mathematics

MDPI

Article

Ninth-Order Two-Step Methods with Varying Step Lengths

Rubayyi T. Alqahtani [1], Theodore E. Simos [2,*] and Charalampos Tsitouras [3]

[1] Department of Mathematics and Statistics, College of Science, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh 11623, Saudi Arabia; rtalqahtani@imamu.edu.sa
[2] Center for Applied Mathematics and Bioinformatics, Gulf University for Science and Technology, West Mishref 32093, Kuwait
[3] General Department, National & Kapodistrian University of Athens, GR34-400 Euripus Campus, Greece; tsitourasc@uoa.gr
* Correspondence: tsimos.conf@gmail.com

Abstract: This study investigates a widely recognized ninth-order numerical technique within the explicit two-step family of methods (a.k.a. hybrid Numerov-type methods). To boost its performance, we incorporate an economical step-size control algorithm that, after each iteration, either preserves the current step length, reduces it by half, or doubles it. Any additional off-grid points needed by this strategy are computed using a local interpolation routine. Indicative numerical experiments confirm the substantial efficiency gains realized by this method. It is particularly adept at resolving differential equations with oscillatory dynamics, delivering high precision with fewer function evaluations. Furthermore, a detailed Mathematica implementation is supplied, enhancing usability and fostering further research in the field. By simultaneously improving computational efficiency and accuracy, this work offers a significant contribution to the numerical analysis community.

Keywords: initial value problem; second-order; two-step methods; step-size change control

MSC: 65L05; 65L06

1. Introduction

Our research delves into a specific type of second-order initial value problem (IVP), characterized by the following equation:

$$\psi'' = f(x, \psi), \quad \psi(x_0) = \psi_0, \quad \psi'(x_0) = \psi'_0, \tag{1}$$

where

$$f : \mathbb{R} \times \mathbb{R}^\nu \longrightarrow \mathbb{R}^\nu \quad \text{and} \quad \psi_0,\ \psi'_0 \in \mathbb{R}^\nu.$$

Notably, ψ' does not appear explicitly in (1).

The Numerov method is a well-established numerical technique for advancing the solution from x_κ to $x_{\kappa+1} = x_\kappa + h$. Its formula is given by the following:

$$\psi_{\kappa+1} = 2\psi_\kappa - \psi_{\kappa-1} + \frac{h^2}{12}(f_{\kappa+1} + 10f_\kappa + f_{\kappa-1}),$$

where

$$\psi_\kappa \approx \psi(x_\kappa) \quad \text{and} \quad f_\kappa \approx \psi''(x_\kappa) = f(x_\kappa, \psi_\kappa),$$

with $f_\kappa,\ \psi_\kappa \in \mathbb{R}^\nu$.

check for updates

Academic Editor: Rodica Luca

Received: 5 March 2025
Revised: 30 March 2025
Accepted: 6 April 2025
Published: 11 April 2025

Citation: Alqahtani, R.T.; Simos, T.E.; Tsitouras, C. Ninth-Order Two-Step Methods with Varying Step Lengths. *Mathematics* **2025**, *13*, 1257. https://doi.org/10.3390/math13081257

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

Pioneers such as Hairer [1], Cash [2], and Chawla [3] introduced hybrid implicit Numerov-type methods approximately 40–45 years ago. These methods utilized off-grid points and addressed the challenge of P-stability, which is crucial for stiff oscillatory problems. Later, Chawla [4] proposed a modified explicit Numerov scheme:

$$\begin{aligned} v_1 &= \psi_{\kappa-1}, \\ v_2 &= \psi_{\kappa}, \\ v_3 &= 2\psi_{\kappa} - \psi_{\kappa-1} + h^2 f(x_{\kappa}, v_2), \\ \psi_{\kappa+1} - 2\psi_{\kappa} + \psi_{\kappa-1} &= \tfrac{1}{12} h^2 \Big(f(x_{\kappa+1}, v_3) + 10 f(x_{\kappa}, v_2) + f(x_{\kappa-1}, v_1) \Big), \end{aligned} \tag{2}$$

where h is the constant step size:

$$h = x_{\kappa} - x_{\kappa-1} = x_{\kappa+1} - x_{\kappa} = \cdots = x_1 - x_0.$$

The vectors $\psi_{\kappa+1}$, ψ_{κ}, $\psi_{\kappa-1}$ approximate $\psi(x_{\kappa} + h)$, $\psi(x_{\kappa})$, $\psi(x_{\kappa} - h)$, respectively, while the stages v_1, v_2, $v_3 \in \mathbb{R}^{\nu}$ represent the function evaluations used by the method. We leverage the information at the grid:

$$v_1 = \psi_{\kappa-1}, \quad v_2 = \psi_{\kappa}.$$

Given that $f(x_{\kappa-1}, v_1)$ is calculated in the previous step, only $f(x_{\kappa+1}, v_3)$ and $f(x_{\kappa}, v_2)$ need to be evaluated at each step, resulting in just two function evaluations per step.

Following this, Tsitouras developed a technique drawing inspiration from Runge–Kutta–Nyström (RKN) methods [5], which significantly reduced computational overhead. As a result, only four steps are required to construct a sixth-order method, whereas earlier approaches necessitated six function evaluations (see [6]). Building upon this progress, our research group has extensively explored this topic. Tsitouras developed eighth-order methods that require nine steps per iteration in [7], and concurrently, a team of Spanish researchers conducted notable studies on similar problems [8–10].

A distinction must be made between the term "second-order initial value problem", which signifies the presence of ψ'', and the algebraic order of a numerical scheme, which defines its degree of precision.

Despite these advancements, current numerical methods often struggle with efficiency when applied to differential equations with oscillatory behavior. Our work addresses this limitation by optimizing the step-length adjustment in a ninth-order Numerov-type method, thereby considerably enhancing computational performance.

Here, we present a ninth-order Numerov-type approach that uniquely integrates a variable step-length algorithm, substantially improving computational efficiency for solving second-order IVPs with oscillatory behavior. This study aims to introduce a novel method specifically designed to better handle problems with periodic solutions. Traditionally, various characteristics of a basic test equation are utilized for this purpose; our innovation involves fine-tuning the free parameters across a broad range of relevant problems. Differential evolution is employed for this optimization, and we anticipate that this strategy will yield a method exceptionally well-suited for oscillatory challenges.

The main contributions of this work are as follows:

- The construction of a ninth-order two-step method with variable step lengths, tailored for second-order oscillatory problems;
- The development of an embedded error estimation mechanism enabling efficient step-size control through symbolic interpolation;
- A detailed Mathematica implementation that facilitates symbolic order verification and adaptive logic; and (iv) numerical evidence demonstrating the method's practical

advantages in terms of accuracy and computational efficiency on classical orbital problems such as the Kepler and Arenstorf systems.

