A new algorithm used Chebyshev pseudospectral method to solve nonlinear second-order Lienard differential equations

L.A. Nhat[1,3], D.S. Kulyabov[1,2], K.P. Lovetskiy[1]

[1]Department of Applied Probability and Informatics of RUDN University, Miklukho-Maklaya str. 6, Moscow, Russia, 117198
[2]Laboratory of Information Technologies Joint Institute for Nuclear Research, Joliot-Curie 6, Dubna, Moscow region, Russia, 141980
[3]Tan Trao University, Tuyen Quang, Vietnam, 22227

Abstract. This article presents a numerical method to determine the approximate solutions of the Lienard equations. It is assumed that the second-order nonlinear Linard differential equations of types $u''(x) + f[u(x)]u'(x) + g[u(x)] = 0$ on the range $[-1, 1]$ with the given boundary values $u[-1]$ and $u[+1]$. We have to build a new algorithm to find the approximate solutions to this problem. This algorithm is based on the pseudospectral method used in the Chebyshev differentiation matrix. In this paper, we used the Mathematica version 10.4 to represent the algorithm, the numerical results and graphics. In the numerical results, we made a comparison between the CPMs numerical results and the Mathematica's numerical results. The biggest odds were very small. Therefore, they will be able to be applied to other nonlinear systems such as the Rayleigh equations and the Emden-fowler equations.

1. Introduction

Lienard equations are applied in mathematics, mechanics, and physics. The general form of the second-order nonlinear Lienard differential equations is as follows

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}u(x) + f\left[u(x)\right]\frac{\mathrm{d}}{\mathrm{d}x}u(x) + g\left[u(x)\right] = 0, \quad -1 \leqslant x \leqslant 1, \quad u[-1] = \alpha, u[+1] = \beta, \tag{1}$$

here, $f \neq 0$ and $g \neq 0$ are the differentiable functions of $u(x)$; the boundary values α and β are given.

The Lienard equations are usually presented in the class autonomous equations, they have been dealt in many places [1–10]. Inside, several approaches have been studied so far dealing with the nonlinear second-order Lienard differential equations such as: the block pulse functions and their operational matrices of integration and differentiation are used to solve the Lienard equation in a large interval [4]; the residual power series method is implemented to find an approximate solution to the Lienard equation, here the author combined the fractional Taylor series and the residual functions [5]; the hybrid heuristic computing technique, stochastic in nature, is used for obtaining an approximate numerical solution of the Lienard equation [6]; the differential transform method based on the Taylor series expansion which constructs an analytical solution in the form of a polynomial to solve the Lienard equation [7]; in the Tiberiu's paper [8],

the first step, the second-order Lienard type equation is transformed into a second kind Abel type first order differential equation. The next, with the use of an exact integrability condition for the Abel equation, the exact general solution of the Abel equation can be obtained, thus leading to a class of exact solutions of the Lienard equation, expressed in a parametric form; The G'/G)–expansion method determined the exact solutions of Lienard equation [9]; the variational homotopy perturbation method determined the exact and numerical solutions for the Lienards equation [10], and others.
In this paper, we study, built a new algorithm based on the pseudo-spectral method used in the Chebyshev differentiation matrix to solve the second-order nonlinear Lienard differential equations.

2. Chebyshev differentiation matrix (CDM)

Let $h(x)$ – a polynomial of degree n have these polynomial values at $n+1$ points $x_0, x_1, ..., x_n$ are $h(x_i)$, $i = \overline{1, n}$; therefore, at these $n+1$ points, the values of the derivatives of $h'(x) = \frac{\mathrm{d}}{\mathrm{d}x}h(x)$ are determined. Each derivative can be expressed as a fixed linear combination of the given values of the function and the entire relation. Likewise, for the relationships for second derivatives $h''(x) = \frac{\mathrm{d}^2}{\mathrm{d}x^2}h(x)$. We can thus write in the matrix form

