"INDIRECT METHODS FOR THE NUMERICAL SOLUTION OF ORDINARY LINEAR BOUNDARY VALUE PROBLEMS"

H. W. LOCKSLEY

ABSTRACT

This thesis is mainly concerned with indirect numerical solution methods for linear two point boundary value problems. We concentrate particularly on problems with separated boundary conditions which have a 'dichotomy' property. We investigate the inter-relationship of various methods including some which have first appeared since the work for this thesis began. We examine the stability of these methods and in particular we consider circumstances in which the methods discussed give rise to well conditioned decoupling transformations. Empirical comparisons of some of the methods are described using a set of test problems including a number of 'stiff' and marginally ill conditioned problems.

In the past the main method of error estimation has been to repeat the whole calculation. Here an alternative error estimation technique is proposed and a related iterative improvement method is considered. Although results for this are not completely conclusive we think they justify the need for further research on the method as it shows promise of being a novel and reliable practical method of solving both well conditioned and ill conditioned problems.

CONTENTS

INTRODUCTION

Numerical methods for the solution of boundary value problems (BVPs) for ordinary differential equations (ODEs) can be broadly classified as either direct or indirect. The former are methods based on finite differences, finite elements or collocation, in all of which the solution of the BVP is obtained discretely by solving linear (global) algebraic systems. Indirect methods are so called because they are based on finding the numerical solution of auxiliary initial value problems (IVPs). Variants of these methods such as multiple (parallel) shooting are really hybrid methods but we classify them here as indirect.

An important and commonly occurring type of BVP is one whose differential system possesses an exponential dichotomy. This thesis is mainly concerned with indirect solution methods for linear two point boundary value problems (LBVPs) which are dichotomic. We concentrate particularly on LBVPs with separated boundary conditions (BCs) for which the concept of dichotomy is very closely related to that of conditioning. We investigate the inter-relationships of the methods and examine their stability. We show that all of the methods considered can be collectively regarded as well conditioned (explicit or implicit) decoupling transformations which ensure the stability of the auxiliary IVPs.

Chapter 0 contains a very brief review of direct methods which we include for the sake of completeness.

In Chapter 1 we discuss the stability of IVPs, the conditioning of LBVPs and the concept of dichotomy.We examine the reliability of eigenvalues as indicators of IVP stability and dichotomic structure and in this connection kinematic similarity transformations are introduced. The close relationship between well conditioning of a LBVP and dichotomy is illustrated for the case of separated BCs. Finally we explain what is meant by stable decoupling transformations and introduce two important examples of these viz. the Riccati and the continuous orthonormal.

Chapter 2 is devoted to 'shooting' methods: single shooting, multiple (parallel) shooting and stabilised marching. We justify the stability of Conte's re-orthonormalisation method by showing how it can be regarded as a well conditioned discrete decoupling transformation.

Chapter 3 deals with two of the main variants of continuous orthonormalisation (the invariant imbedding method of Van Loon and the double sweep method of Davey and Meyer) and examines the relationship between them. The simple superposition method suffers from the well known disadvantage that the homogeneous solutions of the given differential system may lose their independence. The methods of this Chapter seek to overcome this drawback by finding an orthonormal set of solutions of another differential system which span the same subspace as that spanned by the solutions of the original system.

In Chapter 4 we look in detail at the Riccati method (including both the 'double sweep' method and the invariant imbedding

technique) and we show how the disadvantage of possible singularities in the Riccati solution may be overcome by a reimbedding restart strategy. Also included is a description of another method which is related to the Riccati method known as the Compound Matrix method in which these singularities are actually removed.

In Chapter 5 we look again at the Riccati and continuous orthonormalisation methods but this time from the slightly different viewpoint of Babuska and Majer viz. the factorisation method in which a set of conditions equivalent to the initial (final) BC of the given LBVP is propagated forwards (backwards) across the problem interval. Here the forward and backward sweeps are independent in that each employs a different form of the same decoupling transformation of which only part is used. The chief advantage of this approach is that the computed errors in the solved IVPs can (for a well conditioned LBVP) provide a meaningful bound for the computed error in the LBVP solution.

Chapter 6 contains a description of a proposed error estimation and iterative improvement method based upon multiple shooting. We think that this method warrants further investigation and research as it shows promise of being a novel and reliable practical method of solving (ill conditioned) LBVPs.

Finally Appendices I, II and III and a reference list are included at the end.

No attempt has been made in the text to distinguish typographically between matrices and vectors and scalars, but whenever a matrix or vector is introduced its dimensions are given : (m,n) denotes a matrix with m rows and n columns. Also references such as [6] refer to the reference list at the end, ones like [3-2] refer to the relevant section in Appendix I whilst (5.3) denotes equation number 3 of Chapter 5.

All of the numerical results given in Chapter 6 and in Appendices II and III were obtained in double precision from programs especially written in Pascal using a Prospero compiler (Pro Pascal iid 3.143) and run on a stand alone RM Nimbus PC1 microcomputer.

CHAPTER 0

DIRECT METHODS

As stated in the introduction, numerical methods for the solution of two point BVPs with ODEs can be broadly classified as either direct or indirect. In Chapters 1 to 5 we deal in detail with indirect methods for linear BVPs with separated BCs as this is really the focus of this thesis. However, for the sake of completeness, we include here a brief review of some of the main direct methods which are applicable also to non-linear BVPs with general non-linear BCs.

Any n dimensional two point BVP defined on the interval $a <= t <= b$ can be written as a system of n ODEs of the form :

$$\dot{x}(t) = g(t, x(t)) \qquad (0.1a)$$

together with n BCs :

$$r(x(a), x(b)) = 0 \qquad (0.1b)$$

where the solution $x(t)$ $(n,1)$ is assumed to be unique and where g $(n,1)$ and r $(n,1)$ may be non-linear functions.

Direct methods can be subdivided into :

A) segmentation methods

B) series truncation methods

C) function space methods

with further sudivisions of each of these as outlined below.

A) Segmentation methods : In all of these the whole problem interval $I = [a,b]$ is subdivided into N segments :

$I_j = [t_{j-1}, t_j]$ $(1 <= j <= N)$ where

$a = t_0 < t_1 < t_2 < \ldots\ldots < t_N = b$. At nodes t_j $(0 <= j <= N)$ approximations α_j to the exact solution vectors x_j of the BVP are obtained as relations of the form :

$$\Psi_j(\alpha_j, \alpha_{j+1}) = 0 \qquad (0.2a)$$

$(0 <= j <= N - 1)$. Together with the BCs :

$$r(\alpha_0, \alpha_N) = 0 \qquad (0.2b)$$

these provide a system of $(N + 1)$ equations for the calculation of the $(N + 1)$ unknown vectors α_i $(0 <= i <= N)$. This system may be written : $\Phi\alpha = 0$ (0.3) where $\alpha = [\alpha_0, \ldots\ldots, \alpha_N]^T$ and Φ is, in general, a non-linear operator. Now system (0.3) will be stable if, for any two values α^1 and α^2 of α, corresponding to other different segmentations we have : $\| \alpha^1 - \alpha^2 \| < s \| \Phi\alpha^1 - \Phi\alpha^2 \|$ where s is a constant independent of the segmentation. The success of all of these methods depends on choosing operators Ψ_j in equation (0.2a) such that the ODEs (0.1a) are approximated sufficiently well that system (0.3) is stable.

We can subdivide segmentation methods into :

(i) IVP methods (multiple and parallel shooting). These are dealt with as indirect methods in Chapter 2.

(ii) Piecewise polynomial function (or collocation) methods. Here each major subinterval $I_j = [t_{j-1}, t_j]$ $(1 <= j <= N)$ is itself segmented by the insertion of M nodes t_i^j $(1 <= i <= M)$ so as to produce in total a grid of $N(M + 1)$

segments of [a,b]. Now on each major segment I_j we define an Mth order polynomial $v_j(t)$ (n,1) by :

$$v_j(t) = \sum_{k=0}^{M} c_k^j (t - t_{j-1})^k \qquad (0.4)$$

for $t \in I_j$ $(1 <= j <= N)$, where c_k^j $(0 <= k <= M)$ are constant (n,1) vectors to be determined. The piecewise functions $v_j(t)$ defined by (0.4) are required to satisfy the ODEs (0.1a) at all of the sub-grid points t_i^j i.e.

$$\dot{v}_j(t_i^j) - g(t_i^j, v_j(t_i^j)) = 0 \qquad (0.5)$$

for $1 <= j <= N$ and $1 <= i <= M$, where $\dot{v}_j(t) = \sum_{k=0}^{M} k\, c_k^j (t - t_{j-1})^{k-1}$. In addition, the v_j functions must be continuous at the end of each major segment i.e.

$$v_j(t_{j-1}) - v_{j-1}(t_{j-1}) = 0 \qquad (0.6)$$

for $2 <= j <= N$, and also $v_1(a)$ and $v_N(b)$ must satisfy the given BC (0.1b) i.e.

$$r(v_1(t_0), v_N(t_N)) = 0 \qquad (0.7)$$

Thus equations (0.5), (0.6) and (0.7) together provide N(M + 1) vector equations from which the unknowns c_k^j can be obtained, where $c_0^j = v_j(t_{j-1})$ are the required approximations to the LBVP solution x(t) at the major nodes t_{j-1} $(1 <= j <= N)$.

(iii) Finite difference methods : Here each subinterval $I_j = [t_{j-1}, t_j] = \Delta t_j$ is taken sufficiently small as to be acceptable as the steplength of an implicit one-step integration method such as the trapezoidal rule or the mid-point rule [see Lambert: 22]. For example, using the latter rule we get :

$$\alpha_{j+1} - \alpha_j = (\Delta t_j).g\left(\frac{t_j + t_{j+1}}{2}, \frac{\alpha_j + \alpha_{j+1}}{2}\right)$$

which can be written in form (0.2a) as :

$$\left(\frac{\alpha_{j+1} - \alpha_j}{\Delta t_j}\right) - g\left(\frac{t_j + t_{j+1}}{2}, \frac{\alpha_j + \alpha_{j+1}}{2}\right) = 0 \qquad (0.8)$$

$(0 <= j <= N - 1)$. Together with the BC : $r(\alpha_0, \alpha_N) = 0$ this gives us $(N + 1)$ equations from which $\alpha = [\alpha_0, \ldots, \alpha_N]^T$ can be computed.

B) Series truncation methods : In these methods the solution $x(t)$ of BVP (0.1) is expressed in the form of an infinite series of terms of which a finite number is used in the computation. The most common application employs (orthogonal) Chebyshev polynomials. In this case, the problem interval $a <= t <= b$ must first be transformed to $-1 <= s <= 1$ by making the substitution $t = 0.5\{(b - a)s + b + a\}$ in equations (0.1a & b). The kth Chebyshev polynomial is defined as

$$T_K(s) = \cos(k \cos^{-1} s) \qquad (0.9)$$

which satisfies the recursion

$$T_{K+1}(s) = 2s\, T_K(s) - T_{K-1}(s) \qquad (0.10)$$

and hence also

$$\dot{T}_{K+1}(s) = 2\{s\, \dot{T}_K(s) + T_K(s)\} - \dot{T}_{K-1}(s) \qquad (0.11)$$

where $\dot{T} = \frac{dT}{ds}$.

By using recursions (0.10) and (0.11) we can thus express any product of Chebyshev polynomials (or any derivative of a Chebyshev polynomial) as a linear combination of Chebyshev polynomials.

To apply the method we assume an approximate solution $\alpha(s)$

of BVP (0.1) of the form :

$$\alpha(s) = \sum_{k=0}^{N} \beta_k T_k(s) \qquad (0.12)$$

where β_k ($0 <= k <= N$) are constant (n,1) vectors to be determined. $\alpha(s)$ is now substituted into ODE (0.1a) and each side is obtained as a linear combination of Chebyshev polynomials up to order N. By equating coefficients of $T_k(s)$ ($1 <= k <= N$) we can obtain N equations for $\beta = [\beta_0, \beta_1, \ldots\ldots, \beta_N]^T$. Evaluating (0.12) at $s = \pm 1$ and substituting into BC (0.1b) provides another equation for β. Hence we have (N + 1) equations from which β can be computed. The required approximate solution $\alpha(s)$ to BVP (0.1) is then given by (0.12) for all $s \in [-1,1]$.

Alternatively, in an analagous fashion to the above, we can use trigonometric polynomials instead of Chebyshev polynomials by writing equation (0.12) in the form :

$$\alpha(s) = B_0 + \sum_{k=1}^{N} \{A_k \sin(ks) + B_k \cos(ks)\} \qquad (0.13)$$

where now the problem interval [a,b] has been transformed to $[0, 2\pi]$.

C) Function space methods : In these, we obtain an approximation $\alpha(t)$ to the solution $x(t)$ of the BVP where $\alpha(t)$ is of the form : $\alpha(t) = \sum_{k=0}^{N} \beta_k w_k(t)$ (0.14)

where $w_k(t)$ ($0 <= k <= N$) are a set of independent basis functions. Since the Chebyshev polynomials are orthogonal over [-1,1] with respect to weight function $\sigma(t) = \dfrac{1}{\sqrt{1 - t^2}}$

$$\left(\text{i.e.} \int_{-1}^{1} \frac{T_k(t).T_\ell(t)}{\sqrt{1-t^2}} \, dt = 0 \quad \text{if} \quad k \neq \ell \right)$$ the basis functions $w_k(t)$ are often taken to be $T_k(t)$ (assuming that the problem interval has been transformed to $[-1,1]$). The $(n,1)$ constant vectors β_k $(0 <= k <= N)$ are then determined so that $\alpha(t)$ minimises some measure of error. For example, in collocation methods, the error $e(t) = \dot{\alpha}(t) - g(t, \alpha(t))$ in satisfying the given ODE (0.1a) is made zero at N distinct points t_j $(1 <= j <= N)$ in $[-1,1]$ i.e.

$$\dot{\alpha}(t_j) - g\left(t_j, \alpha(t_j)\right) = 0 \qquad (0.16)$$

for $1 <= j <= N$, where $\dot{\alpha}(t) = \sum_{k=0}^{N} \beta_k \dot{T}_k(t)$ and $\dot{T}_k(t) = \frac{k}{\sqrt{1-t^2}} . \sin(k.\cos^{-1} t)$. $\alpha(t)$ is also required to satisfy the BC (0.1b) i.e. $r(\alpha(-1), \alpha(1)) = 0$ (0.17).

This provides us with a total of $(N + 1)$ equations for the determination of $\beta = [\beta_0, ..., \beta_N]^T$, from which the solution $\alpha(t)$ to the BVP is obtained at any value of $t \in [-1,1]$ by using (0.14).

In the least squares method we proceed similarly to that described above but in (0.16) we take $M > N$ points t_j so as to obtain an overdetermined system (i.e. more than $N + 1$ equations) from which we compute the least squares solution.

In the Galerkin method, instead of condition (0.16) the error $e(t)$ is required to be orthogonal to each of the first N basis functions over $[-1,1]$ i.e.

$$\int_{-1}^{1} \sigma(t) \, \| e(t) \| \, . \, T_k(t) \, dt = 0 \qquad (0.18)$$

for $0 <= k <= N - 1$, and in addition (0.17) must be satisfied. Also we may note the Ritz (finite element) method in which the given BVP is replaced by an equivalent variational problem of minimising a certain functional related to the problem. However, the application of this method is limited to a certain class of BVPs which can be variationally formulated.

(A detailed discussion of all of the foregoing methods in sections A, B and C can be found in [12]).

All of the above direct methods have the disadvantage of requiring the solution of a large system of equations. With the possible exception of multiple shooting, this is avoided in indirect methods by obtaining the BVP solution via forward and backward integrations of IVPs over the whole problem interval [a,b]. However, as we shall see in the following Chapters, this requires these methods to be theoretically more convoluted.

CHAPTER 1

STABLE DECOUPLING TRANSFORMATIONS ([1], [7], [13])

Introduction

Any LBVP can be written in the form :

$$\dot{x}(t) = A(t)x(t) + f(t) \qquad (1.1a)$$

$$B_0 x(a) + B_1 x(b) = c \qquad (1.1b)$$

for $a <= t <= b$, where $\dot{x}(t) = \frac{d}{dt} x(t)$, t is the real variable of integration, A is of dimension (n,n), x and f are (n,1), B_0 and B_1 are constant (n,n) matrices and c is constant (n,1).

In [1-1] we show that any single nth order linear differential equation can be written in form (1.1a). For such a LBVP as (1.1), depending on the choice of BCs, there may be a unique solution or no solution or an infinity of solutions as the following (2,2) example shows :

Ex. 1: Take $A = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$, $f=0$, $B_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, $B_1 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}$, $c = \begin{bmatrix} 0 \\ c_2 \end{bmatrix}$, $a=0$. The general solution which satisfies the initial BC $[1 \quad 0].x(0) = 0$ is $x(t) = k \begin{bmatrix} \sin t \\ \cos t \end{bmatrix}$, where k is an arbitrary constant. Thus if $b = \pi$ then the LBVP has no solution if $c_2 \neq 0$ but an infinity of solutions if $c_2 = 0$.

If, in (1.1b), $B_0 = 0$ and B_1 is nonsingular (or viceversa) then the LBVP reduces to an IVP as now all the BCs are given at one point. If $X(t) = [X_1(t) | X_2(t) | \ldots\ldots | X_n(t)]$ is a nonsingular variable (n,n) matrix for which $\dot{X}_i(t) = A(t)X_i(t)$ for

$1 <= i <= n$ (i.e. for which $\dot{X}(t) = A(t)X(t)$) then we say that $X(t)$ is a fundamental solution of the system $\dot{x}(t) = A(t)x(t)$. Now if $X(t)$ is any such fundamental solution then LBVP (1.1) has a unique solution iff matrix Q (n,n) is nonsingular where

$$Q = B_0 X(a) + B_1 X(b) \qquad (1.2).$$

In this case (which is assumed for all LBVPs throughout this thesis) the unique solution of (1.1) is given by [1-2] :

$$x(t) = \Phi(t)c + \int_a^b G(t,s)f(s)\,ds \qquad (1.3a)$$

where $\Phi(t) = X(t)Q^{-1}$ and $G(t,s)$ is the (n,n) Green's function matrix defined by :

$$G(t,s) = \begin{cases} \Phi(t)B_0\Phi(a)\Phi^{-1}(s) & \text{for } s <= t \\ -\Phi(t)B_1\Phi(b)\Phi^{-1}(s) & \text{for } s > t \end{cases} \qquad (1.3b).$$

(This result is really only of theoretical importance due to the considerable cost of obtaining $G(t,s)$ in practice.)

Stability of IVPs

We turn now to the consideration of stability of IVPs because the stability of the solution algorithms for LBVPs that we are to examine will be measured by the stability of the associated auxiliary IVPs. Consider the linear IVP :

$$\dot{x}(t) = A(t)x(t) \qquad (1.4a)$$

$$x(a) = c \qquad (1.4b)$$

for $a <= t <= b$. If $x(t) = X(t)e$ where $e = X^{-1}(a)c$ then $\dot{x}(t) = \dot{X}(t)e = A(t)X(t)e = A(t)x(t)$ and $x(a) = X(a)e = c$. Thus the exact solution of (1.4) can be written $x(t) = X(t)e$ where

$X(t)$ is any fundamental solution of (1.4a). Now we say that IVP (1.4) is stable (well conditioned) iff any small perturbation in the data (i.e. in c or A(t)) does not produce a correspondingly large perturbation in the value of $x(t)$ for any $t \in [a,b]$. More precisely, we say that a solution $x(t)$ of (1.4) is (uniformly) stable over $[a,b]$ iff, given any $\epsilon > 0$ and any point $d >= a$, there exists a $\delta > 0$ such that any other solution $\hat{x}(t)$ of (1.4a) which satisfies $\| x(d) - \hat{x}(d) \| <= \delta$ also satisfies $\| x(t) - \hat{x}(t) \| <= \epsilon$ for all $d < t <= b$. (Here and elsewhere, unless otherwise stated, $\| \cdot \|$ denotes $\| \cdot \|_\infty$).

Note that it is sufficient to consider a homogeneous differential system such as (1.4a) since if $\dot{x}(t) = A(t)x(t) + f(t)$ then $z(t) = x(t) - \hat{x}(t)$ satisfies the homogeneous system $\dot{z}(t) = A(t)z(t)$.

We can quantify the degree of stability of IVP (1.4) by defining a stability constant k where

$$k = \sup_{a<=t<=b} \| X(t)X^{-1}(a) \| \qquad (1.5)$$

Then $x(t) = X(t)X^{-1}(a)c$ ===> $\| x(t) \| <= \| X(t)X^{-1}(a) \| . \| c \|$

===> $\| x(t) \| <= k \| c \|$ (1.6)

for all $t \in [a,b]$. This provides us with a bound on the solution in that if k is small the IVP (1.4) will be well conditioned over $[a,b]$.

We now examine the reliability of the eigenvalues of the system matrix A(t) as indicators of the stability of IVP (1.4).

First consider the case where A is constant with n distinct* eigenvalues. If λ_i and g_i (n,1) are a corresponding eigenvalue and eigenvector pair of A then $Ag_i = \lambda_i g_i$. Now if $u(t) = e^{\lambda_i t} g_i$ then $\dot{u}(t) = e^{\lambda_i t}\lambda_i g_i = e^{\lambda_i t} Ag_i = Au(t)$. Thus $X(t) = GD(t)$, where $G = [g_1 | g_2 | \ldots\ldots | g_n]$ (n,n) and $D(t) = \text{diag}\{ e^{\lambda_1 t}, e^{\lambda_2 t}, \ldots\ldots, e^{\lambda_n t} \}$, is a fundamental solution of $\dot{x}(t) = A(t)x(t)$. Hence the exact solution of IVP (1.4) can be written $x(t) = GD(t).[GD(a)]^{-1} c$ or $x(t) = GD(t).1$ where $1 = [GD(a)]^{-1} c$. In expanded form this becomes :

$$x(t) = \sum_{i=1}^{n} 1_i e^{\lambda_i t} g_i \qquad (1.7)$$

where $1 = [1_1 \ldots\ldots\ldots 1_n]^T$.

We say that the IVP (1.4) is forward stable over an arbitrary interval iff it is stable for any choice of initial value c (i.e. for any 1). From (1.7) this will be so iff all the eigenvalues of A are such that $\text{Re}(\lambda_i) < 0$ $(1 <= i <= n)$ as in this case <u>any</u> forward solution of $\dot{x}(t) = A\,x(t)$ must be a decay vector for increasing t. However, if some of the eigenvalues are such that $\text{Re}(\lambda_i) > 0$ then forward solutions of (1.4) corresponding to choices of 1 (i.e. of c) which exclude all the terms containing these positive exponentials will be forward decay vectors whilst those corresponding to any other choice of c will be forward growth vectors. In this case, the solution space of the system $\dot{x}(t) = A\,x(t)$ is split into a subspace of forward decay

* Slight modification of the following is required for the case where A is constant but does not have a full set of n linearly independent eigenvectors.

solutions and a subspace of forward growth solutions and the IVP (1.4) is unstable for any choice of c. Thus for the constant coefficient case the eigenvalues of A do provide an accurate guide to the stability of the system $\dot{x}(t) = Ax(t)$ over [a,b]. However, for the case where A(t) is variable we shall see that this is not always so. Before dealing with the variable coefficient case though we digress to define what we mean by a kinematic similarity transformation of a differential system.

Kinematic similarity transformations and kinematic eigenvalues

Suppose that T(t) (n,n) is a nonsingular differentiable transformation. Then the substitution :

$x(t) = T(t)y(t)$ in $\dot{x}(t) = A(t)x(t)$ gives

$T(t)\dot{y}(t) + \dot{T}(t)y(t) = A(t)T(t)y(t) \quad ===>$

$\dot{y}(t) = \{T^{-1}(t)A(t)T(t) - T^{-1}(t)\dot{T}(t)\}y(t)$ or $\dot{y}(t) = V(t)y(t)$

where $V(t) = T^{-1}(t)\{A(t)T(t) - \dot{T}(t)\}$. Now from (1.6) we have

$\|x(t)\| <= k\|c\| \quad ===> \quad \|x(t)\| <= k\|T(a)\|.\|y(a)\|$ (as $x(a) = c$)

$===> \quad \|T^{-1}(t)\|.\|x(t)\| <= k\|T^{-1}(t)\|.\|T(a)\|.\|y(a)\|$

$===> \quad \|y(t)\| <= \bar{k}\|y(a)\|$ where $\bar{k} = \|T^{-1}(t)\|.\|T(a)\|.k$.

Thus if T(t) is well conditioned i.e. if cond $T(t,s) = \|T^{-1}(t)\|.\|T(s)\|$ is not large for all $t,s \in [a,b]$ then the condition constants k and $\bar{k}$ will be of the same order of magnitude and we say that the systems $\dot{y}(t) = V(t)y(t)$ and $\dot{x}(t) = A(t)x(t)$ are kinematically similar.

Moreover it can be shown that there exists a (non-unique) orthogonal transformation $T(t)$ for which $V(t)$ will be triangular. In this case, the diagonal elements (i.e. the eigenvalues) of $V(t)$ are called the kinematic eigenvalues of $A(t)$ corresponding to $T(t)$: see [1-3].

Now it is shown in [12] that for a variable coefficient system $\dot{x}(t) = A(t)x(t)$ the kinematic eigenvalues are analagous to the eigenvalues for the case of a constant system i.e. for a variable system it is the kinematic eigenvalues that provide a true indication of the stability properties of IVP (1.4).

In fact, IVP (1.4) will be stable over any interval $[a,b]$ for forward integration from <u>any</u> initial value $x(a)$ iff the kinematic eigenvalues $\{\sigma_1(t), \ldots\ldots, \sigma_n(t)\}$ of $A(t)$ are such that $\forall \alpha, \beta \in [a,b]$ $\mathrm{Re} \int_{\alpha}^{\beta} \sigma_i(t)\, dt < 0$ for $1 <= i <= n$. In this case, we say that any particular solution of system $\dot{x}(t) = A(t)x(t)$ is a forward decay vector or a backward growth vector over $[a,b]$. Also analagous to the constant coefficient case if some of the kinematic eigenvalues are such that $\mathrm{Re} \int_{\alpha}^{\beta} \sigma_i(t)\, dt > 0$ then the solution space of $\dot{x}(t) = A(t)x(t)$ is split into a subspace of forward decay vectors and one of forward growth vectors and the IVP (1.4) is unstable for any value of c.

This leads us to the concept of exponential dichotomy but before we introduce this we give an example which illustrates that we cannot rely on the eigenvalues of a variable system matrix $A(t)$

to correctly indicate stability properties of an associated IVP : [12] :

Ex 2 : Consider the IVP :

$\dot{x}(t) = A(t)x(t)$

$x(a) = c$ for $a <= t <= b$ where

$$A(t) = \begin{bmatrix} (-0.25 + 0.75\cos(2t)) & (1 - 0.75\sin(2t)) \\ (-1 - 0.75\sin(2t)) & (-0.25 - 0.75\cos(2t)) \end{bmatrix}$$

for any given value of c.

A(t) has eigenvalues $0.25*(-1 + i\sqrt{7})$ i.e. both eigenvalues have negative real parts leading us perhaps to expect that the IVP would be stable for any c. However, the transformation $x(t) = T(t)y(t)$ where $T(t) := \begin{bmatrix} \cos(t) & \sin(t) \\ -\sin(t) & \cos(t) \end{bmatrix}$ puts the system $\dot{x}(t) = A(t)x(t)$ into the form $\dot{y}(t) = V(t)y(t)$ where $V(t) = \begin{bmatrix} 0.5 & 0 \\ 0 & -1 \end{bmatrix}$. Thus the kinematic eigenvalues of A(t) corresponding to T(t) are 0.5 and -1, showing in fact that the IVP is unstable for arbitrary c.

The eigenvalues of a variable system $\dot{x}(t) = A(t)x(t)$ will be a good guide to stability properties only when A(t) varies sufficiently slowly over [a,b] to ensure that the eigenvalues of A(t) remain sufficiently close to the kinematic eigenvalues for all $t \in [a,b]$. Therefore, when considering the stability of a given IVP it is advisable to disregard eigenvalues and instead to base the analysis on stability constants as in (1.5) and (1.6) or on kinematic eigenvalues. The disadvantage of the latter is that although they are theoretically important they

are of limited practical value due to the considerable cost of explicitly determining them by an orthogonal transformation deflation method (see [1-3]).

Exponential dichotomies

We now introduce the (theoretically) important concept of exponential dichotomy which underlies the whole subject of stability of decoupling algorithms for the solution of LBVPs. We say that the differential system $\dot{x}(t) = A(t)x(t)$ has an exponential dichotomy over [a,b] with forward growth space of dimension p and forward decay space of dimension q (= n-p) if the spectrum $\{\sigma_1(t), \ldots\ldots, \sigma_n(t)\}$ of kinematic eigenvalues of A(t) corresponding to some orthogonal transformation T(t) is split such that : $\forall\ \alpha, \beta \in [a, b]$

$$\text{Re} \int_{\alpha}^{\beta} \sigma_i(t)\, dt \;\begin{cases} > 0 & \text{for} \quad 1 <= i <= p \quad \text{and} \\ < 0 & \text{for} \quad (p+1) <= i <= n. \end{cases}$$

The solution space of system $\dot{x}(t) = A(t)x(t)$ is split into two subspaces : a forward growth subspace $\Phi_1(t)$ of dimension p and a forward decay subspace $\Phi_2(t)$ of dimension q where p + q = n. In general any solution $\phi(t)$ of $\dot{x}(t) = A(t)x(t)$ will be a combination of solutions belonging to both $\Phi_1(t)$ and $\Phi_2(t)$. Corresponding to any fundamental solution X(t) of system $\dot{x}(t) = A(t)x(t)$ there will exist a constant (n,n) non-singular matrix C such that

$$X(t)C = \Phi(t) \quad \text{where} \quad \Phi(t) = [\, \Phi_1(t) \mid \Phi_2(t) \,] \qquad (1.7a)$$

is a dichotomic fundamental solution. Such a fundamental solution arises naturally in the constant coefficient case. Suppose that A is constant with p positive eigenvalues and q negative eigenvalues then the dimensions of the forward growth and decay subspaces will be p and q respectively

i.e. $\lambda_i > 0$ for $1 <= i <= p$ and

< 0 for $(p+1) <= i <= n$.

Now if s_i (n,1) is the value of the initial vector x(a) corresponding to the choice of vector 1 in which all components are zero except for 1_i then, from (1.7), we have

$x_i(t) = 1_i e^{\lambda_i t} g_i$, where $x_i(t)$ is the particular solution vector corresponding to $x(a) = s_i$. Thus $x_i(t)$ will be growth vectors for $1 <= i <= p$ and decay vectors for $(p+1) <= i <= n$ (for increasing t) and

$\Phi(t) = [\Phi_1(t) \mid \Phi_2(t)] = [x_1(t) \ldots x_p(t) \mid x_{p+1}(t) \ldots x_n(t)]$ will be a dichotomic fundamental solution.

Alternatively, we can define a growth or decay vector for increasing t over [a,b] by means of norm ratios as follows :

(i) $\phi(t)$ is a growth vector if :

$$\frac{\|\phi(\tau)\|}{\|\phi(s)\|} >= \frac{1}{\gamma_1} e^{\lambda(\tau - s)} \qquad (1.8)$$

for all $s, \tau \in [a,b]$ for which $\tau >= s$, where λ and γ_1 are are constants such that $\gamma_1 >= 1$ and $\lambda > 0$ and where it is assumed that λ is not small and γ_1 is not large.

IVPs defined over infinite intervals. Over a finite problem interval [a,b] definitions (1.8),(1.9),(1.10a) and (1.10b) are imprecise because it is possible to find values of the constants to satisfy these conditions for any given differential system.

As stated earlier, when the system matrix A of the given ODE is constant then the split of the spectrum of eigenvalues between those with positive and those with negative real parts accurately reflects the structure of the dichotomy i.e. the dimensions of the growth and decay subspaces respectively. But (analagous to the situation with IVP stability) if A(t) is variable then its eigenvalues may not be a good guide to dichotomy structure. For this we need the kinematic eigenvalues as the following example illustrates :[12]:

Ex 3 : Consider the (2,2) system $\dot{x}(t) = A(t)x(t)$ where

$$A(t) = \begin{bmatrix} (\lambda \cos(2wt)) & (-\lambda \sin(2wt) + w) \\ (-\lambda \sin(2wt) - w) & (-\lambda \cos(2wt)) \end{bmatrix}$$

for $a <= t <= b$ where λ and w are positive parameters. The eigenvalues of $A(t)$ are $\pm \sqrt{\lambda^2 - w^2}$. Now let $x(t) = T(t)y(t)$ where $T(t) = \begin{bmatrix} \cos(wt) & \sin(wt) \\ -\sin(wt) & \cos(wt) \end{bmatrix}$.

The transformed system is $\dot{y} = Vy$ $(a <= t <= b)$ where $V = \begin{bmatrix} \lambda & 0 \\ 0 & -\lambda \end{bmatrix}$ and so the kinematic eigenvalues corresponding to $T(t)$ are $\pm\lambda$, which show that the dimensions of the growth and decay subspaces are $p = q = 1$. The dichotomy does not change with w but we see that as w increases the eigenvalues

of A(t) drift further away from the kinematic eigenvalues i.e. only when A(t) is slowly varying do the eigenvalues provide a good guide to the dichotomy. In fact, when $w > \lambda$ the eigenvalues become imaginary and give no information about the dichotomy structure. (Note that although the eigenvalues of a differential system matrix A(t) may vary under a kinematic similarity transformation T(t), the structure of the dichotomy as shown by the kinematic eigenvalues is invariant).

Not all differential systems possess a dichotomy in the sense described above but we restrict most of our consideration in this thesis to those that do because, as we shall see later, for LBVPs with BCs of separated form there is a close relationship between the existence of a dichotomy and the well conditioning of the LBVP. But before we can deal with that, we must discuss conditioning of LBVPs.

Conditioning of LBVPs

LBVP (1.1) is said to be well conditioned if any small perturbation in A, f, B_0, B_1 or c produces only a small corresponding perturbation in the value of the exact solution $x(t)$ of the LBVP at any value of $t \in [a,b]$. As with IVPs, to quantify the notion of well conditioning we define stability (conditioning) constants k_1 and k_2 by :

$$k_1 = \max_{a <= t <= b} \| X(t)\, Q^{-1} \| \qquad (1.11a)$$

$$k_2 = (b - a) \max_{a <= t,s <= b} \| G(t,s) \| \qquad (1.11b)$$

where X(t) is any fundamental solution of $\dot{x}(t) = A(t)x(t)$ and Q and G(t,s) are as given in (1.2) and (1.3b) respectively.

Note that if $X(t)$ and $Y(t)$ are any two fundamental solutions then $X(t) = Y(t).C$ where C (n,n) is constant. Hence

$X(t).Q^{-1} = X(t)[B_0 X(a) + B_1 X(b)]^{-1} = Y(t).C.[B_0 Y(a)C + B_1 Y(b)C]^{-1}$

i.e. $X(t).Q^{-1} = Y(t).Q_1^{-1}$ where $Q_1 = B_0 Y(a) + B_1 Y(b)$. This shows that in (1.11a) constant k_1 is independent of the choice of fundamental solution $X(t)$.

From (1.3a) we can obtain the following bound on the solution of LBVP (1.1) :

$$\| x \| \leq k_1 \| c \| + k_2 \| f \|_1 \qquad (1.12)$$

(See [12]).

Now consider the perturbed LBVP :

$\dot{w}(t) = A(t)w(t) + f(t) + \delta f(t)$

$B_0 w(a) + B_1 w(b) = c + \delta c$

where $\delta f(t)$ and δc are perturbations in $f(t)$ and c. The difference between the solutions to the perturbed and unperturbed problems at any value of $t \in [a,b]$ is given by $e(t) = w(t) - x(t)$ where $e(t)$ is the solution of the LBVP :

$\dot{e}(t) = A(t)e(t) + \delta f(t)$

$B_0 e(a) + B_1 e(b) = \delta c.$

Thus (1.12) implies that :

$$\| e \| \leq k_1 \| \delta c \| + k_2 \| \delta f \| \qquad (1.13).$$

This shows that $k = \max \{ k_1, k_2 \}$ provides a bound on the effect of perturbations in c and $f(t)$ on the solution $x(t)$, and so k may be taken to be the condition constant of LBVP (1.1) i.e. if k is reasonably small then (1.1) will be well conditioned. In [12] it is shown that the above argument is

still valid when perturbations also occur in A, B_0 and B_1. It is also shown that in fact constant k_1 is redundant i.e. that k_2 small ====> k_1 small, so that in effect we can take k_2 to be the condition constant of LBVP (1.1).

Note that the condition of a LBVP is not significantly altered by a well conditioned kinematic similarity transformation, as we now show.

