Jagannath University Journal of Science
Volume 12, Number II, Dec. 2025, pp. 159−176
https://jnu.ac.bd/journal/portal/archives/science.jsp
ISSN 3005-4486 (Online), ISSN 2224-1698 (Print)

Quantitative Analysis of Error Propagation and Computational Efficiency in Numerical Solutions of Boundary Value Problems

Research Article

Md. Shah Najmus Shakib[*1,2], Bijan Krishna Saha[1], Nipa Roy[1,2], Baysa Binte Azad[3], Moumita Datta[4]

[1]*Department of Mathematics, Faculty of Science, University of Barishal, Barishal-8254, Bangladesh*
[2]*Department of Mathematics, University of Global Village (UGV), Barishal-8200, Bangladesh*
[3]*Barishal Government Women's College, Barishal-8200, Bangladesh*
[4]*Department of Computer Science and Engineering, Faculty of Engineering and Technology, University of Global Village (UGV), Barishal-8200, Bangladesh*

DOI: https://doi.org/10.3329/jnujsci.v12i2.90147
Received: 13.03.2026, Accepted: 10.05.2026, Published: July 2026

ABSTRACT

This study presents a comparative analysis of three widely used numerical methods, the Shooting method, the Galerkin method, and the Modified Galerkin method for solving boundary value problems (BVPs). The Shooting method is applied by converting the BVP into an initial value problem, solved by a Runge-Kutta scheme, and the Galerkin and Modified Galerkin methods are developed in a weighted residual form. The methods are used on a variety of problems, such as application-driven models like population dynamics and epidemiological systems. The accuracy of each method is assessed by an error analysis based on the root mean square (RMS) error. Besides accuracy, the computation time is also a performance metric used to analyze the trade-off between efficiency and precision. The numerical data indicate that the Shooting method is more likely to give a better accuracy in complex problems, and the Modified Galerkin method is more favorable to balance accuracy and computational cost. In contrast, the standard Galerkin method exhibits comparatively lower accuracy under the considered problem settings. This paper points out the comparative weaknesses and strengths of these techniques and can help in understanding how these techniques can be used to solve BVPs.

Keywords: *Boundary Value Problems (BVPs), Ordinary Differential Equations (ODEs), Shooting Method, Modified Galerkin Method, Numerical Analysis*

[*] **Corresponding author:** Md. Shah Najmus Shakib
E-mail: shakibn932@gmail.com

1. Introduction

Boundary value problems (BVPs) of ordinary differential equations (ODEs) contribute significantly to mathematical modeling in many applied sciences and engineering fields. BVPs frequently arise in areas such as fluid flow, structure analysis, control theory, and heat conduction. Precise and effective numerical approaches are required instead of analytical solutions to such problems, which are often difficult to attain. As a result, one of the key issues in the area of computational mathematics is the search for a reliable numerical method for BVPs. There are several numerical methods formulated for solving BVPs, both linear and non-linear, over the years (Pandey 2010). Among them, the Shooting method, the Galerkin method, and the modified Galerkin method are well used.

Although it is easy and computationally efficient, it poses some problems with stiff or highly non-linear problems. It can be found to be very effective in dealing explicitly with complicated situations and has been extensively studied as applied to heat conduction, wave propagation, and vibration analysis. To enhance efficiency, an extension of the method has been suggested, known as the modified Galerkin method. These modifications usually employ part-wise or low rank additions or corrections, which dramatically cut down the cost, at little expense to accuracy (Kant et al. 2023, Laud et al. 2024). These two methods have been discussed on a one-by-one basis in some studies. (Singh 2022) examined the Shooting method on boundary value problem, and (Eichmeir and

Steiner 2026) extended the double Shooting method to multibody problems. For solving higher-order linear and non-linear boundary value problems, a significant modification has been made to the multiple shooting method by dividing the domain into subintervals, constructing semi-analytical approximate solutions that are continuously differentiable up to the required order, and improving accuracy by minimizing the L^2-norm of the residual functions in each subinterval (Kheybari et al. 2025). (Javeed and Hincal 2024) enhances the Shooting Method by adding both Newton-Raphson iterations and a Runge-Kutta fourth-order method to effectively deal with initial value sensitivity and convergence issues in coupled, high-order problems that are non-linear and have boundary values.

In the analysis of the standard Galerkin-based finite element approach for problems involving initial and boundary values, (Thomee 2006) has insights into the BVPs. The wavelet-based Galerkin method has sought numerical solutions to BVPs using a number of families of wavelets as basis functions (Angadi 2025). A high-efficiency 3D Discontinuous Galerkin (DG) technique is an efficient and precise forward simulation technique of ground-penetrating radar (GPR) in complex media (Xue et al. 2025). (Ahmed 2025) introduced new Galerkin operational matrices of derivatives in terms of Robin-modified generalized shifted Jacobi polynomials and applied them to develop a highly accurate spectral method of solving two-point fractional-order non-linear boundary value problems with Robin boundary conditions.

An analysis of weak Galerkin finite element methods shows that it may be used to solve time-dependent singularly perturbed parabolic convection-diffusion equations with layer-adapted meshes to solve the problems of boundary layers effectively (Toprakseven and Dinibutun 2024). The hybrid Galerkin-finite difference approach (with modified Bernoulli polynomials in the Galerkin term in space) simplifies to numerically compute time-space fractional parabolic PDEs (Hossan et al. 2024). A highly precise spectral Galerkin approach, based on new generalized shifted Jacobi polynomials and their derivative matrices, is used to solve high-order ODEs and multi-term variable-order fractional differential equations (Ahmed 2024). (Al-Abadi and Abd 2024) applied Galerkin finite element error estimates of [1+2] dimensional integro-differential equations on a 2D mesh with linear triangular elements.

A weak Galerkin finite element method with modifications is used in solving time-dependent parabolic partial differential equations on highly stretched triangular meshes (Li et al. 2023). Pseudo-Galerkin spectral scheme works very efficiently based on using the second derivatives of Legendre polynomials as the basis functions in order to accurately approximate a solution of a linear and non-linear boundary value problem, using which some real-life models are demonstrated (Abdelhakem et al. 2022). A meshless conservative Galerkin approach has been created by using the Hamilton equations of waves (Sun and Ling 2022). An evaluation provides optimal-order error estimates, both in $L^2(\ L^2)$ and $L^2\ \ (H^1)$ norms, of the weak Galerkin finite element approximations of second-order linear parabolic equations on polygonal meshes, assuming the exact solution has very low regularity (Deka and Kumar 2020).

An analysis constructs a highly reliable and efficient a posteriori error estimator of the residual type in the weak Galerkin finite element method of second-order elliptic problems with mixed boundary conditions and with no stabilizer, demonstrating its reliability and efficiency, given appropriate conditions. A weak Galerkin finite element method applied to second-order elliptic problems with mixed boundary conditions, proving its reliability and efficiency under suitable assumptions (Xie et al. 2020). A discontinuous Galerkin difference method utilizes an energy-based stability analysis to construct high-order, energy-conserving schemes for second-order wave equations on structured grids (Zhang et al. 2021).

A Sinc-Galerkin spatial discretization is used of a time-fractional diffusion equation in 1D and 2D, and a higher-order time-dependent finite difference scheme to ensure exponential convergence in case of non-smooth solutions to the equations (Luo et al. 2024). (Kant et al. 2024) considered the modified Galerkin method as applied to the problem of integral equations, and (Mitsotakis et al. 2016) employed a modified Galerkin/finite element method to find the numerical result of the Serre-Green-Naghdi system. A modified weak Galerkin method based on a weighted L^2-projection is used to obtain a uniform convergence and remove non-physical oscillations when a singularly perturbed parabolic equation is solved on Shishkin meshes (Toprakseven and Gao 2023).

