Addressing Discontinuous Root-Finding for Subsequent Differentiability in Machine Learning, Inverse Problems, and Control

Daniel Johnson[1], Ronald Fedkiw[2]

Stanford University, 353 Jane Stanford Way, Gates Computer Science, Stanford, CA, 94305, United States

Abstract

There are many physical processes that have inherent discontinuities in their mathematical formulations. This paper is motivated by the specific case of collisions between two rigid or deformable bodies and the intrinsic nature of that discontinuity. The impulse response to a collision is discontinuous with the lack of any response when no collision occurs, which causes difficulties for numerical approaches that require differentiability which are typical in machine learning, inverse problems, and control. We theoretically and numerically demonstrate that the derivative of the collision time with respect to the parameters becomes infinite as one approaches the barrier separating colliding from not colliding, and use lifting to complexify the solution space so that solutions on the other side of the barrier are directly attainable as precise values. Subsequently, we mollify the barrier posed by the unbounded derivatives, so that one can tunnel back and forth in a smooth and reliable fashion facilitating the use of standard numerical approaches. Moreover, we illustrate that standard approaches fail in numerous ways mostly due to a lack of understanding of the mathematical nature of the problem (e.g. typical backpropagation utilizes many rules of differentiation, but ignores L'Hopital's rule).

1. Introduction

Recent excitement in deep learning has led to a plethora of interest in utilizing machine learning and data driven techniques for a wide variety of scientific disciplines including computational physics, see e.g. [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. In fact, JCP has dedicated an entire special issue to machine learning methods for physical systems [14]. In this paper, our focus is on the differentiability of one such physical phenomenon (i.e. collisions).

In order to train a neural network, one typically formulates an objective function (i.e. an energy or loss) that is subsequently minimized as a function of various neural network parameters. Many standard approaches to minimization iteratively use the Hessian or approximations to the Hessian or its inverse, often gaining efficiency by utilizing rank one updates of such approximations, see e.g. [15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]. In order to avoid dependence on the existence of second derivatives or robust approximations to them, the Hessian can be crudely approximated with the identity matrix in order to utilize gradient descent methods [26]. Viewing gradient descent as forward Euler discretization of a gradient flow ordinary differential equation has led to a number of adaptive time-step numerical integration approaches including AdaGrad [27], RMSprop [28], and AdaDelta [29]. Moreover, the idea of using previous search directions to escape local minima led to the idea of momentum methods [30] such as Nesterov [31] and Adam [32].

Broad interest in machine and deep learning has led to the development of facilitating software such as PyTorch [33], TensorFlow [34], Torch [35], Caffe [36], Theano [37], Jax [38], etc. These software packages utilize automatic-differentiation [39], specifically backpropagation [40], to compute derivatives (as opposed to computing them analytically, numerically, or symbolically). This is accomplished by combining the derivatives of basic functions (e.g. simple arithmetic, exponential, trigonometric, etc.) based on the various rules of differentiation, e.g. product rule, quotient rule, chain rule, etc. There are (at least) three obvious

[1] dansj@cs.stanford.edu, Corresponding author
[2] fedkiw@cs.stanford.edu

flaws with this approach. Firstly, it is well known that it is quite difficult to write robust code for a number of common physical/mathematical problems at the heart of computational physics, e.g. consider singular, ill-conditioned, and indefinite linear systems, robustness issues for SVD, positivity preservation and cavitation, limiters and TVD, interfaces, etc. Even the simple quadratic formula should be de-rationalized for robustness, see e.g. [41, 42, 43], and Cardano's formula has been found wanting for cubic equations, which are best solved via iterative methods and require double (as opposed to single) precision arithmetic, see e.g. [44]. In fact, some have argued that quadruple precision [45] or exact precision [46] is required for various applications. Considering how much effort has been invested into devising robust numerical methods for various problems, it seems rather unlikely that the additional constraint of making such algorithms differentiable is readily obtainable. Secondly, the aforementioned software does not even do what it claims to do. That is, the basic functions are not actually being differentiated because the software ignores floating point and function approximation errors, e.g. they differentiate $x + y$ instead of the actual $round(x + y)$, and use analytic derivatives of trigonometric, square root, and other functions that computer hardware estimates with various approximation errors. Ignoring potentially adverse effects due to various floating point and function approximation errors seems unwise given the many issues uncovered by numerical analysts over the years. Thirdly, these software packages have simplistic and overly idealistic modularity, which does not properly address issues that arise when disparate chunks of code are combined in various ways. For example, [47] shows how both TensorFlow and PyTorch fail to properly treat the simple functions $x^2 - 4$ and $x - 2$ when the former is divided by the later (aiming for $x + 2$). As is well known, the common remedy of adding a small number ϵ to the denominator (to avoid overflow when dividing by very small numbers) perturbs the result to incorrectly take on values near 0 instead of 4 when x is near 2; unfortunately, both TensorFlow and PyTorch auto-differentiate this code to obtain derivatives with unbounded $O\left(\frac{1}{\epsilon}\right)$ errors.

The computational physics community has long embraced discontinuities via both mathematical formulations and numerical algorithms. Consider, for example, the use of the weak (integral) form for conservation laws in order to correctly model shocks and detonations, which do not possess the differentiability required for existence of the strong form, see e.g. [48, 49]. This has led to a variety of numerical approaches with discontinuous decision making, such as ENO [50]. Also consider, for example, sharp interface methods for contact discontinuities and material interfaces when simulating compressible flows, incompressible flows, solids undergoing fracture, etc., see e.g. level set methods [51], ghost fluids methods [52, 53, 54], immersed interface methods [55], XFEM [56], etc. On the other hand, while respecting nondifferentiability and the need to correctly treat discontinuities, computational physics researchers have aimed for smoothness when desirable and/or appropriate. For example, implicit time integration and steady-state/quasistatic approaches to nonlinear hyperbolic partial differential equations typically lead to nonlinear system solves that require smoothness as well as direct or indirect access to various derivatives; thus, smooth or smoothed approximations to the governing equations are highly beneficial and often sought, see e.g. [57, 58, 59, 60, 61, 62, 63, 64, 65]. This motivated, for example, formulating WENO [66, 67] as a convex combination of the three ENO choices; however, practitioners have struggled to get the smoother WENO scheme to work well enough near shocks and detonations, and a popular approach is to use ENO near discontinuities and WENO in smoother regions of the flow, see e.g. [68] and the references therein. Unfortunately, this careful consideration of the potentially adverse effects caused by smoothing is mostly absent from the software and proposed approaches of the computer science community, which typically naively assume that one can indiscriminantly smooth mathematical formulations of physical processes and/or differentiate code instead of equations, see e.g. [69]. Notable exceptions include: [70] where the authors explain how roundoff errors can cause backpropagation to fail when considering neural ordinary differential equations, [71] which discusses how stiffness and discontinuities can compromise the efficacy of differentiable simulators, and [72] which discusses when differentiation based optimization algorithms struggle due to problems with the Jacobian.

In spite of the aforementioned issues, the machine and deep learning community still obtains rather impressive results on a variety of problems. This is likely because the direct penalization of deviations from the training data in their objective function helps to overcome other flaws in their approach. Particularly questionable is their justification for dropout [73], which claims that randomly setting derivatives to zero while training a neural network is somehow equivalent to averaging various good models together. Although model averaging is certainly justified when a number of good models exist, there is no justification that randomly dropping the subsets of a model that vanish when terminating dependencies by arbitrarily setting derivatives to zero leads to viable models. A better justification would be that that dropout coaxes the neural network to match the training data even when the derivatives are so poorly approximated that they are randomly set to zero; this calls into question the entire paradigm of using differentiability, backpropagation, and optimization for training neural networks. If one were to believe the theoretical justifications regarding differentiability, then at the very least one would ascertain error bounds on the derivatives and randomly perturb derivative estimates within those bounds instead of randomly setting derivatives to zero; additionally, derivatives that are nonexistent or that blow up towards infinite values should be more carefully addressed as well, rather than randomly setting them to zero and hoping for the best.

This paper was motivated by examining collisions between rigid and deformable bodies where the mathematical formulations and numerical methods are both known to be problematic due to differential inclusions [74, 75] and issues with accurately solving cubic equations [44, 76, 42]. The concept of whether or not a collision occurs (i.e. collision detection) is intrinsically discontinuous, and we show that this manifests itself as a derivative blowing up towards infinity as one approaches the decision boundary between colliding or not colliding. Any robust root-finding approach will require discontinuous (and thus non-differentiable) decision making, e.g. consider the hybridization of Newton's method with bisection. Since both the equations and the numerical methods are not readily differentiable, backpropagating through the iterative solver seems unwise; instead, we follow an approach similar to [12, 77] of differentiating the equations (this is typically referred to as an implicit layer, see e.g. [78] and the references therein). This is akin to using the pseudoinverse A^+ to represent the Jacobian $\frac{\partial x}{\partial b}$ from $x = A^+b$ instead of backpropagating through whatever algorithm was used to solve $Ax = b$. For collision detection, this amounts to implicit differentiation of a cubic equation (note, [79, 80] took a similar, albeit incorrect, approach[3]). Although implicit differentiation allows the derivatives to be obtained while still utilizing a state-of-the-art iterative solver with non-differentiable decision making intact, it does not address derivatives blowing up towards infinity. Unfortunately, as was pointed out in [81], the machine learning community has mostly ignored the conditions required to validate the use of the implicit function theorem. To properly address this for collision detection, we lift the solution space to $\mathbb{R}^2$ in order to allow the iterative solver to more readily work its way back and forth between real-valued roots representing collisions and complex-valued roots representing the absence of collisions; then, we mollify the barrier posed by the unbounded derivative in a fashion that does not pollute the accuracy or attainability of solutions when they exist. Notably, this is enabled by devising a new canonical form for cubic equations (perhaps not previously appearing in the literature).

2. Preliminaries

Collision detection and response is important to a wide variety of material modeling problems, where the materials may be approximated as multibody systems with each body either stiff enough to be treated as a rigid body or instead simulated with a deformable finite element approximation. The surface of each body can be discretized into a set of triangles; then, a collision occurs when either a vertex from one surface impacts a triangle of the other or when two edges collide. In both cases, a collision occurs when four points become coplanar. When the objects are close enough together, one can linearize the motion of these points via $\vec{x}_i(t) = \vec{x}_i^o + \vec{v}_i t$ where $\vec{x}_i(t)$ is the position of point i at time t, $\vec{x}_i^o$ is the position before linearization, and $\vec{v}_i$ is the linearized velocity. Choosing $\vec{x}_1(t)$ as a frame of reference, the three edge vectors $\vec{x}_{21}(t) = \vec{x}_2(t) - \vec{x}_1(t)$, $\vec{x}_{31}(t) = \vec{x}_3(t) - \vec{x}_1(t)$, and $\vec{x}_{41}(t) = \vec{x}_4(t) - \vec{x}_1(t)$ describe a would-be tetrahedron (similar to Green strain [82]) where coplanarity is equivalent to the tetrahedron having zero volume, e.g. $\vec{x}_{21}(t) \times \vec{x}_{31}(t) \cdot \vec{x}_{41}(t) = 0$. This results in a cubic equation for t, which (as discussed in [44]) requires a carefully designed/implemented iterative solver using double precision in order to guarantee that potential collisions are not missed. Given a time t of coplanarity, the positions of the points are examined to determine if either the point is inside the triangle or the two edges overlap (depending on which case is being considered). If a collision occurs, the positions and velocities of bodies are used to determine a collision response.

When the object of interest is deformable, the particles are true degrees of freedom. When the body is rigid, the degrees of freedom are its center-of-mass translational and rotational velocity. In both cases, we refer to the degrees of freedom as $\vec{\lambda}$ for the sake of exposition. A typical goal might be to obtain a specific

[3] [79] published an incorrect derivative; meanwhile, that derivative is set to be identically zero in their code (see https://github.com/williamljb/DifferentiableCloth). The code for [80] (see https://github.com/YilingQiao/diffsim) does not set the derivative to zero, but still uses the incorrect formula from [79].

post-collision velocity by somehow modifying $\vec{\lambda}$ via controllable degrees of freedom. Although this seems feasible when considering collision response, changes in $\vec{\lambda}$ have no effect on post-collision velocities when there is no collision. In a real-world scenario, the prospect of missing a collision would inevitably motivate a change in strategy; instead of focusing on the final trajectory of the particle, one might turn their attention towards aiming to create a collision. Mathematically, missing the collision is equivalent to the desired root of a cubic equation being complex-valued instead of real-valued, and aiming to create a collision is equivalent to aiming to change the complex-valued root into a real-valued root. Although the proposition of collision or no collision at first appears binary and unavoidably non-differentiable, the real-world scenario would seem to indicate that the collision response perhaps could be differentiably connected to collision detection by formally lifting the root-finding problem to consider both real and complex roots.

Let $t_{\text{root}}(\vec{\lambda})$ represent a solution/root to the aforementioned cubic equation, and consider minimizing an objective function

$$L(\vec{\lambda}) = \frac{1}{2}\left(\sum_{\text{roots}} ||t_{\text{root}}(\vec{\lambda}) - t_{\text{root,L}}||_2^2 + \left|\left|\vec{p}(\vec{\lambda}) - \vec{p}_L\right|\right|_2^2\right) \tag{1}$$

where each root of interest $t_{\text{root}}(\vec{\lambda})$ may have an aspirational target value $t_{\text{root,L}}$ and the (column vector) coefficients of the cubic $\vec{p}(\vec{\lambda})$ may be regularized towards some $\vec{p}_L$. For optimization, one would utilize the gradient

$$\nabla_{\vec{\lambda}} L = \frac{\partial \vec{p}}{\partial \vec{\lambda}}^T \left(\sum_{\text{roots}} \frac{\partial t_{\text{root}}}{\partial \vec{p}}^T (t_{\text{root}}(\vec{\lambda}) - t_{\text{root,L}}) + \vec{p}(\vec{\lambda}) - \vec{p}_L\right) \tag{2}$$

where the ∂ represents the typical Jacobian and thus the gradients (being the transpose) reverse their order. Here, t_{root} is a column vector of separate real and imaginary parts, and $\frac{\partial t_{\text{root}}}{\partial \vec{p}}$ has two rows (one for the real part and one for imaginary part).

In order to aid both the analysis and the numerics for $\frac{\partial t_{\text{root}}}{\partial \vec{p}}$, we guarantee that the maximum magnitude of any entry in $\vec{p}$ is bounded above by 1 simply by dividing by the largest entry $p_{\max}$ if it has magnitude larger than 1. Generally speaking, this division can be problematic (perhaps requiring asymptotic analysis) when entries of $\vec{p}$ are blowing up; however, since $\vec{p}$ is a function of $\vec{\lambda}$, this is problem specific and we leave it to the reader. Going forward, we will treat $\vec{p}$ as if it were bounded when considering $\frac{\partial t_{\text{root}}}{\partial \vec{p}}$, and use $\vec{p}_{\text{orig}}$ to represent the original not necessarily bounded polynomial coefficients. Since $t_{\text{root}}(\vec{p}_{\text{orig}}) = t_{\text{root}}(\vec{p})$, one can simply treat t_{root} as a function of the bounded parameters instead. For the derivatives, one needs to replace $\frac{\partial t_{\text{root}}}{\partial \vec{p}_{\text{orig}}}$ with $\frac{\partial t_{\text{root}}}{\partial \vec{p}} \frac{\partial \vec{p}}{\partial \vec{p}_{\text{orig}}}$ where

$$\frac{\partial \vec{p}}{\partial \vec{p}_{\text{orig}}} = \frac{1}{p_{\max}^2}\left(p_{\max} I - \vec{p}_{\text{orig}} \hat{e}_k^T\right) \tag{3}$$

and $\hat{e}_k$ is a standard unit basis vector (where k is the index of $p_{\max}$ in p_{orig}). This leads to replacing $\frac{\partial \vec{p}}{\partial \vec{\lambda}}^T$ with $\frac{\partial \vec{p}_{\text{orig}}}{\partial \vec{\lambda}}^T \frac{\partial \vec{p}}{\partial \vec{p}_{\text{orig}}}^T$ in equation 2. We stress that we still allow for the coefficients of the cubic (i.e. $\vec{p}_{\text{orig}}$) to grow rather large, but believe that it is more tidy to address this with equation 3 and a bounded $\vec{p}$ in $\frac{\partial t_{\text{root}}}{\partial \vec{p}}$ than worrying about the case where $\vec{p}_{\text{orig}}$ is large in $\frac{\partial t_{\text{root}}}{\partial \vec{p}_{\text{orig}}}$.

3. Quadratic Equations

We motivate our approach by restricting the cubic equation $qt^3 + at^2 + bt + c = 0$ to the simpler $q = 0$ quadratic equation case. The roots of the quadratic equation are

$$t_{\text{root}}^{\pm} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{4}$$

which are real-valued when $b^2 - 4ac \geq 0$ (repeated when $b^2 - 4ac = 0$) and complex conjugates when $b^2 - 4ac < 0$. As discussed in Section 2, we guarantee that the polynomial coefficients have magnitudes less than or equal to 1; thus, the only numerically problematic case occurs when a is small. As $a \to 0$, $\frac{-b}{2a}$ can be any real number depending on the behavior of b which may also go to zero and do so at speeds faster than, slower than, or commensurate with a; thus, repeated roots and the real part of complex roots may take on any value (perhaps even being unbounded). The imaginary parts of complex roots have equal and opposite sign and may bounded or unbounded. In the case of two distinct real roots, they may appear anywhere, i.e. both bounded, one bounded and the other unbounded, or both unbounded (with the same or opposite signs).

Remark 3.1: For the sake of a practical implementation, one needs to establish the largest magnitude number $t_{\max}$ that can be used to represent a root (on a computer of interest). We refer to roots outside $(-t_{\max}, t_{\max})$ as either unbounded, approaching $\pm\infty$, blowing up, etc. as appropriate. Notably, our approach fully handles this case, including the ability to drive these unbounded roots to smaller obtainable values when desired.

Letting the independent variable be $t_{\text{root}} = [t^R_{\text{root}}, t^I_{\text{root}}]^T$, one can write

$$\frac{\partial t^R_{\text{root}}}{\partial \vec{p}} = \begin{bmatrix} \frac{b\mp\sqrt{b^2-4ac}}{2a^2} \mp \frac{c}{a\sqrt{b^2-4ac}} & \frac{-1}{2a} \pm \frac{b}{2a\sqrt{b^2-4ac}} & \frac{\mp 1}{\sqrt{b^2-4ac}} \end{bmatrix} \tag{5a}$$

$$= \frac{-1}{\pm\sqrt{b^2-4ac}} \begin{bmatrix} \left(\frac{-b\pm\sqrt{b^2-4ac}}{2a}\right)^2 & \frac{-b\pm\sqrt{b^2-4ac}}{2a} & 1 \end{bmatrix} \tag{5b}$$

$$= \frac{-1}{\pm 2\sqrt{-\tilde{c}}} \begin{bmatrix} \left(t^R_{\text{root}}\right)^2 & \left(t^R_{\text{root}}\right)^1 & \left(t^R_{\text{root}}\right)^0 \end{bmatrix} \tag{5c}$$

when the roots are real (i.e. with $t^I = 0$); here, $\tilde{c} = -\frac{1}{4}(b^2 - 4ac) \leq 0$. Equation 5c elucidates a trivial and valid strategy covering the case when a root blows up; as $t^R_{\text{root}} \to \pm\infty$, the direction of $\frac{\partial t^R_{\text{root}}}{\partial \vec{p}}$ approaches $[\mp 1\ 0\ 0]$. This indicates the need to fix a, which makes sense since this degeneracy is caused by $a \to 0$. The magnitude of $\frac{\partial t^R_{\text{root}}}{\partial \vec{p}}$ will also approach infinity; however, it can be clamped without changing the direction to some maximum allowable value that makes sense (for optimization). The magnitude can be similarly clamped when t^R_{root} is bounded and $\tilde{c} \to 0$, indicating the merging of two real roots into a repeated root.

A rather interesting case occurs as both $\tilde{c} \to 0$ and $t^R_{\text{root}} \to 0$, i.e. the roots are merging towards $t^R_{\text{root}} = 0$. Although the aforementioned strategy robustly treats this via $\frac{\partial t^R_{\text{root}}}{\partial \vec{p}}$ having direction $[0\ 0\ 1]$ with a clamped magnitude, some form of L'Hospital's rule is required when one is interested in obtaining values for $\frac{\partial t^R_{\text{root}}}{\partial a}$ and $\frac{\partial t^R_{\text{root}}}{\partial b}$. Writing $b^2 - 4ac \to \gamma b^2$ treats the case when b^2 dominates $4ac$ by $\gamma = 1$, the case when $4ac$ dominates b^2 by $\gamma \to \infty$, and co-dominance by $\gamma \in [0, 1) \cup (1, \infty)$ with $[0, 1)$ when $4ac > 0$ and $(1, \infty)$ when $4ac < 0$. Substituting $b^2 - 4ac \to \gamma b^2$ into equation 4 gives

$$t^R_{\text{root}} \to \frac{-b}{2a}(1 \mp \text{Sign}(b)\sqrt{\gamma}) \tag{6}$$

where Sign(b) is the sign of b. Substituting equation 6 into equation 5c gives

$$\frac{\partial t^R_{\text{root}}}{\partial \vec{p}} \to \begin{bmatrix} \frac{1}{4a^2}\left(\frac{\mp|b|}{\sqrt{\gamma}} + 2b \mp |b|\sqrt{\gamma}\right) & \frac{-1}{2a}\left(\frac{\mp 1}{\text{Sign}(b)\sqrt{\gamma}} + 1\right) & \frac{\mp 1}{|b|\sqrt{\gamma}} \end{bmatrix} \tag{7}$$

where $\gamma b^2 \to 0$ (and thus $b\sqrt{\gamma} \to 0$) implies that $\frac{\partial t^R_{\text{root}}}{\partial c}$ always blows up. $\frac{\partial t^R_{\text{root}}}{\partial b}$ is indeterminate, ranging from $\frac{-1}{2a}$ as $\gamma \to \infty$ to various finite values for finite γ to blowing up as $\gamma \to 0$. The second two terms in $\frac{\partial t^R_{\text{root}}}{\partial a}$ vanish. When $\gamma \neq 0$, the first term in $\frac{\partial t^R_{\text{root}}}{\partial a}$ also goes to zero implying that $\frac{\partial t^R_{\text{root}}}{\partial a} \to 0$. When $\gamma = 0$, b^2 and $4ac$ co-dominate to cancel the $O(b^2)$ terms in $b^2 - 4ac$; however, even though γ does not contain any $O(1)$ terms, it can still contain powers of b. When $\sqrt{\gamma} \to 0$ slower than $b \to 0$, $\frac{\partial t^R_{\text{root}}}{\partial a} \to 0$. When $\sqrt{\gamma} \to 0$ at the same speed as $b \to 0$, $\frac{\partial t^R_{\text{root}}}{\partial a}$ is finite. When $\sqrt{\gamma} \to 0$ faster than $b \to 0$, $\frac{\partial t^R_{\text{root}}}{\partial a}$ blows up. Thus, $\frac{\partial t^R_{\text{root}}}{\partial a}$ is indeterminate. Although one might attempt to remove the set of measure zero sequences where $\gamma \to 0$ by setting $\frac{\partial t^R_{\text{root}}}{\partial a} \to 0$ in all cases, no similar strategy works for $\frac{\partial t^R_{\text{root}}}{\partial b}$.

Remark 3.2: Numerically, these asymptotics will manifest themselves via infinitesimal values for the parameters generated pseudo-randomly because of limited numerical precision, e.g. $-4ac >> b^2$ is $\gamma \to \infty$, $b^2 >> -4ac$ is $\gamma \to 1$, and $b^2 - 4ac << b^2$ is $\gamma \to 0$. The obvious difficulty is that roundoff errors and the representability of small numbers are difficult to predict and control.

Extending equation 5c to include the identically zero imaginary part of the real roots gives

$$\frac{\partial t_{\text{root}}}{\partial \vec{p}} = \frac{-1}{\pm 2\sqrt{-\tilde{c}}} \begin{bmatrix} \left(t^R_{\text{root}}\right)^2 & \left(t^R_{\text{root}}\right)^1 & \left(t^R_{\text{root}}\right)^0 \\ 0 & 0 & 0 \end{bmatrix} \tag{8a}$$

$$= \frac{-1}{\pm 2\sqrt{-\tilde{c}}} \begin{bmatrix} Re(z^2_{\text{root}}) & Re(z^1_{\text{root}}) & Re(z^0_{\text{root}}) \\ Im(z^2_{\text{root}}) & Im(z^1_{\text{root}}) & Im(z^0_{\text{root}}) \end{bmatrix} \tag{8b}$$

where $z_{\text{root}} = t^R_{\text{root}} + it^I_{\text{root}}$, and Re and Im are the real and imaginary parts respectively. Equations 5-8 and the related discussion are only valid in the halfspace where both roots are real. In the halfspace where the roots are complex,

$$\frac{\partial t_{\text{root}}}{\partial \vec{p}} = \begin{bmatrix} \frac{b}{2a^2} & \frac{-1}{2a} & 0 \\ \frac{\mp\sqrt{-(b^2-4ac)}}{2a^2} \pm \frac{c}{a\sqrt{-(b^2-4ac)}} & \mp\frac{b}{2a\sqrt{-(b^2-4ac)}} & \frac{\pm 1}{\sqrt{-(b^2-4ac)}} \end{bmatrix} \tag{9a}$$

$$= \frac{-1}{\pm 2\sqrt{\tilde{c}}} \begin{bmatrix} 2t^R_{\text{root}}t^I_{\text{root}} & t^I_{\text{root}} & 0 \\ -\left((t^R_{\text{root}})^2 - (t^I_{\text{root}})^2\right) & -t^R_{\text{root}} & -1 \end{bmatrix} \tag{9b}$$

$$= \frac{-1}{\pm 2\sqrt{\tilde{c}}} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} Re(z^2_{\text{root}}) & Re(z^1_{\text{root}}) & Re(z^0_{\text{root}}) \\ Im(z^2_{\text{root}}) & Im(z^1_{\text{root}}) & Im(z^0_{\text{root}}) \end{bmatrix} \tag{9c}$$

instead. Noting that $\pm 2\sqrt{\tilde{c}} = \pm\sqrt{-(b^2 - 4ac)} = 2at^I_{\text{root}}$, the real part of equation 9b has direction $[2t^R_{\text{root}}\ 1\ 0]$ indicating the need to fix a when t^R_{root} is big. If either t^R_{root} or t^I_{root} is large, but not both, the imaginary part of equation 9b also indicates fixing a; however, when t^R_{root} and t^I_{root} are both large, they may cancel making the second entry (i.e. modify b) dominate.

Remark 3.3: Strategically, modifying b may be unwise in this case because it leads to real-valued roots whereas fixing a can leave the roots complex-valued. Moreover, one needs to fix a anyways (because t^R_{root} is large) as indicated by the first row in equation 9b.

Next, we revisit $\tilde{c} \to 0$ and $t^R_{\text{root}} \to 0$, i.e. the roots merging to $t^R_{\text{root}} = 0$ but from the complex halfspace this time. Here, $\gamma \in (-\infty, 0]$ with $4ac$ dominating b^2 given by $\gamma \to -\infty$; in addition, $\gamma \to 0$ from the $4ac > b^2$ side. Substituting $b^2 - 4ac \to \gamma b^2$ into equation 4 gives

$$\begin{bmatrix} t^R_{\text{root}} \\ t^I_{\text{root}} \end{bmatrix} \to \frac{-b}{2a} \begin{bmatrix} 1 \\ \mp\text{Sign}(b)\sqrt{-\gamma} \end{bmatrix} \tag{10}$$

which when substituted into equation 9b gives

$$\frac{\partial t_{\text{root}}}{\partial \vec{p}} \to \begin{bmatrix} \frac{b}{2a^2} & \frac{-1}{2a} & 0 \\ \frac{1}{4a^2}\left(\frac{\pm|b|}{\sqrt{-\gamma}} \mp |b|\sqrt{-\gamma}\right) & \frac{-1}{2a}\left(\frac{\pm 1}{\text{Sign}(b)\sqrt{-\gamma}}\right) & \frac{\pm 1}{|b|\sqrt{-\gamma}} \end{bmatrix} \tag{11}$$

where $\frac{\partial t^I_{\text{root}}}{\partial c}$ blows up, and both $\frac{\partial t^I_{\text{root}}}{\partial b}$ and $\frac{\partial t^I_{\text{root}}}{\partial a}$ are indeterminate (as in equation 7).

3.1. Implicit Differentiation

For the sake of exposition, we write the quadratic equation as

$$f(t_{\text{root}}; \vec{p}) = \begin{bmatrix} a(t^R_{\text{root}})^2 - a(t^I_{\text{root}})^2 + bt^R_{\text{root}} + c \\ (2at^R_{\text{root}} + b)t^I_{\text{root}} \end{bmatrix} = \vec{0} \tag{12}$$

letting θ_1 refer to the first variable (i.e. t_{root}) and θ_2 refer to the second variable (i.e. $\vec{p}$) so that the derivatives

$$f_{\theta_1}(t_{\text{root}}; \vec{p}) = \begin{bmatrix} 2at^R_{\text{root}} + b & -2at^I_{\text{root}} \\ 2at^I_{\text{root}} & 2at^R_{\text{root}} + b \end{bmatrix} \tag{13a}$$

$$f^{-1}_{\theta_1}(t_{\text{root}}; \vec{p}) = \frac{1}{s} \begin{bmatrix} 2at^R_{\text{root}} + b & 2at^I_{\text{root}} \\ -2at^I_{\text{root}} & 2at^R_{\text{root}} + b \end{bmatrix} \text{ where } s = (2at^R_{\text{root}} + b)^2 + (2at^I_{\text{root}})^2 \tag{13b}$$

$$f_{\theta_2}(t_{\text{root}}; \vec{p}) = \begin{bmatrix} Re(z^2_{\text{root}}) & Re(z^1_{\text{root}}) & Re(z^0_{\text{root}}) \\ Im(z^2_{\text{root}}) & Im(z^1_{\text{root}}) & Im(z^0_{\text{root}}) \end{bmatrix} \tag{13c}$$

have compact notation. The total derivative of equation 12 is $f_{\theta_1}(t_{\text{root}}, \vec{p})d\theta_1 + f_{\theta_2}(t_{\text{root}}, \vec{p})d\theta_2 = 0$, which can be written as $d\theta_1 = -f_{\theta_1}^{-1}(t_{\text{root}}, \vec{p})f_{\theta_2}(t_{\text{root}}, \vec{p})d\theta_2$ or

$$dt_{\text{root}} = -\frac{1}{s}\begin{bmatrix} 2at^R_{\text{root}} + b & 2at^I_{\text{root}} \\ -2at^I_{\text{root}} & 2at^R_{\text{root}} + b \end{bmatrix}\begin{bmatrix} Re(z^2_{\text{root}}) & Re(z^1_{\text{root}}) & Re(z^0_{\text{root}}) \\ Im(z^2_{\text{root}}) & Im(z^1_{\text{root}}) & Im(z^0_{\text{root}}) \end{bmatrix} d\vec{p} \tag{14}$$

when f_{θ_1} is invertible. In the halfspace where the roots are real, $t^I_{\text{root}} = 0$ and $2at^R_{\text{root}} + b = \pm 2\sqrt{-\tilde{c}}$ showing the equivalence between equations 14 and 8b. In the complex halfspace, $2at^R_{\text{root}} + b = 0$ and $2at^I_{\text{root}} = \pm 2\sqrt{\tilde{c}}$ showing the equivalence between equations 14 and 9c. Interestingly, even though f_{θ_1} is not invertible when $s = 0$ (the repeated roots case) and thus equation 14 is not formally derivable via the total derivative and the implicit function theorem, equation 14 does match equations 8b and 9c.

