

A Modified Simple Shooting Method for Solving Two-Point Boundary-Value Problems

Raymond Holsapple
Texas Tech University
Lubbock, TX 79409
806-742-1293 x423
rholsapp@math.ttu.edu

Ram Venkataraman
Texas Tech University
Lubbock, TX 79409
806-742-2580 x239
rvenkata@math.ttu.edu

David Doman
Air Force Research Laboratory
Wright-Patterson AFB, OH 45433
937-255-8451
david.doman@wpafb.af.mil

Abstract—In the study of mechanics and optimal control, one often encounters what is called a two-point boundary-value problem (TPBVP). A couple of methods exist for solving these problems, such as the Simple Shooting Method (SSM) and its variation, the Multiple Shooting Method (MSM). In this paper a new method is proposed that was designed from the favorable aspects of both the SSM and the MSM. The Modified Simple Shooting Method (MSSM) sheds undesirable aspects of both previously mentioned methods to yield a superior, faster method for solving TPBVPs. The convergence of the MSSM is proven under mild conditions on the TPBVP. A comparison of the MSM and the MSSM is made for a problem where both methods converge. We also provide a second example where the MSM fails to converge while the MSSM converges rapidly.

Although not the first to investigate the solutions of TPBVPs, one of the first publications to approach this subject was by Keller [2]. Those initial methods were and still are referred to as shooting methods.

Keller [3] develops the SSM and the MSM, referring to the MSM as parallel shooting, and also proposes a version of parallel shooting that he calls "stabilized march." Several years later, J. Stoer and R. Bulirsch [5] explored both the SSM and the MSM in great detail, while providing several examples and hints for practical numerical implementation.

In this paper a new method is proposed for the solution of two-point boundary-value problems that seems to converge faster and more accurately than the MSM. The existence and uniqueness of solutions to the TPBVP is assumed.

Generally speaking, existence and uniqueness theorems for two-point boundary value problems can be quite difficult; however, in this next section we quote two results that broach this topic. The first is from Stoer and Bulirsch [5], and the second is from Keller [2].

TABLE OF CONTENTS

1. INTRODUCTION
2. EXISTENCE AND UNIQUENESS THEOREMS
3. CURRENT METHODS
4. MODIFIED SIMPLE SHOOTING
5. CONCLUSIONS

2. EXISTENCE AND UNIQUENESS THEOREMS

1. INTRODUCTION

A general TPBVP can be written in the following form:

$$y'(x) = f(x, y); \quad a \leq x \leq b \quad (1)$$

$$r(y(a), y(b)) = 0, \quad (2)$$

where (2) describes the boundary conditions satisfied by the system. Examples are the familiar initial-value problem (IVP) and first order necessary conditions obtained by an application of the Pontryagin Maximum Principle in optimal control theory. TPBVPs from optimal control (unconstrained) have separated boundary conditions of the type $r_1(y(a)) = 0$ and $r_2(y(b)) = 0$.

An existence and uniqueness theorem for initial-value problems can be found in Hale [1], which is but one of many texts that provide this well known result. On the other hand, TPBVPs may have multiple or no solution at all. For example, consider the following system:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 1 \\ g(x_1, x_2) \end{bmatrix},$$

where $g(\cdot, \cdot)$ is a continuous function of its arguments, and $x_1(0) = 1, x_1(1) = -1$. One can easily see that the first equation will only allow values of x_1 to increase as time increases. Thus, there does not exist a value for $x_2(0)$ that will drive the value of x_1 from 1 at $t = 0$ to -1 at $t = 1$. Because of this fact, existence and uniqueness theory for TPBVPs is consid-

erably less developed and less understood than that of IVPs. Despite these drawbacks, below are two existence and uniqueness theorems that are applicable to much smaller classes of functions $f(x, y)$.

General Boundary Conditions

Theorem 2.1: For the two-point boundary-value problem (1)-(2), let the following assumptions be satisfied:

1. f and $D_y f$ are continuous on $S = \{(x, y) | a \leq x \leq b, y \in \mathbb{R}^n\}$
2. There is a $k(\cdot) \in C[a, b]$ with $\|D_y f(x, y)\| \leq k(x)$ for all $(x, y) \in S$.
3. The matrix

$$P(u, v) = D_u r(u, v) + D_v r(u, v)$$

admits for all $u, v \in \mathbb{R}^n$ a representation of the form $P(u, v) = P_0(I + M(u, v))$ with a constant nonsingular matrix P_0 and a matrix $M = M(u, v)$, and there are constants μ and m with $\|M(u, v)\| \leq \mu < 1$, $\|P_0^{-1} D_v r(u, v)\| \leq m$ for all $u, v \in \mathbb{R}^n$.

