Journal of Engineering and Applied Sciences 14 (22): 8344-8347, 2019
ISSN: 1816-949X

NLVIEEIM Algorithm: A Solution of Nonlinear Volterra Integral Equations using Explicit-Implicit Method

Ronak Bagelany
University of Kirkuk, Kirkuk, Iraq
saeed.runak@yahoo.com

Abstract: Solving system of nonlinear Volterra integral equations has attracted enormous attention due to its importance in different sciences. Therefore, this study proposes a new efficient algorithm called NLVIEEIM. Specifically, the Gear Explicit method is developed to solve Systems of Nonlinear Volterra Integral Equations of the second kind (SNLVIE-2). Additionally, we used backward differentiation formula first-order implicit method to improve this solution. The algorithm is examined by numerical examples and the calculations are carried out using MATLAB 2014b Software. The outcomes show a better accuracy in solving SNLVIE-2.

Key words: Nonlinear integral equation, Volterra second kind, Gear Explicit method, accuracy, algorithm, backward differentiation formula first-order implicit method

INTRODUCTION

Large numbers of mathematical problems (such as mathematical electrostatics, physics, biology and mixed problems of mechanics of continuous media) are reduced to a system of nonlinear integral equations of the second kind (Rice, 2014). To deal with this system, various methods have been proposed and analytically edited (Wazwaz, 2011; Linz, 1985). According to the literature, numerical methods have an essential role in solving systems of nonlinear integral equationss by Maturi (2014). Furthermore, predictor-corrector methods are applied for the numerical solution of System of Nonlinear Volterra Integral Equation of second kind (SNLVIE-2) (Jumaa and Taqi, 2016). Moreover, very recently by Jumaa (2017) the SNLVIE-2 is addressed using the trapezoidal predictor-corrector method.

This study aims to study solving SNLVIE-2 using a new efficient algorithm called NLVIEEIM. The unknown functions manifest outside and inside the integral sign of the form (Taqi and Jumaa, 2009):

$$\varnothing_i(x) = f_i(x) + \sum_{j=1}^{m} \int_0^x k_{ij}\left(x, t\varnothing_j(t)\right) dt \quad i = 1, 2, \ldots, L \qquad (1)$$

where, $\varnothing_i(x)$ are unknown functions, the kernels k_{ij} (, ... ,) and the functions $f_i(x)$ are given real-valued functions on subsets of R^3 and R^1, respectively.

Conditions of the solution for SNLVIE-2: The specific conditions under which a solution exists for the SNLVIE-2 are as follows: the functions f_i (x) is integrable and bounded in $a \le x \le b$. The functions f_i (x) must satisfy the Lipschitz condition in the interval (a, b) which means $|f_i(x)-f_i(y)|<L_i|x-y|$. The function $G_i(x, t, u(t))$ is integrable and bounded $|G_i(x, t, u(t))|<K$ in a«x, t«b. The functions $G_i(x, t, u(t))$ must satisfy the Lipschitz condition:

$$\left|G_i(x, t, z) - G_i(x, t, z')\right| < M_i\left|z - z'\right|$$

where, I = 1, 0, ..., n.

Proof; Wazwaz (2011)

The Gear Explicit method for solving SNLVIE-2: The Gear Explicit is the most elementary approximation method which uses for evaluating initial-value problems, this method is (Fausett, 2008):

$$\begin{aligned} & w_0 = \propto \\ & w_{i+1} = w_i + hf(x_i, w_i), \quad \text{for each } i = 0, 1, \ldots, N-1 \\ & x_i = a + i*h, \quad h = \frac{b-a}{N}, \quad \text{for each } i = 0, 1, \ldots, N \end{aligned}$$

where, N is the number of the mesh points. Now, we develop this method and its application to evaluate an approximate solution of the SNLVIE-2 in Eq. 1 as follows:

$$\begin{aligned} & x_0 = 0,\ x_r = x_0 + r*h \\ & \varnothing_i(x_0) = f_i(x_0) \\ & \varnothing_i(x_r) = f_i(x_r) + \varnothing_i(x_{r-1}) + h*K_i\left(x_r, t_{r-1}, \varnothing_i(t_{r-1})\right) \end{aligned} \qquad (2)$$

where, r = 1, ..., n.