$$\begin{pmatrix} h'(x_0) \\ h'(x_1) \\ \vdots \\ h'(x_n) \end{pmatrix} = \widehat{D} \begin{pmatrix} h(x_0) \\ h(x_1) \\ \vdots \\ h(x_n) \end{pmatrix}, \quad \begin{pmatrix} h''(x_0) \\ h''(x_1) \\ \vdots \\ h''(x_n) \end{pmatrix} = \widehat{D}^2 \begin{pmatrix} h(x_0) \\ h(x_1) \\ \vdots \\ h(x_n) \end{pmatrix}, \tag{2}$$

where $\widehat{D} = \{d_{i,j}\}$, $i, j = \overline{1, n}$ is the so-called differentiation matrix.
For the Chebyshev-Gauss-Lobatto points, there are $n+1$ points $x_k = cos(k\pi/n)$ on the range $[-1, 1]$ of the Chebyshev polynomial $T_n(x)$. The elements of the differential matrix are calculated by the following formulae [11–15]

$$\begin{aligned} d_{0,0} &= -d_{n,n} = \frac{n^2}{3} + \frac{1}{6}, \\ d_{i,i} &= -\frac{cos(\frac{i\pi}{n})}{2sin^2(\frac{i\pi}{n})}, \quad i = 1, 2, .., n-1, \\ d_{i,j} &= \frac{c_i}{2c_j} \frac{(-1)^{i+j}}{sin\left(\frac{i+j}{2n}\pi\right) sin\left(\frac{j-i}{2n}\pi\right)}, \quad i \neq j, \end{aligned} \tag{3}$$

here

$$c_k = \begin{cases} 2, & k = 0 \text{ or } n \\ 1, & \text{otherwise} \end{cases}$$

3. Algorithm use CDM for the nonlinear Lienard differential equations

Suppose that

$$\frac{\mathrm{d}}{\mathrm{d}x}u(x) = f(x), \quad x \in [-1, 1], \quad u(-1) = \alpha, u(1) = \beta, \tag{4}$$

and the collocation points $\{x_i\}$ so that $-1 = x_n < x_{n-1} < ... < x_1 < x_0 = 1$.
We know that

$$\frac{\mathrm{d}}{\mathrm{d}x}u_n(x_i) = \sum_{k=0}^{n} \widehat{D}_{i,k} u_n(x_k). \tag{5}$$

So equation (4) becomes

$$\sum_{k=0}^{n} \widehat{D}_{i,k} u_n(x_k) = f(x_i), \quad i = \overline{1, n-1}, \quad u_n(x_n) = \alpha, u_n(x_0) = \beta, \tag{6}$$

Alternately, we partition the matrix $\widehat{D}$ into matrices [11]

$$e_0^{(1)} = \begin{pmatrix} d_{1,0} \\ d_{2,0} \\ \vdots \\ d_{n-1,0} \end{pmatrix}, E^{(1)} = \begin{pmatrix} d_{1,1} & d_{1,2} & \cdots & d_{1,n-1} \\ d_{2,1} & d_{2,2} & \cdots & d_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n-1,1} & d_{n-1,2} & \cdots & d_{n-1,n-1} \end{pmatrix}, e_n^{(1)} = \begin{pmatrix} d_{1,n-1} \\ d_{2,n-1} \\ \vdots \\ d_{n-1,n-1} \end{pmatrix} \tag{7}$$

we can rewrite $e_0^{(1)} = \{d_{i,0}\}$, $E^{(1)} = \{d_{i,j}\}$, $e_n^{(1)} = \{d_{i,n-1}\}$; here, $i, j = \overline{1, n}$ [16].
Thus, (6) can then be rewritten in the form matrix

$$u_n(x_0)e_0^{(1)} + E^{(1)}u + u_n(x_n)e_n^{(1)} = f \tag{8}$$

where u and f denote the vector

$$u = \begin{pmatrix} u_n(x_1) \\ \vdots \\ u_n(x_{n-1}) \end{pmatrix}, f = \begin{pmatrix} f_n(x_1) \\ \vdots \\ f_n(x_{n-1}) \end{pmatrix}.$$

Similarly with matrix $\widehat{D}^2$, we partition into matrices $e_0^{(2)}$, $E^{(2)}$, $e_n^{(2)}$. Furthermore, we have