Comparison of other numerical methods, e.g., the Shooting method versus finite difference methods (Ahmad and Charan 2019) or the weighted residual schemes (Cicelia 2014) to achieve greater solution accuracy has been attempted in other works. In more recent work, there have been attempts to mix frameworks, e.g., to use Shooting with Galerkin projections (Bizzarri et al. 2018) or low-rank corrections within Galerkin schemes (Lund and Palitta 2024). (Almutairi and Saeed 2025) has worked on a comparative study of the finite difference method and the Galerkin finite element method in solving the boundary value problem, comparing their accuracy, convergence rates, computational efficiency, and the aspects of implementation using a series of numerical test problems. A numerical method using weighted residual (Galerkin, Least Squares, and Collocation) based on the Legendre polynomials basis functions and Caputo fractional derivatives offers approximate solutions, with low absolute error, to both linear and non-linear problems of the first-order fractional equations with a non-homogeneous boundary (Ruman and Islam 2024). A systematic comparison and evaluation of the numerical accuracy and computational efficiency of the Galerkin, modified Galerkin, Shooting, and Homotopy Perturbation methods for solving boundary value problems, and seeks to determine the most effective method (Shakib et al. 2026). A study shows that between Galerkin and Collocation Isogeometric Analysis, Collocation is better in terms of computation efficiency, and Galerkin in terms of accuracy and stability of acoustic wave propagation (Zampieri and Pavarino 2023).

Although a large literature exists on the topic of numerical approaches to solving boundary value problems, the majority of the literature addresses only one or two specific approaches or only provides a comparison of methods with reference to the typical benchmark problems with known analytical solutions. Although these studies are useful to test numerical correctness, they do not tend to reflect the real behavior of such methods in application-relevant situations, where governing equations are determined by real-life models and can have varying degrees of complexity, nonlinearity, and sensitivity. Further, despite the growing focus in recent computational mathematics research on efficiency and scalability, comparatively little focus has been placed on the joint evaluation of accuracy and computational cost, especially in classical methods like the Shooting, Galerkin, and Modified Galerkin methods. In this respect, there is a certain gap in the current knowledge regarding the effectiveness of these well-established techniques when applied to model-based boundary value problems, and the accuracy of such techniques in comparison to the consideration of computational efficiency.

1.1 Novelty of the Study

The present study paper fills this gap with an application-based, systematic comparative analysis of the Shooting, Galerkin, and Modified Galerkin methods. In contrast to traditional literature, which uses only the textbook illustrations, the work integrates application-based problems based on population dynamics and epidemiological (SIR-type) models, thus providing a more realistic assessment of the numerical performance. Besides that, the paper presents a two-metric assessment system, which is a synthesis of root mean square (RMS) error and computation time, to investigate the accuracy-efficiency trade-off. By using this combined method, not only is the numerical accuracy compared, but also the practical adequacy of both methods is described, which shows their relative capabilities and shortcomings in various problem contexts. This work introduces a novel understanding of the practical implementation of these methods, filling in a more thorough comprehension of the role of these methods in contemporary computational modeling, by closing the gap between classical numerical methods and application-oriented analysis and computational performance concerns.

2. Numerical Methods

This section represents three numerical methods studied in this research: the Shooting method, the Galerkin method, and the modified Galerkin method. To introduce the problem, the general second-order linear boundary value problem (BVP) is considered:

$$y'' + f(x, y, y') = 0, a \leq x \leq b$$

With boundary conditions:

$$y(a) = \alpha, y(b) = \beta.$$

2.1 Shooting Method

The shooting method converts the boundary value problem into an equivalent initial value problem by guessing the missing initial condition, then iteratively refines the guess until the boundary condition (Mostoufi and Constantinides 2023).

To implement this, the second-order equation is rewritten as a system of first-order equations:

$$y'(x) = z(x),$$

$$z'(x) = -f(x, y, z).$$

Here, $z(x)$ is an auxiliary variable. With an initial guess for $z(a)$, the system can be integrated forward, and corrections are applied until the boundary condition $y(b) = \beta$ is achieved.

2.2 Galerkin Method

The Galerkin method approximates the solution as $\tilde{y}(x) = \sum_{i=1}^{n} c_i \emptyset_i(x)$ and enforces orthogonality between the residual and each basis function

$$\int_a^b R(x) w_j(x) dx = 0.$$

2.3 Modified Galerkin Method

The modified Galerkin method reduces computational cost by applying integration by parts to transfer derivatives from the trial solution to the weighting function (Mitsotakis et al., 2016):

$$\int_a^b R(x)\emptyset_j(x)dx = \int_a^b (\tilde{y}''(x) + f(x,\tilde{y},\ \tilde{y}'))\emptyset_j(x)\ dx$$

$$\rightarrow \int_a^b \tilde{y}'(x)\emptyset'_j(x)dx + \int_a^b f(x,\tilde{y},\tilde{y}')\emptyset_j(x)dx$$

Detailed formulations and algorithmic implementations are provided in Appendix A**.**

3. Error Analysis

When solving a boundary value problem (BVP) numerically, the overall error is due to discretization, approximation, and numerical implementation. In this work, the error is discussed within a method-specific framework of the Shooting, Galerkin, and Modified Galerkin methods, as opposed to relying on generic descriptions of truncation and round-off error. Where $u(x)$ represents the exact solution and $u_h(x)$ represents the numerical approximation. The pointwise error is given as:

$e(x) = u(x) - u_h(x)$

and the overall accuracy is quantified using the root mean square (RMS) error.

3.1 Error in the Shooting Method

The Shooting method converts the BVP to an analogous initial value problem (IVP) that is resolved numerically by a time-stepping approach. A fourth-order Runge-Kutta (RK4) method is used in this research. In the case of sufficiently smooth solutions, the global truncation error of the RK4 method is:

$$e(h) = O(h^4)$$

where h is the step size.

In addition to the discretization error, the Shooting method is sensitive to the estimation of the unknown initial condition. Some perturbation in the initial slope can propagate over the integration time interval and affect the end solution. But in well-posed and smooth problems, this sensitivity is kept under control, which results in highly accurate approximations.

3.2 Error in the Galerkin Method

In a finite-dimensional subspace of a set of chosen basis functions, the Galerkin method approximates the exact solution. Let $V_h \subset H^1$. On the exact solution and choice of basis functions of standard regularity, the error satisfies the estimate:

$$\|u - u_h\| \leq Ch^p \|u\|_{p+1}$$

where h represents the discretization parameter, p is the degree of the polynomial basis functions, and C is a constant independent of h. This estimate shows that the convergence of the Galerkin method is dependent on the smoothness of the exact solution as well as the ability of the selected basis functions to approximate.

3.3 Error in the Modified Galerkin Method

The Modified Galerkin method is a generalization of the standard Galerkin formulation, where the order of differentiation is lowered by integration by parts. This results in a more computationally efficient formulation, and can frequently increase numerical performance.

Theoretically, the Modified Galerkin method has the same convergence behavior as the standard Galerkin method. Nevertheless, in computations, it often has better accuracy since the residual effects are smaller and the derivative terms are more easily handled.

3.4 Remarks on Stability and Convergence

Although a full mesh refinement study is beyond the scope of the present work, the stability and convergence behavior of the numerical methods can be discussed based on the computed results and known theoretical properties. The error distributions for all test problems remain smooth and bounded across the domain, with no indication of numerical instability or divergence. This behavior is consistent with the known stability characteristics of the Shooting, Galerkin, and Modified Galerkin methods for well-posed and smooth boundary value problems. Moreover, the consistently small error magnitudes and their uniform distribution suggest that the methods exhibit the expected convergence behavior in practical computations.

4. Results and Discussion

The numerical solutions for four problems were computed using the Shooting method, the Galerkin method, and the modified Galerkin method. By comparing these solutions with the exact solutions, the corresponding errors were determined. Analysis of the RMS errors indicates the accuracy of the solutions. Values are shown up to 12 decimal places. Errors are computed using full machine precision. For all four boundary value problems considered in this study, a consistent finite element and spectral approximation framework is used. The Galerkin method employs piecewise linear Lagrange basis functions defined on a uniform mesh with step size $h = 0.5$, leading to a standard P_1 finite element space. The modified Galerkin method uses spectral basis functions constructed from shifted Legendre polynomials multiplied by a quadratic weight function $t(T - t)$, ensuring boundary compatibility. The number of basis functions is problem-dependent and chosen to balance accuracy and computational efficiency. This unified formulation is applied consistently across all test problems to ensure comparability of results.

