This item was submitted to [Loughborough's Research Repository](#) by the author.
Items in Figshare are protected by copyright, with all rights reserved, unless otherwise indicated.

The numerical solution of an optimal temperature problem

PLEASE CITE THE PUBLISHED VERSION

PUBLISHER

Loughborough University of Technology

LICENCE

CC BY-NC 4.0

REPOSITORY RECORD

Walder, T.J.. 2021. "The Numerical Solution of an Optimal Temperature Problem". Loughborough University.
<https://doi.org/10.26174/thesis.lboro.15022233.v1>.

THE NUMERICAL SOLUTION OF
AN OPTIMAL TEMPERATURE
PROBLEM

by

T. J. Walder, B.Sc.

Supervisor: Professor C. Storey

Submitted for the degree of Master of Science of
Loughborough University of Technology.

March, 1969.

SUMMARY

The general problem of system optimization and its solution using Pontryagin's Maximum Principle is discussed. A computational study of the application of the principle to the solution of a particular non-linear problem (optimal temperature profile in a chemical reactor) with constraints on the control variable is presented. The resulting two-point boundary value problem is tackled by the fixed-step gradient, steepest ascent and conjugate gradient methods and the performances of all three compared over a wide range of conditions. In addition, factors influencing computational efficiency are investigated, and the relative merits of several numerical techniques required in the implementation of Pontryagin's method to the non-linear problem are considered.

To provide further comparison, the control function is parameterised and the problem tackled using a suitable hill-climbing technique; several forms for the control are assumed. Computational procedures and flow diagrams are included.

ACKNOWLEDGEMENTS

The author is indebted to Professor C. Storey for the encouragement and valuable advice given during the preparation of this thesis, and also for frequently giving so generously of his time for discussion.

Also, Miss C. Midlane's excellent typing under difficult conditions is gratefully acknowledged.

CONTENTS

Summary	ii
Acknowledgement	iii
 Chapter 1. Introduction	 1
1.1 Introductory.	1
1.2 Contribution of this thesis.	1
1.3 The problem considered and its computational analysis.	2
 Chapter 2. General statement of the problem and its solution using the Maximum Principle	 6
2.1 The problem in general terms.	6
2.2 Method of solution using Pontryagin's Maximum Principle.	8
2.3 Other considerations.	10
2.4 The canonical form.	12
2.5 Iterative techniques.	13
2.6 Types of control.	18
2.7 Ancillary techniques.	18
 Chapter 3. The Gradient Method in Function Space	 20
3.1 The problem considered.	20
3.2 Method of solution.	22
3.3 Solution of the two-point boundary value problem.	23
3.4 Derivation of method.	25
3.5 Choice of step size.	28
3.6 Computational procedure.	29
3.7 Numerical integration methods.	31

3.8	Results and discussion.	34
	Figures and Tables of results.	44
Chapter 4. The Steepest Ascent and Conjugate Gradient Methods.		54
4.1	Introductory.	54
4.2	Modified gradient methods.	55
4.3	The conjugate gradient method applied to the problem of system optimization.	58
4.4	The steepest ascent and conjugate gradient algorithm.	59
4.5	The linear search.	62
4.6	Computational procedure.	64
4.7	Results and discussion.	67
	Figures and tables of results.	78
Chapter 5. A Non-Gradient Method.		87
5.1	Introduction.	87
5.2	Parameterisation of control.	89
5.3	Optimizing routine.	92
5.4	Computational procedure.	94
5.5	Results and discussion.	95
	Figures and tables of results.	106
Chapter 6. Comparison of the Methods.		112
6.1	Solution of the problem.	112
6.2	Performance of the methods.	113
6.3	Conclusion.	118

Appendix A. Supplementary to Chapter 3.	120
A1 General flow diagram for the Gradient Method.	121
A2-A4 Computer programmes	123
A5 Runge Kutta method applied to the system.	132
A6 Hamming's predictor-corrector method	134
Appendix B. Supplementary to Chapter 4.	136
B1 General flow diagram for combined Gradient, Steepest Ascent and Conjugate Gradient Methods.	137
B2-B3 Computer programmes.	141
B4 Estimation of the norms.	149
B5 Linear search at constant step.	151
B6 Linear search with step-doubling.	154
Appendix C. Supplementary to Chapter 5.	157
C1 Flow diagram for discrete/linear temperature assumption with variable final time.	158
C2 Computer programme for linear form.	161
C3-C7 Subroutines for other forms.	163
References	168

CHAPTER 1Introduction

1.1 One of the difficulties which arises in the process of system optimization is the choice of technique to be used on the problem under consideration. It is perhaps true to say that experience so far suggests that no single optimization technique is superior to other techniques in the solution of every type of problem. The choice of method depends on many factors, including the characteristics of the problem and not least the computing facilities available. However, certain types of problem may perhaps best be tackled by one specific technique compared with any other, and as computational experience increases it may well be possible to type-cast problems in this way. The amount of computational experience yet available is not substantial and does not appear great enough to produce a detailed catalogue of the type mentioned for some time to come. Perhaps utilitarian motives have produced some of the work done to date in two particular fields - aerospace systems and chemical engineering [1]-[8].

1.2 Contribution of this Thesis

The work presented here is aimed at increasing the computational knowledge in the latter field, and analyses the results obtained in a study of the behaviour of combinations of selected optimizing techniques and associated numerical methods applied to a practical problem. Essentially it represents

(2)

an extension and modification of some of the methods employed in an optimization study previously presented with more limited computational facilities then available. [3,Ch.10]. Also attempted is the application of new methods in order to extend the experience available on the type of problem involved in the study mentioned.

Since the earlier work was completed, computer technology has advanced at its distinctive rapid pace, and consequently it has been possible to utilise the services of a more powerful machine. In all the computational exercises attempted in this work the programming was done in Fortran language and programmes were subsequently run on an I.C.T. 1905 machine with 32K immediate access store and magnetic disc and tape backing store. Previous to this the work had been pioneered in Autocode on a Ferranti Mercury machine limited to 570 words of fast storage. The latter had an addition time of $420 \mu\text{sec}$ compared with an addition time of $7 \mu\text{sec}$, fixed point, and $12 \mu\text{sec}$ double length floating point for the more powerful machine. Thus the absolute efficiency of a given method will be dependent on the computational facilities available.

1.3 The Problem Considered and Its Computational Analysis

The problem considered in this work is outlined in [3] but is restated in Chapter 3 for completeness.

Essentially it is one of maximizing some measure of the reactor performance through the selection of a temperature profile for a tubular reaction vessel in

(3)

which a chemical reaction is taking place under plug flow conditions. The state equations for the system are given in Chapter 3.

Among the many dynamic optimization techniques proposed, probably the two most successfully applied in recent years have been dynamic programming developed by Bellman [9], and the Maximum Principle derived by Pontryagin and his associates. Rosenbrock and Storey showed [3] that in the problem considered in this work the application of the latter technique appeared to be more promising.

The Maximum Principle was first introduced by Pontryagin in 1956 and was originally confined to continuous processes. Later his co-workers proved that the Maximum Principle was a necessary condition for time optimality and extended it to a general case of extremizing an integral performance index. Details of these results may be found in the English translation of Rozonoer's paper [10]; a statement of the principle as applied to the general problem in optimal control is given in Chapter 2 of this work. Also included in Chapter 2 is a discussion of the techniques available to solve the two-point boundary value problem which always occurs as a result of applying Pontryagin's method or the classical method of the calculus of variations. In Chapter 3 the specific problem mentioned earlier in this section is tackled by the application of the Maximum Principle in association with a gradient method for solving the two-point problem. The

opportunity is taken to extend and modify some of the results produced earlier and at the same time to investigate the comparative suitability of several selected numerical integration methods.

In Chapter 4 a gradient method is again used in conjunction with Pontryagin's method but is applied in a different manner - equivalent to a steepest ascent method. In addition a conjugate gradient method suggested recently by Lasdon et al [5] is applied and the relative merits assessed. Neither of these techniques had previously been used on the problem under consideration. Both methods involved a heavy dependence on linear search routines which have received some considerable attention in the literature [11]-[14]. The general shape of the optimal temperature profile suggested by the results produced by the application of Pontryagin's Maximum Principle allows some measure of parameterisation of the admissible control to be effected. The problem can now be reduced to a hill-climbing problem for which certain powerful computational methods exist [11, 12, 15, 16]. Chapter 5 presents the results of such an approach in which several expansions of the control variable are considered. These considerations are further to the work on variation of parameters done previously [3] and represent a suitable extension of the promising results obtained then. Each of Chapters 3, 4 and 5 contains a summary and discussion of the numerical results obtained and Chapter 6 gives an overall summary of the work done

(5)

and a qualitative comparison of the methods.

Appendices A, B and C include general flow diagrams, a selection of some of the computer programmes written, and details of the numerical techniques required in the application of the various methods to the problem considered.

CHAPTER 2General Statement of the Problem and Its Solution Using the Maximum Principle2.1 The problem in general terms.

The process to be controlled is described by a set of non-linear differential equations of the form:

$$\frac{dx_i}{dt} = f_i(x_1(t), x_2(t), \dots, x_n(t), u_1(t), u_2(t), \dots, u_m(t)) \quad (2.1)$$

$$t_0 \leq t \leq t_f$$

$$\text{where} \quad x_i(t_0) = a_i \quad i = 1, 2, \dots, n \quad (2.2)$$

or in vector form

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}) \quad (2.3)$$

$$\text{where} \quad \underline{x}(t_0) = \underline{a} \quad (2.4)$$

Here $\underline{x}$ is an n -dimensional vector of state variables representing the state of the process at time t and $\underline{u}$ is an m -dimensional vector of control variables. The variable t may alternatively represent some other physical quantity such as distance in a steady state continuous process. The initial values of the state variables at time $t = t_0$, assumed all given here, are represented by the components of the n -dimensional vector $\underline{a}$.

In the form given in equation (2.3) the system is autonomous; the method of dealing with non-autonomous system in which t appears explicitly on the right-hand side of the equation is mentioned briefly later.

A particular optimization problem related to such a process is one of finding a piecewise continuous control vector $\underline{u}(t)$, subject to the constraints

(7)

$$G_i(\underline{u}) \leq 0, \quad i = 1, \dots, m \quad (2.5)$$

which makes some objective function of the final values of the state variables

$$F = \sum_{i=1}^n k_i \mathbf{x}_i(t_f), \quad k_i \text{ constant}, \quad (2.6)$$

a maximum or minimum having regard to the differential constraints (2.3) and initial conditions on the state variables (2.4).

t_f here is the final time which may or may not be given. When variable end time is involved the form of solution given later in section 2.2 is modified only slightly [17, pp.261,262] but obviously leads to greater complexity.

Such a problem might alternatively involve the maximization or minimization of a cumulative measure as the objective function such as for example the integral performance index

$$\int_{t_0}^{t_f} \phi(\underline{x}, \underline{u}) dt \quad (2.7)$$

This case may be conveniently analysed by the introduction of an additional state variable x_{n+1} satisfying the differential equation

$$\dot{x}_{n+1} = \phi(\underline{x}, \underline{u}) \quad (2.8)$$

$$\text{with } x_{n+1}(t_0) = 0 \quad (2.9)$$

and thus simply enlarging the system represented by equation (2.3) by one member [3, p.236].

An important special case is the time optimal control

(8)

problem when $\phi(\underline{x}, \underline{u}) = 1$ in equation (2.7), and in which the values of the state variables are also given at final time t_f (see Section 2.3).

2.2 Method of Solution using Pontryagin's Maximum Principle.

The procedure for solving the problem using the Maximum Principle is to introduce an n -dimensional vector $\underline{p}(t)$, of adjoint variables, satisfying the adjoint differential equations

$$\dot{p}_i = - \sum_{j=1}^n p_j \frac{\partial f_j}{\partial x_i} \quad (2.10)$$
$$i = 1, \dots, n$$

$$\text{where } p_i(t_f) = \frac{\partial F}{\partial x_i(t_f)} = k_i \quad (2.11)$$

when F is as specified in equation (2.6) for example. It can be shown [7] that, provided the control vector is unconstrained, the $\underline{p}(t)$ are equivalent to the Lagrange multipliers which would occur on application of the classical method of the calculus of variations to the problem specified with the differential constraints of equation (2.3). Further, using the classical method the control vector is given at each time t by the solution of the necessary condition

$$\sum_{i=1}^n p_i \frac{\partial f_i}{\partial u_j} = 0 \quad j = 1, \dots, m \quad (2.12)$$

in conjunction with the integration of equations (2.10). This correspondence between the Lagrange multipliers and the adjoint variables $\underline{p}(t)$ breaks down, however, when the magnitudes of the forcing functions are

(9)

limited as would be the case expressed by the conditions in equation (2.5).

Also introduced is the Hamiltonian function H defined by the expression

$$H(\underline{p}, \underline{x}, \underline{u}) = \sum_{i=1}^n p_i f_i(\underline{x}, \underline{u}) \quad (2.13)$$

According to the Principle [10], if, for $t_0 \leq t \leq t_f$, $\underline{u}(t)$ is a piecewise continuous vector subject to constraint, then in order for the objective function F to be a maximum for the process described by equations (2.3) and (2.4), it is necessary that the non-zero continuous vector $\underline{p}(t)$ exists and satisfies equations (2.10) and (2.11), and that $\underline{u}$ is chosen so that H is a maximum for every t , $t_0 \leq t \leq t_f$.

If R is a closed, bounded region containing the admissible control vectors $\underline{u}$ the last necessary condition requires $\underline{u}$ to be chosen such that

$$H(\underline{p}, \underline{x}, \underline{u}) = \sum p_i f_i(\underline{x}, \underline{u}) \quad (2.13a)$$

is a maximum for $\underline{u} \in R$.

Thus if H has its maximum value for some $\underline{u}$ within the region R , the optimal control will be that solution of

$$\frac{\partial H}{\partial u_j} = 0 \quad j = 1, \dots, m \quad (2.14)$$

which makes H greatest.

Alternatively, this equation need not be satisfied when H has its maximum value corresponding to some $\underline{u}$ which has been constrained on the boundary of R .

Thus maximization (or minimization) is not always achieved by setting derivatives to zero; this is

perhaps the main difficulty in the proof of the Principle.

Equation (2.14) is the equivalent of equation (2.12) using the classical calculus of variations when the control vector is unconstrained, but in that context there is no distinction between maxima, minima or points of inflection, nor between local maxima (or minima).

Use of the same classical method on the problem in which control (or state) variables are to be restricted might be handled by including penalty terms in the performance criterion being used. These would have the effect of producing steep ridges when the boundary was approached. Pontryagin's Principle, therefore, can more conveniently handle the situation at least in the case of constrained control vectors. It has also been extended to cases where the state variables are subject to constraint [18].

Details of the analytical relationships between the calculus of variations and the maximum principle (and also dynamic programming) may be found in references [2], [3] and [7].

A summary of the method of solution in canonical form is given later in Section 2.4.

2.3 Other Considerations

With a fixed time interval $t_f - t_0$ there are basically two types of problem: a fixed end point problem and a free end point problem depending on whether or not the values of the state variables are given at final

(11)

time t_f . In the general problem and the method of solution specified in Sections 2.1 and 2.2 the situation is of the latter type.

However, in the fixed end point problem the form of solution given in Section 2.2 is only slightly changed - the modification necessary being that for each new condition on $x_i(t_f)$ the corresponding condition on $p_i(t_f)$ is dropped [18]. Thus if in addition to the conditions on the state variables given by equation (2.4) we consider the problem in which, for example, $x_p(t_f)$ and $x_q(t_f)$ are given and the objective function is

$$F = \sum_{i=1}^n k_i x_i(t_f) \quad i \neq p, i \neq q \quad (2.15)$$

then the method of solution given in Section 2.2 is modified to allow $p_q(t_f)$ and $p_p(t_f)$ from equation (2.11) to be free.

Non-autonomous systems may be dealt with using a similar technique to that mentioned in Section 2.1 for integral performance criteria. Thus if the system has state equations of the form

$$\dot{\underline{x}} = f(\underline{x}, \underline{u}, t) \quad (2.16)$$

$$\underline{x}(t_0) = \underline{a}$$

then the technique is to introduce a new state variable x_{n+1} such that

$$x_{n+1}(t) = t \quad t_0 \leq t \leq t_f \quad (2.17)$$

This variable satisfies the state differential equation

$$\dot{x}_{n+1} = 1 \quad (2.18)$$

with initial condition

(12)

$$x_{n+1}(t_0) = t_0 \quad (2.19)$$

and reduces the system equations (2.16) to the autonomous type of equation (2.3). The method of solution in Section 2.2 now applies to the enlarged system.

2.4 The Canonical Form

The Hamiltonian function H was defined in equation (2.13) as

$$H = \sum_{i=1}^n p_i f_i$$

Differentiating H with respect to p_i we have

$$\begin{aligned} \frac{\partial H}{\partial p_i} &= f_i \quad i = 1, \dots, n \\ &= \dot{x}_i \quad \text{from equation (2.1)} \end{aligned}$$

Differentiating H with respect to x_i we have

$$\begin{aligned} \frac{\partial H}{\partial x_i} &= p_1 \frac{\partial f_1}{\partial x_i} + p_2 \frac{\partial f_2}{\partial x_i} + \dots + p_n \frac{\partial f_n}{\partial x_i} \\ \therefore \frac{\partial H}{\partial x_i} &= \sum_{j=1}^n p_j \frac{\partial f_j}{\partial x_i} \quad i = 1, \dots, n \\ &= -\dot{p}_i \quad \text{from equation (2.10)} \end{aligned}$$

Also as has already been indicated in Section 2.2, equation (2.12) becomes

$$\frac{\partial H}{\partial u_j} = 0 \quad j = 1, \dots, m$$

Pontryagin's Maximum Principle as applied to the general problem specified in Section 2.1 may thus be summarized in conjunction with the following equations in canonical form:

(13)

$$H = \sum_{i=1}^n p_i f_i \quad (2.20)$$

$$\dot{x}_i = \frac{\partial H}{\partial p_i} \quad \text{with } x_i(t_0) = a_i \quad (2.21)$$

$$\dot{p}_i = -\frac{\partial H}{\partial x_i} \quad \text{with } p_i(t_f) = \frac{\partial F}{\partial x_i(t_f)} \quad (2.22)$$

$$\max_{\underline{u} \in R} \left\{ H = \sum_{i=1}^n p_i f_i \right\} \quad (2.23)$$

In this form, if the control vector $\underline{u}$ is subject to constraint, then the set of necessary conditions for a maximum (minimum) of the objective function F are the state and adjoint equations (2.21) and (2.22) together with the requirement of equation (2.23) that $\underline{u}$ must be chosen to maximize (minimize) H either directly from equation (2.14) if $\underline{u}$ is within the region R of admissible control, or by inspection of its values taken on the boundaries of R . In either case the $\underline{u}$ selected is that which determines the greatest (least) H .

2.5 Iterative Techniques

Equations (2.21)-(2.23) are generally difficult to solve when the system equations are non-linear although analytical solutions have been obtained in the case of linear dynamics with linear or quadratic objective functions. In order to obtain the solution of the set of state and adjoint differential equations and determine the optimal solution to the system being considered a two-point boundary value problem must be solved. The reason for this is that the boundary

conditions on the adjoint variables $\underline{p}$ are given at the end of the time interval while the boundary conditions on the state variables $\underline{x}$ are known at the start. This situation occurs whether Pontryagin's method or the calculus of variations is being used and the methods of solution may roughly be divided into three groups:

- (a) boundary value iteration,
- (b) quasilinearization,
- (c) relaxation.

These are discussed briefly in turn below.

2.5.1 Boundary value iteration

This is the classical indirect method and involves estimating either the final values for the state variables or the initial values for the adjoint variables. The two sets of differential equations may then be integrated in either the backward or forward directions in time. The initial values of the state variables or final values of the adjoint variables arising from this process may then be compared with the respective given values in order to determine the starting values for the next cycle. This determination is iterative and may be done by using some static hill-climbing technique to minimize some form of error criterion formed from the specified boundary conditions and the iterated unknown ones. Although this process is usually easy to implement, it is generally found to have poor convergence - unless the iterative process is commenced with a very good approximation to the final solution.

One reason for bad behaviour of the method lies in the numerical difficulties which may arise from the instability of one or other of the sets of state and adjoint differential equations [17, p.246]. For a system with stable state equations, it can be shown that the adjoint equations are generally unstable in the direction of increasing time. Thus small errors in the estimates for the initial values of the adjoint variables tend to cause large errors in the terminal values. The situation is not helped by alternatively assessing final values for the state variables, as although in the backward direction of integration the adjoint equations will be stable, unfortunately the state equations will be unstable. This is probably the main drawback of the method which has been applied to the problem considered later in Chapter 3 with little success [3, Ch.10]. Further examples of the application of the method may be found in [19],[20].

2.5.2 Quasilinearization

The second of these methods, which is also indirect, is referred to by Kalaba [21] as 'quasilinearization', and solves a sequence of linear boundary-value problems in such a way that the solution sequence converges to the required solution of the non-linear problem. The method is a generalisation of the Newton-Raphson method for solving scalar equations and is discussed by Lee [1] with applications to chemical engineering, and by Kenneth and McGill [6] who apply it to problems in aerospace. It has also been applied in comparison with

direct methods (described next) by Kopp et al [8, pp.65-106]. Although from their work Kenneth and McGill and also Lee conclude that the method may be a useful tool for non-linear optimal control problems the emphasis in the work done here is placed on direct methods of solving the two-point boundary value problem.

2.5.3 Relaxation Methods

It has been indicated in the method of boundary value iteration that a principal difficulty lies in the directional instability of either the state or adjoint equations when integrated and used with the coupling equation (2.23). Nevertheless, if an iterative procedure can be devised to produce $\underline{u}$ without reference to the coupling equation the difficulties inherent in the problem are markedly eased. One reason for this is due to the fact that the state equations can be integrated in forward time independently of the adjoint equations which in turn can be integrated in the direction of time decreasing. The instability difficulty associated with the boundary value iteration is now eliminated since for stable process dynamics these computations will both be stable.

Such a method which does not attempt to satisfy the coupling equation directly is variously known as the relaxation method [7], the gradient method in function space [3], [17, Ch.6], or the steepest ascent or descent of the Hamiltonian [22]. These methods, which are classed as direct, proceed by solving a sequence

of non-optimal problems with the property that each successive solution leads to an improved value of the objective function being optimized. A common technique is to perform iterations on the control function which is initially selected arbitrarily in such a manner that the condition for optimality is not satisfied. In this class of methods, which are applied in Chapters 3 and 4, a direct search is made for the extreme value of the objective function. One apparent disadvantage of the method is the slow rate of convergence particularly in the vicinity of the optimum. Included in the same class are the second variation techniques which incorporate feedback based on the second derivatives of the performance measure. These techniques which have been developed by Merriam [7] and Mitter [23] have rapid convergence near the optimum but tend to run into convergence difficulties if control is far from the optimum where the inclusion of second order effects is of relatively little value. The advantages are offset by the vastly increased computing and analytical effort when compared with the gradient methods mentioned earlier. More recently, second order successive approximation methods have been developed through differential dynamic programming techniques [24], and there is also a second order computational scheme involving the solution of integral equations [25] which appears to work well compared with the method developed by Merriam [7]. At present there appears to be little computational evidence that either of these will work

well on complex problems.

2.6 Types of Control

The type of control arising within a given system depends on the form of the objective function being considered and the state equations of the system. For a control vector subject to constraint Takahashi has shown [26] how the optimal control of a given system may be continuous, bang-bang, or continuous with saturation, depending on the form of the objective function. The problem considered in Chapter 3 comes within the latter classification.

An interesting discussion of various possibilities with several examples is given by Fan [2, pp.36-68].

2.7 Ancillary Techniques

For the specific problem considered in later chapters of this work and described in Chapter 3, the state and adjoint differential equations were non-linear and required a numerical method of solution. Some attention was paid to the choice of technique since the efficient solution of this part of the problem in terms of accuracy achieved and computing time taken was highly important.

