World Applied Sciences Journal 21 (Mathematical Applications in Engineering): 129-133, 2013
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2013.21.mae.99937

Solving Boundary Value Problems With Neumann Conditions Using Direct Method

[1]P.S. Phang, [1,2]Z.A. Majid, [2]M. Suleiman and [1]F. Ismail

[1]Department of Mathematics, Faculty Science,
University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
[2]Institute for Mathematical Research, University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia

Abstract: In this paper, the direct method is utilized for solving second order two-point boundary value problem of Neumann type. The method will obtain the solution of the second order boundary value problem directly without reducing it to to first order equations. The method will be implemented using variable step size via shooting technique adapted with the Newton method. Numerical results are given to compare the efficiency of the proposed method with the bvp4c from the Matlab solver.

Key words: Direct method · Neumann type · Shooting technique

INTRODUCTION

Two-point boundary value problems have been widely arisen in modeling of chemical reactions, the boundary layer theory in fluid mechanic and heat power transmission theory. These problems can be presented in several types of boundary conditions: e.g. Dirichlet, Neumann and mixed. Dirichlet boundary condition is the common boundary condition and has been solved by several researchers such as Hamid *et al.* [1] and Mohamad [2]. Liu [3] studied on Neumann-type boundary value problems and Han and Wang [4] proved the existence of solutions to mixed two point boundary-value problem for impulsive differential equations by variational methods. Robin boundary condition is another type of boundary condition; it is a linear combination of Dirichlet and Neumann boundary conditions.

We are concerned for solving the Neumann type boundary value problem. There are many analytical and numerical techniques available to solve boundary value problem with Neumann condition including several well-known methods, such as Adomian decomposition method, finite difference method and collocation method. Dehghan [5] approached the numerical solution of a non-local boundary value problem with Neumann's boundary conditions by using finite difference method. Ramadan [6] and Liu *et al.* [7] solved the Neumann type boundary value problems by polynomial and nonpolynomial spline approach. Siraj-ul-Islam *et al.* [8] proposed the collocation method with the Haar wavelets to solve second order boundary value problem and Yao [9] applied iterative method of nonlinear Neumann boundary value problems. Recently, Aly *et al.* [10] solved the two-point nonlinear boundary value problems with Neumann boundary conditions by using Adomian decomposition method. Besides that, Kierzenka and Shampine [11] introduced a boundary value problem solver based on residual control and the MATLAB which call bvp4c.

The purpose of this paper is to establish a new algorithm for solving the linear and nonlinear second order two-point boundary value problem subjected to Neumann boundary condition directly. The approach for solving higher order ordinary differential equation directly has been suggested by Suleiman [12] and; Majid and Suleiman [13]. We will extend the direct method using variable step size from Majid and Suleiman [13] and adapted with shooting technique via Newton method to solve the boundary value problem.

MATERIALS AND METHODS

Consider the second order two-point boundary value problem of the form:

$$y'' = f(x, y, y'), a \le x \le b \qquad (1)$$

Subject to the Neumann boundary conditions:

$$y'(a) = \alpha, y'(b) = \beta \qquad (2)$$

Corresponding Author: Z.A. Majid, Institute for Mathematical Research, University Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia.

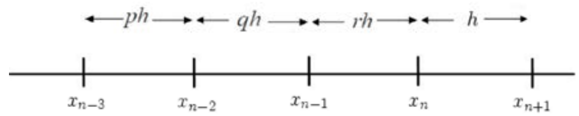

Fig. 1: Direct method variable step size

In Fig. 1 show that the approximated value y_{n+1} has the current step size, h and the previous step size were rh, qh and ph. The corrector formulae will involve the set of points $\{x_{n-3}, x_{n-2}, x_{n-1}, x_n, xn_{+1}\}$, while the predictor formulae will involve the set of points $\{x_{n-3}, x_{n-2}, x_{n-1}, x_n,\}$. The corrector formulae of direct method were derived using Lagrange interpolation polynomial of order five and the predictor formulae were derived using the Lagrange interpolation polynomial of order four. We obtained the approximation values of y_{n+1} at the points x_{n+1} by integrating once and twice over Eq. (1) with respect to x over the interval $[x_n, x_{n+1}$.