Remark 3.4: Our treatment of t_{root} as a column vector with separate real and imaginary parts means that the multiplication of two complex numbers z_1 and z_2 is

$$z_1 z_2 = \begin{bmatrix} Re(z_1) & -Im(z_1) \\ Im(z_1) & Re(z_1) \end{bmatrix}\begin{bmatrix} Re(z_2) \\ Im(z_2) \end{bmatrix} \tag{15}$$

allowing equation 14 to be rewritten as

$$dz_{root} = \frac{-1}{2az_{root} + b}\begin{bmatrix} z^2_{root} & z^1_{root} & z^0_{root} \end{bmatrix} d\vec{p} \tag{16}$$

since

$$\frac{1}{2az_{root} + b} = \frac{1}{2a(t^R_{root} + it^I_{root}) + b} = \frac{1}{s}\left(2at^R_{root} + b - i2at^I_{root}\right). \tag{17}$$

4. Newton's Method

When the roots are real, equation 4 is typically evaluated using de-rationalization

$$t^+_{\text{root}} = \begin{cases} \frac{-b+\sqrt{b^2-4ac}}{2a} & \text{if } b \le 0 \\ \frac{2c}{-b-\sqrt{b^2-4ac}} & \text{if } b > 0 \end{cases} \tag{18a}$$

$$t^-_{\text{root}} = \begin{cases} \frac{-b-\sqrt{b^2-4ac}}{2a} & \text{if } b \ge 0 \\ \frac{2c}{-b+\sqrt{b^2-4ac}} & \text{if } b < 0 \end{cases} \tag{18b}$$

to avoid catastrophic cancellation (see e.g. [41, 42, 43]). Prior works on cubic equations (e.g. [44]) found that Cardano's explicit formula lacked the accuracy required for collision detection and instead used iterative methods; moreover, [44] stressed that double precision (rather than single precision) was required in order to detect collisions accurately enough for their cloth simulations. Since our consideration of the quadratic equation is merely a building block for the cubic equation, we utilize Newton's method. Note that it is important to use $t_{\text{root}}(\vec{p})$ instead of $t_{\text{root}}(\vec{p}_{\text{orig}})$, as discussed in Section 2, in order to avoid numerical issues with convergence (and detecting convergence) when using Newton's method; otherwise, we have observed Newton's method struggling to converge to prescribed tolerances due to cancellation issues. This makes sense since equation 12 is linear in $\vec{p}$.

Newton's method for computing t_n recursively from t_{n-1} is

$$t_n = t_{n-1} - f_{\theta_1}^{-1}(t_{n-1}; \vec{p})f(t_{n-1}; \vec{p}) \tag{19a}$$

$$= t_{n-1} - \frac{1}{s}\begin{bmatrix} 2at^R_{n-1} + b & 2at^I_{n-1} \\ -2at^I_{n-1} & 2at^R_{n-1} + b \end{bmatrix}\begin{bmatrix} a(t^R_{n-1})^2 - a(t^I_{n-1})^2 + bt^R_{n-1} + c \\ (2at^R_{n-1} + b)t^I_{n-1} \end{bmatrix} \tag{19b}$$

where $s = (2at^R_{n-1} + b)^2 + (2at^I_{n-1})^2$ and the iteration proceeds until $f(t_n, \vec{p})$ is small enough. When $a = 0$ and $b \neq 0$, equation 19b becomes

$$t_n = t_{n-1} - \frac{1}{b}\begin{bmatrix} bt^R_{n-1} + c \\ bt^I_{n-1} \end{bmatrix} \tag{20}$$

which converges to $t^R_{\text{root}} = \frac{-c}{b}$ in one iteration. When $a = b = 0$, $f(t;\vec{p}) = c$ is a constant function where either nothing ($c \neq 0$) or everything ($c = 0$) is a root. When $a \neq 0$, equation 19b can be rewritten as

$$t_n = \begin{bmatrix} -\frac{b}{2a} \\ 0 \end{bmatrix} + \frac{1}{4a} \begin{bmatrix} \left(1 - \frac{4\tilde{c}}{s}\right)\left(2at^R_{n-1} + b\right) \\ \left(1 + \frac{4\tilde{c}}{s}\right)\left(2at^I_{n-1}\right) \end{bmatrix} \tag{21}$$

using only algebraic manipulation. In the case of repeated roots, $\tilde{c} = 0$ and the convergence is linear (with the distance between (t^R_n, t^I_n) and $(\frac{-b}{2a}, 0)$ cut in half every iteration) instead of quadratic. When $\tilde{c} \approx 0$ (but $\tilde{c} \neq 0$) with roots close to $(\frac{-b}{2a}, 0)$, converging iterates give values of s close to (perhaps even identically equal to) zero making $\frac{\tilde{c}}{s}$ problematic. Although $2at^R_{n-1} + b$ and $2at^I_{n-1}$ shrink like $\sqrt{s}$ and thus help to eliminate a vanishing s, $\frac{\tilde{c}}{\sqrt{s}}$ is still problematic. One remedy would be to set $\tilde{c} = 0$ (and eliminate $\frac{\tilde{c}}{s}$ from the computation) forcing a repeated root even when the roots are distinct, arguing that this is only a small perturbation of the distinct roots.

For the sake of exposition, let $\tilde{c} \leq 0$ so that only real-valued roots are relevant; then equation 19b can be written as

$$t^R_n = t^R_{n-1} - \frac{a(t^R_{n-1})^2 + bt^R_{n-1} + c + \delta}{2at^R_{n-1} + b + \epsilon} \tag{22}$$

where ϵ is used to avoid division by zero, and δ collects (all) the roundoff errors from computing the numerator and denominator, dividing, and subtracting from t^R_{n-1}. Without loss of generality, equation 22 can be rewritten as

$$\bar{t}^R_n = \frac{a(\bar{t}^R_{n-1})^2 + \epsilon\bar{t}^R_{n-1} - \frac{\tilde{c}}{a} - \delta}{2a\bar{t}^R_{n-1} + \epsilon} \tag{23}$$

via the change of variables $\bar{t}^R = t^R + \frac{b}{2a}$. Assume $a > 0$ and $\bar{t}^R_{n-1} > 0$, so that one would expect to converge to the positive root $\bar{t}^R_{\text{root}} \geq 0$. Assume that $\epsilon > 0$ is chosen (properly) to match the sign of $2a\bar{t}^R_{n-1}$. Ignoring δ, both the numerator and denominator remain strictly positive and convergence to the correct $\bar{t}^R_{\text{root}} \geq 0$ is guaranteed. On the other hand, the roundoff error δ can change the sign of the numerator when $\bar{t}^R_{n-1}$ and $\tilde{c}$ are small, which occurs when the roots are close together. This emphasizes the need to hybridize Newton's method with bisection in order to guarantee convergence to a desired root.

Remark 4.1: It is worth briefly discussing the addition of $\epsilon > 0$ to s in equation 19b (and thus equations 22 and 23). Newton's method discretizes $df(t_{n-1};\vec{p}) = f_{\theta_1}(t_{n-1};\vec{p})dt$ with $df \approx 0 - f(t_{n-1};\vec{p})$ and $dt \approx \Delta t_n = t_n - t_{n-1}$ leading to $f_{\theta_1}(t_{n-1};\vec{p})\Delta t_n = -f(t_{n-1};\vec{p})$. Then, the normal equations $f^T_{\theta_1}(t_{n-1};\vec{p})f_{\theta_1}(t_{n-1};\vec{p})\Delta t_n = -f^T_{\theta_1}(t_{n-1};\vec{p})f(t_{n-1};\vec{p})$ reduce to $s\Delta t_n = -sf^{-1}_{\theta_1}(t_{n-1};\vec{p})f(t_{n-1};\vec{p})$ since $f^T_{\theta_1} = sf^{-1}_{\theta_1}$. Dividing both sides by s gives equation 19a as expected. Instead, a Levenberg-Marquardt approach would modify the coefficient matrix to $f^T_{\theta_1}f_{\theta_1} + \epsilon I = (s + \epsilon)I$ illustrating that modifying s to $s + \epsilon$ in equation 19b to ad hoc remove division by zero is formally equivalent to Levenberg-Marquardt.

In equation 19b, $t^I_{n-1} = 0$ implies $t^I_n = 0$; in other words, iterates on the real axis are stuck on the real axis. In addition, $t^R_{n-1} = -\frac{b}{2a}$ implies $t^R_n = -\frac{b}{2a}$; in other words, such iterates can only obtain complex or repeated roots. Equation 21 illustrates that t^R_n becomes $\frac{-b}{2a}$ (and stays there) when $s = 4\tilde{c}$, which implies $\tilde{c} \geq 0$ and thus complex or repeated roots; similarly, t^I_n becomes zero (and stays there) when $s = -4\tilde{c}$, which implies $\tilde{c} \leq 0$ and thus real or repeated roots. In both cases, this only happens when it should; however, numerical errors, initial guesses, etc. may lead to $s = 4\tilde{c}$ or $s = -4\tilde{c}$ erroneously.

Remark 4.2: The need to hybridize Newton's method with non-differentiable bisection in order to guarantee convergence to the desired root in the face of roundoff errors highlights the folly of aiming to make iterative solvers differentiable for the sake of backpropagation; moreover, the perturbation required in order to avoid spuriously getting stuck on the $t^I = 0$ and $t^R = \frac{-b}{2a}$ lines makes devising a differentiable iterative solver even more unlikely.

Remark 4.3: Embracing a bit of non-differentiable decision-making allows for a straightforward approach. When $\tilde{c} < 0$, both $t^R \in (-\infty, \frac{-b}{2a})$ and $t^R \in (\frac{-b}{2a}, \infty)$ are safe intervals for hybridizing Newton's method with bisection. When $\tilde{c} > 0$, $t^I \in (-\infty, 0)$ and $t^I \in (0, \infty)$ are the safe intervals (with $t^R = \frac{-b}{2a}$).

5. Bisection

Considering only real roots, a typical implementation of bisection (including derivative information) would proceed as follows. Given endpoints E_0^{left} and E_0^{right}, the midpoint

$$t_0 = \frac{E_0^{\text{left}} + E_0^{\text{right}}}{2} \tag{24a}$$

$$\frac{\partial t_0}{\partial \vec{p}} = \frac{1}{2}\left(\frac{\partial E_0^{\text{left}}}{\partial \vec{p}} + \frac{\partial E_0^{\text{right}}}{\partial \vec{p}}\right) \tag{24b}$$

is chosen as the initial guess for the root. Subsequently, the sign of $f(t_0)$ is used to branch the code. Software infrastructures such as PyTorch [33] and TensorFlow [34] compute derivatives by building a computational graph of dependencies, allowing one to ascertain the change in an output parameter with respect to an input parameter. The computational graph only contains branches that are taken, essentially adding a non-differentiable Heaviside function at every code branch. Assuming $f(E_1^{\text{left}}) < 0$ and $f(E_1^{\text{right}}) > 0$ leads to

$$E_1^{\text{left}} = H(f(t_0))E_0^{\text{left}} + (1 - H(f(t_0)))t_0 \tag{25a}$$

$$\frac{\partial E_1^{\text{left}}}{\partial \vec{p}} = \left(\frac{1}{2} + \frac{H(f(t_0))}{2}\right)\frac{\partial E_0^{\text{left}}}{\partial \vec{p}} + \left(\frac{1}{2} - \frac{H(f(t_0))}{2}\right)\frac{\partial E_0^{\text{right}}}{\partial \vec{p}} \tag{25b}$$

$$E_1^{\text{right}} = H(f(t_0))t_0 + (1 - H(f(t_0)))E_0^{\text{right}} \tag{25c}$$

$$\frac{\partial E_1^{\text{right}}}{\partial \vec{p}} = \frac{H(f(t_0))}{2}\frac{\partial E_0^{\text{left}}}{\partial \vec{p}} + \left(1 - \frac{H(f(t_0))}{2}\right)\frac{\partial E_0^{\text{right}}}{\partial \vec{p}} \tag{25d}$$

where equation 24b was used to simplify equations equations 25b and 25d, and $H(f(t_0))$ is a piecewise constant Heaviside function with an identically zero derivative (almost everywhere). Although one might attempt to smooth the Heaviside function by considering both branches, the bisection algorithm cannot be applied to intervals without a sign change.

Proceeding recursively eventually leads to

$$t_n = \eta E_0^{\text{left}} + (1 - \eta)E_0^{\text{right}} \tag{26a}$$

$$\frac{\partial t_n}{\partial \vec{p}} = \eta\frac{\partial E_0^{\text{left}}}{\partial \vec{p}} + (1 - \eta)\frac{\partial E_0^{\text{right}}}{\partial \vec{p}} \tag{26b}$$

for some $\eta \in (0, 1)$; however, equation 26b is obviously incorrect and should actually be

$$\frac{\partial t_n}{\partial \vec{p}} = \eta\frac{\partial E_0^{\text{left}}}{\partial \vec{p}} + (1 - \eta)\frac{\partial E_0^{\text{right}}}{\partial \vec{p}} + (E_0^{\text{left}} - E_0^{\text{right}})\frac{\partial \eta}{\partial \vec{p}} \tag{27}$$

based on equation 26a. Along the lines of Remark 4.3, consider finding $t^R \in (\frac{-b}{2a}, \infty)$. Substituting $E_0^{\text{left}} = \frac{-b}{2a}$, $E_0^{\text{right}} = K$ for large fixed constant K, and exact solution $t_n = \frac{-b+\sqrt{b^2-4ac}}{2a}$ into equation 26a leads to

$$\eta = 1 - \frac{\sqrt{b^2 - 4ac}}{b + 2aK} \tag{28}$$

which satisfies equation 27 (as expected), but not equation 26b (since $\frac{d\eta}{d\vec{p}} \neq \vec{0}$).

Remark 5.1: Although the inability to execute bisection for non-taken code branches causes the typical software infrastructures to incorrectly obtain the result in equation 26b, it appears that bisection could be made to be formally differentiable via an implicit layer (i.e. see equations 26a, 27, and 28).

Comparing equations 26b and 27 leads to a strategy that allows the typical software infrastructures to obtain a correct derivative even while using the incorrect equation 26b (we have verified this numerically). Setting

$$E_0^{\text{right}} = \frac{-b}{2a} + K\frac{\sqrt{b^2 - 4ac}}{2a} \tag{29}$$

with $K > 1$ results in $\eta = 1 - \frac{1}{K}$ and thus $\frac{d\eta}{d\vec{p}} = \vec{0}$.

6. Differentiating Newton's Method

Consider any iterative solver where t_n is a function of t_{n-1} and the parameters $\vec{p}$; then, the total derivative can be written as

$$dt_n = \frac{\partial t_n}{\partial t_{n-1}} dt_{n-1} + \frac{\partial t_n}{\partial \vec{p}} d\vec{p} \tag{30a}$$

$$\begin{bmatrix} dt_n^R \\ dt_n^I \end{bmatrix} = \begin{bmatrix} \frac{\partial t_n^R}{\partial t_{n-1}^R} & \frac{\partial t_n^R}{\partial t_{n-1}^I} \\ \frac{\partial t_n^I}{\partial t_{n-1}^R} & \frac{\partial t_n^I}{\partial t_{n-1}^I} \end{bmatrix} \begin{bmatrix} dt_{n-1}^R \\ dt_{n-1}^I \end{bmatrix} + \begin{bmatrix} \frac{\partial t_n^R}{\partial \vec{p}} \\ \frac{\partial t_n^I}{\partial \vec{p}} \end{bmatrix} d\vec{p} \tag{30b}$$

which becomes

$$dt_n = \prod_{j=0}^{n-1} \frac{\partial t_{j+1}}{\partial t_j} dt_0 + \sum_{i=1}^{n} \left(\prod_{j=i}^{n-1} \frac{\partial t_{j+1}}{\partial t_j} \right) \frac{\partial t_i}{\partial \vec{p}} d\vec{p} \tag{31}$$

using recursion. Focusing on Newton's method from equation 19,

$$\frac{\partial t_n}{\partial t_{n-1}} = I - \left(f_{\theta_1}^{-1}(t_{n-1};\vec{p}) \right)_{\theta_1} f(t_{n-1};\vec{p}) - f_{\theta_1}^{-1}(t_{n-1};\vec{p}) f_{\theta_1}(t_{n-1};\vec{p}) \tag{32a}$$

$$= - \left(f_{\theta_1}^{-1}(t_{n-1};\vec{p}) \right)_{\theta_1} f(t_{n-1};\vec{p}) \tag{32b}$$

$$\frac{\partial t_n}{\partial \vec{p}} = - \left(f_{\theta_1}^{-1}(t_{n-1};\vec{p}) \right)_{\theta_2} f(t_{n-1};\vec{p}) - f_{\theta_1}^{-1}(t_{n-1};\vec{p}) f_{\theta_2}(t_{n-1};\vec{p}) \tag{33}$$

taking some notational liberties for the sake of brevity. In equations 32b and 33, the θ_1 and θ_2 subscripts to the far right of $f_{\theta_1}^{-1}(t_{n-1};\vec{p})$ indicate replacing every term in the matrix and vector (respectively) with its appropriate row vector Jacobian without changing the dimension of the matrix or vector; then, the matrix-vector multiplications lead to a column of row vectors, which is treated as a matrix (as one would expect via tensor operations).

All of the terms in $\left(f_{\theta_1}^{-1}(t_{n-1};\vec{p}) \right)_{\theta_1}$ and $\left(f_{\theta_1}^{-1}(t_{n-1};\vec{p}) \right)_{\theta_2}$ can be accounted for via

$$\frac{\partial f_{\theta_1}^{-1}(t_{n-1};\vec{p})}{\partial t_{n-1}^R} = \frac{1}{s} \begin{bmatrix} 2a & 0 \\ 0 & 2a \end{bmatrix} - \frac{8a^2 t_{n-1}^R + 4ab}{s} f_{\theta_1}^{-1}(t_{n-1};\vec{p}) \tag{34a}$$

$$\frac{\partial f_{\theta_1}^{-1}(t_{n-1};\vec{p})}{\partial t_{n-1}^I} = \frac{1}{s} \begin{bmatrix} 0 & 2a \\ -2a & 0 \end{bmatrix} - \frac{8a^2 t_{n-1}^I}{s} f_{\theta_1}^{-1}(t_{n-1};\vec{p}) \tag{34b}$$

$$\frac{\partial f_{\theta_1}^{-1}(t_{n-1};\vec{p})}{\partial a} = \frac{1}{s} \begin{bmatrix} 2t_{n-1}^R & 2t_{n-1}^I \\ -2t_{n-1}^I & 2t_{n-1}^R \end{bmatrix} - \frac{8a(t_{n-1}^R)^2 + 4bt_{n-1}^R + 8a(t_{n-1}^I)^2}{s} f_{\theta_1}^{-1}(t_{n-1};\vec{p}) \tag{34c}$$

$$\frac{\partial f_{\theta_1}^{-1}(t_{n-1};\vec{p})}{\partial b} = \frac{1}{s} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \frac{4at_{n-1}^R + 2b}{s} f_{\theta_1}^{-1}(t_{n-1};\vec{p}) \tag{34d}$$

$$\frac{\partial f_{\theta_1}^{-1}(t_{n-1};\vec{p})}{\partial c} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{34e}$$

where the columns of $\left(f_{\theta_1}^{-1}(t_{n-1};\vec{p}) \right)_{\theta_1} f(t_{n-1};\vec{p})$ are formed by multiplying equations 34a-b by $f(t_{n-1};\vec{p})$ and the columns of $\left(f_{\theta_1}^{-1}(t_{n-1};\vec{p}) \right)_{\theta_2} f(t_{n-1};\vec{p})$ are formed by multiplying equations 34c-e by $f(t_{n-1};\vec{p})$. As long as s remains bounded away from zero, $\left(f_{\theta_1}^{-1}(t_{n-1};\vec{p}) \right)_{\theta_1} f(t_{n-1};\vec{p}) \to 0$ and $\left(f_{\theta_1}^{-1}(t_{n-1};\vec{p}) \right)_{\theta_2} f(t_{n-1};\vec{p}) \to 0$ as one converges to a solution where $f(t_{n-1};\vec{p}) \to 0$; then, equations 32 and 33 behave like

$$\frac{\partial t_n}{\partial t_{n-1}} \to \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{35}$$

$$\frac{\partial t_n}{\partial \vec{p}} \to -f_{\theta_1}^{-1}(t_{n-1};\vec{p}) f_{\theta_2}(t_{n-1};\vec{p}) \tag{36}$$

leading to

$$dt_n \to -f_{\theta_1}^{-1}(t_{n-1};\vec{p})f_{\theta_2}(t_{n-1};\vec{p})d\vec{p} \tag{37}$$

when plugged into equation 30. This is consistent with the implicit differentiation in Section 3.1, i.e. $d\theta_1 = -f_{\theta_1}^{-1}(t_{\text{root}},\vec{p})f_{\theta_2}(t_{\text{root}},\vec{p})d\theta_2$.

Next, consider the case where $s \to 0$ and thus $\alpha = 2at_{n-1}^R + b \to 0$ and $\beta = 2at_{n-1}^I \to 0$. If $a = 0$, then $b = 0$ and thus either everything or nothing is a root; in such cases, Newton's method takes zero iterations either because the initial guess is considered to be a root or because the derivative is identically equal to zero. Since backpropagating through Newton's method assuredly fails in this $a = 0$ case, assume $a \neq 0$ (note that our proposed method in Section 9 adequately deals with all degeneracies). Although the iterative scheme can result in $s = \alpha^2 + \beta^2 = 0$ any time any iterate has $t_{n-1}^R = \frac{-b}{2a}$ and $t_{n-1}^I = 0$, the most problematic case is when one is converging to such a result (i.e. a repeated root with $\tilde{c} = 0$). As discussed in Section 9.3, the derivatives have indeterminacies when the repeated root is identically equal to zero; thus, assume $b \neq 0$ in order to avoid such cases. Using the definitions of α, β, and $\tilde{c}$ leads to

$$f(t_{n-1};\vec{p}) = \frac{1}{4a}\begin{bmatrix}\alpha^2-\beta^2\\ 2\alpha\beta\end{bmatrix} + \frac{1}{a}\begin{bmatrix}\tilde{c}\\ 0\end{bmatrix} \tag{38a}$$

$$f_{\theta_1}(t_{n-1};\vec{p}) = \begin{bmatrix}\alpha & -\beta\\ \beta & \alpha\end{bmatrix} \tag{38b}$$

$$f_{\theta_1}^{-1}(t_{n-1};\vec{p}) = \frac{1}{\alpha^2+\beta^2}\begin{bmatrix}\alpha & \beta\\ -\beta & \alpha\end{bmatrix} \tag{38c}$$

and leveraging the inequalities

$$\left|\frac{\alpha^2}{\alpha^2+\beta^2}\right| \leq \left|\frac{\alpha^2}{\alpha^2}\right| = 1 \tag{39a}$$

$$\left|\frac{\beta^2}{\alpha^2+\beta^2}\right| \leq \left|\frac{\beta^2}{\beta^2}\right| = 1 \tag{39b}$$

$$\left|\frac{\alpha\beta}{\alpha^2+\beta^2}\right| \leq \max\left(\left|\frac{\alpha^2}{\alpha^2+\beta^2}\right|, \left|\frac{\beta^2}{\alpha^2+\beta^2}\right|\right) \leq 1 \tag{39c}$$

allows one to show that $\frac{1}{s}f(t_{n-1};\vec{p})$ and $\sqrt{s}f_{\theta_1}^{-1}(t_{n-1};\vec{p})$ are bounded when $\tilde{c} = 0$ in equation 38a; thus, $\frac{1}{\sqrt{s}}f_{\theta_1}^{-1}(t_{n-1};\vec{p})f(t_{n-1};\vec{p})$ is bounded and $f_{\theta_1}^{-1}(t_{n-1};\vec{p})f(t_{n-1};\vec{p}) \to 0$ even as $s \to 0$ (i.e., Newton's method converges even as $s \to 0$, see equation 19a). Next, consider multiplying the right hand sides of equations 34a-e by $f(t_{n-1};\vec{p})$ in order form the columns of $\left(f_{\theta_1}^{-1}(t_{n-1};\vec{p})\right)_{\theta_1} f(t_{n-1};\vec{p})$ and $\left(f_{\theta_1}^{-1}(t_{n-1};\vec{p})\right)_{\theta_2} f(t_{n-1};\vec{p})$. Since $\frac{1}{s}f(t_{n-1};\vec{p})$ and $\frac{1}{\sqrt{s}}f_{\theta_1}^{-1}(t_{n-1};\vec{p})f(t_{n-1};\vec{p})$ are bounded, one need only consider $\frac{1}{\sqrt{s}}(8a^2t_{n-1}^R + 4ab) = \frac{1}{\sqrt{s}}4a\alpha$, $\frac{1}{\sqrt{s}}(8a^2t_{n-1}^I) = \frac{1}{\sqrt{s}}4a\beta$, $\frac{1}{\sqrt{s}}(8a(t_{n-1}^R)^2 + 4bt_{n-1}^R + 8a(t_{n-1}^I)^2) = \frac{1}{\sqrt{s}}\frac{2}{a}(\alpha^2+\beta^2-b\alpha)$, and $\frac{1}{\sqrt{s}}(4at_{n-1}^R + 2b) = \frac{1}{\sqrt{s}}2\alpha$ which are all bounded according to equation 39. Thus, both $\left(f_{\theta_1}^{-1}(t_{n-1};\vec{p})\right)_{\theta_1} f(t_{n-1};\vec{p})$ and $\left(f_{\theta_1}^{-1}(t_{n-1};\vec{p})\right)_{\theta_2} f(t_{n-1};\vec{p})$ are bounded. Finally, consider $f_{\theta_1}^{-1}(t_{n-1};\vec{p})f_{\theta_2}(t_{n-1};\vec{p})$, which can be rewritten as

$$\frac{1}{s}\begin{bmatrix}\alpha & \beta\\ -\beta & \alpha\end{bmatrix}\begin{bmatrix}\frac{1}{4a^2}\left(\alpha^2-2\alpha b+b^2-\beta^2\right) & \frac{1}{2a}(\alpha-b) & 1\\ \frac{1}{2a^2}\left(\alpha\beta-b\beta\right) & \frac{1}{2a}\beta & 0\end{bmatrix} \to \frac{\alpha}{s}\begin{bmatrix}\frac{b^2}{4a^2} & \frac{-b}{2a} & 1\\ \frac{-b}{2a^2}\beta & \frac{1}{2a}\beta & 0\end{bmatrix} + \frac{\beta}{s}\begin{bmatrix}\frac{-b}{2a^2}\beta & \frac{1}{2a}\beta & 0\\ -\frac{b^2}{4a^2} & \frac{b}{2a} & -1\end{bmatrix} \tag{40}$$

where the expression to the right splits the diagonal and off-diagonal components of $f_{\theta_1}^{-1}(t_{n-1};\vec{p})$ into separate terms; in addition, all the terms that vanish (when combined with non-vanishing terms) have been eliminated. The first matrix bears similarity to equation 8 (when $\beta = 0$), and the second matrix bears similarity to equation 9 (when $\alpha = 0$). As long as $\alpha \to 0$ slower then $\beta^2 \to 0$, the first row in the first matrix blows up; when β is identically equal to zero, Newton's method is operating on real numbers only and only this top row is used. As long as $\beta \to 0$ slower then $\alpha^2 \to 0$, the second row in the second matrix blows up. These unbounded terms dominate the bounded $\left(f_{\theta_1}^{-1}(t_{n-1};\vec{p})\right)_{\theta_1} f(t_{n-1};\vec{p})$ and $\left(f_{\theta_1}^{-1}(t_{n-1};\vec{p})\right)_{\theta_2} f(t_{n-1};\vec{p})$ terms, implying that $\frac{\partial t_n}{\partial t_{n-1}}$ can still be ignored and equations 36 and equation 37 are still valid.

Remark 6.1: We have shown that backpropagating through Newton's method results in a vanishing (or ignorable) contribution from the recursive $\frac{\partial t_n}{\partial t_{n-1}}$ term when Newton iteration makes sense (i.e. a and b are not both zero), the derivatives make sense (i.e. avoiding the indeterminacies of an identically equal to zero repeated root), the roots are distinct (i.e. $s \neq 0$), and the roots are repeated except under very special circumstances (i.e. $\alpha \to 0$ as fast or faster than $\beta^2 \to 0$, and $\beta \to 0$ as fast or faster than $\alpha^2 \to 0$).

Remark 6.2: The convergence of equation 30 to equation 37 (which we have shown in the vast majority of cases) indicates that the recursive terms should (typically) make no contribution to dt_n as $t_n \to t_{root}$. Strategically, it makes little sense to aim for a robust implementation of backpropagation through Newton's method (including all of the degeneracies) that will at best (typically) do nothing.

6.1. One Parameter Examples

The three parameter family $\vec{p} = [a, b, c]^T$ can be reduced to a two parameter family $\hat{p} = [\hat{b}, \hat{c}]^T$ by dividing the quadratic equation by a, resulting in $\hat{b} = \frac{b}{a}$ and $\hat{c} = \frac{c}{a}$; alternatively, division by a can be avoided using a change of variables $\hat{t} = at$ to obtain $\hat{t}^2 + \hat{b}\hat{t} + \hat{c} = 0$ where $\hat{b} = b$ and $\hat{c} = ac$. The latter approach is used for the examples in this section, although the approaches are equivalent until the mapping between $\hat{p}$ and $\vec{p}$ is considered. The two parameter family $\hat{p}$ can be further reduced to a one parameter family $\tilde{p}$ (represented by a single scalar $\tilde{c}$). This can be accomplished either via the standard approach where $\tilde{t}^R = t^R + \frac{b}{2a}$ and $\tilde{t}^I = t^I$ yield $\tilde{t}^2 + \tilde{c} = 0$ with $\tilde{c} = \frac{-b^2}{4a^2} + \frac{c}{a}$, or via $\tilde{t}^R = at^R + \frac{b}{2}$ and $\tilde{t}^I = at^I$ to also obtain $\tilde{t}^2 + \tilde{c} = 0$ but with $\tilde{c} = \frac{-b^2}{4} + ac$ (which is the $\tilde{c}$ used throughout the paper); once again, the examples are indifferent until one considers the mapping between $\tilde{p}$ and $\vec{p}$.

