Solution of Nonlinear High Order Multi-Point Boundary Value Problems By Semi-Analytic Technique

Luma. N. M. Tawfiq* ***Mariam. M. Hilal****

Received 20, December, 2012
Accepted 25, June, 2013

Abstract:

In this paper, we present new algorithm for the solution of the nonlinear high order multi-point boundary value problem with suitable multi boundary conditions. The algorithm is based on the semi-analytic technique and the solutions are calculated in the form of a rapid convergent series. It is observed that the method gives more realistic series solution that converges very rapidly in physical problems. Illustrative examples are provided to demonstrate the efficiency and simplicity of the proposed method in solving this type of multi- point boundary value problems.

Key words: Differential Equation, Multi-point Boundary Value Problem, Approximate Solution.

1. Introduction

Some problems which have wide classes of application in science and engineering have usually been solved by perturbation methods. These methods have some limitations, e.g., the approximate solution involves a series of small parameters which poses difficulty since the majority of nonlinear problems have no small parameters at all. Although appropriate choices of small parameters do lead to ideal solution while in most other cases, unsuitable choices lead to serious effects in the solutions [1]. The semi-analytic technique employed here, is a new approach for finding the approximate solution that does not require small parameters, thus over-coming the limitations of the traditional perturbation techniques. The method was first proposed by Grundy (2003) and successfully applied by other researchers like Grundy (2003-2007) who examined the feasibility of using two points Hermite interpolation as a systematic tool in the analysis of initial-boundary value problems for nonlinear diffusion equations. In 2005 Grundy analyzed initial - boundary value problems involving nonlocal nonlinearities using two points Hermite interpolation [1], also, in 2006 He showed how two-point Hermite interpolation can be used to construct polynomial representations of solutions to some initial-boundary value problems for the inviscid Proudman-Johnson equation. In 2008, Maqbool [2] used a Semi-analytical Method to Model Effective SINR Spatial Distribution in WiMAX Networks. Also, in 2008, Debabrata [3] studied Elasto-plastic strain analysis by a semi-analytical method.

The existence of positive solutions for multi-point boundary value problems is one of the key areas of research these days owing to its wide application in engineering like in the modeling of physical problems involving vibrations occurring in a wire of uniform cross section and composed of material with different densities, in the theory of elastic stability and also its applications in fluid flow through porous media.

*Department of Mathematics, College of Education Ibn Al-Haitham,University of Baghdad

Gupta [4] studied the existence of solutions for the generalized multi-point BVP in the non-resonance case.
Zhang et al [5] obtained some new existence results of the fourth-order, four-point BVP, by developing the upper and lower solution method and the monotone iterative technique; it is well known that the upper and lower solution method is a powerful tool for proving existence results for BVPs. In many cases it is possible to find a minimal solution and a maximal solution between the lower solution and the upper solution by the monotone iterative technique [5].
Liu [6] established sufficient conditions for the existence of at least one solution of nth order MPBVP.
Anderson et al [7] concerned with the existence and form of solutions to nonlinear third-order, three-point and multi-point boundary-value problems on general time scales.
Wang et al [8] studied the existence of nontrivial solutions for nonlinear higher order MPBVP on time scales with all derivatives.
Graef et al [9] obtained sufficient conditions for the existence of a solution of the higher order MPBVP based on the existence of lower and upper solutions.
Liu et al [10] established the existence results of multiple monotone and convex positive solutions for some fourth-order MPBVPs.
In this paper we use two-point osculatory interpolation; essentially this is a generalization of interpolation using Taylor polynomials. The idea is to approximate a function y by a polynomial P in which values of y and any number of its derivatives at given points are fitted by the corresponding values and derivatives of P.
We are particularly concerned with fitting function values and derivatives at the two end points of a finite interval, say [0,1] where a useful and succinct way of writing osculatory interpolation P_{2n+1} of degree 2n + 1 was given for example by Phillips [11] as:

$$P_{2n+1}(x) = \sum_{j=0}^{n} \{ y^{(j)}(0)\, q_j(x) + (-1)^j\, y^{(j)}(1)\, q_j(1-x) \} \qquad (1)$$

$$q_j(x) = (x^j / j!)(1-x)^{n+1} \sum_{s=0}^{n-j} \binom{n+s}{s} x^s = Q_j(x) / j! \qquad (2)$$

so that (1) with (2) satisfies:

$$y^{(j)}(0) = P_{2n+1}^{(j)}(0) \quad , \quad y^{(j)}(1) = P_{2n+1}^{(j)}(1) \quad , \quad j = 0, 1, 2,\dots, n.$$