2. Derivation of Hybrid Numerov Methods

To address the given problem via advanced algebraic methodologies, we first express the independent parameter t as a component of ψ. Consequently, without any loss of generality, we concentrate on the autonomous formulation:

$$\psi'' = f(\psi).$$

A blended Numerov algorithm utilizing s stages (as delineated in [7]) is formulated as follows:

$$\begin{array}{c} \psi_{\kappa+1} = 2\psi_\kappa - \psi_{\kappa-1} + h^2 \cdot (b \otimes I_s) \cdot f(v) \\ v = (\mathbf{1} + c) \otimes \psi_\kappa - c \otimes \psi_{\kappa-1} + h^2 \cdot (A \otimes I_s) \cdot f(v), \end{array} \tag{3}$$

where $I_s \in \mathbb{R}^{s\times s}$ denotes the identity matrix, and $A \in \mathbb{R}^{s\times s}$, $b^T \in \mathbb{R}^s$, and $c \in \mathbb{R}^s$ signify the coefficient matrices of the approach, with the following:

$$\mathbf{1} = [1\ 1\ \cdots 1]^T \in \mathbb{R}^s.$$

To delineate these coefficients, we employ the Butcher tableau [11,12]:

$$\begin{array}{c|c} c & A \\ \hline & b \end{array}.$$

Since the function assessments are conducted in a sequential manner, these techniques are explicit, implying that A is a strictly lower triangular matrix. For the scenario where $s = 9$, the matrices assume the following configuration:

$$\begin{aligned}
f_{\kappa-1} &= f(x_{\kappa-1}, \psi_{\kappa-1}) = \psi''_{\kappa-1}, \\
f_\kappa &= f(x_\kappa, \psi_\kappa) = \psi''_\kappa, \\
\psi_\alpha &= c_3\psi_{\kappa-1} + (1-c_3)\psi_\kappa + h^2(a_{31}f_{\kappa-1} + a_{d2}f_\kappa), \\
f_\alpha &= f(x_\kappa + c_3h, \psi_\alpha), \\
\psi_\beta &= c_4\psi_{\kappa-1} + (1-c_4)\psi_\kappa + h^2(a_{41}f_{\kappa-1} + a_{42}f_\kappa + a_{43}f_\alpha), \\
f_\beta &= f\Big(x_\kappa + c_4h, \psi_\beta\Big), \\
\psi_\gamma &= c_5\psi_{\kappa-1} + (1-c_5)\psi_\kappa + h^2\Big(a_{51}f_{\kappa-1} + a_{52}f_\kappa + a_{53}f_\alpha + a_{54}f_\beta\Big), \\
f_\gamma &= f(x_\kappa + c_5h, \psi_\gamma), \\
\psi_\delta &= c_6\psi_{\kappa-1} + (1-c_6)\psi_\kappa + h^2\Big(a_{61}f_{\kappa-1} + a_{62}f_\kappa + a_{63}f_\alpha + a_{64}f_\beta + a_{65}f_\gamma\Big), \\
f_\delta &= f(x_\kappa + c_6h, \psi_\delta), \\
\psi_\epsilon &= c_7\psi_{\kappa-1} + (1-c_7)\psi_\kappa + h^2\Big(a_{71}f_{\kappa-1} + a_{72}f_\kappa + a_{73}f_\alpha + a_{74}f_\beta + a_{75}f_\gamma + a_{76}f_\delta\Big), \\
f_\epsilon &= f(x_\kappa + c_7h, \psi_\epsilon), \\
\psi_\zeta &= c_8\psi_{\kappa-1} + (1-c_8)\psi_\kappa + h^2\begin{pmatrix} a_{81}f_{\kappa-1} + a_{82}f_\kappa + a_{83}f_\alpha \\ +a_{84}f_\beta + a_{85}f_\gamma + a_{86}f_\delta + a_{87}f_\epsilon \end{pmatrix}, \\
f_\zeta &= f(x_\kappa + c_8h, \psi_\zeta), \\
\psi_\eta &= c_9\psi_{\kappa-1} + (1-c_9)\psi_\kappa + h^2\begin{pmatrix} a_{91}f_{\kappa-1} + a_{92}f_\kappa + a_{93}f_\alpha + a_{94}f_\beta \\ +a_{95}f_\gamma + a_{96}f_\delta + a_{97}f_\epsilon + a_{98}f_\zeta \end{pmatrix}, \\
f_\eta &= f(x_\kappa + c_9h, \psi_\eta), \\
\psi_{\kappa+1} &= 2\psi_\kappa - \psi_{\kappa-1} + h^2\begin{pmatrix} b_1f_{\kappa-1} + b_2f_\kappa + b_3f_\alpha + b_4f_\beta \\ +b_4f_\gamma + b_6f_\delta + b_6f_\epsilon + b_1f_\zeta + b_1f_\eta \end{pmatrix}.
\end{aligned}$$

Since we assumed the following above:

$$b_3 = 0,\ b_5 = b_4,\ b_7 = b_6,\ b_8 = b_1,\ c_5 = -c_4,\ c_7 = -c_6,\ c_8 = 0,\ c_9 = 1,$$

the associated matrices take the following form:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{31} & a_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{51} & a_{52} & a_{53} & a_{54} & 0 & 0 & 0 & 0 & 0 \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} & 0 & 0 & 0 & 0 \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} & a_{76} & 0 & 0 & 0 \\ a_{81} & a_{82} & a_{83} & a_{84} & a_{85} & a_{86} & a_{87} & 0 & 0 \\ a_{91} & a_{92} & a_{93} & a_{94} & a_{95} & a_{96} & a_{97} & a_{98} & 0 \end{bmatrix},$$

$$b = \begin{bmatrix} b_1 & b_2 & 0 & b_4 & b_4 & b_6 & b_6 & b_1 & b_1 \end{bmatrix}$$

and

$$c = \begin{bmatrix} -1 & 0 & c_3 & c_4 & -c_4 & c_6 & -c_6 & 0 & 1 \end{bmatrix}^T.$$

Since $f_{\kappa-1}$ is derived from the preceding iteration, only seven function evaluations are required per step. To attain an algebraic order of eight, it is crucial to eliminate the corresponding truncation error components (refer to [13]).

The proposed technique comprises a total of 34 parameters. As previously highlighted, there exist 27 coefficients associated with the matrix A, explicitly:

$$a_{31},\ a_{32},\ a_{41},\ a_{42},\ a_{43},\ \cdots,\ a_{87}.$$

Furthermore, 4 coefficients correspond to the vector b, while 3 components belong to the vector c. The total number of order conditions at different accuracy levels aligns with those of Runge–Kutta–Nyström techniques, as illustrated in Table 1. To obtain ninth-order precision, a total of

$$1 + 1 + 2 + 3 + 6 + 10 + 20 + 36 + 72 = 151$$

equations must be fulfilled. Conditions extending to the ninth order can be found in several tables within [13].

Table 1. Number of order constraints.

Order p	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
No of conditions	1	1	2	3	6	10	20	36	72	137	275

Since the number of available parameters is insufficient to match the number of governing equations—a difficulty similar to that encountered in the formulation of Runge–Kutta (RK) schemes—specific simplifying assumptions must be imposed. These reductions decrease the number of constraints, thereby minimizing the number of coefficients. Frequently employed simplifications include the following:

$$\begin{aligned} (A \cdot \mathbf{1})_{(3-8)} &= \tfrac{1}{2}\left(c^2 + c\right)_{(3-9)}, \\ (A \cdot c)_{(3-8)} &= \tfrac{1}{6}\left(c^3 - c\right)_{(3-9)}, \\ (A \cdot c^2)_{(4-8)} &= \tfrac{1}{12}\left(c^4 + c\right)_{(4-9)}, \end{aligned} \tag{4}$$

where

$$c^i = \begin{bmatrix} (-1)^i & 0 & c_3^i & c_4^i & (-c_4)^i & (-c_5)^i & c_5^i & 1 \end{bmatrix}^T,$$

and

$$c^0 = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T = \mathbf{1}.$$

For indices satisfying $\mu_1 < \mu_2$, the notation

$$(v)_{(\mu_1-\mu_2)} = [v_{\mu_1}\ v_{\mu_1+1} \cdots v_{\mu_2}]^T$$

is employed.

