HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

EFFECTIVE NUMERICAL METHODS FOR SOLVING NONLINEAR DIFFERENTIAL EQUATIONS

Achilova Shoira Oltiboyevna
Tashkent City, Shaykhantahur District
Republican Specialized Art School (RIRM)
First-Category Teacher of Mathematics and Computer Science
doi:https://doi.org/10.5281/zenodo.22863694

Abstract. Objective: to identify efficient numerical strategies for nonlinear differential equations by relating accuracy, stability, computational cost, and stiffness to the structural properties of the problem. Methods: comparative analysis of explicit and implicit time-stepping schemes, including explicit Euler, Heun's method, classical fourth-order Runge–Kutta, backward Euler with Newton iteration, and the role of adaptive and multistep approaches. Two reproducible nonlinear initial-value problems were used: a logistic equation with a closed-form solution and a stiff nonlinear relaxation model. Results: for the non-stiff logistic problem, RK4 produced the smallest final-time error among the tested fixed-step methods, while Heun provided a useful cost–accuracy compromise. For the stiff problem, large-step explicit formulas generated severe oscillatory or inaccurate behaviour, whereas backward Euler remained stable and converged toward the reference solution. A method-selection framework was developed that links stiffness, required tolerance, smoothness, and computational budget with the numerical scheme. Conclusion: numerical efficiency cannot be reduced to formal order alone; an effective solver is one whose stability region, error control, and nonlinear algebraic treatment are consistent with the equation's dynamics.

Keywords: nonlinear differential equation, numerical method, Runge–Kutta method, backward Euler method, stiffness, Newton iteration, error analysis, stability, computational efficiency, adaptive step size.

INTRODUCTION

Nonlinear differential equations constitute a central mathematical language for processes in mechanics, transport, biology, chemical kinetics, control, economics, and engineering. In contrast to linear equations, nonlinear models may display multiple equilibria, sharp transitions, saturation, self-excitation, finite-time sensitivity, or stiffness. Exact solutions are available only for restricted classes, so reliable computation requires numerical approximation supported by error and stability analysis. Uzbek textbooks on numerical analysis consistently place approximate solution of ordinary and partial differential equations among the core applications of computational mathematics (1, 2, 3, 4, 5).

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

The difficulty is not simply to obtain a sequence of numbers. A numerical method should reproduce the qualitative and quantitative behaviour of the underlying model within a prescribed tolerance and at an acceptable computational cost. The basic workflow includes problem classification, discretization, construction of an iterative or time-stepping process, control of truncation and round-off errors, and verification of the computed solution. Uzbek methodological literature emphasizes that the choice of algorithm must be linked to the mathematical structure of the problem rather than treated as a mechanical application of formulas (6, 7, 8).

For nonlinear differential equations, method selection is especially important. Explicit formulas are inexpensive per step and convenient for smooth non-stiff problems, but their stability restrictions can force very small time increments. Implicit formulas require the solution of a nonlinear algebraic equation at each step, commonly by Newton iteration, yet their larger stability regions make them attractive for stiff systems. High-order Runge–Kutta methods reduce the discretization error rapidly for sufficiently smooth solutions, whereas adaptive algorithms redistribute computational effort by reducing the step size only where the solution requires it. These trade-offs are developed in classical numerical ODE theory and remain fundamental in modern solver design (13, 14, 15).

National teaching resources also show an increasing integration of numerical computation with mathematical modelling, computer algebra, and software implementation. The works of Noraliyev and Yusupov, Xurramov, Ravshanov, and Xolmatov and Taylaqov connect numerical procedures with model construction and computer experimentation, while Durdiyev extends computational reasoning to partial differential equations and boundary-value problems (8, 9, 10, 11, 12). Such integration is essential because the efficiency of a numerical method depends not only on local accuracy but also on the model scale, parameter values, conditioning, and stiffness.

The purpose of this study is to systematize effective numerical methods for nonlinear differential equations and to formulate a practical selection strategy based on order of accuracy, stability, computational work, and equation stiffness. The research tasks are: (i) to compare the mathematical properties of commonly used explicit and implicit schemes; (ii) to conduct reproducible numerical tests on representative nonlinear equations; (iii) to identify the circumstances in which formal order is or is not a reliable indicator of efficiency; and (iv) to propose an algorithmic decision framework suitable for research and higher-education practice.