If $\dot{x}(t) = A(t)x(t) + f(t)$ then putting $x(t) = T(t)y(t)$ gives $\dot{y}(t) = V(t)y(t) + g(t)$ where $V(t) = T^{-1}(t)\{A(t)T(t) - \dot{T}(t)\}$ and $g(t) = T^{-1}(t)f(t)$. Thus from (1.12) we get :

$$\| x \| <= k_1 \| c \| + k_2 \| T \; g \|$$

$$<= k_1 \| c \| + k_2 \| T \| . \| g \| . \text{ Hence}$$

$$\| T^{-1}(t) \| . \| x \| <= k_1 \| T^{-1}(t) \| . \| c \| + k_2 \| T^{-1}(t) \| . \| T \| . \| g \|$$

====> $\| y(t) \| <= \epsilon \| c \| + \theta \| g \|$ where ϵ and θ are the condition constants of the transformed LBVP and $\epsilon = k_1 \| T^{-1}(t) \|$ and $\theta = k_2 \| T^{-1}(t) \| . \| T \|$. Thus if $\max \{ k_1, k_2 \}$ is large then so will be $\max \{ \epsilon, \theta \}$. On the other hand, this does also mean that the well conditioning of a LBVP is preserved under a well conditioned kinematic similarity transformation and we utilise this property later to justify the stability of transformation decoupling algorithms.

The condition of a LBVP is important because if the problem is not well conditioned (i.e. if k is unreasonably large) then even if a stable algorithm is used to solve it we must still expect large errors in the computed solution $x(t)$. Fortunately, most LBVPs which describe physically realistic situations are

well conditioned. (For those that are not, in Chapter 6, we put forward an error estimation technique based on the multiple shooting method (see Chapter 2)). Note that, in practice, the condition constants k_1 and k_2 are of limited value because for their determination we require a fundamental solution $X(t)$ of system $\dot{x}(t) = A(t)x(t)$ for $t \in [a,b]$: if $A(t)$ is 'stiff' (see Chapter 2) then it may not be possible to find $X(t)$ with sufficient accuracy.

Well conditioned LBVPs with separated BCs

We saw earlier that for any dichotomic differential system $\dot{x}(t) = A(t)x(t) + f(t)$ it is the kinematic eigenvalues that determine both the stability (condition) of an associated IVP and also the structure of the dichotomy. This suggests that for a LBVP there may be a connection between its condition and its dichotomy. For the case where the LBVP has separated BCs this is indeed true. We say that the BC (1.1b) are separated if B_0 and B_1 have the form :

$$B_0 = \begin{bmatrix} 0 \\ B_a \end{bmatrix}_{m}^{n-m} \quad \text{and} \quad B_1 = \begin{bmatrix} B_b \\ 0 \end{bmatrix}_{m}^{n-m} \tag{1.14}$$

where B_a is (m,n) and B_b $(n-m, n)$. This may seem to be unduly restrictive but in fact, as shown below, any LBVP can be re-written in separated form (though at the cost of doubling the size of the problem). Therefore, any theoretical results obtained for LBVPs with separated BCs are applicable also in the case of general BCs. This is the justification for our concentration on

the case of separated BCs throughout much of the remainder of this thesis.

To convert the general LBVP (1.1) into separated form we define an additional ODE $\dot{z}(t) = 0$ (n,1) and let $u(t) = [x(t), z(t)]^T$ (2n,1). The combined ODE can now be written :

$$\begin{bmatrix} \dot{x}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} A(t) & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x(t) \\ z(t) \end{bmatrix} + \begin{bmatrix} f(t) \\ 0 \end{bmatrix} \qquad (1.14a)$$

i.e. $\dot{u}(t) = M(t)u(t) + h(t)$ where $M(t) = \begin{bmatrix} A(t) & 0 \\ 0 & 0 \end{bmatrix}$ and $h(t) = \begin{bmatrix} f(t) \\ 0 \end{bmatrix}$. Now $\dot{z}(t) = 0$ ====> $z(t)$ is constant for all $t \in [a,b]$, i.e. $z(a) = z(b)$. The combined BCs can thus be written:

$$\begin{bmatrix} 0 & 0 \\ B_0 & -I_n \end{bmatrix} \begin{bmatrix} x(a) \\ z(a) \end{bmatrix} + \begin{bmatrix} B_1 & I_n \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x(b) \\ z(b) \end{bmatrix} = \begin{bmatrix} c \\ 0 \end{bmatrix} .$$

Thus the separated form of LBVP (1.1) is :

$$\dot{u}(t) = M(t)u(t) + h(t)$$

$$\bar{B}_0 u(a) + \bar{B}_1 u(b) = \bar{c} \qquad (1.14b)$$

for $a <= t <= b$, where $\bar{B}_0 = \begin{bmatrix} 0 \\ \bar{B}_a \end{bmatrix}$, $\bar{B}_1 = \begin{bmatrix} \bar{B}_b \\ 0 \end{bmatrix}$, $\bar{c} = \begin{bmatrix} c \\ 0 \end{bmatrix}$ and $\bar{B}_a = [\, B_0 \mid -I_n \,]$ and $\bar{B}_b = [\, B_1 \mid I_n \,]$. Note that LBVP (1.14b) is now size (2n,2n).

We can show, as follows, that the condition of the LBVP will not be significantly altered by this conversion. Suppose the condition constants of the original LBVP (1.1) are k_1 and k_2 then from (1.12) :

$$\| x \| \leq k_1 \| c \| + k_2 \| f \|$$

$\Longrightarrow \|x\| \leq k_1 \|\bar{c}\| + k_2 \|h\|$ (1.14c)

Now $\|u\| = \max\{\|x\|, \|z\|\}$ where

$\|z\| = \|B_0 x(a)\|$ because $z(a) = B_0 x(a)$.

If $\max\{\|x\|, \|z\|\} = \|x\|$ then from (1.14c) :

$\|u\| \leq k_1 \|\bar{c}\| + k_2 \|h\|$ and so LBVP (1.14b) also has condition constants k_1 and k_2.

If $\max\{\|x\|, \|z\|\} = \|z\|$ then from (1.14c) :

$\|x(a)\| <= \|x\| \leq k_1 \|\bar{c}\| + k_2 \|h\|$

$\Longrightarrow \|B_0\| . \|x(a)\| \leq k_1 \|B_0\| . \|\bar{c}\| + k_2 \|B_0\| . \|h\|$

$\Longrightarrow \|z\| \leq k_1 \|B_0\| . \|\bar{c}\| + k_2 \|B_0\| . \|h\|$

$\Longrightarrow \|u\| \leq \bar{k}_1 \|\bar{c}\| + \bar{k}_2 \|h\|$ where $\bar{k}_1 = k_1 \|B_0\|$ and $\bar{k}_2 = k_2 \|B_0\|$ are the condition constants of LBVP (1.14b).

Assume now that our original LBVP (1.1) has been written in separated BC form i.e.

$\dot{x}(t) = A(t)x(t) + f(t)$ (1.15a)

$B_0 x(a) + B_1 x(b) = c$ (1.15b)

for $a <= t <= b$, where the size of the problem is (n,n) with

$$B_0 = \begin{bmatrix} 0 \\ B_a \end{bmatrix} \begin{matrix} n-m \\ m \end{matrix} \quad \text{and} \quad B_1 = \begin{bmatrix} B_b \\ 0 \end{bmatrix} \begin{matrix} n-m \\ m \end{matrix}.$$

It is shown in [12] that in order for LBVP (1.15) to be well conditioned it is necessary that the ODE (1.15a) is dichotomic. Moreover, if it is given that the LBVP is well conditioned then the row dimensions of B_a and B_b must respectively match the dimensions of the decay and growth subspaces i.e. $m = q$ and $n - m = p$, where q is the dimension of the decay subspace and p that of the growth subspace,so that the exponentially forward

decaying components in the solution are determined at the left hand side (t = a) and the growing components at the right hand side (t = b). This means that the well conditioning of the LBVP (1.15) imposes natural constraints on the BCs as follows. If u(t) is a solution of the homogeneous system $\dot{x}(t) = A(t)x(t)$ for which $B_a\, u(a) = 0$ then this implies that $u(t) \notin$ span $\Phi_2(t)$ (where $\Phi_2(t)$ is as defined in (1.7a)) i.e. u(t) must be either a significantly or moderately forward growing solution. Similarly, if $B_b\, u(b) = 0$ then u(t) must be a significantly or moderately forward decaying solution. We utilise this result later to justify the stability of the auxiliary IVPs obtained from a well conditioned decoupling transformation. We may also note in this connection that if LBVP (1.15) is well conditioned then the fundamental solution Y(t) of $\dot{x}(t) = A(t)x(t)$ which satisfies :

$B_a\, Y(a) + B_b\, Y(b) = I_n$ will be dichotomic as in (1.7a) i.e. if $Y(t) = [\, Y_1(t) \mid Y_2(t)\,]$ then span $Y_1(t)$ and span $Y_2(t)$ will be growth and decay subspaces respectively. This is so because : $B_a\, Y(a) + B_b\, Y(b) = I_n$ ====>

$$\begin{bmatrix} 0 \\ B_a \end{bmatrix} [Y_1(a) \mid Y_2(a)] + \begin{bmatrix} B_b \\ 0 \end{bmatrix} [Y_1(b) \mid Y_2(b)] = I_n \quad ====>$$

$$\begin{bmatrix} B_b\, Y_1(b) & B_b\, Y_2(b) \\ B_a\, Y_1(a) & B_a\, Y_2(a) \end{bmatrix} = \begin{bmatrix} I_p & 0 \\ 0 & I_q \end{bmatrix} \quad ====>$$

$B_b\, Y_2(b) = 0$ and $B_a\, Y_1(a) = 0$.

We said earlier that if k_1 and k_2 are the conditioning constants of LBVP (1.15) (where k_1 and k_2 are as defined

in (1.11a,b)) then the LBVP is well conditioned iff k_2 is small. However, if we are given that the ODE (1.15a) is dichotomic and also that k_1 is small then this implies that k_2 is small [12] i.e. that LBVP (1.15) is well conditioned. In other words, for a LBVP with a dichotomic differential system we can take k_1 as the conditioning constant. Note that a forward IVP is a special case of a separated LBVP (1.15) for which all n conditions are given at $t = a$ i.e. $m = n$. This re-affirms what we said earlier viz. that a well conditioned dichotomic IVP will have a decay subspace of dimension n i.e. all of its kinematic eigenvalues will be such that : $\forall\ \alpha, \beta \in [a,b]$

$$\text{Re} \int_{\alpha}^{\beta} \sigma_i(t)\, dt < 0 \quad \text{for} \quad 1 <= i <= n.$$

Stable decoupling transformations

We turn now to an explanation of what we mean by a stable decoupling transformation of a LBVP and we introduce two important examples of this. Assume that LBVP (1.15) is well conditioned (i.e. that $m = q$) and partition :

$$A(t) = \begin{bmatrix} A_{11}(t) & A_{12}(t) \\ A_{21}(t) & A_{22}(t) \end{bmatrix} \begin{matrix} p \\ q \end{matrix} \begin{matrix} \\ p \quad q \end{matrix}, \quad c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \begin{matrix} p \\ q \end{matrix}, \quad x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \begin{matrix} p \\ q \end{matrix}$$

and let $B_a = \underset{p}{[\, L_0} \mid \underset{q}{L_1}\,]_q$ and $B_b = \underset{p}{[\, L_2} \mid \underset{q}{L_3}\,]_p$ where L_1 and L_2 are assumed to be non-singular. Now make the kinematic similarity transformation $x(t) = T(t)y(t)$ to obtain the transformed system : $\dot{y}(t) = V(t)y(t) + g(t)$ where

$V(t) = T^{-1}(t)\{ A(t)T(t) - \dot{T}(t) \}$ and $g(t) = T^{-1}(t)f(t)$. Suppose that $T(t) = [\, T_1(t) \mid T_2(t) \,]$ can be chosen so as to make $V(t)$ block upper triangular i.e. so that $V_{21}(t) = 0$ for all t. (We give examples later of two such transformations). LBVP (1.15) has thus been transformed into the LBVP :

$$\dot{y}(t) = V(t)y(t) + g(t) \qquad (1.16a)$$

$$B_0\, T(a)y(a) + B_1 T(b)y(b) = c \qquad (1.16b)$$

for $a <= t <= b$.

Since (as shown earlier) a well conditioned kinematic similarity transformation preserves the condition of a LBVP, the assumption that LBVP (1.15) is well conditioned will ensure that (1.16) is also.

The initial BC of LBVP (1.16) is $B_a\, T(a)y(a) = c_2$

$$\Longrightarrow \quad [\, B_a\, T_1(a) \mid B_a\, T_2(a) \,] \begin{bmatrix} y_1(a) \\ y_2(a) \end{bmatrix} = c_2 \,.$$

Thus if $T_1(a)$ is chosen to satisfy : $B_a\, T_1(a) = 0$ (1.17)

then we get $B_a\, T_2(a) y_2(a) = c_2$ or $y_2(a) = [B_a\, T_2(a)]^{-1} c_2$ (1.18)*.

This choice of $T_1(a)$ to satisfy (1.17) is in fact the only possible practical choice that will decouple the initial BC of (1.16b) and so provide us with initial conditions (1.18) for the integration of the forward sweep auxiliary IVP. Fortunately, (1.17) also serves (in the case of a well conditioned LBVP) to ensure that the auxiliary IVPs will both be stable in their respective directions, as we now show.

From (1.16a), the ODE of the transformed LBVP are

* The non-singularity of matrix $B_a T_2(a)$ is ensured by the assumption that the LBVP has a unique solution.

$$\begin{bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \end{bmatrix} = \begin{bmatrix} V_{11}(t) & V_{12}(t) \\ 0 & V_{22}(t) \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} + \begin{bmatrix} g_1(t) \\ g_2(t) \end{bmatrix}$$

i.e. $\dot{y}_1(t) = V_{11}(t)y_1(t) + V_{12}(t)y_2(t) + g_1(t)$ (1.19)

$\dot{y}_2(t) = V_{22}(t)y_2(t) + g_2(t)$ (1.20)

where IVP (1.20) is to be integrated forwards from initial value (1.18) and then IVP (1.19) backwards from a value $y_1(b)$ yet to be determined. The fact that LBVP (1.16) is well conditioned implies that if $y(t)$ is a solution of $\dot{y}(t) = V(t)y(t)$ for which $[B_a T(a)]y(a) = 0$ (i.e. for which $[0 \mid B_a T_2(a)]\, y(a) = 0$) then $y(t)$ must be a forward growth vector. This means that any vector of the form $(\, y_1(t)\, ,\, 0\,)^T$ must be a forward growth vector, for any initial value $y_1(a)$.

Now $\dot{y}(t) = V(t)y(t)$ ====>

$\dot{y}_1(t) = V_{11}(t)y_1(t) + V_{12}(t)y_2(t)$ (1.21a)

$\dot{y}_2(t) = V_{22}(t)y_2(t)$ (1.21b).

From (1.21b), if $y_2(a) = 0$ then $y_2(t) = 0$ for all t and so (1.21a) becomes $\dot{y}_1(t) = V_{11}(t)y_1(t)$ (1.22).

Thus if $(\, y_1(t)\, ,\, 0\,)^T$ is a solution of $\dot{y}(t) = V(t)y(t)$ then $y_1(t)$ will be a solution of (1.22) and so if we integrate this equation forwards from any initial value $y_1(a)$ the solution $y_1(t)$ will be a forward growth vector of (1.22) i.e. if we integrate (1.22) backwards starting from any initial value $y_1(b)$ then the solution $y_1(t)$ will be a decay vector in this direction. Now the backward IVP (1.19) can be written :

$\dot{y}_1(t) = V_{11}(t)y_1(t) + p_1(t)$ where $p_1(t) = V_{12}(t)y_2(t) + g_1(t)$

and so we see that this IVP will be stable in the backward

direction for any given initial value $y_i(b)$.

Later we justify the stability of the forward IVP (1.20) by actually relating the condition constant C_I of this IVP to the condition constants of the given LBVP (1.15).

Thus we see that a well conditioned transformation $T(t)$ for which $V_{21}(t) = 0$ for all t and for which $B_a T_1(a) = 0$ will split the spectrum of kinematic eigenvalues of $A(t)$ so that the p kinematic eigenvalues of $V_{11}(t)$ (p,p) will be such that : $\forall\ \alpha, \beta \in [a,b]$

$$\text{Re} \int_{\alpha}^{\beta} \sigma_i(t)\, dt > 0 \qquad (1 <= i <= p)$$

and the q kinematic eigenvalues of $V_{22}(t)$ (q,q) will be such that :

$$\text{Re} \int_{\alpha}^{\beta} \sigma_i(t)\, dt < 0 \qquad (p+1 <= i <= n)$$, thereby ensuring the stability of the forward and backward sweep IVPs (1.20) and (1.19) respectively.

We now introduce two important practical examples of continuous (well conditioned) decoupling transformations viz. the Riccati and the continuous orthonormal.

The Riccati transformation

This is defined by : $T(t) = \begin{bmatrix} I_p & 0 \\ R(t) & I_q \end{bmatrix}$ (1.23)

where $R(t)$ (q,p) is the Riccati function matrix. Note that

$T^{-1}(t) = \begin{bmatrix} I_p & 0 \\ -R(t) & I_q \end{bmatrix}$. We now find the conditions imposed on $R(t)$ in order that $T(t)$ will transform the given ODE (1.15a) into ODE (1.16a) with $V(t)$ upper block triangular and how, in this case, $V_{11}(t)$, $V_{12}(t)$ and $V_{22}(t)$ will each depend on $R(t)$.

From $V(t) = T^{-1}(t)\{ A(t)T(t) - \dot{T}(t) \}$ we obtain the Lyapunov equation for $T(t)$ viz. $\dot{T}(t) = A(t)T(t) - T(t)V(t)$ ====>

$$\begin{bmatrix} 0 & 0 \\ \dot{R}(t) & 0 \end{bmatrix} = \begin{bmatrix} A_{11}(t) & A_{12}(t) \\ A_{21}(t) & A_{22}(t) \end{bmatrix}\begin{bmatrix} I_p & 0 \\ R(t) & I_q \end{bmatrix} - \begin{bmatrix} I_p & 0 \\ R(t) & I_q \end{bmatrix}\begin{bmatrix} V_{11}(t) & V_{12}(t) \\ 0 & V_{22}(t) \end{bmatrix}$$

====> $0 = A_{11}(t) + A_{12}(t)R(t) - V_{11}(t)$

$0 = A_{12}(t) - V_{12}(t)$

$\dot{R}(t) = A_{21}(t) + A_{22}(t)R(t) - R(t)V_{11}(t)$

$0 = A_{22}(t) - R(t)V_{12}(t) - V_{22}(t)$.

Thus : $V_{11}(t) = A_{11}(t) + A_{12}(t)R(t)$

$V_{12}(t) = A_{12}(t)$ and $V_{22}(t) = A_{22}(t) - R(t)A_{12}(t)$.

We also see that $R(t)$ must satisfy the Riccati equation :

$$\dot{R}(t) = A_{21}(t) + A_{22}(t)R(t) - R(t)A_{11}(t) - R(t)A_{12}(t)R(t) \qquad (1.24a)$$

the initial conditions for which are obtained from (1.17) viz.

$B_a T_1(a) = 0$ ====> $[\, L_0 \mid L_1 \,]\begin{bmatrix} I_p \\ R(a) \end{bmatrix} = 0$ ====>

$L_0 + L_1 R(a) = 0$ ====> $R(a) = -L_1^{-1} L_0$ (1.24b).

From the transformation equation $x(t) = T(t)y(t)$ we get :

$$\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} I_p & 0 \\ R(t) & I_q \end{bmatrix}\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} \text{ ===> } x_1(t) = y_1(t) \qquad (1.25a)$$

$$x_2(t) = R(t)y_1(t) + y_2(t) \qquad (1.25b)$$

where (1.25b) is the Riccati transformation equation. Equation (1.18) becomes : $y_2(a) = \left[[\, L_0 \mid L_1 \,] \begin{pmatrix} 0 \\ I_q \end{pmatrix} \right]^{-1} c_2$ i.e.

$$y_2(a) = L_1^{-1} c_2 \qquad (1.26)$$

and $g(t) = T^{-1}(t) f(t)$ ===> $g_1(t) = f_1(t)$ (1.27a)

$$g_2(t) = -R(t) f_1(t) + f_2(t) \qquad (1.27b).$$

At $t = b$, the final BC of (1.16b) is $B_b T(b) y(b) = c_1$ ====>

$$[\, L_2 \mid L_3 \,] \begin{bmatrix} I_p & 0 \\ R(b) & I_q \end{bmatrix} y(b) = c_1 \quad ====>$$

$$[\, L_2 + L_3 R(b) \mid L_3 \,] \begin{bmatrix} y_1(b) \\ y_2(b) \end{bmatrix} = c_1 \quad ====>$$

$$y_1(b) = [\, L_2 + L_3 R(b) \,]^{-1} [\, c_1 - L_3 y_2(b) \,] \qquad (1.28).$$

The forward ODE (1.20) now is :

$$\dot{y}_2(t) = \{ A_{22}(t) - R(t) A_{12}(t) \} y_2(t) + \{ f_2(t) - R(t) f_1(t) \} \qquad (1.29)$$

and the backward ODE (1.19) is :

$$\dot{y}_1(t) = \{ A_{11}(t) + A_{12}(t) R(t) \} y_1(t) + A_{12}(t) y_2(t) + f_1(t) \qquad (1.30)$$

The basic outline of the solution algorithm is therefore :

(i) integrate simultaneously forwards (from $t = a$ to $t = b$) ODEs (1.24a) and (1.29) using the initial conditions (1.24b) and (1.26) respectively

(ii) integrate backwards (from $t = b$ to $t = a$) ODE (1.30) using initial condition (1.28).

(iii) obtain the solution $x(t)$ of LBVP (1.15) from (1.25a,b).

The above describes the double sweep Riccati method one disadvantage of which is that considerable storage capacity is

required in the forward sweep. In practice, this is usually avoided by using the invariant imbedding technique whereby all the integrations are performed in one direction. However, this does not overcome the main drawback of the Riccati method which is that the solution $R(t)$ of equation (1.24a,b) may become unbounded at some value of $t \in [a,b]$ because $R(t)$ has a pole at some point in $[a,b]$.

This is tantamount to saying that we must keep transformation $T(t)$ (as defined in (1.23)) well conditioned.

In Chapter 4 we look again at the Riccati transformation method in more detail and explain the operation of invariant imbedding and how we can keep the Riccati transformation well conditioned by the strategy of re-imbedding whenever necessary to avoid $\|R(t)\|$ becoming too large. In this connection, we also describe another method (the Compound Matrix method [8]) which is related to the Riccati method and in which the singularities of $R(t)$ are removed.

We now demonstrate the stability of the Riccati transformation by relating the condition number C_I of the forward auxiliary equation (1.29,26) to the condition constants of the given LBVP (1.15). Consider the case of inhomogeneous BC (1.15b) and assume that $c_2 \neq 0$ (this may require the reversal of the direction of the problem). From (1.12) we have $\|x\| \leq k_1 \|c\| + k_2 \|f\|$ where k_1 and k_2 are the condition constants of LBVP (1.15).

The Riccati transformation is $T(t) = \begin{bmatrix} I_p & 0 \\ R(t) & I_q \end{bmatrix}$.

To show the importance of keeping $T(t)$ well conditioned we consider the case where $R(t)$ is exponentially growing for $t > a$ or has a pole in $[a,b]$.

Now $y(t) = T^{-1}(t)x(t)$ $\Longrightarrow$ $\|y(t)\| \leq \|T^{-1}\|.\|x\|$

$\Longrightarrow$ $\|y(t)\| \leq \{\|R(b)\| + 1\}.\|x\|$ (1.30a)

where $R(b)$ is the value of $R(t)$ at the final point $t = b$ unless $R(t)$ has a pole in $[a,b]$ in which case $R(b)$ is the value of $R(t)$ at the time of re-imbedding (see Chapter 4).

From (1.24b) and (1.26) respectively we have :

$R(a) = -L_1^{-1} L_0$ and $y_2(a) = L_1^{-1} c_2$.

Thus $L_1 R(a) = -L_0$ $\Longrightarrow$ $\|L_0\| \leq \|L_1\|.\|R(a)\|$

$\Longrightarrow$ $\dfrac{\|L_0\|}{\|L_1\|} \leq \|R(a)\|$

Now suppose that the rate of growth of $R(t)$ is such that

$$\|R(b)\| + 1 \leq \bar{\beta}\left[\frac{\|L_0\| + 1}{\|L_1\|}\right].\|c_2\| \qquad (1.30b)$$

where $\bar{\beta}$ is a scalar > 1 i.e.

$$\|R(b)\| + 1 \leq \bar{\beta}\left[\|R(a)\| + \frac{1}{\|L_1\|}\right].\|c_2\| \qquad (1.30c)$$

From (1.26) : $L_1 y_2(a) = c_2$ $\Longrightarrow$ $\|c_2\| \leq \|L_1\|.\|y_2(a)\|$

$\Longrightarrow$ $\dfrac{\|c_2\|}{\|L_1\|} \leq \|y_2(a)\|$

From (1.30b) : $\|R(b)\| + 1 \leq \bar{\beta}.\left[\|L_0\| + 1\right].\dfrac{\|c_2\|}{\|L_1\|}$

$$\Longrightarrow \quad \| R(b) \| + 1 \;<=\; \bar{\beta}\left[\| L_0 \| + 1 \right] . \| y_2(a) \|$$

Now from (1.30a), since $\| y_2(t) \| <= \| y(t) \|$ we have :

$$\| y_2(t) \| <= \bar{\beta} \{ \| L_0 \| + 1 \} . \| y_2(a) \| . \{ k_1 \| c \| + k_2 \| f \| \}$$

i.e. $\| y_2(t) \| <= C_I \| y_2(a) \|$ where the condition constant C_I of the forward IVP is given by :

$$C_I = \bar{\beta} \{ \| L_0 \| + 1 \} \{ k_1 \| c \| + k_2 \| f \| \} .$$

Thus if k_1 and k_2 are small (i.e. the LBVP is well conditioned) and $\bar{\beta}$ is small (i.e. the Riccati solution remains small) then C_I will be small (i.e. the forward auxiliary IVP will be well conditioned). Also the condition number of transformation matrix $T(b)$ is given by

$$\text{cond } T(b) = \| T(b) \| . \| T^{-1}(b) \| = \{ 1 + \| R(b) \| \}^2 .$$

Thus if $\| R(b) \|$ becomes large then so will both cond $T(b)$ and C_I. This shows the importance of keeping transformation $T(t)$ well conditioned.

The Continuous Orthonormal method

Here we obtain a transformation $T(t) = [T_1^{p}(t) \mid T_2^{q}(t)]$, where $T_1(t)$ is (n,p) and $T_2(t)$ (n,q), which is orthonormal for all t and for which the transformed system matrix $V(t)$ of (1.16a) will be block upper triangular. It can be shown that in order for this to be so $T_1(t)$ and $T_2(t)$ must satisfy ODEs of the following form (see appendix [1-4]) :

$$\dot{T}_1(t) = A(t) T_1(t) - T_1(t) C_{11}(t) \quad \text{and}$$

$$\dot{T}_2(t) = -A^T(t) T_2(t) + T_2(t) C_{22}^T(t)$$

where $C_{11}(t)$ (p,p) and $C_{22}(t)$ (q,q) must be such that :

$C_{ii}(t) + C_{ii}^T(t) = T_i^T(t) \{ A(t) + A^T(t) \} T_i(t)$

for $1 <= i <= 2$. Then $V_{ii}(t) = C_{ii}(t)$ ($1 <= i <= 2$) and $V_{12}(t) = T_1^T(t) \{ A(t) + A^T(t) \} T_2(t)$. There are various possibilities for $C_{ii}(t)$ of which the obvious one is $C_{ii}(t) = T_i^T(t)A(t)T_i(t)$. With this choice we get :

$$\dot{T}_1(t) = \{ I_n - T_1(t)T_1^T(t) \} A(t)T_1(t) \qquad (1.31)$$

$$\dot{T}_2(t) = \{ -I_n + T_2(t)T_2^T(t) \} A^T(t)T_2(t) \qquad (1.32).$$

The initial conditions for equation (1.31) are obtained from (1.17) viz. $B_a T_1(a) = 0$ i.e. $T_1(a)$ is chosen to be a unit orthogonal column set such that $[L_0 \mid L_1] T_1(a) = 0$. Then the initial condition for (1.32) is obtained by choosing $T_2(a)$ to be any unit orthogonal column set such that $T_2^T(a)T_1(a) = 0$ i.e. such that $T(a)$ (n,n) is a unit orthogonal matrix.

Also $g(t) = T^{-1}(t)f(t) = T^T(t)f(t) = \begin{bmatrix} T_1^T(t) \\ T_2^T(t) \end{bmatrix} . f(t)$ i.e.

$g_1(t) = T_1^T(t)f(t)$ and $g_2(t) = T_2^T(t)f(t)$.

At $t = b$ the final BC of (1.16b) is $B_b T(b)y(b) = c_1$

$$\Longrightarrow \quad [B_b T_1(b) \mid B_b T_2(b)] \begin{bmatrix} y_1(b) \\ y_2(b) \end{bmatrix} = c_1$$

$$\Longrightarrow \quad y_1(b) = [B_b T_1(b)]^{-1} [c_1 - B_b T_2(b)y_2(b)] \qquad (1.33).$$

The forward IVP (1.20) thus becomes :

$$\dot{y}_2(t) = \{ T_2^T(t)A(t)T_2(t) \} y_2(t) + T_2^T(t)f(t) \qquad (1.34a)$$

$$y_2(a) = [B_a T_2(a)]^{-1} c_2 \qquad (1.34b)$$

and the backward IVP (1.19) is :

$$\dot{y}_1(t) = \{T_1^T(t)A(t)T_1(t)\}y_1(t) + T_1^T(t)\{A(t) + A^T(t)\}T_2(t)y_2(t) + T_1^T(t)f(t) \qquad (1.35)$$

with $y_1(b)$ as given in (1.33).

Thus the double sweep orthonormalisation algorithm would be :

(i) integrate simultaneously forwards (from $t = a$ to $t = b$) the ODEs (1.31), (1.32) and (1.34) from their respective initial values

(ii) integrate backwards (from $t = b$ to $t = a$) IVP (1.35)

(iii) obtain the solution $x(t)$ of LBVP (1.15) from the transformation equation $x(t) = T(t)y(t)$. -

However, as with the Riccati method, the above algorithm would entail considerable storage of $T(t)$ values in the forward sweep for use in the backward sweep, and so in practice we use invariant imbedding to enable us to integrate all of the ODEs in one direction. Also, in theory, matrix $T(t)$, as obtained from the solution of ODEs (1.31) and (1.32), should remain unit orthogonal for all t. However, in practice, it has been found that this may not always be so and that $T(t)$ may become ill conditioned before $t = b$ is reached, particularly if the integrations are being performed by a fixed-step Runge Kutta integrator. To overcome this difficulty an adaptation of equations (1.31) and (1.32), involving the 'generalised inverses' of $T_1(t)$ and $T_2(t)$, has been suggested [3]. In Chapter 3 we look in detail at the operation of invariant imbedding orthonormalisation methods employing 'generalised inverses'.

We saw earlier that the choice of $T_1(a)$ to satisfy the condition $B_a T_1(a) = 0$ (1.17) enables us to obtain the initial value of $y_2(a)$ for the forward sweep and is also essential to ensure the stability of the auxiliary IVPs of the method (whether Riccati or continuous orthonormalisation). Note also that this condition implies (for a well conditioned LBVP) that the columns of $T_1(a)$ must be forward growth vectors of the system $\dot{x}(t) = A(t)x(t)$ at $t = a$. We can show further [1-5] that this means that span $T_1(t)$ will form a basis for a forward growth subspace of $\dot{x}(t) = A(t)x(t)$ for all $t >= a$. This is a characteristic property of a stable decoupling transformation.

N.B. In Appendix I :

[1-1] : Expression of single nth. order ODE as a system of simultaneous first order equations.

[1-2] : Derivation of theoretical solution of LBVP in terms of Green's function matrices.

[1-3] : Deflation method for calculation of kinematic eigenvalues of a system matrix.

[1-4] : Derivation of ODEs for $T_1(t)$ and $T_2(t)$ in the orthonormalisation method.

[1-5] : Characteristic property of stable decoupling transformations.

CHAPTER 2

MULTIPLE SHOOTING METHODS

Consider the well conditioned (n,n) LBVP :

$$\dot{x}(t) = A(t)x(t) + f(t) \qquad (2.1a)$$

$$B_o\, x(a) + B_1\, x(b) = c \qquad (2.1b)$$

for $t \in [a,b]$, where $B_o = \begin{bmatrix} 0 \\ B_a \end{bmatrix} \begin{matrix} p \\ q \end{matrix}$, $B_1 = \begin{bmatrix} B_b \\ 0 \end{bmatrix} \begin{matrix} p \\ q \end{matrix}$, $c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} \begin{matrix} p \\ q \end{matrix}$

and B_a is (q,n), B_b (p,n), c_1 $(p,1)$, $p + q = n$.

Single shooting

The most straightforward method of attempting to solve this LBVP is by reduced superposition (complementary function method) whereby the solution $x(t)$ is represented as a linear combination of solutions of associated IVPs as follows. Let :

$$x(t) = X'(t).\alpha + v_o(t) \qquad (2.2)$$

where $X'(t)$ (n,p) is a part fundamental solution of $\dot{x}(t) = A(t)x(t)$ (i.e. $\dot{X}'(t) = A(t)X'(t)$) for which :

$$B_a\, X'(a) = 0 \qquad (2.3)$$

$v_o(t)$ $(n,1)$ is a particular solution of $\dot{x}(t) = A(t)x(t) + f(t)$ (i.e. $\dot{v}_o(t) = A(t)v_o(t) + f(t)$) for which :

$$B_a\, v_o(a) = c_2 \qquad (2.4)$$

and α is a constant $(p,1)$ vector to be determined.

Then from (2.2): $\dot{x}(t) = \dot{X}'(t).\alpha + \dot{v}_o(t)$

$$= A(t)X'(t).\alpha + A(t)v_o(t) + f(t)$$

$= A(t) \{ X'(t).\alpha + v_o(t) \} + f(t)$

$= A(t)x(t) + f(t)$, and so $x(t)$

satisfies ODE (2.1a). Also :

$B_o x(a) = B_o X'(a).\alpha + B_o v_o(a) = [\ 0,\ c_2\ ]^T$ from (2.3,4)

and so $x(t)$ also satisfies the initial BC of (2.1b). If $x(t)$ is to be the solution of LBVP (2.1) then it must also satisfy the final BC i.e. $B_b x(b) = c_1$ ====>

$B_b X'(b).\alpha + B_b v_o(b) = c_1$ ====>

$\{ B_b X'(b) \} \alpha = c_1 - B_b v_o(b)$ ===> $M\alpha = d$ (2.5)

where $M(p,p) = B_b X'(b)$ and $d(\bar{p},1) = c_1 - B_b v_o(b)$.

Therefore, since $X'(t)$ and $v_o(t)$ can be computed by forward integration from $t = a$ to $t = b$ of the homogeneous and inhomogeneous system from initial conditions satisfying (2.3) and (2.4) respectively, in theory, vector α can be found by solving the linear system (2.5) and hence the solution $x(t)$ of the LBVP obtained for all $t \in [a,b]$ from equation (2.2). In practice, however, if system (2.1a) is 'stiff' (i.e. if the kinematic eigenvalues of $A(t)$ are widely separated in real part) then as these forward integrations proceed the columns of $X'(t)$ may gradually lose their independence and also vector $v_o(t)$ may become dependent on span $X'(t)$. This would cause linear system (2.5) to be ill conditioned with consequent loss in accuracy of the calculated value of α. Moreover, further errors in the computed value of solution $x(t)$ may be caused by cancellation errors arising when $x(t)$ is

obtained from equation (2.2) due to large mod values in the calculated components of $X^{!}(t)$ and $v_o(t)$.

Note that these difficulties stem not from the condition of the given LBVP (2.1) but from the fact that the dichotomy of $A(t)$ is such that system $\dot{x}(t) = A(t)x(t)$ possesses rapidly (forward) growing and decaying solutions which cause the IVPs of the method to be ill conditioned. (If we reverse the statement of the problem i.e. solve from $t = b$ to $t = a$ we will encounter the same difficulties due to the forward decay vectors which will grow rapidly for decreasing values of t). It was the need to overcome these basic difficulties of the single shooting method that led to the development of multiple (parallel) shooting and stabilised marching techniques such as Conte's reorthonormalisation method (which we discuss in this Chapter) and also to the Riccati transformation and continuous reorthonormalisation methods (which we deal with in later Chapters). As we shall see, the success of all of these methods depends on their ability to produce well conditioned IVPs (or their equivalent).

Note that the single shooting method described above employs reduced superposition requiring only $(p + 1)$ forward integrations. Alternatively, we could use full superposition (variation of parameters method) in which we express the solution vector $x(t)$ in the form :

$$x(t) = X(t).\alpha + v_o(t) \qquad (2.6)$$

where now $X(t)$ (n,n) and $v_o(t)$ $(n,1)$ are any fundamental

solution and any particular solution respectively of systems $\dot{x}(t) = A(t)x(t)$ and $\dot{x}(t) = A(t)x(t) + f(t)$ and α is now a constant (n,1) vector to be determined. In this case, from BC (2.1b) we get :

$$B_0 \{ X(a)\alpha + v_0(a) \} + B_1 \{ X(b)\alpha + v_0(b) \} = c$$

$$\Longrightarrow \quad Q\alpha = \gamma \qquad (2.7)$$

where $Q(n,n) = B_0 X(a) + B_1 X(b)$ and

$$\gamma(n,1) = c - B_1 v_0(b) - B_0 v_0(a) .$$

Q and γ are obtained by solving (n + 1) IVPs and hence α can (in theory) be found from (2.7) and then x(t) from (2.6). Obviously, the same drawbacks apply here as in the reduced superposition method and (n - p) more IVPs must be solved. Full superposition is necessary however when the BCs (2.1b) are not separated.

