# SOLUTION OF GENERAL TPBVP'S IN OPTIMAL CONTROL USING THE SENSITIVITY METHOD

APPROVED BY

Doç. Dr. Yorgo Istefanopulos
(Thesis Advisor)

Dr. M. Akif Eyler
(Co-advisor)

Doç. Dr. Bülent Sankur

Date: ../../....

# ACKNOWLEDGEMENTS

This study was conducted under the supervision of Doç. Dr. Yorgo Istefanopulos and Dr. M. Akif Eyler to whom I wish to express my sincere thanks for their invaluable guidance, encouragement and support.

I am also greatful to Doç. Dr. Bülent Sankur for serving on my thesis comittee.

G.M.

# ABSTRACT

In this thesis, general TPBVP's in Optimal Control have been analyzed using the sensitivity method. A solution method, which uses the trajectory sensitivities to solve general TPBVP's, is developed. Existence of the solutions for the boundary value problems with linear boundary conditions is also studied.

The proposed method allows us to parametrize the boundary conditions. If the analytic expressions for the parametric functions are not available then it determines their numerical values. In this way, the method easily converts the original problem into an initial value problem. Then by changing the parameters and using the trajectory sensitivities, it calculates the true parameter values that satisfy the problem constraints, hence finds a solution for the given problem.

# ÖZET


Bu tezde, en iyi denetimde ortaya çıkan genel iki-nokta sınır değer problemleri duyarlılık yöntemi kullanılarak incelenmiş, yörünge duyarlılıklarını kullanarak bu tip problemleri çözebilen bir yöntem geliştirilmiş ve doğrusal sınır koşullu sınır değer problemlemlerinin çözümlerinin varlık sorununa değinilmiştir.

Önerilen yöntem, sınır koşullarını değiştirgenler kullanarak yeniden tanımlamamıza olanak verir. Eğer değiştirgen işlevler için çözümsel ifadeler bulmak olası değilse, yöntem kendisi bu ifadelerin sayısal değerlerini belirler. Bu şekilde, sınır-değer problemini kolayca ilk-değer problemine dönüştürür. Daha sonra ise değiştirgenleri değiştirerek ve yörünge duyarlılıklarını kullanarak, değiştirgenlerin problem kısıtlarını sağlayan doğru değerlerini hesaplar, dolayısıyla problemin çözümünü bulmuş olur.

# TABLE OF CONTENTS

# LIST OF TABLES

# I. INTRODUCTION

When we apply the calculus of variations to the optimal control problems, we generally end up with nonlinear two point boundary value problems. Unless the resulting problems are quite simple, we have to use a numerical technique to determine the optimal control, and trajectories.

The nonlinear two point boundary value problems encountered in a large class of optimal control problems can be summarized as finding 2n state and costate variables and m control inputs, while satisfying the 2n state and costate differential equations, the m optimality conditions, and the initial and the final conditions.

Most of the numerical methods for the solution of such problems necessarily use iterative procedures. Generally a nominal solution is chosen that satisfies some but not all of the necessary conditions, then it is modified so that the rest of the conditions are also satisfied. The most popular methods of this kind are Gradient Methods, Quasilinearization Methods, and Variation of Extremals.

Gradient Methods were developed to overcome the initial guess difficulty associated with direct integration methods. In this approach nominal solution is chosen to satisfy the 2n differential equations, and is characterized by the iterative algorithms for improving the estimates of control inputs so as to come closer to satisfying the optimality and boundary conditions. Although, it is considerably easy to start this method, as an extremal is approached, the gradient becomes small and the method has a tendency to converge slowly.

One version of Quasilinearization Methods chooses nominal functions for states and costates that satisfy as many of the boundary conditions as possible. Then the nominal control inputs are determined by use of optimality conditions. The state and costate differential equations are linearized about the nominal solutions, and successive linear two point boundary value problems are solved to modify the solution until it satisfies the state and costate differential equations. The sequence of solutions of the linearized equations of states and costates, with a rate that is at least quadratic, converges to the desired solutions, if the norm of the deviation of the initial guess from the desired solutions is sufficiently small.

In the Method of Variations of Extremals, every trajectory generated by the algorithm satisfies the state and costate differential equations, hence is an extremal. This method is characterized by iterative algorithms for improving the estimates of the unspecified initial (or final) conditions so as to satisfy the specified final (or initial) conditions. This is achived by finding the transition matrix between the unspecified boundary conditions at one end and the specified boundary conditions at the other end.

The main difficulty of this method is getting started, in other words finding a first estimate of the unspecified conditions at one end. The reason for this difficulty is that the extremal solutions are often very sensitive to small changes in the unspecified boundary conditions and very small perturbations may significantly increase the effects of the inaccuracies caused by numerical integration, and truncation and round off errors. These difficulties can be avoided to some extent by the method we have offered to evaluate the transition matrices which we call sensitivity matrices,in other words, the method determines the differential equations that the sensitivity matrices satisfy. Those matrices can be obtained all over the interval by integrating these differential equations simultaneously with the state and costate differential equations. The appropriate initial conditions for the sensitivity matrices may be calculated very accurately using the method of parametrization of the nonlinear boundary conditions.

Thus in thesis, we have developed a solution method using trajectory sensitivities to determine the optimal control and trajectories. In Chapter 2, the necessary conditions for optimal control problem are given, the various boundary conditions that may occur are discussed, and the general formulation of two point boundary value problems is presented.

Chapter 3 discusses the solution method based on trajectory sensitivities. First the idea is introduced by a special case, then the general case is considered and the parametrization of nonlinear boundary conditions,to determine the final values of states, costates, and the sensitivity matrices, are explained. Finally it is shown that, if parametric functions do not exist analytically, then their numerical values and derivatives with respect to parameters can be determined.

In Chapter 4, we discuss the two point boundary value problems with linear boundary conditions, and show that the problem can be reduced to finding the roots of a system of transcendental equations. Necessary conditions for the existence of the solution and convergence of the iterative method for this special case are examined.

A set of numerical examples are studied in Chapter 5. These examples illustrate how the method given in Chapter 3 is applied to a particular problem.

# II. GENERAL FORMULATION OF TPBVP FOR SOLVING OPTIMAL CONTROL PROBLEMS

## 2.1 The Optimal Control Problem

The Optimal Control Problem is to find an admissible $u^*(t)$ that causes the system

$$\dot{x}(t) = a(x(t),u(t),t) \qquad 2.1.1$$

to follow an admissible trajectory $x(t)$ that mininizes the performance measure

$$J(u) = \phi(x(t_f),t_f) + \int_{t_o}^{t_f} \Omega(x(t),u(t),t)\, dt \qquad 2.1.2$$

where $\phi(x(t_f),t_f)$ is the final state penalty, and $\Omega(x(t),u(t),t)$ is an appropriate cost function.We shall assume that the admissible state and control regions are not bounded, and initial conditions $x(t_o)$ and the initial time $t_o$ are specified. As usual $x(t)$ is the n-dimensional state vector and $u(t)$ is the m-dimensional vector of control inputs.

## 2.2 Necessary Conditions for Optimal Control

If we apply variational methods to the optimal control problem given in equations 2.1.1 and 2.1.2, then the necessary conditions for optimal control are:

$$\dot{x}^*(t) = \frac{\partial H}{\partial y}(x^*(t), y^*(t), u^*(t), t) \qquad 2.2.1.a$$

$$\dot{y}^*(t) = -\frac{\partial H}{\partial x}(x^*(t), y^*(t), u^*(t), t) \qquad \text{for all } t \in [t_o, t_f] \qquad 2.2.1.b$$

$$0 = \frac{\partial H}{\partial u}(x^*(t), y^*(t), u^*(t), t) \qquad 2.2.1.c$$

and

$$\left( \frac{\partial \phi}{\partial x}(x^*(t_f), t_f) - y^*(t_f) \right)^T \delta x_f + \left( H(x^*(t_f), y^*(t_f), u^*(t_f), t_f) + \frac{\partial \phi}{\partial t}(x^*(t_f), t_f) \right) \delta t_f = 0 \qquad 2.2.2$$

where

$$H(x(t), y(t), u(t), t) \triangleq \Omega(x(t), u(t), t) + y^T(t)\,(a(x(t), u(t), t)) \qquad 2.2.3$$

is the Hamiltonian, and

$$y(t) \triangleq (y_1(t), \ldots, y_n(t))^T \qquad 2.2.4$$

is the n-dimensional costate vector. Equation 2.2.2 is also known as the natural boundary conditions, and $\delta x_f$ $\delta t_f$ are the variations introduced on $x(t_f)$ and $t_f$, respectively throughout the derivations of equations 2.2.1 and 2.2.2. Since the derivation of the necessary conditions is beyond the scope of this work, we give only the results here. For those who are interested, details can be found in Kirk (1970) pp 185-188.

Notice that these necessary conditions consist of a set of 2n, first order differential equations - the state and co-state equations 2.2.1.a and b - and a set of m algebric relations - 2.2.1.c - which must be satisfied throughout the interval $[t_o, t_f]$. The solutions of the state and costate equations will contain 2n constants of integration. To evaluate these constants we will use the n equations given at the initial time

$$\overset{*}{x}(t_o) = x_o \ , \qquad 2.2.5$$

and an additional set of n (or n+1), relationships from equation 2.2.2 which are all at the final time. Thus we have to solve a two point boundary value problem to determine the optimal control $\overset{*}{u}(t)$.

Before stating the general formulation of two point boundary value problems we are going to deal with, let us discuss the various boundary conditions that may occur as a result of equation 2.2.2.

## 2.3 Boundary Conditions

To determine the boundary conditions of a particular problem is a matter of making the appropriate substitutions in equation 2.2.2. In all cases we will assume that we have the n equations $x(t_o) = x_o$.

PROBLEMS WITH FIXED FINAL TIME: If the final time $t_f$ is specified, then the variation $\delta t_f = 0$, for the following cases.

CASE i. (Final state specified) Since $x(t_f)$ is also specified then $\delta x_f = 0$ and equation 2.2.2 is automatically satisfied.

Thus the required n equations are

$$x^*(t_f) = x_f \tag{2.3.1}$$

CASE ii. (Final state free) In this case, variation $\delta x_f$ will no more vanish. So we have n equations

$$\frac{\partial \phi}{\partial x}(x^*(t_f)) - y^*(t_f) = 0 \tag{2.3.2}$$

to satisfy equation 2.2.2.

CASE iii. (Final state lying on a surface defined by $m(x(t)) = 0$ ) Now we have n state variables and $1 \leqslant k \leqslant n-1$ relationships that the states must satisfy at $t = t_f$, namely

$$m(x(t)) = \begin{bmatrix} m_1(x(t)) \\ \vdots \\ m_k(x(t)) \end{bmatrix} = 0 \tag{2.3.3}$$

where each component of m represents a hypersurface in the n-dimensional state space. Thus the final state lies on the intersection of these hypersurfaces, and $\delta x_f$ is tangent to each of the hypersurfaces at point $(x^*(t_f), t_f)$. This means that $\delta x_f$ is normal to each of the gradient vectors

$$-\frac{\partial m_1}{\partial x}(x^*(t_f)), \ldots\ldots, -\frac{\partial m_k}{\partial x}(x^*(t_f)) \tag{2.3.4}$$

which are assumed to be linearly independent, and normal also to their linear combination. Thus

$$v^T \frac{\partial m}{\partial x}(x^*(t_f))\, \delta x_f = 0 \tag{2.3.5}$$

where

$$v = (v_1, \ldots\ldots, v_k) \tag{2.3.6}$$

is a k-dimensional coefficient vector. Since

$$\left( \frac{\partial\phi}{\partial x} (\overset{*}{x}(t_f)) - \overset{*}{y}(t_f) \right)^T \delta x_f \triangleq v^T \frac{\partial m}{\partial x} (\overset{*}{x}(t_f)) \, \delta x_f = 0 \qquad 2.3.7$$

we have n equations

$$\frac{\partial\phi}{\partial x} (\overset{*}{x}(t_f)) - \overset{*}{y}(t_f) = \left( \frac{\partial m}{\partial x} (\overset{*}{x}(t_f)) \right)^T v \qquad 2.3.8$$

and k equations

$$m(\overset{*}{x}(t_f)) = 0 \qquad 2.3.9$$

to determine the integration constants and v, and to satisfy the equation 2.2.2.