Implying that P_{2n+1} agrees with the appropriately truncated Taylor series for y about
x = 0 and x = 1. We observe that (1) can be written directly in terms of the Taylor coefficients a_i and b_i about x = 0 and x = 1 respectively, as:

$$P_{2n+1}(x) = \sum_{j=0}^{n} \{a_j\, Q_j(x) + (-1)^j\, b_j\, Q_j(1-x) \} \qquad (3)$$

2. Solution of Multi-Point High Order Nonlinear BVP's for ODE

A general form of n- order, m- point BVP's is:

$$y^{(n)} = f(x, y, y', \dots, y^{(n-1)}), \quad 0 \le x \le 1, \quad n \ge 3 \qquad (4a)$$

Subject to the boundary condition:

$$g(y(0), y'(0), \dots , y^{(n-1)}(0)) = 0, \quad y^{(i)}(\eta_i) = \mu_j , j = 0, 1,\dots, n-3, \quad h(y(1), y'(1), \dots , y^{(n-1)}(1)) = 0 \qquad (4b)$$

where $\eta_i \in (0, 1)$, $\forall i = 1,2,\dots,m-2$, g, h : $R^n \rightarrow R$ are continuous functions, and $\mu_j \in R$,
j = 0, 1,…, n-3.

where f: $[0,1]\times R^n \to R$ is a continuous function, $0 < \eta_1 < \eta_2 < \cdots < \eta_{m-2} < 1$.

The idea is to use a two - point osculator interpolation polynomial P_{2n+1} to solve problem (4) by the following steps:

Step 1:

Divide the interval domain [0, 1] in to m-1 subinterval by η_i, i=1, 2, …,m-2, i.e., $[0,\eta_1]$, $[\eta_1,\eta_2]$, … , $[\eta_{m-3},\eta_{m-2}]$, $[\eta_{m-2},1]$, then *apply suggested method for each subintervals as follows.*

Step 2:

Construct osculator interpolation polynomial P_{2n+1} for each subintervals by evaluating Taylor coefficients of y about x= 0, η_i, 1, $\forall i = 1,2,...,m-2$ respectively.

Step 3:

Insert the series form in step 2 into equation (4a) and put x= 0, η_i, 1, $\forall i = 1,2,...,m-2$ respectively, and equate the coefficients of powers of x, (x-η_i), (x-1), $\forall i = 1,2,...,m-2$, to obtain

$y^{(n)}(0), y^{(n)}(\eta_i), y^{(n)}(1), \forall i = 1,2,...,m-2$ respectively.

i.e., to obtain a_n, $y^{(n)}(\eta_i)$, b_n, $\forall i = 1,2,...,m-2$.

Step 4:

Derive equation (4a) with respect to x to obtain new form of equation:

$$y^{(n+1)}(x) = \frac{df(x, y, y', ..., y^{(n)})}{dx} \quad (5)$$

Step 5:

Insert the series form in step 2 into equation (5) and put x= 0, η_i, 1, $\forall i = 1,2,...,m-2$, respectively and equate the coefficients of powers of x, (x-η_i), (x-1), $\forall i = 1,2,...,m-2$ to obtain $y^{(n+1)}(0)$, $y^{(n+1)}(\eta_i)$, $y^{(n+1)}(1)$, $\forall i = 1,2,...,m-2$, respectively.

i.e., to obtain a_{n+1}, $y^{(n+1)}(\eta_i)$, b_{n+1}, $\forall i = 1,2,...,m-2$.

Step 6:

Iterate the process in step 5 many times to obtain $y^{(j)}(0), y^{(j)}(\eta_i), y^{(j)}(1)$, i=1, …, m-2, j = (n+2), (n+3) ,… respectively. (i.e., to obtain a_j, $y^{(j)}(\eta_i)$, b_j).

The resulting equations can be solved using MATLAB package.

Step 7:

Use the coefficients obtained in above steps to construct P_{2n+1} for each sub-intervals $[0,\eta_1]$, $[\eta_1,\eta_2]$, … , $[\eta_{m-3},\eta_{m-2}]$, $[\eta_{m-2},1]$, the constructing polynomials have unknown coefficients $y^{(k)}(0)$, $y^{(k)}(1)$, $y^{(k)}(\eta_i)$, $\forall i = 1,2,...,m-2$, $\forall k = 1,2,...,n-1$, we can get half of these unknown coefficients by the boundary conditions.