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}u(x) = \frac{\mathrm{d}^2}{\mathrm{d}x^2}u_n(x_i) = \sum_{k=0}^{n} \widehat{D}^2_{i,k} u_n(x_k) = u_n(x_0)e_0^{(2)} + E^{(2)}u + u_n(x_n)e_n^{(2)}. \tag{9}$$

Now, we consider the nonlinear second-order Lienard differential equations (1). We have rewritten this equation in the general form

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}u(x) + f\left[u(x)\right]\frac{\mathrm{d}}{\mathrm{d}x}u(x) + \frac{g\left[u(x)\right]}{u(x)}u(x) = 0, \quad -1 \leqslant x \leqslant 1, \quad u[-1] = \alpha, u[+1] = \beta \tag{10}$$

From (8) and (9), we can rewrite (10) in the matrix form as

$$\left[E^{(2)} + F(u)E^{(1)} + G(u)\right]u + \beta\left(e_0^{(2)} + F(u)e_0^{(1)}\right) + \alpha\left(e_n^{(2)} + F(u)e_n^{(1)}\right) \tag{11}$$

where $F(u)$ and $G(u)$ denotes the square matrices order $(n-1) \times (n-1)$.How to determine $F(u)$ and $G(u)$: We know that u denotes the vector. Moreover, $F(u)$ and $G(u)$ denote the square matrices. So, $F(u)$ and $G(u)$ will denote the diagonal matrices with
elements $f\left[u(x_i)\right]$, and $\frac{g[u(x_i)]}{u(x_i)}$ with $i = \overline{1, n-1}$; the following cases can happen:

- If $F(u) = \delta$ is constant, then $F(u) = \delta I$; here, I is the unit matrix of order $(n-1)$;
- If $F(u) = \delta + \gamma u^m$, $m \in \mathbb{Q}$ then $F(u) = \delta I + \gamma \begin{pmatrix} u^m(x_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & 0 & u^m(x_{n-1}) \end{pmatrix}$;

this is similar to $G(u)$.
To find the solution $u_n(x_i)$, we give the following algorithm:

Algorithm

Set: $u^{(old)} := J^T$; $\varepsilon := 1$; $\varsigma := 10^{-8}$;
While $\varepsilon > \varsigma$ **do**
$F := F(u^{(old)})$;
$G := G(u^{(old)})$;
$M := E^{(2)} + F.E^{(1)} + G$;
$u^{(new)} := M^{-1}\left[-\beta\left(e_0^{(2)} + Fe_0^{(1)}\right) - \alpha\left(e_n^{(2)} + Fe_0^{(1)}\right)\right]$;
$\varepsilon := \left|Min\left\{u_1^{(new)} - u_1^{(old)}, u_2^{(new)} - u_2^{(old)}, ..., u_{n-1}^{(new)} - u_{n-1}^{(old)}\right\}\right|$;
$u^{(old)} := u^{(new)}$;
End while;
Return $u^{(old)}$;

Here, J is a unit vector.

Remasks: to increase the accuracy of $u_n(x_i)$, we can change the error ς of the program; the matrices $F\left(u^{(old)}\right)$ and $G\left(u^{(old)}\right)$ are recalculated after each loop.

4. Applications

In this section, we use the programming language Mathematica 10.4 to represent the algorithm used in CDM. Furthermore, we have used the function *NDSolve* to compute numerical results at the column **NDSolve** in each the example for comparison [17].

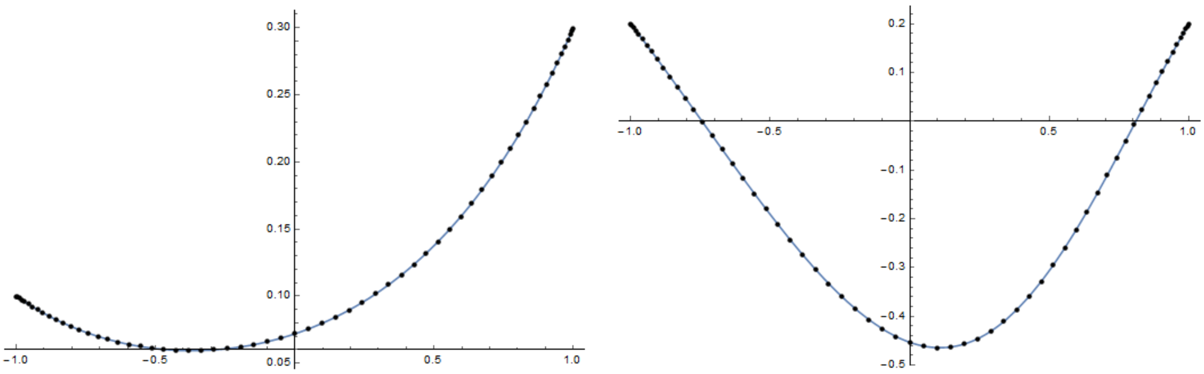
Example 1. Consider the nonlinear Lienard equation:

