2 and ATH3 on Problem 1 with its 26 cases, are given in Appendix 2, Tables 2-8-1, 2-9-1 and 2-10-1 whereas the comparison results of this method with the three values of τ, with the methods of P.R and F.R are given in Appendix 3, Tables 3-8, 3-9 and 3-10.

The Beta Test Hybrid (BTH) Method: The B.T.H method was constructed in an attempt to avoid the use of the angle test that resulted in the storage of an extra n-dimensional vector

and some more computations on every iteration which lead to some extra cpu time as seen above. The improvements achieved by the ATH method over the P.R method were achieved at the expense of some extra cpu time. The idea was then to find an equivalent condition to the one used in ATH, that is condition 4.3.10, that needs less computation. Condition 4.3.6 did indeed achieve what was required in the sense that it obtained the same improvement in the index of computational labour and instead of using some extra cpu time as ATH did it even needed less. The question of its equivalence with condition 4.3.10 however, remains open. The BTH method was 8% better than the P.R method in terms of the index of computational labour and 5% better in terms of cpu time needed to solve the 26 cases of Problem 1. As to its performance against that of the F.R method on Problem 1, it was 68% better in terms of the index of computational labour and the same percentage better in terms of cpu time. The results achieved by the BTH method on Problem 1 with its 26 cases are given in Appendix 2, Table 2-11-1 and the comparison results comparing the BTH method to both the F.R and P.R methods on Problem 1 are given in Appendix 3, Table 3-11.

Hybrid 2 Method : As it was seen in Chapter 4, condition 4.3.10 ensures a descent property to hold and achieves global convergence of ATH with exact line searches. However, it was not possible to use this condition directly to ensure a descent property to hold with which it would be possible to prove the global convergence of the algorithm when inexact line searches are performed throughout the algorithm. By using both conditions,

4.3.10 and 4.3.6 together as explained in Chapter 4 we derived a new hybrid algorithm that not only achieves some improvements over both the F.R and the P.R methods but also ensures a descent property to hold and achieves global convergence in both cases when either exact or inexact line searches are used throughout the algorithm. Since this algorithm uses condition 4.3.10 in particular various values of the scalar τ were tried in an attempt to find the best value of this parameter. Three values of this parameter were found satisfactory; namely $\tau = 10^{-4}$, $\tau = 10^{-6}$ and $\tau = 10^{-8}$. Even though Hybrid 2 with these three values of τ was some 8%, 8% and 2% better than the P.R method respectively and some 68%, 68% and 67% improvement over the F.R method respectively, in terms of the index of computational labour, it was not, expectedly, the best hybrid in terms of cpu time; the reason being the use of condition 4.3.10. The results of this hybrid with its 3 values of τ on Problem 1 with its 26 cases are given in Appendix 2, Tables 2-12-1, 2-13-1 and 2-14-1 whereas the comparison results are given in Appendix 3, Tables 3-12, 3-13 and 3-14.

Hybrid 3 Method : On Problem 1 in particular, Hybrid 3 was not the best of the hybrids computationally although overall it is. In fact it failed to better the P.R method performance but it was not as bad as Hybrid 1. It could only better the F.R method achieving 68%, 62%, 62% and 62% improvements in terms of NI, NF, NC and CPU respectively. The results of this method on Problem 1 with its 26 cases are given in Appendix 2, Table 2-15-1 and those comparing the method to both the F.R and the P.R method are given in Appendix 3, Table 3-15.

F.R and P.R Methods with New Restart : Using the new restarting procedure with the Fletcher-Reeves method did improve the method itself a great deal but did not bring it to compare favourably with the standard Polak-Ribière method. When used with the Polak-Ribière method this procedure failed to better the method on this particular problem. The results of the F.R and the P.R methods with new restart on the 26 cases of Problem 1 are given in Tables 2-16-1 and 2-17-1 respectively, whereas the comparison results with the straight-forward F.R and P.R methods are given in Appendix 3, Tables 3-16 and 3-17 respectively.

The Results Obtained for Problem 2 :

ORIG1 and ORIG2 hybrid methods both achieved some considerable improvements over both the Fletcher-Reeves and the Polak-Ribière methods reducing even the cpu time needed to solve the 26 cases of this particular problem. ORIG2 was however slightly better than ORIG1. ORIG1 achieved 20% improvement in terms of the index of computational labour and 26% improvement in terms of cpu time over the P.R method whereas ORIG2 achieved 22% and 24% respectively over the same method. As to their comparison with the Fletcher-Reeves method ORIG1 achieved 63% improvement in terms of both the index of computational labour and cpu time whereas ORIG2 achieved 64% improvement in terms of NC and CPU. The results of these two methods on the 26 cases of Problem 2 are given in Appendix 2, Tables 2-4-2 and 2-5-2 whereas the comparison results of these methods against the F.R and P.R methods are given in Appendix 3, Tables 3-4 and 3-5.

Surprisingly however, Hybrid 1 performed extremely well on this particular problem. In fact, apart from Hybrid 3 and when using the new restarting criteria, Hybrid 1 performed better than any other conjugate gradient method described in this thesis. Compared to the P.R method it achieved some 23%, 22%, 27% and 27% in terms of the NI, NF, NC and CPU respectively whereas when compared to the F.R method it achieved 64%, 63%, 66% and 63% in terms of the NI, NF, NC and CPU respectively. The results of this method on this problem are given in Appendix 2, Table 2-6-2 and the comparison results are given in Appendix 3, Table 3-6.

ATH1, ATH2 and ATH3 too were able to better the performances of both the F.R and P.R methods in terms of both the NC and CPU. They achieved some 17%, 20% and 20% improvements in terms of the NC respectively and 1%, 5% and 6% improvements in terms of cpu time respectively over the P.R method and 62%, 63% and 63% in terms of the NC and 50%, 52% and 53% improvements in terms of cpu time respectively over the F.R method. We note that on this particular problem, Problem 2, ATH2, ATH3, BTH, Hybrid 2 ($\tau = 10^{-4}$) and Hybrid 2 ($\tau = 10^{-6}$) gave identical results. BTH however, was the best in terms of cpu time as expected. Hybrid 2 ($\tau = 10^{-8}$) on the other hand, was 12% better than the P.R method in terms of the NC for no extra cpu time to speak of and was 60% better than the F.R method in terms of NC with half the cpu time needed by the F.R method. See the respective Tables for the individual and comparison results in Appendices 2 and 3. The results obtained by Hybrid 3 and by the use of the new restarting procedure were by far better than any other result of the other methods on

this particular problem. Hybrid 3 betters the P.R method by 70%, 66%, 71% and 74% in terms of the NI, NF, NC and cpu time respectively and betters the F.R method by 86%, 84%, 87% and 87% in terms of the NI, NF, NC and cpu time respectively. Also the new restarting procedure applied to the F.R and P.R methods achieved about the same percentages as Hybrid 3 on this problem. The results of these methods applied to Problem 2 are given in Appendix 2 and the comparison results in Appendix 3.

The Results Obtained for Problem 3 :

ORIG1 and Hybrid 1 have failed to better the performance of the Polak-Ribière method on this problem. In fact Hybrid 1 has again failed to better even the Fletcher-Reeves method. ORIG2 on the other hand, did perform better than both the F.R and the P.R method. It achieved 16%, 16%, 22% and 24% improvements over the Polak-Ribière method and 61%, 60%, 64% and 66% improvements over the Fletcher-Reeves method in terms of the NI, NF, NC and cpu time respectively. The results of these methods can be found in Appendices 2 and 3 in the respective Tables.

ATH1 and ATH2 were only slightly better than the P.R method whereas ATH3 showed a fairly considerable improvement. The improvements achieved by those three methods were some 3%, 1% and 10% improvements in terms of NC respectively at the expense of some 7.4%, 9% and 0% extra cpu time respectively over the P.R method. As to their performances against that of the F.R method they were 25%, 23% and 30% better in terms of the N.C respectively and 14%, 11% and 20% better in terms of cpu time respectively.

The B.T.H. method on the other hand, achieved a 7% improvement in terms of the N.C and 5% improvement in terms of cpu time over the P.R method and 28% in terms of NC and 24% in terms of cpu time improvements over the F.R method.

As to Hybrid 2 with $\tau = 10^{-4}$, $\tau = 10^{-6}$ and $\tau = 10^{-8}$, it achieved 8%, 7% and 2% improvements over the P.R method in terms of N.C. for the three values of τ respectively at the expense of some 1%, 1% and 6.5% extra cpu time. Compared to the F.R method, Hybrid 2 for these respective values of the scalar τ, achieved 28%, 28% and 23% improvements in terms of the N.C and some 19%, 19% and 14% improvements in terms of the cpu time respectively. See the respective Tables in Appendices 2 and 3 for the explicit results of these method on Problem 3. Hybrid 3 performed as well as the P.R method on this problem in terms of NC and cpu time and achieved 3% improvement in terms of both NI and NF. It however, bettered the Fletcher-Reeves method by 26%. 22%, 22% and 19% in terms of NI, NF, NC and cpu time respectively. The results obtained by this method on this problem with its 26 cases are given in Appendix 2, Table 2-15-3 and those comparing the method to both the F.R and P.R methods are given in Appendix 3, Table 3-15. As to the new restarting procedure used with the F.R and P.R methods, it has in both cases been able to out-perform the straight-forward Fletcher-Reeves method, but failed in both cases to better the performance of the straight-forward Polak-Ribière method. Their results are given in Appendices 2 and 3.

The Results Obtained for Problem 4 :

On this particular problem, most of the hybrid methods have either failed to better the performance of the P.R method or only achieved a slight improvement. The only hybrid method that achieved a very good improvement was Hybrid 3. Also the new restarting procedure achieved excellent results on this problem. Let us look at these results in more detail. ORIG1 has achieved a 3% improvement in terms of the NC and 11% in terms of cpu time over the P.R method and 14% in terms of the NC and 11% in terms of cpu time over the F.R method. ORIG2 did slightly better than that by achieving 7% and 18% improvements in terms of the NC and 11% and 11% improvements in terms of cpu time over the P.R and F.R methods respectively. Hybrid 1 on the other hand, was again worse than the P.R method but was able to better the performance of the F.R method by 4% in terms of the NC for 7.4% extra cpu time. ATH1 was unable to beat the performance of the P.R method and was some 7% better than the F.R method in terms of the NC at the expense of some 23% extra cpu time. ATH2 and Hybrid 2 with $\tau = 10^{-8}$ were able to out-perform the method of P.R achieving 4% and 8% improvements respectively in terms of the NC at the expense of some 16% and 7.4% extra cpu time respectively. As to ATH3, BTH, Hybrid 2 ($\tau = 10^{-4}$) and Hybrid 2 ($\tau = 10^{-6}$) performed as well as the P.R method did on this particular problem and were 10% better than the F.R method in terms of the NC. The B.T.H method was the only method out of these latter methods that did not use any extra cpu time.

Hybrid 3 however, achieved some excellent results on this problem. Its measured improvement over the P.R method was represented by 68%, 66%, 74% and 75% gains in terms of the NI, NF, NC and cpu time respectively, whereas over the Fletcher-Reeves method it achieved some 73%, 71%, 77% and 75% improvements in terms of the NI, NF, NC and cpu time respectively. They are in fact the improvements achieved on this problem and on Problem 2 that made Hybrid 3 the best of the conjugate gradient methods discussed in this thesis. The new restarting procedure applied to the methods of F.R and P.R also produced results as excellent as those of Hybrid 3 on this particular problem.

The Results Obtained for Problem 5 :

On this problem, apart from Hybrid 3 and the P.R method with new restart which achieved some reasonable improvements over the method of P.R all the other hybrids had more or less about the same performance as those of the P.R method, the best of which was ORIG2 which achieved 4%, 3%, 2% and 6% improvements in terms of NI, NF, NC and cpu time respectively over the P.R method and 78%, 76%, 80% and 80% improvements in terms of the NI, NF, NC and cpu time respectively over the F.R method. Hybrid 3 however, achieved 9%, 8%, 9% and 9% improvements in terms of the NI, NF, NC and cpu time respectively over the P.R method and 79%, 77%, 81% and 81% improvements over the F.R method in terms of the NI, NF, NC and cpu time respectively. The new restarting procedure applied to the P.R method performed about as well as Hybrid 3, whereas when applied to the F.R method it failed to improve on the straight-

forward P.R method. The results are given in Appendices 2 and 3.

The Results Obtained for Problems 6 and 7 :

We put those two problems together for the reason that the P.R method has performed so well on these problems that no other method has been able to out-perform it. We therefore only refer to the results given in the Appendices 2 and 3 which are self explanatory.

Discussion of the Overall Results :

The overall performances of these Hybrid methods compared to those of the F.R and P.R methods were as follows:-

ORIG1 achieved some 49%, 46%, 49% and 52% improvements over the method of F.R in terms of NI, NF, NC and cpu time respectively and obtained some 4%, 4%, 6% and 12% improvements over the P.R method in terms of the NI, NF, NC and cpu time respectively. This shows that the idea of replacing the negative updates of the P.R method by F.R updates was a good one but still not as good as that of replacing them by zero as Powell suggested. The idea here however, was to attempt to adapt the method to allow a possibility of a global convergence proof. The use of the Hestenes and Stiefel update in the algorithm which resulted in the algorithm ORIG2 did improve the results a great deal but did not improve the method any further towards finding a global convergence proof. ORIG2 was some 10%, 10%, 12% and 15% better than the P.R method and some 52%, 50%, 52% and 53% better than the F.R method in terms of the NI, NF, NC and cpu time respectively. The idea used in Hybrid 1 however, had led us to find a global

convergence proof of the algorithm but was too restrictive in its choice of updates which resulted in a fairly poor computational result. In fact Hybrid 1 was the only method out of the new hybrids that failed to improve on the Polak-Ribière method. It was 23%, 21%, 20% and 21% worse than the Polak-Ribière and 30%, 29%, 32% and 30% better than the Fletcher-Reeves method in terms of the NI, NF, NC and cpu time respectively. The use of the angle test together with Hybrid 1 that resulted in the ATH method did improve the computational performance of the method but we were only able to prove the global convergence of the method with exact line searches. When inexact line searches are used the improvement achieved by ATH is not only at the expense of some extra cpu time but also at the expense of losing the global convergence that was achieved for Hybrid 1. However, it was necessary to follow this path of ideas to find the best possible hybrid in both theoretical and computational terms. ATH1 achieved 46%, 44%, 46% and 36% improvements over the F.R method in terms of the NI, NF, NC and cpu time respectively. Compared to the P.R method it achieved 2% improvement in terms of the index of computational labour but needed a few more iterations and function evaluations on top of some 14% extra cpu time. When $\tau = 0.01$, the ATH method only needed some 10% extra cpu time to achieve 3%. 3% and 7% improvements over the P.R method in terms of the NI, NF and NC respectively leading to the best value of τ for ATH. For $\tau = 10^{-6}$ ATH was only slightly worse than with $\tau = 10^{-4}$ achieving for some 10% extra cpu time some 3%, 4% and 6% improvements over the P.R method in terms of the NI, NF and NC respectively.

Over the F.R method it achieved 48%, 46%, 48% and 39% improvements in terms of NI, NF, NC and cpu time respectively.

The idea of using the beta test instead of the angle test was interesting for it achieved roughly the same results as did the best ATH but with much less cpu time. It, in fact, achieved 3%, 3%, 6% and 3% improvements over the P.R method and 48%, 46%, 48% and 47% improvements over the F.R method in terms of NI, NF, NC and cpu times respectively. Up to now the global convergence in the case of inexact line search that was achieved for Hybrid 1 had not yet been restored. This was however, again obtained by using the angle test and the beta test simultaneously which resulted in what we called Hybrid 2. For $\tau = 10^{-4}$ and $\tau = 10^{-6}$ roughly the same results were obtained, improving the P.R method by 3%, 3% and 4% respectively, and 5% at the expense of some 9% extra cpu time, in terms of NI, NF and NC respectively and improving the F.R method by 48%, 46%, 48% and 40% in terms of NI, NF, NC and cpu time respectively. For $\tau = 10^{-8}$ better overall results were obtained improving on the P.R method by 3%, 3%, 7% at the expense of some 7% extra cpu time in terms of NI, NF and NC and improving on the F.R method by 48%, 46%, 49% and 41% in terms of NI, NF, NC and cpu time respectively. Although ORIG2 had obtained better results than Hybrid 2 with $\tau = 10^{-8}$, Hybrid 2 has the advantage over ORIG2 that it ensures a descent property to hold on each iteration and achieves global convergence whereas ORIG2 does not. Also ATH2 performed as well as Hybrid 2 with $\tau = 10^{-8}$ but not only is Hybrid 2 globally convergent even when inexact line searches are used but it also achieves its improvements

in less cpu time than ATH2. This therefore makes Hybrid 2 with $\tau = 10^{-8}$ our best Hybrid method so far.

The modification of condition (4.3.6) in Chapter 4 and its replacement by condition 5.2.1 of Chapter 5, and the use of condition (5.2.5) of Chapter 5 instead of condition (4.3.10) of Chapter 4 in the algorithm that resulted in Hybrid 3 algorithm turned out to be the best idea theoretically and computationally. Indeed not only did we obtain an algorithm that ensures a descent property to hold on every iteration and that is globally convergent, for both cases when exact or inexact line searches are used throughout the algorithm, but one that also achieves excellent improvement over both the F.R and the P.R methods computationally. Hybrid 3 uses two scalars λ and μ. Although different values of these scalars work better in different types of problems, Hybrid 3 achieves its best overall results when $\lambda = 10^{-8}$ and $\mu = 0.1$. It betters the F.R method by 75%, 71%, 76% and 75% and betters the P.R method by 53%, 48%, 56% and 55% in terms of NI, NF, NC and cpu time respectively. Good results were also obtained by using the new restarting procedure with the F.R and P.R methods. The F.R method with new restart bettered the straight-forward method of F.R by 72%, 68%, 73% and 72% and bettered the P.R method by 47%, 43%, 50% and 49% in terms of NI, NF, NC and cpu time respectively whereas the P.R method with new restart bettered the straight-forward P.R method by 55%, 49%, 56% and 52% and bettered the F.R method by 76%, 72%, 76% and 74% in terms of NI, NF, NC and cpu time respectively. Hybrid 3 is therfore the best of all the conjugate gradient methods described herein not only because

it achieves the same results as the P.R method with new restart for slightly less cpu time but also because the P.R method with new restart can only be shown to be globally convergent if its update does not at any stage become negative which is obviously not always true.

3. COMPARISON OF HYBRID 3 WITH A QUASI-NEWTON METHOD.

Quasi-Newton methods have always been considered to be far more successful than conjugate gradient methods in numerical computations. Their drawback however, is their need to store an n×n matrix at every iteration to approximate the Hessian of the objective function. This restricts the application of the method to only medium size problems. Because of the considerable improvement of conjugate gradient methods achieved by Hybrid 3 in this thesis, it is felt that it is worthwhile comparing the performance of Hybrid 3 to that of a quasi-Newton method on the test problems considered in this thesis. We used the N.A.G. routine EØ4KBF which contains the code for the quasi-Newton method as described by Gill and Murray (1972). First, an attempt was made to solve all the 182 cases resulting from the seven extended test problems considered in this thesis, but it was found that for some problems such as Problem 2 (Wood), the method failed to solve cases of higher than 160 variables because of the considerable amount of cpu time needed. To solve the Wood test problem with 160 variables, EØ4KBF needed over 6 hours of cpu time which is incredibly high, whereas the Fletcher-Reeves method only needed about 18.4 seconds of cpu time. Therefore, it was

decided to compare the results of the methods considering the seven test problems with up to 160 variables. In Appendix 3, Tables 3-18, 3-19, 3-20 and 3-21 show the summarized results of the F.R, P.R, Q.N and Hybrid 3 methods respectively. These tables also contain the ratios comparing the methods to each other with the problems considered with up to 160 variables. First we note that the Quasi-Newton methods may need more iterations than a conjugate gradient method but at the same time improve a great deal the index of computational labour. This feature is clearly illustrated in Table 3-20, Problem 6 for example. The Q.N method was 40% worse than the P.R method in terms of numbers of iterations but improved the index of computational labour by 55%. As to its comparison with Hybrid 3, Hybrid 3 managed to better the Q.N method overall by 66%, 77%, 33% and 99.3% in terms of NI, NF, NC and CPU respectively.

We also note that the Q.N method had failed to better the P.R method overall (see Table 3-20) and therefore we feel that the situation with the considered test problems is not too realistic and it is therfore only expected that Hybrid 3 would out-perform the Q.N method since Hybrid 3 already out-performs the P.R method. However, we note that the Q.N method has performed rather poorly on Problem 1 and therefore we constructed a new set of results considering only problems 2 to 7. These results that are given in Table 3-22, Appendix 3, do reflect the usual situation where the P.R method out-performs the F.R, the Q.N method out-performs the P.R method and more important, Hybrid 3 still out-performs the Q.N. method.

By considering the test problems for medium size, i.e. up to 160 variables, we draw out another interesting feature of Hybrid 3 by comparing its improvements over the P.R method for the up to 500 variable problems with those for the up to 160 variable problems. The improvements achieved for the former are greater than those achieved for the latter. Consequently we can conclude by saying that Hybrid 3 works even better when the number of variables increases, i.e. for high dimensional problems. This is a very useful aspect of the method.

Appendix 1

Proofs of Theorems Given in Chapter 2.

1. PROOF OF THEOREM 2.2.1.

Part 1: Let x^* be an optimal solution of the problem (2.2.4) and suppose that there exists $s \in Z(x^*)$ such that $s^T \nabla f(x^*) < 0$. This means that f can further be reduced in the feasible direction s. In other words, this means that there exists $\tau > 0$ such that:

$$f(x^*) > f(x^* + \alpha s) \qquad \text{for } 0 < \alpha \leq \tau .$$

But s is in $Z(x^*)$, which implies that there exists $\bar{\alpha}$ such that:

$$x^* + \alpha s \in K \qquad \text{for } 0 \leq \alpha \leq \bar{\alpha} .$$

Finally, if we take α such that $0 < \alpha \leq \min\{\tau, \bar{\alpha}\}$, we obtain:

$$f(x^*) > f(x^* + \alpha s) \quad \text{and} \quad x^* + \alpha s \in K$$

which contradicts the fact that x^* is an optimal solution of (2.2.4) and hence the proof of the first part of the theorem is complete.

Part 2: To prove the second part of the theorem, we only need to show that

$$s^T \nabla f(x^*) \geq 0 \qquad \forall\, s \in \overline{Z(x^*)} \Rightarrow x^* \text{ is optimal,}$$

since the inverse implication is already shown in the first part of the proof above.

The assumption that $g_1, \ldots, g_m$ are quasi-convex implies that

the set K of admissible points is a convex set (This is a convexity result). Therefore if $x \in K$ we obtain:

$$\alpha x + (1-\alpha)x^* = x^* + \alpha(x-x^*) \in K \qquad \text{for all } \alpha : 0 \leqslant \alpha \leqslant 1.$$

So, if we take $s = x - x^*$, we have $s \in Z(x^*)$ which by hypothesis means that:

$$(x-x^*)^T \nabla f(x^*) \geqslant 0.$$

But f is pseudo-convex; therefore, by definition of a pseudo-convex function we obtain:

$$(x-x^*)^T \nabla f(x^*) \geqslant 0 \Rightarrow f(x) \geqslant f(x^*)$$

which completes the proof of the second and last part of the theorem.

2. PROOF OF THEOREM 2.5.2.

$$x \in Z \Rightarrow x = x^{(1)} + z \quad \text{where} \quad z \in [z^{(1)},\ldots,z^{(n)}]$$

$$\Rightarrow x = x^{(1)} + \sum_{i=1}^{m} \alpha_i z^{(i)} \quad \text{where } m \leqslant n$$

$$\Rightarrow f(x) = f\left(x^{(1)} + \sum_{i=1}^{m} \alpha_i z^{(i)}\right) .$$

By definition of the quadratic function f, this implies that:

$$f(x) = a + b^T\left[x^{(1)} + \sum_{i=1}^{m} \alpha_i z^{(i)}\right] + \frac{1}{2}\left[x^{(1)} + \sum_{i=1}^{m} \alpha_i z^{(i)}\right]^T C\left[x^{(1)} + \sum_{i=1}^{m} \alpha_i z^{(i)}\right]$$

$$= a + b^T x^{(1)} + \frac{1}{2} x^{(1)^T} C x^{(1)} + \sum_{i=1}^{m}\left[\alpha_i z^{(i)}(Cx^{(1)} + b) + \frac{1}{2}\alpha_i^2 z^{(i)^T} C z^{(i)}\right]$$

$$= f(x^{(1)}) + \sum_{i=1}^{m}\left[\alpha_i z^{(i)^T} \nabla f(x^{(1)}) + \frac{1}{2}\alpha_i^2 z^{(i)^T} C z^{(i)}\right] \qquad (*)$$

Similarly we have:

$$f(x^{(i)} + \alpha_i z^{(i)}) = f(x^{(i)}) + \alpha_i z^{(i)^T}(Cx^{(i)} + b) + \frac{1}{2}\alpha_i^2 z^{(i)^T} Cz^{(i)} .$$

Since $z^{(i)^T} C(x^{(i)} - x^{(1)}) = 0$, then $z^{(i)^T} Cx^{(i)} = z^{(i)^T} Cx^{(1)}$;

and therefore:

$$f(x^{(i)} + \alpha_i z^{(i)}) = f(x^{(i)}) + \alpha_i z^{(i)^T} \nabla f(x^{(1)}) + \frac{1}{2}\alpha_i^2 z^{(i)^T} Cz^{(i)} \qquad (**)$$

Because of equation (**), equation (*) becomes

$$f(x) = f(x^{(1)}) + \sum_{i=1}^{m} \left[f(x^{(i)} + \alpha_i z^{(i)}) - f(x^{(i)}) \right]$$

$$\geq f(x^{(1)}) + \sum_{i=1}^{m} \left[f(x^{(i)} + \alpha_i^* z^{(i)}) - f(x^{(i)}) \right] = f(x^*)$$

and thus the proof is complete.

Appendix 2

Explicit Sets of Results

This appendix contains the tables of the explicit extensive sets of results. Testing any conjugate gradient method on any of the test problems considered in this thesis involves running the corresponding version of the program 26 times, each run producing a set of results containing the number of variables (NV), the number of iterations (NI), the number of function evaluations (NF), the index of computational labour (NC), the cpu time needed to reach the minimum (CPU) and the function value at the minimum (FV). These extensive sets of results are presented in the tables of this appendix as follows:
Tables 2-i-j contains the results of the 26 runs of method i (i = 1,...,17) on the test problem j (j = 1,...,7). It also contains the totals of the 26 runs in the bottom row. In the case of the Quasi-Newton method, the testing of the method on any of the test problems considered in this thesis only involves running the program 9 times (up to 160 variables). The sets of results produced by this method have the same format as those of the conjugate gradient methods. The notation used in this appendix and the next one is as follows:

NV : Number of Variables.

NI : Number of Iterations.

NF : Number of Function Evaluations.

NC : Index of Computational Labour.

CPU : Cpu time.

FV : The Function Value Obtained On Convergence.

C.5123.
C.
C.2.3.
C.2.7
C3

The figures representing the Cpu time are to the 6^{th} decimal of a second. For example the number 156324192 should be read 156.3 seconds or 2 mins. 36.3 secs.

