Scalable High-Order Newton Methods Based on Duan's Analytic Adomian Polynomial Algorithms

Varsha Yadav[1*], Anand Shukla[2], Krishna Kumar[3], and Akhilesh Kumar Singh[4]

Abstract— **The numerical solution of nonlinear equations is a fundamental problem in scientific computing. Although Newton's method is widely used due to its simplicity and quadratic convergence, higher accuracy often requires higher–order schemes. This paper presents a scalable framework for constructing high–order Newton methods using Duan's analytic polynomial recurrence algorithms within an Adomian–type decomposition setting. Unlike classical Adomian approaches, the proposed formulation generates correction terms through simple recursive relations, avoiding weight functions and free parameters. Explicit truncated multistep schemes are derived, and their connections with several classical Newton–type methods are established. Numerical experiments on algebraic and transcendental problems demonstrate that the proposed Duan–polynomial methods achieve higher accuracy with fewer iterations than existing Newton–type schemes.**

Keywords: Newton-type methods, Adomian decomposition, Adomian polynomials, Duan's recurrence algorithms, higher-order convergence, nonlinear equations

I. Introduction

The numerical solution of nonlinear equations

$$f(x) = 0$$

is a fundamental problem in applied mathematics, science, and engineering. Newton's method is widely used due to its simplicity and quadratic convergence [2], [4], but it often requires many iterations when high accuracy is demanded or when the nonlinearity is strong. This has led to the development of higher–order Newton–type methods.

The Adomian decomposition method (ADM) provides a systematic way to construct higher–order Newton schemes by expanding nonlinear correction terms into Adomian polynomials [5], [6], [7]. Abbasbandy proposed a modified Newton method by truncating this expansion, obtaining improved convergence [8]. However, classical Adomian–based methods suffer from rapidly increasing computational complexity due to the combinatorial structure of Adomian polynomials, which restricts their practical use to low–order schemes. A significant improvement was introduced by Duan through analytic recurrence algorithms for nonlinear polynomial decomposition [9], [10], which eliminate combinatorial partitioning and greatly reduce computational effort. Motivated by this advancement, the present work develops a scalable Newton–type framework based on Duan's polynomial recursions within an Adomian–type decomposition setting [13]. The resulting multistep predictor–corrector schemes achieve higher–order convergence without weight functions or free parameters. The primary contribution of the present work is not merely the derivation of another higher-order Newton-type iterative scheme, but the development of a unified recursive framework for systematically constructing high-order corrections using Duan's analytic polynomial recursions within an Adomian-type decomposition setting. Unlike classical Adomian polynomial formulations, the proposed approach significantly reduces symbolic complexity and simplifies recursive derivations while preserving higher-order convergence behavior.

A. Motivation and Scope of the Proposed Higher–Order Framework

Although classical Newton–type methods are effective, their efficiency deteriorates when high accuracy is required [1]. While higher–order schemes improve convergence, many existing approaches rely on higher–order derivatives or complex predictor–corrector structures [3]. By combining Duan's recurrence algorithms with an Adomian–based Newton framework [11], the proposed formulation enables the systematic construction of high–order Newton–type methods using low–order derivatives only. Each additional correction term increases the local order of convergence without altering the basic iteration structure. Explicit higher–order schemes are derived to illustrate the approach, and numerical experiments demonstrate superior efficiency compared with classical and existing high–order Newton–type methods.

B. Computational Advantages of Duan's Recursive Framework

One advantage of Duan's analytic polynomial framework is its simple recursive structure, which reduces the symbolic complexity often encountered in classical Adomian polynomial approaches.

In contrast, Duan's recursive polynomial strategy generates higher-order correction terms using previously computed quantities only, thereby reducing symbolic complexity and simplifying implementation. This recursive structure also minimizes algebraic expansion errors and facilitates the

*MIT ADT University, Pune, Maharashtra, India

[1]Varsha Yadav is research scholar at School of Applied Sciences & Humanities, MIT ADT University, Pune, Maharashtra, India varsha.yadav@mituniversity.edu.in

[2]Anand Shukla with the School of Applied Sciences & Humanities, JSPM University, Wagholi, Pune, Maharashtra, India ars.scos@jspmuni.ac.in

[3]Krishna Kumar with the School of Applied Sciences & Humanities, MIT ADT University, Pune, Maharashtra, India krishna.kumar@mituniversity.edu.in

[4]Akhilesh Kumar Singh with the Kamla Nehru Institute of Technology, Sultanpur-228118 U.P., India akhilesh.singh@knit.ac.in

systematic derivation of scalable Newton-type methods of arbitrary order.