The additional order constraints are compiled in Table 2. Within this table, the notation "*" represents element-wise multiplication:

$$[v_1\ \ v_2 \cdots v_n]^T * [u_1\ \ u_2 \cdots u_n]^T = [v_1u_1\ \ v_2u_2 \cdots v_nu_n]^T.$$

This operation has lower precedence, meaning that parentheses, exponentiation, and dot products are always evaluated before applying "*".

Table 2. Equations of conditions up to the ninth order, under assumption (4).

$T_1^{(1)} = b \cdot \mathbf{1} = 1$, $T_2^{(3)} = b \cdot c^2 = \frac{1}{6}$,	$T_6^{(5)} = b \cdot c^4 = \frac{1}{15}$,
$b \cdot c^6 = \frac{1}{28}$,	$b \cdot A^2 \cdot c = 0$,
$b \cdot A^3 \cdot \mathbf{1} = \frac{1}{20160}$,	$b \cdot A \cdot (c * Ac) = -\frac{11}{15120}$,
$b \cdot c^8 = \frac{1}{45}$,	$b \cdot A \cdot (c * A^2 \cdot \mathbf{1}) = \frac{1}{7560}$,
$b \cdot (c * A^2 c) = \frac{17}{10080}$,	$b \cdot (c * A \cdot (c * A \cdot c)) = -\frac{1}{720}$,
$b \cdot (c * A^3 \cdot \mathbf{1}) = \frac{23}{60480}$,	$b \cdot (A \cdot \mathbf{1} * A^2 \cdot c) = \frac{17}{20160}$,
$b \cdot A \cdot (A \cdot \mathbf{1} * A^2 \mathbf{1}) = -\frac{1}{17280}$,	$b \cdot (c * A \cdot (c * A^2 \cdot \mathbf{1})) = -\frac{47}{181440}$,
$b \cdot (A \cdot \mathbf{1} * A^3 \cdot \mathbf{1}) = \frac{373}{1814400}$,	$-\frac{799}{907200} = b \cdot (A \cdot \mathbf{1} * A \cdot (c * A \cdot c))$,
$-\frac{1}{16200} = \frac{1}{36} b \cdot (c^3 * A \cdot c^3)$,	$-\frac{31}{181440} = b \cdot (c * A^3 \cdot c)$
$\frac{1}{1814400} = b \cdot A^4 \cdot \mathbf{1}$,	$\frac{1}{1814400} = \frac{1}{2} b \cdot A^3 \cdot c^2$,
$b \cdot A^2 \cdot (c * A \cdot c) = -\frac{13}{907200}$,	$T_7^{(9)} = -\frac{71}{907200} + b \cdot A \cdot (c * A^2 \cdot c) = 0.$

Thus, we arrive at a nonlinear system consisting of 23 equations involving 23 unknown coefficients. For an in-depth discussion on deriving truncation error coefficients, the extensive review in [13] provides valuable insights. Coleman [14] highlighted the significance of the B2 series framework in analyzing the local truncation error, establishing links with T2 rooted tree structures.

To tackle this challenge, we adopt the differential evolution algorithm [15,16]. Differential evolution operates as an iterative optimization strategy where, at each iteration—termed generation λ—a "population" of candidate solutions is maintained, denoted as $\left(a_3^{(\lambda)}, a_4^{(\lambda)}, a_6^{(\lambda)}, w_1^{(\lambda)}, w_2^{(\lambda)}, \cdots, d_{98}^{(\lambda)}\right)_i$, for $i = 1, 2, \cdots, N$, with N representing the population size. Initially, a starting population $\left(a_3^{(0)}, a_4^{(0)}, a_6^{(0)}, w_1^{(0)}, w_2^{(0)}, \cdots, d_{98}^{(0)}\right)_i$, for $i = 1, 2, \cdots, N$, is randomly generated. The fitness function used for evaluation is defined as follows:

$$s = \sqrt{\left(T_1^{(1)}\right)^2 + \left(T_2^{(3)}\right)^2 + \left(T_6^{(5)}\right)^2 + \cdots + \left(T_7^{(9)}\right)^2}$$

which quantifies the residuals for the 23 governing equations listed in Table 2. Each candidate in the initial population is assessed based on this function, with the objective of minimizing its value.

Every iteration g involves a structured three-step process to refine each candidate solution. These phases include the following:

- Mutation (Differentiation): Generating a new potential candidate based on variations within the existing population.
- Crossover: Recombining individuals to introduce diversity.
- Selection: Retaining the fittest individuals for the subsequent generation.

For implementation, we employed the DeMat software [17] within the MATLAB environment [18], which incorporates this evolutionary strategy. Since a single execution does not guarantee optimal results, we conducted several hundred runs of the optimization process to ensure convergence to a viable solution. The obtained result was then further refined to greater accuracy using multi-precision arithmetic, employing the `NMinimize` function in `Mathematica` [19].

The specific parameters that yield an optimal ninth-order scheme `Num9s8` are documented in [20] and can be found implemented in http://users.uoa.gr/~tsitourasc/NumerovSol9.m (accessed on 5 March 2025).

3. Variable Step Implementation

To establish a variable step-size algorithm, it is necessary to devise an error estimator. A similar strategy was used for 8th-order methods [21]. Here, we introduce an auxiliary vector

$$\tilde{b} = [\tilde{b}_1\ \tilde{b}_2\ \tilde{b}_3\ \tilde{b}_4\ \tilde{b}_5\ \tilde{b}_6\ \tilde{b}_7\ \tilde{b}_8\ \tilde{b}_9],$$

with the aim of determining its coefficients so that the solution is approximated locally with sixth-order accuracy, i.e., with an error of $O(h^8)$. Consequently, the following order conditions must be satisfied:

$$\begin{aligned} \widetilde{b}e = 1, \qquad & \widetilde{b}c = 0, \\ \widetilde{b}c^3 = 0, \qquad & \widetilde{b}c^2 = \tfrac{1}{6}, \\ \widetilde{b}c^4 = \tfrac{1}{15}, \qquad & \widetilde{b}c^5 = 0, \\ \tfrac{1}{120}\widetilde{b}c + \tfrac{1}{120}\widetilde{b}Ac + \widetilde{b}A^2c = 0. & \end{aligned}$$

We have obtained a solution for these coefficients, which is provided in the method's implementation in Appendix A. Notably, no additional order conditions are required since the row-simplifying assumption (4) allows the other conditions to be eliminated.

The resulting, cost-free sixth-order approximation is given by the following:

$$\widetilde{\psi}_{\kappa+1} - 2\psi_\kappa + \psi_{\kappa-1} = h_\kappa^2 \cdot \begin{pmatrix} \tilde{b}_1 f_{\kappa-1} + \tilde{b}_2 f_\kappa + \tilde{b}_3 f_\alpha + \tilde{b}_4 f_\beta \\ +\tilde{b}_5 f_\gamma + \tilde{b}_6 f_\delta + \tilde{b}_7 f_\epsilon + \tilde{b}_8 f_\zeta + \tilde{b}_9 f_\eta \end{pmatrix}.$$

Thus, we obtain two approximations—one of order nine and one of order six—without any extra cost. By subtracting these approximations, we derive an $O(h^8)$ estimate of the local error:

$$E = \|\psi_{\kappa+1} - \widetilde{\psi}_{\kappa+1}\| = h_\kappa^2 \cdot \left\| \begin{pmatrix} (b_1 - \tilde{b}_1) f_{\kappa-1} + (b_2 - \tilde{b}_2) f_\kappa + (b_3 - \tilde{b}_3) f_\alpha \\ +(b_4 - \tilde{b}_4) f_\beta + (b_5 - \tilde{b}_5) f_\gamma + (b_6 - \tilde{b}_6) f_\delta \\ +(b_7 - \tilde{b}_7) f_\epsilon + (b_8 - \tilde{b}_8) f_\zeta + (b_9 - \tilde{b}_9) f_\eta \end{pmatrix} \right\|.$$

We set a tolerance τ and keep the step unchanged if $32\tau > E > \tau/32$. If $E < \tau/32$, we double the step for the next interval. In this way, using the available data $\psi_{\kappa-1}, \psi''_{\kappa-1}$ and $\psi_{\kappa+1}, \psi''_{\kappa+1}$, we can compute the subsequent approximation as follows:

$$\psi_{\kappa+2} = \psi(x_{\kappa+1} + 2h_\kappa) + O(h^8) = \psi(x_{\kappa+1} + h_{\kappa+1}) + O(h^8) \quad \text{(locally)}.$$

Proper adjustments in the indices are applied; see Figure 1 for a graphical illustration.