Problem 1

Let $P(t)$ denote the population size of a certain species at time t (in years). The population follows the logistic growth model with intrinsic growth rate $r = 0.2$ (per year) and carrying capacity K = 10000. The population satisfies the first-order logistic equation

$$\frac{dP}{dt} = rP\left(1 - \frac{P}{K}\right)$$

By differentiating this equation with respect to time and substituting the original growth term, one obtains the following equivalent second-order non-linear differential equation:

$$\frac{d^2P}{dt^2} = r\left(1 - \frac{2P}{K}\right)\frac{dP}{dt}, 0 \leq t \leq 10$$

The population is subject to the two-point boundary conditions:

$$P(0) = 500, P(10) = \frac{10000}{1 + 19e^{-2}}$$

The objective is to determine the population after 25 years.

Table 1: Numerical solutions and their error estimation

t	Exact Solution	Shooting Method	Shooting Error	Galerkin Method	Galerkin Error	Modified Galerkin Method	Modified Galerkin Error
0.0	500.000000000000	500.000000000000	0.000000000000	500.000000000000	0.000000000000	500.000000000000	0.000000000000
0.5	549.694863365742	549.694863365742	0.000000000000	549.690203416120	0.004659949622	549.694808872538	0.000054493204
1.0	604.014851376858	604.014851376858	0.000000000000	604.007456708343	0.007394668515	604.014825621766	0.000025755092
1.5	663.325883653535	663.325883653536	0.000000000001	663.317883118478	0.008000535057	663.325970314010	0.000086660475
2.0	728.009691500527	728.009691500530	0.000000000003	728.003397696731	0.006293803796	728.009748508100	0.000057007573
2.5	798.461681518076	798.461681518078	0.000000000002	798.459560629556	0.002120888520	798.461627154124	0.000054363952
3.0	875.088081985363	875.088081985367	0.000000000004	875.092712022445	0.004630037083	875.087974367586	0.000107617777
3.5	958.302277324592	958.302277324596	0.000000000004	958.316293699198	0.014016374606	958.302222021949	0.000055302643
4.0	1048.520240775101	1048.520240775104	0.000000000003	1048.546268562576	0.026027787475	1048.520287475958	0.000046700857
4.5	1146.154986379124	1146.154986379129	0.000000000005	1146.195559222047	0.040572842923	1146.155099266801	0.000112887677
5.0	1251.609979983353	1251.609979983361	0.000000000008	1251.667446403236	0.057466419883	1251.610077256468	0.000097273115
5.5	1365.271476627242	1365.271476627247	0.000000000005	1365.347895529790	0.076418902548	1365.271492277803	0.000015650561
6.0	1487.499789642449	1487.499789642452	0.000000000003	1487.596818011451	0.097028369002	1487.499713525740	0.000076116709

t	Exact Solution	Shooting Method	Shooting Error	Galerkin Method	Galerkin Error	Modified Galerkin Method	Modified Galerkin Error
6.5	1618.619545819636	1618.619545819640	0.000000000004	1618.738322944245	0.118777124609	1618.619423742168	0.000122077468
7.0	1758.909041203891	1758.909041203897	0.000000000006	1759.050075188065	0.141033984173	1758.908943265910	0.000097937981
7.5	1908.588882560118	1908.588882560123	0.000000000005	1908.751946195369	0.163063635251	1908.588862484450	0.000020075668
8.0	2067.810178117853	2067.810178117857	0.000000000004	2067.994222296456	0.184044179603	2067.810245999761	0.000067881908
8.5	2236.642624138978	2236.642624138974	0.000000000004	2236.845717649580	0.203093510603	2236.642744305058	0.000120166080
9.0	2415.062915758573	2415.062915758564	0.000000000009	2415.282220332506	0.219304573933	2415.063027548448	0.000111789875
9.5	2602.943984412788	2602.943984412776	0.000000000012	2603.175773084948	0.231788672160	2602.944033344090	0.000048931302
10.0	2800.045621650739	2800.045621650734	0.000000000005	2800.285346722233	0.239725071494	2800.045585294261	0.000036356478
10.5	3006.007081178494	3006.007081178489	0.000000000005	3006.249495268343	0.242414089849	3006.006978240446	0.000102938048
11.0	3220.342248761577	3220.342248761573	0.000000000004	3220.581578689003	0.239329927426	3220.342128525341	0.000120236236
11.5	3442.437925684176	3442.437925684171	0.000000000005	3442.668094462737	0.230168778561	3442.437843939161	0.000081745015
12.0	3671.555681146765	3671.555681146762	0.000000000003	3671.770568586544	0.214887439779	3671.555674267122	0.000006879643
12.5	3906.837591648695	3906.837591648691	0.000000000004	3907.031319476111	0.193727827416	3906.837660587646	0.000068938951
13.0	4147.316005472132	4147.316005472124	0.000000000008	4147.483229087820	0.167223615688	4147.316116461579	0.000110989447
13.5	4391.927257779574	4391.927257779580	0.000000000006	4392.063444372339	0.136186592766	4391.927358846488	0.000101066914
14.0	4639.529031646793	4639.529031646801	0.000000000008	4639.630703842648	0.101672195856	4639.529077035592	0.000045388799
14.5	4888.920831622429	4888.920831622434	0.000000000005	4888.985757439029	0.064925816599	4888.920802823578	0.000028798851
15.0	5138.866830116855	5138.866830116864	0.000000000009	5138.894143698987	0.027313582133	5138.866743860763	0.000086256092
15.5	5388.120183403660	5388.120183403665	0.000000000005	5388.110426510198	0.009756893462	5388.120082967801	0.000100435859
16.0	5635.447810184569	5635.447810184566	0.000000000003	5635.402891079575	0.044919104994	5635.447744094296	0.000066090273
16.5	5879.654592596254	5879.654592596236	0.000000000018	5879.577666074348	0.076926521906	5879.654590844199	0.000001752055
17.0	6119.606000803019	6119.606000802984	0.000000000035	6119.501279695597	0.104721107422	6119.606060213348	0.000059410329
17.5	6354.248253462379	6354.248253462351	0.000000000028	6354.120767346689	0.127486115690	6354.248339792811	0.000086330432
18.0	6582.625295548307	6582.625295548278	0.000000000029	6582.480615935689	0.144679612619	6582.625361844212	0.000066295905
18.5	6803.892084842304	6803.892084842304	0.000000000000	6803.736037556829	0.156047285475	6803.892097019962	0.000012177658
19.0	7017.323908317883	7017.323908317895	0.000000000012	7017.162293101738	0.161615216145	7017.323864267052	0.000044050831
19.5	7222.321678762247	7222.321678762265	0.000000000018	7222.160013646103	0.161665116144	7222.321609495520	0.000069266727
20.0	7418.413371607367	7418.413371607379	0.000000000012	7418.256675750727	0.156695856640	7418.413322171104	0.000049436263
20.5	7605.251937524898	7605.251937524919	0.000000000021	7605.104561653479	0.147375871419	7605.251937687447	0.000000162549
21.0	7782.610158564812	7782.610158564843	0.000000000031	7782.475667401372	0.134491163440	7782.610201911672	0.000043346860
21.5	7950.373000659347	7950.373000659378	0.000000000031	7950.254107370984	0.118893288363	7950.373048752319	0.000048092972
22.0	8108.528054309780	8108.528054309815	0.000000000035	8108.426603352785	0.101450956995	8108.528067354677	0.000013044898
22.5	8257.154653278880	8257.154653278913	0.000000000033	8257.071645327595	0.083007951285	8257.154626218206	0.000027060674
23.0	8396.412225781929	8396.412225781934	0.000000000005	8396.347876745422	0.064349036507	8396.412195011430	0.000030770499
23.5	8526.528372923069	8526.528372923098	0.000000000029	8526.482198323461	0.046174599608	8526.528375121929	0.000002198860
24.0	8647.787093973539	8647.787093973557	0.000000000018	8647.758010036689	0.029083936850	8647.787113687604	0.000019714065
24.5	8760.517495748791	8760.517495748802	0.000000000011	8760.503929248884	0.013566499907	8760.517492038000	0.000003710791
25.0	8865.083240657419	8865.083240657417	0.000000000002	8865.083240657419	0.000000000000	8865.083240657419	0.000000000000

