A New, Fast Numerical Method for Solving Two-Point Boundary Value Problems

Raymond Holsapple[*], Ram Venkataraman[†]

Texas Tech University, Lubbock, TX 79409-1042.

David Doman [‡]

Wright-Patterson Air Force Base, Ohio 45433-7531.

Introduction

In physics and engineering, one often encounters what is called a two-point boundary-value problem (TPBVP). A number of methods exist for solving these problems including shooting, collocation and finite difference methods.[1,2] Among the shooting methods, the Simple Shooting Method (SSM) and the Multiple Shooting Method (MSM) appear to be the most widely known and used methods.

In this paper, a new method is proposed that was designed to include the favorable aspects of the Simple and the Multiple Shooting methods. This Modified Simple Shooting

[*]Graduate Student, Department of Mathematics and Statistics.

[†]Assistant Professor, Department of Mathematics and Statistics.

[‡]Aerospace Engineer, U.S. Air Force Research Laboratory. Senior Member AIAA

Method (MSSM) sheds undesirable aspects of these methods to yield a fast and accurate method for solving TPBVPs. The convergence of the modified simple shooting method is proved under mild conditions on the TPBVP. A comparison of the MSSM, MSM, collocation (CM) and finite difference methods (FDM) is made for a simple example for which all these methods converge. Further comparison between the MSM and the MSSM can be found in our earlier work,[3] where we studied an optimal control problem with fixed end-points for a non-linear system. For that problem it was shown that the MSM failed to converge while the MSSM converged rapidly.

A general TPBVP can be written in the following form:

$$\dot{y}(t) = f(t, y)\,; \quad a \leq t \leq b \tag{1}$$

$$r[y(a), y(b)] = 0, \tag{2}$$

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 have separated boundary conditions of the type $r_1[y(a)] = 0$ and $r_2[y(b)] = 0$.

Some of the initial publications that deal with TPBVPs are Keller,[4,5] and Roberts and Shipman.[1] Provided it converges, the SSM is the simplest, fastest and most accurate method to solve TPBVPs. However, it is well known that the SSM can fail to converge for problems whose solutions are very sensitive to initial conditions. For such problems, finite difference (FDM) and collocation (CM) methods can provide a solution that satisfies the boundary conditions and is close to the actual solution in some sense. This led to the development of the MSM.[6] Morrison, Riley and Zancanaro[6] first proposed the MSM as a compromise between the SSM and the finite difference methods. Keller[5] refers to the MSM as parallel shooting, and also proposed a version of parallel shooting that he called "stabilized march." The FDM and CM schemes are much harder to set up than the shooting methods.[1] For nonlinear problems, quasi-linearization is used along with finite difference schemes.[7]

In this note, we restrict our attention to problems with no constraints. We compare existing methods with the MSSM in terms of computation times and accuracy of solutions, and show that the MSSM is superior. *Once a numerical method has converged, the norm of the difference between the desired final states and the actual final states obtained after a re-integration of Equation (1) is our criterion for accuracy of the solution.*

Summary of Existing Methods

As separated boundary conditions are commonly encountered in optimal control, we will consider such boundary conditions in the rest of the paper. However, the methods described below can be extended to more general boundary conditions of the type $F_0 y(a) + F_1 y(b) = \alpha$, where F_0 and F_1 are matrices such that $\text{rank}(F_0) + \text{rank}(F_1) = n$, where n is the length of the vector y. Thus the system under consideration is:

$$\dot{y}(t) = f(t, y), \quad a \leq t \leq b, \quad y(t) \in \mathbb{R}^n \quad \text{for each } t; \tag{3}$$

$$Ay(a) = \alpha, \quad By(b) = \beta, \tag{4}$$

where A and B are $m_A \times n$ and $m_B \times n$ matrices with $m_A = \text{rank}(A)$, $m_B = \text{rank}(B)$, and $m_A + m_B = n$. If $m_A = n$, then we have an initial value problem. In the presentation that follows, we assume $n \geq 2$ and $m_A < n$. Though not necessary from a theoretical point of view, it is very useful in practice to have $B = [I_{m_B \times m_B} \; 0_{m_B \times m_A}]$. This can be achieved by a co-ordinate transformation that puts B in the above form.

