International Journal of Engineering, Mathematical and Physical Sciences
ISSN: 2517-9934
Vol:7, No:3, 2013

An Efficient Method for Solving Multipoint Equation Boundary Value Problems

Ampon Dhamacharoen and Kanittha Chompuvised

Abstract**—**In this work, we solve multipoint boundary value problems where the boundary value conditions are equations using the Newton-Broyden Shooting method (NBSM).The proposed method is tested upon several problems from the literature and the results are compared with the available exact solution. The experiments are given to illustrate the efficiency and implementation of the method.

Keywords**—**Boundary value problem; Multipoint equation boundary value problems, Shooting Method, Newton-Broyden method.

I. Introduction

BOUNDARY value problems of ordinary differential equations appear in a large domain of sciences and many practically important problems lead to Multipoint Boundary Value Problems (MBVP's).They are used to describe a large number of physical, biological and chemical phenomena. The works of Prescott [10] and Timoshenko [16] on elasticity, the monographs by Mansfield [5] and the work of Dulácska [4] on the effects of soil settlement are rich sources of such applications.

Let us consider the boundary value problem:
Differential equation:

$$\mathbf{f}(\mathbf{Y}'(t), t) = \mathbf{0} \tag{1a}$$

Boundary value condition:

$$\mathbf{g}(\mathbf{Y}'(t_0), \mathbf{Y}'(t_1), \ldots, \mathbf{Y}'(t_k)) = \mathbf{0} \tag{1b}$$

where $t \in R$, $\mathbf{Y}'(t) = (\mathbf{y}^{(n)}(t), \mathbf{y}^{(n-1)}(t), \ldots, \mathbf{y}'(t), \mathbf{y}(t)) \in R^{m\times(n+1)}$, $\mathbf{f} \in R^m$, $\mathbf{y}(t) \in R^m$, and $\mathbf{g} \in R^{m\times n}$. If $k = 0$ the problems become the initial value problem. The problems are called a two-point boundary value problem if $k = 1$, and a multipoint boundary value problem if $k > 1$. Under certain conditions on the differential equations and the boundary conditions, systems (1a), (1b) have solutions (or a unique solution). The proof of existence, and uniqueness, for some boundary value problems is possible [6,20,21], but for the majority of problems, providing proof is difficult, and so for this reason they remain unproved. Also, solving the problems by analytical methods are rare since most of the problems encountered were difficult, with complicate differential equations or complicate boundary conditions. On the contrary, various numerical methods were developed to approximately solve the problems. Several techniques, such as shooting method [15,19] Adomian decomposition method (ADM) [2,12], differential transform method (DTM) [18], variational iteration method (VIM) [9], successive iteration [13], splines [17], homotopy perturbation method (HPM) [8,11], homotopy analysis method (HAM) [14] have been used to handle these types of problems.

In this paper, we concentrate on method of solving the multipoint boundary value problem where the boundary conditions are equations, which we shall call the multipoint equation boundary value problem. We adapted a method which was success in finding the approximate solution of two-point BVPs (see [22]). The method makes use of the shooting method, but has also adapted the technique of updating the initial values. Several examples are given to illustrate the efficiency and implementation of the method.

II. The Newton and Broyden Shooting Method

The process of shooting method is as follows: Firstly, guess the initial value $\mathbf{z} = \mathbf{Y}'(t_0)$ and solve the differential equation (1a) with this initial value to get $\mathbf{Y}'(t_i)$, $i = 1, 2, \ldots, m$; check whether the condition (1b) is satisfied; if not, adjust the initial $\mathbf{z}$ and repeat until the condition (1b) is met. More precisely, for the k+1-point boundary value problems:

$$\text{Differential equation:} \quad \mathbf{f}(\mathbf{Y}'(t), t) = \mathbf{0} \tag{2a}$$

Boundary value condition:

$$\mathbf{g}(\mathbf{Y}'(t_0), \mathbf{Y}'(t_1), \ldots, \mathbf{Y}'(t_k)) = \mathbf{0} \tag{2b}$$