$$\int_{x_n}^{x_{n+1}} y''(x)\,dx = \int_{x_n}^{x_{n+1}} f\left(x,y,y'\right)dx,$$
$$\int_{x_n}^{x_{n+1}}\int_{x_n}^{x} y''(x)\,dx\,dx = \int_{x_n}^{x_{n+1}}\int_{x_n}^{x} f\left(x,y,y'\right)dx\,dx \tag{3}$$

Let $x_{n+1} = x + h$, the Eq. (3) gives:

$$y'_{n+1} = y'_n + \int_{x_n}^{x_{n+1}} f\left(x,y,y'\right)dx$$
$$y_{n+1} - y_n - hy'_n = \int_{x_n}^{x_{n+1}}\left(x_{n+1}-x\right) f\left(x,y,y'\right)dx \tag{4}$$

The function $f(x,y,y')$ in (4) will be approximated using Lagrange interpolating polynomial, the value of y_{n+1} can be obtained by using MAPLE and the corrector formulae can be obtained as follows:

$$\begin{aligned} y'_{n+1} = y'_n + h\Bigg(& -\left(\frac{-3+5q+10(r+qr+r^2)}{60p(p+q)(p+q+r)(p+q+r+1)}\right)f_{n-3} \\ & +\left(\frac{3+5(p+q)+10(r+pr+qr+r^2)}{60pq(q+r)(q+r+1)}\right)f_{n-2} \\ & -\left(\frac{3+5(p+2q)+10(pq+q^2+r+pr+2qr+r^2)}{60rq(p+q)(r+1)}\right)f_{n-1} \\ & +\Bigg(\frac{3+5(p+2q+3r)+10(pq+q^2+2pr+4qr)}{60r(q+r)(p+q+r)} \\ & \cdot\left(\frac{30(pqr+q^2r+r^2+pr^2+r^3+2qr^2)}{60r(q+r)(p+q+r)}\right)f_n \\ & +\Bigg(\frac{12+15(p+2q+3r)p+20(pq+q^2+2pr+4qr)}{60(1+r)(1+q+r)(p+q+r+1)} \\ & \cdot\left(\frac{30(pqr+q^2r+2r^2+pr^2+r^3+2qr^2)}{60(1+r)(1+q+r)(p+q+r+1)}\right)f_{n+1}\Bigg) \end{aligned}$$

$$\begin{aligned} y_{n+1} = y_n + hy'_n + h^2\Bigg(& -\left(\frac{-1+2q+4r+5r(r+q)}{60p(p+q)(p+q+r)(p+q+r+1)}\right)f_{n-3} \\ & +\left(\frac{1+2(p+q+2r)+5(pr+qr+r^2)}{60pq(q+r)(q+r+1)}\right)f_{n-2} \\ & -\left(\frac{1+2(p+2q+2r)+5(pq+q^2+pr+2qr+r^2)}{60rq(p+q)(r+1)}\right)f_{n-1} \\ & +\Bigg(\frac{1+2(p+2q+3r)+5(pq+q^2+2pr+3r^2)}{60r(q+r)(p+q+r)} \\ & \cdot\left(\frac{20(qr+pqr+q^2r+pr^2+r^3+2qr^2)}{60r(q+r)(p+q+r)}\right)f_n \\ & +\Bigg(\frac{2+3(p+2q+3r)p+5(pq+q^2+2pr+3r^2)}{60(1+r)(1+q+r)(p+q+r+1)} \\ & \cdot\left(\frac{10(2qr+pqr+q^2r+pr^2+r^3+2qr^2)}{60(1+r)(1+q+r)(p+q+r+1)}\right)f_{n+1}\Bigg) \end{aligned} \tag{5}$$

The direct method will solve boundary value problem of Neumann type adapted with shooting technique via Newton method.

Implementation of the Method

Shooting Technique: The shooting technique used to form the boundary value problem of Neumann boundary condition to initial value problems. The idea in shooting technique is to obtain the missing initial value until the boundary condition at the other end converges to its correct value. In order to correcting the guessing value, Newton method is adapted. Eq. (1) and (2) can be written by using shooting technique:

$$y'' = f\left(x,y,y'\right),\quad a \le x \le b$$
$$y(a) = s_v,\quad y'(a) = \alpha,\quad v = 0,1,2,\ldots\,. \tag{6}$$

We choose $s_0 = (\beta - \alpha)/(b - a)$ which is referring to Faires and Burden [14].