MATERIALS AND METHODS

The general initial-value problem is written as $y' = f(t, y)$, $y(t0) = y0$, where f may be nonlinear in y. On a uniform mesh $t(n) = t0 + n\,h$, a one-step method constructs $y(n+1)$ from information available at $t(n)$ and intermediate stages. The explicit Euler formula is $y(n+1) = y(n) + h\,f(t(n), y(n))$. Its principal advantages are simplicity and low cost, but its global error is first order and the admissible step size can be strongly restricted by stability. Heun's method applies an Euler predictor and trapezoidal correction, producing second-order accuracy while retaining an explicit structure (1, 3, 4).

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

The classical fourth-order Runge–Kutta scheme uses four function evaluations per step and has global order four for sufficiently smooth solutions. Its wide use is explained by a favourable balance between accuracy, implementation simplicity, and robustness for non-stiff problems. However, a high order does not remove stiffness restrictions: if the stability domain does not contain the scaled eigenvalues of the local linearization, reducing truncation error alone cannot prevent spurious oscillations or numerical divergence (13, 14).

For stiff nonlinear equations, the backward Euler scheme $y(n+1) = y(n) + h\, f(t(n+1), y(n+1))$ is attractive because of its strong stability properties. The unknown $y(n+1)$ appears implicitly and therefore requires the solution of a nonlinear equation $G(z)=z-y(n)-h\, f(t(n+1),z)=0$. In this study, Newton iteration $z(k+1) = z(k) - G(z(k))/G'(z(k))$ was used until the correction was below 1e-14. The cost of this inner iteration is the price paid for increased stability. The same principle extends to implicit Runge–Kutta and backward differentiation formula (BDF) families used in professional stiff solvers (6, 13, 15).

Adaptive time stepping was analysed conceptually as a complementary strategy. Embedded Runge–Kutta pairs estimate the local error from two formulas of different order and update the step size according to the ratio between the target tolerance and the estimated error. This approach is especially efficient when a solution contains intervals of slow evolution separated by short regions of rapid change. In contrast, fixed-step formulas spend the same computational effort in all regions. Uzbek textbooks on computation and modelling increasingly stress error estimation and algorithmic selection as integral components of numerical practice (2, 5, 8, 9).

Table 1. Comparative characteristics of numerical methods for nonlinear differential equations

Method	Global order	Per-step work	Stability profile	Nonlinear solve	Recommended use
Explicit Euler	1	1 f-eval	Small stability region	No	Basic, non-stiff, small h
Heun / RK2	2	2 f-evals	Larger than Euler	No	Moderate accuracy, non-stiff
Classical RK4	4	4 f-evals	Good for non-stiff	No	Smooth non-stiff IVPs
Adaptive RK	4–5 typical	Variable	Tolerance-controlled	No	Non-stiff problems with variable scales
Backward Euler	1	f +	A-stable	Yes	Stiff nonlinear

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

Method	Global order	Per-step work	Stability profile	Nonlinear solve	Recommended use
		Newton			IVPs
BDF family	1–5	History + Newton	Designed for stiffness	Yes	Large stiff systems

Two computational experiments were used. Test A was the logistic equation y' = 3y(1-y), y(0)=0.1 on 0 <= t <= 2. The exact solution y(t)=1/(1+9 exp(-3t)) allowed direct calculation of the final-time absolute error. Fixed steps h=0.2, 0.1, and 0.05 were used for explicit Euler, Heun, classical RK4, and backward Euler. This problem is smooth and non-stiff; therefore it primarily measures the effect of order and step refinement.

Test B was the nonlinear relaxation equation y' = -40(y-y^3), y(0)=0.5 on 0 <= t <= 1. The model rapidly approaches the stable equilibrium y=0 and has a fast time scale that exposes stability restrictions of explicit methods. A high-resolution RK4 solution with h=1e-4 was used as the reference. The same four fixed-step schemes were tested for h=0.1, 0.05, and 0.025. The purpose was not to rank all numerical solvers universally, but to demonstrate how stiffness changes the meaning of computational efficiency.

