Annealing approach to root finding

Junghyo Jo [1,2,3,*], Alexandre Wagemakers [4,†], and Vipul Periwal [5,‡]

[1]*Department of Physics Education, Seoul National University, Seoul 08826, Korea*
[2]*Center for Theoretical Physics and Artificial Intelligence Institute, Seoul National University, Seoul 08826, Korea*
[3]*School of Computational Sciences, Korea Institute for Advanced Study, Seoul 02455, Korea*
[4]*Nonlinear Dynamics, Chaos and Complex Systems Group, Departamento de Física, Universidad Rey Juan Carlos, Tulipán s/n, Móstoles 28933, Madrid, Spain*
[5]*Laboratory of Biological Modeling, National Institute of Diabetes and Digestive and Kidney Diseases, National Institutes of Health, Bethesda, Maryland 20892, USA*

The Newton-Raphson method is a fundamental root-finding technique with numerous applications in physics. In this study, we propose a parameterized variant of the Newton-Raphson method, inspired by principles from physics. Through analytical and empirical validation, we demonstrate that this approach offers increased robustness and faster convergence during root-finding iterations. Furthermore, we establish connections to the Adomian series method and provide a natural interpretation within a series framework. Remarkably, the introduced parameter, akin to a temperature variable, enables an annealing approach. This advancement sets the stage for a fresh exploration of numerical iterative root-finding methodologies.

I. INTRODUCTION

Root-finding algorithms are important to solve equations, a fundamental task in quantitative theoretical science. Root-finding and fixed-point iterations are intricately connected, serving as essential tools in various fields such as optimization and algorithm development. In physics, root-finding methods can be smartly employed to find periodic orbits [1,2] and estimate parameters [3] in nonlinear systems with chaotic dynamics. Additionally, they can be used to identify stationary states of potential energy functions in classical systems [4,5], in a generic Turing reaction-diffusion system [6], and in a nonlinear Schrödinger lattice [7], as well as to find complex saddle points in quantum many-body systems [8]. This application can be naturally extended to the maximization of log-likelihood [9] and the minimization of free energy [10] of physical systems. They are also applicable in solving differential equations of fluid dynamics [11] and self-consistent equations in quantum heat transport problems [12]. Recently, the rise of deep learning has introduced complex, high-dimensional root-finding challenges, particularly in optimizing architecturally intricate neural networks, driving the need for further advancements in root-finding techniques.

Here, we introduce a concrete example how the root-finding method can be applied to solve a physics problem. The Kuramoto model [13] is paradigmatic in the theoretical study of the spontaneous synchronization of physical constituents. Consider a system of N physical rotors with configurations θ_i,

interacting as described by

$$\frac{d\theta_i}{dt} = \kappa \sum_{j \neq i} \Gamma_{ij} \sin(\theta_j - \theta_i + \Psi_{ij}), \qquad (1)$$

where κ is an overall coupling constant, Γ_{ij} is a real weight matrix describing the edges of a weighted asymmetric directed graph with no multiedges, and Ψ_{ij} is a matrix of phase delays associated with the edge $j \to i$. Note that we consider the case where no rotor has intrinsic angular velocity. By including slow learning dynamics for the Γ and Ψ matrices one could in principle store multiple patterns in the dynamics of this system, which may be relevant for modeling neuronal systems [14,15]. Now, suppose we consider the synchronization state of the rotors in a common angular velocity ω, such that

$$\theta_i(t) = \omega t - \phi_i, \qquad (2)$$

where ϕ_i is a rotor-specific phase shift. Given κ, Γ and Ψ, is there a vector of phase shifts ϕ such that synchronization can occur for $\omega \neq 0$? To exhibit this as a problem of finding roots for a set of $N - 1$ coupled equations, inserting Eq. (2) into Eq. (1) gives

$$\omega = \kappa \sum_{j \neq i} \Gamma_{ij} \sin(\phi_i - \phi_j + \Psi_{ij}) \equiv \kappa f_i(\phi), \quad \forall i. \quad (3)$$

We can set the initial phase shift $\phi_0 = 0$ for the rotor with $i = 0$. Then $\omega(\phi)$ is specified by

$$\omega = \kappa \sum_{j \neq 0} \Gamma_{0j} \sin(-\phi_j + \Psi_{0j}), \qquad (4)$$

---
*Contact author: jojunghyo@snu.ac.kr
†Contact author: alexandre.wagemakers@urjc.es
‡Contact author: vipulp@mail.nih.gov

and inserting this value of ω in the remaining $N-1$ equations gives for all $N > i > 0$:

$$\sum_{j \neq 0} \Gamma_{0j} \sin(-\phi_j + \Psi_{0j}) = f_i(\phi). \tag{5}$$

Now κ does not appear in Eq. (5) which is not surprising because changing κ merely changes the scale of the time variable in this system. When the rotors have intrinsic individual frequencies, κ cannot be scaled out and the Kuramoto synchronization phase transition occurs only for strong enough coupling.

To demonstrate that this problem is an exercise in root finding, consider the case of two rotors. Then there is one equation determining ϕ_1:

$$\Gamma_{01} \sin(\phi_1 - \Psi_{01}) + \Gamma_{10} \sin(\phi_1 + \Psi_{10}) = 0. \tag{6}$$

It is evident that for appropriate choices of coupling weights Γ_{01}, Γ_{10} and edge-specific phase shifts Ψ_{01}, Ψ_{10}, this equation has a real root. In particular, the root then determines the synchronization frequency as shown in Eq. (4).

Given the numerous applications, root-finding methods historically predate even calculus, and continue to evolve, with well-known methods like bisection, regula falsi, Newton-Raphson, and secant methods [16]. Notably, the Newton-Raphson method stands out as a powerful iterative approach for root finding [17]. The iterative update of x_n,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \tag{7}$$

ultimately converges to the root, x_{root}, satisfying $f(x_{\text{root}}) = 0$. Note that $f'(x_n)$ represents the slope of $f(x)$ at $x = x_n$.

Research efforts persist in exploring hybrid methods, parallelized techniques, and symbolic and numerical hybrids, all aimed at enhancing efficiency, convergence, robustness, and global optimization. Physical analogies have long been important in numerical analysis. Can root finding also be phrased in a physically motivated setting? In this study, we develop an extended Newton-Raphson method, inspired by principles from physics. The method can be summarized in two iterative steps:

$$\hat{x}_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)},$$
$$x_{n+1} = \hat{x}_{n+1} - \beta \frac{f(\hat{x}_{n+1})}{f'(x_n)}. \tag{8}$$

Here, we introduce an auxiliary parameter β, where $\beta = 0$ corresponds to the original Newton-Raphson method. Although we present the method for a scalar variable, it can be readily extended to encompass vector functions with multiple variables. For the specific application to the synchronization problem discussed above, we define

$$A_{ik}^{-1}(\phi) \equiv \frac{\partial}{\partial \phi_k} f_i(\phi). \tag{9}$$

The iteration to find ϕ_i involves two steps:

$$\hat{\phi}_k^{(n+1)} = \phi_k^{(n)} - \sum_j A_{kj}(\phi^{(n)}) f_j(\phi^{(n)}), \tag{10}$$

followed by

$$\phi_k^{(n+1)} = \hat{\phi}_k^{(n+1)} - \beta \sum_j A_{kj}(\phi^{(n)}) f_j(\hat{\phi}^{(n+1)}). \tag{11}$$

In fact, in this system there could well be multiple independent synchronization frequencies for subsets of rotors, depending on the structure of Γ and Ψ.

This paper is organized as follows: In Sec. II, we provide a physical perspective on the Newton-Raphson method, highlighting its limitations and proposing solutions. Section III delves into the development of a parameterized variant of the Newton-Raphson method, elucidating its iterative implementation of Eq. (8) and demonstrating superior convergence and robustness through practical examples. Section IV explores the connection between the extended Newton-Raphson method and the Adomian decomposition method, interpreting the auxiliary parameter β as a natural expansion parameter for the Adomian series. Finally, adopting the annealing concept in physics [18], we show further computational improvement by adjusting the β value during iterations.

II. PHYSICAL APPROACH TO FINDING ROOTS

We now rederive the Newton-Raphson method using an alternative perspective from physics, naturally extending it into an enhanced version.

A. Alternative perspective of the Newton-Raphson method

Suppose that we have a scalar function $f(x)$ with a single variable x, aiming to determine a root, x_{root}, such that $f(x_{\text{root}}) = 0$. Then, we consider an integral,

$$Z \equiv \int dx \exp\left[-\frac{1}{2g^2} f(x)^2\right], \tag{12}$$

where g is a real-valued scale parameter. We provide the mathematical justification for this form based on cohomological quantum field theory in Appendix A. For any function h of x, we expect that in the limit $g \downarrow 0$,

$$\langle h \rangle \equiv \frac{1}{Z} \int dx \, h(x) \exp\left[-\frac{1}{2g^2} f(x)^2\right] \to h(x_{\text{root}}), \tag{13}$$

where x_{root} is a minimum or a root of f, assuming a unique root or a unique minimum. In particular, the expectation value of $h(x) = x$ itself should be an estimate of x_{root}.

The systematic approach to analyzing Eq. (13) in the limit $g \downarrow 0$ is by using Laplace's approximation for the integral. In brief, this approximation requires us to find x^* such that $f(x^*)f'(x^*) = 0$, and then do a Taylor expansion about x^*, evaluating the integral as an integral over fluctuations about x^*.