The stop condition for shooting technique is given as follow:

$$\varphi(s) = y'(b, s_v) - \beta \le TOL \tag{7}$$

We compute the $\{s_{v}\}$ by Newton method:

$$s_{v+1} = s_v - \frac{\varphi(s_v)}{\varphi'(s_v)}. \tag{8}$$

Differentiate (6) with respect to s and it is simplify as follows:

$$z'' = \frac{d}{dy} f(x,y,y')z + \frac{d}{dy'} f(x,y,y')z',\, a \le x \le b$$
$$z(a) = 1,\quad z'(a) = 0. \tag{9}$$

Therefore, the solutions of (8) will gives $\varphi'(s_v)$ = iz.(b,s_v). The new guess can be calculated base on the previous guess using:

$$s_{v+1} = s_v - \frac{y'(b,s_v) - \beta}{z'(b,s_v)}, \qquad v = 1,2\ldots \tag{10}$$

Both of equations (6) and (9) will be solved simultaneously using the direct method. The process is repeated over and over until the error $|\beta - y'(b,s_v)| \le TOL$.

Variable Step Size Strategy: In order to reduce the computation time, the variable step size strategy will be implemented and this strategy is referring to Shampine and Gordon [15]. Three basic strategies are proposed for the step size adjustment, where the next step size will be restricted to half, double or the same as the current step size. The successful step size will remain constant for at least two blocks before we considered the next step size to be doubled. When a fail step occurs, the next step size will be halved of the previous step size. The following are some of the cases for choosing step size:

Case 1: First time successful step: (p = q = r = 1). Substitute $p = q = r = 1$ in (5) will produce the following corrector formulae:

$$y'_{n+1} = y'_n - \frac{h}{720}\left(-251f_{n+1} - 646f_n + 264f_{n-1} - 106f_{n-2} + 19f_{n-3}\right)$$

$$y_{n+1} = y_n + hy'_n - \frac{h^2}{1440}\left(-135f_{n+1} - 752f_n + 264f_{n-1} - 96f_{n-2} + 17f_{n-3}\right)$$

Case 2: Second time successful step: (p = q = r = 1/2). Substitute $p = q = r = 1/2$ in (5) will produce the following corrector formulae:

$$y'_{n+1} = y'_n - \frac{h}{900}\left(-269f_{n+1} - 1360f_n + 1220f_{n-1} - 615f_{n-2} + 124f_{n-3}\right)$$

$$y_{n+1} = y_n + hy'_n - \frac{h^2}{1800}\left(-129f_{n+1} - 1430f_n + 1080f_{n-1} - 525f_{n-2} + 104f_{n-3}\right)$$

Case 3: First time failure step: ($p = q = r = 2$). Substitute $p = q = r = 2$ in (5) will produce the following corrector formulae:

$$y'_{n+1} = y'_n - \frac{h}{100800}\left(-40192f_{n+1} - 68005f_n + 10255f_{n-1} - 3423f_{n-2} + 565f_{n-3}\right)$$

$$y_{n+1} = y_n + hy'_n - \frac{h^2}{100800}\left(-13808f_{n+1} - 42175f_n + 4935f_{n-1} - 1617f_{n-2} + 265f_{n-3}\right)$$

The successful step is dependent on the condition local truncation error (LTE) < TOL. If this condition fails, the values of the approximate solution, y_{n+1} are rejected and the current step size is reduce and recalculate the approximate solution.

Algorithm of Direct Method Variable Step Size (DMVS) via Shooting Technique Adapted with Newton Method:

Step 1 : Set $Tol\ s_0$
Step 2 : Set $x = x + h$, evaluate y_{n+1} and z_{n+1} with direct method and compute f_{n+1} and z^n_{n+1}.
Step 3 : If $x < b$, repeat Step 2. If $x = b$, go to Step 4.
Step 4 : If fulfill stop condition: $|\beta - y'(b,s_v)| \le Tol$, go to Step 6. If not, go to Step 5.
Step 5 : Generate the new guessing values by Eq. (9) and go to Step 2.
Step 6 : Complete.

This algorithm was developed in C language.

RESULTS AND DISCUSSION

In this section, four numerical examples are presented. These problems will be tested by the direct method with three different, Tol: 10^{-3} and 10^{-5}. The numerical results will be comparing to the MATLAB solver, bvp4c for two different step sizes, h: 0.1 and 0.05. The MATLAB solver, bvp4c solves two point boundary value problems by collocation method. The following notations are used in the tables:

h	Step size
Tol	Tolerance
MAXE	Maximum error
AVE	Average error
TFC	Total function call
TS	Total step at last iteration
IG	Total iteration of guess
FS	Failure step at last iteration
bvp4c	MATLAB solver
DMVS	Direct method variable step size adapted with shooting technique via Newton method.