Let $\mathbf{z} = \mathbf{Y}'(t_0)$ and $\mathbf{F}(\mathbf{z}) = \mathbf{g}(\mathbf{z}, \mathbf{Y}'(t_1), \ldots, \mathbf{Y}'(t_k)))$, where $\mathbf{Y}'(t)$ and $\mathbf{Y}'(t_j)$, $j = 1, 2, \ldots, k$, are considered to be functions of $\mathbf{z}$ implicitly through the equation (2a). The equation (2b) can be written in the form

$$\mathbf{F}(\mathbf{z}) = \mathbf{0} \tag{3}$$

To solve (3) numerically, the Newton method has the scheme:

$$\mathbf{z}_{i+1} = \mathbf{z}_i - [\mathbf{F}'(\mathbf{z}_i)]^{-1}\mathbf{F}(\mathbf{z}_i), \quad i = 1, 2, 3, \ldots \tag{4}$$

A. Dhamacharoen is with the Department of mathematics, Faculty of Science, Burapha University, Bangsaen, Chonburi 20131, Thailand; (e-mail: ampon@buu.ac.th).

Kanittha Chompuvised is with the Department of mathematics, Faculty of Science, and Technology, Nakhon Ratchasima Rajabhat University, Nakhon Ratchasima, Thailand; (e-mail: jokejin@hotmail.com).

where $\mathbf{F}'(\mathbf{z})$ is the Jacobian matrix of the function $\mathbf{F}$. The method is called Newton Shooting Method if (3) is solved using the Newton scheme (4). The most difficult task in this problem is to find the value $\mathbf{F}'(\mathbf{z}_i)$ in each iteration, since the function $\mathbf{F}$ has implicit form, and its value is derived from solving the differential equations (1a). Let $\mathbf{S}(t, \mathbf{z}) = \mathbf{Y}'(t)$, we have $\mathbf{F}(\mathbf{z}) = \mathbf{g}(\mathbf{z}, \mathbf{S}(t_1, \mathbf{z}), \ldots, \mathbf{S}(t_k, \mathbf{z}))$. Therefore:

$$\mathbf{F}'(\mathbf{z}) = \frac{\partial \mathbf{g}}{\partial \mathbf{z}} + \frac{\partial \mathbf{g}}{\partial \mathbf{S}}\frac{\partial \mathbf{S}_1}{\partial \mathbf{z}} + \ldots + \frac{\partial \mathbf{g}}{\partial \mathbf{S}}\frac{\partial \mathbf{S}_k}{\partial \mathbf{z}} \qquad (5)$$

where $\frac{\partial \mathbf{g}}{\partial \mathbf{z}}$, $\frac{\partial \mathbf{g}}{\partial \mathbf{S}}$ and $\frac{\partial \mathbf{S}_j}{\partial \mathbf{z}}$, j = 1, 2, . . . , k, are Jacobian matrices with appropriate sizes. The values $\frac{\partial \mathbf{g}}{\partial \mathbf{z}}$ and $\frac{\partial \mathbf{g}}{\partial \mathbf{S}}$ will be obtained directly from (2b). The value $\frac{\partial \mathbf{S}_i}{\partial \mathbf{z}}$ can be computed by $\frac{\partial \mathbf{S}_j}{\partial \mathbf{z}} = \left.\frac{\partial \mathbf{S}}{\partial \mathbf{z}}\right|_{t=t_j}$, j = 1, 2, . . . , n.

From (1a) differentiating both sides with respect to $\mathbf{z}$ gives:

$$[\frac{\partial \mathbf{f}}{\partial \mathbf{S}}(\mathbf{Y}'(t), t)]\ \frac{\partial \mathbf{S}}{\partial \mathbf{z}} = \mathbf{0}$$

Let $\mathbf{p}'(t) = \frac{\partial \mathbf{S}}{\partial \mathbf{z}} = \frac{\partial}{\partial \mathbf{z}}\mathbf{Y}'(t)$, then $\frac{\partial \mathbf{S}}{\partial \mathbf{z}}$ can be obtained from solving the system of linear differential equations

$$[\frac{\partial \mathbf{f}}{\partial \mathbf{S}}(\mathbf{Y}'(t), t)]\ \mathbf{p}'(t) = \mathbf{0} \qquad (6)$$

for $\mathbf{p}(t)$ and then set $\frac{\partial \mathbf{S}_j}{\partial \mathbf{z}} = \mathbf{p}(t_j)$, j = 1, 2, . . . , n.