For the purposes of finding roots of f, this is pointless as we have no idea where the root might be. Attempting to localize the integral at an arbitrary specific value x_0 using a Taylor expansion leads to

$$Z = \int dx \exp\left[-\frac{1}{2g^2}(f(x_0) + f'(x_0)(x - x_0) + \ldots)^2\right]. \tag{14}$$

Defining a fluctuation variable $\delta \equiv x - x_0$ with $f_0 \equiv f(x_0)$ and $f_0' \equiv f'(x_0)$,

$$Z = \int d\delta \exp\left[-\frac{1}{2g^2}(f_0 + f_0'\delta + \ldots)^2\right]. \quad (15)$$

Then, we can find a better stationary point by varying

$$\frac{1}{2g^2}\left[f_0 f_0'\delta + \frac{1}{2}(f_0'\delta)^2\right], \quad (16)$$

which is stationary when

$$\delta = -f_0/f_0'. \quad (17)$$

Since (up to higher order derivatives) this defines the expectation value of δ, it follows that the estimate for the root of f given by this is

$$x_{\text{root}} \approx \langle x \rangle = x_0 - f_0/f_0', \quad (18)$$

which is the Newton-Raphson update.

As the basic assumption of the Laplace approximation is that the initial starting point is a stationary point of f^2 so that the expectation value of fluctuations vanishes, this attempt to expand the integral about an arbitrary value x_0 tells us that for a consistent Laplace approximation, we need to replace $x_0 \to x_0 - f_0/f_0'$, and start over.

B. Inconsistency in the Newton-Raphson method

Laplace's approximation holds true only when we expand around a stationary point. As noted, the presence of a nonzero right-hand side in Eq. (17) indicates that this condition does not hold for arbitrary x_0. However, there is another problem with our approach. We assumed that the linear approximation was actually solving the equation of interest, namely $f(x_{\text{root}}) = 0$, but all we really solved was

$$f_L(x) \equiv f_0 + f_0'(x - x_0) = 0. \quad (19)$$

This becomes apparent when we substitute the value $x_0 \to x_0 + \delta$ into Eq. (15), as the term $f_0^2/2g^2$ also undergoes modification, although it did not contribute to the variational equation, Eq. (16). Only if the corrections to Eq. (17) are of order g and higher with $\delta = -f_0/f_0' = O(g)$ and $f_0'' \equiv f''(x_0)$,

$$f(x_0 + \delta) \approx f_0 + f_0'\delta + \frac{1}{2}f_0''\delta^2 = O(g^2), \quad (20)$$

which is higher order in g. This result implies that the Laplace approximation is naïvely self-consistent. However, it is important to note that $f_0/f_0' = O(g) \Leftrightarrow f_0/g = O(f_0')$. For an arbitrarily chosen point x_0, there is no assurance of such a relationship between f_0 and f_0', as g must be as small as possible for the Laplace approximation to be valid. Hence, we deduce that the genuine expansion parameter is f_0/g, and Eq. (17) must be amended.

One approach to addressing both problems at the same time is to seek an enhanced integrand, often referred to as an effective action in analogous contexts within physics. Typically, such an effective integrand is computed by taking short distance scale fluctuations into account, leading to an integrand governing longer distance fluctuations but necessarily dependent on g. For instance, Lepage and Mackenzie [19] demonstrated the reordering of strong-coupling terms

to derive an improved effective action in lattice gauge theory. However, in our scenario, where only a single integral is involved, such a strategy cannot be directly applied. Nonetheless, we can explore the possibility of identifying a conceptually similar effective equation to refine the Laplace approximation, particularly in cases where the Taylor expansion employed to derive Eq. (17) becomes invalid.

f_0/f_0' has the dimensions of length so fluctuations about x_0 are naturally considered big or small relative to this length scale. Two clues toward finding such an effective variant of Eq. (17) are the following:

(i) When an exact stationary point for the Laplace approximation is not used, the leading correction to the logarithm of the integral is singular in a power of g.

(ii) As mentioned above, the term $f(x_0)^2/2g^2$, which did not contribute to Eq. (17), becomes influential in determining the next approximate stationary point of the Laplace approximation.

The consequences of f_0/f_0' not being small can be estimated readily. Consider the next term in the Taylor expansion,

$$f(x_0 + \delta) = f_0 + f_0'\delta + \frac{1}{2}\frac{f_0''}{f_0'^2}(f_0'\delta)^2. \quad (21)$$

Using this equation, we vary $f(x_0 + \delta)^2/2g^2$ with respect to the dimensionless variable $\Delta \equiv f_0'\delta/g$. Note the power of g in this definition, included so that the condition for validity of the Laplace expansion becomes $\Delta \ll 1$. The variation gives

$$\left(\frac{1}{g}f_0 + \Delta + \frac{1}{2}\frac{f_0''}{f_0'^2}g\Delta^2\right)\left(1 + \frac{f_A''}{f_0'^2}g\Delta\right). \quad (22)$$

Ignoring the second derivative, this implies $\Delta = -f_0/g$. This is the standard Newton-Raphson update. For the validity of the Laplace approximation, $\Delta \ll 1$, so we see again that f_0/g is the effective expansion parameter. However, if we include the second derivative terms and estimate Δ from the resulting quadratic equation, then we find that

$$\Delta \approx -\frac{1}{g}f_0 - \frac{1}{2g}f_0^2\frac{f_0''}{f_0'^2}, \quad (23)$$

which implies that including nonlinearities, with higher derivative terms in the function f for example, cannot be accommodated in the Laplace approximation for arbitrarily small f_0/g in a consistent manner.

C. Correcting the Newton-Raphson method

Given the inconsistency in the Newton-Raphson method, an improved effective version of Eq. (23) is needed that gives a consistent order by order expansion in f_0/g, but still gives exactly the same formal stationary point equation for Δ. We begin with a clue by observing

$$\frac{1}{g^2}f(x_0 - f_0/f_0') \approx \frac{1}{2}\frac{f_0^2}{g^2}\frac{f_0''}{f_0'^2} \propto \left(\frac{f_0}{g}\right)^2. \quad (24)$$

This motivates us to formulate an effective stationary point equation,

$$\Delta = -\frac{1}{g}f_0 - \frac{1}{g^2}f(x_0 + g\Delta/f_0')$$

$$\approx -\frac{1}{g}f_0 - \frac{1}{g^2}f(x_0 - f_0/f_0') + \dots. \quad (25)$$

The presence of the second nonlinear term with the lower power of g now becomes influential in determining the consistency of the Laplace approximation, a role absent in the contribution of the term $f(x_0)^2/2g^2$ in Eq. (17). Another insight into the lower power of g is that as g increases, f_0/g diminishes, aligning with the strong coupling limit in physics parlance [20], consequently reducing the impact of the additional term in Eq. (25). This term which becomes more important as $g \downarrow 0$ is enforcing the initial ultralocal exact equation: $f(x) = 0$. The original Newton-Raphson linear term is more important in the strong coupling limit and because it incorporates the derivative of f, and not just the value of f, it corresponds to a 'hopping approximation' to the actual equation of motion. In the Newton-Raphson scenario, the Laplace approximation necessitates further corrections beyond the linear expansion of $f(x)^2/2g^2$ in integral evaluation. Hence, the inclusion of the additional term with the lower power aims to balance the update of the stationary point between the weak and strong coupling evaluations of the integral.

To see that the power of g is now exactly correct in a formal expansion, we use the linear Taylor expansion and get

$$\frac{1}{g^2}f(x_0 + g\Delta/f_0') \approx \frac{1}{g^2}(f_0 + g\Delta) \quad (26)$$

and inserting this expression back into Eq. (25), we get

$$\Delta = -\frac{1}{g}\left(1 + \frac{1}{g}\right)f_0 - \frac{1}{g}\Delta \quad \Leftrightarrow \quad \Delta = -\frac{1}{g}f_0. \quad (27)$$

In other words, the effective equation of motion, when g is small, recapitulates Eq. (17). Finally, we accomplish both objectives of (i) having a leading correction singular in a power of g and (ii) ensuring the term $f(x_0)^2/2g^2$ plays a significant role in the Laplace approximation. By substituting $\Delta = f_0'\delta/g$ with $\delta = x - x_0$, the effective stationary point equation of Eq. (25) can be expressed as follows:

$$\frac{1}{g}f(x) + f_0 + f_0'(x - x_0) = 0. \quad (28)$$

This stationary equation can also be understood more intuitively. Consider the modified form of the partition function in Eq. (15) as follows:

$$Z = \int dx \exp\left[-\frac{1}{2g^2}(\alpha f(x) + (1 - \alpha)f_L(x))^2\right], \quad (29)$$

where $f_L(x) \equiv f_0 + f_0'(x - x_0)$ represents the linear Taylor approximation of $f(x)$ at $x = x_0$. This interpolation between $f(x)$ and $f_L(x)$ encapsulates three essential properties:

(i) Setting $\alpha = 1$ recovers the original partition function.

(ii) It closely resembles the original partition function when $f_L(x) \approx f(x)$ is a valid approximation.

(iii) The term $\alpha f(x)$ serves as a "tadpole" that diminishes as x approaches x_{root}.

With a few algebraic manipulations and redefined parameters in Eq. (29), we obtain

$$Z = \int dx \exp\left[-\frac{1}{2\tilde{g}^2}(\beta f(x) + f_L(x))^2\right], \quad (30)$$

where $\tilde{g} \equiv g(1 - \alpha)$ and $\beta = \alpha/(1 - \alpha)$. Subsequently, the stationary equation leads to:

$$\beta f(x) + f_0 + f_0'(x - x_0) = 0. \quad (31)$$

Aside from coefficient differences, this equation mirrors Eq. (28). As we will demonstrate, the coefficient acts as a control parameter, rendering this distinction insignificant.