Though it is difficult to establish existence and uniqueness for TPBVPs in general, for certain problems that arise from a variational principle (including optimal control problems) one can deduce such properties.[8] As we are interested in numerical methods, we will make some assumptions that ensure the well-posedness of our algorithms.

The remainder of the paper makes use of two assumptions.

Assumption 1 *There exists a unique solution to the TPBVP (3-4).*

Assumption 2 *If $y^*(a)$ denotes the initial condition that leads to the solution of the TP-BVP, then there exists a unique solution defined on $[a, b]$ for every initial condition in a sufficiently small neighborhood of $y^*(a)$. Furthermore, the solution is continuously differentiable with respect to changes in the initial condition.*

These assumptions imply that the matrix $\left.\frac{\partial f(t,y)}{\partial y}\right|_{y=y^*(t)}$ exists and is bounded for $t \in [a, b]$, where $y^*(t)$ denotes the optimal solution. The second assumption ensures that numerical methods based on a modified Newton's method will be convergent.

The Simple Shooting Method transforms a TPBVP into an initial value problem where the initial values of selected parameters are varied to satisfy the desired end conditions.[5] A very desirable property of the SSM (provided that it converges) is that the resulting solution is a continuously differentiable function that satisfies the Equations (3-4). This means that the boundary conditions (4) are satisfied when Equation (3) is integrated over $a \leq t \leq b$ using the initial condition obtained using the SSM. It should be noted that there can be serious problems with the convergence of the SSM, if the starting initial condition $y(a)$ is not close to $y^*(a)$. As we have no way of knowing $y^*(a)$ before hand, the SSM is not a practical method for many applications. This drawback of the SSM can be addressed by implementing what is known as the Multiple Shooting Method (MSM).

The MSM is similar to the SSM, in that one selects unknown parameters at the initial time; however, one does not integrate Equation (3) all the way to the final time. Instead, the "distance" from a corresponding point on a pre-selected reference path is checked continuously as the integration proceeds, and the integration is aborted when the distance exceeds a tolerance value. Then, one starts the integration again from the corresponding point on the reference path and the previous step is repeated, until the system is integrated to the final time. An equation is then formed to "match up" the discontinuous trajectory segments, and a modified Newton's method is used to reduce the "gaps". An advantage of this approach over the SSM is that convergence can now be obtained for a larger class of TPBVPs.[9] A major disadvantage of this method is that the number of parameters to be updated in each iteration can be very large, leading to larger computation times when compared to the SSM (provided that it converges). During each run, one must invert matrices whose row and column dimensions are a linear function of the number of shooting nodes. The number of nodes can be quite large depending on the guesses for the initial unknown parameters. It is also important to note that the number of nodes cannot be reduced, even as the guesses improve. Another serious disadvantage of this method is that if the differential equations are re-integrated to result in one continuous trajectory for the system, the actual final values may not be close to the desired final values. This is a common problem when solving TPBVPs that result from optimal control, due to instability of the systems in the forward direction. For more detail on the multiple shooting method, please refer to Stoer and Bulirsch.[9]

As we mentioned earlier, the FDM and CM are far more complex to set up. For linear systems, the FDM transforms the TPBVP to a linear algebra problem. The FDM is often used in conjunction with quasi-linearization for nonlinear systems. A very good discussion of the Finite Difference schemes can be found in Keller,[5] Roberts and Shipman,[1] while a description of the collocation methods can be found in Reinhardt.[2] The Collocation Method tested in this paper was the 'bvp4c' routine in MATLAB.[10]

Proposed Modified Simple Shooting Method

The Modified Simple Shooting Method (MSSM) combines the attractive quality of convergence for large classes of systems of the MSM, with the satisfaction of the boundary conditions on re-integration property and fast computation times of the SSM. Just like the MSM, there is a choice of a reference path that has the effect of improving convergence.

