1. Introduction

The real world problems in scientific fields such as solid state physics, plasma physics, fluid mechanics, chemical kinetics and mathematical biology are nonlinear in general when formulated as differential equations [1]. Most of the nonlinear differential equations are complicated to be solved using analytical techniques, but these problems are tackled using approximate analytical or numerical methods. For example, various nonlinear differential equations have been solved using the Taylor matrix method, the closed-form method, Chebyshev polynomial approximations, the variational iteration method, the subdomain finite element method, the differential quadrature method, the homotopy perturbation method, the quantic B-spline differential quadrature method, the power series method, the Padé series method, the Legendre polynomial function approximation [2], predictor–corrector method [3], etc.

The Liénard equation is a nonlinear second order differential equation proposed by Liénard [4] and is presented as:

\[ u''(x) + p(u)u'(x) + q(u) = g(x) \] (1)

subject to the initial conditions \( u(x_0) = \alpha, u'(x_0) = \beta \).

Equation (1) is not only regarded as a generalization of the damped pendulum equation or a damped spring–mass system (where \( p(u)u'(x) \) is the damping force, \( q(u) \) is the restoring force, and \( g(x) \) is the external force), but also used as nonlinear models in many physically significant fields when taking different choices for \( p(u), q(u) \) and \( g(x) \). For example, the choices \( p(u) = \varepsilon(u^2 - 1), q(u) = u \) and \( g(x) = 0 \) lead Equation (1) to the Van der Pol equation served as a nonlinear model of electronic oscillation [5,6]. Moreover, in the development of radio and vacuum tube technology, Liénard equations were remarkably investigated as they can be utilized to describe the oscillating circuits [7]. The Liénard-type equations can also be used to model fluid mechanical phenomena [8].

In recent years, many researchers have tried to develop different numerical methods for solving Liénard differential equations. For examples, Equation (1) was studied by many authors in [9–17]. However, still, the accuracy of their solution needs improvement. Thus, this study introduces a sixth order predictor–corrector method which is stable, convergent and more accurate for solving Liénard second order ordinary differential equations.

2. Formulation of the method

Let \( v(x) = u'(x) \), then Equation (1) is changed into a first order system of equations:

\[
\begin{align*}
  u'(x) &= v(x) = f^{(1)}(x, u, v), \quad u(x_0) = \alpha \\
  v'(x) &= g(x) - p(u)v(x) - q(u) = f^{(2)}(x, u, v), \quad v(x_0) = \beta \\
\end{align*}
\] (2)

To describe the methods, let’s denote \( u^{(1)} = u \) and \( u^{(2)} = v \), then Equation (2) can be written as:

\[
\frac{du^{(1)}}{dx} = f^{(1)}(x, u^{(1)}, u^{(2)}), \text{ for } j = 1, 2
\] (3)

subject to \( u^{(1)}(x_0) = \alpha, \quad u^{(2)}(x_0) = \beta \).

Now, divide the interval \([x_0, L]\) into \(N\) equal subintervals of mesh length \(h\) and integrating Equation (3) on...
the interval $[x_i, x_{i+1})$, we obtain:

$$u^{(j)}(x_{i+1}) = u^{(j)}(x_i) + \int_{x_i}^{x_{i+1}} f^{(j)}(x, u^{(1)}, u^{(2)}) \, dx,$$

for $j = 1, 2$ \quad(4)

To formulate the methods, we approximate $f^{(j)}(x, u^{(1)}, u^{(2)})$, for $j = 1, 2$ by a suitable interpolation polynomials.

**Case 1:** Take $k$ data values $(x_i, f^{(j)}_i), (x_{i-1}, f^{(j)}_{i-1}), \ldots, (x_{i-k+1}, f^{(j)}_{i-k+1})$, for $j = 1, 2$. For this data, we fit Newton’s backward difference interpolating polynomial of degree $k-1$ and obtain:

$$f^{(j)}(x_i + sh) = \sum_{m=0}^{k-1} (-1)^m \binom{-s}{m} \sum_{m=0}^{k-1} \binom{-s}{m} \bar{v}^m f^{(j)}_{i+m} + t^{(j)}_{k} \quad(5)$$

where $s = ((x-x_i)/h)$ and

$$t^{(j)}_{k} = \frac{s(s+1)(s+2) \cdots (s+k-1)}{k!} h^k f^{(j)}(\xi) \quad(6)$$

is the error term, when $\xi$ lies in some interval containing the points $x_i, x_{i-1}, x_{i-k+1}$ and $x$.