Diagram 1. Editable decision framework for selecting a numerical method

Problem	→	Stiffness test	→	Accuracy demand	→	Method family
Smooth non-stiff	→	No severe scale separation	→	Medium / high	→	**Heun, RK4, adaptive RK**
Stiff / fast-slow	→	Explicit step strongly restricted	→	Stable integration	→	**Backward Euler, BDF, implicit RK**

RESULTS

The non-stiff experiment confirms the expected convergence hierarchy. At h=0.1, the final-time absolute errors were approximately 4.95e-3 for explicit Euler, 1.22e-3 for Heun, 3.96e-6 for RK4, and 4.34e-3 for backward Euler. Halving the step reduced the Euler-type errors approximately linearly and the Heun error approximately quadratically, while RK4 showed a much more rapid reduction. The implicit method was stable but not competitive in accuracy for this smooth non-stiff problem because its first-order truncation error dominated and each step additionally required Newton iteration.

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

Table 2. Final-time errors for the nonlinear logistic equation, T = 2

h	Method	Numerical y(T)	Exact y(T)	Absolute error
0.20	Explicit Euler	0.988942028	0.978178051	1.076e-02
0.20	Heun (RK2)	0.972321809	0.978178051	5.856e-03
0.20	Classical RK4	0.978104599	0.978178051	7.345e-05
0.20	Backward Euler	0.969882562	0.978178051	8.295e-03
0.10	Explicit Euler	0.983126903	0.978178051	4.949e-03
0.10	Heun (RK2)	0.976962172	0.978178051	1.216e-03
0.10	Classical RK4	0.978174093	0.978178051	3.958e-06
0.10	Backward Euler	0.973837296	0.978178051	4.341e-03
0.05	Explicit Euler	0.980562060	0.978178051	2.384e-03
0.05	Heun (RK2)	0.977894278	0.978178051	2.838e-04
0.05	Classical RK4	0.978177820	0.978178051	2.315e-07
0.05	Backward Euler	0.975945460	0.978178051	2.233e-03

Diagram 2. Relative accuracy at h = 0.1 for Test A (longer colored bar = higher accuracy)

Method	Error	1	2	3	4	5	6	7	8
Explicit Euler	4.95e-03								
Heun (RK2)	1.22e-03								
Classical RK4	3.96e-06								
Backward Euler	4.34e-03								

The stiff relaxation problem produced a different picture. With h=0.1, explicit Euler converged to the spurious value -1 instead of the reference value near zero, and RK4 also generated a large error of about 3.49e-1. Heun reduced the instability but still produced an error close to 3.94e-2. In contrast, backward Euler yielded approximately 5.16e-8, showing that a low-order implicit method can be far

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

more effective than a higher-order explicit formula when stiffness dominates the computation. Reducing the step to h=0.025 restored accurate behaviour for the explicit methods, but at the cost of four times as many steps as h=0.1.

Table 3. Stability-sensitive results for the nonlinear stiff relaxation problem, T = 1

h	Method	Numerical y(T)	Reference y(T)	Absolute error
0.100	Explicit Euler	-1.000000000	2.453e-18	1.000e+00
0.100	Heun (RK2)	0.039350919	2.453e-18	3.935e-02
0.100	Classical RK4	-0.348584169	2.453e-18	3.486e-01
0.100	Backward Euler	0.000000052	2.453e-18	5.164e-08
0.050	Explicit Euler	0.101148639	2.453e-18	1.011e-01
0.050	Heun (RK2)	0.109600280	2.453e-18	1.096e-01
0.050	Classical RK4	0.000000000	2.453e-18	1.043e-10
0.050	Backward Euler	0.000000000	2.453e-18	1.466e-10
0.025	Explicit Euler	0.000000000	2.453e-18	2.453e-18
0.025	Heun (RK2)	0.000000000	2.453e-18	4.565e-13
0.025	Classical RK4	0.000000000	2.453e-18	2.752e-18
0.025	Backward Euler	0.000000000	2.453e-18	4.759e-13

The combined results indicate that efficiency must be evaluated by a work–precision–stability balance. On the smooth logistic problem, RK4 achieved several orders of magnitude smaller error than first-order formulas for the same step size. On the stiff problem, however, stability was the primary requirement: a method with excellent non-stiff accuracy could be ineffective at a large step if its stability region was incompatible with the fast decay scale. Thus, the most efficient method is not necessarily the method with the highest formal order; it is the method that reaches the requested tolerance with the least total computational work while preserving the dynamics of the problem.