Again, one has to choose unknown parameters at the initial time and integrate the system (3) forward in time, while checking the distance from corresponding points on the reference path (in some cases it may be more appropriate to integrate backward in time).[3] The integration is aborted when the distance becomes larger than some tolerance value, just as in the multiple shooting method. Next, the unknown parameters at the initial time are updated so that the trajectory "passes through" a chosen point on the reference path. This procedure amounts to performing simple shooting on a smaller time interval. This process is repeated as described in the algorithm below on progressively larger intervals of time, until we perform simple shooting on the entire time interval. Thus only the exact number of unknown initial parameters have to be updated in each iteration, which leads to faster convergence. Furthermore, the final trajectory is a solution of the given system and it exactly connects the initial and final states. This feature of the MSSM contrasts with the MSM where unavoidable discontinuities may result in significant differences between the desired and actual final state when the resulting approximate solution is used to integrate Equation (3). The Modified Simple-Shooting algorithm is given below.

Initialization: Choose a distance metric $d(\cdot,\cdot)$ for the space $\mathbb{R}^n$. Next, choose a Lipschitz continuous $\mathbb{R}^n-$valued function $\varphi(t)$ (that is: $d[\varphi(t_1),\varphi(t_2)] \leq K_1 |t_1 - t_2|$ for some $K_1 > 0$) that satisfies $A\varphi(a) = \alpha$ and $B\varphi(b) = \beta$. As rank$(A) = m_A$, the equation $A\varphi(a) = \alpha$ can be solved to obtain m_A of the initial states in terms of the other m_B states that are now treated as parameters. At the k-th iteration, we denote this parameter vector of length m_B as s_{k-1}.

1. (*At step 1:*)Choose the parameter vector for the first step $s_0 \in \mathbb{R}^{m_B}$, and compute $\varphi(a)$. The initial vector for Step 1 of the algorithm is $y(a) = \varphi(a)$. Denote $t_0 = a$.

2. (*At step k:*) Solve the system

$$\dot{y}(t) = f(t,y); \quad a \leq t \leq b \tag{5}$$

The initial states $y(a)$ are determined from the parameter vector s_{k-1} and the initial condition: $Ay(a) = \alpha$. Denote the solution as $y(t; s_{k-1})$. If $d[y(t; s_{k-1}), \varphi(t)] < \varepsilon$ for $t \in (t_{k-1}, b)$, go to Step 4, otherwise go to Step 3.

3. If there exists a $\tilde{t} \in (t_{k-1}, b)$ with $d[y(\tilde{t}; s_{k-1}), \varphi(\tilde{t})] = \varepsilon$, then:

   (a) denote $t_k = \tilde{t}$;

   (b) use a modified Newton's method with cost function $g(s) = d[y(t_k; s), \varphi(t_k)]$ (and tolerance parameter $\varepsilon_1 < \varepsilon$) and find an update to the parameter vector s. Increment k and go to Step 2.

4. If $d[By(b; s_{k-1}), \beta] \geq \delta$ (where $\delta < \varepsilon$), then use a modified Newton's method with cost function $g(s) = d[By(b; s), \beta]$ and tolerance parameter δ. Stop.

There are three parameters $\varepsilon, \varepsilon_1$ and δ that must be chosen in addition to the choice of s_0. The parameter ε is chosen such that a SSM converges on the first interval $[t_0, t_1]$. It is possible to choose ε_1 and δ to be equal to each other as long as they are less than ε. In the last step of the MSSM, a SSM is being performed with a starting initial guess s_k that keeps $By(b; s_k)$ close to β. This prevents numerical divergence. Figure 1 illustrates the Modified Simple Shooting Method. In this case, it took three overall "shots" to integrate from $t = a$ to $t = b$. In this illustration, the matrix B is in the form $[I_{m_B \times m_B}\ 0_{m_B \times m_A}]$.

