mathematics

MDPI

Article

Finite-Time Contractivity Profiling of a Two-Parameter Parallel Root-Finding Scheme via a kNN–LLE Proxy

Mudassir Shams [1,2], Andrei Velichko [3] and Bruno Carpentieri [4,*]

[1] Department of Mathematics, Faculty of Arts and Science, Balikesir University, 10145 Balıkesir, Turkey; mudassir.shams@balikesir.edu.tr
[2] Department of Mathematics and Statistics, Riphah International University, I-14, Islamabad 44000, Pakistan
[3] Institute of Physics and Technology, Petrozavodsk State University, 185910 Petrozavodsk, Russia; velichko@petrsu.ru
[4] Faculty of Engineering, Free University of Bozen-Bolzano, 39100 Bolzano, Italy
* Correspondence: bruno.carpentieri@unibz.it

Abstract

Parallel iterative schemes are widely used for the simultaneous computation of all distinct roots of nonlinear equations in scientific computing and engineering. While high-order parallel methods can provide substantial acceleration, their practical performance is often dominated by the choice of internal real-valued parameters introduced by correction/acceleration mechanisms, which may strongly affect convergence speed and numerical robustness. Classical parameter-selection strategies—based on analytical sufficient conditions, trial-and-error experimentation, or qualitative dynamical diagnostics (basins of attraction, bifurcation-style inspection, and parameter planes)—are typically problem-dependent, expensive to scale, and difficult to automate reproducibly. In this work, we propose a data-driven framework for systematic parameter optimization based on finite-time contractivity profiling. The approach uses k-nearest neighbors (kNN) micro-series analysis to estimate a proxy profile of the largest Lyapunov exponent (LLE) along the iteration index, summarizing the transient contraction/expansion behavior of the solver trajectories. Two profile-based scores, the minimum score $S_{\min}$ and the moment score S_{mom}, are introduced to rank candidate parameter pairs and to construct stability landscapes over (α, β) grids. As a testbed, we apply the framework to a bi-parametric two-step parallel Weierstrass-type scheme and demonstrate that the learned parameter regions yield faster and more reliable convergence than generic or manually tuned choices. Extensive numerical experiments show that the proposed profiling-based optimization consistently improves convergence rate and robustness across the considered nonlinear test problems, providing a scalable and reproducible alternative to heuristic and dynamical-system-based tuning.

Keywords: parallel root-finding; parameter tuning; machine learning; k-nearest neighbors; largest Lyapunov exponent; finite-time contractivity; numerical robustness

MSC: 65Yxx; 68T05; 68T45; 65H04; 65H05; 65H17

Academic Editor: Baccouch Mahboub

Received: 30 January 2026
Revised: 28 February 2026
Accepted: 3 March 2026
Published: 5 March 2026

1. Introduction

Nonlinear equations constitute a central pillar of mathematical modeling in science, engineering, and applied technology. Problems arising in fluid mechanics, chemical reaction networks, electrical circuits, biomedical engineering, control theory, optimization, and signal processing frequently reduce to the computation of the roots of nonlinear scalar or

vector-valued functions [1,2], often in settings where robustness and numerical stability are more critical than extreme precision. In particular, polynomial and polynomial-like nonlinear systems play a crucial role in eigenvalue computations [3], spectral methods [4], stability analysis [5], inverse problems [6], and discretizations of differential and integral equations [7,8]. The reliable and efficient numerical solution of nonlinear equations is therefore a long-standing and continuously evolving research area in numerical analysis. Among the available numerical techniques, iterative root-finding methods are especially attractive due to their conceptual simplicity, flexibility, and scalability. Classical single-root methods such as Newton [9], Halley [10], and Chebyshev schemes [11] are well known for their local convergence properties and high asymptotic convergence orders. However, in many applications, especially those involving high-degree polynomials or large-scale systems, it is not sufficient to compute a single root; rather, all distinct roots must be obtained simultaneously with high accuracy and robustness. This requirement has motivated the development of parallel and simultaneous iterative schemes, among which the Weierstrass (or Durand–Kerner) method [12] and its variants occupy a prominent position. Parallel root-finding methods are particularly well-suited for modern computational architectures, including multicore processors, GPUs, and distributed systems. Their ability to compute all roots simultaneously not only reduces computational cost but also avoids deflation procedures, which may introduce numerical instability and error propagation [13]. Consequently, the design, analysis, and acceleration of parallel iterative schemes remain an active and significant area of research.

Despite their widespread use, classical single-root iterative methods suffer from several well-known limitations:

- First, their convergence is typically local, requiring initial guesses sufficiently close to the desired root.
- Second, they are highly sensitive to the choice of initial values, particularly in the presence of multiple or closely spaced roots.
- Third, when applied sequentially to compute multiple roots, they often rely on deflation techniques, which may degrade numerical stability and amplify rounding errors.

Parallel methods, such as the classical Weierstrass scheme, overcome some of these difficulties by computing all roots simultaneously and avoiding deflation. However, they introduce their own challenges. The original Weierstrass method exhibits quadratic convergence under suitable assumptions, but its performance may deteriorate for ill-conditioned problems, clustered roots, or poorly chosen initial approximations. Moreover, the convergence behavior of parallel schemes is often strongly influenced by nonlinear interactions among the approximations of different roots, leading to complex dynamical phenomena [14,15]. To enhance the efficiency of parallel schemes, various correction strategies have been proposed, including single-step and multi-step modifications designed to increase the convergence order from two to three or higher [16]. While these higher-order schemes offer faster asymptotic convergence, they frequently involve additional parameters whose values critically affect stability, convergence rate, and robustness. Selecting appropriate parameter values is therefore a nontrivial and problem-dependent task.

A substantial body of literature has been devoted to the construction of high-order parallel root-finding methods and to the study of their convergence properties. Classical approaches to parameter selection and method assessment rely on analytical convergence analysis, numerical experiments, and tools from real and complex dynamical systems [17,18]. These include parametric planes [19], bifurcation diagrams [20], critical point analysis [15], and fractal visualizations [21]. While such techniques provide deep qualitative insight into the dynamical behavior of iterative schemes, they also exhibit notable limitations. First, they are often computationally expensive, especially when exploring multi-dimensional

parameter spaces. Second, their interpretation may be subjective and heavily dependent on visualization choices. Third, they do not easily scale to large families of methods or to automated parameter optimization. As a result, there remains a clear gap between the theoretical development of parameterized parallel schemes and the systematic, data-driven selection of optimal parameters. Recent advances in machine learning and data-driven modeling [22,23] offer new opportunities to address this gap. In particular, our recent work (Velichko et al. [22]) introduced and validated a machine-learning-based methodology for the largest Lyapunov exponent estimation from multi-horizon forecast-error growth in one-dimensional chaotic time series, demonstrating strong practical accuracy and robustness. In the present paper, we do not reintroduce that methodology as a new standalone estimator; instead, we adapt its core forecasting-error-growth principle to finite-time contractivity profiling and parameter ranking for a parallel root-finding scheme. However, the integration of machine learning techniques into the design and tuning of parallel root-finding methods remains relatively unexplored.

Parallel root-finding schemes are of increasing importance in both theory and practice. From a theoretical standpoint, they provide a natural framework for studying collective dynamics, stability, and interaction effects among approximations to different roots. From a practical standpoint, they are well aligned with modern high-performance computing environments and enable efficient large-scale computations [24].

The efficiency of parameterized parallel schemes, however, strongly depends on the chosen internal parameters. Traditional tuning relies on analytical sufficient conditions, trial-and-error testing, and qualitative dynamical diagnostics such as parameter planes and basins of attraction. While useful, these approaches are often manual, time-consuming, problem-dependent, and difficult to scale reproducibly, especially when the scheme exhibits sensitive transient dynamics.

1.1. Digital Health Motivation

Beyond classical scientific computing, nonlinear root-finding arises naturally in digital health and occupational safety, where a personal digital assistant must continuously integrate heterogeneous sensor streams (wearables, bedside devices, mobile telemetry, and contextual/environmental sensing) into actionable health services [25]. In many physiological and biochemical pipelines, reduced-order dynamical models are calibrated online, and their operating regimes are characterized by equilibria or steady states; these states are frequently computed by repeatedly solving nonlinear algebraic balance equations (or polynomial surrogates) under changing conditions [26]. Since feasibility and safety are closely tied to stability of the underlying model, stability- and resilience-reflecting constraints are often used to restrict admissible parameter sets and to screen out unstable regimes [27]. Practically, this implies repeated re-initialization and frequent re-calibration at scale, motivating robust and computationally efficient solvers and tuning procedures, including parallel and self-adaptive strategies that reduce wall-clock time while maintaining reliability [28]. Similar repeated estimation loops also appear in contactless vital-sign monitoring, where adaptive nonlinear estimation must remain stable under noise, motion, and nonstationarity [29]. In this context, the proposed finite-time contractivity profiling via a kNN micro-series largest Lyapunov exponent (LLE) proxy provides a lightweight stability indicator for selecting solver parameters that remain reliable across operating conditions and are suitable for reproducible deployment on multicore/edge platforms, complementing earlier step-log contraction profiling results [30].

In this work, we adopt a data-driven alternative based on finite-time contractivity profiling. Specifically, we construct a k-nearest neighbors (kNN) micro-series estimator of the largest Lyapunov exponent (LLE) proxy profile along the iteration index, which

serves as a quantitative indicator of local contraction or expansion in the solver dynamics. The rationale for using kNN is that stability is inherently local: by comparing only nearby micro-series windows (in the space of short trajectory segments), the estimator captures local error-growth behavior without imposing a global parametric model. The resulting profile $\lambda_1(t_{\text{end}})$ serves as an interpretable finite-time indicator: negative values correspond to locally contractive behavior, whereas positive values indicate transient instability. To enable systematic parameter selection, the profile is summarized by two scalar scores computed from the smoothed profile: the stability minimum $S_{\min}$ and the stability moment S_{mom}. These metrics quantify the depth and temporal concentration of contractive behavior and allow consistent ranking of candidate parameter pairs across test problems. Unlike purely visual diagnostics, the proposed profiling framework yields reproducible stability landscapes over (α, β) grids and supports automated screening of parameter regions associated with rapid and robust convergence.

1.2. Positioning of the Proposed Framework

Unlike classical basin-of-attraction or bifurcation analyses, which provide primarily qualitative insight, the proposed framework delivers an automated and quantitatively ranked parameter-selection mechanism based on finite-time contractivity profiling. In contrast to generic machine-learning hyperparameter tuning strategies, the approach is explicitly tailored to the dynamical structure of parallel root-finding iterations and exploits short-horizon error-growth behavior as an interpretable stability proxy. This design enables scalable and reproducible screening of parameter regions while preserving a clear connection to the underlying numerical dynamics.

The main contributions of this paper can be summarized as follows:

- A bi-parametric two-step parallel Weierstrass-type scheme, augmented by a single-step correction strategy, increasing the convergence order from two to three while preserving the parallel structure.
- A data-driven, systematic parameter optimization framework based on kNN micro-series LLE-proxy profiling, including smoothing and profile-based ranking via the metrics $S_{\min}$ and S_{mom}.
- Heatmap-based stability landscapes over (α, β) that reveal structured regions of finite-time contractivity and provide an automated alternative to manual parameter tuning.
- Extensive numerical experiments showing that parameters selected by the proposed profiling framework yield consistent improvements in convergence speed and robustness when computing all distinct roots.

In this context, the primary novelty of the present study lies in the proposed data-driven parameter optimization framework based on kNN–LLE stability profiling; the new bi-parametric parallel scheme is used as a representative demonstration testbed for this methodology. To the best of our knowledge, this is among the first works to employ kNN-based finite-time Lyapunov-proxy profiling to systematically optimize parameters in parallel simultaneous root-finding schemes.

The remainder of this paper is organized as follows: Section 2 introduces the proposed parallel scheme and presents its theoretical convergence analysis. Section 3 describes the profiling framework, including the kNN micro-series methodology and the definition of the performance metrics. Section 4 summarizes the algorithmic workflow of the proposed methodology. Section 5 reports numerical experiments and comparative studies illustrating the effectiveness of the proposed approach. Section 6 concludes the paper and outlines directions for future research.

2. Development, Methodology, and Implementation of the Two-Step Parallel Root-Finding Pipeline

Nonlinear equations arise ubiquitously in science and engineering, with applications spanning fluid dynamics, signal processing, control systems, chemical kinetics, optimization, and biomedical modeling. Accurate and efficient solution of such equations is therefore a fundamental computational task. Classical single-root methods, such as Newton, Halley, and their variants, are widely used due to their simplicity and fast local convergence. However, in many practical problems—especially those involving high-degree polynomials or coupled nonlinear systems—it is desirable to compute all roots simultaneously rather than sequentially.

After the impossibility theorems [31], parallel (simultaneous) methods were developed to approximate all roots of nonlinear equations, particularly polynomials. Consider the general nonlinear equation

$$f(z) = 0. \tag{1}$$

If f is a polynomial with n distinct (simple) roots $\xi_1, \xi_2, \ldots, \xi_n$, then it can be written in the factored form

$$f(z) = a\prod_{i=1}^{n}(z - \xi_i), \qquad a \neq 0, \tag{2}$$

with degree

$$\deg(f) = n. \tag{3}$$

Owing to their global convergence, enhanced stability, reliability, and natural suitability for parallel architectures, such simultaneous schemes have become central tools in modern polynomial root-finding and large-scale nonlinear computations. Their ability to compute both distinct and multiple roots concurrently makes them especially effective for high-degree polynomials and large-scale nonlinear problems [32]. The simplicity of these methods, their global convergence properties, and their independence from derivative evaluations render them robust and widely adopted in scientific and engineering computations. Moreover, their inherently parallel structure enables efficient implementation on modern multicore processors, GPU architectures, and distributed platforms, thereby facilitating fast, reliable, and scalable root-finding across a broad range of applications.

In this section, we first propose inverse parallel schemes and then present a detailed exposition of our methodology, including the mathematical formulation of the iterative scheme, the motivation for employing Lyapunov analysis, the micro-series kNN-based Lyapunov estimator, the experimental setup, and the organization of the numerical results.

Among the classical schemes, the well-known Weierstrass–Durand–Kerner method [33], which exhibits local quadratic convergence, updates the approximations $z_i^{[\check{t}]}$ of the n roots ($i = 1, \ldots, n$) at iteration $\check{t}$ according to

$$z_i^{[\check{t}+1]} = z_i^{[\check{t}]} - \frac{f\left(z_i^{[\check{t}]}\right)}{\prod_{j\neq i}^{n}\left(z_i^{[\check{t}]} - z_j^{[\check{t}]}\right)}. \tag{4}$$

Nourein's method [34] is defined as

$$z_i^{[\check{t}+1]} = z_i^{[\check{t}]} - \frac{\mathcal{Q}\left(z_i^{[\check{t}]}\right)}{1 + \sum_{j\neq i}^{n}\left(\frac{\mathcal{Q}(z_j^{[\check{t}]})}{z_i^{[\check{t}]} - \mathcal{Q}(z_j^{[\check{t}]}) - z_j^{[\check{t}]}}\right)}, \tag{5}$$

where

$$\mathcal{Q}(z_i^{[\check{t}]}) = \frac{f(z_i^{[\check{t}]})}{\prod_{j\neq i}\left(z_i^{[\check{t}]} - z_j^{[\check{t}]}\right)}.$$

Zhang et al. [35] introduced the fifth-order method (denoted by ZHM) as

$$z_i^{[\check{t}+1]} = z_i^{[\check{t}]} - \frac{\mathcal{Q}(z_i^{[\check{t}]})}{1 + \mathcal{G}_i^{[*]}(z_i^{[\check{t}]}) + \sqrt{\mathcal{M}_{1.1}}}, \tag{6}$$

where

$$\mathcal{M}_{1.1} = 1 + \left(\mathcal{G}_i^{[*]}(z_i^{[\check{t}]})\right)^2 + 4\mathcal{Q}(z_i^{[\check{t}]}) \sum_{j\neq i}^{n}\left(\frac{\mathcal{Q}(z_j^{[\check{t}]})}{(z_i^{[\check{t}]} - z_j^{[\check{t}]})(z_i^{[\check{t}]} - \mathcal{Q}(z_j^{[\check{t}]}) - z_j^{[\check{t}]})}\right),$$

and

$$\mathcal{G}_i^{[*]}(z_i^{[\check{t}]}) = \frac{\mathcal{Q}(z_i^{[\check{t}]})}{\sum_{j\neq i}^{n}(z_i^{[\check{t}]} - z_j^{[\check{t}]})}.$$

Several researchers have also proposed multi-step simultaneous root-finding schemes, see, e.g., [36–41] and the references therein.