Although any independent initial conditions can be used for the $\dot{X}(t) = A(t)X(t)$ and $\dot{v}_0(t) = A(t)v_0(t) + f(t)$ IVPs it is usual to employ the standard conditions : $X(a) = I_n$ and $v_0(a) = 0$ in which case Q and γ simplify to :

$Q = B_0 + B_1 X(b)$, $\gamma = c - B_1 v_0(b)$ and $\alpha = x(a)$.

Multiple (parallel) shooting

We now turn to multiple shooting in which the range [a,b] of the LBVP is divided into subintervals according to some criterion (see later) and then in each subinterval separately we use single shooting to find the general solution which

satisfies the ODE $\dot{x}(t) = A(t)x(t) + f(t)$ for all t in that subinterval. These subinterval solutions are then 'matched up' at each of the internal nodes and also with the given initial and terminal values prescribed by the BC. Thus we obtain the overall solution vector $x(t)$ of the LBVP which is continuous over $[a,b]$ and which satisfies both the given ODE $\dot{x}(t) = A(t)x(t) + f(t)$, for all $t \in [a,b]$, and the BC.

More precisely, the interval $[a,b]$ is subdivided into N subintervals by the insertion of $(N-1)$ internal nodes thus : $a = t_0 < t_1 < t_2 < \ldots\ldots\ldots < t_{N-1} < t_N = b$. Then in the standard multiple shooting variant we compute for each subinterval $[t_i, t_{i+1}]$ $(0 <= i <= N-1)$ the fundamental solution $X_i(t)$ for which $X_i(t_i) = I_n$ and the particular solution $v_i(t)$ for which $v_i(t_i) = 0$. Using full superposition the corresponding subinterval general solution vectors $x_i(t)$ are : $x_i(t) = X_i(t)\alpha_i + v_i(t)$ (2.8) for $t \in [t_i, t_{i+1}]$ where α_i $(n,1)$ $(0 <= i <= N-1)$ are constant $(n,1)$ vectors to be determined.

For continuity at the internal nodes t_i $(1 <= i <= N-1)$ we must have :

$x_i(t_{i+1}) = x_{i+1}(t_{i+1})$ for $0 <= i <= N-2$

$\Longrightarrow X_i(t_{i+1})\alpha_i + v_i(t_{i+1}) = X_{i+1}(t_{i+1})\alpha_{i+1} + v_{i+1}(t_{i+1})$

$\Longrightarrow X_i(t_{i+1})\alpha_i + v_i(t_{i+1}) = \alpha_{i+1}$ (2.9).

Also to satisfy the given BC (2.1b) we must have :

$B_0 x_0(t_0) + B_1 x_{N-1}(t_N) = c \Longrightarrow$

$B_0\alpha_0 + B_1 \{ X_{N-1}(t_N)\alpha_{N-1} + v_{N-1}(t_N) \} = c$

$$\Longrightarrow \quad B_0 \alpha_0 + B_1 \alpha_N = c \qquad (2.10)$$

where $\alpha_N = X_{N-1}(t_N)\alpha_{N-1} + v_{N-1}(t_N)$.

Equations (2.9) and (2.10) together can be written in the form :

$$Z\alpha = d \qquad (2.11)$$

where $\alpha = [\alpha_0, \alpha_1 \ldots\ldots, \alpha_N]^T$ ($\{N+1\}n$, 1)

$d = [c, v_0(t_1), \ldots . v_{N-1}(t_N)]^T$ ($\{N+1\}n$, 1)

$$Z = \begin{bmatrix} B_0 & & & & B_1 \\ -X_0(t_1) & I_n & & & \\ & -X_1(t_2) & I_n & & \\ & & & & \\ & & & -X_{N-1}(t_N) & I_n \end{bmatrix}$$

($\{N+1\}n$, $\{N+1\}n$).

The above set of linear multiple shooting equations can now be solved by Gaussian elimination to obtain vector α . Hence the solution $x(t)$ of the LBVP is computed piecewise from equation (2.8).

The success of the method can depend very much on the number and positions of the internal nodes (particularly if a fixed step Runge Kutta integrator is used to solve the IVPs for the subinterval fundamental and particular solutions). If the nodes are chosen such that $\| X_i(t_{i+1}) \| <= \bar{k}$ ($0 <= i <= N-1$) then it can be shown [6] that :

$$\text{cond } Z = \| Z \| \| Z^{-1} \| <= \{ \bar{k} + 1 \} . \left(k_1 + \frac{k_2 N}{b - a} \right)$$

where k_1 and k_2 are the condition constants of the given LBVP. Thus for a well conditioned problem the multiple shooting matrix Z will also be well conditioned provided that the nodes t_i $(1 <= i <= N-1)$ are inserted frequently enough to limit sufficiently the growth of the fundamental solutions $X_i(t)$ $(0 <= i <= N-1)$. Unfortunately, for a LBVP for which the system matrix $A(t)$ of the ODE is 'stiff' this could mean that a large number of subintervals may be required resulting in a very large linear system (2.11) to be solved, particularly if the problem size n is large also. It was this weakness of the multiple shooting method with regard to 'stiff' problems that motivated interest in the development of the Riccati and continuous orthonormalisation methods (to be discussed in later Chapters). An advantage of multiple shooting over the latter methods, however, is that it is also directly applicable in the case where the BC are not separated .

We said earlier in Chapter 1 that the success of any method in solving a LBVP with a dichotomic ODE depended on the ability of the method to produce well conditioned IVPs (or their equivalent) by correctly decoupling the (forward) growth components in the fundamental solutions from the decay components. In the case of the multiple shooting method it is not immediately obvious as to how this is achieved, because the decoupling of the differential system occurs implicitly as the

multiple shooting equations $Z\alpha = d$ (2.11) are solved by the Gaussian elimination process. As evidence of this it is shown in [6]* that for the case where no row interchanges are allowed in the Gauss process the latter is equivalent to the Riccati (single imbedding) transformation method in that as the Gausssian elimination process reduces matrix Z to upper triangular form this automatically generates the Riccati solution $R(t)$ down the leading diagonal and so the process is equivalent to the forward integration of this IVP. This might perhaps lead us to expect that (by analogy) the operation of the Gaussian elimination process, where full row interchanges are allowed so as to employ the max modulus element in each column as pivot, would be equivalent to the Riccati method where a re-imbedding (see Chapter 4) occurs at each of the multiple shooting nodes t_i ($1 <= i <= N - 1$). However, how the Gauss process achieves this in this case (if indeed it does) has not yet been clearly established.

Stabilised marching

Multiple shooting methods can be split into two types : those which employ 'parallel' shooting and those which are examples of stabilised marching. The standard multiple shooting method that we described in the previous section is an example of 'parallel' shooting because for $1 <= i <= n$ the fundamental solution values $X_i(t_i)$ and $X_{i-1}(t_i)$ are independent as

* Refers only to the case of separated BC with the multiple shooting matrix in slightly different form from that on page 2-6.

also are the particular solution values $v_i(t_i)$ and $v_{i-1}(t_i)$. In fact, $X_i(t_i) = I_n$ and $v_i(t_i) = 0$ for $1 <= i <= N - 1$ and so the integrations for all N subintervals could be performed simultaneously (i.e. in 'parallel') if a pre-selected number of equal subintervals was used. By contrast, in all of the stabilised marching methods the initial values of $X_i(t_i)$ and (in some cases) of $v_i(t_i)$ at the beginning of each subinterval are derived from the corresponding values $X_{i-1}(t_i)$ and $v_{i-1}(t_i)$, respectively, as we now explain.

The following is a description of a stabilised marching method known as discrete re-orthonormalisation (using full superposition). We first describe the algorithm and then we show how it can be regarded as an example of a stable decoupling transformation. As with parallel shooting we subdivide the problem range [a,b] into N subintervals where the nodes are inserted according to some criterion (see later). For each subinterval $[t_i, t_{i+1}]$ $(0 <= i <= N - 1)$ we obtain the fundamental solution $X_i(t)$ of system $\dot{x}(t) = A(t)x(t)$ where the initial value of $X_i(t_i) = [X_i^1(t_i) | X_i^2(t_i)]$ is an orthonormal matrix obtained by a QU decomposition using the Gram Schmidt process viz. $X_{i-1}(t_i) = X_i(t_i) \Gamma_i$ where Γ_i is an upper triangular (n,n) matrix $(1 <= i <= N - 1)$. At $t = t_0 = a$ the initial value $X_0(t_0)$ is obtained by choosing $X_0^1(t_0)$ to be a unit orthogonal column set such that $B_a X_0^1(t_0) = 0$.

Then $X_0^2(t_0)$ is any unit orthogonal column set satisfying $[X_0^2(t_0)]^T . [X_0^1(t_0)] = 0$ i.e. such that $X_0(t_0)$ is an orthonormal matrix. Also for each subinterval $[t_i, t_{i+1}]$ we obtain the particular solution $v_i(t)$ of system $\dot{x}(t) = A(t)x(t) + f(t)$ for which $v_i(t_i) = 0$ for $0 <= i <= N - 1$.

The node t_{i+1} is inserted such that $\| X_i(t_{i+1}) \| < \bar{k}$ where $\bar{k}$ is sufficiently small to ensure that the columns of $X_i(t_{i+1})$ are still linearly independent. This could be done by checking on the value of cond $X_i(t) = \| X_i(t) \| . \| X_i^{-1}(t) \|$ at each step. Alternatively, Mattheij and Staarink [18] advocate using the growth of the particular solution $v_i(t)$ as a guide to the growth of the fundamental solution : when using a variable step Runge Kutta integrator (such as RKF 45) a node is inserted every pth step of the integration of $v_i(t)$, where p is a pre-selected small number. These nodes are then also used as the restart points of the integration of $X_i(t)$.

By (full) superposition the solution $x(t)$ of the LBVP on subinterval $[t_i, t_{i+1}]$ can be written :

$x_i(t) = X_i(t) . \alpha_i + v_i(t)$, where α_i is a constant (n,1) vector ($0 <= i <= N - 1$). Continuity of $x(t)$ at node t_{i+1} demands that : $x_i(t_{i+1}) = x_{i+1}(t_{i+1})$ ====>

$$X_i(t_{i+1}) \alpha_i - X_{i+1}(t_{i+1}) \alpha_{i+1} = -v_i(t_{i+1}) \qquad (2.12)$$

for $0 <= i <= N - 2$. The BC are : $B_0 x_0(t_0) + B_1 x_{N-1}(t_N) = c$

$$====> \quad B_0 X_0(t_0) \alpha_0 + B_1 X_{N-1}(t_N) \alpha_{N-1} = \gamma \qquad (2.13)$$

where $\gamma = c - B_1 v_{N-1}(t_N)$. Equations (2.12,13) can be combined into a set of Nn linear equations: $Ms = w$ (2.14)

where

$$M = \begin{bmatrix} X_0(t_1) & -X_1(t_1) & & & & \\ & X_1(t_2) & -X_2(t_2) & & & \\ & & & & & \\ & & & & X_{N-2}(t_{N-1}) & -X_{N-1}(t_{N-1}) \\ B_0 X_0(t_0) & & & & & B_1 X_{N-1}(t_N) \end{bmatrix}$$

$s = [\alpha_0, \alpha_1, \ldots\ldots, \alpha_{N-1}]^T$ and

$w = [-v_0(t_1), -v_1(t_2), \ldots\ldots\ldots, -v_{N-2}(t_{N-1}), \gamma]^T$.

If, in (2.14), we now pre-multiply each row block except the last by $-[X_i(t_i)]^{-1} = -[X_i(t_i)]^T$ $(1 <= i <= N-1)$ this converts M into $\bar{M}$ where

$$\bar{M} = \begin{bmatrix} -\Gamma_1 & I_n & & & & \\ & -\Gamma_2 & I_n & & & \\ & & -\Gamma_3 & & & \\ & & & & & \\ & & & & -\Gamma_{N-1} & I_n \\ B_0 X_0(t_0) & & & & & B_1 X_{N-1}(t_N) \end{bmatrix}$$

and w into $\bar{w} = [g_1, g_2, \ldots\ldots\ldots, g_{N-1}, \gamma]^T$ where $g_i = [X_i(t_i)]^T . v_{i-1}(t_i)$. Thus the new linear system corresponding to (2.14) is $\bar{M} s = \bar{w}$ (2.15).

Now $B_o X_o(t_o) = \begin{bmatrix} 0 \\ B_a \end{bmatrix} . [X_o^1(t_o) \mid X_o^2(t_o)]$

$$= \begin{bmatrix} 0 & 0 \\ 0 & B_a X_o^2(t_o) \end{bmatrix}$$ because

$B_a X_o^1(t_o) = 0$ by the initial choice of $X_o(t_o)$.

Also $B_1 X_{N-1}(t_N) = \begin{bmatrix} B_b \\ 0 \end{bmatrix} . X_{N-1}(t_N) = \begin{bmatrix} L & H \\ 0 & 0 \end{bmatrix}$ where

L is (p,p) and H (p,q). Thus from the last (i.e. Nth) row block of (2.15) we get : $B_o X_o(t_o) \alpha_o + B_1 X_{N-1}(t_N) . \alpha_{N-1} = \bar{w}_N$

$\Longrightarrow [L \mid H] \begin{bmatrix} \alpha_{N-1}^1 \\ \alpha_{N-1}^2 \end{bmatrix} = \bar{w}_N^1$ (p,1)

and

$[0 \mid B_a X_o^2(t_o)] \begin{bmatrix} \alpha_o^1 \\ \alpha_o^2 \end{bmatrix} = \bar{w}_N^2$ (q,1) where $\bar{w}_i = \begin{bmatrix} \bar{w}_i^1 \\ \bar{w}_i^2 \end{bmatrix}$ $(1 \le i \le N)$

$\Longrightarrow \quad \alpha_o^2 = [B_a X_o^2(t_o)]^{-1} . \bar{w}_N^2$ (2.16a)

$\alpha_{N-1}^1 = L^{-1} [\bar{w}_N^1 - H \alpha_{N-1}^2]$ (2.16b).

Now the first (N − 1) row blocks of (2.15) can be written recursively as :

$-\Gamma_{i+1} \alpha_i + \alpha_{i+1} = \bar{w}_{i+1} \Longrightarrow \alpha_{i+1} = \Gamma_{i+1} \alpha_i + \bar{w}_{i+1}$ for $0 \le i \le N-2$, where Γ_{i+1} is upper triangular. Hence :

$$\begin{bmatrix} \alpha_{i+1}^1 \\ \alpha_{i+1}^2 \end{bmatrix} = \begin{bmatrix} E_i & F_i \\ 0 & G_i \end{bmatrix} \begin{bmatrix} \alpha_i^1 \\ \alpha_i^2 \end{bmatrix} + \begin{bmatrix} \bar{w}_{i+1}^1 \\ \bar{w}_{i+1}^2 \end{bmatrix}$$ where

$$\Gamma_{i+1} = \begin{bmatrix} E_i & F_i \\ 0 & G_i \end{bmatrix}$$

$\Longrightarrow \quad \alpha_{i+1}^1 = E_i \alpha_i^1 + F_i \alpha_i^2 + \bar{w}_{i+1}^1$ (2.17a)

$\alpha_{i+1}^2 = G_i \alpha_i^2 + \bar{w}_{i+1}^2$ (2.17b).

Iteration (2.17b) is now solved from i = 0 to i = N − 2

starting from initial condition (2.16a) and the values of α_i^2 ($0 <= i <= N-1$) are stored. Using the value of α_{N-1}^2 in (2.16b) enables us to find α_{N-1}^1. Equation (2.17a) is re-written as: $\alpha_i^1 = [E_i]^{-1} . [\alpha_{i+1}^1 - F_i \alpha_i^2 - w_{i+1}^1]$ (2.18) which can be solved backwards from $i = N-2$ to $i = 0$ to obtain the values α_i^1. Thus we have computed the vectors α_i ($0 <= i <= N-1$) from which we can get the solution $x(t)$ of the LBVP in each subinterval $[t_i, t_{i+1}]$ by using $x_i(t) = X_i(t) \alpha_i + v_i(t)$.

To demonstrate the stability of the recursions (2.17a & b) in their respective directions we define the unit orthogonal transformation $T(t_i) = X_i(t_i)$ of the LBVP (2.1) at each node i.e. $x_i(t_i) = T(t_i) . y_i(t_i)$ where $y_i(t_i)$ is the solution vector of the transformed LBVP. In this case,

$y_i(t_i) = T^{-1}(t_i) . x_i(t_i) = [X(t_i)]^T . x_i(t_i)$

$= [X(t_i)]^T . [X_i(t_i) \alpha_i + v_i(t_i)]$

$\Longrightarrow y_i(t_i) = \alpha_i$ since $v_i(t_i) = 0$ $(0 <= i <= N)$.

Now :

(i) the original LBVP (2.1) is assumed to be well conditioned

(ii) the transformation $X_i(t_i)$ is well conditioned for all t_i because we make $X_i(t_i)$ unit orthogonal

(iii) at $t = t_o$ the condition $B_a X_o^1(t_o) = 0$ ensures that the first p columns of $X_o(t_o)$ will form a basis for a (foward) growth space of the dichotomy of $A(t)$ and hence this will be true for $X_o^1(t)$ for all $t \in [t_o, t_1]$

(iv) at each node t_i $(1 <= i <= N-1)$ we perform a QU decomposition of the fundamental solution viz. $X_i(t_i) = X_{i-1}(t_i)(\Gamma_i)^{-1}$ where $(\Gamma_i)^{-1}$ is upper triangular. This means that each of the first p columns of $X_i(t_i)$ is a linear combination of the first p columns of $X_{i-1}(t_i)$ and so span $X_i^1(t_i)$ = span $X_{i-1}^1(t_i)$ i.e. the (forward) growth space basis of $X_i^1(t_i)$ is preserved for all t_i $(1 <= i <= N-1)$ and hence for all $t \in [a,b]$.

Facts (i) to (iv) above imply that the transformed LBVP must also be well conditioned. Now since $\bar{y}_i(t_i) = \alpha_i$ this means that the recursions (2.17a & b) must be stable in their respective directions because they are the backward and forward sweeps of the decoupled system of the transformed LBVP (compare equations (1.19 & 20) of Chapter 1).

Conte's Re-orthonormalisation method [19] :

This is a more economical version of the algorithm described in the previous section which employs reduced superposition and thereby eliminates the forward iteration (2.17b). Let the fundamental solution $X(t)$ (n,n) be partioned $[X^1(t) \mid X^2(t)]$ of which we consider here only $X^1(t)$. The part fundamental solution $X_0^1(t)$ for which $B_a X_0^1(t_0) = 0$ (where $X_0^1(t_0)$ is a unit orthogonal column set) is obtained by forward integration of the homogeneous system $\dot{x}(t) = A(t)x(t)$. Simultaneously we obtain a particular solution $v_0(t)$ corresponding to

$B_a\, v_0(t_0) = c_2$ by forward integration of system $\dot{x}(t) = A(t)x(t) + f(t)$. As in the previous section, a node t_1 is inserted before the columns of $X_0'(t)$ lose their independence and at $t = t_1$ we re-orthonormalise $X_0'(t_1)$ by means of a QU decomposition viz. $X_0'(t_1) = X_1'(t_1).D_0$ where D_0 (p,p) is the upper triangular orthonormalisation matrix and where $X_1'(t_1)$ now has unit orthogonal columns. Also at $t = t_1$ we obtain the orthogonal complement $v_1(t_1)$ of $v_0(t_1)$ from : $v_1(t_1) = v_0(t_1) - X_1'(t_1).[X_1'(t_1)]^T.\, v_0(t_1)$ so that $v_1(t_1)$ is now orthogonal to every column of $X_1'(t_1)$. We continue thus inserting nodes t_i $(1 <= i <= N - 1)$ until $t_N = b$ is reached where a final re-orthonormalisation occurs to convert $X_{N-1}'(t_N)$ into $X_N'(t_N)$ and $v_{N-1}(t_N)$ into $v_N(t_N)$. At each node t_i $(1 <= i <= N)$ the orthonormalisation matrices D_i (p,p) are defined by :

$X_i'(t_{i+1}) = X_{i+1}'(t_{i+1}).D_i$ $(0 <= i <= N - 1)$ where $X_{i+1}'(t_{i+1})$ are the re-orthonormalised part fundamental solutions.

Now, by reduced superposition, in subinterval $[t_0, t_1]$ the solution vector $x_0(t) = X_0'(t).\beta_0 + v_0(t)$ (where β_0 is constant (p,1)) satisfies the ODE (2.1a) of the given LBVP and also the initial BC : $B_a\, x_0(a) = c_2$. Thus the solution vector $x(t)$ will be piecewise continuous of the form :

$x_i(t) = X_i'(t).\beta_i + v_i(t)$ $(0 <= i <= N - 1)$ (2.18)

where $x_i(t)$ is the solution in subinterval $[t_i, t_{i+1}]$, if the continuity conditions : $x_i(t_{i+1}) = x_{i+1}(t_{i+1})$ are satisfied at the nodes t_i $(1 <= i <= N - 1)$. The latter

implies (see [2-1]) that :

$$\beta_{i+1} = D_i \beta_i + [X'_{i+1}(t_{i+1})]^T . v_i(t_{i+1})$$

which can be written :

$$\beta_i = D_i^{-1} . \beta_{i+1} - D_i^{-1} [X'_{i+1}(t_{i+1})]^T . v_i(t_{i+1}) \qquad (2.19).$$

Now from the final BC of (2.1b) we have $B_b\, x(b) = c_1$ ====> $B_b X'_N(t_N)\, \beta_N = c_1 - B_b v_N(t_N)$,from which the value of β_N can be obtained. Hence iteration (2.19) can be solved backwards for β_i ($0 <= i <= N-1$) starting from this value of β_N. The subinterval solution vectors $x_i(t)$, which constitute the solution $x(t)$ of the given LBVP, are now obtained from (2.18). The backward iteration (2.19) corresponds to (2.17a) of the previous section and so its stability is ensured by the same argument as put forward there.

In Appendix I :

[2-1] : Backward iteration of Conte's method.

CHAPTER 3

CONTINUOUS ORTHONORMALISATION METHODS

As we have seen in Chapter 2, if the dichotomy of system $\dot{x}(t) = A(t)x(t)$ is such that $A(t)$ has kinematic eigenvalues (corresponding to some orthogonal transformation $T(t)$) which are large in modulus value, then all of the multiple shooting and stabilised marching methods suffer from the drawback that frequent restarts may be necessary to avoid loss of independence of the columns of fundamental solutions. On the other hand, as we shall see in Chapter 4, if $A(t)$ is rapidly varying, causing rotational activity of the columns of the fundamental solutions, then the Riccati method may require frequent re-imbeddings to prevent the Riccati solution from becoming unbounded in $[a,b]$. To overcome both of these drawbacks was the main motivation for the development of the continuous orthonormalisation methods. There are two principal variants of this method: one due largely to the work of Davey [3], Meyer [4], Bakhvalov [24] and Drury [25], and the other to Van Loon [17] and Mattheij [10]. The latter variation is obviously a decoupling transformation method which employs invariant imbedding whilst the Davey/Meyer method is a double sweep method. We describe the Van Loon method and then, by establishing relationships between this method and that of Davey/Meyer, we show how the latter method also fits into the framework of a well conditioned decoupling transformation method.

Method of Van Loon et al. [17,10] :

The given LBVP is (1.15) which is assumed to be well conditioned i.e. with $m = q$ (the dimension of the decay subspace of the dichotomy of system $\dot{x}(t) = A(t)x(t)$ for increasing t). As outlined in Chapter 1, a continuous orthonormal transformation $T(t) = [T_1(t) \mid T_2(t)]$ is sought which puts the system (1.15a) into upper block triangular form. This will be so if $T_1(t)$ and $T_2(t)$ satisfy the ODEs (see [1-4]) :

$$\dot{T}_1(t) = [I_n - T_1(t)T_1^T(t)].A(t)T_1(t) \qquad (3.1)$$

$$\dot{T}_2(t) = [-I_n + T_2(t)T_2^T(t)].A^T(t)T_2(t)$$

$$= -T_1(t)T_1^T(t)A^T(t)T_2(t) \qquad (3.2).$$

The transformed system matrix of (1.15a) is then :

$$V(t) = \begin{bmatrix} T_1^T(t)A(t)T_1(t) & T_1^T(t)[A(t) + A^T(t)].T_2(t) \\ 0 & T_2^T(t)A(t)T_2(t) \end{bmatrix}$$

In Chapter 1 we described the double sweep orthonormalisation method which we showed to be stable but very costly as regards storage. Here we show how to use invariant imbedding to enable us to integrate all the ODEs in one direction, which avoids having to store values of $T(t)$ in the forward sweep for subsequent use in the backward sweep. By superposition, the general solution of $\dot{y}(t) = V(t)y(t) + g(t)$ (1.16a) can be written : $y(t) = Y(t)y(a) + h(t)$ (3.3)

where $Y(t)$ is the fundamental solution of $\dot{y}(t) = V(t)y(t)$ for which $Y(a) = I_n$ and $h(t)$ is the particular solution of $\dot{y}(t) = V(t)y(t) + g(t)$ for which $h(a) = 0$. Since

$V_{21}(t) = 0$ for all t and $\dot{Y}(t) = V(t)Y(t)$ we have $\dot{Y}_{21}(t) = V_{22}(t)Y_{21}(t)$ where $Y_{21}(a) = 0$. Hence $Y_{21}(t) = 0$ for all t, and so, from (3.3) we have :

$$y_1(t) = Y_{11}(t)y_1(a) + Y_{12}(t)y_2(a) + h_1(t) \qquad (3.4a)$$

$$y_2(t) = Y_{22}(t)y_2(a) + h_2(t) \qquad (3.4b).$$

Also from $\dot{Y}(t) = V(t)Y(t)$ we get the IVPs :

$$\dot{Y}_{11}(t) = V_{11}(t)Y_{11}(t), \quad Y_{11}(a) = I_p \qquad (3.5)$$

$$\dot{Y}_{12}(t) = V_{11}(t)Y_{12}(t) + V_{12}(t)Y_{22}(t), \quad Y_{12}(a) = 0 \qquad (3.6)$$

$$\dot{Y}_{22}(t) = V_{22}(t)Y_{22}(t), \quad Y_{22}(a) = I_q \qquad (3.7)$$

and from $\dot{h}(t) = V(t)h(t) + g(t)$ we get :

$$\dot{h}_1(t) = V_{11}(t)h_1(t) + V_{12}(t)h_2(t) + g_1(t) \qquad (3.8a)$$

$$\dot{h}_2(t) = V_{22}(t)h_2(t) + g_2(t) \qquad (3.8b).$$

However, of the above IVPs, (3.5), (3.6) and (3.8a) would all be unstable for forward integration because all of the kinematic eigenvalues of $V_{11}(t)$ (p,p) are such that $\mathrm{Re}\int_a^b \sigma_i(t)\,dt > 0$ $(1 <= i <= p)$. To obtain only stable IVPs, Van Loon [17] therefore defines $R_{11}(t)$ (p,p) and $R_{12}(t)$ (p,q) by : $R_{11}(t) = Y_{11}^{-1}(t)$, $R_{12}(t) = -Y_{11}^{-1}(t)Y_{12}(t)$ and also $l_1(t)$ $(p,1)$ by $l_1(t) = -Y_{11}^{-1}(t)h_1(t)$, where the non-singularity of $Y_{11}(t)$ is ensured by (3.5). Using these, (3.4a) becomes :

$$y_1(a) = R_{11}(t)y_1(t) + R_{12}(t)y_2(a) + l_1(t) \qquad (3.9)$$

which is known as the recovery transformation equation. We can now also obtain [3-1] the following IVPs for $R_{11}(t)$, $R_{12}(t)$ and $l_1(t)$:

$$\dot{R}_{11}(t) = -R_{11}(t)V_{11}(t) \qquad R_{11}(a) = I_p \qquad (3.10)$$

$$\dot{R}_{12}(t) = -R_{11}(t)V_{12}(t)Y_{22}(t) \qquad R_{12}(a) = 0 \qquad (3.11)$$

$$\dot{l}_1(t) = -R_{11}(t)[V_{12}(t)h_2(t) + g_1(t)], \quad l_1(a) = 0 \qquad (3.12).$$

Note that IVPs (3.11) and (3.12) are simply quadratures and so will be stable. For the case where V_{11} is constant the forward stability of IVP (3.10) is obvious since all the eigenvalues of V_{11} would be positive and these are a true guide to stability. But when $V_{11}(t)$ is variable, which would generally be the case, the forward stability of equation (3.10) requires justification [3-2].

Evaluation of equations (3.4b) and (3.9) at $t = b$ together with the transformed BC (1.16b) produces the following (well conditioned) system of (2n,2n) linear equations :

$$\begin{bmatrix} 0 & -Y_{22}(b) & 0 & I_2 \\ I_1 & -R_{12}(b) & -R_{11}(b) & 0 \\ 0 & 0 & E_2 & E_3 \\ 0 & E_1 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1(a) \\ y_2(a) \\ y_1(b) \\ y_2(b) \end{bmatrix} = \begin{bmatrix} h_2(b) \\ l_1(b) \\ c_1 \\ c_2 \end{bmatrix} \qquad (3.13)$$

where $E_1 = B_a T_2(a)$, $E_2 = B_b T_1(b)$ and $E_3 = B_b T_2(b)$.

The values of $Y_{22}(b)$, $R_{12}(b)$, $R_{11}(b)$, $l_1(b)$ and $h_2(b)$ are obtained by integrating forwards simultaneously the (well conditioned) IVPs (3.7),(3.11),(3.10),(3.12) and (3.8b) together with (3.1) and (3.2). System (3.13) is then solved for $y(a)$ and $y(b)$ from which the solutions $x(a)$ and $x(b)$ to LBVP (1.15) can be obtained from the transformation equation $x(t) = T(t)y(t)$. By subdividing the problem range [a,b] the above algorithm can be adapted to find the solution $x(t)$ at these internal nodes also but this will require the solution

of a much larger multiple shooting type system of linear equations instead of (3.13).

We now describe two alternative versions of the above ortho-normalisation algorithm which are more economical in that the ODEs involved are of smaller dimensions. Recall the double sweep method outlined in Chapter 1. We integrate forwards the IVP :

$$\dot{y}_2(t) = V_{22}(t)y_2(t) + g_2(t), \qquad y_2(a) = E_1^{-1} c_2 \qquad (3.14)$$

together with IVPs (3.1) and (3.2) so as to obtain $y_2(b)$. Then (see (1.33) of Chapter 1) we find $y_1(b)$ from :

$$y_1(b) = (E_2)^{-1}.\,[c_1 - E_3\, y_2(b)] \qquad (3.15).$$

To obtain $y_1(t)$ we now integrate backwards the IVP :

$$\dot{y}_1(t) = V_{11}(t)y_1(t) + V_{12}(t)y_2(t) + g_1(t) \qquad (3.16)$$

from $t = b$ to $t = a$, but this requires storage of the values of $T_1(t)$, $T_2(t)$ and $y_2(t)$ during the forward integrations of (3.1), (3.2) and (3.14) respectively. To avoid this we can instead, as explained below, integrate forwards a general solution for $y_1(t)$.

From (3.16) : $\dot{y}_1(t) = V_{11}(t)y_1(t) + p_1(t)$ (3.17)

where $p_1(t) = V_{12}(t)y_2(t) + g_1(t)$. From (3.4a) the general solution of (3.17) can be written :

$$y_1(t) = Y_{11}(t)y_1(a) + z_1(t) \qquad (3.18)$$

where $z_1(t) = Y_{12}(t)y_2(a) + h_1(t)$ is a particular solution of (3.17). Now let : $R_{11}(t)z_1(t) = w_1(t)$ size (p,1) ====>

$\dot{w}_1(t) = R_{11}(t).[V_{11}(t)z_1(t) + p_1(t)] - R_{11}(t)V_{11}(t)z_1(t)$

(from (3.10)) ====> $\dot{w}_1(t) = R_{11}(t)p_1(t), \quad w_1(a) = 0$ (3.19)

since $z_1(a) = 0$. Also from (3.18) :

$$y_1(t) = R_{11}^{-1}(t)y_1(a) + R_{11}^{-1}(t)w_1(t) \qquad (3.20)$$

$$\Longrightarrow \quad y_1(b) = R_{11}^{-1}(b).[y_1(a) + w_1(b)]$$

$$\Longrightarrow \quad y_1(a) = R_{11}(b)y_1(b) - w_1(b) \qquad (3.21).$$

Thus an alternative algorithm is to integrate simultaneously forwards (from $t = a$ to $t = b$) the IVPs (3.1),(3.2),(3.14), (3.10) and (3.19) for $T_1(t)$, $T_2(t)$, $y_2(t)$, $R_{11}(t)$ and $w_1(t)$ respectively. Then obtain $y_1(b)$ from (3.15) and use (3.21) to get $y_1(a)$. Finally, obtain the solutions $x(a)$ and $x(b)$ to the LBVP (1.15) by using the transformation $x(t) = T(t)y(t)$. This version of the algorithm has the advantage that fewer variables of integration are required. Both versions have the $T_1(t)$ (3.1), $T_2(t)$ (3.2) and $R_{11}(t)$ (3.10) IVPs in common. But this version has $y_2(t)$ (q,1) instead of $Y_{22}(t)$ (q,q), $w_1(t)$ (p,1) instead of $R_{12}(t)$ (p,q) and the $l_1(t)$ and $h_2(t)$ ODEs have been dispensed with. Also in the previous version we had to solve a linear system (3.13) of size (2n,2n) whereas here equation (3.15) for $y_1(b)$ is only (p,p). These could be significant savings if n were large. Note that having found $y(a)$ and $y(b)$ we could then attempt to find $y(t)$ at interior points of [a,b] by re-integrating forwards the above equations for $T_1(t)$, $T_2(t)$, $y_2(t)$, $R_{11}(t)$ and $w_1(t)$ and then using equation (3.20) to obtain $y_1(t)$. However, since (3.20) involves $R_{11}^{-1}(t)$ this may not be successful because, although the $R_{11}(t)$ equation (3.10) should be stable for forward integration, the $R_{11}(t)$ solution so obtained is still

liable to be ill conditioned and therefore unsuitable for inversion. Note also that equation (3.21) is identical to the second equation of (3.13). Although this equation involves $R_{11}(t)$ the possibility of the latter being ill conditioned does not now matter as this term appears on the right hand side of the equation.

We can economise still further in the above algorithm by eliminating the $T_2(t)$ equation, as follows. Let $u(t) = T_2(t)y_2(t)$ where $u(t)$ is $(n,1)$, then :

$$\dot{u}(t) = T_2(t)\dot{y}_2(t) + \dot{T}_2(t)y_2(t) =$$

$$T_2(t)[T_2^T(t)A(t)T_2(t)y_2(t) + T_2^T(t)f(t)] - T_1(t)T_1^T(t)A^T(t)T_2(t)y_2(t)$$

$$= [T_2(t)T_2^T(t)A(t) - T_1(t)T_1^T(t)A^T(t)]u(t) + T_2(t)T_2^T(t)f(t) =$$

$$[(I_n - T_1(t)T_1^T(t))A(t) - T_1(t)T_1^T(t)A^T(t)]u(t) + [I_n - T_1(t)T_1^T(t)]f(t)$$

i.e. $$\dot{u}(t) = A(t)u(t) + T_1(t)G(t) + [I_n - T_1(t)T_1^T(t)]f(t) \qquad (3.22)$$

where $$G(t) = -T_1^T(t)[A(t) + A^T(t)]u(t) \qquad (3.23).$$

The initial conditions for IVP (3.22) are $u(a) = T_2(a)y_2(a)$ $= T_2(a)[B_a T_2(a)]^{-1} c_2$. Also from (3.19) :

$$\dot{w}_1(t) = R_{11}(t)p_1(t) = R_{11}(t)[V_{12}(t)y_2(t) + g_1(t)]$$

$$= R_{11}(t)V_{12}(t)y_2(t) + R_{11}(t)g_1(t)$$

$$= R_{11}(t)T_1^T(t)[A(t) + A^T(t)]T_2(t)y_2(t) + R_{11}(t)g_1(t)$$

$$= -R_{11}(t)G(t) + R_{11}(t)g_1(t)$$

i.e. $$\dot{w}_1(t) = R_{11}(t)[T_1^T(t)f(t) - G(t)] \qquad (3.24)$$

where $w_1(a) = 0$.

Thus the revised algorithm is to integrate simultaneously forwards the equations (3.24),(3.22),(3.10) and (3.1) for $w_1(t)$, $u(t)$, $R_{11}(t)$, and $T_1(t)$ respectively. Now

$$x(t) = T(t)y(t) \quad ====> \quad x(t) = [T_1(t) \mid T_2(t)] \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}$$

$$====> \quad x(t) = T_1(t)y_1(t) + u(t) \qquad (3.25)$$

Hence from the final BC of LBVP (1.15b) we have :

$$B_b\, x(b) = c_1 \quad ===> \quad B_b\,[T_1(b)y_1(b) + u(b)] = c_1 \quad ====>$$

$$y_1(b) = [B_b\, T_1(b)]^{-1}.\,[c_1 - B_b\, u(b)] \qquad (3.26)$$

from which we can obtain $y_1(b)$. Then we use (3.21) to find $y_1(a)$. Finally we obtain the LBVP solutions $x(a)$ and $x(b)$ from (3.25).