Note that computing the value of $\mathbf{F}'(\mathbf{z})$ in each iteration requires so much work, therefore, any method which could be utilized to avoid this work, and which also works satisfactorily well, would be preferred.

To avoid the difficult calculation of $\mathbf{F}'(\mathbf{z})$, the Broyden scheme

$$\mathbf{z}_{i+1} = \mathbf{z}_i - D_i^{-1}\mathbf{F}(\mathbf{z}_i), \quad i = 1, 2, 3, \ldots \qquad (7)$$

can be used to replace (3), that is, replacing $\mathbf{F}'(\mathbf{z})$ with a non-singular matrix D_i. With the appropriate initial $\mathbf{z}_0$ and D_0, and with the Broyden updating

$$D_{i+1} = D_i + \mathbf{a}\mathbf{b}^T$$

where $\mathbf{a} = \frac{1}{\mathbf{b}^T\mathbf{b}}\mathbf{F}(\mathbf{z}_{i+1})$, and $\mathbf{b} = \mathbf{z}_{i+1} - \mathbf{z}_i$, the scheme (7) will produce the sequence that hopefully converges to the solutions.

In solving the system of ordinary equations, the Broyden method performs well, provided that a good initial point $\mathbf{z}$ and D were chosen. In case of convergence, the Broyden method has q-super linearly order of convergence [7]. So, compared to the Newton method, which has quadratic convergence, but involves a large amount of computation, the Broyden method may sometimes be preferable. Unfortunately, it is sometimes difficult to come up with a good initial guess, and thus the solutions cannot be found. For the boundary value problems with simple two-point boundary conditions, experiments have showed that the Broyden works well most of the time, even for non-linear differential equations [3]. If the boundary conditions are equations, the Broyden still works well if the differential equations are linear. However, many times, it can not reach a satisfactory solution if the differential equations are non-linear.

III. The Newton-Broyden Shooting Method

Now, let's consider the function of the form

$$\mathbf{F}(\mathbf{u}(\mathbf{x}), \mathbf{x}) = \mathbf{0} \qquad (8)$$

The Newton scheme of this problem is

$$\mathbf{x}_{i+1} = \mathbf{x}_i - [\mathbf{F}'(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i)]^{-1}\mathbf{F}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i),\ i = 0, 1, 2, \ldots \qquad (9)$$

where $\mathbf{F}'(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i) = \mathbf{F}_\mathbf{u}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i)\mathbf{u}'(\mathbf{x}_i) + \mathbf{F}_\mathbf{x}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i)$

Replace $\mathbf{u}'(\mathbf{x}_i)$ by D_i, gives

$$\mathbf{x}_{i+1} = \mathbf{x}_i - [\mathbf{F}_\mathbf{u}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i)D_i + \mathbf{F}_\mathbf{x}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i)]^{-1}\mathbf{F}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i),\quad i = 0, 1, 2, \ldots \qquad (10)$$

with updating:

$$D_{i+1} = D_i + \frac{1}{\mathbf{b}^T\mathbf{b}}(\mathbf{u}(\mathbf{x}_{i+1}) - \mathbf{u}(\mathbf{x}_i) - D_i\mathbf{b})\mathbf{b}^T \qquad (11)$$

where $\mathbf{b} = \mathbf{x}_{i+1} - \mathbf{x}_i$. This method retains the good part of Newton method, and replaces the difficult part of Newton's by Broyden's. With a good initial guess for $\mathbf{z}_0$ and D_0, the Newton-Broyden scheme (10) will produce the sequence that converges to a solution of (8), with q-super linearly order of convergence [1].