Fixing $\tilde{a} = 1$ and $\tilde{b} = 0$ results in $d\tilde{a} = d\tilde{b} = 0$, and equation 14 reduces to

$$\begin{bmatrix} d\tilde{t}^R_{\text{root}} \\ d\tilde{t}^I_{\text{root}} \end{bmatrix} = -\frac{1}{s}\begin{bmatrix} 2\tilde{t}^R_{\text{root}} & 2\tilde{t}^I_{\text{root}} \\ -2\tilde{t}^I_{\text{root}} & 2\tilde{t}^R_{\text{root}} \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} d\tilde{c} = -\frac{1}{2\left((\tilde{t}^R_{\text{root}})^2 + (\tilde{t}^I_{\text{root}})^2\right)}\begin{bmatrix} \tilde{t}^R_{\text{root}} \\ -\tilde{t}^I_{\text{root}} \end{bmatrix} d\tilde{c} \tag{41}$$

for $\tilde{t}^2 + \tilde{c} = 0$. When $\tilde{c} \leq 0$, $\tilde{t}^R_{\text{root}} = \pm\sqrt{-\tilde{c}}$ and $\tilde{t}^I_{\text{root}} = 0$ reducing equation 41 to

$$d\tilde{t}^R_{\text{root}} = -\frac{1}{2\tilde{t}^R_{\text{root}}} d\tilde{c} = \mp\frac{1}{2\sqrt{-\tilde{c}}} d\tilde{c} \tag{42}$$

where $\frac{\partial \tilde{t}^R_{\text{root}}}{\partial \tilde{c}} \to \mp\infty$ as the root gains multiplicity with $\tilde{c} \to 0$. When $\tilde{c} \geq 0$, $\tilde{t}^R_{\text{root}} = 0$ and $\tilde{t}^I_{\text{root}} = \pm\sqrt{\tilde{c}}$ leading to $\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \tilde{c}} \to \pm\infty$ as $\tilde{c} \to 0$. Here, we present results for the $\tilde{t}^R_{\text{root}} = \sqrt{-\tilde{c}}$ case noting that the results for $\tilde{t}^R_{\text{root}} = -\sqrt{-\tilde{c}}$ and $\tilde{t}^I_{\text{root}} = \pm\sqrt{\tilde{c}}$ are similar. Plugging the final result of Newton iteration into the middle of equation 42 for $\tilde{t}^R_{\text{root}}$ gives

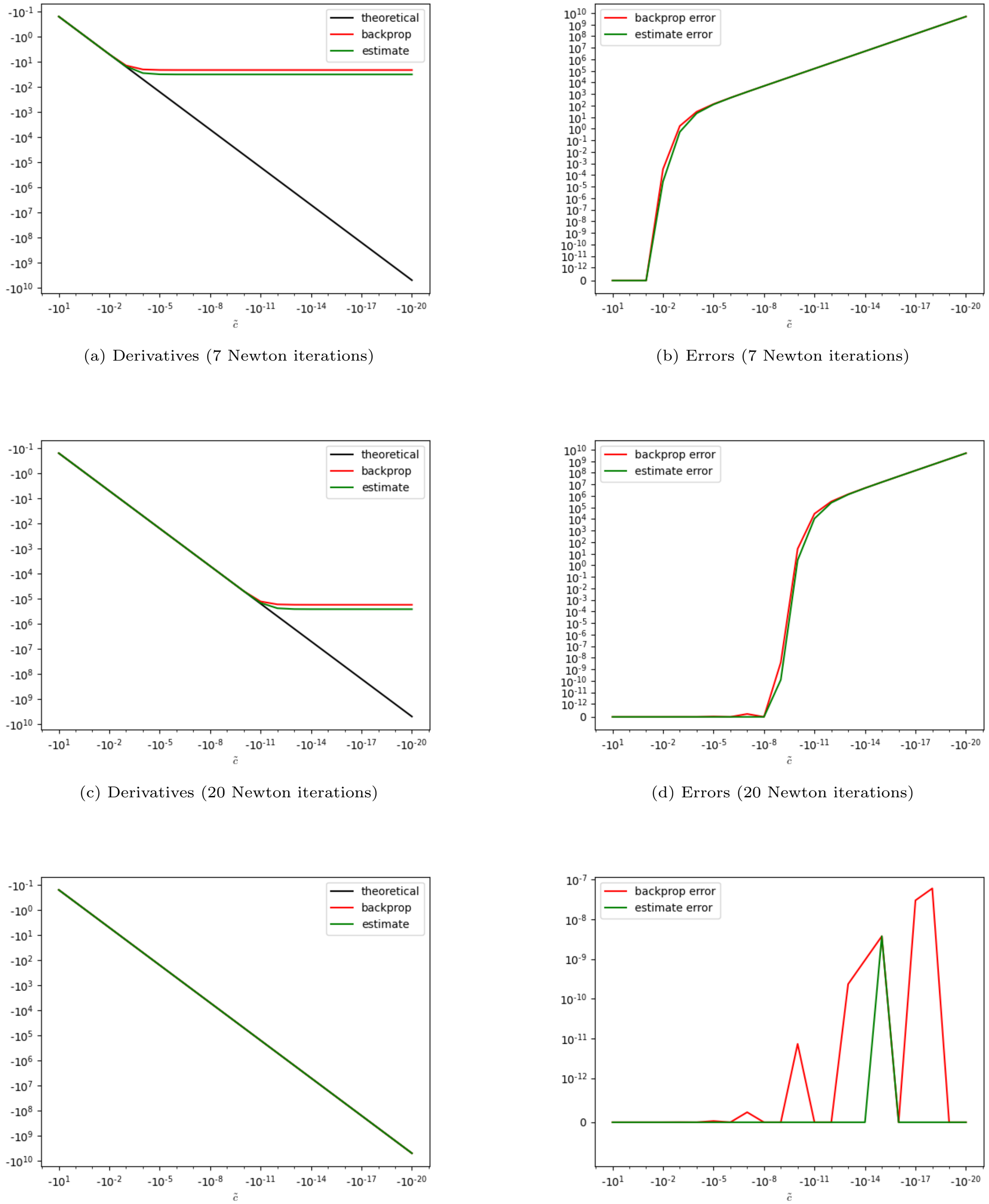
$$\frac{\partial \tilde{t}^R_{\text{root}}}{\partial \tilde{c}} \approx -\frac{1}{2\tilde{t}^R_n} \tag{43}$$

after n iterations of Newton's method.

For the sake of a baseline, we implemented backpropagation of Newton's method in Pytorch (and were careful to avoid the various degeneracies discussed earlier in this section). Figure 1 shows the results obtained using 7, 20, and 50 Newton iterations. As compared to the theoretical value, one would be hard pressed to argue for the benefits of backpropagation over the estimate in equation 43 or vice versa, since both have commensurate errors. Although increasing the number of Newton iterations does eventually give the desired results on smaller and smaller values of $\tilde{c}$, these rather large numbers of Newton iterations would not typically be used by a practitioner. In order to demonstrate that obtaining a reasonable derivative approximation is significantly more difficult than obtaining an accurate root, we experimentally determine the number of Newton iterations required to reduce the relative error to 1% for each. In order to avoid dividing by zero when computing the relative error for the root, we move the repeated root from $t = 0$ to $t = 1$ by using $t^2 - 2t + 1 - \epsilon = 0$ as $\epsilon \to 0$. Focusing on the root to the right, i.e. $t^R_{\text{root}} = 1 + \sqrt{\epsilon}$, equation 14 gives $\frac{\partial t^R_{\text{root}}}{\partial c} = -\frac{1}{2t^R_{\text{root}} - 2} = -\frac{1}{2\sqrt{\epsilon}}$ which resembles equation 42. Figure 2 shows that the root itself is obtained to 1% relative accuracy with 7 Newton iterations for varying values of ϵ, even as the repeated root is approached; however, an excessive number of Newton iterations is required to approximate the derivative to an equivalent 1% relative accuracy. We stress that 1% relative error in the derivative is a rather large absolute error as compared to the same relative error in the root, since the derivative is approaching infinity while the root is bounded. Achieving commensurate absolute error for the derivative approximation would require even more Newton iterations.

Remark 6.3: Given all of the degeneracies discussed earlier in this section, as well as the similar accuracies (and inaccuracies) obtained using either the estimate in equation 43 or standard backpropagation, we prefer to avoid backpropagation and instead proceed by developing a theoretical approach (resembling equation 43) that can be implemented via an implicit layer.

Remark 6.4: If one utilizes the best numerical algorithms available (differentiable or not) and subsequently seeks equations to describe that code, then one is empowered to pursue regularizations of the code's governing equations without requiring changes to the algorithms themselves (one may think of this as constitutive/continuum modeling of code). For example, instead of attempting to differentiate a particularly sensitive implementation of an iterative solver such as MINRES [83] for a poorly conditioned and/or singular $Ax = b$*, one can simply write* $x = A^+b$ *and thus* $\frac{\partial x}{\partial b} = A^+$ *noting that the pseudoinverse contains all the desired derivatives regardless of the algorithm used to solve* $Ax = b$*; then, one can efficiently/robustly estimate* A^+ *to the desired accuracy using robust PCA/SVD approaches including the power method [41], Lanczos iteration [84], etc.*



(a) Derivatives (7 Newton iterations)

(b) Errors (7 Newton iterations)

(c) Derivatives (20 Newton iterations)

(d) Errors (20 Newton iterations)

(e) Derivatives (50 Newton iterations)

(f) Errors (50 Newton iterations)

Figure 1: For $\tilde{t}^2 + \tilde{c} = 0$, the $\frac{d\tilde{t}^R_{\text{root}}}{d\tilde{c}}$ derivatives (as a function of $\tilde{c}$ on a log-log scale) computed using backpropagation (red), the equation 43 estimate based on an implicit layer (green), and the theoretical value from equation 42 (black). The errors are computed by comparing to the theoretical value from equation 42.

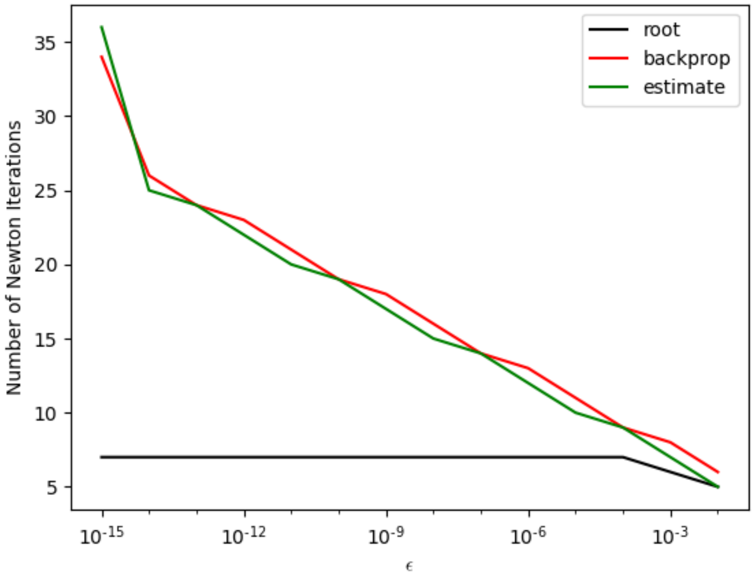


Figure 2: Only seven Newton iterations are required to obtain 1% relative error for the root (black) for varying values of ϵ in $t^2 - 2t + 1 - \epsilon = 0$. In contrast, both backpropagation (red) and the estimate based on the implicit layer (green) require an increasing number of Newton iterations to reach a 1% relative error as $\epsilon \to 0$.

6.2. Two Parameter Examples

Returning to the two parameter quadratic equation from Section 6.1, i.e. $\hat{p} = [\hat{b}\ \hat{c}]^T$, fixing $\hat{a} = 1$ results in $d\hat{a} = 0$ and equation 14 reduces to

$$\begin{bmatrix} d\hat{t}^R_{\text{root}} \\ d\hat{t}^I_{\text{root}} \end{bmatrix} = -\frac{1}{s} \begin{bmatrix} 2\hat{t}^R_{\text{root}} + \hat{b} & 2\hat{t}^I_{\text{root}} \\ -2\hat{t}^I_{\text{root}} & 2\hat{t}^R_{\text{root}} + \hat{b} \end{bmatrix} \begin{bmatrix} \hat{t}^R_{\text{root}} & 1 \\ \hat{t}^I_{\text{root}} & 0 \end{bmatrix} \begin{bmatrix} d\hat{b} \\ d\hat{c} \end{bmatrix} \tag{44}$$

for $\hat{t}^2 + \hat{b}\hat{t} + \hat{c} = 0$. Note that $\tilde{c} = \frac{-b^2}{4} + ac$ can still be used to classify the behavior. When $\tilde{c} \leq 0$, $\hat{t}^R_{\text{root}} = \frac{-\hat{b}}{2} \pm \sqrt{-\tilde{c}}$ and $\hat{t}^I_{\text{root}} = 0$ reducing equation 44 to

$$d\hat{t}^R_{\text{root}} = -\frac{1}{2\hat{t}^R_{\text{root}} + \hat{b}} \begin{bmatrix} \hat{t}^R_{\text{root}} & 1 \end{bmatrix} \begin{bmatrix} d\hat{b} \\ d\hat{c} \end{bmatrix} = \mp \frac{1}{2\sqrt{-\tilde{c}}} \begin{bmatrix} \frac{-\hat{b}}{2} \pm \sqrt{-\tilde{c}} & 1 \end{bmatrix} \begin{bmatrix} d\hat{b} \\ d\hat{c} \end{bmatrix} \tag{45}$$

where $\frac{\partial \hat{t}^R_{\text{root}}}{\partial \hat{c}} \to \mp\infty$ as the root gains multiplicity with $\tilde{c} \to 0$ (consistent with equation 42); in addition, $\frac{\partial \hat{t}^R_{\text{root}}}{\partial \hat{b}} \to \pm\infty$ (depending on the sign of $\hat{b}$) as $\tilde{c} \to 0$. When $\tilde{c} \geq 0$, $\hat{t}^R_{\text{root}} = \frac{-\hat{b}}{2}$ and $\hat{t}^I_{\text{root}} = \pm\sqrt{\tilde{c}}$ leading to $d\hat{t}^I_{\text{root}}$ behaving similarly to $d\hat{t}^R_{\text{root}}$ in equation 45 (also blowing up as $\tilde{c} \to 0$). Here, we present results for the $\hat{t}^R_{\text{root}} = \frac{-\hat{b}}{2} + \sqrt{-\tilde{c}}$ case noting that the results for $\hat{t}^R_{\text{root}} = \frac{-\hat{b}}{2} - \sqrt{-\tilde{c}}$ and the two complex roots are similar. Plugging the final result of Newton iteration into the middle of equation 45 gives

$$d\hat{t}^R_{\text{root}} \approx -\frac{1}{2\hat{t}^R_n + \hat{b}} \begin{bmatrix} \hat{t}^R_n & 1 \end{bmatrix} \begin{bmatrix} d\hat{b} \\ d\hat{c} \end{bmatrix} \tag{46}$$

after n iterations of Newton's method.

Considering only real roots (with $\hat{t}^I_n = d\hat{t}^I_n = 0$ for all n), equation 30b reduces to

$$d\hat{t}^R_n = \frac{\partial \hat{t}^R_n}{\partial \hat{t}^R_{n-1}} d\hat{t}^R_{n-1} + \frac{\partial \hat{t}^R_n}{\partial \hat{p}} d\hat{p} \tag{47}$$

while equations 34a and 34d reduce to

$$\frac{\partial f^{-1}_{\theta_1}(t_{n-1}; \hat{p})}{\partial \hat{t}^R_{n-1}} = -\frac{2}{(2\hat{t}^R_{n-1} + \hat{b})^2} I \tag{48}$$

and

$$\frac{\partial f^{-1}_{\theta_1}(t_{n-1}; \hat{p})}{\partial \hat{b}} = -\frac{1}{(2\hat{t}^R_{n-1} + \hat{b})^2} I \tag{49}$$

respectively. Multiplying equations 48 and 49 by $f(t_{n-1}; \hat{p})$ leads to

$$\frac{\partial \hat{t}^R_n}{\partial \hat{t}^R_{n-1}} = \frac{2((\hat{t}^R_{n-1})^2 + \hat{b}\hat{t}^R_{n-1} + \hat{c})}{(2\hat{t}^R_{n-1} + \hat{b})^2} = \frac{1}{2} + \frac{2\tilde{c}}{(2\hat{t}^R_{n-1} + \hat{b})^2} \tag{50}$$

and

$$\frac{\partial \hat{t}^R_n}{\partial \hat{p}} = \begin{bmatrix} \frac{(\hat{t}^R_{n-1})^2 + \hat{b}\hat{t}^R_{n-1} + \hat{c}}{(2\hat{t}^R_{n-1} + \hat{b})^2} & 0 \end{bmatrix} - \frac{1}{2\hat{t}^R_{n-1} + \hat{b}} \begin{bmatrix} \hat{t}^R_{n-1} & 1 \end{bmatrix} \tag{51a}$$

$$= \left(\frac{1}{2} + \frac{2\tilde{c}}{(2\hat{t}^R_{n-1} + \hat{b})^2} \right) \begin{bmatrix} \frac{1}{2} & 0 \end{bmatrix} - \frac{1}{2\hat{t}^R_{n-1} + \hat{b}} \begin{bmatrix} \hat{t}^R_{n-1} & 1 \end{bmatrix} \tag{51b}$$

respectively (following the derivations earlier in this section). Note that

$$\frac{1}{2} + \frac{2\tilde{c}}{(2\hat{t}^R_{n-1} + \hat{b})^2} \to 0 \tag{52}$$

as $\hat{t}^R_{n-1} \to \hat{t}^R_{\text{root}} = \frac{-\hat{b}}{2} + \sqrt{-\tilde{c}}$; thus, $\frac{\partial \hat{t}^R_n}{\partial \hat{t}^R_{n-1}} \to 0$ and the first term in equation 51b vanishes (leaving only the second term, which is identical to the estimate in equation 46) as $\hat{t}^R_{n-1} \to \hat{t}^R_{\text{root}}$.

As $\tilde{c} \to 0$, indicating closeness to a repeated root, $\hat{t}^R_{n-1} \to \frac{-\hat{b}}{2}$ and L'Hopital's rule is required for equation 52; in addition, $\frac{1}{2\hat{t}^R_{n-1}+\hat{b}}$ blows up. To demonstrate this numerically, set $\hat{b} = -2$ and $\hat{c} = 1 + \epsilon$ (which also sets $\tilde{c} = \epsilon$) so that $\hat{t}^R_{\text{root}} = 1 \pm \epsilon$ is close to being a repeated root. Table 1 shows the results for 10 Newton iterations, and Table 2 shows the results for 100 Newton iterations. The results shown in Table 2 substantiate our analysis; furthermore, Table 1 demonstrates the highly erroneous results obtained using only 10 Newton iterations, even though $\hat{t}^R_{10}$ has two significant digits of accuracy. The results for $\frac{\partial \hat{t}^R_n}{\partial \hat{c}}$ were omitted for brevity, but behave as expected.

ϵ	$(\hat{t}^R_{\text{root}})^+$	$\hat{t}^R_{10}$	$\frac{1}{2} + \frac{2\tilde{c}}{(2\hat{t}^R_{10}+\hat{b})^2}$	$-\frac{\hat{t}^R_{10}}{2\hat{t}^R_{10}+\hat{b}}$	Backprop $\frac{\partial \hat{t}^R_{10}}{\partial \hat{b}}$
-7.203e-09	1.000e+00	1.009e+00	5.000e-01	-5.439e+01	-3.643e+01
-3.023e-09	1.000e+00	1.009e+00	5.000e-01	-5.439e+01	-3.643e+01
-4.170e-10	1.000e+00	1.009e+00	5.000e-01	-5.439e+01	-3.643e+01
-1.468e-11	1.000e+00	1.009e+00	5.000e-01	-5.439e+01	-3.643e+01
-1.144e-12	1.000e+00	1.009e+00	5.000e-01	-5.439e+01	-3.643e+01

Table 1: Even after 10 Newton iterations when $\hat{t}^R_{10}$ has two significant digits of accuracy, columns 4, 5, and 6 all give erroneous values (as compared to the more accurate values in Table 2).

ϵ	$(\hat{t}^R_{\text{root}})^+$	$\hat{t}^R_{100}$	$\frac{1}{2} + \frac{2\tilde{c}}{(2\hat{t}^R_{100}+\hat{b})^2}$	$-\frac{\hat{t}^R_{100}}{2\hat{t}^R_{100}+\hat{b}}$	Backprop $\frac{\partial \hat{t}^R_{100}}{\partial \hat{b}}$
-7.203e-09	1.000e+00	1.000e+00	5.497e-09	-5.892e+03	-5.892e+03
-3.023e-09	1.000e+00	1.000e+00	1.632e-08	-9.094e+03	-9.094e+03
-4.170e-10	1.000e+00	1.000e+00	-1.121e-07	-2.449e+04	-2.449e+04
-1.468e-11	1.000e+00	1.000e+00	3.689e-06	-1.305e+05	-1.305e+05
-1.144e-12	1.000e+00	1.000e+00	-3.699e-05	-4.675e+05	-4.675e+05

Table 2: After 100 Newton iterations, the entries in column 4 are small and the estimates in column 5 match the results of backpropagation in column 6.

Next, consider the even more problematic $\hat{b} \approx 0$ case, and set $\hat{b} = 10^{-4}$ and $\hat{c} = \epsilon$ (which also sets $\tilde{c} = \epsilon - 2.5 \times 10^{-9}$) so that $\hat{t}^R_{\text{root}} = -5 \times 10^{-5} \pm \sqrt{2.5 \times 10^{-9} - \epsilon} \approx 0$ is close to being a repeated root. Table 3 shows the results after 100 Newton iterations where the root is well-converged, equation 52 is valid, and the estimate from equation 46 well-matches the results from backpropagation for $\frac{\partial \hat{t}^R}{\partial \hat{c}}$. Table 4 is a continuation of Table 3 (with matching rows) and shows $\gamma = \frac{\hat{b}^2 - 4\hat{c}}{\hat{b}^2} = \frac{-4\tilde{c}}{\hat{b}^2}$ along with expressions from equation 7. Note that the second column in Table 4 matches the last two columns in Table 3 (as expected). The agreement of equation 7, the equation 46 estimate, and backpropagation in the last three rows of Table 4 numerically validates our discussion of the non-removable singularity.

Remark 6.5: Simply switching from backpropagation to an implicit layer (or similar use of the implicit function theorem) is not enough to deal with the inherent non-removable singularity. This makes differentiating the equations instead of the code even more important since understanding the fundamental structure of the equations may be necessary in order to remedy indeterminate derivatives.

ϵ	$\tilde{c}$	$(\hat{t}^R_{\text{root}})^+$	$\hat{t}^R_{100}$	$\frac{1}{2}+\frac{2\tilde{c}}{(2\hat{t}^R_{100}+\hat{b})^2}$	$-\frac{1}{2\hat{t}^R_{100}+\hat{b}}$	Backprop $\frac{\partial \hat{t}^R_{100}}{\partial \hat{c}}$
-1.977e-06	-1.979e-06	1.357e-03	1.357e-03	-1.110e-16	-3.554e+02	-3.554e+02
-9.335e-08	-9.585e-08	2.596e-04	2.596e-04	5.551e-17	-1.615e+03	-1.615e+03
-2.159e-08	-2.409e-08	1.052e-04	1.052e-04	1.110e-16	-3.221e+03	-3.221e+03
3.743e-10	-2.126e-09	-3.894e-06	-3.894e-06	-1.110e-16	-1.084e+04	-1.084e+04
2.003e-09	-4.970e-10	-2.771e-05	-2.771e-05	3.331e-16	-2.243e+04	-2.243e+04
2.316e-09	-1.836e-10	-3.645e-05	-3.645e-05	1.221e-15	-3.690e+04	-3.690e+04
2.486e-09	-1.367e-11	-4.630e-05	-4.630e-05	5.218e-15	-1.352e+05	-1.352e+05

Table 3: For a problematically small $\hat{b} = 10^{-4}$, each row shows a small $\hat{c}$ leading to a small $\tilde{c}$ indicating that $\hat{t}^R_{\text{root}}$ is close to being a repeated root. See Table 4 for a continuation of the rows.

γ	$\frac{-1}{\lvert\hat{b}\rvert\sqrt{\gamma}}$	$\frac{-1}{2}\left(\frac{-1}{\text{Sign}(\hat{b})\sqrt{\gamma}}+1\right)$	$-\frac{\hat{t}^R_{100}}{2\hat{t}^R_{100}+\hat{b}}$	Backprop $\frac{\partial \hat{t}^R_{100}}{\partial \hat{b}}$
7.917e+02	-3.554e+02	-4.822e-01	-4.822e-01	-4.822e-01
3.834e+01	-1.615e+03	-4.192e-01	-4.192e-01	-4.192e-01
9.637e+00	-3.221e+03	-3.389e-01	-3.389e-01	-3.389e-01
8.503e-01	-1.084e+04	4.223e-02	4.223e-02	4.223e-02
1.988e-01	-2.243e+04	6.214e-01	6.214e-01	6.214e-01
7.346e-02	-3.690e+04	1.345e+00	1.345e+00	1.345e+00
5.469e-03	-1.352e+05	6.261e+00	6.261e+00	6.261e+00

Table 4: Equation 7 (columns 2 and 5), the equation 46 estimate (columns 3 and 6), and backpropagation (columns 4 and 7) all agree with each other as well as the non-removable nature of the derivatives with respect to $\hat{b}$ and the large magnitude of the derivatives with respect to $\hat{c}$.

7. Branch Selection

Correctly identifying the branches of $t^{\pm}_{\text{root}}$ is important, since at least one of them will appear in the objective function where it needs to be differentiated in order to obtain a search direction. Let $t_{\text{root,L}}$ designate a desired target value for t_{root}, and consider the one-parameter quadratic equation $\tilde{t}^2 + \tilde{c} = 0$. In this one-parameter quadratic equation, equation 12 (top) leads to $\tilde{c} = -(\tilde{t}^R_{\text{root,L}})^2 + (\tilde{t}^I_{\text{root,L}})^2$. When the target root is real-valued, $\tilde{c} = -(\tilde{t}^R_{\text{root,L}})^2$ leads to two real roots $(\tilde{t}^R_{\text{root}})^{\pm} = \pm\sqrt{-\tilde{c}} = \pm|\tilde{t}^R_{\text{root,L}}|$; importantly, only one of $(\tilde{t}^R_{\text{root}})^{\pm}$ matches $\tilde{t}^R_{\text{root,L}}$. In other words, one needs to work with $(\tilde{t}^R_{\text{root}})^{+}$ when $\tilde{t}^R_{\text{root,L}} > 0$ and $(\tilde{t}^R_{\text{root}})^{-}$ when $\tilde{t}^R_{\text{root,L}} < 0$, while both work when $\tilde{t}^R_{\text{root,L}} = 0$. When the target root is complex-valued, equation 12 (bottom) leads to $\tilde{t}^R_{\text{root,L}} = 0$ and thus $\tilde{c} = (\tilde{t}^I_{\text{root,L}})^2$; then, $(\tilde{t}^I_{\text{root}})^{\pm} = \pm\sqrt{\tilde{c}} = \pm|\tilde{t}^I_{\text{root,L}}|$. In other words, one needs to work with $(\tilde{t}^I_{\text{root}})^{+}$ when $\tilde{t}^I_{\text{root,L}} > 0$ and $(\tilde{t}^I_{\text{root}})^{-}$ when $\tilde{t}^I_{\text{root,L}} < 0$.

Next, consider the two-parameter quadratic equation $\hat{t}^2 + \hat{b}\hat{t} + \hat{c} = 0$ where equation 12 (top) leads to $\hat{c} = -(\hat{t}^R_{\text{root,L}})^2 + (\hat{t}^I_{\text{root,L}})^2 - \hat{t}^R_{\text{root,L}}\hat{b}$. Plugging this into equation 4 leads to

$$\hat{t}^{\pm}_{\text{root}} = \frac{-\hat{b} \pm \sqrt{\hat{b}^2 + 4(\hat{t}^R_{\text{root,L}})^2 + 4\hat{t}^R_{\text{root,L}}\hat{b} - 4(\hat{t}^I_{\text{root,L}})^2}}{2} \tag{53}$$

which becomes

$$(\hat{t}^R_{\text{root}})^{\pm} = \frac{-\hat{b} \pm |\hat{b} + 2\hat{t}^R_{\text{root,L}}|}{2} \tag{54}$$

when the target root is real-valued. When $\hat{b} \geq -2\hat{t}^R_{\text{root,L}}$, $(\hat{t}^R_{\text{root}})^{+} = \hat{t}^R_{\text{root,L}}$ and $(\hat{t}^R_{\text{root}})^{-} = -\hat{b} - \hat{t}^R_{\text{root,L}}$; otherwise, when $\hat{b} \leq -2\hat{t}^R_{\text{root,L}}$, $(\hat{t}^R_{\text{root}})^{+} = -\hat{b} - \hat{t}^R_{\text{root,L}}$ and $(\hat{t}^R_{\text{root}})^{-} = \hat{t}^R_{\text{root,L}}$. See Figure 3. Here, the choice of which $(\hat{t}^R_{\text{root}})^{\pm}$ to plug into the objective function depends on the value of $\hat{b}$, which itself has a one parameter set of potential values. The three-parameter quadratic equation has an additional degree of freedom, since the one-parameter set of values for $\hat{b} = \frac{b}{a}$ comes from a two-parameter set of values for a

and b. When the target root is complex-valued, equation 12 (bottom) leads to $\hat{t}^R_{\text{root,L}} = -\frac{\hat{b}}{2}$, which uniquely determines $\hat{b}$; however, the three-parameter quadratic equation has an additional degree of freedom, since $\hat{b} = \frac{b}{a}$. Substituting $\hat{t}^R_{\text{root,L}} = -\frac{\hat{b}}{2}$ into equation 53 leads to

$$\begin{bmatrix} (\hat{t}^R_{\text{root}})^{\pm} \\ (\hat{t}^I_{\text{root}})^{\pm} \end{bmatrix} = \begin{bmatrix} \frac{-\hat{b}}{2} \\ \pm \left|\hat{t}^I_{\text{root,L}}\right| \end{bmatrix} \tag{55}$$

illustrating that branch selection is only needed for $(\hat{t}^I_{\text{root}})^{\pm}$.

Remark 7.1: A real-valued target root has a unique solution for the one-parameter quadratic equation, a one-parameter family of solutions for the two-parameter quadratic equation, and a two-parameter family of solutions for the three-parameter quadratic equation. The real part of a complex-valued target root is identically zero (with no dependence on the target root) for the one-parameter quadratic equation, has a unique solution for the two-parameter quadratic equation, and has a one-parameter family of solutions for the three-parameter quadratic equation. The imaginary part of a complex-valued target root always has a unique solution (for all three quadratic equations).

Figure 4 plots the one-parameter family of solutions $(\hat{b}, \hat{c})$ corresponding to a target root $\hat{t}^R_{\text{root,L}} = \frac{1}{2}$, and shows the results obtained minimizing

$$L(\hat{p}) = \frac{1}{2}||\hat{t}_{\text{root}}(\hat{p}) - \hat{t}_{\text{root,L}}||_2^2 \tag{56}$$

using Adam [32] optimization with backpropagation (via PyTorch [33]) starting from an initial guess of $(\hat{b}_0, \hat{c}_0) = (-2, -4)$. Since $(\hat{t}^R_{\text{root}})^+$ is used in the objective function, the iteration converges to a point on the green ray (note that all points on the green ray are valid solutions). In Figure 5, the initial guess is modified to $(\hat{b}_0, \hat{c}_0) = (-5.1, 5)$ so that the parameter iterates enter the complex region on their way to the green ray. This highlights the fact that one needs to consider complex roots in both the Newton iteration and the parameter optimization, even in the case where both the initial guess and the final solutions are real-valued. Moreover, one needs to explicitly set the imaginary part of the root to zero in the objective function, e.g.

$$L(\hat{p}) = \frac{1}{2}\left|\left|\begin{bmatrix} \hat{t}^R_{\text{root}}(\hat{p}) \\ \hat{t}^I_{\text{root}}(\hat{p}) \end{bmatrix} - \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix}\right|\right|_2^2, \tag{57}$$

in order to enforce convergence to real-valued roots. Figure 6 instead minimizes

$$L(\hat{p}) = \frac{1}{2}\left|\left|\hat{t}^R_{\text{root}}(\hat{p}) - \frac{1}{2}\right|\right|_2^2 \tag{58}$$

demonstrating that the optimization can otherwise converge to a complex-valued root with a real part matching the desired $\hat{t}^R_{\text{root,L}}$. Finally, Figure 7 demonstrates what happens when one switches from using $\hat{t}^+_{\text{root}}$ in the objective function to instead using $\hat{t}^-_{\text{root}}$ in the objective function during parameter optimization.