4. There is a number $\lambda > 0$ with $\lambda + \mu < 1$ such that $\int_a^b k(t) dt \leq \ln(1 + \frac{\lambda}{m})$. Then the boundary value problem (1) has exactly one solution $y(x)$.

Proof: For a proof of this theorem, consult Stoer and Bulirsch [5], page 510. ■

Separated Boundary Conditions

For a theorem that will apply to separated boundary conditions, we consult Keller [2]. Consider the following second-order system:

$$\begin{aligned} y'' &= f(x, y, y'); & a \leq x \leq b \\ a_0 y(a) - a_1 y'(a) &= \alpha, & |a_0| + |a_1| \neq 0; \\ b_0 y(b) + b_1 y'(b) &= \beta, & |b_0| + |b_1| \neq 0. \end{aligned} \quad (3)$$

Theorem 2.2: Let the function $f(x, y, y')$ in (3) satisfy the all of the following:

1. $f(x, y, y')$ is continuous on $D = \{(x, y, y') | a \leq x \leq b, y^2 + (y')^2 < \infty\}$
2. $f(x, y, y')$ satisfies a uniform Lipschitz condition on R in y and y' .
3. $f(x, y, y')$ has continuous derivatives on D which satisfy, for some positive constant M ,

$$(a) \quad \frac{\partial f}{\partial y} > 0$$

$$(b) \quad \left| \frac{\partial f}{\partial y'} \right| \leq M$$

4. The coefficients in (3) satisfy $a_0 a_1 \geq 0$, $b_0 b_1 \geq 0$, $|a_0| + |b_0| \neq 0$

Then the boundary-value problem in (3) has a unique solution.

Proof: For proof of this theorem, consult Keller [2], page 9. ■

Remark. Assumptions 1-4 of Theorem 2.1 are very restrictive sufficient conditions. Even simple boundary conditions exist that do not satisfy assumption 3; such is the case with separated boundary conditions.

Optimal Control

Now consider the optimal control problem of finding a $u(\cdot)$ for the following system:

$$\dot{x} = f(x, u), \quad x(0) = x_0, \quad x(1) = x_1, \quad (4)$$

such that

$$J(u) = \int_0^1 L(x, u) dt$$

is minimized. The Pontryagin Maximum Principle yields the existence of functions

$$p(t) = [p_1(t) \ p_2(t) \ \cdots \ p_n(t)]^T$$

with $t \in [0, 1]$; $H(x, u, p) = L(x, u) + p^T f(x, u)$, and $u^* = \arg \min_u H(x, u, p)$ such that

$$\begin{bmatrix} \dot{x} \\ \dot{p} \end{bmatrix} (t) = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \nabla H(x, u^*, p)$$

satisfies $x(0) = x_0$ and $x(1) = x_1$. If $\nabla H(x, \arg \min_u H(x, u, p), p)$ is Lipschitz continuous in the x and p variables then we have uniqueness. A sufficient condition is the twice differentiability of $H(x, u, p)$.

Now that we have proof of existence and uniqueness for small classes of TPBVPs, let's explore the current methods commonly used to numerically solve such a problem and take a look at the new method that we propose.

3. CURRENT METHODS

Although Theorem 2.1 does not apply to the case of separated boundary conditions and Theorem 2.2 itself

may be somewhat restrictive, separated boundary conditions are the most commonly encountered in optimal control. Because of this, separated boundary conditions will be used for explanation purposes. The system now becomes

$$\begin{aligned} y'(x) &= f(x, y); & a \leq x \leq b \\ Ay(a) &= \alpha, & By(b) = \beta, \end{aligned} \quad (5)$$

where A and B are $m \times n$ matrices with $\text{rank}(A) + \text{rank}(B) = n$.

Simple Shooting

The Simple Shooting Method, as the name implies, is the simplest method of finding a solution to such a problem. The idea is to convert (5) into an initial-value problem (IVP):

$$\begin{aligned} y'(x) &= f(x, y); & a \leq x \leq b \\ y(a) &= y_a, \end{aligned} \quad (6)$$

where y_a is composed of known states from $Ay(a) = \alpha$ and guesses for the unknown states s_0 . Now, $y(x) \in \mathbb{R}^n$, $\alpha \in \mathbb{R}^m$, and $\beta \in \mathbb{R}^p$. A necessary condition to keep the problem from being inconsistent is that $m + p = n$. To form an IVP out of (5), one needs to guess initial conditions for the $(n - m)$ components of $y(x)$ that do not already have initial conditions at $x = a$. Let $s_0 \in \mathbb{R}^{n-m}$ be the guess for the unknown initial conditions and s_k ; $k \geq 1$ subsequent corrections of the vector s_0 . With s_0 , one can now integrate (6) forward in the time variable x . Integration is performed from $x = a$ all the way to $x = b$. Then compute the error $e = \|By(b) - \beta\|^2$. With this information, a correction is made to the initial guess s_0 to yield s_1 , and the integration is performed again. This process is repeated over and over until $e < \varepsilon$, where $\varepsilon > 0$ is small. How the correction is made will be addressed shortly.