DISCUSSION

The findings are consistent with the classical distinction between non-stiff and stiff differential equations. For smooth non-stiff problems, explicit Runge–Kutta methods are usually preferred because they avoid nonlinear algebraic solves and can obtain high accuracy with a moderate number of function evaluations. The logistic test illustrates this clearly: RK4 outperformed both Euler variants by orders of magnitude at equal h. This agrees with the standard order analysis presented in international references and with the structure of national numerical-methods courses, where Runge–Kutta formulas are treated as core tools for Cauchy problems (1, 2, 3, 7, 13, 14).

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

For stiff systems, the decisive issue is numerical stability. A solver may approximate local derivatives very accurately and still fail globally if the step is too large relative to rapidly decaying modes. In Test B, explicit Euler and RK4 produced unacceptable results at h=0.1, while backward Euler remained stable. This phenomenon explains why implicit BDF or implicit Runge–Kutta methods are widely used in chemical kinetics, circuit simulation, constrained mechanics, and other multiscale models. The computational burden of Newton iterations is compensated by the ability to take steps that are far larger than those permitted by explicit stability conditions (4, 6, 15).

Newton iteration deserves special attention in nonlinear implicit solvers. Its local quadratic convergence can make the nonlinear correction inexpensive when the initial guess is good, but each iteration may require derivative evaluation or Jacobian factorization. For large systems, the cost can dominate the entire integration. Therefore practical codes reuse Jacobians, apply simplified Newton steps, exploit sparsity, or employ Krylov linear solvers. These ideas connect the numerical integration problem with the broader computational framework described in Uzbek textbooks on approximate solution of nonlinear systems and scientific computing (5, 6, 11).

Adaptive step control provides a second major route to efficiency. Fixed-step experiments are useful for transparent comparison, but real applications rarely have uniform difficulty over time. An adaptive solver increases h when the estimated local error is comfortably below tolerance and decreases it near rapid transitions. This reduces the number of unnecessary evaluations while retaining accuracy. For non-stiff equations, embedded Runge–Kutta pairs such as RK4(5) are typically effective; for stiff equations, modern BDF and implicit Runge–Kutta solvers combine error estimation with stiffness-aware step selection (13, 15).

The results also have implications for partial differential equations. Spatial discretization of nonlinear PDEs by finite differences, finite volumes, or finite elements often produces large systems of nonlinear ODEs. If diffusion or reaction terms introduce stiffness, an explicit time step may be constrained by a severe stability condition. Semi-implicit or fully implicit discretization can then be more efficient despite the algebraic solve. Uzbek resources on partial differential equations and computational modelling emphasize the need to relate discretization type, boundary conditions, and numerical stability to the mathematical character of the model (8, 10, 12).

A practical method-selection strategy can therefore be formulated. First, analyse the differential equation qualitatively: identify nonlinearity, expected time scales, stiffness, conservation laws, discontinuities, and the required output quantity. Second, choose an initial method family: explicit RK for smooth non-stiff dynamics; adaptive RK if the time scale varies; backward Euler or BDF for stiff dynamics; structure-preserving methods if invariants are essential. Third, perform a step-refinement test or tolerance study. Fourth, verify that the numerical solution is insensitive to reasonable changes in h and nonlinear-solver tolerances. Finally, compare computational cost using function evaluations, Jacobian evaluations, Newton iterations, or CPU time rather than step count alone (3, 5, 9, 10).

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

The study has several limitations. The experiments are deliberately low-dimensional in order to make the error and stability mechanisms transparent. Performance on large nonlinear systems also depends on Jacobian sparsity, memory access, vectorization, preconditioning, and software implementation. The reference solution in the stiff test is numerical rather than analytic, although the selected fine RK4 step makes the comparison sufficiently accurate for illustrating the stability effect. Future research should extend the comparison to adaptive RK45, Rosenbrock, BDF, and implicit Runge–Kutta solvers on higher-dimensional benchmark systems such as van der Pol, Robertson kinetics, nonlinear reaction–diffusion models, and discretized conservation laws.

CONCLUSION

Effective numerical solution of nonlinear differential equations requires simultaneous consideration of accuracy, stability, and computational cost. The experiments demonstrate that formal order alone is not a sufficient selection criterion. For a smooth non-stiff logistic model, classical RK4 gave the smallest error at a fixed step and Heun's method provided a reasonable medium-accuracy alternative. For a stiff nonlinear relaxation problem, backward Euler was substantially more reliable at a large step than higher-order explicit schemes, illustrating the practical value of implicit stability.