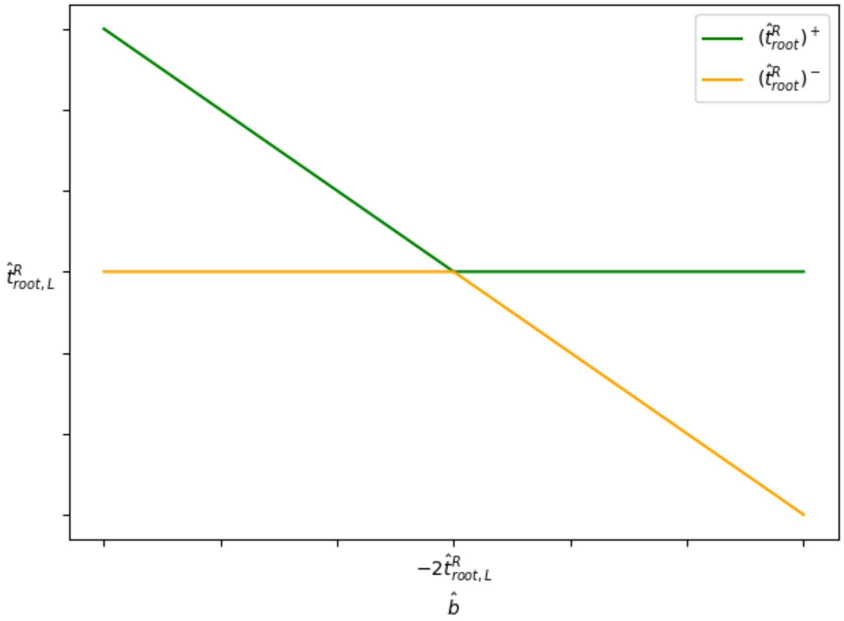


Figure 3: When $\hat{b} \geq -2\hat{t}^R_{\text{root,L}}$, $(\hat{t}^R_{\text{root}})^+ = \hat{t}^R_{\text{root,L}}$ and $(\hat{t}^R_{\text{root}})^- = -\hat{b} - \hat{t}^R_{\text{root,L}}$; otherwise, when $\hat{b} \leq -2\hat{t}^R_{\text{root,L}}$, $(\hat{t}^R_{\text{root}})^+ = -\hat{b} - \hat{t}^R_{\text{root,L}}$ and $(\hat{t}^R_{\text{root}})^- = \hat{t}^R_{\text{root,L}}$. See equation 54.

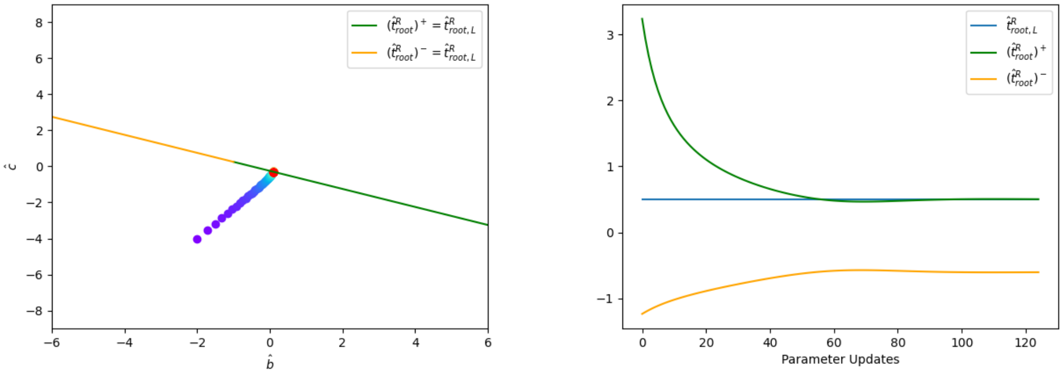


Figure 4: (Left) The line $\hat{c} = -\hat{t}^R_{\text{root,L}}\hat{b} - (\hat{t}^R_{\text{root,L}})^2$ of valid solutions with green denoting $(\hat{t}^R_{\text{root}})^+ = \hat{t}^R_{\text{root,L}}$ and yellow denoting $(\hat{t}^R_{\text{root}})^- = \hat{t}^R_{\text{root,L}}$. Each $(\hat{b}, \hat{c})$ parameter iterate is shown as a separate dot, color-coded from purple to red as iteration proceeds. (Right) The value of the $(\hat{t}^R_{\text{root}})^+$ converges to the target value of $\hat{t}^R_{\text{root,L}}$ as the optimization proceeds.

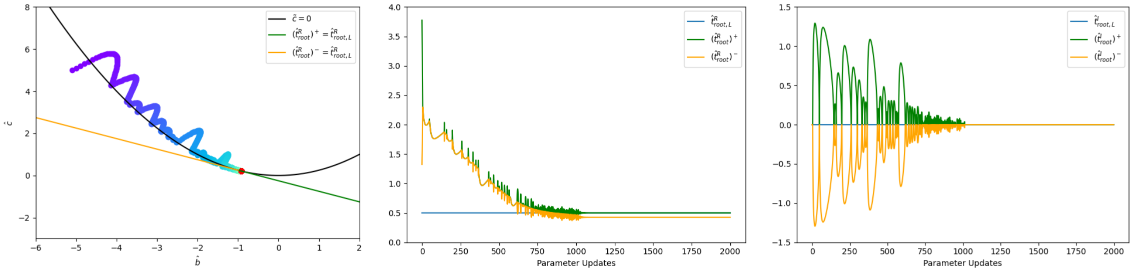


Figure 5: (Left) The parameter iterates oscillate into the complex region (with boundary denoted by the parabola $\hat{c} = \frac{\hat{b}^2}{4}$ where $\tilde{c} = 0$) as they proceed towards an valid solution on the $\hat{t}^+_{\text{root}} = \hat{t}_{\text{root,L}}$ green ray. (Middle) The real parts of the roots. (Right) The imaginary part of the roots.

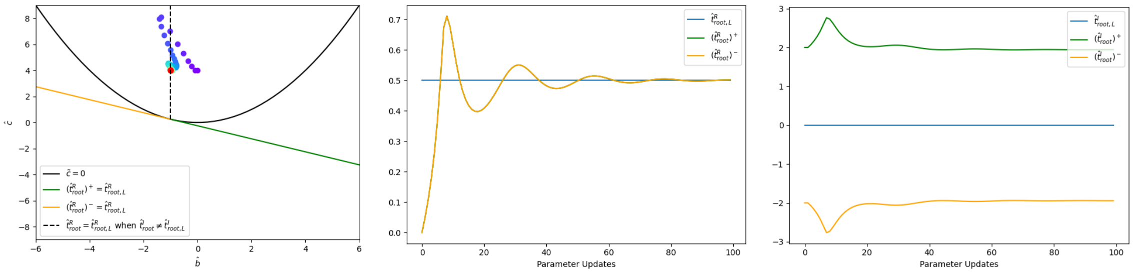


Figure 6: (Left) One can erroneously converge to the $\hat{t}^R_{\text{root}} = \hat{t}^R_{\text{root,L}}$ ray in the complex region, if $\hat{t}^I_{\text{root,L}}$ is not explicitly set to zero in the objective function. (Middle) In the complex region, $(\hat{t}^R_{\text{root}})^\pm$ coincide. (Right) In the complex region, $(\hat{t}^I_{\text{root}})^\pm$ are equal and opposite.

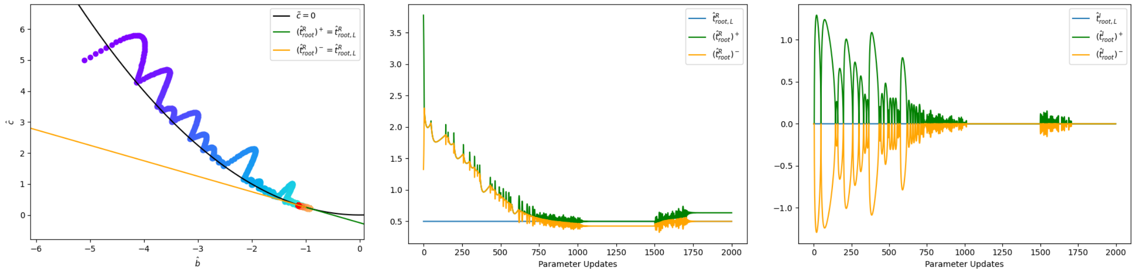


Figure 7: For the first 1500 iterations, $\hat{t}^+_{\text{root}}$ is used in the objective function; afterwards, $\hat{t}^-_{\text{root}}$ is used instead. Note how the iterates move from the green to the yellow ray (in the left figure), and how $(\hat{t}^R_{\text{root}})^-$ converges to $\hat{t}_{\text{root,L}}$ after 1500 iterations (in the middle figure).

8. Difficulties Near Repeated Roots

Many problems of interest will aim to either create or avoid a collision, and thus necessarily spend time iteratively wading back and forth between real and complex roots near $\tilde{c} = 0$, i.e. near coalescence to a repeated root. From equation 56,

$$\frac{\partial L}{\partial \hat{p}} = \frac{\partial L}{\partial \hat{t}_{\text{root}}}\frac{\partial \hat{t}_{\text{root}}}{\partial \hat{p}} = \left[\hat{t}^R_{\text{root}}(\hat{p}) - \hat{t}^R_{\text{root,L}} \quad \hat{t}^I_{\text{root}}(\hat{p}) - \hat{t}^I_{\text{root,L}}\right]\frac{\partial \hat{t}_{\text{root}}}{\partial \hat{p}} \tag{59}$$

where, from equation 44, one can write

$$\frac{\partial \hat{t}_{\text{root}}}{\partial \hat{p}} = \frac{-1}{\pm 2\sqrt{-\tilde{c}}}\begin{bmatrix} -\frac{\hat{b}}{2} & 1 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{bmatrix} \qquad \text{if } \tilde{c} < 0 \tag{60a}$$

$$\frac{\partial \hat{t}_{\text{root}}}{\partial \hat{p}} = \frac{1}{\pm 2\sqrt{\tilde{c}}}\begin{bmatrix} 0 & 0 \\ -\frac{\hat{b}}{2} & 1 \end{bmatrix} - \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{bmatrix} \qquad \text{if } \tilde{c} > 0 \tag{60b}$$

noting that $\tilde{c} < 0$ has $\hat{t}^R_{\text{root}} = -\frac{\hat{b}}{2} \pm \sqrt{-\tilde{c}}$ and $\hat{t}^I_{\text{root}} = 0$ while $\tilde{c} > 0$ has $\hat{t}^R_{\text{root}} = -\frac{\hat{b}}{2}$ and $\hat{t}^I_{\text{root}} = \pm\sqrt{\tilde{c}}$. Starting with an initial guess of $(\hat{b}_0, \hat{c}_0) = (-3, \frac{9}{4} - 10^{-12})$, so that $\tilde{c} = -\frac{\hat{b}^2}{4} + \hat{c} = -10^{-12}$ indicates closeness to a repeated root, Table 5 shows that gradient descent optimization (SGD) with backpropagation (via PyTorch) takes an erroneously large step in $(\hat{b}, \hat{c})$ because of the large values of $\frac{\partial L}{\partial \hat{b}}$ and $\frac{\partial L}{\partial \hat{c}}$. Afterwards, $\frac{\partial L}{\partial \hat{b}}$ and $\frac{\partial L}{\partial \hat{c}}$ are about 10^9 times smaller than $\hat{b}$ and $\hat{c}$, and SGD is unable to make any significant progress towards the target root $\hat{t}^R_{\text{root,L}} = \frac{1}{2}$. Both of these issues can be understood from equation 60a. $\frac{\partial \hat{t}^R_{\text{root}}}{\partial \hat{c}} = \frac{-1}{\pm 2\sqrt{-\tilde{c}}}$ blows up when $\tilde{c}$ is small and vanishes when $\tilde{c}$ is large. $\frac{\partial \hat{t}^R_{\text{root}}}{\partial \hat{b}} = \frac{\hat{b}}{\pm 4\sqrt{-\tilde{c}}} - \frac{1}{2}$ blows up when $\tilde{c}$ is small compared to $\hat{b}$; in addition, when $\hat{b}^2 >> \hat{c}$, $\tilde{c} \approx -\frac{\hat{b}^2}{4}$ and thus $\frac{\partial \hat{t}^R_{\text{root}}}{\partial \hat{b}} \approx \pm\frac{1}{2}\frac{\hat{b}}{|\hat{b}|} - \frac{1}{2}$ which is approximately zero for $(\hat{t}^R_{\text{root}})^+$ which is used in the objective function. Increasing the step size to a rather large value of 10^8 allows SGD to recover and converge after hundreds of iterations, albeit to rather large values for $(\hat{b}, \hat{c})$ as shown in Figure 8. For the sake of comparison, switching to Adam after the first iteration only required increasing the step size to 10^3 (still far too large) in order to recover within a similar number of iterations (see Figure 9).

Iterations	$\hat{b}$	$\hat{c}$	$\tilde{c}$	$(\hat{t}^R_{\text{root}})^+$	L	$\frac{\partial L}{\partial \hat{b}}$	$\frac{\partial L}{\partial \hat{c}}$
0	-3.000e+00	2.250e+00	-1.000e-12	1.500e+00	5.000e-01	-5.146e+05	-3.431e+05
1	5.146e+04	3.431e+04	-6.619e+08	-6.668e-01	6.807e-01	-1.512e-05	2.268e-05
2	5.146e+04	3.431e+04	-6.619e+08	-6.668e-01	6.807e-01	-1.512e-05	2.268e-05
100000	5.146e+04	3.431e+04	-6.619e+08	-6.668e-01	6.807e-01	-1.512e-05	2.268e-05

Table 5: SGD takes an erroneously large step in $(\hat{b}, \hat{c})$ because of the large values of $\frac{\partial L}{\partial \hat{b}}$ and $\frac{\partial L}{\partial \hat{c}}$ close to the repeated root; subsequently, it struggles to recover.

Repeating the aforementioned example using Adam (instead of SGD) alleviates issues with erroneously jumping to large values of $(\hat{b}, \hat{c})$; however, other issues lead to rather slow convergence (thousands of iterations). A representative example is shown in Figure 10. Combining equations 59 and 60 leads to

$$\frac{\partial L}{\partial \hat{p}} = \frac{\partial L}{\partial \hat{t}^R_{\text{root}}}\left(\frac{-1}{\pm 2\sqrt{-\tilde{c}}}\frac{\partial \tilde{c}}{\partial \hat{p}} - \begin{bmatrix} \frac{1}{2} & 0 \end{bmatrix}\right) \qquad \text{if } \tilde{c} < 0 \tag{61a}$$

$$\frac{\partial L}{\partial \hat{p}} = \frac{\partial L}{\partial \hat{t}^R_{\text{root}}}\begin{bmatrix} -\frac{1}{2} & 0 \end{bmatrix} + \frac{\partial L}{\partial \hat{t}^I_{\text{root}}}\frac{1}{\pm 2\sqrt{\tilde{c}}}\frac{\partial \tilde{c}}{\partial \hat{p}} \qquad \text{if } \tilde{c} > 0 \tag{61b}$$

where $\frac{\partial \tilde{c}}{\partial \hat{p}} = \begin{bmatrix} -\frac{\hat{b}}{2} & 1 \end{bmatrix}$ since $\tilde{c} = -\frac{\hat{b}^2}{4} + \hat{c}$. Equation 61 illustrates that small values of $\tilde{c}$ cause $\frac{\partial L}{\partial \hat{p}}$ to align with $\frac{\partial \tilde{c}}{\partial \hat{p}}$ as illustrated by the black arrows in Figure 11. Initially, as shown in Figure 10, the iterates start out strongly attracted to the $\tilde{c} = 0$ parabola where the derivatives tend to blow up and the descent direction is fairly orthogonal to the preferred direction (tangent to the $\tilde{c} = 0$ parabola) for making progress towards valid solutions on the green ray. Later (after entering the blue region of Figure 11), the iterates are no longer attracted to the $\tilde{c} = 0$ parabola allowing them to settle down and converge to a valid solution (on the green ray). See Figure 12.

Remark 8.1: When small values of $\tilde{c}$ on the denominator of equation 61 happen to be cancelled out by equivalently small values in the numerator, the analysis leading to the black arrows in Figure 11 needs some modification. For example, when $\hat{t}^I_{root,L} = 0$, equation 61b becomes

$$\frac{\partial L}{\partial \hat{p}} = \left(\frac{-\hat{b}}{2} - \hat{t}^R_{root,L} \right) \begin{bmatrix} -\frac{1}{2} & 0 \end{bmatrix} + (\pm\sqrt{\tilde{c}} - 0) \frac{1}{\pm 2\sqrt{\tilde{c}}} \begin{bmatrix} \frac{-\hat{b}}{2} & 1 \end{bmatrix} = \begin{bmatrix} \frac{\hat{t}^R_{root,L}}{2} & \frac{1}{2} \end{bmatrix} \tag{62}$$

after substituting the complex roots, $\hat{t}^R_{root} = -\frac{\hat{b}}{2}$ and $\hat{t}^I_{root} = \pm\sqrt{\tilde{c}}$, into $\frac{\partial L}{\partial \hat{t}_{root}}$ in equation 59. In Figure 11 where $\hat{t}^R_{root,L} = \frac{1}{2}$, equation 62 gives $-\frac{\partial L}{\partial \hat{p}} = -[\frac{1}{4}, \frac{1}{2}]$ indicating that the black arrows in the right subfigure of Figure 11 should point down and to the left in the entire complex region. Note that they still point towards the $\tilde{c} = 0$ parabola as long as $\hat{b} < 4$. This boundedness of $\frac{\partial L}{\partial \hat{p}}$ in the complex region is why the upper right subfigure of Figure 10 has many more iterates in the $\tilde{c} > 0$ region than in the $\tilde{c} < 0$ region where $\frac{\partial L}{\partial \hat{p}}$ does blow up as $\tilde{c}$ vanishes. These constant values of $\frac{\partial L}{\partial \hat{b}}$ and $\frac{\partial L}{\partial \hat{c}}$ (whenever $\tilde{c} > 0$) can also be seen in Figure 12.

It is not always be possible for the iterates to separate from the $\tilde{c} = 0$ parabola. In Figure 13, the initial guess is moved further away from the $\tilde{c} = 0$ parabola in order to illustrate how the iterates are still attracted towards it; then, the objective function is modified to prefer solutions close to the initial guess via

$$L(\hat{p}) = \frac{1}{2} \left(\left|\left| \hat{t}_{\text{root}}(\hat{p}) - \hat{t}_{\text{root,L}} \right|\right|_2^2 + \eta \left|\left| \hat{p} - \hat{p}_0 \right|\right|_2^2 \right) \tag{63}$$

where $\hat{p} = [\hat{b}\ \hat{c}]^T$. This relatively common modification of the objective function prevents the iterates from separating far enough away from the $\tilde{c} = 0$ parabola to avoid oscillations (and converge).

Next, consider target roots with nonzero imaginary parts, modifying $\hat{t}_{\text{root,L}} = \left[\frac{1}{2}, 0\right]^T$ to $\hat{t}_{\text{root,L}} = \left[\frac{1}{2}, 2\right]^T$. The behavior near the $\tilde{c} = 0$ parabola, as dictated by equation 61, is illustrated in Figure 14. Unfortunately, iterates that start below the $\tilde{c} = 0$ parabola in the blue region tend to erroneously converge to the green ray, because the identically zero imaginary component provides no information (below the $\tilde{c} = 0$ parabola) to prevent this; in addition, iterates below the green ray will also tend to erroneously converge to it. This can be remedied by including both $\hat{t}^+_{\text{root}}$ and $\hat{t}^-_{\text{root}}$ in the objective function, via

$$L(\hat{p}) = \frac{1}{2} \left(||\hat{t}^+_{\text{root}}(\hat{p}) - \hat{t}_{\text{root,L}}||_2^2 + ||\hat{t}^-_{\text{root}}(\hat{p}) - \hat{t}^*_{\text{root,L}}||_2^2 \right) \tag{64}$$

where $\hat{t}^*_{\text{root,L}} = \left[\frac{1}{2}, -2\right]^T$ is the complex conjugate, in order to create gradients below the $\tilde{c} = 0$ parabola that lead towards the intersection point of the green and yellow rays (i.e. towards the endpoint of the black dashed ray, as desired); in fact, only the real part of $\hat{t}^-_{\text{root}}$ needs to be included in the objective function. When one desires complex roots but is unconcerned with the precise value of the imaginary components, it is enough to force the real parts of both $\hat{t}^+_{\text{root}}$ and $\hat{t}^-_{\text{root}}$ to target the same $\hat{t}^R_{\text{root,L}}$ in the objective function; alternatively, when also unconcerned with the precise values of the real components, it is enough to minimize the difference between the real parts of $\hat{t}^+_{\text{root}}$ and $\hat{t}^-_{\text{root}}$.

To address the oscillations in Figures 5, 10, and 13, we first replace backpropagation (via PyTorch) with an implicit layer using equation 61a and 62, obtaining results quite similar to Figures 5, 10, and 13 as expected (see Figures 15 and 16, top rows). Then, convergence can be improved in all cases by clamping the magnitude of $\tilde{c}$ in equation 61a so that dividing by it no longer causes derivatives to blow up near the $\tilde{c} = 0$ parabola (see Figures 15 and 16, second rows). Alternatively, equation 61a can be rewritten as

$$\frac{\partial L}{\partial \hat{p}} = \frac{\partial L}{\partial \hat{t}^R_{\text{root}}} \left(\frac{-1}{\pm 2\sqrt{-\tilde{c}}} + \frac{\hat{b}}{\hat{b}^2 + 4} \right) \frac{\partial \tilde{c}}{\partial \hat{p}} + \frac{\partial L}{\partial \hat{t}^R_{\text{root}}} \frac{1}{\hat{b}^2 + 4} \begin{bmatrix} -2 & -\hat{b} \end{bmatrix} \quad \text{if } \tilde{c} < 0 \tag{65}$$

by splitting $[-\frac{1}{2}\ 0]$ into components parallel to and orthogonal to $\frac{\partial \tilde{c}}{\partial \hat{p}}$; then, (instead of clamping $\tilde{c}$) one can clamp the magnitude of the first term in equation 65 (see Figures 15 and 16, bottom rows).

Remark 8.2: Serendipitously, a reduced number of Newton iterations may lead to erroneously low values of $\tilde{c}$ (as illustrated in Figure 1) providing accidental, but beneficial, clamping; unfortunately, increasing the number of Newton iterations would then lead to (perhaps surprising) instability.

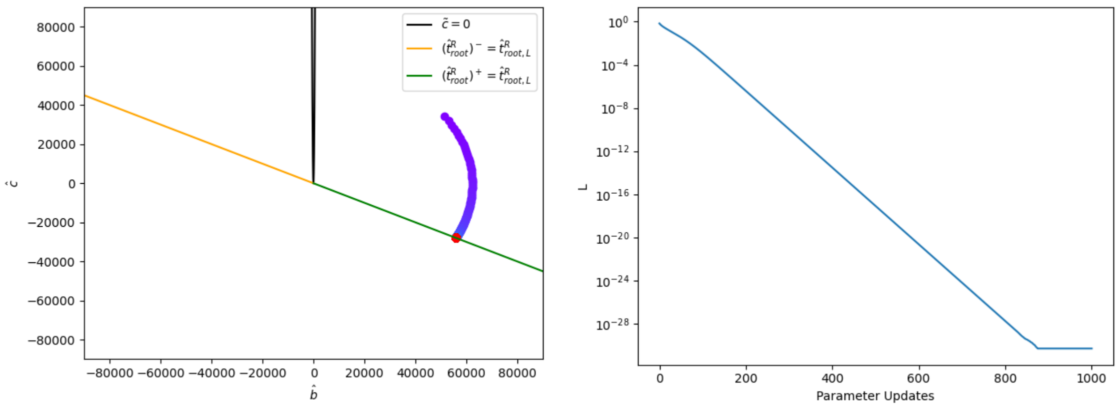


Figure 8: Starting from iteration 1 in Table 5, increasing the step size of SGD to 10^8 allows it to converge to the green ray (in hundreds of iterations).

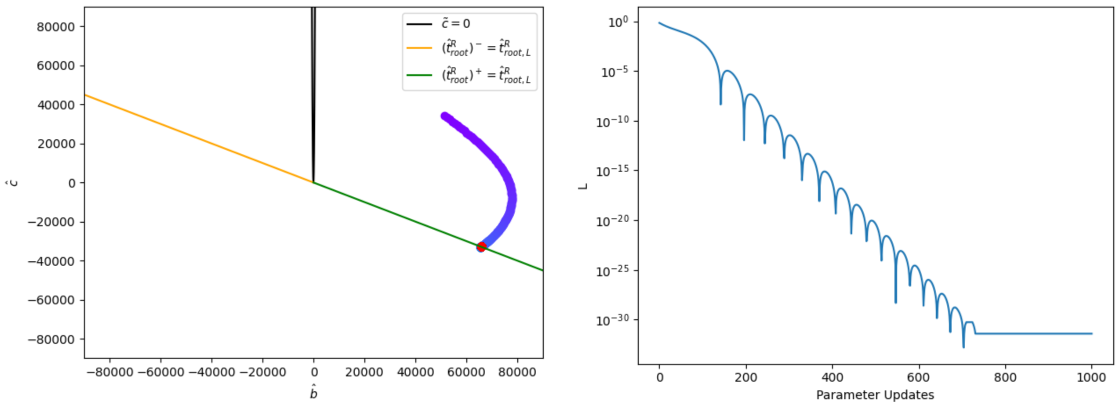


Figure 9: Switching to Adam after the first iteration in Table 5 and increasing the step size to 10^3 enables convergence to the green ray (in hundreds of iterations).

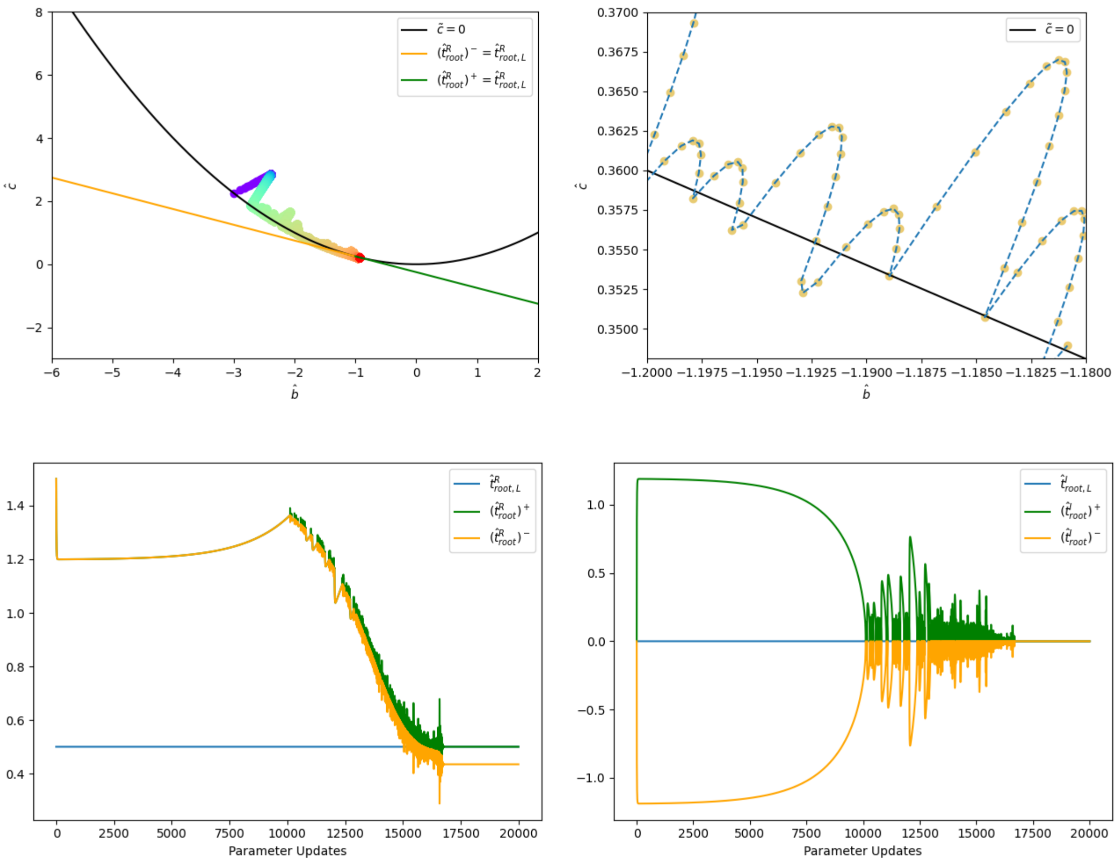


Figure 10: Although Adam performs better than SGD, i.e. not initially taking an erroneously large step in $(\hat{b}, \hat{c})$ because of the large values of $\frac{\partial L}{\partial \hat{b}}$ and $\frac{\partial L}{\partial \hat{c}}$ close to the repeated root case, it suffers from both oscillatory behavior and slow convergence. The top right subfigure shows a zoomed in view of the oscillatory behavior near the $\tilde{c}=0$ parabola.

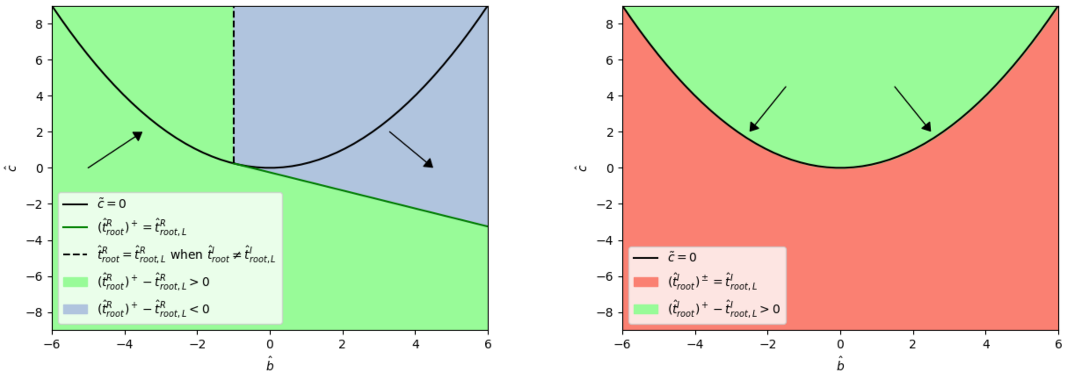


Figure 11: These figures address using $\hat{t}^+_{\text{root}}$ in the objective function (the figures for using $\hat{t}^-_{\text{root}}$ illustrate similar behavior). When the roots are real (below the $\tilde{c} = 0$ parabola), equation 61a indicates that a small $\tilde{c}$ makes the descent direction $-\frac{\partial L}{\partial \hat{p}}$ point in the same direction as $\frac{\partial \tilde{c}}{\partial \hat{p}}$ when $\frac{\partial L}{\partial \hat{t}^R_{\text{root}}}$ is positive (the green region) and opposite $\frac{\partial \tilde{c}}{\partial \hat{p}}$ when $\frac{\partial L}{\partial \hat{t}^R_{\text{root}}}$ is negative (the blue region). This is illustrated by the arrows in the left subfigure. When the roots are complex (above the $\tilde{c} = 0$ parabola), equation 61b illustrates that a small $\tilde{c}$ makes $-\frac{\partial L}{\partial \hat{p}}$ point opposite $\frac{\partial \tilde{c}}{\partial \hat{p}}$ when $\frac{\partial L}{\partial \hat{t}^I_{\text{root}}}$ is positive (the green region). This is illustrated by the arrows in the right subfigure. See Remark 8.1.