Method of Davey, Meyer et al. [3, 4, 24, 25] :

Unlike Van Loon's method this method does not employ invariant imbedding i.e. it is a double sweep method which therefore has the disadvantage of requiring storage of values during the forward sweep for subsequent use in the backward sweep. At first sight it is not easy to relate the Davey/Meyer method (as described in [3] and [4]) to the orthogonal decoupling transformation method of the previous section. However, as we shall see, the Davey/Meyer method is simply the basic double sweep orthonormalisation method (as outlined in Chapter 1) but with the elimination of the $T_2(t)$ IVP (1.32) by the introduction of the new variable $u(t) = T_2(t)y_2(t)$, as in the third version of Van Loon's method described earlier. The backward sweep IVP for $y_1(t)$ is (see (1.35) of Chapter 1)

$$\dot{y}_1(t) = \{T_1^T(t)A(t)T_1(t)\}y_1(t) + T_1^T(t)\{A(t) + A^T(t)\}T_2(t)y_2(t) + T_1^T(t)f(t)$$

i.e. $$\dot{y}_1(t) = V_{11}(t)y_1(t) - G(t) + T_1^T(t)f(t) \qquad (3.27)$$

where $u(t) = T_2(t)y_2(t)$, $V_{11}(t) = T_1^T(t)A(t)T_1(t)$ and $G(t) = -T_1^T(t)\{A(t) + A^T(t)\}u(t)$ as in (3.23). Note that equation (3.27) corresponds to Meyer [4: 2.17] and to Davey [3: 24]. Now equation (3.1) for $T_1(t)$ can be written :

$$\dot{T}_1(t) = A(t)T_1(t) - T_1(t)V_{11}(t) \qquad (3.28)$$

and the equation for $u(t)$ is as in (3.22) viz.

$$\dot{u}(t) = A(t)u(t) + T_1(t)G(t) + \{I_n - T_1(t)T_1^T(t)\}f(t) \qquad (3.29).$$

Equations (3.28) and (3.29) correspond to those of Davey [3: 2 & 18 respectively].

The algorithm is to integrate simultaneously forwards (from $t = a$ to $t = b$) IVPs (3.28) and (3.29) for $T_1(t)$ and $u(t)$ respectively from their initial values given by $B_a T_1(a) = 0$ and $u(a) = T_2(a)[B_a T_2(a)]^{-1}.c_2$ (as in Van Loon's method). During these integrations the values of $T_1(t)$ and $u(t)$ must be stored at the end of each step (if a fixed step integrator is used) or at arbitrary nodes $a = t_0 < t_1 < \ldots\ldots < t_N = b$ in the case of a variable step integrator. The initial condition for the backward integration of $y_1(t)$ is now obtained from (1.33): $y_1(b) = [B_b T_1(b)]^{-1}.[c_1 - B_2 u(b)]$ (as in Davey [3: 23]) from which value ODE (3.27) is now integrated from $t = b$ to $t = a$ using the stored values of $T_1(t)$ and $u(t)$ at the nodes and interpolations between the nodes if necessary. Finally the

solution $x(t)$ of the LBVP (1.15) can be obtained at any output point $t \in [a,b]$ from (3.25) : $x(t) = T_1(t)y_1(t) + u(t)$. Thus the Davey/Meyer method is in effect a straightforward application of the double sweep orthonormalisation transformation method except that instead of integrating forwards the IVP for $y_2(t)$ (as in (1.34) of Chapter1) we integrate forwards IVP (3.29) for $u(t) = T_2(t)y_2(t)$, thereby eliminating the need for the $T_2(t)$ IVP (1.32), as in the last variation of Van Loon's method described earlier. Note that the stability of the $u(t)$ IVP (3.29) is ensured by that of the $y_2(t)$ IVP (1.34) because $u^T(t)u(t) = y_2^T(t)T_2^T(t)T_2(t)y_2(t) = y_2^T(t)y_2(t)$, since $T_2(t)$ is a unit orthogonal column set.

As mentioned earlier, the main reason for interest in developing continuous orthonormalisation methods was an attempt to overcome the practical difficulties associated with superposition methods viz. loss of independence of the columns of the fundamental solutions. To recapitulate, if we are trying to solve LBVP (1.15) by single shooting reduced superposition then we express the solution $x(t)$ of the LBVP in the form :

$x(t) = X_1(t)d + p(t)$, where $X_1(t)$ (n,p) is a part fundamental solution of $\dot{x}(t) = A(t)x(t)$ for which $B_a X_1(a) = 0$ and $p(t)$ is a particular solution of $\dot{x}(t) = A(t)x(t) + f(t)$ for which $B_a p(a) = c_2$ and d is constant $(p,1)$. Troubles may arise (particularly in the case where system $\dot{x}(t) = A(t)x(t)$ is 'stiff' i.e. one for which the kinematic eigenvalues are widely separated in real part) due to loss of independence of

the columns of $X_1(t)$ as t increases and also to increasing dependence of $p(t)$ on span $X_1(t)$. All of the variants of continuous orthonormalisation overcome both these difficulties (at least in theory :see later) by finding an orthonormal basis for span $X_1(t)$, at each value of t, as we now explain.

(Spanning Theorem) $X_1(t)$ satisfies the ODE $\dot{X}_1(t) = A(t)X_1(t)$. Let $T_1(t)$ (n,p) be defined by $T_1(t) = X_1(t)W(t)$ where $W(t)$ (p,p) satisfies $\dot{W}(t) = -W(t)V_{11}(t)$, $W(a) = I_p$,where $V_{11}(t)$ is (p,p). Thus $W(t)$ will be nonsingular for all t and $T_1(a) = X_1(a)$. Now $T_1(t) = X_1(t)W(t)$ ====>

$\dot{T}_1(t) = X_1(t)\dot{W}(t) + \dot{X}_1(t)W(t) = -X_1(t)W(t)V_{11}(t) + A(t)X_1(t)W(t)$

i.e. $\dot{T}_1(t) = A(t)T_1(t) - T_1(t)V_{11}(t)$, which is precisely equation (3.28). Thus if $\dot{X}_1(t) = A(t)X_1(t)$ and $\dot{T}_1(t) = A(t)T_1(t) - T_1(t)V_{11}(t)$ where $T_1(a) = X_1(a)$ then $T_1(t) = X_1(t)W(t)$, where $W(t)$ is nonsingular for all t, which means that span $T_1(t)$ = span $X_1(t)$ for all t. Hence, in the continuous orthonormalisation methods, instead of integrating forwards the system $\dot{X}_1(t) = A(t)X_1(t)$ from $B_a X_1(a) = 0$ and then obtaining the solution from the resolution equation $x(t) = X_1(t)d + p(t)$, we integrate forwards the system $\dot{T}_1(t) = A(t)T_1(t) - T_1(t)V_{11}(t)$ (3.28) from $B_a T_1(a) = 0$ (1.17) and use the resolution equation $x(t) = T_1(t)y_1(t) + u(t)$ (3.25). In effect, we have "replaced" the part fundamental solution $X_1(t)$ by $T_1(t)$ where span $T_1(t)$ = span $X_1(t)$ for all t, and where now the columns of $T_1(t)$ should remain unit orthogonal for all t. Note also that since $u(t) = T_2(t)y_2(t)$

then $u^T(t)T_1(t) = y_2^T(t)T_2^T(t)T_1(t) = 0$ so that $u(t)$ should be orthogonal to span $T_1(t)$ for all t.

As described in Chapter 2, in Conte's re- orthonormalisation method the part fundamental solution $X_1(t)$ is re-orthogonalised whenever necessary at discrete points $t_1 < t_2 < \ldots\ldots < t_{N-1}$ in $[a,b]$ by means of a QU decomposition using the Gram-Schmidt process. In the continuous orthonormalisation methods of this Chapter this re-orthogonalisation takes place at every value of t automatically.

We turn now to a practical difficulty-which may be encountered when using any of the continuous orthonormalisation methods. Although $T_1(t)$ and $T_2(t)$, as computed from IVPs (3.1) and (3.2), should in theory produce a unit orthogonal matrix $T(t) = [T_1(t) \mid T_2(t)]$ for all t,in practice this may not be so. Before $t = b$ is reached the orthogonality of $T(t)$ may be lost and $T(t)$ may even become (seriously) ill conditioned with a consequent effect upon the accuracy of the computed solution $x(t)$ of the LBVP. This may happen because, as described by Davey [3], Meyer [4] and Van Loon [17], the IVPs (3.1) and (3.2) for $T_1(t)$ and $T_2(t)$ may be mathematically unstable in that not every orthonormal solution of these IVPs is asymptotically stable. In theory, the value of $T_1^T(t)T_1(t)$ should remain constant for all t at its initial value of I_p. Now $\frac{d}{dt}\{T_1^T(t)T_1(t)\} = T_1^T(t)\dot{T}_1(t) + \dot{T}_1^T(t)T_1(t)$ and so if $T_1^T(t)\dot{T}_1(t) = 0$ for all t this would ensure that $T_1^T(t)T_1(t)$

remained constant. The ODE for $T_1(t)$ is

$$\dot{T}_1(t) = A(t)T_1(t) - T_1(t)c_{11}(t) \quad (3.29) \quad \Longrightarrow$$

$$T_1^T(t)\dot{T}_1(t) = T_1^T(t)A(t)T_1(t) - \{T_1^T(t)T_1(t)\}c_{11}(t) \quad (3.30)$$

which means that if we choose $c_{11}(t) = T_1^T(t)A(t)T_1(t)$ (as we stated earlier) then $T_1^T(t)\dot{T}_1(t) = 0$ only if $T_1^T(t)T_1(t) = I_p$ which in practice may not be so. In order to numerically stabilise this equation Davey and Meyer therefore suggest that instead we take $c_{11}(t) = \{T_1^T(t)T_1(t)\}^{-1}.\{T_1^T(t)A(t)T_1(t)\}$ so that (3.30) becomes : $T_1^T(t)\dot{T}_1(t) =$

$\{T_1^T(t)A(t)T_1(t)\} - \{T_1^T(t)T_1(t)\}.\{T_1^T(t)T_1(t)\}^{-1}.\{T_1^T(t)A(t)T_1(t)\}$.

In this case, $T_1^T(t)\dot{T}_1(t) = 0$ even if $T_1^T(t)T_1(t) \neq I_p$ exactly i.e. this choice of $c_{11}(t)$ has the effect of stabilising the value of $T_1^T(t)T_1(t)$ in the event that it starts to move away from its theoretical value of I_p. A similar argument applies to the $T_2(t)$ IVP (3.2).

The effect of replacing $c_{11}(t) = T_1^T(t)A(t)T_1(t)$ by $c_{11}(t) = \{T_1^T(t)T_1(t)\}^{-1}.\{T_1^T(t)A(t)T_1(t)\}$ in (3.29) is equivalent to replacing $T_1^T(t)$ by $T_1^+(t) = \{T_1^T(t)T_1(t)\}^{-1}.T_1^T(t)$ in the original version, where $T_1^+(t)$ is called the 'generalised inverse' of $T_1(t)$. The generalised inverse $T_2^+(t)$ of $T_2(t)$ is similarly defined. Thus the 'generalised inverse' versions of the ODEs for $T_1(t)$ and $T_2(t)$ become :

$$\dot{T}_1(t) = A(t)T_1(t) - T_1(t)T_1^+(t)A(t)T_1(t) \quad (3.31a)$$

$$\dot{T}_2(t) = -A^T(t)T_2(t) + T_2(t)T_2^+(t)A^T(t)T_2(t) \quad (3.31b).$$

In Chapter 5 we describe the factorisation method of Babuska and Majer [16], one version of which utilises the orthogonal transformations of this Chapter.

In Appendix I :

[3-1]: Derivation of ODEs for $R_{11}(t)$, $R_{12}(t)$ and $1_1(t)$ of Van Loon's method.

[3-2]: Stability of $R_{11}(t)$ IVP.

CHAPTER 4

THE RICCATI TRANSFORMATION METHOD

Application of invariant imbedding

In Chapter 1 we outlined the basic double sweep Riccati transformation method which, though stable, has the disadvantage of requiring considerable storage during the forward sweep. As with the continuous orthonormal transformation (Chapter 3) this can be avoided by employing invariant imbedding whereby all of the integrations are performed in one direction. The operation of this technique with the Riccati transformation is virtually identical to that described in Chapter 3 for the orthonormal invariant imbedding (see equations (3.1) to (3.22)). But now in this case the decoupling transformation matrix $T(t)$ is $\begin{bmatrix} I_p & 0 \\ R(t) & I_q \end{bmatrix}$. Corresponding to this the upper block triangular transformed system matrix is $V(t) = \begin{bmatrix} V_{11}(t) & V_{12}(t) \\ 0 & V_{22}(t) \end{bmatrix}$

where : $V_{11}(t) = A_{11}(t) + A_{12}(t)R(t)$

$V_{12}(t) = A_{12}(t)$

$V_{22}(t) = A_{22}(t) - R(t)V_{12}(t)$

and where $R(t)$ is the solution of the Riccati equation:

$$\dot{R}(t) = A_{21}(t) + A_{22}(t)R(t) - R(t)A_{11}(t) - R(t)A_{12}(t)R(t) \quad (4.1a)$$

$$R(a) = -L_1^{-1} L_0 \quad (4.1b).$$

As described in Chapter 3, we obtain the simultaneous solution

of the IVPs:

$$\dot{R}_{11}(t) = -R_{11}(t)V_{11}(t), \qquad R_{11}(a) = I_p$$

$$\dot{R}_{12}(t) = -R_{11}(t)V_{12}(t)Y_{22}(t), \qquad R_{12}(a) = 0$$

$$\dot{l}_1(t) = -R_{11}(t)\{V_{12}(t)h_2(t) + g_1(t)\}, \qquad l_1(a) = 0$$

$$\dot{Y}_{22}(t) = V_{22}(t)Y_{22}(t), \qquad Y_{22}(a) = I_q$$

$$\dot{h}_2(t) = V_{22}(t)h_2(t) + g_2(t), \qquad h_2(a) = 0$$

together with (4.1).

(Note that $R_{11}(t)$ and $R_{12}(t)$ are not related to the Riccati solution R(t). The notation used follows that of Van Loon [17]).

Hence, as before, we obtain $y(a)$ and $y(b)$ by solving the (well conditioned) system of (2n,2n) linear equations as given in (3.13) viz.

$$\begin{bmatrix} 0 & -Y_{22}(b) & 0 & I_q \\ I & -R_{12}(b) & -R_{11}(b) & 0 \\ 0 & 0 & E_2 & E_3 \\ 0 & E_1 & 0 & 0 \end{bmatrix} \begin{bmatrix} y_1(a) \\ y_2(a) \\ y_1(b) \\ y_2(b) \end{bmatrix} = \begin{bmatrix} h_2(b) \\ l_1(b) \\ c_1 \\ c_2 \end{bmatrix} \qquad (4.2)$$

but where now $E_1 = L_1$ (see (1.26) of Chapter 1), $E_2 = L_2 + L_3R(b)$ and $E_3 = L_3$ (see (1.28)). Finally, the solutions $x(a)$ and $x(b)$ to LBVP (1.15) are obtained from the transformation equation $x(t) = T(t)y(t)$. As with the orthogonal transformation the solution $x(t)$ at interior points of $[a,b]$ could be found by subdividing $[a,b]$ leading to the solution of a much larger system of linear equations than (4.2). The more economical algorithm described in Chapter 3, equations (3.14) to (3.21), is also applicable to the Riccati transformation, but the version in

equations (3.22) to (3.26), involving the substitution $u(t) = T_2(t)y_2(t)$, is not.

Re-imbedding

However, all of the foregoing pre-supposes that the solution of the Riccati equation (4.1) remains bounded for all $t \in [a,b]$. This may not be so, particularly if the system matrix $A(t)$ of the given ODE is rapidly varying so causing rotational activity of the columns of the fundamental solutions. This is clearly illustrated by the following example :

Ex 1: Consider a LBVP over $[0,1]$ whose ODE is $\dot{x}(t) = A(t)x(t)$ where $A(t) = \begin{bmatrix} \cos(2wt) & w - \sin(2wt) \\ -w - \sin(2wt) & -\cos(2wt) \end{bmatrix}$ so that the parameter w determines the rate of rotation. In this case, a fundamental solution is given by : $X(t) = \begin{bmatrix} \cos(wt) & \sin(wt) \\ -\sin(wt) & \cos(wt) \end{bmatrix} \begin{bmatrix} e^{t} & 0 \\ 0 & e^{-t} \end{bmatrix}$ and the solution of the Riccati equation (4.1a) corresponding to initial conditions $R(0) = 0$ is $R(t) = -\tan(wt)$ which has a pole at $t = \pi/2w$. Thus if $w > \pi/2$ then $R(t)$ will 'blow up' before $t = 1$ is reached.

As described in Chapter 1 (equations (1.24) to (1.30)), suppose we transform the ODE $\dot{x}(t) = A(t)x(t) + f(t)$ (4.3a) of the given LBVP into system $\dot{y}(t) = V(t)y(t) + g(t)$ (4.3b) by means of the transformation $x(t) = T(t)y(t)$ where

$T(t) = \begin{bmatrix} I_p & 0 \\ R(t) & I_q \end{bmatrix}$. Corresponding fundamental solutions $X(t)$ and $Y(t)$ of (4.3a) and (4.3b) respectively are connected by $X(t) = T(t)\,Y(t)$. Now suppose that $X(t)$ is the fundamental solution of (4.3a) where

$$X(a) = T(a) = \begin{bmatrix} I_p & 0 \\ R(a) & I_q \end{bmatrix}$$

so that $Y(a) = I_n$. Now $\dot{Y}(t) = V(t)Y(t)$ where $V(t) = \begin{bmatrix} V_{11}(t) & V_{12}(t) \\ 0 & V_{22}(t) \end{bmatrix}$.

Hence $\dot{Y}_{21}(t) = V_{22}(t)Y_{21}(t)$ where $Y_{21}(a) = 0$. Thus $Y_{21}(t) = 0$ for all t. From $X(t) = T(t)Y(t)$ we therefore get

$$\begin{bmatrix} X_{11}(t) & X_{12}(t) \\ X_{21}(t) & X_{22}(t) \end{bmatrix} = \begin{bmatrix} I_p & 0 \\ R(t) & I_q \end{bmatrix} \begin{bmatrix} Y_{11}(t) & Y_{12}(t) \\ 0 & Y_{22}(t) \end{bmatrix}$$

$\Longrightarrow X_{11}(t) = Y_{11}(t)$ and $X_{21}(t) = R(t)Y_{11}(t)$

$\Longrightarrow X_{21}(t).X_{11}^{-1}(t) = R(t)$.

Therefore the solution of the Riccati equation (4.1a) is given for all t by $R(t) = X_{21}(t)X_{11}^{-1}(t)$ from which we see that $R(t)$ will have a pole whenever $X_{11}(t)$ becomes singular. Now in theory the p columns of the part fundamental solution $X'(t) = \begin{bmatrix} X_{11}(t) \\ X_{21}(t) \end{bmatrix}$ are independent for all t and so column rank = row rank = p i.e. for any value of t there exists p linearly independent rows of $X'(t)$. This provides us with the strategy for preventing the Riccati solution $R(t)$ from becoming unbounded in $[a,b]$.

In practice, at any value of t there will exist p "most linearly independent" rows of $X'(t)$. Ideally, we would like to

have these p rows in $X_{11}(t)$ for all t because this would ensure that R(t) always remained finite. However, this would be very costly to achieve and so we settle for the following compromise. Consider first the forward sweep. As the forward integration of the Riccati IVP (4.1) proceeds from t = a (simultaneously with the $y_2(t)$ IVP (1.29) of Chapter 1) we check at each step on the value of $\left\| \begin{matrix} I_p \\ R(t) \end{matrix} \right\|$. As soon as a value $t = t^*$ is reached where

$$\left\| \begin{matrix} I_p \\ R(t^*) \end{matrix} \right\| > \rho . \left\| \begin{matrix} I \\ R(a) \end{matrix} \right\| \qquad (4.4)$$

(where ρ is a pre-selected small positive constant) we perform a re-imbedding (i.e. a rearrangement of the solution components of the problem) as follows.

Let the given LBVP be :

$$\dot{x}(t) = A(t)x(t) + f(t) \qquad (4.5a)$$

$$B_0\, x(a) + B_1\, x(b) = c \qquad (4.5b)$$

for $a <= t <= b$ where x(t) denotes the solution of the LBVP in the given imbedding. If we use perm matrix $\Pi(n,n)$ (chosen as explained later) to change the imbedding then (4.5) becomes :

$$\Pi \dot{x}(t) = \Pi A(t)(\Pi^T \Pi)x(t) + \Pi f(t)$$

$$B_0 (\Pi^T \Pi)x(a) + B_1 (\Pi^T \Pi)x(b) = c, \quad \text{since} \quad \Pi^T \Pi = I_n$$

$$\text{i.e.} \quad \frac{d}{dt}\{\Pi x(t)\} = \{\Pi A(t) \Pi^T\}\{\Pi x(t)\} + \Pi f(t)$$

$$(B_0 \Pi^T)\{\Pi x(a)\} + (B_1 \Pi^T)\{\Pi x(b)\} = c$$

or : $\frac{d}{dt}\tilde{x}(t) = \tilde{A}(t)\tilde{x}(t) + \tilde{f}(t)$

$$\tilde{B}_0\,\tilde{x}(a) + \tilde{B}_1\,\tilde{x}(b) = c \qquad (4.6)$$

for $t^* <= t <= b$, where $\tilde{x}(t) = \pi\, x(t)$ (4.7)

is the re-arranged solution in the new imbedding and

$\tilde{A}(t) = \pi\, A(t)\,\pi^T$, $\tilde{f}(t) = \pi\, f(t)$, $\tilde{B}_0 = B_0\,\pi^T$, $\tilde{B}_1 = B_1\,\pi^T$

are the corresponding re-imbedded values of $A(t)$, $f(t)$, B_0 and B_1 respectively for all $t >= t^*$. Also, in the new imbedding, the part fundamental solution $X'(t)$ becomes

$$\pi\, X' = \tilde{X}' = \begin{bmatrix} \tilde{X}_{11}(t) \\ \tilde{X}_{21}(t) \end{bmatrix}$$ for $t >= t^*$, i.e. the re-imbedded Riccati solution $\tilde{R}(t)$ is given by $\tilde{R}(t) = \tilde{X}_{21}(t).\tilde{X}_{11}^{-1}(t)$ where now $\tilde{X}_{11}(t)$ will be non-singular until the next pole is reached in this imbedding.

Thus at a restart point $t = t^*$ the re-imbedded Riccati equation viz.

$$\frac{d}{dt}\tilde{R}(t) = \tilde{A}_{21}(t) + \tilde{A}_{22}(t)\tilde{R}(t) - \tilde{R}(t)\tilde{A}_{11}(t) - \tilde{R}(t)\tilde{A}_{12}(t)\tilde{R}(t) \qquad (4.8)$$

and the re-imbedded forward sweep IVP viz.

$$\frac{d}{dt}\tilde{y}_2(t) = \{\tilde{A}_{22}(t) - \tilde{R}(t)\tilde{A}_{12}(t)\}\tilde{y}_2(t) + \{\tilde{f}_2(t) - \tilde{R}(t)\tilde{f}_1(t)\} \qquad (4.9)$$

(compare with (1.29) of Chapter 1) are integrated forwards from their respective initial values $\tilde{R}(t^*)$ and $\tilde{y}_2(t^*)$ which we show in appendix [4-1] are :

$$\tilde{R}(t^*) = [P_{21} + P_{22}R(t^*)].[P_{11} + P_{12}R(t^*)]^{-1} \qquad (4.8a)$$

and $$\tilde{y}_2(t^*) = [P_{22} - \tilde{R}(t^*)P_{12}]y_2(t^*) \qquad (4.9a)$$

where $\Pi = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$. Suppose we are able to remain in this imbedding (using criterion (4.4)) until $t = b$ is reached. Now if $B_1 = \begin{bmatrix} L_2 & L_3 \\ 0 & 0 \end{bmatrix}$ let $\tilde{B}_1 = B_1 \Pi^T$ (4.10)

$= \begin{bmatrix} \tilde{L}_2 & \tilde{L}_3 \\ 0 & 0 \end{bmatrix}$ then the backward integration of the $\dot{\tilde{y}}_1(t)$ equation viz.

$\dot{\tilde{y}}_1(t) = \{\tilde{A}_{11}(t) + \tilde{A}_{12}(t)\tilde{R}(t)\}\tilde{y}_1(t) + \tilde{A}_{12}(t)\tilde{y}_2(t) + \tilde{f}_1(t)$

(compare (1.30) of Chapter 1) is started from :

$\tilde{y}_1(b) = [\tilde{L}_2 + \tilde{L}_3 \tilde{R}(b)]^{-1} \cdot [c_1 - \tilde{L}_3 \tilde{y}_2(b)]$ (compare (1.28)).

This proceeds from $t = b$ to $t = t^*$ where we switch back to the original imbedding for which the ODE is :

$\dot{y}_1(t) = \{A_{11}(t) + A_{12}(t)R(t)\}y_1(t) + A_{12}(t)y_2(t) + f_1(t)$,

where the restart value is shown in [4-2] to be :

$y_1(t^*) = [P_{11} + P_{12} R(t^*)]^{-1} \cdot [\tilde{y}_1(t^*) - P_{12} y_2(t^*)]$.

For the original imbedding, from $t = a$ to $t = t^*$, the solution $x(t)$ of the LBVP (4.5) is obtained directly from the transformation equation (1.25) of Chapter 1 viz.

$x_1(t) = y_1(t)$, $x_2(t) = R(t)y_1(t) + y_2(t)$. Similarly, for the new imbedding from $t = t^*$ to $t = b$ we have :

$\tilde{x}_1(t) = \tilde{y}_1(t)$, $\tilde{x}_2(t) = \tilde{R}(t)\tilde{y}_1(t) + \tilde{y}_2(t)$, from which we can recover solution $x(t)$ from $x(t) = \Pi^T \tilde{x}(t)$ (4.11)

by using (4.7).

In practice (depending on the nature of solution $R(t)$ and the

choice of constant ρ in criterion (4.4)) several re-imbeddings may be required between $t = a$ and $t = b$, in which case at each restart point we must store the perm matrix used to change the imbedding as well as the composite perm matrix π_c which relates the current imbedding to the original imbedding of the problem. Note that in the above description π is the perm matrix which changes the imbedding at any restart point $t = t^*$. If several re-imbeddings were performed then, in equations (4.10) and (4.11), we would require the composite permutation matrix π_c.

We turn now to a procedure [5] for choosing the perm matrix to change the imbedding at any restart point $t = t^*$ so as to keep the Riccati transformation $T(t)$ well conditioned throughout $[a,b]$:

Alternate row interchanges and column operations are performed on the matrix $\begin{bmatrix} I_\rho \\ R(t^*) \end{bmatrix}$, these being of the form :

$$[P_\rho \; P_{\rho-1} \ldots\ldots P_2 \; P_1] \begin{bmatrix} I_\rho \\ R(t^*) \end{bmatrix} [G_1 G_2 \ldots G_{\rho-1}] = \pi \begin{bmatrix} I_\rho \\ R(t^*) \end{bmatrix} G$$

where π is the composite perm $[P_\rho \; P_{\rho-1} \ldots P_1]$ and G is the composite (column) Gaussian elimination matrix where the sequence of operations on

$$\begin{bmatrix} I_\rho \\ R(t^*) \end{bmatrix} \quad \text{is} \quad P_1, G_1, P_2, G_2 \ldots\ldots, P_\rho$$

where these are performed alternately on left and right side.

P_i is the perm matrix (n,n) which takes the max mod element

in the ith column ($1 <= i <= p$) to the ith row and $G_i(p,p)$ is the matrix which performs (column) Gaussian elimination with (i,i)th element as pivot. Suppose that the final transformed value of $\begin{bmatrix} I_p \\ R(t^*) \end{bmatrix}$ is $\begin{bmatrix} E \\ F \end{bmatrix}$ (where E is lower triangular) then, in appendix [4-3], we show that the Riccati restart value $\tilde{R}(t^*)$, as given in (4.8a), is equal to FE^{-1}. Most important, however, is the fact that this procedure ensures that all the elements of $\tilde{R}(t^*)$ will now be in mod value less than or equal to unity (see [16]). Also if the new imbedding chosen by perm matrix Π is an unstable one in that the Riccati solution $\tilde{R}(t)$ is exponentially increasing with t or has a singular point in [a,b], then criterion (4.4) will ensure that the length of the subinterval in this imbedding will be short. (In Chapter 5, we will see that Babuska and Majer's bounded factorisation Riccati method [16], employs a restart re-imbedding strategy analagous to that just described).

Inverse Riccati equation

We may note a special case which occurs when (for a well conditioned LBVP) the dimensions p and q of the growth and decay subspaces are equal i.e. when the associated Riccati solution matrix R(t) is square (p,p) and so possesses an inverse $S(t) = R^{-1}(t)$ for

any value of t for which $R(t)$ is nonsingular. If, in this case, we change the imbedding by means of the perm matrix

$$J = \begin{bmatrix} 0 & I_p \\ I_p & 0 \end{bmatrix}$$ then the re-imbedded LBVP solution is

$$\tilde{x}(t) = J.x(t) = J\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ x_1(t) \end{bmatrix}$$ and the re-imbedded part fundamental solution is $\tilde{X}'(t) = J.X'(t) = J\begin{bmatrix} X_{11}(t) \\ X_{21}(t) \end{bmatrix} = \begin{bmatrix} X_{21}(t) \\ X_{11}(t) \end{bmatrix}$.

Hence the corresponding re-imbedded Riccati solution matrix is $\tilde{R}(t) = X_{11}(t).X_{21}^{-1}(t) = R^{-1}(t)$, since $R(t) = X_{21}(t).X_{11}^{-1}(t)$.

Also the corresponding re-imbedded ODE system matrix will be

$$\tilde{A}(t) = J.A(t).J^T = \begin{bmatrix} A_{22}(t) & A_{21}(t) \\ A_{12}(t) & A_{11}(t) \end{bmatrix}$$ for which the inverse Riccati equation is :

$$\dot{S}(t) = A_{12}(t) + A_{11}(t)S(t) - S(t)A_{22}(t) - S(t)A_{21}(t)S(t), \qquad (4.11a)$$

instead of (4.1a). Equation (4.11a) can be verified by replacing R by S^{-1} in (4.1a). This 'square' Riccati case is in fact not as special as it may seem because as explained in Chapter 1 any given (n,n) LBVP can be re-written in separated BC form for which the Riccati solution matrix $R(t)$ will be (n,n).

Compound matrix method ([8],[9],[15]):

The difficulties caused by the singularities of the Riccati solution, necessitating the cost of switching from one imbedding to another in order to avoid them, prompts us to ask

whether instead we could remove these singularities altogether. Such a method does exist and is known as the Compound Matrix method, the key point about which is that we calculate the normal to the subspace of all solutions which satisfy the known initial conditions.

To simplify the notation and explanation we describe the application of this method to the solution of a LBVP with a 4th order differential system which has a dichotomy with forward growth subspace of dimension $p = 2$ viz.

$$\phi^{IV} - a_1 \phi''' - a_2 \phi'' - a_3 \phi' - a_4 \phi = a_5 \qquad (4.12)$$

where ϕ and a_i $(1 <= i <= 5)$ are all functions of t.

This equation can be re-written (see [1-1]) in the form :

$\dot{x}(t) = A(t)x(t) + f(t)$ where

$x(t) = [\phi(t), \phi'(t), \phi''(t), \phi'''(t)]^T$

$f(t) = [0, 0, 0, a_5(t)]^T$ and

$$A(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ a_4(t) & a_3(t) & a_2(t) & a_1(t) \end{bmatrix} .$$

Thus the given (n,n) LBVP is :

$$\dot{x}(t) = A(t)x(t) + f(t) \qquad (4.13a)$$

$$B_0 x(a) + B_1 x(b) = c \qquad (4.13b)$$

for $a <= t <= b$ where $B_0 = \begin{bmatrix} 0 \\ B_a \end{bmatrix} \begin{matrix} p \\ q \end{matrix}$, $B_1 = \begin{bmatrix} B_b \\ 0 \end{bmatrix} \begin{matrix} p \\ q \end{matrix}$

$c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$ and $n = 4$, $p = q = 2$. Without loss of generality (see [4]) we further assume that B_a has the form $[0 \mid I_2]$.

Now by superposition, any solution of system (4.13a) which satisfies the initial BC of (4.13b) can be written : $x(t) = x_0(t) + \alpha\, x_1(t) + \beta\, x_2(t)$, where α and β are constants, $x_1(t)$ $(n,1)$ and $x_2(t)$ $(n,1)$ are two linearly independent solutions of the homogeneous system $\dot{x}(t) = A(t)x(t)$ satisfying $B_a\, x_i(a) = 0$ $(1 <= i <= 2)$ and $x_0(t)$ $(n,1)$ is a particular solution of system $\dot{x}(t) = A(t)x(t) + f(t)$ satisfying $B_a\, x_0(a) = c_2$. To solve LBVP (4.13) by the standard complementary function method (see Chapter 2) we would separately compute $x_0(t)$ and $x_i(t)$ $(1 <= i <= 2)$ by forward integration of the inhomogeneous and homogeneous system respectively from $t = a$ to $t = b$ starting from the initial conditions $x_0(a) = [0,\ 0,\ c_2^1,\ c_2^2]^T$, $x_1(a) = [1,\ 0,\ 0,\ 0]^T$ and $x_2(a) = [0,\ 1,\ 0,\ 0]^T$, where $c_2 = [c_2^1,\ c_2^2]^T$ (4.14). This would give the solution $x(t)$ in the form :

$$x(t) = x_0(t) + \alpha\, x_1(t) + \beta\, x_2(t) \qquad (4.15)$$

for all $t \in [a,b]$ and hence the value of $x(b)$. If $x(t)$ is to be the solution of LBVP (4.13) then $x(b)$ must satisfy the final BC of (4.13b) i.e. $B_b\, x(b) = c_1$ ====>

$B_b\, x_0(b) + \alpha B_b\, x_1(b) + \beta B_b\, x_2(b) = c_1$ ====>

$$M \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = s \qquad (4.16)$$

where matrix $M = [B_b\, x_1(b) \mid B_b\, x_2(b)]$ $(2,2)$ and

$s = c_1 - B_b\, x_o(b)$ (2,1). Hence vector $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$ can be obtained by solving (4.16) and then used to find the solution $x(t)$ from (4.15). However, as mentioned in Chapter 2, if system $\dot{x}(t) = A(t)x(t)$ is 'stiff' (i.e. having kinematic eigenvalues which are widely separated in real part) then the linear system (4.16) is liable to be ill conditioned. We now explain how the compound matrix method attempts to overcome this difficulty and later we show the relationship between this method and the singularities of the corresponding Riccati equation.

First we define the part fundamental solution matrix $L(t)$ of dimension (n,p) i.e. (4,2) by $L(t) = [x_1(t) \mid x_2(t)]$ (4.17)

and also the solution matrix $J(t)$ of dimension (n, p + 1) i.e. (4,3) by $J(t) = [x_o(t) \mid x_1(t) \mid x_2(t)]$ (4.18)

where $x_i(t) = [\phi_i(t),\ \phi_i'(t),\ \phi_i''(t),\ \phi_i'''(t)]$, $0 <= i <= 2$.

From $L(t)$ we now obtain the six (i.e. ${}_nC_p = {}_4C_2$) (2,2) minors $y_i(t)$, $1 <= i <= 6$, viz.

$$
\begin{aligned}
y_1(t) &= \phi_1(t).\phi_2'(t) - \phi_2(t).\phi_1'(t) \\
y_2(t) &= \phi_1(t).\phi_2''(t) - \phi_2(t).\phi_1''(t) \\
y_3(t) &= \phi_1(t).\phi_2'''(t) - \phi_2(t).\phi_1'''(t) \\
y_4(t) &= \phi_1'(t).\phi_2''(t) - \phi_2'(t).\phi_1''(t) \\
y_5(t) &= \phi_1'(t).\phi_2'''(t) - \phi_2'(t).\phi_1'''(t) \\
y_6(t) &= \phi_1''(t).\phi_2'''(t) - \phi_2''(t).\phi_1'''(t)
\end{aligned}
\qquad (4.19)
$$

$y_1 \ldots\ldots y_6$ are in fact the Plucker coordinates of the line joining the two points $(\phi_1, \phi_1', \phi_1'', \phi_1''')$ and $(\phi_2, \phi_2', \phi_2'', \phi_2''')$ in S_3 (three dimensional projective space).