III. ITERATIVE ROOT FINDING

The revised stationary Eq. (31) serves as an iterative root-finding method. This section elucidates its enhanced convergence, robustness, and interpretation as an iterative algorithm.

A. Iterative algorithm

We rewrite the stationary Eq. (31):

$$\beta f(x(\beta)) + f_0 + f_0'(x(\beta) - x_0) = 0 \quad (32)$$

to find $x(\beta)$, in place of Eq. (17). Once we get to Eq. (32), we can completely ignore the origins of this equation so the numerical coefficient that we have not determined can simply be absorbed into the definition of the abstract parameter β.

Here, we consider intuitively how this approach works. Equation (32) incorporates the complete function f, not just the linear Taylor expansion approximation and therefore makes sense even when $\delta = x(\beta) - x_0$ is not small since, for β large, it dominates the other terms in Eq. (32) to move δ toward x_{root}, as we show geometrically below. If $f(x(\beta))$ and f_0 have the same sign, then increasing β will result in $|x(\beta) - x_0|$ becoming larger, while the opposite sign will lead to a smaller separation between x_0 and $x(\beta)$. This will help the iterative process $[x_0 \to x(\beta)]$ converge faster (Fig. 1). For any value $\beta \neq -1$, if there is a fixed point, x^*, with a nonsingular gradient, we have

$$(\beta + 1)f(x^*) = 0 \Leftrightarrow f(x^*) = 0. \quad (33)$$

The intermediate values of $x(\beta)$ may vary with β, potentially causing shifts in the basins of attraction as β changes (refer to Sec. III E for further elucidation). Nonetheless, the fixed point remains a root of f, unaffected by β, a fact that can also be demonstrated through cohomological quantum field theory (Appendix A).

We now explicitly present the revised iterative method as an alternative to the Newton-Raphson method. Using Eq. (32), let $x(\beta)$ be defined as the solution of

$$x(\beta) = N(x_0) - \frac{\beta f(x(\beta))}{f_0'} \quad (34)$$

with a nonlinear mapping of $N(x) \equiv x - f(x)/f'(x)$. At a fixed point, $f(x^*) = 0$, all β dependence vanishes. As we expect based on our discussion above, the fixed point is a root of f: $x^* = x_{\text{root}}$. At $\beta = 0$, the infinite temperature limit, this is clearly the Newton-Raphson iteration, and does not have
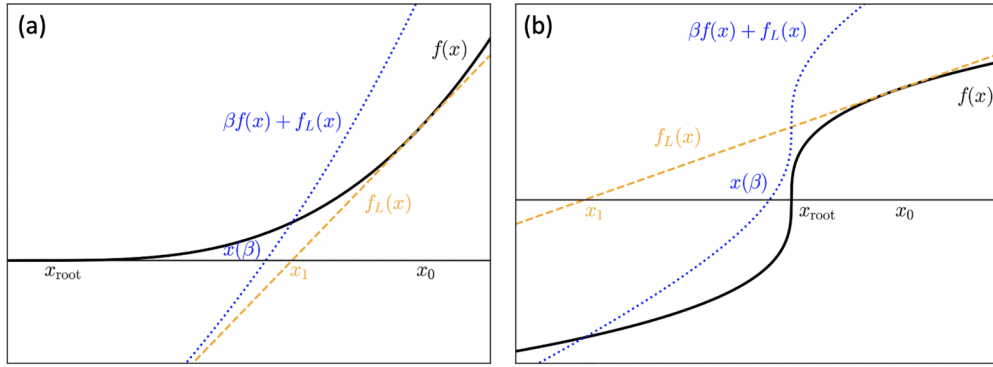
FIG. 1. Improved convergence to roots: Comparison between Newton-Raphson method and the revised method. The Newton-Raphson method progresses from an initial point x_0 to x_1, determined as the solution of $f_L(x_1) = f_0 + f_0'(x_1 - x_0) = 0$, whereas the revised method advances x_0 to $x(\beta)$, found as the solution of $\beta f(x) + f_L(x)$. The separation $|x(\beta) - x_0|$ adjusts, either increasing or decreasing, to enhance convergence toward x_{root}, contingent upon the relationship between $f(x(\beta))$ and f_0: (a) when they share the same sign, or (b) when they have opposite signs.

any $x(\beta)$ on the right hand side. For $\beta > 0$, it could also be solved by inserting this definition of $x(\beta)$ into the right hand side repeatedly. At the first order, we get

$$x(\beta) = N(x_0) - \frac{\beta f(N(x_0))}{f_0'}. \tag{35}$$

A fixed point of this equation does not immediately imply that the fixed point is a root of f, because as it stands this is not the original Eq. (34). A fixed point here implies only that $f(x^*) + \beta f(N(x^*)) = 0$. Now this equation is satisfied when x^* is a root, x_{root}, of f, but the reverse implication is not necessarily true, unless this continues to hold as we vary β continuously.

Let us rewrite the first-order equation as

$$\begin{aligned} x(\beta) &= x_1 - \beta f_1/f_0' \\ &= (1 - \beta)(x_0 - f_0/f_0') + \beta x_1 - \beta f_1/f_0' \\ &= \tilde{x}_0 - \tilde{f}_0/f_0', \end{aligned} \tag{36}$$

where we denote $x_1 \equiv N(x_0) = x_0 - f_0/f_0'$ and $f_1 \equiv f(x_1)$, and define $\tilde{x}_0 \equiv (1 - \beta)x_0 + \beta x_1$ and $\tilde{f}_0 \equiv (1 - \beta)f_0 + \beta f_1$.

Comparing this update with the Newton-Raphson form in Fig. 2, we see that the function pair (x_0, f_0) determining the next estimate is replaced in our approach with the pair $(\tilde{x}_0, \tilde{f}_0)$, but with the same slope, f_0'. In other words, the line going from (x_0, f_0) to $(x_1, 0)$ with slope f_0' is translated to a parallel line going from $(\tilde{x}_0, \tilde{f}_0)$, to $(x(\beta), 0)$ to find the next estimate of the root, $x(\beta)$. Geometrically, this makes the next step in the iteration larger or smaller depending on the relative signs of f_0 and $f(x(\beta))$, as discussed above.

Were we to repeat this process, we would get

$$x(\beta) = N(x_0) - \frac{\beta f\left[N(x_0) - \frac{\beta f(N(x_0))}{f_0'}\right]}{f_0'}, \tag{37}$$

and so on. Each order in β has one more function evaluation and is higher order in β. It turns out that either of these definitions of iterative determination of $x(\beta)$ gives exactly the same fixed point structure, which exhibits a fractal nature. A fixed point value that is a fixed point of both these versions will in fact be a root. Henceforth, we proceed with the simplest
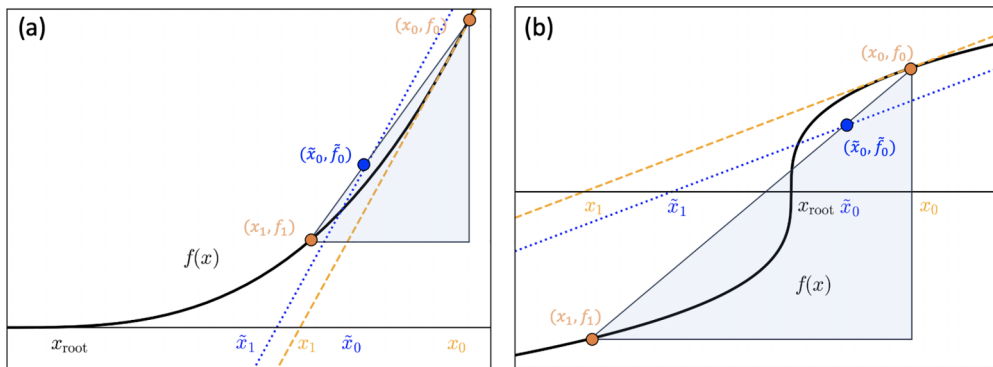

FIG. 2. Improved iteration to roots: Comparison between Newton-Raphson method and the revised method. In the Newton-Raphson method, the progression from an initial point x_0 to $x_1 = x_0 - f_0/f_0'$ with a slope f_0'. The revised method can be interpreted to provide an interpolated point, $(\tilde{x}_0, \tilde{f}_0)$, between (x_0, f_0) and (x_1, f_1), where $f_i \equiv f(x_i)$. Then, the next iterated point $\tilde{x}_1$ is determined by a linear function crossing $(\tilde{x}_0, \tilde{f}_0)$ with the same slope f_0'. The convergence behavior depends on the relationship between f_0 and f_1: (a) when they share the same sign or (b) when they have opposite signs.

choice, which can be described by two iterative steps:

$$\hat{x}_{n+1} = N(x_n) = x_n - \frac{f(x_n)}{f'(x_n)},$$

$$x_{n+1} = \hat{x}_{n+1} - \beta \frac{f(\hat{x}_{n+1})}{f'(x_n)}. \tag{38}$$

This completes our derivation of Eq. (8). It is noteworthy that this modified Newton-Raphson method, with a specific value of $\beta = 1$, has been derived in Refs. [21–23]. The derivation in Ref. [23] based on the Adomian method will be further discussed in the subsequent section.