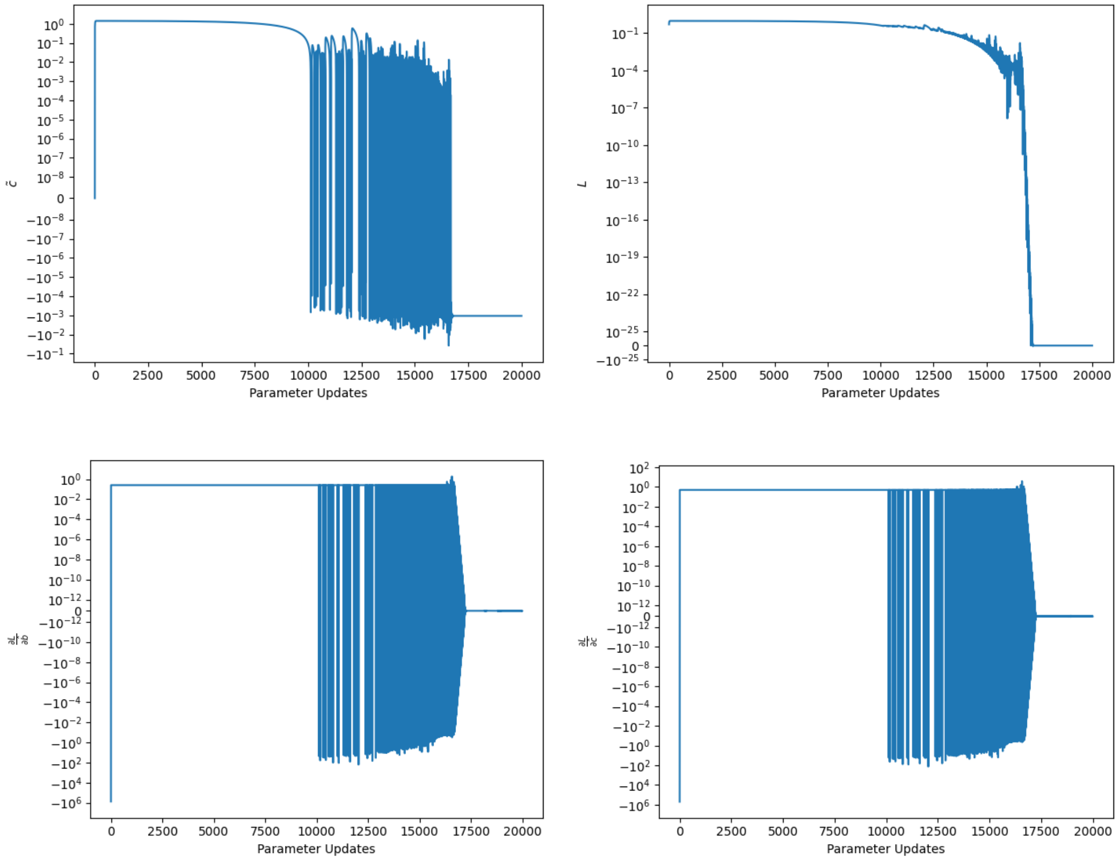


Figure 12: After entering the blue region in the left subfigure of Figure 11, the iterates are no longer attracted to the $\tilde{c} = 0$ parabola and can then converge to valid solutions (on the green ray).

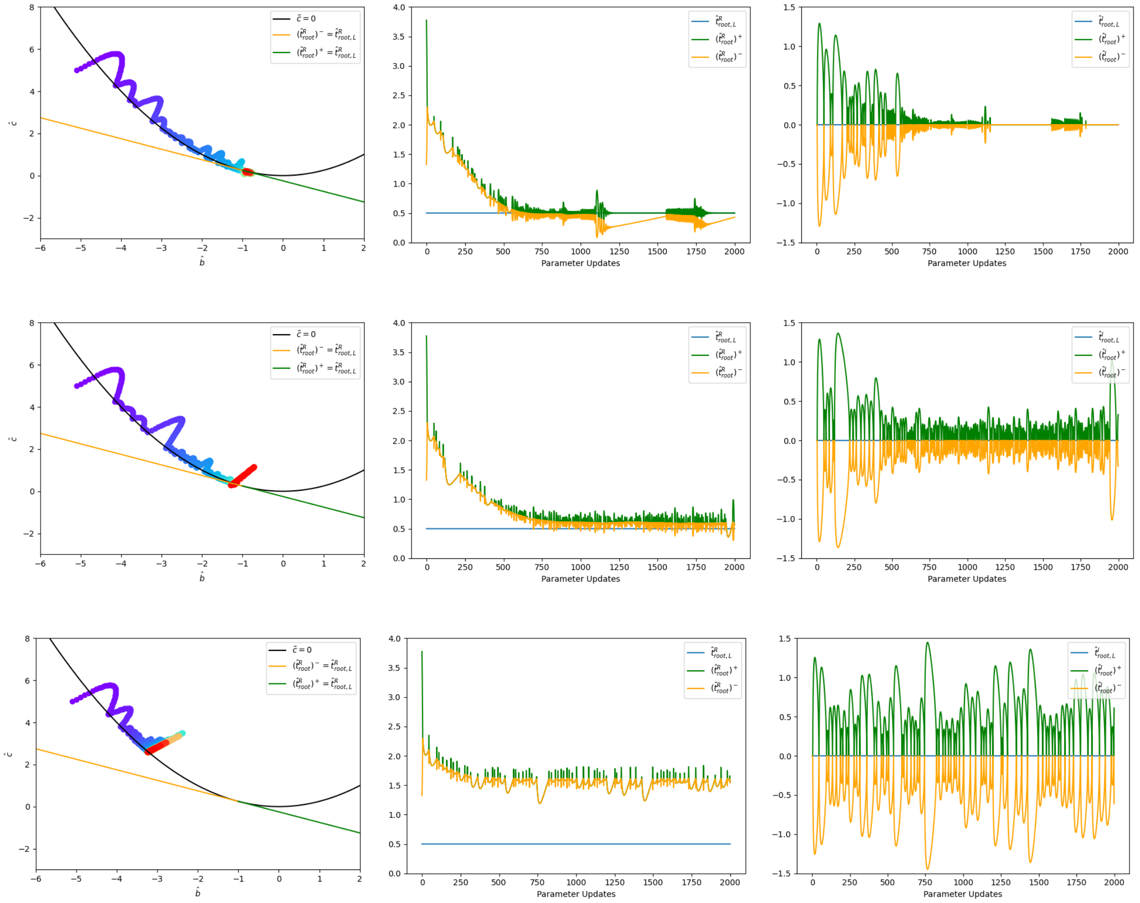


Figure 13: Equation 63 with $\eta = .001$ (top row), $\eta = .01$ (middle row), and $\eta = .1$ (bottom row). The η-term attracts the iterates towards the initial guess, subjecting them to highly oscillatory behavior near the $\tilde{c} = 0$ parabola (preventing convergence).

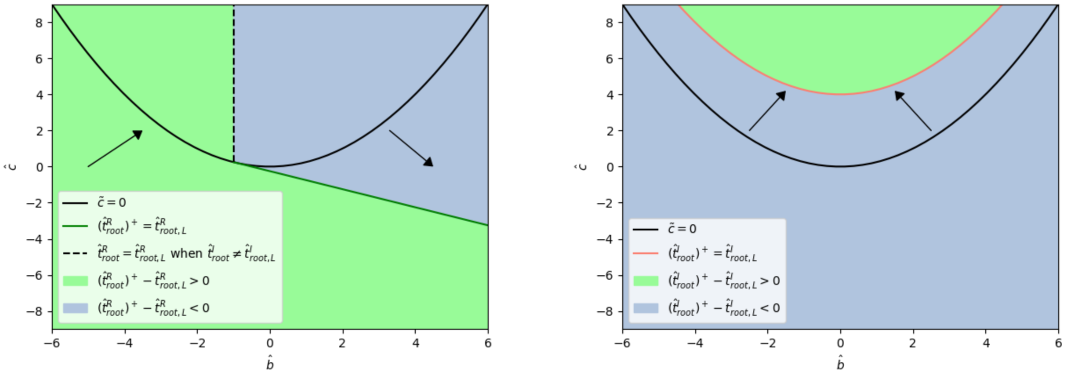


Figure 14: These figures address using $\hat{t}^+_{\text{root}}$ in the objective function (the figures for using $\hat{t}^-_{\text{root}}$ illustrate similar behavior). Modifying $\hat{t}_{\text{root,L}}$ to have a nonzero imaginary part does not affect equation 61a, and thus the left subfigure remains identical to that shown in Figure 11; however, as dictated by equation 61b, the arrows in the right subfigure change to point away from the $\tilde{c} = 0$ parabola and towards $(\hat{t}^I_{\text{root}})^+ = \hat{t}^I_{\text{root,L}}$ as desired.

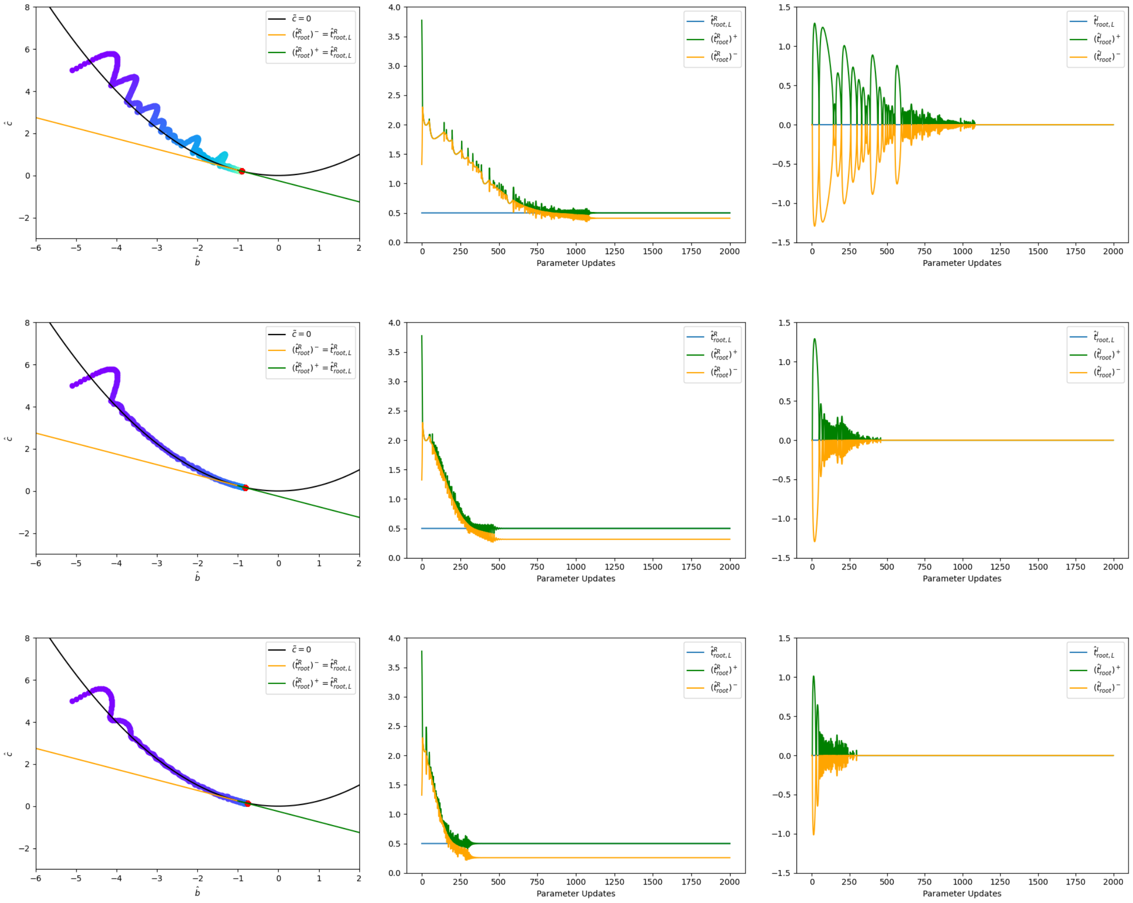


Figure 15: Replacing backpropagation (via PyTorch) with an implicit layer defined by equations 61a and 62 gives results (top row) similar to Figure 5 (as expected). Subsequently limiting the magnitude of $\tilde{c}$ in equation 61a (e.g. $|\tilde{c}| \geq 0.1$ on the second row) improves convergence. Breaking up the derivatives into components parallel to and orthogonal to $\frac{\partial \tilde{c}}{\partial \hat{p}}$ in equation 65 and subsequently clamping the magnitude of the parallel component (e.g. ≥ 1 on the third row) also improves convergence.

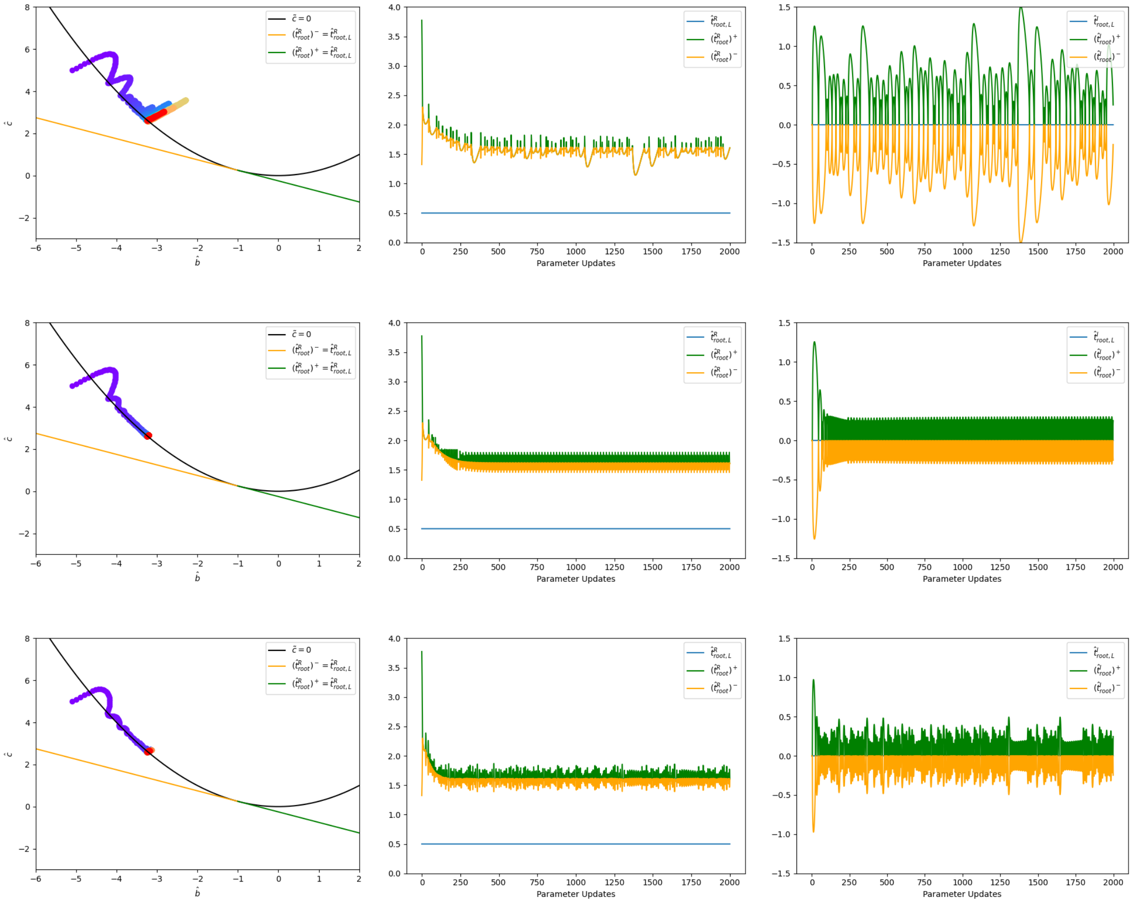


Figure 16: Replacing backpropagation (via PyTorch) with an implicit layer defined by equations 61a and 62 gives results (top row) similar to the bottom row of Figure 13 (as expected). Subsequently limiting the magnitude of $\tilde{c}$ in equation 61a (e.g. $|\tilde{c}| \geq 0.1$ on the second row) improves convergence. Breaking up the derivatives into components parallel to and orthogonal to $\frac{\partial \tilde{c}}{\partial \hat{p}}$ in equation 65 and subsequently clamping the magnitude of the parallel component (e.g. ≥ 1 on the third row) also improves convergence.

9. Proposed Approach for Quadratic Equations

We begin by leveraging the change of variables robust to $a \to 0$ discussed in Section 6.1,

$$\tilde{t} = at + \begin{bmatrix} \frac{b}{2} \\ 0 \end{bmatrix} \tag{66a}$$

$$\begin{bmatrix} \tilde{t}^R \\ \tilde{t}^I \end{bmatrix} = a \begin{bmatrix} t^R \\ t^I \end{bmatrix} + \begin{bmatrix} \frac{b}{2} \\ 0 \end{bmatrix} \tag{66b}$$

giving $\tilde{t}^2 + \tilde{c} = 0$ with $\tilde{c} = \frac{-b^2}{4} + ac$. In this reduced canonical form, equations 12 and 13 become

$$\tilde{f}(\tilde{t}_{\text{root}}; \tilde{p}) = \begin{bmatrix} (\tilde{t}^R_{\text{root}})^2 - (\tilde{t}^I_{\text{root}})^2 + \tilde{c} \\ 2\tilde{t}^R_{\text{root}}\tilde{t}^I_{\text{root}} \end{bmatrix} = af(t_{\text{root}}; \vec{p}) = \vec{0} \tag{67a}$$

$$\tilde{f}_{\theta_1}(\tilde{t}_{\text{root}}; \tilde{p}) = \begin{bmatrix} 2\tilde{t}^R_{\text{root}} & -2\tilde{t}^I_{\text{root}} \\ 2\tilde{t}^I_{\text{root}} & 2\tilde{t}^R_{\text{root}} \end{bmatrix} = f_{\theta_1}(t_{\text{root}}; \vec{p}) \tag{67b}$$

$$\tilde{f}^{-1}_{\theta_1}(\tilde{t}_{\text{root}}; \tilde{p}) = \frac{1}{s} \begin{bmatrix} 2\tilde{t}^R_{\text{root}} & 2\tilde{t}^I_{\text{root}} \\ -2\tilde{t}^I_{\text{root}} & 2\tilde{t}^R_{\text{root}} \end{bmatrix} \text{ where } s = 4((\tilde{t}^R_{\text{root}})^2 + (\tilde{t}^I_{\text{root}})^2) \tag{67c}$$

$$\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}}; \tilde{p}) = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = f_{\theta_2}(t_{\text{root}}; \vec{p}) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{67d}$$

where equation 67a matches equation 12 up to a factor of a, equations 67b-c match equations 13a-b, and equation 67d matches the last column of equation 13c. Note that $\frac{\partial \vec{p}}{\partial \tilde{p}}$ is a 3x3 Jacobian and that $\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}}; \tilde{p})$ is formally size 2x3; however, $\tilde{f}(\tilde{t}_{\text{root}}; \tilde{p})$ only requires the last column of $\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}}; \tilde{p})$, and so only this last column is shown in equation 67d.

The total derivative of equation 67a is $\tilde{f}_{\theta_1}(\tilde{t}_{\text{root}}, \tilde{p})d\theta_1 + \tilde{f}_{\theta_2}(\tilde{t}_{\text{root}}, \tilde{p})d\theta_2 = 0$, which can be written as $d\theta_1 = -\tilde{f}^{-1}_{\theta_1}(\tilde{t}_{\text{root}}, \tilde{p})\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}}, \tilde{p})d\theta_2$ when $\tilde{f}_{\theta_1}$ is invertible; in other words, $\frac{\partial \theta_1}{\partial \theta_2} = -\tilde{f}^{-1}_{\theta_1}(\tilde{t}_{\text{root}}, \tilde{p})\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}}, \tilde{p})$. As long as θ_1 and θ_2 are not independent, one can write $d\theta_1 = \frac{\partial \theta_1}{\partial \theta_2}d\theta_2$ leading to $\tilde{f}_{\theta_1}(\tilde{t}_{\text{root}}, \tilde{p})\frac{\partial \theta_1}{\partial \theta_2} = -\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}}, \tilde{p})$ or

$$\begin{bmatrix} 2\tilde{t}^R_{\text{root}} & -2\tilde{t}^I_{\text{root}} \\ 2\tilde{t}^I_{\text{root}} & 2\tilde{t}^R_{\text{root}} \end{bmatrix} \begin{bmatrix} \frac{\partial \tilde{t}^R_{\text{root}}}{\partial \tilde{c}} \\ \frac{\partial \tilde{t}^I_{\text{root}}}{\partial \tilde{c}} \end{bmatrix} = - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \tag{68}$$

even when $\tilde{f}_{\theta_1}$ is not invertible. When the roots are real (and not repeated) with $\tilde{t}^R_{\text{root}} = \pm\sqrt{-\tilde{c}}$ and $\tilde{t}^I_{\text{root}} = 0$, equation 68 gives $\frac{\partial \tilde{t}^R_{\text{root}}}{\partial \tilde{c}} = \frac{-1}{2\tilde{t}^R_{\text{root}}} = \frac{-1}{\pm 2\sqrt{-\tilde{c}}}$ and $\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \tilde{c}} = 0$. When the roots are complex with $\tilde{t}^R_{\text{root}} = 0$ and $\tilde{t}^I_{\text{root}} = \pm\sqrt{\tilde{c}}$, equation 68 gives $\frac{\partial \tilde{t}^R_{\text{root}}}{\partial \tilde{c}} = 0$ and $\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \tilde{c}} = \frac{1}{2\tilde{t}^I_{\text{root}}} = \frac{1}{\pm 2\sqrt{\tilde{c}}}$. In the repeated root case, $\tilde{c} = \tilde{t}^R_{\text{root}} = \tilde{t}^I_{\text{root}} = 0$ making $\tilde{f}_{\theta_1}$ identically zero so that $\tilde{f}^{-1}_{\theta_1}$ does not exist; then, one can no longer rely on sloppy interpretations of the implicit function theorem in order to write statements such as $d\theta_1 = -\tilde{f}^{-1}_{\theta_1}(\tilde{t}_{\text{root}}, \tilde{p})\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}}, \tilde{p})d\theta_2$ and $\frac{\partial \theta_1}{\partial \theta_2} = -\tilde{f}^{-1}_{\theta_1}(\tilde{t}_{\text{root}}, \tilde{p})\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}}, \tilde{p})$. However, equation 68 is still valid (asymptotically) in spite of the coefficient matrix going to zero. This can be seen from the real side by plugging in the solution $[\frac{-1}{2\tilde{t}^R_{\text{root}}}, 0]^T$ to obtain the right hand side via (a trivial) L'Hospital's rule. Similarly from the complex side, plugging in $[0, \frac{1}{2\tilde{t}^I_{\text{root}}}]^T$ leads to the right hand side.

In this reduced canonical form, $\tilde{t}_{\text{root}}$ only depends on $\tilde{c}$ and thus

$$\frac{\partial \tilde{t}_{\text{root}}}{\partial \vec{p}} = \frac{\partial \tilde{t}_{\text{root}}}{\partial \tilde{c}} \frac{\partial \tilde{c}}{\partial \vec{p}} \tag{69a}$$

$$\tilde{f}_{\theta_1}(\tilde{t}_{\text{root}}, \tilde{p})\frac{\partial \tilde{t}_{\text{root}}}{\partial \vec{p}} = \tilde{f}_{\theta_1}(\tilde{t}_{\text{root}}, \tilde{p})\frac{\partial \tilde{t}_{\text{root}}}{\partial \tilde{c}} \frac{\partial \tilde{c}}{\partial \vec{p}} = - \begin{bmatrix} 1 \\ 0 \end{bmatrix} \frac{\partial \tilde{c}}{\partial \vec{p}} \tag{69b}$$

since the left hand side of equation 68 is $\tilde{f}_{\theta_1}(\tilde{t}_{\text{root}}, \tilde{p})\frac{\partial \tilde{t}_{\text{root}}}{\partial \tilde{c}}$. Using equation 66 to expand the left hand side of equation 69b leads to

$$\tilde{f}_{\theta_1}(\tilde{t}_{\text{root}}, \tilde{p}) \left(a\frac{\partial t_{\text{root}}}{\partial \vec{p}} + \begin{bmatrix} t^R_{\text{root}} & \frac{1}{2} & 0 \\ t^I_{\text{root}} & 0 & 0 \end{bmatrix} \right) = -\begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{bmatrix} c & \frac{-b}{2} & a \end{bmatrix} \tag{70a}$$

$$a\tilde{f}_{\theta_1}(\tilde{t}_{\text{root}}, \tilde{p}) \frac{\partial t_{\text{root}}}{\partial \vec{p}} = -a \begin{bmatrix} (t^R_{\text{root}})^2 - (t^I_{\text{root}})^2 & t^R_{\text{root}} & 1 \\ 2t^R_{\text{root}} t^I_{\text{root}} & t^I_{\text{root}} & 0 \end{bmatrix} \tag{70b}$$

$$\tilde{f}_{\theta_1}(\tilde{t}_{\text{root}}, \tilde{p}) \frac{\partial t_{\text{root}}}{\partial \vec{p}} = -\begin{bmatrix} (t^R_{\text{root}})^2 - (t^I_{\text{root}})^2 & t^R_{\text{root}} & 1 \\ 2t^R_{\text{root}} t^I_{\text{root}} & t^I_{\text{root}} & 0 \end{bmatrix} \tag{70c}$$

where equation 12 was used to simplify the right hand side of equation 70b. Importantly, equation 70c was obtained from equation 70b by dividing by a, which allows for $a \to 0$ but not $a = 0$.

Remark 9.1: Note that equation 70c is identically $f_{\theta_1}(t_{root}; \vec{p}) \frac{\partial t_{root}}{\partial \vec{p}} = -f_{\theta_2}(t_{root}; \vec{p})$*. That is, equation 14 can be treated more carefully via equation 70c (similar in spirit to equation 68).*

From equation 67c, one can obtain

$$\tilde{f}^{-1}_{\theta_1}(\tilde{t}_{\text{root}}; \tilde{p}) = \begin{cases} \frac{1}{\pm 2\sqrt{-\tilde{c}}} I & \text{if } \tilde{c} < 0 \\ \frac{1}{\pm 2\sqrt{\tilde{c}}} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} & \text{if } \tilde{c} > 0 \end{cases} \tag{71a}$$

$$\tilde{f}^{-1}_{\theta_1}(\tilde{t}_{\text{root}}; \tilde{p}) = \begin{cases} \frac{1}{2\tilde{t}^R_{\text{root}}} I & \text{if } \tilde{t}^I_{\text{root}} = 0 \\ \frac{1}{2\tilde{t}^I_{\text{root}}} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} & \text{if } \tilde{t}^I_{\text{root}} \neq 0 \end{cases} \tag{71b}$$

$$\tilde{f}^{-1}_{\theta_1}(\tilde{t}_{\text{root}}; \tilde{p}) = \begin{cases} \frac{1}{2at^R_{\text{root}}+b} I & \text{if } t^I_{\text{root}} = 0 \\ \frac{1}{2at^I_{\text{root}}} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} & \text{if } t^I_{\text{root}} \neq 0 \end{cases} \tag{71c}$$

which are all identical; however, equation 71a explicitly maintains the ±-sign differentiating between the two coalescing roots as the denominator goes to (and becomes identically equal to) zero. We proceed using equation 71c, relying on equation 71a only to motivate the algorithm.

Remark 9.2: As long as the standard definition of $t^{\pm}_{root}$ *is used (i.e. equation 4), both* $2at^R_{root} + b$ *and* $2at^I_{root}$ *should be clamped to be positive/negative when using* $t^{\pm}_{root}$ *in the objective function respectively.*

Remark 9.3: This clamping (in Remark 9.2) alleviates issues when: the roots are close to being repeated and $2at^R_{root} + b \approx 0$ *for both roots but* $2at^R_{root} + b$ *is incorrectly the same sign (instead of opposite signs) for both roots due to numerical errors. Note that similar issues do not arise for complex roots where both* $t^I_{root} \neq 0$ *and* $a \neq 0$*, since the* $\pm-$*sign follows directly (and correctly) from* t^I_{root} *and* a*.*

Remark 9.4: In the degenerate $a = 0$ *case (which has two real roots), the sign of* $2at^R_{root} + b$ *is correctly determined by* $b \neq 0$ *for the finite root, but should be clamped to be opposite the sign of* b *for the infinite root (due to a nonzero value for* $2at^R_{root}$ *from L'Hospital's rule). This is properly treated by the clamping in Remark 9.2. The* $a = b = 0$ *case (with two infinite roots) is also correctly treated by the clamping in Remark 9.2 (see Section 9.1).*

Remark 9.5: The choice between the $\tilde{c} < 0$ *and* $\tilde{c} > 0$ *cases is unimportant for truly repeated roots, since (see e.g. Figures 11 and 14) the search directions across the* $\tilde{c} = 0$ *parabola (whether consistent or inconsistent) cannot be significantly improved by any choice of search direction on the* $\tilde{c} = 0$ *parabola; instead, one needs to carefully craft the objective function.*

Using equation 71c in equation 70c leads to

$$\frac{\partial t_{\text{root}}}{\partial \vec{p}} = \begin{cases} \frac{-1}{2at^R_{\text{root}}+b} \begin{bmatrix} (t^R_{\text{root}})^2 & t^R_{\text{root}} & 1 \\ 0 & 0 & 0 \end{bmatrix} & \text{if } t^I_{\text{root}} = 0 \\ \frac{-1}{2at^I_{\text{root}}} \begin{bmatrix} 2t^R_{\text{root}} t^I_{\text{root}} & t^I_{\text{root}} & 0 \\ -((t^R_{\text{root}})^2 - (t^I_{\text{root}})^2) & -t^R_{\text{root}} & -1 \end{bmatrix} & \text{if } t^I_{\text{root}} \neq 0 \end{cases} \tag{72}$$

where the rows of the matrices provide directions for $\frac{\partial t^R_{\text{root}}}{\partial \vec{p}}$ and $\frac{\partial t^I_{\text{root}}}{\partial \vec{p}}$. In the $t^I_{\text{root}} = 0$ case, consider the top row of the first matrix in equation 72. When $(t^R_{\text{root}})^2 > 1$, we factor $(t^R_{\text{root}})^2$ out into the numerator of the scalar multiplier so that each component of the vector is bounded by 1. One could also factor out the magnitude of the top row, if a unit vector were desired. The scalar multiplier (out front) is then evaluated robustly as follows: Let N be the magnitude of the numerator, D be the magnitude of the denominator, and M be the maximum allowable magnitude of the result. If $N < DM$, then $\frac{N}{D}$ can be robustly computed; otherwise, $\frac{N}{D}$ is set to M without the need for (potentially problematic) division. In the $t^I_{\text{root}} \neq 0$ case, we consider the top and bottom rows of the second matrix in equation 72 separately. For the top row, t^I_{root} is factored out front (where it cancels) leaving only $2a$ on the denominator. Note that the ±-sign from equation 71 is unnecessary for this (non-merging) real part of the complex root, and only the sign of a is required (recall, $a \neq 0$ for complex roots). When $|2t^R_{\text{root}}| > 1$, $2t^R_{\text{root}}$ is also factored out front; then, the scalar multiplier is robustly evaluated (as discussed above). For the bottom row, the larger in magnitude between $(t^R_{\text{root}})^2 - (t^I_{\text{root}})^2$ and t^R_{root} is factored out front when it is larger than 1; then, the scalar multiplier is robustly evaluated.