Using Equations (5) and (6) in Equation (4), we get:

$$u^{(j)}_{i+1} = u^{(j)}_i + h \int_0^1 \left( \sum_{m=0}^{k-1} (-1)^m \binom{-s}{m} \bar{v}^m f^{(j)}_{i+m} + t^{(j)}_{k} \right) \, ds \quad(7)$$

Integrating term by term, Equation (7) with respect to $s$, choosing the value $k = 6$ and simplifying, we obtain:

$$u^{(j)}_{i+1} = u^{(j)}_i + h \left[ \frac{4277 f^{(j)}_{i+1} - 7923 f^{(j)}_{i+1} + 9982 f^{(j)}_{i-2} - 7298 f^{(j)}_{i-3} + 2877 f^{(j)}_{i-4} - 475 f^{(j)}_{i-5}}{1440} \right] + t^{(j)}_6 \quad(8)$$

where

$$t^{(j)}_6 = \frac{19087}{60480} h^7 (u^{(j)}_0) (\xi), \quad \text{for} \ j = 1, 2 \quad(9)$$

is the local truncation error. Hence, Equation (8) is called predictor method.

**Case 2:** Consider the $k + 1$ data values $(x_i, f^{(j)}_i), (x_i, f^{(j)}_{i-1}), \ldots, (x_{i-k+1}, f^{(j)}_{i-k+1})$, for $j = 1, 2$, which include the current data points. For this data, we fit Newton’s backward difference interpolating polynomial of degree $k$ and obtain:

$$f^{(j)}(x_i + sh) = \sum_{m=0}^{k} (-1)^m \binom{1-s}{m} \bar{v}^m f^{(j)}_{i+m} + t^{(j)}_{k} \quad(10)$$

where $s = ((x-x_i)/h), x - x_{i+1} = (x - x_i) - (x_{i+1} - x_i) = sh - h = h(s - 1)$ and

$$T_k = \frac{(s-1)(s+1)(s+2) \cdots (s+k-1)}{(k+1)!} h^{k+1} f^{(j)}(\xi) \quad(11)$$

is the error term, when $\xi$ lies in some interval containing the points $x_{i+1}, x_i, x_{i-1}, \ldots, x_{i-k+1}$ and $x$.

Using Equations (10) and (11) in Equation (4), we get:

$$u^{(j)}_{i+1} = u^{(j)}_i + h \int_0^1 \left( \sum_{m=0}^{k} (-1)^m \binom{1-s}{m} \bar{v}^m f^{(j)}_{i+m} + t^{(j)}_{k} \right) \, ds \quad(12)$$

Integrating term by term, Equation (12) with respect to $s$, choosing $k = 5$ and simplifying, we get:

$$u^{(j)}_{i+1} = u^{(j)}_i + \frac{h}{1440} \left[ 475 f^{(j)}_{i+1} + 1427 f^{(j)}_{i+1} - 798 f^{(j)}_{i-1} + 482 f^{(j)}_{i-2} - 173 f^{(j)}_{i-3} + 27 f^{(j)}_{i-4} \right] + t^{(j)}_5 \quad(13)$$

where $f^{(j)}_{i+1} = f^{(j)}(x_{i+1}, u^{(j)}_{i+1}, u^{(j)}_{i+1}, u^{(j)}_{i+1})$ and $u^{(j)}_{i+1}, u^{(j)}_{i+1}$ are values obtained from Equation (8) and

$$t^{(j)}_5 = -\frac{863}{60480} h^7 (u^{(j)}_0) (\xi), \quad \text{for} \ j = 1, 2 \quad(14)$$

is the local truncation error of a corrector method.

Thus, Liénard second order nonlinear differential equation in Equation (1) can be easily solved by the systems in Equations (8) and (13) using MATLAB software.