$$u''(x) + au(x)u'(x) + (bu^2(x) + c)u(x) = 0, x \in [-1, 1], u[-1] = \alpha, u[1] = \beta,$$

here $a, b, c \in \mathbb{R}$ (problem 2.2.3-2 p. 324 in [1]).
With $n = 64$, $\varsigma = 10^{-8}$ Tab.1. show several numerical results in the two cases:

- The first case $a = 2$, $b = -5$, $c = -3$ and the boundary values $\alpha = 0.1$, $\beta = 0.3$;
- The first case $a = 2$, $b = 1$, $c = 4$ and the boundary values $\alpha = \beta = 0.2$;

and Figure 1 is the corresponding graphics.



a) Graphic of the first case b) Graphic of the second case

Figure 1. Graphics of example 1, here dots are the result of the algorithm and the lines are graphics computed of Mathematica 10.4.

Table 1. Numerical results of example 1 in the first case and the second case.

		The first case		The second case	
i	x_i	$u_n(x_i)$	NDSolve	$u_n(x_i)$	NDSolve
1	0.99879546	0.29943006	0.29943010	0.19882732	0.19882724
5	0.97003125	0.28613857	0.28613860	0.17033285	0.17033278
10	0.88192126	0.24901089	0.24901092	0.07836687	0.07836686
15	0.74095113	0.19953275	0.19953276	-0.07461391	-0.07461381
20	0.55557023	0.14976440	0.14976441	-0.25941161	-0.25941139
25	0.33688985	0.10849297	0.10849297	-0.41065117	-0.41065091
30	0.09801714	0.07975743	0.07975743	-0.46426124	-0.46426105
35	-0.14673047	0.06390680	0.06390680	-0.40670783	-0.40670775
40	-0.38268343	0.05943794	0.05943794	-0.27442251	-0.27442250
45	-0.59569930	0.06409530	0.06409531	-0.11792937	-0.11792938
50	-0.77301045	0.07486935	0.07486935	0.02329408	0.02329406
55	-0.90398929	0.08759392	0.08759392	0.12708425	0.12708423
60	-0.98078528	0.09728577	0.09728577	0.18572562	0.18572561
63	-0.99879546	0.09982627	0.09982627	0.19911056	0.19911056

Example 2. Consider the nonlinear Lienard equation:

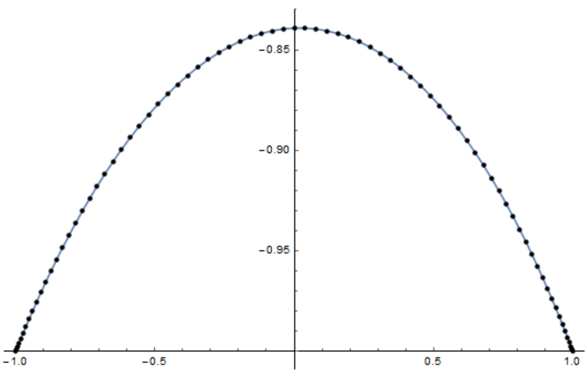
$$u''(x) + [au(x) + 3b]u'(x) + [2b^2 + abu(x) - cu^2(x)]u(x) = 0, x \in [-1, 1], u[-1] = \alpha, u[1] = \beta,$$

here $a, b, c \in \mathbb{R}$ (problem 2.2.3-3 p. 324 in [1]).