Step 8:

To evaluate the remainder coefficients integrate equation (4a) on [0, x], $[\eta_i$,x], i=1, …, m-2, respectively.

Step 9:

Again integrate resulting equations in step 8, (n-1) times on [0, x], $[\eta_i$, x], i=1,…,m-2, respectively.

Step 10:

Use P_{2n+1} as a replacement of y in each equations in step 8 and 9, then put, x =η_i, i=1, …, m-2 and x=1, respectively, in these equations to obtain system of n(m-1) equations with n(m-1) unknown coefficients which can be solved using the MATLAB package to get the unknown coefficients, then insert it into P_{2n+1} of each subintervals.

Step 11:

Summing the P_{2n+1} of each subinterval obtained in step 10 which represent the polynomial solution of problem (4).

Now, we introduce many examples of high order multi-point BVP's for ODE to illustrate suggested method. Accuracy and efficiency of the suggested method is established through comparison with other methods.

Example 1:

Consider the following linear, 4th order, 3-point BVP's:

$$y^{(4)} = y + 4e^{x} \quad , \quad 0 \le x \le 1$$

with BC :

$$y(0)=1,\ y'(0)=2,\ y(1)=2e,\ y(\tfrac{1}{2})=\tfrac{1}{2}\left(3e(\tfrac{1}{2})\right)$$

Hence the exact solution has the form [12]: $y(x)=(1+x)e^{x}$

Solving this example by using suggested method, from equation (4), we get:

P_{15} = 2.15276058 10^{-11} x^{15} + 0.0000000001345 x^{14} + 0.00000000233 x^{13} + 0.000000027027 x^{12} + 0.000000300722 x^{11} + 0.00000303125 x^{10} + 0.00002755733512x^{9} + 0.0002232142836 x^{8} + 0.001587301587 x^{7} + 0.009722222222 x^{6} + 0.05x^{5} + 0.2083333333x^{4} + 0.6666666667x^{3}+1.5x^{2} +2x+1.

For more details, Table (1) gives the results for different nodes in the domain, for n=7, i.e. P_{15}, and the absolute errors obtained by comparing it with the exact solution. Figure (1) illustrates the accuracy of solution by comparing P_{15} with the exact solution.

Table 1: The Accuracy of the Suggested Method P_{15} for Example 1.

x_i	Exact solution y(x)	Suggested solution P_{15}	Error = | y(x) – P_{15} |
0	1.000000000000000	1.000000000000000	0
0.1	1.215688009883213	1.215688009883242	$0.002975397705995e^{-011}$
0.2	1.465683309792204	1.465683309792417	$0.021338486533296e^{-011}$
0.3	1.754816449848804	1.754816449849434	$0.062949645496246e^{-011}$
0.4	2.088554576697778	2.088554576699057	$0.127897692436818e^{-011}$
0.5	2.473081906050192	2.473081906052270	$0.207744932367859e^{-011}$
0.6	2.915390080624814	2.915390080627676	$0.286126677906395e^{-011}$
0.7	3.423379602699810	3.423379602703189	$0.337863070853928e^{-011}$
0.8	4.005973671286442	4.005973671289734	$0.329158922340866e^{-011}$
0.9	4.673245911198205	4.673245911200373	$0.216804352248801e^{-011}$
1	5.436563656918091	5.436563656917566	$0.052491344604277e^{-011}$
Max. error		$3.378630708539276e^{-012}$	
S.S.E		$4.180676135479167e^{-023}$	

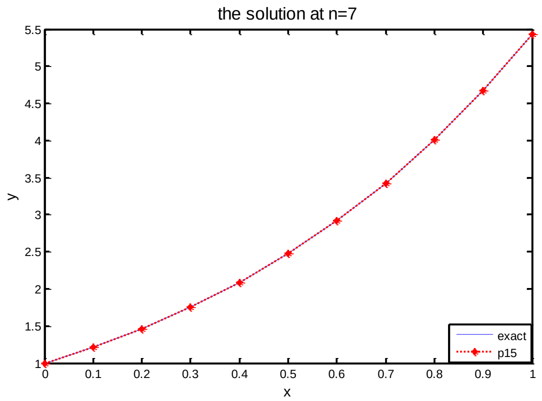

Fig.1: Comparison between P_{15} and the Exact Solution for Example 1.