Problem 1:
Linear boundary value problem:

$$y'' = -y - 1,\ 0 \le x \le 1.$$

Neumann boundary condition:

$$y'(0) = (1 - \cos(1))/\sin(1),\ y(1) = -(1 - \cos(1))/\sin(1).$$

Exact Solution:

$$y = \cos(x) + \frac{1-\cos(1)}{\sin(1)}\sin(x) - 1.$$

Problem 2:
Linear boundary value problem:

$$y'' = -xy + (3 - x - x^2 + x^3)\sin(x) + 4x\cos(x),\ 0 \le x \le 1.$$

Neumann boundary condition:

$$y'(0) = -1,\ y'(1) = 2\sin(1).$$

Exact Solution:

$$y = \sin^2(\pi x).$$

Problem 3:
Non-linear boundary value problem:

$$y'' = y^2 + 2\pi^2\cos(2\pi x) - \sin^4(\pi x),\ 0 \le x \le 1.$$

Neumann boundary condition:

$$y'(0) = 0,\ y'(1) = 0.$$

Exact Solution:

$$y = \cos(x) + \frac{1-\cos(1)}{\sin(1)}\sin(x) - 1.$$

Problem 4:
Non-linear boundary value problem:

$$y'' = -\exp(-2y),\ 0 \le x \le 1.$$

Neumann boundary condition:

$$y'(0) = 1,\ y'(1) = \frac{1}{2}.$$

Exact Solution:

$$y = \ln(1+x).$$

The numerical results of the bvp4c and DMVS to solve Problem 1-4 are presented in Tables 1-4 respectively. Firstly, we are interested to discuss the numerical results obtained by bvp4c with two different step size and DMVS at Tol = 10^{-3}. The maximum error and average error for DMVS is comparable with bvp4c in most of the cases. For example in Table 1, the maximum error for DMVS at Tol = 10^{-3} and bvp4c at *h*=0.1 and *h*=0.05 are 2.54E-06, 6.37E-06 and 6.16E-06 respectively. As the tolerance getting smaller, the maximum error and average error for DMVS is better compare to bvp4c when Tol = 10^{-5} and 10^{-7}. For example in Table 2 the average error for DMVS at Tol = 10^{-7} and bvp4c at *h*=0.05 are 3.26E-07 and 1.34E-05 respectively.

For the non-linear boundary value problem, both methods managed to give the similar conclusion, *e.g.* in Table 3 and 4, the maximum error for DMVS at Tol = 10^{-7} and bvp4c at *h*=0.05 are 1.46E-06, 4.75E-04 and 2.51E-07,

Table 1: Comparison bvp4c with DMVS for Problem 1

	bvp4c		DMVS		
	h=0.1	h=0.05	Tol=e-3	Tol=e-5	Tol=e-7
MAXE	6.4e-6	6.2e-6	2.5e-6	8.7E-8	2.6e-9
AVE	6.6e-6	6.3e-6	2.4e-6	8.3E-8	2.4e-9
TFC	117	227	63	79	102
TS	-	-	26	34	42
IG	-	-	2	3	3
FS	-	-	0	0	0

Table 2: Comparison bvp4c with DMVS for Problem 2

	bvp4c		DMVS		
	h=0.1	h=0.05	Tol=e-3	Tol=e-5	Tol=e-7
MAXE	1.7e-5	1.5e-5	3.2e-4	2.3e-5	3.8e-7
AVE	1.4e-5	1.3e-5	2.9e-4	2.1e-5	3.3e-7
TFC	110	210	64	84	125
TS	-	-	26	34	50
IG	-	-	2	2	2
FS	-	-	0	0	0

Table 3: Comparison bvp4c with DMVS for Problem 3

	bvp4c		DMVS		
	h=0.1	h=0.05	Tol=e-3	Tol=e-5	Tol=e-7
MAXE	1.0e+0	4.8e-4	8.6e-4	7.0e-5	1.5e-6
AVE	9.9e-1	4.4e-4	1.1e-4	9.8e-6	2.9e-7
TFC	220061	74129	90	124	210
TS	-	-	36	49	83
IG	-	-	1	1	1
FS	-	-	0	0	0