For illustration purposes only, consider Figure 1 which represents an example of Simple Shooting for (6). We assume that there exists a unique solution to the problem. Every point on the plot represents a vector in \mathbb{R}^p .

There can be serious problems with the accuracy of the SSM. The problems occur when making the correction to the s_k vector. This vector is usually corrected using a modified Newton's Method, and in practice the system must be linearized to use this method. If e is large, then convergence can be quite slow (please refer to page 511 of Stoer and Bulirsch [5]). This drawback of the SSM can be fixed by implementing what is known as the Multiple Shooting Method.

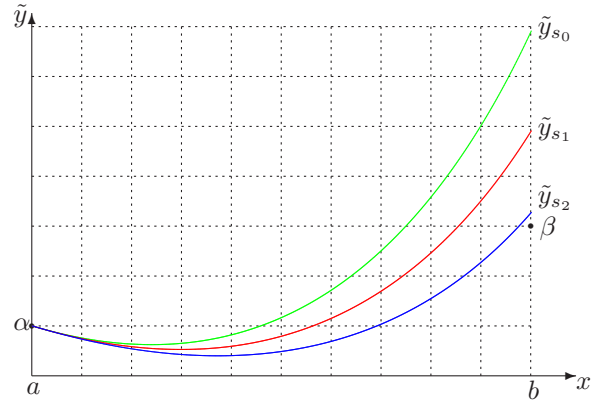


Figure 1. Illustration of the Simple Shooting Method

Multiple Shooting

The Multiple Shooting Method begins with the choice of a Lipschitz continuous function $\varphi(x)$ that satisfies $A\varphi(x) = \alpha$ and $B\varphi(x) = \beta$. An initial guess of unknowns, s_0 , must be made. Then, (6) is integrated until $\|y(x, s_0) - \varphi(x)\| > \varepsilon$ for some $\varepsilon > 0$. We designate the time variable at this point as x_1 . Now the integration of the system continues from x_1 using $\varphi(x_1)$ as the initial 'guess' for the solution. This process continues until the integration reaches $x = b$. Now the error function $e(s) = \|y(x_i) - y_i\|^2$ is formulated where $s = [s_0 \ y_1 \ \dots \ y_{k-1}]^T$ and y_i is the initial state for the trajectory in the interval $[x_i, x_{i+1}]$. After this is accomplished, a correction is made to s using a modified Newton's method, and the process is repeated. The starting trajectory is not used after the first iteration.

Figure 2 illustrates the MSM. Once again, three iterations are displayed for this method. The correction process stops when $e(s) < \varepsilon_1 < \varepsilon$ for some $\varepsilon_1 > 0$.

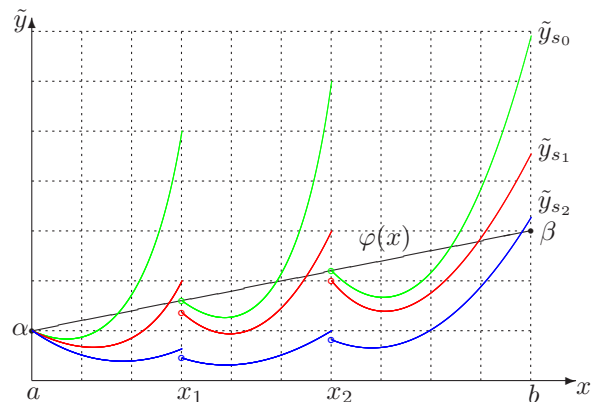


Figure 2. Illustration of the Multiple Shooting Method

A problem with the MSM is the discontinuity of the trajectory found by the MSM at the points x_i ; $i = 1, \dots, k - 1$. The integration and corrections of s will continue until a desired level of closeness is determined, but this final value of the vector s can still be far from an optimal solution due to the unstable nature of many systems in the forward direction. If (6) is re-integrated to result in one continuous trajectory for the system, the end values $By(b)$ need not be anywhere close to β and almost certainly will not be. Another problem is computation. During the process, one must invert many matrices of the size $[n(k - 1) + p]$, where k can be quite large depending on the guesses $[s_0 \ \varphi(x_1) \cdots \varphi(x_{k-1})]^T$ for the initial trajectory. Note that k cannot be reduced even as the guesses improve.