THE EXTENDED ROSENBROCK TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE FLETCHER-REEVES METHOD

NV	NI	NF	NC	CPU	FV
2	30	64	192	64840	0.1132D-18
20	68	140	2940	674957	0.8473D-27
40	65	136	5576	1241256	0.6052D-21
60	67	150	9150	1952171	0.3778D-22
80	77	173	14013	2971258	0.1580D-18
100	83	189	19089	4033481	0.9819D-20
120	83	190	22990	4793818	0.1804D-20
140	105	235	33135	7024407	0.2595D-20
160	117	261	42021	8960484	0.4782D-20
180	74	177	32037	6739122	0.4783D-19
200	79	188	37788	7857354	0.3946D-21
220	68	167	36907	7629157	0.3664D-21
240	84	204	49164	10188917	0.2214D-18
260	83	203	52983	10945863	0.6728D-19
280	82	203	57043	11758400	0.3809D-20
300	77	194	58394	12004704	0.6625D-19
320	81	204	65484	13429246	0.2784D-21
340	82	208	70928	14581035	0.3938D-20
360	76	198	71478	14473047	0.4616D-20
380	95	237	90297	18634275	0.1068D-20
400	107	262	105062	21758017	0.6123D-19
420	89	225	94725	19208354	0.4750D-20
440	82	213	93933	19074731	0.2351D-21
460	78	207	95427	19357379	0.7187D-19
480	78	209	100529	20258698	0.1572D-18
500	90	232	116232	23768797	0.1843D-19
TOTALS	2100	5069	1377517	283383768	

Table 2-1-1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE FLETCHER-REEVES METHOD

NV	NI	NF	NC	CPU	FV
4	35	77	385	104831	0.8013D-24
20	68	146	3066	746409	0.9900D-24
40	86	181	7421	1773015	0.1075D-27
60	126	266	16226	3865756	0.1099D-26
80	210	436	35316	8420825	0.1481D-21
100	206	430	43430	10330467	0.1379D-27
120	259	532	64372	15338790	0.1151D-22
140	292	599	84459	20105139	0.4400D-23
160	232	478	76958	18434256	0.4454D-20
180	429	879	159099	38012331	0.9980D-20
200	451	929	186729	44595626	0.4347D-19
220	480	984	217464	52016684	0.8008D-21
240	486	1000	241000	57811546	0.4918D-28
260	397	820	214020	51196781	0.3316D-19
280	599	1219	342539	82190094	0.3589D-21
300	431	902	271502	64788131	0.3661D-20
320	374	786	252306	60013525	0.2147D-19
340	367	778	265298	62820086	0.1933D-21
360	408	856	309016	73072580	0.7566D-21
380	399	843	321183	75349929	0.1691D-20
400	479	1002	401802	94192784	0.1858D-20
420	530	1100	463100	109190985	0.3495D-20
440	621	1271	560511	132323387	0.2021D-19
460	670	1368	630648	149526557	0.3440D-20
480	721	1470	707070	169734318	0.7875D-20
500	663	1355	670855	162462462	0.3456D-20
TOTALS	10019	20707	6553775	1558417294	

Table 2.1.2

THE MIELE & CANTRELL TEST FUCTION
WITH ANALYTICAL GRADIENT
THE FLETCHER-REEVES METHOD

NV	NI	NF	NC	CPU	FV
4	79	196	980	441291	0.1220D-11
20	77	202	4242	1629716	0.1728D-11
40	82	205	8405	3241434	0.8953D-12
60	69	165	10065	3904201	0.1253D-11
80	92	231	18711	7207860	0.4352D-11
100	78	185	18685	7277770	0.9269D-11
120	106	235	28435	11204181	0.2747D-11
140	89	216	30456	11893148	0.1211D-11
160	74	193	31073	11954371	0.2094D-11
180	95	237	42897	16597995	0.2281D-11
200	98	247	49647	19116321	0.1164D-11
220	86	207	45747	17778571	0.4849D-11
240	74	174	41934	16377514	0.8868D-11
260	99	240	62640	24300743	0.2070D-11
280	77	198	55638	21358449	0.5615D-11
300	105	273	82173	31499075	0.2841D-12
320	96	224	71904	28177893	0.3531D-11
340	96	266	90706	34282741	0.4241D-11
360	80	210	75810	28883900	0.7483D-11
380	90	242	92202	34969296	0.7233D-11
400	87	216	86616	33343680	0.1493D-10
420	69	179	75359	28658095	0.1241D-10
440	81	206	90846	34048597	0.6317D-11
460	103	271	124931	47626876	0.2945D-11
480	80	210	101010	38528989	0.1059D-10
500	99	256	128256	49150443	0.1356D-11
TOTALS	2261	5684	1469368	564253150	

Table 2.1-3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE FLETCHER-REEVES METHOD

NV	NI	NF	NC	CPU	FV
4	130	268	1340	360249	0.2184D-13
20	108	227	4767	1086856	0.2941D-13
40	125	268	10988	2408908	0.1959D-13
60	210	437	26657	5922544	0.6748D-14
80	248	516	41796	9215578	0.1792D-15
100	207	434	43834	9637950	0.2400D-14
120	247	518	62678	13745913	0.3833D-14
140	310	643	90663	19914341	0.2903D-13
160	355	732	117852	25929933	0.6442D-13
180	364	752	136112	29842629	0.6148D-13
200	407	841	169041	37192351	0.3151D-13
220	444	912	201552	44391390	0.3742D-13
240	487	997	240277	52972391	0.1260D-13
260	538	1102	287622	63170152	0.1504D-13
280	567	1160	325960	71454390	0.1572D-14
300	604	1230	370230	81597162	0.1799D-12
320	644	1313	421473	92791265	0.4419D-13
340	687	1402	478082	105320300	0.8224D-14
360	724	1474	532114	117403809	0.3519D-13
380	767	1564	595884	132068697	0.3303D-18
400	571	1170	469170	103741587	0.4280D-13
420	709	1447	609187	134553127	0.2682D-13
440	665	1359	599319	132211297	0.3044D-13
460	736	1503	692883	152296945	0.2712D-13
480	770	1570	755170	165997893	0.2361D-13
500	670	1370	686370	152030540	0.4052D-13
TOTALS	12294	25209	7971021	1757258197	

Table 2.1-4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE FLETCHER-REEVES METHOD

NV	NI	NF	NC	CPU	FV
10	87	190	2090	812365	0.2889D-19
20	52	112	2352	898232	0.2222D-27
40	122	259	10619	4063738	0.8283D-20
60	100	211	12871	4914828	0.1793D-19
80	122	254	20574	7884273	0.2359D-19
100	208	430	43430	16614745	0.1362D-20
120	265	542	65582	25095617	0.4538D-20
140	292	597	84177	32269930	0.1487D-24
160	214	443	71323	27356724	0.1561D-19
180	219	457	82717	31490701	0.1595D-19
200	241	501	100701	38207348	0.8158D-20
220	270	556	122876	46942636	0.9730D-20
240	275	568	136888	51997050	0.3044D-19
260	303	620	161820	61639146	0.1130D-19
280	291	600	168600	64263273	0.1831D-21
300	319	656	197456	75194954	0.1890D-20
320	347	713	228873	87177538	0.9245D-20
340	372	763	260183	99108830	0.2558D-19
360	394	804	290244	110643556	0.1437D-19
380	418	854	325374	123769607	0.7345D-20
400	449	919	368519	140745967	0.1122D-19
420	466	952	400792	153350768	0.2598D-19
440	489	998	440118	167440206	0.1511D-19
460	499	1018	469298	178838060	0.2481D-19
480	527	1073	516113	195855656	0.1164D-19
500	544	1106	554106	211156960	0.1227D-19
TOTALS	7885	16196	5137696	1957732708	

Table 2.1.5

THE EXTENDED BEALE TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE FLETCHER-REEVES METHOD

NV	NI	NF	NC	CPU	FV
2	14	34	102	78333	0.6770D-29
20	26	60	1260	395022	0.2736D-22
40	44	98	4018	1242369	0.3484D-20
60	28	66	4026	1230133	0.1926D-20
80	83	177	14337	4480172	0.1997D-31
100	103	221	22321	6938832	0.4690D-32
120	114	241	29161	8991977	0.3901D-19
140	107	228	32148	9986551	0.3756D-19
160	105	223	35903	11131842	0.1554D-19
180	98	211	38191	11838518	0.1372D-19
200	101	217	43617	13556407	0.6909D-19
220	29	73	16133	4814284	0.2167D-18
240	34	83	20003	6003567	0.1813D-18
260	44	102	26622	8108610	0.6682D-19
280	60	134	37654	11639398	0.9745D-20
300	78	171	51471	16021064	0.9729D-20
320	92	199	63879	19730585	0.2615D-19
340	105	225	76725	23713562	0.1657D-19
360	112	239	86279	26882687	0.1499D-19
380	113	242	92202	28620601	0.4128D-19
400	118	252	101052	31324442	0.1670D-19
420	119	257	108197	33482360	0.2402D-19
440	120	258	113778	35182955	0.2859D-19
460	120	258	118938	36772349	0.3236D-19
480	120	258	124098	38363336	0.3147D-19
500	119	256	128256	39629694	0.3135D-19
TOTALS	2206	4783	1390371	430159650	

Table 2.1.6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE FLETCHER-REEVES METHOD

NV	NI	NF	NC	CPU	FV
2	8	19	57	61564	0.0000D+00
20	8	24	504	133682	0.0000D+00
40	13	35	1435	375031	0.0000D+00
60	12	35	2135	554053	0.0000D+00
80	9	27	2187	569708	0.0000D+00
100	11	32	3232	829767	0.0000D+00
120	12	35	4235	1089169	0.0000D+00
140	14	39	5499	1418954	0.0000D+00
160	13	38	6118	1531300	0.0000D+00
180	15	42	7602	1954061	0.0000D+00
200	13	38	7638	2006604	0.0000D+00
220	13	40	8840	2265610	0.0000D+00
240	12	38	9158	2336289	0.0000D+09
260	13	40	10440	2672838	0.0000D+00
280	12	36	10116	2605205	0.0000D+00
300	10	31	9331	2391748	0.0000D+00
320	9	30	9630	2443517	0.0000D+00
340	10	34	11594	2808130	0.0000D+00
360	12	38	13718	3488736	0.0000D+00
380	12	37	14097	3531279	0.0000D+00
400	11	35	14035	3519873	0.0000D+00
420	11	35	14735	3653909	0.0000D+00
440	12	37	16317	4157439	0.0000D+00
460	13	39	17979	4538303	0.0000D+00
480	12	38	18278	4587867	0.0000D+00
500	13	41	20541	5198793	0.0000D+00
TOTALS	303	913	239451	60723429	

Table 2.1.7

THE EXTENDED ROSENBROCK TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE POLAK-RIBIERE METHOD

NV	NI	NF	NC	CPU	FV
2	33	70	210	79368	0.5894D-19
20	23	52	1092	265901	0.5109D-25
40	23	55	2255	523078	0.2528D-23
60	23	60	3660	812092	0.4174D-26
80	24	67	5427	1176480	0.3881D-24
100	21	64	6464	1352696	0.1607D-24
120	22	63	7623	1642481	0.4094D-30
140	24	68	9588	2041252	0.2387D-27
160	23	65	10465	2229032	0.3698D-23
180	23	69	12489	2594501	0.5666D-30
200	23	66	13266	2850136	0.6221D-24
220	22	69	15249	3162597	0.4104D-28
240	23	71	17111	3594171	0.2623D-28
260	23	72	18792	3939522	0.5984D-29
280	20	67	18827	3809604	0.1104D-23
300	20	73	21973	4345880	0.4632D-22
320	20	69	22149	4455560	0.8756D-22
340	22	75	25575	5138153	0.1342D-25
360	22	80	28380	5670252	0.3047D-33
380	20	67	25527	5176123	0.7250D-22
400	20	68	27268	5484113	0.9034D-22
420	20	69	29049	5811180	0.8887D-34
440	22	78	34398	6819875	0.3448D-26
460	22	82	37802	7384388	0.1645D-26
480	22	76	36556	7276057	0.4281D-26
500	22	77	38577	7702886	0.5224D-26
TOTALS	582	1792	470272	95337378	

Table 2.2.1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE POLAK-RIBIERE METHOD

NV	NI	NF	NC	CPU	FV
4	37	79	395	115752	0.1005D-21
20	46	99	2079	548594	0.2415D-23
40	135	289	11849	3101323	0.3697D-22
60	71	154	9394	2430289	0.2844D-22
80	88	189	15309	3966808	0.7373D-25
100	105	227	22927	5924733	0.3163D-23
120	134	288	34848	9026150	0.5103D-23
140	154	341	48081	12316806	0.2739D-24
160	178	395	63595	16226033	0.2454D-24
180	163	345	62445	16282961	0.2020D-21
200	204	426	85626	22439012	0.1156D-18
220	181	382	84422	21983901	0.4978D-20
240	200	421	101461	26457372	0.2764D-21
260	249	518	135198	35469689	0.5321D-21
280	240	501	140781	36871129	0.3637D-21
300	175	369	111069	28991971	0.3841D-20
320	181	382	122622	32042907	0.3046D-21
340	176	372	126852	33061341	0.6000D-21
360	157	336	121296	31474363	0.4770D-21
380	154	331	126111	32685758	0.2896D-21
400	174	372	149172	38669263	0.4825D-21
420	201	426	179346	46774983	0.3601D-21
440	245	514	226674	59284433	0.3196D-20
460	128	284	130924	33422919	0.6323D-21
480	400	834	401154	105422263	0.9016D-20
500	462	981	491481	128001061	0.1231D-19
TOTALS	4638	9855	3005111	782991014	

Table 2.2.2

THE EXTENDED MIELE & CANTRELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE POLAK-RIBIERE METHOD

NV	NI	NF	NC	CPU	FV
4	58	166	830	331847	0.1936D-11
20	71	181	3801	1519431	0.1929D-12
40	75	192	7872	3142273	0.1225D-11
60	52	144	8784	3444429	0.5846D-12
80	72	170	13770	5598567	0.2843D-12
100	77	184	18584	7552096	0.1125D-10
120	90	221	26741	10796668	0.1006D-10
140	52	158	22278	8602942	0.1106D-10
160	48	113	18193	7427831	0.6271D-11
180	88	229	41449	16434043	0.7495D-11
200	61	179	35979	13982008	0.6106D-11
220	58	146	32266	12989939	0.1018D-10
240	51	128	30848	12384952	0.4884D-11
260	98	265	69165	27359934	0.1510D-11
280	70	186	52266	20684205	0.4993D-12
300	70	177	53277	21359287	0.5483D-11
320	79	204	65484	26120627	0.1714D-10
340	47	125	42625	16850401	0.3766D-11
360	43	113	40793	16151761	0.4861D-11
380	90	256	97536	38177207	0.2002D-10
400	49	130	52130	20649928	0.1101D-10
420	70	203	85463	33235775	0.2127D-10
440	81	223	98343	38591635	0.2070D-10
460	31	91	41951	16094624	0.1039D-11
480	97	241	115921	46767099	0.1301D-10
500	50	137	68637	26874963	0.4137D-11
TOTALS	1728	4562	1144986	453124472	

Table 2.2.3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE POLAK-RIBIERE METHOD

NV	NI	NF	NC	CPU	FV
4	95	197	985	294561	0.2170D-13
20	107	243	5103	1268927	0.4101D-13
40	94	205	8405	2117911	0.8598D-15
60	117	278	16958	4044845	0.6777D-13
80	262	542	43902	11050355	0.1086D-13
100	239	503	50803	12743262	0.6184D-15
120	204	431	52151	12940317	0.4271D-14
140	224	537	75717	17688710	0.1055D-12
160	195	418	67298	16623549	0.1418D-13
180	340	703	127243	32008673	0.1229D-13
200	251	527	105927	26302548	0.1278D-13
220	454	935	206635	51646014	0.2928D-14
240	484	996	240036	59687576	0.5077D-13
260	320	663	173043	42834223	0.1178D-13
280	514	1056	296736	73460512	0.2999D-13
300	607	1240	373240	92722882	0.1346D-13
320	365	753	241713	59896321	0.9186D-16
340	736	1498	510818	127915495	0.3645D-13
360	499	1087	392407	95774080	0.1777D-13
380	405	839	319659	78997018	0.9482D-13
400	804	1638	656838	164146465	0.8892D-14
420	628	1287	541827	135187896	0.1488D-13
440	506	1047	461727	114161192	0.1753D-13
460	604	1238	570718	142820943	0.1027D-13
480	986	2003	963443	240401615	0.8690D-14
500	545	1123	562623	140048518	0.2696D-13
TOTALS	10585	21987	7065955	1756784408	

Table 8.2.4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE POLAK-RIBIERE METHOD

NV	NI	NF	NC	CPU	FV
10	98	224	2464	1003086	0.2184D-24
20	37	81	1701	689156	0.2808D-20
40	86	185	7585	3062522	0.8042D-20
60	52	116	7076	2805256	0.2551D-20
80	52	115	9315	3692209	0.8872D-20
100	67	150	15150	5972472	0.5010D-20
120	109	233	28193	11294926	0.2328D-19
140	114	242	34122	13722239	0.1568D-19
160	76	172	27692	10959327	0.1427D-19
180	57	135	24435	9571936	0.1025D-20
200	65	151	30351	11970942	0.2734D-19
220	46	112	24752	9668451	0.2841D-19
240	50	121	29161	11347059	0.1114D-19
260	58	139	36279	14175218	0.5664D-20
280	44	106	29786	11595568	0.8969D-20
300	43	106	31906	12247847	0.8928D-20
320	53	127	40767	15730455	0.6442D-20
340	49	119	40579	15714331	0.2767D-19
360	46	115	41515	16058358	0.2237D-20
380	54	131	49911	19488692	0.1229D-19
400	66	157	62957	24587498	0.1342D-19
420	68	164	69044	26931728	0.1603D-19
440	99	225	99225	39219902	0.1077D-19
460	87	203	93583	36814325	0.2425D-19
480	106	240	115440	45772488	0.3690D-19
500	100	229	114729	45695658	0.1856D-19
TOTALS	1782	4098	1067718	419791649	

Table 2.2.5

THE EXTENDED BEALE TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE POLAK-RIBIERE METHOD

NV	NI	NF	NC	CPU	FV
2	14	32	96	77597	0.2390D-21
20	10	29	609	184361	0.9438D-22
40	11	36	1476	426874	0.1932D-30
60	11	35	2135	610846	0.1100D-31
80	10	33	2673	756044	0.3775D-21
100	10	35	3535	1007367	0.2568D-30
120	10	36	4356	1231719	0.8121D-22
140	10	37	5217	1464668	0.4371D-21
160	11	42	6762	1893675	0.2092D-30
180	11	38	6878	1948740	0.1502D-29
200	11	38	7638	2136554	0.1321D-30
220	11	39	8619	2414326	0.4479D-31
240	11	40	9640	2710636	0.4450D-31
260	11	40	10440	2945295	0.3485D-31
280	11	41	11521	3237856	0.4410D-31
300	11	42	12642	3527966	0.8458D-31
320	11	41	13161	3704988	0.1752D-30
340	11	40	13640	3847652	0.1230D-31
360	10	39	14079	3920629	0.3194D-23
380	10	38	14478	4059221	0.3776D-23
400	10	39	15639	4363468	0.1039D-29
420	9	40	16840	4662240	0.2531D-32
440	9	41	18081	4922098	0.1692D-20
460	10	41	18901	5221308	0.4680D-24
480	10	41	19721	5436249	0.2827D-21
500	10	42	21042	5796113	0.1714D-20
TOTALS	274	995	259819	72508490	

Table 8.2.6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE POLAK-RIBIERE METHOD

NV	NI	NF	NC	CPU	FV
2	8	18	54	62637	0.0000D+00
20	7	23	483	123846	0.0000D+00
40	8	23	943	253805	0.0000D+00
60	7	24	1464	366673	0.0000D+00
80	7	25	2025	501647	0.0000D+00
100	7	23	2323	582494	0.0000D+00
120	7	23	2783	700384	0.0000D+00
140	8	26	3666	924022	0.0000D+00
160	8	26	4186	1073962	0.0000D+00
180	8	26	4706	1185039	0.0000D+00
200	8	26	5226	1340730	0.0000D+00
220	7	26	5746	1378133	0.0000D+00
240	7	27	6507	1573937	0.0000D+00
260	8	29	7569	1875736	0.0000D+00
280	7	25	7025	1801102	0.0000D+00
300	7	27	8127	1969758	0.0000D+00
320	7	27	8667	2120525	0.0000D+00
340	7	27	9207	2266855	0.0000D+00
360	7	27	9747	2408312	0.0000D+00
380	7	27	10287	2550902	0.0000D+00
400	7	26	10426	2585535	0.0000D+00
420	7	27	11367	2740753	0.0000D+00
440	7	26	11466	2798814	0.0000D+00
460	8	28	12908	3219112	0.0000D+00
480	8	28	13468	3384589	0.0000D+00
500	8	29	14529	3604730	0.0000D+00
TOTALS	192	669	174905	43394032	

Table 2.2-7

THE EXTENDED ROSENBROCK TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE CG/SD POWELL's HYBRID METHOD

NV	NI	NF	NC	CPU	FV
2	32	67	201	104992	0.8004D-25
20	19	45	945	212606	0.2479D-24
40	20	50	2050	431926	0.1084D-27
60	22	56	3416	713401	0.5098D-28
80	21	58	4698	955676	0.3187D-22
100	21	58	5858	1178701	0.4012D-22
120	21	59	7139	1426568	0.3331D-22
140	22	65	9165	1806666	0.3098D-29
160	22	63	10143	1996674	0.5526D-32
180	22	65	11765	2304770	0.1276D-31
200	22	67	13467	2637146	0.1697D-32
220	22	67	14807	2883076	0.2979D-32
240	22	70	16870	3258743	0.1333D-27
260	22	70	18270	3499836	0.1187D-25
280	22	70	19670	3719815	0.1851D-25
300	22	71	21371	4068512	0.6714D-22
320	23	72	23112	4420711	0.7523D-35
340	21	72	24552	4594166	0.1286D-23
360	22	75	27075	5098920	0.5769D-29
380	22	74	28194	5332939	0.1644D-25
400	22	72	28872	5494772	0.5166D-24
420	22	73	30733	5799494	0.2202D-23
440	22	73	32193	6113050	0.2147D-23
460	22	72	33192	6324757	0.5837D-24
480	22	77	37037	6893151	0.4300D-25
500	22	76	38076	7119777	0.5625D-27
TOTALS	574	1737	462871	88390845	

Table 2-3-1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE CG/SD POWELL's HYBRID METHOD

NV	NI	NF	NC	CPU	FV
4	37	79	395	149723	0.1005D-21
20	46	102	2142	515915	0.2370D-23
40	54	117	4797	1145303	0.1768D-19
60	71	154	9394	2222591	0.2844D-22
80	88	189	15309	3635376	0.7373D-25
100	105	227	22927	5440625	0.3208D-23
120	125	267	32307	7674362	0.3817D-24
140	148	312	43992	10475311	0.2419D-21
160	151	319	51359	12202989	0.8712D-21
180	163	345	62445	14902419	0.2020D-21
200	204	426	85626	20427382	0.1156D-18
220	181	382	84422	20024991	0.4978D-20
240	200	421	101461	24069278	0.2764D-21
260	249	518	135198	32339424	0.5321D-21
280	240	501	140781	33633633	0.3637D-21
300	175	369	111069	26303285	0.3841D-20
320	181	382	122622	29210629	0.3046D-21
340	176	372	126852	30257596	0.6000D-21
360	157	336	121296	28626201	0.4770D-21
380	154	331	126111	29761668	0.2896D-21
400	174	372	149172	35176448	0.4825D-21
420	201	426	179346	42651721	0.3601D-21
440	245	514	226674	53456237	0.3196D-20
460	128	284	130924	30398134	0.6323D-21
480	289	606	291486	69234794	0.7068D-20
500	105	240	120240	27393818	0.3026D-20
TOTALS	4047	8591	2498347	591329853	

Table 2-3-2

THE EXTENDED MIELE & CANTRELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE C6/SD POWELL's HYBRID METHOD

NV	NI	NF	NC	CPU	FV
4	68	187	935	401175	0.3262D-11
20	64	170	3570	1342966	0.3873D-11
40	64	152	6232	2384626	0.4600D-11
60	56	158	9638	3557770	0.2817D-11
80	66	183	14823	5496053	0.5893D-11
100	55	135	13635	5156421	0.3740D-11
120	58	147	17787	6696221	0.1151D-10
140	51	142	20022	7437212	0.9181D-11
160	86	215	34615	13104184	0.9657D-11
180	88	263	47603	17410854	0.1208D-10
200	107	296	59496	22082978	0.7409D-11
220	71	179	39559	14969753	0.1353D-10
240	78	213	51333	19074345	0.3848D-11
260	81	222	57942	21540665	0.7006D-11
280	58	154	43274	16132278	0.2189D-12
300	70	178	53578	20175356	0.1276D-10
320	68	189	60669	22471302	0.1920D-10
340	50	140	47740	17691987	0.1202D-10
360	44	109	39349	14824256	0.2476D-11
380	44	122	46482	17214014	0.1045D-10
400	31	83	33283	12364841	0.1012D-10
420	67	189	79569	29548190	0.1671D-10
440	79	208	91728	34546873	0.8796D-11
460	83	221	101881	38203537	0.3908D-11
480	76	195	93795	35497450	0.8649D-11
500	103	287	143787	53313666	0.8311D-11
TOTALS	1766	4737	1212325	452638981	

Table 8-3-3

TENDED POWELL TEST FUNCTION
NALYTICAL GRADIENT
/SD POWELL's HYBRID METHOD

NV	NI	NF	NC	CPU	FV
4	95	197	985	316854	0.2170D-13
20	84	182	3822	890064	0.6335D-15
40	106	224	9184	2084491	0.1121D-16
60	124	280	17080	3760868	0.1842D-13
80	191	404	32724	7348962	0.4898D-13
100	239	503	50803	11395699	0.6200D-15
120	204	431	52151	11632265	0.4271D-14
140	173	374	52734	11653702	0.5458D-14
160	327	683	109963	24491215	0.4756D-14
180	340	703	127243	28593530	0.1229D-13
200	404	833	167433	37597064	0.5219D-13
220	454	935	206635	46460026	0.3001D-14
240	243	511	123151	27365747	0.7157D-13
260	304	632	164952	36667680	0.1459D-13
280	309	644	180964	40238343	0.4775D-16
300	42	112	33712	6749524	0.1448D-12
320	355	734	235614	52269980	0.5785D-15
340	736	1498	510818	114510751	0.3628D-13
360	388	804	290244	64578634	0.5966D-14
380	393	816	310896	69068838	0.2598D-13
400	411	855	342855	76284406	0.6114D-18
420	628	1287	541827	121091233	0.1488D-13
440	481	995	438795	97547260	0.4353D-14
460	516	1068	492348	109529194	0.5497D-13
480	986	2003	963443	215469104	0.8688D-14
500	511	1055	528555	117590886	0.2607D-13
TALS	9044	18763	5988931	1335186320	

Table 2-3-4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE CG/SD POWELL's HYBRID METHOD

NV	NI	NF	NC	CPU	FV
10	98	224	2464	954725	0.2642D-27
20	37	81	1701	639526	0.2808D-20
40	92	198	8118	3037653	0.2429D-23
60	52	116	7076	2572163	0.2551D-20
80	52	115	9315	3450041	0.8872D-20
100	67	150	15150	5531588	0.5010D-20
120	96	207	25047	9275452	0.1728D-19
140	113	240	33840	12504787	0.1822D-19
160	76	172	27692	10094039	0.1427D-19
180	57	135	24435	8845403	0.1025D-20
200	64	149	29949	10890426	0.1145D-19
220	46	112	24752	8890198	0.2841D-19
240	50	121	29161	10520282	0.1114D-19
260	58	139	36279	13061721	0.5664D-20
280	44	106	29786	10681489	0.8969D-20
300	43	106	31906	11368398	0.8928D-20
320	53	127	40767	14638371	0.6442D-20
340	49	119	40579	14669588	0.2767D-19
360	46	115	41515	14859094	0.2237D-20
380	54	131	49911	18016987	0.1229D-19
400	66	157	62957	22814954	0.1342D-19
420	71	167	70307	25541216	0.1263D-19
440	88	204	89964	32796484	0.5581D-20
460	87	203	93583	33998321	0.2425D-19
480	106	240	115440	42248950	0.3690D-19
500	100	229	114729	41847869	0.1856D-19
TOTALS	1765	4063	1056423	383749725	

Table 2-3-5

THE EXTENDED BEALE TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE CG/SD POWELL HYBRID METHOD

NV	NI	NF	NC	CPU	FV
2	14	34	102	82518	0.5657D-24
20	11	31	651	200355	0.2941D-28
40	11	35	1435	422798	0.2981D-28
60	11	35	2135	624446	0.1001D-27
80	11	35	2835	827298	0.1176D-27
100	11	39	3939	1116636	0.1290D-31
120	11	42	5082	1430652	0.5355D-31
140	11	39	5499	1554249	0.3592D-29
160	11	41	6601	1845663	0.7302D-28
180	11	38	6878	1962396	0.4870D-27
200	11	39	7839	2189792	0.6010D-27
220	11	42	9282	2612426	0.2769D-27
240	11	41	9881	2734051	0.7351D-28
260	11	41	10701	2995607	0.4008D-28
280	11	41	11521	3249642	0.4076D-28
300	11	41	12341	3473051	0.5586D-28
320	11	41	13161	3695385	0.8174D-28
340	11	40	13640	3858033	0.5796D-24
360	11	41	14801	4126270	0.1492D-27
380	12	45	17145	4821829	0.9124D-27
400	11	43	17243	4789637	0.5159D-31
420	9	40	16840	4632339	0.1358D-23
440	9	38	16758	4647170	0.8211D-20
460	10	41	18901	5240822	0.5148D-22
480	11	43	20683	5834710	0.2777D-30
500	11	44	22044	6100274	0.1412D-28
TOTALS	285	1030	267938	75060049	

Table 2-3-6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE CG/SD POWELL HYBRID METHOD

NV	NI	NF	NC	CPU	FV
2	8	18	54	68795	0.0000D+00
20	7	23	483	126564	0.0000D+00
40	8	23	943	250725	0.0000D+00
60	7	24	1464	368175	0.0000D+00
80	7	25	2025	506020	0.0000D+00
100	7	23	2323	584127	0.0000D+00
120	7	23	2783	703939	0.0000D+00
140	8	26	3666	927935	0.0000D+00
160	8	26	4186	1087559	0.0000D+00
180	8	26	4706	1193140	0.0000D+00
200	8	26	5226	1336141	0.0000D+00
220	7	26	5746	1384927	0.0000D+00
240	7	27	6507	1569262	0.0000D+00
260	8	29	7569	1891967	0.0000D+00
280	7	26	7306	1792617	0.0000D+00
300	7	27	8127	1959656	0.0000D+00
320	7	27	8667	2123563	0.0000D+00
340	7	27	9207	2277468	0.0000D+00
360	7	27	9747	2387969	0.0000D+00
380	7	27	10287	2524568	0.0000D+00
400	7	26	10426	2529851	0.0000D+00
420	7	27	11367	2758311	0.0000D+00
440	7	26	11466	2779477	0.0000D+00
460	8	28	12908	3199809	0.0000D+00
480	8	28	13468	3412553	0.0000D+00
500	8	29	14529	3603910	0.0000D+00
TOTALS	192	670	175186	43349028	

Table 2.3.7

THE EXTENDED ROSENBROCK TEST FUNCTION
WITH ANALYTICAL GRADIENT
ORIG1 HYBRID METHOD

NV	NI	NF	NC	CPU	FV
2	32	71	213	112004	0.1831D-27
20	21	49	1029	237627	0.2128D-34
40	25	58	2378	529042	0.1270D-24
60	18	55	3355	669867	0.2060D-29
80	19	54	4374	884986	0.1438D-22
100	20	57	5757	1165951	0.3705D-32
120	22	64	7744	1546547	0.1355D-32
140	22	62	8742	1760042	0.1402D-29
160	20	60	9660	1906709	0.1373D-26
180	20	59	10679	2135119	0.1829D-23
200	22	63	12663	2556134	0.2103D-25
220	23	66	14586	2932770	0.4719D-30
240	18	65	15665	2936793	0.5282D-27
260	19	60	15660	3054324	0.1595D-22
280	18	61	17141	3253364	0.1282D-28
300	18	65	19565	3691048	0.1427D-27
320	19	63	20223	3925741	0.3471D-22
340	19	64	21824	4194260	0.0000D+00
360	18	61	22021	4215857	0.5298D-32
380	18	61	23241	4422033	0.1456D-20
400	18	62	24862	4761464	0.9135D-29
420	20	71	29891	5678034	0.1787D-32
440	23	78	34398	6559652	0.1882D-22
460	23	77	35497	6816265	0.1023D-30
480	22	78	37518	7130671	0.9513D-31
500	22	75	37575	7140596	0.5760D-33
TOTALS	539	1659	436261	84216900	

Table 2-4-1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
ORIG1 HYBRID METHOD

NV	NI	NF	NC	CPU	FV
4	37	79	395	124899	0.1005D-21
20	46	102	2142	530493	0.2370D-23
40	59	129	5289	1286686	0.6388D-21
60	71	154	9394	2291117	0.2844D-22
80	88	189	15309	3729396	0.7373D-25
100	105	227	22927	5561556	0.3208D-23
120	125	266	32186	7763378	0.1444D-24
140	147	309	43569	10562356	0.1332D-20
160	165	348	56028	13556822	0.1638D-26
180	163	345	62445	15122377	0.2020D-21
200	204	426	85626	21087197	0.1156D-18
220	181	382	84422	20629143	0.4978D-20
240	200	421	101461	24712945	0.2764D-21
260	249	518	135198	32805266	0.5321D-21
280	240	501	140781	34346759	0.3637D-21
300	175	369	111069	26961467	0.3841D-20
320	181	382	122622	29768057	0.3046D-21
340	176	372	126852	30772453	0.6000D-21
360	157	336	121296	29321915	0.4770D-21
380	154	331	126111	30363369	0.2896D-21
400	174	372	149172	35924512	0.4825D-21
420	201	426	179346	43299053	0.3601D-21
440	245	514	226674	54857552	0.3196D-20
460	128	284	130924	31075557	0.6323D-21
480	155	337	162097	38730462	0.1122D-19
500	139	305	152805	36450297	0.3036D-21
TOTALS	3965	8424	2406140	581635084	

Table 2-4-2

THE EXTENDED MIELE & CANTRELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
ORIG1 HYBRID METHOD

NV	NI	NF	NC	CPU	FV
4	80	205	1025	414555	0.1577D-11
20	69	200	4200	1593604	0.5050D-11
40	70	146	5986	2420114	0.1957D-11
60	68	176	10736	4152092	0.3671D-12
80	81	210	17010	6633037	0.5222D-11
100	79	214	21614	8350652	0.6148D-12
120	107	272	32912	12854009	0.4757D-11
140	68	180	25380	9821347	0.7975D-11
160	70	197	31717	12124729	0.8190D-11
180	54	143	25883	9989000	0.4470D-11
200	118	298	59898	23283279	0.2759D-12
220	46	112	24752	9688941	0.7797D-11
240	65	175	42175	16301750	0.2043D-11
260	110	278	72558	28245089	0.7626D-11
280	79	208	58448	22384469	0.1327D-10
300	22	67	20167	7511638	0.4242D-11
320	51	132	42372	16300747	0.4533D-11
340	57	151	51491	19794080	0.2314D-10
360	52	145	52345	19999603	0.2106D-12
380	83	241	91821	34576567	0.4298D-11
400	69	189	75789	28697423	0.9621D-11
420	58	155	65255	25086697	0.1689D-10
440	56	158	69678	26166525	0.6162D-11
460	80	224	103264	38852285	0.1225D-10
480	56	150	72150	27400925	0.1423D-10
500	81	222	111222	42017907	0.3566D-11
TOTALS	1829	4848	1189848	454661064	

Table 2-4-3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
ORI61 HYBRID METHOD