PROBLEMS WITH FREE FINAL TIME: If the final time is free, then the assumption $\delta t_f = 0$ is no more valid.

CASE iv. (Final state fixed) The appropriate substitution in equation 2.2.2 is $\delta x_f = 0$. So we obtain (2n+1)st relationship, to determine the final time $t_f$, as

$$H(\overset{*}{x}(t_f),\overset{*}{y}(t_f),\overset{*}{u}(t_f),t_f) + \frac{\partial\phi}{\partial t} (\overset{*}{x}(t_f),t_f) = 0 \qquad 2.3.10$$

whereas the other 2n equations are $\overset{*}{x}(t_o)=x_o$, and

$$\overset{*}{x}(t_f) = x_f \qquad 2.3.11$$

CASE v. (Final state free) In this case both $\delta x_f$ and $\delta t_f$ are arbitrary, and furthermore, independent. So to satisfy equation 2.2.2 we must have their coefficients to vanish. Thus

$$\frac{\partial\phi}{\partial x} (\overset{*}{x}(t_f),t_f) - \overset{*}{y}(t_f) = 0 \qquad 2.3.12$$

and

$$H(\overset{*}{x}(t_f),\overset{*}{y}(t_f),\overset{*}{u}(t_f),t_f) + \frac{\partial\phi}{\partial t}(\overset{*}{x}(t_f),t_f) = 0 \qquad 2.3.13$$

CASE vi. ($x(t_f)$ lies on the moving point $\theta(t)$) Now, $\delta x_f$ and $\delta t_f$ are related by

$$\delta x_f \doteq \left( \frac{d\theta}{dt}(t_f) \right) \delta t_f \qquad 2.3.14$$

to a first order, making this substitution into equation 2.2.2 yields

$$H(\overset{*}{x}(t_f),\overset{*}{y}(t_f),\overset{*}{u}(t_f),t_f) + \frac{\partial\phi}{\partial t}(\overset{*}{x}(t_f),t_f)$$

$$+ \left( \frac{\partial\phi}{\partial x}(\overset{*}{x}(t_f),t_f) - \overset{*}{y}(t_f) \right)^T \left( \frac{d\theta}{dt}(t_f) \right) = 0 \qquad 2.3.15$$

This gives one equation and remaining n required relationships are

$$\overset{*}{x}(t_f) = \theta(t_f) \qquad 2.3.16$$

CASE vii. (Final state lying on the surface defined by $m(x(t))=0$) The reasoning used in CASE iii with fixed final time also applies here. Since the surface where the final state lies does not move, i.e., has no explicit dependence on t, then the variation in $x(t_f)$, namely $\delta x_f$ is independent of $\delta t_f$. Thus the required n+k+1 equations are

$$\frac{\partial\phi}{\partial x}(\overset{*}{x}(t_f),t_f) - \overset{*}{y}(t_f) = \left( \frac{\partial m}{\partial x}(\overset{*}{x}(t_f)) \right)^T v \qquad 2.3.17$$

$$m(\overset{*}{x}(t_f)) = 0 \qquad 2.3.18$$

and

$$H(\overset{*}{x}(t_f),\overset{*}{y}(t_f),\overset{*}{u}(t_f),t_f) + \frac{\partial\phi}{\partial t}(\overset{*}{x}(t_f),t_f) = 0 \qquad 2.3.19$$

CASE viii. (Final state lying on the moving surface defined by $m(x(t),t)=0$) Notice that $\delta t_f$ now influences the admissible values of $\delta x_f$, that is, to remain on the surface $m(x(t),t)=0$, the value of $\delta x_f$ does depend on $\delta t_f$. The vector with components $\delta x_f$ and $\delta t_f$ must be contained in a plane tangent to the surface at point $(\overset{*}{x}(t_f),t_f)$. This means that

$$\begin{bmatrix} \delta x_f \\ \cdots \\ \delta t_f \end{bmatrix} \qquad 2.3.20$$

is normal to each of the gradient vectors

$$\begin{bmatrix} \frac{\partial m_1}{\partial x}(\overset{*}{x}(t_f),t_f) \\ \cdots\cdots \\ \frac{\partial m_1}{\partial t}(\overset{*}{x}(t_f),t_f) \end{bmatrix} \cdots\cdots \begin{bmatrix} \frac{\partial m_k}{\partial x}(\overset{*}{x}(t_f),t_f) \\ \cdots\cdots \\ \frac{\partial m_k}{\partial t}(\overset{*}{x}(t_f),t_f) \end{bmatrix} \qquad 2.3.21$$

which are assumed to be linearly independent,and normal also to their linear combination. Rewriting the equation 2.2.2 as

$$\begin{bmatrix} \frac{\partial \phi}{\partial x}(\overset{*}{x}(t_f),t_f) - \overset{*}{y}(t_f) \\ \cdots\cdots\cdots \\ H(\overset{*}{x}(t_f),\overset{*}{y}(t_f),\overset{*}{u}(t_f),t_f) + \frac{\partial \phi}{\partial t}(\overset{*}{x}(t_f),t_f) \end{bmatrix}^T \begin{bmatrix} \delta x_f \\ \cdots \\ \delta t_f \end{bmatrix}$$

$$\triangleq v^T \begin{bmatrix} \frac{\partial m}{\partial x}(\overset{*}{x}(t_f),t_f) & \frac{\partial m}{\partial t}(\overset{*}{x}(t_f),t_f) \end{bmatrix} \begin{bmatrix} \delta x_f \\ \cdots \\ \delta t_f \end{bmatrix} = 0 \qquad 2.3.22$$

yields the necessary n+k+1 equations

$$\frac{\partial \phi}{\partial x}(\overset{*}{x}(t_f),t_f) - \overset{*}{y}(t_f) = \begin{bmatrix} \frac{\partial m}{\partial x}(\overset{*}{x}(t_f),t_f) \end{bmatrix}^T v \qquad 2.3.23$$

$$H(\overset{*}{x}(t_f),\overset{*}{y}(t_f),\overset{*}{u}(t_f),t_f) + \frac{\partial \phi}{\partial t}(\overset{*}{x}(t_f),t_f)$$

$$= \begin{bmatrix} \frac{\partial m}{\partial t}(\overset{*}{x}(t_f),t_f) \end{bmatrix}^T v \qquad 2.3.24$$

and

$$m(\overset{*}{x}(t_f),t_f) = 0 \qquad 2.3.25$$

## 2.4 General Formulation

Now let us try to state the general formulation of two point boundary value problems such that it covers all cases we have discussed above.

Since the technique we have offered determines an open-loop control, we will assume that equation 2.2.1.c can be solved to obtain an expression for $\overset{*}{u}(t)$ in terms of $\overset{*}{x}(t)$, $\overset{*}{y}(t)$, and t, that is

$$\overset{*}{u}(t) = z(\overset{*}{x}(t),\overset{*}{y}(t),t) \qquad 2.4.1$$

If this expression is substituted into equations 2.2.1.a, and 2.2.1.b, we have a set of 2n first order ordinary differential equations (so called the reduced differential equations) involving only $\overset{*}{x}(t)$, $\overset{*}{y}(t)$, and t, namely

$$\dot{x}(t) = f(x(t),y(t),t)^{\dagger} \qquad 2.4.2$$

$$\dot{y}(t) = h(x(t),y(t),t) \qquad 2.4.3$$

where

$$f(x(t),y(t),t) \triangleq \frac{\partial H}{\partial y}(x(t),y(t),z(x(t),y(t),t),t)$$
$$h(x(t),y(t),t) \triangleq -\frac{\partial H}{\partial x}(x(t),y(t),z(x(t),y(t),t),t) \qquad 2.4.4$$

† From now on we will not use $*$ to indicate $(\ )^*$ is an optimal quantity

The boundary conditions for these differential equations (which are of nonlinear nature) can be generalized as

$$x(t_o) = x_o \qquad 2.4.5.a$$

$$m(x(t_f),t_f) = 0 \qquad 2.4.5.b$$

$$l(x(t_f),y(t_f),v,t_f) = 0 \qquad 2.4.5.c$$

$$g(x(t_f),y(t_f),v,t_f) = 0 \qquad 2.4.5.d$$

where equation 2.4.5.a is given n relationships. Equation 2.4.5.b forms $0 \leqslant k \leqslant n-1$ relationship(s) to define the surface where $x(t_f)$ will lie at the final time. In equations 2.4.5.c and 2.4.5.d, l and g are an n-dimensional vector, and a scalar valued functions respectively, to satisfy the equation 2.2.2.

Notice that, a very little effort is necessary to put the boundary conditions we have discussed into the form of equation 2.4.5.

# III. THE SOLUTION METHOD USING TRAJECTORY SENSITIVITIES

## 3.1 Introduction

As we see in the previos chapter, in general the variational approach to optimal control problems leads to a nonlinear two point boundary value problem that cannot be solved analytically to obtain the optimal control. So, somehow we have to use a numerical technique to determine the optimal control and trajectory functions.

If the boundary conditions were given, all at either $t_o$ or $t_f$, we could numerically integrate the reduced differential equations to obtain $x(t)$, $y(t)$ for $t \in [t_o, t_f]$. The optimal control history could then be found by substituting $x(t)$, $y(t)$ into equation 2.4.1. Unfortunately the boundary conditions are split, so this method cannot be directly applied. But the iterative method we have developed somehow, makes use of this fact. In other words, we make an initial guess for $x(t_f)$, $y(t_f)$, $v$, and $t_f$ to convert the problem into an initial value problem in which equations 2.4.5.b through 2.4.5.d are satisfied. Then we integrate the differential equations 2.4.2 and 2.4.3 backwards numerically, to obtain a value for $x(t_o)$, which will most probably be different from $x_o$. Then this value is used to adjust the initial guess in an attempt to make next value of $x(t_o)$ come closer to $x_o$. If these steps are repeated and the iterative procedure converges, then the optimal trajectories and control satisfying all boundary conditions will eventually be found.

## 3.2 A Special Case

To introduce the idea of the method, let us consider the following special case of two point boundary value problem given by equations 2.4.2 through 2.4.5.

$$\dot{x}(t) = f(x(t),y(t),t) \tag{3.2.1}$$

$$\dot{y}(t) = h(x(t),y(t),t) \tag{3.2.2}$$

$$x(t_o) = x_o \quad , \quad y(t_f) = y_f \tag{3.2.3}$$

where $x(t)$, and $y(t)$ are n-vectors, $x_o$, $y_f$ are known and $t_f$ is fixed. Furthermore let us assume that $f(x(t),y(t),t)$ and $h(x(t),y(t),t)$ are bounded and continuous functions of $x(t)$, $y(t)$, and $t$, and that satisfy a uniform Lipschitz condition to ensure a unique solution of the initial value problem for a given set of initial (or as in our case final) conditions.

Suppose we make an initial guess for $x(t_f)$ then we have

$$x(t_f) = x_f \quad , \quad y(t_f) = y_f \tag{3.2.4}$$

and the equation 3.2.4 together with the equations 3.2.1 and 3.2.2 convert the original problem into an initial value problem that can be solved, e.g. by integration, easily. But now, the calculated $x(t_o)$ will probably differ from the desired value $x_o$. So, somehow we have to adjust our initial guess $x_f$ to catch the desired value $x_o$.

Certainly, introduction of $x_f$ causes a parametric dependence of $x(t)$ and $y(t)$ on $x_f$, that is

$$x(t) \triangleq x(t;x_f) \quad , \quad y(t) \triangleq y(t;x_f) \tag{3.2.5}$$

Now also suppose that, we introduce small changes in $x_f$. Since the assumptions made on f and h ensure that the solutions $x(t)$ and $y(t)$ are continuously differentiable with respect to $x_f$, the effect of such changes on $x(t)$ and $y(t)$ can be predicted by the trajectory sensitivities. Furthermore the sensitivities can be calculated while the initial value is being integrated.