Consider the function of the form:

$$\mathbf{F}(\mathbf{u}_1(\mathbf{x}), \mathbf{u}_2(\mathbf{x}), \ldots, \mathbf{u}_k(\mathbf{x}), \mathbf{x}) = \mathbf{0} \qquad (12)$$

where $\mathbf{x}, \mathbf{u}_j(\mathbf{x}) \in R^m$, j = 1, 2, . . . , k. The derivative is:

$$\mathbf{F}'(\mathbf{u}_1(\mathbf{x}), \mathbf{u}_2(\mathbf{x}), \ldots, \mathbf{u}_k(\mathbf{x}), \mathbf{x}) = \sum_{j=1}^{k}\mathbf{F}_{\mathbf{u}_j}(\mathbf{u}_1(\mathbf{x}), \ldots, \mathbf{u}_k(\mathbf{x}), \mathbf{x})\,\mathbf{u}_j'(\mathbf{x}) + \mathbf{F}_\mathbf{x}(\mathbf{u}_1(\mathbf{x}), \mathbf{u}_2(\mathbf{x}), \ldots, \mathbf{u}_k(\mathbf{x}), \mathbf{x})$$

The Newton scheme of this problem is

$$\mathbf{x}_{i+1} = \mathbf{x}_i - [\sum_{j=1}^{k}\mathbf{F}_{\mathbf{u}_j}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i)\,\mathbf{u}_j'(\mathbf{x}_i) + \mathbf{F}_\mathbf{x}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i)]^{-1}\mathbf{F}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i),\quad i = 0, 1, 2, \ldots \qquad (13)$$

where $\mathbf{u}(\mathbf{x}) = (\mathbf{u}_1(\mathbf{x}), \mathbf{u}_2(\mathbf{x}), \ldots, \mathbf{u}_k(\mathbf{x}))$.

Replace $\mathbf{u}_j'(\mathbf{x}_i)$ by D_{ji}, gives

$$\mathbf{x}_{i+1} = \mathbf{x}_i - [\sum_{j=1}^{k}\mathbf{F}_{\mathbf{u}_j}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i)\,D_{ji} + \mathbf{F}_\mathbf{x}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i)]^{-1}\mathbf{F}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i),\quad i = 0, 1, 2, \ldots \qquad (14)$$

with updating:

$$D_{ji+1} = D_{ji} + \frac{1}{\mathbf{b}^T\mathbf{b}}(\mathbf{u}_j(\mathbf{x}_{i+1}) - \mathbf{u}_j(\mathbf{x}_i) - D_{ji}\mathbf{b})\mathbf{b}^T \quad (15)$$

where $\mathbf{b} = \mathbf{x}_{i+1} - \mathbf{x}_i$.

Consider the equation (5) in the form of (13), the derivatives take the forms:

$$(\mathbf{F}_{\mathbf{u}_1}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i), \mathbf{F}_{\mathbf{u}_2}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i), \ldots, \mathbf{F}_{\mathbf{u}_k}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i)) = \frac{\partial \mathbf{g}}{\partial S},$$

$$\mathbf{u}_j'(\mathbf{x}_i) = \frac{\partial \mathbf{S}_j}{\partial \mathbf{z}}, \ j = 1, 2, \ldots, k, \text{ and } \mathbf{F}_{\mathbf{x}}(\mathbf{u}(\mathbf{x}_i), \mathbf{x}_i) = \frac{\partial \mathbf{g}}{\partial \mathbf{z}}.$$

In using the scheme (14), the matrix $\frac{\partial \mathbf{S}_j}{\partial \mathbf{z}}$ at the i^{th} iteration was replaced by D_{ji}, and the Broyden updating technique was used to obtain D_{ji+1}. The full process (which should be called the Newton-Broyden Shooting method (NBSM)) will be as follows:

1. Let $\mathbf{Y}'(t_0) = \mathbf{z}_0$ and a matrix D_{j0} (same size of $\frac{\partial \mathbf{S}}{\partial \mathbf{z}}$) be given. Solve the differential equation (2a) with the initial values $\mathbf{z}_0$ to obtain $\mathbf{Y}'(t_i)$, i = 1, 2, . . . , k. Compute $\mathbf{F} = \mathbf{F}(\mathbf{z}) = \mathbf{g}(\mathbf{Y}'(t_1), \mathbf{Y}'(t_2), \ldots, \mathbf{Y}'(t_k), \mathbf{z})$. Let $\mathbf{u}_{ji} = \mathbf{Y}'(t_j)$, j = 1, 2, . . . , k.
2. For i = 0, 1, 2, . . . do the following:

   Compute $\frac{\partial \mathbf{g}}{\partial \mathbf{z}}$, $\frac{\partial \mathbf{g}_j}{\partial S} = \left.\frac{\partial \mathbf{g}}{\partial S}\right|_{t_j}$, j = 1, 2, . . . , k, and

   $$H = \frac{\partial \mathbf{g}}{\partial \mathbf{z}} + \sum_{j=1}^{k} \frac{\partial \mathbf{g}_j}{\partial S} D_{ji}$$

   Solve the system of linear equations $H\mathbf{b} = -\mathbf{F}$ for $\mathbf{b}$.