B. Convergence

We now demonstrate that there is actually a benefit of this β dependent reformulation in terms of root-finding performance. The quadratic convergence of the Newton-Raphson method, near a root, x_{root}, is demonstrated by Taylor expanding

$$f(x) \approx f'_{\text{root}}(x - x_{\text{root}}) + \frac{1}{2} f''_{\text{root}}(x - x_{\text{root}})^2, \tag{39}$$

where $f'_{\text{root}} \equiv f'(x_{\text{root}})$ and $f''_{\text{root}} \equiv f''(x_{\text{root}})$. By inserting this Taylor approximation into the Newton-Raphson update of $x_{n+1} = N(x_n)$, we obtain the the expected quadratic convergence:

$$x_{n+1} - x_{\text{root}} \approx \frac{1}{2} \frac{f''_{\text{root}}}{f'_{\text{root}} + f''_{\text{root}}(x_n - x_{\text{root}})} (x_n - x_{\text{root}})^2. \tag{40}$$

Similarly, for our β dependent update, we get

$$x_{n+1} - x_{\text{root}} \approx \frac{1 - \beta}{2} \frac{f''_{\text{root}}}{f'_{\text{root}}} (x_n - x_{\text{root}})^2, \tag{41}$$

exactly in line with the intuition we gave above. Of course, the number of iterations is never zero so, more precisely, this only shows that the convergence at $\beta = 1$ is faster than quadratic. See Homeier [24,25] for the significance of cubic convergence in modified Newton-Raphson methods. Here we see that for problems where the first derivative at the root is finite, we should use the largest value of β for which a fixed point exists. However, as we shall show explicitly in the case of $f(x) = x^{1/3}$, a fixed point may not exist for $\beta = 1$.

Close to a root, the order of convergence depends on the parameter β, being quadratic for the Newton-Raphson case with $\beta = 0$. Theorem 1 in Ref. [26] proves that the order of convergence is maximized for $\beta = 1$, achieving local order three for a similar iterative algorithm. The cubic convergence for $\beta = 1$ has also been demonstrated in other studies [27,28]. However, the theorem does not specify the convergence order for other values of β. We conducted numerical validation to assess the convergence order across a range of nonlinear functions, and confirmed the quadratic convergence for $\beta \neq 1$ and the cubic convergence only for $\beta = 1$ (refer to Sec. III D in details).

C. Cube-root function

The cube root is a well-known analytically solvable function that is not well-suited for finding roots using the Newton-Raphson algorithm. In fact, for $f(x) = x^{1/3}$, we get

$f(x)/f'(x) = 3x$. It follows that

$$x_{n+1} = N(x_n) = -2x_n, \tag{42}$$

which clearly does not converge. For our β dependent update, Eq. (38), we have $N(x) = -2x$, and so $f(N(x))/f'(x) = (-2x)^{1/3}/(x^{-2/3}/3) = -2^{1/3} \times 3x$. Therefore, Eq. (38) implies

$$x_{n+1} = x_n(-2 + 3 \times 2^{1/3}\beta). \tag{43}$$

For convergence we must have

$$|-2 + 3 \times 2^{1/3}\beta| < 1, \tag{44}$$

which requires

$$\frac{1}{3 \times 2^{1/3}} < \beta < \frac{1}{2^{1/3}}. \tag{45}$$

Moreover, the swiftest convergence occurs at $\beta_{\min} = 2^{2/3}/3$. It is worth noting that $\beta = 0$ (Newton-Raphson method) and $\beta = 1$ (Adomian's method) do not fall within the range of convergent values. These observations regarding the strong-coupling approach to root determination of the cube-root function are empirically validated in one of the examples. Remarkably, by tuning β to this particular value, convergence becomes seemingly independent of the initial point in this instance. This surprising convergence is indicative of a more interesting role for β in the size of the basins of attraction to a given root as we shall show in Sec. III E.

D. Numerical results

We now explicitly compare key numerical figures of the extended Newton-Raphson method. Specifically, the original Newton-Raphson method corresponds to the case of $\beta = 0$ in the parameterized Newton-Raphson method. To conduct the comparison, we tested various types of nonlinear functions, some of which were adopted from Weerakoon and Fernando [29].

Tables I and II show four metrics:

(i) Iteration number: The average number of iterations per initial point after discarding divergent initial points.

(ii) Convergence percentage: The percentage of initial points that successfully converge to a root before reaching a preset maximum number of iterations (set to 50 iterations in this study).

(iii) Computation time: The average execution time per point relative to the Newton-Raphson method. Divergent initial points are excluded from this calculation.

(iv) Convergence order: The estimated order of convergence as iteration progresses.

Given that finding real roots for real-valued function $f(x) = 0$ with $x \in \mathbb{R}$ can be extended to finding zeros for complex-valued functions $f(z) = 0$ with $z \in \mathbb{C}$, we explore the Newton-Raphson methods in the complex plane. For each function, we iterate the method starting from an initial point $z_0 \in \mathbb{C}$, chosen from a regular grid in the square $[-2, 2] \times [-2, 2]$. We select a grid of 1000 by 1000 points evenly spaced across this square and iterate each initial point until the criterion $|z_{n+1} - z_n| < \varepsilon$ is satisfied, with a threshold $\varepsilon = 10^{-14}$. If the iteration does not satisfy the criterion within 50 iterations, then we consider the trajectory divergent. Here,

TABLE I. Numerical performance of the extended Newton-Raphson method. For five different values of β, we evaluate: (i) the average number of iterations required for convergence (Iteration number); (ii) the percentage of initial points that converge (Convergence percentage); and (iii) the relative computation time compared to the $\beta = 0$ case (Computation time). Note that $\beta = 0$ represents the original Newton-Raphson method. In each category, the best performance—shortest iteration number, highest convergence percentage, and least computation time among different β values—is highlighted in bold.

Functions	Iteration number					Convergence percentage					Computation time				
	$\beta = -1$	-0.5	0	0.5	1	$\beta = -1$	-0.5	0	0.5	1	$\beta = -1$	-0.5	0	0.5	1
$f_1(x) = (x^2 - 1)(x^2 + 1)$	18.4	15.7	11.3	11.0	**10.5**	75	80	**100**	91	90	1.50	1.29	1.0	0.92	**0.89**
$f_2(x) = x^3 - 1$	15.5	13.3	9.1	9.1	**8.4**	92	96	**100**	99	99	1.55	1.35	1.0	0.94	**0.93**
$f_3(x) = x^{12} - 1$	18.5	16.3	16.4	12.7	**11.5**	44	56	**84**	67	66	1.11	0.93	1.0	0.73	**0.67**
$f_4(x) = (x^2 - 4)(x + 1.5)(x - 0.5)$	12.6	10.4	8.0	7.4	**6.4**	98	97	**100**	**100**	**100**	1.50	1.27	1.0	0.89	**0.79**
$f_5(x) = (x + 2)(x + 1.5)^2(x - 0.5)(x - 2)$	12.1	**10.3**	28.9	25.6	22.2	42	43	96	**99**	**99**	0.42	**0.35**	1.0	0.85	0.76
$f_6(x) = \sin(x)$	9.7	7.7	6.6	6.0	**5.5**	87	84	**100**	95	92	1.42	1.09	1.0	0.87	**0.82**
$f_7(x) = (x - 1)^3 + 4(x - 1)^2 - 10$	17.9	14.5	9.7	9.1	**8.4**	97	97	**100**	**100**	**100**	1.74	1.44	1.0	0.92	**0.83**
$f_8(x) = \sin(x - 14/10)^2 - (x - 14/10)^2 + 1$	11.3	9.4	8.9	7.5	**6.5**	60	59	**71**	64	64	1.25	1.04	1.0	0.86	**0.76**
$f_9(x) = x^2 - e^x - 3x + 2$	8.1	7.1	6.2	5.8	**5.0**	97	97	**100**	99	98	1.20	1.13	1.0	0.93	**0.79**
$f_{10}(x) = \cos(x - 3/4) - x + 3/4$	10.8	8.8	8.6	7.1	**6.3**	61	63	**93**	75	72	1.23	0.97	1.0	0.8	**0.72**
$f_{11}(x) = (x + 1)^3 - 1$	15.1	12.7	9.0	8.9	**8.1**	93	96	**100**	99	99	1.50	1.29	1.0	0.96	**0.85**
$f_{12}(x) = (x - 2)^3 - 10$	17.5	14.7	9.3	9.8	**8.9**	87	92	**100**	97	98	1.84	1.54	1.0	1.01	**0.96**
$f_{13}(x) = (x + 5/4)e^{(x+5/4)^2} - \sin(x + 5/4)^2$ $+3\cos(x + 5/4) + 5$	17.0	13.2	11.6	9.7	**9.4**	55	60	**96**	82	80	1.46	1.12	1.0	0.81	**0.80**
$f_{14}(x) = x + \sin(2/x)x^2$	15.5	13.5	10.9	**9.5**	10.3	35	57	**99**	88	74	1.41	1.22	1.0	**0.86**	1.08

the convergence order was estimated using the following formula [30]:

$$q_n = \frac{\log |e_{n+1}/e_n|}{\log |e_n/e_{n-1}|}, \qquad (46)$$

where $e_n \equiv z_n - z_{n-1}$. When $|e_n| < \varepsilon$ the iterations stop and we pick the last value of the series q_n as the order of convergence. For each function, we select an initial point that

converges to one of the roots for at least eight iterations to obtain a valid estimation of the convergence.