NV	NI	NF	NC	CPU	FV
4	95	197	985	282212	0.2170D-13
20	89	195	4095	966030	0.1517D-18
40	87	209	8569	1893982	0.3344D-13
60	185	387	23607	5401609	0.2455D-13
80	303	623	50463	11578307	0.2260D-13
100	239	503	50803	11482094	0.6200D-15
120	204	431	52151	11855749	0.4271D-14
140	310	652	91932	20893505	0.2306D-16
160	324	684	110124	24894830	0.5563D-14
180	340	703	127243	29049818	0.1229D-13
200	339	755	151755	33600332	0.9921D-15
220	455	937	207077	47317908	0.3001D-14
240	506	1037	249917	57094515	0.7089D-16
260	301	624	162864	37077988	0.1210D-13
280	300	621	174501	39554426	0.3391D-13
300	342	727	218827	49103896	0.1531D-12
320	432	886	284406	65423103	0.3296D-13
340	736	1498	510818	117116429	0.3620D-13
360	440	910	328510	74750828	0.2316D-13
380	396	821	312801	70616652	0.2272D-13
400	445	919	368519	83304911	0.1711D-13
420	628	1287	541827	123587754	0.1488D-13
440	508	1048	462168	105647894	0.1569D-14
460	927	1891	871751	200048915	0.3346D-13
480	986	2003	963443	220307411	0.8680D-14
500	526	1084	543084	123516822	0.3910D-13
TOTALS	10443	21632	6872240	1566367920	

Table 2-4-4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
ORIG1 HYBRID METHOD

NV	NI	NF	NC	CPU	FV
10	127	280	3080	1224184	0.9614D-20
20	37	81	1701	665770	0.2808D-20
40	74	160	6560	2533356	0.3602D-20
60	52	116	7076	2692454	0.2551D-20
80	52	115	9315	3536273	0.8872D-20
100	67	150	15150	5735607	0.5010D-20
120	103	221	26741	10226386	0.1165D-19
140	106	227	32007	12254833	0.5678D-20
160	76	172	27692	10452289	0.1427D-19
180	57	135	24435	9181268	0.1025D-20
200	60	141	28341	10655102	0.8264D-20
220	46	112	24752	9205514	0.2841D-19
240	50	121	29161	10876081	0.1114D-19
260	58	139	36279	13402860	0.5664D-20
280	44	106	29786	11102109	0.8969D-20
300	43	106	31906	11808415	0.8928D-20
320	53	127	40767	15173145	0.6442D-20
340	49	119	40579	15037902	0.2767D-19
360	46	115	41515	15342417	0.2237D-20
380	54	131	49911	18519888	0.1229D-19
400	66	157	62957	23371841	0.1342D-19
420	75	177	74517	27796076	0.2296D-20
440	90	207	91287	34316973	0.1954D-19
460	87	203	93583	34929212	0.2425D-19
480	106	240	115440	43201435	0.3690D-19
500	100	229	114729	42892323	0.1856D-19
TOTALS	1778	4087	1059267	396133713	

Table 2-4-5

THE EXTENDED BEALE TEST FUNCTION
WITH ANALYTICAL GRADIENT
ORIG1 HYBRID METHOD

NV	NI	NF	NC	CPU	FV
2	14	32	96	74489	0.5590D-23
20	12	34	714	215097	0.5350D-20
40	12	38	1558	452298	0.3634D-20
60	12	38	2318	670442	0.1552D-19
80	12	38	3078	894856	0.2140D-19
100	9	34	3434	958720	0.9158D-22
120	11	39	4719	1333051	0.2730D-21
140	11	40	5640	1594856	0.2592D-20
160	12	43	6923	1954245	0.1157D-19
180	12	41	7421	2104516	0.4517D-19
200	12	41	8241	2352222	0.6806D-19
220	12	41	9061	2592882	0.4711D-19
240	12	43	10363	2949037	0.1849D-19
260	12	43	11223	3170912	0.1139D-19
280	12	43	12083	3422382	0.1161D-19
300	12	44	13244	3707209	0.1503D-19
320	12	44	14124	3953034	0.2044D-19
340	11	40	13640	3835190	0.5798D-24
360	12	42	15162	4297794	0.3699D-20
380	12	42	16002	4515552	0.1718D-22
400	9	38	15238	4167958	0.3663D-21
420	10	40	16840	4669300	0.2266D-18
440	10	40	17640	4897054	0.2264D-19
460	11	44	20284	5612947	0.1674D-24
480	11	44	21164	5853729	0.1243D-20
500	12	47	23547	6570284	0.7583D-20
TOTALS	299	1053	273757	76820056	

Table 2-4-6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
ORIG1 HYBRID METHOD

NV	NI	NF	NC	CPU	FV
2	8	18	54	59010	0.0000D+00
20	7	23	483	125329	0.0000D+00
40	8	23	943	249167	0.0000D+00
60	7	24	1464	361441	0.0000D+00
80	7	25	2025	501284	0.0000D+00
100	7	23	2323	585152	0.0000D+00
120	7	23	2783	692591	0.0000D+00
140	8	26	3666	929896	0.0000D+00
160	8	26	4186	1066857	0.0000D+00
180	8	26	4706	1198487	0.0000D+00
200	8	26	5226	1329822	0.0000D+00
220	7	26	5746	1403938	0.0000D+00
240	7	27	6507	1569608	0.0000D+00
260	8	29	7569	1876768	0.0000D+00
280	8	27	7587	1905537	0.0000D+00
300	7	27	8127	1959482	0.0000D+00
320	7	27	8667	2105591	0.0000D+00
340	7	27	9207	2254187	0.0000D+00
360	7	27	9747	2409696	0.0000D+00
380	7	27	10287	2526101	0.0000D+00
400	7	26	10426	2575818	0.0000D+00
420	7	27	11367	2742333	0.0000D+00
440	7	26	11466	2811943	0.0000D+00
460	8	28	12908	3245355	0.0000D+00
480	8	28	13468	3380527	0.0000D+00
500	8	29	14529	3620922	0.0000D+00
TOTALS	193	671	175467	43486842	

Table 2-4-7

THE EXTENDED ROSENBROCK TEST FUNCTION
WITH ANALYTICAL GRADIENT
ORIG2 HYBRID METHOD

NV	NI	NF	NC	CPU	FV
2	29	68	204	112451	0.4083D-26
20	20	47	987	243265	0.1190D-21
40	24	55	2255	540120	0.4805D-28
60	19	50	3050	676813	0.2786D-19
80	19	54	4374	929436	0.2621D-21
100	20	56	5656	1207264	0.4806D-26
120	21	58	7018	1499555	0.6249D-22
140	22	65	9165	1906847	0.9874D-19
160	23	64	10304	2185768	0.2780D-23
180	22	63	11403	2412900	0.1155D-22
200	22	63	12663	2661072	0.5557D-20
220	21	66	14586	2947178	0.1999D-28
240	19	61	14701	2985660	0.1831D-19
260	19	62	16182	3227944	0.5427D-19
280	19	63	17703	3523108	0.2083D-22
300	21	69	20769	4132612	0.4519D-26
320	19	65	20865	4093341	0.1203D-21
340	23	73	24893	5021351	0.5030D-27
360	19	66	23826	4661739	0.5092D-25
380	19	66	25146	4952411	0.9650D-23
400	20	68	27268	5391624	0.4039D-28
420	22	73	30733	6154635	0.2309D-26
440	23	77	33957	6782313	0.2878D-26
460	22	78	35958	6924011	0.1400D-29
480	21	71	34151	6705534	0.5698D-22
500	22	74	37074	7312098	0.3345D-30
TOTALS	550	1675	444891	89191050	

Table 2.5.1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
ORI62 HYBRID METHOD

NV	NI	NF	NC	CPU	FV
4	38	81	405	177540	0.2012D-25
20	46	101	2121	559679	0.8320D-25
40	74	158	6478	1667198	0.6950D-19
60	73	158	9638	2459090	0.2796D-22
80	88	189	15309	3887730	0.1054D-22
100	105	227	22927	5785732	0.7805D-23
120	125	266	32186	8162809	0.2516D-27
140	148	312	43992	11203238	0.8140D-20
160	165	347	55867	14216611	0.3700D-27
180	83	185	33485	8344803	0.9231D-19
200	205	427	85827	21945547	0.1382D-26
220	200	420	92820	23643224	0.3148D-21
240	200	421	101461	25802842	0.7760D-21
260	249	518	135198	34550759	0.8926D-21
280	241	504	141624	36036521	0.5625D-21
300	168	356	107156	27191950	0.6398D-20
320	179	378	121338	30819195	0.1599D-21
340	179	379	129239	32782877	0.4363D-21
360	141	304	109744	27594369	0.4552D-21
380	138	299	113919	28589762	0.1622D-19
400	174	372	149172	37989042	0.6181D-21
420	201	426	179346	45663081	0.3957D-21
440	247	517	227997	58362809	0.3433D-20
460	115	257	118477	29573132	0.8465D-20
480	188	403	193843	48537818	0.4483D-21
500	113	253	126753	31375775	0.3199D-19
TOTALS	3883	8258	2356322	596923133	

Table 2-5-2

THE EXTENDED MIELE & CANTRELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
ORIG2 HYBRID METHOD

NV	NI	NF	NC	CPU	FV
4	46	116	580	282938	0.1829D-11
20	43	105	2205	876980	0.5450D-12
40	32	85	3485	1353087	0.1057D-11
60	42	111	6771	2572183	0.5404D-12
80	47	110	8910	3558491	0.2450D-11
100	40	103	10403	4034011	0.5352D-14
120	45	116	14036	5418221	0.1031D-11
140	46	122	17202	6658723	0.3126D-11
160	57	163	26243	9891029	0.3410D-11
180	44	121	21901	8398271	0.1343D-11
200	60	158	31758	12291109	0.6794D-12
220	38	105	23205	8711847	0.8505D-11
240	53	147	35427	13667713	0.2806D-11
260	50	131	34191	13223427	0.3887D-11
280	42	108	30348	11797030	0.6540D-12
300	23	70	21070	7906961	0.1492D-10
320	37	96	30816	11940320	0.2154D-11
340	34	92	31372	12068017	0.5126D-12
360	49	125	45125	17259342	0.1252D-10
380	64	172	65532	25197921	0.3919D-12
400	58	154	61754	23475722	0.9919D-11
420	46	124	52204	20028585	0.3854D-11
440	53	136	59976	23147327	0.5883D-11
460	40	107	49327	18940953	0.3875D-11
480	46	123	59163	22359215	0.7673D-13
500	34	99	49599	18699089	0.8642D-11
TOTALS	1169	3099	792603	303758512	

Table 2.5.3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
ORI62 HYBRID METHOD

NV	NI	NF	NC	CPU	FV
4	115	238	1190	400532	0.9963D-14
20	119	259	5439	1342500	0.2341D-14
40	65	149	6109	1447035	0.2096D-13
60	124	274	16714	3966217	0.6271D-13
80	245	504	40824	9904817	0.5874D-13
100	220	471	47571	11274396	0.8806D-19
120	202	427	51667	12300050	0.1413D-14
140	227	478	67398	16046698	0.1811D-13
160	289	612	98532	23420994	0.2725D-14
180	381	787	142447	34063916	0.2332D-13
200	254	534	107334	25486173	0.4611D-13
220	469	963	212823	51369737	0.9243D-14
240	407	849	204609	48926227	0.3473D-14
260	263	579	151119	35302085	0.6544D-13
280	458	949	266669	63914718	0.1785D-14
300	441	959	288659	67835079	0.2068D-13
320	510	1054	338334	81172750	0.9850D-13
340	550	1127	384307	92571560	0.1827D-13
360	724	1476	532836	128015112	0.1104D-12
380	473	976	371856	88885773	0.4758D-13
400	522	1083	434283	104101839	0.5131D-14
420	494	1025	431525	103119876	0.1830D-12
440	510	1061	467901	111681612	0.4855D-13
460	722	1476	680436	163212403	0.1252D-13
480	630	1294	622414	149682955	0.1156D-13
500	574	1178	590178	141080856	0.1416D-13
TOTALS	9980	20782	6563174	1570525910	

Table 2-5-4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
ORIG 2 HYBRID METHOD

NV	NI	NF	NC	CPU	FV
10	75	173	1903	793532	0.1732D-20
20	37	81	1701	674547	0.3497D-20
40	72	154	6314	2449869	0.8182D-21
60	54	120	7320	2808139	0.7015D-20
80	51	113	9153	3521471	0.2325D-19
100	66	147	14847	5679290	0.7926D-20
120	101	216	26136	10073219	0.1773D-19
140	112	239	33699	12987227	0.1666D-19
160	72	164	26404	10004090	0.1650D-19
180	52	124	22444	8414411	0.1480D-19
200	61	144	28944	10869448	0.9158D-20
220	48	116	25636	9617294	0.7752D-20
240	50	121	29161	10915115	0.7682D-20
260	68	158	41238	15629846	0.1970D-19
280	44	106	29786	11183571	0.1062D-19
300	43	106	31906	11887242	0.1598D-19
320	53	127	40767	15307897	0.6574D-20
340	49	119	40579	15212652	0.2914D-19
360	46	115	41515	15451330	0.2003D-20
380	53	129	49149	18361913	0.1085D-19
400	63	152	60952	22772088	0.1876D-19
420	66	160	67360	25340703	0.1364D-19
440	93	213	93933	35810892	0.4220D-20
460	87	202	93122	35304764	0.8578D-20
480	102	231	111111	42296873	0.6093D-20
500	97	224	112224	42462098	0.1190D-19
TOTALS	1715	3954	1047304	395829521	

Table 2-5-5

THE EXTENDED BEALE TEST FUNCTION
WITH ANALYTICAL GRADIENT
ORIG 2 HYBRID METHOD

NV	NI	NF	NC	CPU	FV
2	14	32	96	85436	0.3286D-23
20	11	32	672	212490	0.3085D-21
40	11	34	1394	422887	0.1158D-22
60	11	36	2196	653971	0.4252D-20
80	11	36	2916	853170	0.1234D-20
100	11	36	3636	1081223	0.1304D-27
120	12	41	4961	1460793	0.4754D-20
140	12	42	5922	1724276	0.8560D-20
160	11	38	6118	1764596	0.8925D-22
180	12	43	7783	2261704	0.9885D-20
200	12	41	8241	2426550	0.5638D-19
220	12	41	9061	2633400	0.3110D-19
240	12	43	10363	3005269	0.1467D-19
260	12	43	11223	3269967	0.9256D-20
280	12	43	12083	3506938	0.7233D-20
300	12	44	13244	3794615	0.6607D-20
320	12	44	14124	4018253	0.6805D-20
340	10	38	12958	3717134	0.6328D-22
360	10	39	14079	3961387	0.1013D-22
380	10	39	14859	4242792	0.2712D-20
400	11	39	15639	4508268	0.5213D-27
420	11	45	18945	5353501	0.9669D-24
440	12	47	20727	5882899	0.2664D-20
460	12	47	21667	6180057	0.1154D-19
480	12	47	22607	6445118	0.2362D-19
500	12	48	24048	6820877	0.3519D-19
TOTALS	300	1058	279562	80287571	

Table 2-5-6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
ORIG2 HYBRID METHOD

NV	NI	NF	NC	CPU	FV
2	8	19	57	61365	0.0000D+00
20	7	21	441	124179	0.0000D+00
40	8	23	943	262485	0.0000D+00
60	7	24	1464	377389	0.0000D+00
80	7	23	1863	487687	0.0000D+00
100	7	23	2323	611621	0.0000D+00
120	7	23	2783	717157	0.0000D+00
140	8	26	3666	965954	0.0000D+00
160	8	26	4186	1100767	0.0000D+00
180	8	26	4706	1239104	0.0000D+00
200	8	26	5226	1383684	0.0000D+00
220	7	26	5746	1439806	0.0000D+00
240	7	27	6507	1608401	0.0000D+00
260	7	27	7047	1749070	0.0000D+00
280	8	27	7587	1963223	0.0000D+00
300	7	26	7826	1974089	0.0000D+00
320	7	27	8667	2170895	0.0000D+00
340	7	28	9548	2364546	0.0000D+00
360	7	27	9747	2475765	0.0000D+00
380	7	27	10287	2601817	0.0000D+00
400	7	26	10426	2647514	0.0000D+00
420	7	26	10946	2741319	0.0000D+00
440	7	26	11466	2871474	0.0000D+00
460	8	28	12908	3363502	0.0000D+00
480	8	28	13468	3500627	0.0000D+00
500	8	29	14529	3743186	0.0000D+00
TOTALS	192	665	174363	44546626	

Table 2-5-7

HE EXTENDED ROSENBROCK TEST FUNCTION
ITH ANALYTICAL GRADIENT
YBRID 1 METHOD

NV	NI	NF	NC	CPU	FV
2	30	64	192	72510	0.1051D-18
20	45	97	2037	518198	0.1871D-21
40	44	101	4141	994904	0.2525D-25
60	91	192	11712	2917817	0.1032D-20
80	84	183	14823	3605610	0.5006D-21
100	58	133	13433	3180819	0.3217D-19
120	123	264	31944	7873823	0.1192D-26
140	78	174	24534	5906680	0.4632D-20
160	163	342	55062	13585771	0.8277D-24
180	183	384	69504	17192884	0.2467D-27
200	196	412	82812	20390045	0.1681D-19
220	57	133	29393	6877582	0.9979D-21
240	244	506	121946	30034378	0.5660D-23
260	263	549	143289	35189148	0.9794D-32
280	83	190	53390	12588531	0.6877D-19
300	30	86	25886	5553894	0.5401D-22
320	62	149	47829	11051076	0.1109D-19
340	57	139	47399	10800592	0.1169D-19
360	206	437	157757	38467713	0.4986D-19
380	122	270	102870	24722491	0.5637D-19
400	28	83	33283	7082497	0.2485D-18
420	423	871	366691	90393424	0.8324D-27
440	54	136	59976	13480396	0.1725D-20
460	327	681	313941	77049730	0.1317D-19
480	41	111	53391	11682239	0.8799D-22
500	127	278	139278	33497729	0.4766D-19
TOTALS	3219	6965	2006513	484710481	

Table 2.6.1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 1 METHOD

NV	NI	NF	NC	CPU	FV
4	40	87	435	139577	0.1147D-21
20	55	119	2499	671906	0.1923D-22
40	64	138	5658	1486337	0.1763D-21
60	72	157	9577	2484006	0.1247D-22
80	85	185	14985	3894744	0.1828D-27
100	106	227	22927	5972613	0.2107D-21
120	132	279	33759	8789796	0.1161D-21
140	101	218	30738	7953107	0.7983D-20
160	165	348	56028	14455414	0.4116D-26
180	94	208	37648	9700034	0.1084D-20
200	150	319	64119	16575091	0.1105D-20
220	187	394	87074	22841802	0.3761D-19
240	203	426	102666	26862207	0.2454D-20
260	238	498	129978	33758155	0.5180D-20
280	102	229	64349	16377340	0.1823D-19
300	223	472	142072	37107942	0.4674D-20
320	100	224	71904	18391543	0.1168D-19
340	152	326	111166	28826033	0.8290D-21
360	137	295	106495	27493400	0.2485D-20
380	159	340	129540	33524587	0.9440D-20
400	187	397	159197	41346778	0.1292D-19
420	177	378	159138	41070165	0.1282D-19
440	162	351	154791	39934013	0.7612D-20
460	112	251	115711	29515943	0.8802D-20
480	59	145	69745	17165902	0.1400D-20
500	303	631	316131	82766251	0.4857D-20
TOTALS	3565	7642	2198330	569112686	

Table 2-6-2

THE EXTENDED MIELE & CANTRELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 1 METHOD

NV	NI	NF	NC	CPU	FV
4	33	81	405	172242	0.7068D-12
20	81	220	4620	1836094	0.2537D-11
40	90	243	9963	3887147	0.5237D-11
60	86	222	13542	5403028	0.5805D-11
80	73	184	14904	5980691	0.7380D-11
100	69	168	16968	6885688	0.5053D-11
120	73	199	24079	9473440	0.4350D-11
140	91	246	34686	13656027	0.2782D-12
160	86	227	36547	14507038	0.7522D-11
180	72	173	31313	12680937	0.3137D-11
200	88	218	43818	17617632	0.1630D-11
220	56	141	31161	12442221	0.6187D-11
240	47	122	29402	11748568	0.4813D-12
260	106	281	73341	29304926	0.4825D-11
280	64	171	48051	19125166	0.1289D-10
300	99	235	70735	28853107	0.1252D-11
320	91	228	73188	29409480	0.5050D-11
340	84	222	75702	30182771	0.7532D-11
360	74	185	66785	26809383	0.1851D-10
380	86	229	87249	34760359	0.1285D-10
400	111	292	117092	46849974	0.1469D-10
420	81	206	86726	34839873	0.3299D-11
440	78	204	89964	35989195	0.1098D-10
460	102	278	128158	50986134	0.7986D-11
480	107	287	138047	54821119	0.1286D-10
500	103	279	139779	55692174	0.4843D-11
TOTALS	2131	5541	1486225	593914414	

Table 2.6-3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 1 METHOD

NV	NI	NF	NC	CPU	FV
4	77	166	830	249118	0.4808D-15
20	89	191	4011	1029852	0.4958D-13
40	155	322	13202	3334360	0.2476D-13
60	188	391	23851	5959366	0.4537D-13
80	248	512	41472	10292912	0.2298D-14
100	207	432	43632	10838118	0.2008D-13
120	297	615	74415	18495839	0.2762D-13
140	302	623	87843	21739281	0.7740D-14
160	382	788	126868	31559775	0.2346D-13
180	396	815	147515	36589878	0.3483D-13
200	435	894	179694	44048504	0.2416D-13
220	444	912	201552	49862079	0.1192D-12
240	487	996	240036	59031557	0.2450D-13
260	291	606	158166	38871735	0.2578D-13
280	564	1150	323150	79803673	0.1705D-12
300	607	1237	372337	92446767	0.3427D-13
320	647	1314	421794	104945049	0.4089D-14
340	684	1393	475013	118508759	0.9591D-13
360	576	1179	425619	106026963	0.3081D-13
380	764	1554	592074	146822506	0.3146D-13
400	737	1500	601500	148871111	0.2936D-13
420	593	1212	510252	127062924	0.2666D-13
440	676	1378	607698	150758731	0.3999D-13
460	603	1233	568413	140911896	0.4914D-13
480	609	1246	599326	147817363	0.3943D-13
500	804	1637	820137	203673999	0.2614D-13
TOTALS	11862	24296	7660400	1899552115	

Table 2-6-4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 1 METHOD

NV	NI	NF	NC	CPU	FV
10	81	191	2101	863459	0.4218D-20
20	37	81	1701	701132	0.3838D-20
40	65	140	5740	2340112	0.1726D-19
60	53	119	7259	2911082	0.2242D-19
80	45	102	8262	3290215	0.1858D-20
100	67	147	14847	5971108	0.2035D-19
120	100	214	25894	10494046	0.7687D-20
140	101	216	30456	12339479	0.3110D-20
160	72	164	26404	10537558	0.6736D-20
180	57	133	24073	9531910	0.1656D-19
200	60	140	28140	11182473	0.9602D-20
220	49	119	26299	10266791	0.3039D-20
240	53	127	30607	11957214	0.1353D-20
260	66	153	39933	15778876	0.2271D-19
280	40	100	28100	10973292	0.3491D-20
300	51	122	36722	14504600	0.2264D-19
320	45	112	35952	14076207	0.1101D-19
340	49	121	41261	16183930	0.6519D-20
360	50	124	44764	17516896	0.8395D-20
380	54	131	49911	19638039	0.1748D-20
400	65	156	62556	24569211	0.3211D-20
420	71	169	71149	28131164	0.3724D-20
440	85	200	88200	34994624	0.9397D-20
460	95	220	101420	40255996	0.9265D-20
480	97	222	106782	42497968	0.1923D-19
500	95	218	109210	43536648	0.3279D-20
TOTALS	1703	3941	1047751	415044030	

Table 2-6-5

THE EXTENDED BEALE TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 1 METHOD

NV	NI	NF	NC	CPU	FV
2	14	33	99	78137	0.1176D-27
20	30	68	1428	462149	0.3059D-19
40	46	101	4141	1323416	0.6343D-26
60	64	137	8357	2659520	0.7195D-20
80	83	178	14418	4619247	0.6457D-22
100	103	219	22119	7025554	0.1752D-30
120	123	264	31944	10104102	0.6018D-33
140	142	299	42159	13427927	0.9291D-19
160	132	276	44436	14337669	0.1457D-19
180	141	295	53395	17240144	0.2500D-19
200	146	307	61707	19906044	0.1772D-19
220	147	308	68068	21909384	0.2236D-19
240	148	310	74710	24081311	0.2198D-19
260	148	313	81693	26172689	0.2489D-19
280	148	313	87953	28095582	0.2599D-19
300	148	313	94213	29926336	0.2731D-19
320	148	313	100473	31906322	0.2880D-19
340	113	243	82863	26498119	0.3034D-19
360	121	261	94221	30068819	0.6332D-19
380	132	283	107823	34601599	0.1485D-19
400	139	297	119097	38274500	0.1094D-19
420	141	304	127984	40698485	0.2595D-19
440	141	304	134064	42897127	0.3593D-19
460	145	311	143371	45789241	0.1774D-19
480	146	313	150553	47939689	0.2156D-19
500	147	315	157815	50112677	0.2559D-19
TOTALS	3136	6678	1909104	610155789	

Table 2-6-6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 1 METHOD

NV	NI	NF	NC	CPU	FV
2	8	18	54	64207	0.0000D+00
20	7	22	462	128464	0.0000D+00
40	8	24	984	256311	0.0000D+00
60	8	26	1586	407312	0.0000D+00
80	7	23	1863	472921	0.0000D+00
100	8	27	2727	686427	0.0000D+00
120	8	28	3388	855503	0.0000D+00
140	8	28	3948	998509	0.0000D+00
160	8	27	4347	1094753	0.0000D+00
180	8	27	4887	1237248	0.0000D+00
200	8	28	5628	1428541	0.0000D+00
220	8	30	6630	1622587	0.0000D+00
240	8	29	6989	1732597	0.0000D+00
260	8	30	7830	1875766	0.0000D+00
280	12	36	10116	2672976	0.0000D+00
300	10	31	9331	2430525	0.0000D+00
320	8	28	8988	2246694	0.0000D+00
340	8	29	9889	2502644	0.0000D+00
360	8	29	10469	2524322	0.0000D+00
380	8	30	11430	2733631	0.0000D+00
400	8	30	12030	2877386	0.0000D+00
420	8	30	12630	3019177	0.0000D+00
440	8	29	12789	3140752	0.0000D+00
460	8	30	13830	3360586	0.0000D+00
480	8	31	14911	3615613	0.0000D+00
500	8	31	15531	3782921	0.0000D+00
TOTALS	212	731	193267	47768373	

Table 2_6_7

THE EXTENDED ROSENBROCK TEST FUNCTION
WITH ANALYTICAL GRADIENT
SHANNO's HYBRID METHOD

NV	NI	NF	NC	CPU	FV
2	35	74	222	128528	0.2581D-26
20	21	48	1008	281533	0.7357D-22
40	22	51	2091	547499	0.1114D-23
60	24	58	3538	910800	0.2924D-30
80	26	66	5346	1318920	0.1146D-24
100	23	62	6262	1524024	0.8131D-30
120	24	67	8107	1938934	0.3267D-30
140	26	74	10434	2444380	0.1922D-32
160	25	70	11270	2677949	0.1843D-33
180	23	66	11946	2803627	0.3251D-25
200	25	70	14070	3335788	0.2159D-24
220	24	71	15691	3645168	0.5136D-23
240	24	72	17352	4004321	0.8618D-29
260	23	71	18531	4209736	0.3743D-20
280	23	72	20232	4547788	0.5332D-33
300	23	74	22274	4909233	0.1610D-28
320	23	75	24075	5340702	0.9901D-29
340	24	78	26598	5931668	0.2547D-27
360	23	76	27436	6092943	0.1197D-20
380	22	73	27813	6074179	0.8444D-29
400	21	69	27669	6114265	0.3852D-27
420	18	64	26944	5774844	0.2352D-26
440	24	81	35721	7799523	0.6040D-24
460	24	81	37341	8092485	0.7843D-26
480	24	80	38480	8500638	0.4522D-28
500	24	80	40080	8838250	0.2601D-29
TOTALS	618	1823	480531	107787733	

Table 2.7.1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
SHANNO's HYBRID METHOD

NV	NI	NF	NC	CPU	FV
4	37	79	395	176136	0.1005D-21
20	46	102	2142	621730	0.2370D-23
40	141	301	12341	3617147	0.3663D-19
60	71	154	9394	2712282	0.2844D-22
80	88	189	15309	4417917	0.7373D-25
100	105	227	22927	6579701	0.3208D-23
120	134	288	34848	10004767	0.5091D-23
140	154	341	48081	13573455	0.2473D-24
160	178	395	63595	17953416	0.3520D-22
180	163	345	62445	18105290	0.2020D-21
200	204	426	85626	24905860	0.1156D-18
220	181	382	84422	24513783	0.4978D-20
240	200	421	101461	29507715	0.2764D-21
260	249	518	135198	39678704	0.5321D-21
280	240	501	140781	41271052	0.3637D-21
300	175	369	111069	32455995	0.3841D-20
320	181	382	122622	35789965	0.3046D-21
340	176	372	126852	36975902	0.6000D-21
360	157	336	121296	35135006	0.4770D-21
380	154	331	126111	36403687	0.2896D-21
400	174	372	149172	43208383	0.4825D-21
420	201	426	179346	51945393	0.3601D-21
440	245	514	226674	65929554	0.3196D-20
460	128	284	130924	37279355	0.6323D-21
480	400	834	401154	117866614	0.9016D-20
500	478	1013	507513	148122739	0.1930D-20
TOTALS	4660	9902	3021698	878751548	

Table 2.7.2

THE EXTENDED MIELE & CANTRELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
SHANNO's HYBRID METHOD

NV	NI	NF	NC	CPU	FV
4	85	195	975	487728	0.2007D-15
20	59	153	3213	1349554	0.3407D-11
40	66	161	6601	2847880	0.1164D-11
60	59	149	9089	3836955	0.5523D-11
80	73	183	14823	6284560	0.6644D-11
100	71	170	17170	7380957	0.1211D-10
120	75	201	24321	10074838	0.1165D-11
140	74	189	26649	11139783	0.1573D-11
160	89	236	37996	15779740	0.9151D-12
180	73	186	33666	14106027	0.1097D-10
200	66	178	35778	14763891	0.1370D-10
220	72	215	47515	19130105	0.9540D-11
240	48	122	29402	12349141	0.2371D-11
260	86	222	57942	24379460	0.1105D-10
280	76	214	60134	24766662	0.4239D-11
300	69	182	54782	22991776	0.1308D-10
320	92	257	82497	34176456	0.6783D-11
340	75	207	70587	29204509	0.2150D-10
360	70	200	72200	29689795	0.9824D-11
380	78	204	77724	32611194	0.4158D-11
400	74	201	80601	33407661	0.2439D-10
420	71	191	80411	33387529	0.1292D-10
440	63	178	78498	32067347	0.3962D-11
460	31	91	41951	17005925	0.1039D-11
480	98	269	129389	53646035	0.2631D-10
500	50	139	69639	28823710	0.6972D-11
TOTALS	1843	4893	1243553	515689218	

Table 2-7-3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
SHANNO's HYBRID METHOD