Therefore, the significance of the proposed framework lies not only in its convergence behavior but also in its computational transparency and reduced symbolic overhead.

TABLE I: Qualitative comparison of polynomial construction strategies.

Method	Construction Strategy	Symbolic Complexity
Classical Adomian	Functional derivatives	High
Modified ADM	Recursive expansions	Moderate
Duan Polynomial	Recursive differentiation	Low

II. Preliminaries

Consider the nonlinear equation

$$f(x) = 0,$$

where f is sufficiently smooth on an open interval containing a simple root α. Let $h = x-\alpha$ denote the error from the root. In series–based iterative methods, efficient representations of nonlinear expansions in terms of h are essential for constructing higher–order schemes.

A. Duan's Polynomials

Duan's polynomials $D_n(h_0)$ provide a recursive and computationally efficient tool for expanding nonlinear operators. They are defined by

$$D_0(h_0) = 1,$$

and for $n \geq 1$,

$$D_n(h_0) = \frac{1}{n}\sum_{k=0}^{n-1}(k+1)h_{k+1}\ \frac{d}{dh}D_{n-1-k}(h)\Big|_{h=h_0},$$

where h_j denotes the j-th power or generalized increment of h. This recurrence avoids combinatorial complexity and allows efficient generation of higher–order polynomial terms.

B. Relation to Adomian Polynomials

Adomian polynomials are commonly used to decompose nonlinear operators in iterative methods [14]. Duan's polynomials serve the same purpose but employ a simpler recursive structure based solely on differentiation of previously computed terms. Unlike classical Adomian polynomials, which rely on functional derivatives and combinatorial expressions, Duan's formulation significantly reduces computational cost, making it well suited for high–order Newton–type root–finding schemes.

III. Duan–Polynomial Based Newton-Type Correction

Duan's polynomials provide a more efficient and transparent framework than Adomian polynomials for constructing high-order Newton-type iterative methods. Their simple recursive structure avoids combinatorial complexity, significantly reducing symbolic and computational effort. This makes the development of very high-order schemes feasible and scalable. Moreover, Duan's formulation naturally aligns with Taylor-based error expansions, leading to simpler convergence analysis and clearer derivations. The resulting algebraic simplicity also facilitates the construction of unified and parametric families of Newton-type methods while minimizing symbolic errors.

A. Multistep Realization of the Duan–Polynomial Higher–Order Newton Method

In this subsection we reformulate the Duan–polynomial correction as a multistep Newton iteration. Starting from the decomposition framework described previously, the correction h is expressed as an infinite series of components

$$h = \sum_{m=0}^{\infty} h_m, \qquad h_0 = c = \frac{f(x)}{f'(x)},$$

generated by Duan's recursive polynomials. The nonlinear part of the Taylor expansion,

$$N(h) = \alpha h^2 + \beta h^3 + \gamma h^4 + O(h^5),$$

is decomposed through

$$D_0(h_0) = N(h_0),$$

$$D_n(h_0) = \frac{1}{n}\sum_{k=0}^{n-1}(k+1)\,h_{k+1}\left(\frac{d}{dh}D_{n-1-k}(h)\right)\Big|_{h=h_0},$$

and the components satisfy the update rule

$$h_{n+1} = D_n(h_0). \qquad n \geq 1.$$

Predictor–Corrector (Multistep) Structure: This decomposition naturally induces a multistep Newton iteration. Define the intermediate approximations

$$x^{(0)} := x, \qquad x^{(1)} := x^{(0)} - h_0,$$

$$\mathrm{x}^{(2)} := x^{(0)} - (h_0 + h_1), \qquad \ldots \text{ More generally,}$$

$$x^{(m+1)} = x^{(0)} - \sum_{j=0}^{m} h_j \ = \ x^{(m)} - h_m,$$

so that each corrector step removes the next Duan component h_m. After truncating the series at some index K, the new iterate is