2.1. An Optimal Bi-Parametric Modified Newton Method

Here, we first propose a single-step optimal bi-parametric family for nonlinear equations. This scheme is subsequently embedded as a correction stage within a simultaneous framework and later optimized using kNN–LLE Lyapunov profiling to identify high-performance parameter regimes, ensuring fast and stable convergence for both single-root and simultaneous nonlinear solvers.

The proposed single-root iteration is defined as

$$z^{[\check{t}+1]} = z^{[\check{t}]} - \left(\beta f(z^{[\check{t}]}) + \left(1 + \frac{\alpha f(z^{[\check{t}]})}{1 + f(z^{[\check{t}]})}\right)^{-1}\right)\frac{f(z^{[\check{t}]})}{f'(z^{[\check{t}]})}, \qquad \alpha, \beta \in \mathbb{R}. \tag{7}$$

Here, the free parameters α and β act as nonlinear control weights that regulate both the local correction magnitude and the global stability of the iteration. This formulation enables systematic tuning of convergence behavior, which we later exploit through kNN–LLE Lyapunov profiling to extract optimal parameter sets for parallel schemes.

Theorem 1. *Let $I \subset \mathbb{R}$ be an open interval, and let $\xi \in I$ be a simple root of a sufficiently smooth function $f : I \to \mathbb{R}$, i.e., $f'(\xi) \neq 0$. Define the error by $\varepsilon^{[\check{t}]} = z^{[\check{t}]} - \xi$. Then the iterative scheme (7) converges locally to ξ with quadratic order. Moreover, the associated error equation is given by*

$$\varepsilon^{[\check{t}+1]} = (\alpha - \beta + c_2)\,(\varepsilon^{[\check{t}]})^2 + \mathcal{O}\left((\varepsilon^{[\check{t}]})^3\right), \tag{8}$$

where

$$c_m = \frac{f^{(m)}(\xi)}{m!\, f'(\xi)}, \qquad m \geq 2.$$

Proof. Since ξ is a simple root, we have $f(\xi) = 0$ and $f'(\xi) \neq 0$. Expanding $f(z^{[\check{t}]})$ and $f'(z^{[\check{t}]})$ in Taylor series about ξ yields

$$f(z^{[\check{t}]}) = f'(\xi)\Big(\varepsilon^{[\check{t}]} + c_2(\varepsilon^{[\check{t}]})^2 + c_3(\varepsilon^{[\check{t}]})^3 + c_4(\varepsilon^{[\check{t}]})^4 + c_5(\varepsilon^{[\check{t}]})^5 + \mathcal{O}((\varepsilon^{[\check{t}]})^6)\Big), \tag{9}$$

$$f'(z^{[\check{t}]}) = f'(\xi)\Big(1 + 2c_2(\varepsilon^{[\check{t}]}) + 3c_3(\varepsilon^{[\check{t}]})^2 + 4c_4(\varepsilon^{[\check{t}]})^3 + 5c_5(\varepsilon^{[\check{t}]})^4 + \mathcal{O}((\varepsilon^{[\check{t}]})^5)\Big). \tag{10}$$

Dividing (9) by (10), we obtain

$$\frac{f(z^{[\check{t}]})}{f'(z^{[\check{t}]})} = \varepsilon^{[\check{t}]} + c_2(\varepsilon^{[\check{t}]})^2 + (2c_2^2 - 4c_3)(\varepsilon^{[\check{t}]})^3 + \mathcal{O}((\varepsilon^{[\check{t}]})^4). \tag{11}$$

Next, expanding the nonlinear weight term gives

$$1 + \frac{\alpha f(z^{[\check{t}]})}{1 + f(z^{[\check{t}]})} = 1 + \alpha\varepsilon^{[\check{t}]} + (\alpha c_2 - \alpha)(\varepsilon^{[\check{t}]})^2 + (-2\alpha c_2 + \alpha c_3 + \alpha)(\varepsilon^{[\check{t}]})^3 + \mathcal{O}((\varepsilon^{[\check{t}]})^4). \tag{12}$$

Its inverse is

$$\begin{aligned}\left(1 + \frac{\alpha f(z^{[\check{t}]})}{1 + f(z^{[\check{t}]})}\right)^{-1} &= 1 - \alpha\varepsilon^{[\check{t}]} + (\alpha^2 - \alpha c_2 + \alpha)(\varepsilon^{[\check{t}]})^2 + \Big(2\alpha c_2 - \alpha c_3 - \alpha + \alpha(\alpha c_2 - \alpha) \\ &\quad + (-\alpha^2 + \alpha c_2 - \alpha)\alpha\Big)(\varepsilon^{[\check{t}]})^3 + \mathcal{O}((\varepsilon^{[\check{t}]})^4).\end{aligned} \tag{13}$$

For the composite correction term, we obtain

$$\begin{aligned}&\left(\beta f(z^{[\check{t}]}) + \left(1 + \frac{\alpha f(z^{[\check{t}]})}{1 + f(z^{[\check{t}]})}\right)^{-1}\right)\frac{f(z^{[\check{t}]})}{f'(z^{[\check{t}]})} = \varepsilon^{[\check{t}]} + (-\alpha + \beta - c_2)(\varepsilon^{[\check{t}]})^2 \\ &+(\alpha^2 + 2c_2^2 + \alpha - 2c_3)(\varepsilon^{[\check{t}]})^3 + \mathcal{O}((\varepsilon^{[\check{t}]})^4).\end{aligned} \tag{14}$$

Finally, the updated iterate satisfies

$$z^{[\check{t}+1]} - \xi = (-\alpha^2 - 2c_2^2 - \alpha + 2c_3)(\varepsilon^{[\check{t}]})^3 + (\alpha - \beta + c_2)(\varepsilon^{[\check{t}]})^2 + \mathcal{O}((\varepsilon^{[\check{t}]})^4). \tag{15}$$

Substituting (11)–(15) into the scheme (7) and collecting like powers of $\varepsilon^{[\check{t}]}$ yields

$$\varepsilon^{[\check{t}+1]} = (\alpha - \beta + c_2)(\varepsilon^{[\check{t}]})^2 + \mathcal{O}((\varepsilon^{[\check{t}]})^3), \tag{16}$$

which confirms quadratic convergence and completes the proof. □

Remark 1. *The presence of two free parameters (α, β) enables fine-grained control over both stability and convergence speed. In particular, α primarily scales the local correction magnitude and thus affects the convergence rate, while β influences the coupling and damping of the iterative dynamics, playing a key role in stability; their combined effect is systematically optimized in later sections via kNN–LLE Lyapunov profiling, allowing the method to transition seamlessly from single-root solvers to high-efficiency parallel schemes with superior dynamical stability properties.*

2.2. Proposed Parallel Bi-Parametric Scheme for Nonlinear Problems

We start from a parameterized single-root iterative method and incorporate a Weierstrass-type correction to construct a parallel framework for the simultaneous ap-

proximation of all simple roots of the nonlinear Equation (1). The resulting inverse parallel fractional scheme is defined by

$$\begin{cases} y_i^{[\check{t}]} = z_i^{[\check{t}]} - \dfrac{f\left(z_i^{[\check{t}]}\right)}{\prod\limits_{\substack{j=1 \\ j\neq i}}^{n}\left(z_i^{[\check{t}]} - z_j^{[\check{t}]} + \dfrac{f\left(z_j^{[\check{t}]}\right)}{f'\left(z_j^{[\check{t}]}\right)}\left[\left(1+\dfrac{\alpha f\left(z_j^{[\check{t}]}\right)}{1+\beta f\left(z_j^{[\check{t}]}\right)}\right)^{-1} + \beta f\left(z_j^{[\check{t}]}\right)\right]\right)}, \\ z_i^{[\check{t}+1]} = y_i^{[\check{t}]} - \dfrac{f\left(y_i^{[\check{t}]}\right)}{\prod\limits_{\substack{j=1 \\ j\neq i}}^{n}\left(y_i^{[\check{t}]} - y_j^{[\check{t}]}\right)}. \end{cases} \tag{17}$$

where $\alpha, \beta \in \mathbb{R}$ are free parameters. The method (17), denoted MVB$_*$, is analyzed for its local convergence order in the following theorem.

Theorem 2. *Let $\xi_1, \ldots, \xi_n$ be distinct simple zeros of a function $f \in C^7(\Omega)$, where $\Omega \subset \mathbb{C}$ is an open set containing these zeros. Let*

$$d = \min_{i\neq j} |\xi_i - \xi_j| > 0$$

denote the minimum root separation.

Then there exists a constant $\delta > 0$, with $0 < \delta < \frac{d}{2}$, such that if the initial approximations satisfy

$$\max_{1\leq i\leq n} |z_i^{[0]} - \xi_i| < \delta,$$

and the denominators in (17) remain nonzero, the iterative scheme (17) converges locally to $\xi_1, \ldots, \xi_n$ with order six. More precisely, for k sufficiently large,

$$\varepsilon_i^{[\check{t}+1]} = O\left((\varepsilon_i^{[\check{t}]})^6\right), \quad i = 1, \ldots, n.$$

Remark 2. *The above result is a local convergence theorem. The condition $\max_i |z_i^{[0]} - \xi_i| < \delta$ means that the initial vector lies in a neighborhood where (i) the roots remain separated so that denominator terms do not vanish, and (ii) the Taylor expansions used in the proof are valid. An explicit universal numerical value of δ cannot be given, since it depends on the derivative bounds of f and the root separation d.*

Proof. Define the iteration errors by

$$\varepsilon_i = z_i^{[\check{t}]} - \xi_i, \qquad \varepsilon_i' = y_i^{[\check{t}]} - \xi_i, \qquad \varepsilon_i'' = z_i^{[t\check{+}1]} - \xi_i.$$

From the first step of (17), we obtain

$$y_i^{[\check{t}]} - \xi_i = z_i^{[\check{t}]} - \xi_i - \frac{f(z_i^{[\check{t}]})}{\prod\limits_{\substack{j=1 \\ j\neq i}}^{n}(z_i^{[\check{t}]} - q_j^{[\check{t}]})}, \tag{18}$$

where

$$q_j^{[\check{t}]} = z_j^{[\check{t}]} - \frac{f(z_j^{[\check{t}]})}{f'(z_j^{[\check{t}]})}\left(\left(1 + \frac{\alpha f(z_j^{[\check{t}]})}{1+\beta f(z_j^{[\check{t}]})}\right)^{-1} + \beta f(z_j^{[\check{t}]})\right).$$

Rewriting (18) in terms of error variables yields

$$\varepsilon_i' = \varepsilon_i\left(1 - \prod_{\substack{j=1\\ j\neq i}}^{n} \frac{z_i^{[\check{t}]} - \xi_j}{z_i^{[\check{t}]} - q_j^{[\check{t}]}}\right). \tag{19}$$

Using the expansion

$$\frac{z_i^{[\check{t}]} - \xi_j}{z_i^{[\check{t}]} - q_j^{[\check{t}]}} = 1 + \frac{z_j^{[\check{t}]} - \xi_j}{z_i^{[\check{t}]} - q_j^{[\check{t}]}},$$

together with $q_j^{[\check{t}]} - \xi_j = O(\varepsilon_j^2)$, we obtain

$$\prod_{\substack{j=1\\ j\neq i}}^{n} \frac{z_i^{[\check{t}]} - \xi_j}{z_i^{[\check{t}]} - q_j^{[\check{t}]}} = 1 + (n-1)O(\varepsilon_j^2),$$

where $|\varepsilon_i| = |\varepsilon_j| = |\varepsilon|$ is assumed. Hence,

$$\varepsilon_i' = O(\varepsilon^3). \tag{20}$$

For the second step of (17), we have

$$z_i^{[\check{t}+1]} - \xi_i = y_i^{[\check{t}]} - \xi_i - \frac{f(y_i^{[\check{t}]})}{\prod_{\substack{j=1\\ j\neq i}}^{n}(y_i^{[\check{t}]} - y_j^{[\check{t}]})},$$

which implies

$$\varepsilon_i'' = \varepsilon_i'\left(1 - \prod_{\substack{j=1\\ j\neq i}}^{n} \frac{y_i^{[\check{t}]} - \xi_j}{y_i^{[\check{t}]} - y_j^{[\check{t}]}}\right). \tag{21}$$

Using the expansion

$$\frac{y_i^{[\check{t}]} - \xi_j}{y_i^{[\check{t}]} - y_j^{[\check{t}]}} = 1 + \frac{y_j^{[\check{t}]} - \xi_j}{y_i^{[\check{t}]} - y_j^{[\check{t}]}},$$

together with $y_j^{[\check{t}]} - \xi_j = O(\varepsilon_j')$, we obtain

$$\prod_{\substack{j=1\\ j\neq i}}^{n} \frac{y_i^{[\check{t}]} - \xi_j}{y_i^{[\check{t}]} - y_j^{[\check{t}]}} = 1 + (n-1)O(\varepsilon_j'),$$

where $|\varepsilon_i'| = |\varepsilon_j'| = |\varepsilon'|$. Therefore,

$$\varepsilon_i'' = O\Big((\varepsilon')^2\Big) = O(\varepsilon^6),$$

which proves that the method converges with order six. □

While analytical convergence analysis provides theoretical guarantees, the practical performance of iterative schemes is highly sensitive to the choice of control parameters. To systematically identify stable and optimal parameter regions, we complement the theoretical study with a data-driven profiling strategy. In the next section, we introduce a profile-based kNN–LLE framework for stability estimation and parameter tuning, which uses local trajectory information to identify robust convergence regions and guide the selection of effective parameter values for the proposed methods.

3. Profile-Based kNN–LLE Contractivity Estimation and Parameter Tuning Framework

In our earlier work [30], the stability of parallel Weierstrass-type solvers was quantified through a direct step-log contraction profile (log-ratio averaging of successive step norms).

In the present study, we keep the same contractivity-oriented viewpoint, but replace the direct averaging by a data-driven finite-time contractivity proxy based on micro-series and a k-nearest neighbors (kNN) estimate of the largest Lyapunov exponent (LLE)-like slope.

Throughout this manuscript, the term kNN–LLE denotes such a k-nearest neighbor-based largest Lyapunov exponent estimate.

3.1. Goal: Parameter Tuning via a Finite-Time Contractivity Proxy

Parallel high-order root-finders may exhibit fast convergence only for specific internal parameters. Here, the objective is to rank parameter pairs (α, β) using an automatically computed finite-time contractivity profile extracted from solver trajectories. The profile acts as a stability/contractivity proxy: negative values indicate locally contractive dynamics (desirable for convergence), while sustained positive values indicate transient instability.