Example 3.1: Consider the following system:

$$\begin{bmatrix} y_1'(x) \\ y_2'(x) \\ y_3'(x) \\ y_4'(x) \end{bmatrix} = \begin{bmatrix} y_3(x) \\ y_4(x) \\ y_2(x) \\ y_1(x) \end{bmatrix}$$

$$\begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} y_1(1) \\ y_2(1) \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \quad (7)$$

where $0 \leq x \leq 1$. This system was solved with the ‘bad’ initial guess $s_0 = [-100 \ 2]^T$ with the parameters of the code as follows:

- The time step $h = 0.01$.
- ε was set to 1.
- Newton’s Method is stopped when the solution is found to be in an ε_1 ball of size $\varepsilon_1 = 10^{-3}$

For this example $\varphi(x) = x[1 \ 1]^T + [1 \ 1]^T$. The following results were obtained. The MSM corrects s_0 to $[0.70 \ 0.70]^T$ in 8.543 seconds. Figure 3 shows the discontinuous segments obtained after the convergence of the iterations. One can see the discontinuous segments connecting the given initial and desired final states. However, when this very system is re-integrated using the s_0 vector determined to be the correct one by the MSM, the first two components of the solution do not end up within 10^{-3} of $[2 \ 2]^T$ (please see Figure 4). Instead, the solution ends up being close to $[2.25 \ 2.25]^T$. Obviously, these are not desirable results.

Note that while using the MSM *one does not have control over the error in the states at the final time*. It depends on the particular system being considered. For the sample problem, in order for the error between the

actual and desired final states to be small, we reduced the time step to $h = 0.001$. This time MSM corrects the unknown states s_0 to $[0.42 \ 0.42]^T$ in 16.634 seconds. Figure 5 shows both the result of MSM and the final trajectory after re-integration.

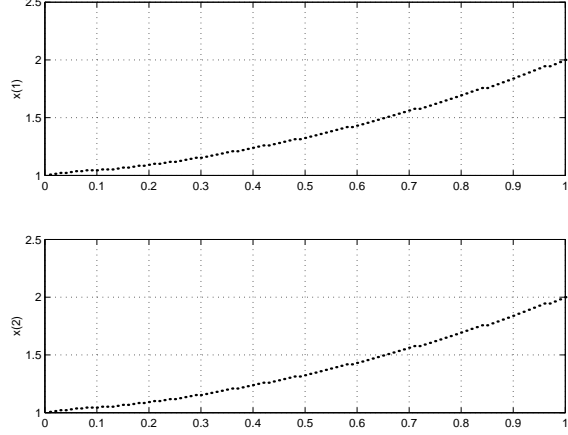


Figure 3. Discontinuous segments connecting the initial and desired final states while using the MSM.

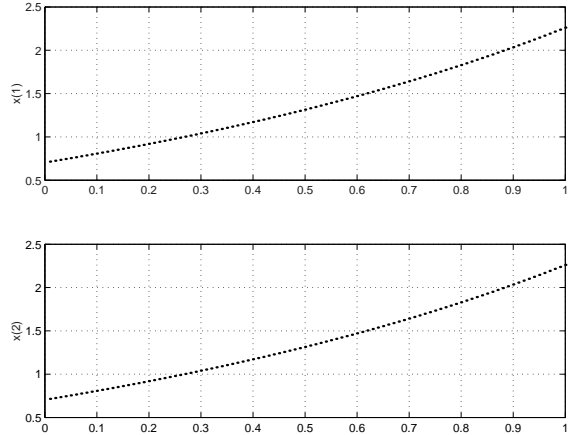


Figure 4. Plot of the states while using the result of the MSM ($h = 0.01$ seconds).

4. MODIFIED SIMPLE SHOOTING

Description of Algorithm

The Modified Simple Shooting Method begins with the selection of a Lipschitz continuous starting path of integration, $\varphi(x)$, such that $A\varphi(a) = \alpha$ and $B\varphi(b) = \beta$. Again, an initial guess of unknowns, s_0 must be made. Then, (6) is integrated until $\|y(x, s_0) - \varphi(x)\| > \varepsilon$ for some $\varepsilon > 0$. Then we designate $s_{1_0} = s_0$. A modified Newton’s Method is then used to correct the ‘guess’ s_{1_0} . The iteration stops when $\|y(x, s_{1_k}) - \varphi(x)\| < \varepsilon_1$ where ε_1 is chosen such that $\varepsilon_1 < \varepsilon$. We then let $s_1 = s_{1_k}$, and proceed with the integration of the