We investigate the impact of the parameter β in the extended Newton-Raphson method by using some representative values of $\beta \in \{-1, -0.5, 0, 0.5, 1\}$. We found that positive values of β improve the efficiency of the root-finding iteration compared to the original Newton-Raphson method ($\beta = 0$). Specifically, the case $\beta = 1$ consistently shows superior performance (Table I). The mean number of iterations

TABLE II. Annealing effect on iterative root finding. The numerical performances of fixed β versus annealing β_n are compared. The evaluations include: (i) the average number of iterations required for convergence (Iter. number); (ii) the percentage of initial points that converge (Conv. percentage); (iii) the relative computation time compared to the $\beta = 0$ case (Comp. time); and (iv) the order of convergence (Conv. order). Note that the fixed $\beta = 0$ and $\beta = 1$ represent the original and the modified Newton-Raphson method, respectively. The iteration-dependent β_n is formulated in Eq. (71). The convergence order is estimated by Eq. (46). In each category, the best performance—shortest iteration number, highest convergence percentage, least computation time, and largest convergence order among different β values—is highlighted in bold.

| Functions	Iter. number			Conv. percentage			Comp. time			Conv. order		
	$\beta = 0$	$\beta = 1$	β_n	$\beta = 0$	$\beta = 1$	β_n	$\beta = 0$	$\beta = 1$	β_n	$\beta = 0$	$\beta = 1$	β_n
$f_1(x) = (x^2 - 1)(x^2 + 1)$	11.3	10.5	**7.9**	**100**	90	**100**	1.0	**0.89**	1.02	2.00	2.99	**4.08**
$f_2(x) = x^3 - 1$	9.1	8.4	**6.5**	**100**	99	**100**	1.0	**0.93**	1.07	2.00	3.01	**4.14**
$f_3(x) = x^{12} - 1$	16.4	**11.5**	12.8	84	66	**87**	1.0	**0.67**	1.26	2.00	2.92	**4.17**
$f_4(x) = (x^2 - 4)(x + 1.5)(x - 0.5)$	8.0	6.4	**5.4**	**100**	**100**	**100**	1.0	**0.79**	1.07	1.99	2.94	**3.95**
$f_5(x) = (x + 2)(x + 1.5)^2(x - 0.5)(x - 2)$	28.9	22.2	**18.6**	96	99	**100**	1.0	**0.76**	1.17	0.99	**1.00**	**1.00**
$f_6(x) = \sin(x)$	6.6	5.5	**5.0**	**100**	92	99	1.0	**0.82**	1.18	3.00	4.97	**6.85**
$f_7(x) = (x - 1)^3 + 4(x - 1)^2 - 10$	9.7	8.4	**7.3**	**100**	**100**	**100**	1.0	**0.83**	1.44	2.00	3.02	**4.36**
$f_8(x) = \sin(x - 14/10)^2 - (x - 14/10)^2 + 1$	8.9	6.5	**6.1**	71	64	**73**	1.0	**0.76**	1.27	2.00	**2.98**	2.58
$f_9(x) = x^2 - e^x - 3x + 2$	6.2	5.0	**4.2**	**100**	98	**100**	1.0	**0.79**	0.99	2.00	3.09	**4.25**
$f_{10}(x) = \cos(x - 3/4) - x + 3/4$	8.6	**6.3**	**6.3**	**93**	72	**93**	1.0	**0.72**	1.17	2.00	3.02	**3.89**
$f_{11}(x) = (x + 1)^3 - 1$	9.0	8.1	**6.3**	**100**	99	**100**	1.0	**0.85**	1.23	2.00	2.93	**4.08**
$f_{12}(x) = (x - 2)^3 - 10$	9.3	8.9	**7.0**	**100**	98	**100**	1.0	**0.96**	1.34	2.00	2.99	**3.92**
$f_{13}(x) = (x + 5/4)e^{(x+5/4)^2} - \sin(x + 5/4)^2 + 3\cos(x + 5/4) + 5$	11.6	9.4	**8.5**	96	80	**97**	1.0	**0.80**	1.34	2.00	3.02	**3.64**
$f_{14}(x) = x + \sin(2/x)x^2$	10.9	10.3	**8.0**	**99**	74	81	**1.0**	1.08	1.28	2.00	2.99	**3.99**
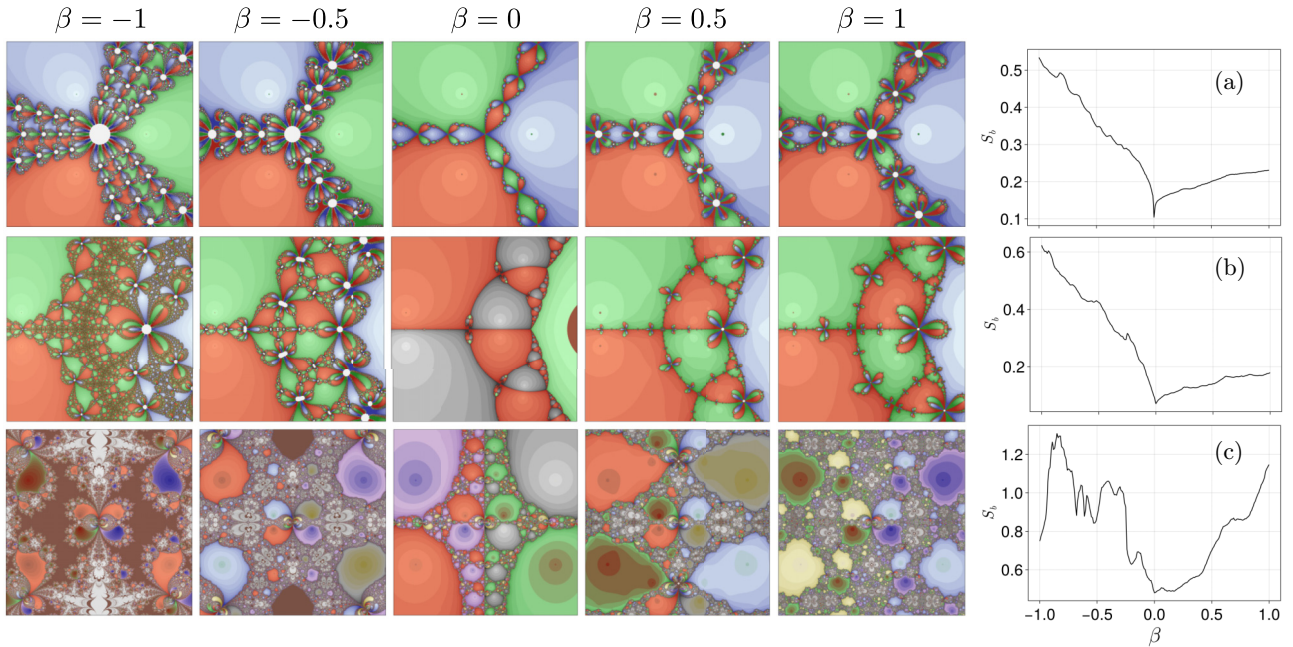
FIG. 3. Newton fractals and basin entropy. Colors represent different basins of attractors (roots). In particular, the colors in the basins of attraction have been shaded such that darker colors correspond to longer iterations to converge to roots. The basins of three functions are computed for $\beta \in \{-1, -0.5, 0, 0.5, 1\}$: (a) top row $f_2(z) = z^3 - 1$, (b) central row $f_7(z) = (z-1)^3 + 4(z-1)^2 - 10$, and (c) bottom row $f_{14}(z) = z + z^2 \sin(2/z)$. The corresponding basin entropy S_b in the right plot quantifies the unpredictability of roots as a function of β over the range $[-1, 1]$ in steps of 0.01. S_b has been computed on a grid of 1000 times 1000 initial points with a covering of boxes of size 20 times 20. The method to determine the final root is described in Sec. III D.

decreases monotonically as β changes from -1 to 1. However, this trend for positive β is accompanied by higher instabilities as the convergence percentage of initial points decreases. The relative computation time compared to the original Newton-Raphson method is reduced. Since additional operations for the extended method are negligible, the reduced number of iterations directly leads to shorter computation times.

In general, negative values of β lead to poorer performance, except for the function f_5. A closer examination of the convergence percentage reveals more than half of the initial points do not converge for negative β. This can be understood visually from Fig. 1, where unlike postive β, negative β causes the next iteration points to move farther away from the roots.

Table II showed quadratic convergence for $\beta = 0$ and cubic convergence for $\beta = 1$. Notably, we encountered anomalous convergence in the case of the function $f_6(x) = \sin(x)$, which exhibits vanishing curvature with $f_6''(x_{\mathrm{root}}) = 0$. Additionally, we noted linear convergence near a doubly degenerate real root for $f_5(x)$.

E. Basin entropy

For nonlinear functions with multiple roots, different initial points z_0 approach different roots or zeros or attractors. Consequently, the set of initial points converging to the same root z^* is defined as the basin of attraction of the root z^*. It is well known that the basin of attractors exhibits a fractal structure for the Newton-Raphson method. Here, we investigate whether our iteration method alters the Newton-Raphson basin of attractors. A remarkable characteristic of the basins

resulting from the numerical method is the intricate nature of the boundary between them. For initial points located on this boundary, the uncertainty in the final root is maximized. This aspect could be perceived as a drawback of the method, as a "poor" initial point might result in convergence to any root of the function. Nonetheless, for functions with unknown roots, our interest lies in identifying all roots in principle. As we will explore in the next section, this characteristic can also contribute to an enhancement in the speed of convergence.