Let us define the sensitivity matrices as,

$$R(t) \triangleq \frac{\partial x(t;x_f)}{\partial x_f} \tag{3.2.6}$$

$$S(t) \triangleq \frac{\partial y(t;x_f)}{\partial x_f} \tag{3.2.7}$$

then $R(t)$ and $S(t)$ happens to be nxn matrices, and

$$\frac{d}{dt} R(t) = \frac{d}{dt} \frac{\partial x(t;x_f)}{\partial x_f} \tag{3.2.8}$$

Since t and $x_f$ are not dependent, the order of the differentiation in equation 3.2.8 can be changed, yielding

$$\dot{R}(t) = \frac{\partial}{\partial x_f} \dot{x}(t;x_f) \qquad 3.2.9$$

or substituting $\dot{x}(t)$ from equation 3.2.1 into equation 3.2.9, we obtain

$$\dot{R}(t) = \frac{\partial}{\partial x_f} f(x(t;x_f),y(t;x_f),t) \qquad 3.2.10$$

Momentarily omitting the arguments, and applying the chain rules of partial differentiation,

$$\dot{R}(t) = \frac{\partial f}{\partial x}\frac{\partial x}{\partial x_f} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial x_f}$$

or

$$\dot{R}(t) = \frac{\partial f}{\partial x} R(t) + \frac{\partial f}{\partial y} S(t) \qquad 3.2.11$$

Similarly

$$\dot{S}(t) = \frac{\partial h}{\partial x} R(t) + \frac{\partial h}{\partial y} S(t) \qquad 3.2.12$$

To integrate the equations 3.2.11 and 3.2.12 we also have to know the values of the sensitivity matrices at $t_f$, there follows from the definition of $R(t)$,

$$R(t_f) = \left.\frac{\partial x(t;x_f)}{\partial x_f}\right|_{t=t_f} \qquad 3.2.13$$

or first evaluating $x(t;x_f)$ at $t_f$ and then taking the derivative with respect to $x_f$, we have

$$R(t_f) = \frac{\partial x(t_f;x_f)}{\partial x_f} \qquad 3.2.14$$

But $x(t_f;x_f)$ is nothing but $x_f$, thus

$$R(t_f) = I \tag{3.2.15}$$

and similarly $S(t_f)$ can be found as

$$S(t_f) = 0 \tag{3.2.16}$$

If we want to write these results in a more compact form, we obtain

$$\begin{bmatrix} \dot{R} \\ \dot{S} \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y} \end{bmatrix} \begin{bmatrix} R \\ S \end{bmatrix} \tag{3.2.17}$$

and

$$\begin{bmatrix} R(t_f) \\ S(t_f) \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} \tag{3.2.18}$$

Now, we can integrate the system of equations 3.2.1, 3.2.2 and 3.2.17 using 3.2.4 and 3.2.18 to obtain a value for $x(t_o)$ and its sensitivity for the variations in $x_f$, namely $R(t_o)$. Suppose that we introduce a finite change $\Delta x_f$ on the initial guess $x_f$, then the resulting change in $x(t_o)$ can be approximated by

$$\Delta x(t_o) \cong R(t_o)\,\Delta x_f \tag{3.2.19}$$

Since our aim is to make $x(t_o)$ equal to $x_o$, the desired change in $x(t_o)$ is $\Delta x(t_o) = x(t_o) - x_o$, and the necessary change in $x_f$ can be found approximately by using the formula

$$\Delta x_f \cong R^{-1}(t_o)\,(x(t_o) - x_o) \tag{3.2.20}$$

provided that $R(t_o)$ is invertable. If the system given by equations 3,2.1 and 3.2.2 is linear (see Appendix B) or the initial guess is very close to the true value, then it is possible to have $x(t_o)$ to be very close to $x_o$, after making the change on $x_f$ found from equation 3.2.20, and solving the initial value problem again. But in general this is not the case and a few more iterations are necessary.

ALGORITHM

Now we can summarize the method described above as an algorithm

1. Choose a norm in $E^n$ and an accuracy $\delta_x$.
2. Make an initial guess for $x_f$.
3. Solve the initial value problems given by equations 3.2.1, 3.2.2 and 3.2.4 for states and costates, and the one given by 3.2.17 and 3.2.18 for sensitivities from $t_f$ to $t_o$. Store only the values $x(t_o)$ and nxn matrix $R(t_o)$.
4. Check to see if the termination criterion $\|x(t_o)-x_o\|<\delta_x$ is satisfied. If this is the case, use the final iterated value of $x_f$ to reintegrate the system equations 3.2.1 and 3.2.2 while evaluating the optimal control history using $u^*(t)=z(x^*(t),y^*(t),t)$. If the stopping criterion is not satisfied, then determine the new value of $x_f$, using

$$x_f = x_f - R^{-1}(t_o)\ (x(t_o) - x_o) \qquad 3.2.21$$

and return to step 3.

## 3.3 General Case with Analytic Parametric Functions

Now let us turn back to general Two Point Boundary Value Problem that we have stated at the end of Chapter 2, and for the sake of completeness let us restate it again. We have a set of 2n first order differential equations, namely

$$\dot{x}(t) = f(x(t),y(t),t) \qquad 3.3.1$$

and

$$\dot{y}(t) = h(x(t),y(t),t) \qquad 3.3.2$$

where $x(t)$, and $y(t)$ are n-vectors and $f(x(t),y(t),t)$, and $h(x(t),y(t),t)$ are bounded and continuous functions of $x(t),y(t)$, and t. Furthermore they satisfy a uniform Lipschitz condition.

We have also a set of boundary conditions

$$x(t_o) = x_o \qquad 3.3.3$$

$$m(x(t_f),t_f) = 0 \qquad 3.3.4.a$$

$$l(x(t_f),y(t_f),v,t_f) = 0 \qquad 3.3.4.b$$

$$g(x(t_f),y(t_f),v,t_f) = 0 \qquad 3.3.4.c$$

where $t_o$ and $t_f$ are the initial and the final times respectively. It is assumed that $t_o$ and $x_o$ are given a priori. $m(x(t_f),t_f)$ is an $0 \leqslant k \leqslant n-1$ dimensional vector function to define the surface where $x(t_f)$ will lie at $t=t_f$.
Of course if k=0 then there will be no constraint on $x(t_f)$ defined by m, so k-dimensional coefficient vector v will

also disappear from equations 3.3.4.b and 3.3.4.c. $l$ and $g$ are an n-dimensional vector function and a scalar valued function respectively to satisfy the generalized natural boundary conditions. If $t_f$ is also fixed, then $g$ will necessarily disappear. We also assume that $m$, $l$, and $g$ are continuous functions of their arguments.

The first thing we have to do, is to somehow determine $x(t_f)$, $y(t_f)$, $v$, and $t_f$ to satisfy equations 3.3.4. But we have $2n+k+1$ unknowns with only $n+k+1$ equations. Thus we can determine $n+k+1$ of them interms of remaining $n$ unknowns, at least in principle. Alternatively we can parametrize the equations 3.3.4 such that

$$x(t_f) \triangleq x_f(p) \qquad 3.3.5.a$$

$$y(t_f) \triangleq y_f(p) \qquad 3.3.5.b$$

$$v \triangleq v(p) \qquad 3.3.5.c$$

$$t_f \triangleq T(p) \qquad 3.3.5.d$$

where $p$ is an n-dimensional parameter vector, $x_f(p)$, $y_f(p)$, $v(p)$, and $T(p)$ are continuous functions of $p$, and equations 3.3.4 are satisfied for all values of $p$.

Now a question may come to mind that "Is it always possible to find those parametric functions in equations 3,3.5 analytically?". Answer to this question is, unfortunately and of course "no". But we leave the answer of the question "What shall we do then?" to the next section and at the moment, assume that they exist.

Suppose that we make an initial guess for p, then through equations 3.3.5 we have the necessary set of final values $x(t_f)$, $y(t_f)$, and $t_f$ to integrate the differential equations 3.3.1 and 3.3.2 as an initial value problem.

Since the different values of p, give different set of final values, thus different solutions for the initial value problem, then x(t), and y(t) depend on final values, i.e.,

$$x(t) \triangleq x(t;x_f(p),y_f(p),v(p),T(p)) \qquad 3.3.6.a$$

$$y(t) \triangleq y(t;x_f(p),y_f(p),v(p),T(p)) \qquad 3.3.6.b$$

and consequently depend on p parametrically, that is

$$x(t) \triangleq x(t;p) \quad , \quad y(t) \triangleq y(t;p) \qquad 3.3.7$$

Furthermore, they are continuously differentiable with respect to p. Differentiability is guaranteed by the assumptions on f and h.

The second thing is to determine the trajectory sensitivities in the sense that we can predict the effects of small changes in p, on the system response. We define the sensitivity matrices in a same manner as we did before. Let

$$R(t) \triangleq \frac{\partial x(t;p)}{\partial p} \qquad 3.3.8$$

and

$$S(t) \triangleq \frac{\partial y(t;p)}{\partial p} \qquad 3.3.9$$

where R(t) and S(t) happen to be nxn matrices. The differential equations which R(t) and S(t) satisfy can be obtained by taking the derivative of the equations 3.3.1 and 3.3.2 with respect to p, as

$$\frac{\partial \dot{x}(t;p)}{\partial p} = \frac{\partial f}{\partial x}\frac{\partial x(t;p)}{\partial p} + \frac{\partial f}{\partial y}\frac{\partial y(t;p)}{\partial p} \qquad 3.3.10.a$$

$$\frac{\partial \dot{y}(t;p)}{\partial p} = \frac{\partial h}{\partial x}\frac{\partial x(t;p)}{\partial p} + \frac{\partial h}{\partial y}\frac{\partial y(t;p)}{\partial p} \qquad 3.3.10.b$$

or

$$\begin{bmatrix} \dot{R} \\ \dot{S} \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial y} \\ \frac{\partial h}{\partial x} & \frac{\partial h}{\partial y} \end{bmatrix} \begin{bmatrix} R \\ S \end{bmatrix} \qquad 3.3.11$$

Calculation of the final values of R(t) and S(t) is not very simple as the previous case and should be considered very carefully and in detail. Let us begin from the definition of R(t) to calculate $R(t_f)$.

$$R(t_f) = \left.\frac{\partial x(t;p)}{\partial p}\right|_{t=t_f} \qquad 3.3.12$$

Making use of equations 3.3.5 and 3.3.6, equation 3.3.12 can be rewritten as

$$R(t_f) = \left.\frac{\partial x(t;x_f(p),y_f(p),v(p),T(p))}{\partial p}\right|_{t=T(p)} \qquad 3.3.13$$

Expanding equation 3.3.13, by employing chain rules of partial differentiation, it follows that

$$R(t_f) = \left.\frac{\partial x(t;x_f(p),y_f(p),v(p),T(p))}{\partial x_f(p)}\right|_{t=T(p)} \frac{\partial x_f(p)}{\partial p}$$

$$+ \left.\frac{\partial x(t;x_f(p),y_f(p),v(p),T(p))}{\partial y_f(p)}\right|_{t=T(p)} \frac{\partial y_f(p)}{\partial p}$$

$$+ \left.\frac{\partial x(t;x_f(p),y_f(p),v(p),T(p))}{\partial v(p)}\right|_{t=T(p)} \frac{\partial v(p)}{\partial p}$$

$$+ \left.\frac{\partial x(t;x_f(p),y_f(p),v(p),T(p))}{\partial T(p)}\right|_{t=T(p)} \frac{\partial T(p)}{\partial p} \qquad 3.3.14$$

Momentarily dropping the p arguments, let us evaluate the expression in the first line of equation 3.3.14, namely

$$\left.\frac{\partial x(t;x_f,y_f,v,T)}{\partial x_f}\right|_{t=T}$$

Since the evaluation of the expression at $t=T$ brings no extra terms involving $x_f$, we can first do that and then take the partial derivative with respect to $x_f$, so we have

$$\left.\frac{\partial x(t;x_f,y_f,v,T)}{\partial x_f}\right|_{t=T} = \frac{\partial x(T;x_f,y_f,v,T)}{\partial x_f} ,$$

but $x(T;x_f,y_f,v,T)$ equals nothing but $x_f$, that is

$$\left.\frac{\partial x(t;x_f,y_f,v,T)}{\partial x_f}\right|_{t=T} = \frac{\partial x_f}{\partial x_f} = I \qquad 3.3.15$$