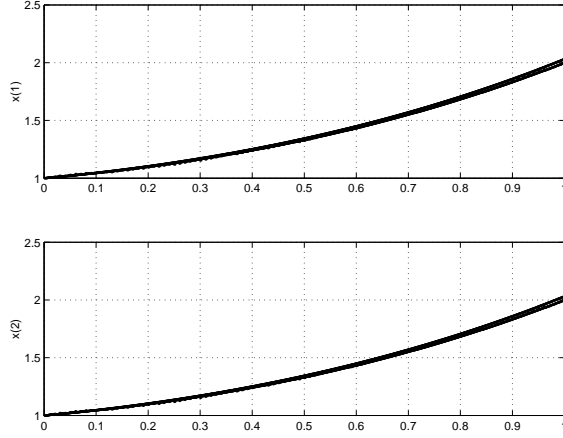


Figure 5. Plot of the states while using the result of the MSM ($h = 0.001$ seconds).

system,(6) where $y(a)$ is found using $Ay(a) = \alpha$ and s_1 .

The modified Newton's Method mentioned above is found in Stoer and Bulirsch [5]. The following is an outline of that method for the first iteration:

1. Choose a starting point $s_0 \in \mathbb{R}^{n-m}$
2. For each $i = 0, 1, \dots$ define s_{i+1} from s_i as follows:
 - (a) Set

$$d_i = DF(s_{1_i})^{-1}F(s_{1_i}),$$

$\gamma_i = \frac{1}{\text{cond}(DF(s_{1_i}))}$, and let $h_i(\tau) = h(s_{1_i} - \tau d_i)$, where $h(s) = F(s)^T F(s)$. Determine the smallest integer $j \geq 0$ satisfying $h_i(2^{-j}) \leq h_i(0) - 2^{-j} \frac{\gamma_i}{4} \|d_i\| \|Dh(s_{1_i})\|$.

(b) Determine λ_i so that $h(s_{1_{i+1}}) = \min_{0 \leq \kappa \leq j} h_i(2^{-\kappa})$, and let $s_{1_{i+1}} = s_{1_i} + \lambda_i d_i$.

The MSSM continues until $\|y(x, s_q) - \varphi(x)\| < \varepsilon$ at each $x \in [a, b]$. In this last step, we are performing exactly the SSM for the original system, but with a starting initial guess s_q that keeps $By(b)$ close to β . This will prevent any numerical divergence.

Figure 3 illustrates the Modified Simple Shooting Method. In this case, it took three overall 'shots' to integrate from $x = a$ to $x = b$.

Convergence of the MSSM

Theorem 2.1 provided us with a somewhat limited existence and uniqueness theorem. Theorem 2.2 was more useful in the sense that it applied to the case of separated boundary conditions. For the purpose of this section, we shall assume the existence of a solution to

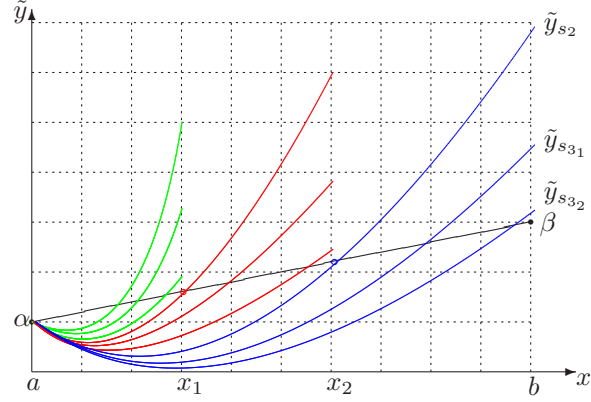


Figure 6. Illustration of the Modified Simple Shooting Method

(5) for an entire family of boundary conditions. More specifically, suppose that the BVPs

$$y'(x) = f(x, y)$$

$$Ay(a) = \alpha, \quad By(x) = B\varphi(x)$$

have unique solutions where $\varphi(\cdot)$ was the function chosen initially. This is necessary to assume, as a solution to the overall problem may not directly imply the existence of a solution to one of the intermediate reduced problems. Now we can consider the main theorem of this report on the convergence of the Modified Simple Shooting Method with certain assumptions.

Theorem 4.1: Consider the Two-Point Boundary-Value Problem as described in (5). Let $y(x)$ denote the solution to this problem. The Modified Simple Shooting Method, as described earlier, converges to $y(x)$ when applied to (5).

Proof: In order for the MSSM to converge to $y(x)$, it must first converge to $\bar{y}_i(x)$ at each intermediate point x_i , where $i = 1, 2, \dots, k-1$, and $\bar{y}_i(x)$ is the solution of the reduced problem on $[a, x_i]$ for $i = 1, 2, \dots, k-1$. For example, $\bar{y}_1(x)$ is the solution to the problem

$$\bar{y}'_1(x) = f(x, \bar{y}_1), \quad a \leq x \leq x_1$$

$$Ay(a) = \alpha, \quad By(x_1) = \varphi(x_1)$$

where $\varphi(x)$ is the reference path mentioned in the description of the algorithm. As such, it is only necessary to show two things to complete this proof.