It was also of interest to determine whether there was any interaction between the chosen hill-climbing method and numerical integration technique. Details of three techniques used and a discussion of their performances are given in Chapter 3.

A further technique was required in the analysis of the

same problem by the methods described in Chapter 4. In that chapter the implementation of Pontryagin's method involved the frequent use of a linear search routine. A description of two such routines together with a discussion of their respective computational efficiency is presented in Chapter 4.

CHAPTER 3The Gradient Method in Function Space

In this Chapter and in Chapters 4 and 5 a specific problem is analysed. This problem, which concerns the determination of the optimal temperature profile in a tubular reactor under certain conditions, was first investigated computationally by Rosenbrock and Storey [3]. Their work suggested the promising nature of the gradient method in function space as a method of solving the two-point boundary value problem associated with Pontryagin's Maximum Principle applied in this context. The problem they considered is detailed here for completeness.

3.1 The Problem Considered

The reaction scheme is of the kind shown in the figure, and is assumed to take place in a

tubular reactor with plug flow conditions.

The concentrations of the feed material A, the required yield C and the intermediate product B are denoted by x_1 , x_3 and x_2 respectively; D is a side product which is to be wasted.

With this notation the kinetic equations for the system may be written [3,p.242]:

(21)

$$\begin{aligned}\frac{dx_1}{dt} &= -(k_1 + k_2 + k_3)x_1 = f_1 \\ \frac{dx_2}{dt} &= k_1x_1 - k_4x_2 = f_2 \\ \frac{dx_3}{dt} &= k_4x_2 - k_5x_3 = f_3\end{aligned}\tag{3.0}$$

with initial conditions $x_1 = 1, x_2 = 0, x_3 = 0$ (3.1)

at time $t = 0$.

In vector form

$$\dot{\underline{x}} = \underline{f}(\underline{x}, T)\tag{3.2}$$

where $\dot{\underline{x}}$ is the column vector with elements $\left\{ \dot{x}_1, \dot{x}_2, \dot{x}_3 \right\}$
 $\underline{f}$ is the column vector with elements $\left\{ f_1, f_2, f_3 \right\}$

$T = T(t)$ is the absolute temperature.

These forms are equivalent to the state equations described for the general system in Section 2.1 of Chapter 2. In this particular problem the kinetic rate constants $k_1, \dots, k_5$ are given in terms of the absolute temperature T at any point in the reactor by the expressions

$$k_i = C_i \exp \left\{ -\frac{E_i}{R} \left(\frac{1}{T} - \frac{1}{658} \right) \right\}\tag{3.3}$$

$$i = 1, \dots, 5$$

R is the gas constant, and the frequency factors C_i and activation energies E_i used in all this work are as listed in Table 3.12. In this form the problem is one of estimating a temperature gradient $T(t)$ along the length of the reactor which will give maximum yield x_3 at the final time t_f .

The nature of the reaction scheme being studied is such that a falling temperature profile from inlet to outlet would be expected for optimum yield; there

(22)

will also be an optimal contact time t_f [3]. From practical considerations of catalyst stability the temperatures along the reactor are constrained to be $\leq 550^\circ\text{C}$.

3.2 Method of Solution

The adjoint set of differential equations required by Pontryagin are derived in this problem from equation (2.10)

$$\dot{p}_i = - \sum_{j=1}^3 p_j \frac{\partial f_j}{\partial x_i} \quad i = 1, 2, 3$$

where the f_j are taken from equations (3.0).

We thus have the set:-

$$\begin{aligned} \frac{dp_1}{dt} &= (k_1 + k_2 + k_3)p_1 - k_1 p_2 \\ \frac{dp_2}{dt} &= k_1(p_2 - p_3) \\ \frac{dp_3}{dt} &= k_5 p_3 \end{aligned} \quad (3.4)$$

with end conditions given by equation (2.11)

$$p_i(t_f) = \frac{\partial F}{\partial x_i(t_f)}$$

In this case, since $F = x_3(t_f)$ we have,

$$p_1(t_f) = 0 \quad p_2(t_f) = 0 \quad p_3(t_f) = 1 \quad (3.5)$$

Also the Hamiltonian function H is from equation (2.13)

$$H = \sum_{i=1}^3 p_i f_i$$

thus

$$\begin{aligned} H = & -(k_1 + k_2 + k_3)x_1 p_1 + k_1 x_1 p_2 - k_4 x_2 p_2 + k_4 x_2 p_3 \\ & - k_5 x_3 p_3 \end{aligned} \quad (3.6)$$

Also

$$\frac{\partial H}{\partial T} = \frac{1}{RT^2} \left\{ -(k_1 E_1 + k_2 E_2 + k_3 E_3) x_1 p_1 + k_1 E_1 x_1 p_2 \right. \\ \left. - k_4 E_4 x_2 (p_2 - p_3) - k_5 E_5 x_3 p_3 \right\}$$

The estimate of the temperature profile $T(t)$ which maximizes the yield x_3 at final time now requires at each point in time the maximization of H either from the equation

$$\frac{\partial H}{\partial T} = 0 \quad (3.7)$$

or by selecting T on the constraint boundary, i.e. in this case 550°C .

This situation which requires the solution of the two-point boundary value problem posed by the non-linear sets of differential equations (3.0) and (3.4) is beset by the difficulties already described in Chapter 2.

In the previous analysis of this problem it was shown that the use of a boundary value iteration technique was unsatisfactory due to the extreme sensitivity of the temperature profile $T(t)$ to the guesses for the concentrations x_1 , x_2 and x_3 at time t_f ; these starting values necessary for the backward integration of the kinetic equations gave quite inadequate values for $x_i(0)$, $i = 1, 2, 3$ [3]. Much better results were obtained and are extended here by use of a relaxation method [7] - the mechanics of which are now discussed.

3.3 Solution of the Two-point Boundary Value Problem

As has been pointed out in Section 2.5 the difficulty lies in the requirement that equations (3.0) and (3.4) must be solved simultaneously with the coupling equation (3.7). Since the boundary conditions (3.1)

and (3.5) are given for the state variables at the reactor inlet ($t = 0$) and for the adjoint variables at the reactor outlet ($t = t_f$) respectively at neither of $t = 0$ nor $t = t_f$ can the solutions be commenced simultaneously. The method of approach to be adopted here is one in which the control vector is selected arbitrarily as a function of time in the first instance so that the condition for optimality in the form of equation (3.7) is not satisfied. The corresponding state and adjoint vectors may then be obtained by integrating equations (3.0) and (3.4) in the forward and backward directions respectively. Thus if the control vector is known the problem may be decoupled into a pair of one-point boundary value problems. In addition, for stable system dynamics both directions of integration initiated by this procedure will be stable, but the solutions obtained together with the assumed control will not generally form an optimal set due to the arbitrary initial choice made for the control vector.

What is now required is a method of selecting the next control vector in such a way that the choice leads to an improvement in system performance. In this way the two-point boundary value problem may be solved by considering a sequence of pairs of one-point boundary value problems.

For the problem being considered here the method of selecting the new control function for the next iteration is derived in the next section. The method

(25)

has been referred to as the gradient method in function space [3, 17, 27].

3.4 Derivation of Method

The performance index to be maximized is

$$J = x_3(t_f) \quad (3.8)$$

subject to the differential constraints (3.2) in vector form

$$\dot{\underline{x}} = \underline{f}$$

which are the state equations for the problem.

Introducing Lagrange multipliers in the form of the column vector $\underline{p}$ with components (p_1, p_2, p_3) and using matrix notation we have the unconstrained performance index

$$I = J + \int_0^{t_f} dt(\underline{p}'(\underline{f} - \dot{\underline{x}})) \quad (3.9)$$

where $\underline{p}'$ denotes the transpose of $\underline{p}$.

It is convenient at this stage to introduce the Hamiltonian defined as in equation (2.13)

$$H = \sum_{i=1}^3 p_i f_i$$

or in vector form

$$H = \underline{p}'\underline{f}$$

so that

$$I = x_3(t_f) + \int_0^{t_f} dt(H - \underline{p}'\dot{\underline{x}}) \quad (3.10)$$

If we now allow the small changes $\delta x_3(t_f)$, $\delta \dot{\underline{x}}$, δT , $\delta \underline{p}$ and $\delta \underline{x}$ to occur so that I becomes I_1 we have

$$I_1 = I + \delta x_3(t_f) + \int_0^{t_f} dt \left\{ (\nabla_{\underline{x}} H)' \delta \underline{x} + (\nabla_{\underline{p}} H)' \delta \underline{p} + (\nabla_T H)' \delta T - \underline{p}' \delta \dot{\underline{x}} - \dot{\underline{x}}' \delta \underline{p} \right\} \quad (3.11)$$

(26)

where $\nabla_{\underline{x}}H$ is the column vector with components

$\left(\frac{\partial H}{\partial x_1}, \frac{\partial H}{\partial x_2}, \frac{\partial H}{\partial x_3}\right)$ and $\nabla_{\underline{p}}H$ similarly has components

$\left(\frac{\partial H}{\partial p_1}, \frac{\partial H}{\partial p_2}, \frac{\partial H}{\partial p_3}\right)$. Also $\nabla_T H = \frac{\partial H}{\partial T}$.

Integration by parts gives

$$-\int_0^{t_f} \underline{p}' \delta \underline{x} dt = \int_0^{t_f} \underline{\dot{p}}' \delta \underline{x} dt - \underline{p}'(t_f) \delta \underline{x}(t_f) \quad (3.12)$$

so that now

$$I_1 = I + \delta x_3(t_f) - \underline{p}'(t_f) \delta \underline{x}(t_f) + \int_0^{t_f} dt \left\{ (\nabla_{\underline{x}}H)' \delta \underline{x} + (\nabla_{\underline{p}}H)' \delta \underline{p} + (\nabla_T H)' \delta T + \underline{\dot{p}}' \delta \underline{x} - \underline{\dot{x}}' \delta \underline{p} \right\} \quad (3.13)$$

For unconstrained T the necessary conditions for a maximum or a minimum of I are now obtained by selecting

$$\begin{aligned} \underline{\dot{x}} &= \nabla_{\underline{p}}H \\ -\underline{\dot{p}} &= \nabla_{\underline{x}}H \end{aligned}$$

These conditions represent the state and adjoint equations in canonical form as quoted in Section 2.4 of Chapter 2.

Thus, assuming these conditions are satisfied and taking account of the boundary conditions for $\underline{x}$ and $\underline{p}$,

$$I_1 = I + \int_0^{t_f} dt \left\{ (\nabla_T H)' \delta T \right\} \quad (3.14)$$

where $\delta T = T_{\text{new}} - T_{\text{old}}$

$$= T_{i+1} - T_i \quad (\text{say})$$

Also since $\nabla_T H = \frac{\partial H}{\partial T} = (\nabla_T H)'$ and taking $\delta I = I_1 - I$ we have finally the relation

$$\delta I = \int_0^{t_f} \left\{ \frac{\partial H}{\partial T} \delta T \right\} dt \quad (3.15)$$

(27)

This is the equation which leads to the required method of selecting T_{i+1} from T_i . By setting $\frac{\partial H}{\partial T} = 0$ one might conceivably obtain the required conditions for an exact maximum (or minimum); however, this procedure is ill-advised as it leads to the same difficulties inferred by the original two-point boundary value problem. The δT which increases the value of the objective function need only make δI positive and is consequently not unique.

The choice made for the method derived here is that

$$\delta T = \epsilon \frac{\partial H}{\partial T} \quad \text{where } \epsilon > 0$$

so that $T_{i+1} = T_i + \epsilon \frac{\partial H}{\partial T}$ (3.16)

Other possibilities are [7], [28]

$$\delta T = \epsilon \operatorname{sgn}\left\{\frac{\partial H}{\partial T}\right\} \quad \text{where } \epsilon > 0 \quad (3.17)$$

where $\operatorname{sgn}(y) = \begin{matrix} 1 & y > 0 \\ -1 & y < 0 \end{matrix}$

and $\delta T = \epsilon W(t) \frac{\partial H}{\partial T}$ (3.18)

where $W(t)$ is an arbitrary positive definite matrix. Equation (3.16) provides the essential step in the solution of the two-point boundary value problem by the gradient method in function space. The fundamental idea involved is that T_i , $\underline{x}$ and $\underline{p}$ and hence $\frac{\partial H}{\partial T}$ are all specified at the time previous to when T_{i+1} is required. Thus the estimation of the optimal temperature profile does not involve the solution of the necessary coupling equation.

(28)

When T is subject to constraints of the form

$$T_L \leq T \leq T_u$$

the selection process provided by equation (3.16) is modified to give

$$T_{i+1} = T_u \quad \text{when } T_i + \epsilon \frac{\partial H}{\partial T} \geq T_u$$

$$\text{and } T_{i+1} = T_L \quad \text{when } T_i + \epsilon \frac{\partial H}{\partial T} \leq T_L$$

Merriam [7,pp.246,247] establishes these results in an interesting geometrical manner.

Further detailed discussion of the method may be found in [3], [7,ch.10], [17,Ch.6], [27,Ch.5].

3.5 Choice of Step Size

The considerations involved here are discussed in some detail by Merriam [7,Ch.10] for example.

If ϵ has been chosen so that $\delta I > 0$ then T_{i+1} is a better solution to the optimal profile than T_i . If this condition is always satisfied throughout the application of the method then the process should converge to the optimal profile.

For satisfactory behaviour of the process the step should be reasonably small so that δT and consequently $\delta \underline{x}$ are small enough to substantiate the neglect of the second and higher order terms in the Taylor expansions used in Section 3.4. However, with the choice of a small value for ϵ there can be an intolerably slow rate of convergence to the optimum which can seriously limit the efficiency of the method. Therefore, some compromise is seen to be necessary in order to satisfy both the condition that the objective function should

show some improvement and also that this improvement should be sustained at a substantial rate. This rate of convergence to the optimum which is an important measure of efficiency is not only measured by the number of iterations required but also by the time taken per iteration.

The simplest form for ϵ is a constant and since it is used in conjunction with $\frac{\partial H}{\partial T}$ to determine T_{i+1} from T_i the terms "large" and "small" ϵ are relative terms as $\frac{\partial H}{\partial T}$ plays a similar role to ϵ and will vary between problems.

In the analysis of the problem being considered here the behaviour of the method under a full range of constant step sizes was considered; no attempt at doubling or halving an initially chosen step was made. No doubt a general programme utilising the relaxation technique involved here could have a suitable criterion such as last mentioned for creating an acceptable step size within the computations.

3.6 Computational Procedure

The general flow diagram for the complete method is shown in Appendix A1. The main steps involved are outlined below - the scheme for other problems would involve similar logic as outlined in the computational procedure of Chapter 4.

- (1) A starting temperature profile for the reactor is assumed and the kinetic rate constants calculated according to equation (3.3) at the

points along the reactor required in (ii) and (iii) below.

- (ii) The state equations (3.0) can now be integrated in the forward direction and the values of the state variables x_1 , x_2 and x_3 stored at each integration step. (The choice of numerical integration routine and integration step size is discussed in the next section.)
- (iii) The adjoint equations (3.4) with boundary conditions (3.5) are also integrated but in the direction of time decreasing, the equations being stable when integrated in this direction for a system with stable state equations. The values of p_1 , p_2 and p_3 are also stored at each integration step.
- (iv) It is now possible to evaluate $\frac{\partial H}{\partial T}$ from the expression in equation (3.6) at each integration step and this in conjunction with a constant value of ϵ provides the incremental changes in temperature profile $\epsilon \frac{\partial H}{\partial T}$ along the reactor length in accordance with equation (3.16). The value for ϵ is kept constant throughout the computation.
- (v) If, during the execution of stage (iv), the temperature at any point exceeds the allowable limit of 550°C ., the corresponding temperature is set to that value prior to further computation.
- (vi) The temperature profile $T(t)$ and yield $x_3(t_f)$ are printed out at selected points along the reactor for each iteration.

(vii) The kinetic rate constants are recalculated for the new temperature profile from (iv) and stages (ii)-(vi) repeated until no further significant change in profile is detectable. At the completion of a given investigation a print out of the temperature profile and the distributions of the concentrations x_1 , x_2 and x_3 is available along the length of the reactor.

Computer programmes based on this procedure are presented in Appendices A2, A3 and A4.

3.7 Numerical Integration Methods

One aspect of the investigation which was considered to be of some importance was the choice of numerical integration routine in the solution of the kinetic and adjoint differential equations. The main factors influencing the choice included accuracy, the absence of strongly unstable characteristics and reasonable simplicity leading to speed of computation. Three methods were selected for comparison of behaviour when applied to the problem being considered here.

The trapezoidal rule was the first of these methods and this was the method used to integrate the differential equations in [3]. With it the finite difference equations representing the state and adjoint differential equations take particularly simple forms. One disadvantage is the loss of accuracy at the larger integration step lengths since the principle truncation error of the method with integration step h is of order h^3 . However, this can perhaps be relatively

offset by the speed of computation implicit in the use of the method due to the simple form of the equations being processed. Another advantage is that no special starting procedure is required.

The second of these methods which was also a direct method was the standard fourth order Runge Kutta method [29]. This method also requires no special starting procedure, involves light demands on storage and is ideally suited for automatic computation since a straightforward procedure is repeated several times. The available sub-routine on the I.C.T. 1905 machine was not used here since its use would have involved some modification in any case. Instead the fourth order method was programmed directly; the fairly simple form this took for the set of kinetic equations (3.0) is shown in Appendix A5. A disadvantage of the method is perhaps that the time taken in the continuous evaluation of the functions may be somewhat lengthy. However, on the credit side the principal truncation error of the method is of order h^5 and the method should, therefore, be capable of effective use at the larger integration step lengths with a consequent reduction in computing time.

This now raises the problem that in operating at a relatively large integration interval the partial instability, of which the Runge Kutta method is perhaps suspect, might feature in the computation. Problems of stability are always difficult to resolve, especially in the case of simultaneous sets of non-

linear differential equations, but a suitable range of step lengths will be used as part of the comparison with the trapezoidal rule. It is of interest to note that, on the basis of applying the trapezoidal rule to the traditional linear first order equation, this method could avoid partial instability completely [30]. The third method considered, although still in the same class of step by step methods, was essentially different in character being a predictor-corrector type of method. In view of the known strongly unstable properties of the classic predictor-corrector methods reference was made to a method by Hamming [31] in which the stability characteristics are shown to be favourable up to relatively large integration step lengths. In this method the predictor and corrector formulae have the same order of accuracy and, therefore, the difference between the results from each formula can be used to estimate the truncation error of the process. This difference is, in fact, fed back into the process and eliminates part of the truncation error thus producing a method of higher accuracy. Details of the method which has a principle truncation error of order h^6 may be found in Appendix A6. A disadvantage is that a special starting procedure is necessary; this is provided by the Runge Kutta method already mentioned. One advantage claimed for Hamming's method over the Runge Kutta process is that it should be quicker since only two function evaluations are required for each step forward as against the four required in Runge Kutta.

It should be interesting to see how this works out for the complex problem being considered here.

3.8 Results and Discussion

This section contains a selection of the results obtained in a series of runs on an I.C.T. 1905 series computer using the gradient method described earlier in this chapter. These results which are summarized and discussed here are presented in the form of graphs and tables at the end of this chapter.

The earlier results in the analysis of the problem were obtained on a Ferranti Mercury machine with relatively severe limitations on available fast storage and also much slower computing time (see Chapter 1).

Consequently, using the method mentioned on the faster machine has enabled an extension of the analysis of the problem; some modification has also been possible.

In addition to a comparison of the performances of the selected numerical integration routines, this analysis includes an investigation into the effects of using various step sizes in the direction of search, the choice of numerical integration step length and how these react on the yield and temperature profile.

Consideration is also given to the variation of yield at final time with choice of starting temperature profile and also with reactor length or contact time. Flow diagrams and programmes for the method may be found in Appendix A.

3.8.1 Reactor Yield and Temperature Profile

Fig. 3.1 shows typical growth curves for the yield x_3

along the reactor length as re-cycling of the method takes place. Characteristic changes in the temperature profile along the reactor which are associated with the yields of Fig. 3.1 are indicated in Fig. 3.2.

Variations shown are for a reactor length or contact time of 1 sec. and a starting isothermal temperature of 400°C . limited to 550°C . at any stage in the process; this limitation in the steady state is seen to occur over the region occupying some 12-13% of the reactor's length at the inlet. The integration of the kinetic and adjoint differential equations was performed using the trapezoidal rule with 160 integration steps. The choice of step length used here in the direction of search is seen to be such that the best yield at final time is reached in effectively few iterations. The background to this choice is indicated in Fig. 3.3 where the effect on yield of increasing the step factor ϵ along the direction of search is shown; this aspect is considered in more detail later.

It is perhaps worth mentioning at this stage that a direct comparison with previous results using this method is not possible since during routine checks on the computation it was realised that in the analysis given in [3] the adjoint equations were inadvertently integrated incorrectly - in effect producing a forward type of integration rather than the necessary integration in the direction of time decreasing. This may account for the pronounced instability encountered in the computation there at relatively

small values of ϵ . No such instability was encountered in the present investigation, thus confirming the promise of the method used on this problem as compared with the classical treatment of the 2-point boundary value problem discussed in Section 2.5.1 of Chapter 2. In this work, somewhat larger values of ϵ were found necessary to produce any progress in the method; this scale effect may perhaps be traced to relatively small slopes of the 'hill' in function space for this problem particularly at some distance from the optimum value. Other differences include larger final values of the yield and a more 'positive' climb to these values - the temperature profile was also changed along the entire reactor length, a feature not produced previously by the method.

3.8.2 Effect of Increase of Step-size in Directions of Search

The best yield obtained at the reactor outlet under the conditions described in 3.8.1 was 43.8% although higher values were recorded at points within the reactor as noted in previous work [3]. (Thus, if the temperatures in the reactor after these points were reduced to a low value the reaction would be frozen with consequent improvement in final yield.)

In order to assess how quickly the best yield could be obtained the effect of choosing from a range of step lengths in the direction of search was investigated and Figs. 3.3 and 3.4 indicate the manner in which an increase in step length can reduce the number of iterations and hence time required to produce the best

(and lower) values of yield. The dependence of the results on choice of integration routine also shown in Fig. 3.4 is discussed in the next section. The size of step was in fact restricted only by oscillatory behaviour of the final yield at very large values of ϵ . This behaviour is illustrated in Table 3.5 in comparison with normal circumstances; at this step size the method makes no progress. At large step lengths below the latter critical size a type of "overshoot" was encountered (see also Table 3.5) in the first iteration but quickly cleared allowing the method to progress normally - i.e. as for lower values of ϵ . This early behaviour under these conditions appears to be a direct consequence of retaining only first order terms in the Taylor expansion on which the method is based. Nevertheless, although this behaviour was an early sign of limitation on step length, the violation at step lengths below critical was temporary only and subsequent iterations involving smaller increments of temperature and yield satisfied the linearity assumption and enabled the method under these conditions to lead to the quickest results. As can be seen from Fig. 3.4 the best value of 0.438 commencing with an isothermal temperature profile of 400°C. along the reactor can be first achieved in as few as 10 iterations which seems remarkable for a first order process. From the computer results for this case and also as indicated in Figs. 3.1 and 3.2 further re-cycling of the method produced only small changes in the temperature profile with no

further improvement in yield (to the accuracy quoted here).

Of course, the few iterations required to produce this result has only been possible at the expense of the computation implied in the results shown in Fig. 3.4. However, this approach was considered preferable and beneficial in this part of the analysis of the problem since it created a feel for the method being applied and also because the experience gained could be used to economise on computing time in the intended further work involving stepping along the gradient trajectory. (See later this chapter and Chapter 4). It is suggested that this style of approach could also prove useful in other problems in which an extensive analysis involving this type of method was contemplated.