NV	NI	NF	NC	CPU	FV
4	95	200	1000	374217	0.3627D-13
20	141	290	6090	1766746	0.5452D-14
40	102	220	9020	2484454	0.6485D-13
60	123	263	16043	4416475	0.1121D-13
80	130	279	22599	6192526	0.9571D-13
100	156	336	33936	9277909	0.1028D-13
120	244	512	61952	17074976	0.7752D-13
140	296	614	86574	23997413	0.1954D-14
160	208	440	70840	19341356	0.3641D-13
180	187	399	72219	19789281	0.3981D-13
200	239	504	101304	27806419	0.1762D-13
220	265	554	122434	33851980	0.2224D-13
240	246	517	124597	34109164	0.1693D-14
260	277	578	150858	41219424	0.1491D-12
280	293	609	171129	47067741	0.1950D-14
300	328	682	205282	56547493	0.8112D-13
320	350	725	232725	63909593	0.5587D-14
340	355	737	251317	68700962	0.6573D-14
360	387	806	290966	79733849	0.1369D-13
380	421	873	332613	91617736	0.8738D-14
400	418	870	348870	95460250	0.1013D-18
420	628	1287	541827	149755222	0.1488D-13
440	472	977	430857	118756795	0.9847D-14
460	492	1015	467915	128986440	0.6706D-13
480	507	1048	504088	138692654	0.3415D-13
500	545	1123	562623	154763267	0.2696D-13
TOTALS	7905	16458	5219678	1435694342	

Table 8-7-4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
SHANNO's HYBRID METHOD

NV	NI	NF	NC	CPU	FV
10	98	223	2453	1081239	0.2184D-24
20	37	81	1701	719359	0.2808D-20
40	81	173	7093	3023527	0.1181D-19
60	52	116	7076	2961358	0.2551D-20
80	52	115	9315	3901329	0.8872D-20
100	66	147	14847	6224423	0.2755D-20
120	109	233	28193	11932872	0.2328D-19
140	114	242	34122	14461348	0.1568D-19
160	76	172	27692	11531598	0.1427D-19
180	57	135	24435	10000903	0.1025D-20
200	65	151	30351	12468325	0.2734D-19
220	46	112	24752	10037084	0.2841D-19
240	50	121	29161	11821107	0.1114D-19
260	58	139	36279	14688758	0.5664D-20
280	44	106	29786	12135769	0.8969D-20
300	43	106	31906	12833545	0.8928D-20
320	53	127	40767	16575844	0.6442D-20
340	49	119	40579	16399219	0.2767D-19
360	46	115	41515	16702858	0.2237D-20
380	54	131	49911	20064906	0.1229D-19
400	66	157	62957	25606559	0.1342D-19
420	68	164	69044	28097360	0.1603D-19
440	99	225	99225	41105199	0.1077D-19
460	87	203	93583	38271506	0.2425D-19
480	106	240	115440	47917805	0.3690D-19
500	100	229	114729	47045869	0.1856D-19
TOTALS	1776	4082	1066912	437609669	

Table 2-7-5

THE EXTENDED BEALE TEST FUNCTION
WITH ANALYTICAL GRADIENT
SHANNO's HYBRID METHOD

NV	NI	NF	NC	CPU	FV
2	14	32	96	79225	0.2390D-21
20	10	29	609	202719	0.9438D-22
40	11	36	1476	469947	0.1932D-30
60	11	35	2135	678910	0.1100D-31
80	10	33	2673	848195	0.3775D-21
100	10	35	3535	1099161	0.2568D-30
120	10	36	4356	1344059	0.8121D-22
140	10	37	5217	1600158	0.4371D-21
160	11	42	6762	2061843	0.2092D-30
180	11	38	6878	2139886	0.1502D-29
200	11	38	7638	2367065	0.1321D-30
220	11	39	8619	2649656	0.4479D-31
240	11	40	9640	2992638	0.4450D-31
260	11	40	10440	3187307	0.3485D-31
280	11	41	11521	3508623	0.4410D-31
300	11	42	12642	3895049	0.8458D-31
320	11	41	13161	4069847	0.1752D-30
340	11	40	13640	4230680	0.1230D-31
360	10	39	14079	4292682	0.3194D-23
380	10	38	14478	4427785	0.3776D-23
400	10	39	15639	4769022	0.1039D-29
420	9	40	16840	5040511	0.2531D-32
440	9	41	18081	5236730	0.1692D-20
460	10	41	18901	5672524	0.4680D-24
480	10	41	19721	5923085	0.2827D-21
500	10	42	21042	6287193	0.1714D-20
TOTALS	274	995	259819	79074500	

Table 2-7-6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
SHANNO's HYBRID METHOD

NV	NI	NF	NC	CPU	FV
2	8	18	54	69191	0.0000D+00
20	7	23	483	140388	0.0000D+00
40	8	23	943	284273	0.0000D+00
60	7	24	1464	405476	0.0000D+00
80	7	25	2025	547427	0.0000D+00
100	7	23	2323	659877	0.0000D+00
120	7	23	2783	775748	0.0000D+00
140	8	26	3666	1045062	0.0000D+00
160	8	26	4186	1188536	0.0000D+00
180	8	26	4706	1346952	0.0000D+00
200	8	26	5226	1491971	0.0000D+00
220	7	26	5746	1545031	0.0000D+00
240	7	27	6507	1712227	0.0000D+00
260	8	29	7569	2089953	0.0000D+00
280	7	25	7025	1977751	0.0000D+00
300	7	27	8127	2171120	0.0000D+00
320	7	27	8667	2327398	0.0000D+00
340	7	27	9207	2504093	0.0000D+00
360	7	27	9747	2653598	0.0000D+00
380	7	27	10287	2782198	0.0000D+00
400	7	26	10426	2828695	0.0000D+00
420	7	27	11367	3040253	0.0000D+00
440	7	26	11466	3087606	0.0000D+00
460	8	28	12908	3570593	0.0000D+00
480	8	28	13468	3760753	0.0000D+00
500	8	29	14529	4021978	0.0000D+00
TOTALS	192	669	174905	48028148	

Table 2-7-7

THE EXTENDED ROSENBROCK TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.1)

NV	NI	NF	NC	CPU	FV
2	32	74	222	88650	0.4892D-28
20	22	51	1071	306121	0.1266D-22
40	27	62	2542	722443	0.4475D-26
60	18	50	3050	792134	0.1147D-23
80	19	54	4374	1130993	0.1023D-18
100	21	58	5858	1502967	0.6798D-30
120	21	61	7381	1839612	0.6953D-21
140	20	60	8460	2040118	0.4595D-30
160	21	63	10143	2451507	0.4683D-23
180	21	61	11041	2661418	0.2053D-30
200	27	75	15075	3949333	0.8380D-32
220	20	61	13481	3206137	0.2121D-33
240	21	70	16870	3891877	0.9253D-33
260	18	62	16182	3827687	0.2387D-26
280	18	60	16860	3968809	0.5259D-22
300	18	62	18662	4268311	0.4986D-22
320	18	60	19260	4647299	0.7244D-24
340	22	70	23870	5781830	0.3164D-28
360	18	61	22021	5011564	0.5298D-32
380	20	65	24765	5937948	0.1946D-20
400	18	62	24862	5747659	0.9135D-29
420	25	80	33680	8031042	0.4798D-25
440	24	77	33957	8525366	0.1322D-29
460	21	72	33192	7531182	0.9509D-25
480	21	73	35113	8094387	0.5722D-20
500	21	72	36072	8376687	0.4783D-21
TOTALS	552	1676	438064	104333081	

Table 2-8-1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.1)

NV	NI	NF	NC	CPU	FV
4	37	79	395	148116	0.1005D-21
20	46	99	2079	650025	0.2415D-23
40	59	129	5289	1652265	0.6388D-21
60	71	154	9394	2896626	0.2844D-22
80	88	189	15309	4786915	0.7373D-25
100	105	227	22927	7019196	0.3163D-23
120	125	266	32186	10016191	0.1582D-24
140	147	309	43569	13592956	0.1332D-20
160	165	347	55867	17409403	0.7173D-27
180	163	345	62445	19320138	0.2020D-21
200	204	426	85626	26986001	0.1156D-18
220	181	382	84422	26236789	0.4978D-20
240	200	421	101461	31976813	0.2764D-21
260	249	518	135198	42370059	0.5321D-21
280	240	501	140781	43876732	0.3637D-21
300	175	369	111069	34371467	0.3841D-20
320	181	382	122622	38077072	0.3046D-21
340	176	372	126852	39649984	0.6000D-21
360	157	336	121296	37596556	0.4770D-21
380	154	331	126111	38926167	0.2896D-21
400	174	372	149172	46067206	0.4825D-21
420	201	426	179346	55981205	0.3601D-21
440	245	514	226674	70680497	0.3196D-20
460	128	284	130924	39839289	0.6323D-21
480	155	337	162097	49233746	0.1122D-19
500	236	499	249999	77263368	0.2424D-20
TOTALS	4062	8614	2503110	776624782	

Table 2-8-2

THE EXTENDED MIELE & CANTRELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.1)

NV	NI	NF	NC	CPU	FV
4	108	305	1525	680901	0.2625D-11
20	59	162	3402	1479966	0.8818D-13
40	58	152	6232	2715709	0.1199D-11
60	80	219	13359	5775341	0.1847D-11
80	76	206	16686	7378685	0.7089D-11
100	62	148	14948	6852477	0.2210D-11
120	51	130	15730	7201223	0.6029D-11
140	74	212	29892	12929717	0.9236D-11
160	69	192	30912	13174535	0.3017D-11
180	66	171	30951	13738382	0.7705D-12
200	63	169	33969	14812650	0.2196D-11
220	59	148	32708	14647763	0.7500D-11
240	54	152	36632	15939581	0.2134D-11
260	71	168	43848	19970740	0.2508D-11
280	56	143	40183	17457950	0.7729D-11
300	22	67	20167	8390570	0.4242D-11
320	65	177	56817	24389194	0.1454D-10
340	54	139	47399	21247073	0.1130D-10
360	80	194	70034	31714267	0.1190D-10
380	91	225	85725	38175493	0.9170D-11
400	72	190	76190	33874863	0.4197D-11
420	63	165	69465	30683594	0.1178D-11
440	67	180	79380	35070255	0.3046D-11
460	62	181	83441	35805148	0.7217D-11
480	70	200	96200	41488462	0.1612D-12
500	54	143	71643	32477715	0.8703D-11
TOTALS	1706	4538	1107438	488072262	

Table 2-8-3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.1)

NV	NI	NF	NC	CPU	FV
4	66	142	710	256353	0.8277D-15
20	105	221	4641	1522909	0.1218D-13
40	166	343	14063	4532981	0.5952D-13
60	185	387	23607	7365779	0.6109D-14
80	193	404	32724	10323358	0.2852D-13
100	245	512	51712	16154069	0.1762D-13
120	273	571	69091	21348486	0.3167D-13
140	287	595	83895	26050459	0.3227D-13
160	429	878	141358	44517723	0.3378D-13
180	233	491	88871	27505644	0.1796D-13
200	380	784	157584	48615966	0.1636D-13
220	588	1202	265642	82350276	0.1642D-13
240	487	997	240277	73613235	0.7062D-14
260	527	1080	281880	85990995	0.1174D-14
280	358	739	207659	63642041	0.1779D-13
300	604	1233	371133	114201722	0.2336D-13
320	544	1110	356310	110868529	0.2581D-13
340	687	1405	479105	146400903	0.4083D-16
360	736	1503	542583	166248219	0.6068D-14
380	418	866	329946	101430365	0.8465D-13
400	444	917	367717	113192530	0.1867D-13
420	628	1287	541827	165338036	0.1488D-13
440	887	1805	796005	243743279	0.1524D-14
460	927	1886	869446	266673494	0.7924D-16
480	545	1121	539201	166294820	0.2370D-13
500	546	1124	563124	173737471	0.3996D-13
TOTALS	11488	23603	7420111	2281919642	

Table 2.8.4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.1)

NV	NI	NF	NC	CPU	FV
10	108	238	2618	1245408	0.2167D-20
20	37	81	1701	752273	0.2808D-20
40	86	182	7462	3371198	0.3776D-20
60	52	116	7076	3228970	0.2551D-20
80	52	115	9315	4173164	0.8872D-20
100	67	150	15150	6571865	0.5010D-20
120	103	221	26741	12057413	0.1165D-19
140	106	227	32007	14147014	0.5678D-20
160	76	172	27692	11737909	0.1427D-19
180	57	135	24435	10675147	0.1025D-20
200	60	141	28341	12073094	0.8264D-20
220	46	112	24752	10677182	0.2841D-19
240	50	121	29161	12775573	0.1114D-19
260	58	139	36279	15569244	0.5664D-20
280	44	106	29786	12830515	0.8969D-20
300	43	106	31906	13812132	0.8928D-20
320	53	127	40767	17662227	0.6442D-20
340	49	119	40579	17211036	0.2767D-19
360	46	115	41515	17453280	0.2237D-20
380	54	131	49911	21920542	0.1229D-19
400	66	157	62957	26406901	0.1342D-19
420	75	177	74517	32157653	0.2296D-20
440	90	207	91287	40402305	0.1954D-19
460	87	203	93583	39993707	0.2425D-19
480	106	240	115440	50585489	0.3690D-19
500	100	229	114729	50556639	0.1856D-19
TOTALS	1771	4067	1059707	460047880	

Table 2-8-5

THE EXTENDED BEALE TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.1)

NV	NI	NF	NC	CPU	FV
2	14	32	96	81145	0.2390D-21
20	10	29	609	203959	0.9438D-22
40	11	36	1476	476050	0.1932D-30
60	11	35	2135	687694	0.1100D-31
80	10	33	2673	855986	0.3775D-21
100	10	35	3535	1103137	0.2568D-30
120	10	36	4356	1343037	0.8121D-22
140	10	37	5217	1606977	0.4371D-21
160	11	42	6762	2079109	0.2092D-30
180	11	38	6878	2162652	0.1502D-29
200	11	38	7638	2394835	0.1321D-30
220	11	39	8619	2682475	0.4479D-31
240	11	40	9640	2983784	0.4450D-31
260	11	40	10440	3255734	0.3485D-31
280	11	41	11521	3551481	0.4410D-31
300	11	42	12642	3855417	0.8458D-31
320	11	41	13161	4041586	0.1752D-30
340	11	40	13640	4224614	0.1230D-31
360	10	39	14079	4235510	0.3194D-23
380	10	38	14478	4377096	0.3776D-23
400	10	39	15639	4768267	0.1039D-29
420	9	40	16840	5023229	0.2531D-32
440	9	41	18081	5280535	0.1692D-20
460	10	41	18901	5699680	0.4680D-24
480	10	41	19721	5843660	0.2827D-21
500	10	42	21042	6293633	0.1714D-20
TOTALS	274	995	259819	79111282	

Table 2-8-6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.1)

NV	NI	NF	NC	CPU	FV
2	8	18	54	66593	0.0000D+00
20	7	23	483	139357	0.0000D+00
40	8	23	943	288621	0.0000D+00
60	7	24	1464	404453	0.0000D+00
80	7	25	2025	555812	0.0000D+00
100	7	23	2323	654100	0.0000D+00
120	7	23	2783	779824	0.0000D+00
140	8	26	3666	1052907	0.0000D+00
160	8	26	4186	1198582	0.0000D+00
180	8	26	4706	1349361	0.0000D+00
200	8	26	5226	1497848	0.0000D+00
220	7	26	5746	1527229	0.0000D+00
240	7	27	6507	1709084	0.0000D+00
260	8	29	7569	2094297	0.0000D+00
280	7	25	7025	2000096	0.0000D+00
300	7	27	8127	2173018	0.0000D+00
320	7	27	8667	2298135	0.0000D+00
340	7	27	9207	2503721	0.0000D+00
360	7	27	9747	2650380	0.0000D+00
380	7	27	10287	2785667	0.0000D+00
400	7	26	10426	2841764	0.0000D+00
420	7	27	11367	3026547	0.0000D+00
440	7	26	11466	3083785	0.0000D+00
460	8	28	12908	3609153	0.0000D+00
480	8	28	13468	3743800	0.0000D+00
500	8	29	14529	4002980	0.0000D+00
TOTALS	192	669	174905	48037114	

Table 2.8.7

THE EXTENDED ROSENBROCK TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.01)

NV	NI	NF	NC	CPU	FV
2	32	74	222	94222	0.4892D-28
20	20	48	1008	283148	0.2682D-29
40	25	57	2337	672084	0.6869D-22
60	20	54	3294	851623	0.3490D-28
80	19	54	4374	1053702	0.1436D-22
100	21	59	5959	1502965	0.3138D-27
120	23	63	7623	1951679	0.1613D-23
140	22	62	8742	2158763	0.1402D-29
160	20	60	9660	2306882	0.1372D-26
180	20	59	10679	2606193	0.1829D-23
200	24	69	13869	3550375	0.1537D-22
220	21	63	13923	3323835	0.1293D-35
240	19	60	14460	3542426	0.1411D-33
260	19	60	15660	3634067	0.1594D-22
280	18	62	17422	4087513	0.7553D-25
300	18	63	18963	4378090	0.1000D-19
320	19	63	20223	4585964	0.3462D-22
340	19	64	21824	5086342	0.0000D+00
360	18	61	22021	5063636	0.5298D-32
380	18	61	23241	5448450	0.1456D-20
400	18	62	24862	5791829	0.9135D-29
420	21	70	29470	6867119	0.4981D-24
440	24	79	34839	8235996	0.3217D-26
460	23	76	35036	8457411	0.5888D-25
480	23	77	37037	8699512	0.1788D-22
500	22	75	37575	8699821	0.4232D-33
TOTALS	546	1655	434323	102934447	

Table 2_9_1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.01)

NV	NI	NF	NC	CPU	FV
4	37	79	395	191708	0.1005D-21
20	46	99	2079	646151	0.2415D-23
40	59	129	5289	1627857	0.6388D-21
60	71	154	9394	2872123	0.2844D-22
80	88	189	15309	4673927	0.7373D-25
100	105	227	22927	6968534	0.3163D-23
120	125	266	32186	9934994	0.1582D-24
140	147	309	43569	13532428	0.1332D-20
160	165	347	55867	17303549	0.7173D-27
180	163	345	62445	19221018	0.2020D-21
200	204	426	85626	26932974	0.1156D-18
220	181	382	84422	26242644	0.4978D-20
240	200	421	101461	31812046	0.2764D-21
260	249	518	135198	42239922	0.5321D-21
280	240	501	140781	43873058	0.3637D-21
300	175	369	111069	34318219	0.3841D-20
320	181	382	122622	37792528	0.3046D-21
340	176	372	126852	39378073	0.6000D-21
360	157	336	121296	37233042	0.4770D-21
380	154	331	126111	38404541	0.2896D-21
400	174	372	149172	45795171	0.4825D-21
420	201	426	179346	55491253	0.3601D-21
440	245	514	226674	70368538	0.3196D-20
460	128	284	130924	39830764	0.6323D-21
480	155	337	162097	49268339	0.1122D-19
500	139	305	152805	44782468	0.3036D-21
TOTALS	3965	8420	2405916	740735869	

Table 2-9-2

THE EXTENDED MIELE & CANTRELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.01)

NV	NI	NF	NC	CPU	FV
4	118	318	1590	725062	0.1547D-11
20	64	159	3339	1528540	0.5483D-12
40	83	206	8446	3822293	0.1115D-11
60	50	122	7442	3387260	0.3651D-11
80	61	176	14256	6163140	0.3625D-11
100	69	184	18584	8207524	0.8933D-11
120	48	125	15125	6832811	0.1183D-10
140	69	176	24816	11118219	0.7428D-11
160	57	155	24955	10915586	0.1241D-10
180	64	163	29503	13063370	0.5768D-11
200	66	175	35175	15186976	0.1484D-10
220	50	127	28067	12655362	0.3421D-11
240	88	224	53984	23774291	0.3050D-11
260	79	194	50634	22671903	0.1740D-11
280	67	173	48613	21533228	0.2085D-11
300	22	67	20167	8418951	0.4242D-11
320	65	177	56817	24396041	0.1454D-10
340	61	163	55583	24402848	0.2603D-11
360	68	187	67507	29286034	0.1829D-10
380	54	150	57150	24819709	0.1088D-10
400	58	147	58947	26262809	0.5259D-11
420	67	176	74096	32452232	0.3816D-11
440	67	190	83790	35862516	0.1083D-10
460	91	233	107413	46969899	0.2644D-12
480	87	234	112554	49259615	0.3012D-11
500	60	160	80160	35904467	0.7120D-11
TOTALS	1733	4561	1138713	499620686	

Table 2.9.3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.01)

NV	NI	NF	NC	CPU	FV
4	95	197	985	357603	0.2170D-13
20	89	191	4011	1263031	0.5427D-13
40	166	345	14145	4358023	0.3445D-13
60	185	388	23668	7275604	0.4628D-13
80	303	623	50463	16013262	0.2260D-13
100	325	670	67670	20980406	0.4181D-13
120	247	517	62557	19488139	0.3389D-13
140	284	614	86574	26467077	0.2846D-13
160	339	700	112700	34596896	0.1297D-14
180	364	751	135931	42488759	0.1995D-13
200	380	784	157584	48902223	0.1636D-13
220	454	935	206635	62922976	0.2928D-14
240	496	1017	245097	74307972	0.2668D-14
260	523	1069	279009	86385315	0.1403D-13
280	354	730	205130	63000369	0.2075D-13
300	306	636	191436	58257028	0.1033D-14
320	430	882	283122	87796434	0.3679D-13
340	736	1498	510818	156450198	0.3645D-13
360	455	940	339340	103961952	0.1352D-13
380	764	1558	593598	179745575	0.1682D-13
400	445	919	368519	112500740	0.1711D-13
420	628	1287	541827	164696602	0.1488D-13
440	605	1240	546840	167203958	0.1588D-13
460	792	1620	746820	226939391	0.3587D-13
480	493	1023	492063	149672802	0.1020D-13
500	522	1076	539076	165775628	0.4348D-13
TOTALS	10780	22210	6805618	2081807963	

Table 2-9-4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.01)

NV	NI	NF	NC	CPU	FV
10	108	238	2618	1181740	0.2167D-20
20	37	81	1701	766083	0.2808D-20
40	74	160	6560	2947528	0.3602D-20
60	52	116	7076	3215080	0.2551D-20
80	52	115	9315	4156477	0.8872D-20
100	67	150	15150	6516033	0.5010D-20
120	103	221	26741	11946605	0.1165D-19
140	106	227	32007	14081314	0.5678D-20
160	76	172	27692	11669999	0.1427D-19
180	57	135	24435	10691495	0.1025D-20
200	60	141	28341	12087003	0.8264D-20
220	46	112	24752	10640169	0.2841D-19
240	50	121	29161	12745446	0.1114D-19
260	58	139	36279	15547845	0.5664D-20
280	44	106	29786	12879566	0.8969D-20
300	43	106	31906	13860210	0.8928D-20
320	53	127	40767	17575387	0.6442D-20
340	49	119	40579	17202013	0.2767D-19
360	46	115	41515	17412854	0.2237D-20
380	54	131	49911	21954538	0.1229D-19
400	66	157	62957	26281180	0.1342D-19
420	75	177	74517	32003786	0.2296D-20
440	90	207	91287	40204933	0.1954D-19
460	87	203	93583	39738461	0.2425D-19
480	106	240	115440	50211294	0.3690D-19
500	100	229	114729	50503743	0.1056D-19
TOTALS	1759	4045	1058805	458020782	

Table 2-3-5

THE EXTENDED BEALE TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.01)

NV	NI	NF	NC	CPU	FV
2	14	32	96	81145	0.2390D-21
20	10	29	609	203959	0.9438D-22
40	11	36	1476	476050	0.1932D-30
60	11	35	2135	687694	0.1100D-31
80	10	33	2673	855986	0.3775D-21
100	10	35	3535	1103137	0.2568D-30
120	10	36	4356	1343037	0.8121D-22
140	10	37	5217	1606977	0.4371D-21
160	11	42	6762	2079109	0.2092D-30
180	11	38	6878	2162652	0.1502D-29
200	11	38	7638	2394835	0.1321D-30
220	11	39	8619	2682475	0.4479D-31
240	11	40	9640	2983784	0.4450D-31
260	11	40	10440	3255734	0.3485D-31
280	11	41	11521	3551481	0.4410D-31
300	11	42	12642	3855417	0.8458D-31
320	11	41	13161	4041586	0.1752D-30
340	11	40	13640	4224614	0.1230D-31
360	10	39	14079	4235510	0.3194D-23
380	10	38	14478	4377096	0.3776D-23
400	10	39	15639	4768267	0.1039D-29
420	9	40	16840	5023229	0.2531D-32
440	9	41	18081	5280535	0.1692D-20
460	10	41	18901	5699680	0.4680D-24
480	10	41	19721	5843660	0.2827D-21
500	10	42	21042	6293633	0.1714D-20
TOTALS	274	995	259819	79111282	

Table 2-9-6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.01)

NV	NI	NF	NC	CPU	FV
2	8	18	54	66593	0.0000D+00
20	7	23	483	139357	0.0000D+00
40	8	23	943	288621	0.0000D+00
60	7	24	1464	404453	0.0000D+00
80	7	25	2025	555812	0.0000D+00
100	7	23	2323	654100	0.0000D+00
120	7	23	2783	779824	0.0000D+00
140	8	26	3666	1052907	0.0000D+00
160	8	26	4186	1198582	0.0000D+00
180	8	26	4706	1349361	0.0000D+00
200	8	26	5226	1497848	0.0000D+00
220	7	26	5746	1527229	0.0000D+00
240	7	27	6507	1709084	0.0000D+00
260	8	29	7569	2094297	0.0000D+00
280	7	25	7025	2000096	0.0000D+00
300	7	27	8127	2173018	0.0000D+00
320	7	27	8667	2298135	0.0000D+00
340	7	27	9207	2503721	0.0000D+00
360	7	27	9747	2650380	0.0000D+00
380	7	27	10287	2785667	0.0000D+00
400	7	26	10426	2841764	0.0000D+00
420	7	27	11367	3026547	0.0000D+00
440	7	26	11466	3083785	0.0000D+00
460	8	28	12908	3609153	0.0000D+00
480	8	28	13468	3743800	0.0000D+00
500	8	29	14529	4002980	0.0000D+00
TOTALS	192	669	174905	48037114	

Table 2.9.7

THE EXTENDED ROSENBROCK TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.001)

NV	NI	NF	NC	CPU	FV
2	32	74	222	83166	0.7928D-28
20	21	50	1050	287549	0.1175D-34
40	25	58	2378	667464	0.1270D-24
60	18	53	3233	818272	0.2033D-29
80	19	54	4374	1053821	0.1436D-22
100	20	57	5757	1465722	0.3705D-32
120	22	63	7623	1973054	0.1820D-32
140	22	62	8742	2165545	0.1402D-29
160	20	60	9660	2300868	0.1372D-26
180	20	59	10679	2583105	0.1829D-23
200	22	63	12663	3220595	0.2103D-25
220	23	66	14586	3502616	0.3662D-30
240	18	61	14701	3465122	0.5253D-27
260	19	60	15660	3617114	0.1594D-22
280	18	63	17703	4122361	0.1289D-28
300	18	66	19866	4526665	0.8181D-33
320	19	63	20223	4577335	0.3462D-22
340	19	64	21824	5106619	0.0000D+00
360	18	61	22021	5012665	0.5298D-32
380	18	61	23241	5405042	0.1456D-20
400	18	62	24862	5765338	0.9135D-29
420	20	72	30312	6940398	0.4937D-32
440	23	78	34398	7905632	0.1882D-22
460	23	77	35497	8293380	0.4331D-31
480	22	77	37037	8683559	0.7396D-31
500	22	75	37575	8650950	0.4232D-33
TOTALS	539	1659	435887	102193957	

Table 2.10.1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.001)

NV	NI	NF	NC	CPU	FV
4	37	79	395	145849	0.1005D-21
20	46	99	2079	650139	0.2415D-23
40	59	129	5289	1621573	0.6388D-21
60	71	154	9394	2878580	0.2844D-22
80	88	189	15309	4723478	0.7373D-25
100	105	227	22927	7013935	0.3163D-23
120	125	266	32186	9969132	0.1582D-24
140	147	309	43569	13601336	0.1332D-20
160	165	347	55867	17318597	0.7173D-27
180	163	345	62445	19116708	0.2020D-21
200	204	426	85626	26675456	0.1156D-18
220	181	382	84422	26021210	0.4978D-20
240	200	421	101461	31595334	0.2764D-21
260	249	518	135198	41999415	0.5321D-21
280	240	501	140781	43623746	0.3637D-21
300	175	369	111069	34145997	0.3841D-20
320	181	382	122622	37788100	0.3046D-21
340	176	372	126852	39206504	0.6000D-21
360	157	336	121296	37153697	0.4770D-21
380	154	331	126111	38450130	0.2896D-21
400	174	372	149172	45668081	0.4825D-21
420	201	426	179346	55265682	0.3601D-21
440	245	514	226674	70021001	0.3196D-20
460	128	284	130924	39621045	0.6323D-21
480	155	337	162097	48934399	0.1122D-19
500	139	305	152805	44892175	0.3036D-21
TOTALS	3965	8420	2405916	738101299	

Table 2-10-2

HE EXTENDED MIELE & CANTRELL TEST FUNCTION
ITH ANALYTICAL GRADIENT
HE ANGLE TEST HYBRID METHOD (Tau = 0.001)

NV	NI	NF	NC	CPU	FV
4	75	186	930	432026	0.2507D-11
20	64	159	3339	1475879	0.5483D-12
40	52	128	5248	2378217	0.2456D-11
60	50	122	7442	3349954	0.3651D-11
80	51	123	9963	4572573	0.4945D-12
100	69	184	18584	8070963	0.8933D-11
120	48	125	15125	6760214	0.1183D-10
140	69	176	24816	10925714	0.7428D-11
160	57	155	24955	10853756	0.1241D-10
180	58	156	28236	12241458	0.7950D-12
200	66	175	35175	15146127	0.1484D-10
220	57	148	32708	14468900	0.3610D-11
240	80	197	47477	21166845	0.1138D-11
260	79	194	50634	22637632	0.1740D-11
280	67	173	48613	21535736	0.2085D-11
300	22	67	20167	8483689	0.4242D-11
320	65	177	56817	24501330	0.1454D-10
340	61	163	55583	24544730	0.2603D-11
360	60	155	55955	24967292	0.7695D-11
380	54	150	57150	24787601	0.1088D-10
400	58	147	58947	26003991	0.7649D-11
420	67	176	74096	32615054	0.1923D-11
440	50	139	61299	26665381	0.9041D-11
460	91	233	107413	46981646	0.2644D-12
480	54	139	66859	30013309	0.7492D-12
500	44	115	57615	25675648	0.7380D-11
TOTALS	1568	4062	1025146	451255665	

Table 2.10.3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.001)

NV	NI	NF	NC	CPU	FV
4	95	197	985	360555	0.2170D-13
20	89	195	4095	1272719	0.1517D-18
40	100	220	9020	2751672	0.1476D-13
60	154	324	19764	6149762	0.2427D-14
80	303	623	50463	16058281	0.2260D-13
100	305	636	64236	19795976	0.2246D-13
120	188	399	48279	15028317	0.3430D-13
140	302	627	88407	27288109	0.5072D-13
160	311	659	106099	31531163	0.2059D-13
180	367	757	137017	42345392	0.4208D-13
200	380	784	157584	48762156	0.1636D-13
220	454	935	206635	62858710	0.2928D-14
240	503	1031	248471	75673305	0.3186D-15
260	338	707	184527	56848361	0.7771D-15
280	354	730	205130	62992422	0.2075D-13
300	604	1233	371133	112500122	0.1217D-12
320	432	886	284406	87935017	0.3296D-13
340	736	1498	510818	155923212	0.3645D-13
360	418	864	311904	94997261	0.1144D-13
380	396	821	312801	94230519	0.2272D-13
400	445	919	368519	111394197	0.1711D-13
420	628	1287	541827	162981167	0.1488D-13
440	516	1063	468783	142206231	0.7712D-14
460	927	1891	871751	263732780	0.3085D-13
480	986	2003	963443	291252562	0.8690D-14
500	526	1084	543084	166041264	0.3918D-13
TOTALS	10857	22373	7079181	2152919232	