1. The sequence of points $\{x_n\}_{n=1}^{\infty} \in \mathbb{R}$ converges to the right endpoint b .
2. The SSM converges when existence is known.

The latter is a result of the modified Newton's Method, which is guaranteed to find a solution for large classes

of functions, if it exists. As existence of a solution $y(x)$ is being assumed, the modified Newton's Method guarantees that the SSM converges to $y(x)$.

Now assume that after the modified Newton's Method is performed at $x \in [a, b]$, the integration proceeds to $x^* \geq x$. It is necessary to show that $(x^* - x)$ is bounded below by a positive number. $y(x, s_m)$ is a continuous function of x and $\varphi(x)$ is a Lipschitz continuous function of x . Thus, $d(\varphi; y, s_m)(x) = \|y(x, s_m) - \varphi(x)\|$ is a Lipschitz continuous function of x . By the compactness of $[a, b]$, there is a uniform Lipschitz constant, k for $d(\varphi; y, \cdot)$. Thus, $|d(\varphi; y, s_m)(x) - d(\varphi; y, s_m)(x^*)| \leq k|x - x^*|$. When the integration of the system stops, $|d(\varphi; y, s_m)(x) - d(\varphi; y, s_m)(x^*)| = \varepsilon - \varepsilon_1$. Then $\varepsilon - \varepsilon_1 \leq k(x^* - x)$, or $(x^* - x) \geq \frac{\varepsilon - \varepsilon_1}{k} > 0$. This means that the integration continues past x , and by the compactness of $[a, b]$, the process must extend all the way to $x = b$. ■

Examples

Remark. All computations in the following examples were performed in the MATLAB environment, Version 6.1.0.450 Release 12.1, running on a Microsoft Windows 2000 Professional operating system with an AMD Athlon processor running at 1.2 GHz.

A Linear Example—

Example 4.1: Consider the simple system of Example 3.1. The MSSM was applied to the same system with the same parameters in the code.

The MSSM finds a solution in 1.001 seconds with s corrected to $[0.39 \ 0.39]^T$, which is about 8 times faster than the MSM. The corrections to s by both methods do not give the same result. The discontinuity problem does not arise with the MSSM, since the final trajectory is a continuous one.

*An Application in $SO(3)$ —*Now we will look at an example in the three dimensional special orthogonal group or $SO(3)$. $SO(n)$ is defined as follows:

$$SO(n) = \{A \in \mathbb{R}^{n \times n} \mid \det A = 1, A^T A = I_{n \times n}\}.$$

This example is actually an optimization problem on the set of orientation matrices in three dimensional space. Consider the following system.

$$\begin{aligned} \dot{Q} &= Q\hat{\Omega} \\ \begin{bmatrix} \dot{\Omega} \\ \dot{P}_1 \\ \dot{P}_2 \end{bmatrix} &= \begin{bmatrix} P_2 \\ -\frac{1}{2}P_1 \times \Omega + \Omega \times \frac{1}{4}(P_2 \times \Omega) \\ -\frac{1}{2}P_2 \times \Omega - P_1 \end{bmatrix} \end{aligned} \quad (8)$$

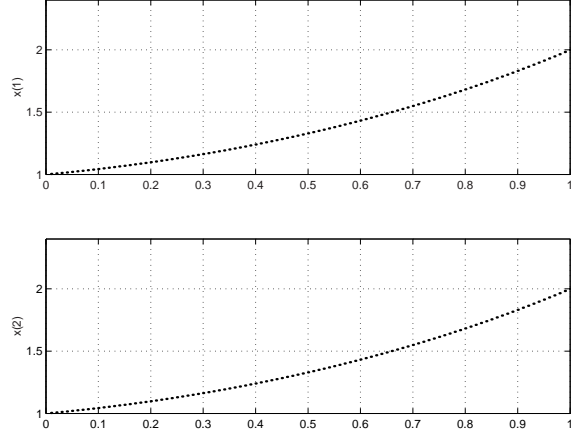


Figure 7. Plot of the states for the trajectory planning problem using MSSM.

Q, Ω, P_1 , and P_2 are all functions of time, but the dependence on time, t , has been suppressed for ease of writing. Ω, P_1 , and P_2 are vectors in \mathbb{R}^3 , whereas Q is a matrix in $\mathbb{R}^{3 \times 3}$. $\hat{\Omega}$ is a little more complicated; it is a skew-symmetric matrix formed from the vector Ω as such.