Remark 9.6: It can be problematic to evaluate $(t^R_{root})^2 - (t^I_{root})^2$ when both t^R_{root} and t^I_{root} are quite large. This can be alleviated to some degree by instead considering $(t^R_{root} + t^I_{root})(t^R_{root} - t^I_{root})$. Since t^R_{root} and t^I_{root} can only be large when a is small, one would likely prefer to have this term dominate the second row in this case (as opposed to letting perhaps erroneous and commensurate values of t^R_{root} and t^I_{root} cancel, which is quite likely to happen when they are both clamped to t_{max}).

9.1. Quadratic Root Solver

When $b^2 - 4ac < 0$, the roots are complex and $a \neq 0$. One can (using $\pm t_{\max}$ as the upper bound) divide $-b$ by $2a$ for the real part and $\sqrt{-(b^2 - 4ac)}$ by $2a$ for the imaginary part. Otherwise, the roots are real. In the $a = 0$ case, a pseudo-sign for a is required in order to assign the larger/smaller computed root to $t^{\pm}_{\text{root}}$ for Remark 9.2.

When $b = 0$, equation 18 reduces to $\frac{\pm\sqrt{-ac}}{a}$. When $a \neq 0$, robust division can be used to compute $\pm\frac{\sqrt{|c|}}{\sqrt{|a|}}$. When $a = 0$ and $c \neq 0$, $\pm\frac{\sqrt{|c|}}{\sqrt{|a|}} \to \pm t_{\max}$; in addition, the pseudo-sign for a is set opposite the sign of c (which is required in order for the roots to be real). When $a = 0$ and $c = 0$ making everything a root, we return a repeated root at 0 (keeping the problem symmetric) and arbitrarily set a pseudo-sign of $a > 0$. This leads to [0 0 1] in the top row of the first matrix in equation 72, which fixes c to be nonzero creating roots at $\pm t_{\max}$; then, the top row subsequently becomes [1 0 0], which fixes a to make the roots smaller.

When $b \neq 0$, the $-b \pm \sqrt{b^2 - 4ac}$ terms in equation 18 are always nonzero; thus, only division by $2a$ (when $a = 0$) is problematic. Both problematic quotients reduce to $\frac{-b}{a} \to \pm t_{\max}$ depending on the pseudo-sign of a, and we arbitrarily set a pseudo-sign of $a > 0$.

Remark 9.7: Since a, b, and c do not grow too large in our examples, we use the $\vec{p}_{orig}$ values in the root solver (not the normalized coefficients).

Remark 9.8: Replacing t with t^{-1} leads to a reversed quadratic $ct^2 + bt + a = 0$, which can be solved to obtain one over the roots (changing the roles of a and c in equation 18). In the case of complex roots, one can thus choose to divide by either 2a or 2c. In the case of real roots and $b = 0$, choosing to divide by c instead of a does not help much since the result needs to be flipped anyways. When $b \neq 0$, the problematic $\frac{-b}{a}$ case with $a = 0$ is robustly replaced by $\frac{a}{-b}$ in equation 18 which is identically zero (but unfortunately needs to be flipped). Overall, this strategy of solving for t^{-1} is not necessarily beneficial for the quadratic equation.

9.2. Branch Selection

The formulation of the objective function will typically be problem dependent. As can be seen in equations 1 and 2, each root used will typically need to be differentiated with respect to its parameters. The root solver (in Section 9.1) provides values for both roots for any set of parameters, and those values can be used in equation 72 to robustly compute derivatives.

One particular case that is worth addressing is when a changes sign. When $a = 0$, the quadratic degenerates to a linear function with one root; however, perturbations of a cause the second root to be near $\pm\infty$ depending on the sign of a. As a changes sign, the root of the linear equation switches from being the smaller/larger root to being the larger/smaller root respectively. As can be seen in equation 18, the linear root only depends on b (it is t^+_{root} when $b > 0$ and t^-_{root} when $b < 0$) and not on a; thus, the roles of $t^{\pm}_{\text{root}}$ in the objective function do not need to change when a changes sign. See Figure 21.

9.3. Examples

In this section, we show the efficacy for our proposed approach using the root solver from Section 9.1 while computing derivatives according to equation 72 using the robust division discussed in the text (after equation 72). The sign of the denominator of each scalar multiplier in equation 72 is chosen according to Remark 9.2. Similar to equation 56 (and equation 57), we use

$$L(p) = \frac{1}{2}||t_{\text{root}}(p) - t_{\text{root,L}}||_2^2 \tag{73}$$

as the objective function, and choose t^+_{root} as the branch under consideration. Adam was used for the optimization.

We first reconsider the examples from Figures 5, 10, and 13 using our proposed approach; however, for the sake of a clean comparison, we remove the first column in equation 72 by setting $da = 0$ and holding $a = 1$. Keeping $a = 1$ fixed leads to $\hat{t} = at = t$, $\hat{b} = b$, and $\hat{c} = ac = c$ for the change of variables in the beginning of Section 6.1 (used in Figures 5, 10, and 13). Figure 17 repeats the example from Figure 5 illustrating some of the results one might expect when choosing different values for M in the robust division. For larger values of M, the results most closely match those that would be obtained using analytic derivatives except when the analytic derivatives lead to division by small numbers causing catastrophic overflow. Smaller values of M can alleviate the oscillations along the $\tilde{c} = 0$ parabola allowing for faster convergence. Figure 18 repeats the example from Figure 10. Even with a large value of $M = 10000$, the initial derivative is bounded enough to obtain convergence more than twice as fast as in Figure 10. The middle row of Figure 17 and the bottom row of Figure 18 were chosen to illustrate an occasional lack of convergence caused by the iterates settling down towards the left endpoint of the green ray very close to the problematic $\tilde{c} = 0$ parabola. The derivatives in this region (which tend to point orthogonal to the $\tilde{c} = 0$ parabola) can sometimes overcome the momentum from Adam stopping rightward motion (sometimes even driving the iterates to the left). Although we occasionally observed such behavior, the iterates typically contain enough momentum to continue moving to the right. To verify this explanation, Figure 19 illustrates what one would expect when the direction orthogonal to the $\tilde{c} = 0$ parabola tends to point towards the right (instead of towards the left) in the region of interest (near the left endpoint of the green ray). Finally, Figure 20 repeats the example from Figure 13.

In the subsequent examples, a is allowed to vary; thus, the first column in equation 72 is included. For the sake of visualization only, we still plot some results using the change of variables $\hat{t} = at$, $\hat{b} = b$, and $\hat{c} = ac$. The line of acceptable solutions $\hat{c} = -\hat{t}^R_{\text{root,L}}\hat{b} - (\hat{t}^R_{\text{root,L}})^2 = -at^R_{\text{root,L}}\hat{b} - a^2(t^R_{\text{root,L}})^2$ varies as a varies; thus, to minimize confusion, we only plot it for the last iteration. In spite of the line moving around, the analysis leading to Figure 3 is still valid implying that $(\hat{t}^R_{\text{root}})^- = \hat{t}^R_{\text{root,L}}$ on the left and $(\hat{t}^R_{\text{root}})^+ = \hat{t}^R_{\text{root,L}}$ on the right. Finally, note that $\hat{t}^R_{\text{root,L}} = at^R_{\text{root,L}}$ varies as a varies, as can be seen in Figure 21 (top right).

Next, we demonstrate robustness with regard to degeneracies. Figure 22 (top row) starts with $[a, b, c]^T = [0, -5.1, 5]^T$, and Table 6 shows the first few iterations. In spite of $(t^R_{\text{root}})^+$ not being the linear root and thus starting out at our maximum allowable value of 10^{150}, our proposed approach recovers to obtain $(t^R_{\text{root}})^+ = t_{\text{root,L}}$ as desired. The robust root solver in Section 9.1 used a pseudo-sign of $a > 0$ in order to obtain $(t^R_{\text{root}})^+ = 10^{150}$. As discussed in Remark 9.2, $2at^R_{\text{root}} + b$ is clamped to be positive in equation 72 leading to $\frac{\partial(t^R_{\text{root}})^+}{\partial a} < 0$; then, $\frac{\partial L}{\partial a} = \frac{\partial L}{\partial(t^R_{\text{root}})^+}\frac{\partial(t^R_{\text{root}})^+}{\partial a} = -10^{153}$ since $\frac{\partial L}{\partial(t^R_{\text{root}})^+} \approx (t^R_{\text{root}})^+ = 10^{150}$ and $\frac{\partial(t^R_{\text{root}})^+}{\partial a} = -M = -1000$ via clamping. Figure 22 (middle row) starts with $[a, b, c]^T = [0, 0, 5]^T$, and Table 7 shows the first few iterations. Since $c > 0$, the robust root solver in Section 9.1 assumes $a < 0$ to obtain $(t^R_{\text{root}})^\pm = \mp 10^{150}$ respectively; then, $\frac{\partial L}{\partial a} = 10^{153}$ since $\frac{\partial L}{\partial(t^R_{\text{root}})^+} \approx -10^{150}$ and $\frac{\partial(t^R_{\text{root}})^+}{\partial a} = -M = -1000$ via clamping. Figure 22 (bottom row) starts with $[a, b, c]^T = [0, 0, 0]^T$, and Table 8 shows the first few iterations. The robust root solver in Section 9.1 used a pseudo-sign of $a > 0$ and sets $(t^R_{\text{root}})^\pm = 0$; then, $\frac{\partial L}{\partial c} = 500$ since $\frac{\partial L}{\partial(t^R_{\text{root}})^+} = (t^R_{\text{root}})^+ - t^R_{\text{root,L}} = -.5$ and $\frac{\partial(t^R_{\text{root}})^+}{\partial c} = -M = -1000$ via clamping. This leads to $c < 0$ in the next iteration, and the method recovers in a manner similar to the $a = b = 0$ example shown in Table 7.

Itr.	a	b	c	$(t^R_{\text{root}})^+$	$(t^R_{\text{root}})^-$	$\tilde{c}$	$\frac{\partial L}{\partial a}$	$\frac{\partial L}{\partial b}$	$\frac{\partial L}{\partial c}$
0	0.00e+00	-5.10e+00	5.00e+00	1.00e+150	9.80e-01	-6.50e+00	-1.00e+153	-1.00e+03	-1.00e-147
1	1.00e-01	-5.00e+00	5.00e+00	4.90e+01	1.02e+00	-5.75e+00	-2.43e+04	-4.95e+02	-1.01e+01
2	1.67e-01	-4.91e+00	5.07e+00	2.83e+01	1.07e+00	-5.17e+00	-4.90e+03	-1.73e+02	-6.11e+00

Table 6: Starting with $[a, b, c]^T = [0, -5.1, 5]^T$

Itr.	a	b	c	$(t^R_{\text{root}})^+$	$(t^R_{\text{root}})^-$	$\tilde{c}$	$\frac{\partial L}{\partial a}$	$\frac{\partial L}{\partial b}$	$\frac{\partial L}{\partial c}$
0	0.00e+00	0.00e+00	5.00e+00	-1.00e+150	1.00e+150	0.00e+00	1.00e+153	-1.00e+03	1.00e-147
1	-1.00e-01	1.00e-01	5.00e+00	-6.59e+00	7.59e+00	-5.02e-01	2.17e+02	-3.29e+01	5.00e+00
2	-1.67e-01	1.69e-01	4.93e+00	-4.95e+00	5.96e+00	-8.30e-01	7.32e+01	-1.48e+01	2.99e+00

Table 7: Starting with $[a, b, c]^T = [0, 0, 5]^T$

Itr.	a	b	c	$(t^R_{\text{root}})^+$	$(t^R_{\text{root}})^-$	$\tilde{c}$	$\frac{\partial L}{\partial a}$	$\frac{\partial L}{\partial b}$	$\frac{\partial L}{\partial c}$
0	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	5.00e+02
1	0.00e+00	0.00e+00	-1.00e-01	1.00e+150	-1.00e+150	0.00e+00	-1.00e+153	-1.00e+03	-1.00e-147
2	7.44e-02	7.44e-02	-1.67e-01	1.08e+00	-2.08e+00	-1.38e-02	-2.87e+00	-2.66e+00	-2.46e+00

Table 8: Starting with $[a, b, c]^T = [0, 0, 0]^T$

Finally, we repeat the examples shown in Figures 5, 10, and 13 using the proposed approach with a, b, and c allowed to vary. Figure 23 repeats the example shown in Figure 5 (and Figure 17). Figures 24 and 25 repeat the example shown in Figure 10 (and Figure 18). Figure 26 repeats the example shown in Figure 13 (and Figure 20).

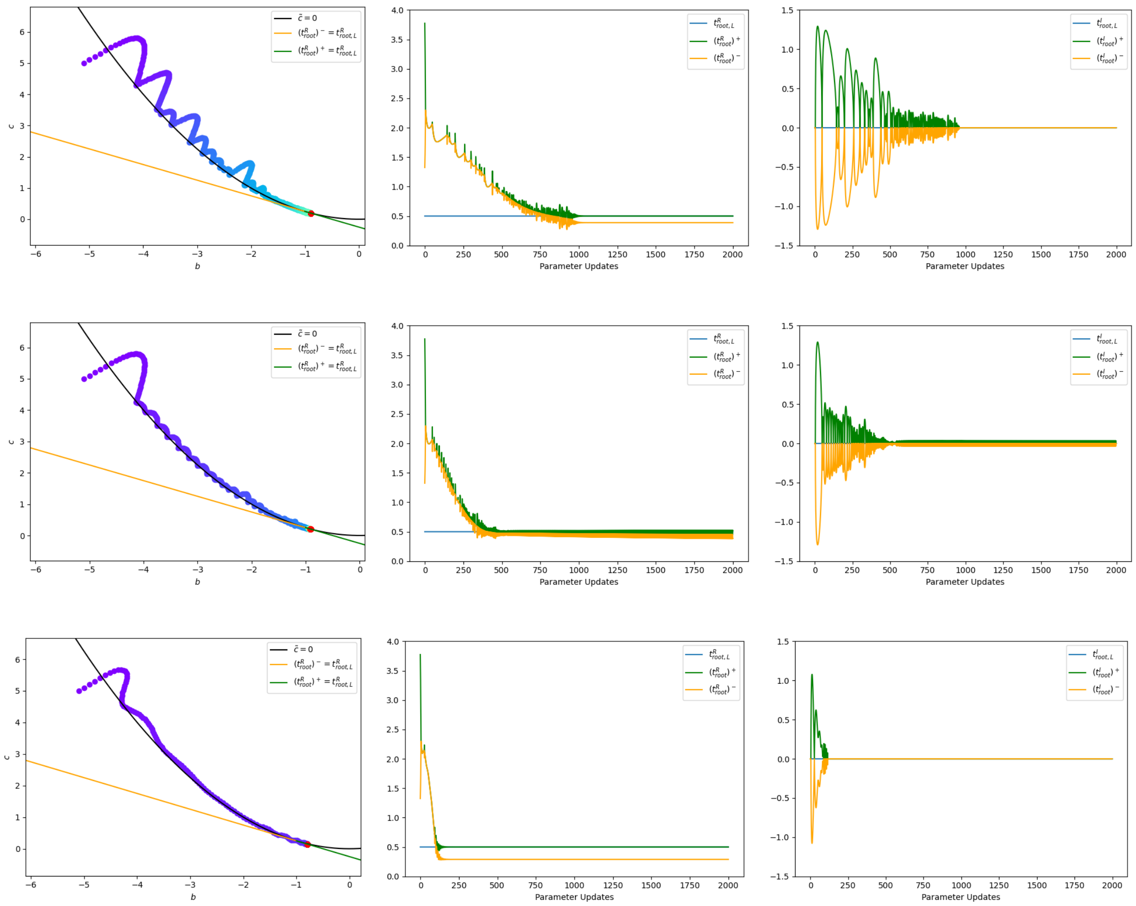


Figure 17: This example uses the same initial conditions as Figure 5, the root solver from Section 9.1, and equation 72 with robust division with $M = 10000$ (top row), $M = 10$ (middle row), and $M = 1$ (bottom row). The lack of convergence in the middle row is caused by the iterates settling down towards the left endpoint of the green ray near the $\tilde{c} = 0$ parabola (as can be seen by the nonzero imaginary part of the roots). See also Figure 19.

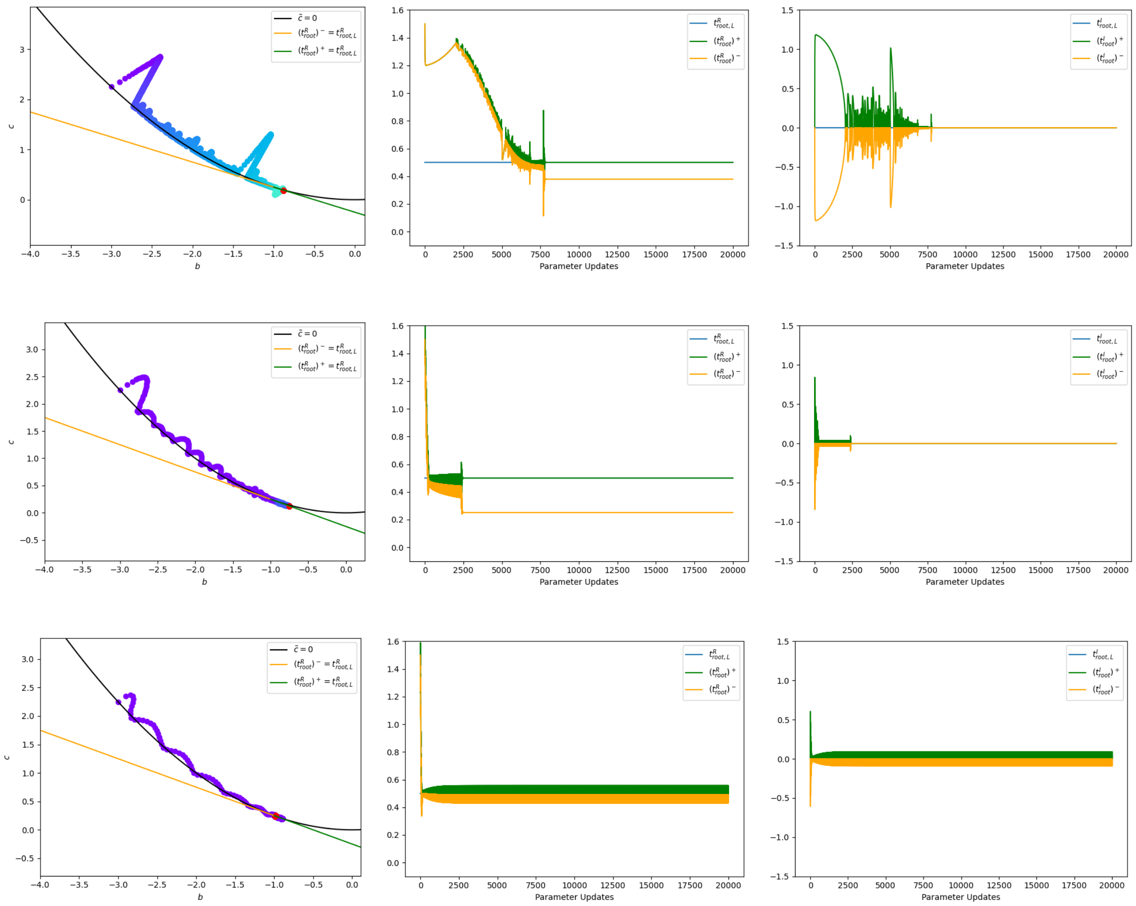


Figure 18: This example uses the same initial conditions as Figure 10, the root solver from Section 9.1, and equation 72 with robust division with $M = 10000$ (top row), $M = 10$ (middle row), and $M = 1$ (bottom row). The lack of convergence in the bottom row is caused by the iterates settling down towards the left endpoint of the green ray near the $\tilde{c} = 0$ parabola (as can be seen by the nonzero imaginary part of the roots). See also Figure 19.

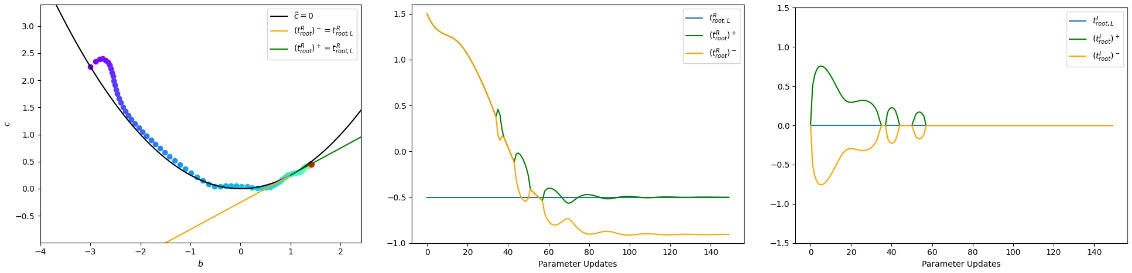


Figure 19: This example uses the same initial conditions as Figure 10, the root solver from Section 9.1, and equation 72 with robust division with $M = .01$. Here, we change $t^R_{\text{root,L}} = \frac{1}{2}$ to $t^R_{\text{root,L}} = -\frac{1}{2}$ so that the direction orthogonal to the $\tilde{c} = 0$ parabola tends to point towards the right (instead of towards the left) in the region of interest (near the left endpoint of the green ray). Even with very aggressive clamping in the robust division (using $M = .01$) to limit momentum accumulation in Adam, the iterates still safely converge further to the right on the green ray away from the $\tilde{c} = 0$ parabola.

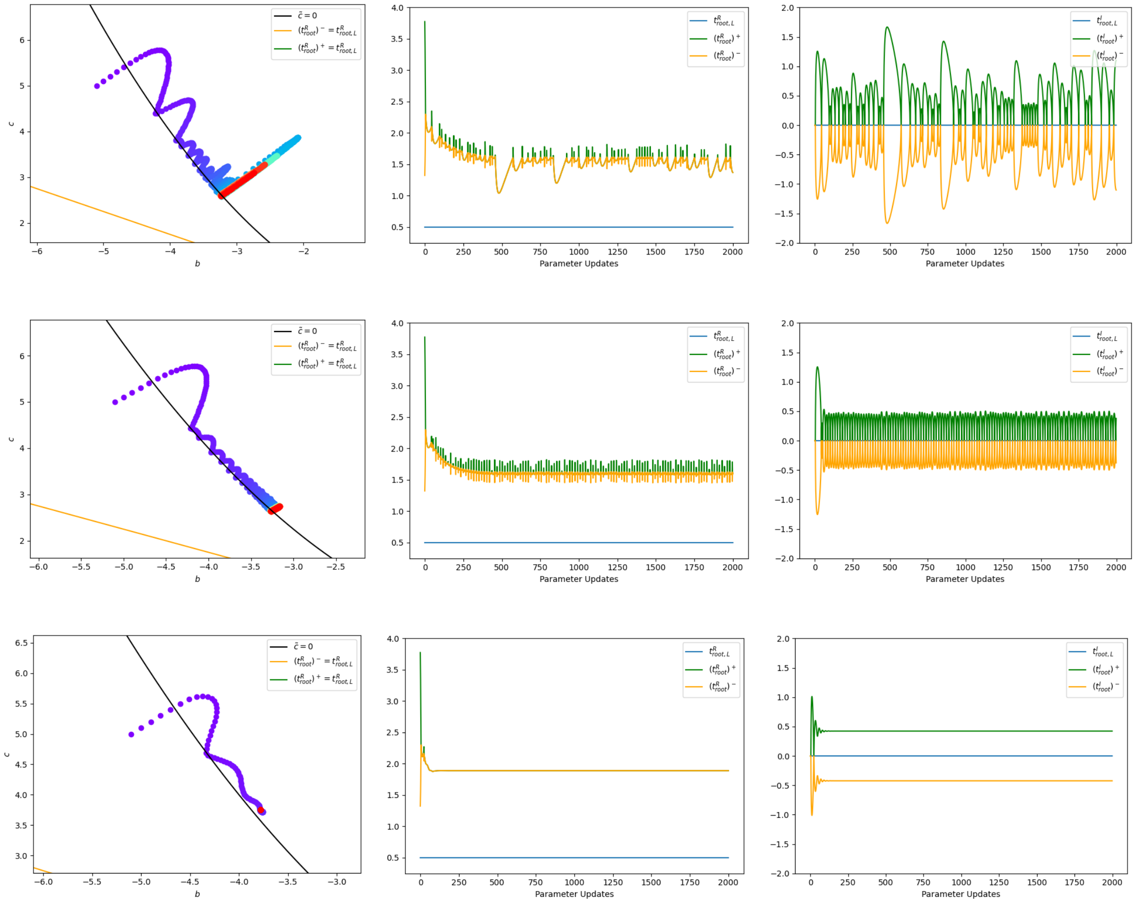


Figure 20: This example uses the same initial conditions as the bottom row of Figure 13, the root solver from Section 9.1, and equation 72 with robust division with $M = 10000$ (top row), $M = 10$ (middle row), and $M = 1$ (bottom row).

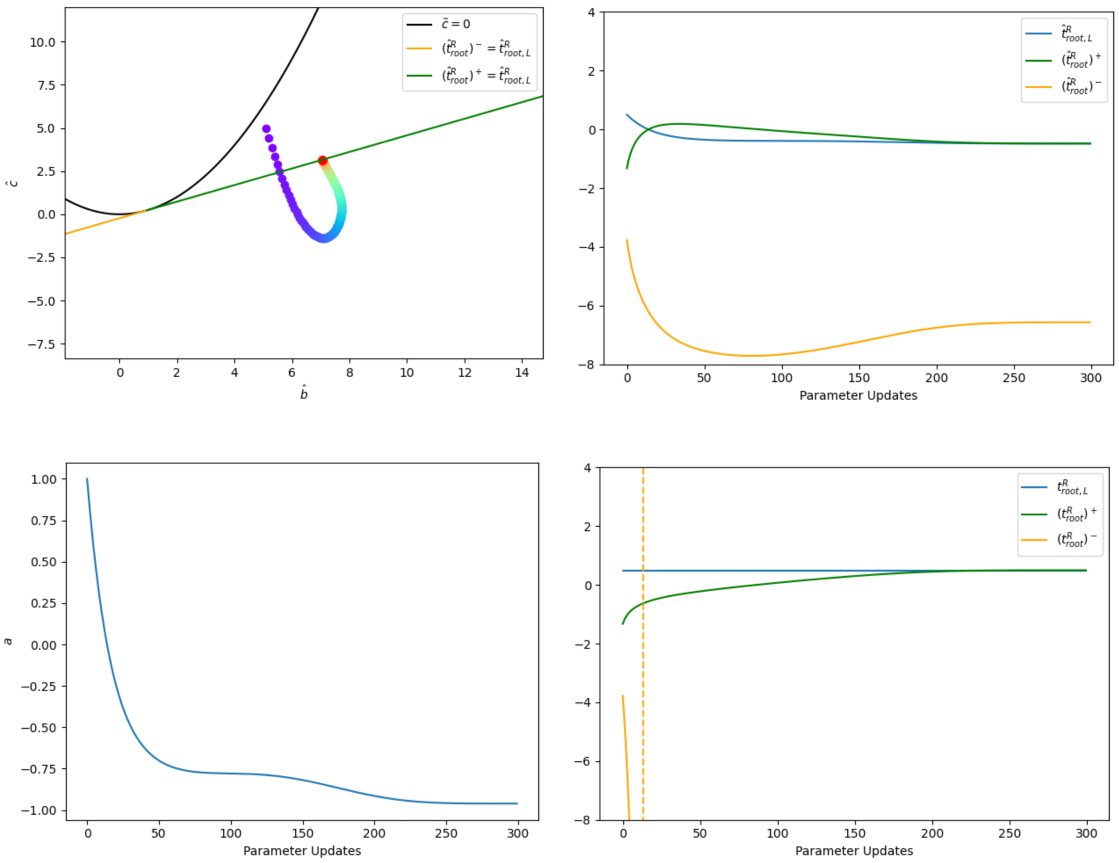


Figure 21: This example demonstrates what happens when $a \to 0$ and subsequently changes sign. Even though $(\hat{t}^R_{\text{root}})^{\pm}$ are well behaved, $(t^R_{\text{root}})^- \to -\infty$ before jumping discontinuously towards $+\infty$ when a changes sign; meanwhile, $(\hat{t}^R_{\text{root}})^+ \to \hat{t}^R_{\text{root,L}}$ and $(t^R_{\text{root}})^+ \to t^R_{\text{root,L}}$ as expected (see Section 9.2). Note that the dotted yellow line in the lower right subfigure does not include any actual values of $(t^R_{\text{root}})^-$, but is shown to indicate the jump from large negative values to large positive values.

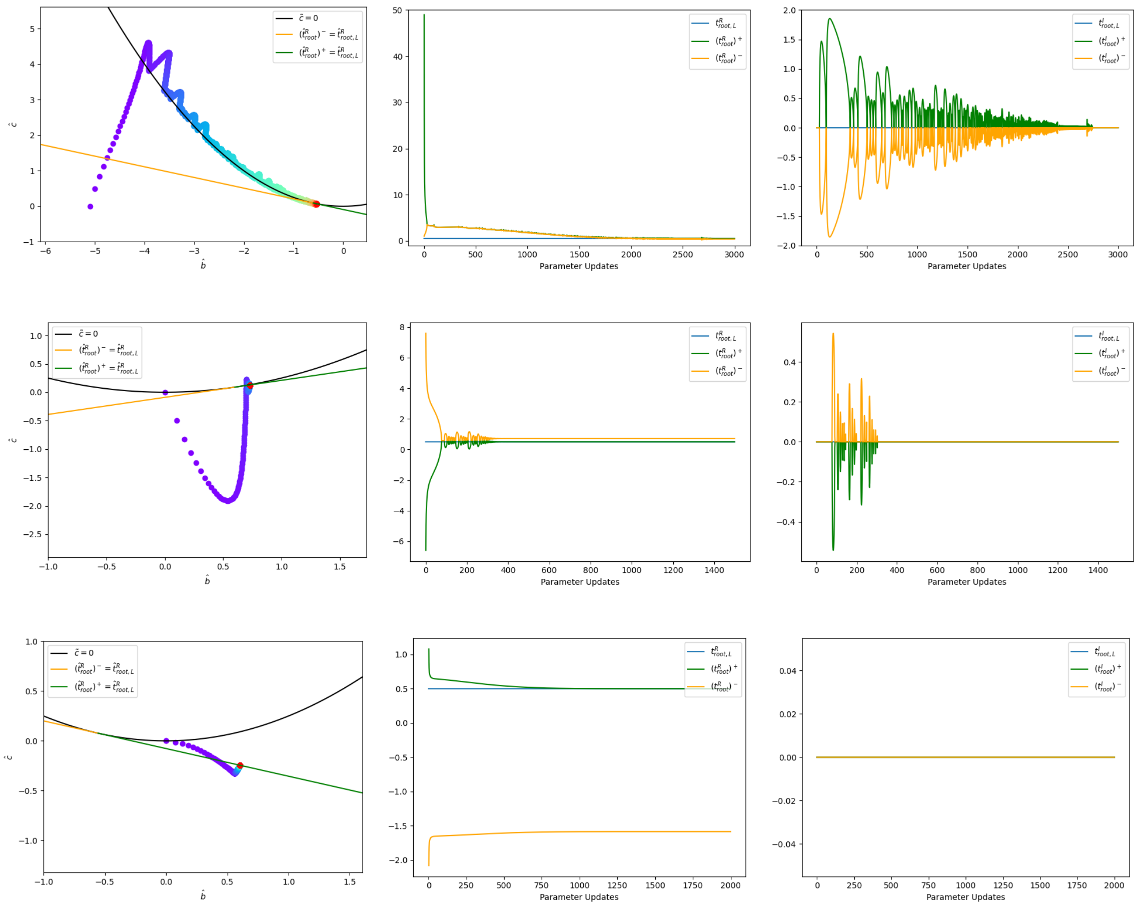


Figure 22: For this example, we set the robust division parameter $M = 1000$. The top row starts with $[a, b, c]^T = [0, -5.1, 5]^T$, the middle row starts with $[a, b, c]^T = [0, 0, 5]^T$, and the bottom row starts with $[a, b, c]^T = [0, 0, 0]^T$.