Same reasoning also applies to the expressions in the second and the third lines of equation 3.3.14, yielding

$$\left.\frac{\partial x(t;x_f,y_f,v,T)}{\partial y_f}\right|_{t=T} = \frac{\partial x_f}{\partial y_f} = 0 \qquad 3.3.16$$

and

$$\left.\frac{\partial x(t;x_f,y_f,v,T)}{\partial v}\right|_{t=T} = \frac{\partial x_f}{\partial v} = 0 \qquad 3.3.17$$

For the last expression of equation 3.3.14, situation is a bit different. It can be shown that (see Appendix A),

$$\left.\frac{\partial x(t;x_f,y_f,v,T)}{\partial T}\right|_{t=T} = -f(x_f(p),y_f(p),T(p)) \qquad 3.3.18$$

Substituting the values of the expressions found in equations 3.3.15 through 3.3.18 into equation 3.3.14, we have

$$R(t_f) = \frac{\partial x_f(p)}{\partial p} - f(x_f(p),y_f(p),T(p))\,\frac{\partial T(p)}{\partial p} \qquad 3.3.19$$

Similarly $S(t_f)$ can be calculated as

$$S(t_f) = \frac{\partial y_f(p)}{\partial p} - h(x_f(p),y_f(p),T(p))\,\frac{\partial T(p)}{\partial p} \qquad 3.3.20$$

or in a more compact form

$$\begin{bmatrix} R(t_f) \\ S(t_f) \end{bmatrix} = \begin{bmatrix} I & 0 & -f(x_f(p),y_f(p),T(p)) \\ 0 & I & -h(x_f(p),y_f(p),T(p)) \end{bmatrix} \begin{bmatrix} \partial x_f(p)/\partial p \\ \partial y_f(p)/\partial p \\ \partial T(p)/\partial p \end{bmatrix} \qquad 3.3.21$$

Thus we have now two initial value problems, for $x(t),y(t)$ given by equations 3.3.1 and 3.3.2, and for $R(t),S(t)$ given by equations 3.3.11, and their values at $t=t_f$ given by equations 3.3.5 and 3.3.21, respectively. So we can integrate them simultaneously.

The rest of the method is the same as the previous case. To repeat, as a result of the initial value problems, we obtain a value for $x(t_o)$, which is probably different from the desired value $x_o$, and a value for its sensitivity to the variations in p, namely $R(t_o)$. Then these values $x(t_o)$, $R(t_o)$, and $x_o$ are used to adjust the initial guess p in an attempt to make the next value of $x(t_o)$ come closer to $x_o$, namely

$$p_{new} = p_{old} - R^{-1}(t_o)\ (x(t_o) - x_o) \qquad 3.3.22$$

provided that $R(t_o)$ is nonsingular.

Algorithm that summarizes the method for the general case is almost same as the one we have discussed before, however for the sake of completeness let us go over it again, while making the appropriate modifications.

1. Choose a norm in $E^n$ and an accuracy $\delta_x$.

2. Make an initial guess for the parameter vector p.

3. Determine the values for $x_f(p)$, $y_f(p)$, $v(p)$, and $T(p)$ using equations 3.3.5, and the values for $R(t_f)$, and $S(t_f)$ using equations 3.3.21.

4. Using these values, solve the initial value problems given by equations 3.3.1, and 3.3.2 for states and costates, and given by equations 3.3.11 for sensitivities, from $t_f$ to $t_o$. Store only values $x(t_o)$ and nxn matrix $R(t_o)$.

5. Check to see if the termination criterion $\|x(t_o)-x_o\| < \delta_x$ is satisfied. If this the case, use the final iterated value of p to reintegrate the system equations 3.3.1 and 3.3.2 while evaluating the optimal control history using $u^*(t)=z(x^*(t),y^*(t),t)$. If the stopping criterion is not satisfied, then determine the new value of p using equation 3.3.22 and return to step 3.

## 3.4 Determination of The Values of The Parametric Functions For a Given p, Numerically

Let us now answer the question "What shall we do if the parametric functions given in equation 3.3.5 do not exist, analytically?". What we have done is to parametrize a given set of n+k+1 nonlinear algebraic equations

$$m(x(t_f),t_f) = 0 \quad . \qquad 3.4.1.a$$

$$l(x(t_f).y(t_f),v,t_f) = 0 \qquad 3.4.1.b$$

$$g(x(t_f),y(t_f),v,t_f) = 0 \qquad 3.4.1.c$$

using an n-dimensional parameter vector p,yielding

$$x(t_f) \triangleq x_f(p) \qquad 3.4.2.a$$

$$y(t_f) \triangleq y_f(p) \qquad 3.4.2.b$$

$$v \triangleq v(p) \qquad 3.4.2.c$$

$$t_f \triangleq T(p) \qquad 3.4.2.d$$

such that $x_f(p)$, $y_f(p)$, $v(p)$, and $T(p)$ are continuous functions of p, and equations 3.4.1 are satisfied for all possible values of p. Since we cannot always find analytic expressions for parametric functions given in equation 3.4.2, we will determine them numerically.

Suppose that we know a set of values for $x_f(p)$, $y_f(p)$, $v(p)$, and $T(p)$ such that they satisfy equations 3.4.1. If we now introduce some finite changes on those values, then the resulting changes in m, l, and g can be approximated by the algebraic vector-matrix equation

$$\begin{bmatrix} m \\ l \\ g \end{bmatrix} = Q \begin{bmatrix} \Delta x_f(p) \\ \Delta y_f(p) \\ \Delta v(p) \\ \Delta T(p) \end{bmatrix} \qquad 3.4.3$$

where Q happens to be n+k+1x2n+k+1 matrix

$$\begin{bmatrix} M_x & M_y & M_v & M_T \\ L_x & L_y & L_v & L_T \\ G_x & G_y & G_v & G_T \end{bmatrix} \qquad 3.4.4$$

and

$$M_x \triangleq \frac{\partial m}{\partial x_f} \qquad L_x \triangleq \frac{\partial l}{\partial x_f} \qquad G_x \triangleq \frac{\partial g}{\partial x_f}$$

$$M_y \triangleq \frac{\partial m}{\partial y_f} = 0 \qquad L_y \triangleq \frac{\partial l}{\partial y_f} \qquad G_y \triangleq \frac{\partial g}{\partial y_f} \qquad 3.4.5$$

$$M_v \triangleq \frac{\partial m}{\partial v} = 0 \qquad L_v \triangleq \frac{\partial l}{\partial v} \qquad G_v \triangleq \frac{\partial g}{\partial v}$$

$$M_T \triangleq \frac{\partial m}{\partial T} \qquad L_T \triangleq \frac{\partial l}{\partial T} \qquad G_T \triangleq \frac{\partial g}{\partial T}$$

Of course the equation 3.4.3 can be used to determine the necessary changes on the values of $x_f(p)$, $y_f(p)$, $v(p)$, and $T(p)$, to satisfy equations 3.4.1. This can be done only if we have fixed n out of 2n+k+1 unknowns beforehand, such that Q is reduced to a square matrix. Without loss of generality we can set $x_f(p)$ equal to p, then by the initial guess made for p, the values of $x_f(p)$ become fixed, namely

$$x_f(p) \triangleq p \qquad 3.4.6$$

Now suppose that we guess also the values of the remaining n+k+1 unknowns, namely $y_f(p)$, $v(p)$, and $T(p)$ and evaluate the functions m, l, and g which probably do not vanish. Then we have to change the values of $y_f(p)$, $v(p)$, and $T(p)$ to satisfy equations 3.4.1 while the values of $x_f(p)$ kept fixed. Thus the desired changes, in order to make the next values of m, l, and g come closer to zero, can be obtained by substituting $\Delta x_f(p)=0$ in equation 3.4.3, yielding

$$\begin{bmatrix} \Delta y_f(p) \\ \Delta v(p) \\ \Delta T(p) \end{bmatrix} = \begin{bmatrix} 0 & 0 & M_T \\ L_y & L_v & L_T \\ G_y & G_v & G_T \end{bmatrix}^{-1} \begin{bmatrix} m \\ l \\ g \end{bmatrix} \qquad 3.4.7$$

provided that the indicated inverse exists.

Certainly any n of the unknowns, as well as $x_f(p)$, can be set equal to p, and will lead a square matrix, by substituting zero for their variations in equation 3.4.3.

We also have to determine the derivatives of the parametric functions with respect to p numerically, to use in the evaluation of $R(t_f)$ and $S(t_f)$. Taking the partial derivatives of the equations 3.4.1 with respect to p, we obtain

$$\begin{bmatrix} M_x & M_y & M_v & M_T \\ L_x & L_y & L_v & L_T \\ G_x & G_y & G_v & G_T \end{bmatrix} \begin{bmatrix} \partial x_f(p)/\partial p \\ \partial y_f(p)/\partial p \\ \partial v(p)/\partial p \\ \partial T(p)/\partial p \end{bmatrix} = 0 \qquad 3.4.8$$

or substituting $x_f(p)/ p=I$ for the case of $x_f(p)=p$, then the equation 3.4.8 reduces to

$$\begin{bmatrix} \partial y_f(p)/\partial p \\ \partial v(p)/\partial p \\ \partial T(p)/\partial p \end{bmatrix} = - \begin{bmatrix} 0 & 0 & M_T \\ L_y & L_v & L_T \\ G_y & G_v & G_T \end{bmatrix}^{-1} \begin{bmatrix} M_x \\ L_x \\ G_x \end{bmatrix} \qquad 3.4.9$$

where the matrix inverted is the same as we have used in equation 3.4.7.

In the view of this section only the steps 2, and 3 of the algorithm, needs to be modified as follows.

2a Choose n of the final values to form the parameter vector p, and make an initial guess for p.

2b Choose norms in $E^n$, $E^k$ and accuracies $\delta_m$, $\delta_1$, $\delta_g$.

2c Also guess the remaining n+k+1 final values

3a Evaluate m, l, and g, and check to see if $\|m(x_f(p),T(p))\| < \delta_m$, and $\|l(x_f(p),y_f(p),v(p),T(p))\| < \delta_1$, and $|g(x_f(p),y_f(p),v(p),T(p))| < \delta_g$ are satisfied. If it is, store the values of $x_f(p)$, $y_f(p)$, $v(p)$, $T(p)$ to use in the next iterate (or as true values) and goto step 3c.

3b If it is not, then making the appropriate substitutions, solve the algebraic equations 3.4.3 to determine new final values and return to step 3a.

3c Using the final values found in step 3a, and again making the appropriate substitutions, solve the equation 3.4.8 to determine the partial derivatives of $x_f(p)$, $y_f(p)$, $v(p)$, $T(p)$ with respect to p, and evaluate $R(t_f)$ and $S(t_f)$ using equation 3.3.21.

# IV. EXISTENCE OF SOLUTIONS AND CONVERGENCE

## 4.1 Introduction

In the previous chapter, we made some assumptions on $f(x,y,t)$ and $h(x,y,t)$ to ensure that the initial value problem has a unique solution for a given set of initial conditions. Infact the uniqueness of the solution of the initial value problem is necessary, since then we are solving transcendental equations (in which initial values are the variables of the equations) to find its roots and evidently determine the solution of the boundary value problem.

In this chapter, we will study the necessary conditions for the existence of solutions for the boundary value problems of a special type, namely with linear boundary conditions, by stating the necessary theorems and proving some of them. This study is done simply to give an intuition. The existence theory for more general boundary value problems should be considered as an independent research topic.

## 4.2 Initial Value Problems

One of the basic results of the initial value problems can be stated as follows.