If

$$\Omega = \begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \end{bmatrix},$$

then

$$\hat{\Omega} = \begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix}.$$

$Q_{initial}, \Omega_{initial}, Q_{desired}$ and $\Omega_{desired}$ are known. The solution is sought for $0 \leq t \leq 1$.

For this particular example, further obstacles were to be overcome. It was required that $Q(t) \in SO(3)$ at every instant of time t . Furthermore, there was the obstacle that we must integrate forward in time four equations, three of which are vector valued equations and one matrix valued equation. But, the matrix equation depended only upon the value of $\hat{\Omega}$ and hence Ω at each instant of time. Because of this and the fact that none of the other equations depend on Q , it was possible to integrate the matrix equation separately, but still forward in time at the same time as the vector equations. To keep $Q(t) \in SO(3)$, Rodrigues' formula was used, which can be found in Murray, Li, and Sastry [4]. The formula is

$$e^{\hat{\Omega}\Theta} = I + \hat{\Omega} \sin \Theta + \hat{\Omega}^2 (1 - \cos \Theta).$$

The initial and final values of Q and Ω were generated randomly by MATLAB and were exactly the same for

both the MSSM and the MSM. Those values were

$$Q_{initial} = \begin{bmatrix} 0.57 & 0.82 & 0.08 \\ -0.78 & 0.57 & -0.25 \\ -0.25 & 0.08 & 0.97 \end{bmatrix},$$

$$\Omega_{initial} = \begin{bmatrix} 0.95 \\ 4.34 \\ 7.09 \end{bmatrix},$$

$$Q_{desired} = \begin{bmatrix} -0.27 & 0.96 & -0.01 \\ -0.84 & -0.23 & 0.50 \\ 0.48 & 0.14 & 0.87 \end{bmatrix},$$

$$\Omega_{desired} = \begin{bmatrix} 1.90 \\ 8.67 \\ 4.18 \end{bmatrix}.$$

The results for this example again heavily favored the Modified Simple Shooting Method. After 47.12 seconds, the MSSM obtained the following values for Ω and Q at time $t = 1$,

$$Q(1) = \begin{bmatrix} -0.27 & 0.96 & -0.01 \\ -0.84 & -0.23 & 0.50 \\ 0.48 & 0.14 & 0.87 \end{bmatrix},$$

$$\Omega(1) = \begin{bmatrix} 1.90 \\ 8.67 \\ 4.19 \end{bmatrix}.$$

The Multiple Shooting Method proved inadequate for this problem. After 606.89 seconds, the MSM obtained these results,

$$Q(1) = \begin{bmatrix} 0.09 & -0.01 & 1.00 \\ 0.99 & -0.06 & -0.09 \\ 0.06 & 1.00 & 0.00 \end{bmatrix},$$

$$\Omega(1) = \begin{bmatrix} 2.10 \\ 10.81 \\ 5.89 \end{bmatrix}.$$

To measure closeness the normal Euclidean norm can be used to compare $\Omega_{desired}$ and $\Omega(1)$; however, this is not the case with $Q_{desired}$ and $Q(1)$. One must be more careful. It is desired to have a measure of closeness within the group $SO(3)$, not the space of all 3×3 matrices. To do so, we take the matrix logarithm of the quantity $Q(1)^T Q_{desired}$, which will yield a skew-symmetric matrix. We then take the norm of this matrix. The Multiple Shooting Method yields

$$\|\Omega_{desired} - \Omega(1)\| = 2.74,$$

$$\|\log(Q(1)^T Q_{desired})\| = 2.62.$$

The Modified Simple Shooting Method yields

$$\|\Omega_{desired} - \Omega(1)\| = 0.0013,$$

$$\|\log(Q(1)^T Q_{desired})\| = 0.0003.$$

The MSM took more than 10 times as long and the results are not close at all to the desired values; these results speak for themselves.

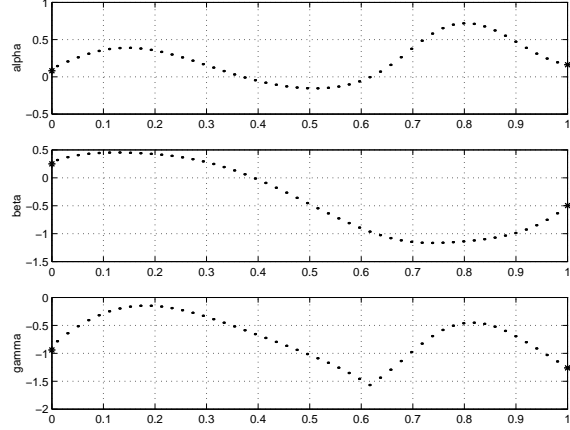


Figure 8. Plot of ZYX Euler angles.