3.8.3 Computational Comparison of Numerical Integration Techniques

It would be possible to considerably reduce the time taken in these investigations by choosing a larger integration step length than that used so far (160 steps along the reactor). However, since in the trapezoidal formula for integration the principal truncation error is of order h^3 , where h is the integration step length, reasonable care must be taken here in order to avoid undue errors in the results. The two other integration routines mentioned in Chapter 2 were therefore introduced, and to compare the performances of all three methods computing was done over a range of integration step lengths and all computer runs were automatically timed. In this way

the results gave some idea of the errors involved at the larger integration step lengths, and also indicated the efficiency of each method in terms of the compromise between accuracy of computation and time taken. The adaption of the fourth-order Runge Kutta method and Hamming's predictor-corrector method to the differential equations of this problem is shown in Appendix A5 and Appendix A6 respectively; the corresponding forms for the trapezoidal rule have already been referenced in Chapter 2. As was expected the Runge Kutta method took longer in execution than the trapezoidal rule but, of course, the interest here was the interplay between additional time taken and retained accuracy at larger integration step.

In practice, the trapezoidal and Runge Kutta methods gave substantially the same results at 160 steps of integration; the scale of Fig. 3.4 exaggerates the very minor difference in the results at this step size. Further computing was done at 20, 40 and 80 steps for both methods and also an overall check computation was done with 320 steps ($h = 0.00325$) to substantiate the earlier accuracy. Some of these results are shown in Fig. 3.6 and these suggest the suitability of the Runge Kutta method with regard to the level of accuracy required in this work and also the margin of error likely in using the trapezoidal rule at the larger integration steps. This retention of required accuracy at the larger step in using Runge Kutta easily compensates for the additional computing time implied.

The attractiveness of Hamming's method lies in its built-in stability compared with the traditional predictor-corrector methods and also the fact that the corrector formula has only to be applied once with each predictor. This is because the method includes a modification which allows a correction for the principal truncation error (which is of order h^6) at each step in the integration. The method is more accurate than the Runge Kutta with principal truncation error of order h^5 , and it was to be hoped that it was also quicker. This hope - partially supported by the fact that Hamming's method involves only two function evaluations per iteration as against four in the Runge Kutta scheme - did not materialise. The reason for this was the ancilliary computation implicit in Hamming's method compared with the more straightforward procedure of Runge Kutta (see again Appendix A6); no doubt the former would perform more favourably for a much more complicated function. It is of interest to note that in this application the predictor-corrector method did eventually behave in an unstable manner at the 20 step size $h = 0.05$. Apart from this the behaviour was quite satisfactory, the results, effectively coincident with Runge Kutta, being included in Fig. 3.4. The comparative execution times for the three methods are shown in Table 3.11.

3.8.4 Variation of Yield with Final Time

In some situations the objective function will continue to show improvement so long as the final time increases,

in others it will be some specific contact time for which the objective function is maximized (or minimized). The latter is the case here and in fact it may easily be shown how the best isothermal yield depends on a compromise between the temperature of the reactor and the reaction time [3, p.243]. The modification of the Pontryagin form of solution [17, pp.261,262], however, involves some considerable complication within the method being used here to solve the two-point boundary value problem. To alleviate this a number of computer runs were made with different final times in order to indicate the order of improvement produced in the yield as contact time was increased. It was not possible to proceed far with this particular aspect of the investigation using this method in the earlier analysis due to the limitation on the available fast storage mentioned previously. The results from the present investigation using the Runge Kutta integration routine with 80 steps are shown in Figs. 3.7 and 3.9. The main effect on the temperature profile can be seen to be a progressive lowering of reactor temperature with increase of contact time, but with little change in the region of constrained temperature at the reactor inlet. The yield shows a substantial improvement in value as the contact time is increased from 0.5 to 4.0 sec. Beyond this the increase is less marked, a yield of 45.0% being recorded for a contact time of 8 sec. for example. A further computation at 12 sec. contact time produced a yield

of 45.2% after 120 iterations and with the temperature profile effectively having reached the steady state. Evidently the rate of variation of yield with reactor length tends to become extremely small as the latter is increased and further satisfactory investigation at larger contact times using this method would require substantial amounts of computer time with no guarantee of success.

From the evidence available some compromise between yield and contact time could conceivably be taken from a practical standpoint - this would no doubt be the deciding factor in many problems.

3.8.5 Performance from Different Starting Temperature Profiles

Reverting back to the contact time of 1 sec., several computer runs were done starting with assumed isothermal reactor temperatures within the range 10°C-500°C to assess the effect of different starting conditions and to test the reliability of the method. Fig. 3.10 shows the results for a value of $\epsilon = 30 \times 10^3$ and indicates the effectiveness of the method when started from isothermal temperature profiles within the range 300-500°C. Fig. 3.8 also shows the best isothermal yield at a contact time of 1 sec. obtainable from the same computer runs. As one of the many checks on the computation generally these last values were confirmed independently by using the analytical solution to the problem for the isothermal case and computing the yield for the various temperatures. This check was done by hand and the results corresponded (to the

accuracy used in this work) with these obtained using the approximate integration routines.

The performance of the method from the 200°C starting temperature profile was sluggish and required 37 iterations to achieve the best yield of 43.8%. This compares with some 14 iterations for the starting profiles in the range 300-500°C. The method made no progress when started from the 10°C and 100°C profiles due to the small gradient components encountered at the start. Further discussion on progress under these conditions may be found in Chapter 4.

(44)

Fig. 3.1. Progressive Growth of Yield.

Fig. 3.2 Progressive Changes in Temperature Profile.

(46)

Fig. 3.3. Effect of ϵ on Growth of Yield.

Fig. 3.4. Variation, with ϵ , of the Number of Iterations Required to Achieve the Best Yield.

(48)

$\epsilon \times 10^{-3}$		15	30	50	70
Start	Temp. °C	400	400	400	400
	Yield (t_f)	0.357	0.357	0.357	0.357
1st Iteration	Temp. °C	418.7	437.3	462.2	487.1
	Yield	0.400	0.385	0.319	0.269
2nd Iteration	Temp. °C	412.6	405.3	399.8	390.4
	Yield	0.407	0.421	0.422	0.405
5th Iteration	Temp. °C	390.7	373.1	369.0	421.9
	Yield	0.430	0.435	0.436	0.399
6th Iteration	Temp. °C	385.1	368.5	362.6	357.6
	Yield	0.432	0.436	0.437	0.394
10th Iteration	Temp. °C	371.6	358.6	354.0	357.2
	Yield	0.436	0.437	0.438	0.394
15th Iteration	Temp. °C	363.1	353.6	350.8	419.7
	Yield	0.437	0.438	0.438	0.400
		NORMAL		'OVERSHOOT'	OSCILLATORY

Table 3.5. Behaviour with increase of ϵ ; contact time = 1 sec.

Fig. 3.6. Computational Error at Large Integration Steps - Trapezoidal and Runge Kutta Methods.

(50)

Fig. 3.7. Effect of Reactor Length (Contact Time) on Final Yield.

Fig. 3.8. Best Isothermal Yield; Contact Time = 1 sec.

Fig. 3.9. Effect of Contact Time on Temperature Profile.

Fig. 3.10. Performance of the Method from Different Starting Temperature Profiles.

Computer	I.C.T. 1905	Ferranti Mercury
Integration Routine	Time per iteration (sec)	Time per iteration (sec)
Trapezoidal	2.40	43*
Fourth Order Runge Kutta	2.65	-
Hamming Predictor-Corrector	3.45	-

*Estimated from figure for 20 steps [3].

Table 3.11. Comparative computing times for the gradient method using three different numerical integration schemes. Figures shown are for 160 integration steps; estimates for other choices may be approximated by scaling.

i	1	2	3	4	5
Frequency factors C_i	1.02	0.93	0.386	3.28	0.084
Activation energies E_i	16000	14000	15000	10000	15000

Table 3.12. Basic data for the problem of Chapter 3.

CHAPTER 4Steepest Ascent and Conjugate Gradient Methods

4.1 Most of the procedures for locating the position

$\underline{x} = \underline{x}_m$ of the maximum or minimum of the function $F(\underline{x})$ may be conveniently described by means of the iterative formula

$$\underline{x}^{k+1} = \underline{x}^k + t_k \underline{s}_k \quad (4.1)$$

where $\underline{s}_k$ denotes the vector direction of search and t_k the step size in that direction at the k^{th} iteration.

$\underline{x}_m$ is then located as the limit of the sequence $\underline{x}^0, \underline{x}^1, \underline{x}^2, \dots$ where $\underline{x}^0$ is the initial approximation to $\underline{x}_m$.

A gradient method selects

$$\underline{s}_k = \underline{g}_k = \nabla F(\underline{x}^k) \quad (4.2)$$

i.e. the direction of fastest increase of F , when the maximum of $F(\underline{x})$ is being sought; conversely the negative gradient is chosen as the direction of search for a minimum.

The various gradient methods differ in their choice of t_k . In Chapter 3 this choice was $t_k = \text{constant}$ and a single step only was taken in each prescribed direction - this will be referred to as the gradient method here.

Perhaps the most widely used idea is that of taking successive steps along the given direction until a maximum (or minimum) is reached and then, having defined the next direction for search at this point, to repeat the process in this and subsequent directions. Thus in each direction $\underline{s}$ the step t would be chosen such that it produces from a starting point $\underline{x}$,

$$\begin{array}{c} \text{max. } F(\underline{x} + t\underline{s}) \\ \text{(min.)} \end{array} \quad (4.3)$$

The maximum or minimum of $F(\underline{x})$ is then located as the limit point $\underline{x} = \underline{x}_m$ of several linear searches of the type specified in equation (4.3).

This type of method with successive directions $\underline{s}$ selected as in equation (4.2) will be referred to here as the method of steepest ascent (or descent). The manner in which the linear search may be executed in practice is discussed in Section (4.5).

Unfortunately, it is often the case that the gradient direction is only the best direction locally and may become steadily worse as the direction is followed with, as a consequence, slow or oscillatory approach to the limit $\underline{x}_m$.

4.2 Modified Gradient Methods

To combat this, other methods have been developed which modify the gradient in order to improve the directional properties. One of these methods selects as the direction of search

$$\underline{s}_k = H(\underline{x}_k) \underline{g}_k \quad (4.4)$$

where H is a suitable matrix of second derivatives of $F(\underline{x})$. For a quadratic function, H is a matrix of constants and it may be shown that the problem of maximizing $F(\underline{x})$ may be solved in one step by taking

$$H(\underline{x}) = [\nabla^2 F(\underline{x})]^{-1} \quad (4.5)$$

where $\nabla^2 F(\underline{x})$ is the matrix of second order partial derivatives of $F(\underline{x})$, the typical element $\frac{\partial^2 F}{\partial x_i \partial x_j}$

being located in row i and column j .

More generally, H is not a matrix of constants and is often not known analytically so that numerical differentiation techniques need to be used. In Davidon's method [14], (see also [13]), H is generated by successive approximations involving successive values of $\underline{g}_k$; the unit matrix is the usual first choice for H . The attractive features of the method are that it combines the descent (or ascent) property of the conventional gradient method together with quadratic terminal convergence, thus avoiding the characteristic final behaviour of the conventional gradient methods. The method is relatively complex and requires a large amount of computer storage particularly when the number of variables involved is large; consequently in many problems a computationally simpler scheme may be preferred. Another method which is certainly more economical on computer storage and manipulation is the Fletcher and Reeves [32] adaptation of the conjugate gradient method of Hestenes and Stiefel [33]. In this method when used for minimization, the direction vectors are successively formed using the relations

$$\underline{s}_k = -\underline{g}_k + \beta_{k-1}\underline{s}_{k-1} \quad (4.6)$$

$$(k = 1, 2, \dots)$$

where the scalar β_{k-1} is given by

$$\beta_{k-1} = \frac{|\underline{g}_k|^2}{|\underline{g}_{k-1}|^2} \quad (4.7)$$

$$\text{and} \quad |\underline{g}_k|^2 = \underline{g}'_k \underline{g}_k \quad (4.8)$$

$\underline{g}'_k$ being the transpose of the vector $\underline{g}_k$.

Initially $\underline{s}_0 = -\underline{g}_0$, so that the first linear search is made along the same direction as that in the scheme given by equation (4.2).

Myers [34] has shown that, for a quadratic function, the direction vectors of the Davidon algorithm (as reformulated by Fletcher and Powell in [13]) and the conjugate gradient algorithm just mentioned are scalar multiples of each other, and, therefore, the methods will generate identical steps (to a minimum) assuming no round-off error and a perfect one-dimensional search. For general functions the directions $\underline{s}_k$ generated by equations (4.6) and (4.7) are those corresponding to the local quadratic approximation in the current iteration of the process, and the overall rate of convergence will depend on how quickly the process reaches some suitable neighbourhood of the minimum in which the function resembles a quadratic and fast terminal convergence is expected.

The two methods have been compared in [32] on Rosenbrock's banana-shaped valley resulting in the superiority of Davidon's method in terms of number of iterations required for convergence; however, as mentioned earlier a disadvantage of the method is its greater complexity and, therefore, a comparison of computer run times is entirely another matter.

Further to this it has been suggested by Kelley [35] on the basis of limited computations, that because of the numerical differentiation error effect in the Davidon scheme it was considered essential to compute

in double precision arithmetic in order to avoid loss of definiteness of the Davidson H-matrix and subsequent loss of the descent property. The conjugate gradient algorithm did not experience this difficulty (as long as the gradient components were correct) and, therefore, could be worked with single precision arithmetic.

In view of all these considerations it was decided to apply the method of steepest ascent and the conjugate gradient method described in this section to the problem defined in Chapter 3.

4.3 The Conjugate Gradient Method Applied to the Problem of System Optimization

Whereas the method of steepest descent (ascent) has been one of the methods readily used in the solution of optimal control problems, and its behaviour and limitations noted [e.g. 8, pp. 65-106], only comparatively recently has mention been made of the use of the conjugate gradient method outlined in Section 4.2.

In [5] Lasdon et al extend the conjugate gradient minimization method of Fletcher and Reeves [32] to optimal control problems and advocate the method as a promising compromise between present first order gradient techniques and the more sophisticated second variation methods which require a great deal more computation. Convergence of the method is also proved and the comparative results of tackling three problems using the steepest ascent and conjugate gradient methods are presented. These results indicate, for the problems

considered, superiority in rate of convergence for the conjugate gradient method compared with the method of steepest ascent, with very little additional computation per iteration. These problems were essentially simple in structure, although one was non-linear in the state variables; the question naturally arises as to whether similar conclusions would result on a problem of greater complexity.

In a further report [36], Sinnott and Luenberger extend the conjugate gradient algorithm to systems in which the final state must satisfy a linear terminal constraint. In their work a more exact form for the parameter β of equation (4.7) is assumed at the expense of much greater complexity in the computation. Their development of the theory is applied to a fairly simple system involving a linear terminal constraint and they obtain similar conclusions to those in [5].

Apart from a small amount of numerical work on the computation of optimal orbit transfers [35] there appears to be very little else available in the way of practical application of this conjugate gradient algorithm to the problem of system optimization.

4.4 The Steepest Ascent and Conjugate Gradient Algorithms

The steepest ascent algorithm may be derived in exactly the same manner as the gradient method of Chapter 3; the difference in application is through the definition of steps taken as described in Section (4.1). The method may also be recovered from the conjugate gradient algorithm of [5] which is restated here for completeness.

(60)

(i) The problem:-

In the notation of Section (2.1) of Chapter 2 we wish to minimize

$$J = \phi(\underline{x}(t_f)) \quad (4.9)$$

subject to the differential constraints

$$\dot{\underline{x}} = \underline{f}(\underline{x}, \underline{u}) \quad (4.10)$$

with initial conditions

$$\underline{x}(t_0) = \underline{a} \quad (4.11)$$

Here $\underline{u} = \underline{u}(t)$ is the control function and the performance index J is a function of the state variables at final time $t = t_f$. In the remainder of this section a single control function $u(t)$ is assumed ($m = 1$ in Section 2.1); the theory is easily extended to the multicontrol case.

In addition we introduce as in Section (2.2) of the same chapter, the adjoint differential equations

$$\dot{p}_i = - \sum_{j=1}^n p_j \frac{\partial f_j}{\partial x_i} \quad (4.12)$$

$i = 1, 2, \dots, n$

where $p_i(t_f) = \frac{\partial \phi}{\partial x_i}(t_f) \quad (4.13)$

and also the Hamiltonian function

$$H = \sum_{i=1}^n p_i f_i(\underline{x}, \underline{u}) \quad (4.14)$$

The gradient is then

$$g(u) = \frac{\partial H}{\partial u} \quad (4.15)$$

(ii) The algorithm:-

An arbitrary starting value for the control, $u = u_0$, is selected, then:-

(61)

$$g_0 = g(u_0) \quad (4.16)$$

and $s_0 = -g_0 \quad (4.17)$

is the first direction for linear search.

In this and subsequent directions s_k , $t = t_k$ is chosen to minimize

$$J(u_k + ts_k) \quad (4.18)$$

this is the function space analogue of equation (4.3).

The $(k+1)^{th}$ approximation to the optimal control is then formed from

$$u_{k+1} = u_k + t_k s_k \quad (4.19)$$

so that $g_{k+1} = g(u_{k+1}) \quad (4.20)$

with the next search direction

$$s_{k+1} = -g_{k+1} + \beta_k s_k \quad (4.21)$$

where

$$\beta_k = \frac{(g_{k+1}, g_{k+1})}{(g_k, g_k)} \quad (4.22)$$

and $(g_k, g_j) = \int_{t_0}^{t_f} g_k(t) g_j(t) dt, -$

these again being the function space analogues of equations (4.6)-(4.8).

Putting

$$\beta_k = 0 \quad (4.23)$$

in equation (4.21) recovers the method of steepest descent from the algorithm; both methods use the same direction for search in the first iteration.

The computational procedure for the algorithm is set out in Section (4.6).

For the direct application of this scheme to the problem of Chapter 3, the problem may be reformulated to minimize $-x_3(t_f)$; in this context the linear searches^e involved in equation (4.18) are equivalent to linear maximizations of $x_3(t_f)$.

The choice of linear search technique is discussed in the next section.

4.5 The Linear Search

In the methods to be used in this chapter the problem of maximizing a function along a one dimensional curve must be solved many times during the solution of the main problem. This linear search can, therefore, occupy a large proportion of the total time for the solution of the problem and should, therefore, obviously be as efficient as possible. Some suitable balance should be struck between the accuracy and required computation for a given process; the accuracy implied by fitting a high order polynomial may be outweighed by exorbitant computation time due to the complexity.

One scheme used in [5] is that proposed by Davidon and described by Fletcher and Powell [13]. The scheme is in three stages and the interpolation is performed using a cubic fit; a single application of the formulae involved is expected to yield the exact value of the optimum step in the limit as the function becomes quadratic in the neighbourhood of the local maximum or minimum. In addition it is shown through numerical tests that no significant reduction in the number of iterations can be achieved by using higher accuracy

interpolation.

It was not possible to use this scheme in the problem considered here since estimates of the first derivatives of the function required for the cubic fit were not available.

A common practice is the use of a parabolic fit to the function. Three items of information are needed, one of which certain to be available is the current best estimate of the solution; in the absence of derivatives, the value of the function at two other points may be used. A second degree expression approximating the function is obtained, and the point at which it attains its maximum is used to estimate the next direction for search (after comparison with the current best estimate to check whether the approximation was sufficiently good).

Two variations of this method were used here and are described fully in Appendices B5 and B6

- (i) In the first technique the step was retained at a constant value until a reduction in the objective function was achieved; the quadratic was then fitted through the last three points. Obviously, in the regions remote from the final solution, the smaller the step the better the fit. This type of search was considered to be of interest because a direct comparison with the results obtained in Chapter 3 using the gradient method there would then be possible. Of particular interest was whether the large step sizes found possible in the

latter method could be sustained in the methods of this chapter. Computational experience with this technique in conjunction with the conjugate gradient method necessitated the inclusion of a step-halving procedure. Thus, if at any stage the step was found to be too large, it was automatically halved and retained at the new value in subsequent searches unless further reduction was necessary.

- (ii) In the second method of search, the technique adopted was one of starting with a nominally small step and repeatedly doubling the length until a reduction in the objective function was again achieved. At this stage a half step back was taken and the maximum 'boxed in' with the three points much closer together as described in Appendix B6. This method is as described by Swann [11] and increases the accuracy of the interpolation without increasing the number of function evaluations required. As in method (i) the maximum obtained through the quadratic fit was compared with the current best estimate in the search being used before the next stage of the computation.

4.6 Computational Procedure

4.6.1 The procedure in general is as follows:-

- (1) An arbitrary starting control $u = u_0$ is chosen and the state and adjoint equations (4.10)-(4.13) for the system integrated in the directions of increasing and decreasing time respectively over the range $t = t_0$ to $t = t_f$.

(65)

The values of the state and adjoint variables at each point of time in the integration processes are stored and the performance index corresponding to $u = u_0$ calculated and stored.

- (ii) The gradient g_0 is calculated at each instant using equations (4.15)-(4.16) and $-g_0$ is taken as the first direction of search s_0 . The norm of g_0 is also calculated and stored according to the formula

$$\int_{t_0}^{t_f} g_0^2 dt$$

using some suitable integration formula.

- (iii) A step t is now taken in direction s_0 from $u = u_0$ to give the new control $u = u_0 + ts_0$. The state equations (4.10)-(4.11) are re-integrated and the performance index recalculated.

- (iv) Step (iii) is repeated until the linear search being used has satisfactorily achieved the required minimization of the objective at step $t = t_0$ and control $u_1 = u_0 + t_0 s_0$.

Then for $k = 0$;

- (v) The gradient g_{k+1} at each point of time and its norm are calculated with the control at u_{k+1} and stored as in (i) and (ii). The new direction of search s_{k+1} at each point is estimated using equation (4.21) where the scalar parameter β_k is available from the ratio of the previously calculated and stored norms as in equation (4.22).

- (vi) Steps (iii) and (iv) are now repeated with s_{k+1} replacing s_0 , u_{k+1} replacing u_0 and the new

control obtained according to equation (4.19)

For $k = 1, 2, \dots$

(vii) The whole process now revolves around successive applications of steps (v) and (vi) until the objective function and the corresponding control have been stabilised at a predetermined accuracy.

Setting $\beta_k = 0$ at any stage results in the method of steepest descent being applied.

4.6.2 For the problem defined in Chapter 3 the flow diagram showing the application of this computational procedure is shown in Appendix B1.

Also shown in Appendix B2 and B3 are two of the computer programmes written for the problem. The first of these is a combined programme which, by selection of a parameter, results in the application of any one of the gradient method (Chapter 3), method of steepest ascent or conjugate gradient method. The linear search routine for the programme works with constant step size which must be specified.

The second programme is a combined steepest ascent - conjugate gradient with the linear search performed with step doubling from an arbitrary starting step size. Further programmes used for the application of the methods of this chapter were modifications of these two. Details concerning the integration of the state and adjoint differential equations and the numerical integration processes required in the estimation of the gradient trajectory norms are given in Section 4.7 in which computer results are presented and discussed.

4.7 Results and Discussion

4.7.1 This section contains a summary of the main results.

It was considered of interest not only to compare the relative merits of the steepest ascent and conjugate gradient methods when applied to the problem under consideration but also wherever possible to directly compare results with those obtained using the gradient method defined in Chapter 3. Since that method used a fixed step length throughout, this comparison was indeed possible by using, for the methods of this chapter, the linear search technique with constant step as described in Appendix B5. The linear search with step doubling allowed comparison only of the methods of this chapter, but the comparative merits of both searches was also of interest.

As in Chapter 3 other points of interest were an analysis of search step sizes and their effect on the time taken by the methods and also the behaviour of each method from several different starting profiles. The effect of using variable final time was not included as it was felt that this could be adequately assessed from the results in Chapter 3; a contact time of 1 sec. is used throughout. Graphical and tabular displays of results referred to in this section may be found at the end of the chapter.