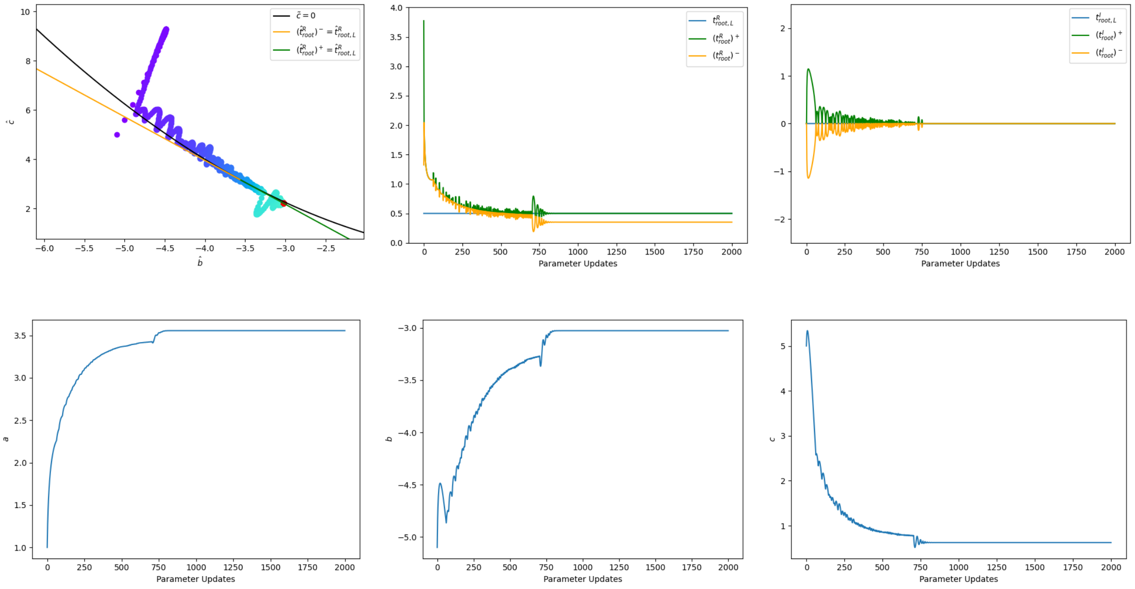


Figure 23: This figure repeats the example shown in Figures 5 and 17 (using $M = 1000$).

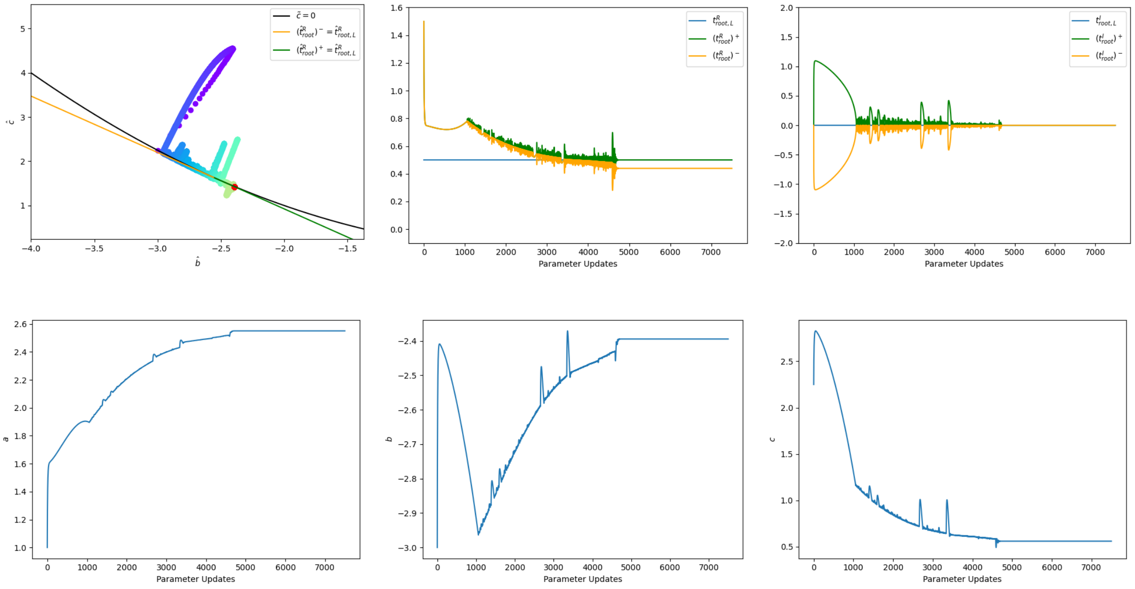


Figure 24: This figure repeats the example shown in Figures 10 and 18 (with $M = 1000$). Note that Figures 10 and 18 show 20000 iterations, whereas only 7500 iterations are shown here (for the sake of comparison with Figure 25).

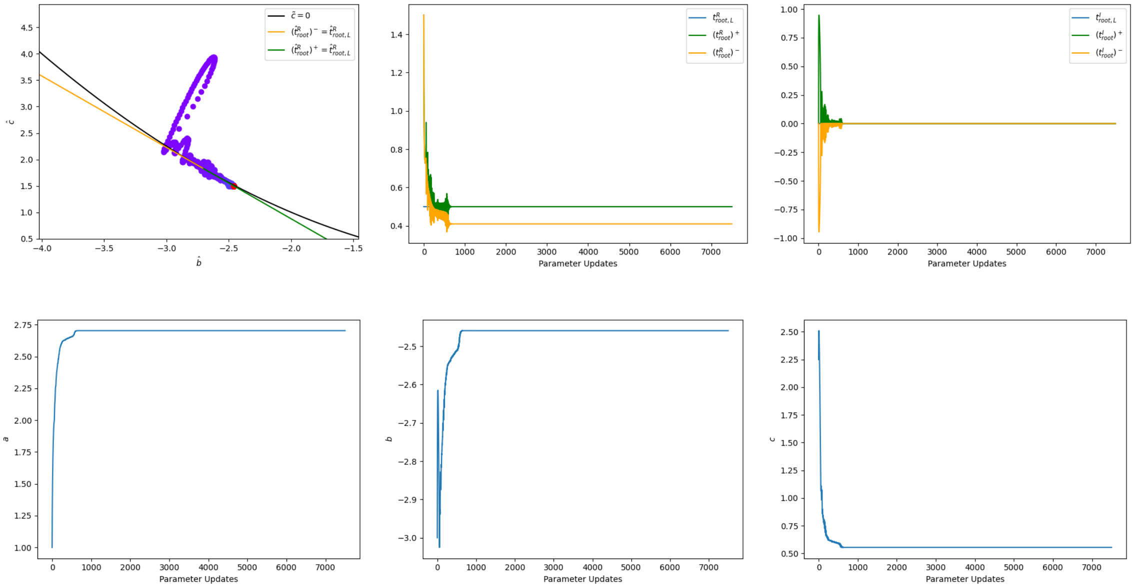


Figure 25: This figure repeats the example shown in Figure 24 but with $M = 10$ in order to demonstrate the accelerated convergence.

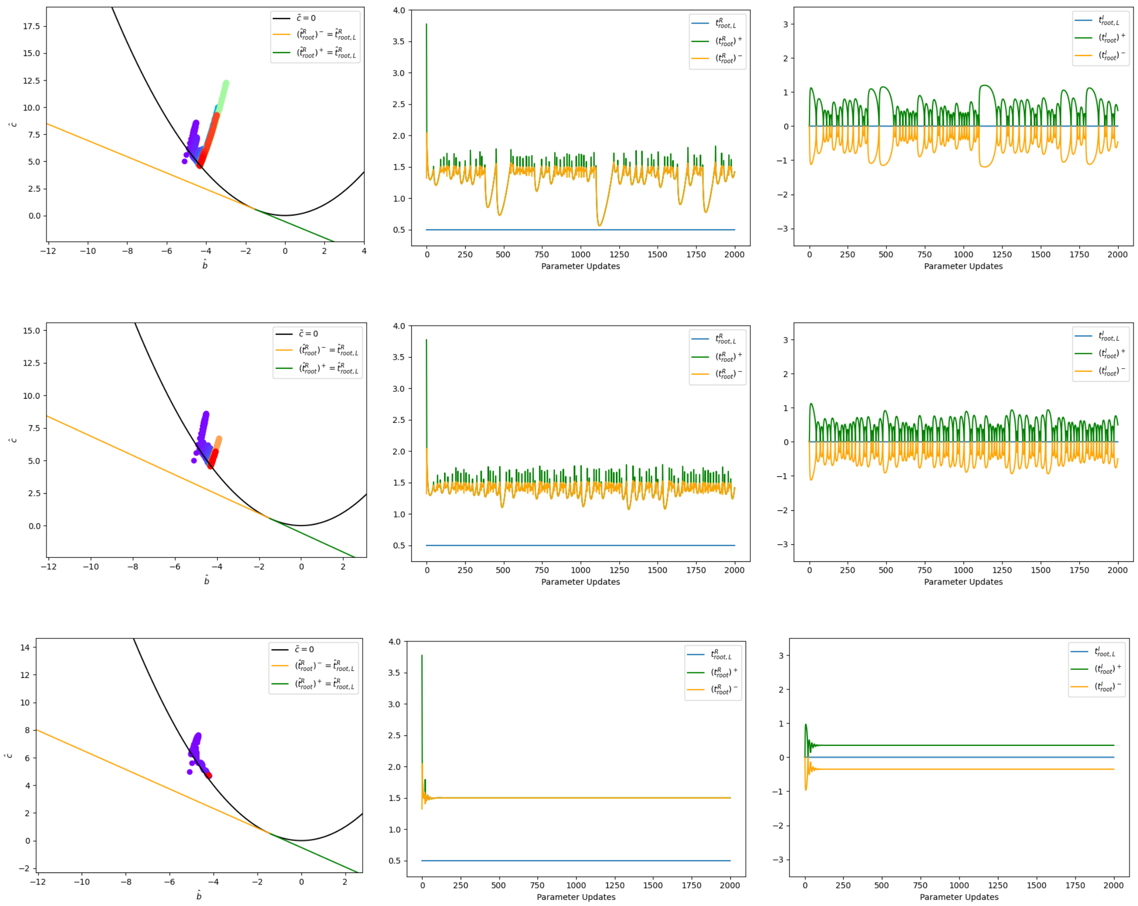


Figure 26: This figure repeats the example shown in Figures 13 (bottom row) and 20 using $M = 1000$ (top row), $M = 10$ (middle row), and $M = 1$ (bottom row).

10. Cubic Equations

In Section 10.1, we begin by introducing a novel and robust approach to finding roots (both real and complex) for cubic equations, even when any or all of the coefficients degenerate to be identically zero. In Section 10.2, we briefly address implicit differentiation. In Section 10.3, we analyze the cubic equation in its standard reduced canonical form where the roots are a function of only two parameters. In Section 10.4, we propose a new reduced canonical form, which enables a more robust treatment of both the roots and the derivatives of the roots with respect to the parameters. Finally, Section 11 details our proposed approach for cubic equations.

10.1. Cubic Root Solver

Whereas [44] only considered cubic roots that were in a pre-specified time interval (indicating a potential collision during a time step), we instead would like to find all roots (including complex roots and roots that blow up). If any of the coefficients has magnitude larger than 1, then we divide through by it; in addition, we flip the sign of all of the coefficients whenever $q < 0$. The resulting cubic equation can be written as $f(t,\vec{p}) = qt^3 + at^2 + bt + c$ where $0 \leq q \leq 1$, $|a| \leq 1$, $|b| \leq 1$, and $|c| \leq 1$.

Rewriting the cubic equation as

$$f(t,\vec{p}) = \frac{1}{3}t^2\left(qt + 3a\right) + \frac{1}{3}t\left(qt^2 + 3b\right) + \frac{1}{3}\left(qt^3 + 3c\right) = 0 \tag{74}$$

facilitates the computation of an interval $[t_{\text{left}}, t_{\text{right}}]$ containing the roots. t_{left} is determined by $t_{\text{left}} < 0$, $qt_{\text{left}} < -3a$, $q(t_{\text{left}})^2 > -3b$, $q(t_{\text{left}})^3 < -3c$, and t_{right} is determined by $t_{\text{right}} > 0$, $qt_{\text{right}} > -3a$, $q(t_{\text{right}})^2 > -3b$, $q(t_{\text{right}})^3 > -3c$. Choosing $t_{\text{left}} < -3/q$ and $t_{\text{right}} > 3/q$ is sufficient. When q is large enough, we proceed as follows: To determine monotonic intervals, we examine the roots of the derivative $3qt^2 + 2at + b = 0$. When these critical points are complex or repeated (i.e. $a^2 - 3qb \leq 0$), the cubic is monotonically increasing with one (possibly repeated) real root (which can be found robustly with a mixture of Newton's method and bisection). Otherwise, the critical points can be computed via

$$t_{\text{crit,L}} = \begin{cases} \frac{-a-\sqrt{a^2-3qb}}{3q} & \text{if } a \geq 0 \\ \frac{b}{-a+\sqrt{a^2-3qb}} & \text{if } a < 0 \end{cases} \tag{75a}$$

$$t_{\text{crit,R}} = \begin{cases} \frac{-a+\sqrt{a^2-3qb}}{3q} & \text{if } a \leq 0 \\ \frac{b}{-a-\sqrt{a^2-3qb}} & \text{if } a > 0 \end{cases} \tag{75b}$$

to avoid cancellation error (see e.g. [41, 42, 43]). This should result in $t_{\text{left}} < t_{\text{crit,L}} < t_{\text{crit,R}} < t_{\text{right}}$; however, critical points can be discarded (leaving less intervals as candidates for sign changes) whenever numerical errors cause these conditions to be violated. The signs of $f(t_{\text{crit,L}})$ and $f(t_{\text{crit,R}})$ can be used to determine intervals that contain roots (which can be found robustly with a mixture of Newton's method and bisection). After finding one real root, r_1, the cubic can be factored as

$$(t - r_1)(qt^2 + (qr_1 + a)t + qr_1^2 + ar_1 + b) = 0 \tag{76}$$

so that the remaining roots are given by

$$r_{2,3} = \frac{-(qr_1 + a) \pm \sqrt{(qr_1 + a)^2 - 4q(qr_1^2 + ar_1 + b)}}{2q} \tag{77}$$

noting that the most robust way to find r_2 and r_3 from the quadratic factor in equation 76 is to use the robust quadratic root-solver in Section 9.1. When exactly two roots are found via the iterative solver, we recommend using the root that would give the largest positive number under the square root in equation 77 when factoring the cubic in equation 76. A similar strategy can be used even when three roots are found via the iterative solver, especially when one suspects that numerical errors may have led to less accurate than desired roots.

Remark 10.1: Given that the real roots (and thus r_1) are bounded by $\pm\frac{3}{q}$, equation 77 leads to a bound of $\pm\frac{2}{q}$ on the real part and $\pm\frac{\sqrt{38}}{2q}$ $\left(\approx \pm\frac{3.1}{q}\right)$ on the imaginary parts of the complex roots; thus, whenever a real root or a real/imaginary part of a complex root grows large, q must be small.

When q is too small to robustly use $[-\frac{3}{q}, \frac{3}{q}]$ to bound the roots, one could attempt to use a smaller interval such as $[-t_{\max}, +t_{\max}]$ and proceed with the approach outlined above. One could also replace t with t^{-1} to obtain a reversed cubic $ct^3 + bt^2 + at + q = 0$, flipping the sign of all of the coefficients if necessary so that $c \geq 0$. When c is too small to robustly use $[-\frac{3}{c}, \frac{3}{c}]$, one could (again) attempt to use a smaller interval. Note that either the original or the reversed cubic is guaranteed to have a root in the interval $[-1, 1]$, since the left endpoint evaluates to either $-q + a - b + c$ or $-c + b - a + q$ while the right endpoint evaluates to $q + a + b + c$ for both cubics. Note that we reverse the quadratic factor in equation 76 to be

$$(cr_1^2 + br_1 + a)t^2 + (cr_1 + b)t + c = 0 \tag{78}$$

whenever r_1 is obtained from the reversed cubic, in order to avoid the problematic case of having to flip an identically zero r_2 or r_3 (see Remark 9.8).

It is problematic to flip an identically zero r_1 from the reversed cubic, which should only occur when $q = 0$; however, when q is smaller than the tolerance of the iterative solver, one may also obtain an identically zero r_1. We address this as follows: As $q \to 0$, the cubic equation degenerates into a quadratic equation. Assuming that two of the roots are approximately governed by $at^2 + bt + c = 0$, one can uncover the behavior of the remaining root from

$$\frac{qt^3 + at^2 + bt + c}{at^2 + bt + c} = \frac{q}{a}\left(t + \frac{a}{q} - \frac{b}{a} + \frac{1}{a}\frac{(b^2 - ac)t + bc}{at^2 + bt + c}\right) = 0 \tag{79}$$

indicating that the remaining root r_1 blows up like $\frac{-a}{q}$ as $\frac{q}{a} \to 0$ via

$$r_1 = -\left(\frac{q}{a}\right)^{-1} + \frac{b}{a}\left(\frac{q}{a}\right)^{0} + \left(\frac{b^2}{a^2} - \frac{c}{a}\right)\left(\frac{q}{a}\right)^{1} + O\left(\left(\frac{q}{a}\right)^{2}\right) \tag{80}$$

where the $O\left(\frac{q}{a}\right)$ term was obtained from the last term in equation 79 as $t \to \pm\infty$. Plugging this r_1 into equation 77 gives

$$r_{2,3} = -\frac{b}{2a} + O\left(\frac{q}{a}\right) \pm \sqrt{\frac{b^2 - 4ac}{(2a)^2} + O\left(\frac{q}{a}\right)} \tag{81}$$

as expected. Note that equation 81 is only a valid approximation as $\frac{q}{a} \to 0$ when all of the $\frac{b}{a}$ and $\frac{c}{a}$ terms are robustly bounded even when raised to various powers. In summary, when $q = 0$ identically (or the iterative solver returns $r_1 = 0$ from the reversed cubic), we use the quadratic root solver (from Section 9.1) on $at^2 + bt + c = 0$ to find r_2 and r_3 (i.e. consistent with equation 78); then, we choose $r_1 = -\text{Sign}(a)\infty$ (clamped to a maximum magnitude) consistent with $q > 0$ in equation 80. When $a = 0$, the quadratic solver determines a pseudo-sign for a (and thus $\text{Sign}(a)$) except when $a = b = c = 0$ where we choose $r_1 = 0$ as a triply repeated root.

Remark 10.2: Notably, one would not want to be compelled into making the code for this cubic solver differentiable.

10.2. Implicit Differentiation

Similar to Section 3.1, we write the cubic equation as

$$f(t_{\text{root}}; \vec{p}) = \begin{bmatrix} q(t^R_{\text{root}})^3 - 3qt^R_{\text{root}}(t^I_{\text{root}})^2 + a(t^R_{\text{root}})^2 - a(t^I_{\text{root}})^2 + bt^R_{\text{root}} + c \\ -q(t^I_{\text{root}})^3 + 3q(t^R_{\text{root}})^2 t^I_{\text{root}} + 2at^R_{\text{root}} t^I_{\text{root}} + bt^I_{\text{root}} \end{bmatrix} = \vec{0} \tag{82}$$

letting θ_1 refer to the first variable (i.e. t_{root}) and θ_2 refer to the second variable (i.e. $\vec{p}$) so that the derivatives

$$f_{\theta_1}(t_{\text{root}};\vec{p}) = \begin{bmatrix} 3q(t^R_{\text{root}})^2 - 3q(t^I_{\text{root}})^2 + 2at^R_{\text{root}} + b & -(6qt^R_{\text{root}}t^I_{\text{root}} + 2at^I_{\text{root}}) \\ 6qt^R_{\text{root}}t^I_{\text{root}} + 2at^I_{\text{root}} & 3q(t^R_{\text{root}})^2 - 3q(t^I_{\text{root}})^2 + 2at^R_{\text{root}} + b \end{bmatrix} \tag{83a}$$

$$f^{-1}_{\theta_1}(t_{\text{root}};\vec{p}) = \frac{1}{s}\begin{bmatrix} 3q(t^R_{\text{root}})^2 - 3q(t^I_{\text{root}})^2 + 2at^R_{\text{root}} + b & 6qt^R_{\text{root}}t^I_{\text{root}} + 2at^I_{\text{root}} \\ -(6qt^R_{\text{root}}t^I_{\text{root}} + 2at^I_{\text{root}}) & 3q(t^R_{\text{root}})^2 - 3q(t^I_{\text{root}})^2 + 2at^R_{\text{root}} + b \end{bmatrix} \tag{83b}$$

$$\text{where } s = (3q(t^R_{\text{root}})^2 - 3q(t^I_{\text{root}})^2 + 2at^R_{\text{root}} + b)^2 + (6qt^R_{\text{root}}t^I_{\text{root}} + 2at^I_{\text{root}})^2$$

$$f_{\theta_2}(t_{\text{root}};\vec{p}) = \begin{bmatrix} Re(z^3_{\text{root}}) & Re(z^2_{\text{root}}) & Re(z^1_{\text{root}}) & Re(z^0_{\text{root}}) \\ Im(z^3_{\text{root}}) & Im(z^2_{\text{root}}) & Im(z^1_{\text{root}}) & Im(z^0_{\text{root}}) \end{bmatrix} \tag{83c}$$

have compact notation. The total derivative of equation 82 is $f_{\theta_1}(t_{\text{root}},\vec{p})d\theta_1 + f_{\theta_2}(t_{\text{root}},\vec{p})d\theta_2 = 0$, which can be written as $d\theta_1 = -f^{-1}_{\theta_1}(t_{\text{root}},\vec{p})f_{\theta_2}(t_{\text{root}},\vec{p})d\theta_2$ or

$$dt_{\text{root}} = -f^{-1}_{\theta_1}(t_{\text{root}},\vec{p})\begin{bmatrix} Re(z^3_{\text{root}}) & Re(z^2_{\text{root}}) & Re(z^1_{\text{root}}) & Re(z^0_{\text{root}}) \\ Im(z^3_{\text{root}}) & Im(z^2_{\text{root}}) & Im(z^1_{\text{root}}) & Im(z^0_{\text{root}}) \end{bmatrix} d\vec{p} \tag{84}$$

when f_{θ_1} is invertible.

10.3. Understanding the Asymptotics

Similar to completing the square for the quadratic equation, one can make a change of variables $\tilde{t} = t + \frac{a}{3q}$ to obtain

$$q\tilde{t}^3 + \left(\frac{3bq - a^2}{3q}\right)\tilde{t} + \left(\frac{2a^3 - 9abq + 27cq^2}{27q^2}\right) = 0 \tag{85}$$

which can be reduced to a two parameter family by dividing by q and setting $\hat{b} = \frac{3bq-a^2}{3q^2}$ and $\hat{c} = \frac{2a^3-9abq+27cq^2}{27q^3}$ to obtain $\tilde{t}^3 + \hat{b}\tilde{t} + \hat{c} = 0$.

Remark 10.3: Since $(\tilde{t} - r_1)(\tilde{t} - r_2)(\tilde{t} - r_3) = \tilde{t}^3 - (r_1 + r_2 + r_3)\tilde{t}^2 + (r_1r_2 + r_1r_3 + r_2r_3)\tilde{t} - r_1r_2r_3$, $r_1 + r_2 + r_3 = 0$ *for this reduced cubic; furthermore, since any imaginary parts cancel, the sum of the real parts is identically zero.*

The critical points can be found from $3\tilde{t}^2 + \hat{b} = 0$ as $\tilde{t} = \pm\sqrt{\frac{-\hat{b}}{3}}$. In order for there to be three real roots, the reduced cubic must be non-negative at $-\sqrt{\frac{-\hat{b}}{3}}$ and non-positive at $\sqrt{\frac{-\hat{b}}{3}}$, i.e. $-\left(\sqrt{\frac{-\hat{b}}{3}}\right)^3 - \hat{b}\sqrt{\frac{-\hat{b}}{3}} + \hat{c} \geq 0$ and $\left(\sqrt{\frac{-\hat{b}}{3}}\right)^3 + \hat{b}\sqrt{\frac{-\hat{b}}{3}} + \hat{c} \leq 0$ which lead to $\hat{c} \geq -2\left(\frac{-\hat{b}}{3}\right)^{3/2}$ and $\hat{c} \leq 2\left(\frac{-\hat{b}}{3}\right)^{3/2}$ respectively. Graphs of the boundary curves in the valid $\hat{b} \leq 0$ region are shown in Figure 27, and the region where both inequalities are strictly valid is shown in green. Excluding the origin (in Figure 27) where zero is a triply repeated root, the smallest of the three real roots is always strictly negative and the largest is always strictly positive. As the cubic becomes negative at $-\sqrt{\frac{-\hat{b}}{3}}$ crossing from the green to the red region, the middle and negative roots merge to become complex while the positive root remains real; similarly, crossing from the green to the blue region merges the two larger roots. The boundary between the red and the blue regions occurs when the single real root is zero, which requires $\hat{c} = 0$.

For the reduced cubic, equation 83b has $s = (3(\tilde{t}^R_{\text{root}})^2 - 3(\tilde{t}^I_{\text{root}})^2 + \hat{b})^2 + (6\tilde{t}^R_{\text{root}}\tilde{t}^I_{\text{root}})^2$ where $s = 0$ only when $\tilde{t}^I_{\text{root}} = 0$ and $3(\tilde{t}^R_{\text{root}})^2 + \hat{b} = 0$, i.e. when $\tilde{t}^R_{\text{root}} = \pm\sqrt{\frac{-\hat{b}}{3}}$ (the boundaries of the green region). Note that the non-merging real root only takes on the value of a critical point when all three roots are merging (the origin in Figure 27); otherwise, $s \neq 0$ for a non-merging real root.

Consider a real root (with $\tilde{t}^I = 0$) where equation 84 reduces to

$$\begin{bmatrix} d\tilde{t}^R_{\text{root}} \\ d\tilde{t}^I_{\text{root}} \end{bmatrix} = -\frac{1}{3(\tilde{t}^R_{\text{root}})^2 + \hat{b}}\begin{bmatrix} \tilde{t}^R_{\text{root}} & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} d\hat{b} \\ d\hat{c} \end{bmatrix} \tag{86}$$

implying that $d\tilde{t}^I_{\text{root}}$ is identically zero. Let $\tilde{t}^{\pm}_{\text{root}}$ denote the largest/smallest real roots in the green region, noting that $\tilde{t}^{+}_{\text{root}}$ is also defined in the red region (including the red curve) and $\tilde{t}^{-}_{\text{root}}$ is also defined in the blue region (including the blue curve). $\tilde{t}^{+}_{\text{root}}$ always occurs where the function is increasing with derivative $3(\tilde{t}^{+}_{\text{root}})^2 + \hat{b} > 0$, except when it is merging where $3(\tilde{t}^{+}_{\text{root}})^2 + \hat{b} = 0$; similarly, $\tilde{t}^{-}_{\text{root}}$ also always occurs where the function is increasing, except when it is merging. Since the middle root has $-\sqrt{\frac{-\hat{b}}{3}} < \tilde{t}^{M}_{\text{root}} < \sqrt{\frac{-\hat{b}}{3}}$ except when it is merging, $3(\tilde{t}^{M}_{\text{root}})^2 + \hat{b} < 0$. Thus, $\frac{\partial \tilde{t}^{\pm}_{\text{root}}}{\partial \hat{c}} \to -\infty$ whenever $\tilde{t}^{\pm}_{\text{root}}$ is merging on a boundary of the green region, and $\frac{\partial \tilde{t}^{M}_{\text{root}}}{\partial \hat{c}} \to +\infty$ on all boundaries of the green region. On the red curve (excluding the origin), the merging roots are negative implying that $\frac{\partial \tilde{t}^{-}_{\text{root}}}{\partial \hat{b}} \to +\infty$ and $\frac{\partial \tilde{t}^{M}_{\text{root}}}{\partial \hat{b}} \to -\infty$; similarly, on the blue curve (excluding the origin), the merging roots are positive implying that $\frac{\partial \tilde{t}^{+}_{\text{root}}}{\partial \hat{b}} \to -\infty$ and $\frac{\partial \tilde{t}^{M}_{\text{root}}}{\partial \hat{b}} \to +\infty$. At the origin, $\frac{\partial \tilde{t}^{R}_{\text{root}}}{\partial \hat{b}}$ is a bit more complicated.

When $\hat{b} \le 0$, one can write $0 \le 3(\tilde{t}^{\pm}_{\text{root}})^2 + \hat{b} \le 3(\tilde{t}^{\pm}_{\text{root}})^2$ leading to

$$0 < \frac{1}{3\tilde{t}^{+}_{\text{root}}} \le \frac{\tilde{t}^{+}_{\text{root}}}{3(\tilde{t}^{+}_{\text{root}})^2 + \hat{b}} \tag{87a}$$

$$\frac{\tilde{t}^{-}_{\text{root}}}{3(\tilde{t}^{-}_{\text{root}})^2 + \hat{b}} \le \frac{1}{3\tilde{t}^{-}_{\text{root}}} < 0 \tag{87b}$$

implying that $\frac{\partial \tilde{t}^{\pm}_{\text{root}}}{\partial \hat{b}} \to \mp\infty$ respectively approaching the origin from anywhere with $\hat{b} \le 0$. When $\hat{b} > 0$, we choose curves of the form $\tilde{t}^{\pm}_{\text{root}} = \gamma \hat{b}^p$ with $p > 0$ to illustrate the behavior. See Figure 28 left. When $p > 1$, $\frac{\partial \tilde{t}^{\pm}_{\text{root}}}{\partial \hat{b}} \to 0$. When $p = 1$, $\frac{\partial \tilde{t}^{\pm}_{\text{root}}}{\partial \hat{b}} \to -\gamma$ (where $\gamma > 0$ is in the red region, $\gamma < 0$ is in the blue region, $\gamma = 0$ is the positive $\hat{b}$-axis). When $0 < p < 1$, $\frac{\partial \tilde{t}^{\pm}_{\text{root}}}{\partial \hat{b}} \to \mp\infty$ respectively. Thus, both $\frac{\partial \tilde{t}^{\pm}_{\text{root}}}{\partial \hat{b}}$ are nonremovable singularities at the origin. For $\tilde{t}^{M}_{\text{root}}$, we choose curves of the form $\tilde{t}^{M}_{\text{root}} = \gamma(-\hat{b})^p$. Choosing $p > \frac{1}{2}$ makes both $3p > \frac{3}{2}$ and $p+1 > \frac{3}{2}$, so that the curves are in the green region as they approach the origin. See Figure 28 right. On these curves, $3(\tilde{t}^{M}_{\text{root}})^2 = 3\gamma^2(-\hat{b})^{2p} < -\hat{b}$ as $\hat{b} \to 0$ for $p > \frac{1}{2}$, implying that $3(\tilde{t}^{M}_{\text{root}})^2 + \hat{b} < 0$ and thus that these curves do indeed represent the middle root as $\hat{b} \to 0$. When $p > 1$, $\frac{\partial \tilde{t}^{M}_{\text{root}}}{\partial \hat{b}} \to 0$. When $p = 1$, $\frac{\partial \tilde{t}^{M}_{\text{root}}}{\partial \hat{b}} \to \gamma$. When $\frac{1}{2} < p < 1$, $\frac{\partial \tilde{t}^{M}_{\text{root}}}{\partial \hat{b}} \to \text{Sign}(\gamma)\infty$. Thus, $\frac{\partial \tilde{t}^{M}_{\text{root}}}{\partial \hat{b}}$ is a nonremovable singularity at the origin.