The graphical depiction of problem 1, derived from the computed solutions, is presented below:

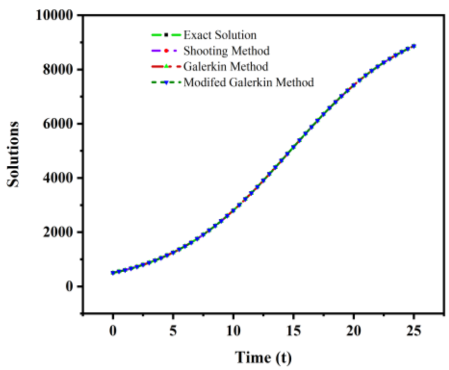

Figure 1. Numerical solution plot for problem 1

Figure 2 presents the plotted error for problem 1, derived from the obtained numerical data:

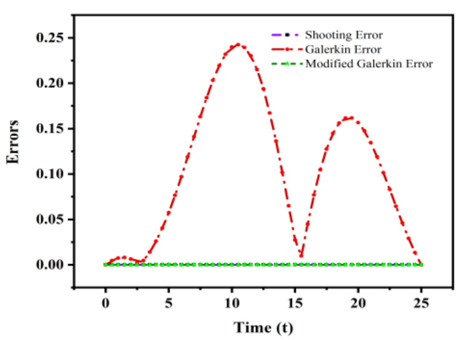

Figure 2. Error representation plot for problem 1

The numerical results in Table 1 are obtained using three numerical methods, the Shooting method, the Galerkin method, and the modified Galerkin method; each solution is compared against the exact solution. The data indicate that the Shooting method produces small errors compared to other methods, giving the least RMS error as 1.44×10^{-11} and therefore the highest accuracy. The modified Galerkin method provides low accuracy compared to the Shooting method, with RMS errors of 6.82×10^{-5}. In comparison, the Galerkin method has a comparatively high RMS error of 1.30×10^{-1} and the lowest accuracy. The Shooting method provides the least error as it converts the BVP problem into the IVP and subsequently employs the use of extremely accurate ODE solvers, in most cases, Runge-Kutta methods. The solvers are automatically high-order and adaptive, thus they follow the precise solution closely throughout the interval.

The modified Galerkin method is an improvement in the standard Galerkin method, which employs superior basis functions or weighted projection that minimize approximation error and reduce oscillations. This refinement enables it to model the non-linear behavior more precisely than the classical Galerkin method, so its RMS error is lower than the Galerkin but still larger than Shooting. The Galerkin method is the most common and is the one with the largest error, as it is based on an approximation space that is finite-dimensional. The Shooting method is proven to capture the whole growth curve effectively, including the inflection point. The Modified Galerkin method has only slightly higher errors in the accelerated growth phase, suggesting some difficulties in approximating the strong curvature with the chosen basis functions. The standard Galerkin method had the largest deviations in the same area, indicating that this method has some limitations in the case of moderate nonlinearity, when the number of basis functions is fixed.

In terms of computation time, the Shooting method is the quickest, requiring 6108.61 milliseconds. The modified Galerkin method takes slightly more time because of the extra corrections or projections, which is 7146.51 milliseconds. The Galerkin method is the most time-intensive in computation since it converts the differential equation into a very large system of simultaneous algebraic equations, needing to build and invert massive dense matrices. This method has a computational time of 42104.66 milliseconds. The most accurate and have the lowest RMS error are the Shooting method and the modified Galerkin method, which have an optimized balance between accuracy and computational efficiency.

Problem 2

Let $D(t)$ denote the concentration of a drug in the bloodstream (in mg/L) at time t (in hours). The drug follows a non-linear saturation-elimination model given by

$$\frac{dD}{dt} = -kD\left(1 - \frac{D}{D_{max}}\right)$$

where $k = 0.3$ is the elimination rate constant and $D_{max} = 80$ is the saturation concentration. By differentiating this equation, the system is equivalently described by the following second-order non-linear differential equation:

$$\frac{d^2D}{dt^2} = -k\left(1 - \frac{2D}{D_{max}}\right)\frac{dD}{dt}, 0 \le t \le 6.$$

The system satisfies the boundary conditions: $D(0) = 20, D(6) = \frac{80}{1+3e^{-1.8}}$. Determine the value of $D(15)$.

Table 2: Numerical solutions and their error estimation

t	Exact Solution	Shooting Method	Shooting Error	Galerkin Method	Galerkin Error	Modified Galerkin Method	Modified Galerkin Error
0.0	20.000000000000	20.000000000000	0.000000000000	20.000000000000	0.000000000000	20.000000000000	0.000000000000
0.5	22.333119004118	22.333119014116	0.000000009999	22.333212712818	0.000093708700	22.333115455778	0.000003548340
1.0	24.825795360990	24.825795370124	0.000000009134	24.825800462292	0.000005101302	24.825794808377	0.000000552607
1.5	27.464185840389	27.464185840389	0.000000000000	27.463897936421	0.000287903968	27.464191610272	0.000005769883
2.0	30.229347319157	30.229347325617	0.000000006460	30.228560952731	0.000786366426	30.229348913874	0.000001594717
2.5	33.097518229270	33.097518233979	0.000000004709	33.096051211863	0.001467017407	33.097512673848	0.000005555422
3.0	36.040760635228	36.040760635229	0.000000000000	36.038478181588	0.002282453640	36.040755014710	0.000005620518
3.5	39.027945645326	39.027945646009	0.000000000683	39.024780304482	0.003165340844	39.027946836176	0.000001190850
4.0	42.026018990788	42.026018989367	0.000000001420	42.021982722665	0.004036268123	42.026026198561	0.000007207774
4.5	45.001442753458	45.001442753458	0.000000000000	44.996628753454	0.004814000004	45.001449153869	0.000006400411
5.0	47.921682157107	47.921682151756	0.000000005351	47.916255916415	0.005426240692	47.921681853629	0.000000303478
5.5	50.756599254068	50.756599247052	0.000000007016	50.750780381390	0.005818872678	50.756592475212	0.000006778856
6.0	53.479629903399	53.479629903399	0.000000000000	53.473667887492	0.005962015907	53.479622449229	0.000007454170
6.5	56.068653243868	56.068653234393	0.000000009474	56.062801201122	0.005852042746	56.068651417633	0.000001826235
7.0	58.506506539017	58.506506528791	0.000000010226	58.500996922474	0.005509616543	58.506512021796	0.000005482779
7.5	60.781143744150	60.781143744151	0.000000000001	60.776169127159	0.004974616991	60.781152360850	0.000008616700
8.0	62.885475195812	62.885475184989	0.000000010823	62.881175916960	0.004299278852	62.885480285642	0.000005089830
8.5	64.816952978873	64.816952968142	0.000000010732	64.813412032850	0.003540946023	64.816950653215	0.000002325658
9.0	66.576979806351	66.576979806352	0.000000000001	66.574224205240	0.002755601111	66.576972063437	0.000007742914
9.5	68.170219796030	68.170219786049	0.000000009982	68.168226883233	0.001992912797	68.170213043906	0.000006752124
10.0	69.603880524913	69.603880515500	0.000000009412	69.602587428294	0.001293096619	69.603880456904	0.000000068008
10.5	70.887021131712	70.887021131714	0.000000000002	70.886335615011	0.000685516701	70.887027831381	0.000006699670
11.0	72.029924689352	72.029924681273	0.000000008079	72.029735971452	0.000188717900	72.029932153913	0.000007464562
11.5	73.043557364443	73.043557357067	0.000000007377	73.043745892602	0.000188528159	73.043558608161	0.000001243717
12.0	73.939123716706	73.939123716708	0.000000000002	73.939569298195	0.000445581489	73.939117758919	0.000005957787
12.5	74.727717588091	74.727717582080	0.000000006011	74.728305639862	0.000588051771	74.727711467558	0.000006120533
13.0	75.420061388416	75.420061383041	0.000000005375	75.420687327636	0.000625939220	75.420062883613	0.000001495197
13.5	76.026322766877	76.026322766879	0.000000000001	76.026894734249	0.000571967372	76.026329200992	0.000006434115
14.0	76.555996040901	76.555996036667	0.000000004234	76.556436237111	0.000440196210	76.555995676299	0.000000364602
14.5	77.017835688253	77.017835684518	0.000000003735	77.018080631534	0.000244943281	77.017831489182	0.000004199071
15.0	77.419830125436	77.419830125437	0.000000000002	77.419830125436	0.000000000000	77.419830125436	0.000000000000