To analyze the MSSM, we presume that Assumptions 1 and 2 stated earlier hold. Let D be a neighbourhood of $y^*(a)$ such that the two conditions are satisfied, and let $y_D(t) = \{y(t; a, z) | z \in D\}$ denote the neighborhood around $y^*(t)$ formed by the solutions that start in D. In order to prove convergence of this method, we have to show that the sequence of stopping times $\{t_k\}$ that is produced by the algorithm converges to b for some finite k. At each of these times, we employ the modified Newton's method as described in Step 3b. Therefore we denote the sequence of initial states by $y_{k,l}$ where k corresponds to the stopping time t_k and l corresponds to the subsequence generated by the modified Newton's method. In the following theorem, $\mathbf{N}$ represents the set of natural numbers.

Theorem 1 *Consider the Two-Point Boundary-Value Problem as described in (3-4), along with the Assumptions 1 and 2. Denote the solution to the problem by* $y^*(t)$*. Suppose that the reference function is chosen such that* $\varphi(t) \in y_D(t)$ *for each* $t \in [a, b]$*. Assume that the initial choice* s_0 *is in* D*. Then the Modified Simple Shooting Method results in a sequence* $\{t_k\};\ k \in \mathbf{N}$ *such that* $t_N = b$ *for some finite* N*. Furthermore, the sequence of initial states* $\{y_{k,l}(a)\};\ k,\, l \in \mathbf{N}$*, converges to* $y^*(a)$ *in the limit as* $\delta \to 0$*.*

Proof. By our choice of the reference function and initial state, the solutions to the intermediate step $y(t; a, s_{k-1})$ always lie in the set $y_D(t)$. Suppose that the sequence t_k converges to $t^* < b$. Then there exists an $N \in \mathbf{N}$ such that $|t_N - t^*| < \eta$, where η is a positive number that will be specified later. Furthermore, as a result of the modified Newton's method in Step 3b, there exists a parameter vector s such that $d[y(t_N; s), \varphi(t_N)] < \frac{\varepsilon}{6}$. By our assumption on the intervals of existence for solutions that start in D, we can extend the solution $y(t; s)$ to $[a, t^*]$. Now the function $y(t; s)$ is a differentiable function of t and so let K_2 denote its Lipschitz constant on the interval $[a, t^*]$. Also let K_1 denote the Lipschitz constant of the reference function on $[a, b]$. Now suppose that η is chosen so that $\eta = \varepsilon/(6 \max\{K_1, K_2\})$. Then we have:

$$\begin{aligned} d[y(t^*, s), \varphi(t^*)] &\leq d(y(t^*, s), y(t_N, s)) + d(\varphi(t^*), \varphi(t_N)) + d(\varphi(t_N), y(t_N, s)) \\ &< \frac{\varepsilon}{6} + \frac{\varepsilon}{6} + \frac{\varepsilon}{6} \\ &< \frac{\varepsilon}{2}. \end{aligned} \tag{6}$$

Now by the existence of solutions over the interval $[a, b]$ for all initial states in D, we can extend the solution $y(t; s)$ beyond the interval $[a, t^*]$ to an interval $[a, t^* + \mu]$ for some $\mu > 0$. Furthermore, we can choose μ so that $d[y(t^* + \mu; s), \phi(t^* + \mu)] < \varepsilon$, because of Inequality (6) and the continuity of $y(t;s)$ and $\phi(t)$ as functions of t. Thus $t^* < b$ cannot be true and we must have $t^* = b$. The finiteness of N follows because μ can be chosen to be at least η which depends only on K_1, K_2 and the parameter ε. The last claim follows from the convergence properties of the modified Newton's method.[9] □

Example

In this section, we study an example in order to compare and contrast the MSM, MSSM, Finite Difference and Collocation methods . The computations were performed in a MATLAB environment on a standard desktop computer. The comparison was done for the Problem (7) with two different final times $t_f = 1$ and $t_f = 35$.