It can be verified that the above $y_i(t)$ satisfy the Monge identity : $y_1(t)y_6(t) - y_2(t)y_5(t) + y_3(t)y_4(t) = 0$. (4.20)

We also obtain the four (i.e. ${}_nC_{p+1} = {}_4C_3$) (3,3) minors of $J(t)$ viz. $z_i(t)$, $1 <= i <= 4$:

$$z_1(t) = y_1(t) . \phi_0''(t) - y_2(t) . \phi_0'(t) + y_4(t) . \phi_0(t)$$
$$z_2(t) = y_1(t) . \phi_0'''(t) - y_3(t) . \phi_0'(t) + y_5(t) . \phi_0(t)$$
$$z_3(t) = y_2(t) . \phi_0'''(t) - y_3(t) . \phi_0''(t) + y_6(t) . \phi_0(t)$$
$$z_4(t) = y_4(t) . \phi_0'''(t) - y_5(t) . \phi_0''(t) + y_6(t) . \phi_0'(t) \quad (4.21)$$

We now define the pth (i.e. 2nd) compound vector of $L(t)$ as $y(t) = [y_1(t), \ldots\ldots y_6(t)]^T$ and the (p + 1)th (i.e. 3rd) compound vector of $J(t)$ as $z(t) = [z_1(t), \ldots\ldots z_4(t)]^T$ for all $t \in [a,b]$. By differentiating equations (4.19) and using (4.12) we can show that $y(t)$ satisfies the compound differential system : $\dot{y}(t) = B(t)y(t)$ where $B(t)$ (6,6) is given by :

$$B(t) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ a_3 & a_2 & a_1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ -a_4 & 0 & 0 & a_2 & a_1 & 1 \\ 0 & -a_4 & 0 & -a_3 & 0 & a_1 \end{bmatrix} .$$

Corresponding to the initial values for $x_i(a)$ $(1 <= i <= 2)$ given in (4.14) we get $y(a) = [1, 0, 0, 0, 0, 0,]^T$. Similarly we can show that $z(t)$ satisfies the system :

$\dot{z}(t) = E(t)z(t) + g(t)$, where $E(t)$ (4,4) is given by

$$E(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_2 & a_1 & 1 & 0 \\ -a_3 & 0 & a_1 & 1 \\ a_4 & 0 & 0 & a_1 \end{bmatrix}$$

and $g(t) = [0,\ y_1 a_5,\ y_2 a_5,\ y_4 a_5]^T$ and for which the initial condition corresponding to $x_0(a)$ in (4.14) is

$z(a) = [c_2^1, c_2^2, 0, 0]$ from (4.21).

Now instead of computing $x_i(t)$ $(0 <= i <= 2)$ directly as in the complementary function method we obtain $y(t)$ and $z(t)$ by forward integration of the IVPs :

$$\dot{y}(t) = B(t)y(t), \quad y(a) = [1, 0, 0, 0, 0, 0]^T \qquad (4.22)$$

$$\dot{z}(t) = E(t)z(t) + g(t), \quad z(a) = [c_2^1, c_2^2, 0, 0]^T \qquad (4.23).$$

The solution $x(t)$ of the LBVP satisfies :

$$x(t) - x_0(t) = \alpha x_1(t) + \beta x_2(t) \qquad (4.24)$$

which is four linear equations for α and β. If we denote $x(t)$ by $[\theta(t), \theta'(t), \theta''(t), \theta'''(t)]^T$ then by eliminating α and β from (4.24) in four different ways and then using equations (4.21) we find that the latter are satisfied with function $\phi_0(t)$ replaced by $\theta(t)$ i.e.

$$y_1(t).\theta''(t) - y_2(t).\theta'(t) + y_4(t).\theta(t) = z_1(t) \qquad (i)$$

$$y_1(t).\theta'''(t) - y_3(t).\theta'(t) + y_5(t).\theta(t) = z_2(t) \qquad (ii)$$

$$y_2(t).\theta'''(t) - y_3(t).\theta''(t) + y_6(t).\theta(t) = z_3(t) \qquad (iii)$$

$$y_4(t).\theta'''(t) - y_5(t).\theta''(t) + y_6(t).\theta'(t) = z_4(t) \qquad (iv)$$

(4.25)

for all $t \in [a,b]$. Equations (4.25) can be evaluated at $t = b$ and written in the form of the linear system : $N(b)x(b) = z(b)$ (4.26)

where :

$$N(t) = \begin{bmatrix} y_4 & -y_2 & y_1 & 0 \\ y_5 & -y_3 & 0 & y_1 \\ y_6 & 0 & -y_3 & y_2 \\ 0 & y_6 & -y_5 & y_4 \end{bmatrix} .$$

From the set of equations (4.26) we can in fact obtain only two independent equations for $\theta(b)$, $\theta'(b)$, $\theta''(b)$, $\theta'''(b)$. For example, assuming that $y_4(b) \neq 0$, by applying row operations to (4.26) and using identity (4.20) we can reduce this set of linear equations to the form :

$$\begin{bmatrix} y_4(b) & -y_2(b) & y_1(b) & 0 \\ 0 & y_6(b) & -y_5(b) & y_4(b) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \theta(b) \\ \theta'(b) \\ \theta''(b) \\ \theta'''(b) \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ 0 \\ 0 \end{bmatrix}$$

where h_1 and h_2 are functions of $z_i(b)$ $(1 <= i <= 4)$.

The final BC of (4.13b) provide two more independent linear equations and so we get the linear system :

$$\left[\begin{array}{cccc} y_4(b) & -y_2(b) & y_1(b) & 0 \\ 0 & y_6(b) & -y_5(b) & y_4(b) \\ \hline & & B_b & \end{array}\right] \cdot \begin{bmatrix} \theta(b) \\ \theta'(b) \\ \theta''(b) \\ \theta'''(b) \end{bmatrix} = \begin{bmatrix} h_1 \\ h_2 \\ c_1^1 \\ c_1^2 \end{bmatrix}$$

where $c_1 = [\, c_1^1 , c_1^2 \,]^T$ which can be written :

$$D.x(b) = w \qquad (4.27).$$

The fact that the given LBVP is assumed to have a unique solution ensures that matrix D (n,n) will (theoretically) be non-singular so that x(b) can be computed.

Any one of equations (4.25) can now be used to obtain some of the components of solution $x(t)$ by backward integration from $t = b$. The remaining components can then be directly obtained from equations (4.25) since $z(t)$ is known for all $t \in [a,b]$. For example, the differential equation (4.25 i) viz. $y_1(t) . \theta''(t) - y_2(t) . \theta'(t) + y_4(t) . \theta(t) = z_1(t)$ can be written as the differential system :

$$\dot{u}(t) = F(t)u(t) + 1(t) \qquad (2,2) \qquad (4.28)$$

where $u(t) = [\ \theta(t),\ \theta'(t)]^T$, $F(t) = \begin{bmatrix} 0 & 1 \\ -y_4/y_1 & y_2/y_1 \end{bmatrix}$ and $1(t) = [\ 0,\ z_1/y_1\ ]^T$. Now since $u(b)$ is known from $x(b)$, assuming that $y_1(t) \neq 0$ for all $t \in [a,b]$, system (4.28) can be integrated backwards from $t = b$ to $t = a$ to obtain $u(t)$ for all t. The remaining components $\theta''(t)$ and $\theta'''(t)$ can now be found by simultaneous solution of (4.25)(i) and (ii). Hence we have obtained solution $x(t)$ of the LBVP for all $t \in [a,b]$.

Davey [8] shows that the method will be stable for the case where the system matrix A of the given ODE (4.13a) is constant with eigenvalues of the form $\pm \lambda_1$, $\pm \lambda_2$ or where $A(t)$ is variable but with eigenvalues which are relatively unchanging over the interval $[a,b]$. The method could be extended to the solution of higher order problems but for $n > 4$ the dimensions of the forward IVPs (4.22) and (4.23) rapidly increase. For example, for a problem of size $n = 6$, with $p = 3$, $y(t)$ would be (20,1). A disadvantage of the method though is that it is

only applicable to a LBVP whose ODE is a single nth order equation, as in (4.12).

Relationship of compound matrix method to Riccati method

The main interest of the compound matrix method lies in its relationship to the Riccati transformation and this we now show for the case $n = 4$, $p = q = 2$. Let :

$X(t) = \begin{bmatrix} X_{11}(t) & X_{12}(t) \\ X_{21}(t) & X_{22}(t) \end{bmatrix}$ be the fundamental solution of system $\dot{x}(t) = A(t)x(t)$ corresponding to the initial condition $X(a) = I_4$, then the solution $R(t)$ (2,2) of the Riccati equation :

$\dot{R}(t) = A_{21}(t) + A_{22}(t)R(t) - R(t)A_{11}(t) - R(t)A_{12}(t)R(t)$, for which $R(a) = 0$, is given by $R(t) = X_{21}(t).X_{11}^{-1}(t)$, as shown in the previous section. Now the part fundamental solution $L(t)$ (4,2) is $L(t) = [x_1(t) \mid x_2(t)]$ where $x_1(a) = [1, 0, 0, 0]^T$ and $x_2(a) = [0, 1, 0, 0]^T$, from (4.14), so that $L(a) = \begin{bmatrix} I_2 \\ 0 \end{bmatrix}$.

Thus $\begin{bmatrix} X_{11}(t) \\ X_{21}(t) \end{bmatrix} = L(t) = \left[\begin{array}{cc} \phi_1(t) & \phi_2(t) \\ \phi_1'(t) & \phi_2'(t) \\ \hline \phi_1''(t) & \phi_2''(t) \\ \phi_1'''(t) & \phi_2'''(t) \end{array}\right]$

and therefore if we denote $R(t)$ by $\begin{bmatrix} r_1(t) & r_2(t) \\ r_3(t) & r_4(t) \end{bmatrix}$

we get : $\begin{bmatrix} r_1 & r_2 \\ r_3 & r_4 \end{bmatrix} = \begin{bmatrix} \phi_1'' & \phi_2'' \\ \phi_1''' & \phi_2''' \end{bmatrix} \cdot \begin{bmatrix} \phi_1 & \phi_2 \\ \phi_1' & \phi_2' \end{bmatrix}^{-1}$

$= \begin{bmatrix} \phi_1'' & \phi_2'' \\ \phi_1''' & \phi_2''' \end{bmatrix} \cdot \begin{bmatrix} \phi_2' & -\phi_2 \\ -\phi_1' & \phi_1 \end{bmatrix} \cdot \frac{1}{(\phi_1\phi_2' - \phi_2\phi_1')}$

$\Longrightarrow r_1 = \frac{-y_4}{y_1}$, $r_2 = \frac{y_2}{y_1}$, $r_3 = \frac{-y_5}{y_1}$, $r_4 = \frac{y_3}{y_1}$ (4.29)

where $y_1(a) = 1$. Thus we see that whenever $y_1(t)$ becomes zero in the solution $y(t)$ of the IVP $\dot{y}(t) = B(t)y(t)$, $y(a) = [1, 0, 0, 0, 0, 0]^T$ then the Riccati solution $R(t)$ will have a pole at that value of t. Alternatively, if we define $r_i(t)$ $(1 <= i <= 4)$ by (4.29) and then differentiate these equations to obtain $\dot{r}_i(t)$ and substitute from the compound differential system $\dot{y}(t) = B(t)y(t)$ we can show that these $r_i(t)$ are indeed the elements of the Riccati solution matrix $R(t)$. For example, from (4.29) : $\dot{r}_1 = \frac{-y_1\dot{y}_4 + y_4\dot{y}_1}{y_1^2}$

and from $\dot{y} = By$ we have : $\dot{y}_1 = y_2$, $\dot{y}_4 = y_5$. Thus

$\dot{r}_1 = \frac{-y_1 y_5 + y_4 y_2}{y_1^2} = \frac{-y_5}{y_1} + \frac{y_4}{y_1}\cdot\frac{y_2}{y_1}$

i.e. $\dot{r}_1 = r_3 - r_1 r_2$. Now the Riccati equation

$\dot{R} = A_{21} + A_{22}R - RA_{11} - RA_{12}R$ (4.30)

$\Longrightarrow$

$$\begin{bmatrix} \dot{r}_1 & \dot{r}_2 \\ \dot{r}_3 & \dot{r}_4 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ a_4 & a_3 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ a_2 & a_1 \end{bmatrix}\begin{bmatrix} r_1 & r_2 \\ r_3 & r_4 \end{bmatrix} - \begin{bmatrix} r_1 & r_2 \\ r_3 & r_4 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} r_1 & r_2 \\ r_3 & r_4 \end{bmatrix}\begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} r_1 & r_2 \\ r_3 & r_4 \end{bmatrix}$$

$\Longrightarrow \dot{r}_1 = r_3 - r_1 r_2$. Similarly we can verify the

elements r_2 , r_3 and r_4 . Thus the six linear equations of the compound system $\dot{y}(t) = B(t)y(t)$ can be reduced to a system of four non-linear Riccati equations.

Finally we may note that the Riccati transformation can also be used in the factorisation method of Babuska and Majer [16]] and this is the subject of the next Chapter.

In Appendix I :

[4-1] : Restart values for $\tilde{R}(t^*)$ and $\tilde{y}_2(t^*)$.

[4-2] : Restart value for $y_1(t^*)$.

[4-3] : $\tilde{R}(t^*) = FE^{-1}$.

CHAPTER 5

FACTORISATION METHODS

Partitioning

Up to now, in all of the previous Chapters, we have partitioned the system matrix $A(t)$ of the given (well conditioned) LBVP (1.15) as

$$A(t) = \begin{matrix} \\ p \\ q \end{matrix} \begin{matrix} \begin{matrix} p & \quad q \end{matrix} \\ \begin{bmatrix} A_{11}(t) & A_{12}(t) \\ A_{21}(t) & A_{22}(t) \end{bmatrix} \end{matrix}$$

and the solution $x(t)$ as $\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \begin{matrix} p \\ q \end{matrix}$ to correspond with the partitioning of the transformation matrix $T(t) = [T_1(t) \overset{p}{|} T_2(t) \overset{q}{}]$ where p and q were the dimensions of the (forward) growth and decay subspaces respectively. This follows the notation used by Van Loon [10, 17], Mattheij [1] and Russell [12].

In this Chapter, however, to facilitate understanding of the bounded factorisation methods, as put forward by Babuska and Majer in [16], we partition the problem as shown below where n_1 is the dimension of the (forward) decay subspace and n_2 that of the (forward) growth subspace. We therefore restate the given (well conditioned) (n,n) LBVP as :

$$\dot{x}(t) = B(t)x(t) - f(t) \qquad (5.1a)$$

$$D_1 x(a) + D_2 x(b) = c \qquad (5.1b)$$

for $a <= t <= b$ where $B(t) = \begin{bmatrix} B_{11}(t) & B_{12}(t) \\ B_{21}(t) & B_{22}(t) \end{bmatrix} \begin{matrix} n_1 \\ n_2 \end{matrix}$ (columns n_1, n_2),

$x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \begin{matrix} n_1 \\ n_2 \end{matrix}$, $c = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$, $D_1 = \begin{bmatrix} U_1 \\ 0 \end{bmatrix}$

$D_2 = \begin{bmatrix} 0 \\ U_2 \end{bmatrix}$, $f(t) = \begin{bmatrix} f_1(t) \\ f_2(t) \end{bmatrix}$ and

$U_1 = n_1 \begin{bmatrix} K_0 & K_1 \end{bmatrix}$ (n_1, n_2 columns), $U_2 = \begin{bmatrix} K_2 & K_3 \end{bmatrix} n_2$ (n_1, n_2 columns) where K_0 and K_3 are assumed to be non-singular and where n_1 and n_2 are the dimensions of the (forward) decay and growth subspaces respectively. This follows the notation used by Babuska in [16].

Propagation of BC

The notion which underlies the factorisation methods is that we propagate forwards (backwards) a set of conditions equivalent to the initial (final) boundary conditions of LBVP (5.1) so as to obtain a complete set of n independent conditions at any point $t = t^*$ in $[a,b]$ at which the solution $x(t)$ of (5.1) is required. Factorisation is a double sweep method which can be applied to either the continuous orthonormal or to the Riccati transformation or even to single shooting superposition. However, unlike the double sweep methods that we have looked at in previous Chapters, here the forward and backward sweeps are independent in that each sweep employs a different form of the same transformation (either continuous orthonormal or Riccati)

of which only one part is used. Let us call the forward and backward sweep transformation matrices $T(t)$ and $L(t)$ respectively where these are partitioned $[T_1(t) \mid T_2(t)]$ (n_1, n_2) and likewise for $L(t)$. For the forward sweep let :

$$T^{-1}(t) = S(t) = \begin{bmatrix} \Phi_1(t) \\ * \end{bmatrix} \begin{matrix} n_1 \\ n_2 \end{matrix}$$, where $*$ denotes a (n_2, n) matrix with which we will not be concerned here. Then

$x(t) = T(t)y(t) \quad \Longrightarrow$

$$y(t) = T^{-1}(t)x(t) \quad \Longrightarrow \quad \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} \Phi_1(t) \\ * \end{bmatrix} . x(t)$$

$$\Longrightarrow \quad \Phi_1(t).x(t) = y_1(t) \quad \text{or} \quad \Phi_1(t).x(t) = \phi_1(t) \qquad (5.2)$$

putting $\phi_1(t) = y_1(t)$

where $\Phi_1(t)$ (n_1, n) and $\phi_1(t)$ $(n_1, 1)$ are the forward transition matrix and vector respectively.

Likewise for the backward sweep let $L^{-1}(t) = J(t) = \begin{bmatrix} * \\ \Phi_2(t) \end{bmatrix} \begin{matrix} n_1 \\ n_2 \end{matrix}$

where now, of course, for backward integration from $t = b$ to $t = a$, n_1 is the dimension of the growth subspace. Then $x(t) = L(t)r(t) \Longrightarrow r(t) = L^{-1}(t)x(t)$, where $r(t)$ is the solution of the transformed system, $\Longrightarrow$

$$\begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix} = \begin{bmatrix} * \\ \Phi_2(t) \end{bmatrix} .x(t) \qquad \Longrightarrow \qquad \Phi_2(t)\, x(t) = r_2(t)$$

or $\Phi_2(t)\, x(t) = \phi_2(t)$ (5.3)

where $\Phi_2(t)$ (n_2, n) and $\phi_2(t)$ $(n_2, 1)$ are the

backward transition matrix and vector respectively. Equations (5.2) and (5.3) are the forward and backward transition equations and if we combine them at any value of $t^* \in [a,b]$ we get the combined transition equation :

$$\begin{bmatrix} \Phi_1(t^*) \\ \Phi_2(t^*) \end{bmatrix} . \; x(t^*) \; = \; \begin{bmatrix} \phi_1(t^*) \\ \phi_2(t^*) \end{bmatrix} \qquad (5.4)$$

from which the solution $x(t^*)$ of the LBVP (5.1) is obtained. It is shown in [16] that if LBVP (5.1) has a unique solution then so will linear system (5.4). Also, provided the IVPs for $\Phi_i(t)$ and $\phi_i(t)$ $(1 <= i <= 2)$ are well conditioned then so will be system (5.4). In later sections we derive these IVPs (known as the factorisation transition equations).

The distinguishing feature of the factorisation methods is the way in which the given initial (final) BC is propagated forwards (backwards) across the problem interval [a,b]. For the forward sweep, the given initial BC of LBVP (5.1) is $U_1 x(a) = c_1$ or $[K_0 \mid K_1] x(a) = c_1$. As explained later, this BC is transformed into an equivalent set of conditions $\Phi_1(a) x(a) = d_1$, where $\Phi_1(a)$ must be of the same form as the first n_1 rows of transformation $T^{-1}(a)$ i.e. $T^{-1}(a) = \begin{bmatrix} \Phi_1(a) \\ * \end{bmatrix} \begin{matrix} n_1 \\ n_2 \end{matrix}$. Thus for the Riccati application $\Phi_1(a)$ must have the form $[I_{n_1} \mid R(a)]$ because here $T^{-1}(t) = \begin{bmatrix} I_{n_1} & R(t) \\ 0 & I_{n_2} \end{bmatrix}$, whilst for the orthonormal

application $\Phi_1(a)$ must be a row set of orthonormal vectors because in this case $T^{-1}(t) = T^T(t) = [T_1(t) \mid T_2(t)]^T$ $= \begin{bmatrix} T_1^T(t) \\ T_2^T(t) \end{bmatrix}$ where $T(t)$ is an orthonormal matrix for all t. In the Riccati case this transformation could simply be achieved by pre-multiplication by K_0^{-1} i.e. $[K_0 \mid K_1]\, x(a) = c_1$ ====> $[I_{n_1} \mid K_0^{-1} K_1]\, x(a) = K_0^{-1} c_1 = d_1$. But this might result in some of the elements of $R(a) = K_0^{-1} K_1$ being large in modulus. To avoid this we adopt a preliminary procedure of alternate column interchanges and row operations described later so as to finish up with $R(a)$ having all its elements in modulus less than or equal to one. (This will most likely cause a re-arrangement of the components of the LBVP solution $x(t)$ thereby necessitating a re-imbedding of the ODE).

Now for either the Riccati or the orthonormal application at any value of $t \in [a,b]$ we have $x(t) = T(t)y(t)$ ====> $\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} \Phi_1(t) \\ * \end{bmatrix} \cdot x(t)$ ====>

$$\Phi_1(t)x(t) = y_1(t) = \phi_1(t) \quad ====> \quad \Phi_1(a)x(a) = \phi_1(a).$$

If we compare the latter with the equivalent set of initial conditions $\Phi_1(a)x(a) = d_1$ that we have obtained we see that forward integration of the ODEs for $\Phi_1(t)$ and $\phi_1(t)$ (viz. (5.6) and (5.5) below) from initial conditions of $\Phi_1(a)$ and $\phi_1(a) = d_1$ will produce a set of conditions equivalent to the initial BC of LBVP (5.1) for all $t \in [a,b]$.

Likewise, in the backward sweep the final given BC is propagated

from $t = b$ to $t = a$ in a similar fashion.

Forward Sweep

Consider the forward sweep. First the given LBVP (5.1) must be recast so that the initial BC matrix U_1 is in the correct form (see later) for the particular transformation to be used (either continuous orthonormal or Riccati). In the case of the Riccati transformation this will probably require a re-imbedding of the LBVP. We give details later of precisely how this is done when we examine each transformation individually. For now we will assume that this preliminary transformation has been done so that matrix U_1 in (5.1) is already in the correct form.

Recall from Chapter 1 that if in general we apply the transformation $x(t) = N(t)v(t)$ or $v(t) = N^{-1}(t)x(t) = M(t)x(t)$ to the ODE $\dot{x}(t) = B(t)x(t) - f(t)$ of LBVP (5.1) we get the transformed system : $\dot{v}(t) = W(t)v(t) - \tilde{f}(t)$ where $W(t)$ and $M(t)$ are connected by the Lyapunov equation $\dot{M}(t) = W(t)M(t) - M(t)B(t)$ and $\tilde{f}(t) = M(t)f(t)$.

For the forward sweep we use the transformation $x(t) = T(t)y(t)$ or $y(t) = S(t)x(t)$ on $\dot{x}(t) = B(t)x(t) - f(t)$ to obtain the transformed system $\dot{y}(t) = \tilde{B}(t)y(t) - g(t)$ where $T(t)$ is chosen such that $\tilde{B}(t)$ is block <u>lower</u> triangular. Then we have : $\dot{S}(t) = \tilde{B}(t)S(t) - S(t)B(t)$ and $g(t) = S(t)f(t)$ where $S(t) = \begin{bmatrix} \Phi_1(t) \\ * \end{bmatrix}$. Thus $\dot{y}(t) = \tilde{B}(t)y(t) - g(t)$ ====>

$$\begin{bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \end{bmatrix} = \begin{bmatrix} \tilde{B}_{11}(t) & 0 \\ \tilde{B}_{21}(t) & \tilde{B}_{22}(t) \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} - \begin{bmatrix} \Phi_1(t) \\ * \end{bmatrix} . f(t)$$

i.e. $\dot{y}_1(t) = \tilde{B}_{11}(t) y_1(t) - \Phi_1(t) f(t)$ or

$$\dot{\phi}_1(t) = -\Phi_1(t) f(t) + Z_1(t) . \phi_1(t) \qquad (5.5)$$

putting $y_1 = \phi_1(n_1, 1)$ and $Z_1 = \tilde{B}_{11}$ where $Z_1(t)$ (n_1, n_1) is the conditioning matrix of this IVP.

Also : $\dot{S}(t) = \tilde{B}(t)S(t) - S(t)B(t)$ ====>

$$\begin{bmatrix} \dot{\Phi}_1(t) \\ * \end{bmatrix} = \begin{bmatrix} \tilde{B}_{11}(t) & 0 \\ \tilde{B}_{21}(t) & \tilde{B}_{22}(t) \end{bmatrix} \begin{bmatrix} \Phi_1(t) \\ * \end{bmatrix} - \begin{bmatrix} \Phi_1(t) \\ * \end{bmatrix} . B(t)$$

i.e. $\dot{\Phi}_1(t) = \tilde{B}_{11}(t) . \Phi_1(t) - \Phi_1(t)B(t)$ or

$$\dot{\Phi}_1(t) = -\Phi_1(t)B(t) + Z_1(t) . \Phi_1(t) \qquad (5.6).$$

To obtain the initial conditions for these IVPs (5.5) and (5.6) note that from equation (5.2) we get : $\Phi_1(a)x(a) = \phi_1(a)$ whilst from the initial BC of LBVP (5.1) we have :

$U_1 . x(a) = c_1$. Thus we take $\Phi_1(a) = U_1$ and $\phi_1(a) = c_1$.

Now the transformed initial BC is $U_1 T(a) y(a) = c_1$

====> $\begin{bmatrix} U_1 T_1(a) & | & U_1 T_2(a) \end{bmatrix} \begin{bmatrix} y_1(a) \\ y_2(a) \end{bmatrix} = c_1$

====> $U_1 T_1(a) . \phi_1(a) + U_1 T_2(a) y_2(a) = c_1$ since $y_1 = \phi_1$

Hence the transformation $T(a)$ must be such that

$$U_1 T_1(a) = I_{n_1} \qquad (5.7a)$$

$$U_1 T_2(a) = 0 \qquad (5.7b)$$

where $T(t)$ may be either continuous orthonormal or Riccati.

Note that condition (5.7b) corresponds to (1.17) of Chapter 1.

Backward Sweep

Now we consider the backward sweep, for which a similar argument applies. First the LBVP (5.1) must be recast to obtain the final BC matrix U_2 in the correct form for either the Riccati or the continuous orthonormal transformation. Again, for now, we will assume that this has already been done in (5.1). For the backward sweep we use the transformation $x(t) = L(t)r(t)$ or $r(t) = J(t)x(t)$ on $\dot{x}(t) = B(t)x(t) - f(t)$ to obtain the transformed system $\dot{r}(t) = C(t)r(t) - h(t)$ where $L(t)$ is chosen so that $C(t)$ is block upper triangular. Thus we have $\dot{J}(t) = C(t)J(t) - J(t)B(t)$ and $h(t) = J(t)f(t)$ where

$$J(t) = \begin{bmatrix} * \\ \Phi_2(t) \end{bmatrix} \begin{matrix} n_1 \\ n_2 \end{matrix} \quad \text{and so : } \dot{r}(t) = C(t)r(t) - h(t) \quad ====>$$

$$\begin{bmatrix} \dot{r}_1(t) \\ \dot{r}_2(t) \end{bmatrix} = \begin{bmatrix} C_{11}(t) & C_{12}(t) \\ 0 & C_{22}(t) \end{bmatrix} \begin{bmatrix} r_1(t) \\ r_2(t) \end{bmatrix} - \begin{bmatrix} * \\ \Phi_2(t) \end{bmatrix} . f(t)$$

i.e. $\dot{r}_2(t) = C_{22}(t)r_2(t) - \Phi_2(t)f(t)$ or

$$\dot{\phi}_2(t) = - \Phi_2(t)f(t) + Z_2(t) . \phi_2(t) \qquad (5.8)$$

putting $r_2(t) = \phi_2(t)\ (n_2, 1)$ and $C_{22}(t) = Z_2(t)\ (n_2, n_2)$.

Also : $\dot{J}(t) = C(t)J(t) - J(t)B(t) \quad ====>$

$$\begin{bmatrix} * \\ \dot{\Phi}_2(t) \end{bmatrix} = \begin{bmatrix} C_{11}(t) & C_{12}(t) \\ 0 & C_{22}(t) \end{bmatrix} \begin{bmatrix} * \\ \Phi_2(t) \end{bmatrix} - \begin{bmatrix} * \\ \Phi_2(t) \end{bmatrix} . B(t)$$

$====>$ $\dot{\Phi}_2(t) = C_{22}(t) . \Phi_2(t) - \Phi_2(t)B(t)$ or

$$\dot{\Phi}_2(t) = - \Phi_2(t)B(t) + Z_2(t) . \Phi_2(t) \qquad (5.9).$$

From (5.3) we have : $\Phi_2(b)x(b) = \phi_2(b)$ and from the final BC of (5.1) : $U_2 x(b) = c_2$. Hence we have initial conditions

for IVPs (5.8) and (5.9) of $\phi_2(b) = c_2$ and $\Phi_2(b) = U_2$.

Also : $U_2\, x(b) = c_2$ ====> $U_2\, L(b) r(b) = c_2$ ====>

$U_2\, L_1(b) r_1(b) + U_2\, L_2(b) . \phi_2(b) = c_2$ (since $r_2 = \phi_2$)

and so the transformation $L(t)$ must be such that :

$U_2\, L_1(b) = 0$ and $U_2\, L_2(b) = I_{n_2}$.

Note that if we define $\underline{\Phi}(t)$ (n, n), $\underline{\phi}(t)$ $(n,1)$ and $Z(t)$ (n,n) by :

$$\underline{\Phi}(t) = \begin{bmatrix} \Phi_1(t) \\ \Phi_2(t) \end{bmatrix} \begin{matrix} n_1 \\ n_2 \end{matrix}, \quad \underline{\phi}(t) = \begin{bmatrix} \phi_1(t) \\ \phi_2(t) \end{bmatrix} \begin{matrix} n_1 \\ n_2 \end{matrix} \quad \text{and}$$

$$Z(t) = \begin{bmatrix} Z_1(t) & 0 \\ 0 & Z_2(t) \end{bmatrix} \begin{matrix} n_1 \\ n_2 \end{matrix} -$$

then we can combine the forward and backward transition matrix equations (5.6) and (5.9) into $\dot{\underline{\Phi}}(t) = - \underline{\Phi}(t) B(t) + Z(t) . \underline{\Phi}(t)$ and the corresponding transition vector equations (5.5) and (5.8) into $\dot{\underline{\phi}}(t) = - \underline{\Phi}(t) f(t) + Z(t) . \underline{\phi}(t)$.

Application to Riccati transformation

We look now in more detail at how the factorisation method can be applied to the Riccati transformation. For the forward sweep we must first re-cast the given LBVP (5.1) so as to put the initial BC matrix U_1 into the required form as follows. The given initial BC of (5.1) is :

$$[K_0 \mid K_1] . x(a) = c_1 \qquad (5.10).$$

Alternate column interchanges and row operations are now performed on matrix $[K_0 \mid K_1]$ of the form :

$G_{n_1-1} \ldots\ldots\ldots G_1\ [K_0 \mid K_1]\ P_1 \ldots\ldots\ldots P_{n_1}$ where P_i (n,n) is the perm matrix which takes the max mod element of the ith row $(1 <= i <= n_1)$ to the ith column and G_i (n_1, n_1) is the matrix which performs row Gaussian elimination using the (ith, ith) element as pivot. The sequence of operations is thus : $P_1, G_1, P_2, G_2 \ldots\ldots\ldots P_{n_1}$ e.g. after the first cycle of operations we have from (5.10) :

$G_1\ [K_0 \mid K_1]\ P_1\ P_1^T\ x(a) = G_1\ c_1$ (since $P_1\ P_1^T = I_n$)

or $[\bar{K}_0 \mid \bar{K}_1]\ \bar{x}(a) = \bar{c}_1$ where $\bar{x}(a) = P_1^T\ x(a)$.

Therefore if we denote the composite perm $P_1\ P_2 \ldots\ldots\ldots P_{n_1}$ by P we finish with a set of conditions equivalent to (5.10) of the form : $[E \mid F]\ P^T\ x(a) = d_1$ which can be written : $[I_{n_1} \mid E^{-1}\ F]\ \tilde{x}(a) = u_1$ or

$$[I_{n_1} \mid R(a)]\ \tilde{x}(a) = u_1 \qquad (5.11)$$

where $R(a) = E^{-1}F$, $u_1 = E^{-1}d_1$ and $\tilde{x}(a) = P^T\ x(a)$.

It is shown in [16] that the above procedure will ensure that all the components of R(a) will now be in modulus less than or equal to one. However, it has caused a re-arrangement of the components of solution $x(t)$ defined by $\tilde{x}(t) = P^T\ x(t)$ and so we must re-imbed the ODE of LBVP (5.1) to take account of this. In the original given imbedding the ODE (5.1a) was

$$\dot{x}(t) = B(t)x(t) - f(t) \qu\Longrightarrow$$

$$P^T\ \dot{x}(t) = P^T\ B(t)\{P\ P^T\}x(t) - P^T\ f(t) \quad \Longrightarrow$$

$$\frac{d}{dt}\ \tilde{x}(t) = A(t)\tilde{x}(t) - 1(t) \qquad (5.12)$$

where $A(t) = P^T\ B(t)P$ and $1(t) = P^T\ f(t)$. Thus after

re-casting the given initial BC (5.10) into the Riccati form (5.11), the corresponding re-imbedded ODE is now (5.12).

For the forward sweep we now define the Riccati transformation by $T(t) = \begin{bmatrix} I_{n_1} & -R(t) \\ 0 & I_{n_2} \end{bmatrix}$ (5.13)

where $R(t)$ (n_1 , n_2) is the Riccati matrix, so that $S(t) = T^{-1}(t) = \begin{bmatrix} I_{n_1} & R(t) \\ 0 & I_{n_2} \end{bmatrix}$. In order that the transformation $\tilde{x}(t) = T(t)z(t)$ will put ODE (5.12) into the form $\dot{z}(t) = \tilde{A}(t)z(t) - h(t)$ where $\tilde{A}(t)$ is block lower triangular we show in [5-1] that $R(t)$ must satisfy the Riccati equation :

$$\dot{R}(t) = A_{11}(t)R(t) + R(t)A_{21}(t)R(t) - A_{12}(t) - R(t)A_{22}(t) \quad (5.14)$$

and that $\tilde{A}(t)$ is then given by :

$$\tilde{A}(t) = \begin{bmatrix} A_{11}(t) + R(t)A_{21}(t) & 0 \\ A_{21}(t) & A_{22}(t) - A_{21}(t)R(t) \end{bmatrix} . \quad (5.15)$$

Also, since $S(t) = T^{-1}(t) = \begin{bmatrix} I_{n_1} & R(t) \\ 0 & I_{n_2} \end{bmatrix}$, the forward transition matrix $\Phi_1(t) = [I_{n_1} , R(t)]$ (n_1 , n) and the condition matrix $Z_1(t) = \tilde{A}_{11}(t) = A_{11}(t) + R(t)A_{21}(t)$. From (5.11), the initial condition for the forward integration of the $\dot{\Phi}_1(t)$ IVP is $\Phi_1(a) = [I_{n_1} , R(a)]$ and for the $\dot{\varnothing}_1(t)$ IVP we have $\varnothing_1(a) = u_1$. Note that $[I_{n_1} , R(a)].T_1(a) = [I_{n_1} , R(a)].\begin{bmatrix} I_{n_1} \\ 0 \end{bmatrix} = I_{n_1}$ and that $[I_{n_1} , R(a)].T_2(a) =$

$[I_{n_1}, R(a)]. \begin{bmatrix} -R(a) \\ I_{n_2} \end{bmatrix} = 0$, as required in (5.7a & b).

In practice, we do not need to integrate $\Phi_1(t)$ forwards because $\Phi_1(t) = [I_{n_1}, R(t)]$ for all t. Instead we integrate forwards the Riccati equation (5.14) from its initial value of $R(a) = E^{-1}F$ (which gives us $\Phi_1(t)$) simultaneously with the $\phi_1(t)$ IVP (5.5) viz.

$$\dot{\phi}_1(t) = -\Phi_1(t)f(t) + Z_1(t).\phi_1(t), \qquad \phi_1(a) = u_1 .$$

Hence we obtain $\Phi_1(t)$ and $\phi_1(t)$ for all $t \in [a,b]$ (assuming that $R(t)$ remains bounded : we return to discussion of this point later).

For the backward sweep, the given LBVP (5.1) must first be re-cast so as to put the final BC matrix U_2 into the required form as follows. The final BC of (5.1) is :

$$[K_2, K_3].x(b) = c_2 \qquad (5.16)$$

Alternate column interchanges and row operations (similar to those described earlier for the forward sweep) are now performed on matrix $[K_2, K_3]$ except that in this case each perm matrix Q_i (n,n) takes the max mod element in the ith row ($1 <= i <= n_2$) to the $(n - i + 1)$th column and the $(i, n - i + 1)$th element is then used as the pivot for the row Gaussian elimination. We thus obtain a set of conditions equivalent to (5.16) of the form : $[E_1 \mid F_1].Q^T x(b) = e_1$ where Q is the composite perm matrix $Q_1 Q_2 \ldots\ldots Q_{n_2}$. This can now be written : $[P(b) \mid I_{n_2}].\tilde{x}(b) = \bar{v}_1$ (5.17) where $P(b) = F_1^{-1} E_1$, $\bar{v}_1 = F_1^{-1} e_1$ and $\tilde{x}(b) = Q^T x(b)$.