Table 2-10-4

'HE EXTENDED DIXON TEST FUNCTION
ITH ANALYTICAL GRADIENT
'HE ANGLE TEST HYBRID METHOD (Tau = 0.001)

NV	NI	NF	NC	CPU	FV
10	108	238	2618	1267334	0.2167D-20
20	37	81	1701	772018	0.2808D-20
40	74	160	6560	2969290	0.3602D-20
60	52	116	7076	3250154	0.2551D-20
80	52	115	9315	4188752	0.8872D-20
100	67	150	15150	6568007	0.5010D-20
120	103	221	26741	12118250	0.1165D-19
140	106	227	32007	14296062	0.5678D-20
160	76	172	27692	11874446	0.1427D-19
180	57	135	24435	10793549	0.1025D-20
200	60	141	28341	12209190	0.8264D-20
220	46	112	24752	10746158	0.2841D-19
240	50	121	29161	12904110	0.1114D-19
260	58	139	36279	15663615	0.5664D-20
280	44	106	29786	12971458	0.8969D-20
300	43	106	31906	13955150	0.8928D-20
320	53	127	40767	17756356	0.6442D-20
340	49	119	40579	17376766	0.2767D-19
360	46	115	41515	17519288	0.2237D-20
380	54	131	49911	21955562	0.1229D-19
400	66	157	62957	26515294	0.1342D-19
420	75	177	74517	32154775	0.2296D-20
440	90	207	91287	40268123	0.1954D-19
460	87	203	93583	39869415	0.2425D-19
480	106	240	115440	50421298	0.3690D-19
500	100	229	114729	50462361	0.1856D-19
TOTALS	1759	4045	1058805	460846781	

Table 2-10-5

THE EXTENDED BEALE TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.001)

NV	NI	NF	NC	CPU	FV
2	14	32	96	81145	0.2390D-21
20	10	29	609	203959	0.9438D-22
40	11	36	1476	476050	0.1932D-30
60	11	35	2135	687694	0.1100D-31
80	10	33	2673	855986	0.3775D-21
100	10	35	3535	1103137	0.2568D-30
120	10	36	4356	1343037	0.8121D-22
140	10	37	5217	1606977	0.4371D-21
160	11	42	6762	2079109	0.2092D-30
180	11	38	6878	2162652	0.1502D-29
200	11	38	7638	2394835	0.1321D-30
220	11	39	8619	2682475	0.4479D-31
240	11	40	9640	2983784	0.4450D-31
260	11	40	10440	3255734	0.3485D-31
280	11	41	11521	3551481	0.4410D-31
300	11	42	12642	3855417	0.8458D-31
320	11	41	13161	4041586	0.1752D-30
340	11	40	13640	4224614	0.1230D-31
360	10	39	14079	4235510	0.3194D-23
380	10	38	14478	4377096	0.3776D-23
400	10	39	15639	4768267	0.1039D-29
420	9	40	16840	5023229	0.2531D-32
440	9	41	18081	5280535	0.1692D-20
460	10	41	18901	5699680	0.4680D-24
480	10	41	19721	5843660	0.2827D-21
500	10	42	21042	6293633	0.1714D-20
TOTALS	274	995	259819	79111282	

Table 2.10.6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE ANGLE TEST HYBRID METHOD (Tau = 0.001)

NV	NI	NF	NC	CPU	FV
2	8	18	54	66593	0.0000D+00
20	7	23	483	139357	0.0000D+00
40	8	23	943	288621	0.0000D+00
60	7	24	1464	404453	0.0000D+00
80	7	25	2025	555812	0.0000D+00
100	7	23	2323	654100	0.0000D+00
120	7	23	2783	779824	0.0000D+00
140	8	26	3666	1052907	0.0000D+00
160	8	26	4186	1198582	0.0000D+00
180	8	26	4706	1349361	0.0000D+00
200	8	26	5226	1497848	0.0000D+00
220	7	26	5746	1527229	0.0000D+00
240	7	27	6507	1709084	0.0000D+00
260	8	29	7569	2094297	0.0000D+00
280	7	25	7025	2000096	0.0000D+00
300	7	27	8127	2173018	0.0000D+00
320	7	27	8667	2298135	0.0000D+00
340	7	27	9207	2503721	0.0000D+00
360	7	27	9747	2650380	0.0000D+00
380	7	27	10287	2785667	0.0000D+00
400	7	26	10426	2841764	0.0000D+00
420	7	27	11367	3026547	0.0000D+00
440	7	26	11466	3083785	0.0000D+00
460	8	28	12908	3609153	0.0000D+00
480	8	28	13468	3743800	0.0000D+00
500	8	29	14529	4002980	0.0000D+00
TOTALS	192	669	174905	48037114	

Table 2-10-7

THE EXTENDED ROSENBROCK TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE BETA TEST HYBRID METHOD

NV	NI	NF	NC	CPU	FV
2	32	71	213	81936	0.1785D-27
20	21	50	1050	259954	0.1175D-34
40	25	58	2370	583030	0.1270D-24
60	19	53	3233	720665	0.0000D+00
80	19	54	4374	964474	0.1436D-22
100	20	57	5757	1276781	0.3486D-20
120	22	60	7260	1626386	0.2186D-21
140	22	62	8742	1938727	0.1402D-29
160	20	60	9660	2081550	0.1372D-26
180	20	59	10679	2327394	0.1829D-23
200	22	63	12663	2795614	0.2128D-19
220	23	66	14586	3215604	0.3662D-30
240	19	63	15183	3142388	0.1411D-33
260	19	60	15660	3307558	0.1594D-22
280	19	63	17703	3678437	0.5931D-32
300	18	65	19565	3950641	0.4187D-20
320	19	63	20223	4215056	0.3462D-22
340	19	64	21824	4546482	0.1582D-19
360	18	61	22021	4544668	0.5298D-32
380	18	61	23241	4778922	0.1456D-20
400	18	61	24461	4995267	0.1065D-22
420	20	72	30312	6106855	0.4937D-32
440	23	78	34398	7103882	0.1882D-22
460	23	77	35497	7340178	0.4331D-31
480	22	77	37037	7521811	0.7396D-31
500	22	74	37074	7655190	0.4177D-23
TOTALS	542	1652	434794	90759450	

Table 2-11-1

HE EXTENDED WOOD TEST FUNCTION
ITH ANALYTICAL GRADIENT
HE BETA TEST HYBRID METHOD

NV	NI	NF	NC	CPU	FV
4	37	79	395	128864	0.1005D-21
20	46	99	2079	563923	0.2415D-23
40	59	129	5289	1401248	0.6388D-21
60	71	154	9394	2481419	0.2044D-22
80	88	189	15309	4042537	0.7373D-25
100	105	227	22927	6040970	0.3163D-23
120	125	266	32186	8541278	0.1582D-24
140	147	309	43569	11623363	0.1332D-20
160	165	347	55867	14857628	0.7173D-27
180	163	345	62445	16668934	0.2020D-21
200	204	426	85626	22885662	0.1156D-18
220	181	382	84422	22491289	0.4978D-20
240	200	421	101461	27048704	0.2764D-21
260	249	518	135198	36349443	0.5321D-21
280	240	501	140781	37799887	0.3637D-21
300	175	369	111069	29680485	0.3041D-20
320	181	382	122622	32727395	0.3046D-21
340	176	372	126852	33866058	0.6000D-21
360	157	336	121296	32138675	0.4770D-21
380	154	331	126111	33269377	0.2896D-21
400	174	372	149172	39678736	0.4825D-21
420	201	426	179346	47887609	0.3601D-21
440	245	514	226674	60724015	0.3196D-20
460	128	284	130924	34221762	0.6323D-21
480	155	337	162097	42476575	0.1122D-19
500	139	305	152805	40203418	0.3036D-21
TOTALS	3965	8420	2405916	639799254	

Table 2-11-2

THE EXTENDED MIELE & CANTRELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE BETA TEST HYBRID METHOD

NV	NI	NF	NC	CPU	FV
4	75	186	930	407048	0.2507D-11
20	64	159	3339	1395913	0.5483D-12
40	97	249	10209	4149052	0.1358D-11
60	48	121	7381	3045007	0.1631D-11
80	51	123	9963	4124616	0.4945D-12
100	69	188	18988	7673902	0.9037D-11
120	48	125	15125	6134717	0.1183D-10
140	69	176	24816	10197792	0.7428D-11
160	57	155	24955	10015248	0.1241D-10
180	58	156	28236	11331986	0.7950D-12
200	57	155	31155	12449808	0.7732D-11
220	81	209	46189	18739888	0.1981D-12
240	64	153	36873	15228455	0.2284D-12
260	81	203	52983	21633261	0.9377D-11
280	66	173	48613	19639537	0.4052D-11
300	22	67	20167	7825811	0.4242D-11
320	65	177	56817	22813438	0.1454D-10
340	61	163	55583	22418272	0.2603D-11
360	63	157	56677	23222119	0.4778D-12
380	55	146	55626	22362281	0.1960D-11
400	58	147	58947	23897453	0.7649D-11
420	67	176	74096	29751957	0.1923D-11
440	50	139	61299	24606725	0.9041D-11
460	91	233	107413	44006607	0.2644D-12
480	54	139	66859	27211263	0.7492D-12
500	65	180	90180	36102706	0.5544D-12
TOTALS	1636	4255	1063419	430384862	

Table 2.11-3

HE EXTENDED POWELL TEST FUNCTION
ITH ANALYTICAL GRADIENT
HE BETA TEST HYBRID METHOD

NV	NI	NF	NC	CPU	FV
4	95	197	985	297734	0.2170D-13
20	89	195	4095	1032719	0.1517D-18
40	87	209	8569	2026637	0.3344D-13
60	185	387	23607	5910590	0.2455D-13
80	303	623	50463	12639303	0.2260D-13
100	239	503	50803	12575162	0.6184D-15
120	204	431	52151	12948258	0.4271D-14
140	284	599	84459	21015336	0.3893D-13
160	215	466	75026	18465839	0.6429D-13
180	340	703	127243	31946701	0.1229D-13
200	326	706	141906	34746898	0.6670D-13
220	454	935	206635	51812507	0.2928D-14
240	506	1037	249917	63006914	0.7089D-16
260	321	670	174870	43656077	0.4958D-13
280	300	621	174501	43495350	0.3391D-13
300	607	1274	383474	95164306	0.1935D-13
320	432	886	284406	71083977	0.3296D-13
340	736	1498	510818	128429045	0.3645D-13
360	437	912	329232	81430490	0.3716D-13
380	396	821	312801	77391728	0.2272D-13
400	445	919	368519	91363345	0.1711D-13
420	628	1287	541827	134882426	0.1488D-13
440	677	1385	610785	153024435	0.2380D-13
460	927	1891	871751	217973901	0.3085D-13
480	986	2003	963443	239218731	0.8690D-14
500	526	1084	543084	133458995	0.3918D-13
TOTALS	10745	22242	7145370	1778997404	

Table 2-11-4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE BETA TEST HYBRID METHOD

NV	NI	NF	NC	CPU	FV
10	108	238	2618	1143999	0.2167D-20
20	37	81	1701	692563	0.2808D-20
40	74	160	6560	2651339	0.3602D-20
60	52	116	7076	2863882	0.2551D-20
80	52	115	9315	3799361	0.8872D-20
100	67	150	15150	6127556	0.5010D-20
120	103	221	26741	10916451	0.1165D-19
140	106	227	32007	13136261	0.5678D-20
160	76	172	27692	11192538	0.1427D-19
180	57	135	24435	9755286	0.1025D-20
200	60	141	28341	11245968	0.8264D-20
220	46	112	24752	9755104	0.2841D-19
240	50	121	29161	11501784	0.1114D-19
260	58	139	36279	14343049	0.5664D-20
280	44	106	29786	11734571	0.8969D-20
300	43	106	31906	12544362	0.8928D-20
320	53	127	40767	16117771	0.6442D-20
340	49	119	40579	15963609	0.2767D-19
360	46	115	41515	16184436	0.2237D-20
380	54	131	49911	19662870	0.1229D-19
400	66	157	62957	24885345	0.1342D-19
420	75	177	74517	29458306	0.2296D-20
440	90	207	91287	36483780	0.1954D-19
460	87	203	93583	37087559	0.2425D-19
480	106	240	115440	46096240	0.3690D-19
500	100	229	114729	45778676	0.1856D-19
TOTALS	1759	4045	1058805	421122666	

Table 2.11.5

HE EXTENDED BEALE TEST FUNCTION
ITH ANALYTICAL GRADIENT
HE BETA TEST HYBRID METHOD

NV	NI	NF	NC	CPU	FV
2	14	32	96	79860	0.5590D-23
20	15	39	819	255827	0.2299D-24
40	15	42	1722	520396	0.2242D-22
60	15	42	2562	761287	0.1158D-22
80	15	44	3564	1051070	0.9194D-24
100	15	45	4545	1328669	0.4342D-23
120	15	47	5607	1655764	0.1320D-22
140	15	48	6768	1955893	0.1252D-22
160	15	49	7889	2272710	0.5015D-22
180	15	47	8507	2471075	0.3938D-22
200	15	47	9447	2739491	0.3347D-22
220	15	47	10387	3016489	0.3044D-22
240	15	49	11809	3397038	0.2959D-22
260	15	49	12789	3705429	0.2811D-22
280	15	50	14050	4041020	0.1630D-22
300	15	50	15050	4340659	0.4495D-23
320	15	51	16371	4692954	0.2579D-23
340	15	51	17391	4988330	0.0000D+00
360	13	45	16245	4640521	0.1116D-30
380	15	51	19431	5536097	0.8196D-23
400	15	52	20852	5917956	0.1700D-22
420	15	55	23155	6577509	0.1228D-22
440	15	54	23814	6751524	0.3082D-22
460	15	54	24894	7080458	0.4058D-22
480	15	54	25974	7354598	0.4742D-22
500	15	54	27054	7691818	0.5184D-22
TOTALS	387	1248	330872	94824442	

Table 8.11.6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE BETA TEST HYBRID METHOD

NV	NI	NF	NC	CPU	FV
2	8	18	54	59171	0.0000D+00
20	8	23	483	133732	0.0000D+00
40	9	25	1025	281996	0.0000D+00
60	8	25	1525	395824	0.0000D+00
80	8	25	2025	524845	0.0000D+00
100	8	25	2525	654812	0.0000D+00
120	8	25	3025	776569	0.0000D+00
140	8	26	3666	929693	0.0000D+00
160	9	28	4508	1172729	0.0000D+00
180	9	28	5068	1318622	0.0000D+00
200	8	26	5226	1323237	0.0000D+00
220	8	28	6188	1549404	0.0000D+00
240	8	28	6748	1691075	0.0000D+00
260	8	29	7569	1863301	0.0000D+00
280	7	24	6744	1668496	0.0000D+00
300	8	28	8428	2098308	0.0000D+00
320	8	29	9309	2265745	0.0000D+00
340	7	28	9548	2348983	0.0000D+00
360	7	26	9386	2337019	0.0000D+00
380	8	28	10668	2671739	0.0000D+00
400	8	28	11228	2793211	0.0000D+00
420	8	28	11788	2922440	0.0000D+00
440	8	28	12348	3075618	0.0000D+00
460	8	28	12908	3208458	0.0000D+00
480	8	28	13468	3341472	0.0000D+00
500	8	29	14529	3578494	0.0000D+00
TOTALS	208	691	179987	44984993	

Table 2-11-7

THE EXTENDED ROSENBROCK TEST FUNCTION
WITH ANALYTICA GRADIENT
HYBRID 2 METHOD (Tau = 1.0D-04)

NV	NI	NF	NC	CPU	FV
2	32	74	222	85005	0.7928D-28
20	21	50	1050	286017	0.1175D-34
40	25	58	2378	663456	0.1270D-24
60	19	53	3233	831841	0.0000D+00
80	19	54	4374	1055021	0.1436D-22
100	20	57	5757	1433975	0.3486D-20
120	22	60	7260	1859328	0.2186D-21
140	22	62	8742	2148427	0.1402D-29
160	20	60	9660	2275590	0.1372D-26
180	20	59	10679	2578110	0.1829D-23
200	22	63	12663	3165174	0.2128D-19
220	23	66	14586	3454576	0.3662D-30
240	19	63	15183	3584433	0.1411D-33
260	19	60	15660	3590026	0.1594D-22
280	19	63	17703	4142283	0.5931D-32
300	18	65	19565	4407027	0.4187D-20
320	19	63	20223	4531400	0.3462D-22
340	19	64	21824	4958627	0.1582D-19
360	18	61	22021	4989549	0.5298D-32
380	18	61	23241	5387517	0.1456D-20
400	18	61	24461	5614650	0.1065D-22
420	20	72	30312	6958542	0.4937D-32
440	23	78	34398	7882241	0.1882D-22
460	23	77	35497	8214418	0.4331D-31
480	22	77	37037	8581653	0.7396D-31
500	22	74	37074	8445854	0.4177D-23
TOTALS	542	1655	434803	101124740	

Table 2.12.1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHOD (Tau = 1.0D-04)

NV	NI	NF	NC	CPU	FV
4	37	79	395	145725	0.1005D-21
20	46	99	2079	641115	0.2415D-23
40	59	129	5289	1593650	0.6388D-21
60	71	154	9394	2834092	0.2844D-22
80	88	189	15309	4627726	0.7373D-25
100	105	227	22927	6899195	0.3163D-23
120	125	266	32186	9794742	0.1582D-24
140	147	309	43569	13290708	0.1332D-20
160	165	347	55867	17010228	0.7173D-27
180	163	345	62445	18854362	0.2020D-21
200	204	426	85626	26375078	0.1156D-18
220	181	382	84422	25694452	0.4978D-20
240	200	421	101461	31152411	0.2764D-21
260	249	518	135198	40860328	0.5321D-21
280	240	501	140781	42663462	0.3637D-21
300	175	369	111069	33671396	0.3841D-20
320	181	382	122622	37128597	0.3046D-21
340	176	372	126852	38715570	0.6000D-21
360	157	336	121296	36550108	0.4770D-21
380	154	331	126111	37747340	0.2896D-21
400	174	372	149172	45058186	0.4825D-21
420	201	426	179346	54719612	0.3601D-21
440	245	514	226674	68930176	0.3196D-20
460	128	284	130924	39050258	0.6323D-21
480	155	337	162097	48228722	0.1122D-19
500	139	305	152805	44025039	0.3036D-21
TOTALS	3965	8420	2405916	726262278	

Table 2-12-2

THE EXTENDED MIELE & CANTRELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHOD (Tau = 1.0D-04)

NV	NI	NF	NC	CPU	FV
4	75	186	930	434559	0.2507D-11
20	64	159	3339	1482474	0.5483D-12
40	97	249	10209	4496012	0.1358D-11
60	51	125	7625	3395306	0.4948D-11
80	51	123	9963	4472766	0.4945D-12
100	69	184	18584	8010825	0.8933D-11
120	48	125	15125	6643783	0.1183D-10
140	69	176	24816	10829072	0.7428D-11
160	57	155	24955	10636806	0.1241D-10
180	58	156	28236	12011396	0.7950D-12
200	57	155	31155	13293712	0.7732D-11
220	57	148	32708	14322527	0.3610D-11
240	64	153	36873	16353308	0.2284D-12
260	81	203	52983	23345246	0.9377D-11
280	66	173	48613	21225705	0.4052D-11
300	22	67	20167	8335135	0.4242D-11
320	65	177	56817	24253441	0.1454D-10
340	61	163	55583	24162139	0.2603D-11
360	63	157	56677	25336911	0.4778D-12
380	54	150	57150	24334612	0.1088D-10
400	58	147	58947	25721581	0.7649D-11
420	67	176	74096	32116036	0.1923D-11
440	50	139	61299	26295221	0.9041D-11
460	91	233	107413	46421671	0.2644D-12
480	54	139	66859	29661163	0.7492D-12
500	65	180	90180	39224216	0.5544D-12
TOTALS	1614	4198	1051302	456815623	

Table 2-12-3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHOD (Tau = 1.0D-04)

NV	NI	NF	NC	CPU	FV
4	95	197	985	351358	0.2170D-13
20	89	195	4095	1235454	0.1517D-18
40	87	205	8405	2373263	0.3077D-13
60	185	387	23607	7082392	0.2455D-13
80	303	623	50463	15438360	0.2260D-13
100	268	570	57570	17073660	0.2863D-15
120	204	431	52151	15696121	0.4271D-14
140	302	627	88407	26379286	0.5072D-13
160	215	466	75026	21634387	0.6429D-13
180	340	703	127243	38456319	0.1229D-13
200	304	646	129846	38407985	0.1019D-15
220	454	935	206635	61165200	0.2928D-14
240	503	1031	248471	74028712	0.1803D-15
260	371	779	203319	60210908	0.1649D-13
280	300	621	174501	50535690	0.3391D-13
300	607	1274	383474	109639307	0.1935D-13
320	432	886	284406	85391158	0.3296D-13
340	736	1498	510818	152078977	0.3645D-13
360	421	870	314070	93693139	0.2308D-14
380	396	821	312801	92445864	0.2272D-13
400	445	919	368519	109383075	0.1711D-13
420	628	1287	541827	160020742	0.1488D-13
440	677	1385	610785	180691694	0.2380D-13
460	927	1891	871751	257151793	0.3085D-13
480	986	2003	963443	284217998	0.8690D-14
500	526	1084	543084	162289464	0.3918D-13
TOTALS	10801	22334	7155702	2117072306	

Table 2.12.4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHOD (Tau = 1.0D-04)

NV	NI	NF	NC	CPU	FV
10	108	238	2618	1193728	0.2167D-20
20	37	81	1701	766748	0.2808D-20
40	74	160	6560	2940175	0.3602D-20
60	52	116	7076	3198535	0.2551D-20
80	52	115	9315	4087540	0.8872D-20
100	67	150	15150	6431099	0.5010D-20
120	103	221	26741	11834588	0.1165D-19
140	106	227	32007	13980541	0.5670D-20
160	76	172	27692	11641146	0.1427D-19
180	57	135	24435	10596953	0.1025D-20
200	60	141	28341	11943735	0.8264D-20
220	46	112	24752	10583314	0.2841D-19
240	50	121	29161	12681523	0.1114D-19
260	58	139	36279	15418232	0.5664D-20
280	44	106	29786	12773912	0.8969D-20
300	43	106	31906	13716322	0.8928D-20
320	53	127	40767	17493540	0.6442D-20
340	49	119	40579	17124360	0.2767D-19
360	46	115	41515	17205573	0.2237D-20
380	54	131	49911	21797961	0.1229D-19
400	66	157	62957	26191823	0.1342D-19
420	75	177	74517	31715884	0.2296D-20
440	90	207	91287	39836195	0.1954D-19
460	87	203	93583	39262407	0.2425D-19
480	106	240	115440	49400700	0.3690D-19
500	100	229	114729	49437720	0.1856D-19
TOTALS	1759	4045	1058805	453262254	

Table 2-12-5

THE EXTENDED BEALE TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHOD (Tau = 1.0 D-04)

NV	NI	NF	NC	CPU	FV
2	14	32	96	76390	0.5590D-23
20	13	36	756	244578	0.1681D-23
40	13	39	1599	506838	0.1250D-22
60	13	40	2440	763530	0.7171D-23
80	13	40	3240	1000282	0.6726D-23
100	10	35	3535	1031801	0.4897D-21
120	13	43	5203	1583136	0.6741D-26
140	13	43	6063	1849921	0.9427D-23
160	13	44	7084	2150746	0.3947D-22
180	13	42	7602	2317906	0.7083D-22
200	13	42	8442	2573435	0.7856D-22
220	13	42	9282	2836913	0.7125D-22
240	13	44	10604	3220085	0.4229D-22
260	13	44	11484	3433763	0.3133D-22
280	13	44	12364	3706745	0.3262D-22
300	13	45	13545	4045108	0.4246D-22
320	13	45	14445	4317502	0.5796D-22
340	13	46	15686	4580041	0.1508D-19
360	13	45	16245	4775445	0.1116D-30
380	13	45	17145	5069496	0.6630D-25
400	10	39	15639	4465412	0.1959D-20
420	12	51	21471	6091385	0.3369D-33
440	11	43	18963	5605192	0.3383D-18
460	13	47	21667	6430522	0.1856D-29
480	13	47	22607	6728755	0.5880D-25
500	13	48	24048	7164000	0.4157D-23
TOTALS	330	1111	291255	86568927	

Table 2-12-6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHOD (Tau = 1.0D-04)

NV	NI	NF	NC	CPU	FV
2	8	18	54	62114	0.0000D+00
20	7	21	441	133105	0.0000D+00
40	9	25	1025	295915	0.0000D+00
60	8	25	1525	426218	0.0000D+00
80	7	23	1863	520139	0.0000D+00
100	8	25	2525	701108	0.0000D+00
120	8	25	3025	820293	0.0000D+00
140	8	26	3666	970710	0.0000D+00
160	9	28	4508	1233414	0.0000D+00
180	10	28	5068	1446036	0.0000D+00
200	8	26	5226	1397292	0.0000D+00
220	8	28	6108	1645275	0.0000D+00
240	8	28	6748	1797730	0.0000D+00
260	8	29	7569	2016295	0.0000D+00
280	7	24	6744	1801171	0.0000D+00
300	7	26	7826	2075614	0.0000D+00
320	7	27	8667	2307763	0.0000D+00
340	7	28	9548	2511467	0.0000D+00
360	7	26	9386	2533900	0.0000D+00
380	8	28	10668	2836175	0.0000D+00
400	8	28	11228	2981941	0.0000D+00
420	8	28	11788	3147159	0.0000D+00
440	8	28	12348	3296997	0.0000D+00
460	8	28	12908	3454338	0.0000D+00
480	8	28	13468	3517850	0.0000D+00
500	8	29	14529	3765945	0.0000D+00
TOTALS	205	683	178539	47696065	

Table 2-12-7

THE EXTENDED ROSENBROCK TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHOD (Tau = 1.0D-06)

NV	NI	NF	NC	CPU	FV
2	32	71	213	86287	0.1785D-27
20	21	50	1050	282624	0.1175D-34
40	25	58	2378	649743	0.1270D-24
60	19	53	3233	819253	0.0000D+00
80	19	54	4374	1033944	0.1436D-22
100	20	57	5757	1426053	0.3486D-20
120	22	60	7260	1844699	0.2186D-21
140	22	62	8742	2106534	0.1402D-29
160	20	60	9660	2210036	0.1372D-26
180	20	59	10679	2515935	0.1829D-23
200	22	63	12663	3135788	0.2128D-19
220	23	66	14586	3372074	0.3662D-30
240	19	63	15183	3491303	0.1411D-33
260	19	60	15660	3509540	0.1594D-22
280	19	63	17703	4086796	0.5931D-32
300	18	65	19565	4324154	0.4187D-20
320	19	63	20223	4467695	0.3462D-22
340	19	64	21824	4910950	0.1582D-19
360	18	61	22021	4912314	0.5298D-32
380	18	61	23241	5268520	0.1456D-20
400	18	61	24461	5527530	0.1065D-22
420	20	72	30312	6820890	0.4937D-32
440	23	78	34398	7730304	0.1882D-22
460	23	77	35497	8089587	0.4331D-31
480	22	77	37037	8519297	0.7396D-31
500	22	74	37074	8295527	0.4177D-23
TOTALS	542	1652	434794	99437377	

Table 2_13_1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHOD (Tau = 1.0D-06)

NV	NI	NF	NC	CPU	FV
4	37	79	395	145946	0.1005D-21
20	46	99	2079	632133	0.2415D-23
40	59	129	5289	1579942	0.6388D-21
60	71	154	9394	2798704	0.2844D-22
80	88	189	15309	4568860	0.7373D-25
100	105	227	22927	6792104	0.3163D-23
120	125	266	32186	9642219	0.1582D-24
140	147	309	43569	13242221	0.1332D-20
160	165	347	55867	16760150	0.7173D-27
180	163	345	62445	18511565	0.2020D-21
200	204	426	85626	26098469	0.1156D-18
220	181	382	84422	25386456	0.4978D-20
240	200	421	101461	30915983	0.2764D-21
260	249	518	135198	41013376	0.5321D-21
280	240	501	140781	42726964	0.3637D-21
300	175	369	111069	33411024	0.3841D-20
320	181	382	122622	37114622	0.3046D-21
340	176	372	126852	38396751	0.6000D-21
360	157	336	121296	36321921	0.4770D-21
380	154	331	126111	37547772	0.2896D-21
400	174	372	149172	44746956	0.4825D-21
420	201	426	179346	54152306	0.3601D-21
440	245	514	226674	68732668	0.3196D-20
460	128	284	130924	38841082	0.6323D-21
480	155	337	162097	47850837	0.1122D-19
500	139	305	152805	43598997	0.3036D-21
TOTALS	3965	8420	2405916	721538028	

Table 2-13-2

THE EXTENDED MIELE & CANTRELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHOD (Tau = 1.0D-06)

NV	NI	NF	NC	CPU	FV
4	75	186	930	431239	0.2507D-11
20	64	159	3339	1481345	0.5483D-12
40	97	249	10209	4471930	0.1358D-11
60	51	125	7625	3398347	0.4948D-11
80	51	123	9963	4487892	0.4945D-12
100	69	184	18584	7997144	0.8933D-11
120	48	125	15125	6636646	0.1183D-10
140	69	176	24816	10861922	0.7428D-11
160	57	155	24955	10705317	0.1241D-10
180	58	156	28236	12153189	0.7950D-12
200	57	155	31155	13189043	0.7732D-11
220	81	209	46189	19830985	0.1981D-12
240	64	153	36873	16267089	0.2284D-12
260	81	203	52983	23186884	0.9377D-11
280	66	173	48613	20931353	0.4052D-11
300	22	67	20167	8321471	0.4242D-11
320	65	177	56817	23890830	0.1454D-10
340	61	163	55583	23806480	0.2603D-11
360	63	157	56677	25054112	0.4778D-12
380	55	146	55626	23829799	0.1960D-11
400	58	147	58947	25510399	0.7649D-11
420	67	176	74096	31881980	0.1923D-11
440	50	139	61299	26122459	0.9041D-11
460	91	233	107413	46021864	0.2644D-12
480	54	139	66859	29450220	0.7492D-12
500	65	180	90180	38679544	0.5544D-12
TOTALS	1639	4255	1063259	458599483	

Table 2-13-3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHOD (Tau = 1.0D-06)

NV	NI	NF	NC	CPU	FV
4	95	197	985	340205	0.2170D-13
20	89	195	4095	1236766	0.1517D-18
40	87	209	8569	2389980	0.3344D-13
60	185	387	23607	7123442	0.2455D-13
80	303	623	50463	15462700	0.2260D-13
100	239	503	50803	15312396	0.6184D-15
120	204	431	52151	15684394	0.4271D-14
140	284	599	84459	24798096	0.3893D-13
160	215	466	75026	21586147	0.6429D-13
180	340	703	127243	38363291	0.1229D-13
200	326	706	141906	40694993	0.6670D-13
220	454	935	206635	60895231	0.2928D-14
240	506	1037	249917	74214021	0.7089D-16
260	321	670	174870	52052994	0.4958D-13
280	300	621	174501	50316274	0.3391D-13
300	607	1274	383474	109595596	0.1935D-13
320	432	886	284406	85499078	0.3296D-13
340	736	1498	510818	152051687	0.3645D-13
360	437	912	329232	97127100	0.3716D-13
380	396	821	312801	92298780	0.2272D-13
400	445	919	368519	109296543	0.1711D-13
420	628	1287	541827	159700522	0.1488D-13
440	677	1385	610785	179303894	0.2380D-13
460	927	1891	871751	255513540	0.3085D-13
480	986	2003	963443	282961398	0.8690D-14
500	526	1084	543084	161564541	0.3918D-13
TOTALS	10745	22242	7145370	2105383609	

Table 2-13-4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHD (Tau = 1.0D-06)