4.7.2 Preliminary Considerations

Simpson's rule [37, p.120] was chosen for the estimation of the norms involved in equation (4.22) because of its simplicity and reasonable accuracy, the principal

truncation error being of order h^4 for integration interval h in the full application of the rule. Since the gradient components required in this estimation were directly available following the integration of the state and adjoint differential equations (4.10)-(4.13) in conjunction with equation (4.15), the integration step length h could conveniently be the same in both processes. It was recognised that in this way, the relatively large integration step lengths found possible in the method of Chapter 3 with the Runge Kutta scheme would not be possible unless some more accurate estimate of β_k was available; this point is considered again later. For the method of steepest ascent, where the norms are not involved, similar conclusions to those of Chapter 3 concerning the size of h still apply. 160 integration steps were used with the Trapezoidal method of Chapter 3 for the integration of the state and adjoint differential equations. Fig. 4.1 shows the gradient trajectory for the first direction of search starting from the 400°C isothermal profile (available from the results of Chapter 3). A hand calculation of the first norm using this data gave a check on the computer result of 0.526×10^{-5} . It was appreciated that in other stages of the analysis this figure could be considerably smaller (0.432×10^{-6} for the second direction for example) and again this point is considered later.

By cross-plotting appropriate results from Chapter 3 it was possible to obtain the variation of yield in

the direction of the first search; this is shown in Fig. 4.2. This allowed a check to be made on the computations from each of the linear search techniques and also gave some indication of suitable step sizes - at least in the early stages of application of the method. Without this information, an analysis similar to that of the last chapter could have been performed.

4.7.3 Linear Search at Constant Step

From the evidence of Fig. 4.2, an ϵ of 1.0×10^4 was selected and results computed for the steepest ascent and conjugate gradient methods starting from a profile isothermal at 400°C . The results are shown in Fig. 4.3 where they are compared with results using the same step size from application of the method of Chapter 3. This comparison, which is on the basis of the number of steps taken, does not account for the differences in computing time per step for each method. It was found that the time taken computing the new directions of search at each step in the gradient method was greater than the time taken in the linear search routines of the other two methods. These times are shown in Table 4.5; the difference between the steepest ascent and conjugate gradient is accounted for by calculation of the norms and different search directions for the latter.

Inspection of Fig. 4.3 shows the progress in the conjugate gradient method to be considerably worse than the other two methods; this may well be due to the approximate nature of the factor β in regions away from

the optimum. Further information on this behaviour was available from inspection of the temperature profiles for the two methods. The conjugate gradient had forced considerably more of the inlet region into the temperature constraint than the steepest ascent method - and in fact more than that reached in the optimum configuration of Chapter 3. The net effect of this was detrimental to the yield and in subsequent directions of search the method was clearly trying to remedy the situation in which the temperature constraint appears partially to blame. The steepest ascent method, which approached the constraint more uniformly, worked very well and produced similar final results for temperature to those by the gradient method of Chapter 3 shown in Fig. 3.2. Some measure of the differences in trajectory which account for some of the behaviour during the second direction of search is shown in Fig. 4.1 for the two methods of this chapter. These differences arise inherently from application of equation (4.21). Further inspection of the computer print-out indicated that although the step-halving procedure had not been called for by the steepest ascent method it had been used by the conjugate gradient method during the later searches. Therefore next, from the same starting profile, computations were carried out using reduced ϵ of 5×10^3 and 2.5×10^3 . It was anticipated that the choice of a smaller step would provide a more accurate quadratic fit to the local maximum and also allow a more gradual build up of

temperature avoiding perhaps the addition of relatively large increments at critical moments - a feature which had proved adverse for the conjugate gradient method in the earlier computations. Fig. 4.4 gives results for $\epsilon = 5 \times 10^3$ and indicates substantial improvement for the conjugate gradient method. Results for the smaller step with $\epsilon = 2.5 \times 10^3$ gave the same trends and simply confirmed that at this step size the process took approximately twice as long to reach a given stage in the computation. On the basis of this evidence and for the reasons mentioned earlier there appears to be a limit on step size for satisfactory working of the conjugate gradient method as applied to the problem considered here. Below this limit the method has worked well, requiring substantially less iterations to attain a yield of 0.437 than the steepest ascent and gradient methods. A comparison of iterations or steps required for the three methods together with computing time per iteration is presented in Table 4.5, the figures indicating some 30% reduction in the number of iterations required using the method of conjugate gradients. This tendency is in line with the general conclusions of [5] and [36].

The marginally greater yield of 0.438 achieved by both gradient and steepest ascent methods in substantially more iterations (see Fig. 4.7) was not attained by the conjugate gradient method suggesting possibly accumulation of error in the former or failure to converge in the latter. The convergence proofs of

[5] for the conjugate gradient algorithm assume exact location of the turning point in each linear search,- this is not possible in practice for the general problem and may account here for the small difference in yield between the methods after a substantial number of iterations. Inspection of computer print-out gives no real clue here although an interesting feature is provided by the comparison of temperature profiles for a given yield shown in Fig. 4.6. This comparison suggests that near the optimum the yield is rather insensitive to small variations in temperature outside of the region of constraint; this point is pursued further in Chapter 5.

Finally, the performance of the methods was evaluated at larger step lengths consistent with those used in Chapter 3. The results from the conjugate gradient method are summarized in Table 4.9 and show rather inferior rates of convergence for the reasons mentioned already in this section. The steepest ascent method showed a sustained increased rate of convergence over a wide range of step sizes and Figs. 4.7 and 4.8 give a comparison of the results with those obtained using the gradient technique of Chapter 3. The more favourable results using the conjugate gradient method with smaller steps are also included in Fig. 4.7. Fig. 4.8 indicates the gradient method's greater efficiency at the larger step lengths but with little or no saving in computer time.

It is of interest to note that the gradient method as

used in Chapter 3 can represent, particularly at the larger step lengths, the method of steepest ascent with 'subrelaxation' [38, p.98]. This is to say the optimum step t arising from equation (4.3) is replaced by ρt where ρ is a constant such that $0 < \rho < 1$; of course in the comparison here, ρ may vary between linear searches. In practice this modification of steepest ascent can produce improvement. Wolfe [39, p.114] quotes the value $\rho = 0.9$ as an effective choice in a minimization problem involving six variables. An average figure of 0.7 is indicated here from the computer print-outs at large step with $\epsilon = 50 \times 10^3$.

4.74 Linear Search With Step-Doubling

To provide a comparison with the results of Section 4.7.3 an alternative linear search was now introduced in which the steps taken within each search were progressively doubled starting from the same arbitrary starting step.

The particular technique used here was discussed in Section 4.5 and a full description is presented in Appendix B6.

Although the computing times per step remained effectively as for the constant step procedure of Section 4.7.3 (see Table 4.5), some saving in total computing time was expected particularly when testing the two methods of this chapter from starting profiles remote from the optimum. Also, in these regions, results obtained in Chapter 3 indicated small gradient components, and the accuracy provided by Simpson's rule in the evaluation of the required norms was, therefore,

likely to be affected unless a very small integration step h was chosen - with associated increase in computer time. To avoid this, Simpson's rule was replaced by a more accurate method having principal truncation error of order h^4 [37, p.124]. This method, which is described fully in Appendix B4, was found to be completely suitable throughout the subsequent computations. Additionally, its inclusion makes possible the use of a larger integration step in conjunction with the Runge Kutta scheme for the conjugate gradient method compatible with the suggestions in Chapter 3. The first point of investigation was the choice of step size in the linear search routine since a starting step was required and this was quite arbitrary. An interesting discussion of the considerations involved in the selection of step size is given by Merriam [7, p.242]; a frequently used starting choice in the general case is unity, but the best choice, i.e. that leading to the greatest rate of convergence, depends entirely on the particular problem being considered.

Starting from the 400°C isothermal temperature profile with $\epsilon = 1$ the method of steepest ascent achieved a yield of 0.437 in 51 steps and an apparently optimum yield of 0.438 in 87 steps. This latter yield remained unchanged after a total of 175 steps (10 linear searches), although small changes in the temperature profile occurred similar to those referred to in the corresponding situation in Chapter 3.

The corresponding results using the conjugate gradient

method showed slower convergence - a yield of 0.432 only being achieved in 53 steps and sustained to 135 steps with slowly changing profile; this performance was again initiated by the adverse features mentioned in Section 4.7.3 which were accentuated here by the large steps generated in the step doubling procedure. Further investigation of the effect of increase in starting step size on convergence rate, similar to that of the previous section, was then carried out and the results are summarized in Figs. 4.10 and 4.11 and Table 4.12.

The Table shows earlier increases in yield as the starting step size is increased for the conjugate gradient method but there is no real improvement in final rate of convergence. In these results, for the reasons outlined in Section 4.7.3, a greater proportion of the temperatures over the reactor inlet were reaching the constraint early in the computation compared with the corresponding temperatures arising from use of the steepest ascent method. Since these higher temperatures give rise to increased kinetic rate constants, calculations were repeated at this stage with some 500 integration steps to check whether increased truncation error was being introduced into the analysis. The results of this check indicated no significant changes. Results for the steepest ascent method shown in Fig. 4.10 indicate an increased rate of convergence sustained over a wide range of starting step sizes; Fig. 4.11 suggests that the best yield can be achieved in as few as 17 steps

compared with the 87 required at unit starting step. The final temperature profile throughout these computations was substantially that already shown in Fig. 4.6.

An estimate of the relative value of each linear search is provided by direct comparison of Figs. 4.7 and 4.8 with Fig. 4.11. This comparison indicates the magnitude of the considerable reduction in computing time anticipated for the process with step doubling except at the larger steps where the method using constant step is more efficient.

The situation for the conjugate gradient method at the larger step sizes for each process is summarized in Tables 4.9 and 4.12 and also Fig. 4.7. For the application being considered here, the search at constant step is seen to produce better results.

Next, the performance of the steepest ascent and conjugate gradient methods was assessed from several different starting profiles using a starting step with $\epsilon = 100$. Results are summarized in Tables 4.13 and 4.14. Table 4.13 indicates the very good performance of the steepest ascent method from each of the starting profiles; the method found no difficulty in locating the best yield even from the most remote starting profile, although more steps were required. Again the final profile in each case was substantially that already shown for the 400°C start.

The performance of the conjugate gradient method from these starting profiles is summarized in Table 4.14.

This performance tended to be unpredictable, a feature largely attributable to the inadequacy of the scalar factor β of equation (4.7) in the determination of the appropriate search directions in regions at some distance from the optimum. As an example, the first two norms calculated during computations for the isothermal starting profile of 100°C were 0.157×10^{-14} and 0.268×10^{-5} respectively. The value of β contributing to the formation of the second direction of search was, therefore, very large and as a result every temperature along the reactor reached the constraint; subsequent computations made no further progress.

Results from other starting profiles were better but inferior in convergence rate to those obtained using the method of steepest ascent. One exception to this was the result obtained starting from an isothermal profile of 300°C. Here the factor β was presumably more appropriate and a good result was obtained in terms of an appreciably higher convergence rate compared with the corresponding result using the method of steepest ascent (see Tables 4.13 and 4.14). Overall, however, for the reasons already discussed, the linear search technique of this section has produced better results in conjunction with the method of steepest ascent.

Fig. 4.1. Initial Search Trajectories

Fig. 4.2. Variation of Yield Along First Direction of Search (from Ch. 3)

Fig. 4.3. Comparative Growth of Yield; 400°C Isothermal Starting Profile. $\epsilon = 1 \times 10^4$

Fig. 4.4. Comparative Growth of Yield; 400°C Isothermal Starting Profile. $\epsilon = 5 \times 10^3$

Time Per Iteration (Sec.)	2.4	2.0	2.1
$\epsilon (\times 10^{-3})$	Gradient (Chap. 3)	Steepest Ascent	Conjugate Gradient
2.5	76	77	52
5.0	38	39	27
10.0	19	22	---

Table 4.5. Iterations to 0.437 Yield; 400°C Isothermal Starting Profile

Fig. 4.6. Temperature Profile Sensitivity

Fig. 4.7. Effect of Increase in ϵ ; Linear Search with Constant Step

Fig. 4.8. Effect of Further Increase in ϵ

$\epsilon \times 10^{-3}$	Starting Yield	Yield After	
		25 Steps	50 Steps
10	0.357	0.426	0.434
15	0.357	0.434	0.434
50	0.357	0.436	

Table 4.9. Conjugate Gradient Method, Using Linear Search
At Constant Step. Starting Profile 400°C Isothermal

Fig. 4.10. Effect of Increase in the Starting Value of ϵ on Performance

Fig. 4.11. Performance at Large Starting Step; 400°C Isothermal Starting Profile

$\epsilon (\times 10^{-3})$	Starting Yield	Yield After		
		25 Steps	50 Steps	100 Steps
0.001	0.357	0.401	0.426	0.432
0.1	0.357	0.423	0.432	0.432
0.5	0.357	0.429	0.432	
1.0	0.357	0.431	0.432	
5.0	0.357	0.432	0.432	

Table 4.12. Performance of Conjugate Gradient Method with Various Starting Steps in the Linear Search With Step Doubling. 400°C Isothermal Starting Temperature Profile.

Starting Temp ^r . °C	Starting Yield	Final Yield	Number of Steps	Final Yield	Number of Steps
10	0	0.437	77	0.438	82
100	0	0.437	53	0.438	65
200	0.001	0.437	52	0.438	76
300	0.054	0.437	47	0.438	70
400	0.357	0.437	34	0.438	60
500	0.331	0.437	47	0.438	78

Table 4.13. Performance of Steepest Ascent Method From Different Starting Profiles Using a Linear Search With Step Doubling.

Starting Temp ^r . °C	Starting Yield	Yield After		
		25 Steps	50 Steps	100 Steps
100	0	0.005	0.373	0.373
200	0.001	0.408	0.410*	-
300	0.054	0.437	0.437	0.437
400	0.357	0.423	0.432	0.432
500	0.331	0.430	0.433	0.435

*Constraint not reached.

Table 4.14. Performance of Conjugate Gradient Method From Different Starting Profiles Using a Linear Search With Step Doubling.

CHAPTER 5A Non-Gradient Method5.1 Introduction

In this chapter a non-gradient method is applied to the problem specified mathematically in Chapter 3. The difficulties associated with the application of Pontryagin's Maximum Principle to the problem of system optimization have already been discussed in Chapter 2. The procedures necessary to deal with the two-point boundary value problem may be avoided if, for example, sufficient knowledge is available concerning the form of the control variable. In particular, if the control can be represented by a series or function then the performance index in turn will depend on the parameters involved in the selected control variable representation. The problem then becomes one of optimization of the performance index with respect to the parameters and may be tackled by some suitable hill-climbing method.

An obvious drawback is the problem of providing a suitable form for the control; also it is suspected that the use of a direct method such as that proposed here may take longer computationally - especially when the number of parameters involved is large. In the analysis of the problem by the methods of Chapters 3 and 4, the results obtained provide good information as to the form of the control; assuming this was not available, however, there still remains the theoretical reasoning of the falling temperature profile for

optimum yield as mentioned in Chapter 2.

Little computational experience on the solution of problems in optimization using these ideas appears to be available but previous attempts on the problem to be considered here gave some promising results [3].

In that work, which again, as in the earlier work using the gradient method described in Chapter 2, was carried out using a Ferranti Mercury computer, first approximations to the temperature profile along the reactor were by means of linear and quadratic forms. These representations were respectively used in conjunction with an optimizing procedure [16] which continuously modified in effect the intercept, slope or curvature of the form assumed until no further improvement in the yield could be achieved. Combinations of linear and curved forms were also tried and results provided adequate evidence to invite further investigation along these lines.

Such an extension of the earlier work is presented in this chapter using an I.C.T. 1905 computer and Fortran language.

Although the method being used here replaces the problem of optimizing a functional of a continuous function by that of optimizing a function of few parameters only we may think of the latter approximating in some manner to the infinite number of degrees of freedom of the functional as the number of parameters is increased. In any event, a suitable representation with few parameters only may often give results

sufficiently close to those physically likely so as not to merit some more complex arrangement.

5.2 Parameterisation of Control

5.2.1 In the results presented later in this chapter several assumptions were made concerning the form of the temperature profile $T(t)$ along the length of the reactor, some being natural developments dictated by successive computer results.

To summarize, these were:-

- (i) A simple profile in which the temperature varied linearly from inlet to outlet. In this form the temperature variation with time t along the length of the reactor was given by

$$T = a_2 - a_1 t \quad (5.1)$$

a_1 and a_2 being the two parameters involved.

This type of profile was selected as the first in order to provide a check on the computation by comparison with previous results.

- (ii) A profile in which the temperature was constant over a variable region at the inlet followed by an exponential reduction to the outlet. This combination was possible in terms of three parameters a_1 , a_2 and a_3 where

$$T = a_1, \quad t \leq a_2$$

and

$$T = a_1 e^{a_3(t-a_2)}, \quad t > a_2 \quad (5.2)$$

Here a_2 was the variable point of intersection of the two forms. This representation had one

parameter less than a similar one tried previously and should, therefore, considerably reduce computing time.

- (iii) A profile consisting of the combination of three straight lines, two having variable slope; with the addition of a variable intercept at the inlet and two variable points of intersection the number of parameters was now increased to five. In this form, the temperature T was given by the set of expressions

$$T = a_1, \quad t \leq a_2$$

$$T = a_1 + a_2 a_3 - a_3 t, \quad a_2 < t \leq a_4 \quad (5.3)$$

$$T = a_1 + a_2 a_3 + a_4 (a_5 - a_3) - a_5 t, \quad t > a_4.$$

Here a_1 is the intercept, a_2 and a_4 the variable points of intersection, and a_3 and a_5 the variable slopes of two of the lines.

- (iv) A form was then considered in which the discrete temperatures at selected points along the reactor were themselves chosen to be the parameters involved. One difficulty here was the compromise necessary between burdening the optimizing routine with a large number of parameters as variables and having an inadequate description of the temperature profile with too few.

In this representation then

$$T = a_i, \quad (i = 1, \dots, n) \quad (5.4)$$

- (v) Finally, a profile was assumed in which the temperature was represented by a combination

of discrete temperatures at selected points together with a linear form along the remaining portion of the reactor. This type of profile was suggested largely on the basis of results from (iv) and was considered a convenient method of reducing the number of parameters involved and yet retaining an adequate temperature representation. The discrete temperatures were selected over the inlet region where the temperature varied considerably and with this choice the temperatures for a contact time of 1 sec. were given by the expressions

$$\begin{aligned}
 T &= a_i, & (i = 1, \dots, m-1), & \quad t < x \\
 T &= \frac{(a_m - a_{m-1})t + a_{m-1} - xa_m}{1-x}, & \quad t \geq x
 \end{aligned}
 \tag{5.5}$$

Here a_m was the temperature at the outlet and it was possible to choose the value of m considerably less than n in (iv). x was the point of division between the discrete and linear forms and this was varied for effect outside of the optimizing routine.

5.2.2 Having selected a profile of the type specified in equations (5.1) to (5.5), a starting temperature profile is obtained by initialising the parameters involved. The state equations (3.0) for the system may then be integrated in the forward direction to produce the corresponding yield $x_3(t_f)$ and for this both the trapezoidal and Runge Kutta methods were used as

detailed in Section 5.5. The next and subsequent temperature profiles (and corresponding yields) are then determined by the optimizing routine which varies the parameters involved in a specific manner (see Section 5.3). A full description of the computational procedure is given in Section 5.4.

One significant advantage implicit in the method of control parameterisation is the convenient manner in which a variable final time may be considered. In the problem being considered here this represents the length of the reactor or contact time and may be relatively easily incorporated by means of an additional parameter which is adjusted, with the other parameters involved, by the optimizing routine. This additional parameter was included in two of the parametric control forms considered.

5.3 Optimizing Routine

The well-known optimizing technique due to Rosenbrock, which incorporates the facility for constraining the parameters being varied, was used. This facility was necessary in the problem being considered here since at all times the temperatures along the reactor were constrained to be $\leq 550^{\circ}\text{C}$, this implies in turn constraints on the parameters in the form assumed for the temperature profile.

The technique used is described fully in [16] and [3] and a brief description only is given here.

For a function F of n variables, n orthogonal unit vector directions $\underline{\xi}_1, \underline{\xi}_2, \dots, \underline{\xi}_n$ are selected rather

than a single direction in which to progress at each stage. An arbitrary step is then taken in the direction ξ_1 and the new value of F calculated. If a maximum is being sought and this new value is greater than or equal to the value of F at the point from which the step was made, this is classed as a success and the step length is multiplied by 3 for future use in this direction. Otherwise we have a failure and the step is multiplied by $-\frac{1}{2}$. These step lengths represent the changes which have occurred in the parameter being varied and in the event of a success the parameter and the value of F are retained at their current values; a failure results in the values being retained at the levels before the step was made.

This process is repeated for each vector direction ξ_r (corresponding to changes in the r^{th} parameter) and the whole process recycles until at least one step has been successful and one has failed in each direction used.

When this situation has been reached a new set of orthogonal directions $\xi_1, \xi_2, \dots, \xi_n$ are computed making use of current information in the Gram Schmidt procedure [16] and the complete process repeated.

A description of the behaviour of the routine when any of the parameters involved reaches a constraint is also given in [3] and [16].

A modification of Rosenbrock's method which includes a univariate search along each direction ξ_i to locate the maximum (or minimum) has been developed by Davies, Swann and Campey and described by Swann [11]. A review of

other direct methods of optimization, including the latter, has been given by Fletcher [15], but none of the methods he summarizes allow for the inclusion of constraints on the variables involved. Much more recently Powell's survey of numerical methods for unconstrained optimization [41] gives an account of some of the more successful techniques and indicates likely future developments in this field.

5.4 Computational Procedure

5.4.1 For this particular application it was possible to utilise an existing subroutine available on the 1905 computer system. Such a subroutine searches for the maximum (or minimum) value of a function within a specified region according to the technique described briefly in Section 5.3. It was, however, necessary to write separate subroutines to evaluate the function being optimized (in this case the yield) and also a main programme to link the various subroutines together. The computational procedure which follows describes briefly the logical steps involved for the general application.

- 5.4.2 (i) A form for the control variable is first assumed and an arbitrary starting value is set up by initialising the parameters involved.
- (ii) The state differential equations for the system may now be integrated in the direction of increasing time and the performance index evaluated.
- (iii) Entry is now made to the hill-climbing subroutine

which systematically modifies the parameters involved in the form assumed for the control.

(See Section 5.3.)

(iv) For each parameter change in (iii) a corresponding value for the control variable is produced and step (ii) is repeated.

(v) Stages (ii) and (iii) are continued until a pre-defined number of parameter changes has been reached at which point the computation terminates.

5.4.3 For the specific problem being considered here the state equations for the system are those of equation (3.0) in Chapter 3, and the forms of control to be assumed are those already described in equations (5.1)-(5.5) of Section 5.2. For each of these forms the subroutine which integrates the state differential equations and computes the yield may be found in Appendix C. In addition a main programme is necessary to link any of these subroutines to the hill-climbing subroutine to which they are complementary. This main programme (plus subroutine, plus optimizing routine) is shown for the linear control form given by equation (5.1). The main programmes for the other subroutines are similar and are not repeated. A flow diagram showing the typical calculations for the yield and the interaction between the main programme and subroutines is also shown in Appendix C1 for the control form of equation (5.5) with variable final time.

5.5 Results and Discussion

This section contains a selection of some of the main

results obtained which are presented in the form of tables and graphs at the end of the chapter. In addition to investigating the suitability of each assumed form for the profile from the points of view of yield obtained and convergence rate, consideration is also given to the relative performances for several different starting profiles and the variability of the contact time or reactor length. Both the Runge Kutta and Trapezoidal methods were used for integrating the state equations as indicated in the following discussion, and a check was carried out on the variation of results with choice of numerical integration step length. Comparison is made, where possible, with results from methods applied and discussed in earlier chapters.

5.5.1 Preliminary Computations

As a check on the programming and correct use of the subroutine the first temperature profile assumed was the linear form given by equation (5.1), the two parameters involved being the slope and intercept. The main programme and subroutine necessary to evaluate the yield in this case are shown in Appendix C2. The trapezoidal rule with 100 steps was used for the integration of the state equations and the starting profile was one in which the temperature fell from 527°C at the inlet to 277°C at the reactor outlet. These assumptions were exactly as those considered in [3], and the final yield after 100 iterations (i.e. parameter changes) of 42.57% corresponding to an

inlet temperature of 550°C falling linearly to a temperature of 394°C at the outlet was identical with that achieved earlier on the Mercury computer. The time taken, however, was considerably less, being 56 secs. in execution compared with some 18 mins. on the Ferranti machine; this is indeed some measure of the rapid advance in computer technology.