**Remark 2.1:** For using these methods, we require the starting values $u^{(j)}_0, u^{(j)}_1, u^{(j)}_2, u^{(j)}_3, u^{(j)}_4$, for $j = 1, 2$. So in this study, we used the classical Runge–Kutta method for the first five nodal points.

### 3. Stability and convergence analysis

**Definition 3.1:** Let $\lambda_1, \lambda_2, \ldots, \lambda_k$ and $\gamma_1, \gamma_2, \ldots, \gamma_k$ are respectively the (not necessarily distinct) roots of the characteristic equations given by:

$$p_1(\lambda) = \lambda^k - \alpha^{(1)}_{k-1}\lambda^{k-1} - \cdots - \alpha^{(1)}_1\lambda - \alpha^{(1)}_0 = 0,$$

for $j = 1, 2$

$$p_2(\gamma) = \gamma^k - a^{(2)}_{k-1}\gamma^{k-1} - \cdots - a^{(2)}_1\gamma - a^{(2)}_0 = 0,$$

for $j = 1, 2$

associated with the system of the multistep difference method of Equations (8) and (13) given as:

$$u^{(j)}_{i+1} = a^{(j)}_{k-1} u^{(j)}_i + a^{(j)}_{k-2} u^{(j)}_{i-1} + \cdots + a^{(j)}_0 u^{(j)}_{i-k} + hF(x_i, h, u^{(j)}_i, u^{(j)}_{i-1}, \ldots, u^{(j)}_{i-k}),$$

where $a^{(j)}_0, a^{(j)}_1, \ldots, a^{(j)}_{k-1}$ are constants, for each $i = k - 1, k, \ldots, N - 1$ and $j = 1, 2$.

If $|\lambda_i| \leq 1$, and $|\gamma_i| \leq 1$ for $i = 1, 2, \ldots, k$, and all roots with absolute value 1 are simple roots, then the system of the difference method is said to satisfy the root condition.

**Definition 3.2:** Method that satisfy the root condition and have $\lambda = 1$ as the only root of the characteristic
equation with magnitude one is called strongly stable [18].

**Theorem 3.1:** The scheme in Equation (8) is strongly stable.

**Proof:** The scheme in Equation (8) can be expressed as:

\[ u^{(j)}_{i+1} = u^{(j)}_i + hF(x_i, h, u^{(j)}_i, u^{(j)}_{i-1}, u^{(j)}_{i-2}, u^{(j)}_{i-3}, u^{(j)}_{i-4}, u^{(j)}_{i-5}), \]

where

\[
F(x_i, h, u^{(j)}_i, u^{(j)}_{i-1}, u^{(j)}_{i-2}, u^{(j)}_{i-3}, u^{(j)}_{i-4}, u^{(j)}_{i-5}) = \frac{1}{1440} (4277f_i(x_i, u^{(1)}_i, u^{(2)}_i) - 7923f_i(x_i, u^{(1)}_i, u^{(2)}_i) + 9982f_i(x_i, u^{(1)}_i, u^{(2)}_i) - 7298f_i(x_i, u^{(1)}_i, u^{(2)}_i) + 2877f_i(x_i, u^{(1)}_i, u^{(2)}_i) - 475f_i(x_i, u^{(1)}_i, u^{(2)}_i))
\]

In this case, \( k = 6 \) so by Definition 3.1, we have:

\[ a_0^{(j)} = a_1^{(j)} = a_2^{(j)} = a_3^{(j)} = a_4^{(j)} = 0, \quad \text{and} \quad a_5^{(j)} = 1, \quad \text{for} \quad j = 1, 2. \]

The characteristic equations for the method becomes:

\[ p_1(\lambda) = \lambda^5 - \lambda^4 = \lambda^5(\lambda - 1) = 0, \quad \text{for} \quad j = 1, \]

\[ p_2(\gamma) = \gamma^5 - \gamma^4 = \gamma^5(\gamma - 1) = 0, \quad \text{for} \quad j = 2. \]

which implies \( \lambda_i = \gamma_i = 1 \) and \( \lambda_i = \gamma_i = 0 \) for \( i = 2, 3, \ldots, 6 \) are the roots of the polynomials.