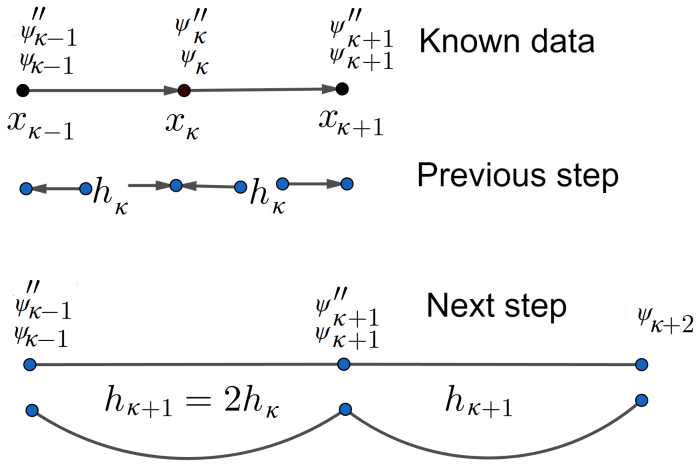


Figure 1. The next step is doubled.

If instead $E > 32\tau$, we reject the trial step $\tilde{h}_\kappa = h_{\kappa-1}$ and instead compute the solution at $x_\kappa + \frac{1}{2}\tilde{h}_\kappa = x_\kappa + h_\kappa$, where h_κ is the halved step used for advancing the solution. We then use ψ_κ, ψ'_κ and suitable approximations for $\psi_{\kappa-\frac{1}{2}}$, $\psi''_{\kappa-\frac{1}{2}}$ to apply methods of the form (3), after adjusting indices appropriately (see Figure 2 for details).

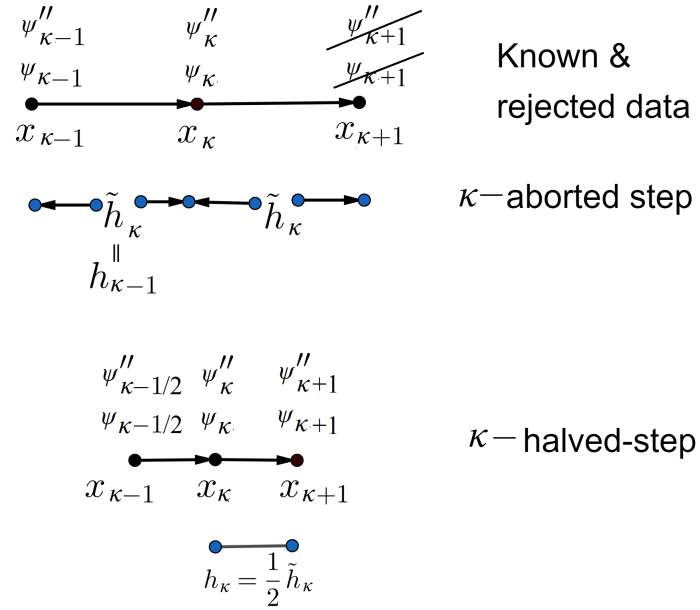


Figure 2. Abortion and halving a step.

A local $O(h^{11})$ approximation for $\psi_{\kappa-\frac{1}{2}}$ and $h^2\psi''_{\kappa-\frac{1}{2}}$ is necessary to prevent order reduction due to error propagation. Such approximations can be obtained by applying the interpolation theory as introduced in [22]. Specifically, we add four additional stages f_1, f_2, f_3 and f_4:

$$\begin{aligned}
\psi_{1^*} &= c_{10}\psi_{\kappa-1} + (1-c_{10})\psi_\kappa + h_\kappa^2\Big(a_{10,1}f_{\kappa-1} + a_{10,2}f_\kappa + a_{10,3}f_\alpha + a_{10,4}f_\beta \\
&\quad + a_{10,5}f_\gamma + a_{10,6}f_\delta + a_{10,7}f_\epsilon + a_{10,8}f_\zeta + a_{10,9}f_\eta\Big), \\
f_{1^*} &= f\left(x_\kappa + c_{10}h_\kappa, \psi_{1^*}\right), \\
\psi_{2^*} &= c_{11}\psi_{\kappa-1} + (1-c_{11})\psi_\kappa + h_\kappa^2\Big(a_{11,1}f_{\kappa-1} + a_{11,2}f_\kappa + a_{11,3}f_\alpha + a_{11,4}f_\beta \\
&\quad + a_{11,5}f_\gamma + a_{11,6}f_\delta + a_{11,7}f_\epsilon + a_{11,8}f_\zeta + a_{11,9}f_\eta + a_{11,10}f_{1^*}\Big), \\
f_{2^*} &= f\left(x_\kappa + c_{11}h_\kappa, \psi_{2^*}\right), \\
\psi_{3^*} &= c_{12}\psi_{\kappa-1} + (1-c_{12})\psi_\kappa + h_\kappa^2\Big(a_{12,1}f_{\kappa-1} + a_{12,2}f_\kappa + a_{12,3}f_\alpha + a_{12,4}f_\beta \\
&\quad + a_{12,5}f_\gamma + a_{12,6}f_\delta + a_{12,7}f_\epsilon + a_{12,8}f_\zeta + a_{12,9}f_\eta + a_{12,10}f_{1^*} + a_{12,11}f_{2^*}\Big), \\
f_{3^*} &= f\left(x_\kappa + c_{12}h_\kappa, \psi_{3^*}\right),
\end{aligned}$$

and

$$\begin{aligned}
\psi_{4^*} &= c_{13}\psi_{\kappa-1} + (1-c_{13})\psi_\kappa + h_\kappa^2\Big(a_{13,1}f_{\kappa-1} + a_{13,2}f_\kappa + a_{13,3}f_\alpha + a_{13,4}f_\beta + a_{13,5}f_\gamma \\
&\quad + a_{13,6}f_\delta + a_{13,7}f_\epsilon + a_{13,8}f_\zeta + a_{13,9}f_\eta + a_{13,10}f_{1^*} + a_{13,11}f_{2^*} + a_{13,12}f_{3^*}\Big), \\
f_{4^*} &= f\left(x_\kappa + c_{12}h_\kappa, \psi_{4^*}\right).
\end{aligned}$$

This yields the following:

$$\begin{aligned}
\psi_{\kappa-\frac{1}{2}} = \frac{1}{2}\psi_\kappa + \frac{1}{2}\psi_{\kappa-1} + h_\kappa^2 \begin{pmatrix} \hat{b}_1 f_{\kappa-1} + \hat{b}_2 f_\kappa + \hat{b}_3 f_\alpha + \hat{b}_4 f_\beta \\ + \hat{b}_5 f_\gamma + \hat{b}_6 f_\delta + \hat{b}_7 f_\epsilon + \hat{b}_8 f_\zeta \\ + \hat{b}_9 f_\eta + \hat{b}_{10} f_{1^*} + \hat{b}_{11} f_{2^*} + \hat{b}_{12} f_{3^*} + \hat{b}_{13} f_{4^*} \end{pmatrix} \\
= \psi\left(x_\kappa - \frac{1}{2}h_\kappa\right) + O(h^{11}),
\end{aligned}$$

where the approximations (i.e., big O notation) are considered locally.