The optimizing routine was continued to a total of 400 iterations but no further improvement in yield was produced and only a slight temperature change at the outlet. With a contact time increased to one sec. the comparative yield after 100 iterations was 43.08% and a drop in temperature to 210°C had been sustained at the outlet. These trends are consistent with those observed using the gradient method in Chapter 3.

The same programme was run with starting profiles of 100°C and 400°C (a standard starting profile for the methods used in Chapters 3 and 4); for each a final time of 1 sec. and 160 integration steps were used. After approximately 200 iterations the former had converged and in the latter changes taking place were extremely small. The comparative results are summarised in Table 5.1 in which the evidence suggests that the best yield and best linear profile have been reached for the contact time of 1 sec. considered. Fig. 5.2 shows rates of increase of yield; the 400°C isothermal start is seen to be the most difficult for the optimizing routine.

5.5.2 Second Assumption

Next, the combination profile given by equation (5.2)

was considered consisting of a straight line of zero slope at the inlet and an exponential fall-away towards the outlet. The number of parameters involved was three and the subroutine necessary to evaluate the yield for this assumed control form is shown in Appendix C3. The main programme necessary to initialise the data is similar to that for Section 5.5.2 and is not repeated. 150 iterations starting from an isothermal profile of 400°C with a final time of 1 sec. produced a yield of 40.07% with the straight line portion of the profile at 443°C . After 400 iterations the yield was 43.29% and the linear portion of the profile had reached its constraint at 550°C ; no further change in profile or yield was observed up to 600 iterations. The trapezoidal rule was used for integrating the state differential equations with 160 steps and then 80 steps; a check on the output at 150 iterations indicated no significant loss of accuracy at the larger step length. This point was also apparent in the work using the gradient method of Chapter 3, when a reduction to 40 steps with the trapezoidal rule resulted in a loss of accuracy. The best profile of this type is shown in Fig. 5.3; the time taken per iteration at the 80 step integration level was 0.95 sec. The comparison with the best linear profile is seen to be quite close.

5.5.3 Variable Final Time

The final time t_f was introduced as an additional parameter in the scheme of Section 5.5.2. The subroutine

necessary to evaluate the yield was also rewritten to include the fourth order Runge Kutta integration technique so that a larger integration step could be used with consequent reduction in the otherwise excessive computation time at the larger values of t_f . This subroutine, which is shown in Appendix C4, was tested down to an integration step size of 0.05 at a final time of 1 sec. The results after 600 iterations with 80 and 40 steps gave an excellent comparison with those previously found using the trapezoidal rule with 160 (and 80) steps; at the Runge Kutta 20 step level there was a slight difference in yield with identical profile. In a 30 minute computer run commencing from the 400°C isothermal profile and using 40 integration steps a yield of 43.33 was achieved at an apparently stabilised final time of approximately $14\frac{1}{2}$ sec. The tendency, as in the results of Chapter 3, is for there to be only a relatively small increase in yield at substantially large increase of contact time. A summary of the progress in the computer run is given in Table 5.5.

5.54 Third Assumption

The next stage was to retain the linear portion but to replace the exponential reduction by two separate straight lines, thus representing the profile by a combination of three straight lines involving a total of five parameters. The subroutine is shown in Appendix C5. The results obtained indicated that this combination was rather ineffective. At a starting

profile of 400°C isothermal, with the variable points of intersection located at $t = 0.25$ sec. and $t = 0.5$ sec. and a final time of 1 sec., the starting yield was 35.7%. An outstanding feature of the computer results was the manner in which the hill-climbing routine insisted on reducing the three-line combination to one. (This had previously been experienced with a profile consisting of two straight lines [3].) Thus after 250 iterations (i.e. 50 changes in each parameter) when the yield had reached 40.97%, the inlet temperature had reached 500°C and the three line combination had been virtually reduced to two (see Fig. 5.4). After a further 200 iterations, during which time the inlet temperature remained practically constant, the yield had increased to 41.43% and the hill-climbing routine had succeeded in further reducing the combination to one line, - the points of intersection now being at $t = 0.35 \times 10^{-3}$ and $t = 0.9998$ respectively.

At 1500 iterations a yield of 41.53% had been achieved with the inlet temperature still not at its maximum and the outlet at 350°C, - the system had apparently converged at this stage.

It would appear on the basis of these results that no significant benefits can be produced by this more complex linear combination compared with the simple linear form.

5.5.5 Fourth Assumption

In all of the previous assumptions regarding the form of the control each analysis resulted in a final state

approximating to the best control of a particular type - thus in Section 5.5.1 the best linear profile was produced for a fixed contact time. It was considered at this stage that, having experimented with perhaps certain slightly restrictive forms, some more flexible scheme might be appropriate. Consideration was, therefore, given to the representation of the profile by a set of discrete temperatures equidistant at points along the reactor - the temperature between each pair remaining constant. The number of temperatures was chosen as twenty so as to provide a reasonably adequate representation along the reactor and to avoid placing too much strain on the optimizing routine. The subroutine is shown in Appendix C6. The starting yield was 35.7% with each temperature at 400°C and a contact time of 1 sec. After 400 iterations (i.e. 20 changes of each parameter) the yield had increased to 43.48% and the leading temperature had reached the 550°C constraint. The time taken for this operation was 0.25 sec. per iteration using Runge Kutta with twenty steps (greater speed with marginally less accuracy) and the temperature profile had progressed to the form shown in Fig. 5.6. After 2000 iterations the yield had reached 43.54% and at 4000 only a further 0.07 had been gained. Fig. 5.6 indicates the corresponding changes in temperature profile which accompanied these small additional increases in yield. It is to be seen that some 5-10% of the profile at the inlet progressively reaches the constraint while the profiles as a whole

generally approach the type obtained in the results of Chapters 3 and 4 - a tendency to be expected as the number of iterations increases.

It was noticed during the later iterations that the hill-climbing routine, having effectively stabilised the inlet and centre portion of the reactor, produced relatively large variations in the prevailing temperatures towards the outlet with no effective change in yield. The outlet temperatures, which were reasonably linear, appear to have relatively less influence on yield than the more forward parts of the reactor - a tendency which was also noted during application of the methods of Chapters 3 and 4 and in the earlier work [3]. Apart from this, it was also apparent that the hill-climbing routine had not yet produced convergence, although increases were only small, and that after good progress in the early stages the number of parameters was proving rather burdensome.

5.5.6 Final Assumption

In view of the preceding observations it was decided that a course of action warranting investigation was to retain in essence the scheme of Section 5.5.5 but to reduce the number of parameters.

Accordingly, a form was chosen which retained the discrete temperature over the more sensitive inlet region and replaced those towards the outlet by a linear form. The point at which to separate the two regions was not obvious and it was, therefore, decided to make

this position a variable one, several choices for it to be made as the analysis proceeded. This is then the scheme given by equation (5.5). With a separation point at $t = 0.4$ the number of variables required to be handled had been reduced to 10, 9 of which were discrete temperatures situated at equidistant points over the inlet 40% of the reactor, the remaining parameter being the outlet temperature. There was thus no loss of representation over the sensitive inlet region.

Appendix C7 contains the necessary subroutine. The various combinations considered, which were all tested under identical conditions, are summarized in Table 5.7 which includes the yields after approximately 500 iterations and the further progress made by two of the combinations after a substantially larger number of iterations. The Runge Kutta method was again used to integrate the kinetic equations and tested with 20, 40, 80 and 160 integration steps for the form involving 5 parameters over the contact time of 1 sec; the results shown are those using 40 steps which gave good accuracy compared with that at the finer integration interval confirming the conclusion of Chapter 3 on this point. The time taken at this integration interval was approximately 0.44 sec. per iteration. It will be seen that all results are only marginally different at the later stages but all improve on the situation of Section 5.5.5.

The comparative profiles are illustrated in Fig. 5.8, and the variation of yield during the early iterations

is shown in Fig. 5.10, the growth for the representation involving fewest parameters being particularly rapid. Further results were then obtained for the configuration involving 5 variables from different starting profiles. With a 300°C isothermal start the initial yield was 5.43% increasing to 43.52% after 1000 iterations (i.e. 200 changes per variable). From a profile isothermal at 100°C the initial yield was effectively zero and rose to 43.67% again after 1000 iterations. The time taken for each run was 7.3 mins. and in neither case had the process converged although at the stage considered the inlet temperatures had been constrained as in the corresponding results starting from 400°C isothermal. The profiles for all three cases were very similar - that for the 400°C start having previously been shown in Fig. 5.8. A computer run commencing from a 10°C isothermal profile yielded 43.54% after 1000 iterations - the leading temperature was, however, very low and the hill-climbing routine was experiencing some difficulty in this case. Continuation to 2500 iterations failed to improve on this situation.

Overall, this form assumed for the control has allowed a reduction in the number of variables to be handled and has produced reasonably consistent results comparable to those produced by the methods of Chapters 3 and 4.

5.5.7 Variable Final Time

This was also investigated for the form assumed in Section 5.5.3. As a further study of the effect of varying the final time in order to improve the yield,

the scheme involving fewest parameters was adjusted to include the final time t_f as an additional parameter. The results of a computer run commencing from the 400°C isothermal starting profile are summarised in Table 5.9. The situation has apparently converged at a final time of 5.1 sec. where the yield is 44.4%. This result bears comparison with the figures obtained in Chapter 3 and shown in Fig. 3.7.

A flow diagram for the computations necessary in this case is shown in Appendix C1; the subroutine to evaluate the yield is only slightly different to that shown in Appendix C7.

5.5.8 It is of interest to note that although in the majority of cases considered the yields obtained are similar in value, the temperature profiles by comparison can differ substantially. The suggestion also is that the best contact time can vary with the type of profile assumed. These points may perhaps be explained by the fact that the hill-climbing technique is forced to act within the framework set up by the assumed form of control and should, therefore, theoretically give the best control of a type. If the form assumed is substantially flexible then one obviously has more chance of attaining the best overall control form. This is evident for example in the results obtained for the discrete temperature and discrete-linear combination forms of profile which tend to approach reasonably closely to the yields and more general final profile produced by the methods of Chapters 3 and 4.

Start			After 100 Iterations		
Inlet °C	Outlet °C	Yield %	Inlet °C	Outlet °C	Yield %
527	27	35.39	550	210	43.08
400	400	35.59	549	230	43.0
100	100	0.4×10^{-6}	550	218	43.06
			After 200 Iterations		
400	400	35.59	550	209	43.07
100	100	0.4×10^{-6}	550	209	43.07

Table 5.1. Linear Profile, 1 sec. Contact Time

Fig. 5.2. Rates of Increase of Yield; Linear Profile, 1 sec. Contact Time.

Fig. 5.3. Comparison of Final Profiles; 400°C Isothermal Starting Temperatures, 1 sec. Contact Time

Fig. 5.4. Progressive Changes in Profile—Three Straight Line Assumption. Starting Profile 400°C Isothermal.

No. Iterations	Linear Parameter	Separation Point	Exponential Parameter	Final Time	Yield
0	673.0	0	0	1.0	0.3568
35	691.2	0.288	-0.04	1.5	0.3920
68	694.9	0.543	-0.18	2.4	0.3989
92	698.2	0.576	-0.23	2.8	0.4000
112	712.3	0.400	-0.24	3.9	0.4052
134	741.3	0.233	-0.32	6.5	0.4137
150	757.7	0.143	-0.31	8.1	0.4190
178	773.8	0.088	-0.35	9.7	0.4231
189	794.4	0.049	-0.40	11.7	0.4276
205	806.1	0.028	-0.44	12.8	0.4300
253	822.6	0.025	-0.54	14.5	0.4332
343-708	822.9	0.025	-0.54	14.5	0.4333

Table 5.5. Effect of Variable Final Time; Exponential/Linear
Combination

Fig. 5.6. Progressive Changes in Profile - Discrete Temperatures Assumption. Starting Profile 400°C Isothermal, Contact Time 1 sec.

Combination					
(1)	(2)	(3)	(4)	(5)	
87	72	58	45	38	Iterations
41.94	41.52	41.61	41.76	41.80	Yield
157	133	130	108	156	Iterations
43.47	43.43	43.54	43.61	43.65	Yield
252	276	221	259	276	Iterations
43.68	43.73	43.67	43.65	43.65	Yield
315	360	300	305	359	Iterations
43.71	43.74	43.67	43.65	43.66	Yield
500	500	500	500	500	Iterations
43.71	43.74	43.67	43.65	43.71	Yield
	2500			1000	Iterations
	43.74			43.74	Yield

Table 5.7. Comparative Yields; Discrete/Linear Combinations.
Starting Yield 35.68 at 400°C Isothermal Starting Profile.
Contact Time = 1 sec.

Fig. 5.8. Comparative Temperature Profiles; Discrete/Linear Combinations.
400°C Isothermal Starting Profile.

No. Iterations	Inlet Temp.	Temp. at 5%	Outlet Temp.	Final Time	Yield
0	400	400	400	1.0	0.3568
45	509	509	388	0.95	0.4219
104	549	547	361	0.95	0.4362
172	550	544	358	1.23	0.4376
315	550	549	358	1.30	0.4382
653	550	549	356	1.30	0.4384
802	550	550	353	1.60	0.4389
1815	550	549	348	1.69	0.4395
1860	542	533	305	4.94	0.4417
1902	550	533	304	4.69	0.4436
2061 -3833	550	534	304	5.15	0.4441

Table 5.9. Effect of Variable Final Time. Discrete/Linear Combination With 5 Variables.

Fig. 5.10. Initial Variation of Yield; Discrete/Linear Combinations. 400°C Isothermal Starting Profile.

Chapter 6Comparison of the Methods

The computational study presented in this thesis had two objectives. The first of these was to provide a practical solution to a fairly complex problem. The second aimed at using the problem as a test vehicle in order to compare relative merits of the selected methods of solution. A considerable amount of quantitative information has already been presented, summarised and discussed at length in Chapters 3, 4 and 5, and in this chapter a qualitative assessment of the methods used on the problem of Chapter 3 is given.

6.1 Solution of the Problem

Solutions to the problem specified mathematically in Chapter 3 were obtained by each of the four methods described in Chapters 3, 4 and 5.

Among these solutions were consistent results which had reached the same converged state from several different starting configurations using different methods. Typically a yield of 43.8% was such a result with an associated contact time of 1 sec. and a temperature profile constrained at 550°C over some 12-13% of the reactor's length at the inlet, and falling to 350°C at the outlet. This yield was the best achieved but increases of the order of 1% in the value were obtained by substantial increase in contact time and associated changes in temperature profile. Whether such increases would be worthwhile in practice

would depend on many factors.

Also among the results were those which were considered to be the best obtainable consistent with the restrictions implied by the method used - the reference here is to the conclusions of Chapter 5. In that chapter the method gave results based on assumed forms for the temperature profile - several of which would perhaps be relatively easier to produce in practice, which is an important consideration. It is true to say that in many engineering problems the production of the precise mathematical optimum, even if this is possible, may often be a misplaced objective from a practical viewpoint. Often the desired attainment of this ultimate can, with present techniques, lead to difficult and substantial programming or excessive computation or perhaps both. Inspection of the computational results from all the methods used in this study indicated that in many cases sub-optimal solutions, differing only marginally from the best achieved, were more readily available; of course this was evident in this problem only because the optimum had been fairly well established. In addition these solutions proved to be considerably less sensitive to variations in temperature profile than the best achieved and this again is important when trying to assess what can be achieved from a practical standpoint.

6.2 Performance of the Methods

Several aspects are involved in the assessment of the performance of each method when applied to the problem

considered in this work. Application of the methods to other problems has already been mentioned in the previous chapters.

Important considerations, which are included in the following summary of the computational merits of each method, are:-

- (i) ease of programming,
- (ii) numerical techniques required,
- (iii) convergence rate,
- (iv) versatility.

6.2.1 Method (1). Gradient Method

The method was relatively easy to programme and required the least storage compared with the other gradient techniques (see for example Appendices A and B).

A numerical technique was required to integrate the state and adjoint differential equations as in methods 2 and 3 to follow. These techniques are reasonably well established and free choice here allowed considerable reductions in computing time. Of three techniques tried, namely Trapezoidal formula, Runge Kutta fourth order method and the Hamming predictor-corrector technique, the Runge Kutta proved best. Typically, computation using this method was possible with an integration step some four times the size of that required using the Trapezoidal formula to obtain similar accuracy. A detailed comparison of the performance using these three techniques is given in Chapter 3. The penalty in terms of additional programming for the more accurate methods is indicated by inspection of the programmes in

Appendix A.

The best yield using this method was consistently 43.8% and the associated temperature profile is shown in Fig. 3.2. In attaining this yield the proportion of the inlet temperatures under constraint was critical. The characteristics of the problem solved by this method made increases in search step sizes tolerable with associated reductions in computing time. With larger steps the same yield and profile were recovered in as few as 10 iterations and the method produced the fastest solution of all.

Predictably, for this problem, the method was not a good one when used from starting conditions remote from the optimum, nor was it convenient to incorporate variable contact time within the programme although interesting trends were established by varying this quantity separately.

6.22 Method (2). Steepest Ascent

This involved substantially more programming (see Appendix B) because of the linear search technique required as part of the method; an existing subroutine for this would have been useful here.

Two types of linear search were investigated, one at constant step, the other with step-doubling, although both used a quadratic fit to the local maximum. The second of these was faster for the problem being considered.

Results obtained using method (1) were confirmed, not

only from identical starting conditions, but also from a wide range of other starting configurations, including those remote from the optimum. An indication of the versatility of the method applied to this problem may be obtained by inspection of the results presented in Chapter 4. Comments regarding the effects of variable contact time are similar to those of method (1). Again, as in the analysis by method (1), a considerable reduction in computing time was achieved by increase of step size in both types of linear search, although this reduction was greater when the linear search with step-doubling was being used. The computing time to the optimum configuration using either of the linear searches was less than in method (1) under almost all the conditions considered.

6.2.3 Method (3). Conjugate Gradient Method.

A further numerical technique was necessary here to estimate the norm of each gradient trajectory. Both Simpson's rule and a formula of higher accuracy were used, the former being generally sufficiently accurate provided an appropriate integration step was used. The latter formula allowed a reduction in computing time through a coarser integration step and was essential where gradient components were very small. The additional programming required for either numerical technique was very small and the overall programming required was, therefore, similar to that in method (2). This programme required the most storage and there was a small increase in computing time per iteration over

that required for method (2).

The method worked well in conjunction with the linear search at constant step so long as a relatively small step was used. Results at this stage were produced more quickly than any other method working under comparable conditions.

With large search steps and also when using the search with step-doubling from a nominal starting size the method brought a greater proportion of the temperatures at the reactor inlet under constraint early in the computation with a correspondingly lower yield. This again emphasises the delicate relationship between the proportion of inlet region under constraint and the yield in the optimum state.

A factor detracting rather from the overall performance of this method was the rather approximate nature of the method for determining directions of search except when in the vicinity of the optimum. Away from this vicinity directions generated were often inferior to method (2) and, at distances remote from the optimum, perhaps not even directions of ascent. Starting from these latter regions the method was often quite unworkable.

6.2.4 Method (4). Non-gradient Method

In this method, which made no use of the Maximum Principle, the control variable was parameterised and the optimum configuration investigated by means of the direct search technique described in Chapter 5. An available subroutine for the latter made actual programming minimal and the storage required was least

for all methods although approximating to that required in method (1) using the Trapezoidal Rule. The numerical technique for integration of the differential equations was now only necessary for the state equations.

Computation under various conditions confirmed that computer time was greater than that required in methods (1), (2) and (3).

Several distinct assumptions as to the form to be taken by the control variable led to different final temperature profiles with associated differing yields lower in value than those achieved earlier by the other methods. However, these earlier results were again confirmed when other initial assumptions gave a final temperature profile approximating to those realised by methods (1), (2) and (3).

The method worked reasonably well from several different starting profiles but some difficulty was experienced starting from profiles remote from the optimum. A useful feature of this method is the ease with which variable final time (contact time) may be incorporated as another parameter even though considerable additional computation is necessary to produce results. These results suggested that different temperature profiles determined different final or contact times.

6.3 Conclusion

In this work all of the methods used produced consistent results, and each had certain advantages to offer, but none exhibited overall superiority to the others.

Quantitative comparison of the computational results obtained suggests that the Maximum Principle in conjunction with the method of steepest ascent performed best on average for this problem, but it is obviously unwise to generalise on this conclusion with respect to other problems.

The characteristics of the problem considered here made it possible, by various means, to considerably reduce computing times. One likely generalisation, whatever the characteristics of other problems, is that computing times can be reduced by appropriate choices of optimization and numerical techniques. The uncertainties involved in these choices will be lessened as computational experience grows.

The possibility of a multi-purpose computer programme to deal with problems of system optimization has already been explored [42], but it is too early to comment on the success or otherwise of this project.