The computed solutions are illustrated graphically. The graph for problem 2 is shown below:

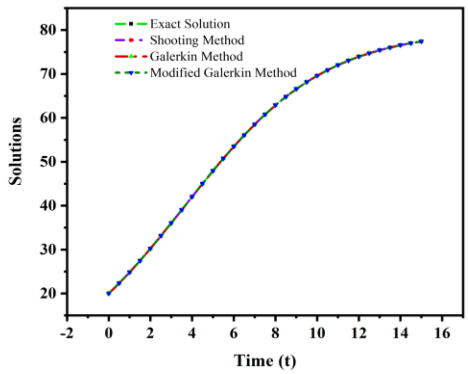

Figure 3. Numerical solution plot for problem 2.

Figure 4 presents the error profile for problem 2, as determined from the computational results.

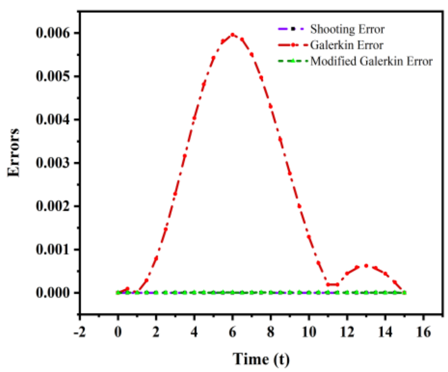

Figure 4. Error representation plot for problem 2.

In Table 2, all three numerical procedures give an approximation of the exact solution with relatively good precision, although their relative precision varies greatly when evaluated using the overall error indicators such as the RMS error. The Shooting method has given the smallest RMS error of 6.11×10^{-9} compared to the other two methods. Also, the modified Galerkin method provides an RMS error of 5.04×10^{-6}. Conversely, the standard Galerkin method has the largest RMS error, which is 3.07×10^{-3}.

The Shooting method is useful in converting the boundary value problem to an initial value problem and integrates the problem directly, using highly accurate ODE solvers, which reduces error accumulation. The modified Galerkin method provides better accuracy than the classical Galerkin method as it uses better basis functions or correction terms, and therefore has a smaller approximation error.

The standard Galerkin method, however, uses a small number of basis functions, which might not effectively represent the exact behavior of the solution, resulting in larger residual errors.

In this model, the Galerkin method, however, was found to increase the error in the initial transient before settling down close to the plateau. Such behavior is indicative of its sensitivity to regions of varying curvature. The Shooting method is performed with consistently low errors due to the high-order Runge-Kutta integration.

The Shooting method is generally the fastest in terms of computational time since its computational time is relatively low, 326.93 milliseconds to solve a system of ODEs. The Galerkin method with the modification involves supplementing the method with correction terms or higher-order basis functions, which require more time to compute as it adds to the size of matrices, the complexity of the integration, and the number of floating-point operations, which is 1559.59 milliseconds. The Galerkin method has the longest computational time, 17399.50 milliseconds, which is the most time-consuming method because it is used to assemble and solve a system of algebraic equations.

These variations in computation time are due to the trade-off between direct integration and system-based approximations and the complexity in the reformulated expression. Among these three methods, the Shooting method has the highest accuracy and the minimum computational time. The Galerkin method is less accurate than the modified Galerkin and Shooting method.

Problem 3

Let $C(t)$ denote the concentration of reactant A (in mol/L) at time t (in hours) in a batch reactor. The reaction follows a second-order kinetics model given by

$$\frac{dC}{dt} = -kC^2$$

where $k = 0.05$ L/mol·h is the rate constant.

By differentiating this equation, the system is equivalently described by the following second-order non-linear differential equation:

$$\frac{d^2C}{dt^2} = -2kC\frac{dC}{dt}, 0 \leq t \leq 10$$

The system satisfies the boundary conditions $C(0) = 2, C(10) = 1$.

The objective is to determine the concentration after 18 hours.

Table 3: Numerical solutions and their error estimation.

t	Exact Solution	Shooting Method	Shooting Error	Galerkin Method	Galerkin Error	Modified Galerkin Method	Modified Galerkin Error
0.0	2.000000000000	2.000000200000	0.000000200000	2.000020000000	0.000020000000	2.000000600000	0.000000600000
0.5	1.904761905000	1.904762111000	0.000000206000	1.904782857000	0.000020952500	1.904762528000	0.000000623500
1.0	1.818181818000	1.818182032000	0.000000214000	1.818203636000	0.000021818000	1.818182465000	0.000000647000
1.5	1.739130435000	1.739130657000	0.000000222000	1.739153810000	0.000023375000	1.739131122000	0.000000687000
2.0	1.66666666700	1.666666903000	0.000000236000	1.666691667000	0.0000250000000	1.666667467000	0.000000800000
2.5	1.60000000000	1.600000250000	0.000000250000	1.600027500000	0.000027500000	1.600001900000	0.000001900000
3.0	1.53846153800	1.538461804000	0.000000266000	1.538491538000	0.000030000000	1.538463738000	0.000002200000
3.5	1.48148148100	1.481481763000	0.000000282000	1.481514981000	0.000033500000	1.481484381000	0.000002900000
4.0	1.42857142900	1.428571727000	0.000000298000	1.428608929000	0.000037500000	1.428575329000	0.000003900000
4.5	1.37931034500	1.379310659000	0.000000314000	1.379352345000	0.000042000000	1.379315845000	0.000005500000
5.0	1.33333333300	1.333333633000	0.000000300000	1.333363333000	0.000030000000	1.333334833000	0.000001500000
5.5	1.29032258100	1.290322913000	0.000000332000	1.290367581000	0.000045000000	1.290330181000	0.000007600000
6.0	1.25000000000	1.250000350000	0.000000350000	1.250048000000	0.000048000000	1.250008200000	0.000008200000
6.5	1.21212121200	1.212121578000	0.000000366000	1.212172212000	0.000051000000	1.212130012000	0.000008800000
7.0	1.17647058800	1.176470970000	0.000000382000	1.176525588000	0.000055000000	1.176481988000	0.000011400000
7.5	1.14285714300	1.142857541000	0.000000398000	1.142916143000	0.000059000000	1.142869343000	0.000012200000
8.0	1.11111111100	1.111111525000	0.000000414000	1.111174111000	0.000063000000	1.111124911000	0.000013800000
8.5	1.08108108100	1.081081511000	0.000000430000	1.081148081000	0.000067000000	1.081097681000	0.000016600000
9.0	1.05263157900	1.052632025000	0.000000446000	1.052702579000	0.000071000000	1.052651179000	0.000019600000
9.5	1.02564102600	1.025641488000	0.000000462000	1.025716026000	0.000075000000	1.025663726000	0.000022700000
10	1.00000000000	1.000000400000	0.000000400000	1.000050000000	0.000050000000	1.000002400000	0.000002400000
10.5	0.97560975610	0.975610222100	0.000000466000	0.975689756100	0.000080000000	0.975633356100	0.000023600000
11.0	0.952380952400	0.952381434400	0.000000482000	0.952464952400	0.000084000000	0.952407652400	0.000026700000
11.5	0.93023255810	0.930233056100	0.000000498000	0.930320558100	0.000088000000	0.930262558100	0.000030000000
12.0	0.90909090910	0.909091423100	0.000000514000	0.909182909100	0.000092000000	0.909123309100	0.000032400000
12.5	0.88888888890	0.888889418900	0.000000530000	0.888984888900	0.000096000000	0.888923988900	0.000035100000
13.0	0.86956521740	0.869565763400	0.000000546000	0.869665217400	0.000100000000	0.869603217400	0.000038000000
13.5	0.85106382980	0.851064391800	0.000000562000	0.851167829800	0.000104000000	0.851104729800	0.000041000000
14.0	0.83333333330	0.833333911300	0.000000578000	0.833441333300	0.000108000000	0.833377033300	0.000043700000
14.5	0.81632653060	0.816327124600	0.000000594000	0.816438530600	0.000112000000	0.816373130600	0.000046600000