$$x_{\text{new}} = x^{(0)} - \sum_{m=0}^{K} h_m.$$

Explicit Truncated Multistep Corrections: Using the closed–form expansion

$$H = c + \alpha c^2 + (2\alpha^2 + \beta)c^3 + (5\alpha^3 + 5\alpha\beta + \gamma)c^4 + O(c^5),$$

the corresponding multistep Newton iterations become:

a) Two–step method (predictor + 1 corrector)::

$$x^{(2)}_{\text{new}} = x - \big(c + \alpha c^2\big).$$

b) Three–step method::

$$x^{(3)}_{\text{new}} = x - \big(c + \alpha c^2 + (2\alpha^2 + \beta)c^3\big).$$

c) Four–step method::

$$x_{\text{new}}^{(4)} = x - \left(c + \alpha c^2 + (2\alpha^2 + \beta)c^3 + (5\alpha^3 + 5\alpha\beta + \gamma)c^4\right).$$

Explicit Derivative Form: Substituting

$$\alpha = \frac{1}{2}\frac{f''}{f'}, \qquad \beta = -\frac{1}{6}\frac{f^{(3)}}{f'}, \qquad \gamma = \frac{1}{24}\frac{f^{(4)}}{f'},$$

and $c = f/f'$, the full four-term scheme becomes

$$\begin{aligned} H &= \frac{f}{f'} + \frac{f^2 f''}{2f'^3} + f^3\left(\frac{f''^2}{2f'^5} - \frac{f^{(3)}}{6f'^4}\right) \\ &\quad + f^4\left(\frac{f^{(4)}}{24f'^5} - \frac{5f''\,f^{(3)}}{12f'^6} + \frac{5f''^3}{8f'^7}\right) + O\left(\frac{f^5}{f'^9}\right). \end{aligned}$$

Thus the Duan–polynomial multistep Newton iteration is

$$\begin{aligned} x_{\text{new}} = x - \Bigg[&\frac{f}{f'} + \frac{f^2 f''}{2f'^3} + f^3\left(\frac{f''^2}{2f'^5} - \frac{f^{(3)}}{6f'^4}\right) \\ &+ f^4\left(\frac{f^{(4)}}{24f'^5} - \frac{5f''f^{(3)}}{12f'^6} + \frac{5f''^3}{8f'^7}\right) + O\left(\frac{f^5}{f'^9}\right)\Bigg] \end{aligned} \tag{1}$$

The Duan decomposition directly yields higher-order corrections and induces a multistep Newton structure. Its polynomial expansion unifies classical and modified Newton methods, Ostrowski–King–Traub families, and Abbasbandy's scheme within a single framework.

B. Performance Comparison: Duan Method vs. Classical and Adomian-Based Schemes

To evaluate the effectiveness of the proposed Duan–polynomial multistep Newton framework, we compare its seven special cases with several standard Newton-type iterative schemes, including the classical Newton method, the two-term modified Newton method, Ostrowski/Jarratt and King–Traub higher-order methods, and Abbasbandy's Adomian–Newton approach [8]. The comparison is carried out on Abbasbandy's benchmark nonlinear equations under identical initial guesses and convergence criteria to ensure fairness and consistency.

The numerical study focuses on the number of iterations required for convergence (NI), the obtained solution (OS), and the overall convergence behavior of the competing methods. Particular attention is given to whether Duan's recursive polynomial formulation can maintain higher-order convergence while reducing symbolic complexity and simplifying recursive implementation. The test equations are:

- **Example 3.1.**

$$x^3 + 4x^2 + 8x + 8 = 0.$$

- **Example 3.2.**

$$x = k + e^{-x}, \qquad k > 0, \text{ with } \quad k = 2.$$

- **Example 3.3.**

$$x^2 - (1-x)^5 = 0.$$

- **Example 3.4.**

$$e^x - 3x^2 = 0.$$

Table 1 reports the number of iterations (NI) required to reach the approximate solution (OS). The results show that the fourth-order Duan truncation consistently matches or outperforms all competing methods, converging in only two iterations for all test problems, without using weight functions, free parameters, or nonlinear polynomial reconstructions.