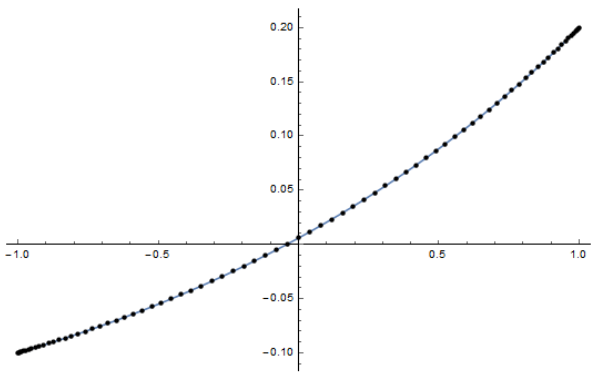
With $n = 80$, $\varsigma = 10^{-8}$ Tab.2. show several numerical results in the two cases:

- The first case $a = 0.2$, $b = 0.1$, $c = 0.5$ and the boundary values $\alpha = \beta = -1$;
- The first case $a = 0.5$, $b = 0.2$, $c = 0.3$ and the boundary values $\alpha = -0.1$, $\beta = 0.2$;

and Figure 2 is the corresponding graphics.



a) Graphic of the first case



b) Graphic of the second case

Figure 2. Graphics of example 2, here dots are the result of the algorithm and the lines are graphics computed of Mathematica 10.4.

Example 3. Consider the nonlinear Lienard equation:

$$u''(x) + a\sin[\lambda u(x)]u'(x) + b\sin[\lambda u(x)] = 0, \quad x \in [-1, 1], u[-1] = \alpha, u[1] = \beta,$$

here $a, b, \lambda \in \mathbb{R}$ (problem 2.2.3-19 p. 326 in [1]).

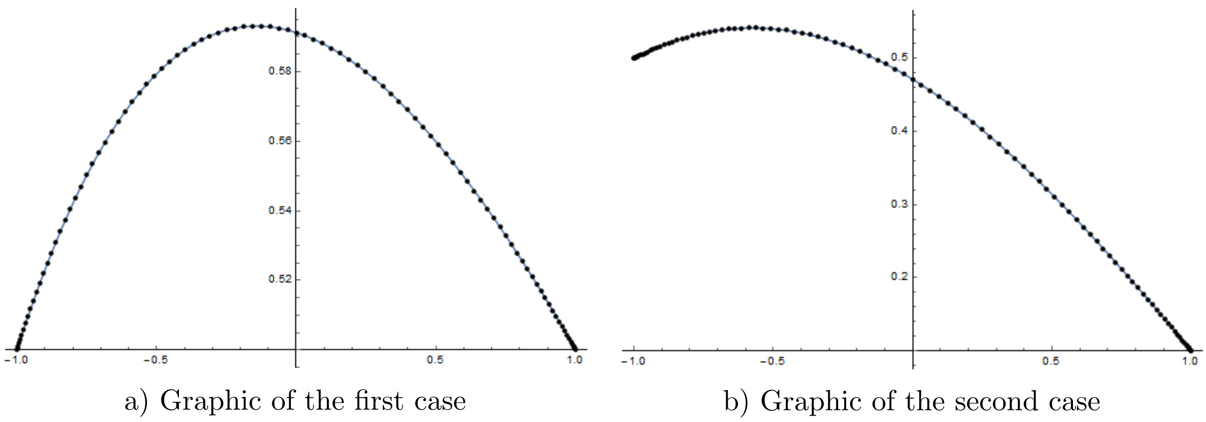
With $n = 100$, $\varsigma = 10^{-8}$ Tab.3. show several numerical results in the two cases:

Table 2. Numerical results of example 2 in the first case and the second case.