The proposed framework recommends matching the solver to the equation: explicit high-order Runge–Kutta methods for smooth non-stiff problems; adaptive Runge–Kutta methods when local scales vary; backward Euler, BDF, or implicit Runge–Kutta families for stiff systems; and semi-implicit or implicit time discretization for stiff systems arising from nonlinear PDEs. Step refinement, error estimation, and verification of qualitative behaviour should be regarded as mandatory parts of the computational procedure rather than optional post-processing.

From an educational and research perspective, the numerical method should be presented not as an isolated formula but as a controlled approximation process. The researcher must justify the discretization, evaluate convergence and stability, document nonlinear iteration tolerances, and report enough information for the computation to be reproducible. Such practice is consistent with the methodological orientation of Uzbek computational-mathematics literature and with contemporary international numerical analysis (1, 4, 8, 13, 15).

REFERENCES:

1. Normurodov, Ch.B. Sonli usullar. I qism [Numerical Methods. Part I]. Tashkent: Termiz Publishing Center, 2025. 298 p. (in Uzbek).
2. Bahromov, S.A. Sonli usullar [Numerical Methods]. Tashkent: National University of Uzbekistan Regional Center, 2022. (in Uzbek).
3. Toshboev, S., Imamov, A. Hisoblash usullari [Computational Methods]. Tashkent: O'ZKITOBSAVDONASHRIYOT NMIU, 2023. 134 p. (in Uzbek).
4. Ismatullayev, G.P., Kosbergenova, M.S. Hisoblash usullari [Computational Methods]. Tashkent: Tafakkur-Bo'stoni, 2014. 240 p. (in Uzbek).

HSR (LONDON),
HOUGHTON STREET REVIEW JOURNAL
ISSN: 2058-5519|Impact factor-15.9
http://houghtonstreetreview.com/index.php/hsr

5. Akramxo‘jayev, Y.T., Usmanov, K.I., Yakubova, N.S. Hisoblashning sonli usullari [Numerical Methods of Computation]. Tashkent: Fan ziyosi, 2025. 153 p. (in Uzbek).
6. Abdirashidov, A. Nochiziqli tenglamalar sistemasini yechishning sonli usullari bo‘yicha uslubiy ko‘rsatmalar [Methodical Guidelines on Numerical Solution of Nonlinear Systems]. Samarkand: SamDU, 2014. 60 p. (in Uzbek).
7. Abdirashidov, A. Birinchi tartibli oddiy differensial tenglamalarni bir qadamli sonli usullar yordamida yechish [One-Step Numerical Methods for First-Order ODEs]. Samarkand: SamDU, 2018. 56 p. (in Uzbek).
8. Noraliyev, N.X., Yusupov, M. Matematik modellashtirish va sonli usullar [Mathematical Modelling and Numerical Methods]. Tashkent: Fan zargari, 2024. 220 p. (in Uzbek).
9. Xurramov, A. Matematik va kompyuterli modellashtirish [Mathematical and Computer Modelling]. Tashkent: City of Book, 2023. 228 p. (in Uzbek).
10. Ravshanov, N. Matematik va kompyuterli modellashtirish asoslari [Fundamentals of Mathematical and Computer Modelling]. Tashkent, 2016. 84 p. (in Uzbek).
11. Xolmatov, T.X., Taylaqov, N.I. Amaliy matematika, dasturlash va kompyuterning dasturiy ta’minoti [Applied Mathematics, Programming and Computer Software]. Tashkent: Mehnat, 2000. 304 p. (in Uzbek).
12. Durdiyev, D.Q. Xususiy hosilali differensial tenglamalardan misol va masalalar to‘plami [Problems in Partial Differential Equations]. Tashkent: International Islamic Academy of Uzbekistan, 2020. 176 p. (in Uzbek).
13. Butcher, J.C. Numerical Methods for Ordinary Differential Equations. 3rd ed. Chichester: Wiley, 2016.
14. Hairer, E., Nørsett, S.P., Wanner, G. Solving Ordinary Differential Equations I: Nonstiff Problems. 2nd rev. ed. Berlin: Springer, 1993.
15. Hairer, E., Wanner, G. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. 2nd rev. ed. Berlin: Springer, 1996.