NV	NI	NF	NC	CPU	FV
10	108	238	2618	1237473	0.2167D-20
20	37	81	1701	757348	0.2808D-20
40	74	160	6560	2915026	0.3602D-20
60	52	116	7076	3210221	0.2551D-20
80	52	115	9315	4124206	0.8872D-20
100	67	150	15150	6428069	0.5010D-20
120	103	221	26741	11771549	0.1165D-19
140	106	227	32007	14012334	0.5678D-20
160	76	172	27692	11632617	0.1427D-19
180	57	135	24435	10527597	0.1025D-20
200	60	141	28341	11866423	0.8264D-20
220	46	112	24752	10496143	0.2841D-19
240	50	121	29161	12584893	0.1114D-19
260	58	139	36279	15324733	0.5664D-20
280	44	106	29786	12702472	0.8969D-20
300	43	106	31906	13679008	0.8928D-20
320	53	127	40767	17443795	0.6442D-20
340	49	119	40579	17017168	0.2767D-19
360	46	115	41515	17301629	0.2237D-20
380	54	131	49911	21631600	0.1229D-19
400	66	157	62957	26119087	0.1342D-19
420	75	177	74517	31639999	0.2296D-20
440	90	207	91287	39742825	0.1954D-19
460	87	203	93583	39376479	0.2425D-19
480	106	240	115440	49677892	0.3690D-19
500	100	229	114729	49661449	0.1856D-19
TOTALS	1759	4045	1058805	452882035	

Table 2-13-5

HE EXTENDED BEALE TEST FUNCTION
ITH ANALYTICAL GRADIENT
YBRID 2 METHOD (Tau = 1.0 D-06)

NV	NI	NF	NC	CPU	FV
2	14	32	96	76390	0.5590D-23
20	13	36	756	244578	0.1681D-23
40	13	39	1599	506838	0.1250D-22
60	13	40	2440	763530	0.7171D-23
80	13	40	3240	1000282	0.6726D-23
100	10	35	3535	1031801	0.4897D-21
120	13	43	5203	1583136	0.6741D-26
140	13	43	6063	1849921	0.9427D-23
160	13	44	7084	2150746	0.3947D-22
180	13	42	7602	2317906	0.7083D-22
200	13	42	8442	2573435	0.7856D-22
220	13	42	9282	2836913	0.7125D-22
240	13	44	10604	3220085	0.4229D-22
260	13	44	11484	3433763	0.3133D-22
280	13	44	12364	3706745	0.3262D-22
300	13	45	13545	4045108	0.4246D-22
320	13	45	14445	4317502	0.5796D-22
340	13	46	15686	4580041	0.1508D-19
360	13	45	16245	4775445	0.1116D-30
380	13	45	17145	5069496	0.6630D-25
400	10	39	15639	4465412	0.1959D-20
420	12	51	21471	6091385	0.3369D-33
440	11	43	18963	5605192	0.3383D-18
460	13	47	21667	6430522	0.1856D-29
480	13	47	22607	6728755	0.5880D-25
500	13	48	24048	7164000	0.4157D-23
TOTALS	330	1111	291255	86568927	

Table 2-13-6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHOD (Tau = 1.0 D-06)

NV	NI	NF	NC	CPU	FV
2	8	18	54	62114	0.0000D+00
20	7	21	441	133185	0.0000D+00
40	9	25	1025	295915	0.0000D+00
60	8	25	1525	426218	0.0000D+00
80	7	23	1863	520139	0.0000D+00
100	8	25	2525	701108	0.0000D+00
120	8	25	3025	820293	0.0000D+00
140	8	26	3666	970718	0.0000D+00
160	9	28	4508	1233414	0.0000D+00
180	10	28	5068	1446036	0.0000D+00
200	8	26	5226	1397292	0.0000D+00
220	8	28	6188	1645275	0.0000D+00
240	8	28	6748	1797730	0.0000D+00
260	8	29	7569	2016295	0.0000D+00
280	7	24	6744	1801171	0.0000D+00
300	7	26	7826	2075614	0.0000D+00
320	7	27	8667	2307768	0.0000D+00
340	7	28	9548	2511467	0.0000D+00
360	7	26	9386	2533908	0.0000D+00
380	8	28	10668	2836175	0.0000D+00
400	8	28	11228	2981941	0.0000D+00
420	8	28	11788	3147159	0.0000D+00
440	8	28	12348	3296997	0.0000D+00
460	8	28	12908	3454338	0.0000D+00
480	8	28	13468	3517850	0.0000D+00
500	8	29	14529	3765945	0.0000D+00
TOTALS	205	683	178539	47696065	

Table 2-13-7

HE EXTENDED ROSENBROCK TEST FUNCTION
ITH ANALYTICAL GRADIENT
YBRID 2 METHOD (Tau = 1.0D-08)

NV	NI	NF	NC	CPU	FV
2	32	71	213	87778	0.1785D-27
20	20	48	1008	285506	0.3973D-23
40	29	65	2665	743699	0.3412D-22
60	23	61	3721	905361	0.6383D-34
80	21	57	4617	1109383	0.1476D-20
100	24	64	6464	1590994	0.2132D-22
120	24	66	7986	1958080	0.1410D-28
140	24	66	9306	2221479	0.1911D-25
160	22	65	10465	2428708	0.1953D-29
180	21	61	11041	2554226	0.2053D-30
200	24	68	13668	3348231	0.3133D-19
220	22	65	14365	3312325	0.4658D-29
240	22	68	16388	3689101	0.5291D-30
260	20	64	16704	3836773	0.6320D-22
280	20	63	17703	3968964	0.2450D-18
300	20	65	19565	4316647	0.1357D-18
320	21	67	21507	4863774	0.9090D-20
340	21	71	24211	5305654	0.1619D-33
360	21	67	24187	5487469	0.1277D-24
380	21	68	25908	5826771	0.3929D-20
400	21	67	26867	6073305	0.1085D-21
420	22	73	30733	6854899	0.6280D-29
440	24	78	34398	7785725	0.1629D-20
460	27	84	38724	9127224	0.6397D-25
480	24	79	37999	8562869	0.2838D-27
500	25	80	40080	9149650	0.2636D-21
TOTALS	595	1751	460493	105394595	

Table 2-14-1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHOD (Tau = 1.0D-08)

NV	NI	NF	NC	CPU	FV
4	34	77	385	141701	0.1233D-26
20	54	118	2478	753731	0.9169D-22
40	59	129	5289	1613898	0.6388D-21
60	81	175	10675	3217084	0.1631D-23
80	85	187	15147	4571509	0.7686D-23
100	108	231	23331	6911589	0.5449D-22
120	125	266	32186	9779676	0.1582D-24
140	147	309	43569	13210763	0,4153D-20
160	165	347	55867	16853871	0.7173D-27
180	163	345	62445	18675623	0.2020D-21
200	204	426	85626	26353408	0.1156D-18
220	202	424	93704	28785140	0.7469D-21
240	202	424	102184	31156775	0.1527D-21
260	249	518	135198	41032417	0.5321D-21
280	240	501	140781	42896511	0.3637D-21
300	229	487	146587	43927485	0.6600D-20
320	264	553	177513	54348709	0.1253D-19
340	194	411	140151	42442443	0.3537D-20
360	137	296	106856	31779318	0.1668D-20
380	152	326	124206	36979719	0.4155D-21
400	189	401	160801	48207821	0.4482D-21
420	228	479	201659	60552289	0.1016D-19
440	245	514	226674	68577055	0.3196D-20
460	128	284	130924	38672786	0.6323D-21
480	155	337	162097	47676922	0.1122D-19
500	236	499	249999	75069460	0.2424D-20
TOTALS	4275	9064	2636332	794187783	

Table 2-14-2

HE EXTENDED MIELE & CANTRELL TEST FUNCTION
ITH ANALYTICAL GRADIENT
/BRID 2 METHOD (Tau = 1.0D-08)

NV	NI	NF	NC	CPU	FV
4	75	186	930	425942	0.2507D-11
20	103	243	5103	2234994	0.8055D-12
40	70	151	6191	2856566	0.2685D-12
60	46	128	7808	3286710	0.3298D-11
80	55	125	10125	4653553	0.1216D-11
100	69	184	18584	7935723	0.8933D-11
120	67	169	20449	8816134	0.4152D-11
140	79	205	28905	12354992	0.7145D-12
160	69	167	26887	11810616	0.2905D-11
180	60	169	30589	12908561	0.6248D-11
200	88	246	49446	20537267	0.1742D-11
220	61	160	35360	15179901	0.9613D-11
240	78	184	44344	19507505	0.9334D-12
260	74	200	52200	21952330	0.1324D-10
280	43	113	31753	13648835	0.5761D-11
300	19	60	18060	7284710	0.4399D-11
320	65	177	56817	24223323	0.1454D-10
340	76	220	75020	31642457	0.6889D-11
360	63	157	56677	25253977	0.4778D-12
380	55	146	55626	24045126	0.1960D-11
400	53	142	56942	24364781	0.7973D-11
420	72	198	83358	35081547	0.9784D-11
440	60	156	68796	29553026	0.2002D-11
460	53	136	62696	27551767	0.3170D-11
480	67	166	79846	35321682	0.1895D-10
500	98	288	144288	60207532	0.5314D-11
TOTALS	1718	4476	1126800	482639557	

Table 2.14.3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHOD (Tau = 1.0D-08)

NV	NI	NF	NC	CPU	FV
4	95	197	985	362906	0.2170D-13
20	128	273	5733	1790810	0.2496D-14
40	128	270	11070	3444533	0.2152D-13
60	186	397	24217	7211537	0.4298D-13
80	180	381	30861	9460608	0.2007D-13
100	279	583	58883	17970311	0.9526D-14
120	252	529	64009	19357646	0.3044D-13
140	302	627	88407	26594646	0.5072D-13
160	317	664	106904	32050957	0.2006D-13
180	364	751	135931	41291512	0.1995D-13
200	380	784	157584	47323368	0.1636D-13
220	510	1046	231166	68573239	0.8545D-14
240	487	999	240759	72040493	0.4387D-20
260	576	1175	306675	91758977	0.3240D-13
280	295	614	172534	50710783	0.6181D-14
300	306	651	195951	56363184	0.6787D-13
320	659	1342	430782	129067934	0.5517D-14
340	736	1498	510818	152484092	0.3645D-13
360	724	1476	532836	159529283	0.2792D-13
380	419	868	330708	99151361	0.8431D-13
400	445	919	368519	110065066	0.1711D-13
420	460	952	400792	118795417	0.1035D-13
440	551	1133	499653	148753253	0.3261D-13
460	526	1084	499724	149053910	0.1880D-14
480	570	1171	563251	168097042	0.2594D-13
500	526	1084	543084	162144304	0.3918D-13
TOTALS	10401	21468	6511836	1943447172	

Table 2-14-4

HE EXTENDED DIXON TEST FUNCTION
ITH ANALYTICAL GRADIENT
YBRID 2 METHOD (Tau = 1.0D-08)

NV	NI	NF	NC	CPU	FV
10	108	238	2618	1166162	0.2167D-20
20	37	81	1701	747778	0.2808D-20
40	74	160	6560	2850160	0.3602D-20
60	52	116	7076	3129017	0.2551D-20
80	52	115	9315	3998530	0.8872D-20
100	67	150	15150	6343800	0.5010D-20
120	103	221	26741	11637921	0.1165D-19
140	106	227	32007	13834677	0.5678D-20
160	76	172	27692	11493519	0.1427D-19
180	57	135	24435	10438723	0.1025D-20
200	60	141	28341	11795093	0.8264D-20
220	46	112	24752	10517158	0.2841D-19
240	50	121	29161	12537113	0.1114D-19
260	58	139	36279	15217451	0.5664D-20
280	44	106	29786	12641490	0.8969D-20
300	43	106	31906	13506189	0.8928D-20
320	53	127	40767	17327386	0.6442D-20
340	49	119	40579	16911305	0.2767D-19
360	46	115	41515	16955432	0.2237D-20
380	54	131	49911	21287922	0.1229D-19
400	66	157	62957	25852217	0.1342D-19
420	75	177	74517	31231054	0.2296D-20
440	90	207	91287	39411355	0.1954D-19
460	87	203	93583	38775318	0.2425D-19
480	106	240	115440	49005193	0.3690D-19
500	100	229	114729	49391703	0.1856D-19
TOTALS	1759	4045	1058805	448003674	

Table 2-14-5

THE EXTENDED BEALE TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHOD (Tau = 1.0 D - 08)

NV	NI	NF	NC	CPU	FV
2	14	32	96	76390	0.5590D-23
20	13	36	756	244578	0.1681D-23
40	13	39	1599	506838	0.1250D-22
60	13	40	2440	763530	0.7171D-23
80	13	40	3240	1000282	0.6726D-23
100	10	35	3535	1031801	0.4897D-21
120	13	43	5203	1583136	0.6741D-26
140	13	43	6063	1849921	0.9427D-23
160	13	44	7084	2150746	0.3947D-22
180	13	42	7602	2317906	0.7083D-22
200	13	42	8442	2573435	0.7856D-22
220	13	42	9282	2836913	0.7125D-22
240	13	44	10604	3220085	0.4229D-22
260	13	44	11484	3433763	0.3133D-22
280	13	44	12364	3706745	0.3262D-22
300	13	45	13545	4045108	0.4246D-22
320	13	45	14445	4317502	0.5796D-22
340	13	46	15686	4580041	0.1508D-19
360	13	45	16245	4775445	0.1116D-30
380	13	45	17145	5069496	0.6630D-25
400	10	39	15639	4465412	0.1959D-20
420	12	51	21471	6091385	0.3369D-33
440	11	43	18963	5605192	0.3383D-18
460	13	47	21667	6430522	0.1856D-29
480	13	47	22607	6728755	0.5800D-25
500	13	48	24048	7164000	0.4157D-23
TOTALS	330	1111	291255	86568927	

Table 2.14.6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 2 METHOD (Tau = 1.0D-08)

NV	NI	NF	NC	CPU	FV
2	8	18	54	62114	0.0000D+00
20	7	21	441	133185	0.0000D+00
40	9	25	1025	295915	0.0000D+00
60	8	25	1525	426218	0.0000D+00
80	7	23	1863	520139	0.0000D+00
100	8	25	2525	701108	0.0000D+00
120	8	25	3025	820293	0.0000D+00
140	8	26	3666	970718	0.0000D+00
160	9	28	4508	1233414	0.0000D+00
180	10	28	5068	1446036	0.0000D+00
200	8	26	5226	1397292	0.0000D+00
220	8	28	6188	1645275	0.0000D+00
240	8	28	6748	1797730	0.0000D+00
260	8	29	7569	2016295	0.0000D+00
280	7	24	6744	1801171	0.0000D+00
300	7	26	7826	2075614	0.0000D+00
320	7	27	8667	2307768	0.0000D+00
340	7	28	9548	2511467	0.0000D+00
360	7	26	9386	2533900	0.0000D+00
380	8	28	10668	2836175	0.0000D+00
400	8	28	11228	2981941	0.0000D+00
420	8	28	11788	3147159	0.0000D+00
440	8	28	12348	3296997	0.0000D+00
460	8	28	12908	3454338	0.0000D+00
480	8	28	13468	3517850	0.0000D+00
500	8	29	14529	3765945	0.0000D+00
TOTALS	205	683	178539	47696065	

Table 2-14-1

THE EXTENDED ROSENBROCK TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 3 METHOD (Mu = 0.1 , Lambda = 1.0D-08)

NV	NI	NF	NC	CPU	FV
2	33	71	213	91932	0.5640D-19
20	25	57	1197	297456	0.9957D-20
40	26	59	2419	585456	0.3483D-20
60	23	59	3599	817590	0.3229D-20
80	23	62	5022	1109791	0.3785D-20
100	23	65	6565	1412797	0.3390D-22
120	24	67	8107	1764042	0.8768D-30
140	22	64	9024	1939272	0.1185D-33
160	27	73	11753	2587812	0.1956D-18
180	28	76	13756	3021416	0.1242D-22
200	28	77	15477	3393132	0.4969D-23
220	19	59	13039	2736613	0.2660D-22
240	23	68	16388	3505720	0.7375D-20
260	23	69	18009	3803406	0.2234D-21
280	23	69	19389	4110597	0.1618D-19
300	23	72	21672	4537707	0.1839D-19
320	23	71	22791	4793805	0.2695D-20
340	28	85	28985	6112915	0.1349D-20
360	28	86	31046	6514348	0.1679D-20
380	28	86	32766	6850017	0.2353D-20
400	28	86	34486	7222865	0.3393D-20
420	28	87	36627	7600499	0.3429D-20
440	28	87	38367	7948716	0.3726D-20
460	28	87	40107	8302709	0.4241D-20
480	28	89	42809	8792355	0.4719D-20
500	28	89	44589	9213509	0.4909D-20
TOTALS	668	1920	518202	109066477	

Table 2-15-1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 3 METHOD (Mu = 0.1 , Lambda = 1.0D-00)

NV	NI	NF	NC	CPU	FV
4	49	104	520	175901	0.2362D-20
20	52	113	2373	630185	0.1028D-19
40	45	103	4223	1072981	0.7259D-20
60	59	132	8052	2083580	0.9497D-21
80	62	139	11259	2874936	0.8215D-20
100	52	120	12120	3059276	0.2575D-22
120	67	149	18029	4562320	0.7968D-21
140	60	137	19317	4851439	0.3899D-22
160	49	116	18676	4615814	0.8913D-22
180	54	128	23168	5719761	0.1325D-21
200	54	128	25728	6316246	0.1088D-21
220	47	116	25636	6223180	0.2490D-23
240	56	135	32535	7950737	0.1973D-21
260	50	123	32103	7749852	0.2339D-23
280	46	117	32877	7881400	0.2939D-20
300	87	212	63812	15469885	0.4169D-19
320	56	143	45903	10931850	0.6463D-23
340	42	114	38874	9113506	0.9287D-24
360	57	146	52706	12591963	0.3419D-25
380	30	86	32766	7470010	0.1754D-22
400	47	119	47719	11375176	0.3561D-20
420	49	124	52204	12483052	0.5028D-22
440	51	129	56889	13683625	0.7213D-22
460	50	128	59008	14152561	0.1710D-21
480	58	145	69745	16803078	0.1088D-21
500	56	144	72144	17174771	0.1120D-18
TOTALS	1385	3350	858386	207017085	

Table 2-15-2

THE EXTENDED MIELE & CANTRELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 3 METHOD (Mu = 0.1 , Lambda = 1.0D-08)

NV	NI	NF	NC	CPU	FV
4	61	129	645	306000	0.1657D-12
20	59	164	3444	1387509	0.1388D-11
40	75	188	7708	3155339	0.1280D-11
60	46	128	7808	3113082	0.3298D-11
80	72	193	15633	6220525	0.2947D-11
100	62	164	16564	6640992	0.7017D-11
120	52	142	17182	6841176	0.5365D-11
140	81	221	31161	12408840	0.7162D-11
160	73	188	30268	12189958	0.4947D-11
180	65	177	32037	12713485	0.1361D-10
200	56	146	29346	11763358	0.1125D-10
220	64	158	34918	14133341	0.2645D-11
240	55	138	33258	13399111	0.6829D-11
260	94	248	64728	25850897	0.1230D-10
280	44	113	31753	12727850	0.5268D-11
300	19	60	18060	6890122	0.4399D-11
320	91	253	81213	32137201	0.1253D-11
340	59	147	50127	20209462	0.5035D-11
360	78	199	71839	28794371	0.4173D-11
380	55	146	55626	22230718	0.1960D-11
400	49	132	52932	20945299	0.3933D-11
420	75	184	77464	31487324	0.5320D-11
440	84	228	100548	40042575	0.1121D-11
460	59	171	78831	30887012	0.4902D-11
480	73	195	93795	37569971	0.5520D-11
500	80	218	109218	43578040	0.6395D-11
TOTALS	1681	4430	1146106	457623558	

Table 2.15.3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 3 METHOD (Mu = 0.1 , Lambda = 1.0D-08)

NV	NI	NF	NC	CPU	FV
4	118	244	1220	429119	0.8263D-14
20	115	245	5145	1355081	0.4775D-14
40	121	259	10619	2666942	0.7737D-14
60	56	132	8052	1916166	0.3482D-13
80	90	201	16281	3942234	0.5351D-14
100	86	193	19493	4699522	0.4283D-13
120	158	337	40777	10053752	0.8858D-13
140	110	241	33981	8272234	0.3854D-13
160	205	433	69713	17290263	0.9393D-14
180	225	477	86337	21436588	0.1858D-13
200	129	286	57486	13918223	0.1837D-13
220	136	305	67405	16160656	0.6891D-14
240	157	342	82422	20021694	0.6521D-13
260	228	482	125802	31108909	0.3067D-13
280	163	352	98912	24106446	0.5462D-13
300	142	313	94213	22746067	0.9133D-13
320	181	389	124869	30504987	0.7245D-14
340	128	285	97185	23395325	0.5473D-13
360	106	241	87001	20808552	0.3699D-13
380	138	307	116967	28213578	0.4093D-13
400	62	160	64160	14438514	0.2970D-13
420	95	222	93462	21951652	0.8963D-13
440	73	180	79380	18272190	0.1905D-13
460	168	367	169187	41142884	0.8367D-13
480	96	242	116402	26522625	0.6419D-13
500	78	189	94689	21865207	0.5159D-14
TOTALS	3364	7424	1861160	447239410	

Table 2.15_4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 3 METHOD (Mu = 0.1 , Lambda = 1.0D-08)

NV	NI	NF	NC	CPU	FV
10	95	217	2387	995454	0.2771D-25
20	40	87	1827	748097	0.1656D-19
40	93	198	8118	3307938	0.3368D-20
60	45	102	6222	2477877	0.1409D-19
80	39	89	7209	2890739	0.1815D-19
100	63	141	14241	5706770	0.7249D-20
120	87	188	22748	9171067	0.1146D-20
140	85	184	25944	10405980	0.7734D-21
160	69	156	25116	10002008	0.7931D-21
180	61	141	25521	10089970	0.6038D-20
200	60	142	28542	11213122	0.7658D-20
220	44	107	23647	9205389	0.2065D-26
240	45	111	26751	10377744	0.2305D-20
260	56	132	34452	13487150	0.1576D-22
280	40	98	27538	10696548	0.1065D-19
300	39	97	29197	11323549	0.9820D-20
320	43	106	34026	13157955	0.3922D-19
340	43	107	36487	14121439	0.8724D-20
360	43	112	40432	15458369	0.1055D-24
380	57	138	52578	20521949	0.2215D-20
400	65	152	60952	23991102	0.1061D-19
420	70	165	69465	27241820	0.2114D-20
440	88	203	89523	35193829	0.1720D-19
460	94	216	99576	39179713	0.7149D-20
480	81	189	90909	35519496	0.5498D-20
500	77	180	90180	35241123	0.1237D-19
TOTALS	1622	3758	973588	381726997	

Table 2-15-5

HE EXTENDED BEALE TEST FUNCTION
ITH ANALYTICAL GRADIENT
YBRID 3 METHOD

NV	NI	NF	NC	CPU	FV
2	12	29	87	75822	0.2622D-27
20	13	36	756	231906	0.1681D-23
40	13	39	1599	476625	0.1250D-22
60	13	40	2440	717523	0.7171D-23
80	13	40	3240	949393	0.6726D-23
100	10	35	3535	1006164	0.4897D-21
120	13	43	5203	1503458	0.6741D-26
140	13	43	6063	1738953	0.9427D-23
160	13	44	7084	2023663	0.3947D-22
180	13	42	7602	2161323	0.7083D-22
200	13	42	8442	2439042	0.7856D-22
220	13	42	9282	2650310	0.7125D-22
240	13	44	10604	2997027	0.4229D-22
260	13	44	11484	3289662	0.3133D-22
280	13	44	12364	3482282	0.3262D-22
300	13	45	13545	3873205	0.4246D-22
320	13	45	14445	4121320	0.5796D-22
340	13	46	15686	4414248	0.1508D-19
360	13	45	16245	4636200	0.1116D-30
380	13	45	17145	4866399	0.6630D-25
400	10	39	15639	4330858	0.1959D-20
420	12	51	21471	5915313	0.3369D-33
440	11	43	18963	5269699	0.3383D-18
460	13	47	21667	6082529	0.1856D-29
480	13	47	22607	6395397	0.5080D-25
500	13	48	24048	6753163	0.4157D-23
TOTALS	328	1108	291246	82401484	

Table 2-15-6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
HYBRID 3 METHOD

NV	NI	NF	NC	CPU	FV
2	7	16	48	65989	0.0000D+00
20	7	21	441	121370	0.0000D+00
40	9	25	1025	285942	0.0000D+00
60	8	25	1525	397003	0.0000D+00
80	7	23	1863	477456	0.0000D+00
100	8	25	2525	654329	0.0000D+00
120	8	25	3025	781579	0.0000D+00
140	8	26	3666	929414	0.0000D+00
160	9	28	4508	1179213	0.0000D+00
180	10	28	5068	1352664	0.0000D+00
200	8	26	5226	1331096	0.0000D+00
220	8	28	6188	1555635	0.0000D+00
240	8	28	6748	1690694	0.0000D+00
260	8	29	7569	1878765	0.0000D+00
280	7	24	6744	1685918	0.0000D+00
300	7	26	7826	1934307	0.0000D+00
320	7	27	8667	2137485	0.0000D+00
340	7	28	9548	2328310	0.0000D+00
360	7	26	9386	2337829	0.0000D+00
380	8	28	10668	2663375	0.0000D+00
400	8	28	11228	2795032	0.0000D+00
420	8	28	11788	2931660	0.0000D+00
440	8	28	12348	3071190	0.0000D+00
460	8	28	12908	3203782	0.0000D+00
480	8	28	13468	3340579	0.0000D+00
500	8	29	14529	3595960	0.0000D+00
TOTALS	204	681	178533	44726576	

Table 2.15.7

HE EXTENDED ROSENBROCK TEST FUNCTION
ITH ANALYTICAL GRADIENT
HE F-R METHOD WITH NEW RESTART

NV	NI	NF	NC	CPU	FV
2	51	118	354	166098	0.4533D-26
20	47	101	2121	496723	0.4628D-24
40	33	73	2993	670127	0.1379D-21
60	52	113	6893	1530429	0.4404D-19
80	48	106	8586	1882605	0.6275D-23
100	36	91	9191	1900733	0.1036D-27
120	41	99	11979	2533898	0.1690D-20
140	41	100	14100	2944324	0.1522D-19
160	41	103	16583	3425641	0.1113D-21
180	43	111	20091	4135723	0.9814D-23
200	44	113	22713	4671408	0.1402D-24
220	44	112	24752	5122932	0.3674D-24
240	41	109	26269	5355376	0.3732D-20
260	40	107	27927	5697996	0.1554D-21
280	45	120	33720	6851574	0.1792D-20
300	46	123	37023	7483989	0.7409D-20
320	50	131	42051	8547057	0.2457D-21
340	51	133	45353	9249901	0.8770D-20
360	42	116	41876	8384096	0.6596D-21
380	51	135	51435	10493657	0.1193D-20
400	47	128	51328	10322650	0.1551D-18
420	50	134	56414	11454150	0.1363D-21
440	50	134	59094	11983100	0.2265D-21
460	50	134	61774	12541017	0.2807D-21
480	50	134	64454	13048331	0.3609D-21
500	50	134	67134	13648940	0.5487D-21
TOTALS	1184	3012	806208	164542475	

Table 2_16_1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE F-R METHOD WITH NEW RESTART

NV	NI	NF	NC	CPU	FV
4	46	99	495	153599	0.1184D-25
20	42	93	1953	484857	0.7360D-19
40	46	101	4141	1003649	0.4823D-19
60	55	124	7564	1802683	0.3277D-20
80	54	124	10044	2360104	0.7175D-21
100	42	101	10201	2353767	0.2223D-24
120	46	109	13189	3044622	0.6066D-20
140	45	107	15087	3452675	0.4433D-25
160	47	111	17871	4117653	0.8211D-21
180	55	131	23711	5430956	0.2622D-21
200	53	127	25527	5835376	0.2566D-21
220	47	116	25636	5822140	0.1192D-21
240	53	130	31330	7119416	0.4998D-19
260	63	150	39150	8979256	0.8626D-21
280	62	149	41869	9517659	0.6406D-21
300	75	186	55986	12680823	0.1197D-22
320	58	144	46224	10484351	0.4223D-21
340	51	131	44671	9989213	0.3866D-22
360	53	133	48013	10816581	0.3435D-24
380	49	124	47244	10624227	0.2605D-20
400	52	130	52130	11796385	0.2950D-22
420	43	113	47573	10534195	0.1329D-19
440	60	147	64827	14668576	0.4728D-19
460	52	132	60852	13647490	0.3323D-23
480	52	134	64454	14339561	0.1046D-21
500	62	153	76653	17353680	0.1005D-22
TOTALS	1363	3299	876395	198413494	

Table 2.16.2

HE EXTENDED MIELE & CANTRELL TEST FUNCTION
ITH ANALYTICAL GRADIENT
HE F-R METHOD WITH NEW RESTART

NV	NI	NF	NC	CPU	FV
4	66	148	740	325754	0.1050D-11
20	92	247	5187	1987988	0.1892D-11
40	65	161	6601	2538233	0.1540D-11
60	83	205	12505	4818832	0.1789D-11
80	84	197	15957	6164919	0.2872D-11
100	81	212	21412	8207184	0.3510D-11
120	78	204	24684	9409743	0.6024D-11
140	75	183	25803	9991205	0.3498D-11
160	78	183	29463	11539491	0.3818D-11
180	92	240	43440	16592737	0.1648D-11
200	86	228	45828	17460721	0.2061D-11
220	87	212	46852	18099921	0.3445D-11
240	98	262	63142	24012273	0.2742D-11
260	70	169	44109	16994763	0.2394D-11
280	79	201	56481	21588523	0.3362D-11
300	75	188	56588	21648845	0.7969D-11
320	87	202	64842	25205819	0.4559D-11
340	71	184	62744	23971348	0.7228D-11
360	79	203	73283	28000836	0.4601D-11
380	69	174	66294	25470420	0.1101D-11
400	111	276	110676	42663487	0.4502D-11
420	88	215	90515	34993022	0.1136D-10
440	86	220	97020	37204403	0.5496D-11
460	87	233	107413	40881540	0.5508D-11
480	86	226	100706	41482436	0.1170D-10
500	80	209	104709	39974799	0.1348D-10
TOTALS	2133	5382	1384994	531230042	

Table 2-16-3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE F-R METHOD WITH NEW RESTART

NV	NI	NF	NC	CPU	FV
4	115	236	1180	348118	0.1737D-13
20	147	306	6426	1528780	0.2210D-13
40	80	179	7339	1632288	0.8276D-14
60	103	224	13664	3023674	0.5826D-13
80	138	294	23814	5317667	0.2468D-13
100	115	252	25452	5565043	0.1305D-13
120	122	266	32186	7059782	0.6850D-14
140	105	236	33276	7178109	0.4070D-14
160	101	230	37030	7969287	0.1358D-13
180	134	296	53576	11656173	0.9436D-13
200	105	239	48039	10343817	0.1501D-13
220	126	279	61659	13347802	0.3237D-13
240	159	344	82904	18183605	0.5515D-13
260	113	254	66294	14330803	0.1202D-13
280	125	281	78961	17080557	0.9649D-14
300	130	287	86387	18783961	0.1123D-12
320	90	211	67731	14382312	0.4667D-14
340	96	224	76384	16289134	0.3209D-14
360	103	238	85918	18385344	0.1201D-12
380	126	284	108204	23344814	0.3383D-13
400	125	280	112280	24211147	0.1144D-13
420	73	177	74517	15740340	0.7354D-13
440	113	256	112896	24283953	0.1916D-12
460	121	277	127697	27376485	0.9234D-14
480	112	259	124579	26415088	0.2867D-13
500	87	209	104709	22003543	0.1329D-12
TOTALS	2964	6618	1653102	355781706	