		The first case		The second case	
i	x_i	$u_n(x_i)$	NDSolve	$u_n(x_i)$	NDSolve
1	0.99922904	-0.99971806	-0.99971808	0.19979988	0.19979988
5	0.98078528	-0.99306774	-0.99306778	0.19504123	0.19504122
10	0.92387953	-0.97366093	-0.97366104	0.18070249	0.18070250
15	0.83146961	-0.94552646	-0.94552666	0.15847740	0.15847741
20	0.70710678	-0.91372105	-0.91372139	0.13050468	0.13050469
25	0.55557023	-0.88342877	-0.88342924	0.09917569	0.09917569
30	0.38268343	-0.85905388	-0.85905448	0.06677511	0.06677512
35	0.19509032	-0.84373476	-0.84373545	0.03522807	0.03522808
40	0	-0.83920250	-0.83920322	0.00597392	0.00597392
45	-0.19509032	-0.84582528	-0.84582596	-0.02004993	-0.02004993
50	-0.38268343	-0.86270156	-0.86270215	-0.04234803	-0.04234803
55	-0.55557023	-0.88773432	-0.88773479	-0.06076638	-0.06076639
60	-0.70710678	-0.91770271	-0.91770303	-0.07538008	-0.07538009
65	-0.83146961	-0.94843058	-0.94843077	-0.08638896	-0.08638897
70	-0.92387953	-0.97519840	-0.97519850	-0.09403151	-0.09403151
75	-0.98078528	-0.99349314	-0.99349316	-0.09852053	-0.09852054
79	-0.99922904	-0.99973563	-0.99973564	-0.09994099	-0.09994099

- The first case $a = 0.9$, $b = 0.2$, $\lambda = \pi$ and the boundary values $\alpha = \beta = 0.5$;
- The first case $a = 0.3$, $b = 0.6$, $\lambda = \pi/2$ and the boundary values $\alpha = 0.5$, $\beta = 0.1$;

and Figure 3 is the corresponding graphics.



a) Graphic of the first case b) Graphic of the second case

Figure 3. Graphics of example 3, here dots are the result of the algorithm and the lines are graphics computed of Mathematica 10.4.

Alternately, from the programs, we also have other results: number of loops to find the solution $u_n(x_i)$ of the algorithm; the biggest odds between two columns $u_n(x_i)$ and **NDSolve**. All these results are shown in the Table 4.

5. Conclusions

In this work, we have investigated a new algorithm to solve nonlinear Lienard equations based on the pseudo-spectral method using the Chebyshev differentiation matrix. From tables 1-3, we see

Table 3. Numerical results of example 3 in the first case and the second case.

		The first case		The second case	
i	x_i	$u_n(x_i)$	NDSolve	$u_n(x_i)$	NDSolve
1	0.99950656	0.50006953	0.50006954	0.10022698	0.10022698
5	0.98768834	0.50172952	0.50172951	0.10565786	0.10565785
10	0.95105652	0.50680870	0.50680863	0.12242123	0.12242122
15	0.89100652	0.51491144	0.51491128	0.14963676	0.14963673
20	0.80901699	0.52550202	0.52550172	0.18615131	0.18615127
25	0.70710678	0.53784935	0.53784887	0.23025423	0.23025416
30	0.58778525	0.55105204	0.55105135	0.27966240	0.27966231
35	0.45399050	0.56407834	0.56407741	0.33159060	0.33159048
40	0.30901699	0.57582422	0.57582305	0.38293136	0.38293122
45	0.15643447	0.58519121	0.58518982	0.43053930	0.43053915
50	0	0.59118240	0.59118084	0.47158118	0.47158102
55	-0.15643447	0.59301190	0.59301026	0.50388839	0.50388824
60	-0.30901699	0.59022036	0.59021874	0.52624121	0.52624107
65	-0.45399050	0.58278554	0.58278405	0.53852521	0.53852509
70	-0.58778525	0.57121108	0.57120981	0.54172602	0.54172592
75	-0.70710678	0.55656685	0.55656586	0.53775978	0.53775969
80	-0.80901699	0.54044682	0.54044613	0.52916891	0.52916886
85	-0.89100652	0.52481517	0.52481475	0.51873865	0.51873863
90	-0.95105652	0.51173861	0.51173841	0.50910206	0.50910204
95	-0.98768834	0.50304693	0.50304689	0.50239491	0.50239490
99	-0.99950656	0.50012335	0.50012335	0.50009735	0.50009734

Table 4. Several other results.