   Let $\mathbf{z}_{i+1} = \mathbf{z}_i + \mathbf{b}$.

   Solve the differential equation (2a) with the new initial values $\mathbf{z}_{i+1}$, to obtain $\mathbf{Y}'(t_j)$, j = 1, 2, . . . , k. Compute $\mathbf{u}_{ji+1} = \mathbf{Y}'(t_j)$ and

   $$\mathbf{F} = \mathbf{F}(\mathbf{z}_{i+1}) = \mathbf{g}(\mathbf{Y}'(t_1), \mathbf{Y}'(t_2), \ldots, \mathbf{Y}'(t_k), \mathbf{z}_{i+1}).$$

   Compute $\mathbf{b}^T\mathbf{b}$ and $\mathbf{a}_j = \frac{1}{\mathbf{b}^T\mathbf{b}}(\mathbf{u}_{ji+1} - \mathbf{u}_{ji} - D_{ji}\mathbf{b})$, and update D_i by $D_{ji+1} = D_{ji} + \mathbf{a}_j\mathbf{b}^T$
3. Stop the iteration if i ≤ n, a prescribed positive integer, or $\|\mathbf{F}\| < \varepsilon$, or $\|\mathbf{b}\| < \varepsilon$, a given small number.

In the algorithm, the matrix $\frac{\partial \mathbf{S}_j}{\partial \mathbf{z}}$ was replaced by D_{ji}, so that skipping the step which involves solving the system of linear differential equations can save a large amount of work, in terms of the number of computation and implementation.

The success of the method for solving boundary value problems generally depends on two phases, which are namely the method of solving the initial value problems, and the method of solving systems of equations. In this proposed method, the first part use the standard technique of solving the initial value problems, such as the Taylor's method or the Runge-Kutta's method; the second part use the new technique for solving the system of non-linear equations derived by A. Dhamacharoen [1].

IV. Experiments

Various systems of boundary value problems, with ordinary differential equations and multi-point equation boundary values conditions, have been solved using the scheme in section III. The problems are as follow:

Problem 1: We consider the following third-order linear ordinary differential equation:

$$x''' = tx + (t^3 - 2t^2 - 5t - 3)e^t$$

with the following boundary conditions:

$$x''(0.5) - x'(0.5) - 2x'(1) + x(0)^2 + x''(0)x''(1) = 2e - 2e^{1/2}$$
$$x'(0)x(1) + x(0.5)x''(0) + 4x'(1) - x''(1) = 0$$
$$x'(0)x(0.5) - x'(0.5) + x(1)^2 - x(0)x''(0.5) = 0$$

The exact solution of this problem is $x(t) = (t - t^2)e^t$

Define the vectors $\mathbf{x} = (x(0), x'(0), x''(0))$; $\mathbf{u_1(x)} = (x(0.5), x'(0.5), x''(0.5))$ and $\mathbf{u_2(x)} = (x(1), x'(1), x''(1))$, the function $\mathbf{F(u_1(x), u_2(x), x)}$ is the vector in the boundary condition. The value of $\mathbf{u}$ can be obtained by solving the differential equation with the initial value $\mathbf{x}$, so that the function $\mathbf{F}$ is well defined. The Jacobian $\mathbf{u}'(\mathbf{x})$ could be computed by solving the system of linear differential equations. However, use the approximating matrix D instead.