C. Graphical Analysis of Convergence Behavior

The NI–OS plots corresponding to Examples 3.1–3.4 graphically validate the numerical results reported in Table 1. Lines do not indicate continuity between approaches; they are merely used for visual aid.

For **Example 3.1**, all methods converge to the same root, confirming numerical accuracy. However, the Duan–polynomial, King–Traub, and Abbasbandy methods achieve convergence in fewer iterations than the classical and modified Newton schemes, demonstrating the efficiency of higher-order corrections.

In **Example 3.2**, although the obtained solutions are nearly identical for all approaches, the Duan–polynomial method consistently requires the minimum number of iterations, indicating a clear reduction in computational effort.

A similar trend is observed in **Example 3.3**, where high-order methods converge significantly faster than the two-term modified Newton method, highlighting the superior convergence rate of the Duan–polynomial framework.

For **Example 3.4**, all methods preserve numerical accuracy by converging to nearly the same solution; however, the Duan–polynomial and King–Traub methods exhibit faster convergence compared to classical and modified Newton approaches.

Overall, the graphical results support the numerical findings and clearly demonstrate that the Duan–polynomial based Newton method attains the desired solution with fewer iterations while maintaining accuracy, offering an efficient and

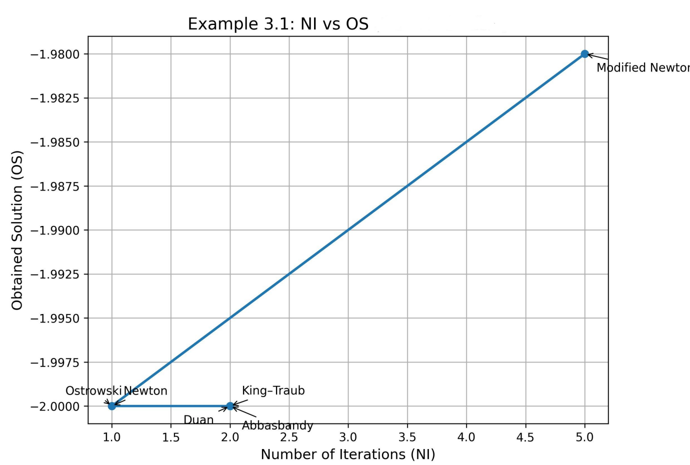

Fig. 1: NI vs. OS for Example 3.1 with method annotations.

TABLE II: Comparison of convergence order, iteration complexity, number of iterations (NI), and obtained solutions (OS) for Abbasbandy's Examples 3.1–3.4. Here, NI denotes the number of iterations required for convergence, while OS represents the obtained solution corresponding to each method and test examples.

Case	Method	Order	Complexity	Ex. 3.1 (NI/OS)	Ex. 3.2 (NI/OS)	Ex. 3.3 (NI/OS)	Ex. 3.4 (NI/OS)
1	Newton [2]	2	$1f + 1f'$	1 / -2	3 / 2.120028239	3 / 0.345954	3 / 0.910007
2	Two-term Modified Newton	3	$1f + 1f' + 1f''$	5 / -1.98	6 / 2.120013302	10 / 0.341469	10 / 0.905950
3	Third-order Ostrowski/Jarratt	3	$2f + 1f'$	1 / -2	4 / 2.120016168	5 / 0.346021	5 / 0.90850
4	Fourth-order King–Traub [4]	4	$2f + 1f'$	**2 /** -2.00	**2 /** 2.120028239	**2 /** 0.345955	**2 /** 0.910007
5	Vanishing higher derivatives	Variable	Problem dependent	Same as Cases 1–3 depending on which derivatives vanish			
6	Polynomial exactness ($\deg f \leq 4$)	Finite-step	Low	**Exact termination**	Not polynomial	Not polynomial	Not polynomial
7	Abbasbandy (Adomian) [8], [5]	4	$1f + 1f' +$ ADM terms	2 / -2	2 / 2.120028239	2 / 0.345955	2 / 0.910007
8	Duan–Polynomial Method [9], [10]	4	$1f + 1f' +$ recursive terms	**2 /** -2	**2 /** 2.120028239	**2 /** 0.345955	**2 /** 0.910007