t	Exact Solution	Shooting Method	Shooting Error	Galerkin Method	Galerkin Error	Modified Galerkin Method	Modified Galerkin Error
15.0	0.80000000000	0.800000520000	0.000000520000	0.800065000000	0.000065000000	0.800003900000	0.000003900000
15.5	0.78431372550	0.784314335500	0.000000610000	0.784431725500	0.000118000000	0.784363725500	0.000050000000
16.0	0.76923076920	0.769231395200	0.000000626000	0.769352769200	0.000122000000	0.769283769200	0.000053000000
16.5	0.75471698110	0.754717623100	0.000000642000	0.754842981100	0.000126000000	0.754772981100	0.000056000000
17.0	0.74074074070	0.740741398700	0.000000658000	0.740870740700	0.000130000000	0.740799740700	0.000059000000
17.5	0.727272727300	0.727273401300	0.000000674000	0.727406727300	0.000134000000	0.727334727300	0.000062000000
18.0	0.71428571430	0.714286404300	0.000000690000	0.714423714300	0.000138000000	0.714350714300	0.000065000000

The obtained solutions have been represented graphically, and the solution for problem 3 is displayed in the following figure:

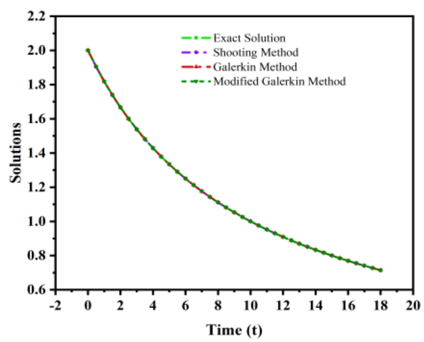

Figure 5. Numerical solution plot for problem 3.

The error corresponding to problem 3, obtained from the computations, is illustrated in the graph below:

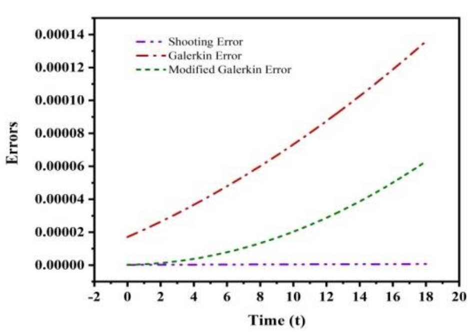

Figure 6. Error representation plot for problem 3

The results in Table 3 indicate that all three approaches, Shooting, Galerkin, and modified Galerkin, give good approximations to the exact solution across the entire domain, but a major difference is observed when the accuracy of the methods is assessed in terms of the RMS error. The RMS error in the Shooting method is 4.56×10^{-7}, whereas the Galerkin method has the largest RMS error, which tends to give the RMS error of 8.65×10^{-5}. The modified Galerkin method exhibits the RMS error of 2.29×10^{-5}, which is higher than the shooting method.

Based on root mean square errors, the Shooting method gives the highest accuracy, and the modified Galerkin method provides better accuracy than the Galerkin method. This difference in accuracy is due to the underlying differences in the numerical formulations. A unique strength of the Shooting method in this problem lies in its direct conversion of the second-order nonlinear BVP into a system of first-order IVPs, which are then integrated using the RK4 scheme. The solution provided by this model is smoothly decaying and strictly monotonic, so it is possible to propagate the solution forward in time using the Shooting method with very little error accumulation and high accuracy. The Galerkin and modified Galerkin methods are global, with polynomial basis functions, and this results in a consistent error in the approximation of the solution, even in this relatively simple problem of decay.

The Shooting Method is typically the most efficient in terms of computational time as well, 18.56 milliseconds, because it uses direct ODE solvers whose computational complexity is relatively simple. The Galerkin method involves a collection of algebraic equations that have to be assembled and solved, which takes computational time of 3644.75

milliseconds. The modified Galerkin method has a computational time of 63.50 milliseconds due to the extra alterations, including higher-order terms or more complicated integrations, which increase the size of the matrices and the number of floating-point operations.

Problem 4

In a closed population, the spread of an infectious disease is modelled by the SIR system:

$$\frac{dS}{dt} = -\beta SI, \frac{dI}{dt} = \beta SI - \gamma I,$$

Where,

S(t): susceptible population

$I(t)$: infected population

β = 0.002, γ = 0.5

Assume the total population is constant:

$$S(t) + I(t) = N = 500$$

After reducing to a single non-linear equation, using $S = 500 - I$, we get,

$$\frac{dI}{dt} = \beta(500 - I)I - \gamma I = I(100\beta - \beta I - \gamma)$$

Then substituting the values,

$$\frac{dI}{dt} = I(1 - 0.002I - 0.5) = I(0.5 - 0.002I)$$

Converting to second-order non-linear BVP

$$\frac{d^2I}{dt^2} = \frac{dI}{dt}(0.5 - 0.004I)$$

With boundary condition:

$$I(0) = 10,\ I(10) = \frac{250}{1+24e^{-5}}.$$

Determine the value of $I(15)$.

Table 4: Numerical solutions and their error estimation.

t	Exact Solution	Shooting Method	Shooting Error	Galerkin Method	Galerkin Error	Modified Galerkin Method	Modified Galerkin Error
0.0	10.000000000000	10.000000000000	0.000000000000	10.000000000000	0.000000000000	10.000000000000	0.000000000000
0.5	12.696014534144	12.696027140811	0.000012606667	12.696014534144	0.001102971937	12.695318972690	0.000695561454
1.0	16.070209244540	16.070235790871	0.000026546331	16.070209244540	0.006335466837	16.071039644272	0.000830399732
1.5	20.264578773080	20.264623830435	0.000045057355	20.264578773080	0.001085779101	20.266605331882	0.002026558802
2.0	25.434661610273	25.434734443059	0.000072832786	25.434661610273	0.007606258830	25.435190752926	0.000529142653
2.5	31.741537045052	31.741652195690	0.000115150638	31.741537045052	0.004944179641	31.738953546034	0.002583499018
3.0	39.338336459524	39.338512144638	0.000175685114	39.338336459524	0.003278224613	39.334242569899	0.004093889626
3.5	48.350525283995	48.350778248889	0.000252964894	48.350525283995	0.008430043278	48.348708379222	0.001816904773
4.0	58.850575783819	58.850912362689	0.000336578870	58.850575783819	0.005656466042	58.853692310074	0.003116526255
4.5	70.829929221860	70.830335017578	0.000405795718	70.829929221860	0.002354454788	70.836624531590	0.006695309730
5.0	84.173951455187	84.174385399150	0.000433943963	84.173951455187	0.008273564077	84.179088671985	0.005137216797
5.5	98.647788635014	98.648188312794	0.000399677780	98.647788635014	0.006697362455	98.646302129872	0.001486505142
6.0	113.900943058625	113.901243745660	0.000300687034	113.900943058625	0.000988559491	113.892697556597	0.008245502029
6.5	129.494696963735	129.494857324175	0.000160360440	129.494696963735	0.007576896431	129.485974617958	0.008722345777
7.0	144.949618811747	144.949638706881	0.000019895135	144.949618811747	0.006522474151	144.948601819339	0.001016992408
7.5	159.803030564140	159.802949276580	0.000081287560	159.803030564140	0.001276394873	159.811718767761	0.008688203621
8.0	173.662325417712	173.662201462001	0.000123955711	173.662325417712	0.007155625387	173.672317651758	0.009992234046
8.5	186.241711816789	186.241597484788	0.000114332001	186.241711816789	0.003440949493	186.241092444803	0.000619371986
9.0	197.376329051944	197.376253868331	0.000075183613	197.376329051944	0.004920721885	197.365902539656	0.010426512287
9.5	207.015218293924	207.015186818725	0.000031475198	207.015218293924	0.002046230373	207.014520135593	0.000698158331
10.0	215.199872032794	215.19985058722	0.000021445567	215.19759337845	0.002278654340	215.19518297636	0.004689056433
10.5	222.036611145350	222.036624611882	0.000013466533	222.036611145350	0.024860950199	221.994149852825	0.042461292524