Therefore, the predictor method in Equation (8) satisfies the root condition and is strongly stable by Definition 3.2.

**Theorem 3.2:** The scheme in Equation (13) is also strongly stable.

**Proof:** The scheme in Equation (13) can be expressed as:

\[ u^{(j)}_{i+1} = u^{(j)}_i + hF(x_i, h, u^{(j)}_i, u^{(j)}_{i-1}, u^{(j)}_{i-2}, u^{(j)}_{i-3}, u^{(j)}_{i-4}, u^{(j)}_{i-5}), \]

where

\[
F(x_i, h, u^{(j)}_i, u^{(j)}_{i-1}, u^{(j)}_{i-2}, u^{(j)}_{i-3}, u^{(j)}_{i-4}, u^{(j)}_{i-5}) = \frac{h}{1440} \left[ 475f_i(x_i, u^{(1)}_i, u^{(2)}_i) + 1427f_i(x_i, u^{(1)}_i, u^{(2)}_i) - 798f_i(x_i, u^{(1)}_i, u^{(2)}_i) + 482f_i(x_i, u^{(1)}_i, u^{(2)}_i) - 173f_i(x_i, u^{(1)}_i, u^{(2)}_i) + 27f_i(x_i, u^{(1)}_i, u^{(2)}_i) \right]
\]

Following the similar procedure as we have done in Theorem 3.1, here, \( k = 5 \) which implies \( a_0^{(j)} = a_1^{(j)} = a_2^{(j)} = a_3^{(j)} = 0, \quad \text{and} \quad a_4^{(j)} = 1, \quad \text{for} \quad j = 1, 2. \]

The characteristic equations for the method become:

\[ p_1(\lambda) = \lambda^5 - \lambda^4 = \lambda^5(\lambda - 1) = 0, \quad \text{for} \quad j = 1, \]

\[ p_2(\gamma) = \gamma^5 - \gamma^4 = \gamma^5(\gamma - 1) = 0, \quad \text{for} \quad j = 2. \]

Thus, the polynomials have roots \( \lambda_i = \gamma_i = 1 \) and \( \lambda_i = \gamma_i = 0 \) for \( i = 2, 3, 4, 5. \)

Therefore, the corrector method in Equation (13) satisfies the root condition and is strongly stable by Definition 3.2.

**Definition 3.3 (Consistency):** The method is consistent if the local truncation error \( r_k(h) \to 0 \) as \( h \to 0. \)

**Lemma 3.1:** The schemes in Equations (8) and (13) are both sixth-order convergence.

**Proof:** From Equations (9) and (14), we have:

\[ r_5 = -\frac{863}{60480} h^7 \left( u^{(j)}(\xi) \right) \]

\[ r_6 = \frac{19087}{60480} h^7 \left( u^{(j)}(\xi) \right) \]

Thus, \( r_k(h) \to 0 \) as \( h \to 0 \) for \( k = 5, 6 \) and \( j = 1, 2. \)

---

**Table 1.** The pointwise absolute errors for Example 4.1 with \( x \in [0, 0.5]. \)

| \( x \) | Magnus series | Present method |
|--------|---------------|----------------|
| \( [0, 0.5] \) | \( h = 0.001 \) | \( h = 0.1 \) |
| 0.1    | 1.4409e-04    | 2.6794e-07  | 1.2975e-11 | 4.4409e-16 |
| 0.2    | 1.1547e-03    | 5.7177e-07  | 1.6940e-10 | 4.4409e-16 |
| 0.3    | 3.9018e-03    | 8.8764e-07  | 2.1276e-11 | 1.3332e-15 |
| 0.4    | 9.2132e-03    | 1.1954e-06  | 2.4793e-11 | 1.3332e-15 |
| 0.5    | 1.78199e-02   | 1.2849e-06  | 1.2747e-10 | 1.6653e-15 |

| \( x \) | \( h = 0.1 \) | \( h = 0.1 \) |
|--------|---------------|----------------|
| 0.1    | 1.4409e-04    | 2.6794e-07  | 1.2975e-11 | 4.4409e-16 |
| 0.2    | 1.1547e-03    | 5.7177e-07  | 1.6940e-10 | 4.4409e-16 |
| 0.3    | 3.9018e-03    | 8.8764e-07  | 2.1276e-11 | 1.3332e-15 |
| 0.4    | 9.2132e-03    | 1.1954e-06  | 2.4793e-11 | 1.3332e-15 |
| 0.5    | 1.78199e-02   | 1.2849e-06  | 1.2747e-10 | 1.6653e-15 |