We investigate the impact of our method on the boundaries of the basins generated by iterations from initial points. Upon initial inspection of the basins for the function $f_2(z) = z^3 - 1$, a significant alteration in the boundary between $\beta = -1$ and $\beta = 1$ becomes evident [Fig. 3(a) and top row]. Formally, in the complex plane, this boundary constitutes the set of points that never converge to a root and forms a Julia set [31]. Furthermore, the example illustrated in Fig. 3(a) top row exhibits the property of Wada, wherein a single boundary simultaneously separates three basins. This property entails a unique form of unpredictability, as a point on the boundary can ultimately converge to any of the roots. Similar phenomena have been observed on boundaries for other modified Newton methods dependent on parameters, as discussed in Ref. [32], where the modifications introduce nontrivial transformations of the boundaries. Here, we provide quantitative insight into these transformations.

The unpredictability of the final root for initial points near the boundary can be quantified using the *basin entropy* [33]. This metric assesses the local entropy within boxes of side length l by initially estimating the probability p_i of each final attractor i inside the box with the naïve frequency estimator.

These estimated probabilities serves to compute the Gibbs (or Shannon) entropy of the box: $S_b = -\sum_i p_i \log p_i$. The basin entropy is then calculated as the average of the box entropy with a covering of nonoverlapping boxes over the portion of the phase space studied.

In the examples shown in Fig. 3, the basins of the roots are computed over a grid of 1000×1000 initial points in the complex plane using the methods of Sec. III D. A box size of 20×20 initial points is used to estimate the p_i probabilities.

In Fig. 3, we depict the evolution of the basin entropy for three functions across the range of β values from -1 to 1. It is evident that unpredictability consistently increases for $\beta \neq 0$. While we cannot offer a rigorous explanation for this phenomenon, we can provide heuristic arguments regarding the emergence of new fractal structures in the phase space. One mechanism contributing to the formation of fractal boundaries is the stretching and folding action of small areas. The roots of the derivative $f'(x)$ serve as sources of instabilities in the Newton-Raphson method, as initial points near these points can be dramatically dispersed. Subsequent iterations tend to bring these points back toward one of the roots. Small variations in the initial points near these roots can result in convergence to any of the roots, characteristic of chaotic behavior.

We consider our iterative method, expressed as $x_{n+1} = N(x_n) - \beta f(N(x_n))/f'(x_n)$, where $N(x) = x - f(x)/f'(x)$. When close to a singular x_s, where $f'(x_s) = 0$, we can approximate a linear relation, $f'(x) \simeq c_2 \delta x$ with $c_2 \equiv f''(x_s)$ and $\delta x \equiv x - x_s$. Consequently, $N(x) \simeq -c_0/\delta x$ with $c_0 \equiv f(x_s)$. The amplitude of the next iteration, given by

$$x_{n+1} \simeq -\frac{c_0}{\delta x_n} - \frac{\beta f(-c_0/\delta x_n)}{c_2 \delta x_n}, \tag{47}$$

depends on the proximity to the singularity, $\delta x_n = x_n - x_s$, and the value of $f(-c_0/\delta x_n)$. Generally, uncertainty increases near these points due to these significant jumps. For complex functions, new structures emerge near the singularity, visible as the blobs in Fig. 3 for $\beta \neq 0$.

This rise in basin entropy signifies an enhanced mixing property in the phase space, facilitating more thorough exploration of roots. The algorithm can traverse a broader range of regions in the phase space before settling on a local solution. A recent study implemented a deflated version of the Newton-Raphson method [34], aiming to reveal additional roots of a function by avoiding convergence to already known solutions. Similarly, our parameterized Newton-Raphson method enhances root searching efficiency in phase space due to the fractal nature of the boundary.

IV. LINK TO THE ADOMIAN METHOD

The self-consistent stationary Eq. (34) is reminiscent of the Adomian method [35,36]. Notably, the Adomian method has previously yielded an enhanced Newton-Raphson approach, corresponding precisely to the scenario when $\beta = 1$ in our formulation. In this section, we provide an overview of the Adomian method, establish connections between our approach and the Adomian method, and extend the fixed β method to an annealing approach with varying β.

A. Adomian method

We first give an extensive introduction to the Adomian method because it may not be familiar to all readers. The canonical form for employing the Adomian decomposition is given by

$$a = C + F(a), \tag{48}$$

where C is a constant and $F(a)$ is a nonlinear function of the variable a. Utilizing a series solution approach, we express

$$a = a_0 + a_1 + a_2 + \cdots,$$
$$F(a) = A_0(a_0) + A_1(a_0, a_1) + A_2(a_0, a_1, a_2) + \cdots. \tag{49}$$

Equating terms between the two sides in Eq. (48) requires explicit choices because there is no formal expansion parameter in the series. With specific choices, this leads to the following relationships:

$$
\begin{aligned}
a_0 &= C, \\
a_1 &= A_0(a_0), \\
a_2 &= A_1(a_0, a_1), \\
&\ \ \vdots,
\end{aligned} \tag{50}
$$

where A_n represents the Adomian polynomial:

$$A_n = \frac{1}{n!}\left[\frac{d^n}{d\beta^n}F(a_0 + \beta a_1 + \beta^2 a_2 + \cdots)\right]_{\beta=0}. \tag{51}$$

Here are a few initial terms:

$$
\begin{aligned}
A_0(a_0) &= F(a_0), \\
A_1(a_0, a_1) &= a_1 F'(a_0).
\end{aligned} \tag{52}
$$

We emphasize that there is no unambiguous way to compare terms between the left and right hand sides of Eq. (50): For example,

$$
\begin{aligned}
a_0 &= C - \epsilon, \\
a_1 &= \epsilon - \epsilon^2 + A_0(a_0), \\
a_2 &= \epsilon^2 - \epsilon^3 + A_1(a_0, a_1), \\
&\ \ \vdots
\end{aligned} \tag{53}
$$

for an arbitrary ϵ with $|\epsilon| < 1$, also solves Eq. (48). Results on convergence have been proved with hypotheses on the size of derivatives and it has been noted that the decomposition must be chosen appropriately [37], as there is no canonical choice for the matching of terms.

Let us apply the Adomian decomposition to the root-finding problem, $f(x) = 0$. Utilizing a linear approximation of $f(x)$ in proximity to x gives

$$f(x - a) = f(x) - af'(x). \tag{54}$$

Rearranging this equation into the canonical form for Adomian decomposition yields

$$a = \underbrace{\frac{f(x)}{f'(x)}}_{C} - \underbrace{\frac{f(x-a)}{f'(x)}}_{F(a)}. \tag{55}$$

By focusing solely on the zeroth-order term, $a_0 = C = f(x)/f'(x)$, we can approximate $a \approx a_0$. The iterative update of $x_{n+1} = x_n - a$ is expected to satisfy $f(x_{n+1}) = f(x_n - a) = 0$ for root finding. Then, $x_n - x_{n+1} = a = a_0$ results in

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \qquad (56)$$

which is equivalent to the Newton-Raphson method.

Chun [23] has expanded the analysis by considering an additional step:

$$f(x - a) = f(x) - af'(x) + \underbrace{\frac{1}{2}a^2 f''(x)}_{g(x,a)}. \qquad (57)$$

Once again, rearranging this equation, assuming $f(x - a) = 0$ at $x - a$, allows it to be expressed in the canonical form for Adomian decomposition:

$$a = \underbrace{\frac{f(x)}{f'(x)}}_{C} + \underbrace{\frac{g(x, a)}{f'(x)}}_{F(a)}. \qquad (58)$$

This decomposition is certainly not unique (see Abbasbandy [38] for an alternative decomposition).

This time, considering $a \approx a_0 + a_1$ up to the first order, the Adomian decomposition provides

$$a_0 = C = \frac{f(x)}{f'(x)}, \qquad (59)$$

$$a_1 = F(a_0) = \frac{g(x, a_0)}{f'(x)} = \frac{f(x - a_0)}{f'(x)}, \qquad (60)$$

where we utilized $g(x, a_0) = f(x - a_0) - f(x) + a_0 f'(x) = f(x - a_0)$. Consequently,

$$a \approx a_0 + a_1 = \frac{f(x)}{f'(x)} + \frac{f(x - a_0)}{f'(x)}. \qquad (61)$$

Again, given that $x_n - x_{n+1} = a = a_0 + a_1$, this leads to

$$\begin{aligned} x_{n+1} &= x_n - \frac{f(x_n)}{f'(x_n)} - \frac{f\left(x_n - \frac{f(x_n)}{f'(x_n)}\right)}{f'(x_n)} \\ &= N(x_n) - \frac{f(N(x_n))}{f'(x_n)}, \end{aligned} \qquad (62)$$

which corresponds to Eq. (37) in our formulation with $\beta = 1$.

The Adomian polynomial approach, when truncated to a specific number of terms in a_i, $i \leqslant m$, offers a pathway to derive alternative root-finding iterations. Although the higher-order forms involve algebraic complexities, these truncations result in progressively higher orders of convergence for well-behaved functions. This observation led to Chun's conjecture [23] that such truncations exhibit convergence order $m + 2$. Therefore, when $m = 0$, it corresponds to the Newton-Raphson iteration, which demonstrates quadratic convergence.