t	Exact Solution	Shooting Method	Shooting Error	Galerkin Method	Galerkin Error	Modified Galerkin Method	Modified Galerkin Error
11.0	227.669593763240	227.669605246701	0.000011483461	227.669593763240	0.045087059488	227.592595232601	0.076998530639
11.5	232.258528384754	232.258528854129	0.000000469375	232.258528384754	0.061167548044	232.154076421340	0.104451963414
12.0	235.962574137856	235.962561199516	0.000012938340	235.962574137856	0.073704457342	235.836720021577	0.125854116279
12.5	238.930151230864	238.930127410331	0.000023820553	238.930151230864	0.083315746422	238.787887978220	0.142263252663
13.0	241.293516873310	241.293487401745	0.000029471566	241.293516873310	0.090576506660	241.138854036351	0.154662836959
13.5	243.166744350073	243.166715278029	0.000029072044	243.166744350073	0.095989528656	243.002832743017	0.163911607056
14.0	244.645883931889	244.645860861087	0.000023070801	244.645883931889	0.099975723841	244.475156239282	0.170727692607
14.5	245.810362622875	245.810350034390	0.000012588485	245.810362622875	0.102876440568	245.634669630857	0.175692992019
15.0	246.724966653252	246.724966653252	0.000000000000	246.724966653252	0.000000000000	246.724966653252	0.000000000000

The computed solutions have been visualized in a graph, with problem 4 displayed in the following figure:

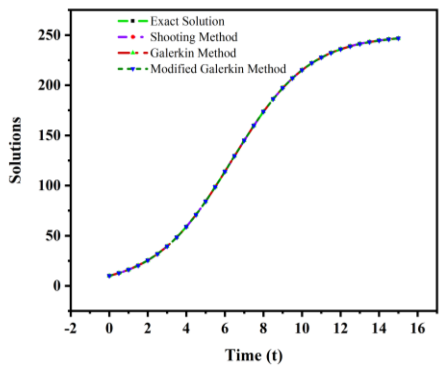

Figure 7. Numerical solution plot for problem 4.

A graphical representation of all obtained errors has been created, with problem 4 illustrated below:

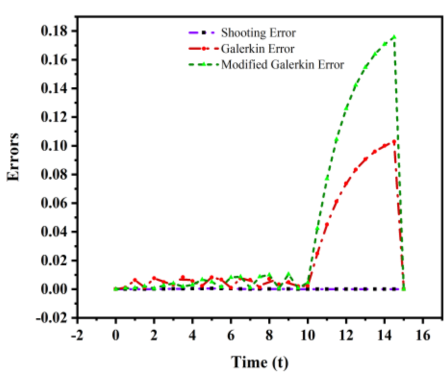

Figure 8. Error representation plot for problem 4.

The numerical data represented in Table 4 shows that all three techniques: Shooting, Galerkin, and modified Galerkin, yield solutions that are close to the exact solution throughout the domain. But with a more accurate comparison in terms of the root mean square (RMS) error, it can be seen that they differ significantly in their accuracy. The Shooting technique has the lowest RMS error of 1.71×10^{-4}, thus pointing to higher accuracy and numerical stability. The modified Galerkin method has a significant RMS error of 4.69×10^{-2}. The Galerkin method exhibits the greatest RMS error of 7.99×10^{-2} as compared to the other two.

This model showed the highest level of non-linearity of all four test problems. The standard Galerkin method had an enormous problem around the epidemic peak, where the curvature changed rapidly. Larger errors in this region show the ability of the classical Galerkin approach to capture strong nonlinear interactions. Compared to the other methods, the Shooting method proved to be robust and accurate even in this complex example, which illustrates very well the robustness of the Shooting method for highly nonlinear epidemiological BVPs.

The Shooting method is generally less computationally time-consuming because it is fairly simple to implement and directly integrates, and takes time of 111.24 milliseconds. However, the Galerkin method, instead, consists of the construction and resolution of a set of algebraic equations, which adds the highest time of computation, 9041.68 milliseconds. The modified Galerkin method is the one that has the computational time of 136.46

milliseconds, since it adds more correction terms and higher order approximations to the system, making it more complex. The disparities in both RMS error and computational time are a result of the trade-off between the simplicity of the algorithm, approximation strategy, and the numerical stability of each approach.

5. Conclusion

According to the outcomes of this study, it can be concluded that among the three methods, the Shooting method provides the highest accuracy, which is the most suitable method when maximum precision is required. In certain graphs, the solutions derived from the three methods closely match the exact solution, leading to overlapping curves in the plotted graphs. The modified Galerkin method can be used as an alternative to the Shooting method when a slightly lower accuracy is acceptable, especially if computational cost needs to be reduced.

The modified Galerkin method offers a more accurate solution than the Galerkin method and can be considered a balanced option between accuracy and computational efficiency. In contrast, the standard Galerkin method produces the largest error.

In terms of computational time, the Shooting method is the most efficient among the three as it transforms the boundary value problem into an equivalent initial value problem, which is then solved using highly accurate ODE solvers. The Modified Galerkin method requires slightly higher computational time due to additional projection or correction steps, while still maintaining a good level of accuracy. In contrast, the standard Galerkin method is significantly more time-consuming, primarily due to the formation and solution of large systems of algebraic equations, making it less efficient overall.

6. Limitations

Although this study has been useful, there are a number of limitations to note. Firstly, the current work is dedicated to the use and comparative analysis of available numerical methods, as opposed to the development of new methodologies. The Shooting, Galerkin, and Modified Galerkin methods used in this analysis are standard methods, and no novel numerical scheme is presented. Second, a limited set of smooth and well-posed problems involving boundary values is analyzed, which might not be a complete representation of more complex real-world systems with strong nonlinearity, discontinuities, or stiffness. This has led to a lack of exploration of the performance of the considered methods under these difficult conditions.

Third, despite the error behavior being addressed on theoretical grounds and numerical findings, the research lacks rigorous mathematical demonstrations of convergence, in-depth stability analysis, and formal derivations of error estimates of the particular methods used to address the specified BVPs. Rather, the analysis is based on known theoretical properties and numerical behavior. Fourth, both Galerkin and Modified Galerkin methods are based on certain selections of basis functions, and this can be restrictive in their ability to approximate some types of problems. Possibly, better accuracy could be achieved using a different choice of basis or a higher-order approximation space. One of the future directions is to come up with better or hybrid numerical methods to improve the accuracy and efficiency of the current numerical methods, like the Shooting, Galerkin, and Modified Galerkin methods. This can involve the construction of adaptive schemes, higher-order formulations, or problem-specific changes to more complicated boundary value problems. The other significant line is the inclusion of rigorous mathematical analysis, with formal proofs of convergence, detailed analysis of stability, and derivation of error bounds that are specific to the methods used to apply the methods to the specified classes of BVPs. This kind of analysis would present more solid theoretical support and supplement the numerical data on the same in this study. Moreover, further analysis of more complex and non-linear boundary value problems, such as stiff systems and models that are the results of complex real-world applications, would also be used as an indication of the strength and usability of the techniques. Lastly, the discussion of higher-dimensional problems and exploration of more basis functions and their use would yield more insight into the practical abilities and constraints of this numerical method.

References

Al-Abadi AK, Abd SK. 2024. Galerkin finite element methods error estimates analysis for [1+2] equations of integro-differential on linear triangular. Baghdad Sci J (in press).

Almutairi SSA, Saeed AM. 2025. A comparative study of finite difference and Galerkin finite element methods for solving boundary value problems. Eur J Pure Appl Math 18: 5895.