**Table 2.** The maximum absolute errors for different values of \( h \) and \( x \in [0, 1]. \)

| \( h \) | \( x \) | \( h = 1/50 \) | \( h = 1/100 \) |
|--------|--------|----------------|-----------------|
| 6.633e-10 | 6.770e-10 | 2.6704e-11 | 1.3498e-12 |
| 6.706e-11 | 2.8566e-11 | 1.9988e-12 | 2.3725e-13 |

| \( h \) | \( x \) | \( h = 1/200 \) | \( h = 1/300 \) |
|--------|--------|----------------|-----------------|
| 6.633e-10 | 6.770e-10 | 2.6704e-11 | 1.3498e-12 |
| 6.706e-11 | 2.8566e-11 | 1.9988e-12 | 2.3725e-13 |

| \( h \) | \( x \) | \( h = 1/400 \) | \( h = 1/500 \) |
|--------|--------|----------------|-----------------|
| 6.633e-10 | 6.770e-10 | 2.6704e-11 | 1.3498e-12 |
| 6.706e-11 | 2.8566e-11 | 1.9988e-12 | 2.3725e-13 |
Therefore, the methods in Equations (8) and (13) are consistent and hence they are sixth-order convergence.

4. Numerical examples and Results

Three model examples of Liénard second order nonlinear differential equations with initial conditions have been considered to validate the applicability of the method. For each number of nodal points $N$, the pointwise absolute errors were approximated by the formula $|E| = |u(x_i) - u_i|$, for $i = 0, 1, 2, \ldots, N$, where $u(x_i)$ and $u_i$ are exact and computed an approximate solution of the given problems respectively, at the nodal point $x_i$. Numerical examples are presented to illustrate the reliability and efficiency of the method. Comparison of numerical results is made with a Magnus series expansion method [15] which is based on

Table 3. The pointwise absolute errors for Example 4.2 with $x \in [0, 1]$ and different values of $h$.

| Differential transform [16] | Hybrid heuristic computation [17] | Present method |
|-----------------------------|----------------------------------|----------------|
| $x$                         | $h = 0.1$                        | $h = 0.01$     | $h = 0.002$     |
| $0.1$                       | 2.2683e−06                      | 6.7141e−08     | 6.7141e−08     |
| $0.2$                       | 2.5222e−06                      | 1.4010e−07     | 1.4010e−07     |
| $0.3$                       | 1.4404e−05                      | 1.4865e−07     | 1.4865e−07     |
| $0.4$                       | 6.1806e−05                      | 1.4573e−07     | 1.4573e−07     |
| $0.5$                       | 2.0999e−04                      | 9.8898e−08     | 9.8898e−08     |
| $0.6$                       | 6.5187e−04                      | 9.9460e−12     | 9.9460e−12     |
| $0.7$                       | 1.6408e−03                      | 3.4195e−08     | 3.4195e−08     |
| $0.8$                       | 3.6469e−03                      | 1.3746e−08     | 1.3746e−08     |
| $0.9$                       | 7.3935e−03                      | 1.4832e−08     | 1.4832e−08     |
| $1.0$                       | 1.3692e−02                      | 2.9187e−08     | 2.9187e−08     |

Table 4. The maximum absolute errors for Example 4.2 for different values of $h$ and $x \in [0, 1]$.

| $h = 1/50$ | $h = 1/100$ | $h = 1/200$ | $h = 1/300$ | $h = 1/400$ | $h = 1/500$ |
|------------|------------|------------|------------|------------|------------|
| 3.5866e−10 | 1.0288e−11 | 3.9369e−13 | 6.5170e−14 | 1.8652e−14 | 7.6605e−15 |
| Refinement |            |            |            |            |            |
| 1.5083e−10 | 7.0087e−12 | 3.4439e−13 | 6.1284e−14 | 1.8208e−14 | 7.5495e−15 |
Lie groups and Lie algebras on some specific nonlinear differential equations, a differential transform method [16] based on the Taylor series expansion which constructs an analytical solution in the form of a polynomial and a hybrid heuristic computation [17] based technique as an alternative to the existing deterministic standard numerical methods, including variational iteration method and differential transform method for solving the Liénard differential equations numerically.