Problem 2: We consider the following third-order nonlinear ordinary differential equation:

$$x''' = e^{-t}x^2$$

with the following boundary conditions:

$$x(0) + 2x'(0.5)^2 - x(1) - x''(1) - x(0.5) = 1 - e^{1/2}$$
$$x'(0)^2 - x''(0)x'(1) + x(0.5)x''(0.5) + x''(1) = 1 + e$$
$$x'(0)x(1) - x'(1) + x'(0.5) + x(0) - x''(0) - x''(0.5) = 0$$

The exact solution of this problem is, $x(t) = e^t$

Problem 3: We consider the following fourth-order nonlinear ordinary differential equation:

$$x^{(4)} + xx' - 4t^7 - 24 = 0$$

with the following boundary conditions:

$$2x'''(0) + x^2(0)x'(0.25) + x''(0.5) - x'(1) + x^2(1) - 4x''(0.25) = -3$$
$$2x(0) + x''(0)x'''^2(1) - x''(1) + x''(0.5)x'''(0.25) - 16x(0.25) + x(0.5) = 6$$
$$x''(0) + 2x'(0) + x'''(1) - 2x'''(0.5) + 2x'(0.5) - x'''(0.25) = -5$$
$$x'''(0) + x'''(1)x''(0.25) - x'(0.25) + x(0.5) - 2x'''(0.25) = 6$$

The exact solution of this problem is, $x(t) = t^4$.

Define the vectors $\mathbf{x} = (x(0), x'(0), x''(0), x'''(0))$;

$\mathbf{u_1(x)} = (x(0.25), x'(0.25), x''(0.25), x'''(0.25))$;

u₂(x) = $\left(x(0.5), x'(0.5), x''(0.5), x'''(0.5)\right)$ and **u₃(x)** = $\left(x(1), x'(1), x''(1), x'''(1)\right)$, the function **F(u₁(x), u₂(x), u₃(x), x)** is the vector in the boundary condition. The value of **u** can be obtained by solving the differential equation with the initial value **x**, so that the function **F** is well defined.

Problem 4: We consider the following nonlinear system of second-order ordinary differential equation:

$$x'' - ty' + x = t^3 - 2t^2 + 6t$$
$$y'' + tx' + xy = t^5 - t^4 + 2t^3 + t^2 - t + 2$$

with the following boundary conditions:

$$x^2(0)y(0.5) + x'(0)x'(1) + y(1) = -2$$
$$x^2(1) + y'(0.5) + y'^2(1) - x(0.5)y(0) = 1$$
$$4x'(0.5) - y'(0) + 2x(0) + y'(1)x'(1) = 2$$
$$2y(0.5) + x'(0.5) - 2x(0.5) + y'(0) + x'^2(0) + y(1) = 0$$

The exact solution of this problem is, $x(t) = t^3 - t$ and $y(t) = t^2 - t$.

Define the vectors **x** $= (x(0), x'(0), y(0), y'(0))$; **u₁(x)** = $\left(x(0.5), x'(0.5), y(0.5), y'(0.5)\right)$ and **u₂(x)** = $\left(x(1), x'(1), y(1), y'(1)\right)$, the function **F(u₁(x), u₂(x), x)** is the vector in the boundary condition. The value of **u** can be obtained by solving the differential equation with the initial value **x**, so that the function **F** is well defined.

The above problems were solved using three methods for comparison; the Broyden Shooting Method (BSM), the Newton Shooting Method (NSM) and the Newton-Broyden Shooting Method (NBSM).

The computations for each problem were made by using the same initial guess for all methods. The suitable initial value, and the number of iterations (N) are shown in Table I. The number of operations were counted, and the saving cost of the proposed methods, which is the percentage of saving number of operations relative to the Newton method, is shown in Table II.

The results show that the NBSM can compete with NSM. Although the number of iterations of the NBSM to reach the solution is more than that of the NSM, overall the number of operations is less than that of the NSM.

TABLE I
COMPARISON OF VARIOUS ITERATIVE METHODS

Initial guess z_0	Solution	Suitable Initial values for the solutions		
		BSM	NSM	NBSM
Problem 1 (2, 0, 0.5)	(0, 1, 0)	N = 14 (3.6E-08, 0.99999976, 1E-09)	N = 10 (2E-09, 0.99999978, 1.6E-08)	N = 9 (-1.2E-08, 0.999999864, 9E-09)
Problem 2 (1.5, 1.5, 1.5)	(1, 1, 1)	N = 64 (1.000000015, 1.00000002, 1.000000036)	N = 13 (0.999999994, 1.000000025, 1.000000044)	N = 27 (0.99999999, 1.000000002, 1.000000000)
Problem 3 (1, -1, -1, -1)	(0, 0, 0, 0)	Diverge	N = 11 (-6.870E-09, -2.718E-09, -2.24E-10, 1.21E-10)	N = 16 (-1.077E-08, 2.794E-09, -2.44E-10, 7.321E-09)
Problem 4 (1, 1, 1, 0)	(0, -1,0,-1)	Diverge	N = 18 (-9.902E-09, -1.00000000044, -6.994E-09, -1.00000000230)	N = 18 (-7.680E-09, -1.0000000007, 7.659E-09, -1.00000000016)