Algorithm; Nonlinear Volterra Integral Equations via. Gear Explicit (NLVIEGE):

Step 1: 1- Assume $h = \frac{b-a}{n}, n \in N$

2- Set $\varnothing_{i,0} = f_{i,0}$

Step 2: Compute $\varnothing_{i,r}; r = 1, 2, ..., N$ using Eq. (2)

BDF-first order implicit method: Backward Differentiation Formula family (BDF) are derived from a different use of interpolating polynomials and are all implicit methods. The BDF-first order implicit method is approximation technique for solving initial-value problems and given by Stoer and Bulirsch (2013) and Epperson (2013):

$$w_0 = \propto$$
$$w_{i+1} = w_i + hf(x_{i+1}, w_{i+1}), \quad \text{for each } i = 0, 1, ..., N-1$$
$$x_i = a + i*h, \quad h = \frac{b-a}{N}, \quad \text{for each } i = 0, 1, ..., N$$

where, N is the number of the mesh points. In this study, we developed this method and applied it to evaluate an approximate solution of the SNLVIE-2 in Eq. 1 as follows:

$$\begin{aligned} &x_0 = 0, \quad x_r = x_0 + r*h \\ &\varnothing_i(x_0) = f_i(x_0) \\ &\varnothing_i(x_r) = f_i(x_r) + \varnothing_i(x_{r-1}) + h*K_i(x_r, t_{r-1}, \varnothing_i(t_{r-1})) \end{aligned} \tag{3}$$

where, r = 1, ..., n.

Algorithm; Nonlinear Volterra Integral Equations via. BDF (NLVIEBDF):

Step 1: 1- Assume $h = \frac{b-a}{n}, n \in N$

2- Set $\varnothing_{i,0} = f_{i,0}$

Step 2: Compute $\varnothing_{i,r}; r = 1, 2, ..., N$ using Eq. (3)

Explicit-implicit method: Explicit-implicit method to evaluate an approximate solution of the SNLVIE-2 in the Eq. 1. This method is written as the following:

$$x_0 = 0, \quad x_r = x_0 + r*h$$
$$\varnothing_i(x_0) = f_i(x_0)$$
$$\varnothing_i(x_r) = f_i(x_r) + \varnothing_i(x_{r-1}) + h*K_i(x_r, t_{r-1}, \varnothing_i(t_{r-1}))$$

where, r = 1, ..., n (4)

$$\varnothing_i(x_r) = f_i(x_r) + \varnothing_i(x_{r-1}) + h*K_i(x_r, t_r, \varnothing_i(t_r)) \tag{5}$$

where, r = 1, ..., n.

Table 1: Comparison between the exact solution φ_1 (x) and the numerical solution via. applying (NLVIEEIM) algorithm

x	Exact $\varphi_1(x)$	Numerical $\varphi_1(x)$	Error $\varphi_1(x)$
0.0	1.000000000000000	1.000000000000000	0.000000000000000
0.1	0.995004165278026	1.000004165278026	0.005000000000000
0.2	0.980066577841242	0.980089361398244	0.000022783557002
0.3	0.955336489125606	0.950321052704141	0.005015436421465
0.4	0.921060994002885	0.910883201436098	0.010177792566787
0.5	0.877582561890373	0.861951220827917	0.015631341062456
0.6	0.8253356149096678	0.803796770968461	0.021538843941217
0.7	0.764842187284488	0.736763736670047	0.028078450614441
0.8	0.696706709347165	0.661267154797583	0.035439554549582
0.9	0.621609968270664	0.57779395635212	0.043816011920452
1.0	0.540302305868140	0.486905946301797	0.053396359566343
LSE			0.007677378735987