Consider the following system:

$$\begin{bmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \\ \dot{y}_3(t) \\ \dot{y}_4(t) \end{bmatrix} = \begin{bmatrix} y_3(t) \\ y_4(t) \\ y_2(t) \\ y_1(t) \end{bmatrix} \quad \text{with} \quad \begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} y_1(t_f) \\ y_2(t_f) \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} \tag{7}$$

where $0 \le t \le t_f$. This system was solved with the "bad" initial guess $s_0 = [-100 \;\; 2]^T$ with the time step 0.01, $\varepsilon = 2$, and $\varepsilon_1 = \delta = 10^{-3}$. The reference path was chosen to be $\varphi(t) = t[1 \;\; 1]^T + [1 \;\; 1]^T$. The parameters, initial guess for the unknown initial states and reference path were kept the same for both the MSM and MSSM.

The computations were carried out for $t_f = 1$ and $t_f = 35$, and the results are reported in Table 1. In the first case ($t_f = 1$), one can see the advantage the MSSM enjoys over the other methods with regard to computation time. The second case ($t_f = 35$) illustrates the accuracy of the MSSM clearly. Only the first few significant digits for the states are displayed in Table 1 though the computations were performed in double precision. The integration method employed for the MSSM, MSM and CM was the classical fourth-order Runge-Kutta method. Figure 2 shows the result of the MSSM including the reference path and the intermediate trajectories. It should be noted that the SSM failed for this problem (with $t_f = 35$), while both the MSM and MSSM were successful.

The Collocation method tested was the "bvp4c" routine in MATLAB. For this method, the mesh selection is based on the residual of the C^1 continuous solution that is fourth order accurate uniformly in the time interval.[10] For $t_f = 1$, we chose 101 uniformly spaced nodes corresponding to a time step of 0.01 seconds as the initial mesh. For $t_f = 35$, a uniform mesh corresponding to a time-step of 0.004 seconds was chosen at the initial step.

Conclusions

A new method for solving two-point boundary-value problems has been presented. An example was provided that clearly illustrates that the MSSM results in an accurate solution that takes significantly less computation time than the MSM, Finite Difference and Collocation methods.

Among its desirable features are that it requires the inversion of much smaller matrices than those required to be inverted in the MSM, Finite Difference and Collocation Methods. Another fact that makes the MSSM more appealing is that the solution results in a trajectory that satisfies the system differential equations. This property is very important in optimal control problems where the systems are unstable in forward time. The MSM, Finite Difference and Collocation methods do not share this property with the MSSM. Due to the instability of many systems in the forward direction, these other methods can lead to erroneous solutions, in the sense that the solution trajectory on re-integration does not satisfy the boundary conditions at the final time.

Acknowledgement

R. Holsapple was supported by an Air Force Research Assistantship during Summer 2002. R. Venkataraman was supported by an NRC/USAF Summer Faculty Fellowship during Summer 2002. The authors gratefully acknowledge the discussions they had with F. Fahroo during the development of this paper.

References

[1] Roberts, S. M. and Shipman, J. S., *Two-Point Boundary Value Problems: Shooting Methods*, Elsevier, New York, NY., 1972, pp. 1–231.

[2] Reinhardt, H.-J., *Analysis of Approximation Methods for Differential and Integral Equations*, Springer-Verlag, New York, NY., 1985, pp. 72–73.

[3] Holsapple, R., Venkataraman, R., and Doman, D., "A Modified Simple Shooting Method for Solving Two Point Boundary Value Problems," *Proceedings of the IEEE Aerospace Conference, Big Sky, MT*, Vol. 6, IEEE, New York, NY 10016-5997, March 2003, pp. 2783–2790.

[4] Keller, H. B., *Numerical Methods For Two-Point Boundary-Value Problems*, Blaisdell, Waltham, MA., 1968, pp. 7–18; 39–71.

[5] Keller, H. B., *Numerical Solution of Two Point Boundary Value Problems*, CBMS-NSF Regional Conference Series in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, PA., 1976, pp. 1–19.