Let us revisit our self-consistent stationary Eq. (34):

$$x(\beta) = \underbrace{N(x_0)}_{C} - \underbrace{\frac{\beta f(x(\beta))}{f'_0}}_{F(x(\beta))}. \qquad (63)$$

This equation adheres to the canonical form of the Adomian method. One notable distinction is the inclusion of the parameter β. In the limit of small β, the so-called "high-temperature" limit, the second term can be regarded as a perturbation. We now explore the high-temperature limit, and consider a series solution:

$$x(\beta) = a_0 + \beta a_1 + \beta^2 a_2 + \dots. \qquad (64)$$

Here, the nonlinear function $F(x(\beta))$ can also be expressed using the Adomian polynomial in Eq. (51). When we truncate $x(\beta) = a_0 + \beta a_1$ to the first order of β, we obtain:

$$\begin{aligned} a_0 &= N(x_0), \\ a_1 &= -\frac{f(a_0)}{f'_0}. \end{aligned} \qquad (65)$$

This yields:

$$x(\beta) = a_0 + \beta a_1 = N(x_0) - \frac{\beta f(N(x_0))}{f'_0}, \qquad (66)$$

which corresponds to Eq. (35). Hence, our formulation can be interpreted within the context of the Adomian method. However, the inclusion of the auxiliary parameter β holds significant importance in the Adomian decomposition. It ensures an unambiguous comparison between series and the Adomian polynomial, providing an exact term-by-term mapping with correct orders of β.

We note that there is no need to use Eq. (49) at all as we can work directly with $x(\beta)$ in our iteration, Eq. (66). Also noteworthy is the fact that as we directly find $x(\beta)$ fixed points, no derivatives of f beyond the first are ever explicitly required in the iteration.

B. Annealing

Indeed, the physical intuition gained from considering Eq. (63) as a high-temperature expansion suggests that annealing the temperature during iterations could improve the performance of the root-finding algorithm. In our physical analogy, the auxiliary parameter β in Eq. (37) can be interpreted as the inverse of temperature in statistical mechanics. With the high-temperature analogy in mind, it is natural to view the original Newton-Raphson update as the high-temperature limit ($\beta = 0$), where only the linear Taylor expansion of the function is used, making the hopping term dominant. In contrast, the Adomian update represents the low-temperature limit ($\beta = 1$), where the exact minimum encoded in the β-dependent term is balanced with the linear hopping term. As the temperature is decreased, the importance of the ultralocal exact root should increase. While exploration over as broad an area as possible is essential at the beginning, focusing on the root location by gradually decreasing the temperature should help in getting to the actual root. In fact, if we increase the inverse temperature from $\beta = 0$, which corresponds to Newton-Raphson, to a final value β after even one step, then we find that the total number of iterations needed decreases and the dependence of the number of iterations on the final value of β becomes much smoother as well. However, if β is ramped up too gradually, then there is no benefit to be had for most of the test functions we considered. This

annealing approach is similar to multipoint methods in that the update formulation changes with each iteration step [39].

Let us now derive an annealing schedule from first principles. First, we introduce shorthand notation:

$$f_n \equiv f(x_n), \quad f_n' \equiv f'(x_n),$$

$$\hat{f}_{n+1} \equiv f(\hat{x}_{n+1}), \quad \hat{f}_{n+1}' \equiv f'(\hat{x}_{n+1}).$$

Then, our update is represented as

$$\hat{x}_{n+1} = x_n - \frac{f_n}{f_n'},$$

$$x_{n+1} = \hat{x}_{n+1} - \beta \frac{\hat{f}_{n+1}}{f_n'}, \tag{67}$$

where $\hat{x}_{n+1}$ is the Newton-Raphson update. A Taylor expansion of $f(x_{n+1})$ with the second equation gives the following relation:

$$f(x_{n+1}) = f\left(\hat{x}_{n+1} - \beta \frac{\hat{f}_{n+1}}{f_n'}\right)$$

$$\approx f(\hat{x}_{n+1}) - f'(\hat{x}_{n+1})\beta \frac{\hat{f}_{n+1}}{f_n'}. \tag{68}$$

This implies

$$f_{n+1}^2 = \left[\hat{f}_{n+1} - \beta \frac{\hat{f}_{n+1}'\hat{f}_{n+1}}{f_n'}\right]^2 \tag{69}$$

with our shorthand notation. Therefore,

$$f_{n+1}^2 \leqslant \hat{f}_{n+1}^2, \quad \text{if} \quad \left|1 - \beta \frac{\hat{f}_{n+1}'}{f_n'}\right| \leqslant 1. \tag{70}$$

In other words, our update makes $f^2(x_{n+1})$ smaller than the Newton-Raphson update $f^2(\hat{x}_{n+1})$, when $\beta \hat{f}_{n+1}'/f_n'$ is small enough to justify the Taylor expansion of f around $\hat{x}_{n+1}$. Now, suppose we take

$$\beta_n \equiv \frac{2f_n'^2}{\hat{f}_{n+1}'^2 + f_n'^2} \geqslant 0. \tag{71}$$

This choice guarantees Eq. (70) because

$$1 - \beta_n \frac{\hat{f}_{n+1}'}{f_n'} = \frac{(\hat{f}_{n+1}' - f_n')^2}{\hat{f}_{n+1}'^2 + f_n'^2} \leqslant 1. \tag{72}$$

Therefore, the annealing schedule in Eq. (71) suggests an approach to setting β_n depending on the derivatives of the function at the previous value of x_n and the Newton-Raphson update value $\hat{x}_{n+1}$. Equation (71) has appropriate limits of 1 or 0 when f_n' is large or small depending on the value of $\hat{f}_{n+1}'$ of course. When f_n' is large and x_n is proximate to roots, we expect the two derivative values are close, so $\beta_n \approx 1$ with $|1 - \beta_n \hat{f}_{n+1}'/f_n'| \leqslant 1$. When it is small at the singularity, however, $\beta_n \approx 0$ with a finite derivative value $\hat{f}_{n+1}'$.

Table II summarizes the results for fixed values of β, and for the annealing schedule β_n defined in Eq.(71). The annealing schedule enhances the effectiveness of the β value near roots. The estimated order of convergence near a root is $q_n \simeq 4$ for most functions.

The mean number of iterations has been computed for a 1000×1000 regularly spaced grid using the same methods and parameters as in Sec. III D. The best results have been highlighted in bold. There is no function for which the Newton-Raphson algorithm performs better than the annealing schedule. We caution that the derived annealing schedule in Eq. (71) does require an additional derivative evaluation $f'(\hat{x}_{n+1})$. This extra computational cost is reflected in the total time per iterations. The time per point is indeed higher despite the reduction in the number of iterations due to the added derivative. A possible optimization is to approximate this derivative with first-order approximations using the previously evaluated functions.

In the supplemental Julia scripts available at [40], computations are provided for all the standard test functions listed in Table I.

V. DISCUSSION

Root finding is a technique that is central to many quantitative science and engineering problems. This paper has investigated an alternative root-finding approach that has the potential to offer several benefits: improved efficiency and enhanced numerical stability, broader applicability to complex nonlinear functions that abound in real-world problems, and more thorough investigation of possible roots because of higher basin entropy, all while not increasing the order of derivatives needed for application. More fundamentally, the physical picture of the variation in the root basins of attraction as the temperature is decreased and the cohomological field theory explication of roots as transcendental cohomology classes are, to the best of our knowledge, new contributions to the theoretical underpinnings of numerical analysis and algorithms. We have just scratched the surface of the applications of this technique, as is evident from the connection we made to the Adomian method, which has been used for solving nonlinear, and even stochastic, ordinary and partial differential equations. This connection implies that our approach could be applied to all these problem areas as well.

In particular, the homotopy analysis method (HAM) is a well-known approach to solving nonlinear problems due to Liao [41], motivated by topological homotopy theory. HAM unifies the Adomian method and a host of other numerical methods using an auxiliary parameter that constructs a homotopy to handle the nonlinear nature of the problem. This so-called convergence control parameter is then used to show convergence of a series solution. As HAM can be combined with spectral or Padé approximation methods, it would be interesting to see if HAM could be combined with our approach, Eq. (32), to completely avoid a series expansion.

APPENDIX A: TOPOLOGICAL FORMULATION OF ROOT FINDING

We formulate a rigorous topological formulation of root finding to show that root-finding iterations are explicit representatives of the same cohomology class. The derivation below is self-contained but we note that it is an extension of cohomological quantum field theory [42].

We introduce two anticommuting variables, b and c, with the following properties:

$$bc = -cb, \quad b^2 = 0, \quad c^2 = 0, \tag{A1}$$

and a Grassmann integration over these variables defined by

$$\int db1 = 0, \quad \int db\,b = 1, \quad \int dc1 = 0,$$

$$\int dc\,c = 1, \quad \int db\,c = 0, \quad \int dc\,b = 0, \tag{A2}$$

extended by linearity and all the standard properties of integration. We also introduce an anticommuting derivation, s (called the BRST operator in the physics literature) as follows:

$$sx = c, \quad sc = 0, \quad sb = \lambda, \quad s\lambda = 0, \tag{A3}$$

where λ is a commuting variable for consistency with the fact that combinations of anticommuting quantities, in this case s and b, are commuting quantities. It is trivial to verify that $s^2 = 0$, which implies that s can be used to define a cohomology theory.

Now we define a gauge fermion depending on a parameter g as follows:

$$\Psi = b(if(x) - g^2\lambda/2), \tag{A4}$$

where f is the function for which we want to find roots. The action of s on Ψ gives

$$s\Psi = \lambda(if(x) - g^2\lambda/2) - ibf'(x)c, \tag{A5}$$

where the negative sign of the second term arises when the derivation s anticommutes past b.