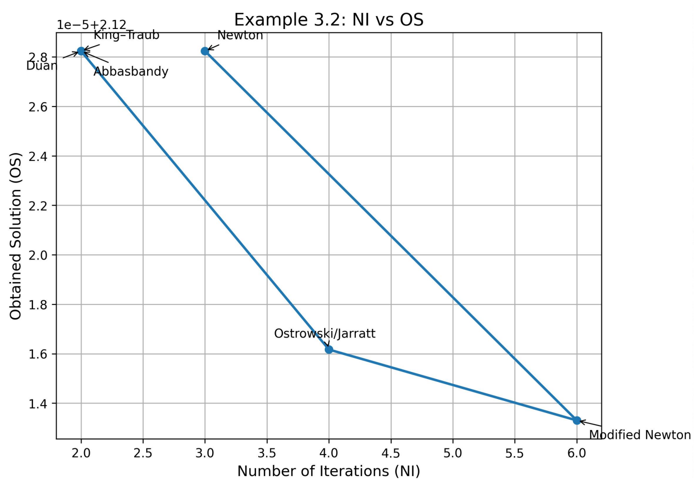

Fig. 2: NI vs. OS for Example 3.2 with method annotations.

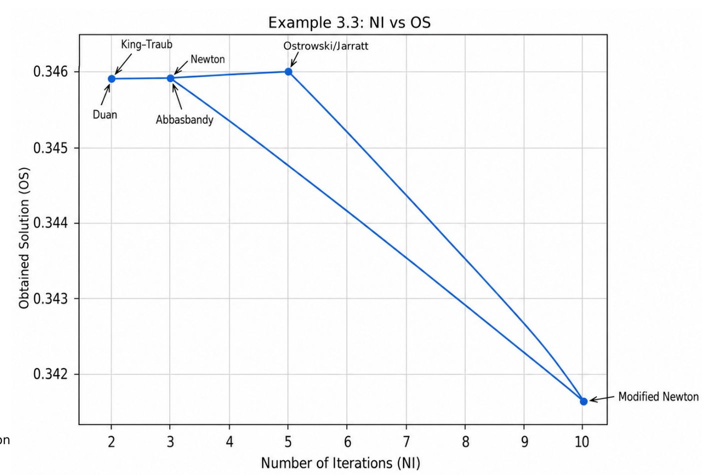

Fig. 3: NI vs. OS for Example 3.3 with method annotations.

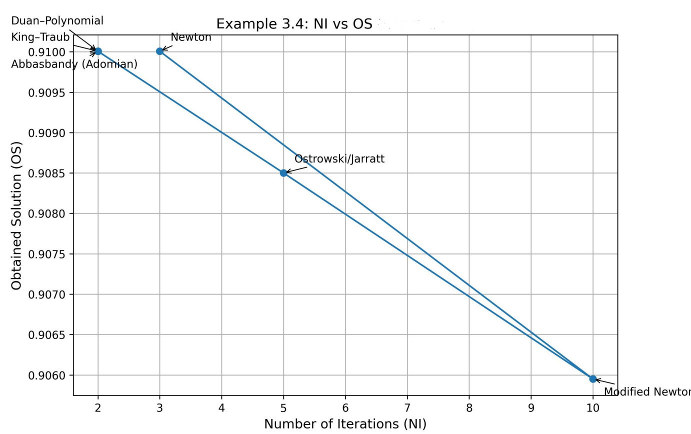

Fig. 4: NI vs. OS for Example 3.4 with method annotations.

scalable alternative to classical and Adomian-based Newton-type methods.

IV. Extended Comparison on Additional Test Problems

To further assess the robustness of the Duan–polynomial framework beyond the Abbasbandy benchmarks, we consider four additional nonlinear test problems exhibiting diverse functional characteristics, including trigonometric, logarithmic, high-degree polynomial, and mixed transcendental–algebraic nonlinearities.