THEOREM 4.1 Let an n-vector $s_0$ and positive numbers $\rho$, K, and M be given such that, with

$$R_{\rho,M}(s_o) = \left\{(t,u)\,\middle|\, |u-s_o| \leqslant \rho + M(t-t_o),\quad t_o \leqslant t \leqslant t_f\right\},$$

(a) $f(t,u)$ is continuous in $R_{\rho,M}(s_o)$

(b) $|f(t,u)| \leqslant M$ for all $(t,u) \in R_{\rho,M}(s_o)$

(c) $|f(t,u)-f(t,v)| \leqslant K|u-v|$ for all $(t,u),(t,v) \in R_{\rho,M}(s_o)$

Then the initial value problem

$$\dot{u} = f(t,u) \quad , \quad u(t_o) = s \qquad 4.2.1$$

has a unique solution $u=u(t;s)$ on $[t_o,t_f]$ for each

$$s \in N_\rho(s_o) \triangleq \left\{s \,\middle|\, |s-s_o| \leqslant \rho\right\}$$

Furthermore, the solution is Lipschitz-continuous in s, more precisely

$$|u(t,s)-u(t,p)| \leqslant e^{K|t-t_o|}\,|s-p| \qquad 4.2.2$$

for all $t \in [t_o,t_f]$, and all s and $p \in N_\rho(s_o)$.

Proof: For the moment suppose that $u(t)$ is known and satisfies the relation

$$u(t) = s_o + \int_{t_o}^{t} f(\tau,u(\tau))\,d\tau \qquad 4.2.3$$

Now regard $u(t)$ as unknown, let t lie in the interval $[t_o,t_f]$, and consider the sequence of functions $u_o(t)$, $u_1(t),\ldots\ldots,u_n(t)$ defined as follows,

$$u_o(t) = s_o$$

$$u_1(t) = s_o + \int_{t_o}^{t} f(\tau,u_o(\tau))\,d\tau$$

$$u_2(t) = s_o + \int_{t_o}^{t} f(\tau, u_1(\tau))\, d\tau$$

$$\vdots$$

$$u_n(t) = s_o + \int_{t_o}^{t} f(\tau, u_{n-1}(\tau))\, d\tau$$

It will now be proved

i) that as n increases indefinitely, the sequence of functions $u_n(t)$ tends to limit which is a continuous function of t,

ii) that the limit function satisfies the differential equation 4.2.1,

iii) that solution is unique and assumes $s_o$ when $t=t_o$.

We have to show that $|u_n(t)-s_o| \leqslant \rho + M(t-t_o) \quad t_o \leqslant t \leqslant t_f$.
Suppose that $|u_{n-1}(t)-s_o| \leqslant \rho + M(t-t_o)$, then

$$|f(t, u_{n-1}(t))| \leqslant M \,,$$

and consequently

$$|u_n(t)-s_o| \leqslant \int_{t_o}^{t} |f(\tau, u_{n-1}(\tau))|\, d\tau$$

$$\leqslant M(t-t_o)$$

$$\leqslant \rho + M(t-t_o)$$

Clearly

$$|u_1(t)-s_o| \leqslant \rho + M(t-t_o) \,,$$

it is therefore true that

$$u_n(t) \in R_{\rho, M}(s_o)$$

for all values of n.

CLAIM:

$$\left|u_n(t)-u_{n-1}(t)\right| \leqslant \frac{M\ K^{n-1}}{n!}\,(t-t_o)^n$$

For n=1, it is obvious. Suppose it to be true that when $t \in [t_o,t_f]$

$$\left|u_{n-1}(t)-u_{n-2}(t)\right| \leqslant \frac{M\ K^{n-2}}{(n-1)!}\,(t-t_o)^{n-1},$$

then

$$\left|u_n(t)-u_{n-1}(t)\right| \leqslant \int_{t_o}^{t} \left|f(\tau,u_{n-1}(\tau))-f(\tau,u_{n-2}(\tau))\right| d\tau$$

$$\leqslant \int_{t_o}^{t} K\left|\ddot{u}_{n-1}(\tau)-u_{n-2}(\tau)\right| d\tau$$

$$\leqslant \int_{t_o}^{t} K\,\frac{M\ K^{n-2}}{(n-1)!}\,(\tau-t_o)^{n-1}\, d\tau$$

$$\leqslant \frac{M\ K^{n-1}}{n!}\,(t-t_o)^n$$

It follows that the series

$$s_o + \sum_{r=1}^{\infty}\left\{u_r(t)-u_{r-1}(t)\right\}$$

is absolutely and uniformly convergent when $t$ $t_o,t_f$, moreover each term is a continuous function of t. But

$$u_n(t) = s_o + \sum_{r=1}^{n}\left\{u_r(t)-u_{r-1}(t)\right\};$$

consequently the limit function

$$u(t) = \lim_{n \to \infty} u_n(t)$$

exists and continuous function of $t \in [t_o, t_f]$. It can be shown that

$$\lim_{n\to\infty} u_n(t) = s_o + \lim_{n\to\infty} \int_{t_o}^{t} f(\tau, u_{n-1}(\tau))\, d\tau$$

$$= s_o + \int_{t_o}^{t} \lim_{n\to\infty} f(\tau, u_{n-1}(\tau))\, d\tau$$

$$= s_o + \int_{t_o}^{t} f(\tau, u(\tau))\, d\tau ,$$

thus $u(t)$ is a solution of the integral equation 4.3.2, and since f is a continuous function

$$\frac{d\,u(t)}{dt} = \frac{d}{dt}\int_{t_o}^{t} f(\tau, u(\tau))\, d\tau$$

$$= f(t, u(t))$$

Now suppose that there exists $U(t)$ which is a solution distinct from $u(t)$ satisfying the initial condition $U(t_o)=s_o$ and continuous in an interval $[t_o, t_f']$ where $t_f' < t_f$ such that

$$|U(t) - s_o| < \rho + M(t - t_o)$$

then it satisfies the integral equation

$$U(t) = s_o + \int_{t_o}^{t} f(\tau, U(\tau))\, d\tau$$

and consequently

$$U(t) - u_n(t) = \int_{t_o}^{t} \left\{ f(\tau, U(\tau)) - f(\tau, u_{n-1}(\tau)) \right\} d\tau$$

Successive applications of the Lipschitz condition, yield

$$\left| U(t) - u_{n-1}(t) \right| \leqslant \frac{K^n \, (\varsigma + M(t-t_o)) \, (t-t_o)^n}{n!}$$

Hence

$$U(t) = \lim_{n \to \infty} u_n(t) = u(t)$$

for all values of $t$ in the interval $[t_o, t_f']$, and therefore the new solution is identical with the old one.

Demostration of equation 4.2.2 can be found in Keller (1968) pp 3. ∎

Replacing condition (c) of Theorem 4.1 by the requirement that $\partial f(t,u)/\partial u$ be continuous on $R_{\varsigma,M}(s_o)$, one can show that the solution $u(t;s)$ of equation 4.2.1 are continuously differentiable with respect to s in $N_{\varsigma}(s_o)$ and $t \in [t_o, t_f]$. Ince (1944), pp 68-69, had shown it for scalar case, and its extention to vector case is immediate.

## 4.3 Two-Point Boundary-Value Problems (TPBVP)

Let us consider the general system of n first order differential equations

$$\dot{y} = f(t,y) \quad , \quad t_o \leqslant t \leqslant t_f \qquad 4.3.1.a$$

subject to the most general linear boundary conditions

$$Ay(t_o) + By(t_f) = \alpha \qquad 4.3.1.b$$

where $y(t)$ is an n-dimensional vector, $f(t,y)$ is an n-vector function of $t$ and $y$. Most of the TPBVP's with linear boundary conditions can be written in the form of equation 4.3.1.b, and they are linearly independent if and only if the nx2n order matrix $[A,B]$ has rank n. Let us also consider the initial value problem

$$\dot{u} = f(t,u) \quad , \quad u(t_o) = s \qquad 4.3.2$$

where s is an n-vector to be determined. In terms of the $u=u(t;s)$ of the problem 4.3.2 we define the system of n equations, namely

$$\phi(s) \triangleq As + Bu(t_f;s) - \alpha = 0 \qquad 4.3.3$$

Clearly, if $s=s^*$ is a root of this equation, we expect that

$$y(t) = u(t;s^*)$$

is a solution of the boundary value problem 4.3.1. In fact it is and we have the following theorem.

THEOREM 4.2 Let $f(t,u)$ be continuous on

$$R: \; t_o \leqslant t \leqslant t_f \quad , \quad |u| < \infty$$

and satisfy there a uniform Lipschitz condition in u. Then the boundary value problem 4.3.1 has many solutions as there are distinct roots $s=s^i$ of equation 4.3.3. These solutions

$$y(t) = u(t,s^i)$$

are also the solutions of the initial value problem 4.3.2 with $s=s^i$.

Proof: In Keller 1968, pp 8-9, it is proved for n=2, and in this case, the proof is almost identical. ∎

We have now reduced the problem of solving the boundary value problem 4.3.1 to that of finding the roots of a system of n transcendental equations. A very effective class of numerical methods are based on this equivalance. It is generally quite difficult to prove the existence of roots of such systems. For the special case that we are dealing with, we have the following theorem.

THEOREM 4.3 Let an n-vector $s_o$ and positive constants $\rho$, K, M satisfy, for

$$R_{\rho,M}(s_o) = \left\{ (t,u) \,\middle|\, |u-s_o| \leqslant \rho + M(t-t_o) \quad , \quad t_o \leqslant t \leqslant t_f \right\},$$

(a) $f(t,u)$ is continuous in $R_{\rho,M}(s_o)$

(b) $|f(t,u)| \leqslant M$ in $R_{\rho,M}(s_o)$

(c) $|f(t,u)-f(t,v)| \leqslant K|u-v|$ for all $(t,u),(t,v) \in R_{\rho,M}(s_o)$

Further, let the matrices and the interval length, $|t_f - t_o|$, be such that

(d) $(A+B)$ is nonsingular

(e) $\|(A+B)^{-1}B\| \cdot M(t_f-t_o) + \|(A+B)^{-1}\alpha - s_o\| \leqslant \rho.$

Then the boundary value problem

$$\dot{y} = f(t,y) \quad , \quad Ay(t_o) + By(t_f) = \alpha \qquad 4.3.4$$

has at least one solution with $y(t_o) \in N_\rho(s_o)$.

Proof: From conditions (a) through (c), theorem 4.1 is applicable, hence 4.3.4 has a solution if and only if the n-transcendental equations

$$\phi(s) \triangleq As + Bu(t_f;s) - \alpha = 0$$

has a solution. We also know that $\phi(s)=0$ has identical roots with that of $s=g(s)$ where

$$g(s) \triangleq s - Q(s)\phi(s) \ ,$$

if $Q(s)$ is any n-th order matrix which is bounded and nonsingular for all s. Without loss of generality we can choose $Q(s)=(A+B)^{-1}$ by condition (d), then

$$g(s) = s - (A+B)^{-1}\left[As + Bu(t_f;s) - \alpha\right]$$

$$= s - (A+B)^{-1}\left[(A+B)s + B(u(t_f;s)-s) - \alpha\right]$$

$$= (A+B)^{-1}\alpha - (A+B)^{-1}B(u(t_f;s)-s)$$

Since $u(t_f;s)$ is continuous with respect to s in $N_\rho(s_o)$, so $g(s)$ is also continuous with respect to s in $N_\rho(s_o)$.