3.2. Two-Step Parallel Scheme and Scalar Observable

Let $z^{[i]} \in \mathbb{C}$ denote the i-th component of the vector iterate $z \in \mathbb{C}^n$. We consider the modified two-step parallel scheme (denominator stabilizer $\varepsilon_{\text{den}} > 0$):

Step 1 ($z \to y$; parameterized correction). For $i = 1, \ldots, n$,

$$y_i^{[\check{t}]} = z_i^{[\check{t}]} - \frac{f(z_i^{[\check{t}]})}{\prod_{j \neq i}(z_i^{[\check{t}]} - z_j^{[\check{t}]} + W_i) + \varepsilon_{\text{den}}}, \qquad W_i = \frac{\dfrac{f(z_i^{[\check{t}]})}{f'(z_i^{[\check{t}]}) + \varepsilon_{\text{den}}}}{1 + \alpha \left(\dfrac{f(z_i^{[\check{t}]})}{1 + \beta f(z_i^{[\check{t}]})} \right)}. \tag{22}$$

Step 2 ($y \to z_+$; plain Weierstrass/Durand–Kerner update).

$$z_{i,+}^{[\check{t}]} = y_i^{[\check{t}]} - \frac{f(y_i^{[\check{t}]})}{\prod_{j \neq i}(y_i^{[\check{t}]} - y_j^{[\check{t}]}) + \varepsilon_{\text{den}}}. \tag{23}$$

To obtain a scalar trajectory from each run, we record the step norm

$$s_k = \| z_k^{[\check{t}+1]} - z_k^{[\check{t}]} \|_2, \tag{24}$$

and store the log-step observable

$$x_k = \log(s_k + \varepsilon_{\text{tiny}}), \tag{25}$$

where $\varepsilon_{\text{tiny}}$ is used only for log-safety.

3.3. Ensemble Dataset (Multiple Runs per (α, β))

For each fixed (α, β) we generate an ensemble of N_{runs} independent trajectories with random complex initial vectors $z^{[0]} \in \mathbb{C}^n$, and iterate for K steps. Collecting the scalar series yields the numeric matrix

$$X = [x_{r,k}] \in \mathbb{R}^{N_{\text{runs}} \times K}, \tag{26}$$

where each row corresponds to one run r and contains the time series $(x_{r,1}, \ldots, x_{r,K})$. This CSV matrix is the only input to the profiling stage.

To ensure reproducibility across MATLAB (R2023b) and Python (version 3.11, NumPy 1.26) implementations, it is important that the random initialization consumes the random number generator in the same order. In our generators, for each run, all real parts are drawn first, followed by all imaginary parts; changing this order leads to slightly different CSV datasets and can result in noticeable shifts in the downstream contractivity profiles.

3.4. kNN Micro-Series LLE Profiling

We emphasize that the quantity estimated in this subsection is used as a finite-time contractivity proxy for parameter ranking and stability profiling, rather than as a strict classical finite-time Lyapunov exponent (FTLE) of the full coupled iterative map. The construction is adapted from the machine-learning LLE-estimation methodology introduced and validated in Velichko et al. [22], where multi-horizon forecast-error growth was used to estimate Lyapunov behavior from time-series data. In the present work, the same forecasting-error-growth principle is transferred to solver-generated scalar observables in order to profile transient contraction/expansion behavior across parameter pairs (α, β).

For a chosen window end index t_{end}, we extract from each run the last L_{micro} samples:

$$U^{(t_{\text{end}})} = X[:, t_{\text{end}} - L_{\text{micro}} + 1 : t_{\text{end}}] \in \mathbb{R}^{N_{\text{runs}} \times L_{\text{micro}}}, \qquad L_{\text{micro}} = L_b + H_{\max}, \tag{27}$$

where L_b is `LOOK_BACK` and $H_{\max}$ denotes the maximum forward offset (prediction horizon) over which local kNN-based trajectory evolution is evaluated when estimating the finite-time Lyapunov proxy.

For each t_{end}, we build supervised pairs (input window of length L_b and target h steps ahead), fit a kNN regressor, and compute a horizon-dependent prediction error. Let $y(h)$ denote the logarithm of an aggregated prediction error at horizon h. Under approximate short-horizon exponential error growth/decay, $y(h)$ is approximately linear in h, which motivates slope-based finite-time profiling. We estimate the slope by a robust line (or two-segment) fit and define

$$\lambda_1(t_{\text{end}}) := \text{slope of the short-horizon fit of } y(h). \tag{28}$$

Scanning t_{end} produces a profile $\lambda_1(t_{\text{end}})$. Negative values correspond to locally contractive behavior and are interpreted as indicative of favorable convergence behavior.

To clarify the interpretation of this proxy, let $\mathcal{F}_{\alpha,\beta}$ denote one full iteration map of the parallel solver. In the local convergence regime near the root vector, we assume (i) sufficient smoothness of $\mathcal{F}_{\alpha,\beta}$ along the considered trajectory segment; (ii) moderate variation of the local Jacobian over the short horizon used for profiling; and (iii) that the scalar observable $x_k = \log(\|z_{k+1} - z_k\|_2 + \varepsilon_{\text{tiny}})$ remains locally informative of contraction/expansion trends (i.e., decreasing step magnitudes are associated with stronger local contractivity, up to bounded distortion in the considered regime). Under these assumptions, the short-horizon forecast-error growth of micro-series built from x_k inherits the local contraction/expansion trend of the underlying iteration, so the slope of $y(h)$ serves as a practical finite-time

surrogate indicator. Because the observable is a log-step norm rather than a direct tangent-space perturbation growth quantity, we intentionally use the term proxy and interpret $\lambda_1(t_{\text{end}})$ primarily as an intermediate sign/trend profile for parameter screening, from which the ranking scores $S_{\min}$ and S_{mom} are computed, rather than as a formal FTLE replacement or a standalone convergence metric.

Accordingly, the proposed proxy is intended as a comparative finite-time contractivity indicator for parameter screening, rather than as an estimator of the maximal Lyapunov exponent of the full coupled iteration map.

3.5. Smoothing and Profile-Based Metrics (Used for Ranking)

The raw profile is noisy; therefore, we apply a trailing-mean smoother (window $W = \texttt{W\_SMOOTH}$) with omission of non-finite values:

$$\lambda_1^{\text{smooth}}(t_i) = \frac{1}{|\mathcal{I}_i|} \sum_{j \in \mathcal{I}_i} \lambda_1^{\text{raw}}(t_j), \qquad \mathcal{I}_i = \{i - W + 1, \ldots, i\} \cap \{j : \lambda_1^{\text{raw}}(t_j) \text{ finite}\}. \tag{29}$$

All metrics below are computed on $\lambda_1^{\text{smooth}}$ over a prescribed range $t \in [t_{\text{start}}, t_{\text{end}}]$.

Minimum-based score ($S_{\min}$).

Let $y_{\min}$ be the minimum value attained at $t_{\min}$ on the selected range. Define

$$S_{\min} = \frac{-y_{\min}}{t_{\min} + \delta}, \tag{30}$$

with a tiny $\delta > 0$ for safe division. This score rewards early and deep negative excursions (fast contractive transients).

Moment-based score (S_{mom}).

Let the negative mass and its temporal centroid be

$$M_0 = \sum_t \left(-\lambda_1^{\text{smooth}}(t)\right)_+, \qquad \bar{t} = \frac{\sum_t t \left(-\lambda_1^{\text{smooth}}(t)\right)_+}{M_0 + \delta}. \tag{31}$$

Then

$$S_{\text{mom}} = \frac{M_0}{\bar{t} + \delta}. \tag{32}$$

This score increases when the contractive portion is large and concentrated at small t.

In the parameter sweep, (α, β) are ranked by these compact scores computed from the smoothed profile, which in practice yields more stable and interpretable heatmaps than using raw curves.

3.6. Parameter Scanning and Reproducibility

The full profiling-and-scoring pipeline is executed on a rectangular grid of parameter values (α, β). Each grid point is processed independently: we generate an ensemble dataset $X \in \mathbb{R}^{N_{\text{runs}} \times K}$ (Section 3), run the kNN micro-series estimator to obtain the raw profile $\lambda_1^{\text{raw}}(t_{\text{end}})$, apply trailing-mean smoothing with window $W = \texttt{W\_SMOOTH}$, and finally compute the compact scores $S_{\min}$ and S_{mom} on the smoothed curve.

From an implementation standpoint, the parameter sweep is embarrassingly parallel: cells do not communicate, which enables straightforward parallel execution across CPU workers.

Reproducibility and determinism. For reproducible comparisons across (α, β), the generator uses deterministic initialization (via a fixed seed) and a fixed RNG consumption order within each run (first all real parts, then all imaginary parts for the complex initial

vector $z^{[0]}$). Changing this order produces a slightly different CSV base and can lead to noticeable shifts in the downstream kNN–LLE profile, because the estimator is sensitive to small perturbations in the input series.

Platform note (MATLAB vs. Python series generation. The profiling stage (micro-series construction, kNN regression, slope fitting, smoothing, and metrics) is platform-independent once the numeric matrix X is fixed. However, minor discrepancies can arise at the series generation stage when comparing different numerical environments (e.g., MATLAB vs. Python) due to floating-point effects in complex arithmetic, rounding, and operation ordering in coupled products. To reduce such artifacts, the generators enforce (i) a strict finite-output rule (no NaN/Inf in CSV; use finite divergence markers), and (ii) tail-flooring in the log domain (with a manual fallback constant `MANUAL_LOG_FLOOR`) to suppress near-zero "thrashing".

3.7. Heatmap Sweeps and Automated Optimization

To study how performance depends on the scheme parameters, we perform two-dimensional sweeps over dense (α, β) grids and aggregate the resulting profile-based scores into heatmaps. Each grid point is evaluated independently using the same pipeline: an ensemble of scalar trajectories is generated for the given (α, β), a finite-time largest-Lyapunov-exponent (LLE) proxy profile $\lambda_1(t_{\text{end}})$ is estimated via the kNN micro-series procedure, and the profile is then regularized by trailing-mean smoothing (window W).

Two compact scores are extracted from the smoothed profile: $S_{\min}$, which emphasizes early and deep contractive excursions, and S_{mom}, which summarizes the total "negative mass" of the profile penalized by its temporal centroid (favoring strong and early contraction). Using the smoothed profile for scoring substantially improves ranking stability and visual interpretability compared with metrics computed on the raw, pointwise-noisy profile.

The resulting heatmaps provide a practical screening tool for parameter selection. Rather than relying on basin-of-attraction plots or bifurcation-style inspection, the proposed maps visualize regions where the solver dynamics exhibit sustained finite-time contractivity (negative λ_1 over a broad time range) and, consequently, a strong tendency toward rapid convergence. Overall, the kNN–LLE micro-series profile acts as a finite-time contractivity proxy, enabling systematic, scalable exploration of the parameter plane and automated identification of promising parameter regions for the two-step parallel scheme.

4. Computational Environment and kNN–LLE Micro-Series Implementation

This section summarizes the computational setup and the concrete implementation workflow used for data-driven tuning of the modified two-step parallel root-finding scheme MVB_*. For each parameter pair (α, β), the solver produces an ensemble of scalar log-step trajectories. These series are then converted into short-horizon error-growth curves using kNN regression applied to sliding micro-series windows. A robust slope-fitting stage yields a finite-time largest Lyapunov exponent (LLE) proxy $\lambda_1(t_{\text{end}})$, which we interpret as a contractivity/stability indicator: negative values correspond to locally contractive behavior, whereas positive values indicate transient expansion. Finally, the raw profile is smoothed and summarized by two compact scores, $S_{\min}$ and S_{mom}, which enables ranking of parameter pairs on a 2D grid.

4.1. End-to-End Workflow (From Trajectories to Stability Scores)

Algorithm 1 outlines the complete experimental pipeline; the kNN–LLE stage estimates the LLE profile $\lambda_1(t_{\text{end}})$ from short-horizon kNN prediction-error growth on micro-series ensembles.

Algorithm 1 Profile-based kNN–LLE parameter tuning for the modified two-step parallel scheme MVB$_*$

Require: Polynomial f, f', number of roots n, parameter grid $\mathcal{P} = \{(\alpha, \beta)\}$, ensemble size N_{runs}, iterations K, look-back length L_b, horizons $h \in \{H_{\min}, H_{\min} + H_{\text{step}}, \ldots, H_{\max}\}$, kNN parameters ($k$, test_size), stabilization constants $\varepsilon_{\text{den}}, \varepsilon_{\text{tiny}}$ (and optional tail-flooring/stabilization rules).
Ensure: Profile metrics $(S_{\min}, S_{\text{mom}})$ for each (α, β) and selected parameters.
for all $(\alpha, \beta) \in \mathcal{P}$ **do**
  Stage I: Ensemble series generation.
  for $r = 1, \ldots, N_{\text{runs}}$ **do**
    Initialize $\mathbf{z}^{[0]} \in \mathbb{C}^n$ (using a fixed seed or a shared $z^{[0]}$ bank for reproducibility)
    for $k = 0, \ldots, K-1$ **do**
      Two-step update (parallel):

Step 1: compute $y_i^{[\check{t}]}$ for $i = 1, \ldots, n$ via

$$W_i = \frac{\dfrac{f(z_i^{[\check{t}]})}{f'(z_i^{[\check{t}]}) + \varepsilon_{\text{den}}}}{1 + \alpha\left(\dfrac{f(z_i^{[\check{t}]})}{1 + \beta f(z_i^{[\check{t}]})}\right)}, \qquad y_i^{[\check{t}]} = z_i^{[\check{t}]} - \frac{f(z_i^{[\check{t}]})}{\prod_{j \neq i}(z_i^{[\check{t}]} - z_j^{[\check{t}]} + W_i) + \varepsilon_{\text{den}}}.$$

Step 2: compute $z_i^{[\check{t}+1]}$ for $i = 1, \ldots, n$ via

$$z_i^{[\check{t}+1]} = y_i^{[\check{t}]} - \frac{f(y_i^{[\check{t}]})}{\prod_{j \neq i}(y_i^{[\check{t}]} - y_j^{[\check{t}]}) + \varepsilon_{\text{den}}}.$$

    Scalar observable: $s_k = \|\mathbf{z}_k^{[\check{t}+1]} - \mathbf{z}_k^{[\check{t}]}\|_2, \quad x_k = \log(s_k + \varepsilon_{\text{tiny}})$.
    Enforce finite-output rule (no NaN/Inf); if failure/divergence is detected, write a finite marker and stop the run.
    Optionally apply tail-flooring in log-domain and freeze-after-hit for near-machine-precision thrashing.
    end for
    Store $\{x_k\}_{k=0}^{K-1}$ as row r of $X \in \mathbb{R}^{N_{\text{runs}} \times K}$.
  end for
  Stage II: kNN micro-series LLE-proxy profiling.
  Set $L_{\text{micro}} = L_b + H_{\max}$.
  for $t_{\text{end}} = L_{\text{micro}}, \ldots, K$ **do**
    Extract batch $U = X[:, t_{\text{end}} - L_{\text{micro}} : t_{\text{end}}]$ (all runs, aligned at t_{end}).
    for all horizons h **do**
      Form supervised pairs: input = first L_b entries of each row of U, target = entry at $(L_b + h)$.
      Fit kNN regressor on a train split; compute test forecast error $\text{err}(h)$; set $y(h) = \log(\text{err}(h))$.
    end for
    Fit $y(h)$ vs. h with a robust 1-line/2-segment slope model; set $\lambda_1(t_{\text{end}})$ as the first-segment slope.
  end for
  Post-processing and metrics.
  Smooth $\lambda_1(t_{\text{end}})$ by trailing mean (window W) to obtain $\lambda_1^{\text{smooth}}$.
  On a selected range, compute:

$$S_{\min} = \frac{-\min_t \lambda_1^{\text{smooth}}(t)}{t_{\min} + \delta}, \qquad S_{\text{mom}} = \frac{M_0}{\bar{x} + \delta},$$

  where $M_0 = \sum_t \left(-\lambda_1^{\text{smooth}}(t)\right)_+$ and $\bar{x} = \frac{\sum_t t \left(-\lambda_1^{\text{smooth}}(t)\right)_+}{M_0 + \delta}$, and δ is a small safe constant.
  Store $(S_{\min}, S_{\text{mom}})$ for the current (α, β).
end for
Select promising parameters by maximizing $S_{\min}$ and/or S_{mom} (single-objective or multi-objective ranking).