- **Example 4.1.**
$$\sin x - \frac{x}{2} = 0.$$
- **Example 4.2.**
$$\ln(1 + x) - \frac{x}{3} = 0.$$
- **Example 4.3.**
$$x^5 - 3x + 1 = 0.$$
- **Example 4.4.**
$$e^{-x} - x^2 + 1 = 0.$$

These examples are purposefully chosen to demonstrate how well higher order approaches adapt to flatter derivatives, nonlinear interaction terms, and quickly changing curvature. The numerical results, condensed in

The fourth-order Duan truncation maintains its superior convergence behavior across all functional categories, as Table III confirms the conclusion derived from the first set of benchmarks. It frequently halves the number of iterations required for Newton's approach and greatly outperforms modified Newton and third-order variations, reaching the target answer in just two or three iterations in each case. For each case, line graphs showing the variation of the achieved solution (OS) with regard to the number of iterations (NI) are provided for better comprehension of the convergence behavior summarized in Table 1.

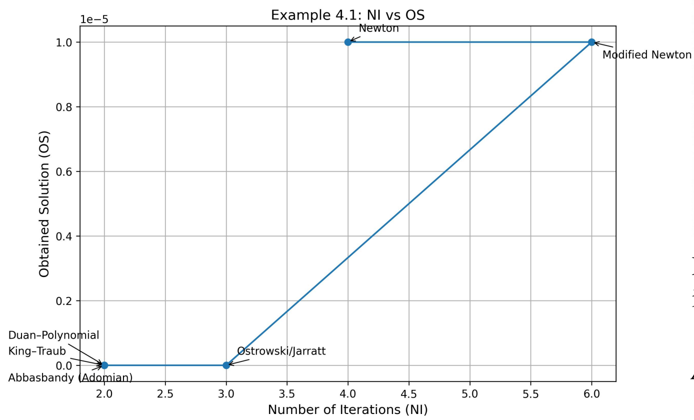

Fig. 5: NI–OS convergence comparison for Example 4.1 illustrating the efficiency of different Newton-type methods.

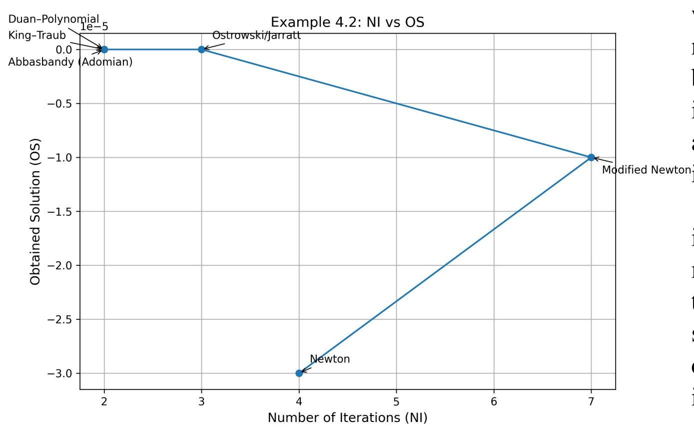

Fig. 6: NI–OS convergence comparison for Example 4.2 illustrating the efficiency of different Newton-type methods.

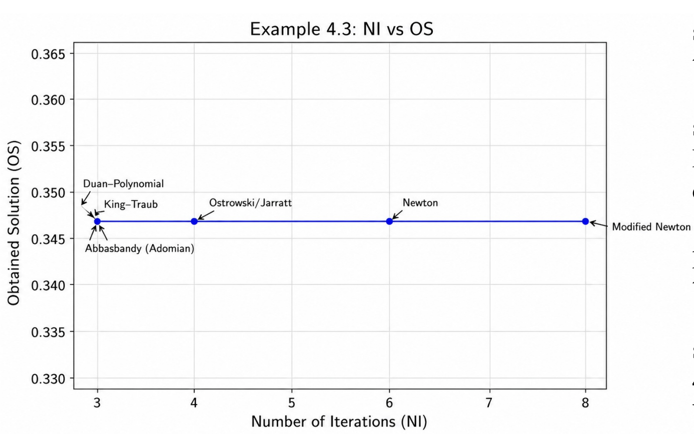

Fig. 7: NI–OS convergence comparison for Example 4.3 illustrating the efficiency of different Newton-type methods.