Table 4: Comparison bvp4c with DMVS for Problem 4

	bvp4c		DMVS		
	h=0.1	h=0.05	Tol=e-3	Tol=e-5	Tol=e-7
MAXE	2.5e-5	1.5e-5	3.9e-5	4.7e-6	2.5e-7
AVE	2.5e-5	1.5e-5	3.6e-5	4.4e-6	2.3e-7
TFC	202	382	65	85	123
TS	-	-	27	36	52
IG	-	-	6	6	7
FS	-	-	0	0	0

1.50E-05 respectively. We observed that the total function call for DMVS is less than bvp4c in all cases, this is expected because the DMVS solving boundary value problem directly with variable step size but bvp4c reduce the boundary value problem to the first order system equations and solve it using constant step size. For example in Table 3, the function call for DMVS is only 210 but bvp4c need 74129 function calls. As the tolerance getting smaller DMVS obtain better accuracy.

CONCLUSION

The main advantage of this paper is to apply the direct way of dealing with the second order boundary value problem. We have shown the proposed direct method with shooting technique using variable step size is suitable for solving second order linear and non-linear two-point boundary value problems with Neumann type. The numerical result shown that in term of accuracy, our proposed method can generate better accuracy even with less function calls.

ACKNOWLEDGEMENTS

The authors gratefully acknowledged the financial support of Research University Grant Scheme (RUGS), No. project: 05-03-10-0973RU and sponsorship for MyBrain15 under Ministry of Higher Education.

REFERENCES

1. Hamid, N.N.A., A.A. Majid and A.I.M. Ismail, 2011. Extended Cubic B-spline Method for Linear Two-Point Boundary Value Problems. Sains Malaysiana, 40(11): 1285-1290.
2. Mohamad, A.J., 2010. Solving second order non-linear boundary value problems by four numerical methods. Eng. and Tech. Journal, 28(2): 369-381.
3. Liu, Y., 2009. Studies on Neumann-type boundary value problems for second-order nonlinear p-Laplacian-like functional differential equations. Nonlinear Analysis: Real World Applications, 10: 333-344.
4. Han, Z. and S. Wang, 2011. Mixed two-point boundary value problems for impulsive differential equations. Electronic Journal of Differential Equations, 35: 1-14.
5. Dehghan, M., 2003. Numerical solution of a non-local boundary value problem with Neumann's boundary conditions. Commun. Numer. Meth. Engng, 19: 1-12.
6. Ramadan, M.A., I.F. Lashien and W.K. Zahra, 2007. Polynomial and nonpolynomial spline approaches to the numerical solution of second order boundary value problems. Applied Mathematics and Computation, 184: 476-484.
7. Liu, B.L., H.W. Liu and Y. Chen, 2011. Polynomial spline approach for solving second-order boundary-value problems with Neumann conditions. Applied Mathematics and Computation, 217: 6972-6882.
8. Siraj-ul-Islam, Aziz, I. and B. Sarler, 2010. The numerical solution of second-order boundary-value problems by collocation method with the Haar wavelets. Mathematical and Computer Modelling, 52: 1577-1590.
9. Yao, Q., 2010. Successively iterative method of nonlinear Neumann boundary value problems. Applied Mathematics and Computation, 217: 2301-2306.
10. Aly, E.H., E. Abdelhalim and R. Randolph, 2011. Advances in the Adomian decomposition method for solving two-point nonlinear boundary value problems with Neumann boundary conditions. Computers and Mathematics with Applications, doi:10.1016/j.camwa.2011.12.010.
11. Kierzenka, J. and L.F. Shampine, 2001. A BVP solver based on residual control and the MATLAB PSE. ACM Transactions on Mathematical Software, 27(3): 299-316.
12. Suleiman, M.B., 1989. Solving nonsiff higher order ODEs directly by direct integration method. Applied Mathematics and Computation, 33: 197-219.
13. Majid, Z.A. and M. Suleiman, 2006. Direct integreation implicit variable steps method for solving higher order systems of ordinary differential equations directly. Sains Malaysiana, 35: 63-68.
14. Faires, D. and R.L. Burden, 1998. Numerical Analysis., USA: International Thomson Publishing Inc.
15. Shampine, L.F. and M.K. Gordon, 1975. Computer Solution of Ordinary Differential Equations: The Initial Value Problem. San Francisco: W. H. Freeman and Company.