**Example 4.1:** Consider the following Liénard Equation:

\[
\dot{u}(x) + u(x)u'(x) + u(x) + u^2(x) = \cos^2(x) - \sin(x) \cos(x)
\]

with the initial conditions

\[
u(0) = 1, \ u'(0) = 0
\]

The exact solution of the given problem is given by \(u(x) = \cos(x)\). Numerical results are presented in Tables 1 and 2 and Figures 1 and 2.

Table 1 exhibits the pointwise absolute errors for different values of mesh size \(h\), and it can be seen that the present method improves the findings of Sure et al. [15]. In Table 2, the maximum absolute errors of the present method and its refinement are presented for different values of \(h\). From Tables 1 and 2, one can observe that as the mesh size \(h\) decreases the absolute errors also decrease.

**Example 4.2:** Consider the following Liénard Equation:

\[
\dot{u}(x) = u(x) + 4u^2(x) - 3u^3(x) = 0
\]

with the initial conditions,

\[
u(0) = 1/sqrt(2), \ u'(0) = sqrt(2)/4
\]

The exact solution of the given problem is given by \(u(x) = sqrt((1 + tanh(x))/2)\). Numerical results are presented in Tables 3 and 4 and Figures 3 and 4.

Table 3 exposes the pointwise absolute errors for different values of \(h\), and it can be seen that the present method improves the findings of Mashallah et al. [16] and Suheel et al. [17]. Table 4 shows the maximum absolute errors of the present method and its refinement for different values of \(h\). In Tables 3 and 4, it can be seen that as the mesh size \(h\) decreases the absolute errors also decrease.

**Example 4.3:** Consider the following Liénard Equation:

\[
\dot{u}(x) = u(x) + 4u^2(x) - 3u^3(x) = 0
\]

with the initial conditions,

\[
u(0) = \frac{1}{\sqrt{1 + \sqrt{2}}}, \ u'(0) = 0
\]

The exact solution of the given problem is given by \(u(x) = \frac{1}{\sqrt{1 + \sqrt{2}}}\). Numerical results are presented in Tables 5 and 6 and Figures 5 and 6.

**Example 4.4:** Consider the following Liénard Equation:

\[
\dot{u}(x) = u(x) + 4u^2(x) - 3u^3(x) = 0
\]

with the initial conditions,

\[
u(0) = 1/sqrt(2), \ u'(0) = sqrt(2)/4
\]

The exact solution of the given problem is given by \(u(x) = sqrt((1 + tanh(x))/2)\). Numerical results are presented in Tables 7 and 8 and Figures 7 and 8.

Table 5 exposes the pointwise absolute errors for different values of \(h\), and it can be seen that the present method improves the findings of Mashallah et al. [16] and Suheel et al. [17]. Table 6 shows the maximum absolute errors of the present method and its refinement for different values of \(h\). In Tables 5 and 6, it can be seen that as the mesh size \(h\) decreases the absolute errors also decrease.
The exact solution of the given problem is given by
\[ u(x) = \sqrt{(\sec h^2(x))/(2\sqrt{2} + (1 - \sqrt{2}) \sec h^2(x))} \]

Figure 5. Numerical solution versus exact solution for Example 4.3 when \( h = 0.05 \).

The exact solution of the given problem is given by
\[ u(x) = \sqrt{(\sec h^2(x))/(2\sqrt{2} + (1 - \sqrt{2}) \sec h^2(x))} \]

Numerical results are presented in Tables 5 and 6 and Figure 5.