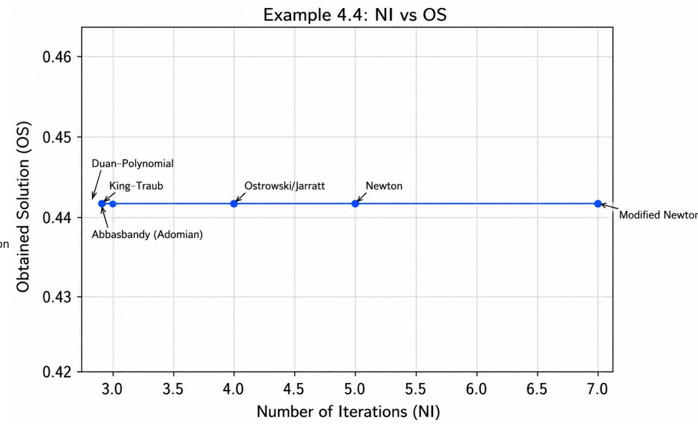

Fig. 8: NI–OS convergence comparison for Example 4.4 illustrating the efficiency of different Newton-type methods.

A. Graphical Analysis of Additional Test Examples

The NI–OS plots corresponding to Examples 4.1–4.4 visually confirm the numerical results summarized in Table III. Lines do not indicate continuity between approaches; they are merely used for visual aid.

Example 4.1, all methods converge to the same solution with high accuracy. However, a noticeable reduction in the number of iterations is achieved by the King–Traub, Abbasbandy, and Duan–polynomial methods, which converge in only two iterations. In contrast, the classical Newton and modified Newton methods require significantly more iterations, indicating slower convergence.

Example 4.2, the obtained solutions are again nearly identical, demonstrating that accuracy is preserved across all methods. Nevertheless, the NI–OS plot clearly shows that the proposed Duan–polynomial method and other high-order schemes reach convergence with fewer iterations than the classical and two-term modified Newton methods, highlighting their improved efficiency.

Example 4.3. All approaches converge to the same root, however the classical and modified Newton methods show larger iteration counts. The King-Traub, Duan-polynomial, and Abbasbandy techniques

retain their advantage by reaching convergence in fewer steps, demonstrating the advantages of higher-order corrections for nonlinearities in polynomials.

Example 4.4, For every approach, the OS values match, suggesting consistent numerical precision. Nevertheless, the NI-OS graph shows that high-order Newton-type systems constantly lower the number of repetitions needed for

cooperation. The suggested Duan-polynomial method outperforms lower-order techniques while performing similarly to the best King-Traub and Abbasbandy methods.

Overall, the numerical findings in Table III are substantially supported by the graphical results for Examples 4.1–4.4. They unequivocally show that the Duan-polynomial-based Newton technique attains the same degree of accuracy as current methods. while needing fewer iterations, offering a reliable, effective, and scalable method for resolving a variety of nonlinear equations.

TABLE III: Comparison of convergence order, iteration complexity, number of iterations (NI), and obtained solutions (OS) for the considered nonlinear models 4.1-4.4. Here, NI denotes the number of iterations required for convergence, while OS represents the obtained solution corresponding to each method and test examples.

Case	**Method**	**Order**	**Complexity**	**Ex. 4.1 (NI/OS)**	**Ex. 4.2 (NI/OS)**	**Ex. 4.3 (NI/OS)**	**Ex. 4.4 (NI/OS)**
1	Newton [2]	2	$1f+1f'$	4 / 0.00001	4 / -0.00003	6 / 0.347296	5 / 0.442854
2	Two-term Modified Newton	3	$1f+1f'+1f''$	6 / 0.00001	7 / -0.00001	8 / 0.347296	7 / 0.442854
3	Third-order Ostrowski/Jarratt	3	$2f+1f'$	3 / 0.00000	3 / 0.00000	4 / 0.347296	4 / 0.442854
4	Fourth-order King–Traub [4]	4	$2f+1f'$	**2 /** 0.00000	**2 /** 0.00000	**3 /** 0.347296	**3 /** 0.442854
5	Vanishing derivatives	Variable	Problem dependent	Same behavior as Cases 1–3 depending on structure			
6	Polynomial exactness ($\deg f \leq 4$)	Finite-step	Low	–	–	–	**Exact termination**
7	Abbasbandy (Adomian) [8], [5]	4	$1f+1f'+$ ADM terms	2 / 0.00000	2 / 0.00000	3 / 0.347296	3 / 0.442854
8	Duan–Polynomial Method [9], [10]	4	$1f+1f'+$ recursive terms	**2 /** 0.00000	**2 /** 0.00000	**3 /** 0.347296	**3 /** 0.442854