Therefore, we approximate the solution of the SNLVIE-2 in the Eq. 1 using the Eq. 4 and 5 in the following algorithm (NLVIEIEM):

Algorithm (NLVIEEIM)

Step 1: 1- Set $h = \frac{b-a}{n}, x_i = x_0 + ih, i = 1, 2, ...N, x_0 = a, x_{N+1} = b$

Step 2: Set $\varnothing_{i,0} = f_{i,0}$

Step 3: for i = 1, 2, …, M do step 4-6

Step 4: for i = 1, 2, …, N do step 5-6

Step 5: Compute $\varnothing_{i,r}$ using Eq. 4

Step 6: Compute $\varnothing_{i,r}$ using Eq. 5

Numerical example: In this study, we exam our algorithm (described in Section 4) in the following examples. The examples have proposed in order to show the differences between the exact and numerical solutions, respectively. The conclusions will be drawn in the next section.

Example 1; Consider the following SNLVILE-2:

$$\varphi_1(x) = x - \left(\frac{2}{3}\right)x^3 + \int_0^x \left(\varphi_1^2(t) + \varphi_2(t)\right)dt$$
$$\varphi_2(x) = x^2 - \frac{1}{4}x^4 + \int_0^x \left(\varphi_1(t) * \varphi_2(t)\right)dt$$

which has the exact solution: $(\varphi_1(x), \varphi_2(x)) = (x, x^2)$. We reported the experimental results of $\varphi_1(x)$ and $\varphi_2(x)$ in Table 1 and 2, respectively. Specifically, Table 1 shows the comparison between the exact solution and the numerical solution via. NLVIEEIM algorithm. The results depend on the Least Square Error (LSE) method which defined as:

$$LSE = \sum_{i=1}^{N} (\text{Exact solution} - \text{Numerical solution})^2$$

when:
$h = 0.1, x = x_i + ih, N = 10$ and $i = 0, 1, ..., 10$

Table 2: Comparison between the exact solution $\varphi_2(x)$ and the numerical solution via. applying (NLVIEEIM) algorithm

x	Exact $\varphi_2(x)$	Numerical $\varphi_2(x)$	Error $\varphi_2(x)$
0.0	0.000000000000000	0.000000000000000	0.000000000000000
0.1	0.010000000000000	0.009975000000000	0.000025000000000
0.2	0.040000000000000	0.039798466546079	0.000201533453903
0.3	0.090000000000000	0.089641189331539	0.000358810668461
0.4	0.016000000000000	0.159691688712280	0.000308311287720
0.5	0.2500000000000000	0.249987677484052	0.000012322515948
0.6	0.360000000000000	0.360336516120571	0.000336516120571
0.7	0.490000000000000	0.490220183582245	0.000220183582245
0.8	0.640000000000000	0.638679678719897	0.001320321280103
0.9	0.810000000000000	0.804174768080632	0.00582523191368
1.0	1.0000000000000000	0.984417259107345	0.015582740892655
LSE			2.7892530635 e-04

Table 3: Comparison between the exact solution $\varphi_1(x)$ and the numerical solution via. applying (NLVIEEIM) algorithm

x	Exact $\varphi_1(x)$	Numerical $\varphi_1(x)$	Error $\varphi_1(x)$
0.0	1.000000000000000	1.000000000000000	0.000000000000000
0.1	0.995004165278026	1.000004165278026	0.005000000000000
0.2	0.980066577841242	0.980089361398244	0.000022783557002
0.3	0.955336489125606	0.950321052704141	0.005015436421465
0.4	0.921060994002885	0.910883201436098	0.010177792566787
0.5	0.877582561890373	0.861951220827917	0.015631341062456
0.6	0.825335614909678	0.803796770968461	0.021538843941217
0.7	0.76484218728448	0.736763736670047	0.028078450614441
0.8	0.696706709347165	0.661267154797583	0.035439554549582
0.9	0.621609968270664	0.57779395635212	0.043816011920452
1.0	0.5403023305868140	0.486905946301797	0.053396359566343
LSE			0.007677378735987