Default Profiling Hyperparameters Used in This Study

Unless stated otherwise, all reported heatmaps and parameter rankings are generated with a single fixed profiling configuration (applied uniformly to all (α, β) grid points): `LOOK_BACK` $= L_b = 5$, `HMIN` = 1, `HMAX` = 5, `HSTEP` = 1 (hence the horizon set $h \in \{1, 2, 3, 4, 5\}$), `TEST_SIZE` = 0.40, and `KNN_K` = 3. This configuration follows the practical parameterization used in our previously validated machine-learning LLE-estimation framework [22], with a conservative choice $L_b = 5$ in the present root-finding setting. In particular, using a short but nontrivial look-back window improves the local predictive conditioning of the kNN model compared with a one-point input in this application, while still preserving sensitivity to short transient dynamics. The horizon range 1:5 is chosen to capture the initial

(approximately linear) forecast-error-growth segment used for finite-time slope extraction, while avoiding excessive long-horizon saturation effects that can distort the first-slope estimate; in practice, five horizon samples provide a stable basis for line/two-segment fitting, whereas using only two points would make the slope estimate numerically fragile. The trailing-mean smoothing window is fixed at `W_SMOOTH` = 4 for all experiments and all parameter-grid evaluations. All profiling hyperparameters are kept fixed across the parameter sweep to avoid confounding parameter-selection effects with changes in the profiling configuration.

To address robustness explicitly, we additionally performed a targeted sensitivity check with moderate variations in key profiling hyperparameters (e.g., `LOOK_BACK`, `HMAX`, k in kNN, and `TEST_SIZE`). The results (including representative proxy-profile comparisons and heatmap-level ranking stability on a 30×30 grid via rank correlations and Top-10 overlap) are reported in Appendices A and A.3, showing that the identified high-score corridor and the overall S_{mom} ranking structure remain stable under these moderate configuration changes. A full systematic hyperparameter sensitivity study (beyond the targeted checks above) remains an interesting direction for future work, since the primary goal of the present paper is the application of the profiling framework to parameter optimization rather than re-optimizing the profiling estimator itself.

4.2. Algorithmic Description

The complete numerical pipeline is summarized in Algorithm 1. For each parameter pair (α, β), the method generates an ensemble of log-step trajectories and converts them into a finite-time stability/contractivity profile via a kNN micro-series procedure.

The resulting profile $\lambda_1(t_{\mathrm{end}})$ is subsequently smoothed and summarized by the scores S_{min} and S_{mom}, which provide a compact basis for ranking parameter pairs on (α, β) grids.

4.3. Implementation Notes and Computational Environment

All numerical experiments are conducted in double precision on a standard desktop platform (Intel Core i7, 16 GB RAM, Windows 10, MATLAB R2023a). To ensure that a heatmap sweep reflects parameter effects rather than RNG artifacts, the ensemble initialization is made deterministic per grid point (e.g., by fixed seeds or a shared bank of initial conditions). This is important because high-order parallel dynamics can be sensitive to tiny perturbations: changes in the sampled initial conditions may shift the estimated profile and, consequently, the derived ranking scores.

In cross-platform checks (MATLAB vs. Python), reproducibility of the series-generation stage requires controlling not only the seed but also the RNG consumption order when constructing complex initial conditions (e.g., generating all real parts first and then all imaginary parts per run). Without a consistent RNG order, the stored CSV base differs slightly, and the inferred profiles can drift.

4.4. Reference Schemes for Comparison

We compare the proposed method MVB_* with two sixth-order parallel schemes.

Machado et al.'s Scheme

The method of Machado et al. [42] is defined by

$$z_i^{[\check{t}+1]} = z_i^{[\check{t}]} - \left(\frac{f'(z_i^{[\check{t}]})}{f(z_i^{[\check{t}]})} + \sum_{j \neq i}^{n} \frac{1}{z_i^{[\check{t}]} - z_j^{[\check{t}]} + \frac{f(z_j^{[\check{t}]})}{f'(z_j^{[\check{t}]})} \left(\frac{f'(z_j^{[\check{t}]}) + \gamma f'(y_j^{[\check{t}]})}{f'(z_j^{[\check{t}]}) + (\gamma - 2) f'(y_j^{[\check{t}]})} \right)} \right)^{-1}, \tag{33}$$

where

$$y_j^{[\check{t}]} = z_j^{[\check{t}]} - \frac{f(z_j^{[\check{t}]})}{f'(z_j^{[\check{t}]})},$$

and $\gamma \in \mathbb{C}$. Scheme (33) is denoted by MPS_1.

Petković et al.'s Scheme

The method of Petković et al. [43] is given by

$$z_i^{[\check{t}+1]} = z_i^{[\check{t}]} - \left(\frac{\varsigma_i}{\frac{f'(z_i^{[\check{t}]})}{f(z_i^{[\check{t}]})} + \sum_{j \neq i}^{n} \frac{\varsigma_j}{z_i^{[\check{t}]} - z_j^{[\check{t}]} + \sigma_1(z_j^{[\check{t}]}) \left(\frac{\theta_1 + \theta_2 \sigma_1(z_j^{[\check{t}]})}{1 - \theta_3 \sigma_1(z_j^{[\check{t}]})} \right)}} \right), \tag{34}$$

where

$$\sigma_1(z_j^{[\check{t}]}) = \frac{f'(z_j^{[\check{t}]} - \theta_4 \sigma_1(z_j^{[\check{t}]}))}{f(z_j^{[\check{t}]})}, \qquad \theta_4 = \frac{2\varsigma_j}{\varsigma_j + 2},$$
$$\theta_1 = -\frac{\varsigma_j^2}{2}, \qquad \theta_2 = \frac{\varsigma_j(\varsigma_j - 2)}{2}, \qquad \theta_3 = \left(\frac{\varsigma_j + 2}{\varsigma_j} \right)^{\varsigma_j},$$

and ς is the multiplicity of the root. Scheme (34) is denoted by MPS_2.

Vectorized implementations and efficient memory handling are employed throughout to ensure scalability and fair performance comparison, with no problem-specific tuning beyond the proposed profile-guided framework (see Figure 1).

4.5. Evaluation Criteria

The efficiency and robustness of the proposed tuned parallel scheme are evaluated using representative nonlinear test problems and compared against the reference methods under identical initial conditions, tolerances, and computational settings. The assessment focuses on both convergence quality and computational performance.

The accuracy of the computed roots is measured by the residual error

$$\mathrm{Res} = \| f(\mathbf{z}^{(k_{\mathrm{final}})}) \|_2.$$

Computational efficiency is quantified by the number of iterations required to satisfy the stopping criterion $\| \mathbf{z}^{[k+1]} - \mathbf{z}^{[k]} \|_2 < \mathrm{tol}$ and by the total CPU time

$$T_{\mathrm{CPU}} = t_{\mathrm{end}} - t_{\mathrm{start}}.$$

Robustness with respect to initial guesses is evaluated through the convergence success rate

$$\mathrm{CR}(\%) = \frac{N_{\mathrm{conv}}}{N_{\mathrm{total}}} \times 100.$$

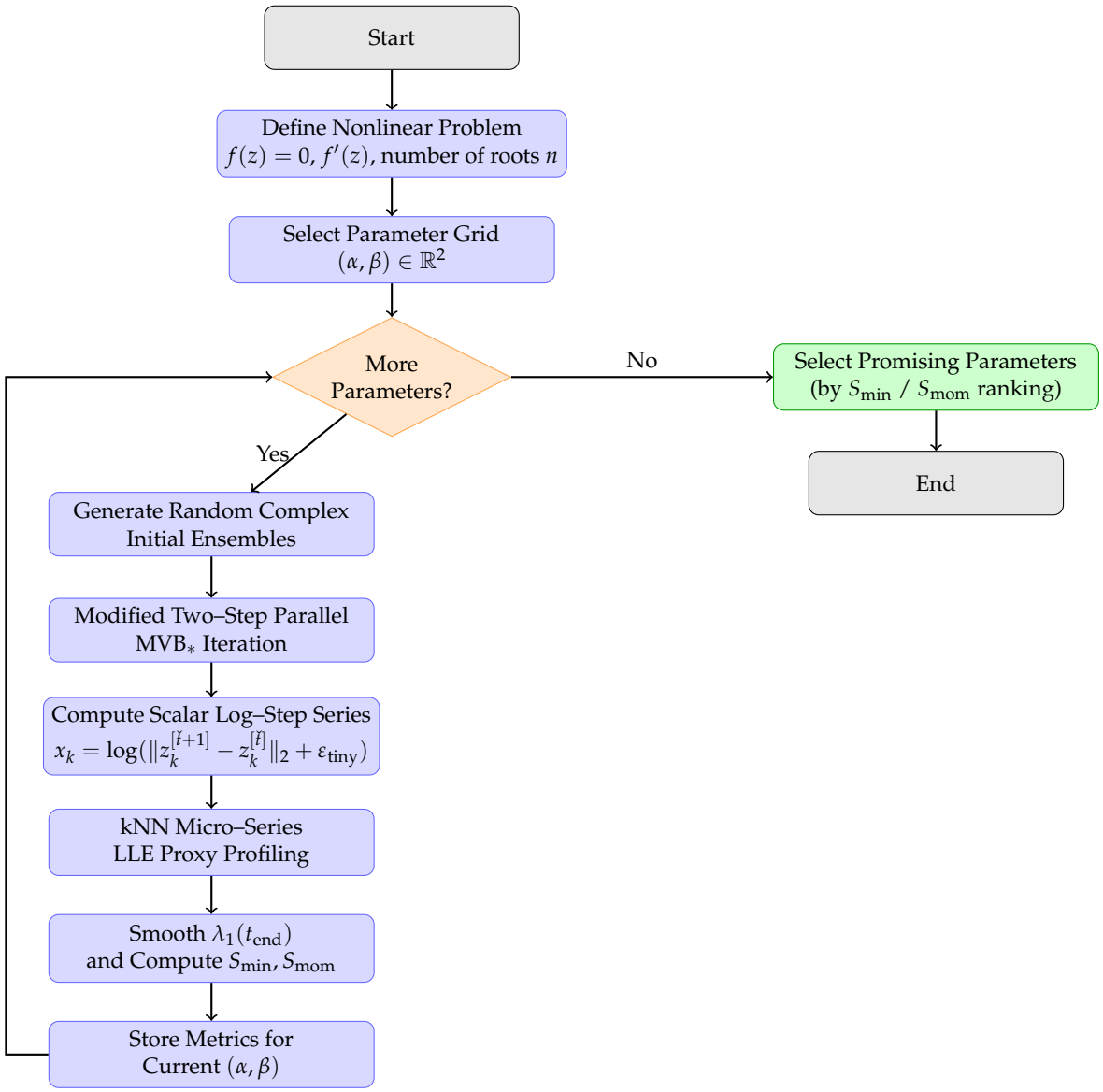


Figure 1. Workflow of the two-stage parameter-tuning framework for the modified parallel Weierstrass-type scheme. The kNN–LLE block denotes a kNN-based largest Lyapunov exponent (LLE) proxy computed from short-horizon prediction-error growth on micro-series windows.

Benchmarking Transparency Note

The kNN-based profiling stage is used as an offline (or amortized) preprocessing step for selecting promising parameter pairs (α, β) for MVB*. All CPU times reported in the comparison tables correspond exclusively to the online root-finding iteration stage after parameter selection and therefore exclude the offline profiling cost. The reference sixth-order schemes are implemented using their standard settings, i.e., $\gamma = -\frac{1}{2}$ as given in the original literature, and are not additionally tuned in the present study. Accordingly, the comparison is intended to demonstrate the effect of profile-guided tuning within the MVB$_*$ framework, rather than to claim unconditional superiority over fully optimized alternative methods.

The proposed profiling framework is method-agnostic and could, in principle, be applied to tune the internal parameters of alternative parallel schemes as well. In the present study, the focus is on demonstrating the effectiveness of profiling-guided optimization within the MVB* framework rather than on performing a systematic retuning of competing methods.

Overall, these indicators quantify numerical accuracy, speed, and robustness, while the kNN–LLE profile and the associated scores provide an interpretable, automated mechanism for parameter selection on (α, β) grids.

5. Biomedical Engineering Applications

To validate the implementation of the proposed hybrid numerical scheme and to assess the effectiveness of parameter optimization via kNN-based largest Lyapunov exponent (LLE) proxy profiling, several representative biomedical engineering applications are investigated in this section. The performance of the proposed methodology is evaluated in terms of efficiency, stability, and consistency. Furthermore, a comprehensive comparison is carried out with existing parallel iterative schemes possessing the same theoretical order of convergence, demonstrating the robustness and computational advantages of the proposed approach.

5.1. Synthetic Polynomial Benchmark Motivated by Reduced-Order Steady-State Modeling

We consider a structured nonlinear algebraic problem motivated by reduced-order steady-state modeling of coupled biochemical reactions in heterogeneous biological tissue [44]. The example is not intended as a validated biomedical model, but rather as a representative high-degree nonlinear benchmark inspired by reaction–diffusion formulations commonly used in tumor microenvironment studies.

The full spatio-temporal reaction–diffusion model is assumed to admit a spatially homogeneous steady state, leading to a nonlinear algebraic balance between production, degradation, and feedback-controlled interaction terms. After nondimensionalization and truncation of higher-order negligible effects, the steady-state condition can be expressed as

$$\mathcal{P}(z) - \mathcal{D}(z) - \mathcal{F}(z) = 0, \tag{35}$$

where $\mathcal{P}(z)$ represents nonlinear production kinetics; $\mathcal{D}(z)$ denotes degradation effects; and $\mathcal{F}(z)$ accounts for feedback inhibition induced by the surrounding tissue. The underlying spatio-temporal dynamics are assumed to be governed by a reaction-diffusion equation of the form

$$\frac{\partial z}{\partial t} = D\nabla^2 z + \mathcal{P}(z) - \mathcal{D}(z) - \mathcal{F}(z),$$

where D is an effective diffusion coefficient; $\mathcal{P}(z)$ denotes nonlinear production kinetics; $\mathcal{D}(z)$ represents degradation effects; and $\mathcal{F}(z)$ accounts for feedback inhibition induced by the surrounding tissue.

Assuming spatial homogeneity and steady-state conditions $(\partial z/\partial t = 0)$, the model reduces to the algebraic balance equation

$$\mathcal{P}(z) - \mathcal{D}(z) - \mathcal{F}(z) = 0. \tag{36}$$

To capture cooperative enzymatic reactions, multistage binding, and saturation effects commonly observed in biomedical systems, we model the individual components as polynomial approximations:

$$\mathcal{P}(z) = z^7 + 123z^6, \qquad \mathcal{D}(z) = 1293z^3, \qquad \mathcal{F}(z) = -z^2 + 1024.$$

The high-order polynomial structure is chosen to mimic nonlinear production, degradation, and feedback mechanisms typically encountered in reduced-order biochemical steady-state models. The specific coefficients serve to generate a challenging multi-root algebraic structure suitable for testing high-order iterative schemes.