TABLE II
THE NUMBER OF OPERATIONS AND THE SAVING COST OF ITERATIONS

Initial guess z_0	Flops			Saving(%)
	BSM	NSM	NBSM	NBSM
Problem 1 (2,0,0.5)	397105	389554	272095	30.15
Problem 2 (1.5,1.5,1.5)	1187676	556434	391171	29.70
Problem 3 (1,-1,-1,-1)	Diverge	1037594	713454	31.24
Problem 4 (1,1,1,0)	Diverge	1631716	832400	48.99

A. Figures and Tables

Comparisons of the solution from the NBSM with the exact values are given in table III-VI below. We observed from the table that the values obtained by the NBSM are reasonably closed to the exact values, and the NBSM still remain efficient, even for higher order BVP and nonlinear systems of BVP.

TABLE III
RESULT OF PROBLEM 1

t	Exact solution	NBSM solution	Error
0.0	0	1.83787E-09	-1.83787E-09
0.1	0.099465383	0.099465369	1.3303E-08
0.2	0.195424441	0.195424415	2.64714E-08
0.3	0.28347035	0.283470312	3.72598E-08
0.4	0.358037927	0.358037882	4.52059E-08
0.5	0.412180318	0.412180268	4.97858E-08
0.6	0.437308512	0.437308462	5.04056E-08
0.7	0.422888069	0.422888022	4.63891E-08
0.8	0.356086549	0.356086512	3.69644E-08
0.9	0.22136428	0.221364259	2.12456E-08
1.0	-1.207E-15	1.78887E-09	-1.78887E-09

TABLE IV
RESULT OF PROBLEM 2

t	Exact solution	NBSM solution	Error
0.0	1	1.000000003	-2.9238E-09
0.1	1.105170918	1.105170921	-2.54891E-09
0.2	1.221402758	1.22140276	-2.11432E-09
0.3	1.349858808	1.349858809	-1.61361E-09
0.4	1.491824698	1.491824699	-1.03828E-09
0.5	1.648721271	1.648721271	-3.7744E-10
0.6	1.8221188	1.8221188	3.8246E-10
0.7	2.013752707	2.013752706	1.25794E-09
0.8	2.225540928	2.225540926	2.26892E-09
0.9	2.459603111	2.459603108	3.43911E-09
1.0	2.718281828	2.718281824	4.79652E-09

TABLE V
RESULT OF PROBLEM 3

t	Exact solution	NBSM solution	Error
0.0	0	-.00000000011784	1.17843E-10
0.1	0.00010	0.00009999988531	1.14684E-10
0.2	0.00160	0.00159999988853	1.11468E-10
0.3	0.00810	0.00809999989189	1.08109E-10
0.4	0.02560	0.02559999989547	1.04524E-10
0.5	0.06250	0.06249999989937	1.00625E-10
0.6	0.12960	0.12959999990368	9.6325E-11
0.7	0.24010	0.24009999990847	9.1529E-11
0.8	0.40960	0.40959999991386	8.6135E-11
0.9	0.65610	0.65609999991997	8.0032E-11
1.0	1.00000	0.99999999992691	7.3088E-11

TABLE VI
RESULT OF PROBLEM 4

t	Exact solution x(t)	NBSM solution	Error
0.0	-0.00000000	0.000000	7.680E-09
0.1	-0.09900000	-0.099000	7.715E-09
0.2	-0.19200000	-0.192000	7.672E-09
0.3	-0.27300000	-0.273000	7.554E-09
0.4	-0.33600000	-0.336000	7.360E-09
0.5	-0.37500000	-0.375000	7.093E-09
0.6	-0.38400000	-0.384000	6.756E-09
0.7	-0.35700000	-0.357000	6.354E-09
0.8	-0.28800000	-0.288000	5.890E-09
0.9	-0.17100000	-0.171000	5.373E-09
1.0	-0.00000000	0.000000	4.810E-09

V. CONCLUSION

The multipoint equation boundary value problem, can be solved using the Newton-Broyden Shooting method. The experiments have shown that the method worked well, as expected, which is to say that the good initial guesses were easily found.