Accordingly,

$$\psi''\left(x_\kappa - \frac{1}{2}h_\kappa\right) \approx f\left(x_\kappa - \frac{1}{2}h_\kappa, \psi_{\kappa-\frac{1}{2}}\right) = \psi''_{\kappa-\frac{1}{2}}.$$

It is evident that rejecting a step incurs a cost of 12 function evaluations compared to the 8 evaluations used in a conventional step—namely, the usual seven stages f_α, f_β, f_γ, f_δ, f_ϵ, f_ζ, f_η plus the extra stages f_{1^*}, f_{2^*} f_{3^*} f_{4^*} and the evaluation of $f\left(x_\kappa - \frac{1}{2}h_\kappa, \psi_{\kappa-\frac{1}{2}}\right)$. Stage f_k remains unchanged and is reused.

The set of new coefficients, i.e., $\hat{b}$, and the enhanced matrix, $\hat{A}$, containing original matrix A and $\hat{c} = [-1, 0, c_3, c_4, -c_4, c_6, -c_6, 0, 1, c_{10}, c_{11}, c_{12}, c_{13}]^T$, have to fulfill a modified set of 151 order conditions. This set has the form shown in Table 3 (see [22]). Here, we set $\rho = -\frac{1}{2}$ to properly approximate $\psi_{\kappa-\frac{1}{2}}$.

All coefficients are provided in the Mathematica listing in the appendix. It can also be retrieved from http://users.uoa.gr/~tsitourasc/NumerovSol9var.m (accessed on 5 March 2025), where we have implemented the method in Mathematica format. In that file, we also

include, in a separate section, the verification of the given coefficients over the 151 equations of the condition for the interpolatory approximation of $\psi_{\kappa-\frac{1}{2}}$ described above.

Table 3. Error conditions for the sixth-order interpolant.

$$\tfrac{1}{2}\cdot(-\rho-\rho^2)+\hat{b}e,$$
$$-\tfrac{1}{6}\rho+\tfrac{1}{6}\rho^3-\hat{b}\hat{c},$$
$$-\tfrac{1}{24}\rho-\tfrac{1}{24}\rho^4-\tfrac{1}{2}\hat{b}\hat{c}+\hat{b}A^*e,$$
$$-\tfrac{1}{24}\rho-\tfrac{1}{24}\rho^4+\tfrac{1}{2}\hat{b}c^{*2},$$
$$\tfrac{1}{120}\rho-\tfrac{1}{120}\rho^5+\tfrac{1}{6}\hat{b}\hat{c}+\hat{b}A^*\hat{c},$$
$$\tfrac{1}{40}\rho-\tfrac{1}{40}\rho^5-\tfrac{1}{2}\hat{b}c^{*2}+\hat{b}(\hat{c}*A^*e),$$
$$-\tfrac{1}{120}\rho+\tfrac{1}{120}\rho^5-\tfrac{1}{6}\hat{b}\hat{c}^3,$$
$$-\tfrac{1}{720}\rho-\tfrac{1}{720}\rho^6-\tfrac{1}{24}\hat{b}c-\tfrac{1}{2}\hat{b}A\hat{c}+\hat{b}A^2e,$$
$$-\tfrac{1}{720}\rho-\tfrac{1}{720}\rho^6-\tfrac{1}{24}\hat{b}\hat{c}+\tfrac{1}{2}\hat{b}A\hat{c}^2,$$
$$-\tfrac{1}{180}\rho-\tfrac{1}{180}\rho^6+\tfrac{1}{6}\hat{b}\hat{c}^2+\hat{b}(\hat{c}*A\hat{c}),$$
$$-\tfrac{1}{240}\rho-\tfrac{1}{240}\rho^6+\tfrac{1}{8}\hat{b}c^2-\tfrac{1}{2}\hat{b}(\hat{c}*Ae)+\tfrac{1}{2}\hat{b}(Ae)^2,$$
$$-\tfrac{1}{120}\rho-\tfrac{1}{120}\rho^6-\tfrac{1}{4}\hat{b}\hat{c}^3+\tfrac{1}{2}\hat{b}(\hat{c}^2*Ae),$$
$$-\tfrac{1}{720}\rho-\tfrac{1}{720}\rho^6+\tfrac{1}{24}\hat{b}\hat{c}^4.$$

4. Numerical Tests

The innovative variable step method is designated as NumerovSol9var, reflecting its connection to **Num**erov, its **9**th-order accuracy, and its **var**iable step capability. It is compared with a method labeled as `Num8s9` in [20] which is the underlying method here when the step is kept constant. This comparison is justified because `Num8s9` has demonstrated superior performance over other methods of similar or lower orders, as evidenced in [20].

Initially, we implement the new method with various tolerance settings, documenting the number of stages used alongside the precision achieved for approximations of ψ at the endpoint. Subsequently, at a comparable computational cost (i.e., the total number of stages utilized in the variable step implementation), we execute the constant step method `Num9s8` and also record the precision at the endpoint. The results are presented in efficiency curves. All tests were conducted using Mathematica.

The problems on which the methods were evaluated are as follows.

4.1. The Two-Body Problem

At the outset, we select the well-known two-body problem (a.k.a., Kepler problem), characterized by eccentricity e:

$$\psi_1''=-\frac{y_1}{(\psi_1^2+\psi_2^2)^{3/2}},\ \psi_2''=-\frac{\psi_2}{(\psi_1^2+\psi_2^2)^{3/2}},$$
$$\psi_1(0)=1-e,\ \psi_1'=0,\ \psi_2(0)=0,\ \psi_2'=\sqrt{\frac{1+e}{1-e}}.$$

The Kepler problem, a fundamental issue in celestial mechanics, entails determining the trajectory of a planet or another celestial object influenced by a central gravitational

force. Within the framework of this paper, the Kepler problem acts as a crucial test case for assessing the efficacy of the variable step-size implementation of the ninth-order hybrid Numerov-type method. Its significance in this research stems from its oscillatory characteristics and well-established periodic solutions, which offer a stringent evaluation of the new numerical technique's effectiveness. By applying this method to such a classical problem, we can showcase its robustness and potential for wider applications in both theoretical and practical contexts involving differential equations with oscillatory dynamics.

Refer to [23] for the theoretical solution. It is important to note that in this problem's definition, the indices correspond to the coordinates of ψ. We solved the aforementioned equation over the interval $[0, 6\pi]$.

We examined three scenarios for eccentricities $e = 0.5$ (problem D3 of DETEST as outlined in [23]), $e = 0.7$ (problem D4 of DETEST), and $e = 0.9$ (problem D5 of DETEST), with results documented in Figures 3, 4, and 5, respectively. These findings demonstrate that utilizing the variable step algorithm provides a distinct advantage.