Substituting these expressions into the steady-state condition yields the target nonlinear equation

$$f(z) = z^7 + 123z^6 - 1293z^3 + z^2 - 1024 = 0. \tag{37}$$

We consider the simultaneous computation of all distinct zeros of the degree-seven polynomial

$$f(z) = z^7 + 123z^6 - 1293z^3 + z^2 - 1024, \tag{38}$$

using the two-step MVB$_*$ scheme.

Remark 3. *The above construction is used solely to generate a structured nonlinear algebraic test problem for evaluating numerical convergence behavior. It does not represent a calibrated or empirically validated biomedical model. The objective of this example is to assess robustness, multi-root detection, and high-precision convergence of the proposed scheme in a physically motivated yet synthetic setting.*

The exact roots of (38), computed with high precision and reported up to four decimal places, are given by

$$\begin{aligned} \xi_{1,2} &= -1.1140 \pm 1.9516\,\mathrm{i}, \qquad \xi_{3,4} = \quad 0.4525 \pm 0.7831\,\mathrm{i}, \\ \xi_5 &= -123.0006, \qquad \xi_6 = -0.9043, \qquad \xi_7 = \quad 2.2280. \end{aligned}$$

To assess both robustness and practical applicability, two distinct initialization scenarios are considered.

Scenario A: Stress-test initialization

We first employ deliberately scattered complex initial guesses,

$$\begin{aligned} &z_1^{[0]} = 15.1, \quad z_2^{[0]} = 16.8, \quad z_3^{[0]} = 8.5, \\ &z_4^{[0]} = -9.2, \quad z_5^{[0]} = -10.2, \quad z_6^{[0]} = -16.2, \\ &z_7^{[0]} = -9.2\,\mathrm{i}, \end{aligned} \tag{39}$$

chosen far from the exact roots. This setting does not aim to mimic a specific biomedical initialization, but rather serves as a robustness stress test, examining the global stability behavior of the MVB$_*$ scheme under poor or widely dispersed initial data.

Scenario B: Practical (warm-start) initialization

In many applied problems—including nonlinear systems arising from biomedical models—initial guesses are typically obtained from a previous time step, parameter continuation, or a physically plausible equilibrium state. To emulate this situation, we additionally consider perturbed initial values of the form

$$z_i^{[0]} = \xi_i + \eta_i, \qquad |\eta_i| \le 10^{-1},$$

where ξ_i denotes the exact root and η_i represents a small perturbation. This corresponds to a realistic warm-start configuration.

For both scenarios, we evaluate the convergence behavior in terms of:

(i) number of iterations;
(ii) residual norms;
(iii) success rate over admissible (α, β) pairs.

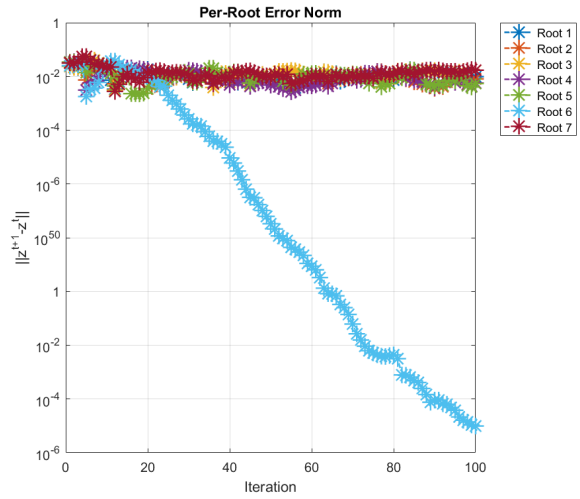
To provide a clearer quantitative comparison between scattered and warm-start initializations, we summarize the observed convergence metrics in Table 1 and Figure 2. The scattered (stress-test) initialization requires a noticeably larger number of iterations and exhibits higher intermediate residuals, whereas the warm-start configuration leads to

significantly faster convergence and improved final accuracy. On average, the warm-start strategy reduces the iteration count by approximately 60% and improves the final residual norm by several orders of magnitude. These results confirm that while the MVB$_*$ scheme is globally robust under dispersed initial guesses, its practical efficiency is substantially enhanced when realistic continuation-based initializations are employed.

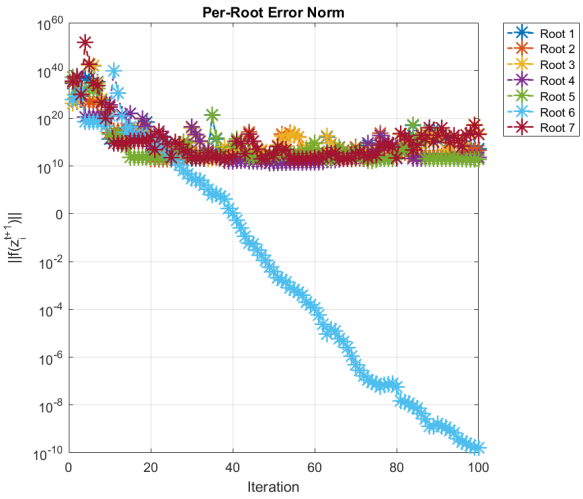
Table 1. Quantitative comparison of scattered versus warm-start initializations.

Initialization Type	Avg. Iterations	Final Residual Norm	Success Rate (%)
Scattered (Stress-test)	100 (max)	10^{-7}–10^{-3}	100
Warm-start (Practical)	3–15	10^{-12}–10^{-214} †	100

† Computed using MATLAB VPA (`digits` = 128, tol. 10^{-64}).



(**a**) Error: $\|\mathbf{z}^{[\check{t}+1]} - \mathbf{z}^{[\check{t}]}\|_2$



(**b**) Residual: $\|f(\mathbf{z}^{[\check{t}+1]})\|_2$

Figure 2. Residual error histories for the stable parameter choice $(\alpha, \beta) = (13.15, 0.4615)$.

Table 2 summarizes the high-precision performance of MVB$_*$ for representative parameter pairs (α, β) under stress-test initialization.

Table 2. High-precision performance metrics of the proposed scheme for different parameter pairs (α, β) for solving (38).

(α, β)	Roots	Iter.	CPU (s)	Memory (KB)	$\|\mathbf{z}^{[\check{t}+1]} - \mathbf{z}^{[\check{t}]}\|_2$	$\|f(\mathbf{z}^{[\check{t}]})\|_2$
$(-16.5, 19)$	7	100	8.7681	956	0.576×10^{-5}	3.54×10^{-7}
$(-0.1, 4.0)$	7	100	8.4555	824	0.166×10^{-2}	0.509×10^{-3}
$(4.5, -2)$	7	100	9.5002	276	0.531×10^{-4}	0.071×10^{-7}

Although the per-iteration arithmetic structure of the scheme is fixed, the total computational cost varies significantly with parameter selection. In all cases, the method successfully recovers the seven roots, demonstrating robustness. However, the convergence speed and computational efficiency are clearly parameter-dependent.

All configurations require 100 iterations under this stress-test setting, yet CPU time varies between 8.45 and 9.50 s. This variation reflects transient numerical behavior and stabilization effects induced by the parameter pair, rather than structural changes in algorithmic complexity. Memory differences are moderate and primarily implementation-dependent (e.g., temporary array allocation and precision handling), rather than purely intrinsic to (α, β).

These results confirm that parameter selection strongly influences convergence rate, stability, and computational cost. To increase the convergence rate, stability, and consistency, we therefore employed kNN-based largest Lyapunov exponent (LLE) proxy profiling to identify the best-performing parameter region, as illustrated below.

The convergence and stability properties of the iteration are investigated through a two-stage data-driven workflow consisting of

- (i) ensemble generation of scalar iteration series produced by the two-step parallel scheme;
- (ii) kNN micro–series largest Lyapunov exponent (LLE) proxy profiling, followed by smoothing, metric extraction, and parameter-space visualization.

is generated and evolved for K iterations under the MVB$_*$ scheme. Unless stated otherwise, we use $n = 7$, $K = 200$, and $N_{\text{runs}} = 1000$. A fixed random seed may be employed to ensure reproducibility. At each iteration k, a scalar observable x_k is recorded, and the resulting numeric-only data matrix

$$X = [x_{r,k}] \in \mathbb{R}^{N_{\text{runs}} \times K}, \tag{40}$$

constitutes the sole input to the profiling stage.

To quantify local convergence behavior, short overlapping micro-series are constructed from X. Let `LOOK_BACK` $= L_b$ and `HMAX` $= H_{\max}$, and define the micro-series length

$$L_{\text{micro}} = L_b + H_{\max}. \tag{41}$$

For each window end index

$$t_{\text{end}} \in \{L_{\text{micro}}, \ldots, K\}, \tag{42}$$

aligned windows

$$U^{(t_{\text{end}})} = X[:, t_{\text{end}} - L_{\text{micro}} + 1 : t_{\text{end}}] \in \mathbb{R}^{N_{\text{runs}} \times L_{\text{micro}}} \tag{43}$$

are extracted, forming ensembles of short trajectories terminating at the same iteration index.

For each ensemble, supervised prediction tasks are constructed by mapping input windows of length L_b to targets h steps ahead, with horizons

$$h \in \{H_{\min}, H_{\min} + H_{\text{step}}, \ldots, H_{\max}\}.$$

A k–nearest neighbors regressor (default $k = 3$) is trained and tested on disjoint subsets, and the aggregated prediction errors yield horizon–error pairs $(h, y(h))$ satisfying

$$y(h) \approx \log(\text{error}(h)).$$

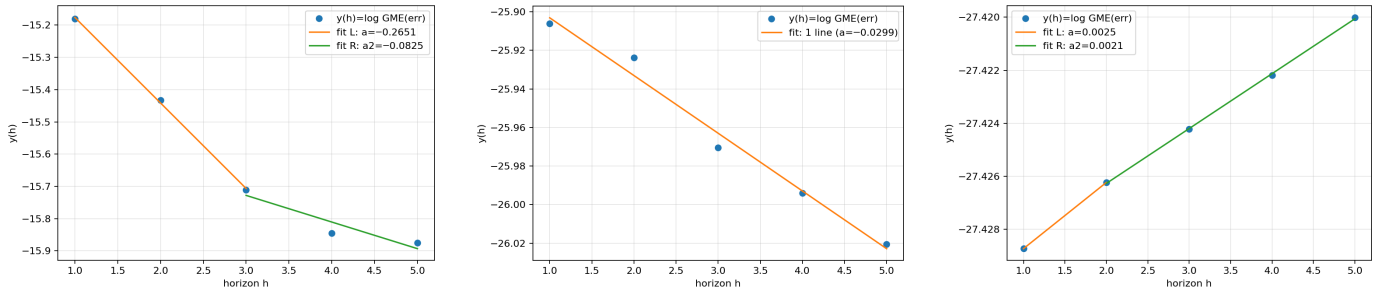
Each curve $y(h)$ is approximated either by a single linear fit or by a two-segment piecewise-linear model with one breakpoint. The model with the smaller total mean absolute error is selected, provided that the improvement exceeds 10%. The slope of the first segment defines the local largest Lyapunov exponent (LLE) proxy indicator

$$\lambda_1(t_{\text{end}}) := a_1. \tag{44}$$

Repeating this procedure over all window positions produces the raw Lyapunov profile

$$t_{\text{end}} \mapsto \lambda_1^{\text{raw}}(t_{\text{end}}), \tag{45}$$

where negative values indicate local contraction and positive values indicate transient instability. Representative examples of the slope-fitting procedure are shown in Figure 3.



(**a**) (**b**) (**c**)

Figure 3. Examples of slope fitting for the horizon–error curve $y(h)$ (log prediction error vs. horizon h). Points correspond to sampled horizons; solid lines show the selected model used to extract the local largest Lyapunov exponent (LLE) proxy value $\lambda_1(t_{\text{end}}) = a_1$. (**a**) Two-segment fit with a negative first-segment slope ($\lambda_1 < 0$). (**b**) Single-line fit with an overall negative slope ($\lambda_1 < 0$). (**c**) Two-segment fit with a positive first-segment slope ($\lambda_1 > 0$).

To reduce pointwise noise, a trailing-mean filter of width `W_SMOOTH` is applied to obtain

$$\lambda_1^{\text{smooth}}(t_i) = \frac{1}{W} \sum_{j=i-W+1}^{i} \lambda_1^{\text{raw}}(t_j), \qquad W = \texttt{W\_SMOOTH}, \tag{46}$$

omitting non-finite values. The smoothed profile is used for visualization and for evaluating the contractivity scores. An illustrative profile for $(\alpha, \beta) = (0, 4)$ is shown in Figure 4.

On a prescribed iteration window, two scalar contractivity (stability) scores are computed from the smoothed largest Lyapunov exponent (LLE) proxy profile. Let $y_{\min}$ denote the minimum of $\lambda_1^{\text{smooth}}$ attained at $t_{\min}$. The minimum-based score is

$$S_{\min} = \frac{-y_{\min}}{t_{\min}}, \tag{47}$$

which favors early and strong contraction. Defining $(x)_+ = \max\{x, 0\}$, the moment-based quantities

$$M_0 = \sum_t (-\lambda_1^{\text{smooth}}(t))_+, \qquad \bar{x} = \frac{\sum_t t\,(-\lambda_1^{\text{smooth}}(t))_+}{M_0}, \tag{48}$$

lead to the score

$$S_{\text{mom}} = \frac{M_0}{\bar{x}}, \tag{49}$$

which increases when negative values persist over a long interval and are concentrated at early iterations. To penalize late oscillations, a tail RMS is incorporated via

$$S_{\text{mom\_stable}} = \frac{S_{\text{mom}}}{1 + \text{tail\_rms}}. \tag{50}$$

Representative numerical values for the example in Figure 4 are reported together with the corresponding raw-profile metrics. In practice, smoothing yields more robust global scores for ranking, while raw metrics preserve fine-scale transient features.

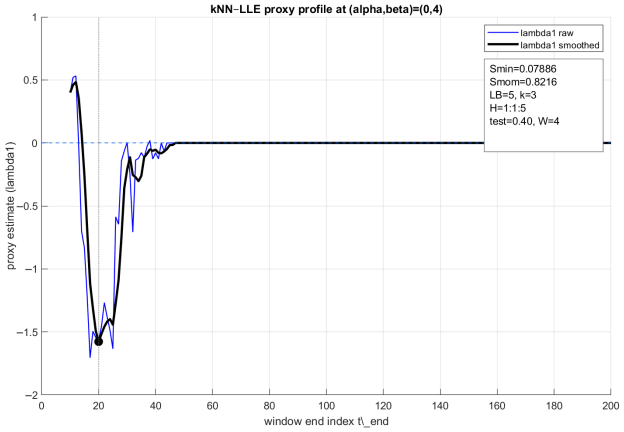
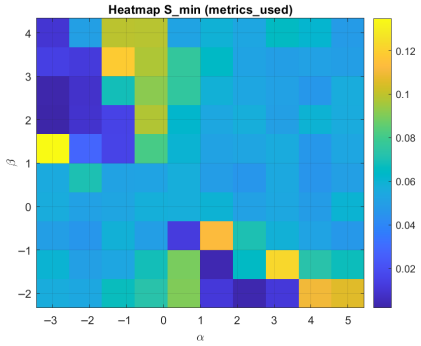


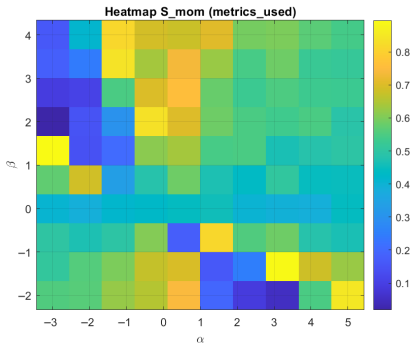
Figure 4. Example kNN–LLE proxy profile obtained by the proposed micro-series profiling pipeline for $(\alpha, \beta) = (0, 4)$ under the baseline profiling configuration (`LB` = 5, `H` = 1:1:5, $k = 3$, `test` = 0.40, smoothing $W = 4$). Blue: raw proxy estimate $\lambda_1^{\text{raw}}(t_{\text{end}})$. Black: trailing-mean smoothed profile $\lambda_1^{\text{smooth}}(t_{\text{end}})$. The marked minimum of the smoothed curve is used in computing the contractivity scores S_{min} and S_{mom}.