Example	Loop	The biggest odds
The first case of exmaple 1	5	3.84138×10^{-8}
The second case of exmaple 1	16	2.61979×10^{-7}
The first case of exmaple 2	9	7.12725×10^{-7}
The second case of exmaple 2	6	1.55962×10^{-8}
The first case of exmaple 3	8	1.64654×10^{-6}
The second case of exmaple 3	8	1.61575×10^{-7}

that the numerical results of two columns $u_n(x_i)$ and *NDSolve* are equivalent, the biggest odds between two columns $u_n(x_i)$ and *NDSolve* in all three examples is 1.64654×10^{-6}; Repeatability to find the solution $u_n(x_i)$ is low (see table 4). So, this new algorithm is reliable to solve the nonlinear Lienard equations class.

6. References

[1] Andrei, D.P. Handbook of exact solutions for ordinary differential equations / D.P. Andrei, F.Z. Valentin // Chapman & Hall/CRC, 2003. – 802 p.

[2] Jordan, D.W. Nonlinear Ordinary Differential Equations: An introduction for Scientists and Engineers / D.W. Jordan, P. Smith. – Oxford, 2007. – 540 p.

[3] Sachdev, P. L. Nonlinear Ordinary differential equations and their applications // Marcel Dekker, Inc., 1991. – 578 p.

[4] Heydari, M.H. A Good Approximate Solution for Lienard Equation in a Large Interval Using Block Pulse Functions / M.H. Heydari, M.R. Hooshmandasl, F.M. Maalek Ghaini // Journal of Mathematical Extension. – 2013. – Vol. 7(1). – P. 17-32.
[5] Muhammed, I.S. A Numerical Solution of Fractional Lienards Equation by Using the Residual Power Series Method // Mathematics. – 2018. – Vol. 6(1). – P. 9.
[6] Suheel, A.M. Numerical Solution of Lienard Equation Using Hybrid Heuristic Computation / A.M. Suheel, M.Q. Ijaz, A. Muhammad, H. Ihsanul // World Applied Sciences Journal. – 2013. – Vol. 28(5). – P. 636-643.
[7] Mashallah, M. Solving the Lienard equation by differential transform method / M. Mashallah, R.B. Saber, G. Maryam // World Journal of Modelling and Simulation. – 2012. – Vol. 8(2). – P. 142-146.
[8] Tiberiu, H. A class of exact solutions of the Lienard type ordinary non-linear differential equation / H. Tiberiu, S.N. Lobo, M.K. Mak // J. Eng. Math. – 2014. – Vol. 89. – P. 193-205.
[9] Salehpour, E. Application of (G'/G)-expansion method to nonlinear Lienard equation / E. Salehpour, H. Jafari, N. Kadkhoda // Indian J. Sci. Technol. – 2012. – Vol. 5(4). – P. 2454-2456.
[10] Matinfar, M. Exact and Numerical Solution of Lienard's Equation by the Variational Homotopy Perturbation Method / M. Matinfar, M. Mahdavi, Z. Raeisy // J. Inf. Comput. Sci. – 2011. – Vol. 6(1). – P. 73-80.
[11] Mason, J.C. Chebyshev Polynomials / J.C. Mason, D.C. Handscomb. – CRC Press LLC, 2003. – 335 p.
[12] Trefethen, L.N. Spectral Methods in Matlab. – Oxford: SIAM, 2000. – 181 p.
[13] Don, W.S. Accuracy and speed in computing the Chebyshev collocation derivative / W.S. Don, A. Solomonoff // SISC. – 1991. – Vol. 16(6). – P. 1253-1268.
[14] Tinuade, O. Application of the Chebyshev pseudospectral method to van der Waals fluids / O. Tinuade, M. Abdolmajid, S. Ousmane // Commun. Nonlinear Sci. Numer. Simulat. – 2012. – Vol. 17. – P. 3499-3507.
[15] Arne, J. Lecture Notes on Spectra and Pseudospectra of Matrices and Operators //Aalborg University, 2009. – 126 p.
[16] Nhat, L.A. Using differentiation matrices for pseudospectral method solve Duffing Oscillator // J. Nonlinear Sci. Appl. – 2018. – Vol. 11(9). – P. 1331-1336.
[17] Martha, L. Braselton Differential Equations with Mathematica / L. Martha, J.P. Abell // Elsevier Inc, 2004. – 893 p.

Acknowledgments

The publication was prepared with the support of the *RUDN University Program 5-100*.