[6] Morrison, D. D., Riley, J. D., and Zancanaro, J. F., "Multiple Shooting Method for Two-Point Boundary Value Problems," *Communications of the ACM*, 1962, pp. 613–614.

[7] Holt, J. F., "Numerical Solution of Nonlinear Two-Point Boundary Problems by Finite Difference Methods," *Communications of the ACM*, 1964, pp. 366–373.

[8] Vinter, R., *Optimal Control*, Birkhäuser, Boston, MA., 2000, pp. 18–20; 91–108.

[9] Stoer, J. and Bulirsch, R., *Introduction to Numerical Analysis*, Springer-Verlag, New York, NY., 2nd ed., 1993, pp. 272–286; 502–535.

[10] Shampine, L. F., Kierzenka, J., and Reichelt, M. W., *Solving Boundary Value Problems for Ordinary Differential Equations in MATLAB with bvp4c*, Natick, MA., 10/26/2000, ftp://ftp.mathworks.com/pub/doc/papers/bvp/tutorial.pdf.

List of Figures

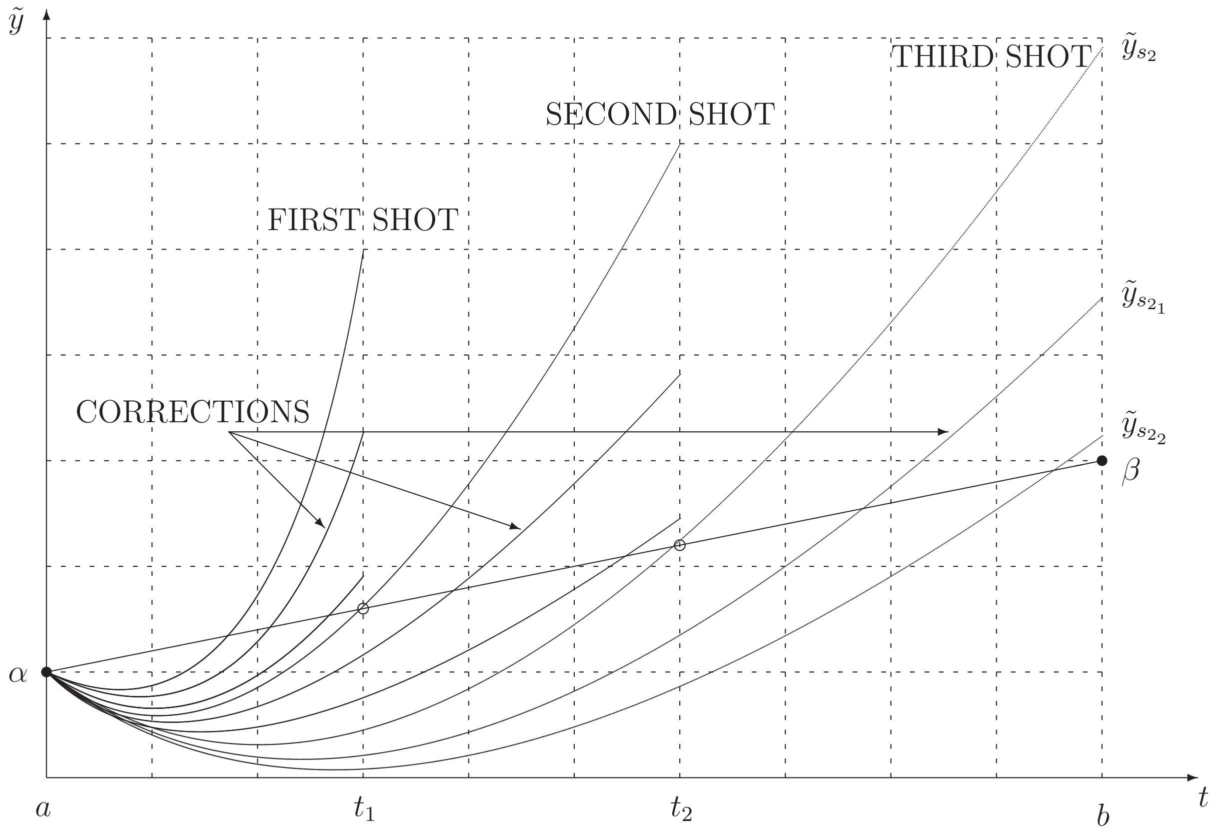