Consider the measure

$$d\mu \equiv dx\,d\lambda\,db\,dc \exp(s\Psi)g/\sqrt{2\pi}, \tag{A6}$$

which is invariant under the action of s. If we integrate a general function $h(x, c, b, \lambda)$ with respect to this measure, then the value of the integral will depend on the parameter g. However, suppose that h satisfies $sh = 0$. Then a change in g will lead to a change in the value of the integral of the form

$$\delta\mathcal{I} = \int d\mu(-sc\lambda/2)h, \tag{A7}$$

but the invariance of the measure under the action of s allows an integration by parts to give us

$$\delta\mathcal{I} = \int d\mu(c\lambda/2)sh = 0. \tag{A8}$$

Thus, for very specific functions that satisfy $sh = 0$, the value of their integral does not depend on the parameter g. Moreover, if we change h to $h + sk$, where k is an arbitrary function

of the variables, then we see by the same argument that the value of the integral of h does not depend on k. Functions that satisfy $sh = 0$ are called closed functions and functions that are of the form sk are called exact functions.

These are the axioms that define a cohomology theory, and therefore closed functions fall into equivalence classes, defined up to the arbitrary addition of exact functions.

How is this formal development relevant for finding roots? A formal Laplace approximation to the measure gives the "equations of motion":

$$x_{\text{root}}: f(x_{\text{root}}) = 0, \quad \lambda = 0, \quad cf'(x_{\text{root}}) = 0, \quad f'(x_{\text{root}})b = 0. \tag{A9}$$

Thus, this measure is localizing at the roots of f. Integrating over λ in

$$\mathcal{Z} = \int d\mu \exp(s\Psi) \tag{A10}$$

gives

$$\hat{\mathcal{Z}} = \int dx\,db\,dc \exp(-f(x)^2/2g^2 - icf'(x)b), \tag{A11}$$

where we should now expect to use the λ equation of motion to verify that $s^2 = 0$. The striking thing about this expression is that if we make g small, we see that any closed function will depend only on the value of x at which $f(x) = 0$. Therefore, this formulation of our physical approach exhibits root finding as a mathematically rigorous cohomology theory. Any nontrivial cohomology class in this problem will have to be a transcendental function, since it is easily demonstrated that the cohomology class of any polynomial function of the variables is guaranteed to be trivial.

Let us now demonstrate that the cohomology operator s can help us check if an expression for a root of $f(x)$ is, or is not, g independent. Consider the formal series

$$x_{\text{root}} \equiv x - h(x) - h(x - h(x)) - \ldots, \tag{A12}$$

where we have defined $h \equiv f/f'$, and this is simply the series of corrections of the Newton-Raphson algorithm. We claim that $sx_{\text{root}} = 0$. This is explicitly verified as follows:

$$sx_{\text{root}} = c - \frac{f'(x)}{f'(x)}c + \frac{f(x)}{f'(x)^2}c + \ldots, \tag{A13}$$

where the first two terms obviously cancel. This general pattern continues because s acting on the nth term gives

$$s(x_n - x_{n-1}) = \frac{f(x_{n-1})}{f'(x_{n-1})^2}sx_{n-1} - sx_{n-1}. \tag{A14}$$

As the previous $n - 1$ terms sum up precisely to sx_{n-1} because this is a telescoping series, we have a cancellation. Since the equation of motion is $f(x_{\text{root}}) = 0$, we see that if the series converges, it is a representative of the cohomology class defined by the root.

In particular, some algebra shows that our β dependent variations on the representative series also have the same type

of telescoping structure:

$$s(x_n - x_{n-1}) = \frac{f(x_{n-1})}{f'(x_{n-1})^2}\left[1 + \beta\left(\frac{f(\hat{x}_{n-1})}{f(x_{n-1})} - \frac{f'(\hat{x}_{n-1})}{f'(x_{n-1})}\right)\right]sx_{n-1} - sx_{n-1}, \tag{A15}$$

where we have defined $\hat{x}_{n-1} \equiv x_{n-1} - h(x_{n-1})$. Thus, independent of β, this defines a cohomology class. To see the promised simplification produced by our choice of coefficient in Eq. (25), a little scrutiny reveals that if the series is converging, the term $f'(\hat{x}_{n-1})/f'(x_{n-1}) \to 1$, which leads to a cancellation at $\beta = 1$, giving

$$s(x_n - x_{n-1}) = \frac{f(\hat{x}_{n-1})}{f'(x_{n-1})^2}sx_{n-1} - sx_{n-1}. \tag{A16}$$

As $\hat{x}_{n-1}$ is the Newton-Raphson iterate of x_{n-1}, this suggests faster convergence, provided of course that the assumptions made in this simplification are valid. This cohomological explanation also removes any mystery in the invariance of the roots no matter what value of g we use for approximating the integral: we simply have to ensure that the expression for the root is a closed function for the cohomology operator s. We show in Sec. III B that the faster convergence suggested above can be explicitly demonstrated with the usual derivation used to show the quadratic convergence of the Newton-Raphson algorithm.

[1] A. Abad, R. Barrio, and A. Dena, Phys. Rev. E **84**, 016701 (2011).

[2] D. Wang, P. Wang, and Y. Lan, Phys. Rev. E **98**, 042204 (2018).

[3] R. Amritkar, Phys. Rev. E **80**, 047202 (2009).

[4] D. J. Wales and J. P. Doye, J. Chem. Phys. **119**, 12409 (2003).

[5] D. Mehta, J. D. Hauenstein, and M. Kastner, Phys. Rev. E **85**, 061103 (2012).

[6] T. E. Woolley, R. E. Baker, P. K. Maini, J. L. Aragón, and R. A. Barrio, Phys. Rev. E **82**, 051929 (2010).

[7] U. Naether, R. A. Vicencio, and M. Johansson, Phys. Rev. E **83**, 036601 (2011).

[8] S. Tomsovic, Phys. Rev. E **98**, 023301 (2018).

[9] D. Pfefferlé and S. I. Abarzhi, Phys. Rev. E **102**, 053107 (2020).

[10] P. Maragakis, M. Spichty, and M. Karplus, Phys. Rev. Lett. **96**, 100602 (2006).

[11] M. A. Herrada and J. M. Montanero, J. Comput. Phys. **306**, 137 (2016).

[12] K. Sääskilahti, J. Oksanen, and J. Tulkki, Phys. Rev. E **88**, 012128 (2013).

[13] Y. Kuramoto, in *Proceedings of the International Symposium on Mathematical Problems in Theoretical Physics: January 23–29, 1975, Kyoto University, Kyoto/Japan* (Springer, Berlin, 1975), pp. 420–422.

[14] G. Buzsaki and A. Draguhn, Science **304**, 1926 (2004).

[15] C. M. Kim, U. Egert, and A. Kumar, Phys. Rev. E **102**, 022308 (2020).

[16] W. H. Press, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed. (Cambridge University Press, Cambridge, UK, 2007).

[17] T. J. Ypma, SIAM Rev. **37**, 531 (1995).

[18] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, Science **220**, 671 (1983).

[19] G. P. Lepage and P. B. Mackenzie, Phys. Rev. D **48**, 2250 (1993).

[20] N. Svaiter, Physica A **345**, 517 (2005).

[21] S. Amat, S. Busquier, and S. Plaza, Appl. Math. Comput. **154**, 735 (2004).

[22] S. Amat, S. Busquier, and S. Plaza, J. Math. Anal. Appl. **366**, 24 (2010).

[23] C. Chun, Comput. Math. Appl. **50**, 1559 (2005).

[24] H. H. Homeier, J. Comput. Appl. Math. **169**, 161 (2004).

[25] H. H. Homeier, J. Comput. Appl. Math. **176**, 425 (2005).

[26] M. Grau and M. Noguera, Appl. Math. Lett. **17**, 509 (2004).

[27] J. F. Traub, *Iterative Methods for the Solution of Equations* (American Mathematical Society, Providence, RI, 1982), Vol. 312.

[28] M. S. Petković, B. Neta, L. D. Petković, and J. Džunić, Appl. Math. Comput. **226**, 635 (2014).

[29] S. Weerakoon and G. Fernando, Appl. Math. Lett. **13**, 87 (2000).

[30] M. Grau-Sánchez, M. Noguera, and J. M. Gutiérrez, Appl. Math. Lett. **23**, 472 (2010).

[31] K. Falconer, *Fractal Geometry: Mathematical Foundations and Applications* (John Wiley & Sons, New York, NY, 2004).

[32] H. Susanto and N. Karjanto, Appl. Math. Comput. **215**, 1084 (2009).

[33] A. Daza, A. Wagemakers, B. Georgeot, D. Guéry-Odelin, and M. A. Sanjuán, Sci. Rep. **6**, 31416 (2016).

[34] J. Cisternas and A. Concha, Chaos, Solitons Fractals **179**, 114468 (2024).

[35] G. Adomian, *Solving Frontier Problems of Physics: The Decomposition Method* (Springer Science & Business Media, Cham, 1994).

[36] R. Rach, Kybernetes **41**, 1087 (2012).

[37] K. Abbaoui and Y. Cherruault, Math. Comput. Model. **20**, 69 (1994).

[38] S. Abbasbandy, Appl. Math. Comput. **145**, 887 (2003).

[39] M. S. Petković and L. D. Petković, Appl. Anal. Discrete Math. **4**, 1 (2010).

[40] https://github.com/awage/RootFinding.

[41] S. Liao, *Beyond Perturbation: Introduction to the Homotopy Analysis Method* (Chapman and Hall/CRC, Boca Raton, FL, 2003).

[42] E. Witten, Int. J. Mod. Phys. A **06**, 2775 (1991).