Table 4: Comparison between the exact solution $\varphi_2(x)$ and the numerical solution via. applying (NLVIEEIM) algorithm

x	Exact $\varphi_2(x)$	Numerical $\varphi_2(x)$	Error $\varphi_2(x)$
0.0	0.000000000000000	0.000000000000000	0.000000000000000
0.1	0.099833416646828	0.104850061107139	0.005016644460310
0.2	0.198669330795061	0.198553747462216	0.000115583332845
0.3	0.295520206661340	0.289898581666503	0.005621624994837
0.4	0.389418342308651	0.377927759041815	0.011490583266836
0.5	0.479425538604203	0.461684365348749	0.017741173255454
0.6	0.564642473395035	0.540309513862413	0.024332959532623
0.7	0.644217687237691	0.613032323084653	0.031185364153038
0.8	0.717356090899523	0.679176265280304	0.038179825619219
0.9	0.783326909627483	0.738164129386050	0.045162780241433
1.0	0.841470984807897	0.789521871497710	0.051949113310187
LSE			0.008264271515161

Example 2; Consider the following SNLVIE-2:

$$\varphi_1(x) = \cos(x) - \frac{1}{2}x^2 + \int_0^x (x-t)\left(\varphi_1^2(t) + \varphi_2^2(t)\right)dt$$

$$\varphi_2(x) = \sin(x) - \frac{1}{2}\sin^2(x) + \int_0^x (x-t)\left(\varphi_1^2(t) - \varphi_2^2(t)\right)dt$$

which has the exact solution $((\varphi_1(x), \varphi_2(x)) = (\cos(x), \sin(x))$. Table 3 and 4 list the experimental results of $\varphi_1(x)$ and $\varphi_2(x)$, respectively. Similar to example 1, they report the comparison between the exact solution and the numerical solution via. NLVIEEIM algorithm. The results depend on the (LSE) as defined in the example 1. Here, also, h = 0.1, $x = x_i+ih$, N = 10 and i = 0, 1, ..., 10

Table 5: Comparison between the exact solution $\varphi_1(x)$ and the numerical solution via. applying (NLVIEEIM) algorithm

x	Exact $\varphi_1(x)$	Numerical $\varphi_1(x)$	Error $\varphi_1(x)$
0.0	1.000000000000000	1.000000000000000	0.000000000000000
0.1	1.105170918075648	1.104502916805101	0.000668001270547
0.2	1.221402758160170	1.220047909666774	0.001354848493396
0.3	1.349858807576003	1.347857394302066	0.002001413273937
0.4	1.491824697641270	1.489278540636438	0.002546157004833
0.5	1.648721270700128	1.645803531778705	0.002917738921423
0.6	1.822118800390509	1.819091670701534	0.003027129688975
0.7	2.013752707470477	2.010989014708243	0.002763692762234
0.8	2.225540928492468	2.223533473956186	0.002007454536282
0.9	2.459603111156950	2.458902171599545	0.000700939557405
1.0	2.718281828459040	2.719123226164825	0.000841397705780
LSE			4.331426305301807 e-05