CLAIM: $g(s)$ maps the closed sphere $N_\rho(s_o)$ into itself, that is to say,

$$\|g(s)-s_o\| \leq \rho \quad \text{for all} \quad s \in N_\rho(s_o)$$

The expression for $g(s)-s_o$ is

$$g(s)-s_o = (A+B)^{-1}\alpha - s_o - (A+B)^{-1}B(u(t_f;s)-s)$$

consequently

$$\|g(s)-s_o\| \leqslant \|(A+B)^{-1}\alpha - s_o\| + \|(A+B)^{-1}B\| \|u(t_f,s)-s\| \qquad 4.3.5$$

but we have

$$u(t_f;s) = s + \int_{t_o}^{t_f} f(t,u(t))\, dt$$

which implies

$$\|u(t_f;s)-s\| \leqslant \int_{t_o}^{t_f} \|f(t,u(t))\|\, dt$$

$$\leqslant M(t_f-t_o)$$

substituting above inequality into equation 4.3.5, we have

$$\|g(s)-s_o\| \leqslant \|(A+B)^{-1}B\| \cdot M(t_f-t_o) + \|(A+B)^{-1}\alpha - s_o\| ,$$

and it follows from the condition (e), that

$$\|g(s)-s_o\| \leqslant \rho$$

which shows that $g(s)$ maps $N_\rho(s_o)$ into itself. Then by Brouwer Fixed Point Theorem, $s=g(s)$ has at least one root in $N_\rho(s_o)$, so $\phi(s)=0$ does. Therefore boundary value problem has at least one solution with $y(t_o)$ $N_\rho(s_o)$. ∎

## 4.4 Convergence

To solve equations of the form

$$g(s) = s - Q(s)\phi(s)$$

we need only to determine the matrix $Q(s)$ such that $g(s)$ is continuous and maps $N_{\rho}(s_o)$ into itself. A particularly effective procedure of this method is Newton's Method, in which we take $Q(s) \triangleq J^{-1}(s)$ where

$$J(s) \triangleq \frac{\partial \phi(s)}{\partial s}$$

is the Jacobian matrix of $\phi$ with respect to s. The corresponding iteration scheme is then

$$s^{(i+1)} = s^{(i)} - J^{-1}(s^{(i)})\phi(s^{(i)}) \qquad i=0,1,.. \qquad 4.4.1$$

The convergence of the Newton's method is frequently rapid, even better than the geometric type of convergence. We do not go into the details here, but simply state the following theorem.

THEOREM 4.4 Let the function $\phi(s)$ have a zero $s=s_o$, continuous first order derivatives in some neighbourhood $N_{\rho}(s_o)$ of $s_o$ and nonsingular Jacobian at $s_o$, that is, $\det J(s_o) \neq 0$. Then for each $\lambda$ in $0<\lambda<1$ there exists a positive number $\rho_\lambda$ such that for any $s^{(o)}$ $N_{\rho_\lambda}(s_o)$ the Newton iterates (4.4.1) converge to $s_o$ with

$$\left|s_o - s^{(i)}\right| \leqslant \lambda^i \left|s_o - s^{(o)}\right| .$$

Proof: It is given in Keller 1968, pp 33-35. ∎

# V. NUMERICAL EXAMPLES AND RESULTS

We are now going to study some examples to illustrate the method, we have discussed so far.

EXAMPLE 5.1 Assume that the system equations are given by

$$\dot{x}(t) = x(t) - y(t) \qquad 5.1.1.a$$

$$\dot{y}(t) = x(t) + y(t) \qquad 5.1.1.b$$

and that the boundary conditions are

$$x(0) = 1 \qquad 5{,}1.2.a$$

$$y(t_f) - \beta e^2 = 0 \qquad 5.1.2.b$$

$$x(t_f) - \alpha e^{t_f} = 0 \qquad 5.1.2.c$$

where $\alpha$=Cos2-Sin2, and $\beta$=Cos2+Sin2. Since the analytic solution is known for parametric functions of this particular problem, we have chosen $t_f$ to be our parameter p, then it follows that

$$x_f(p) = \alpha e^p \qquad 5.1.3.a$$

$$y_f(p) = \beta e^2 \qquad 5.1.3.b$$

$$T(p) = p \qquad 5.1.3.c$$

and from equation 3.3.21, we have

$$R(t_f) = \alpha e^p - (\alpha e^p - \beta e^2)\ 1$$

$$= \beta e^2 \qquad 5.1.4.a$$

and

$$S(t_f) = -\alpha e^p - \beta e^2 \qquad 5.1.4.b$$

The differential equations for R(t) and S(t) are found as

$$\dot{R}(t) = R(t) - S(t) \qquad 5.1.5.a$$

$$\dot{S}(t) = R(t) + S(t) \qquad 5.1.5.b$$

The initial guess used to start the iterative procedure was

$$p = 2.7000$$

and

$$|x(0)-1| < 0.0001$$

was used as a stopping criterion. The method converged after 4 iterations (with a norm 0.000016164) to

$$p = 2.000014904$$

which yielded as the initial value of state

$$x(0) = 0.999983836$$

TABLE 5.1 Solution of the example 5.1

| Iteration | Sensitivity R(0) | Final Time p | Initial X x(0) |
|---|---|---|---|
| 0 | 0.2017 | 2.7000 | 1.27393 |
| 1 | 0.5681 | 1.3420 | 0.63375 |
| 2 | 0.5225 | 1.9867 | 0.99251 |
| 3 | 0.5394 | 2.0010 | 1.00046 |
| 4 | 0.5384 | 2.0001 | 0.99998 |

EXAMPLE 5.2 We worked with the same problem again except that the boundary condition 5.1.2.b was changed to

$$y(t_f) - \beta e^{t_f} = 0 \qquad 5.2.1$$

then it follows that

$$y_f(p) = \beta e^p \qquad 5.2.2$$

and

$$S(t_f) = -\alpha e^p \qquad 5.2.3$$

and the other relationship, remained unchanged. The initial guess used to start the iterative procedure was

$$p = 2.6000$$

which yielded the solution given in Table 5.2. An initial guess of p=2.7000 was also tried. The first iteration found a negative value as a final time. This points out the importance of making a good initial guess.

TABLE 5.2 Solution of the example 5.2

| Iteration | Sensitivity $R(0)$ | Final Time $p$ | Final X $x_f(p)$ | Initial X $x(0)$ |
|---|---|---|---|---|
| 0 | 0.2443 | 2.6000 | -17.8456 | 1.37867 |
| 1 | 1.3602 | 1.0501 | -3.7882 | -0.18420 |
| 2 | 1.0185 | 1.9207 | -9.0476 | 0.92142 |
| 3 | 0.9780 | 1.9979 | -9.7732 | 0.99780 |
| 4 | 0.9802 | 2.0001 | -9.7952 | 1.00000 |

EXAMPLE 5.3 The version of example 5.2 was worked again, but this time $x(t_f)$ was taken as the parameter p. Then we have

$$x_f(p) = p \qquad 5.3.1.a$$

$$y_f(p) = \frac{\beta}{\alpha} p \qquad 5.3.1.b$$

$$T(p) = \ln\left(\frac{p}{\alpha}\right) \qquad 5.3.1.c$$

and

$$R(t_f) = 1 - \left(p - \frac{\beta}{\alpha}p\right)\frac{1}{p}$$

$$= \frac{\beta}{\alpha} \qquad 5.3.2.a$$

$$S(t_f) = \frac{\beta}{\alpha} - \left(p + \frac{\beta}{\alpha}p\right)\frac{1}{p}$$

$$= -1 \qquad 5.3.2.b$$

An initial guess of

$$p = -3.0000$$

was used to start the iterative procedure. We obtained the results given in Table 5.3.

EXAMPLE 5.4 The last version of the same problem was worked once more. This time the values of the parametric functions were determined numerically rather than analytically. The set of initial guesses

$$p = -3.0000$$

$$y_f(p) = 6.0000$$

TABLE 5.3 Solution of the examples 5.3 and 5.4

| Iteration | Sensitivity $R(0)$ | Final Time $T(p)$ | Final X $p$ | Initial X $x(0)$ |
|---|---|---|---|---|
| 0 | -0.4287 | 0.8169 | -3.0000 | -0.5056 |
| 1 | -0.1940 | 1.5919 | -6.5123 | 0.5423 |
| 2 | -0.1191 | 1.9010 | -8.8710 | 0.8997 |
| 3 | -0.1001 | 1.9917 | -9.7135 | 0.9917 |
| 4 | -0.1001 | 2.0002 | -9.7964 | 1.0002 |
| 5 | -0.1001 | 2.0001 | -9.7947 | 1.0000 |

and

$$T(p) = 2.5000$$

and the corresponding matrix

$$Q = \begin{bmatrix} 0 & 1 & -\beta e^{T(p)} \\ 1 & 0 & -\alpha e^{T(p)} \end{bmatrix}$$

were used to start the procedure, and we obtained the same results given in Table 5.3.

EXAMPLE 5.5 Consider the following nonlinear system of equations

$$\dot{x}(t) = y(t) \qquad 5.5.1.a$$

$$\dot{y}(t) = -1 - x(t) - y^2(t) \qquad 5.5.1.b$$

and the boundary conditions

$$x(0) = 0 \qquad 5.5.2.a$$

$$x(t_f) - 1 = 0 \qquad 5.5.2.b$$

$$y(t_f) = 0 \qquad 5.5.2.c$$

where the last equation is necessary to determine the free boundary $t_f$. Analytic expressions for the boundary conditions are

$$x_f(p) = 1 \qquad 5.5.3.a$$

$$y_f(p) = 0 \qquad 5.5.3.b$$

$$T(p) = p \qquad 5.5.3.c$$

The differential equations that the sensitivities satisfy are

$$\dot{R}(t) = S(t) \qquad 5.5.4.a$$

$$\dot{S}(t) = -R(t) - 2y(t)S(t) \qquad 5.5.4.b$$

and their corresponding final values are found as

$$R(t_f) = 0 \quad , \quad S(t_f) = 2 \qquad 5.5.5$$

For an initial guess of $p=1.0000$, and the stopping criteria of $|x(0)|<0.01$, we obtained the results given in the following table. For this particular problem, we could not reach to the desired value of $x(0)$ with an error less

TABLE 5.4 Solution of example 5.5

| Iteration | Sensitivity R(0) | Final Time p | Initial X x(0) |
|---|---|---|---|
| 0 | -6.1555 | 1.0000 | -0.61312 |
| 1 | -3.7646 | 0.9004 | -0.13561 |
| 2 | -3.2390 | 0.8644 | 0.00400 |

than 0.004. When we have used a stopping criteria of greater accuracy, the method has converged to $x(0)=0.00400$ and stayed there. This is because of the truncation errors and the nonlinear nature of the problem.

EXAMPLE 5.6 The last example that will be considered is a 2-dimensional problem given by

$$\dot{x}_1(t)=y_1(t) \quad , \quad \dot{x}_2(t)=y_2(t) \qquad 5.6.1.a$$

$$\dot{y}_1(t)=x_2(t) \quad , \quad \dot{y}_2(t)=x_1(t) \qquad 5.6.1.b$$

and the boundary conditions for the problem are

$$x_1(0) = 0 \quad , \quad x_2(0) = 0 \qquad 5.6.2.a$$

$$x_1(t_f)-5t_f-3=0 \quad , \quad x_2(t_f)-\tfrac{1}{2}t_f^2=0 \qquad 5.6.2.b$$

and

$$2x_1(t_f)x_2(t_f)+10y_1(t_f)t_f+6y_1(t_f)$$

$$+y_2(t_f)t_f^2-y_1^2(t_f)-y_2^2(t_f) = 0 \qquad 5.6.2.c$$

Notice that the last equation is too complex to find an analytic expression for either $y_1(t_f)$ or $y_2(t_f)$, whereas the other equations are so simple to apply a numerical technique to determine the values of $x_1(t_f)$ and $x_2(t_f)$, of course if $t_f$ is chosen to be one of the parameters. So in such a case, we can combine the methods, discussed in sections 3.3 and 3.4, to reduce the dimension of the matrix Q which needs to be inverted. We proceed as follows: Letting

$$p^T = \begin{bmatrix} t_f & y_1(t_f) \end{bmatrix} \qquad 5.6.3$$

then we have

$$x_{1f}(p) = 5p_1+3 \qquad 5.6.4.a$$

$$x_{2f}(p) = \tfrac{1}{2}p_1^2 \qquad 5.6.4.b$$

$$y_{1f}(p) = p_2 \qquad 5.6.4.c$$

$$T(p) = p_1 \qquad 5.6.4.d$$

and the equation, that we are going to make use of, to determine $y_{2f}(p)$ numerically,

$$G = 5p_1^3+3p_1^2+10p_1p_2+6p_2+y_{2f}(p)p_1^2-p_2^2-y_{2f}^2(p) = 0 \qquad 5.6.5$$