APPENDIX A

Supplementary to Chapter 3

APPENDIX A1

General Flow Diagram for the Gradient Method

(122)

APPENDIX A2

Computer Programme for Gradient Method with Integration of State
and Adjoint Differential Equations by Trapezoidal Rule.

```

MASTER FSPACE
C   OPTIMAL TEMPERATURE PROFILES
C   GRADIENT METHOD IN FUNCTION SPACE
    DIMENSION E(5),C(5),RK1(321),RK2(321),RK3(321),RK4(321),RK5(321),
1   RK(321),CORRT(321),X1(321),X2(321),X3(321),T(321),PSI1(321),
2   PSI2(321),PSI3(321)
    JAL=0
    READ (1,103) NDATA
16  WRITE (4,201)
201 FORMAT(/29H OPTIMAL TEMPERATURE PROFILES //34H GRADIENT METHOD IN
1FUNCTION SPACE // 12H PROGRAMME 2 )
    CALL TIME(SECS)
    WRITE(4,206) SECS
    READ (1,101) EPS,H,M,N
    READ (1,102) CTEMP
    IF(CTEMP) 14,14,15
15  DO 12 I=1,N+1
12  T(I)=CTEMP
    GO TO 13
14  READ (1,109) (T(I),I=1,N+1)
13  READ (1,105) (E(I),I=1,5)
    READ (1,106) (C(I),I=1,5)
    READ (1,107) X1(1),X2(1),X3(1),PSI1(N+1),PSI2(N+1),PSI3(N+1)
    READ (1,108) NSP
    J=0
    DO 1 I=1,5
1   E(I) = E(I)/1.9865
9   DO 2 I=1,N+1
    CORRT(I) = (658.0-T(I))/(658.0*T(I))
    RK1(I) = H/2.0*C(1)*EXP(E(1)*CORRT(I))
    RK2(I) = H/2.0*C(2)*EXP(E(2)*CORRT(I))
    RK3(I) = H/2.0*C(3)*EXP(E(3)*CORRT(I))
    RK4(I) = H/2.0*C(4)*EXP(E(4)*CORRT(I))
    RK5(I) = H/2.0*C(5)*EXP(E(5)*CORRT(I))
2   RK(I) = RK1(I) + RK2(I) + RK3(I)
    DO 3 I=2,N+1
    X1(I) = X1(I-1)*(1.0-RK(I-1))/(1.0+RK(I))
    X2(I) = (X2(I-1)*(1.0-RK4(I-1))+X1(I-1)*RK1(I-1)+X1(I)*RK1(I))
1/(1.0+RK4(I))
    X3(I) = (X3(I-1)*(1.0-RK5(I-1))+X2(I-1)*RK4(I-1)+X2(I)*RK4(I))
1/(1.0+RK5(I))
3   CONTINUE
    DO 7 I=1,N+1,NSP
    T(I)=T(I)-273.0
7   CONTINUE
    WRITE (4,202) (T(I),I=1,N+1,NSP)
    DO 8 I=1,N+1,NSP
8   T(I) = T(I)+273.0

```

(124)

```
WRITE (4,203) (X3(I),I=1,N+1,NSP)
WRITE(4,207)
J=J+1
IF(J-M)18,10,10
18 DO 4 K=0,N-1
  I = N+1-K
  PSI3(I-1)=PSI3(I)*(1.0-RK5(I))/(1.0+RK5(I-1))
  PSI2(I-1)=(PSI2(I)*(1.0-RK4(I))+PSI3(I)*RK4(I)+PSI3(I-1)*
1RK4(I-1))/(1.0+RK4(I-1))
  PSI1(I-1)=(PSI1(I)*(1.0-RK(I))+PSI2(I)*RK1(I)+PSI2(I-1)*R
1K1(I-1))/(1.0+RK(I-1))
4 CONTINUE
DO 5 I=1,N+1
  DHDT = (-RK1(I)*E(1)-RK2(I)*E(2)-RK3(I)*E(3))*PSI1(I)*X1(I)
1+RK1(I)*E(1)*PSI2(I)*X1(I)
  DHDT = (DHDT-RK4(I)*E(4))*(PSI2(I)-PSI3(I))*X2(I)-RK5(I)*E(5
1)*X3(I)*PSI3(I))/(T(I)*T(I))
  DHDT = DHDT/(H/2.0)
  T(I) = T(I) +EPS*DHDT
  IF(T(I)-823.0) 5,5,6
6 T(I)=823.0
5 CONTINUE
GO TO 9
10 WRITE (4,205)
205 FORMAT (//2X,5H TEMP,5X,2HX1,6X,2HX2,6X,2HX3)
DO 11 I=1,N+1
  T(I)= T(I)-273.0
  WRITE (4,204) T(I), X1(I), X2(I), X3(I)
11 CONTINUE
  JAI = JAI + 1
  CALL TIME(SECS)
  WRITE(4,206)SECS
  IF(JAI - NDATA) 16,17,17
101 FORMAT (2F10.0,2I4)
102 FORMAT (F10.0)
103 FORMAT (I4)
105 FORMAT (5F10.0)
106 FORMAT (5F10.0)
107 FORMAT (6F10.0)
108 FORMAT (I4)
109 FORMAT (10F8.0)
202 FORMAT (15F8.1)
203 FORMAT (15F8.3)
204 FORMAT (F8.1,3F8.3)
206 FORMAT(A8)
207 FORMAT(/)
17 STOP
END
```

END OF SEGMENT, LENGTH 1189, NAME FSPACE

FINISH

PROGRAM NAME ZO16,

END OF COMPILATION - NO ERRORS

APPENDIX A3

Computer Programme for Gradient Method with Integration of State
and Adjoint Differential Equations by Runge Kutta Fourth Order Process

```

MASTER FSPACE
C  OPTIMAL TEMPERATURE PROFILES
C  GRADIENT METHOD IN FUNCTION SPACE          RUNGE KUTTA
    DIMENSION E(5),C(5),RK1(321),RK2(321),RK3(321),RK4(321),RK5(321),
1  RK(321),CORRT(321),X1(321),X2(321),X3(321),T(321),PSI1(321),
2  PSI2(321),PSI3(321)
    JAL=0
    READ (1,103) NDATA
16  WRITE (4,201)
201  FORMAT(/29H OPTIMAL TEMPERATURE PROFILES //34H GRADIENT METHOD IN
1  FUNCTION SPACE //12H RUNGE KUTTA /12H PROGRAMME 3 // )
    CALL TIME(SECS)
    WRITE(4,206) SECS
    READ (1,101) EPS,H,M,N
    READ (1,102) CTEMP
    IF(CTEMP) 14,14,15
15  DO 12 I=1,N+1
12  T(I)=CTEMP
    GO TO 13
14  READ (1,109) (T(I),I=1,N+1)
13  READ (1,105) (E(I),I=1,5)
    READ (1,106) (C(I),I=1,5)
    READ (1,107) X1(1),X2(1),X3(1),PSI1(N+1),PSI2(N+1),PSI3(N+1)
    READ (1,108) NSP
    J=0
    DO 1 L=1,5
1  E(L) = E(L)/1.9865
9  DO 2 I=1,N+1
    CORRT(I) = (658.0-T(I))/(658.0*T(I))
    RK1(I) = H/2.0*C(1)*EXP(E(1)*CORRT(I))
    RK2(I) = H/2.0*C(2)*EXP(E(2)*CORRT(I))
    RK3(I) = H/2.0*C(3)*EXP(E(3)*CORRT(I))
    RK4(I) = H/2.0*C(4)*EXP(E(4)*CORRT(I))
    RK5(I) = H/2.0*C(5)*EXP(E(5)*CORRT(I))
2  RK(I) = RK1(I) + RK2(I) + RK3(I)
    DO 3 I=2,N+1
        U1=-RK(I-1)*X1(I-1)*2.0
        V1=(RK1(I-1)*X1(I-1)-RK4(I-1)*X2(I-1))*2.0
        W1=(RK4(I-1)*X2(I-1)-RK5(I-1)*X3(I-1))*2.0
        U2=U1*(1.0-RK(I-1))
        V2=V1*(1.0-RK4(I-1))+RK1(I-1)*U1
        W2=W1*(1.0-RK5(I-1))+RK4(I-1)*V1
        U3=U1-RK(I-1)*U2
        V3=V1-(RK4(I-1)*V2-RK1(I-1)*U2)
        W3=W1-(RK5(I-1)*W2-RK4(I-1)*V2)
        U4=U1-RK(I-1)*U3*2.0
        V4=V1+(RK1(I-1)*U3-RK4(I-1)*V3)*2.0
        W4=W1+(RK4(I-1)*V3-RK5(I-1)*W3)*2.0
        X1(I)=X1(I-1)+(U1+2.0*(U2+U3)+U4)/6.0
        X2(I)=X2(I-1)+(V1+2.0*(V2+V3)+V4)/6.0

```

```

      X3(I)=X3(I-1)+(W1+2.0*(W2+W3)+W4)/6.0
3  CONTINUE
      DO 7 I=1,N+1,NSP
      T(I)=T(I)-273.0
7  CONTINUE
      WRITE (4,202) (T(I),I=1,N+1,NSP)
      DO 8 I=1,N+1,NSP
8  T(I) = T(I)+273.0
      WRITE (4,203) (X3(I),I=1,N+1,NSP)
      WRITE(4,207)
      J=J+1
      IF(J-M)18,10,10
18 DO 4 K=0,N-1
      I = N+1-K
      U1=(RK1(I-1)*PSI2(I)-RK(I-1)*PSI1(I))*2.0
      V1=(RK4(I-1)*(PSI3(I)-PSI2(I))*2.0)
      W1=-RK5(I-1)*PSI3(I)*2.0
      U2=U1*(1.0-RK(I-1))+RK1(I-1)*V1
      V2=V1*(1.0-RK4(I-1))+RK4(I-1)*W1
      W2=W1*(1.0-RK5(I-1))
      U3=U1-(RK(I-1)*U2-RK1(I-1)*V2)
      V3=V1-RK4(I-1)*(V2-W2)
      W3=W1-RK5(I-1)*W2
      U4=U1+(RK1(I-1)*V3-RK(I-1)*U3)*2.0
      V4=V1+RK4(I-1)*(W3-V3)*2.0
      W4=W1-RK5(I-1)*W3*2.0
      PSI1(I-1)=PSI1(I)+(U1+2.0*(U2+U3)+U4)/6.0
      PSI2(I-1)=PSI2(I)+(V1+2.0*(V2+V3)+V4)/6.0
      PSI3(I-1)=PSI3(I)+(W1+2.0*(W2+W3)+W4)/6.0
4  CONTINUE
      DO 5 I=1,N+1
      DHDT = (-RK1(I)*E(1)-RK2(I)*E(2)-RK3(I)*E(3))*PSI1(I)*X1(I)+R
      1K1(I)*E(1)*PSI2(I)*X1(I)
      DHDT = (DHDT-RK4(I)*E(4)*(PSI2(I)-PSI3(I))*X2(I)-RK5(I)*E(5)*
      1X3(I)*PSI3(I))/(T(I)*T(I))
      DHDT = DHDT/(H/2.0)
      T(I) = T(I) +EPS*DHDT
      IF(T(I)-823.0) 5,5,6
6  T(I)=823.0
5  CONTINUE
      GO TO 9
10 WRITE (4,205)
205 FORMAT (//2X,5H TEMP,5X,2HX1,6X,2HX2,6X,2HX3)
      DO 11 I=1,N+1
      T(I)= T(I)-273.0
      WRITE (4,204) T(I), X1(I), X2(I), X3(I)
11 CONTINUE
      JAI = JAI + 1
      CALL TIME(SECS)
      WRITE(4,206)SECS
      IF(JAI - NDATA) 16,17,17
101 FORMAT (2F10.0,2I4)
102 FORMAT (F10.0)
103 FORMAT (I4)
105 FORMAT (5F10.0)

```

(127)

```
106 FORMAT (5F10.0)
107 FORMAT (6F10.0)
108 FORMAT (I4)
109 FORMAT (10F8.0)
202 FORMAT (15F8.1)
203 FORMAT (15F8.3)
204 FORMAT (F8.1,3F8.3)
206 FORMAT(A8)
207 FORMAT(/)
17 STOP
END
```

END OF SEGMENT, LENGTH 1617, NAME FSPACE

FINISH

PROGRAM NAME Z016,

END OF COMPILATION - NO ERRORS

APPENDIX A4

Computer Programme for Gradient Method with Integration of State
and Adjoint Differential Equations by Hamming's Predictor-
Corrector Method

```

MASTER FSPACE
C   OPTIMAL TEMPERATURE PROFILES
C   GRADIENT METHOD IN FUNCTION SPACE
DIMENSION E(5),C(5),RK1(501),RK2(501),RK3(501),RK4(501),RK5(501),
1RK(501),CORRT(501),X1(501),X2(501),X3(501),T(501),PSI1(501),
2PSI2(501),PSI3(501)
JAI=0
READ (1,103) NDATA
16 WRITE (4,201)
201 FORMAT(/29H OPTIMAL TEMPERATURE PROFILES //34H GRADIENT METHOD IN
1FUNCTION SPACE // 40H PROGRAMME 4 HAMMING PREDICTOR CORRECTOR //)
CALL TIME(SECS)
WRITE(4,206) SECS
READ (1,101) EPS,H,M,N
READ (1,102) CTEMP
IF(CTEMP) 14,14,15
15 DO 12 I=1,N+1
12 T(I)=CTEMP
GO TO 13
14 READ (1,109) (T(I),I=1,N+1)
13 READ (1,105) (E(I),I=1,5)
READ (1,106) (C(I),I=1,5)
READ (1,107) X1(1),X2(1),X3(1),PSI1(N+1),PSI2(N+1),PSI3(N+1)
READ (1,108) NSP
J=0
DO 1 L=1,5
1 E(L) = E(L)/1.9865
9 DO 2 I=1,N+1
CORRT(I) = (658.0-T(I))/(658.0*T(I))
RK1(I) = H*C(1)*EXP(E(1)*CORRT(I))
RK2(I) = H*C(2)*EXP(E(2)*CORRT(I))
RK3(I) = H*C(3)*EXP(E(3)*CORRT(I))
RK4(I) = H*C(4)*EXP(E(4)*CORRT(I))
RK5(I) = H*C(5)*EXP(E(5)*CORRT(I))
2 RK(I) = RK1(I) + RK2(I) + RK3(I)
DO 19 I=2,4
U1= -RK(I-1)*X1(I-1)
V1= RK1(I-1)*X1(I-1)- RK4(I-1)*X2(I-1)
W1= RK4(I-1)*X2(I-1)- RK5(I-1)*X3(I-1)
U2= U1*(1.0-RK(I-1)/2.0)
V2= V1*(1.0-RK4(I-1)/2.0) + U1*RK1(I-1)/2.0
W2= W1*(1.0-RK5(I-1)/2.0) + V1*RK4(I-1)/2.0
U3= U1 - U2*RK(I-1)/2.0
V3= V1-V2*RK4(I-1)/2.0+U2*RK1(I-1)/2.0
W3= W1-W2*RK5(I-1)/2.0+V2*RK4(I-1)/2.0
U4= U1-U3*RK(I-1)
V4= V1-V3*RK4(I-1)+ U3*RK1(I-1)
W4= W1-W3*RK5(I-1)+ V3*RK4(I-1)

```

```

X1(I)=X1(I-1)+(U1+2.0*(U2+U3)+U4)/6.0
X2(I)=X2(I-1)+(V1+2.0*(V2+V3)+V4)/6.0
X3(I)=X3(I-1)+(W1+2.0*(W2+W3)+W4)/6.0
19 CONTINUE
F1=4.0/3.0
F3= 9.0/121.0
F2=112.0/121.0
Q1=0.0
Q2=0.0
Q3=0.0
DO 3 I=4,N
  Z1= RK(I-1)*X1(I-1)-2.0*RK(I)*X1(I)
  Z2= 2.0*(RK1(I)*X1(I)-RK4(I)*X2(I))- RK1(I-1)*X1(I-1)+RK4(I-1)*
1X2(I-1)
  Z3= 2.0*(RK4(I)*X2(I)-RK5(I)*X3(I))- RK4(I-1)*X2(I-1)+RK5(I-1)*
1X3(I-1)
  U1=X1(I-3) + F1 * (Z1-2.0*RK(I-2)*X1(I-2))
  U2=X2(I-3) +F1*(Z2+2.0*(RK1(I-2)*X1(I-2)-RK4(I-2)*X2(I-2)))
  U3=X3(I-3) +F1*(Z3+2.0*(RK4(I-2)*X2(I-2)-RK5(I-2)*X3(I-2)))
  V1 = U1- F2*Q1
  V2 = U2- F2*Q2
  V3 = U3- F2*Q3
  W1=0.125*(9.0*X1(I)-X1(I-2)+3.0*(Z1-RK(I+1)*V1))
  W2=0.125*(9.0*X2(I)-X2(I-2)+3.0*(Z2+RK1(I+1)*V1-RK4(I+1)*V2))
  W3=0.125*(9.0*X3(I)-X3(I-2)+3.0*(Z3+RK4(I+1)*V2-RK5(I+1)*V3))
  Q1 = U1 - W1
  Q2 = U2 - W2
  Q3 = U3 - W3
  X1(I+1) = W1 + F3*Q1
  X2(I+1) = W2 + F3*Q2
  X3(I+1) = W3 + F3*Q3
3 CONTINUE
DO 7 I=1,N+1,NSP
  T(I)=T(I)-273.0
7 CONTINUE
  WRITE (4,202) (T(I),I=1,N+1,NSP)
DO 8 I=1,N+1,NSP
8 T(I) = T(I)+273.0
  WRITE (4,203) (X3(I),I=1,N+1,NSP)
  WRITE(4,207)
  J=J+1
  IF(J-M)18,10,10
18 DO 20 K=0,2
  I = N+1-K
  U1 =RK1(I-1)*PSI2(I) - RK(I-1)*PSI1(I)
  V1 =RK4(I-1)*(PSI3(I)-PSI2(I))
  W1 =-RK5(I-1)*PSI3(I)
  U2 =U1*(1.0-RK(I-1)/2.0)+ V1*RK1(I-1)/2.0
  V2 =V1*(1.0-RK4(I-1)/2.0)+W1*RK4(I-1)/2.0
  W2 =W1*(1.0-RK5(I-1)/2.0)
  U3 =U1-(U2*RK(I-1)-V2*RK1(I-1))/2.0
  V3 =V1-(V2-W2)*RK4(I-1)/2.0
  W3 =W1-W2*RK5(I-1)/2.0
  U4 =U1+RK1(I-1)*V3-RK(I-1)*U3
  V4 =V1+(W3-V3)*RK4(I-1)

```



```

W4 = W1 - W3 * RK5(I-1)
PSI1(I-1) = PSI1(I) + (U1 + 2.0 * (U2 + U3) + U4) / 6.0
PSI2(I-1) = PSI2(I) + (V1 + 2.0 * (V2 + V3) + V4) / 6.0
PSI3(I-1) = PSI3(I) + (W1 + 2.0 * (W2 + W3) + W4) / 6.0
20 CONTINUE
Q1 = 0.0
Q2 = 0.0
Q3 = 0.0
DO 4 K = 0, N-4
  I = N-K
  Z1 = 2.0 * (RK(I-2) * PSI1(I-2) - RK1(I-2) * PSI2(I-2)) - RK(I-1) * PSI1(I-1) +
1 RK1(I-1) * PSI2(I-1)
  Z2 = 2.0 * RK4(I-2) * (PSI2(I-2) - PSI3(I-2)) - RK4(I-1) * (PSI2(I-1) -
1 PSI3(I-1))
  Z3 = 2.0 * RK5(I-2) * PSI(I-2) - RK5(I-1) * PSI3(I-1)
  U1 = PSI1(I+1) - F1 * (Z1 + 2.0 * (RK(I) * PSI1(I) - RK1(I) * PSI2(I)))
  U2 = PSI1(I+1) - F1 * (Z2 + 2.0 * (RK4(I) * (PSI2(I) - PSI3(I))))
  U3 = PSI3(I+1) - F1 * (Z3 + 2.0 * RK5(I) * PSI3(I))
  V1 = U1 - F2 * Q1
  V2 = U2 - F2 * Q2
  V3 = U3 - F2 * Q3
  W1 = 0.125 * (9.0 * PSI1(I-2) - PSI1(I) - 3.0 * (Z1 + RK(I-3) * V1 - RK1(I-3) * V2))
  W2 = 0.125 * (9.0 * PSI2(I-2) - PSI2(I) - 3.0 * (Z2 + RK4(I-3) * (V2 - V3)))
  W3 = 0.125 * (9.0 * PSI3(I-2) - PSI3(I) - 3.0 * (Z3 + RK5(I-3) * V3))
  Q1 = U1 - W1
  Q2 = U2 - W2
  Q3 = U3 - W3
  PSI1(I-3) = W1 + F3 * Q1
  PSI2(I-3) = W2 + F3 * Q2
  PSI3(I-3) = W3 + F3 * Q3
4 CONTINUE
DO 5 I = 1, N+1
  DHDT = (-RK1(I) * E(1) - RK2(I) * E(2) - RK3(I) * E(3)) * PSI1(I) * X1(I) + RK1(I) *
1 E(1) * PSI2(I) * X1(I)
  DHDT = (DHDT - RK4(I) * E(4) * (PSI2(I) - PSI3(I)) * X2(I) - RK5(I) * E(5) * X3(I) *
1 PSI3(I)) / (T(I) * T(I))
  DHDT = DHDT / H
  T(I) = T(I) + EPS * DHDT
  IF(T(I) - 823.0) 5, 5, 6
6 T(I) = 823.0
5 CONTINUE
GO TO 9
10 WRITE (4, 205)
205 FORMAT (//2X, 5H TEMP, 5X, 2HX1, 6X, 2HX2, 6X, 2HX3)
DO 11 I = 1, N+1
  T(I) = T(I) - 273.0
  WRITE (4, 204) T(I), X1(I), X2(I), X3(I)
11 CONTINUE
JAI = JAI + 1
CALL TIME(SECS)
WRITE(4, 206) SECS
IF(JAI - NDATA) 16, 17, 17
101 FORMAT (2F10.0, 2I4)
102 FORMAT (F10.0)
103 FORMAT (I4)

```

(131)

```
105 FORMAT (5F10.0)
106 FORMAT (5F10.0)
107 FORMAT (6F10.0)
108 FORMAT (I4)
109 FORMAT (10F8.0)
202 FORMAT (15F8.1)
203 FORMAT (15F8.3)
204 FORMAT (F8.1,3F8.3)
206 FORMAT(A8)
207 FORMAT(/)
17 STOP
END
```

END OF SEGMENT, LENGTH 2716, NAME FSPACE

FINISH

PROGRAM NAME ZO16

END OF COMPILATION - NO ERRORS

APPENDIX A5Runge Kutta Fourth Order Method

This well-known method is described in many books on computing or numerical analysis - for example [29] and [30].

The purpose of this appendix is to show the form of numerical solution when this technique is applied to the state equations (3.0) for the problem of Chapter 3.

The simultaneous non-linear set are:-

$$\frac{dx_1}{dt} = -(k_1 + k_2 + k_3)x_1 = -kx_1$$

$$\frac{dx_2}{dt} = k_1x_1 - k_4x_2$$

$$\frac{dx_3}{dt} = k_4x_2 - k_5x_3$$

with $x_1(0) = 1$, $x_2(0) = 0$, $x_3(0) = 0$.

The kinetic rate constants k_i , $i = 1, \dots, 5$ are evaluated from equation (3.3) of Chapter 3.

The application of the Runge Kutta scheme gives the step-by-step solution for values of $i = 1, 2, \dots, n$:-

$$x_{1,i} = x_{1,i-1} + \frac{1}{6}(u_1 + 2u_2 + 2u_3 + u_4)$$

$$x_{2,i} = x_{2,i-1} + \frac{1}{6}(v_1 + 2v_2 + 2v_3 + v_4)$$

$$x_{3,i} = x_{3,i-1} + \frac{1}{6}(w_1 + 2w_2 + 2w_3 + w_4)$$

where there are n integration steps of width h over the time interval $t = t_f - t_0$. With some manipulation, the quantities involved take the following simplified forms:-

$$u_1 = -h k x_1$$

$$v_1 = h(k_1 x_1 - k_4 x_2)$$

$$w_1 = h(k_4 x_2 - k_5 x_3)$$

$$u_2 = h[-k(x_1 + \frac{1}{2}u_1)] = u_1(1 - \frac{1}{2}hk)$$

$$v_2 = h[k_1(x_1 + \frac{1}{2}u_1) - k_4(x_2 + \frac{1}{2}v_1)] = v_1(1 - \frac{1}{2}hk_4) + \frac{1}{2}hk_1u_1$$

$$w_2 = h[k_4(x_2 + \frac{1}{2}v_1) - k_5(x_3 + \frac{1}{2}w_1)] = w_1(1 - \frac{1}{2}hk_5) + \frac{1}{2}hk_4v_1$$

$$u_3 = h[-k(x_1 + \frac{1}{2}u_2)] = u_1 - \frac{1}{2}hku_2$$

$$v_3 = h[k_1(x_1 + \frac{1}{2}u_2) - k_4(x_2 + \frac{1}{2}v_2)] = v_1 + \frac{1}{2}h(k_1u_2 - k_4v_2)$$

$$w_3 = h[k_4(x_2 + \frac{1}{2}v_2) - k_5(x_3 + \frac{1}{2}w_2)] = w_1 + \frac{1}{2}h(k_4v_2 - k_5w_2)$$

$$u_4 = h[-k(x_1 + u_3)] = u_1 - hku_3$$

$$v_4 = h[k_1(x_1 + u_3) - k_4(x_2 + v_3)] = v_1 + h(k_1u_3 - k_4v_3)$$

$$w_4 = h[k_4(x_2 + v_3) - k_5(x_3 + w_3)] = w_1 + h(k_4v_3 - k_5w_3)$$

where each of x_1, x_2, x_3 is evaluated at stage (i-1)

and each of k_1, k_2, k_3, k_4, k_5 is evaluated at stage (i-1).

The computational scheme for the adjoint equations (3.4) takes a similar form, although the integration is in the direction of decreasing time.

APPENDIX A6Predictor-Corrector Method Due to Hamming

Hamming's method [31] may be summarized by the following sequence of operations required in the step-by-step solution of the differential equation

$$y' \equiv \frac{dy}{dx} = f(x, y)$$

The sequence, per step, is for $n \geq 3$:-

predict: $p_{n+1} = y_{n-3} + \frac{4h}{3}(2y'_n - y'_{n-1} + 2y'_{n-2})$

modify: $m_{n+1} = p_{n+1} - \frac{112}{121}(p_n - c_n)$

correct: $c_{n+1} = \frac{1}{8}[9y_n - y_{n-2} + 3h(m'_{n+1} + 2y'_n - y'_{n-1})]$

final value: $y_{n+1} = c_{n+1} + \frac{9}{121}(p_{n+1} - c_{n+1})$ at $x = x_{n+1}$.