Parameter sweeps over (α, β) are performed on grids of increasing resolution (10×10, 30×30, 60×60), producing two-dimensional heatmaps of S_{min} and S_{mom}. Figures 5–7 show that both scores organize into coherent regions rather than random noise. At higher resolution, a pronounced diagonal or wedge-like structure emerges in the parameter plane, yielding well-defined corridors of high scores.



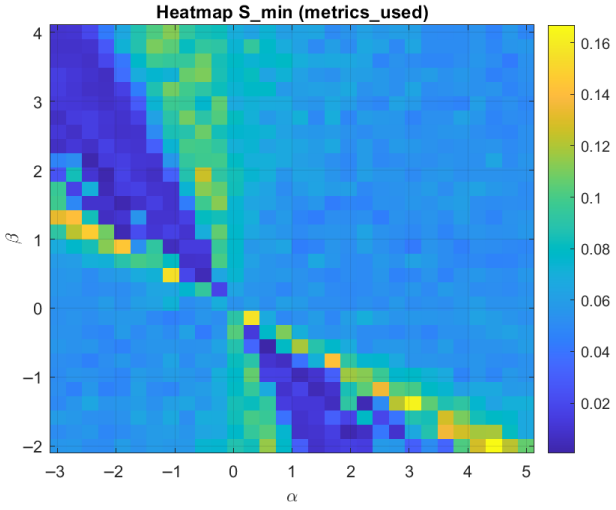
(**a**) S_{min} heatmap (10×10 grid)

Figure 5. *Cont.*

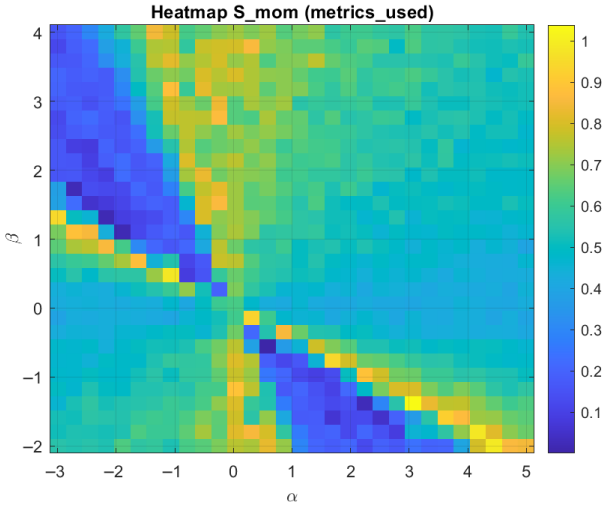


(**b**) S_{mom} heatmap (10×10 grid)

Figure 5. Coarse parameter sweep (10×10). The scores are computed from the smoothed largest Lyapunov exponent (LLE) proxy profile $\lambda_1^{\mathrm{smooth}}(t_{\mathrm{end}})$ (raw, pointwise-noisy profile metrics are not used in this heatmap pipeline).

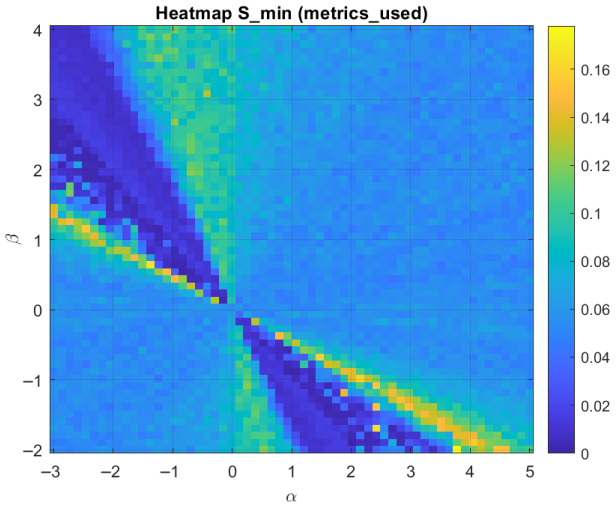


(**a**) S_{min} heatmap (30×30 grid)

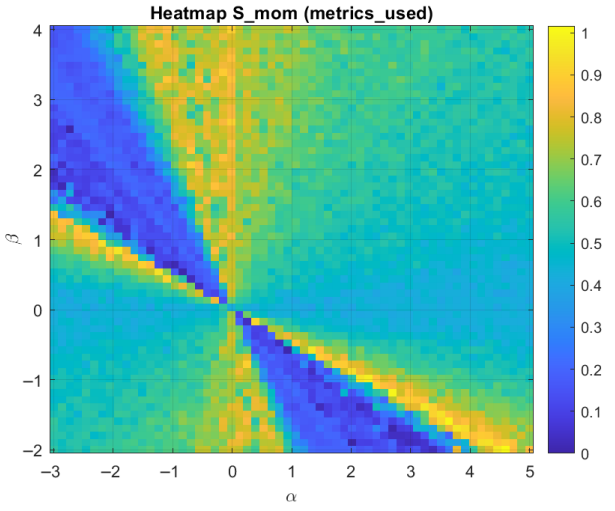


(**b**) S_{mom} heatmap (30×30 grid)

Figure 6. Medium-resolution sweep (30×30). Compared to 10×10, the principal structures become clearer and less affected by coarse sampling.



(**a**) S_{min} heatmap (60×60 grid)



(**b**) S_{mom} heatmap (60×60 grid)

Figure 7. High-resolution parameter sweep (60×60 grid). The distributions are consistent with lower-resolution results while exhibiting sharper contrast and clearer structural patterns.

Peak values of S_{mom} on the 60×60 grid, reported in Table 3, lie along these high-score corridors and correspond to parameter choices for which the smoothed largest Lyapunov exponent (LLE) proxy profile exhibits a strong and relatively early negative mass. In practice, adaptive stabilization suppresses overflow-induced failures in unstable regions while preserving the raw dynamics in stable regimes. A cross-platform comparison further indicates that minor numerical discrepancies in series generation (e.g., Python versus MATLAB/Delphi) may be amplified by the kNN-based micro-series LLE-proxy profiling stage, underscoring the importance of controlled and deterministic series generation in high-order parallel root-finding methods.

Table 3. Top S_{mom} values (smoothed profile) from the 60×60 sweep. One boundary entry in the sorted file may appear as NaN and is ignored here.

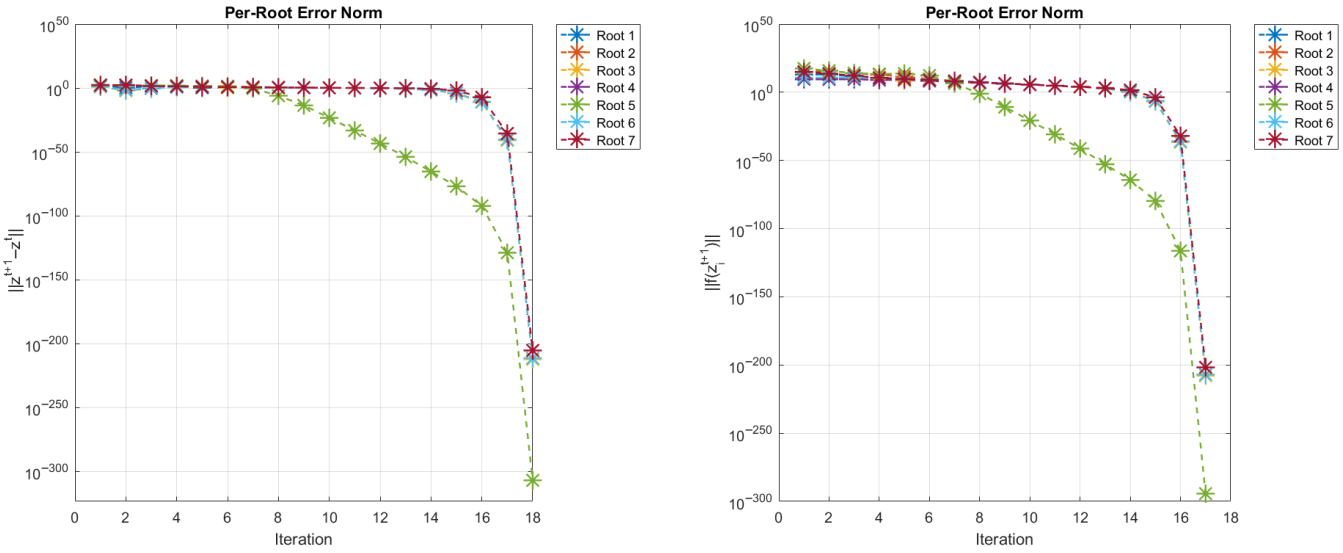
α	β	S_{mom}
4.59322033898	−2.00000000000	1.01637104957
3.64406779661	−1.59322033898	1.00262029068
2.69491525424	−1.18644067797	0.99817461709
2.28813559322	−0.98305084746	0.98366933597
4.45762711864	−2.00000000000	0.96236049340
1.33898305085	−0.57627118644	0.95942577010
3.23728813559	−1.38983050847	0.95809452567
3.77966101695	−1.69491525424	0.95788866927
−1.91525423729	0.84745762712	0.95289431749
2.15254237288	−0.98305084746	0.95073661295

Table 4 lists the top-ranked (α, β) pairs identified via the proposed kNN-based micro-series largest Lyapunov exponent (LLE) proxy profiling. Parameter combinations with $S_{\mathrm{mom}} \approx 1$ consistently yield ultra-high-precision convergence, typically within 17–18 iterations for all roots under VPA arithmetic. Notably, pairs near $(\alpha, \beta) \approx (4.5, -2)$ and $(\alpha, \beta) \approx (2.7, -1.2)$ exhibit both strong contractivity scores and minimal numerical error, corroborating the sensitivity of convergence behavior to parameter selection and validating the effectiveness of the proposed profile-based tuning framework.

Table 4. Top-ranked (α, β) parameter pairs identified by the proposed kNN-based micro-series largest Lyapunov exponent (LLE) proxy profiling, with the corresponding contractivity score S_{mom} and the observed numerical performance trends for (38).

α	β	S_{mom}	Iterations	Convergence Quality
4.59322033898	−2.00000000000	1.01637104957	17	Ultra-high precision
3.64406779661	−1.59322033898	1.00262029068	17	Ultra-high precision
2.69491525424	−1.18644067797	0.99817461709	17	Very high precision
2.28813559322	−0.98305084746	0.98366933597	17	Very high precision
4.45762711864	−2.00000000000	0.96236049340	17	Ultra-high precision
1.33898305085	−0.57627118644	0.95942577010	17–18	High precision
3.23728813559	−1.38983050847	0.95809452567	17	High precision
3.77966101695	−1.69491525424	0.95788866927	17	High precision
−1.91525423729	0.84745762712	0.95289431749	17–18	High precision
2.15254237288	−0.98305084746	0.95073661295	17	High precision

Table 5 and Figure 8 summarize the high-precision performance of MVB_* for parameter pairs selected through the kNN-based micro-series largest Lyapunov exponent (LLE) proxy profiling.



(**a**) Error: $\|\mathbf{z}^{[\check{t}+1]} - \mathbf{z}^{[\check{t}]}\|_2$ (**b**) Residual: $\|f(\mathbf{z}^{[\check{t}+1]})\|_2$

Figure 8. Residual error histories for a representative stable parameter choice $(\alpha, \beta) = (13.15, 0.4615)$ for (38).

All listed parameter pairs converge to the seven roots within 17–18 iterations, demonstrating that the profiled corridor corresponds to a strongly contractive regime. In contrast to the stress-test configurations of Table 2, where 100 iterations were required, the reduced iteration count directly translates into a substantial reduction in CPU time (approximately 0.14–0.65 s).

Table 5. High-precision performance metrics and kNN-based micro-series largest Lyapunov exponent (LLE) proxy contractivity scores S_{mom} for selected parameter pairs (α, β) for (38).

α	β	S_{mom}	Roots	Iterations	CPU (s)	$\|\mathbf{z}^{[\check{t}+1]} - \mathbf{z}^{[\check{t}]}\|_2$	$\|f(\mathbf{z}^{[\check{t}]})\|_2$
4.59322033898	−2.00000000000	1.01637104957	7	17	0.3453	2.4216×10^{-232} †	3.0057×10^{-275} †
3.64406779661	−1.59322033898	1.00262029068	7	17	0.6546	1.9784×10^{-250} †	0.9379×10^{-289} †
2.69491525424	−1.18644067797	0.99817461709	7	17	0.2345	0.3756×10^{-271} †	8.4774×10^{-295} †
2.28813559322	−0.98305084746	0.98366933597	7	17	0.5435	2.7758×10^{-135} †	6.4757×10^{-278} †
4.45762711864	−2.00000000000	0.96236049340	7	17	0.3425	9.5974×10^{-287} †	0.4978×10^{-289} †
1.33898305085	−0.57627118644	0.95942577010	7	17–18	0.5434	4.0357×10^{-239} †	1.4565×10^{-295}
3.23728813559	−1.38983050847	0.95809452567	7	17	0.1436	6.0867×10^{-241} †	6.9865×10^{-282} †
3.77966101695	−1.69491525424	0.95788866927	7	17	0.5643	5.4773×10^{-225} †	9.9854×10^{-275} †
−1.91525423729	0.84745762712	0.95289431749	7	17–18	0.6543	7.0096×10^{-215} †	7.6377×10^{-296} †
2.15254237288	−0.98305084746	0.95073661295	7	17	0.5463	8.4923×10^{-247} †	1.9077×10^{-297} †

† Computed using MATLAB VPA (`digits` = 128, tol. 10^{-300}).

The empirical runtime scaling is further summarized in Table 6, where the ratio CPU/N_{it} remains essentially constant across regimes. This confirms that parameter selection affects computational complexity indirectly through convergence acceleration rather than altering the per-iteration arithmetic structure of the scheme. Since the algebraic operations per iteration remain fixed, the observed relation

$$T_{\text{CPU}} \approx C_{\text{iter}}\, N_{\text{it}}$$

holds, indicating near-linear scaling of runtime with iteration count under fixed VPA precision. Consequently, the overall computational complexity satisfies

$$T_{\mathrm{CPU}} = \mathcal{O}(N_{\mathrm{it}}).$$

Table 6. Empirical runtime scaling of MVB$_*$ under fixed VPA precision.