Table 2.16.4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE F-R METHOD WITH NEW RESTART

NV	NI	NF	NC	CPU	FV
10	87	201	2211	866329	0.1891D-20
20	46	99	2079	807070	0.3611D-19
40	87	192	7872	2955353	0.1479D-20
60	53	118	7198	2705049	0.3665D-21
80	47	104	8424	3161416	0.5600D-20
100	83	186	18706	7048535	0.1923D-19
120	101	215	26015	9889613	0.1819D-19
140	101	215	30315	11505241	0.2458D-20
160	81	178	28658	10811810	0.5283D-20
180	68	154	27874	10426966	0.1303D-19
200	68	154	30954	11582780	0.9092D-20
220	53	125	27625	10209531	0.1338D-19
240	56	132	31812	11727716	0.3220D-19
260	78	175	45675	17067993	0.7034D-20
280	40	98	27538	10068667	0.1031D-19
300	46	111	33411	12237959	0.2787D-20
320	50	121	38841	14262767	0.6703D-20
340	52	126	42966	15867787	0.5343D-20
360	56	131	47291	17570272	0.7422D-20
380	57	134	51054	18900633	0.5060D-20
400	83	190	76190	28363699	0.2078D-20
420	83	190	79990	29764745	0.2945D-20
440	88	200	88200	32707094	0.9030D-20
460	112	250	115250	42934617	0.6263D-20
480	104	233	112073	41664889	0.1289D-19
500	104	235	117735	43703148	0.9806D-20
TOTALS	1884	4267	1126037	418811679	

Table 2.16.5

THE EXTENDED BEALE TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE F-R METHOD WITH NEW RESTART

NV	NI	NF	NC	CPU	FV
2	23	50	150	101627	0.3556D-20
20	20	48	1008	313551	0.7417D-20
40	25	60	2460	748616	0.3266D-21
60	28	66	4026	1240486	0.1926D-20
80	25	61	4941	1500253	0.3931D-20
100	21	54	5454	1637727	0.2421D-18
120	23	60	7260	2156672	0.3015D-23
140	25	65	9165	2748378	0.2773D-20
160	25	64	10304	3081466	0.1277D-20
180	25	66	11946	3565079	0.1015D-21
200	25	66	13266	3924034	0.9182D-21
220	29	73	16133	4885887	0.2167D-18
240	34	83	20003	6045668	0.1813D-18
260	19	54	14094	4120098	0.6875D-25
280	22	59	16579	4909394	0.9621D-20
300	22	60	18060	5293629	0.1734D-19
320	25	66	21186	6266422	0.1378D-21
340	25	66	22506	6695139	0.4047D-20
360	26	68	24548	7260781	0.3630D-20
380	25	67	25527	7505612	0.4510D-20
400	25	67	26867	7911385	0.4066D-20
420	32	84	35364	10469001	0.4250D-21
440	30	79	34839	10313270	0.1444D-18
460	30	79	36419	10762470	0.1333D-18
480	30	79	37999	11184573	0.1668D-18
500	30	79	39579	11667494	0.2474D-18
TOTALS	669	1723	459683	136308712	

Table 2.16.6

HE EXTENDED ENGWALL TEST FUNCTION
ITH ANALYTICAL GRADIENT
HE F-R METHOD WITH NEW RESTART

NV	NI	NF	NC	CPU	FV
2	13	31	93	74449	0.0000D+00
20	8	24	504	132932	0.0000D+00
40	13	35	1435	376967	0.0000D+00
60	12	35	2135	564681	0.0000D+00
80	9	27	2187	564881	0.0000D+00
100	11	32	3232	826096	0.0000D+00
120	12	35	4235	1094976	0.0000D+00
140	14	39	5499	1415398	0.0000D+00
160	13	38	6118	1545262	0.0000D+00
180	15	42	7602	1979157	0.0000D+00
200	13	38	7638	1986182	0.0000D+00
220	13	40	8840	2274040	0.0000D+00
240	12	38	9158	2310870	0.0000D+00
260	13	40	10440	2662934	0.0000D+00
280	12	36	10116	2578123	0.0000D+00
300	10	31	9331	2382356	0.0000D+00
320	9	30	9630	2448083	0.0000D+00
340	10	34	11594	2808923	0.0000D+00
360	12	38	13718	3465611	0.0000D+00
380	12	37	14097	3552844	0.0000D+00
400	11	35	14035	3488687	0.0000D+00
420	11	35	14735	3647893	0.0000D+00
440	12	37	16317	4116959	0.0000D+00
460	13	39	17979	4623185	0.0000D+00
480	12	38	18278	4581621	0.0000D+00
500	13	41	20541	5208613	0.0000D+00
TOTALS	308	925	239487	60711723	

Table 2-16-1

THE EXTENDED ROSENBROCK TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE P-R METHOD WITH NEW RESTART

NV	NI	NF	NC	CPU	FV
2	26	57	171	68682	0.1043D-30
20	23	54	1134	280400	0.1424D-30
40	24	57	2337	554397	0.1494D-18
60	25	61	3721	869843	0.3412D-26
80	23	63	5103	1136343	0.4820D-22
100	23	66	6666	1454257	0.5548D-29
120	22	64	7744	1662927	0.6365D-22
140	23	68	9588	2044319	0.6297D-31
160	25	70	11270	2448673	0.4561D-31
180	25	71	12851	2772400	0.1060D-31
200	24	70	14070	3009316	0.1116D-21
220	23	68	15028	3200583	0.2012D-32
240	21	65	15665	3288611	0.4365D-23
260	26	76	19836	4235210	0.3455D-29
280	26	75	21075	4533397	0.3030D-29
300	23	72	21672	4535737	0.2600D-19
320	24	76	24396	5092314	0.8729D-21
340	25	79	26939	5609966	0.2165D-29
360	26	83	29963	6225555	0.6602D-31
380	26	83	31623	6570796	0.9609D-31
400	26	83	33283	6857715	0.3464D-30
420	24	78	32838	6763359	0.1943D-25
440	23	76	33516	6895782	0.0000D+00
460	25	80	36880	7658557	0.2503D-28
480	25	81	38961	8046169	0.1994D-29
500	25	82	41082	8425981	0.1696D-30
TOTALS	631	1858	497412	104241289	

Table 2-17-1

HE EXTENDED WOOD TEST FUNCTION
ITH ANALYTICAL GRADIENT
HE P-R METHOD WITH NEW RESTART

NV	NI	NF	NC	CPU	FV
4	57	119	595	203296	0.2186D-21
20	67	143	3003	826948	0.1306D-19
40	155	324	13284	3567394	0.3238D-21
60	58	130	7930	2038803	0.2943D-20
80	63	142	11502	2977398	0.1742D-21
100	46	108	10908	2747154	0.1043D-20
120	79	183	22143	5596669	0.1117D-18
140	139	313	44133	11244443	0.1125D-22
160	131	298	47978	12211660	0.2016D-25
180	55	130	23530	5938181	0.1879D-19
200	54	128	25728	6445580	0.5685D-21
220	46	117	25857	6317362	0.7164D-26
240	53	129	31089	7692956	0.6495D-21
260	50	123	32103	7933487	0.2339D-23
280	46	117	32877	7997504	0.2939D-20
300	88	214	64414	15768983	0.1424D-20
320	48	127	40767	9703723	0.1330D-21
340	38	106	36146	8448752	0.5067D-20
360	57	146	52706	12767005	0.3419D-25
380	37	100	38100	9055195	0.8748D-20
400	47	119	47719	11610018	0.3561D-20
420	50	126	53046	12930762	0.9107D-20
440	51	129	56889	13872197	0.7213D-22
460	50	128	59008	14317791	0.1710D-21
480	58	145	69745	17059746	0.1088D-21
500	56	144	72144	17423368	0.1120D-18
TOTALS	1679	3988	923344	226696375	

Table 2-17-2

THE EXTENDED MIELE & CANTRELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE P-R METHOD WITH NEW RESTART

NV	NI	NF	NC	CPU	FV
4	31	87	435	191349	0.1272D-12
20	54	134	2814	1170855	0.8940D-12
40	61	151	6191	2544063	0.5241D-11
60	83	219	13359	5373328	0.3780D-12
80	81	180	14580	6173773	0.7622D-11
100	76	184	18584	7658435	0.1125D-10
120	45	136	16456	6467282	0.1189D-10
140	83	217	30597	12458829	0.4270D-11
160	66	172	27692	11266719	0.9130D-11
180	81	199	36019	14892539	0.4118D-11
200	61	173	34773	13955561	0.2432D-11
220	59	148	32708	13399856	0.9300D-11
240	82	204	49164	20228528	0.1766D-11
260	107	297	77517	31117142	0.1030D-11
280	66	189	53109	21230279	0.4382D-11
300	90	242	72842	29263781	0.1805D-10
320	74	204	65484	26125145	0.8342D-11
340	47	125	42625	17117921	0.3766D-11
360	47	141	50901	20077962	0.9699D-11
380	77	214	81534	32797087	0.6571D-11
400	77	197	78997	32228154	0.2570D-11
420	71	195	82095	32804096	0.2768D-11
440	81	222	97902	39072092	0.1235D-10
460	70	197	90817	35943906	0.2466D-11
480	85	231	111111	44470191	0.7776D-11
500	83	228	114228	45449488	0.9658D-11
TOTALS	1838	4886	1302534	523478361	

Table 8-17-3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE P-R METHOD WITH NEW RESTART

NV	NI	NF	NC	CPU	FV
4	71	150	750	241441	0.1480D-15
20	101	215	4515	1173324	0.9687D-13
40	108	234	9594	2373182	0.5934D-14
60	81	182	11102	2708595	0.1890D-13
80	109	236	19116	4765645	0.1679D-14
100	86	193	19493	4737751	0.4283D-13
120	58	142	17182	4001814	0.1919D-13
140	124	270	38070	9342770	0.2228D-13
160	155	332	53452	13167019	0.9394D-14
180	91	208	37648	8911580	0.1499D-12
200	138	302	60702	14772213	0.1762D-13
220	132	290	64090	15639834	0.2001D-13
240	86	206	49646	11649879	0.2207D-13
260	93	216	56376	13373339	0.1927D-14
280	103	242	68002	16076452	0.7342D-13
300	113	261	78561	18746961	0.3067D-14
320	120	267	85707	20740832	0.1738D-13
340	89	211	71951	16866065	0.1137D-12
360	163	354	127794	31367587	0.2224D-13
380	109	248	94488	22589779	0.9357D-13
400	72	178	71378	16526537	0.4922D-13
420	54	144	60624	13655581	0.1957D-13
440	91	217	95697	22631902	0.3762D-20
460	118	266	122626	29625098	0.1888D-13
480	102	239	114959	27191577	0.2010D-14
500	115	262	131262	31589966	0.3625D-14
TOTALS	2682	6065	1564785	374466723	

Table 2. 17-4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE P-R METHOD WITH NEW RESTART

NV	NI	NF	NC	CPU	FV
10	106	230	2530	1074122	0.2449D-19
20	40	87	1827	763126	0.1656D-19
40	86	183	7503	3106334	0.1978D-20
60	45	102	6222	2506719	0.1409D-19
80	39	89	7209	2914699	0.1815D-19
100	63	141	14241	5704031	0.7249D-20
120	87	188	22748	9191030	0.1146D-20
140	86	186	26226	10568575	0.2854D-20
160	69	156	25116	10038044	0.7931D-21
180	61	141	25521	10172309	0.6038D-20
200	60	142	28542	11284348	0.7658D-20
220	44	107	23647	9244179	0.2065D-26
240	45	111	26751	10413521	0.2305D-20
260	56	132	34452	13497106	0.1576D-22
280	40	98	27538	10757646	0.1065D-19
300	39	97	29197	11386962	0.9820D-20
320	43	106	34026	13219222	0.3922D-19
340	43	107	36487	14188293	0.8724D-20
360	43	112	40432	15461950	0.1055D-24
380	57	138	52578	20528658	0.2215D-20
400	65	152	60952	24063858	0.1061D-19
420	70	165	69465	27337050	0.2114D-20
440	88	203	89523	35359377	0.1720D-19
460	94	216	99576	39409383	0.7149D-20
480	92	212	101972	40238282	0.1051D-21
500	77	180	90180	35480846	0.1237D-19
TOTALS	1638	3781	984461	387909670	

Table 2-17.5

HE EXTENDED BEALE TEST FUNCTION
ITH ANALYTICAL GRADIENT
HE P-R METHOD WITH NEW RESTART

NV	NI	NF	NC	CPU	FV
2	10	26	78	77828	0.7679D-23
20	12	35	735	222032	0.2815D-22
40	12	39	1599	466806	0.8703D-21
60	12	39	2379	684827	0.1626D-21
80	12	39	3159	915639	0.1126D-21
100	10	35	3535	1007372	0.2614D-18
120	12	42	5082	1434219	0.9970D-26
140	12	42	5922	1671092	0.2327D-21
160	13	45	7245	2059616	0.1735D-23
180	13	43	7783	2226588	0.6291D-23
200	13	43	8643	2463389	0.4978D-23
220	13	43	9503	2707983	0.2198D-23
240	12	43	10363	2920852	0.7483D-21
260	12	43	11223	3176969	0.3685D-21
280	12	43	12083	3421067	0.3239D-21
300	12	44	13244	3731154	0.4051D-21
320	12	44	14124	3978504	0.5881D-21
340	14	48	16368	4722946	0.2499D-35
360	12	47	16967	4721097	0.1329D-29
380	11	43	16383	4550397	0.4690D-30
400	11	45	18045	4995917	0.2125D-31
420	9	39	16419	4497944	0.1864D-21
440	11	44	19404	5406095	0.6486D-25
460	12	46	21206	5955792	0.1278D-29
480	12	46	22126	6205454	0.5501D-25
500	12	47	23547	6558591	0.5944D-23
TOTALS	308	1093	287165	80780170	

Table 8-17.6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE P-R METHOD WITH NEW RESTART

NV	NI	NF	NC	CPU	FV
2	7	16	48	60960	0.0000D+00
20	7	21	441	121846	0.0000D+00
40	8	23	943	256068	0.0000D+00
60	8	25	1525	394753	0.0000D+00
80	7	23	1863	481288	0.0000D+00
100	8	25	2525	652322	0.0000D+00
120	8	25	3025	773273	0.0000D+00
140	9	28	3948	1025086	0.0000D+00
160	9	28	4508	1167460	0.0000D+00
180	9	28	5068	1304471	0.0000D+00
200	8	26	5226	1314972	0.0000D+00
220	8	28	6188	1526783	0.0000D+00
240	8	28	6748	1667843	0.0000D+00
260	8	29	7569	1847875	0.0000D+00
280	6	22	6182	1502749	0.0000D+00
300	7	27	8127	1973608	0.0000D+00
320	7	27	8667	2126085	0.0000D+00
340	7	27	9207	2267396	0.0000D+00
360	7	26	9386	2309696	0.0000D+00
380	8	28	10668	2626634	0.0000D+00
400	8	28	11228	2775666	0.0000D+00
420	8	28	11788	2905039	0.0000D+00
440	8	28	12348	3045976	0.0000D+00
460	8	28	12908	3184241	0.0000D+00
480	8	28	13468	3323116	0.0000D+00
500	8	29	14529	3569188	0.0000D+00
TOTALS	202	679	178131	44204394	

Table 2. 17. 7

THE EXTENDED ROSENBROCK TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE QUASI-NEWTON METHOD

NV	NI	NF	NC	CPU	FV
2	32	115	181	73605823	0.5137D-21
20	103	389	2469	248102379	0.6870D-20
40	155	622	6862	401862731	0.2112D-14
60	201	801	12921	549334636	0.3080D-14
80	264	1028	22228	750982310	0.4989D-15
100	327	1298	34098	1024481741	0.4952D-15
120	348	1388	43268	1203191345	0.1513D-14
140	387	1501	55821	1418788328	0.1177D-14
160	422	1646	68326	1706613499	0.1955D-14
TOTALS	2239	8788	246174	7376962792	

Table 2.18.1

THE EXTENDED WOOD TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE QUASI-NEWTON METHOD

NV	NI	NF	NC	CPU	FV
4	65	223	487	142905238	0.3688D-19
20	232	713	5373	473493568	0.2547D-18
40	281	891	12171	608462018	0.8937D-19
60	357	1184	22664	853338688	0.2186D-14
80	390	1326	32606	1028179255	0.1206D-14
100	442	1556	45856	1288371349	0.2178D-15
120	475	1721	58841	1539596973	0.2029D-14
140	530	1964	76304	1921347142	0.1527D-15
160	566	2089	92809	2244049569	0.6184D-14
TOTALS	3338	11667	347111	10099643800	

Table 2.18.2

THE EXTENDED MIELE & CANTRELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE QUASI-NEWTON METHOD

NV	NI	NF	NC	CPU	FV
4	50	209	413	129443510	0.1894D-14
20	51	205	1245	128906826	0.2378D-14
40	53	210	2370	137141594	0.8241D-15
60	53	212	3452	146639927	0.1272D-14
80	53	212	4532	156951007	0.1696D-14
100	54	211	5711	171018890	0.4929D-15
120	54	211	6811	187195078	0.5914D-15
140	54	211	7911	206131601	0.6900D-15
160	54	211	9011	228460451	0.7886D-15
TOTALS	476	1892	41456	1491888884	

Table 2.18.3

THE EXTENDED POWELL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE QUASI-NEWTON METHOD

NV	NI	NF	NC	CPU	FV
4	43	145	321	91834950	0.4056D-22
20	96	327	2267	206464260	0.4885D-16
40	194	626	8426	415855182	0.5712D-16
60	163	537	10377	376675510	0.1411D-16
80	228	728	19048	557425474	0.5038D-16
100	275	887	28487	748803277	0.3792D-16
120	313	993	38673	933175211	0.1124D-16
140	306	990	43970	1017842159	0.5427D-17
160	351	1147	57467	1302794620	0.3750D-17
TOTALS	1969	6380	209036	5650870643	

Table 2.18.4

THE EXTENDED DIXON TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE QUASI-NEWTON METHOD

NV	NI	NF	NC	CPU	FV
10	59	188	788	125318871	0.1098D-19
20	70	207	1627	141458665	0.1319D-17
40	84	251	3651	179060288	0.5820D-18
60	103	321	6561	238318962	0.3260D-17
80	111	347	9307	277049745	0.2300D-15
100	126	373	13073	337468826	0.3408D-16
120	122	364	15124	363140671	0.3324D-16
140	129	387	18587	430738994	0.6572D-16
160	119	364	19564	448614713	0.9053D-16
TOTALS	923	2802	88282	2551169735	

Table 2.18.5

THE EXTENDED BEALE TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE QUASI-NEWTON METHOD

NV	NI	NF	NC	CPU	FV
2	14	43	73	29070927	0.3557D-21
20	17	49	409	34320780	0.5578D-20
40	15	49	689	34712618	0.6282D-20
60	15	48	1008	36503359	0.7931D-19
80	15	49	1329	39313600	0.7044D-19
100	15	49	1649	43180497	0.1074D-18
120	15	48	1968	46844676	0.8426D-19
140	15	49	2289	52751398	0.1071D-18
160	15	48	2608	57856232	0.1275D-18
TOTALS	136	432	12022	374554087	

Table 2.18.6

THE EXTENDED ENGWALL TEST FUNCTION
WITH ANALYTICAL GRADIENT
THE QUASI-NEWTON METHOD

NV	NI	NF	NC	CPU	FV
2	11	36	60	24062347	0.0000D+00
20	11	35	275	24057143	0.0000D+00
40	11	35	515	24902444	0.0000D+00
60	10	33	693	24916928	0.0000D+00
80	10	33	913	26885750	0.0000D+00
100	10	33	1133	29200541	0.0000D+00
120	10	34	1354	32517389	0.0000D+00
140	10	33	1573	35542945	0.0000D+00
160	10	33	1793	39406506	0.0000D+00
TOTALS	93	305	8309	261492093	

Table 2.18.7

Appendix 3

Summarized and Comparison Results

This appendix contains the tables of the summarized results and the comparison of the new methods with the existing ones. Tables (3-i) $i = 1,2,\ldots,17$ contain the totals from the bottom rows of Tables (2-i-j) $i = 1,\ldots,17$, $j = 1,\ldots,7$ of Appendix 2, followed by the ratios of these totals for the particular method over those of the Fletcher-Reeves method and over those of the Polak-Ribière method. Table (3-18) contains the totals for the F.R method on problems up to 160 variables. Table (3-19) shows the same totals for the P.R method, followed by the ratios comparing the P.R method to the F.R method. Table (3-20) contains the bottom rows of Tables (2-i-j) $i = 18$ and $j = 1,\ldots,7$ from Appendix 2, followed by the ratios comparing the Q.N method with both the F.R and the P.R methods. Table (3-21) shows the totals for Hybrid 3 on problems up to 160 variables, followed by the ratios comparing Hybrid 3 to the F.R, the P.R and the Q.N methods. Finally Table (3-22) shows the results obtained by not considering the Rosenbrock test problem. The same notation is used as in Appendix 2.

The F.R method : (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	2100	5069	1377517	283383768
2	10019	20707	6553775	1558417294
3	2261	5684	1469368	564253150
4	12294	25203	7971021	1757258197
5	7885	16196	5137696	1957732708
6	2206	4783	1390371	430159650
7	303	913	239451	60723429
TOTALS	37068	78561	24139199	6611928196

TABLE 3-1

The P.R method : (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	582	1792	470272	95337378
P.R/F.R	0.28	0.35	0.34	0.34
2	4638	9855	3005111	782991814
P.R/F.R	0.46	0.48	0.46	0.50
3	1728	4562	1144986	453124472
P.R/F.R	0.76	0.80	0.78	0.80
4	10585	21987	7065955	1756784408
P.R/F.R	0.86	0.87	0.89	1.00
5	1782	4098	1067718	419791649
P.R/F.R	0.23	0.25	0.21	0.21
6	274	995	259819	72508490
P.R/F.R	0.12	0.21	0.19	0.17
7	192	669	174905	43394032
P.R/F.R	0.63	0.73	0.73	0.71
TOTALS	19781	43958	13188766	3623932243
P.R/F.R	0.53	0.56	0.55	0.55

TABLE 3-2

The CG/SD Powell's Hybrid Method (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	574	1737	462871	88390845
PH/F.R	0.27	0.34	0.34	0.31
PH/P.R	0.99	0.97	0.98	0.93
2	4047	8591	2498347	591329853
PH/F.R	0.40	0.41	0.38	0.38
PH/P.R	0.87	0.87	0.83	0.76
3	1766	4737	1212325	45638981
PH/F.R	0.78	0.83	0.83	0.80
PH/P.R	1.02	1.04	1.06	1.00
4	9044	18763	5988931	1335186320
PH/F.R	0.74	0.74	0.75	0.76
PH/P.R	0.85	0.85	0.85	0.76
5	1765	4063	1056423	383749725
PH/F.R	0.22	0.25	0.21	0.20
PH/P.R	0.99	0.99	0.99	0.91
6	285	1030	267938	75068049
PH/F.R	0.13	0.22	0.19	0.17
PH/P.R	1.04	1.04	1.03	1.04
7	192	670	175186	43349028
PH/F.R	0.63	0.73	0.73	0.71
PH/P.R	1.00	1.00	1.00	1.00
TOTALS	17673	39591	11662021	2969716801
PH/F.R	0.48	0.50	0.48	0.45
PH/P.R	0.89	0.90	0.88	0.82

TABLE 3-3

ORIG1 Hybrid Method : (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	539	1659	436261	84216900
01/F.R	0.26	0.33	0.32	0.30
01/P.R	0.93	0.93	0.93	0.88
2	3965	8424	2406140	581635084
01/F.R	0.40	0.41	0.37	0.37
01/P.R	0.85	0.85	0.80	0.74
3	1829	4848	1189848	454661064
01/F.R	0.81	0.85	0.81	0.81
01/P.R	1.06	1.06	1.04	1.00
4	10443	21632	6872240	1566367920
01/F.R	0.85	0.86	0.86	0.89
01/P.R	0.99	0.98	0.97	0.89
5	1778	4087	1059267	396133713
01/F.R	0.23	0.25	0.21	0.20
01/P.R	1.00	1.00	0.99	0.94
6	299	1053	273757	76820056
01/F.R	0.14	0.22	0.20	0.18
01/P.R	1.09	1.06	1.05	1.06
7	193	671	175467	43486842
01/F.R	0.64	0.73	0.73	0.72
01/P.R	1.00	1.00	1.00	1.00
TOTALS	19046	42374	12412980	3203321579
01/F.R	0.51	0.54	0.51	0.48
01/P.R	0.96	0.96	0.94	0.88

TABLE 3-4

ORIG2 Hybrid Method : (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	550	1675	444891	89191050
02/F.R	0.26	0.33	0.32	0.31
02/P.R	0.95	0.93	0.95	0.94
2	3883	8258	2356322	595923133
02/F.R	0.39	0.40	0.36	0.34
02/P.R	0.84	0.84	0.78	0.76
3	1169	3099	792603	303758512
02/F.R	0.52	0.55	0.54	0.54
02/P.R	0.68	0.68	0.69	0.67
4	9988	20782	6563174	1570525910
02/F.R	0.81	0.82	0.82	0.89
02/P.R	0.94	0.95	0.93	0.89
5	1715	3954	1047304	395829521
02/F.R	0.22	0.24	0.20	0.20
02/P.R	0.96	0.97	0.98	0.94
6	300	1058	279562	80287571
02/F.R	0.14	0.22	0.20	0.19
02/P.R	1.09	1.06	1.08	1.11
7	192	665	174363	44546626
02/F.R	0.63	0.73	0.73	0.73
02/P.R	1.00	0.99	1.00	1.03
TOTALS	17797	39491	11658219	3081062323
02/F.R	0.48	0.50	0.48	0.47
02/P.R	0.90	0.90	0.88	0.85

TABLE 3-5

Hybrid 1 Method : (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	3219	6965	2006513	484710481
H1/F.R	1.53	1.37	1.46	1.71
H1/P.R	5.53	3.89	4.27	5.08
2	3565	7642	2198330	569112686
H1/F.R	0.36	0.37	0.34	0.37
H1/P.R	0.77	0.78	0.73	0.73
3	2131	5541	1486225	593914414
H1/F.R	0.94	0.97	1.01	1.05
H1/P.R	1.23	1.21	1.30	1.31
4	11862	24296	7660400	1899552115
H1/F.R	0.96	0.96	0.96	1.08
H1/P.R	1.12	1.11	1.08	1.08
5	1703	3941	1047751	415044030
H1/F.R	0.22	0.24	0.20	0.21
H1/P.R	0.96	0.96	0.98	0.99
6	3136	6678	1909104	610155789
H1/F.R	1.40	1.40	1.37	1.42
H1/P.R	11.44	6.70	7.35	8.42
7	212	731	193267	47768373
H1/F.R	0.70	0.80	0.81	0.79
H1/P.R	1.09	1.09	1.10	1.10
TOTALS	25828	55794	16501590	4620257888
H1/F.R	0.70	0.71	0.68	0.70
H1/P.R	1.31	1.27	1.25	1.27

TABLE 3-6

Shanno's Hybrid Method : (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	618	1823	480531	107787733
SH/F.R	0.29	0.36	0.35	0.38
SH/P.R	1.06	1.02	1.02	1.13
2	4660	9902	3021698	878751548
SH/F.R	0.47	0.48	0.46	0.56
SH/P.R	1.00	1.00	1.01	1.12
3	1843	4893	1243553	515689218
SH/F.R	0.82	0.86	0.85	0.91
SH/P.R	1.07	1.07	1.09	1.14
4	7905	16458	5219678	1435694342
SH/F.R	0.64	0.65	0.65	0.82
SH/P.R	0.74	0.75	0.74	0.82
5	1776	4082	1066912	437609669
SH/F.R	0.23	0.25	0.21	0.22
SH/P.R	1.00	1.00	1.00	1.04
6	274	995	259819	79074500
SH/F.R	0.12	0.21	0.19	0.18
SH/P.R	1.00	1.00	1.00	1.09
7	192	669	174905	48028148
SH/F.R	0.63	0.73	0.73	0.79
SH/P.R	1.00	1.00	1.00	1.11
TOTALS	17268	38822	11467096	3502637158
SH/F.R	0.47	0.49	0.48	0.53
SH/P.R	0.87	0.88	0.87	0.97

TABLE 3-7

The Angle Test Hybrid Method ($\tau = 0.1$) : (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	552	1676	438064	104333081
ATH1/F.R	0.26	0.33	0.32	0.37
ATH1/P.R	0.95	0.94	0.93	1.09
2	4062	8614	2503110	776624782
ATH1/F.R	0.41	0.42	0.38	0.50
ATH1/P.R	0.88	0.87	0.83	0.99
3	1706	4538	1107438	488072262
ATH1/F.R	0.75	0.80	0.75	0.86
ATH1/P.R	0.99	0.99	0.97	1.08
4	11488	23603	7420111	2281919642
ATH1/F.R	0.93	0.94	0.93	1.30
ATH1/P.R	1.09	1.07	1.05	1.30
5	1771	4067	1059707	460047880
ATH1/F.R	0.22	0.25	0.21	0.23
ATH1/P.R	0.99	0.99	0.99	1.10
6	274	995	259819	79111282
ATH1/F.R	0.12	0.21	0.19	0.18
ATH1/P.R	1.00	1.00	1.00	1.09
7	192	669	174905	48037114
ATH1/F.R	0.63	0.73	0.73	0.79
ATH1/P.R	1.00	1.00	1.00	1.11
TOTALS	20045	44162	12963154	4238146043
ATH1/F.R	0.54	0.56	0.54	0.64
ATH1/P.R	1.01	1.00	0.98	1.17

TABLE 3-8

The Angle Test Hybrid Method (τ = 0.01): (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	546	1655	434323	102934447
ATH2/F.R	0.26	0.33	0.32	0.36
ATH2/P.R	0.94	0.92	0.92	1.08
2	3965	8420	2405916	740735869
ATH2/F.R	0.40	0.41	0.37	0.48
ATH2/P.R	0.85	0.85	0.80	0.95
3	1733	4561	1138713	499620686
ATH2/F.R	0.77	0.80	0.77	0.89
ATH2/P.R	1.00	1.00	0.99	1.10
4	10780	22210	6805618	2081807963
ATH2/F.R	0.88	0.88	0.85	1.18
ATH2/P.R	1.02	1.01	0.96	1.19
5	1759	4045	1058805	458020782
ATH2/F.R	0.22	0.25	0.21	0.23
ATH2/P.R	0.99	0.99	0.99	1.09
6	274	995	259819	79111282
ATH2/F.R	0.12	0.21	0.19	0.18
ATH2/P.R	1.00	1.00	1.00	1.09
7	192	669	174905	48037114
ATH2/F.R	0.63	0.73	0.73	0.79
ATH2/P.R	1.00	1.00	1.00	1.11
TOTALS	19249	42555	12278099	4010268143
ATH2/F.R	0.52	0.54	0.51	0.60
ATH2/P.R	0.97	0.97	0.93	1.11