Here h is the integration step length and the procedure has a principal truncation error of order h^6 . Unlike traditional predictor-corrector methods which repeatedly use the corrector formula until no significant changes occur, this modified method is applied only once per step. In his original paper [31], Hamming demonstrates the favourable stability properties of the method. One disadvantage of all multi-step methods is that they are not self-starting. Thus in this case, to estimate y_4 requires the values y_0, y_1, y_2 and y_3 . Assuming y_0 is available from the initial condition, the Runge Kutta method may be used to compute the next three steps. Also, in obtaining m_4 it is usually recommended that the term $p_3 - c_3$ is set to zero.

In the application of the method to the step-by-step solution of the simultaneous, non-linear set of differential

equations given in Appendix 5, it is assumed that this necessary preliminary information has been obtained by the Runge Kutta scheme described there.

Applying Hamming's method to the set with n integration steps of width h we have for successive values of $i = 3, 4, \dots, n$:-

$$\begin{aligned} \text{predict:} \quad p_1 &= x_{1,i-3} + \frac{4h}{3}(u_1 - 2k_{1-2}x_{1,i-2}) \\ p_2 &= x_{2,i-3} + \frac{4h}{3}[u_2 + 2(k_{1,i-2}x_{1,i-2} - k_{4,i-2}x_{2,i-2})] \\ p_3 &= x_{3,i-3} + \frac{4h}{3}[u_3 + 2(k_{4,i-2}x_{2,i-2} - k_{5,i-2}x_{3,i-2})] \end{aligned}$$

$$\begin{aligned} \text{where} \quad u_1 &= k_{i-1}x_{1,i-1} - 2k_i x_{1,i} \\ u_2 &= 2(k_{1,i}x_{1,i} - k_{4,i}x_{2,i}) - k_{1,i-1}x_{1,i-1} + k_{4,i-1}x_{2,i-1} \\ u_3 &= 2(k_{4,i}x_{2,i} - k_{5,i}x_{3,i}) - k_{4,i-1}x_{2,i-1} + k_{5,i-1}x_{3,i-1} \end{aligned}$$

$$\begin{aligned} \text{modify:} \quad m_j &= p_j - \frac{112}{121}q_j \quad j = 1, 2, 3 \\ (q_j &= 0 \text{ initially}) \end{aligned}$$

$$\begin{aligned} \text{then} \quad m'_1 &= -k_{i+1}m_1 \\ m'_2 &= k_{1,i+1}m_1 - k_{4,i+1}m_2 \\ m'_3 &= k_{4,i+1}m_2 - k_{5,i+1}m_3 \end{aligned}$$

$$\text{correct:} \quad c_j = \frac{1}{8}[9x_{j,i} - x_{j,i-2} + 3h(m'_j + u_j)] \quad j = 1, 2, 3$$

$$\text{final value:} \quad x_{j,i+1} = c_j + \frac{9}{121}(p_j - c_j) \quad j = 1, 2, 3$$

This cycle is repeated for each value of i (i.e. each integration step) over the time interval.

For each step after the first,

$$q_j = p_j - c_j \quad j = 1, 2, 3.$$

The computational scheme for the adjoint equations (3.4) takes a similar form.

APPENDIX B

Supplementary to Chapter 4

General Flow Diagram for Combined Gradient, Steepest Ascent and
Conjugate Gradient Methods

(139)

(140)

APPENDIX B2

Combined Programme for Gradient, Steepest Ascent and Conjugate Gradient Methods. Linear Search Performed at Constant Step.

```

MASTER FSPACE
C   OPTIMAL TEMPERATURE PROFILES
C   GRADIENT METHOD IN FUNCTION SPACE
    DIMENSION E(5),C(5),RK1(321),RK2(321),RK3(321),RK4(321),RK5(321),
1   RK(321),CORRT(321),X1(321),X2(321),X3(321),T(321),PSI1(321),
2   PSI2(321),PSI3(321),DHD(321),S(321),UT(321)
    JAL=0
    READ (1,103) NDATA
16  READ (1,108) IA,IB
    IF(IA) 48,48,47
48  WRITE (4,201)
201 FORMAT(/29H OPTIMAL TEMPERATURE PROFILES //34H GRADIENT METHOD IN
1FUNCTION SPACE // 12H PROGRAMME 2 )
    GO TO 53
47  IF(IB) 51,51,52
52  WRITE (4,221)
221 FORMAT(/29H OPTIMAL TEMPERATURE PROFILES //26H CONJUGATE GRADIENT
1METHOD // 12H PROGRAMME 6 )
    GO TO 53
51  WRITE (4,222)
222 FORMAT(/29H OPTIMAL TEMPERATURE PROFILES //16H STEEPEST ASCENT //
112H PROGRAMME 5 )
53  CALL TIME(SECS)
    WRITE(4,206) SECS
    KA =0
    KAY=0
    READ (1,101) EPS,H,M,N
    READ (1,102) CTEMP,VALUE
    IF(CTEMP) 14,14,15
15  DO 12 I=1,N+1
12  T(I)=CTEMP
    GO TO 13
14  READ (1,109) (T(I),I=1,N+1)
13  READ (1,105) (E(I),I=1,5)
    READ (1,106) (C(I),I=1,5)
    READ (1,107) X1(1),X2(1),X3(1),PSI1(N+1),PSI2(N+1),PSI3(N+1)
    READ (1,108) NSP,KAM
    J=0
    C1=1.0
    DO 1 I=1,5
1   E(I) = E(I)/1.9865
    DO 60 I=1,N+1
60  UT(I) = T(I)
    IF(IB) 9,9,42
42  DO 43 I=1,N+1
43  S(I)=0.0
    DO 9 I=1,N+1
    CORRT(I) =(658.0-T(I))/(658.0*T(I))
    RK1(I) = H/2.0*C(1)*EXP(E(1)*CORRT(I))
    RK2(I) = H/2.0*C(2)*EXP(E(2)*CORRT(I))
    RK3(I) = H/2.0*C(3)*EXP(E(3)*CORRT(I))
    RK4(I) = H/2.0*C(4)*EXP(E(4)*CORRT(I))

```

(142)

```

      RK5(I) = H/2.0*C(5)*EXP(E(5)*CORRT(I))
2  RK(I) = RK1(I) + RK2(I) + RK3(I)
      DO 3 I=2,N+1
      X1(I) = X1(I-1)*(1.0-RK(I-1))/(1.0+RK(I))
      X2(I) = (X2(I-1)*(1.0-RK4(I-1))+X1(I-1)*RK1(I-1)+X1(I)*RK1(I))/(1.0
1+RK4(I))
      X3(I) = (X3(I-1)*(1.0-RK5(I-1))+X2(I-1)*RK4(I-1)+X2(I)*RK4(I))/(1.0
1+RK5(I))
3  CONTINUE
      IF(X3(N+1)-VALUE) 65,66,66
66  IF(IB) 67,67,69
67  DO 68 I=1,N+1
68  S(I)=0.0
      IB=1
      GO TO 65
69  IF(IA) 70,70,65
70  KA=0
      IA=1
      J=0
      READ(1,103) M
65  IF(KA) 54,24,24
24  DO 7 I=1,N+1,NSP
      T(I)=T(I)-273.0
7  CONTINUE
      WRITE (4,202) (T(I),I=1,N+1,NSP)
      DO 8 I=1,N+1,NSP
8  T(I) = T(I)+273.0
      WRITE (4,203) (X3(I),I=1,N+1,NSP)
      WRITE(4,207)
      IF(IA) 50,50,46
54  IF(X3(N+1)-XB) 56,55,55
55  KA=0
      KAY=0
      GO TO 24
56  IF(KAY) 57,58,58
57  EPSM = -EPSM
      KAY = 0
      GO TO 25
58  EPSM = -(EPSM+EPS)
      GO TO 25
46  IF(KA-1) 21,22,23
21  XA = X3(N+1)
      WRITE (4,230)
230  FORMAT (// 13H END OF CYCLE /)
      CALL TIME (SECS)
      WRITE (4,206) SECS
      GO TO 50
22  XB = X3(N+1)
      IF(KAY) 36,35,35
35  IF(XB-XA) 32,32,25
23  XC = X3(N+1)
      IF(XC-XB) 28,28,30
28  Z1 = XB-XA
      Z2 = XC-XB
      IF(KAY) 29,26,29
26  EPSM = -0.5*EPSM*(Z1-3.0*Z2)/(Z1-Z2)
```

```

      GO TO 33
30 IF (KOUNT -KAM) 27,31,31
27 XA = XB
   XB = XC
   GO TO 25
31 KA = 0
   WRITE (4,220)
220 FORMAT (/24H STEPS LIMITED IN SEARCH )
   GO TO 21
33 KA = -2
   GO TO 25
32 EPSM = -0.5*EPSM
   XC = XB
   KAY = -1
   KA = KA-1
   GO TO 25
36 EPSM = -EPSM
   IF (XB-XA) 32,32,28
29 IF(J-1) 80,80,81
81 EPS = EPSM
80 EPSM = 0.5*EPSM*(Z1+Z2)/(Z1-Z2)
   GO TO 33
50 J = J+1
   KOUNT = 0
   IF(J-M)18,10,10
18 DO 4 K=0,N-1
   I = N+1-K
   PSI3(I-1)=PSI3(I)*(1.0-RK5(I))/(1.0+RK5(I-1))
   PSI2(I-1)=(PSI2(I)*(1.0-RK4(I))+PSI3(I)*RK4(I)+PSI3(I-1)*RK4(I-1)
1)/((1.0+RK4(I-1))
   PSI1(I-1)=(PSI1(I)*(1.0-RK(I))+PSI2(I)*RK1(I)+PSI2(I-1)*RK1(I-1))
1/((1.0+RK(I-1))
4 CONTINUE
   DO 41 I=1,N+1
   DHDT1=(-RK1(I)*E(1)-RK2(I)*E(2)-RK3(I)*E(3))*PSI1(I)*X1(I)+RK1(I)*
1E(1)*PSI2(I)*X1(I)
   DHDT1=(DHDT1-RK4(I)*E(4)*(PSI2(I)-PSI3(I))*X2(I)-RK5(I)*E(5)*X3(I)
1*PSI3(I))/(T(I)*T(I))
41 DHDT(I) =-DHDT1/(H/2.0)
   EPSM = EPS
   IF(IB) 44,44,37
44 DO 45 I=1,N+1
45 S(I)=DHDT(I)
25 DO 5 I=1,N+1
   UT(I) = UT(I) + EPSM*S(I)
   T(I) = UT(I)
   IF(T(I)-823.0) 5,5,6
6 T(I)=823.0
5 CONTINUE
   KA = KA +1
   KOUNT = KOUNT + 1
   GO TO 9
37 Z1 = 0.0
   Z2 = 0.0
   DO 38 I=1,N/2

```

(144)

```
38 Z1=Z1+DHDT(2*I)*DHDT(2*I)
   DO 39 I=1,N/2-1
39 Z2=Z2+DHDT(2*I+1)*DHDT(2*I+1)
   C2=H/3.0*(DHDT(1)*DHDT(1)+4.0*Z1+2.0*Z2+DHDT(N+1)*DHDT(N+1))
   DO 40 I=1,N+1
40 S(I)=DHDT(I)+ C2*S(I)/C1
   C1=C2
   GO TO 25
10 WRITE (4,205)
205 FORMAT (//2X,5H TEMP,5X,2HX1,6X,2HX2,6X,2HX3)
   DO 11 I=1,N+1
   T(I)= T(I)-273.0
   WRITE (4,204) T(I), X1(I), X2(I), X3(I)
11 CONTINUE
   JAI = JAI + 1
   CALL TIME(SECS)
   WRITE(4,206)SECS
   IF(JAI - NDATA) 16,17,17
101 FORMAT (2F10.0,2I4)
102 FORMAT (2F10.0)
103 FORMAT (I4)
105 FORMAT (5F10.0)
106 FORMAT (5F10.0)
107 FORMAT (6F10.0)
108 FORMAT (2I4)
109 FORMAT (10F8.0)
202 FORMAT (15F8.1)
203 FORMAT (15F8.3)
204 FORMAT (F8.1,3F8.3)
206 FORMAT(A8)
207 FORMAT(/)
17 STOP
   END
```

END OF SEGMENT, LENGTH 1845, NAME FSPACE

FINISH

END OF COMPILATION - NO ERRORS

APPENDIX B3

Combined Programme for Steepest Ascent and Conjugate Gradient
Methods. Linear Search Performed With Step-doubling.

```

MASTER FSPACE
C   OPTIMAL TEMPERATURE PROFILES
C   STEEPEST ASCENT AND CONJUGATE GRADIENT
    DIMENSION E(5),C(5),RK1(161),RK2(161),RK3(161),RK4(161),RK5(161),
    1RK(161),CORRT(161),X1(161),X2(161),X3(161),T(161),PSI1(161),
    2PSI2(161),PSI3(161),DHDT(161),S(161),UT(161),ET(161)
    JAL=0
    READ (1,103) NDATA
16  READ (1,103) IA
    WRITE (4,201)
201  FORMAT (/29H OPTIMAL TEMPERATURE PROFILES //34H LINEAR SEARCH WITH
1DOUBLING STEP // )
    IF(IA) 51,51,52
52  WRITE (4,221)
221  FORMAT (/29H OPTIMAL TEMPERATURE PROFILES //26H CONJUGATE GRADIENT
1METHOD // 12H PROGRAMME 6 )
    GO TO 53
51  WRITE (4,222)
222  FORMAT (/29H OPTIMAL TEMPERATURE PROFILES //16H STEEPEST ASCENT //
112H PROGRAMME 5 )
53  CALL TIME(SECS)
    WRITE(4,206) SECS
    KA = 0
    KAY=0
    KAYO = 0
    READ (1,101) EPS,H,M,N
    READ (1,102) CTEMP
    IF(CTEMP) 14,14,15
15  DO 12 I=1,N+1
12  T(I)=CTEMP
    GO TO 13
14  READ (1,109) (T(I),I=1,N+1)
13  READ (1,105) (E(I),I=1,5)
    READ (1,106) (C(I),I=1,5)
    READ (1,107) X1(1),X2(1),X3(1),PSI1(N+1),PSI2(N+1),PSI3(N+1)
    READ (1,108) NSP,KAM
    J=0
    C1=1.0
    DO 1 I=1,5
1  E(I) = E(I)/1.9865
    DO 60 I=1,N+1
60  UT(I) = T(I)
    IF(IA) 9,9,42
42  DO 43 I=1,N+1
43  S(I)=0.0
9  DO 2 I=1,N+1
    CORRT(I) =(658.0-T(I))/(658.0*T(I))
    RK1(I) = H/2.0*C(1)*EXP(E(1)*CORRT(I))
    RK2(I) = H/2.0*C(2)*EXP(E(2)*CORRT(I))
    RK3(I) = H/2.0*C(3)*EXP(E(3)*CORRT(I))
    RK4(I) = H/2.0*C(4)*EXP(E(4)*CORRT(I))

```



```

      RK5(I) = H/2.0*C(5)*EXP(E(5)*CORRT(I))
2  RK(I) = RK1(I) + RK2(I) + RK3(I)
      DO 3 I=2,N+1
        X1(I) = X1(I-1)*(1.0-RK(I-1))/(1.0+RK(I))
        X2(I) = (X2(I-1)*(1.0-RK4(I-1))+X1(I-1)*RK1(I-1)+X1(I)*RK1(I))/
1      (1.0+RK4(I))
        X3(I) = (X3(I-1)*(1.0-RK5(I-1))+X2(I-1)*RK4(I-1)+X2(I)*RK4(I))/
1      (1.0+RK5(I))
3  CONTINUE
      IF(KA) 54,75,75
75  IF(KAYO) 55,24,24
24  DO 7 I=1,N+1,NSP
      T(I)=T(I)-273.0
7  CONTINUE
      WRITE (4,202) (T(I),I=1,N+1,NSP)
      DO 8 I=1,N+1,NSP
8  T(I) = T(I)+273.0
      WRITE (4,203) (X3(I),I=1,N+1,NSP)
      WRITE(4,207)
      GO TO 46
54  IF(X3(N+1)-XC) 56,55,55
55  KA = 0
      DO 70 I=1,N+1
70  UT(I) = ET(I)
      KAY = 0
      KAYO = 0
      GO TO 24
56  EPSM = REPSM
      KAYO = -1
      GO TO 25
46  IF(KA-1) 21,22,23
21  XA = X3(N+1)
      EPSM = 0.0
      DELTA = EPS
      WRITE (4,230)
230  FORMAT (// 13H END OF CYCLE  /)
      CALL TIME (SECS)
      WRITE (4,206)SECS
      GO TO 50
22  XB = X3(N+1)
      IF(XA-XB) 32,32,80
23  XB = X3(N+1)
      IF(KAY) 62,61,81
62  IF(XB-XC) 28,27,27
61  IF(XB-XC) 28,30,30
81  IF(XB-XC) 82,83,83
32  XC = XB
      IF(KAY) 65,66,66
66  DELTA = 2.0*DELTA
      GO TO 48
65  EPSM = EPSM -DELTA*(XA-XD)/(4.0*(2.0*XC-XA-XD))
      KA = -2
      GO TO 25
80  XC=XA
      XA=XB
      DELTA = -DELTA

```

```

      EPSM = 0.0
      KAY = 1
      GO TO 48
27  XA = XC
      GO TO 32
28  XD = XB
      EPSM = EPSM-0.5*DELTA
      REPSM = EPSM
      IF(KAY) 65,67,67
67  KAY = -1
      GO TO 25
30  IF(KOUNT -KAM) 27,31,31
31  KA = 0
      WRITE (4,220)
220 FORMAT (/24H STEPS LIMITED IN SEARCH )
      GO TO 21
82  XD = XB
      EPSM = 0.0
      DELTA =2.0*DELTA
      REPSM = EPSM
      KAY =-1
      GO TO 65
83  KAY = 0
      GO TO 30
50  J = J+1
      KOUNT = 0
      IF(J-M)18,10,10
18  DO 4 K=0,N-1
      I = N+1-K
      PSI3(I-1)=PSI3(I)*(1.0-RK5(I))/(1.0+RK5(I-1))
      PSI2(I-1)=(PSI2(I)*(1.0-RK4(I))+PSI3(I)*RK4(I)+PSI3(I-1)*RK4(
1I-1))/(1.0+RK4(I-1))
      PSI1(I-1)=(PSI1(I)*(1.0-RK(I))+PSI2(I)*RK1(I)+PSI2(I-1)*RK1(I
1-1))/(1.0+RK(I-1))
4  CONTINUE
      DO 41 I=1,N+1
      DHDT1=(-RK1(I)*E(1)-RK2(I)*E(2)-RK3(I)*E(3))*PSI1(I)*X1(I)+RK1
1(I)*E(1)*PSI2(I)*X1(I)
      DHDT1=(DHDT1-RK4(I)*E(4)*(PSI2(I)-PSI3(I))*X2(I)-RK5(I)*E(5)*X
13(I)*PSI3(I))/(T(I)*T(I))
41  DHDT(I) =-DHDT1/(H/2.0)
      IF(IA) 44,44,37
44  DO 45 I=1,N+1
45  S(I)=DHDT(I)
48  EPSM = EPSM + DELTA
25  DO 5 I=1,N+1
      T(I) = UT(I) + EPSM*S(I)
      ET(I) =T(I)
      IF(T(I)-823.0) 5,5,6
6  T(I)=823.0
5  CONTINUE
      KA = KA +1
      KOUNT = KOUNT + 1
      GO TO 9
37  Z1 = DHDT(1)*DHDT(1)+DHDT(N+1)*DHDT(N+1)
      Z2 = DHDT(2)*DHDT(2)+DHDT(N) *DHDT(N)
      Z3 = DHDT(3)*DHDT(3)+DHDT(N-1)*DHDT(N-1)

```

(148)

```
Z4 = DHDT(4)*DHDT(4)+DHDT(N-2)*DHDT(N-2)
Z5 = DHDT(5)*DHDT(5)+DHDT(N-3)*DHDT(N-3)
DO 91 I = 1, N/8-1
  Z1 = Z1 + 2.0*DHDT(8*I+1)*DHDT(8*I+1)
  Z2 = Z2 + DHDT(8*I)*DHDT(8*I)+DHDT(8*I+2)*DHDT(8*I+2)
91 Z4 = Z4 + DHDT(8*I-2)*DHDT(8*I-2)+DHDT(8*I+4)*DHDT(8*I+4)
DO 92 I = 1, N/4-2
  Z3 = Z3 + DHDT(4*I+3)*DHDT(4*I+3)
DO 93 I = 1, N/8-2
  Z5 = Z5 + DHDT(8*I+5)*DHDT(8*I+5)
C2 = 8.0*H/28350.0*(989.0*Z1+5888.0*Z2-928.0*Z3-4540.0*Z5
  +10496.0*Z4)
WRITE (4,240) C2,C1
240 FORMAT (///5X,5H C2 =,E18.8,5H C1 =,E18.8,///)
DO 40 I = 1, N+1
  40 S(I) = DHDT(I) + C2*S(I)/C1
  C1=C2
GO TO 48
10 WRITE (4,205)
205 FORMAT (//2X,5H TEMP,5X,2HX1,6X,2HX2,6X,2HX3)
DO 11 I = 1, N+1
  T(I) = T(I)-273.0
  WRITE (4,204) T(I), X1(I), X2(I), X3(I)
11 CONTINUE
  JAI = JAI + 1
  CALL TIME(SECS)
  WRITE(4,206)SECS
  IF(JAI - NDATA) 16,17,17
101 FORMAT (2F10.0,2I4)
102 FORMAT (F10.0)
103 FORMAT (I4)
105 FORMAT (5F10.0)
106 FORMAT (5F10.0)
107 FORMAT (6F10.0)
108 FORMAT (2I4)
109 FORMAT (10F8.0)
202 FORMAT (15F8.1)
203 FORMAT (15F8.3)
204 FORMAT (F8.1,3F8.3)
206 FORMAT(A8)
207 FORMAT(/)
17 STOP
END
```

END OF SEGMENT, LENGTH 2151, NAME FSPACE

FINISH

END OF COMPILATION - NO ERRORS

APPENDIX B4Estimation of the Norms

The numerical integration required in the estimation of the scalar factor β of equation (4.22) was performed by use of Simpson's rule and also, when greater accuracy was necessary, by use of an accurate Newton-Cotes formula.

The formula selected may be derived by the methods discussed for example in [Ch.4, 37] or [Ch.3, 40].

Specifically, the formula used here was based upon application of the result

$$\int_{x_0}^{x_8} y dx = \frac{8h}{28350} \left\{ A(y_0 + y_8) + B(y_1 + y_7) + C(y_2 + y_6) + D(y_3 + y_5) + E y_4 \right\}$$

where $A = 989$, $B = 5888$, $C = -928$, $D = 10496$, $E = -4540$

and in which the truncation error is of order h^4 for integration step h .

This may be adapted to any interval of integration provided h is chosen so that the number of integration steps is a multiple of eight; this apparent disadvantage is outweighed by the accuracy achieved.

For a general interval x_0 to x_n , and suitable h , successive applications of the formula over subranges gives, for $n \geq 24$,

$$\int_{x_0}^{x_n} y dx = \frac{8h}{28350} \left\{ A \sum_1 + B \sum_2 + C \sum_3 + D \sum_4 + E \sum_5 \right\}$$

where

$$\sum_1 = y_0 + y_n + 2 \sum_{i=1}^{\frac{n}{8}-1} y_{8i}$$

$$\sum_2 = y_1 + y_{n-1} + \sum_{i=1}^{\frac{n}{8}-1} (y_{8i-1} + y_{8i+1})$$

$$\sum_3 = y_2 + y_{n-2} + \sum_{i=1}^{\frac{n}{4}-2} y_{4i+2}$$

$$\sum_4 = y_3 + y_{n-3} + \sum_{i=1}^{\frac{n}{8}-1} (y_{8i-3} + y_{8i+3})$$

$$\sum_5 = y_4 + y_{n-4} + \sum_{i=1}^{\frac{n}{8}-2} y_{8i+4}$$

(For $n < 24$, appropriate summations are not performed.)

This more general formula is ideally suited to automatic computation.