Fazal-i-Haq [12] solved this example by numerical method Based on uniform Haar wavelets with Maximum absolute $5.2462e^{-09}$ and relative Errors $2.2622e^{-09}$ and better performance of the suggested algorithm can be observed.

Example 2:

Consider the following nonlinear, 4th order, 4-point BVP's:

$$y^{(4)} - y^2 + x^8 - 2x^5 + x^2 - 24 = 0 \quad , \quad 0 \le x \le 1$$

with BC: $y^{(3)}\left(\frac{1}{4}\right) = -6 \qquad y'(0) = 1$

$$y\left(\frac{1}{2}\right) = \frac{7}{16} \qquad y'(1) = -3$$

Hence the exact solution has the form [13]: $y(x) = -x^4 + x$

Solving this example by using suggested method from equation (4), we get:

$P_5 = -x^4 + x$, which is the exact solution.

3. Error Estimation for Multi-point Boundary Value Problems:

This paper, present a new, carefully designed modification of this error estimate which not only results in less computational work but also appears to perform satisfactorily for nonlocal MPBVP, and gives a full analytical justification for the asymptotical correctness of the error estimate when it is applied to a general nonlinear problem.

3.1. Error / Defect Weights

The weights used to scale either the error or the maximum defect differs among BVP software. Therefore, the BVP component of pythODE allows users to select the weights they wish to use. The default weights depend on whether an estimate of the error or maximum defect is being used. If the error is being estimated, then the BVP component of pythODE uses [14]. In this paper we modify this package to consist MPBVP and named "pythMPODE", defined as:

$$\frac{\|y(x) - p(x)\|_\infty}{1 + \|p(x)\|_\infty} \quad ; \quad 0 \le x \le 1 \quad (6)$$

where y(x) is exact solution and P(x) is suggested solution of MPBVP.

If the maximum defect is being estimated, then the MPBVP component of "pythMPODE" uses:

$$\frac{\|p_{2n+1}^{(n)}(x) - f(x, p(x), p'(x), \ldots, p^{(n-1)}(x))\|_\infty}{1 + \|f(x, p(x), p'(x), \ldots, p^{(n-1)}(x))\|_\infty} \quad ; \quad 0 \le x \le 1 \quad (7)$$

The relative estimate of both the error and the maximum defect are slightly modified from the one used in BVP SOLVER.

We apply this package for example 1 as follows:

$$\frac{\|p^{(4)}{}_{2n+1} - f(x, p, \ldots, p^{(3)})\|_\infty}{1 + \|f(x, p, \ldots, p^{(3)})\|_\infty} = \frac{3.996802888650564e^{-013}}{1 + 16.309690970754275} = 2.308997252119286e^{-014}$$

For more details see Table (2).

Table 2: The Maximum Relative Defect of Example 1.

x_i	$f(x, p, \ldots, p^{(3)})$	$1+\lvert f(x, p, \ldots, p^{(3)})\rvert$	$(P_{2n+1})^{(4)}$	$\lvert(P_{2n+1})^{(4)} - f(x, p, \ldots, p^{(3)})\rvert$	$(\lvert(P_{2n+1})^{(4)} - f(x, \ldots, p^{(3)})\rvert / 1+\lvert f(x, p, \ldots, p^{(3)})\rvert)$
0	6.000000000000000	5.000000000000000	5.000000000000000	0	0
0.1	5.636371682185802	6.636371682185802	5.636371682185732	0.070166095156310e^{-012}	0.040535729537205e^{-013}
0.2	6.351294342432883	7.351294342432883	6.351294342432856	0.027533531010704e^{-012}	0.015906425514600e^{-013}
0.3	7.154251680152817	8.154251680152817	7.154251680153186	0.368594044175552e^{-012}	0.212940857695445e^{-013}
0.4	8.055853367262859	9.055853367262859	8.055853367263067	0.207833750209829e^{-012}	0.120067857110203e^{-013}
0.5	9.067966988850706	10.067966988850706	9.067966988850307	0.399680288865056e^{-012}	0.230899725211929e^{-013}
0.6	10.203865282186849	11.203865282186849	10.203865282186456	0.392574861507455e^{-012}	0.226794841208161e^{-013}
0.7	11.478390432581719	12.478390432581719	11.478390432581815	0.095923269327614e^{-012}	0.055415934050863e^{-013}
0.8	12.908137385256314	13.908137385256314	12.908137385256508	0.193622895494627e^{-012}	0.111858089102668e^{-013}
0.9	14.511658355826007	15.511658355826007	14.511658355826029	0.021316282072803e^{-012}	0.012314652011303e^{-013}
1	16.309690970754275	17.309690970754275	16.309690970754275	0	0
	Max. error			3.996802888650564e^{-013}	2.308997252119 29e^{-014}