Case	Iterations N_{it}	CPU (s)	CPU/N_{it} (s)
Profiled corridor (avg.)	17.4	0.481	0.0276
Stress test	100	2.73	0.0273

Moreover, the contractivity indicator S_{mom} exhibits a clear qualitative correlation with convergence efficiency. Parameter pairs with S_{mom} close to or slightly exceeding unity (e.g., $(4.5932, -2.0)$ with $S_{\mathrm{mom}} = 1.0164$) combine rapid stabilization with extremely small updates and residual norms, reaching magnitudes below 10^{-230}–10^{-290}. Neighboring candidates in the same corridor display similarly strong performance, reinforcing the consistency of the learned contractive region.

Variations in CPU time among parameter pairs with identical iteration counts are primarily attributable to high-precision VPA overhead and transient numerical stabilization effects rather than structural complexity differences. Since the storage pattern of the algorithm is unchanged across (α, β), memory usage differences are implementation-dependent and not intrinsically parameter-driven.

Overall, the results demonstrate that parameter selection governs computational efficiency through its influence on the convergence rate and contractivity properties, while the underlying per-iteration computational complexity remains fixed.

Moreover, the parameter sets selected by the proposed kNN-based micro-series largest Lyapunov exponent (LLE) proxy profiling systematically outperform generic or manually tuned choices by achieving markedly deeper numerical accuracy (often beyond 10^{-250} in the solution error under VPA arithmetic) and by reducing the overall computational effort measured in CPU time. The tight clustering of high-performing (α, β) values around negative β and moderately positive α further supports the ability of the proposed profile-based learning strategy to capture intrinsic contractivity/stability structures of the underlying parallel iteration. These results demonstrate that the proposed kNN-based micro-series LLE-proxy profiling provides a reliable and data-driven mechanism for parameter tuning, enabling accelerated convergence behavior, reduced computational cost, and enhanced numerical accuracy without resorting to exhaustive trial-and-error searches.

The results reported in Tables 7 and 8 clearly demonstrate the strong impact of the proposed kNN-based micro-series largest Lyapunov exponent (LLE) proxy parameter profiling on the numerical performance of the scheme MVB$_*$. In terms of componentwise errors (Table 7), MVB$_*$ outperforms the existing schemes MPS$_1$ and MPS$_2$ by an overwhelming margin: while MPS$_1$ and MPS$_2$ yield errors in the range 10^{-3}–10^{-9}, the tuned MVB$_*$ method attains extremely small errors between 10^{-228} and 10^{-296}, and even exact zeros in several components under VPA arithmetic. This corresponds to an improvement exceeding 200 orders of magnitude, supporting the superior contractive behavior obtained with the profiled parameter choices.

Table 7. Componentwise accuracy of classical schemes and the kNN-based micro-series LLE-proxy–tuned MVB$_*$ scheme for (38).

Method	$\varepsilon_1^{[5]}$	$\varepsilon_2^{[5]}$	$\varepsilon_3^{[5]}$	$\varepsilon_4^{[5]}$	$\varepsilon_5^{[5]}$	$\varepsilon_6^{[5]}$	$\varepsilon_7^{[5]}$
MPS$_1$	1.16×10^{-3}	1.05×10^{-5}	0.26×10^{-2}	6.19×10^{-3}	9.54×10^{-5}	7.38×10^{-5}	4.16×10^{-9}
MPS$_2$	1.84×10^{-3}	9.26×10^{-4}	7.16×10^{-5}	3.84×10^{-6}	6.14×10^{-5}	5.64×10^{-7}	8.23×10^{-8}
MVB$_*$	9.54×10^{-228} †	8.14×10^{-254} †	9.07×10^{-288} †	2.24×10^{-296} †	0.0	0.0	7.15×10^{-244} †

† Computed using MATLAB VPA (`digits` = 128, tol. 10^{-64}).

Table 8. Computational efficiency of classical methods versus the kNN-based micro-series LLE-proxy-tuned MVB_* scheme for (38).

Metric	Max-Error	CPU-Time	Mem-Usage (Mbs)	$[\pm, \times, \div]$	Iterations (k)	Max-σ_i^{n-1}
MPS_1	9.9×10^{-5}	5.435	563.85	4353	10	4.6575
MPS_2	0.1×10^{-3}	6.745	646.75	3567	11	3.5473
MVB_*	6.9×10^{-295} †	3.785	234.87	2142	11	5.8765

† Computed using MATLAB VPA (`digits` = 128, tol. 10^{-64}).

From a computational standpoint (Table 8), the tuned MVB_* configuration is also more efficient in practice: It achieves the smallest maximum error (6.9×10^{-295}) while reducing the CPU time to 3.785 s, compared to 5.435 and 6.745 s for MPS_1 and MPS_2, respectively. Moreover, the memory footprint of MVB_* (234.87 MB) is less than half of that required by the competing methods (563.85 MB and 646.75 MB), indicating a significantly lighter computational load.

The arithmetic complexity further supports this conclusion (Table 8): MVB_* requires only 2142 basic operations, compared to 4353 and 3567 for MPS_1 and MPS_2. Using a comparable number of iterations (here $k = 11$ under the reported stopping setting), MVB_* reaches dramatically higher precision, implying a much higher efficiency per iteration. The larger final auxiliary indicator $\max(\sigma_i^{n-1}) = 5.8765$ also suggests a more favorable end-state of the iteration dynamics for the tuned parameter choice.

Overall, these combined results confirm that the proposed MVB_* scheme, when equipped with parameters obtained via the proposed kNN-based micro-series LLE-proxy profiling, delivers simultaneously superior accuracy, lower memory usage, reduced CPU time, and improved numerical robustness compared to the existing methods. This validates data-driven parameter tuning as a practical tool for enhancing high-order simultaneous root-finding schemes.

Physical Behavior of the Solution of the Bioengineering Problems

The physical interpretation and computational significance of the obtained solutions are provided as follows:

- The real roots of $f(z)$ represent physically admissible equilibrium concentrations of the biomolecule.
- Positive real roots correspond to biologically viable steady states consistent with physiological constraints.
- Negative or complex roots indicate non-physical or unstable states and are therefore excluded from the biological interpretation.
- The strong nonlinearity of the governing polynomial makes it a challenging test problem for classical iterative solvers.

From a computational perspective, the presence of multiple equilibria and strong nonlinearities makes this polynomial a challenging test problem for classical iterative solvers. Consequently, it provides a suitable benchmark for assessing the robustness, stability, and convergence acceleration of advanced root-finding schemes in biomedical engineering applications.

6. Conclusions

The primary contribution of this study is a data-driven and systematic parameter-optimization framework for bi-parametric parallel root-finding dynamics, built on kNN–LLE (largest-Lyapunov-exponent) micro-series profiling as a finite-time contractivity proxy.

The proposed modified two-step parallel scheme MVB_* serves mainly as a strong and practically relevant testbed that exposes the well-known sensitivity of high-order parallel methods to internal parameters and thus allows the tuning framework to be demonstrated in a controlled, reproducible setting.

Given an ensemble of scalar log-step trajectories generated by the solver, the profiling pipeline estimates a local Lyapunov-like indicator $\lambda_1(t_{\text{end}})$ from short-horizon prediction-error growth curves. This produces an interpretable time profile of transient contraction/expansion without requiring explicit Jacobian construction in the profiling stage. Two compact stability/contractivity scores computed from the smoothed profile, $S_{\min}$ and S_{mom}, provide actionable criteria for ranking (α, β) configurations and for mapping stable corridors in the parameter plane. A key practical advantage is that the optimization target is directly aligned with finite-time contractivity rather than relying on qualitative dynamical visualizations.

The numerical experiments (Tables 4–8 and Figure 8) confirm that parameter sets selected by the kNN–LLE contractivity proxy consistently improve convergence behavior relative to generic or manually tuned choices: lower error/residual norms, reduced iteration counts and CPU time, and demonstrated robustness across the tested initializations. The heatmap sweeps (Figures 5–7) further show that the scores organize into structured regions (stable corridors) that are difficult to localize analytically and that become more distinct at higher grid resolution.

6.1. Implications for Digital Health Assistants

The proposed contractivity-profiling framework is directly compatible with repeated, real-time nonlinear estimation cycles in digital health and occupational monitoring. When a personalized model must be solved frequently under drifting parameters, the stability proxy and heatmap-based screening provide a reproducible way to select solver parameters that reduce the risk of transient divergence and improve time-to-solution on multicore devices. As a future direction, this enables integrating contractivity-aware root solving into decision-support pipelines that translate estimated regimes into actionable recommendations (e.g., workload adjustment, recovery prompts, or safety alerts), subject to application-specific validation.

6.2. Limitations

Despite its strengths, the proposed tuning framework has several limitations:

- The profiling stage adds computational overhead due to ensemble generation and repeated kNN evaluations, although this cost is amortized when parameters are reused across many solves or when stable regions are screened once and then exploited.
- The current implementation uses a fixed profiling configuration (LOOK_BACK, horizon set, k in kNN, train/test split, and smoothing width) across the full parameter sweep. While this improves comparability across (α, β) and yields stable/structured score landscapes in the reported experiments, a dedicated systematic sensitivity analysis of ranking stability with respect to these hyperparameters remains an important direction for future work.
- The proposed kNN–LLE quantity is an observable-based finite-time contractivity proxy and does not, by itself, constitute a strict classical FTLE of the full coupled iteration map or a closed-form global convergence guarantee for all parameter choices and all initial conditions.

6.3. Future Directions

Future directions include the following:

- Adaptive or sequential tuning strategies (coarse-to-fine sweeps, surrogate modeling, or online updates) to reduce sweep cost while preserving reproducibility;
- Systematic hyperparameter selection (e.g., automated calibration of micro-series length, horizons, and smoothing) and uncertainty-aware scoring of parameter regions;
- A systematic tuning study of competing parameterized schemes MPS_1–MPS_2 remains an interesting direction for future work;
- Jacobian-based validation of the proxy is nontrivial for the coupled high-dimensional parallel iteration map; therefore, a systematic Jacobian/linearization-level contraction study is left as a dedicated direction for future work;
- Extension to broader classes of simultaneous root-finding schemes (including higher-order and fractional-order variants) and to large-scale nonlinear systems arising from discretized PDE models.

Overall, the results support the central message: finite-time contractivity profiling via the kNN–LLE largest-Lyapunov-exponent proxy provides a practical, scalable, and reproducible route to systematic parameter optimization in sensitive high-order parallel root-finding dynamics, with the MVB_* scheme used here as a representative and demanding demonstration case.

Author Contributions: Conceptualization, M.S., A.V., and B.C.; methodology, M.S., A.V., and B.C.; software, M.S. and A.V.; validation, M.S., A.V. and B.C.; formal analysis, B.C.; investigation, M.S., A.V., and B.C.; resources, B.C., and A.V.; data curation, M.S. and A.V.; writing—original draft preparation, M.S., A.V., and B.C.; writing—review and editing, A.V. and B.C.; visualization, M.S. and A.V.; supervision, B.C., and A.V.; project administration, B.C.; funding acquisition, B.C. and A.V. All authors have read and agreed to the published version of the manuscript.

Funding: Sections 2–4, 6, and Appendix A were financially supported by the Ministry of Education and Science of Russia within project FSNW-2026-0003 of the state research assignment no. 075-03-2026-472 (Andrei Velichko). Sections 1 and 5 were supported by European Regional Development and Cohesion Funds (ERDF) 2021–2027 under Project AI4AM–EFRE1052 (Bruno Carpentieri). The APC was funded by European Regional Development and Cohesion Funds (ERDF) 2021–2027 under Project AI4AM–EFRE1052.

Institutional Review Board Statement: All authors declare that this work complies with ethical guidelines set by the Committee on Publication Ethics (COPE).

Data Availability Statement: The data supporting the findings of this study are included within this article.

Acknowledgments: The authors thank J. Alberto Conejero for reading the manuscript and for his constructive comments and suggestions, which improved its clarity and overall quality. The authors also wish to express sincere gratitude to the editor and reviewers for their insightful and constructive feedback on the manuscript.

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

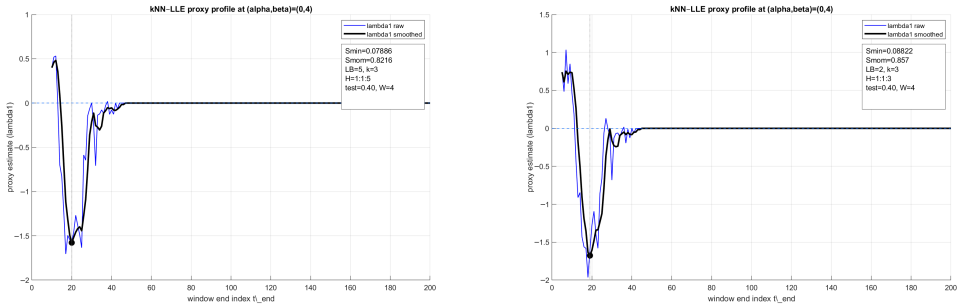
Appendix A. Hyperparameter Robustness: Illustrative Proxy-Profile Comparison

This appendix provides an illustrative sensitivity check of the kNN–LLE proxy profile and the derived ranking scores with respect to moderate changes in profiling hyperparameters, for the representative example $(\alpha, \beta) = (0, 4)$ used in the main text.

Appendix A.1. Two-Profile Illustration (Baseline vs. a Variation)

Figure A1 compares the proxy profiles obtained under the baseline profiling configuration and one representative variation. The early negative excursion (contractive transient)

and the location of the minimum remain in the same early-iteration region, while the derived scores $S_{\min}$ and S_{mom} change only moderately.



(**a**) (**b**)

Figure A1. Illustrative kNN–LLE proxy profiles for $(\alpha, \beta) = (0, 4)$ under two profiling configurations. Blue: raw proxy estimate $\lambda_1^{\mathrm{raw}}(t_{\mathrm{end}})$. Black: trailing-mean smoothed profile $\lambda_1^{\mathrm{smooth}}(t_{\mathrm{end}})$ with $W = 4$. The marked minimum is used in computing $S_{\min}$ and S_{mom}. (**a**) Baseline: `LB` = 5, `H` = 1:1:5, $k = 3$, `test` = 0.40. $t_{\min} = 20$, $y_{\min} = -1.5772$, $S_{\min} = 0.0789$, $S_{\mathrm{mom}} = 0.8216$. (**b**) Variation: `LB` = 2, `H` = 1:1:3, $k = 3$, `test` = 0.40. $t_{\min} = 19$, $y_{\min} = -1.6761$, $S_{\min} = 0.0882$, $S_{\mathrm{mom}} = 0.8570$.

Appendix A.2. Summary Across the Tested Configurations

Table A1 summarizes the derived scores for all tested configurations in this single-point illustration. Across all settings, the profile exhibits an early minimum (typically in the range $t_{\min} \approx 18$–22), and the score values remain in a comparable range, indicating that the observable contractive transient used by $S_{\min}$ and S_{mom} is not an artifact of a single hyperparameter choice for this representative case.

Table A1. Summary of proxy-profile-derived scores for $(\alpha, \beta) = (0, 4)$ under multiple profiling configurations.

Cfg	LB	HMAX	k	Test	$t_{\min}$	S_{mom}
BASE	5	5	3	0.40	20	0.8216
LB2	2	5	3	0.40	20	0.8368
H3	5	3	3	0.40	18	0.8446
LB2_H3	2	3	3	0.40	19	0.8570
K5	5	5	5	0.40	20	0.8086
LB2_K5	2	5	5	0.40	20	0.8198
H3_K5	5	3	5	0.40	18	0.8733
TS30	5	5	3	0.30	22	0.8262
LB2_TS30	2	5	3	0.30	22	0.8648

Appendix A.3. Ranking Robustness on a 30×30 *Parameter Grid (Heatmap-Level)*

To assess robustness at the ranking level, we repeated the (α, β) sweep on a 30×30 grid under the baseline profiling configuration and several moderate variations of the profiling hyperparameters. We quantify stability by (i) the Spearman rank correlation between the resulting score landscapes and (ii) the overlap of the top-10 parameter pairs ranked by the score.