As in the forward sweep, this process ensures that all of the components of $P(b)$ will be in modulus less than or equal to one. It also necessitates a re-imbedding of the given ODE (5.1) defined by $\tilde{x}(t) = Q^T x(t)$. Let us suppose that the re-imbedded version of ODE $\dot{x}(t) = B(t)x(t) - f(t)$ is

$$\frac{d}{dt}\tilde{x}(t) = \tilde{B}(t)\tilde{x}(t) - \tilde{f}(t) \qquad (5.18)$$

where $\tilde{B}(t) = Q^T B(t)Q$ and $\tilde{f}(t) = Q^T f(t)$, for which the corresponding final BC is (5.17).

For the backward sweep we define the Riccati transformation by

$$L(t) = \begin{bmatrix} I_{n_1} & 0 \\ -P(t) & I_{n_2} \end{bmatrix}$$ where $P(t)$ is the Riccati solution matrix (n_2, n_1) and for which $L^{-1}(t) = J(t) = \begin{bmatrix} I_{n_1} & 0 \\ P(t) & I_{n_2} \end{bmatrix}$

The transformation $\tilde{x}(t) = L(t)r(t)$ will put ODE (5.18) into the form $\dot{r}(t) = C(t)r(t) - h(t)$, where $C(t)$ is block upper triangular, provided $P(t)$ satisfies the Riccati equation :

$$\dot{P}(t) = P(t)\tilde{B}_{12}(t)P(t) + \tilde{B}_{22}(t)P(t) - P(t)\tilde{B}_{11}(t) - \tilde{B}_{21}(t) \qquad (5.19)$$

and in this case :

$$C(t) = \begin{bmatrix} \tilde{B}_{11}(t) - \tilde{B}_{12}(t)P(t) & \tilde{B}_{12}(t) \\ 0 & \tilde{B}_{22}(t) + P(t)\tilde{B}_{12}(t) \end{bmatrix} \qquad (5.20).$$

Also since $J(t) = L^{-1}(t) = \begin{bmatrix} I_{n_1} & 0 \\ P(t) & I_{n_2} \end{bmatrix}$ we have

$\Phi_2(t) = [P(t) \mid I_{n_2}]$ (n_2, n) and the condition matrix $Z_2(t) = C_{22}(t) = \tilde{B}_{22}(t) + P(t)\tilde{B}_{12}(t)$. Thus we integrate

simultaneously backwards (from $t = b$ to $t = a$) the Riccati equation (5.19) from its initial condition $P(b) = F_1^{-1} E_1$ together with the $\phi_2(t)$ IVP (5.8) viz.

$\dot{\phi}_2(t) = - \Phi_2(t) f(t) + Z_2(t) . \phi_2(t), \quad \phi_2(b) = \bar{v}_1 .$

Hence we obtain $\Phi_2(t)$ and $\phi_2(t)$ for all $t \in [a,b]$, provided $P(t)$ remains bounded.

Restart re-imbedding procedure

As described in Chapter 4, the chief drawback of the Riccati method is that the Riccati solution (either $R(t)$ forwards or $P(t)$ backwards) may have a pole at some value of t in $[a,b]$. We avoid this by keeping the transition IVPs $\Phi_i(t)$ and $\phi_i(t)$ $(1 <= i <= 2)$ well conditioned by adopting the following restart re-imbedding procedure. For the forward sweep the integration of the Riccati equation and $\phi_1(t)$ equation are continued from $t = a$ until a value of t is reached at which :

$$\| \Phi_1(t) \| > \rho \cdot \| \Phi_1(a) \| \qquad (5.21)$$

where ρ is a pre-selected small positive constant. If this happens at $t = t_1 < b$ then we restart. Suppose the ODE for the first subinterval $[a,t_1]$ is :

$\dot{x}^\circ(t) = B^\circ(t) x^\circ(t) - f^\circ(t)$, where superscript $^\circ$ denotes values in the imbedding for this subinterval. At $t = t_1$ the propagated initial BC (equivalent to the initial BC at $t = a$ of the given LBVP) are $\Phi_1^\circ(t_1) x^\circ(t_1) = \phi_1^\circ(t_1)$ or

$[I_{n_1} \mid R^{\circ}(t_1)]\, x^{\circ}(t_1) = \phi_1^{\circ}(t_1)$ (5.22).

In order to make all of the components of the restart Riccati solution matrix, at $t = t_1$, in mod value less than or equal to one, we perform the previously described row and column operations on $[I_{n_1} \mid R^{\circ}(t_1)]$ causing a change of imbedding. This produces a set of conditions equivalent to (5.22) of the form $[I_{n_1} \mid R'(t_1)]\, x'(t_1) = \phi_1'(t_1)$ where superscript $'$ denotes values in the new imbedding for $t >= t_1$. The ODEs in this new imbedding are $\dot{x}'(t) = B'(t)x'(t) - f'(t)$ and the transition matrix and vector are now $\Phi_1'(t)$ and $\phi_1'(t)$ respectively. The Riccati equation corresponding to this imbedding viz.

$\dot{R}'(t) = B_{11}'(t)R'(t) + R'(t)B_{21}'(t)R'(t) - B_{12}'(t) - R'(t)B_{22}'(t)$

is now restarted from the value $R'(t_1)$ and the transition vector equation $\dot{\phi}_1'(t) = -\Phi_1'(t)f'(t) + Z_1'(t) \cdot \phi_1'(t)$ (where $Z_1'(t) = B_{11}'(t) + R'(t)B_{21}'(t)$) is restarted from $\phi_1'(t_1)$. Note that the Riccati transformation matrix corresponding to this new imbedding is $\begin{bmatrix} I_{n_1} & -R'(t) \\ 0 & I_{n_2} \end{bmatrix}$ for all $t >= t_1$.

The integrations are now continued until a value of t is reached where $\| \Phi_1'(t) \| > \rho \cdot \| \Phi_1'(t_1) \|$. If this occurs at some value of $t = t_2 < b$ then the above restart procedure must be repeated at $t = t_2$, and so on until $t = b$ is reached. After each restart at $t = t_i$ the restart Riccati solution matrix $R'(t_i)$ will have all of its components in

mod value less than or equal to one, which helps to reduce the number of restarts likely to be needed. Also if, after a restart, the new imbedding is an unstable one in that the corresponding Riccati solution is exponentially increasing or has a singular point at some value of $t > t_i$, then the criterion $\| \Phi_1^i(t) \| > \rho . \| \Phi_1^i(t_i) \|$ will ensure that the length of the subinterval in this imbedding will be short. Similar remarks to those above apply also to the backward integration of the Riccati P(t) equation (5.19).

Note that whenever a re-imbedding occurs (in either sweep) the composite perm matrix which produces this re-arrangement must be stored. Suppose that re-imbeddings occur at the nodes $a = t_0 < t_1 < \ldots\ldots < t_{N-1} < t_N = b$ during the forward sweep and at $a = s_M < s_{M-1} < \ldots\ldots\ldots < s_2 < s_1 < s_0 = b$ in the backward sweep. For the forward sweep, in subinterval $[t_i , t_{i+1}]$ $(0 <= i <= N - 1)$ denote the re-imbedded solution vector by $x^i(t)$ and the transition matrix and vector by $\Phi_1^i(t)$ and $\phi_1^i(t)$ respectively. Also let the composite perm matrix which changes the imbedding at $t = t_i$ be P_i^T. Then if $x(t)$ denotes the original imbedding of the solution (as in the given LBVP) we have :

$x^i(t) = P_i^T P_{i-1}^T \ldots\ldots\ldots P_0^T x(t) = P\, x(t)$, say.

Similarly, for the backward sweep, in subinterval $[s_{j+1}, s_j]$ $(0 <= j <= M - 1)$ denote the LBVP solution, transition matrix and vector by $x^j(t)$, $\Phi_2^j(t)$ and $\phi_2^j(t)$ respectively.

Let the composite perm matrix which changes the imbedding at $t = s_j$ be Q_j^T. Then we have :

$x^j(t) = Q_j^T \ldots\ldots\ldots Q_0^T \, x(t) = Q\, x(t)$, say.

Now suppose that we require the solution to the given LBVP at $t = t^*$ where $t^* \in [t_i , t_{i+1}]$ in the forward sweep and $t^* \in [s_{j+1}, s_j]$ in the backward sweep, for some values of i and j. Then at t^* the forward transition equation is

$\Phi_1^i(t^*)\, x^i(t^*) = \phi_1^i(t^*) \Longrightarrow$

$\Phi_1^i(t^*)\, P\, x(t^*) = \phi_1^i(t^*)$, and the backward equation

$\Phi_2^j(t^*)\, x^j(t^*) = \phi_2^j(t^*) \Longrightarrow$

$\Phi_2^j(t^*)\, Q\, x(t^*) = \phi_2^j(t^*)$. Then the combined transition equation at $t = t^*$ is

$$\begin{bmatrix} \Phi_1^i(t^*)\, P \\ \Phi_2^j(t^*)\, Q \end{bmatrix} . \; x(t^*) = \begin{bmatrix} \phi_1^i(t^*) \\ \phi_2^j(t^*) \end{bmatrix}$$

from which solution $x(t^*)$ can be obtained.

Application to continuous orthonormalisation

We now consider the application of the factorisation method to continuous orthonormalisation. For the forward sweep we must first re-write the initial BC of the given LBVP in an equivalent suitable form, as follows. By using the Gram Schmidt process we find matrix $P(n_1 , n_1)$ such that $P[K_0 \mid K_1] = V_1$ is a row set of unit orthogonal vectors i.e. $V_1 V_1^T = I_{n_1}$. The given initial BC of (5.1) is :

$$[K_0 \mid K_1]\, x(a) = c_1 \qquad (5.23)$$

$$\Longrightarrow \quad P\,[K_0 \mid K_1]\,x(a) = P\,c_1$$

$$\Longrightarrow \quad V_1\,x(a) = d_1 \qquad (5.24)$$

where $d_1 = P\,c_1$. (5.24) is now a set of initial conditions equivalent to (5.23). For the forward sweep we use the transformation $x(t) = T(t)y(t)$ or $y(t) = S(t)x(t)$ on ODE $\dot{x}(t) = B(t)x(t) - f(t)$ to obtain the transformed system $\dot{y}(t) = \tilde{B}(t)y(t) - \tilde{f}(t)$ where $T(t) = [\hat{T}_1(t) \mid \hat{T}_2(t)]$ must be orthonormal for all t and such that $\tilde{B}(t)$ is block lower triangular. In [5-2] we show that this will be so if $T_1(t)$ satisfies the ODE : $\dot{T}_1(t) = -B^T(t)T_1(t) + T_1(t)T_1^T(t)B^T(t)T_1(t)$ or $\dot{T}_1^T(t) = -T_1^T(t)B(t) + \{T_1^T(t)B(t)T_1(t)\}T_1^T(t)$ (5.25).

(Note that since only one part of the transformation is to be used in the forward sweep we need not concern ourselves with the ODE for $T_2(t)$). In this case, $\tilde{B}(t)$ is given by :

$$\tilde{B}(t) = \begin{bmatrix} T_1^T(t)B(t)T_1(t) & 0 \\ T_2^T(t)\{B(t) + B^T(t)\}T_1(t) & T_2^T(t)B(t)T_2(t) \end{bmatrix}.$$

Now since $T(t)$ is orthonormal :

$$S(t) = T^{-1}(t) = T^T(t) = [T_1(t) \mid T_2(t)]^T = \begin{bmatrix} T_1^T(t) \\ T_2^T(t) \end{bmatrix}.$$

Therefore the forward transition matrix $\Phi_1(t) = T_1^T(t)$ and the condition matrix $Z_1(t) = \tilde{B}_{11}(t) = T_1^T(t)B(t)T_1(t)$.

Note that if we substitute these expressions into (5.25) it becomes : $\dot{\Phi}_1(t) = -\Phi_1(t)B(t) + Z_1(t)\,.\,\Phi_1(t)$, the forward transition matrix ODE. The initial conditions for the $\Phi_1(t)$ and $\varnothing_1(t)$ ODEs are $\Phi_1(a) = V_1$ and

$\Phi_1(a) = d_1$ from (5.24). Note also that $V_1 T_1(a) = V_1 \Phi_1^T(a) = V_1 V_1^T = I_{n_1}$, as required in (5.7a).

For the backward sweep, we first obtain a set of conditions equivalent to the final BC of LBVP (5.1) viz.

$$[K_2 \mid K_3]\, x(b) = c_2 \qquad (5.26).$$

We use the Gram Schmidt process to find matrix $Q(n_2, n_2)$ such that $Q[K_2 \mid K_3] = V_2$ is a row set of unit orthogonal vectors. Then from (5.26) we get

$$V_2 x(b) = d_2 \qquad (5.27)$$

where $d_2 = Qc_2$. We now use the transformation $x(t) = L(t)r(t)$ or $r(t) = J(t)x(t)$ on ODE $\dot{x}(t) = B(t)x(t) - f(t)$ to obtain the transformed system $\dot{r}(t) = C(t)r(t) - h(t)$ where $L(t) = [L_1(t) \mid L_2(t)]$ is orthonormal for all t and such that $C(t)$ is block upper triangular for all t. For this to be so (see [5-3]) $L_2(t)$ must satisfy the ODE :

$$\dot{L}_2(t) = -B^T(t)L_2(t) + L_2(t)L_2^T(t)B^T(t)L_2(t) \quad ====>$$

$$\dot{L}_2^T(t) = -L_2^T(t)B(t) + \{L_2^T(t)B(t)L_2(t)\}L_2^T(t) \qquad (5.28).$$

(We will not require the ODE for $L_1(t)$ for this sweep).

In this case, $C(t)$ is given by :

$$C(t) = \begin{bmatrix} L_1^T(t)B(t)L_1(t) & L_1^T(t)\{B(t) + B^T(t)\}L_2(t) \\ 0 & L_2^T(t)B(t)L_2(t) \end{bmatrix}.$$

Now since $L(t)$ is orthonormal :

$$J(t) = L^{-1}(t) = L^T(t) = [L_1(t) \mid L_2(t)]^T = \begin{bmatrix} L_1^T(t) \\ L_2^T(t) \end{bmatrix}$$

and so $\Phi_2(t) = L_2^T(t)$ and $Z_2(t) = C_{22}(t) = L_2^T(t)B(t)L_2(t)$.

Substitution of these expressions into (5.28) gives us the ODE

for $\Phi_2(t)$ viz. $\dot{\Phi}_2(t) = -\Phi_2(t)B(t) + Z_2(t).\Phi_2(t)$.
The initial conditions for the $\Phi_2(t)$ and $\phi_2(t)$ IVPs are $\Phi_2(b) = V_2$, $\phi_2(b) = d_2$ from (5.27). Note that $V_2 L_2(b) = V_2 \Phi_2^T(b) = V_2 V_2^T = I_{n_2}$.

Use of 'generalised inverses'

A practical difficulty associated with continuous orthonormalisation methods has already been discussed in Chapter 3 viz. although the solutions $T_1(t)$ and $L_2(t)$ of ODEs (5.25) and (5.28) respectively should each in theory be a unit orthogonal column set for all t, in practice this may not always be so. Therefore, to reduce the risk of loss of orthogonality it is suggested in [16] that the following alternative forms of the conditioning matrices $Z_\alpha(t)$ $(1 <= \alpha <= 2)$ should be used instead of $Z_\alpha(t) = \dot{\Phi}_\alpha(t)B(t).\Phi_\alpha^T(t)$ (5.29) in ODEs (5.25) and (5.28) :

$$Z_\alpha(t) = \{\dot{\Phi}_\alpha(t)B(t).\Phi_\alpha^T(t)\}\{\Phi_\alpha(t).\Phi_\alpha^T(t)\}^{-1} \quad (5.30)$$

$$Z_\alpha(t) = \{\dot{\Phi}_\alpha(t)B(t).\Phi_\alpha^T(t) + Q_\alpha(t)\}\{\Phi_\alpha(t).\Phi_\alpha^T(t)\}^{-1} \quad (5.31)$$

where $Q_\alpha(t) = S_\alpha\{U_\alpha U_\alpha^T - \Phi_\alpha(t).\Phi_\alpha^T(t)\}$, $S_1 > 0$ and $S_2 < 0$. Version (5.30) corresponds to the 'generalised inverse' method of Chapter 3 (see 3.31a & b) whilst version (5.31) is usually referred to as continuous stabilised orthonormalisation. Note that if we assume that $\Phi_\alpha(t).\Phi_\alpha^T(t) = I_{n_\alpha}$ for all t then (5.30) and (5.31) both reduce to the basic orthogonal version (5.29).

Error control by factorisation methods

The main advantage claimed for the approach of Babuska's factorisation method over the Riccati and orthonormalisation methods described in previous Chapters, is that it provides us with a means of error control i.e. in the case of a well conditioned LBVP the errors in the computed values of the transition vectors $\phi_\alpha(t)$ $(1 <= \alpha <= 2)$ as obtained from ODEs (5.5) and (5.8) should provide a meaningful bound for the error vector $e(t)$ in the computed solution $x(t)$ of the LBVP, so long as the transition matrices $\Phi_\alpha(t)$ $(1 <= \alpha <= 2)$ remain sufficiently small over $[a,b]$.
To understand this we must regard the computed solution of a given IVP (LBVP) as the exact solution of a corresponding perturbed IVP (LBVP). Thus the computed solutions of the transition vector equations (5.5) and (5.8) can be regarded as the exact solutions of the perturbed IVPs :

$$\dot{\phi}_\alpha(t) = -\Phi_\alpha(t)f(t) + Z_\alpha(t)\,.\,\phi_\alpha(t) + \delta_\alpha(t)$$

$\phi_\alpha(t_\alpha) = c_\alpha + v_\alpha$, where $t_1 = a$, $t_2 = b$, and $\delta_\alpha(t)$ and v_α are the perturbations in the right hand sides. Likewise, the computed solution of the LBVP (5.1) can be regarded as the exact solution of the perturbed LBVP :

$\dot{x}(t) = B(t)x(t) - f(t) + r(t)$

$U_\alpha x(t_\alpha) = c_\alpha + w_\alpha$, where $r(t)$ and w_α are the perturbations in the right hand sides. Now suppose that for all $t \in [a,b]$ the transition matrices $\Phi_\alpha(t)$ satisfy the

following boundedness conditions :

$$\| \Phi_\alpha(t) \| \leq M_\alpha \quad \text{and} \quad \| [\Phi_\alpha(t) . \Phi_\alpha^T(t)]^{-1} \| \leq \frac{1}{m_\alpha}$$

$(1 \leq \alpha \leq 2)$, where M_α and m_α are constants of moderate size and $\frac{M_\alpha}{m_\alpha} = O(1)$. Then it is shown in [16] that at any value of $t \in [a,b]$:

$$\| r(t) \| \leq \max_{\alpha = 1,2} \left\{ \frac{M_\alpha}{m_\alpha} . \| \delta_\alpha(t) \| \right\}$$

and $\| w_\alpha \| \leq \| v_\alpha \|$ i.e. the perturbations in the LBVP are bounded above by the corresponding perturbations in the transition vector IVPs (5.5) and (5.8). Therefore, if these IVPs are solved using a variable step Runge Kutta integrator with a small error tolerance this will ensure that $\| \delta_\alpha(t) \|$ and $\| v_\alpha \|$ will both be small and hence that $\| r(t) \|$ and $\| w_\alpha \|$ will be small also. Further, if the given LBVP (5.1) is assumed to be well conditioned this means (see 1.13 of Chapter 1) that the error $\| e(t) \|$ in the computed solution $x(t)$ of the LBVP should also be small.

Thus Babuska's bounded factorisation methods provide us with a means of controlling the size of $\| e(t) \|$ .In Chapter 6, we describe a proposed error estimation method (based upon multiple shooting) from which we can actually obtain an estimate for the error vector $e(t)$, and this is applicable even when the given LBVP is not well conditioned. We also extend this method into an iterative correction algorithm which can be used to solve the LBVP.

In Appendix III we give the results of some of our numerical experience with the factorisation methods in the solution of both well and ill conditioned LBVPs.

In Appendix I :

[5-1]: Derivation of Riccati equation.

[5-2]: Derivation of $\dot{T}_1(t)$ equation for forward sweep of orthonormal method.

[5-3]: Derivation of $\dot{L}_2(t)$ equation for backward sweep of orthonormal method.

CHAPTER 6

ERROR ESTIMATION AND ITERATIVE IMPROVEMENT METHODS

Introduction

In this Chapter we describe an error estimation method based upon multiple (parallel) shooting which we extend into an iterative correction algorithm to converge to an improved solution of the given LBVP i.e. to produce successive improvements on the first calculated solution. Results of some of our numerical experience are included to show the success that we achieved with the method. However, the situation is complex and the results are not completely conclusive so that further investigation and research is needed.

As explained in Chapter 5, Babuska's bounded factorisation methods enable us to control the size of the computed error in the solution $x(t)$ to a well conditioned LBVP. But in practice the condition of a given LBVP will most likely be unknown and the calculation of the conditioning constants k_1 and k_2 (see 1.11) is costly. Even if these are found we can still only obtain a bound on the size of the computed error (see 1.13) and this bound could be very pessimistic. Therefore, below we propose a method of estimation of the computed error which is obtained as the LBVP is solved rather than an estimate for the error bound. Moreover (as our results show) the method

can be successfully applied to LBVPs which are quite stiff and ill conditioned.

Suppose the given (n,n) LBVP is :

$$\dot{x}(t) = A(t)x(t) + f(t)$$

$$B_0 x(a) + B_1 x(b) = c \qquad (6.1)$$

for which the exact solution is $x(t)$ for all $t \in [a,b]$.

The method may be summarised as follows. First we find an approximate solution of LBVP (6.1) and then we use an interpolant of this calculated solution $u(t)$ to obtain a residual function $r(t)$ at any required value of $t \in [a,b]$. This enables us to re-solve LBVP (6.1) with the forcing function $f(t)$ now replaced by $r(t)$ and with $c = 0$. The exact solution of this LBVP will be the actual error $e(t)$ in $u(t)$. Our calculated solution though will be subject to the combined effects of interpolation and integration error but we hope that this approximate solution will provide a good estimate for the actual error.

More precisely, we proceed as follows. First we solve LBVP (6.1) by the standard parallel shooting method (as described in Chapter 2). In doing so we use either a pre-selected number of equally spaced nodes or node positions determined by a pre-selected maximum value (cmax) of the condition number of the fundamental solution $X_i(t)$ in each subinterval : the value of cond $X_i(t) = \| X_i(t) \| . \| X_i^{-1}(t) \|$ is checked at the end of

each integration step and a node t_{i+1} is inserted as soon as cond $X_i(t) >= cmax$. The calculated solution vectors $u_1(t_i)$ obtained at these nodes $a = t_0 < t_1 < \ldots\ldots < t_N = b$, are stored.

These $u_1(t_i)$ vectors can now be substituted in the given ODE of LBVP (6.1) to obtain derivative values :

$\dot{u}_1(t_i) = A(t_i)u_1(t_i) + f(t_i)$ and hence also

$\ddot{u}_1(t_i) = A(t_i)\dot{u}_1(t_i) + \dot{A}(t_i)u_1(t_i) + \dot{f}(t_i)$ and

$\dddot{u}_1(t_i) = A(t_i)\ddot{u}_1(t_i) + \ddot{A}(t_i)u_1(t_i) + 2\dot{A}(t_i)\dot{u}_1(t_i) + \ddot{f}(t_i)$.

(If the expressions for the derivatives of A are not readily obtained then it may be necessary to make use of computer algebra software written for this purpose).

Thus we have values for $u_1(t_i)$ and its derivatives at each node t_i and so we can use Hermite (cubic, quintic or septenary) interpolation between each pair of nodes $[t_i, t_{i+1}]$ $(0 <= i <= N - 1)$ to obtain an interpolated approximate solution $u_1(t)$ for all $t \in [a,b]$. Now this solution can be regarded as the exact solution of a perturbed LBVP :

$$\dot{u}_1(t) = A(t)u_1(t) + f(t) + r_1(t)$$

$$B_0 u_1(a) + B_1 u_1(b) = c \qquad (6.2)$$

for some residual function $r_1(t)$ (n,1) given by :

$$r_1(t) = \dot{u}_1(t) - A(t)u_1(t) - f(t) \qquad (6.3)$$

for all $t \in [a,b]$. Note that $r_1(t_i) = 0$ at each node t_i.

Now the error $e_1(t)$ in this solution $u_1(t)$ at any value $t \in [a,b]$ is given by $e_1(t) = u_1(t) - x(t)$. From LBVPs (6.1) and (6.2) we see that $e_1(t)$ is the solution of the LBVP :

$$\dot{e}_1(t) = A(t)e_1(t) + r_1(t)$$

$$B_0 e_1(a) + B_1 e_1(b) = 0 \qquad (6.4).$$

LBVP (6.4) is now solved by the same multiple shooting algorithm used previously to solve LBVP (6.1) i.e. the subinterval fundamental solutions $X_i(t)$ $(0 <= i <= N - 1)$ will be the same as before but the particular solutions $v_i(t)$ must be re-calculated because the forcing function is now $r_1(t)$ instead of $f(t)$. The value of $r_1(t)$ is obtained from (6.3) at any required value of $t = t^*$ in subinterval $[t_i, t_{i+1}]$ by using Hermite interpolation between the nodes t_i and t_{i+1} to find values for $u_1(t^*)$ and $\dot{u}_1(t^*)$. The solution $e_1(t_i)$ is saved at each node t_i and provides us with an estimate of the error in the calculated solution $u_1(t_i)$ of the given LBVP (6.1). The error in $e_1(t_i)$ is a combination of interpolation error and integration error, where these two are interdependent. This complexity makes analysis difficult. The error estimation method described above can be further extended into an iterative correction algorithm as follows.

If the error estimate $e_1(t_i)$ is sufficiently good then a better approximation $u_2(t_i)$ to the solution of LBVP (6.1) should now be given by : $u_2(t_i) = u_1(t_i) - e_1(t_i)$ at each node t_i $(0 <= i <= N)$. We can now repeat the above error estimation procedure this time using the $u_2(t_i)$ values to obtain an interpolated solution $u_2(t)$ for any $t \in [a,b]$ and hence a residual function $r_2(t)$ from which an error estimate

$e_2(t_i)$ in $u_2(t_i)$ can be obtained. We hope that the iteration : $u_{j+1}(t_i) = u_j(t_i) - e_j(t_i)$ $(j >= 1)$ will produce successive solution vectors $u_j(t_i)$ which are improvements on the first calculated solution $u_1(t_i)$ of LBVP (6.1) at each node t_i.

In practice, as we shall see, we found that the success of this proposed iterative residual correction method depended very much on the choice of type of integrator used to solve the IVPs necessary for the solution of the original LBVP (6.1) and the residual associated LBVPs (6.4). Equally important was the overall accuracy of the interpolant used.

In obtaining the numerical results given in this section all calculations were performed in double precision and the multiple shooting node positions were determined by pre-selecting a value (cmax) for the maximum allowable size of the condition number of the fundamental solutions as explained earlier. Note that the number (ns) of subintervals is reduced as cmax is increased. For the iterative correction method, in each case the maximum number of iterations allowed was six and the accuracy of the final calculated iterative solution u(t) to the LBVP was measured by the maximum actual absolute error incurred in the components of $[u(\alpha), u(\beta)]^T$ where $[\alpha, \beta]$ is the problem interval.

The test problems were chosen to illustrate the behaviour of the iterative correction

algorithm in solving LBVPs differing widely in 'stiffness' and in condition. Each of the problems given below is in effect a family of LBVPs with a common known exact solution from which the actual errors in the calculated iterative solutions were obtained.

Error estimation method

At the outset it was our intention only to use one iteration (i.e. one solution of the residual LBVP (6.4)) to obtain an estimate of the error in the first calculated solution. However, we found the agreement between actual and estimated errors to be often much better than we had expected. Some of our numerical results for the single iteration estimation method are given in Appendix II, where the integrator used was a variable step Runge Kutta (RKF 45). The results for test problem II (detailed later) were particularly accurate as can be seen from tables A2.2.1 to A2.2.5 in Appendix II : nearly all of the estimated and actual errors agree to at least two significant figures here. The results obtained for the other two test problems were not quite as good as this - though for the well conditioned BC cases of problem III the errors obtained agreed in most cases to the same order of magnitude (see tables A2.3.1 to A2.3.3). As might be expected agreement between actual and corresponding estimated errors deteriorates as the given LBVP becomes very stiff (see tables A2.2.6 to

A2.2.8) or very ill conditioned (see tables A2.1.11 and A2.1.12 and A2.3.4 to A2.3.6).

The success that we had with this error estimation method on moderately difficult problems motivated us to investigate the use of further iterations to improve on the first calculated solution (as explained earlier) :

Iterative improvement method

Initially we tested this method using the variable step Runge Kutta integrator (RKF 45) to solve all the auxiliary IVPs of LBVPs (6.1) and (6.4) and Hermite cubic interpolation was used between the nodes, but the results obtained were not encouraging. In some cases successive iterative calculated solutions were improvements but we found that this was not generally so. Replacing the cubic Hermite interpolation by a quintic or septenary produced similar unreliable results. We found that the cause of this was the Runge error estimation criterion used to vary the steplength : in many cases it was occasionally allowing through very large steplengths and this we attributed to the fact that, in general, the norm of the solutions to the residual IVPs in (6.4) was much less than the norm of the first solution (6.1). In other words, the reason for failure of the method was that we were using the the same Runge Kutta tolerance to integrate forwards both the original IVP of (6.1) and also the residual IVPs of (6.4).

We give below some of our results obtained using integrator RKF45 with the third, fifth and seventh degree Hermite interpolation (programs ITVAR.3, ITVAR.5 and ITVAR.7 respectively). These show that the success of the method depends on the degree of Hermite interpolation used, the initial steplength of integration and Runge Kutta tolerance and also on the problem itself. We found that the large number of variable factors involved (each contributing to the final error) made analysis difficult. It did seem clear though that, for the residual integrations, the Runge Kutta step adjustment criterion should in some way be related to the size of the solution.

We therefore decided to investigate the effect of using a different Runge Kutta integration tolerance (μ) for the residual IVPs in (6.4) from that used for the first solution (6.1). Denoting the latter by ϵ we found that in general improvement was obtained but only for sufficiently small value of the ratio μ/ϵ, this being dependent on both ϵ and on the problem. This was true whether we used the cubic, quintic or septenary interpolation.

We replaced the RKF 45 integrator by another variable step adjusting Runge Kutta system but with the same result. We therefore concluded that our iterative residual correction method was not reliable in practice if used with a variable step integrator as there was no obvious way of determining how small the ratio μ/ϵ must be to ensure

convergence in any particular case. Further research in this area is needed to find a step adjustment criterion which will automatically take account of this.

We now modified our RKF 45 variable step integrator to convert it into a fixed step integrator. This we did by retaining the Runge Kutta system equations but eliminating the error estimate criterion by which the step length was either halved or doubled. We also now included the option of being able to choose a different fixed step length (h_2) for the residual integrations in (6.4) from the step length (h_1) used to obtain the first solution in (6.1). As the results given below show our iterative correction algorithm was now found to be much more reliable - but still not completely so. Improvement on the first solution was obtained in nearly every case, the amount of improvement being generally (but not always) increased as the degree of the Hermite interpolation was increased from three to five to seven. However, we did find cases where (particularly with ITER.3 with large steplengths) the iterative solutions computed did not show improvement. In fact the difficulty lay in knowing in any given case for which Hermite (cubic, quintic or septenary) the norm of the residual would be least overall as this would be most likely to provide the closest approximation to the exact solution.

(In the results below ITER.3 refers to the fixed step program employing the Hermite cubic interpolation while ITER.5 and

ITER.7 used the quintic and septenary respectively).

In an attempt to find an algorithm which we could propose as practically reliable we therefore combined our fixed step programs ITER.3, ITER.5 and ITER.7 so that whenever the value of a residual $r_i(t)$ is required at a time $t = t^*$ this value is separately calculated from (6.3) using respectively the Hermite cubic, quintic and septenary. From these three residual vectors we then choose the one having the least norm and this $r_i(t)$ is then used to integrate forward the IVP in (6.4). In other words the interpolation used is now analytically discontinuous but provides at each evaluation the closest approximation to the exact solution. This program we called ITMIN.357. (The residual norm used to obtain the results given below was the l_2 norm. We did compare results for some cases with those obtained using the l_∞ norm instead but we found no significant differences. However, it is possible that the choice of norm could, for some problems, have a measurable effect on the rate of improvement of the iterates).

As can be seen from the results below ITMIN.357 was successful in all the test cases in producing iterates which improved on the first solution to the LBVP though the rate of improvement was slow for large steplengths such as $h = 0.1$.

(See tables 1.1, 1.2, 1.4, 1.5, 1.6, 1.7, 1.8, 2.1, 2.2, 2.3, 2.4, 2.5, 3.2, 3.3, 3.4, 3.5, 3.6).

Also it can be

seen that in many cases more rapid improvement can be obtained by using ITER.7 or ITER.5 instead of ITMIN.357 but as stated earlier the former cannot always be relied upon to produce improvement. (ITMIN.357 was also tested on several other LBVPs not detailed below and we found it to be successful in every case using fixed steplengths ranging from $h = 0.1$ to $h = 0.005$). Program running times are obviously longer for ITMIN.357 than for ITER.3 or ITER.5 or ITER.7 but we found that when using ITMIN.357 running time can be much reduced by using a much larger steplength (h_2) for the residual integrations of (6.4) than the steplength used for the first solution (6.1) e.g. $h_1 = 0.01$ and $h_2 = 0.05$. As the results show , in many cases, this does not reduce the rate of improvement - indeed, in some cases, the solution improves more rapidly than when h_2 is taken to be smaller (with the same h_1).

We may also note (surprisingly) that in the problems tested the effectiveness of ITMIN.357 seems to be little affected by the poor condition of the given LBVP but only by the stiffness of its ODE. Also although running time may be long with ITMIN.357 it is quite economical as regards storage because the value of the residual $r_i(t)$ is calculated as and when required for the forward integration of the particular solution of system $\dot{e}_i(t) = A(t)e_i(t) + r_i(t)$ in each subinterval.

Perhaps the main drawback of ITMIN.357 is that common to all multiple shooting methods viz. the large number of subintervals likely to be required for the solution of a stiff LBVP which in turn necessitates the solution of a very large system of linear equations. Developments in the application of 'block diagonal' methods of solving linear systems could be used to overcome this difficulty. Alternatively, large linear systems can be avoided by instead using only a few multiple shooting sub-intervals and allowing more iterations for improvement - but at the expense of increased program running time.

Another disadvantage of ITMIN.357 is that it must be supplied with not only the system matrix A(t) and vector f(t) of the given LBVP but also their first and second derivatives. However, computer algebra software now available could be used to facilitate this.

Test problems and results for iterative residual correction method :

In the following

h = fixed integration steplength used if this is the same for the first solution and for iterative solutions - if not then h_1 = steplength for the first solution and h_2 = steplength for iterative solutions

err1 = maximum modulus actual error in first solution

errf = maximum modulus actual error in final iterative solution after a maximum of six iterations

(err1 and errf are both taken over the components of $[u(\alpha), u(\beta)]^T$ where $[\alpha, \beta]$ is the problem interval and u(t) is the calculated solution)

cmax = maximum allowed value of fundamental solution norm. (This determines the node positions and the number of multiple shooting subintervals (ns)).

All numerical results have been given to an accuracy of two significant figures as this is sufficient to show whether improvement has been obtained and the order of the size of the actual error in the calculated solution.

Test problem (I) :

This is a (3,3) LBVP for which :

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -j^2 k & j^2 & k \end{bmatrix}$$

$$f(t) = [0,\ 0,\ (1 + j^2 k - j^2 - k)e^t]^T$$

$$B_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{(I.1)}$$

where the problem interval is [0,1] and j and k are constant parameters. In fact, the eigen-values of system matrix A are k, $\pm j$ so that for large (positive) values of j and k the problem becomes very stiff and also (with BC (I.1))

ill conditioned e.g. for the case j = 20, k = 30 the LBVP has a condition number (see (1.11a) of Chapter 1) of approximately 1e10. The exact solution of the LBVP is $x(t) = [e^t, e^t, e^t]^T$, for all values of parameters j and k, so specifying c.

For comparison we also give results below for the same ODE but with the following BC with which the LBVP is well conditioned :

$$B_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad B_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \text{(I.2)}$$

Results :

(In this problem I section a reference such as table 3 is to table 1.3).

Tables 1 to 8 contain results obtained using fixed step integrations (programs ITER.3, ITER.5, ITER.7 and ITMIN.357). Tables 9 and 10 show some of our results obtained using a variable step integrator (RKF 45).

In tables 1 to 5 the BC were ill conditioned (I.1) and parameters j and k were as given :

Table 1.1 : j = 2, k = 3, cmax = 10, ns = 5 :

h	err1	errf : ITMIN.357
0.1	2.4e-6	1.9e-7
0.05	2.1e-7	2.1e-7
0.02	6.3e-9	6.3e-9
0.01	4.2e-10	4.2e-10

Table 1.2 : j = 5, k = 10, cmax = 100, ns = 10 :

h	err1	errf : ITMIN.357
0.1	5.7e-4	2.5e-8
0.05	7.6e-6	7.7e-8
0.02	7.0e-7	2.6e-7
0.0125	1.4e-7	1.4e-7
0.01	6.5e-8	6.5e-8

We see from tables 1 and 2 that for these easier problems for which the first solution is already very accurate there is no improvement in most cases.