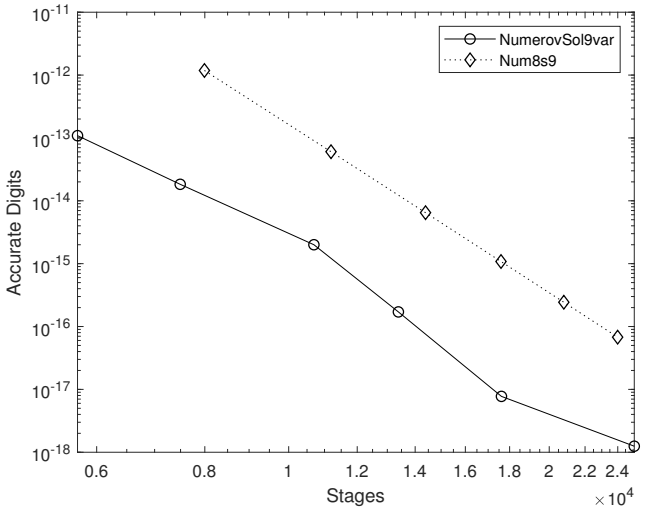


Figure 3. Efficiency curves for DETEST problem D3.

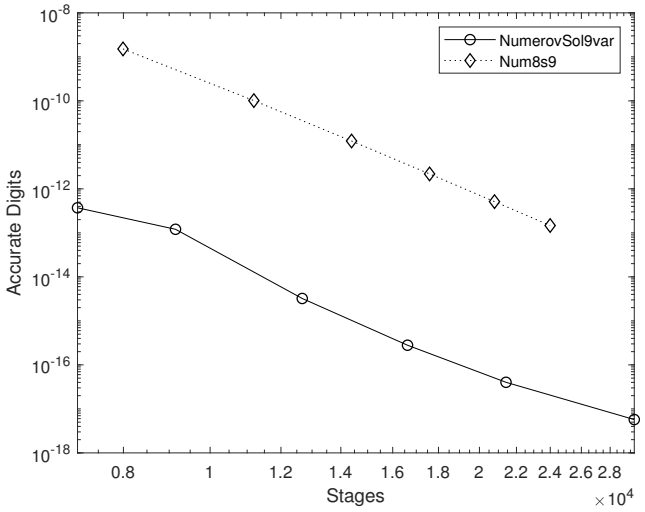


Figure 4. Efficiency curves for DETEST problem D4.

In the context of an elliptical solution (i.e., when $e >> 0$), the significant gain in precision is evident for `NumerovSol9var` across all scenarios. However, when applying an eccentricity of $e = 0.9$, the difference becomes pronounced. Although the proposed ninth-order Numerov-type method with variable step lengths offers considerable efficiency

enhancements, its benefits may be less pronounced for problems that do not necessitate frequent step adjustments, such as the Kepler problem with zero eccentricity. Conversely, when rapidly changing dynamics are present (e.g., Kepler with high eccentricity), the ability to adjust the step size is of paramount importance.

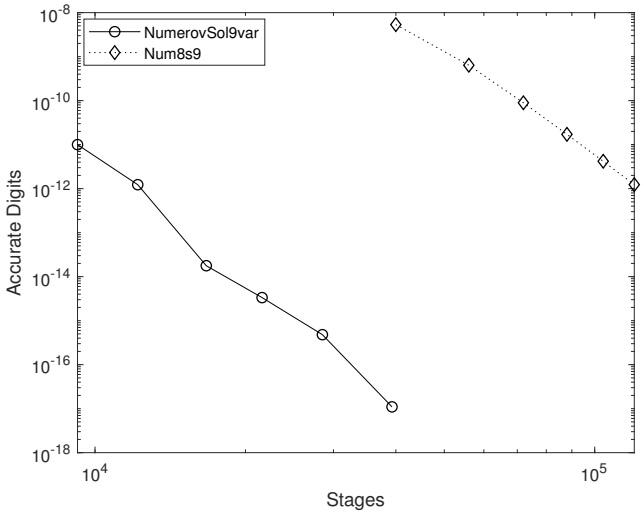


Figure 5. Efficiency curves for DETEST problem D5.

4.2. The Arenstorf Problem

The Arenstorf orbit is a stable orbit between Earth and Moon [24]. In form (1), it reads [25], (p. 296).

$$\begin{aligned} \psi_1'' &= (1-\mu)\frac{-\mu\cos x-\psi_1}{\left((\psi_1+\mu\cos x)^2+(\psi_2+\mu\sin x)^2\right)} \\ &+ \mu\frac{(1-\mu)\cos x-\psi_1}{\left((\psi_1-(1-\mu)\cos x)^2+(\psi_2-(1-\mu)\sin x)^2\right)}, \end{aligned}$$

$$\begin{aligned} \psi_2'' &= (1-\mu)\frac{-\mu\sin x-\psi_2}{\left((\psi_1+\mu\cos x)^2+(\psi_2+\mu\sin x)^2\right)} \\ &+ \mu\frac{(1-\mu)\sin x-\psi_2}{\left((\psi_1-(1-\mu)\cos x)^2+(\psi_2-(1-\mu)\sin x)^2\right)}, \end{aligned}$$

with initial values:

$$\psi_1(0)=0.994,\ \psi_1'(0)=0,\ \psi_2(0)=0,\ \psi_2'(0)=-1.00758510637908252,$$

$\mu=0.012277471$ and end point $x_{\text{end}}=17.0652165601579625589$.

The solution at end–point is as follows:

$$\psi_1(x_{\text{end}})=0.994\cos(x_{\text{end}}),\ \psi_2(x_{\text{end}})=0.994\sin(x_{\text{end}}),$$

and the corresponding efficiency curves are shown in Figure 6.

Here, the results are really astonishing in favor of `NumerovSol9var` since the behavior of the solution requires many rapid and significant changes in the step size.

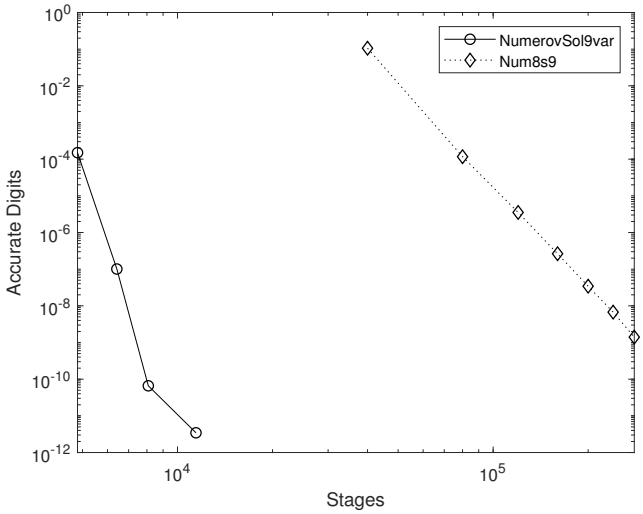


Figure 6. Efficiency curves for Arenstorf.

5. Conclusions

In this work, we introduced a novel ninth-order explicit two-step method equipped with a variable step-size mechanism specifically tailored for second-order initial value problems with oscillatory characteristics. The main contribution lies in the combination of high-order accuracy with a computationally efficient step-size control strategy, which employs error estimation through embedded methods and local interpolation to manage step doubling and halving.

We demonstrated the method's performance across several challenging test problems, including the Kepler and Arenstorf orbits, where the proposed approach showed significant improvements in both accuracy and efficiency compared to fixed-step counterparts. The robustness of the method in handling stiff and rapidly varying dynamics underscores its potential for a wide range of scientific and engineering applications.

Moreover, the implementation in Mathematica, including symbolic verification of order conditions and adaptive control logic, enhances reproducibility and encourages further exploration by the research community.

Future work could investigate more flexible step-size control policies beyond binary step changes (doubling or halving), potentially guided by continuous error estimators. Additionally, extending the method to systems with variable coefficients, discontinuities, or multiphysics coupling would broaden its scope. The optimization of method parameters for specific subclasses of differential equations remains a promising direction for ongoing research.