For this particular problem, equation 3.4.3 reduces to

$$G = \frac{\partial G}{\partial y_{2f}(p)} \Delta y_{2f}(p) \qquad 5.6.6$$

where

$$\frac{\partial G}{\partial y_{2f}(p)} = p_1^2 - 2y_{2f}(p) \qquad 5.6.7$$

and the corresponding derivatives of $y_{2f}(p)$ with respect to $p_1$ and $p_2$ may be found from

$$\begin{bmatrix} \dfrac{\partial y_{2f}(p)}{\partial p_1} \\ \\ \dfrac{\partial y_{2f}(p)}{\partial p_2} \end{bmatrix} = - \left( \frac{\partial G}{\partial y_{2f}(p)} \right)^{-1} \begin{bmatrix} \partial G/\partial p_1 \\ \\ \partial G/\partial p_2 \end{bmatrix} \qquad 5.6.8$$

where

$$\partial G/\partial p_1 = 15p_1^2+6p_1+10p_2+2y_{2f}(p)p_1 \qquad 5.6.9.a$$

and

$$\partial G/\partial p_2 = 10p_1+6-2p_2 \qquad 5.6.9.b$$

Sensitivity matrices $R(t)$ and $S(t)$ satisfy the following differential equations

$$\dot{R}(t) = S(t) \tag{5.6.10.a}$$

and

$$\dot{S}(t) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} R(t) \tag{5.6.10.b}$$

and their final values may be calculated as

$$R(t_f) = \begin{bmatrix} 5 & 0 \\ p_1 & 0 \end{bmatrix} - \begin{bmatrix} p_2 \\ y_{2f}(p) \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} 5-p_2 & 0 \\ p_1 - y_{2f}(p) & 0 \end{bmatrix} \tag{5.6.11.a}$$

and

$$S(t_f) = \begin{bmatrix} 0 & 1 \\ \dfrac{\partial y_{2f}(p)}{\partial p_1} & \dfrac{\partial y_{2f}(p)}{\partial p_2} \end{bmatrix} - \begin{bmatrix} \frac{1}{2}p_1^2 \\ 5p_1+3 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{1}{2}p_1^2 & 1 \\ \dfrac{\partial y_{2f}(p)}{\partial p_1} - 5p_1 - 3 & \dfrac{\partial y_{2f}(p)}{\partial p_2} \end{bmatrix} \tag{5.6.11.b}$$

The set of initial guesses

$$p_1 = 1.0000$$

$$p_2 = 5.0000$$

$$y_{2f}(p) = 10.0000$$

TABLE 5.5 Solution of the example 5.6

| Itr. | Sensitivity Matrix $R(0)$ | | Initial X $x_1(0), x_2(0)$ | Final Y $p_2, y_{2f}(p)$ | Final Time $p_1$ |
|---|---|---|---|---|---|
| 0 | -14.1184<br>-5.2136 | -3.1548<br>-2.8204 | -5.45909<br>-9.28671 | 5.0000<br>14.6887 | 2.0000 |
| 1 | 0.3711<br>16.9981 | -7.7273<br>-7.8836 | 11.25268<br>14.84934 | 0.6078<br>15.1111 | 2.5948 |
| 2 | -7.2626<br>7.1809 | -5.5864<br>-5.6203 | 2.54387<br>2.81605 | 2.0542<br>15.1508 | 2.3921 |
| 3 | -9.7499<br>4.6189 | -5.4024<br>-5.3844 | -0.08058<br>0.28539 | 2.5326<br>15.4958 | 2.3744 |
| 4 | -9.5580<br>4.4542 | -5.2308<br>-5.2092 | 0.04247<br>-0.03663 | 2.5638<br>15.3278 | 2.3489 |

and the stopping criteria

$$|x_1(0)| < 0.1 \quad \text{and} \quad |x_2(0)| < 0.1$$

were used to start the procedure, and we obtained the results given in the above table.

# VI. CONCLUSION

Let us now summarize the features of the method.

An initial n-vector parameter p must be selected to start the iterative procedure. It is advisable and may be better to set the final values "about which we have more knowledge from the physical nature of the problem" equal to the parameter vector p.

If the parametric functions introduced in equations 3.3.5 cannot be solved analytically, then it is necessary to guess the remaining final values, but this not a drawback. As explained in section 3.4, the algorithm brings them to the necessary values to satisfy the boundary conditions, before starting each iteration.

No trajectories need to be stored. Only the values of the sensitivity matrix $R(t_o)$, the iterated final values or simply the parameter vector p, the given initial state value, and the appropriate boundary conditions (if the analytic parametric functions do not exist) are retained in the computer memory.

If the initial guesses are such that the iterated final values are sufficiently close to the true values, then the method will converge quite rapidly. However if the initial guesses are very poor, then the method may not converge at all. In this case, it should be restarted using a different set of initial guesses. Of course all physical insight should be used to guide us in selecting the parameter vector p.

2n(n+1) first order differential equations given by 3.3.1, 3.3.2, and 3.3.11 must be numerically integrated and nxn matrix $R(t_o)$ inverted in each iteration. If again the parametric functions cannot be solved analytically, an additional n+k+1xn+k+1 (at most) matrix inversion is necessary.

The iterative procedure is terminated when

$$\| x(t_o) - x_o \| < \delta_x$$

is satisfied, where ${}_x$ is a pre-selected positive constant. Higher accuracy needs more iterations. But, because of the truncation and roundoff errors in some nonlinear complex problems, accuracy has an upper limit.

Finally, it should be noted that the algorithms presented above are quite easy to understand. The simplifying and unifying view of the sensitivity theory has made this possible.

# APPENDIX A

## Derivation of Equation 3.3.18

Let us consider the following simple TPBVP problem

$$\dot{x}(t) = f(x(t),t) \qquad A.1$$

$$x(t_o)=x_o \quad , \quad x(t_f)=x_f \qquad A.2$$

where $t_o, x_o, x_f$ are known, whereas $t_f$ is unknown. If we try to apply our method to this particular problem, we obtain

$$x_f(p) = x_f \qquad A.3a$$

$$T(p) = p \qquad A.3b$$

$$\dot{R}(t) = \frac{\partial f}{\partial x} R(t) \qquad A.4$$

and

$$R(t_f) = \left.\frac{\partial x(t;p)}{\partial p}\right|_{t=p} \qquad A.5$$

Where equation A.5 should be evaluated to solve the problem.

Suppose that we want to calculate the change $\Delta x(t,p)$ when both t and p are changed by $\Delta p$, thus we have

$$\Delta x(t;p) = x(t+\Delta p;p+\Delta p) - x(t;p) \qquad A.6$$

or expanding the right-hand side of the equation A.6 into its Taylor series, we obtain

$$x(t+\Delta p;p+\Delta p) - x(t;p)$$

$$= x(t,p) + \frac{\partial x(t,p)}{\partial t}\Delta p + \frac{\partial x(t,p)}{\partial p}\Delta p + o(\Delta p^2) - x(t,p) \qquad A.7$$

Simplifying and dividing both sides by $\Delta p$, we get

$$\frac{x(t+\Delta p;p+\Delta p)-x(t;p)}{\Delta p} = \frac{\partial x(t;p)}{\partial t} + \frac{\partial x(t;p)}{\partial p} + o(\Delta p) \qquad \text{A.8}$$

Let us evaluate both sides at $t=p$, then we have

$$\frac{x(p+\Delta p)-x(p)}{\Delta p} = \left.\frac{\partial x(t;p)}{\partial t}\right|_{t=p} + \left.\frac{\partial x(t;p)}{\partial p}\right|_{t=p} + o(\Delta p) \qquad \text{A.9}$$

taking the limit of both sides as $\Delta p \to 0$

$$\frac{\partial x_f}{\partial p} = \left.\frac{\partial x(t;p)}{\partial t}\right|_{t=p} + \left.\frac{\partial x(t;p)}{\partial p}\right|_{t=p} \qquad \text{A.10}$$

but $x_f$ is fixed while $p$ varies then $\partial x_f/\partial p=0$. Therefore

$$0 = \left.\frac{\partial x(t;p)}{\partial t}\right|_{t=p} + \left.\frac{\partial x(t;p)}{\partial p}\right|_{t=p} \qquad \text{A.11}$$

or

$$\left.\frac{\partial x(t;p)}{\partial p}\right|_{t=p} = -f(x_f,p) \qquad \text{A.12}$$

Equation 3.3.18 is simply an extention of the equation A.12, where we have more variables but are refered as fixed with respect to T(p).

# APPENDIX B

# One-step Convergenge
# of
# Linear Systems

Let us consider the following linear TPBVP, given by

$$\dot{x}(t) = A(t)x(t) + B(t)y(t) + e(t) \qquad \text{B.1a}$$

$$\dot{y}(t) = C(t)x(t) + D(t)y(t) + f(t) \qquad \text{B.1b}$$

and the boundary conditions

$$x(t_o)=x_o \quad , \quad y(t_f)=y_f \qquad \text{B.1c}$$

where $t_o, x_o, t_f, y_f$ are all known quantities. The time response of this system may be found from

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \varphi(t_f,t)\begin{bmatrix} x(t_f) \\ y(t_f) \end{bmatrix} + \int_{t_f}^{t} \varphi(t_f,\tau)\begin{bmatrix} e(\tau) \\ f(\tau) \end{bmatrix} d\tau \qquad \text{B.2}$$

where

$$\varphi(t_f,t) = \begin{bmatrix} \varphi_{11}(t_f,t) & \varphi_{12}(t_f,t) \\ \varphi_{21}(t_f,t) & \varphi_{22}(t_f,t) \end{bmatrix} \qquad \text{B.3}$$

is the state-transition matrix and $x(t_f)$ needs to be determined. In terms of $x(t_f)$, $x(t)$ can be found as

$$x(t) = \varphi_{11}(t_f,t)x(t_f) + \int_{t_f}^{t} \varphi_{11}(t_f,\tau)e(\tau)\, d\tau$$

$$+ \varphi_{12}(t_f,t)y_f + \int_{t_f}^{t} \varphi_{12}(t_f,\tau)f(\tau)\, d\tau \qquad \text{B.4}$$

and its value at $t=t_o$

$$x(t_o) = X_o + \varphi_{11}(t_f,t_o)x(t_f) \qquad \text{B.5}$$

where the constant $X_o$ is equal to

$$\varphi_{12}(t_f,t_o)y_f + \int_{t_f}^{t_o} (\varphi_{11}(t_f,\tau)e(\tau)+\varphi_{12}(t_f,\tau)f(\tau))\, d\tau$$

Now let us apply our method to determine the value of $x(t_f)$. We are going to guess $x(t_f)$, so we need to know the changes caused on $x(t_o)$, when we change the value of $x(t_f)$. From equation B.5 this relation is given by

$$R(t_o) = \frac{\partial x(t_o)}{\partial x(t_f)} = \varphi_{11}(t_f,t_o) \qquad \text{B.6}$$

Then by the algorithm

$$\begin{aligned} x_{new}(t_f) &= x_{old}(t_f) - R^{-1}(t_o)(x(t_o)-x_o) \\ &= x(t_f) - \varphi_{11}^{-1}(t_f,t_o)(X_o+\varphi_{11}(t_f,t_o)x(t_f)-x_o) \\ &= x(t_f)-\varphi_{11}^{-1}(t_f,t_o)X_o-x(t_f)+\varphi_{11}^{-1}(t_f,t_o)x_o \\ &= -\varphi_{11}^{-1}(t_f,t_o)\ (X_o - x_o) \qquad \text{B.7} \end{aligned}$$

If we now substitute this new value value of $x(t_f)$ into equation B.5, then the next value of $x(t_o)$

$$x_{new}(t_o) = X_o - \varphi_{11}(t_f,t_o)\varphi_{11}^{-1}(t_f,t_o)(X_o-x_o)$$

or

$$x_{new}(t_o) = x_o \qquad \text{B.8}$$

Hence, whatever the initial guess for $x(t_f)$, the algorithm converges to the desired solution in one-step if the system is linear.