Appendix A.3.1. Visualization (Smom Landscapes)

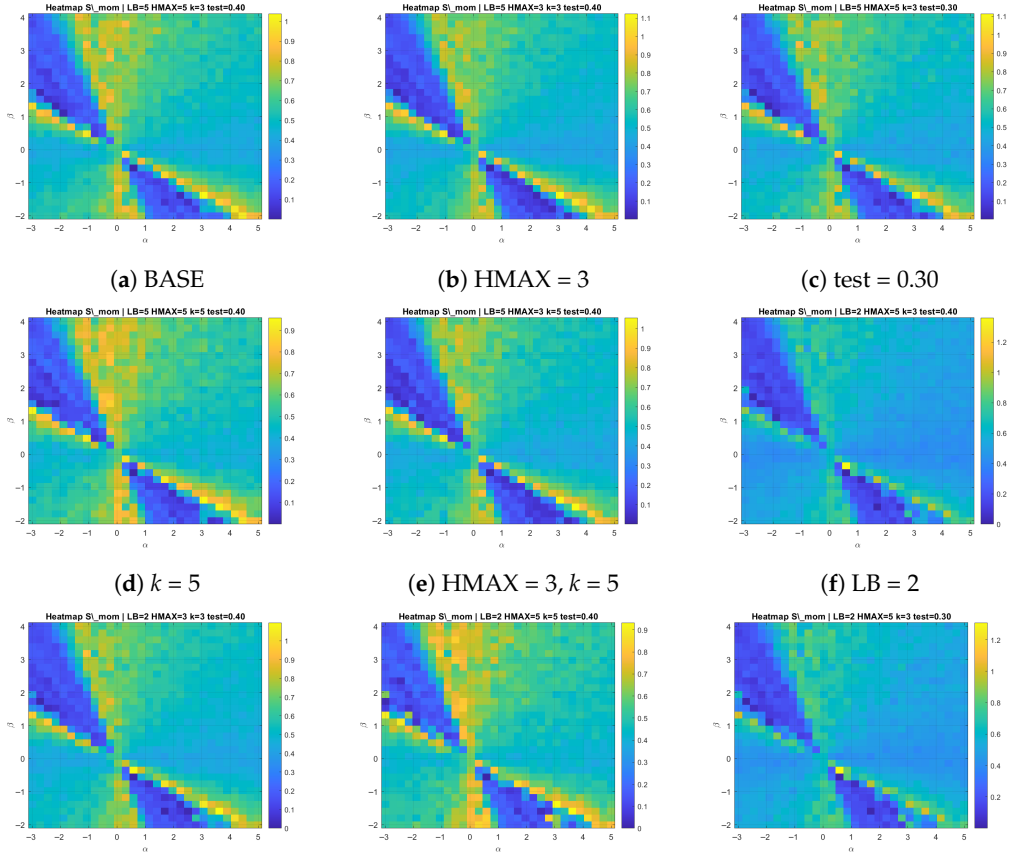
Figure A2 shows the S_{mom} landscapes for the baseline and eight representative variations (one configuration is intentionally omitted for compactness). The high-score corridor remains clearly visible across all shown settings.

Appendix A.3.2. Quantitative Stability Metrics

Table A2 reports Spearman rank correlations and top-10 overlap with the baseline.

Table A2. Ranking robustness under moderate profiling-hyperparameter changes on a 30 × 30 grid (baseline: LB = 5, HMAX = 5, k = 3, test = 0.40).

Variation	$\rho_S(S_{\mathrm{mom}})$	Top-10 Overlap	$\rho_S(S_{\mathrm{min}})$	Top-10 Overlap
BASE	1.00	10/10	1.00	10/10
H3	0.98	8/10	0.92	9/10
H3K5	0.97	7/10	0.89	8/10
K5	0.97	8/10	0.92	8/10
LB2	0.96	6/10	0.89	2/10
LB2H3	0.97	6/10	0.88	6/10
LB2K5	0.93	5/10	0.86	2/10
TS30	0.98	7/10	0.92	8/10
LB2TS30	0.94	6/10	0.84	2/10



(**a**) BASE (**b**) HMAX = 3 (**c**) test = 0.30

(**d**) k = 5 (**e**) HMAX = 3, k = 5 (**f**) LB = 2

(**g**) LB = 2, HMAX = 3 (**h**) LB = 2, k = 5 (**i**) LB = 2, test = 0.30

Figure A2. Heatmap-level robustness illustration on a 30 × 30 grid (only S_{mom} shown for compactness). All panels use the same solver setup and parameter grid; only the profiling hyperparameters are varied, as indicated. Here, BASE denotes the baseline profiling configuration (LB = 5, HMAX = 5, k = 3, test = 0.40).

Appendix A.3.3. Discussion

Overall, the stability results are positive. The S_{mom} landscape remains highly consistent across all tested variations (Spearman $\rho_S \approx 0.93$–0.98), and the Top-10 overlap stays moderate to high (typically 5–8 common parameter pairs), indicating that the high-score corridor is not an artifact of a single profiling configuration.

Among the tested hyperparameters, reducing the horizon range (HMAX: $5 \rightarrow 3$) and changing the train/test split (test: $0.40 \rightarrow 0.30$) have the smallest impact on the ranking, while decreasing the look-back length (LB: $5 \rightarrow 2$) has the largest effect, especially on $S_{\min}$. This behavior is expected: $S_{\min}$ depends strongly on the exact location and depth of the profile minimum, so small shifts in $t_{\min}$ can reorder close candidates, whereas S_{mom} aggregates the negative "mass" of the profile and is therefore more robust to minor local shifts.

These findings support the use of the proposed profiling pipeline for practical parameter screening: the identified high-score corridor is stable under moderate changes of the key profiling hyperparameters, and the overall ranking structure is primarily governed by intrinsic contractivity properties of the underlying solver dynamics rather than by tuning artifacts of the profiling configuration.

References

1. Gao, Y.; Yang, X.M. Properties of the nonlinear scalar functional and its applications to vector optimization problems. *J. Glob. Optim.* **2019**, *73*, 869–889. [CrossRef]
2. Agrawal, A.; Verschueren, R.; Diamond, S.; Boyd, S. A rewriting system for convex optimization problems. *J. Control Decis.* **2018**, *5*, 42–60. [CrossRef]
3. Gautier, A.; Tudisco, F.; Hein, M. Nonlinear Perron–Frobenius Theorems for Nonnegative Tensors. *SIAM Rev.* **2023**, *65*, 495–536. [CrossRef]
4. Gamal, M.; Zaky, M.A.; El-Kady, M.; Abdelhakem, M. Chebyshev polynomial derivative-based spectral tau approach for solving high-order differential equations. *Comput. Appl. Math.* **2024**, *43*, 412. [CrossRef]
5. Varona, J.L. Graphic and numerical comparison between iterative methods. *Math. Intell.* **2002**, *24*, 37–47. [CrossRef]
6. Jiang, Y.; Liu, H.; Ni, T.; Zhang, K. Inverse problems for nonlinear progressive waves. *arXiv* **2023**, arXiv:2308.07808. [CrossRef]
7. Stetter, H.J. *Analysis of Discretization Methods for Ordinary Differential Equations*; Springer: Berlin/Heidelberg, Germany, 1973. [CrossRef]
8. Bhat, I.A.; Mishra, L.N.; Mishra, V.N.; Tunç, C. Analysis of efficient discretization technique for nonlinear integral equations of Hammerstein type. *Int. J. Numer. Methods Heat Fluid Flow* **2024**, *34*, 4257–4280. [CrossRef]
9. Ortega, J.M.; Rheinboldt, W.C. *Iterative Solution of Nonlinear Equations in Several Variables*; Academic Press: New York, NY, USA, 1970. [CrossRef]
10. Noor, K.I.; Noor, M.A. Predictor–corrector Halley method for nonlinear equations. *Appl. Math. Comput.* **2007**, *188*, 1587–1591. [CrossRef]
11. Gutiérrez, J.M.; Galilea, V. Two dynamic remarks on the Chebyshev–Halley family of iterative methods for solving nonlinear equations. *Axioms* **2023**, *12*, 1114. [CrossRef]
12. Weierstraß, K. Neuer Beweis des Satzes, dass jede ganze rationale Funktion einer Veränderlichen dargestellt werden kann als ein Produkt aus linearen Funktionen derselben Veränderlichen. *Ges. Werke* **1903**, *3*, 251–269. [CrossRef]
13. Atanassova, L.; Kyurkchiev, N.; Yamamoto, T. *Methods for Computing All Roots of a Polynomial Simultaneously—Known Results and Open Problems*; Computing Supplementa; Springer: Vienna, Austria, 2002; Volume 16, pp. 23–42. [CrossRef]
14. Brari, Z.; Belghith, S. A new algorithm for largest Lyapunov exponent determination for noisy chaotic signals with application to EEG analysis. *Chaos Solitons Fractals* **2022**, *165*, 112757. [CrossRef]
15. Magreñán, Á.A. A new tool to study real dynamics: The convergence plane. *Appl. Math. Comput.* **2014**, *248*, 215–224. [CrossRef]
16. Kyurkchiev, N. *Initial Approximations and Root Finding Methods*; Wiley–VCH: Berlin, Germany, 1998. [CrossRef]
17. Chicharro, F.I.; Cordero, A.; Torregrosa, J.R. Drawing dynamical and parameter planes of iterative families and methods. *Sci. World J.* **2013**, *2013*, 780153. [CrossRef] [PubMed]
18. Rockwood, M.P.; Loiselle, T.; Green, M.A. Practical concerns of implementing finite-time Lyapunov exponent analysis with under-resolved data. *Exp. Fluids* **2019**, *60*, 74. [CrossRef]
19. Clauss, P.; Loechner, V. Parametric analysis of polyhedral iteration spaces. *J. VLSI Signal Process. Syst. Signal Image Video Technol.* **1998**, *19*, 179–194. [CrossRef]
20. Shams, M.; Carpentieri, B. Chaos in inverse parallel schemes for solving nonlinear engineering models. *Mathematics* **2024**, *13*, 67. [CrossRef]
21. Varga, R.S. *Iterative Analysis*; Springer: New York, NY, USA, 1962. [CrossRef]
22. Velichko, A.; Belyaev, M.; Boriskov, P. A novel approach for estimating largest Lyapunov exponents in one-dimensional chaotic time series using machine learning. *Chaos* **2025**, *35*, 101101. [CrossRef]

23. Ma, Z.S.; Li, X.; He, M.X.; Jia, S.; Yin, Q.; Ding, Q. Recent advances in data-driven dynamics and control. *Int. J. Dyn. Control* **2020**, *8*, 1200–1221. [CrossRef]
24. Shams, M. Theoretical advances in two-step iterative schemes for nonlinear problems. *Trans. Comput. Model. Intell. Syst.* **2025**, *1*, 10011. [CrossRef]
25. Korzun, D.; Meigal, A. Multi-source data sensing in mobile personalized healthcare systems: Semantic linking and data mining. In *Proceedings of the 24th Conference of Open Innovations Association (FRUCT)*; IEEE: New York, NY, USA, 2019; pp. 187–192. [CrossRef]
26. Rosenblatt, M.; Timmer, J.; Kaschek, D. Customized steady-state constraints for parameter estimation in non-linear ordinary differential equation models. *Front. Cell Dev. Biol.* **2016**, *4*, 41. [CrossRef]
27. Kariya, Y.; Honma, M.; Tokuda, K.; Konagaya, A.; Suzuki, H. Utility of constraints reflecting system stability on analyses for biological models. *PLoS Comput. Biol.* **2022**, *18*, e1010441. [CrossRef]
28. Peñas, D.; González, P.; Egea, J.; Doallo, R.; Banga, J. Parameter estimation in large-scale systems biology models: A parallel and self-adaptive cooperative strategy. *BMC Bioinform.* **2017**, *18*, 52. [CrossRef]
29. Beltrão, G.; Martins, W.; Alaee-Kerahroodi, M.; Schroeder, U.; Tatarinov, D. Adaptive nonlinear least squares framework for contactless vital sign monitoring. *IEEE Trans. Microw. Theory Tech.* **2023**, *71*, 1696–1710. [CrossRef]
30. Shams, M.; Velichko, A.; Carpentieri, B. Direct finite-time contraction (step-log) profiling–driven optimization of parallel schemes for nonlinear problems on multicore architectures. *arXiv* **2026**, arXiv:2601.13637. [CrossRef]
31. Rosen, M.I. Niels Hendrik Abel and equations of the fifth degree. *Am. Math. Mon.* **1995**, *102*, 495–505. [CrossRef]
32. Papp, D. Semi-infinite programming using high-degree polynomial interpolants and semidefinite programming. *SIAM J. Optim.* **2017**, *27*, 1858–1879. [CrossRef]
33. Reinke, B.; Schleicher, D.; Stoll, M. The Weierstrass–Durand–Kerner root finder is not generally convergent. *Math. Comput.* **2023**, *92*, 839–866. [CrossRef]
34. Anourein, A.W.M. An improvement on two iteration methods for simultaneous determination of the zeros of a polynomial. *Int. J. Comput. Math.* **1977**, *6*, 241–252. [CrossRef]
35. Zhang, X.; Peng, H.; Hu, G. A high order iteration formula for the simultaneous inclusion of polynomial zeros. *Appl. Math. Comput.* **2006**, *179*, 545–552. [CrossRef]
36. Cordero, A.; Torregrosa, J.R.; Triguero-Navarro, P. Jacobian-free vectorial iterative scheme to find simple several solutions simultaneously. *Math. Methods Appl. Sci.* **2025**, *48*, 5718–5730. [CrossRef]
37. Zeng, Z. Computing multiple roots of inexact polynomials. *Math. Comput.* **2005**, *74*, 869–903. [CrossRef]
38. Nedzhibov, G. Dynamic mode decomposition via polynomial root-finding methods. *Mathematics* **2025**, *13*, 709. [CrossRef]
39. Proinov, P.D.; Petkova, M.D. A new semilocal convergence theorem for the Weierstrass method for finding zeros of a polynomial simultaneously. *J. Complex.* **2014**, *30*, 366–380. [CrossRef]
40. Shams, M.; Kausar, N.; Samaniego, C.; Agarwal, P.; Ahmed, S.F.; Momani, S. On efficient fractional Caputo-type simultaneous scheme for finding all roots of polynomial equations with biomedical engineering applications. *Fractals* **2023**, *31*, 2340075. [CrossRef]
41. Shams, M. On a stable multiplicative calculus-based hybrid parallel scheme for nonlinear equations. *Mathematics* **2024**, *12*, 3501. [CrossRef]
42. Machado, R.N.; Lopes, L.G. A family of Ehrlich-type accelerated methods with King's correction for the simultaneous approximation of polynomial complex zeros. *Glob. J. Pure Appl. Math.* **2019**, *15*, 789–802.
43. Petković, M.S.; Petković, L.D.; Džunić, J. On an efficient method for the simultaneous approximation of polynomial multiple roots. *Appl. Anal. Discrete Math.* **2014**, *8*, 73–94. [CrossRef]
44. Egbo, M.K. A fundamental review on composite materials and some of their applications in biomedical engineering. *J. King Saud Univ.-Eng. Sci.* **2021**, *33*, 557–568. [CrossRef]