Table 1.3 : j = 20, k = 30, cmax = 1e7, ns = 10 :

h	err1	errf		
		ITER.3	ITER.5	ITER.7
0.1	3.8e4	2.9e1	4.4e-1	2.2e0
0.05	4.8e3	1.0e0	1.4e-3	6.5e-6
0.02	2.6e1	3.0e-3	2.7e-6	5.0e-8
0.0125	5.2e0	1.1e-3	5.8e-7	3.4e-8
0.01	3.6e0	6.1e-4	3.1e-7	7.4e-8
0.005	4.4e-1	5.0e-5	7.0e-8	8.2e-8

We see that here improvement occurred in all cases and this was most rapid with ITER.7 except for h = 0.1.

Table 1.4 : j = 20, k = 30, cmax = 1e7, ns = 10 :

h	err1	errf : ITMIN.357
0.1	3.8e4	2.1e2
0.05	4.8e3	7.6e-1
0.02	2.6e1	4.0e-3
0.0125	5.2e0	2.6e-2
0.01	3.6e0	1.9e-3
0.005	4.4e-1	1.5e-5

We see from table 4 that for this difficult problem program ITMIN.357 produces an improved solution in all cases though the amount of improvement is not as much as that obtained by using ITER.5 or ITER.7 (see table 3). But as we said earlier the latter may not always be reliable (see test problem III: table 3.1 for an example of where ITER.3 fails).

Table 1.5 : j = 15, K = 20, cmax = 1e5, ns = 10 :

h	err1	errf : ITMIN.357
0.1	1.3e2	1.6e0
0.05	7.5e0	1.5e-5
0.02	3.2e-2	6.6e-5
0.0125	1.9e-2	1.5e-8
0.01	1.0e-2	6.0e-9

Again we see that ITMIN.357 produces improvement for all values of h and that errf is acceptably small except for h = 0.1 and even in this case we obtained an accurate final solution by allowing more iterations (e.g. errf = 1.3e-3 after ten

iterations).

For comparison, tables 6 and 7 give results obtained using BC (I.2) for which the LBVP is well conditioned :

Table 1.6 : j = 15, k = 20, cmax = 1e5, ns = 10 :

h	err1	errf : ITMIN.357
0.1	3.5e-3	5.0e-6
0.05	1.8e-5	1.2e-8
0.02	8.8e-6	7.4e-8
0.0125	1.8e-6	1.6e-7
0.01	8.2e-7	7.7e-8

Table 1.7 : j = 20, k = 30, cmax = 1e7, ns = 10 :

h	err1	errf :ITMIN.357
0.1	1.5e-2	1.0e-4
0.05	5.9e-4	1.1e-7
0.02	2.1e-5	1.9e-7
0.0125	5.6e-6	5.3e-7
0.01	2.7e-6	2.6e-7

Comparison of tables 6 and 7 with corresponding tables 5 and 4 respectively shows that although the condition of the given LBVP has a marked effect on the accuracy of the first solution (as expected) it surprisingly has no significant effect on the rate of improvement with ITMIN.357 : the errf values are very similar in tables 5 and 6 though in table 4 a few more iterations were required to obtain errf values comparable to

those in table 7 e.g. with h = 0.05 in table 4 errf = 1.6e-4 after twelve iterations reducing to 3.5e-7 after sixteen.

Table 8 shows results obtained with ITMIN.357 using a larger steplength $h_2 = 0.05$ for the residual integrations of (6.4) than that ($h_1 = 0.01$) for the first solution (6.1).

Table 1.8 : Ill conditioned BC (I.1) :

	err1	errf :ITMIN.357
j = 20, k = 30 :	3.6e0	1.1e-4
j = 15, k = 20 :	1.0e-2	6.6e-8

Note that for the case j = 20, k = 30 the value of errf obtained here is actually smaller than that obtained with $h_1 = h_2 = 0.01$ (1.9e-3 from table 4) and running time was considerably reduced.

Table 9 shows results obtained using variable step integrator RKF 45 : programs ITVAR.3, ITVAR.5 and ITVAR.7 employing the Hermite cubic, quintic and septenary interpolation respectively. RK and h_0 denote the Runge Kutta tolerance and initial steplength used respectively :

Table 1.9 : j = 20, k = 30, cmax = 1e7, ns = 10, BC (I.1) :

RK	h_o	err1	errf		
			ITVAR.3	ITVAR.5	ITVAR.7
1e-1	1e-1	9.3e-1	2.9e1	2.1e-2	3.3e-4
1e-1	1e-2	4.8e-1	3.7e-1	6.1e-5	6.7e-8
1e-1	1e-3	1.8e-1	2.7e-1	2.3e-5	6.3e-8
1e-2	1e-1	8.3e-2	3.0e1	2.1e-2	2.7e-5
1e-2	1e-2	3.7e-2	3.7e-1	6.1e-5	5.9e-8
1e-2	1e-3	1.4e-2	2.7e-1	2.3e-5	3.5e-8
1e-3	1e-1	6.1e-3	3.0e1	2.1e-2	4.4e-7
1e-3	1e-2	2.5e-3	3.7e-1	6.1e-5	9.1e-8
1e-3	1e-3	1.2e-2	2.4e-1	1.5e-5	1.0e-7
1e-4	1e-1	4.1e-4	1.0e0	2.1e-2	1.2e-6
1e-4	1e-2	1.3e-3	3.7e-1	6.1e-5	9.1e-8
1e-4	1e-3	1.0e-3	1.5e-1	1.5e-5	1.5e-7
1e-5	1e-1	2.4e-5	2.5e-2	2.1e-2	1.6e-6
1e-5	1e-2	1.4e-4	5.7e-4	6.1e-5	5.7e-8
1e-5	1e-3	7.9e-5	1.5e-1	1.3e-5	9.4e-8

We see that for this problem the effectiveness of the method improved with the degree of Hermite interpolation used : with ITVAR.7 there was improvement of the solution in every case, with ITVAR.5 there was improvement in most cases (but not all) but with ITVAR.3 improvement occurred only for

the case RK = 1e-1, h_o = 1e-2.

Table 10 shows results obtained using a modification of ITVAR.3 in which the Runge Kutta tolerance (μ) used for the residual integrations is in every case ϵ/1e7, where ϵ is the tolerance used in the integration of the first solution. The initial steplength h_o is 0.1 in each case, and the LBVP solved is the same as that in table 9 - but with very different results :

Table 1.10 : j = 20, k = 30, cmax = 1e7, ns = 10, h_o = 0.1 :

RK = ϵ	err1	errf
1e-1	9.3e-1	1.3e-4
1e-2	8.3e-2	1.6e-5
1e-3	6.1e-3	6.5e-7
1e-4	4.1e-4	6.9e-8
1e-5	2.4e-5	1.7e-8

We see that for sufficiently small value of μ (the residual integrations tolerance) improvement is obtained in every case and the final solutions are all acceptably accurate. For the corresponding cases in table 9 (where the same tolerance was used for the residual integrations as for the first) the method failed in every case. However, without a criterion for determining how small the ratio μ/ϵ must be to ensure improvement in any given case this is of little practical use. Further investigation and research to provide a better theoretical understanding of the method might well suggest a

reliable criterion.

Test problem (II) :

The following (4,4) LBVP is taken from Conte's paper [19] :

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k^2 & 0 & k^2 + 1 & 0 \end{bmatrix}$$

$$f(t) = [0,\ 0,\ 0,\ \frac{k^2 t^2}{2} - 1]^T$$

$$B_0 = \begin{bmatrix} 1 & 3 & 17 & -21 \\ 5 & -2 & 1 & -4 \\ 3 & 6 & -8 & -1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad B_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 8 & 6 & 4 & 2 \end{bmatrix} .$$

The problem interval is [0,1] and the exact solution is $x(t) = [1 + \frac{t^2}{2} + sh(t),\ t + ch(t),\ 1 + sh(t),\ ch(t)\ ]^T$ for all values of the constant parameter k. System matrix A has eigenvalues $\pm 1, \pm k$ so that as k increases the problem becomes more stiff and more ill conditioned (with the above BC).

Results :

(In this problem II section a reference such as table 3 is to table 2.3).

All of the following results were obtained using program ITMIN.357 with the k parameter values given :

Table 2.1 : k = 3, cmax = 100, ns = 3 :

h	err1	errf
0.1	1.9e-5	7.5e-8
0.05	1.5e-6	7.6e-8
0.02	4.3e-8	4.3e-8
0.0125	6.7e-9	6.7e-9
0.01	2.8e-9	2.8e-9

Table 2.2 : k = 10, cmax = 1e3, ns = 5 :

h	err1	errf
0.1	1.8e-4	8.3e-8
0.05	2.8e-2	3.1e-7
0.02	1.4e-3	5.6e-7
0.0125	2.4e-4	6.0e-8
0.01	1.0e-4	5.0e-7

Table 2.3 : k = 15, cmax = 1e5, ns = 5 :

h	err1	errf
0.1	1.2e2	9.1e-2
0.05	6.2e0	7.3e-4
0.02	5.7e-1	5.0e-6
0.0125	1.1e-1	9.1e-8
0.01	4.6e-2	1.4e-8

Table 2.4 : k = 20, cmax = 1e6, ns = 5 :

h	err1	errf
0.1	5.9e4	4.6e2
0.05	8.6e-5 *	3.4e-6
0.02	1.6e2	1.2e-1
0.0125	3.4e1	8.6e-3
0.01	1.5e1	4.3e-4
0.005	1.1e0	8.3e-7

We see from the above that ITMIN.357 produced significant improvement on the first solution in all cases for values of $k <= 15$. For $k = 20$ the LBVP is very stiff and, although improvement is still obtained for all h, in some cases more iterations are required to produce an acceptably accurate solution e.g. for $h = 0.1$ errf = 3.8e-4 after twenty iterations.

(* This unexpected result we attribute to a peculiarity of the problem.)

The following results were obtained with ITMIN.357 using steplengths of $h_1 = 0.01$ (for the first solution) and $h_2 = 0.05$ (for the residual solutions) :

Table 2.5 :

	err1	errf
k = 15, cmax = 1e5 :	4.6e-2	1.1e-8
k = 20, cmax = 1e6 :	1.5e1	1.6e-3

Note that for the k = 15 case the value of errf is virtually the same as that obtained using $h_1 = h_2 = 0.01$ (as in table 3).

Test problem (III) :

The following (3,3) variable coefficient LBVP is a generalisation of one discussed in Mattheij's paper [1] :

$$A(t) = \begin{bmatrix} 1 - k\,d & 0 & 1 + k\,s \\ 0 & k & 0 \\ 1 + k\,s & 0 & 1 + k\,d \end{bmatrix}$$

and $f(t) = \exp(t) * (-1 + k\,(d - s),\ -(k - 1),\ -1 - k\,(d + s))^T$

where $d = \cos(2t)$ and $s = \sin(2t)$.

(Mattheij considers the case for which $k = 19$).

We give results below for two sets of BC :

$$B_0 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \qquad B_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \text{(III.1)}$$

and

$$B_0 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \qquad B_1 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \text{(III.2)}$$

Mattheij states that with BC (III.2) the LBVP is well conditioned but with BC (III.1) it is very ill conditioned having a condition number of about 1.9e27.

For any value of parameter k the exact solution of the LBVP is $x(t) = \exp(t) * (1, 1, 1)^T$, and the problem interval is $[0, \pi]$, so specifying c.

Results :

(In this problem III section a reference such as table 4 is to table 3.4).

Table 3.1 : k = 19, cmax = 5e4, ns = 11, BC (III.1) :

h	err1	errf	
		ITER.3	ITER.7
0.1	3.5e0	6.1e0	2.0e0
0.05	6.9e-1	1.8e-2	3.7e-2
0.02	3.1e-2	3.2e-5	7.9e-8
0.0125	4.1e-3	6.8e-6	1.4e-8
0.01	1.7e-3	1.7e-6	2.1e-6
0.005	9.3e-5	9.2e-8	2.9e-7

In nearly all cases above we see that ITER.3 and ITER.7 produce improvement on the first solution this being most rapid generally (but not always) with ITER.7. But the case of h = 0.1 for ITER.3 shows that we must not place too much reliance on results obtained with these algorithms particularly when using large step-lengths.

Now compare the above results with those given in the table below for the same LBVP solved by ITMIN.357 :

Table 3.2 : k = 19, cmax = 5e4, ns = 11, BC (III.1) :

h	err1	errf : ITMIN.357
0.1	3.5e0	2.0e0
0.05	6.9e-1	3.7e-2
0.02	3.1e-2	5.3e-7
0.0125	4.1e-3	1.0e-6
0.01	1.7e-3	3.8e-9
0.005	9.3e-5	4.3e-7

ITMIN.357 produces improvement on the first solution in all cases including h = 0.1 for which ITER.3 failed.

The following results were also obtained with ITMIN.357 for the values of parameter k given and with the ill conditioned BC each time :

Table 3.3 : k = 6, cmax = 5e3, ns = 5, BC (III.1) :

h	err1	errf
0.1	4.1e-2	6.3e-8
0.05	2.4e-3	1.7e-8
0.02	5.9e-5	1.6e-9
0.0125	8.5e-6	2.9e-11
0.01	3.5e-6	4.4e-12

Table 3.4 : k = 12, cmax = 15e3, ns = 8, BC (III.1) :

h	err1	errf
0.1	1.3e0	2.1e-2
0.05	1.1e-1	3.7e-7
0.02	2.7e-3	7.1e-9

0.0125	3.6e-4	1.1e-7
0.01	1.5e-4	2.1e-8

Again we see that ITMIN.357 is successful in all cases in producing significant improvement on the first solution.

For comparison we include the results below which were obtained with ITMIN.357 using the well conditioned BC (III.2) :

Table 3.5 : k = 19, cmax = 50e3, ns = 12 :

h	err1	errf
0.1	2.4e-3	4.7e-7
0.05	1.1e-4	1.7e-7
0.02	2.4e-6	3.0e-7
0.0125	2.9e-7	2.9e-7
0.01	1.2e-7	1.2e-7

We see that here the amount of improvement decreases with the size of the integration steplength (h) used.

Finally the following result was obtained with ITMIN.357 using the ill conditioned BC and steplengths of $h_1 = 0.01$ and $h_2 = 0.05$:

Table 3.6 : k = 19, cmax = 50e3, ns = 12 :

err1 = 1.7e-3 errf = 2.7e-7.

We see that the value obtained for errf here is almost as small as that obtained when using $h_1 = h_2 = 0.01$ viz. 3.8e-9 (see table 2).

We attempted to solve the above LBVP using our variable step integrator RKF 45 and the Hermite cubic interpolation (program

ITVAR.3) but with little success as the results below show :

(RK = Runge Kutta integration tolerance used for both the first solution and also for the residual solutions)

Table 3.7 : k = 19, cmax = 50e3, BC (III.1), ns = 12 :

RK	h_o	err1	errf
1e-1	1e-1	6.6e-2	5.2e0
1e-1	1e-2	1.4e-2	5.9e1
1e-1	1e-3	5.0e-3	2.1e0
1e-2	1e-1	3.0e-3	1.0e0
1e-2	1e-2	1.4e-3	2.7e0
1e-2	1e-3	6.3e-4	2.0e-1
1e-3	1e-1	4.1e-4	7.3e-3
1e-3	1e-2	1.2e-4	1.1e-1
1e-3	1e-3	5.2e-5	2.0e-1
1e-4	1e-1	3.0e-5	1.6e-2
1e-4	1e-2	3.1e-5	4.0e-3
1e-4	1e-3	3.8e-5	1.6e-3

In each case above the method fails : errf > err1. However as we said earlier we can obtain improvement with ITVAR.3 by sufficiently reducing the size of the integration tolerance used in the solution of the residual IVPs. The results below were obtained by taking this tolerance to be ϵ/ 1e7 where ϵ is the tolerance used to integrate the first solution.

The initial steplength h_o was 0.1 in all cases.

Table 3.8 : k = 19, cmax = 50e3, BC (III.1), ns = 12 :

RK = ϵ	err1	errf
1e-1	6.6e-2	4.2e-6
1e-2	3.0e-3	1.1e-7
1e-3	4.1e-4	2.4e-7
1e-4	3.0e-5	2.6e-9

We now obtain improvement in every case and the values of errf are acceptably small. Running times are of course much longer than for the corresponding cases in table 7. However, these results were only obtained by trying successively smaller and smaller values for the residual integration tolerance until eventually we were successful with the value ϵ/1e7.

Variable step adjustment criteria:

We recall that the IVPs necessary to solve the given LBVP (6.1) and the residual LBVP (6.4) are the same except for the particular solutions. Also the first and residual particular solutions differ only in their respective forcing functions. These IVPs are, in each multiple shooting interval $[t_i, t_{i+1}]$ $(0 <= i <= N - 1)$:

$\dot{x}(t) = A(t)x(t) + f(t)$ $\qquad x(t_i) = 0$, and

$\dot{e}(t) = A(t)e(t) + r(t)$ $\qquad e(t_i) = 0$ respectively.

We give below details and test results of two criteria for

automatically adjusting the steplength of our variable step integrator RKF.45. These criteria both relate the integration tolerance to the size of the calculated particular solution. As the results below show, they proved successful in obtaining improvement on the first solution for each of the test LBVPs detailed earlier in this Chapter.

We denote the tolerances used in the integration of the first solution and the residual solutions by ϵ_1 and ϵ_r respectively.

Criterion A :

Initially ϵ_r is set equal to ϵ_1. At the end of each step of the residual integrations (i.e. at time $t = t^*$) we check that :

$$\epsilon_r < \frac{\| r(t^*, \epsilon_r) \|}{\| f(t^*) \|} \cdot \epsilon_1$$

. If not then we set $\epsilon_r := 0.1 * \epsilon_r$ and repeat the residual integrations from the beginning of the multiple shooting interval in which t^* lies. (We found that it is not sufficient simply to repeat the last step because the calculated value of $r(t^*)$ depends on the value of ϵ_r being used).

Criterion B :

We proceed as in A but instead we use the condition :

$$\epsilon_r < \frac{\| p_r(t^*, \epsilon_r) \|}{\| p_1(t^*, \epsilon_1) \|} \cdot \epsilon_1$$

where $p_r(t^*)$ is the calculated value of the residual particular solution and is dependent on the value of ϵ_r being used. $p_1(t^*)$ is the value of the first

particular solution at $t = t^*$. This is obtained by saving the final values of $p_i(t)$ at the end of each multiple shooting interval in the first integration and then using Hermite (cubic) interpolation over the interval containing t^* to find $p_i(t^*)$.

As the results below show, we found no significant difference between the effectiveness of these two criteria on our test problems. Both however suffer from the disadvantage of long program running times due to the very small values of ϵ_r employed (as small as 1e-12) and the need to restart from the beginning of the multiple shooting interval whenever ϵ_r is reduced (though most restarts did occur in the first step of the interval). By comparison, we obtained equally accurate solutions to these problems using fixed step integrations with algorithm ITMIN.357 in much less running time.

Results for Criteria A and B :

In each case h_o (the initial steplength) and ϵ_i were 1e-1, and Hermite cubic interpolation was used to find both the residual and first particular solution values :

Problem I : j = 20, k = 30, cmax = 1e7, ns = 10, ill conditioned BC (I.1) :

err1 = 9.3e-1 errf = 1.4e-7 (A)

errf = 3.0e-8 (B)

Problem II : k = 20, cmax = 1e6, ns = 5, BC as given :

err1 = 2.8e2 errf = 4.8e-7 (A)

errf = 8.1e-7 (B)

Problem III : k = 19, cmax = 50e3, ns = 12, ill conditioned

BC (III.1) :

err1 = 6.6e-2 errf = 2.3e-6 (A)

errf = 2.4e-6 (B).

With a view to putting a lower limit on the size of the residual integration tolerance ϵ_r we also tested the following criterion which is a variation of Criterion A :

Criterion C :

With ϵ_r set equal to ϵ_1 we perform the first residual integration, calculating as we do so the norm of this residual at the middle and the end of each integration step. These values are used to obtain an estimate of the maximum norm of the first residual over the problem range $[\alpha, \beta]$ which we will denote by rm1. The first residual integration is now repeated starting with ϵ_r equal to ϵ_1, but this time at the end of each integration step (i.e. at time $t = t^*$) we apply the following adjustment criterion :

if $\epsilon_r >= \dfrac{rm1}{\|f(t^*)\|} . \epsilon_1$ then we reduce ϵ_r by a factor of ten and repeat the residual integration from the beginning of

the multiple shooting interval in which t^* lies.

Results for Criterion C :

Problem I : j = 20, k = 30, cmax = 1e7, ns = 10, BC (I.1) :

ϵ_1	err1	errf	final ϵ_r
1e-1	9.3e-1	1.0e0	1e-4
1e-2	8.3e-2	1.1e-3	1e-6
1e-3	2.5e-3	6.8e-6	1e-9

Problem I : j = 5, k = 10, cmax = 5e3, ns = 6, BC (I.1) :

ϵ_1	err1	errf	final ϵ_r
1e-1	6.8e-6	5.4e-9	1e-7
1e-2	1.5e-6	1.9e-9	1e-8
1e-3	1.4e-7	1.3e-7	1e-9

Problem II : k = 20, cmax = 1e6, ns = 5, BC as given :

ϵ_1	err1	errf	final ϵ_r
1e-1	2.8e2	4.3e-4	1e-1
1e-2	3.1e1	2.4e-5	1e-3
1e-3	2.5e0	1.0e-6	1e-5

Problem II : k = 10, cmax = 1e3, ns = 5, BC as given :

ϵ_1	err1	errf	final ϵ_r
1e-1	2.8e-2	9.7e-9	1e-6
1e-2	3.0e-3	3.4e-10	1e-8
1e-3	2.7e-3	3.3e-7	1e-9

Problem III : k = 19, cmax = 50e3, ns = 12, BC (III.1) :

ϵ_1	err1	errf	final ϵ_r
1e-1	6.6e-2	2.1e-5	1e-6
1e-2	3.0e-3	2.4e-5	1e-7
1e-3	1.2e-4	1.8e-6	1e-8

Problem III : k = 6, cmax = 5e3, ns = 5, BC (III.1) :

ϵ_1	err1	errf	final ϵ_r
1e-1	3.9e-2	1.5e-4	1e-5
1e-2	1.1e-3	8.6e-7	1e-6
1e-3	1.8e-4	7.1e-8	1e-7

We see from the above results that Criterion C produced improvement on the first solution in all cases except one and the smallest value of ϵ_r employed was 1e-9. In some cases however the amount of improvement was small and this criterion has the disadvantage of having to repeat the whole of the first residual integration.

(Note that when using either Criterion A or C it will be necessary to account for the possibility of the forcing function f(t) of the given LBVP becoming zero in the problem interval. This can be done by making an initial transformation of the LBVP by putting x(t) = y(t) + k, where k is an arbitrary non-zero constant (n,1) vector, so that the forcing function f(t) becomes f(t) + A(t).k).

Conclusions :

The behaviour of the iterative residual correction method is complex when used with a variable step integrator because the final error in the calculated solution is a combination of integration error (dependent on the Runge Kutta tolerance and initial steplength used) and interpolation error (dependent on the degree of Hermite interpolant used). This complexity makes analysis difficult and our results obtained with the variable step integrator are inconclusive.

However, our fixed step integrator results (programs ITER.3, ITER.5, ITER.7 and especially ITMIN.357) show how successful the method can be in solving even quite ill conditioned LBVPs. We think that this justifies the need for further research in this area with a view to formulating a theoretical foundation for the method which will show the inter-relationship between integration error and interpolation error and their combined effect on the estimated error obtained. Hopefully, this will then suggest an efficient step adjustment criterion for the variable step integrator which will make the iterative residual correction method a reliable and useful practical solver of LBVPs.

For comparison we attempted to solve some of the ill conditioned LBVPs detailed in this Chapter by using the factorisation methods of Babuska and Majer [16] as described in Chapter 5. These results are given in Appendix

III. They show that the factorisation methods fail to produce accurate solutions to these difficult problems. This was only to be expected since the success of these 'double sweep' methods depends very much on the stability of the auxiliary IVPs in their respective directions [16] and the good condition of the problem is necessary to ensure this. By contrast, our fixed step iterative method (ITMIN.357) produced acceptably accurate solutions to all of these ill conditioned problems. This underlines the justification for further investigation of this iterative method.

APPENDIX I

[1-1] : Consider the nth order differential equation $x^{n} = F(t, x, \dot{x}, \ddot{x}, \ldots\ldots, x^{n-1})$ where F is linear in $(x, \dot{x}, \ldots\ldots, x^{n-1})$. Let $x_1 = x$, $x_2 = \dot{x}$, $x_3 = \ddot{x}$, $\ldots\ldots$ $x_n = x^{n-1}$. Then $\dot{x}_1 = \dot{x} = x_2$; $\dot{x}_2 = \ddot{x} = x_3$; $\ldots\ldots\ldots \dot{x}_{n-1} = x_n$ and $\dot{x}_n = \dot{x}^{n-1} = x^{n} = F$. Hence the given nth order differential equation can be written in the form :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & & & & \\ 0 & 0 & 1 & & & \\ 0 & 0 & 0 & 1 & \ddots & \\ & & & & \ddots & 1 \\ 0 & 0 & 0 & 0 & \cdots\cdots & 1 \\ k_1 & k_2 & k_3 & k_4 & \cdots\cdots & k_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ g(t) \end{bmatrix}$$

where k_i $(1 <= i <= n)$ are functions of t or constants i.e. it can be written in the form $\dot{x}(t) = A(t)x(t) + f(t)$.

[1-2] : First note that for any continuous function u :

$\frac{d}{dt} \int_a^t u(s).ds = u(t)$, where a is a constant and s and t are real variables. Now consider $x(t) = X(t)\alpha + p(t)$ (1) where X(t) is any fundamental solution of system $\dot{x}(t) = A(t)x(t)$, p(t) is a particular solution of $\dot{x}(t) = A(t)x(t) + f(t)$ and α is constant (n,1). From (1) :

$$\dot{x}(t) = \dot{X}(t)\alpha + \dot{p}(t) = A(t)X(t)\alpha + A(t)p(t) + f(t)$$

$$= A(t)\{X(t)\alpha + p(t)\} + f(t)$$

$$= A(t)x(t) + f(t).$$

Thus any function of form (1) satisfies the ODE

$\dot{x}(t) = A(t)x(t) + f(t)$. Now take :

$$\alpha = Q^{-1}\{c - B_1 X(b) \int_a^b u(s).ds\} \qquad (2)$$

where $Q = B_0 X(a) + B_1 X(b)$, and take $p(t) = X(t) \int_a^t u(s).ds$ (3)

where $u(s) = X^{-1}(s)f(s)$ (4) .

Note that from (3) we get :

$$\dot{p}(t) = X(t) \frac{d}{dt} \int_a^t u(s).ds + \dot{X}(t) \int_a^t u(s).ds$$

$$= X(t)X^{-1}(t)f(t) + A(t)X(t) \int_a^t u(s).ds$$

i.e. $\dot{p}(t) = A(t)p(t) + f(t)$, as required.

From (1), (2) and (3) : $x(a) = X(a)Q^{-1}\{c - B_1 X(b)J\}$ where $J = \int_a^b u(s).ds$, since $p(a) = 0$.

Also $x(b) = X(b)Q^{-1}\{c - B_1 X(b)J\} + X(b)J$. Hence

$$B_0 x(a) + B_1 x(b) = \{B_0 X(a) + B_1 X(b)\}Q^{-1}\{c - B_1 X(b)J\} + B_1 X(b)J$$

$$= c - B_1 X(b)J + B_1 X(b)J = c$$

i.e. $B_0 x(a) + B_1 x(b) = c$.

Hence the exact solution of the given LBVP :

$\dot{x}(t) = A(t)x(t) + f(t)$

$B_0 x(a) + B_1 x(b) = c$

can be written $x(t) = X(t)\alpha + p(t)$ where α and $p(t)$ are as defined in (2) and (3).

Now $X(t).\alpha = X(t)Q^{-1}c - X(t)Q^{-1}B_1 X(b)J$

$= \Phi(t)c - \Phi(t)B_1 X(b) \int_a^b X^{-1}(s)f(s).ds$

(where $X(t) = \Phi(t).Q$)

$= \Phi(t)c - \Phi(t)B_1 \Phi(b)Q \int_a^b Q^{-1} \Phi^{-1}(s)f(s).ds$

$= \Phi(t)c + \int_a^b - \Phi(t)B_1 \Phi(b) \Phi^{-1}(s)f(s).ds$

Also

$p(t) = X(t) \int_a^t X^{-1}(s)f(s).ds = \Phi(t)Q \int_a^t Q^{-1} \Phi^{-1}(s)f(s).ds$

i.e. $p(t) = \int_a^t \Phi(t) \Phi^{-1}(s)f(s).ds$ and so

$x(t) = X(t).\alpha + p(t)$

$= \Phi(t)c + \int_a^b - \Phi(t)B_1 \Phi(b) \Phi^{-1}(s)f(s).ds + \int_a^t \Phi(t). \Phi^{-1}(s)f(s)ds$

$= \Phi(t)c + \int_a^t \Phi(t)\{I_n - B_1 \Phi(b)\}. \Phi^{-1}(s)f(s).ds$

$+ \int_t^b - \Phi(t)B_1 \Phi(b) \Phi^{-1}(s)f(s).ds$

$= \Phi(t)c + \int_a^t \Phi(t)B_0 \Phi(a). \Phi^{-1}(s)f(s).ds$

$+ \int_t^b - \Phi(t)B_1 \Phi(b). \Phi^{-1}(s)f(s).ds$

because $B_0 X(a) + B_1 X(b) = Q \implies B_0 \Phi(a) + B_1 \Phi(b) = I_n$.

[1-3] : Recall that the substitution $x(t) = T(t)y(t)$ transforms the system $\dot{x}(t) = A(t)x(t)$ (n,n) into $\dot{y}(t) = V(t)y(t)$ where $V(t) = T^{-1}(t)A(t)T(t) - T^{-1}(t)\dot{T}(t)$. Thus if $T(t)$ has the block diagonal form $\begin{matrix} \\ \ell \\ m \end{matrix}\begin{matrix} \begin{matrix} \ell & m \end{matrix} \\ \begin{bmatrix} I_\ell & 0 \\ 0 & D \end{bmatrix} \end{matrix}$ and $A(t)$ has block upper triangular form $\begin{matrix} \\ \ell \\ m \end{matrix}\begin{matrix} \begin{matrix} \ell & m \end{matrix} \\ \begin{bmatrix} F & C \\ 0 & B \end{bmatrix} \end{matrix}$ then $V(t)$ will also be block

upper triangular i.e. $V(t) = \begin{bmatrix} F & CD \\ 0 & W \end{bmatrix}$ where $W = D^{-1}BD - D^{-1}\dot{D}$. Therefore we first find an orthonormal transformation $T_0(n,n)$ which puts $A(n,n)$ into the form :

$A_1 = \begin{bmatrix} F & C \\ 0 & B \end{bmatrix}$. (See Chapter 1, (1.31) and (1.32). For initial conditions for these equations we can here take $T_0(a)$ to be any orthonormal matrix e.g. $T_0(a) = I_n$).

We then obtain orthonormal transformation D_1 $(n-1, n-1)$ such that $W_1 = D_1^{-1}BD - D_1^{-1}\dot{D}_1$ is block upper triangular of the form $W_1 = \begin{bmatrix} F_1 & C_1 \\ 0 & B_1 \end{bmatrix}$ and let $T_1 = \begin{bmatrix} I_1 & 0 \\ 0 & D_1 \end{bmatrix}$. Similarly, we obtain orthonormal transformation D_2 $(n-2, n-2)$ such that $W_2 = D_2^{-1}B_1D_2 - D_2^{-1}\dot{D}_2$ is block upper triangular of the form $W_2 = \begin{bmatrix} F_2 & C_2 \\ 0 & B_2 \end{bmatrix}$ and we let $T_2 = \begin{bmatrix} I_2 & 0 \\ 0 & D_2 \end{bmatrix}$. This process is repeated to obtain T_3 ..... T_{n-2} where $T_{n-2} = \begin{bmatrix} I_{n-2} & 0 \\ 0 & D_{n-2} \end{bmatrix}$.

Then the required transformation which will put A into upper triangular form A_{n-1} is $\prod_{i=n-2}^{0} T_i$. The kinematic eigenvalues of A are now the diagonal elements of A_{n-1}.

(This deflation method shows the existence of kinematic eigen-

values but in practice would be very costly).

[1-4] : In the following we abbreviate orthonormal transformation $T(t) = [T_1(t) \mid T_2(t)]$ to $T = [T_1 \mid T_2]$, transformed system matrix $V(t)$ to V and functions $c_{11}(t)$, $c_{22}(t)$ to c_{11}, c_{22}. Now from the Lyapunov equation $TV = AT - \dot{T}$ we get $V = T^T(AT - \dot{T})$. Thus $V = \begin{bmatrix} T_1^T \\ T_2^T \end{bmatrix} . [AT_1 - \dot{T}_1 \mid AT_2 - \dot{T}_2]$

$\Longrightarrow$ $V_{11} = T_1^T (AT_1 - \dot{T}_1)$; $V_{12} = T_1^T (AT_2 - \dot{T}_2)$

$V_{21} = T_2^T (AT_1 - \dot{T}_1)$; $V_{22} = T_2^T (AT_2 - \dot{T}_2)$.

Now for V to be block upper triangular $V_{21} = 0$ $\Longrightarrow$ $T_2^T (AT_1 - \dot{T}_1) = 0$. But $T_2^T T_1 = 0$ $\Longrightarrow$ $T_2^T (T_1 c_{11}) = 0$ where c_{11} is any (p,p) matrix i.e. T_1 is a basis for the space othogonal to T_2 and so any matrix orthogonal to T_2 can be written in the form $T_1 c_{11}$ for some c_{11}. Hence

$AT_1 - \dot{T}_1 = T_1 c_{11}$ for some c_{11}

$\Longrightarrow$ $\dot{T}_1 = AT_1 - T_1 c_{11}$ (1)

Now $T_1^T T_1 = I_p$ $\Longrightarrow$ $T_1^T \dot{T}_1 + \dot{T}_1^T T_1 = 0$ (2).

From (1): $T_1^T \dot{T}_1 = T_1^T AT_1 - T_1^T T_1 c_{11} = T_1^T AT_1 - c_{11}$

$\Longrightarrow$ $c_{11} = T_1^T AT_1 - T_1^T \dot{T}_1 = V_{11}$

$\Longrightarrow$ $c_{11}^T = T_1^T A^T T_1 - \dot{T}_1^T T_1$

$\Longrightarrow$ $c_{11} + c_{11}^T = T_1^T (A + A^T) T_1$ (from (2)).

Also $T_1^T T_2 = 0$ $\Longrightarrow$ $T_1^T \dot{T}_2 + \dot{T}_1^T T_2 = 0$ $\Longrightarrow$

$T_1^T \dot{T}_2 + (AT_1 - T_1 c_{11})^T T_2 = 0$ (from (1)) $\Longrightarrow$

$T_1^T \dot{T}_2 + T_1^T A^T T_2 = 0$ $\Longrightarrow$ $T_1^T (\dot{T}_2 + A^T T_2) = 0$.

But $T_1^T (T_2 c_{22}^T) = 0$ where c_{22}^T is (q,q) $\Longrightarrow$

$\dot{T}_2 + A^T T_2 = T_2 c_{22}^T$ for some c_{22}^T $\Longrightarrow$

$\dot{T}_2 = -A^T T_2 + T_2 c_{22}^T$ $\Longrightarrow$ $T_2^T \dot{T}_2 = -T_2^T A^T T_2 + c_{22}^T$

$\Longrightarrow$ $c_{22}^T = T_2^T A^T T_2 + T_2^T \dot{T}_2$ $\Longrightarrow$

$c_{22} = T_2^T A T_2 + \dot{T}_2^T T_2$ and $c_{22} + c_{22}^T = T_2^T (A + A^T) T_2$.

Also : $c_{22} = T_2^T A T_2 - T_2^T \dot{T}_2 = T_2^T (A T_2 - \dot{T}_2) = V_{22}$

and $V_{12} = T_1^T A T_2 - T_1^T \dot{T}_2$

$= T_1^T A T_2 - T_1^T (-A^T T_2 + T_2 c_{22}^T)$

$= T_1^T A T_2 + T_1^T A^T T_2$

$= T_1^T (A + A^T) T_2$.

N.B. Since $T(n,n)$ is orthonormal we have $TT^T = I_n$ $\Longrightarrow$

$$[T_1 \mid T_2] \begin{bmatrix} T_1^T \\ T_2^T \end{bmatrix} = I_n \qquad \Longrightarrow \quad T_1 T_1^T + T_2 T_2^T = I_n .$$

Thus equation (1.32) of Chapter 1 viz.

$\dot{T}_2 = [-I_n + T_2 T_2^T] A^T T_2$ can also be written

$\dot{T}_2 = -T_1 T_1^T A^T T_2$ as in (3.2) of Chapter 3.

[1-5] : Any pair of corresponding fundamental solutions of the systems $\dot{x}(t) = A(t)x(t)$ and $\dot{y}(t) = V(t)y(t)$ are related by $X(t) = T(t)Y(t)$ for all t. Thus if we choose $X(a) = T(a)$ then $Y(a) = I_n$. Now $Y(t)$ satisfies $\dot{Y}(t) = V(t)Y(t)$ where $V_{21}(t) = 0$ and so $\dot{Y}_{21}(t) = V_{22}(t)Y_{21}(t)$ where $Y_{21}(a) = 0$. Hence $Y_{21}(t) = 0$ for all t, and so $\dot{Y}_{11}(t) = V_{11}(t)Y_{11}(t)$ where $Y_{11}(a) = I_p$ $\Longrightarrow$ $Y_{11}(t)$ is non-singular for all t.