TABLE 3-9

The Angle Test Hybrid Method ($\tau = 0.001$): (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	539	1659	435887	102193957
ATH3/F.R	0.26	0.33	0.32	0.36
ATH3/P.R	0.93	0.93	0.93	1.07
2	3965	8420	2405916	738101299
ATH3/F.R	0.40	0.41	0.37	0.47
ATH3/P.R	0.85	0.85	0.80	0.94
3	1568	4062	1025146	451255665
ATH3/F.R	0.69	0.71	0.70	0.80
ATH3/P.R	0.91	0.89	0.90	1.00
4	10857	22373	7079181	2152919232
ATH3/F.R	0.88	0.89	0.89	1.23
ATH3/P.R	1.03	1.02	1.00	1.23
5	1757	4045	1058805	460846781
ATH3/F.R	0.22	0.25	0.21	0.24
ATH3/P.R	0.99	0.99	0.99	1.10
6	274	995	259819	79111289
ATH3/F.R	0.12	0.21	0.19	0.18
ATH3/P.R	1.00	1.00	1.00	1.09
7	192	669	174905	48037114
ATH3/F.R	0.63	0.73	0.73	0.79
ATH3/P.R	1.00	1.00	1.00	1.11
TOTALS	19154	42223	12439659	4032465330
ATH3/F.R	0.52	0.54	0.52	0.61
ATH3/P.R	0.97	0.96	0.94	1.11

TABLE 3-10

The Beta Test Hybrid Method : (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	542	1652	434794	90759450
BTH/F.R	0.26	0.33	0.32	0.32
BTH/P.R	0.93	0.92	0.92	0.95
2	3965	8420	2405916	639799254
BTH/F.R	0.40	0.41	0.37	0.41
BTH/P.R	0.85	0.85	0.80	0.82
3	1636	4255	1063419	430384868
BTH/F.R	0.72	0.75	0.72	0.75
BTH/P.R	0.95	0.93	0.93	0.95
4	10745	22242	7145370	1778997404
BTH/F.R	0.87	0.88	0.90	1.01
BTH/P.R	1.02	1.01	1.01	1.01
5	1759	4045	1058805	421122666
BTH/F.R	0.22	0.25	0.21	0.22
BTH/P.R	0.99	0.99	0.99	1.00
6	387	1248	330872	94824442
BTH/F.R	0.18	0.26	0.24	0.22
BTH/P.R	1.41	1.25	1.27	1.31
7	208	691	179987	44984993
BTH/F.R	0.69	0.76	0.75	0.74
BTH/P.R	1.08	1.03	1.03	1.04
TOTALS	19242	42553	12619163	3500878077
BTH/F.R	0.52	0.54	0.52	0.53
BTH/P.R	0.97	0.97	0.96	0.97

TABLE 3-11

Hybrid 2 Method ($\tau = 10^{-4}$) : (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	542	1655	434807	101124740
H21/F.R	0.26	0.33	0.32	0.36
H21/P.R	0.93	0.92	0.92	1.06
2	3965	8420	2405916	726262278
H21/F.R	0.40	0.41	0.37	0.47
H21/P.R	0.85	0.85	0.80	0.93
3	1614	4198	1051302	456815623
H21/F.R	0.71	0.74	0.72	0.81
H21/P.R	0.93	0.92	0.92	1.01
4	10801	22334	7155702	2117072306
H21/F.R	0.88	0.89	0.90	1.20
H21/P.R	1.02	1.02	1.01	1.21
5	1759	4045	1058805	453262254
H21/F.R	0.22	0.25	0.21	0.23
H21/P.R	0.99	0.99	0.99	1.08
6	330	1111	291255	86568927
H21/F.R	0.15	0.23	0.21	0.20
H21/P.R	1.20	1.12	1.12	1.19
7	205	683	178539	47696065
H21/F.R	0.68	0.75	0.75	0.79
H21/P.R	1.07	1.02	1.02	1.10
TOTALS	19216	42446	12576326	3988802193
H21/F.R	0.52	0.54	0.52	0.60
H21/P.R	0.97	0.97	0.95	1.10

TABLE 3-12

Hybrid 2 Method ($\tau = 10^{-6}$) : (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	542	1652	434794	99437377
H22/F.R	0.26	0.33	0.32	0.35
H22/P.R	0.93	0.92	0.92	1.04
2	3965	8420	2405916	721538028
H22/F.R	0.40	0.41	0.37	0.46
H22/P.R	0.85	0.85	0.80	0.92
3	1639	4255	1063259	458599483
H22/F.R	0.72	0.75	0.72	0.81
H22/P.R	0.95	0.93	0.93	1.01
4	10745	22242	7145370	2105383609
H22/F.R	0.87	0.88	0.90	1.20
H22/P.R	1.02	1.01	1.01	1.20
5	1759	4045	1058805	452882035
H22/F.R	0.22	0.25	0.21	0.23
H22/P.R	0.99	0.99	0.99	1.08
6	330	1111	291255	86568927
H22/F.R	0.15	0.23	0.21	0.20
H22/P.R	1.20	1.12	1.12	1.19
7	205	683	178539	47696065
H22/F.R	0.68	0.75	0.75	0.79
H22/P.R	1.07	1.02	1.02	1.10
TOTALS	19185	42408	12577938	3972105524
H22/F.R	0.52	0.54	0.52	0.60
H22/P.R	0.97	0.96	0.95	1.10

TABLE 3-13

Hybrid 2 Method ($\tau = 10^{-8}$) : (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	595	1751	460493	105394595
H23/F.R	0.28	0.35	0.33	0.37
H23/P.R	1.02	0.98	0.98	1.11
2	4275	9064	2636332	794187783
H23/F.R	0.43	0.44	0.40	0.51
H23/P.R	0.92	0.92	0.88	1.01
3	1718	4476	1126800	482639557
H23/F.R	0.76	0.79	0.77	0.86
H23/P.R	0.99	0.98	0.98	1.07
4	10401	21468	6511836	1943447172
H23/F.R	0.85	0.85	0.82	1.11
H23/P.R	0.98	0.98	0.92	1.11
5	1759	4045	1058805	448003674
H23/F.R	0.22	0.25	0.21	0.23
H23/P.R	0.99	0.99	0.99	1.07
6	330	1111	291255	86568927
H23/F.R	0.15	0.23	0.21	0.20
H23/P.R	1.20	1.12	1.12	1.19
7	205	683	178539	47696065
H23/F.R	0.68	0.75	0.75	0.79
H23/P.R	1.07	1.02	1.02	1.10
TOTALS	19283	42598	12264060	3907937773
H23/F.R	0.52	0.54	0.51	0.59
H23/P.R	0.97	0.97	0.93	1.08

TABLE 3-14

Hybrid 3 Method : (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	668	1920	518202	109066477
H3/F.R	0.32	0.38	0.38	0.38
H3/P.R	1.15	1.08	1.10	1.14
2	1385	3350	858386	207017085
H3/F.R	0.14	0.16	0.13	0.13
H3/P.R	0.30	0.34	0.29	0.26
3	1681	4430	1146106	457623558
H3/F.R	0.74	0.78	0.78	0.81
H3/P.R	0.97	0.97	1.00	1.01
4	3364	7424	1861160	447239410
H3/F.R	0.27	0.29	0.23	0.25
H3/P.R	0.32	0.34	0.26	0.25
5	1622	3758	973588	381726997
H3/F.R	0.21	0.23	0.19	0.19
H3/P.R	0.91	0.92	0.91	0.91
6	328	1108	291246	82401484
H3/F.R	0.15	0.23	0.21	0.19
H3/P.R	1.20	1.11	1.12	1.14
7	204	681	178533	44726576
H3/F.R	0.67	0.75	0.75	0.74
H3/P.R	1.06	1.02	1.02	1.03
TOTALS	9252	22671	5827221	1629801587
H3/F.R	0.25	0.29	0.24	0.25
H3/P.R	0.47	0.52	0.44	0.45

TABLE 3-15

The F.R Method with New Restarts : (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	1184	3012	806208	164542475
NFR/F.R	0.56	0.59	0.59	0.58
NFR/P.R	2.03	1.68	1.71	1.73
2	1363	3299	876395	198413494
NFR/F.R	0.14	0.16	0.13	0.13
NFR/P.R	0.29	0.33	0.29	0.25
3	2133	5382	1384994	531230042
NFR/F.R	0.94	0.95	0.94	0.94
NFR/P.R	1.23	1.18	1.21	1.17
4	2964	6618	1653102	355781706
NFR/F.R	0.24	0.26	0.21	0.20
NFR/P.R	0.28	0.30	0.23	0.20
5	1884	4264	1126037	418811679
NFR/F.R	0.24	0.26	0.22	0.21
NFR/P.R	1.06	1.04	1.05	1.00
6	669	1723	459683	136308712
NFR/F.R	0.30	0.36	0.33	0.32
NFR/P.R	2.44	1.73	1.77	1.88
7	308	925	239487	60711723
NFR/F.R	1.02	1.01	1.00	1.00
NFR/P.R	1.60	1.38	1.37	1.40
TOTALS	10505	25223	6545906	1865799831
NFR/F.R	0.28	0.32	0.27	0.28
NFR/P.R	0.53	0.57	0.50	0.51

TABLE 3-16

The P.R Method with New Restarts : (Totals up to 500 variables)

PROBLEM	NI	NF	NC	CPU
1	631	1858	497412	104241289
NPR/F.R	0.30	0.37	0.36	0.37
NPR/P.R	1.08	1.04	1.06	1.09
2	1679	3988	923344	226696375
NPR/F.R	0.17	0.19	0.14	0.15
NPR/P.R	0.36	0.40	0.31	0.29
3	1838	4886	1302534	523478361
NPR/F.R	0.81	0.86	0.89	0.93
NPR/P.R	1.06	1.07	1.14	1.16
4	2682	6065	1564785	374466723
NPR/F.R	0.22	0.24	0.20	0.21
NPR/P.R	0.25	0.28	0.22	0.21
5	1638	3781	984461	387909670
NPR/F.R	0.21	0.23	0.19	0.20
NPR/P.R	0.92	0.92	0.92	0.92
6	308	1093	287165	80780170
NPR/F.R	0.14	0.23	0.21	0.19
NPR/P.R	1.12	1.10	1.11	1.11
7	202	679	178131	44204394
NPR/F.R	0.67	0.74	0.74	0.73
NPR/P.R	1.05	1.01	1.02	1.02
TOTALS	8978	22350	5737832	1741776982
NPR/F.R	0.24	0.28	0.24	0.26
NPR/P.R	0.45	0.51	0.44	0.48

TABLE 3-17

The F.R Method (Up to 160 variables)

PROBLEM	NI	NF	NC	CPU
1	695	1538	149106	31716628
2	1514	3145	331633	79119488
3	746	1828	151052	58753972
4	1940	4043	400575	88222272
5	1462	3038	313018	119910452
6	624	1348	143276	44475231
7	100	284	25402	6563228
TOTALS	7081	15224	1514062	428761271

TABLE 3-18

The P.R Method (Up to 160 variables)

PROBLEM	NI	NF	NC	CPU
1	216	564	46784	10122380
PR/FR	0.31	0.37	0.31	0.32
2	948	2061	208477	53656488
PR/FR	0.63	0.66	0.63	0.68
3	595	1529	120853	48416084
PR/FR	0.80	0.84	0.80	0.82
4	1537	3354	231322	78772437
PR/FR	0.79	0.83	0.58	0.89
5	691	1518	133298	53201193
PR/FR	0.47	0.50	0.43	0.44
6	97	315	26859	7653151
PR/FR	0.16	0.23	0.19	0.17
7	67	211	17927	4589470
PR/FR	0.67	0.74	0.71	0.70
TOTALS	4151	9552	785520	256411203
PR/FR	0.59	0.63	0.52	0.60

TABLE 3-19

The Q.N Method (Up to 160 variables)

PROBLEM	NI	NF	NC	CPU
1	2239	8788	246174	7376962792
QN/F.R	3.22	5.71	1.65	232.6
QN/P.R	10.36	15.50	5.26	728.8
2	3338	11667	347111	10099643800
QN/F.R	2.20	3.71	1.05	127.7
QN/P.R	3.52	5.66	1.66	188.2
3	476	1892	41456	1491888884
QN/F.R	0.64	1.04	0.27	25.39
QN/P.R	0.80	1.23	0.34	30.81
4	1969	6380	209036	5650870643
QN/F.R	1.01	1.58	0.52	64.05
QN/P.R	1.28	1.90	0.90	71.74
5	923	2802	88282	2551169735
QN/F.R	0.63	0.92	0.28	21.28
QN/P.R	1.34	1.85	0.66	47.95
6	136	432	12022	374554087
QN/F.R	0.22	0.32	0.08	8.42
QN/P.R	1.40	1.37	0.45	48.94
7	93	305	8309	261492093
QN/F.R	0.93	1.07	0.33	39.84
QN/P.R	1.39	1.45	0.46	56.98
TOTALS	9174	32266	952390	27806582034
QN/F.R	1.30	2.12	0.63	64.85
QN/P.R	2.21	3.38	1.21	108.45

TABLE 3-20

The H3 Method (Up to 160 variables)

PROBLEM	NI	NF	NC	CPU
1	226	577	47899	10606148
H3/F.R	0.33	0.38	0.32	0.33
H3/P.R	1.05	1.02	1.02	1.05
H3/Q.N	0.10	0.07	0.19	0.0014
2	495	1113	94569	23926432
H3/F.R	0.33	0.35	0.29	0.30
H3/P.R	0.50	0.54	0.45	0.45
H3/Q.N	0.15	0.10	0.27	0.0024
3	581	1517	130413	52263421
H3/F.R	0.78	0.83	0.86	0.89
H3/P.R	0.98	0.99	1.08	1.08
H3/Q.N	1.22	0.80	3.15	0.0092
4	1059	2285	205281	50625313
H3/F.R	0.55	0.57	0.51	0.57
H3/P.R	0.69	0.68	0.89	0.64
H3/Q.N	0.54	0.36	0.98	0.0090
5	616	1362	113812	45706730
H3/F.R	0.42	0.45	0.36	0.38
H3/P.R	0.89	0.90	0.85	0.86
H3/Q.N	0.67	0.49	1.29	0.0179
6	113	349	30007	8723507
H3/F.R	0.18	0.26	0.21	0.20
H3/P.R	1.16	1.11	1.12	1.14
H3/Q.N	0.83	0.81	2.50	0.0232
7	71	214	18626	4892295
H3/F.R	0.71	0.75	0.73	0.75
H3/P.R	1.06	1.01	1.04	1.07
H3/Q.N	0.76	0.70	2.24	0.0187
TOTALS	3161	7417	640607	196743846
H3/F.R	0.45	0.49	0.42	0.46
H3/P.R	0.76	0.78	0.82	0.77
H3/Q.N	0.34	0.23	0.67	0.0071

TABLE 3-21

The summarized results without the Rosenbrock problem (Up to 160 variables)

METHOD	NI	NF	NC	CPU
F.R	6386	13686	1364946	397044643
P.R	3935	8988	738736	246288823
P.R/F.R	0.67	0.67	0.54	0.62
Q.N	6935	23478	706216	20429619242
Q.N/F.R	1.09	1.72	0.52	51.45
Q.N/P.R	1.76	2.61	0.96	82.95
H3	2935	6840	592708	186137698
H3/F.R	0.46	0.50	0.43	0.47
H3/P.R	0.75	0.76	0.80	0.76
H3/Q.N	0.42	0.29	0.84	0.0091

TABLE 3-22

REFERENCES :

[1] Ablow, C.M. and Brigham, G. (1955): "An Analog Solution of Programming Problems." Operat. Res. 3, pp. 388-394.

[2] Al-Baali, M. (1985):"Descent Property and Global Convergence of the Fletcher_Reeves Method with Inexact Line Search." I.M.A. Journal of Numerical Analysis., 5, pp. 121-124.

[3] Al-Baali, M. and Fletcher, R. (1986): "An Efficient Line Search for Nonlinear Least-Squares." J.O.T.A., 48, No.3, pp. 359-377.

[4] Armijo, L. (1966): "Minimization of Functions Having Lipschitz-Continuous First Partial Derivatives." Pacific, J. Math., 16, pp. 1-3.

[5] Beale, E.M.L. (1972): "A Derivation of Conjugate Gradients" in F.A. Lootsma Ed., Numerical Methods for Nonlinear Optimization, Academic Press, London, pp. 39-43.

[6] Beltrami, E.J. (1969): "A Constructive Proof of the Kuhn-Tucker Multiplier Rule." J. Math. Anal. Appls., 26, pp. 297-306.

[7] Box, G.E.P. and Draper, N.R. (1969): "Evolutionary Operation: A Statistical Method for Process Improvement." Wiley, New York.

[8] Box, M.J. (1965): "A New Method of Constrained Optimization and a comparison with Other Methods." Comp. J., 8, pp. 42-52.

[9] Broyden, C.G. (1967): "Quasi-Newton Methods and their Application to Function Minimization." Math. Comp., 21, pp. 368-381.

[10] Broyden,C.G. (1970): "The Convergence of a Class of Double Rank Minimization Algorithms." Parts 1 and 2: "General Considerations and the New Algorithm." J.I.M.A., 6, pp.76-90, and pp. 222-231.

[11] Broyden, C.G., Dennis, J.E. and More, J.J. (1973): "On the Local and Superlinear Convergence of Quasi-Newton Methods." J.I.M.A., 12, pp.223-245.

[12] Buttler, T. and Martin, A.V. (1962): "On a Method of Courant for Minimizing Functions." J. Math. Phys., 41, pp. 291-299.

[13] Cauchy, A. (1847): "Methode Generale pour la Resolution des Systemes d'Equations Simultanees." C.R. Acd. Sci., 27, pp. 536-538.

[14] Courant, R. (1943): "Variational Methods for the Solution of Problems of Equilibrium and Vibration." Bull. Am. Math. Soc., 49, pp. 1-23.

[15] Crowder, H. and Wolfe, P. (1971): "Linear Convergence of the Conjugate Gradient Method." I.B.M. Journal of Research and Development, 16, pp. 431-433.

[16] Dahlquist, G. and Bjorck, A. (1974): "Numerical Methods." Englewood Cliffs, NJ: Prentice-Hall.

[17] Dambrauskas, A.P. (1970): "The Simplex Optimization Method with Variable Step." Engng. Cybern., 8, pp. 28-36.

[18] Dambrauskas, A.P. (1972): "Investigation of the Efficiency of the Simplex Method of Optimization with Variable Step in Noise Situation." Engng. Cybern., 10, pp. 590-599.

[19] Davidon, W.C. (1959): "Variable Metric Method for Minimization." Argonne Nat. Lab. Report A.N.L.-5990 Rev., Lemont., Ill, Nov. 1959.

[20] Davies, D. (1970): "Some Practical Methods of Optimization." in Integer and Nonlinear Programming (J. Abadie Ed.) pp. 87-117. North-Holland Publishing Company. Amsterdam.

[21] Dennis, J.E. and More,J.J. (1974): "A Characterization of Superlinear Convergence and its Application to Quasi-Newton Methods." Math. Comp., 28, pp. 549-560.

[22] Dennis, J.E. and Schnabel, R.B. (1983): "Numerical Methods for Unconstrained Optimization and Nonlinear Equations." Prentice Hall Inc. Englewood Cliffs. New Jersey. 07632.

[23] Dixon, L.C.W. (1972): "Conjugate Gradient Algorithms: Quadratic Termination Without Linear Searches." The Hatfield Polytechnic, Numerical Optimization Center T.R. 38.

[24] Dixon, L.C.W. (1975): "Conjugate Gradient Algorithms; Quadratic Termination Without Linear Searches." J.I.M.A., 15, pp. 9-18.

[25] Faure, P. and Huard, P. (1965) "Resolution des Programes Mathematiques a Fonctions Nonlineaires par la Methode du Gradient Reduit." Revue Fr. Rech. Oper., 9, pp. 167-205.

[26] Faure, P. and Huard, P. (1966): "Resultats Nouveaux Relatifs a la Methode des Centres." Quatrieme Conference de Recherche Operationnelle, Cambridge.

[27] Fiacco, A.V. (1967): "Sequential Unconstrained Minimization Methods for Nonlinear Programming." Thesis. Northwestern University, Evanston, Illinois.

[28] Fiacco, A.V. and Mc.Cormick, G.P. (1966): "Extensions of S.U.M.T. for Nonlinear Programming: Equality Constraints and Extrapolation." Man. Sci., 12, (11), pp. 816-829.

[29] Fiacco, A.V. and Mc.Cormick, G.P. (1967a): "The Slacked Unconstrained Minimization Techniques for Convex Programming." S.I.A.M. J. Appl. Math., 15, pp. 505-515.

[30] Fiacco, A.V. and Mc.Cormick, G.P. (1967b): "The Sequential Unconstrained Minimization Techniques Without Parameters." Operat. Res., 15, pp. 820-827.

[31] Fletcher, R. (1969): UKAEA Research Report TP 368.

[32] Fletcher, R. (1970): "A New Approach to Variable Metric Algorithms." Computer J., 13, pp. 317-322.

[33] Fletcher, R. (1980): "Practical Methods of Optimization." Vol. 1: "Unconstrained Optimization." John Wiley & Sons.

[34] Fletcher, R. (1981): "Practical Methods of Optimization." Vol. 2: "Constrained Optimization". John Wiley & Sons.

[35] Fletcher, R. and Powell, M.J.D. (1963): "A Rapidly Convergent Descent Method for Minimization." Comp. J., 6, pp. 163-168.

[36] Fletcher, R. and Reeves, C.M. (1964): "Function Minimization by Conjugate Gradients." Comp. J., 7, pp. 149-154.

[37] Gill, P.E. and Murray, W. (1972): "Quasi-Newton Methods for Unconstrained Optimization." J.I.M.A., 9, pp. 91-108.

[38] Gill, P.E., Murray, W. and Wright, M.H. (1981): "Practical Optimization." Academic Press. London.

[39] Goldfarb, D. (1970): "A Family of Variable Metric Methods Derived by Variational Means." Math. Comp., 24, pp. 23-26.

[40] Goldstein, A.A. (1962): "Cauchy's Method of Minimization." Numer. Math., 4, pp. 146-150.

[41] Goldstein, A.A. (1965a): "On Newton's Methods." Numer. Math., 7, pp. 391-393.

[42] Goldstein, A.A. (1965b): "On Steepest Descent." S.I.A.M. J. Control, 3, pp. 147-151.

[43] Goldstein, A.A. (1967): "Constructive Real Analysis." Harper & Row. New York

[44] Hestenes, M. (1980): "Conjugate Direction Methods in Optimization." Springer Verlag.

[45] Hestenes, M. and Stiefel, E. (1952):" Methods of Conjugate Gradients for Solving Linear Systems." Journal of Research of the National Bureau of Standards., 29, pp. 409-439.

[46] Hooke, R. and Jeeves, T.A. (1961): "Direct Search Solution of Numerical and Statistical Problems." J.A.C.M., 8, pp. 212-229.

[47] Huard, P. (1964): "Resolution de Programes Mathematiques a Contraintes Nonlineaires par la Methode des Centres." Note E.D.F., H.R. 5690/3/317.

[48] Huard, P. (1967): "Resolution of Mathematical Programming with Nonlinear Constraints by the Method of Centres." in Nonlinear Programming, (J. Abadie Ed.), pp. 207-219. North-Holland Publishing Co. Amsterdam.

[49] Huard, P. (1968): "Programmation Mathematique Convexe" R.I.R.O., 7, pp. 43-59.

[50] John, F. (1948): "Extremum Problems with Inequalities as Subsidiary Conditions." in Studies and Essays, Courant Anniversary Volume (Interscience, New York), pp. 187-204.

[51] Kowalik, J. and Osborne, M.R. (1968): "Methods for Unconstrained Optimization Problems." American Elsevier, New York.

[52] Kuhn, H.W. and Tucker, A.W. (1951): "Nonlinear Programming." in Proc. Second Berkeley Symposium (University of California Press. Berkeley, 1951), pp. 481-492.

[53] Le, D. (1985): "A Fast and Robust Unconstrained Optimization Method Requiring Minimum Storage." Math. Prog., 30, pp. 41-68.

[54] Lieu, B.T. and Huard, P. (1966): "La Methode des Centres dans un Espace Topologique." Numer. Math., 8, pp. 56-67.

[55] Lootsma, F.A. (1969): "Hessian Matrices of Penalty Functions for Solving Constrained Optimization Problems." Philips Res. Repts., 24, pp. 322-331.

[56] Lootsma, F.A. (1972): "A Survey of Methods for Solving Constrained Minimization Problems Via Unconstrained Minimization." in Numerical Methods for Nonlinear Optimization, (F.A. Lootsma Ed.). Academic Press.

[57] Mc.Guire, M.F. and Wolfe, P. (1973):"Evaluating a Restart Procedures for Conjugate Gradients." Report RC-4382, IBM Research Centre, Yorktown Heights.

[58] Minoux, M. (1986): "Mathematical Programming, Theory and Algorithms." John Wiley & Sons, Chichester.

[59] Murray, W. (1967): "Ill-Conditioning in Barrier and Penalty Functions Arising in Constrained Nonlinear Programming." The 6th Math. Prog. Symp., Princeton University.

[60] Nelder, J.A. and Mead, R. (1965): "A Simplex Method for Function Minimization." Comp. J., 7, pp. 308-313.

[61] Nemirovsky, A.S. and Yudin, D.B. (1983): "Problem Complexity and Method Efficiency in Optimization." John

Wiley & Sons, Chichester.

[62] Oren, S.S. and Luenburger, D.G. (1974): "Self Scaling Variable Metric Algorithms." Part 1, Criteria and Sufficient Conditions for scaling a Class of Algorithms. Man. Sci., 20, (5)

[63] Osborne, M.R. (1985): "Finite Algorithms in Optimization and Data Analysis." John Wiley & Sons Ltd. Chichester.

[64] Ostrowski, A.M. (1966): "Solution of Equations and Systems of Equations." 2cd Edition. Academic Press, New York.

[65] Polak, E. (1971): "Computational Methods in Optimization: A Unified Approach." Academic Press, New York.

[66] Polak, E. and Ribière,G. (1969): "Note sur la Convergence des Methodes de Directions Conjuguees." Revue Francaise Infor. Rech. oper., 16, R1, pp. 35-43.

[67] Polak, E., Sargeant, R.W.H. and Sebastian, D.J. (1974): "On the Convergence of Sequential Minimization Algorithms." J.O.T.A., 14, pp. 439-442.

[68] Powell, M.J.D. (1964): "An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives." Computer J. 7, pp. 155-162.

[69] Powell, M.J.D. (1969): "A Method for Nonlinear Constraints

in Minimization Problems." in Optimization, R. Fletcher Ed., Academic Press, London.

[70] Powell, M.J.D. (1971): "On the Convergence of Variable Metric Algorithms." J.I.M.A., 7, pp. 21-36.

[71] Powell, M.J.D. (1974): "Introduction to Constrained Optimization." in Numerical Methods for Constrained Optimization, P.E. Gill and W. Murray Eds., Academic Press Inc.

[72] Powell, M.J.D. (1975): "Some Global Convergence Properties of a Variable Metric Algorithm for Minimization Without Exact Line Searches." AERE Harwell Report Css15.

[73] Powell, M.J.D. (1976a): "Some Global Convergence Properties of Variable Metric Algorithms forMinimization Without Linear Searches." in "Nonlinear Programming." R.W. Cottle and C.E. Lemke, Eds., SIAM-AMS Proceedings IX, SIAM.

[74] Powell, M.J.D. (1976b): "Some Convergence Properties of the Conjugate Gradient Method." Math. Prog., 11, pp. 42-49.

[75] Powell, M.J.D. (1977): "Restart Procedures for Conjugate Gradient Methods." in Math. Progra. 12, pp.241-254.

[76] Powell, M.J.D. (1983): "Nonconvex Minimization Calculations and the Conjugate Gradient Method." Report No. DAMTP 1983/NA14, Dept of Applied Mathematics and Theoretical

Physics, University of Cambridge (Cambridge, England 1983).

[77] Powell, M.J.D. (1985): "Convergence Properties of Algorithms for Nonlinear Optimization." Report No. DAMTP 1985/NA1, Department of Applied Mathematics and Theoretical Physics, University of Cambridge (Cambridge, England 1985).

[78] Reklaitis, G.V., Ravindran, A. and Ragsdell, K.M. (1983): "Engineering Optimization Methods and Applications." A Wiley-Interscience Publication. John Wiley and Sons. New York.

[79] Rockafellar, R.T. (1974): "Augmented Lagrange Multiplier Functions and Duality in Nonconvex Programming." S.I.A.M. J. Control and Optimization, 12, pp. 268-285.

[80] Rockafellar, R.T. (1976): "Lagrange Multipliers in Optimization." in S.I.A.M.-A.M.S. Proceedings Nonlinear Programming, Vol. IX, pp. 145-168.

[81] Rosenbrock, H.H. (1960): "An Automatic Method for Finding the Greatest or Least Values of a Function." Comput. J., 3, pp. 175-184.

[82] Rosenbrock, H.H.. and Storey, C. (1965): "Computational Techniques for Chemical Engineers." Pergamon Press, London, England.

[83] Sargeant, R.W.H. and Sebastian, D.J. (1973): "On the

Convergence of Sequential Minimization Algorithms." J.O.T.A., 12, pp. 567-575.

[84] Schwefel, H.P. (1981): "Numerical Optimization of Computer Models." John Wiley & Sons. Chichester.

[85] Shanno, D.F. (1970): "Conditioning of Quasi-Newton Methods." Math. Comp. J., 24, pp. 647-656.

[86] Shanno, D.F. (1985): "Globally Convergent Conjugate Gradient Algorithms." Math. Prog., 33, pp. 61-67.

[87] Shanno, D.F. and Phua, K.H. (1978): "Matrix Conditioning and Nonlinear Optimization." Math. Prog., 14, pp. 149-160.

[88] Spendley, W., Hext, G.R. and Himsworth, F.R. (1962): "Sequential Application of Simplex Designs in Optimization and Evolutionary Operation ." Technometrics, 4, pp. 441-461.

[89] Storey, C. (1962): "Application of a Hill-Climbing Method of Optimization." Chemical Engng. Sci., 17, 1, pp. 45-52.

[90] Swann, W.H. (1969): "A Survey of Nonlinear Optimization Techniques." F.E.B.S.- Letters, 2, March, Suppl., pp. 539-555.

[91] Swann, W.H. (1972): "Direct Search Methods." in Numerical Methods for Unconstrained Optimization, W. Murray Ed.,

Academic Press. pp. 13-28.

[92] Thomson, J.R. (1977): "Examples of Non-Convergence of Conjugate Descent Algorithms with Exact Line Searches." Math. Prog., 12, pp. 356-360.

[93] Touati-Ahmed, D. and Storey, C. (1986): "Globally Convergent Hybrid Conjugate Gradient Methods." Math. Res. No. 196. Loughborough University of Technology (Loughborough, England).

[94] Touati-Ahmed, D. and Storey, C. (1987): "Efficient Hybrid Conjugate Gradient Techniques." Math. Res. No. 204. Loughborough University of Technology (Loughborough, England).

[95] Touati-Ahmed, D. and Storey, C. (1990): "Efficient Hybrid Conjugate Gradient Techniques." J.O.T.A., , pp.
Vol. 64, N°2, February 1990 pp. 379-397

[96] Tremolieres, R. (1968): "La Methode des Centres a Troncatures Variables." These, Paris.

[97] Van Der Hoek, G. (1980): "Reduction Methods in Nonlinear Programming." Mathematisch Centrum. Amsterdam.

[98] Ward, L., Nag, A. and Dixon, L.C.W. (1969): "Hill-Climbing Techniques as a Method of Calculating the Optical Constants and Thickness of a Thin Metalic Film." Brit. J. Appl. Phys. (J. Phys. D.), Ser. 2, 2, pp. 301-304.

[99] Wolfe, M.A. (1978): "Numerical Methods for Unconstrained Optimization." Van Nastran Reinhold Company.

[100] Wolfe, P. (1969): "Convergence Conditions for Ascent Methods." S.I.A.M. Review, 11, pp. 226-235.

[101] Zoutendijk, G. (1970): "Nonlinear Programming, Computational Methods." in Integer and Nonlinear Programming, J. Abadie, Ed. North-Holland Publishing Co. Amsterdam.

[102] Zoutendijk, G. (1976): "Mathematical Programming Methods." North-Holland Publishing Co. Amsterdam.