3.2. Global - Error Methods

There are a number of different algorithms that can be used to estimate the global error effectively. These algorithms are based on the use of Richardson extrapolation, higher-order formulae, deferred corrections, and a conditioning constant. The first and second global-error estimation algorithms are modified and described below.

3.2.1. Richardson extrapolation

This algorithm starts with a discrete numerical solution Y_h for a given mesh. Next, the software determines a more accurate numerical solution $Y_{h/2}$ by halving each subinterval of the original mesh.

Then, an estimate of the norm of the global error, e_{RE}, is given by:

$$e_{RE} = \left\| \left(\frac{2^p}{2^p - 1} \right) \left(Y_h - Y_{h/2} \right) \right\|_\infty \quad (8)$$

where p is the order of the discretization formula.

In this paper, we modify this algorithm to represent the suggested method that starts with a discrete solution P_{2n+1} for a given mesh. Next, the software determines a more accurate solution $P_{2(n+1)+1}$ by increasing n number of fit order for derivative of approximate with derivative of exact.

Then, an estimate of the norm of the global error, e_{MPRE}, is given by:

$$e_{MPRE} = \left\| \frac{2^{2(n+1)+1}}{\left(2^{2(n+1)+1} - 1\right)} \left(P_{2(n+1)+1} - P_{2n+1} \right) \right\|_\infty \quad (9)$$

We apply this algorithm for example 1, as follows:

Apply equation (9) by the following (for more details see Table (3)):

Table 3: Appling Modify Richardson Extrapolation for Example 1.

8	P13	P15	| P15 - P13 |	Mod. Richardson
0	1.000000000000000	1.000000000000000	0	0
0.1	1.215688009883212	1.215688009883212	0	0
0.2	1.465683309792205	1.465683309792204	0.008881784197001e-013	0.008882055255816e-013
0.3	1.754816449848809	1.754816449848804	0.053290705182008e-013	0.053292331534899e-013
0.4	2.088554576697792	2.088554576697779	0.133226762955019e-013	0.133230828837246e-013
0.5	2.473081906050211	2.473081906050192	0.182076576038526e-013	0.182082132744237e-013
0.6	2.915390080624828	2.915390080624814	0.137667655053519e-013	0.137671856465155e-013
0.7	3.423379602699815	3.423379602699810	0.053290705182008e-013	0.053292331534899e-013
0.8	4.005973671286443	4.005973671286442	0.008881784197001e-013	0.008882055255816e-013
0.9	4.673245911198205	4.673245911198205	0	0
1	5.436563656918091	5.436563656918091	0	0
Max. error			1.820765760385257e-014	1.820821327442369e-014

$$e_{MPRE} = \left\| \left(\frac{2^{2(n+1)+1}}{2^{2n+1} - 1} \right) \left(p_{2(n+1)+1} - p_{2n+1} \right) \right\|_\infty = \left\| \left(\frac{2^{15}}{2^{15} - 1} \right) \left(p_{15} - p_{13} \right) \right\|_\infty = 1.820821327442369e^{-014}$$

3.2.2. Higher - Order Formulae

Higher – order formulae can be used to determine a more accurate numerical solution with the same mesh as for the original solution. Specifically, the global error can be estimated by:

$$e_{HO} = \| Y_p - Y_q \|_\infty \quad (10)$$

where Y_p is the original discrete solution of order p and Y_q is the more accurate discrete solution of order q > p. In [14] symmetric formulae are used, q = p + 2.

In this paper, we modify this algorithm to represent the suggested method that starts with suggested solution P_{2n+1} for a given mesh. Next, the software determines a more accurate solution $P_{2(n+1)+1}$ by increase n, number of fit order for derivative of approximate with derivative of exact.