Table 6: Comparison between the exact solution $\varphi_2(x)$ and the numerical solution via. applying (NLVIEEIM) algorithm

x	Exact $\varphi_2(x)$	Numerical $\varphi_2(x)$	Error $\varphi_2(x)$
0.0	1.000000000000000	1.000000000000000	0.000000000000000
0.1	0.904837418035960	0.915352371280369	-0.010514953244409
0.2	0.818730753077982	0.821044555673211	-0.002313802595229
0.3	0.740818220681718	0.746721041683056	-0.005902821001339
0.4	0.670320046035639	0.682421783077179	-0.012101737041540
0.5	0.606530659712633	0.608952743261895	-0.002422083549262
0.6	0.548811636094026	0.548768582785854	0.000043053308173
0.7	0.496585303791409	0.498273364201384	-0.001688060409974
0.8	0.449328964117222	0.444464294804639	0.004864669312583
0.9	0.406569659740599	0.406465678765523	0.000103980975076
1.0	0.367879441171442	0.364927077691805	0.002952363479638
LSE			3.383234193127457 e-04

Example 3; Consider the following SNLVIE-2:

$$\varphi_1(x) = e^x + x - \left(\frac{1}{2}\right)\sinh(2x) + \int_0^x (x-t)\left(\varphi_1^2(t) - \varphi_2^2(t)\right)dt$$

$$\varphi_2(x) = e^{-x} + x - xe^x + \int_0^x x\left(\varphi_1^2(t)\varphi(t)\right)dt$$

which has the exact solution: $((\varphi_1(x), \varphi_2(x)) = (e^x, e^{-x})$ similar to before, Table 5 and 6 show the results of $\varphi_1(x)$ and $\varphi_2(x)$, respectively. They list the comparison between the exact solution and the numerical solution via. NLVIEEIM algorithm. Additionally, the results depend on the (L.S.E.) defined in the example 1. Additionally, h = 0.1, $x = x_i+ih$, N = 10 and i = 0, 1, ..., 10

CONCLUSION

In this study proposed a new algorithm called NLVIEEIM have been proposed for solving SNLVIE-2. For this purpose, the algorithm used Gear Explicit-BDF-first order implicit method. The results demonstrated a marked improvement in the (LSE). Furthermore, the examples showed that the gear explicit-BDF-first order implicit method showed better accuracy in the solution of the SNLVIE-2.

REFERENCES

Epperson, J.F., 2013. An Introduction to Numerical Methods and Analysis. 2nd Edn., John Wiley & Sons, New Yrok, USA., ISBN:978-1-118-36759-9, Pages: 614.

Fausett, L.V., 2008. Applied Numerical Analysis Using MATLAB. Pearson, Upper Saddle River, New Jersey, USA., ISBN:9780132397285, Pages: 672.

Jumaa, B.F. and A.H. Taqi, 2016. Non-polynomial spline method for the solution of the system of two nonlinear Volterra integral equations. Kirkuk Univ. J. Sci. Stud., 11: 15-25.

Jumaa, B.F., 2017. Solving the system of two nonlinear Voltera intehral equations of the second kind using the trapezoidal predictor-corrector method. Br. J. Adv. Math. Comput. Sci., 20: 1-10.

Linz, P., 1985. Analytical and Numerical Methods for Volterra Equations. Vol. 7, SIAM, Philadelphia, Pennsylvania, USA., ISBN:9781611970852, Pages: 227.

Maturi, D.A., 2014. Numerical solution of system of two nonlinear Volterra integral equations. Intl. J. Comput. Technol., 12: 3967-3975.

Rice, J.R., 2014. Numerical Methods in Software and Analysis. 2nd Edn., Academic Press, Cambridge, Massachusetts, USA., ISBN:9781483295688, Pages: 721.

Stoer, J. and R. Bulirsch, 2013. Introduction to Numerical Analysis. 2nd Edn., Springer, New York, USA., ISBN:9781475722727, Pages: 660.

Taqi, A.H. and B.F. Jumaa, 2009. Using predictor-corrector methods for numerical solution of system of non-linear Volterra integral equations of second kind. AL. Raf. J. Comput. Sci. Math., 6: 93-109.

Wazwaz, A.M., 2011. Linear and Nonlinear Integral Equations: Methods and Applications. Springer, Heidelberg, Germany, ISBN:9783642214493, Pages: 638.