Ahmed HM. 2024. New generalized Jacobi polynomial Galerkin operational matrices of derivatives: an algorithm for solving boundary value problems. Fractal Fract 8: 199.

Ahmed HM. 2025. New Jacobi Galerkin operational matrices of derivatives: a highly accurate method for solving two-point fractional-order non-linear boundary value problems with Robin boundary conditions. Fractal Fract 9: 686.

Angadi LM. 2025. Wavelet based Galerkin method for the numerical solution of singular boundary value problems using

Fibonacci wavelets. J Sci Res 17: 227–234.

Abdelhakem M, Fawzy M, El-Kady M, Moussa H. 2022. An efficient technique for approximated BVPs via the second derivative Legendre polynomials pseudo-Galerkin method: certain types of applications. Results Phys 43: 106067.

Ahmad N, Charan S. 2019. A comparative study of numerical solutions of second order ordinary differential equations with boundary value problems by shooting method and finite difference method. Int J Stat Appl Math 4: 18–22.

Bizzarri F, Brambilla A, Codecasa L. 2018. Shooting by a two-step Galerkin method. IEEE Trans Circuits Syst I Regul Pap 66: 383–390.

Cicelia JE. 2014. Solution of weighted residual problems by using Galerkin's method. Indian J Sci Technol 7: 52–54.

Deka B, Kumar N. 2020. Error estimates in weak Galerkin finite element methods for parabolic equations under low regularity assumptions. Appl Numer Math 162: 81–105.

Eichmeir P, Steiner W. 2026. A double shooting method for two-point boundary value problems in multibody dynamics. Multibody Syst Dyn (in press).

Hossan MS, Datta T, Islam MS. 2024. Galerkin-finite difference method for fractional parabolic partial differential equations. MethodsX 12: 102763.

Javeed S, Hincal E. 2024. Solving coupled non-linear higher order BVPs using improved shooting method. Int J Math Comput Eng 2: 165–178.

Kant K, Das P, Nelakanti G, Kumar R. 2024. Modified Galerkin method for derivative dependent Fredholm–Hammerstein integral equations of second kind. Adv Appl Math Mech 16: 905–926.

Kheybari S, Darvishi MT, Alizadeh F, Boulaaras SM, Hosseini K. 2025. A modified scheme to the multiple shooting method for boundary value problems. Alex Eng J 118: 649–663.

Lund K, Palitta D. 2024. Low-rank-modified Galerkin methods for the Lyapunov equation. Electron Trans Numer Anal 62: 1–21.

Luo M, Xu D, Pan X. 2024. Sinc-Galerkin method and a higher-order method for 1D and 2D time-fractional diffusion equations. Bound Value Probl 2024: 1.

Mostoufi N, Constantinides A. 2023. Applied Numerical Methods for Chemical Engineers. Academic Press: New York.

Mitsotakis D, Synolakis C, McGuinness M. 2016. A modified Galerkin/finite element method for the numerical solution of the Serre–Green–Naghdi system. Int J Numer Methods Fluids 83: 755–778.

Pandey PK. 2010. Finite difference method for a second-order ordinary differential equation with a boundary condition of the third kind. Comput Methods Appl Math 10: 109–116.

Ruman U, Islam MS. 2024. Legendre-weighted residual methods for system of fractional order differential equations. J Appl Math Phys 12: 3163–3184.

Shakib MSN, Roy N, Datta K, Saha BK. 2026. A study on the numerical accuracy of Galerkin, modified Galerkin, shooting and homotopy perturbation methods in solving boundary value problems. Int J Math Stat Comput Sci 4: 476–489.

Singh J. 2022. Shooting method for solving two-point boundary value problems in ODEs numerically. arXiv:2208.13221.

Sun Z, Ling L. 2022. A kernel-based meshless conservative Galerkin method for solving Hamiltonian wave equations. SIAM J Sci Comput 44: A2789–A2807.

Toprakseven S, Dinibutun S. 2024. A weak Galerkin finite element method for parabolic singularly perturbed convection-diffusion equations on layer-adapted meshes. Electron Res Arch 32: 5033–5066.

Toprakseven S, Gao F. 2023. A modified weak Galerkin finite element method for singularly perturbed parabolic convection-diffusion-reaction problems. J Comput Math 41: 1246–1280.

Thomée V. 2006. The standard Galerkin method. In Lecture Notes in Mathematics. Springer: Berlin; 1–16.

Xie S, Han M, Zhu P. 2020. A posteriori error estimate of weak Galerkin FEM for second order elliptic problem with mixed boundary condition. Discrete Contin Dyn Syst Ser B 26: 5217–5226.

Xue S, Yin C, Li J, Zhu J, Liu W. 2025. A high efficiency discontinuous Galerkin method for 3D ground-penetrating radar simulation. Remote Sens 17: 228.

Zampieri E, Pavarino LF. 2023. A numerical comparison of Galerkin and collocation isogeometric approximations of acoustic wave problems. Appl Numer Math 200: 453–465.

Zhang L, Appelö D, Hagstrom T. 2021. Energy-based discontinuous Galerkin difference methods for second-order wave equations. Commun Appl Math Comput 4: 855–879.

Appendix A

A.1 Algorithm for Shooting Method

The algorithm for the shooting method can be described and presented as follows (Ahmad and Charan 2019):

i. Define the BVP

Consider

$$y''(x) + f(x, y, y') = 0, \; y(a) = \alpha, y(b) = \beta.$$

ii. Transform into IVPs

Introduce auxiliary functions $u(x)$ and $v(x)$ such that:

$$u(a) = \alpha, \quad u'(a) = 1,$$
$$v(a) = 0, \quad v'(a) = 1.$$

iii. Solve the IVPs

Numerically integrate both IVPs using a standard method (e.g., Runge-Kutta).

iv. Construct the General Solution

The solution of the original BVP can be expressed as

$$y(x) = u(x) + \lambda v(x)$$

where λ is chosen such that the boundary condition, $y(b) = \beta.$ is satisfied.

A.2 Algorithm for Galerkin Method

The procedure of the Galerkin method is outlined and formulated as follows (J. E. Cicelia 2014):

i. Formulate the BVP

Consider

$$y''(x) + f(x, y, y') = 0, \quad y(a) = \alpha, y(b) = \beta.$$

ii. Assume a Trial Solution

Choose

$$y(x) \approx \tilde{y}(x) = \sum_{i=1}^{n} c_i \emptyset_i(x),$$

where $\phi_i(x)$ satisfy the boundary conditions.

iii. Formulate the Residual

$$R(x) = \tilde{y}''(x) + f(x, \tilde{y}, \tilde{y}').$$

iv. Apply the Galerkin Condition

For each basis function $\phi_j(x)$, enforce

$$\int_a^b R(x)\phi_j(x)dx = 0, \quad j = 1,2, \dots, n$$

v. Solve the Algebraic System

The above equations yield a system of equations in the unknowns. c_i. Solving this system provides the coefficients, hence the approximate solution.

A.3 Algorithm for Modified Galerkin Method

The modified Galerkin method is expressed through the following algorithmic steps:

i. Formulate the BVP

Consider

$$y''(x) + f(x, y, y') = 0, \quad y(a) = \alpha, y(b) = \beta.$$

ii. Assume a Trial Solution

Select

$$y(x) \approx \tilde{y}(x) = \sum_{i=1}^{n} c_i \emptyset_i(x),$$

Where, $\emptyset_i(x)$ satisfies boundary conditions.

iii. Construct the Residual

$$R(x) = \tilde{y}''(x) + f(x, \tilde{y}, \tilde{y}').$$

iv. Apply the Modified Galerkin Condition

Multiply the residual by $\phi_j(x)$, integrate by parts to reduce differentiation, and enforce.

$$\int_a^b \tilde{y}'(x)\emptyset'_j(x)dx + \int_a^b f(x, \tilde{y}, \tilde{y}')\emptyset_j(x)dx = 0, \quad j = 1,2, \dots, n.$$

v. Solve the Resulting System

The reduced-order algebraic system is solved for the coefficients. c_i, giving the approximate solution $\tilde{y}(x)$.