Author Contributions: Conceptualization, R.T.A., T.E.S. and C.T.; Methodology, T.E.S. and C.T.; Software, R.T.A., T.E.S. and C.T.; Validation, R.T.A., T.E.S. and C.T.; Formal analysis, R.T.A., T.E.S. and C.T.; Investigation, R.T.A., T.E.S. and C.T.; Resources, T.E.S. and C.T.; Data curation, R.T.A., T.E.S. and C.T.; Writing—original draft, C.T.; Writing—review & editing, T.E.S.; Visualization, R.T.A., T.E.S. and C.T.; Supervision, T.E.S.; Project administration, T.E.S.; Funding acquisition, R.T.A. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported and funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) (grant number IMSIU-DDRSP-RP25)

Data Availability Statement: The MATLAB implementation containing the coefficients of the method presented can be retrieved from http://users.uoa.gr/~tsitourasc/NumerovSol9var.m (accessed on 5 March 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Mathematica Listing

We provide a simple description of the program outlining the new implementation. In the input, we specify `fcn` as the function defining the problem, `x0` as the initial point x_0, `x1` as the second point $x_1 = x_0 + h$, `xe` as the endpoint, `ystart0` as the initial value $y(x_0)$, `ystart1` as the initial value $y(x_1)$, `tol` as the tolerance τ. `ystart1` approximates $y(x_0 + h)$ and can be computed using a high-order Runge–Kutta method [26]. In the output, we obtain the grid in `xout` and the matrix containing the approximations of y at these points in `yout`.

It can be retrieved from http://users.uoa.gr/~tsitourasc/NumerovSol9var.m (accessed on 5 March 2025)

```
%-------------------------------------------------------------------------
NumerovSol9var[fcn_, x0_, x1_, xe_, ystart0_, ystart1_, tol_] :=
  Module[{a, b, bb, bs, c, h, x, m, y, f, f0, f1, f2, erro, k, doub,
          half, xout, yout, y0, y1, y2, stages},
   a = SetAccuracy[
   {{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
   {1138818207635569/17888788647582928,3333378954260163/7639424778862807,
    0,0,0,0,0,0,0,0,0,0,0},
   {-738961295452201/18555377494643451,-479030696518956/4630982952248939,
    221463395534051/9040541347820963,0,0,0,0,0,0,0,0,0},
   {-89356055271778/8266078115467089,833921977842237/28086135942568280,
    260430744922526/8296325243696293,1917235959869338/8741740453475911,
    0,0,0,0,0,0,0,0,0},{54946321864723/7072718417900302,
    3292070461962517/15476896927194413,426310606302047/28894894451906660,
    -4872875879837389/20794769713847894,-741532042493254/7746353663357285,
    0,0,0,0,0,0,0,0},{-251224423618045/7424952508545912,
    125088230976712189/272926094612794188,218775165123335/17123639767701491,
    -666835004016636/9078203056179917,864760805293144/11070910018268767,
    960281482715860/4596707163596037,0,0,0,0,0,0,0},
   {-10576773872709440/16766773173543071,-8690145331927382/15796105007055177,
    89083185893050186/132307523842851153,12269900814562454/22837330783350247,
    -5034035877437941/6599359324070268,11389219613319343/15600320848904422,
    52072619848771/16647033931706302,0,0,0,0,0,0},{990106812662497/8608786706779221,
    47709993460294071/35784093553878749,-9736725818190157/25059812990564244,
    -160956717120047/3569163443077247,5224345334971628/20563103273326743,
    -136008468143191/15698515604364771,3475890738044597/11147604929887509,
    -1552113540014879/2714171497526320,0,0,0,0,0},
   {18416007371935979/420432758898246951,-156620231619701803/465915346979262697,
    163401025172665942/235885004367966089,12677558998648686/78953837587788515,
    -79019852266036174/470480554218988571,-38169819233982202/219967012131834185,
    -365063788309013031/906680012125429813,21925574036176765/268754978565376991,
    -41238414961974253/1762793429763497792,0,0,0,0},
   {34543668982039331/410545724822936983,286693809749635373/382376499899843436,
    -114182916006835913/337561389990306790,-25360637462765367/137962475652853258,
    -86500568247232663/235939559792002114,-19491967257348275/230984832785452168,
    11691031589066714/166076370337883255,18961131399369651/71663521282690571,
    44526216726901119/242101211697875540,-1463041441794741/310164447684055957,0,0,0},
   {124667255698865847/539737388738833342,-129716831316448379/246500686057727329,
    48349690501025998/43630540272032909,-8473022563345553/141327709915383754,
    63812242784755555/359395283617477737,-89886296068392911/654585800092602840,
    -72595035575763859/263483678412286769,38977102072170969/314686089940312541,
    -535342268702277563/1292605365671680301,-33999657291563413/201808439863022228,
    5297568021958511/740760492329053756,0,0},{28454082777317276/110034918691513293,
    680667020876240761/679623745715236211,-357139447610076715/476075397256183029,
    -111573303042065203/4510705512098184618,25086062842791575/124338736766908913,
    -7591298801595348/18655481620423638l,227391033138592 4/89681727067232285,
    92929059981689939/293465840017692885,21998919671668353/59906699649424436,
    -31983348834452542/232251852269372751,-40207523454342055/153739002754436812,
    -52822886315457520/176061236801621323,0}}, 33];
   bs = SetAccuracy[{-0.0017997292162861869406821038692212663,
    -0.013112505745942721318478283750136349, 0.003059539915403286950767l066168104193,
```

```
 -0.05491871779622651039791912584989816l, 0.00637383835921743854115332383060568281,
 -0.04005674338916625582769296283246998l, -0.00311408830158357124024021275121572411,
 0.00213090818816288028869909440478959741, 6.97157694938930197838814578484793961*10^-6,
 -0.02100328820928113717731448979576719l, -0.00699132437119350132127783603105345741,
 0.00191499513543995965843107381175567111, 0.00251014385450692948257602807001591281},
 33];
c = SetAccuracy[{-1,0,14472334024676221/23416728348467685,
  -4657513458471779/11992749807882580, 4657513458471779/11992749807882580,
  -9433332829198626/12651088817059513, 9433332829198626/12651088817059513,
  0,1,-1/2,1/2,-3/4,3/4}, 33];
bb = SetAccuracy[{-(2663124765893345/414307129713565604),
 152575928519150547/327517848104256076, 45144011139774057/463099482327322321,
 125109351980123885/778979402784152446, 31069253439689137/277130437312338148,
 13097948064811857/111693808119155519, 5147129187992789/96921094680830912,
 0, 0, 0, 0, 0, 0}, 33];
b = SetAccuracy[{74093748099261/21747946101134003,
    5117072112257713/13418955899996630,0, 2826332804819903/11901194670302015,
    2826332804819903/11901194670302015, 208925646744011/2633580729670742,
    208925646744011/2633580729670742,-706481309125889/32445337026320775,
    74093748099261/21747946101134003, 0, 0, 0, 0}, 33];
h = x1 - x0; doub = 0; stages = 0;
f0 = fcn[x0, ystart0];
xout = Join[{x0}, {x1}, Array[0 &, 100000]];
m = Length[ystart0];
yout = Array[Array[0 &, m] &, 100002];
yout[[1]] = ystart0; y0 = ystart0;
yout[[2]] = ystart1; y1 = ystart1; k = 2;
f = Array[Array[0 &, m] &, 13];
f1 = fcn[x1, yout[[2]]]; x = x1; stages = 2;
While[x < xe, f[[1]] = f0; f[[2]] = f1;
 Do[f[[o]] =
   fcn[x + c[[o]]*h, (1 + c[[o]])*y1 - c[[o]]*y0 +
     h^2*a[[o]] . f], {o, 3, 9}];
     stages = stages + 7;
 erro = 1/2*Max[Abs[h^2*(b - bb) . f]];
 If[erro < 32*tol, x = x + h; k = k + 1;               (* step accepted *)
  y2 = SetAccuracy[2*y1 - y0 + h^2*b . f, 33];
  f2 = fcn[x, y2]; stages = stages + 1;
  xout[[k]] = x; yout[[k]] = y2;
  If[erro < tol/32 && doub == 0, doub = 1; h = h + h;  (* step doubled  *)
   y1 = y2; f1 = f2, y0 = y1; f0 = f1;
   y1 = y2; f1 = f2; doub = 0],
  Do[f[[o]] =                                          (* step halved   *)
    fcn[x + c[[o]]*h, (1 + c[[o]])*y1 - c[[o]]*y0 +
                      h^2*a[[o]] . f], {o, 10, 13}];
  y0 = SetAccuracy[1/2*y1 + 1/2*y0 + h^2*bs . f, 33];
  f0 = fcn[x - h/2, y0];
  stages = stages + 5; h = h/2]];
Return[{xout[[1 ;; k]], yout[[1 ;; k]], stages}]];

%-------------------------------------------------------------------------
```