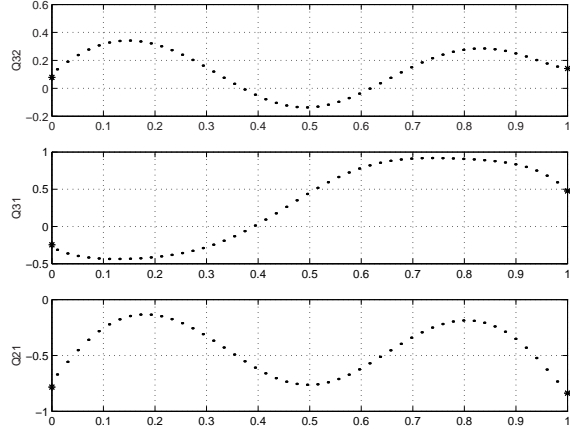


Figure 9. Q matrix components.

5. CONCLUSIONS

In this report, a new method for solving two-point boundary-value problems was described. Although convergence is not so laborious to investigate, it was shown by two examples of theorems how difficult it is to prove existence and uniqueness for two-point boundary-value problems. Nonetheless, three examples were given, among many performed, that clearly show that the Modified Simple Shooting Method performs better and faster than the Multiple Shooting Method.

First, it requires the inversion of much smaller matrices than those required to be inverted in the MSM. The MSSM requires the inversion of matrices that are $n \times n$;

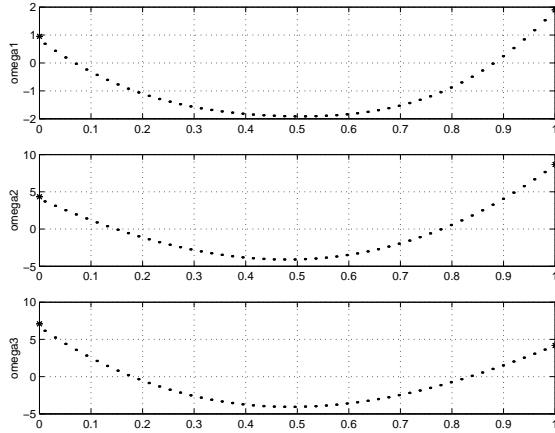


Figure 10. Angular velocity.

whereas, the MSM requires the inversion of matrices that are $[n(k-1)+p] \times [n(k-1)+p]$. This fact alone could account for many seconds of computation time saved as systems become larger and larger.

Another fact that makes the MSSM more appealing is continuity of integration trajectory. This property is very important in optimal control problems where the systems are unstable in forward time. The Modified Simple Shooting Method integrates the system in one continuous path every time it shoots the system for an updated s vector. The Multiple Shooting Method does not have this characteristic. In fact, for a particular example of the MSM, if k intermediate shots are taken then every overall shot of the system from $x = a$ to $x = b$ will consist of $k - 1$ discontinuities. Each of these discontinuities are impossible to correct. The best the method has to offer is to reduce the magnitude of the discontinuities. Due to the instability of many systems in the forward direction, an erroneous solution may be obtained if the system is re-integrated with what is supposed to be an accurate approximation to the actual unknown initial conditions.

REFERENCES

- [1] J. Hale. *Ordinary Differential Equations*. Krieger Publishing Company, 1980.
- [2] Herbert B. Keller. *Numerical Methods For Two-Point Boundary-Value Problems*. Blaisdell, 1968.
- [3] Herbert B. Keller. *Numerical Solution of Two Point Boundary Value Problems*. SIAM, 1976.
- [4] Zexiang Li Richard M. Murray and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press LLC, 1994.
- [5] J. Stoer and R. Bulirsch. *Introduction to Numerical*

Analysis. Springer-Verlag, second edition, 1993.



Raymond Holsapple is currently pursuing a doctoral degree in Mathematics at Texas Tech University. His interests include control theory, numerical analysis, and the theory of ordinary and partial differential equations. He is on inactive reserve for the Air Force and worked at the WPAFB, OH, on an AFRL graduate student fellowship, during the summer of 2002.



Ram Venkataraman currently holds an Assistant Professor position in the Department of Mathematics and Statistics at Texas Tech University. He has a Ph.D. in Electrical Engineering from the University of Maryland at College Park. His research interests include differential geometry and functional analysis, and their applications to system, control theory and smart structures.



David Doman is a senior Aerospace Engineer at the Control Theory and Optimization Branch, Control Sciences Division, Air Force Research Laboratory, WPAFB, OH. He is the technical lead for the hypersonics program at AFRL. He has a Ph.D. in Aerospace engineering from Virginia Tech, and is a senior member of AIAA.