Figure 1: Illustration of Modified Simple Shooting Method

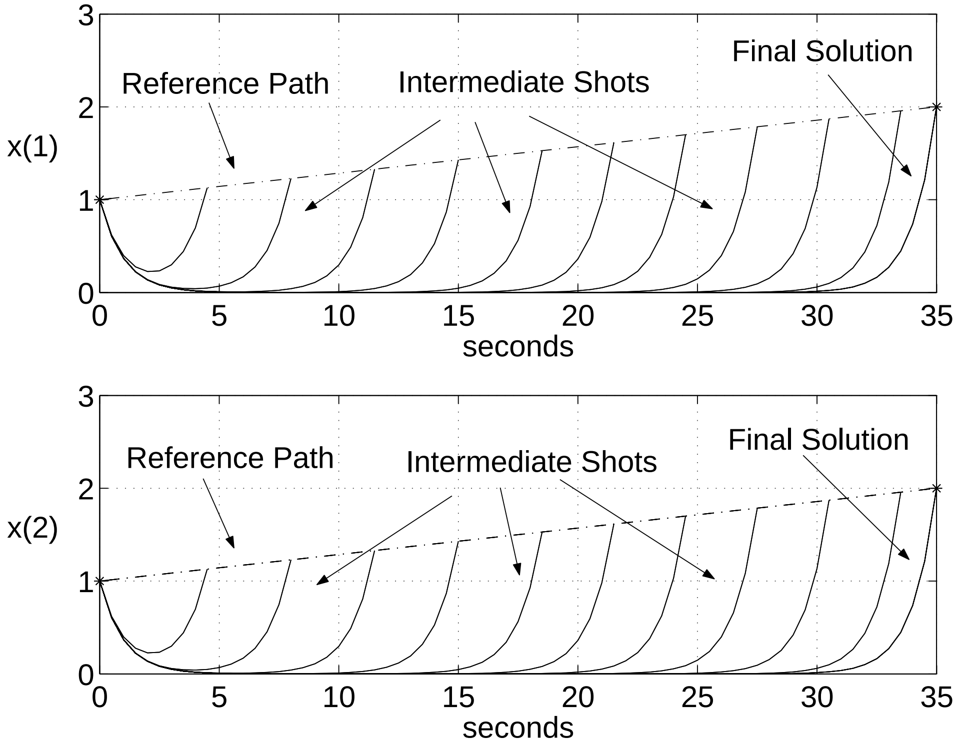

Figure 2: The MSSM applied to the linear system example.

List of Tables

Final Time	Solution Method	Unknown States $[y_3(0) \;\; y_4(0)]$	Time step (s)	Error after re-integration	Computation time (s)
$t_f = 1$	MSSM	$[0.3888 \;\; 0.3888]$	0.1	6.28×10^{-16}	0.02
	MSM	$[0.3888 \;\; 0.3888]$	0.1	9.93×10^{-16}	0.15
	Collocation	$[0.3888 \;\; 0.3888]$	10^{-2}	9.39×10^{-3}	0.17
	FDM	$[0.3888 \;\; 0.3888]$	10^{-2}	9.47×10^{-3}	0.11
$t_f = 35$	MSSM	$[-1.000 \;\; -1.000]$	0.25	2.4×10^{-3}	1.61
	MSM	$[-1.000 \;\; -1.000]$	0.25	1.2×10^{-2}	3.20
	Collocation	$[-1.000 \;\; -1.000]$	0.004	24.33	34.41
	FDM	$[-0.9992 \;\; -0.9992]$	0.04	0.161	490.2

Table 1: Comparison of different solution schemes for the simple example.