Now : $X(t) = T(t)Y(t)$ ====>

$$[X_1(t) \mid X_2(t)] = [T_1(t) \mid T_2(t)] \begin{bmatrix} Y_{11}(t) & Y_{12}(t) \\ 0 & Y_{22}(t) \end{bmatrix}$$

====> $X_1(t) = T_1(t)Y_{11}(t)$ ====> $T_1(t) = X_1(t) \cdot Y_{11}^{-1}(t)$.

Therefore span $T_1(t)$ = span $X_1(t)$, since each column of $T_1(t)$ is a linear combination of the columns of $X_1(t)$. Now $X_1(a) = T_1(a)$ where span $T_1(a)$ forms a basis for a growth subspace of system $\dot{x}(t) = A(t)x(t)$. Thus span $X_1(t)$ and hence span $T_1(t)$ must form a basis for a growth subspace for all t.

[2-1] : $x_i(t_{i+1}) = x_{i+1}(t_{i+1})$ $(0 <= i <= N-2)$ ====>

$X_i'(t_{i+1})\beta_i + v_i(t_{i+1}) = X_{i+1}'(t_{i+1})\beta_{i+1} + v_{i+1}(t_{i+1})$ (by 2.18)

====> $X_{i+1}'(t_{i+1})D_i\beta_i + v_i(t_{i+1}) = X_{i+1}'(t_{i+1})\beta_{i+1} + v_i(t_{i+1})$

$- X_{i+1}'(t_{i+1})\{X_{i+1}'(t_{i+1})\}^T v_i(t_{i+1})$

====> $D_i\beta_i + \{X_{i+1}'(t_{i+1})\}^T v_i(t_{i+1}) = \beta_{i+1} + \{X_{i+1}'(t_{i+1})\}^T v_i(t_{i+1})$

$- \{X_{i+1}'(t_{i+1})\}^T v_i(t_{i+1})$,

(multiplying by $\{X_{i+1}'(t_{i+1})\}^T$)

====> $\beta_{i+1} = D_i\beta_i + \{X_{i+1}'(t_{i+1})\}^T v_i(t_{i+1})$.

[3-1] : In the following R_{ij} , Y_{ij} , V_{ij} , l_i , h_i and g_i are all functions of t.

$R_{11}Y_{11} = I_p$ ====> $\dot{R}_{11}Y_{11} + R_{11}\dot{Y}_{11} = 0$ ====>

$\dot{R}_{11}Y_{11} = -R_{11}\dot{Y}_{11}$ ====> $\dot{R}_{11} = -R_{11}\dot{Y}_{11}Y_{11}^{-1}$ where $\dot{Y}_{11} = V_{11}Y_{11}$

====> $\dot{R}_{11} = -R_{11}V_{11}$.

Also $Y_{11}R_{12} = -Y_{12}$ ====> $\dot{Y}_{11}R_{12} + Y_{11}\dot{R}_{12} = -\dot{Y}_{12}$ ===>

$Y_{11}\dot{R}_{12} = -\dot{Y}_{11}R_{12} - \dot{Y}_{12}$ ====>

$$\dot{R}_{12} = - R_{11} \dot{Y}_{11} R_{12} - R_{11} \dot{Y}_{12}$$

$$= - R_{11} V_{11} Y_{11} R_{12} - R_{11} \{V_{11} Y_{12} + V_{12} Y_{22}\}$$

$$= - R_{11} V_{11} Y_{11} R_{12} + R_{11} V_{11} Y_{11} R_{12} - R_{11} V_{12} Y_{22}$$

$$= - R_{11} V_{12} Y_{22} .$$

Also $l_1 = - R_{11} h_1 \quad \Longrightarrow \quad \dot{l}_1 = - R_{11} \dot{h}_1 - \dot{R}_{11} h_1$

$$\Longrightarrow \quad \dot{l}_1 = - R_{11} \{V_{11} h_1 + V_{12} h_2 + g_1\} + R_{11} V_{11} h_1$$

$$= - R_{11} \{V_{12} h_2 + g_1\} .$$

[3-2] : In the following X, A, Y, V, $\bar{V}$, T, R_{11}, x, w are all functions of t.

Note first that if X (n,n) is any fundamental solution of system $\dot{x} = Ax$ then $XX^{-1} = I_n \Longrightarrow$

$$\dot{X} X^{-1} + X \frac{d}{dt}(X^{-1}) = 0 \quad \Longrightarrow \quad X \frac{d}{dt}(X^{-1}) = - \dot{X} X^{-1}$$

$$\Longrightarrow \quad \frac{d}{dt}(X^{-1}) = - X^{-1} \dot{X} X^{-1}$$

$$\Longrightarrow \quad \frac{d}{dt}(X^{-1}) = - X^{-1} A X X^{-1} = - X^{-1} A$$

$$\Longrightarrow \quad \frac{d}{dt}(X^{-T}) = - A^T X^{-T} .$$

We now show that if transformation T (non-singular) exists such that :

$$\dot{x} = Ax \quad (1a) \qquad \xrightarrow{\quad x = Tw \quad} \qquad \dot{w} = Vw \quad (1b)$$

then transformation T^{-T} will be such that :

$$\dot{x} = - A^T x \quad (2a) \qquad \xrightarrow{\quad x = T^{-T} w \quad} \qquad \dot{w} = \bar{V} w \quad (2b)$$

where $\bar{V} = -V^T$.

If X is any fundamental solution of $\dot{x} = Ax$ then X^{-T} is a fundamental solution of the adjoint system $\dot{x} = -A^T x$, because $\frac{d}{dt}(X^{-T}) = -A^T X^{-T}$, as shown above. Now suppose that X and Y are corresponding fundamental solutions of systems (1a) and (1b) above respectively i.e. $X = TY$ and hence $X^{-T} = T^{-T} Y^{-T}$ (3). Then X is a fundamental solution of (1a) ====> X^{-T} is a fundamental solution of (2a) ====> Y^{-T} is a fundamental solution of (2b), (from (3)). But Y is a fundamental solution of (1b) ====> Y^{-T} is a fundamental solution of the adjoint to (1b) i.e. of $\dot{w} = -V^T w$. Hence $\bar{V} = -V^T$.

Thus if V is upper triangular with all of its diagonal elements positive then $\bar{V}$ will be lower triangular with all of its diagonal elements negative i.e. if all of the kinematic eigenvalues of (1a) are positive then all of the kinematic eigenvalues of the adjoint system (2a) will be negative at the same value of t. Now equation (3.10) viz. $\dot{R}_{11} = -R_{11} V_{11}$ can be written $\dot{R}_{11}^T = -V_{11}^T R_{11}^T$ where all of the kinematic eigenvalues of V_{11} are such that $\mathrm{Re} \int_a^b \sigma_i(t).dt > 0$. This shows that IVP (3.10) will be (forward) stable even when V_{11} is variable.

[4-1] : In the following X, R, $\tilde{R}$, $\tilde{X}$, x, y, $\tilde{x}$, $\tilde{y}$ are all functions of t, but P and π are constants.

$$\tilde{X}_1 = \Pi X_1 \quad ====> \begin{bmatrix} \tilde{X}_{11} \\ \tilde{X}_{21} \end{bmatrix} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} \cdot \begin{bmatrix} X_{11} \\ X_{21} \end{bmatrix} \quad ====>$$

$$\tilde{X}_{11} = P_{11} X_{11} + P_{12} X_{21} \quad \text{and} \quad \tilde{X}_{21} = P_{21} X_{11} + P_{22} X_{21} \quad ====>$$

$$\tilde{X}_{21} \tilde{X}_{11}^{-1} = (P_{21} X_{11} + P_{22} X_{21}) . (P_{11} X_{11} + P_{12} X_{21})^{-1}$$

$$= (P_{21} + P_{22} R) X_{11} . \{(P_{11} + P_{12} R) X_{11}\}^{-1}$$

$$= (P_{21} + P_{22} R) X_{11} X_{11}^{-1} (P_{11} + P_{12} R)^{-1} \quad ====>$$

$$\tilde{R} = (P_{21} + P_{22} R) . (P_{11} + P_{12} R)^{-1}$$

Now $\begin{bmatrix} \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix} = \Pi \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ ====> $\tilde{x}_1 = P_{11} x_1 + P_{12} x_2$

$\tilde{x}_2 = P_{21} x_1 + P_{22} x_2$.

From the Riccati transformation equation :

$\tilde{x}_2 = \tilde{R} \tilde{x}_1 + \tilde{y}_2$ and $x_2 = R x_1 + y_2$.

Hence : $(P_{21} x_1 + P_{22} x_2) = \tilde{R} (P_{11} x_1 + P_{12} x_2) + \tilde{y}_2$

===> $P_{21} x_1 + P_{22} (R x_1 + y_2) = \tilde{R} P_{11} x_1 + \tilde{R} P_{12} (R x_1 + y_2) + \tilde{y}_2$

$= \{\tilde{R}(P_{11} + P_{12} R)\} x_1 + \tilde{R} P_{12} y_2 + \tilde{y}_2$

$= (P_{21} + P_{22} R) x_1 + \tilde{R} P_{12} y_2 + \tilde{y}_2$

====> $P_{22} y_2 = \tilde{R} P_{12} y_2 + \tilde{y}_2$

====> $\tilde{y}_2 = (P_{22} - \tilde{R} P_{12}) y_2$

[4-2] : $\tilde{x}_1 = P_{11} x_1 + P_{12} x_2 = P_{11} x_1 + P_{12} (R x_1 + y_2)$

====> $\tilde{x}_1 - P_{12} y_2 = (P_{11} + P_{12} R) x_1$ ====>

$x_1 = (P_{11} + P_{12} R)^{-1} . (\tilde{x}_1 - P_{12} y_2)$ or

$y_1 = (P_{11} + P_{12} R)^{-1} . (\tilde{y}_1 - P_{12} y_2)$.

[4-3] : In the following, all values are functions of t except P, I, G and Π which are constants.

Suppose that $P_1 \begin{bmatrix} I_p \\ R \end{bmatrix} = \begin{bmatrix} A \\ B \end{bmatrix}$ and $\begin{bmatrix} A \\ B \end{bmatrix} . G_1 = \begin{bmatrix} C \\ D \end{bmatrix}$.

Then $D\,C^{-1} = B\,A^{-1} = S$.

Let $P_2 \begin{bmatrix} A \\ B \end{bmatrix} = \begin{bmatrix} L \\ M \end{bmatrix}$ (1)

and $P_2 \begin{bmatrix} C \\ D \end{bmatrix} . G_2 = \begin{bmatrix} E \\ F \end{bmatrix}$ (2).

Now (1) ====> $P_2 \begin{bmatrix} I_p \\ S \end{bmatrix} = \begin{bmatrix} L \\ M \end{bmatrix} . A^{-1} = \begin{bmatrix} N \\ T \end{bmatrix}$ where

$T\,N^{-1} = M\,L^{-1}$.

(2) ====> $P_2 \begin{bmatrix} I_p \\ S \end{bmatrix} [C.G_2] = \begin{bmatrix} E \\ F \end{bmatrix}$ ====>

$P_2 \begin{bmatrix} I_p \\ S \end{bmatrix} = \begin{bmatrix} E \\ F \end{bmatrix} . U = \begin{bmatrix} E.U \\ F.U \end{bmatrix}$.

Therefore $\begin{bmatrix} E\,U \\ F\,U \end{bmatrix} = \begin{bmatrix} N \\ T \end{bmatrix}$ ====> $F\,E^{-1} = T\,N^{-1} = M\,L^{-1}$

i.e. $F\,E^{-1} = M\,L^{-1}$.

Thus if : $P_2\,P_1 \begin{bmatrix} I_p \\ R \end{bmatrix} = \begin{bmatrix} L \\ M \end{bmatrix}$ and $P_2\,P_1 \begin{bmatrix} I_p \\ R \end{bmatrix} G_1\,G_2 = \begin{bmatrix} E \\ F \end{bmatrix}$

then $M\,L^{-1} = F\,E^{-1}$.

Hence, in general, if $\Pi \begin{bmatrix} I_p \\ R \end{bmatrix} . G = \begin{bmatrix} E \\ F \end{bmatrix}$, where P_i, G_i are performed alternately, and if $\Pi \begin{bmatrix} I_p \\ R \end{bmatrix} = \begin{bmatrix} L \\ M \end{bmatrix}$

then $F\,E^{-1} = M\,L^{-1}$. Now $\Pi = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$ and so

$L = (P_{11} + P_{12} R)$ and $M = (P_{21} + P_{22} R)$.

Therefore $\tilde{R} = (P_{21} + P_{22} R).(P_{11} + P_{12} R)^{-1}$ (see (4.8a))

$= M L^{-1} = F E^{-1}$.

[5-1] : In the following T, A, $\tilde{A}$, R are all functions of t.

From the Lyapunov equation : $\dot{T} = A T - T \tilde{A}$ we get

$$\begin{bmatrix} 0 & -\dot{R} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} I_{n_1} & -R \\ 0 & I_{n_2} \end{bmatrix} - \begin{bmatrix} I_{n_1} & -R \\ 0 & I_{n_2} \end{bmatrix} \begin{bmatrix} \tilde{A}_{11} & 0 \\ \tilde{A}_{21} & \tilde{A}_{22} \end{bmatrix}$$

$$\Longrightarrow \quad 0 = A_{11} - \tilde{A}_{11} + R \tilde{A}_{21}$$

$$-\dot{R} = -A_{11} R + A_{12} + R \tilde{A}_{22}$$

$$0 = A_{21} - \tilde{A}_{21}$$

$$0 = -A_{21} R + A_{22} - \tilde{A}_{22} .$$

Thus : $\tilde{A}_{11} = A_{11} + R A_{21}$; $\tilde{A}_{21} = A_{21}$;

$\tilde{A}_{22} = A_{22} - A_{21} R$ and

$$\dot{R} = A_{11} R - A_{12} - R \tilde{A}_{22}$$

$$= A_{11} R - A_{12} - R(A_{22} - A_{21} R)$$

i.e. $\dot{R} = A_{11} R + R A_{21} R - A_{12} - R A_{22}$.

[5-2]: The proof is similar to that given in [1-4] except that now $V_{12}(t)$ (instead of $V_{21}(t)$) $= 0$. Also $c_{11}(t) = T_1^T(t) B(t) T_1(t)$.

[5-3] : The ODE for $L_2(t)$ corresponds to the $T_2(t)$ equation derived in [1-4], but with $c_{22}(t) = T_2^T(t) B(t) T_2(t)$.

APPENDIX II :

RESULTS FOR ERROR ESTIMATION METHOD

In obtaining all of the results in this appendix the variable step integrator RKF 45 was used to solve the auxiliary IVPs. The tables below give, for each test problem, comparative values of estimated and actual absolute errors in the components of the first calculated LBVP solution vector at each end of the problem range.

est0 = estimated error at initial point of problem range
act0 = actual error at initial point of problem range
est1 = estimated error at final point of problem range
act1 = actual error at final point of problem range
All results are given correct to two significant figures.
The details of the test problems are as given in Chapter 6 .

TEST PROBLEM I :

(In this section a reference such as table 3 is to table A2.1.3).

Here we used a Runge Kutta tolerance of 1e-4 and an initial steplength of 0.01 for all integrations :

Table A2.1.1 : j = 2, k = 3, cmax = 10, BC (I.1), ns = 7 :

est0	act0	est1	act1
0	0	3.2e-9	4.6e-9
6.9e-9	1.5e-9	0	0
6.9e-8	6.9e-8	0	0

Table A2.1.2 : j = 5, k = 10, cmax = 1e4, BC (I.1), ns = 5 :

est0	act0	est1	act1
0	0	9.5e-9	5.0e-11
3.1e-9	1.4e-9	0	0
1.0e-7	1.4e-8	0	0

Table A2.1.3 :j = 10, k = 15, cmax = 1e5, BC (I.1), ns = 7 :

est0	act0	est1	act1
1.5e-25	0	2.4e-9	9.9e-12
5.6e-9	5.7e-11	0	0
7.8e-8	1.1e-9	0	0

Table A2.1.4 : j = 15, k = 20, cmax = 1e6, BC (I.1), ns = 9 :

est0	act0	est1	act1
4.3e-26	0	1.0e-9	1.5e-12
2.6e-9	4.0e-11	0	0
6.0e-8	6.3e-10	0	0

Table A2.1.5 : j = 20, k = 25, cmax = 1e7, BC (I.1), ns = 10 :

est0	act0	est1	act1
0	0	1.4e-9	3.4e-13
5.7e-9	2.4e-11	0	0
9.1e-8	5.0e-10	0	0

Table A2.1.6 : j = 25, k = 30, cmax = 1e8, BC (I.1), ns = 9 :

est0	act0	est1	act1
0	0	6.3e-10	1.4e-13
1.0e-8	3.4e-12	0	0
1.2e-7	9.6e-11	0	0

As the well conditioned BC were used in the above we see, as expected, that the size of the actual errors decreases as parameters j and k increase i.e. as the problem becomes stiffer. However, estimated errors are greater than the corresponding actual errors in nearly all cases.

In tables 7 to 12 below the ill conditioned BC were used :

Table A2.1.7 : j = 2, k = 3, cmax = 10, BC (I.2), ns = 7 :

est0	act0	est1	act1
1.1e-23	2.2e-16	1.2e-8	6.5e-9
1.6e-24	0	1.1e-8	2.5e-9
7.1e-8	7.0e-8	0	0

Table A2.1.8 : j = 5, k = 10, cmax = 1e4, BC (I.2), ns = 5 :

est0	act0	est1	act1
8.4e-25	0	4.4e-8	1.6e-8
5.4e-24	0	1.1e-7	5.3e-8
1.0e-7	1.4e-8	0	0

Table A2.1.9 : j = 10, k = 15, cmax = 1e5, BC (I.2), ns = 7 :

est0	act0	est1	act1
0	0	3.5e-6	3.5e-8
1.3e-24	0	2.1e-5	2.1e-7
7.8e-8	1.1e-9	0	0

Table A2.1.10 : j = 15, k = 20, cmax = 1e6, BC (I.2), ns = 9 :

est0	act0	est1	act1
4.3e-26	0	1.2e-4	1.9e-6
0	0	1.1e-3	1.6e-5
6.0e-8	6.4e-10	0	0

Table A2.1.11 : j = 20, k = 25, cmax = 1e7, BC (I.2), ns = 10 :

est0	act0	est1	act1
1.2e-25	0	2.5e-2	1.1e-4
8.3e-25	2.2e-16	2.8e-1	1.2e-3
9.1e-8	5.0e-10	0	0

Table A2.1.12 : j = 25, k = 30, cmax = 1e8, BC (I.2), ns = 9 :

est0	act0	est1	act1
2.6e-24	0	4.6e0	1.5e-3
1.6e-24	0	6.2e1	2.1e-2
1.2e-7	9.6e-11	0	0

As expected with the ill conditioned BC the actual errors increase in size as the problem becomes stiffer and more ill conditioned as parameters j and k increase. For values of

these up to j = 15 , k = 20 estimated errors are a reasonable guide to actual errors but as the problem becomes more ill conditioned than this the estimated errors at the right hand side become very inaccurate (tables 11 and 12).

Test Problem II :

(In this section a reference such as table 3 is to table A2.2.3).

Here again we used a Runge Kutta tolerance of 1e-4 and an initial steplength of 0.01.

Table A2.2.1 : k = 5, cmax = 100, ns = 7 :

est0	act0	est1	act1
8.3e-8	8.3e-8	2.1e-7	2.1e-7
8.3e-8	8.3e-8	1.7e-7	1.7e-7
8.3e-8	8.3e-8	6.2e-8	6.1e-8
8.3e-8	8.3e-8	1.2e-6	1.2e-6

Table A2.2.2 : k = 10, cmax = 1e3, ns = 9 :

est0	act0	est1	act1
5.2e-6	5.2e-6	1.4e-5	1.4e-5
5.2e-6	5.2e-6	1.3e-5	1.3e-5
5.2e-6	5.2e-6	2.7e-6	2.7e-6
5.2e-6	5.2e-6	1.0e-4	1.0e-4

Table A2.2.3 : k = 15, cmax = 1e5, ns = 6 :

est0	act0	est1	act1
1.5e-4	1.5e-4	4.2e-4	4.2e-4
1.5e-4	1.5e-4	4.0e-4	4.0e-4
1.5e-4	1.5e-4	1.8e-4	1.8e-4
1.5e-4	1.5e-4	3.2e-3	3.2e-3

Table A2.2.4 : k = 20, cmax = 1e6, ns = 7 :

est0	act0	est1	act1
3.2e-2	3.2e-2	8.7e-2	8.8e-2
3.2e-2	3.2e-2	8.5e-2	8.6e-2
3.2e-2	3.2e-2	4.8e-2	4.8e-2
3.2e-2	3.2e-2	7.0e-1	7.0e-1

Table A2.2.5 : k = 25, cmax = 1e6, ns = 8 :

est0	act0	est1	act1
9.0e-1	9.1e-1	2.5e0	2.5e0
9.0e-1	9.1e-1	2.4e0	2.4e0
9.0e-1	9.1e-1	1.5e0	1.6e0
9.0e-1	9.1e-1	2.0e1	2.0e1

Table A2.2.6 : k = 30, cmax = 1e7, ns = 8 :

est0	act0	est1	act1
1.7e2	1.8e2	4.7e2	4.9e2
1.7e2	1.8e2	4.7e2	4.9e2
1.7e2	1.8e2	3.2e2	3.4e2
1.7e2	1.8e2	3.9e3	4.1e3

Table A2.2.7 : k = 35, cmax = 1e8, ns = 8 ;

est0	act0	est1	act1
1.7e3	2.9e3	4.6e3	7.9e3
1.7e3	2.9e3	4.6e3	7.8e3
1.7e3	2.9e3	3.4e3	5.8e3
1.7e3	2.9e3	3.9e4	6.7e4

Table A2.2.8 : k = 40, cmax = 1e9, ns = 8 :

est0	act0	est1	act1
1.1e2	3.9e3	3.1e2	1.1e4
1.1e2	3.9e3	3.0e2	1.0e4
1.1e2	3.9e3	2.3e2	8.0e3
1.1e2	3.9e3	2.6e3	8.9e4

This problem becomes stiffer as parameter k is increased and this is reflected in the size of the actual errors which are small for k <= 15 but increase rapidly for larger k. But very good agreement is obtained between actual and estimated errors for k <= 25 even though for k = 25 the errors are not small. For k = 30 and k = 35 the actual and estimated errors agree in order of magnitude but for k = 40 the estimated errors are very inaccurate and too small.

TEST PROBLEM III :

(In this section a reference such as table 4 is to table A2.3.4).

Here we used an initial steplength of 0.01 each time with a Runge Kutta tolerance (RK) as given. The problem parameter k = 19 for all cases. In tables 1 to 3 the well conditioned BC (III.1) were used.

Table A2.3.1 : RK = 1e-4, cmax = 50e3, ns = 12 :

est0	act0	est1	act1
5.7e-24	1.3e-15	2.7e-7	8.6e-10
8.4e-10	1.2e-9	8.4e-10	1.2e-9
3.0e-9	1.5e-9	3.0e-9	1.5e-9

Table A2.3.2 : RK = 1e-3, cmax = 50e3, ns = 12 :

est0	act0	est1	act1
3.8e-24	1.1e-15	1.7e-5	5.0e-9
1.0e-9	3.1e-9	1.0e-9	3.1e-9
6.7e-10	3.9e-9	6.7e-10	3.9e-9

Table A2.3.3 : RK = 1e-2, cmax = 50e3, ns = 12 :

est0	act0	est1	act1
1.2e-22	2.4e-15	2.1e-5	7.3e-8
2.5e-8	2.8e-8	2.5e-8	2.8e-8
2.7e-8	3.4e-8	2.7e-8	3.4e-8

The results in tables 4 to 6 below were obtained using the

ill conditioned BC (III.1) :

Table A2.3.4 : RK = 1e-4, cmax = 50e3, ns = 12 :

est0	act0	est1	act1
5.7e-24	1.3e-15	3.0e-9	1.5e-9
8.4e-10	1.2e-9	8.4e-10	1.2e-9
3.0e-9	1.5e-9	3.6e-3	3.1e-5

Table A2.3.5 : RK = 1e-3, cmax = 50e3, ns = 12 :

est0	act0	est1	act1
3.8e-24	1.1e-15	6.7e-10	3.9e-9
1.0e-9	3.1e-9	1.0e-9	3.1e-9
6.7e-10	3.9e-9	2.2e-1	1.2e-4

Table A2.3.6 : RK = 1e-2, cmax = 50e3, ns = 12 :

est0	act0	est1	act1
1.2e-22	2.4e-15	2.7e-8	3.4e-8
2.5e-8	2.8e-8	2.5e-8	2.8e-8
2.7e-8	3.4e-8	2.6e-1	1.4e-3

The agreement between actual and estimated errors is not so good for this problem as for problem II - particularly for the ill conditioned BC cases. This is probably partly due to the the fact that for this problem (unlike problem II) the differential system matrix A is variable so that its derivatives will be involved in the calculations and these derivatives have some quite large components which are liable to magnify any errors incurred.

APPENDIX III : RESULTS FOR FACTORISATION METHODS

Here we give some of our numerical results obtained using Babuska's factorisation methods described in Chapter 5. In the following, program RIC used the Riccati transformation and programs ORT1 and ORTO the orthonormal transformation with and without the generalised inverse respectively. All integrations were performed with the variable step Runge Kutta RKF45, using an initial step-length of 0.01 in all cases and with a tolerance per unit step (RK) as stated. All results are given to an accuracy of two significant figures.

* indicates that the method failed because the integration step-length became too small (< 1e-9) in the backward sweep

* * indicates failure due to overflow in the calculation of function values during integration.

Problems I and II below are the test problems detailed in Chapter 6. We can thus compare the results of the factorisation methods given here with those obtained using our proposed iterative residual correction method in Chapter 6.

The results given are (as in Chapter 6) the maximum absolute error in the components of $[u(\alpha), u(\beta)]^T$ where $u(t)$ is the calculated LBVP solution and $[\alpha, \beta]$ is the problem interval.

Results :

Problem I :

(In this section a reference such as table 4 is to table A3.1.4).

The results in tables 1 to 4 were all obtained using the well conditioned BC (I.2).

Table A3.1.1 : j = 2, k = 3, BC (I.2) :

RK	RIC	ORT1	ORTO
1e-2	2.7e-4	6.9e-4	8.6e-4
1e-3	5.4e-5	9.5e-5	2.9e-5
1e-4	1.4e-5	2.5e-5	2.9e-5

Table A3.1.2 : j = 5, k = 10, BC (I.2) :

RK	RIC	ORT1	ORTO
1e-2	2.4e-4	2.6e-4	2.6e-4
1e-3	2.6e-4	2.4e-4	2.5e-4
1e-4	1.2e-5	1.2e-5	1.2e-5

Table A3.1.3 : j = 15, k = 20, BC (I.2) :

RK	RIC	ORT1	ORTO
1e-2	6.7e-4	6.7e-4	* *
1e-3	7.7e-5	7.7e-5	* *
1e-4	2.4e-6	2.4e-6	* *

Table A3.1.4 : j = 20, k = 30, BC (I.2) :

RK	RIC	ORT1	ORTO
1e-2	1.8e-3	9.4e-5	* *
1e-3	1.5e-4	1.5e-4	* *
1e-4	6.0e-6	5.9e-6	* *

We see that for these well conditioned LBVPs the results for the Riccati method and the orthonormal method with generalised inverse are very similar and, for RK = 1e-4, are acceptably accurate in all cases. Notice that because the problem is well conditioned there is no loss of accuracy as parameters j and k increase i.e. as the problem becomes stiffer. This is because the good condition of the LBVP with separated BCs ensures the stability of the auxiliary IVPs in their respective directions. The orthonormal method without the generalised inverse, however, gives results comparable to those obtained with RIC and with ORT1 for small values of parameters j and k but fails to produce a solution for $j > 5$, $k > 10$ confirming the need for the generalised inverse as found by Davey [3] and Meyer [4]. We also tested the factorisation methods on LBVPs that are not well conditioned. The results in tables 5 to 8 below were obtained using the ill conditioned BC (I.1) :

Table A3.1.5 : j = 2, k = 3, BC (I.1) :

RK	RIC	ORT1	ORTO
1e-2	3.9e-4	4.5e-4	5.7e-4
1e-3	2.8e-5	4.0e-5	4.6e-5
1e-4	4.8e-6	9.7e-6	1.0e-5

Table A3.1.6 : j = 5, k = 10, BC (I.1) :

RK	RIC	ORT1	ORTO
1e-2	4.9e-3	4.9e-3	1.0e-2
1e-3	4.2e-4	4.1e-4	4.1e-4
1e-4	7.5e-5	7.4e-5	7.4e-5

Table A3.1.7 : j = 15, k = 20, BC (I.1) :

RK	RIC	ORT1	ORTO
1e-2	2.9e0	2.9e0	*
1e-3	8.5e-1	8.5e-1	*
1e-4	1.0e-1	1.0e-1	*

Table A3.1.8 : j = 20, k = 30, BC (I.1) :

RK	RIC	ORT1	ORTO
1e-2	3.9e2	3.9e2	*
1e-3	2.0e2	2.0e2	*
1e-4	3.2e1	3.2e1	*
1e-5	3.1e0	3.1e0	*
1e-6	2.4e-1	2.4e-1	*

As with the well conditioned BC we see that there is good agreement between the results of programs RIC and ORT1 but

now the accuracy of the calculated solution deteriorates rapidly as the condition of the LBVP worsens : for the case j = 20, k = 30 even with a RK tolerance of 1e-6 the error incurred is unacceptably large. As might be expected program ORTO produces similar results to the other two methods for small values of parameters j and k for which the LBVP is not too badly conditioned but it fails completely for larger values. These inaccurate results were expected because, as with all of the 'double sweep' methods, correct decoupling of the differential system is essential to ensure the stability of both of the auxiliary IVPs in their respective directions and the good condition of the problem is necessary to ensure this.

Problem II :

(In this section a reference such as table 4 is to table A3.2.4).

The results in tables 1 to 4 were obtained using the BC as given for problem II in Chapter 6 for which the problem is ill conditioned :

Table A3.2.1 : k = 5 :

RK	RIC	ORT1
1e-2	2.7e-3	2.2e-3
1e-3	5.5e-4	7.5e-4
1e-4	1.4e-4	1.4e-4
1e-5	1.6e-5	1.6e-5

Table A3.2.2 : k = 10 :

RK	RIC	ORT1
1e-2	8.2e-1	1.8e0
1e-3	1.8e0	1.7e0
1e-4	4.5e-2	4.9e-2
1e-5	2.2e-3	2.0e-3

Table A3.2.3 : k = 15 :

RK	RIC	ORT1
1e-2	3.0e3	7.6e2
1e-3	3.9e1	3.1e1
1e-4	1.3e0	1.1e0
1e-5	6.4e-2	1.5e0

Table A3.2.4 : k = 20 :

RK	RIC	ORT1
1e-2	2.2e4	2.9e4
1e-3	1.9e4	3.1e4
1e-4	4.3e2	5.5e2
1e-5	2.2e1	2.5e1

This problem becomes stiffer and more ill conditioned as the parameter k is increased. As for the ill conditioned case of problem I above, there is generally good agreement between the results of the Riccati and the generalised inverse orthonormal method. But for k > 10 the calculated LBVP solutions are very inaccurate even for small RK tolerances because, as

stated above, as k increases the auxiliary IVPs become more unstable.

We also tested the factorisation methods on problem II as above but now with the following well conditioned BCs instead :

$$B_o = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad B_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Table A3.2.5 : k = 5 :

RK	RIC	ORT1
1e-2	1.9e-3	2.1e-3
1e-3	5.0e-5	8.5e-5
1e-4	2.2e-6	2.1e-6

Table A3.2.6 : k = 10 :

RK	RIC	ORT1
1e-2	9.0e-4	1.1e-3
1e-3	3.3e-5	3.3e-5
1e-4	2.7e-5	9.3e-6

Table A3.2.7 : k = 15 :

RK	RIC	ORT1
1e-2	9.5e-4	5.8e-4
1e-3	7.2e-5	1.5e-4
1e-4	3.4e-6	4.9e-6

Table A3.2.8 : k = 20 :

RK	RIC	ORT1
1e-2	3.0e-4	4.4e-4
1e-3	1.6e-4	1.8e-4
1e-4	1.1e-5	1.2e-5

As expected the errors incurred by both methods are now acceptably small and there is no significant loss in accuracy as parameter k increases.

The results given in this appendix confirm that the factorisation methods of Babuska and Majer [16] are efficient solvers of well conditioned LBVPs but they also indicate that these methods cannot be relied upon to produce accurate solutions to LBVPs which are at all ill conditioned. This emphasises the advantage of our proposed iterative residual correction method (as described in Chapter 6) in this respect. We gave there results for the ill conditioned cases of test problems I and II obtained with this correction method. These results show that accurate solutions can be computed for these difficult problems by our proposed iterative correction method.

REFERENCES

[1] : 'Decoupling and stability of algorithms for boundary value problems' : R.M.M. Mattheij (Siam Review 27/1 March 1985)

[2] : 'Invariant imbedding, the box scheme and an equivalence between them' : H.B. Keller and M. Lentini (Siam J. Num. Anal. 19/5 Oct 1982)

[3] : 'An automatic orthonormalisation method for solving stiff boundary value problems' : A. Davey (J. of Comput. Physics 51 1983)

[4] : 'Continuous orthonormalisation for boundary value problems : G. H. Meyer (J. of Comput. Physics 62/1, Jan 1986)

[5] : 'A Riccati Transformation method for solving linear BVPs' L. Dieci, M.R. Osborne, R.D. Russell (Siam J. Num. Anal. 25/5 Oct 1988)

[6] : 'The close relationships between methods for solving two point boundary value problems' : M. Lentini, M.R. Osborne, R.D. Russell (Siam J. Num. Anal. 22/2 April 1985)

[7] : 'The conditioning of linear boundary value problems' R.M.M Mattheij (Siam J. Num. Anal. 19/5 Oct 1982)

[8] : 'On the removal of the singularities from the Riccati method' : A.Davey (J. Comput. Physics 30,1979)

[9] : 'A numerical method for linear two point boundary value problems using compound matrices' : B.S. Ng, W.H. Reid (J. of Comput. Physics 33, 1979)

[10] : 'Stable continuous orthonormalisation techniques for linear boundary value problems' : P.M. van Loon, R.M. Mattheij (Eindhoven University of Technology, Computing Centre Note 27)

[11] : 'The Riccati transformation in the solution of boundary value problems' : M.R. Osborne, R.D. Russell (Siam J. of Num. Anal. 23/5, Oct 1986)

[12] : 'Numerical solution of boundary value problems for ordinary differential equations' : U.M. Ascher, R.M.M. Mattheij R.D. Russell (Prentice Hall Series in Computational Mathematics

[13] : 'On dichotomy and well conditioning in BVP' :F.R. de Hoog & R. M.M. Mattheij (Siam J. of Num. Anal. 24/1, Feb 1987)

[14] : 'The factorisation method for two point boundary value problems for ODEs and its relation to the finite difference method' : I. Babuska & V. Majer (Proc. Centre for Mathematical Analysis of the Australian National Univ.)

[15] : 'On the numerical solution of difficult boundary value problems' : A.Davey (J. Comput. Physics 35/1 March 1980)

[16] : 'The factorisation method for the numerical solution of two point boundary value problems for linear ODEs' : I. Babuska & V. Majer (Siam J. Num. Anal. 24/6 Dec 1987)

[17] : 'Riccati transformation : when and how to use' : P. van Loon (Progress in Scientific Computing, 5, 1985)

[18] : 'An efficient algorithm for solving general linear two point BVPs' : R.M.M. Mattheij & G.W.M. Staarink (Siam J. Sci. Stat. Comput. 5/4, Dec 1984)

[19] : 'The numerical solution of linear boundary value problems : S.D. Conte (Siam Review, 8/3, July 1966)

[20] : 'Riccati and other methods for singularly perturbed BVPs' : L. Dieci & R.D. Russell (LCCR TR 86-2, Dept. Maths., Simon Fraser Univ.)

[21] : 'Continuous decoupling transformations for linear boundary value problems' : P.M. van Loon (Ph.D thesis 1987, Eindhoven Univ. of Technology)

[22] : 'Computational methods in ordinary differential equations : J.D. Lambert (J. Wiley, 1973)

[23] : 'A classification and survey of numerical methods for boundary value problems in ordinary differential equations' : Z. Aktas & H.J. Stetter (International J. for Numerical Methods in Engineering, Vol. 11, 1977)

[24] : 'Numerical methods' : N.S. Bakhvalov (MIR, Moscow 1977)

[25] : 'Numerical solution of Orr-Sommerfield type equations' : L.O'C. Drury (J. Comput. Physics, 37, 1980)

[26] : 'Criteria for Mesh Selection in Collocation Algorithms for Ordinary Differential Boundary Value Problems' : K. Wright, A.H.A. Ahmed, A.H. Seleman (Technical Report Series No. 289, July 1989, Comp. Lab., Univ. of Newcastle upon Tyne).