# APPENDIX C

## Realization of example 5.6

## BASIC Program Listing

```
]LIST 1-330

1  PR# 1: POKE 1657,80
2  PRINT "     SENSITIVITY  MATRICES    X1(0),X2(0)
      Y1(T),Y2(T)        FINAL TIME"
3  PR# 0
15  DIM YP(2,2),RT(2,2),ST(2,2),RJ(2,2),SJ(2,2),PR(2
     ,2),PS(2,2),QR(2,2),QS(2,2),RR(2,2),RS(2,2),SR(
     2,2),SS(2,2),RN(2,2),SN(2,2),RI(2,2)
20  REM  INITIALIZATION
25 P(1) = 2.34886795:P(2) = 2.56156465
30 Y(2) = 15.3278465
35 L(1) = 5 * P(1) + 3:L(2) = P(1) * P(1) / 2
40 G = 5 * P(1) * P(1) * P(1) + 3 * P(1) * P(1) + 10
      * P(1) * P(2) + 6 * P(2) + Y(2) * P(1) * P(1) -
     P(2) * P(2) - Y(2) * Y(2)
45 GY(2) = P(1) * P(1) - 2 * Y(2)
50  IF  ABS (G) < 0.01001 THEN 60
55 Y(2) = Y(2) - G / GY(2): GOTO 40
60 GP(1) = 15 * P(1) * P(1) + 6 * P(1) + 10 * P(2) +
     2 * Y(2) * P(1):GP(2) = 10 * P(1) + 6 - 2 * P(2
     )
65 YP(2,1) =  - GP(1) / GY(2):YP(2,2) =  - GP(2) / G
     Y(2)
70  PRINT P(1);"  ";P(2);"  ";Y(2)
150  REM  FINAL VALUES
155 TF = P(1)
160 XT(1) = L(1):XT(2) = L(2)
165 YT(1) = P(2):YT(2) = Y(2)
170 RT(1,1) = 5 - P(2):RT(1,2) = 0
175 RT(2,1) = P(1) - Y(2):RT(2,2) = 0
180 ST(1,1) =  - P(1) * P(1) / 2:ST(1,2) = 1
185 ST(2,1) = YP(2,1) - 5 * P(1) - 3:ST(2,2) = YP(2,
     2)
200  REM  INTEGRATION VARIABLES
205 J = TF
210 XJ(1) = XT(1):XJ(2) = XT(2)
215 YJ(1) = YT(1):YJ(2) = YT(2)
220 RJ(1,1) = RT(1,1):RJ(1,2) = RT(1,2):RJ(2,1) = RT
     (2,1):RJ(2,2) = RT(2,2)
225 SJ(1,1) = ST(1,1):SJ(1,2) = ST(1,2):SJ(2,1) = ST
     (2,1):SJ(2,2) = ST(2,2)
230 H = 0.03
250  REM BACKWARD INTEGRATION
275  REM FOURTH-ORDER RUNGE-KUTTA
300  REM  1ST STEP
305 PX(1) = YJ(1) * H
310 PX(2) = YJ(2) * H
315 PY(1) = XJ(2) * H
320 PY(2) = XJ(1) * H
325 PR(1,1) = SJ(1,1) * H
330 PR(1,2) = SJ(1,2) * H
```

```
]LIST 335-710

335 PR(2,1) = SJ(2,1) * H
340 PR(2,2) = SJ(2,2) * H
345 PS(1,1) = RJ(2,1) * H
350 PS(1,2) = RJ(2,2) * H
355 PS(2,1) = RJ(1,1) * H
360 PS(2,2) = RJ(1,2) * H
400  REM  2ND STEP
405 QX(1) = (YJ(1) - PY(1) / 2) * H
410 QX(2) = (YJ(2) - PY(2) / 2) * H
415 QY(1) = (XJ(2) - PX(2) / 2) * H
420 QY(2) = (XJ(1) - PY(1) / 2) * H
425 QR(1,1) = (SJ(1,1) - PS(1,1) / 2) * H
430 QR(1,2) = (SJ(1,2) - PS(1,2) / 2) * H
435 QR(2,1) = (SJ(2,1) - PS(2,1) / 2) * H
440 QR(2,2) = (SJ(2,2) - PS(2,2) / 2) * H
445 QS(1,1) = (RJ(2,1) - PR(2,1) / 2) * H
450 QS(1,2) = (RJ(2,2) - PR(2,2) / 2) * H
455 QS(2,1) = (RJ(1,1) - PR(1,1) / 2) * H
460 QS(2,2) = (RJ(1,2) - PR(1,2) / 2) * H
500  REM  3RD STEP
505 RX(1) = (YJ(1) - QY(1) / 2) * H
510 RX(2) = (YJ(2) - QY(2) / 2) * H
515 RY(1) = (XJ(2) - QX(2) / 2) * H
520 RY(2) = (XJ(1) - QY(1) / 2) * H
525 RR(1,1) = (SJ(1,1) - QS(1,1) / 2) * H
530 RR(1,2) = (SJ(1,2) - QS(1,2) / 2) * H
535 RR(2,1) = (SJ(2,1) - QS(2,1) / 2) * H
540 RR(2,2) = (SJ(2,2) - QS(2,2) / 2) * H
545 RS(1,1) = (RJ(2,1) - QR(2,1) / 2) * H
550 RS(1,2) = (RJ(2,2) - QR(2,2) / 2) * H
555 RS(2,1) = (RJ(1,1) - QR(1,1) / 2) * H
560 RS(2,2) = (RJ(1,2) - QR(1,2) / 2) * H
600  REM  4TH STEP
605 SX(1) = (YJ(1) - RY(1)) * H
610 SX(2) = (YJ(2) - RY(2)) * H
615 SY(1) = (XJ(2) - RX(2)) * H
620 SY(2) = (XJ(1) - RY(1)) * H
625 SR(1,1) = (SJ(1,1) - RS(1,1)) * H
630 SR(1,2) = (SJ(1,2) - RS(1,2)) * H
635 SR(2,1) = (SJ(2,1) - RS(2,1)) * H
640 SR(2,2) = (SJ(2,2) - RS(2,2)) * H
645 SS(1,1) = (RJ(2,1) - RR(2,1)) * H
650 SS(1,2) = (RJ(2,2) - RR(2,2)) * H
655 SS(2,1) = (RJ(1,1) - RR(1,1)) * H
660 SS(2,2) = (RJ(1,2) - RR(1,2)) * H
700  REM  NEXT STATE
705 XN(1) = XJ(1) - (PX(1) + 2 * QX(1) + 2 * RX(1) +
     SX(1)) / 6
710 XN(2) = XJ(2) - (PX(2) + 2 * QX(2) + 2 * RX(2) +
     SX(2)) / 6
```

```
]LIST 710-

710 XN(2) = XJ(2) - (PX(2) + 2 * QX(2) + 2 * RX(2) +
     SX(2)) / 6
715 YN(1) = YJ(1) - (PY(1) + 2 * QY(1) + 2 * RY(1) +
     SY(1)) / 6
720 YN(2) = YJ(2) - (PY(2) + 2 * QY(2) + 2 * RY(2) +
     SY(2)) / 6
725 RN(1,1) = RJ(1,1) - (PR(1,1) + 2 * QR(1,1) + 2 *
     RR(1,1) + SR(1,1)) / 6
730 RN(1,2) = RJ(1,2) - (PR(1,2) + 2 * QR(1,2) + 2 *
     RR(1,2) + SR(1,2)) / 6
735 RN(2,1) = RJ(2,1) - (PR(2,1) + 2 * QR(2,1) + 2 *
     RR(2,1) + SR(2,1)) / 6
740 RN(2,2) = RJ(2,2) - (PR(2,2) + 2 * QR(2,2) + 2 *
     RR(2,2) + SR(2,2)) / 6
745 SN(1,1) = SJ(1,1) - (PS(1,1) + 2 * QS(1,1) + 2 *
     RS(1,1) + SS(1,1)) / 6
750 SN(1,2) = SJ(1,2) - (PS(1,2) + 2 * QS(1,2) + 2 *
     RS(1,2) + SS(1,2)) / 6
755 SN(2,1) = SJ(2,1) - (PS(2,1) + 2 * QS(2,1) + 2 *
     RS(2,1) + SS(2,1)) / 6
760 SN(2,2) = SJ(2,2) - (PS(2,2) + 2 * QS(2,2) + 2 *
     RS(2,2) + SS(2,2)) / 6
900  REM  END OF INTEGRATION
910 J = J - H
920  PRINT J;"    ";RN(1,2);"    ";RN(2,2)
930  IF J <  = 0 THEN 1000
940 XJ(1) = XN(1):XJ(2) = XN(2)
945 YJ(1) = YN(1):YJ(2) = YN(2)
950 RJ(1,1) = RN(1,1):RJ(1,2) = RN(1,2):RJ(2,1) = RN
     (2,1):RJ(2,2) = RN(2,2)
955 SJ(1,1) = SN(1,1):SJ(1,2) = SN(1,2):SJ(2,1) = SN
     (2,1):SJ(2,2) = SN(2,2)
960  GOTO 250
1000  REM    IS PROBLEM SOLVED?
1010  PR# 1
1015  PRINT
1020  PRINT "   ";RN(1,1);"   ";RN(1,2);"   ";XN(1);"
     ";YT(1);"   ";TF
1025  PRINT "   ";RN(2,1);"   ";RN(2,2);"   ";XN(2);"
     ";YT(2)
1030  PR# 0
1050  IF ( ABS (XN(1)) < 0.1) AND ( ABS (XN(2)) < 0.
     1) THEN  STOP
1100 M = RN(1,1) * RN(2,2) - RN(1,2) * RN(2,1)
1110 RI(1,1) = RN(2,2) / M:RI(1,2) =  - RN(1,2) / M
1120 RI(2,1) =  - RN(2,1) / M:RI(2,2) = RN(1,1) / M
1130 P(1) = P(1) - RI(1,1) * XN(1) - RI(1,2) * XN(2)

1140 P(2) = P(2) - RI(2,1) * XN(1) - RI(2,2) * XN(2)

1200  GOTO 35
```

# BIBLIOGRAPHY


1. MEYER H.G., "Initial Value Methods for Boundary Value Problems", New York and London: Academic Press, 1973
2. ISACSON E., KELLER H.B., "Analysis of Numerical Methods", New York: John Wiley and Sons, Inc., 1966
3. KELLER H.B., "Numerical Methods for TPBVP's", Massachusetts: Blaisdell Publishing Company, 1968
4. BAILEY P.B., SHAMPINE L.F., WALTMAN P.E., "Nonlinear TPBVP's", New York and London: Academic Press, 1968
5. JACOBSON D.H., MAYNE D.Q., "Differential Dynamic Programming", New York: American Elsevier Publishing Company, Inc., 1970
6. BIRKHOFF G., ROTA G.C., "Ordinary Differential Equations", Massachusetts: Blaisdell Publishing Company, 1969
7. KANTORVICH L.V., AKILOV G.P., "Functional Analysis in Normed Spaces", New York: Pergamon Press Ltd., 1964
8. BECKETT R., HURT J., "Numerical Calculations and Algorithms", New York: McGraw-Hill, Inc., 1967
9. INCE E.L., "Ordinary Differential Equations", New York: Dover, 1944
10. KIRK D.E., "Optimal Control Theory", New Jersey: Prentice-Hall, Inc., 1970
11. BRYSON A.E., HO Y.C., "Applied Optimal Control", New York: John Wiley and Sons, Inc., 1975

12. BARTLE R.G., "The Elements of Real Analysis", New York: John Wiley and Sons, Inc., 1976

13. KAPLAN W., LEWIS D.J., "Calculus and Linear Algebra", New York: John Wiley and Sons, Inc., 1971

14. HILDEBRAND F.B., "Methods of Applied Mathematics", New Jersey: Prentice-Hall, Inc., 1965

15. ISTEFANOPULOS Y., "State Variables and Linear Control Systems", Bogazici University, 1981

16. FRANK M.P., "Introduction to System Sensitivity Theory" New York: Academic Press, 1978

17. EYLER M.A., EKSIN I., ISTEFANOPULOS Y., "Sensitivity Approach to TPBVP's", <u>Proceedings MELECON,</u> Vol. 2, 1983