Table 5 exhibits the pointwise absolute errors for different values of \( h \), and it can be seen that the present method improves the findings of Mashallah et al. [16] and Suheel et al. [17]. From Tables 5 and 6, one can observe that as the mesh size \( h \) decreases, the absolute errors also decrease.

5. Conclusion

This study considered a sixth order predictor–corrector method for solving Liénard second order nonlinear differential equations. The stability and convergence of the method have been investigated. To further collaborate the applicability of the proposed method, tables of absolute errors and graphs have been plotted. Tables 1, 3 and 5 show that the pointwise absolute errors obtained by the present method have been compared with pointwise absolute errors obtained by different authors. From Tables 1–6, one can observe that all the absolute errors decrease rapidly as \( h \) decreases, which in turn show the convergence of the computed solution. Figures 1, 3 and 5 indicate that the present method approximates the exact solution very well. Figures 2 and 4 show that as \( h \) decreases, the absolute error goes to zero which shows that small step size provides a better approximation.

Concisely, the present method is computationally stable, efficient, convergent and simple to use. Furthermore, it gives more accurate solution than some earlier existing numerical methods, including the more recent method in [15–17].

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Gashu Gadisa Kiltu http://orcid.org/0000-0003-3541-2630
Gemechis File Duressa http://orcid.org/0000-0003-1889-4690

References

[1] Prajapati RN, Mohan R, Kumar P. Numerical solution of generalized Abel’s integral equation by variational iteration method. Am J Comput Math. 2012;2:312–315.
[2] Yüzbas S. A numerical scheme for solutions of a class of nonlinear differential equations. J Taibah Univ Sci. 2017;11:1165–1181.
[3] Roba G, Gadisa G, Hailu K. Fifth order predictor-corrector method for quadratic Riccati differential equations. Int J Eng Appl Sci. 2017;9(4):51–64.
[4] Liénard A. Etude des oscillations entretenues. Rev Gen Electr. 1928;23:901–912 and 946–954.
[5] Guckenheimer J. Dynamics of the van der Pol equation. IEEE Trans Circ Syst. 1980;27:983–989.
[6] Zhang ZF, Ding T, Huang HW, et al. Qualitative theory of differential equations. Peking: Science Press; 1985.
[7] Kumar D, Agarwal PR, Singh J. A modified numerical scheme and convergence analysis for a fractional model of Liénard’s equation. J Comput Appl Math. 2017. doi:10.1016/j.cam.2017.03.011
[8] Harko T, Lobo SNF, Mak MK. A class of exact solutions of the Liénard type ordinary non-linear differential equation. J Eng Math. 2014;89(1):193–205.
[9] Feng Z. On explicit exact solutions for the Liénard equation and its applications. Phys Lett A. 2002;293:50–56.
[10] Kong D. Explicit exact solutions for the Liénard equation and its applications. Phys Lett A. 1995;196:301–306.
[11] Matinfar M, Hosseinzadeh H, Ghanbari M. A numerical implementation of the variational iteration method for the Liénard equation. World J Model Simul. 2008;4:205–210.
[12] Matinfar M, Mahdavi M, Raeesy Z. Exact and numerical solution of Liénard’s equation by the variational homotopy perturbation method. J Inf Comput Sci. 2011;6(1):73–80.
[13] Sun J, Wang W, Wu L. On explicit exact solutions for the Liénard equation and its applications. Phys Lett A. 2003;318:93–101.
[14] Kaya D, El-Sayed SM. A numerical implementation of the decomposition method for the Liénard equation. Appl Math Comput. 2005;171:1095–1103.
[15] Sure K, Aytekin E, Mehmet TA. A numerical approximation to some specific nonlinear differential equations using Magnus series expansion method. NTMSCI. 2016;4(1):125–129.
[16] Mashallah M, Saber RB, Maryam G. Solving the Liénard equation by differential transform method. World J Model Simul. 2012;8(2):142–146.
[17] Suheel AM, Ijaz MQ, Muhammad A, et al. Numerical solution of Liénard equation using Hybrid Heuristic computation. World Appl Sci J. 2013;28(5):636–643.
[18] Eberly D. Stability analysis for systems of differential equations. Mountain View: Geometric Tools, LLC; 2008.