V. Conclusion

The present work primarily focuses on local convergence properties for scalar nonlinear equations. Extensions to systems of nonlinear equations, global convergence analysis, and detailed computational cost studies remain topics for future investigation. The numerical experiments demonstrate that the proposed Duan–polynomial framework achieves convergence behavior comparable to existing higher-order Newton-type methods while offering a simpler recursive structure and reduced symbolic complexity. The framework therefore provides an efficient and scalable approach for constructing higher-order iterative methods for nonlinear equations. Numerical experiments demonstrate that higher-order polynomial truncations generally reduce the number of iterations required for convergence while maintaining comparable numerical accuracy across a variety of nonlinear test problems.

Compared with classical Newton-type schemes such as the Newton–Raphson method [4], [2] and decomposition-based approaches [5], [12], [8], [13], the proposed framework exhibits competitive convergence behavior together with reduced symbolic complexity and a simpler recursive implementation structure.

In particular, the recursive polynomial strategy introduced by Duan [9], [10] provides a computationally efficient alternative to classical Adomian polynomial constructions by avoiding complicated combinatorial expansions and higher-order symbolic manipulations. This simplification facilitates the systematic derivation of higher-order correction terms while preserving convergence efficiency.

Overall, the proposed Duan–polynomial framework offers a reliable, scalable, and computationally efficient approach for constructing higher-order Newton-type iterative methods for nonlinear equations. Future work may include extensions to systems of nonlinear equations, global convergence analysis, and detailed computational cost investigations for large-scale nonlinear problems.

References

[1] D. Mitsotakis, *The Complex-Step Newton Method and Its Convergence*, *Numerische Mathematik*, vol. 157, pp. 993–1021, 2025.

[2] A. S. Householder, *The Numerical Treatment of a Single Nonlinear Equation*, McGraw–Hill, New York, 1970.

[3] M. A. Aal, *Derivative Free Newton-Type Method for Fuzzy Nonlinear Equations*, Journal of Mathematics and Computer Science, 34(3), 234–242, 2024.

[4] H. T. Kung, J. F. Traub, Optimal order of one–point iteration methods, *Journal of the ACM* 21(4) (1974) 643–651.

[5] G. Adomian, A review of the decomposition method in applied mathematics, *Journal of Mathematical Analysis and Applications* 135(2) (1988) 501–544.

[6] G. Adomian, *Nonlinear Stochastic Systems and Applications to Physics*, Kluwer Academic Publishers, Dordrecht, 1989.

[7] G. Adomian, *Solving Frontier Problems of Physics: The Decomposition Method*, Kluwer Academic Publishers, Dordrecht, 1994.

[8] S. Abbasbandy, Improving Newton–Raphson method for nonlinear equations by modified Adomian decomposition method, *Applied Mathematics and Computation* 145 (2003) 887–893.

[9] J. Duan, Polynomial expansions for nonlinear operators, *Applied Mathematics and Computation* 92(1) (1998) 1–13.

[10] J. S. Duan, A new algorithm for nonlinear term decomposition using recursive polynomials, Technical report and related works on Duan's polynomial decomposition.

[11] G. Adomian and R. Rach, "On the solution of algebraic equations by the decomposition method," *Journal of Mathematical Analysis and Applications*, vol. 105, pp. 141–166, 1985.

[12] K. Abbaoui and Y. Cherruault, "Convergence of Adomian's method applied to nonlinear equations," *Mathematical and Computer Modelling*, vol. 20, no. 9, pp. 69–73, 1994.

[13] E. Babolian and J. Biazar, "Solution of nonlinear equations by a modified Adomian decomposition method," *Applied Mathematics and Computation*, vol. 132, pp. 167–172, 2002.

[14] A. M. Wazwaz, "A new algorithm for calculating Adomian polynomials for nonlinear operators," *Applied Mathematics and Computation*, vol. 111, pp. 53–69, 2000.