Thus, we may verify the rightmost lower entry (i.e., for `NumerovSol9var`) in Figure 5 by writing the following (see [23]):

```
In[1]:= gcn[x_,z_]:={-z[[1]]/(z[[1]]^2+z[[2]]^2)^(3/2),
            -z[[2]]/(z[[1]]^2+z[[2]]^2)^(3/2)};

In[2]:= (* for estimation of ystart1 *)
          uu=NMinimize[(u-9/10*Sin[u]-6*Pi/84000)^2,u,WorkingPrecision->33,
          AccuracyGoal->24,PrecisionGoal->24,MaxIterations->1200][[2,1,2]];
In[3]:= sol1=NumerovSol9var[gcn,0,6*Pi/84000,6*Pi,{1/10,0},{Cos[uu]-9/10,
                    Sqrt[1-(81/100)]*Sin[uu]},10^-20];
In[4]:= Print[{sol1[[3]],Max[Abs[sol1[[2]][[Length[sol1[[1]]]]]-{1/10,0}]]}]
Out[4]= {39282,1.099249555737560*10^-17}
```

i.e., using less than 40,000 stages, we achieved an accuracy of almost 17 digits. The step size varied from as low as $\frac{\pi}{14000}$ to as large as $\frac{4\pi}{875}$, i.e., 64 times larger. We remark here that for Arenstorf, the respective variance reaches as high as 256. This variability assisted the new variable-step method in vastly outperforming its counterpart that uses a constant step size.

References

1. Hairer, E. Unconditionally stable methods for second order differential equations. *Numer. Math.* **1979**, *32*, 373–379. [CrossRef]
2. Cash, J.R. High-order P-stable formulae for the numerical integration of periodic initial value problems. *Numer. Math.* **1981**, *37*, 355–370. [CrossRef]
3. Chawla, M.M. Two-step fourth order P-stable methods for second order differential equations. *BIT* **1981**, *21*, 190–193. [CrossRef]
4. Chawla, M.M. Numerov Made Explicit has Better Stability. *BIT* **1984**, *24*, 117–118. [CrossRef]
5. Tsitouras, C. Explicit Numerov type methods with reduced number of stages. *Comput. Maths Appl.* **2003**, *45*, 37–42. [CrossRef]
6. Chawla, M.M.; Rao, P.S. An explicit sixth—Order method with phase—Lag of order eight for $y'' = f(t, y)$. *J. Comput. Appl. Math.* **1987**, *17*, 365–368. [CrossRef]
7. Tsitouras, C. Explicit eighth order two–step methods with nine stages for integrating oscillatory problems. *Int. J. Modern Phys. C* **2006**, *17*, 861–876. [CrossRef]
8. Franco, J.M. A class of explicit two-step hybrid methods for second-order IVPs. *J. Comput. Appl. Math.* **2006**, *187*, 41–57. [CrossRef]
9. Franco, J.M.; Randez, R. Explicit exponentially fitted two-step hybrid methods of high order for second-order oscillatory IVPs. *Appl. Maths. Comput.* **2016**, *273*, 493–505. [CrossRef]
10. Franco, J.M.; Randez, R. Eighth-order explicit two-step hybrid methods with symmetric nodes and weights for solving orbital and oscillatory IVPs. *Int. J. Modern. Phys. C* **2018**, *29*, 1850002. [CrossRef]
11. Butcher, J.C. Implicit Runge Kutta processes. *Math. Comput.* **1964**, *18*, 50–64. [CrossRef]
12. Butcher, J.C. On Runge–Kutta processes of high order. *J. Austral. Math. Soc.* **1964**, *4*, 179–194. [CrossRef]
13. Simos, T.E.; Tsitouras, C.; Famelis, I.T. Explicit Numerov Type Methods with Constant Coefficients: A Review. *Appl. Comput. Math.* **2017**, *16*, 89–113.
14. Coleman, J.P. Order conditions for a class of two-step methods for $y'' = f(x, y)$. *IMA J. Numer. Anal.* **2003**, *23*, 197–220. [CrossRef]
15. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
16. Price, K.V.; Storn, R.M.; Lampinen, J.A. *Differential Evolution, A Practical Approach to Global Optimization*; Springer-Verlag: Berlin/Heidelberg, Germany, 2005.
17. Storn, R.M.; Price, K.V.; Neumaier, A.; Zandt, J.V. DeMat. Available online: https://github.com/mikeagn/DeMatDEnrand (accessed on 8 August 2024).
18. Matlab. *MATLAB Version 2018a*; The MathWorks Inc.: Natick, MA, USA, 2018.
19. Wolfram Research, Inc. *Mathematica, Version 11.1*; Wolfram Research, Inc.: Champaign, IL, USA, 2017.
20. Jerbi, H.; Aoun, S.B.; Alshammari, O.; Simos, T.E.; Tsitouras, C.; Kchaou, M. On explicit ninth-order, two-step methods addressing $y'' = f(x, y)$. *Math. Meth. Appl. Sci.* **2025**, *48*, 2517–2528. [CrossRef]
21. Alshammari, O.; Aoun, S.B.; Kchaou, M.; Simos, T.E.; Tsitouras, C.; Jerbi, H. Eighth-Order Numerov-Type Methods Using Varying Step Length. *Mathematics* **2024**, *12*, 2294. [CrossRef]
22. Alolyan, I.; Simos, T.E.; Tsitouras, C. Interpolants for sixth order Numerov-type methods. *Math. Methods Appl. Sci.* **2019**, *42*, 7349–7358. [CrossRef]
23. Enright, W.; Pryce, J.D. Two FORTRAN packages for assessing initial value methods. *ACM Trans. Math. Softw.* **1987**, *13*, 1–27. [CrossRef]
24. Arenstorf, R.F. Periodic Solutions of the Restricted Three Body Problem Representing Analytic Continuations of Keplerian Elliptic Motions. *Amer.* J. Maths **1963**, *85*, 27–35. [CrossRef]
25. Hairer, E.; Nørsett, S.P.; Wanner, G. *Solving Ordinary Differential Equations I, Nonstiff Problems*, 2nd ed.; Springer-Verlag: Berlin/Heidelberg, Germany, 1993.
26. "ExplicitRungeKutta" for NDSolve. Available online: https://reference.wolfram.com/language/tutorial/NDSolveExplicitRungeKutta.html (accessed on 8 August 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.