Then, the global error can be estimated by:

$$e_{HO} = \| P_{2(n+1)+1} - P_{2n+1} \|_\infty \quad (11)$$

We apply this algorithm for the example 1, as follows:
Apply equation (11) by the following (for more details see Table (3)):
$e_{HO} = \|y_p - y_q\|_\infty = \|p_{15} - p_{13}\|_\infty =$ $1.820765760385257e^{-014}$.

References:

[7] D. R. Anderson and J. Hoffacker, "Existence of Solutions to A Third-Order Multi-Point Problem on Time Scales", Electronic Journal of Differential Equations, Vol. 2007, No. 107, pp. 1–15, 2007.

[3] Debabrata, D.; Prasanta, S. and Kashinath, S. (2008) "Elasto-plastic strain analysis by a semi-analytical method" Sādhanā, © Printed in India, Vol. 33, Part 4, pp. 403–432.

[12] Fazal-i-Haq, "Numerical Solution of Fourth Order Boundary-Value Problems Using Haar Wavelets", Applied Mathematical Sciences, Vol. 5, No. 63, PP: 3131 – 3146, 2011.

[1] Grundy, R. E. (2005) "The application of Hermite interpolation to the analysis of non-linear diffusive initial-boundary value problems", IMA Journal of Applied Mathematics 70, pp.814-838.

[4] Gupta C. P. "A Non-Resonant Generalized Multi-Point Boundary Value Problem of Dirichelet Type Involving A P-Laplacian Type Operator, Electronic Journal of Differential Equations" Conference 15, PP: 127-139. (2007).

[14] J. J. Boisvert, "A Problem-Solving Environment for the Numerical Solution of Boundary Value Problems", M.Sc. Thesis, Department of Computer Science, College of Graduate Studies and Research, University of Saskatchewan, Saskatoon, 2011.

[9] J. R. Graef, L. Kong and Q. Kong, "Higher Order Multi-Point Boundary Value Problems", Mathematische Nachrichten, Vol. 284, Issue 1, pp: 39–52, January 2011.

[8] J. Wang and F. Xu, "Nontrivial Solutions for Nonlinear Higher Order Multi-Point Boundary Value Problem on Time Scales with All Derivatives", Applied Mathematical Sciences, Vol. 3, No. 48, pp: 2349 – 2358, 2009.

[13] M. Dehghan and F. Shakeri, "A Semi-Numerical Technique for Solving the Multi-Point Boundary Value Problems and Engineering Applications", International Journal of Numerical Methods for Heat & Fluid Flow, Vol. 21, Iss: 7, pp: 794 – 809, 2011.

[2] Maqbool, M.; Coupechoux, M. and Godlewski, P. (2008) "A Semi-analytical Method to Model Effective SINR Spatial Distribution in WiMAX Networks", Draft Version, Institut Telecom Paris Tech, December 23.

[11] Phillips, G .M. (1973) "Explicit forms for certain Hermite approximations", BIT 13, 177-180.

[5] Q. Zhang, S. Chen and J. Lü, "Upper and Lower Solution Method for Fourth-Order Four-Point Boundary Value Problems", Journal of Computational and Applied Mathematics, Vol. 196, pp: 387 – 393, 2006.

[6] Y. Liu, "Solutions of Sturm–Liouville Type Multipoint boundary value problems for Higher order Differential Equations", J. Appl. Math. & Computing, Vol. 23, No. 1 – 2, pp: 167 – 182, 2007.

[10] Y. Liu, Z. Weiguo and S. Chunfang, "Monotone and convex positive solutions for fourth-order multi-point boundary value problems", Boundary Value Problems a Springer Open Journal Vol. 2011, pp: 1-12, 2011.

حل مسائل القيم الحدودية متعددة النقاط من الرتب العالية غير خطية باستخدام التقنية شبه التحليلية

لمى ناجي محمد توفيق* ***مريم محمد هلال****

*قسم الرياضيات، كلية التربية ابن الهيثم ، جامعة بغداد

الخلاصة:

في هذا البحث نعرض خوارزمية جديدة لحل معادلات تفاضلية اعتيادية من الرتب العالية غير الخطية ذات الشروط الحدودية عند نقاط متعددة، الخوارزمية تعمل على أساس التقنية شبه التحليلية والحل يحسب بصيغة متسلسلة سريعة التقارب وهذا يتضح أكثر في المسائل الفيزيائية، أيضا ناقشنا بعض الأمثلة لتوضيح الدقة و الكفاءة وسهولة أداء الطريقة المقترحة في حل هذا النوع من المسائل الحدودية متعددة النقاط .