APPENDIX B5Linear Search at Constant Step

To locate the maximum of the function $X = X(\epsilon)$

set $\epsilon = 0$ calculate $X = X_A$

$\epsilon = \epsilon + \delta$ calculate $X = X_B$

Case (1) or Case (2) is now appropriate

Case (1) $X_B \geq X_A$

Stage (i) $\epsilon = \epsilon + \delta$ calculate $X = X_C$

Stage (ii) if $X_C \geq X_B$

 set $X_A = X_B$, $X_B = X_C$

 Repeat stages (i) and (ii)

 until $X_C < X_B$

Stage (iii) when $X_C < X_B$

 use formula (1) for $\epsilon_{\max}$

 Calculate $X = X(\epsilon_{\max})$

 The larger of X_B and X is then selected.

Case (2) $X_B < X_A$

Stage (i) Set $X_C = X_B$

$\epsilon = \frac{1}{2}\epsilon$ calculate $X = X_B$

Stage (ii) if $X_B \leq X_A$

 Repeat stages (i) and (ii)

 until $X_B > X_A$

(152)

Stage (iii) when $X_B > X_A$

use formula (2) for $\epsilon_{\max}$

Calculate $X = X(\epsilon_{\max})$

The larger of X_B and X is then selected.

Formula (1)

$$\epsilon_{\max} = \epsilon - \frac{\delta}{2} \frac{(z_1 - 3z_2)}{(z_1 - z_2)}$$

Formula (2)

$$\epsilon_{\max} = \frac{(3z_1 - z_2)}{2(z_1 - z_2)} \epsilon$$

where $z_1 = X_B - X_A$

$$z_2 = X_C - X_B$$

Derivation

Assuming quadratic fit

$$X_C = a_0 + a_1\epsilon + a_2\epsilon^2$$

$$X_B = a_0 + a_1(\epsilon - \delta) + a_2(\epsilon - \delta)^2$$

$$X_C = a_0 + a_1(\epsilon - 2\delta) + a_2(\epsilon - 2\delta)^2$$

When $X_B > X_A$, $X_C < X_B$

$X = X(\epsilon)$ has a maximum value at $\epsilon_{\max}$ where

$$\epsilon_{\max} = - \frac{a_1}{2a_2}$$

Defining $z_1 = X_B - X_A$, $z_2 = X_C - X_B$ and eliminating a_1 leads to

$$a_2 = \frac{z_2 - z_1}{2\delta^2}$$

together with

$$a_1 = \frac{\delta(3z_2 - z_1) - 2\epsilon(z_2 - z_1)}{2\delta^2}$$

$$\therefore \epsilon_{\max} = \epsilon - \frac{\delta}{2} \frac{(z_1 - 3z_2)}{(z_1 - z_2)}$$

which is formula (1)

Similarly in Case (2) where X_B is at half step we have

$$a_2 = \frac{z_2 - z_1}{2\epsilon^2}$$

together with

$$a_1 = \frac{3z_1 - z_2}{2\epsilon}$$

$$\therefore \epsilon_{\max} = \frac{(3z_1 - z_2)}{2(z_1 - z_2)} \epsilon$$

which is formula (2).

APPENDIX B6Linear Search With Step Doubling

To locate the maximum of the function $X = X(\epsilon)$

Set $\epsilon = 0$ calculate $X = X_A$

$\epsilon = \epsilon + \delta$ calculate $X = X_B$

Case (1) or Case (2) is now appropriate.

Case (1) $X_B \geq X_A$

Stage (i) Set $X_C = X_B$, $\delta = 2\delta$
 $\epsilon = \epsilon + \delta$ calculate $X = X_B$

Stage (ii) if $X_B \geq X_C$
 set $X_A = X_C$, $X_C = X_B$, $\delta = 2\delta$
 $\epsilon = \epsilon + \delta$ calculate $X = X_B$
 Repeat stage (ii) until $X_B < X_C$

Stage (iii) when $X_B < X_C$
 set $X_D = X_B$
 $\epsilon = \epsilon - \frac{1}{2}\delta$ calculate X_B
 Stage (iv) or (v) is now appropriate

Stage (iv) if $X_B \geq X_C$
 set $X_A = X_C$, $X_C = X_B$
 use formula (1) for $\epsilon_{\max}$, calculate $X = X(\epsilon_{\max})$

Stage (v) if $X_B < X_C$
 set $X_D = X_B$, $\epsilon = \epsilon - \frac{1}{2}\delta$
 use formula (1) for $\epsilon_{\max}$, calculate $X = X(\epsilon_{\max})$

Case (2) $X_B < X_A$

Stage (i) Set $\epsilon = 0$ calculate $X = X_A$

$\epsilon = \epsilon + \delta$ calculate $X = X_B$

Set $X_C = X_A$, $X_A = X_B$

$\delta = -\delta$, $\epsilon = 0$

$\epsilon = \epsilon + \delta$ calculate $X = X_B$

Stage (ii) or (iii) is now appropriate.

Stage (ii) If $X_B < X_C$

set $X_D = X_B$

$\epsilon = 0$, $\delta = 2\delta$

use formula (1) for $\epsilon_{\max}$, calculate $X = X(\epsilon_{\max})$

Stage (iii) If $X_B \geq X_C$

set $X_A = X_C$, $X_C = X_B$

and the situation is now as described in
Case (1) at stage (ii).

Formula (1)

$$\epsilon_{\max} = \epsilon - \frac{\delta}{4} \frac{(X_A - X_D)}{(2X_C - X_A - X_D)}$$

This formula is derived in a similar manner to that of the linear search at constant step.

As in the search technique at constant step the value of $X(\epsilon_{\max})$ is compared with the current greatest value of X which in this technique is always the central point X_C of the three used in the interpolation.

$X(\epsilon_{\max})$ is accepted as the maximum value in the direction being considered if and only if $X(\epsilon_{\max}) \geq X_C$; otherwise

X_C is selected as the starting value for the next search.

(157)

APPENDIX C

Supplementary to Chapter 5

Flow Diagram for Discrete/Linear Temperature Combination With Variable Final Time

(159)

APPENDIX C2Main Programme for Linear Form of Temperature Profile

```

MASTER PARVA
C   OPTIMAL TEMPERATURE PROFILES
C   VARIATION OF PARAMETERS - ROSENBROCK HILL CLIMB
    DIMENSION X(2),G(4),H(4),A(6),D(2),E(2),W(30)
    JAI=0
    READ (1,101) NDATA
    1 WRITE (4,201)
201  FORMAT (29H OPTIMAL TEMPERATURE PROFILES //47H VARIATION OF PARA
1METERS -ROSENBROCK HILL-CLIMB // 13H PROGRAMME 10 )
    CALL TIME (SECS)
    WRITE (4,202) SECS
    READ (1,102) N,M,KMAX,IPRINT,BJ,W(11)
    READ (1,103) (W(I),I=1,10)
    READ (1,104) W(12)
    READ (1,103) (X(I),I=1,N)
    READ (1,105) (G(I),H(I),I=1,M)
    NW = 30
    NA = N*(N+1)
    NG = 2*M
    DO 3 I=1,5
    3 W(I) = W(I)/1.9865
    CALL PXS6C(N,M,KMAX,IPRINT,BJ,F,X,G,H,W,NW,NA,NG,A,D,E )
    JAI = JAI + 1
    CALL TIME (SECS)
    WRITE (4,202) SECS
    IF(JAI - NDATA) 1,2,2
101  FORMAT (I4)
102  FORMAT (4I4,2F10.0)
103  FORMAT (5F10.0)
104  FORMAT (F10.0)
202  FORMAT (A8)
105  FORMAT (8F10.0)
    2 STOP
    END

```

END OF SEGMENT, LENGTH 175, NAME PARVA

APPENDIX C2 (continued)

Linear Form of Temperature Profile; Subroutine to Calculate the
Yield (Trapezoidal Method) and Rosenbrock Subroutines

```

SUBROUTINE CALXGH (N,M,IT,F,X,G,H,W)
DIMENSION X(2),G(4),H(4),W(30)
J=12
X4 =1.0
X5 =0.0
X6 =0.0
TI = 0.0
6 CORRT = X(2)-X(1)*TI
CORRT = (658.0-CORRT)/(658.0*CORRT)
DO 2 I=1,5
K = J+I
2 W(K) = W(11)/2.0*W(I+5)*EXP(W(I)*CORRT)
W(J+6) = W(J+1)+W(J+2)+W(J+3)
IF(J-12) 4,4,3
3 X1 = X4* (1.0-W(18))/(1.0+W(24))
X2 = (X5* (1.0-W(16)) +X4*W(13) + X1*W(19))/(1.0+W(22))
X3 = (X6* (1.0-W(17)) +X5*W(16) + X2*W(22))/(1.0+W(23))
J =12
X4 = X1
X5 = X2
X6 = X3
DO 5 I=1,6
K = J+I
5 W(K) = W(K+6)
4 J = J+6
TI = TI + W(11)
IF(TI - W(12)) 6,6,7
7 F = X3
RETURN
END

```

END OF SEGMENT, LENGTH 313, NAME CALXGH

PXS6C	(SEMI-COMPILED)
PXS6C1	(SEMI-COMPILED)
PXS6C2	(SEMI-COMPILED)
PXS6C3	(SEMI-COMPILED)
PXS6C4	(SEMI-COMPILED)

FINISH

END OF COMPILATION - NO ERRORS

APPENDIX C3

Linear/Exponential Form of Temperature Profile; Subroutine to
Calculate the Yield (Trapezoidal Method)

```

SUBROUTINE CALXGH (N,M,IT,F,X,G,H,W )
DIMENSION X(6),G(12),H(12),W(30)
J=12
X4 =1.0
X5 =0.0
X6 =0.0
TI =0.0
8 IF(X(2)-TI) 2,3,3
3 CORRT = X(1)
7 CORRT =(658.0-CORRT)/(658.0*CORRT)
GO TO 4
2 CORRT = X(1)*EXP(X(3)*(TI - X(2)))
GO TO 7
4 DO 5 I=1,5
K = J+I
5 W(K) = W(11)/2.0*W(I+5)*EXP(W(I)*CORRT)
W(J+6) = W(J+1)+ W(J+2)+ W(J+3)
IF(J-12) 12,12,6
6 X1 = X4* (1.0-W(18))/(1.0+W(24))
X2 =(X5* (1.0-W(16)) +X4*W(13) + X1*W(19))/(1.0+W(22))
X3 =(X6* (1.0-W(17)) +X5*W(16) + X2*W(22))/(1.0+W(23))
J=12
X4 = X1
X5 = X2
X6 = X3
DO 10 I=1,6
K = J+I
10 W(K) = W(K+6)
12 J = J+6
TI = TI + W(11)
IF(TI - W(12)) 8,8,11
11 F = X3
RETURN
END

```

END OF SEGMENT, LENGTH 342, NAME CALXGH

APPENDIX C4

Linear/Exponential Form of Temperature Profile with Variable Final Time; Subroutine to Calculate the Yield (Runge Kutta Method)

```

SUBROUTINE CALXGH (N,M,IT,F,X,G,H,W )
DIMENSION X(6),G(12),H(12),W(30)
X4 =1.0
X5 =0.0
X6 =0.0
TI =0.0
8 IF(X(2)-TI) 2,3,3
3 CORRT = X(1)
7 CORRT =(658.0-CORRT)/(658.0*CORRT)
GO TO 4
2 CORRT = X(1)*EXP(X(3)*(TI - X(2)))
GO TO 7
4 DO 5 J =1,5
5 W(J+12) = W(11)/2.0*W(J+5)*EXP(W(J)*CORRT)
W(18) = W(13)+W(14)+W(15)
U1=-W(18)*X4*2.0
V1=(W(13)*X4 -W(16)*X5)*2.0
W1=(W(16)*X5 -W(17)*X6)*2.0
U2=U1*(1.0-W(18))
V2=V1*(1.0-W(16))+W(13)*U1
W2=W1*(1.0-W(17))+W(16)*V1
U3=U1-W(18)*U2
V3=V1-(W(16)*V2-W(13)*U2)
W3=W1-(W(17)*W2-W(16)*V2)
U4=U1-W(18)*U3*2.0
V4=V1+(W(13)*U3-W(16)*V3)*2.0
W4=W1+(W(16)*V3-W(17)*W3)*2.0
X1=X4 +(U1+2.0*(U2+U3)+U4)/6.0
X2=X5 +(V1+2.0*(V2+V3)+V4)/6.0
X3=X6 +(W1+2.0*(W2+W3)+W4)/6.0
X4 = X1
X5 = X2
X6 = X3
TI = TI + W(11)
IF (TI - X(4)) 8,11,11
11 F = X3
RETURN
END

```

END OF SEGMENT, LENGTH 409, NAME CALXGH

APPENDIX C5

Temperature Profile Consisting of Three Straight Lines; Subroutine
to Calculate the Yield (Trapezoidal Method)

```

SUBROUTINE CALXGH (N,M,IT,F,X,G,H,W)
DIMENSION X(6),G(12),H(12),W(30)
J=12
X4 =1.0
X5 =0.0
X6 =0.0
TI=0.0
8 IF(X(2)-TI) 2,3,3
3 CORRT = X(1)
7 CORRT = (658.0 - CORRT)/(658.0*CORRT)
GO TO 4
2 IF(X(4) - TI) 13,14,14
14 CORRT = X(1) + X(2)*X(3)-X(3)*TI
GO TO 7
13 CORRT = X(1) + X(2)*X(3) + X(4)*(X(5) - X(3))-X(5)*TI
GO TO 7
4 DO 5 I=1,5
K = J+I
5 W(K) = W(11)/2.0*W(I+5)*EXP(W(I)*CORRT)
W(J+6) = W(J+1)+ W(J+2) +W(J+3)
IF(J-12) 12,12,6
6 X1 = X4* (1.0-W(18))/(1.0+W(24))
X2 = (X5* (1.0-W(16)) +X4*W(13) + X1*W(19))/(1.0+W(22))
X3 = (X6* (1.0-W(17)) +X5*W(16) + X2*W(22))/(1.0+W(23))
J=12
X4 = X1
X5 = X2
X6 = X3
DO 10 I=1,6
K = J+I
10 W(K) = W(K+6)
12 J = J+6
TI = TI + W(11)
IF(TI - W(12)) 8,8,11.
11 F = X3
RETURN
END

```

END OF SEGMENT, LENGTH 415, NAME CALXGH

APPENDIX C6

Discrete Temperature Form for Profile; Subroutine to Calculate
the Yield (Runge Kutta Method)

```

SUBROUTINE CALXGH (N,M,IT,F,X,G,H,W)
DIMENSION X(21),G(42),H(42),W(30)
X4 =1.0
X5 =0.0
X6 =0.0
DO 2 I=1,N
CORRT = X(I)
CORRT = (658.0-CORRT)/(658.0*CORRT)
DO 3 J=1,5
3 W(J+12) = W(11)/2.0*W(J+5)*EXP(W(J)*CORRT)
W(18) = W(13)+W(14)+W(15)
U1=-W(18)*X4*2.0
V1=(W(13)*X4 -W(16)*X5)*2.0
W1=(W(16)*X5 -W(17)*X6)*2.0
U2=U1*(1.0-W(18))
V2=V1*(1.0-W(16))+W(13)*U1
W2=W1*(1.0-W(17))+W(16)*V1
U3=U1-W(18)*U2
V3=V1-(W(16)*V2-W(13)*U2)
W3=W1-(W(17)*W2-W(16)*V2)
U4=U1-W(18)*U3*2.0
V4=V1+(W(13)*U3-W(16)*V3)*2.0
W4=W1+(W(16)*V3-W(17)*W3)*2.0
X1=X4 +(U1+2.0*(U2+U3)+U4)/6.0
X2=X5 +(V1+2.0*(V2+V3)+V4)/6.0
X3=X6 +(W1+2.0*(W2+W3)+W4)/6.0
X4=X1
X5=X2
2 X6=X3
F = X3
RETURN
END

```

END OF SEGMENT, LENGTH 360, NAME CALXGH

APPENDIX C7

Discrete/Linear Form for Temperature Profile; Subroutine to
Calculate the Yield (Runge Kutta Method)

```

SUBROUTINE CALXGH (N,M,IT,F,X,G,H,W)
DIMENSION X(21),G(42),H(42),W(30)
X4 =1.0
X5 =0.0
X6 =0.0
TI = 0.0
NS = W(30)
I = 1
11 IF(TI-W(12)) 4,5,5
5 CORRT = ((X(M)-X(M-1))*TI + X(M-1) - W(12)*X(M))/(1.0 - W(12))
NS = 0
GO TO 6
4 CORRT = X(I)
NSC = 0
6 CORRT = (658.0-CORRT) / (658.0*CORRT)
DO 3 J=1,5
3 W(J+12) = W(11)/2.0*W(J+5)*EXP(W(J)*CORRT)
W(18) = W(13)+W(14)+W(15)
7 U1=-W(18)*X4*2.0
V1=(W(13)*X4 -W(16)*X5)*2.0
W1=(W(16)*X5 -W(17)*X6)*2.0
U2=U1*(1.0-W(18))
V2=V1*(1.0-W(16))+W(13)*U1
W2=W1*(1.0-W(17))+W(16)*V1
U3=U1-W(18)*U2
V3=V1-(W(16)*V2-W(13)*U2)
W3=W1-(W(17)*W2-W(16)*V2)
U4=U1-W(18)*U3*2.0
V4=V1+(W(13)*U3-W(16)*V3)*2.0
W4=W1+(W(16)*V3-W(17)*W3)*2.0
X1=X4 + (U1+2.0*(U2+U3)+U4)/6.0
X2=X5 + (V1+2.0*(V2+V3)+V4)/6.0
X3=X6 + (W1+2.0*(W2+W3)+W4)/6.0
TI = TI + W(11)
X4=X1
X5=X2
X6=X3
IF(TI-1.0) 8,2,2
8 IF(NS) 9,5,9
9 NSC = NSC + 1
IF(NSC-NS) 7,10,10
10 I = I+1
GO TO 11
2 F = X3
RETURN
END

```

REFERENCES

1. Lee, E. S. "Quasilinearization and Invariant Imbedding". (Academic Press), 1968.
2. Fan, L.T. "The Continuous Maximum Principle". (John Wiley), 1966.
3. Rosenbrock, H. H. and Storey, C. "Computational Techniques for Chemical Engineers". (Pergamon), 1966.
4. Paiewonsky, B. Synthesis of Optimal Controls. Ch. 9 of "Topics in Optimization", G. Leitmann, editor. (Academic Press), 1967.
5. Lasdon, L.S., Mitter, S. K. and Waren, A.D. The Conjugate Gradient Method for Optimal Control Problems. IEEE Transactions on Automatic Control, Vol. AC-12, No.2. (April, 1967).
6. Kenneth, P and McGill, R. Two-Point Boundary Value Problem Techniques. pp. 69-109 of "Advances in Control Systems III", C. T. Leondes, editor. (Academic Press), 1966.
7. Merriam, C. W. III "Optimization Theory and the Design of Feedback Control Systems". (McGraw-Hill), 1964.
8. Balakrishnan, A. V. and Neustadt, L. W., editors. "Computing Methods in Optimization Problems". (Academic Press), 1964.
9. Bellman, R. and Dreyfus, S. E. "Applied Dynamic Programming". (Princeton U.P.), 1962.
10. Rozonoer, L. I. L. S. Pontryagin Maximum Principle in the Theory of Optimum Systems. A.R.C. 20, 1288, 1405, 1517. (1959)
11. Swann, W. H. Report on the Development of a New Direct Searching Method of Optimization. C.I.L. Research Note 64/3. (1964)
12. Powell, M. J. D. An Efficient Method for finding the Minimum of a Function of Several Variables without Calculating Derivatives. Computer J. 7, 2, pp. 155-162. (July, 1964)
13. Fletcher, R. and Powell, M. J. D. A Rapidly Convergent Descent Method for Minimization. Computer J. 6, p.163. (1963)
14. Davidon, W. C. Variable Metric Method for Minimization. A.E.C. Research and Development Report, ANL-5990. (1959)

15. Fletcher, R. Function Minimization without Evaluating Derivatives - a Review. Computer J. 8, 1, pp. 33-41. (April, 1965)
16. Rosenbrock, H. H. An Automatic Method for Finding the Greatest or Least Value of a Function. Computer J. 3, pp. 175-184. (October, 1960)
17. Leitmann, G., editor. "Optimization Techniques". (Academic Press), 1962.
18. Katz, S. Best Temperature Profiles in Plug-Flow Reactors: Methods of the Calculus of Variations. Ann. N.Y. Acad. Sci., 84, 441-478. (1960)
19. Levine, M. D. A Steepest Ascent Method for Synthesising Optimal Control Programmes. I.Mech.E. Conference on Advances in Automatic Control, Nottingham. (1965)
20. Knapp, C. H. and Frost, P. A. Determination of Optimal Control and Trajectories using the Maximum Principle in Association with a Gradient Technique. J.A.C.C. Preprints, Session VIII, Paper 2. (1964)
21. Kalaba, R. On Non-Linear Differential Equations, the Maximum Operation, and Monotone Convergence. J. Maths. and Mechs. 8, p. 519. (1959)
22. Bryson, A. E. and Denham, W. E. A Steepest Ascent Method for Solving Optimal Programming Problems. J. App. Mech. 29, 2, pp. 247-57. (1962)
23. Mitter, S. K. Successive Approximation Methods for the Solution of Optimal Control Problems. Imperial College Report (1965). (See also Automatica, Vol. 3, pp. 135-49, 1966).
24. Jacobson, D. H. Differential Dynamic Programming Methods for Determining Optimal Control of Non-Linear Systems. Imperial College Report (1966). (See also Int. J. Control, Vol. 7, No. 2, pp. 175-196, 1968).
25. Tracz, G. S. Optimal Control Problems: A Second-Order Computational Scheme. Ninth Annual Conference of CORS, Ottawa. (May, 1967)
26. Takahashi, Y. Some Applications of the Maximum Principle to Second-Order Systems Subject to Input Saturation, Minimizing Error, and Effort. A.S.M.E. Paper 62-WA-135. (1962)
27. Noton, A. R. M. "Introduction to Variational Methods in Control Engineering". (Pergamon), 1965.

28. Jenkins, K. W. and Spang, H. A. III. Optimal Control for the Engineer I. G. E. Research and Development Centre, Report No. 67-C-129. (April, 1967)
29. Ralston, A. and Wilf, H. S., editors. "Mathematical Methods for Digital Computers". (Wiley), 1960.
30. Fox, L., editor. "Numerical Solution of Ordinary and Partial Differential Equations". (Pergamon), 1962.
31. Hamming, R. W. Stable Predictor-Corrector Methods for Ordinary Differential Equations. J. Assoc. Comp. Mach. 6, pp. 37-47. (1959)
32. Fletcher, R. and Reeves, C. M. Function Minimization by Conjugate Gradients. Computer J. 7, 2, pp. 149-154. (July, 1964)
33. Hestenes, M. R. and Stiefel, E. Methods of Conjugate Gradients for Solving Linear Systems. J. Res. N.B.S. Vol. 49, p. 409. (1952)
34. Myers, G. E. Properties of the Conjugate Gradient and Davidson Methods. Journal of Optimization Theory and Applications, to appear.
35. Kelley, H. J. and Myers, G. E. Conjugate Direction Methods for Parameter Optimization. Presented 18th Congress of I.A.F., Belgrade (September, 1967). To appear in Astronautica Acta.
36. Sinnott, J. F. and Luenberger, D. G. Solution of Optimal Control Problems by the Method of Conjugate Gradients. (To be published)
37. Milne, W. E. "Numerical Calculus". (Princeton University Press), first published 1949.
38. Booth, A. D. "Numerical Methods". (Butterworths), 1957.
39. Abadie, J., editor. "Non-Linear Programming" (North-Holland Publishing Co.), 1967.
40. Hildebrand, F. B. "Introduction to Numerical Analysis". (McGraw-Hill), 1956.
41. Powell, M. J. D. A Survey of Numerical Methods for Unconstrained Optimization. A.E.R.E. Report T.P. 340. (July, 1968)
42. Schley, C. H. Jr. Optimal Control for the Engineer II. G. E. Research and Development Centre, Report No. 67-C-130. (1967)