ACKNOWLEDGMENT

The author would like to thank the referees for their comments. This research was supported by the Research Fund from the National Research Council of Thailand).

REFERENCES

[1] A. Dhamacharoen, “Two-Stage Iteration Methods for Solving Systems of Nonlinear Equations”, In The 15th Annual Meeting in Mathematics., Bangkok., Annu. 2010, 6-8.

[2] A. M. Wazwaz, The numerical solution of sixth-order boundary value problems by the modified decomposition method, Appl. Math. Comput. 118 (2001), 311-325.

[3] A. Prachanuruk, Broyden method in solving equations boundary value problems, Master's thesis, Department of Mathematics, Graduate School, Burapha University (2007).

[4] E. Dulácska, Soil Settlement Effects on Buildings,Developments Geotechn. Engrg., vol. 69, Elsevier, Amsterdam,1992.

[5] E. H. Mansfield, The Bending and Stretching of Plates Internat. Ser. Monogr. Aeronautics and Astronautics, vol. 6 Pergamon, New York, 1964.

[6] H. B. Keller, Existence theory for two point boundary value problems, Bull. Amer. Math. Soc., 72 (1966), 728-731.

[7] J. E. Dennis, and R. B. Schnabel, Numerical Methods forUnconstrained Optimization and Nonlinear Equations, SIAM,1966.

[8] J. H. He, Homotopy perturbation method for solving boundary value problems, Phys. Lett. A. 350 (2006), 87-88.

[9] J. H. He, Variational approach to the sixth-order boundaryvalue problems, Appl. Math. Comput. 143 (2-3) (2003), 53-538.

[10] J. Prescott, Applied Elasticity, Dover, New York, 1961.

[11] M. A. Noor, S.T. Mohyud-Din, An efficient algorithm for solving fifth-order boundary value problems, Math. Comput. Model. 45 (7-8) (2007), 954-964.

[12] M. Tatari, M. Dehgan, The use of the Adomian decomposition method for solving multipoint boundary value problems, Phys. Scr. 73 (2006), 672-676.

[13] Q. Yao, Successive iteration and positive solution for nonlinear second-order three-point boundary value problem, Comput. Math. Appl. 50 (2005), 433-444.

[14] S. J. Liao, Beyond Perturbation: Introduction to the Homotopy Analysis Method, Chapman & Hall/CRC, Florida, 2004.

[15] S. N. Ha, A nonlinear shooting method for two-point boundary value problems, Comput. Math. Appl., 42 (10-11) (2001), 1411-1420.

[16] S. P. Timoshenko, Theory of Elastic Stability, McGraw–Hill, New York, 1961.

[17] Siraj-Ul-Islam, I.A. Tirmizi, Fazal-i-Haq, M.A. khan, Non-polynomial splines approach to the solution of sixth-order boundary-value problems, Appl. Math. Comput. 195 (2008), 270-284.

[18] Siraj-Ul-Islam, Sirajul-Haq, Javed Ali, Numerical solution of special 12th-order boundary value problems using differential transform method, Comm. Nonl. Sci. Numeric. Simul. 14 (4) (2009), 1132-1138.

[19] U. M. Ascher, R. M. M. Mattheij, R. D. Russell, Numerical solution of boundary value problems for ordinary differential equations, Englewood Cliffs: Prentice Halll (1988).

[20] W. Jiang, M. Cui, Constructive proof for existence of nonlinear two-point boundary value problems., Appl. Math. Comput., 215 (5) (2009), 1937-1948.

[21] Y. Tao, G. Gao, Existence, uniqueness and approximation of classical solutions to nonlinear two-point boundary value problems., Nonlinear Anal., 50 (7) (2002), 981-994.

[22] K. Chompuvised, A. Dhamacharoen, Solving boundary value problems of ordinary differential equations with non-separatedboundary conditions, Applied Mathematics and Computation,Volume 217, Issue 24, 15 August 2011, Pages 10355-10360.