Next, consider the case where the roots are complex with $\tilde{t}^{I}_{\text{root}} \ne 0$. The second line of equation 82 gives $-(\tilde{t}^{I}_{\text{root}})^2 + 3(\tilde{t}^{R}_{\text{root}})^2 + \hat{b} = 0$, implying that $0 < (\tilde{t}^{I}_{\text{root}})^2 = 3(\tilde{t}^{R}_{\text{root}})^2 + \hat{b}$. Plugging this into equation 83b leads to $s = 4(\tilde{t}^{I}_{\text{root}})^2 \left(12(\tilde{t}^{R}_{\text{root}})^2 + \hat{b}\right)$. Since $12(\tilde{t}^{R}_{\text{root}})^2 + \hat{b} \ge 3(\tilde{t}^{R}_{\text{root}})^2 + \hat{b} = (\tilde{t}^{I}_{\text{root}})^2 > 0$, $s \to 0$ implies $\tilde{t}^{I}_{\text{root}} \to 0$ (i.e. one is approaching the boundaries of the green region). Equation 84 can be written as

$$\begin{bmatrix} d\tilde{t}^{R}_{\text{root}} \\ d\tilde{t}^{I}_{\text{root}} \end{bmatrix} = -\frac{1}{2\left(12(\tilde{t}^{R}_{\text{root}})^2 + \hat{b}\right)} \begin{bmatrix} 2\tilde{t}^{R}_{\text{root}} & -1 \\ \frac{-6(\tilde{t}^{R}_{\text{root}})^2 - \hat{b}}{\pm\sqrt{3(\tilde{t}^{R}_{\text{root}})^2 + \hat{b}}} & \frac{-3\tilde{t}^{R}_{\text{root}}}{\pm\sqrt{3(\tilde{t}^{R}_{\text{root}})^2 + \hat{b}}} \end{bmatrix} \begin{bmatrix} d\hat{b} \\ d\hat{c} \end{bmatrix} \tag{88}$$

using $(\tilde{t}^{I}_{\text{root}})^2 = 3(\tilde{t}^{R}_{\text{root}})^2 + \hat{b}$. $d\tilde{t}^{R}_{\text{root}}$ is the same for both complex conjugates, while $d\tilde{t}^{I}_{\text{root}}$ differs only in sign. Approaching the red and blue curves (away from the origin), $\frac{\partial \tilde{t}^{R}_{\text{root}}}{\partial \hat{b}}$ and $\frac{\partial \tilde{t}^{R}_{\text{root}}}{\partial \hat{c}}$ are well-behaved while $\frac{\partial \tilde{t}^{I}_{\text{root}}}{\partial \hat{b}} \to \pm\infty$ and $\frac{\partial \tilde{t}^{I}_{\text{root}}}{\partial \hat{c}} \to \pm\infty$ depending on which complex conjugate root (and which curve in the $\frac{\partial \tilde{t}^{I}_{\text{root}}}{\partial \hat{c}}$ case) is being considered.

Since $12(\tilde{t}^{R}_{\text{root}})^2 + \hat{b} > 0$, $\frac{\partial \tilde{t}^{R}_{\text{root}}}{\partial \hat{c}} \to +\infty$ approaching the origin. For $\frac{\partial \tilde{t}^{I}_{\text{root}}}{\partial \hat{b}}$, note that $\frac{6(\tilde{t}^{R}_{\text{root}})^2 + \hat{b}}{12(\tilde{t}^{R}_{\text{root}})^2 + \hat{b}} \le 1$. When $\hat{b} \ge 0$, $\frac{6(\tilde{t}^{R}_{\text{root}})^2 + \hat{b}}{12(\tilde{t}^{R}_{\text{root}})^2 + \hat{b}} \ge \frac{6(\tilde{t}^{R}_{\text{root}})^2 + \hat{b}}{12(\tilde{t}^{R}_{\text{root}})^2 + 2\hat{b}} = \frac{1}{2}$. When $\hat{b} \le 0$, $\frac{6(\tilde{t}^{R}_{\text{root}})^2 + \hat{b}}{12(\tilde{t}^{R}_{\text{root}})^2 + \hat{b}} > \frac{3(\tilde{t}^{R}_{\text{root}})^2}{12(\tilde{t}^{R}_{\text{root}})^2} = \frac{1}{4}$ using $3(\tilde{t}^{R}_{\text{root}})^2 + \hat{b} > 0$. Together, all of this implies that $\frac{1}{4} < \pm 2\sqrt{3(\tilde{t}^{R}_{\text{root}})^2 + \hat{b}}\frac{\partial \tilde{t}^{I}_{\text{root}}}{\partial \hat{b}} \le 1$; thus, $\frac{\partial \tilde{t}^{I}_{\text{root}}}{\partial \hat{b}} \to \pm\infty$ approaching the origin (and thus on all boundaries of the green region).

Next, consider $\tilde{t}^{R}_{\text{root}} = \gamma \hat{b}^p$ with $\hat{b} > 0$. Writing $\hat{c} = -(\tilde{t}^{R}_{\text{root}})^3 + 3\tilde{t}^{R}_{\text{root}}(\tilde{t}^{I}_{\text{root}})^2 - \hat{b}\tilde{t}^{R}_{\text{root}} = 8(\tilde{t}^{R}_{\text{root}})^3 + 2\hat{b}\tilde{t}^{R}_{\text{root}}$

leads to $\hat{c} = 8\gamma^3\hat{b}^{3p} + 2\gamma\hat{b}^{p+1}$. When $p > \frac{1}{2}$,

$$\frac{\partial \tilde{t}^R_{\text{root}}}{\partial \hat{b}} \to \frac{-\gamma\hat{b}^{p-\frac{1}{2}}}{\sqrt{\hat{b}}} \tag{89a}$$

$$\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \hat{c}} \to \frac{\pm 3\gamma\hat{b}^{p-\frac{1}{2}}}{2\hat{b}} \tag{89b}$$

as $\hat{b} \to 0$. When $0 < p < \frac{1}{2}$,

$$\frac{\partial \tilde{t}^R_{\text{root}}}{\partial \hat{b}} \to \frac{-1}{12\gamma\hat{b}^p} \tag{90a}$$

$$\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \hat{c}} \to \frac{\pm 1}{8\gamma\hat{b}^{2p}\sqrt{3\gamma^2}} \tag{90b}$$

as $\hat{b} \to 0$. When $p > 1$, $\frac{\partial \tilde{t}^R_{\text{root}}}{\partial \hat{b}} \to 0$. When $p = 1$, $\frac{\partial \tilde{t}^R_{\text{root}}}{\partial \hat{b}} \to -\gamma$. When $0 < p < 1$, $\frac{\partial \tilde{t}^R_{\text{root}}}{\partial \hat{b}} \to \text{Sign}(-\gamma)\infty$. Thus, $\frac{\partial \tilde{t}^R_{\text{root}}}{\partial \hat{b}}$ is a nonremovable singularity. When $p > \frac{3}{2}$, $\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \hat{c}} \to 0$. When $p = \frac{3}{2}$, $\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \hat{c}} \to \pm\frac{3\gamma}{2}$. When $0 < p < \frac{3}{2}$, $\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \hat{c}} \to \pm\text{Sign}(\gamma)\infty$. Thus, $\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \hat{c}}$ is a nonremovable singularity. For completeness, consider $\tilde{t}^R_{\text{root}} = \gamma(-\hat{b})^p$ with $\hat{b} < 0$ where $\hat{c} = 8\gamma^3(-\hat{b})^{3p} - 2\gamma(-\hat{b})^{p+1}$. Choosing $0 < p < \frac{1}{2}$ makes $3p < \frac{3}{2}$ and $p + 1 < \frac{3}{2}$, so that the curves are outside of the green region as they approach the origin. In this case, equation 90 would have $\hat{b}^p$ and $\hat{b}^{2p}$ replaced with $(-\hat{b})^p$ and $(-\hat{b})^{2p}$ respectively; thus $\frac{\partial \tilde{t}^R_{\text{root}}}{\partial \hat{b}} \to \text{Sign}(-\gamma)\infty$ and $\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \hat{c}} \to \pm\text{Sign}(\gamma)\infty$ as $\hat{b} \to 0$.

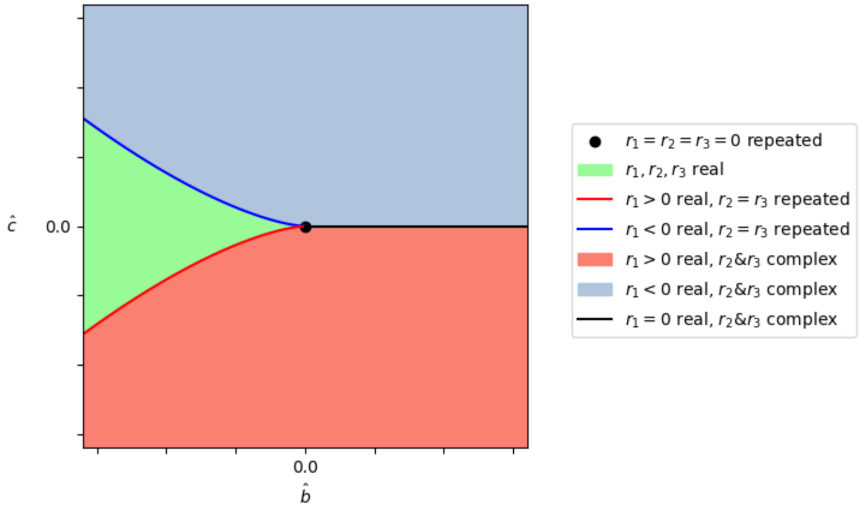


Figure 27: Root classification for the reduced cubic.

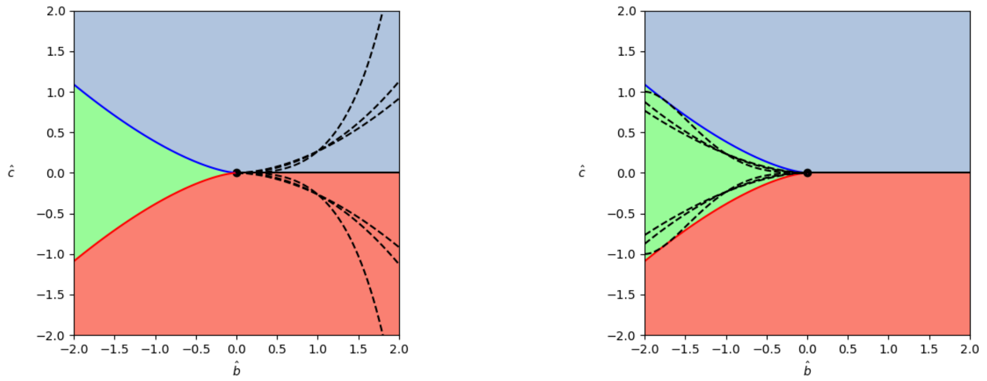


Figure 28: (Left) Graphs of $\tilde{t}^{\pm}_{\text{root}} = \gamma\hat{b}^p$ and thus $\hat{c} = -\gamma^3\hat{b}^{3p} - \gamma\hat{b}^{p+1}$ for $\hat{b} > 0$, $\gamma = \pm.25$, and $p = .75, 1, 2$ (black dotted lines). (Right) Graphs of $\tilde{t}^{M}_{\text{root}} = \gamma(-\hat{b})^p$ and thus $\hat{c} = -\gamma^3(-\hat{b})^{3p} + \gamma(-\hat{b})^{p+1}$ for $\hat{b} < 0$, $\gamma = \pm.25$, and $p = .75, 1, 2$ (black dotted lines).

10.4. A New Canonical Form

We begin with a change of variables robust to $q \to 0$,

$$\tilde{t} = qt + \begin{bmatrix} \frac{a}{3} \\ 0 \end{bmatrix} \tag{91a}$$

$$\begin{bmatrix} \tilde{t}^R \\ \tilde{t}^I \end{bmatrix} = q \begin{bmatrix} t^R \\ t^I \end{bmatrix} + \begin{bmatrix} \frac{a}{3} \\ 0 \end{bmatrix} \tag{91b}$$

giving $\tilde{t}^3 + \hat{b}\tilde{t} + \hat{c} = 0$ with $\hat{b} = -\frac{a^2}{3} + qb$ and $\hat{c} = \frac{2a^3}{27} - \frac{qab}{3} + q^2 c$. In spite of the fact that $\hat{b}$, $\hat{c}$, and $\tilde{t}$ are a different function of q, a, b, c, and t than they were in Section 10.3, the analysis in Section 10.3 is still valid since it only depended on the cubic equation having the form $\tilde{t}^3 + \hat{b}\tilde{t} + \hat{c} = 0$. In this section, we consider a further change of variables $\tilde{b} = \hat{c}$ and $\tilde{c} = \frac{\hat{b}^3}{27} + \frac{\hat{c}^2}{4}$ giving $\tilde{t}^3 - 3\left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{1}{3}}\tilde{t} + \tilde{b} = 0$.

The critical points can be found from $\tilde{t}^2 - \left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{1}{3}} = 0$ as $\tilde{t} = \pm\left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{1}{6}}$, implying that $\tilde{c} \le \frac{\tilde{b}^2}{4}$ is required for the critical points to exist. In order for there to be three real roots, the reduced cubic must be non-negative at $-\left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{1}{6}}$ and non-positive at $\left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{1}{6}}$, i.e. $-\left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{1}{2}} + 3\left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{1}{3}}\left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{1}{6}} + \tilde{b} = \sqrt{\tilde{b}^2 - 4\tilde{c}} + \tilde{b} \ge 0$ and $\left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{1}{2}} - 3\left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{1}{3}}\left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{1}{6}} + \tilde{b} = -\sqrt{\tilde{b}^2 - 4\tilde{c}} + \tilde{b} \le 0$; equivalently, $\tilde{c} \le 0$ when $\tilde{b} \le 0$ and $\tilde{c} \le 0$ when $\tilde{b} \ge 0$ respectively. Graphs of the boundary curves and the region where both inequalities are strictly valid are shown in Figure 29. Excluding the origin (in Figure 29), the smallest of the three real roots is always strictly negative and the largest is always strictly positive. Crossing from the green to the red region merges the middle and negative roots to become complex while the positive root remains real; similarly, crossing from the green to the blue region merges the two larger roots. The boundary between the red and the blue regions occurs when the single real root is zero, which requires $\tilde{b} = 0$.

It is worth briefly discussing the change of variables,

$$\tilde{b} = \hat{c} \tag{92a}$$

$$\tilde{c} = \frac{\hat{b}^3}{27} + \frac{\hat{c}^2}{4} \tag{92b}$$

$$\begin{bmatrix} d\tilde{b} \\ d\tilde{c} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{\hat{b}^2}{9} & \frac{\hat{c}}{2} \end{bmatrix} \begin{bmatrix} d\hat{b} \\ d\hat{c} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{2}{3}} & \frac{\tilde{b}}{2} \end{bmatrix} \begin{bmatrix} d\hat{b} \\ d\hat{c} \end{bmatrix} \tag{92c}$$

$$\hat{b} = -3\left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{1}{3}} \tag{92d}$$

$$\hat{c} = \tilde{b} \tag{92e}$$

$$\begin{bmatrix} d\hat{b} \\ d\hat{c} \end{bmatrix} = \frac{1}{\left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{2}{3}}} \begin{bmatrix} \frac{-\tilde{b}}{2} & 1 \\ \left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{2}{3}} & 0 \end{bmatrix} \begin{bmatrix} d\tilde{b} \\ d\tilde{c} \end{bmatrix} = \frac{9}{\hat{b}^2} \begin{bmatrix} \frac{-\hat{c}}{2} & 1 \\ \frac{\hat{b}^2}{9} & 0 \end{bmatrix} \begin{bmatrix} d\tilde{b} \\ d\tilde{c} \end{bmatrix} \tag{92f}$$

that leads from Figure 27 to Figure 29. Switching the roles of b and c leads to a rotation and reflection, allowing the sign of $\tilde{c}$ to be used to determine whether the roots are real or complex (similar to the role of $\tilde{c}$ for the reduced quadratic equation). Notably, the use of $\hat{b}^3$ in equation 92b causes compression towards the $\hat{b} = 0$ axis, which is subsequently transformed into a parabola by the $\hat{c}^2$ term; importantly, this also transforms the boundaries of the green region in Figure 27 to an independent axis in Figure 29. The inverse of a $\hat{b}^3$ mapping contains a cube root (see equation 92d), which has non-differentiable cusps at $\hat{b} = 0$. Although this leads to the Jacobian blowing up on the $\tilde{c} = \frac{\tilde{b}^2}{4}$ parabola, all of the interesting behavior we wish to address occurs near the $\tilde{b}$-axis (away from $\tilde{c} = \frac{\tilde{b}^2}{4}$) except for the origin which is highly problematic in any case due to nonremovable singularities.

Once again, equation 83b has $s = 0$ only on the boundaries of the green region where $\tilde{c} = 0$ (and only for merging roots). On this $\tilde{b}$-axis, the reduced cubic can be factored as

$$\left(\tilde{t} - \left(\frac{\tilde{b}}{2}\right)^{\frac{1}{3}}\right)^2 \left(\tilde{t} + 2\left(\frac{\tilde{b}}{2}\right)^{\frac{1}{3}}\right) = 0 \tag{93}$$

where $\tilde{t}^-_{\text{root}} = \tilde{t}^M_{\text{root}} = \left(\frac{\tilde{b}}{2}\right)^{\frac{1}{3}}$ and $\tilde{t}^+_{\text{root}} = -2\left(\frac{\tilde{b}}{2}\right)^{\frac{1}{3}}$ when $\tilde{b} \leq 0$, while $\tilde{t}^+_{\text{root}} = \tilde{t}^M_{\text{root}} = \left(\frac{\tilde{b}}{2}\right)^{\frac{1}{3}}$ and $\tilde{t}^-_{\text{root}} = -2\left(\frac{\tilde{b}}{2}\right)^{\frac{1}{3}}$ when $\tilde{b} \geq 0$.

For a real root (with $\tilde{t}^I = 0$), substituting equation 92f into equation 86 gives

$$\begin{bmatrix} d\tilde{t}^R_{\text{root}} \\ d\tilde{t}^I_{\text{root}} \end{bmatrix} = -\frac{1}{3(\tilde{t}^R_{\text{root}})^2 + \hat{b}}\frac{9}{\hat{b}^2}\begin{bmatrix} \frac{-\hat{c}}{2}\tilde{t}^R_{\text{root}} + \frac{\hat{b}^2}{9} & \tilde{t}^R_{\text{root}} \\ 0 & 0 \end{bmatrix}\begin{bmatrix} d\tilde{b} \\ d\tilde{c} \end{bmatrix} \tag{94}$$

where the problematic $3(\tilde{t}^R_{\text{root}})^2 + \hat{b} = 0$ corresponds to $\tilde{c} = 0$. Using the reduced cubic $\tilde{t}^3 + \hat{b}\tilde{t} + \hat{c} = 0$ to write

$$\frac{-\hat{c}}{2}\tilde{t}^R_{\text{root}} + \frac{\hat{b}^2}{9} = \left((\tilde{t}^R_{\text{root}})^3 + \hat{b}\tilde{t}^R_{\text{root}}\right)\frac{\tilde{t}^R_{\text{root}}}{2} + \frac{\hat{b}^2}{9} = \frac{1}{9}\left(3(\tilde{t}^R_{\text{root}})^2 + \hat{b}\right)\left(\frac{3(\tilde{t}^R_{\text{root}})^2}{2} + \hat{b}\right) \tag{95}$$

leads to

$$\frac{\partial\tilde{t}^R_{\text{root}}}{\partial\tilde{b}} = -\frac{1}{\hat{b}^2}\left(\frac{3(\tilde{t}^R_{\text{root}})^2}{2} + \hat{b}\right) \tag{96}$$

removing the need for L'Hospital's rule. This allows $\frac{\partial\tilde{t}^R_{\text{root}}}{\partial\tilde{b}}$ to be robustly evaluated on and near the $\tilde{b}$-axis, except near the origin. For $\frac{\partial\tilde{t}^R_{\text{root}}}{\partial\tilde{c}}$,

$$\frac{\partial\tilde{t}^R_{\text{root}}}{\partial\tilde{c}} = \frac{9}{\hat{b}^2}\frac{\partial\tilde{t}^R_{\text{root}}}{\partial\hat{b}} \tag{97}$$

implies that $\frac{\partial\tilde{t}^-_{\text{root}}}{\partial\tilde{c}} \to +\infty$ and $\frac{\partial\tilde{t}^M_{\text{root}}}{\partial\tilde{c}} \to -\infty$ approaching the red curve while $\frac{\partial\tilde{t}^+_{\text{root}}}{\partial\tilde{c}} \to -\infty$ and $\frac{\partial\tilde{t}^M_{\text{root}}}{\partial\tilde{c}} \to +\infty$ approaching the blue curve (similar to $\frac{\partial\tilde{t}^R_{\text{root}}}{\partial\tilde{b}}$).

Remark 10.4: This new canonical form no longer has the derivatives with respect to both reduced parameters blowing up as roots merge (as was the case for equation 86). Only derivatives with respect to $\tilde{c}$ blow up, while the derivatives with respect to $\tilde{b}$ remain bounded (away from the origin). Importantly, this enables one to readily move tangentially to the red and blue curves, changing the value of the roots without changing the closeness to merging.

Rewriting equation 96 as

$$\frac{\partial\tilde{t}^R_{\text{root}}}{\partial\tilde{b}} = -\frac{1}{2\hat{b}^2}\left(\left(3(\tilde{t}^R_{\text{root}})^2 + \hat{b}\right) + \hat{b}\right) \tag{98}$$

shows that $\frac{\partial\tilde{t}^M_{\text{root}}}{\partial\tilde{b}} \to +\infty$ approaching the origin, since $3(\tilde{t}^M_{\text{root}})^2 + \hat{b} \leq 0$ and $\hat{b} < 0$ implies that $\frac{\partial\tilde{t}^M_{\text{root}}}{\partial\tilde{b}} \geq \frac{-1}{2\hat{b}}$. For $\tilde{t}^\pm_{\text{root}}$, $3(\tilde{t}^\pm_{\text{root}})^2 + \hat{b} \geq 0$ and $\hat{b} < 0$ cancel when $3(\tilde{t}^R_{\text{root}})^2 + 2\hat{b} = 0$ or $\tilde{t}^R_{\text{root}} = \pm\sqrt{\frac{2}{3}}(-\hat{b})^{\frac{1}{2}}$. These are $\tilde{t}^R_{\text{root}} = \gamma(-\hat{b})^p$ curves with $\gamma = \pm\sqrt{\frac{2}{3}}$ and $p = \frac{1}{2}$ implying that $\hat{c} = \pm\sqrt{\frac{2}{27}}(-\hat{b})^{\frac{3}{2}}$, which can only be satisfied when $\tilde{c} = -\frac{\tilde{b}^2}{4}$ (see Figure 30). $\frac{\partial\tilde{t}^+_{\text{root}}}{\partial\tilde{b}}$ is identically zero on the $\tilde{b} > 0$ portion of $\tilde{c} = -\frac{\tilde{b}^2}{4}$, while $\frac{\partial\tilde{t}^-_{\text{root}}}{\partial\tilde{b}}$ is identically zero on the $\tilde{b} < 0$ portion of $\tilde{c} = -\frac{\tilde{b}^2}{4}$. Examining the roots determined from equation 93 leads to $\frac{\partial\tilde{t}^+_{\text{root}}}{\partial\tilde{b}} \to -\infty$ and $\frac{\partial\tilde{t}^-_{\text{root}}}{\partial\tilde{b}} \to +\infty$ approaching the origin on the red curve, while $\frac{\partial\tilde{t}^+_{\text{root}}}{\partial\tilde{b}} \to +\infty$ and $\frac{\partial\tilde{t}^-_{\text{root}}}{\partial\tilde{b}} \to -\infty$ approaching the origin on the blue curve. Thus, both $\frac{\partial\tilde{t}^\pm_{\text{root}}}{\partial\tilde{b}}$ are nonremovable singularities at the origin.

Remark 10.5: Choosing a perturbed $\tilde{t}^R_{root} = \pm\sqrt{\frac{2}{3}(-\hat{b})(1+\gamma\hat{b})}$ leads to $\frac{\partial\tilde{t}^R_{root}}{\partial\tilde{b}} = \gamma$ on the resulting $\hat{c} = \pm\frac{1}{3}(-\hat{b})\left(1 - 2\gamma\hat{b}\right)\sqrt{\frac{2}{3}(-\hat{b})(1+\gamma\hat{b})}$ curves. Note that $\gamma = 0$ reduces to the curves shown in Figure 30.

For $\frac{\partial\tilde{t}^R_{\text{root}}}{\partial\tilde{c}}$, equation 97 allows the results obtained from analyzing equation 86 to be utilized. Equation 87 shows that $\frac{\partial\tilde{t}^\pm_{\text{root}}}{\partial\tilde{c}} \to \mp\infty$ respectively when approaching the origin from anywhere with $\hat{b} < 0$ (i.e. $\tilde{c} < \frac{\tilde{b}^2}{4}$).

For $\tilde{t}^M_{\text{root}}$, consider $\tilde{t}^M_{\text{root}} = \gamma(-\hat{b})^p$. When $p > 3$, $\frac{\partial \tilde{t}^M_{\text{root}}}{\partial \tilde{c}} \to 0$. When $p = 3$, $\frac{\partial \tilde{t}^M_{\text{root}}}{\partial \tilde{c}} \to 9\gamma$. When $\frac{1}{2} < p < 3$, $\frac{\partial \tilde{t}^M_{\text{root}}}{\partial \tilde{c}} \to \text{Sign}(\gamma)\infty$. Thus, $\frac{\partial \tilde{t}^M_{\text{root}}}{\partial \tilde{c}}$ is a nonremovable singularity at the origin.

Next, consider the case where the roots are complex with $\tilde{t}^I_{\text{root}} \neq 0$. Equation 88 can be written as

$$\begin{bmatrix} d\tilde{t}^R_{\text{root}} \\ d\tilde{t}^I_{\text{root}} \end{bmatrix} = -\frac{1}{2\left(12(\tilde{t}^R_{\text{root}})^2 + \hat{b}\right)} \frac{9}{\hat{b}^2} \begin{bmatrix} -\hat{c}\tilde{t}^R_{\text{root}} - \frac{\hat{b}^2}{9} & 2\tilde{t}^R_{\text{root}} \\ \frac{1}{6}\frac{3\hat{c}\left(6(\tilde{t}^R_{\text{root}})^2+\hat{b}\right)-2\hat{b}^2\tilde{t}^R_{\text{root}}}{\pm\sqrt{3(\tilde{t}^R_{\text{root}})^2+\hat{b}}} & \frac{-6(\tilde{t}^R_{\text{root}})^2-\hat{b}}{\pm\sqrt{3(\tilde{t}^R_{\text{root}})^2+\hat{b}}} \end{bmatrix} \begin{bmatrix} d\tilde{b} \\ d\tilde{c} \end{bmatrix} \tag{99}$$

using equation 92f. Approaching the red and blue curves (away from the origin), $\frac{\partial \tilde{t}^R_{\text{root}}}{\partial \tilde{b}}$ and $\frac{\partial \tilde{t}^R_{\text{root}}}{\partial \tilde{c}}$ are well-behaved. Once again, the reduced cubic can be used to obtain $\hat{c} = 8(\tilde{t}^R_{\text{root}})^3 + 2\hat{b}\tilde{t}^R_{\text{root}}$, which leads to

$$3\hat{c}\left(6(\tilde{t}^R_{\text{root}})^2 + \hat{b}\right) - 2\hat{b}^2\tilde{t}^R_{\text{root}} = 6\tilde{t}^R_{\text{root}}\left(8(\tilde{t}^R_{\text{root}})^2 + \frac{2\hat{b}}{3}\right)\left(3(\tilde{t}^R_{\text{root}})^2 + \hat{b}\right) \tag{100}$$

and thus

$$\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \tilde{b}} = -\frac{1}{2\left(12(\tilde{t}^R_{\text{root}})^2 + \hat{b}\right)} \frac{9}{\hat{b}^2} \tilde{t}^R_{\text{root}} \left(8(\tilde{t}^R_{\text{root}})^2 + \frac{2\hat{b}}{3}\right)\left(\pm\sqrt{3(\tilde{t}^R_{\text{root}})^2 + \hat{b}}\right) \tag{101}$$

removing the need for L'Hospital's rule (except near the origin). For $\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \tilde{c}}$,

$$\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \tilde{c}} = \frac{9}{\hat{b}^2}\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \hat{b}} \tag{102}$$

implies that $\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \tilde{c}} \to \pm\infty$, depending on which complex conjugate root is being considered, on all boundaries of the green region.

Remark 10.6: Similar to the $\tilde{t}^I = 0$ case (see Remark 10.4), the $\tilde{t}^I \neq 0$ case also has derivatives with respect to $\tilde{b}$ remaining bounded (away from the origin).

Since $\hat{c}\tilde{t}^R_{\text{root}} \geq 0$ (see Figure 27 and Remark 10.3),

$$\frac{\partial \tilde{t}^R}{\partial \tilde{b}} = \frac{1}{2\left(12(\tilde{t}^R_{\text{root}})^2 + \hat{b}\right)}\left(\frac{9\hat{c}\tilde{t}^R_{\text{root}}}{\hat{b}^2} + 1\right) \geq \frac{1}{2\left(12(\tilde{t}^R_{\text{root}})^2 + \hat{b}\right)} \tag{103}$$

implying that $\frac{\partial \tilde{t}^R}{\partial \tilde{b}} \to +\infty$ approaching the origin (consistent with $\frac{\partial \tilde{t}^M_{\text{root}}}{\partial \tilde{b}}$ as expected, since $\tilde{t}^M_{\text{root}}$ is continuous with the real part of the complex roots across the $\tilde{b}$-axis). Considering only the $\hat{b} < 0$ (i.e. $\tilde{c} < \frac{\tilde{b}^2}{4}$) region,

$$\left|\frac{\partial \tilde{t}^R}{\partial \tilde{c}}\right| = \frac{9\left|\tilde{t}^R_{\text{root}}\right|}{\left(12(\tilde{t}^R_{\text{root}})^2 + \hat{b}\right)\hat{b}^2} > \frac{9\left|\tilde{t}^R_{\text{root}}\right|}{12(\tilde{t}^R_{\text{root}})^2\hat{b}^2} = \frac{9}{12\left|\tilde{t}^R_{\text{root}}\right|\hat{b}^2} \tag{104}$$

illustrating that $\frac{\partial \tilde{t}^R}{\partial \tilde{c}} \to +\infty$ approaching the origin from the red region (below $\tilde{c} = \frac{\tilde{b}^2}{4}$), while $\frac{\partial \tilde{t}^R}{\partial \tilde{c}} \to -\infty$ approaching the origin from the blue region (below $\tilde{c} = \frac{\tilde{b}^2}{4}$).

Consider equation 101 in the $\hat{b} < 0$ (i.e. $\tilde{c} < \frac{\tilde{b}^2}{4}$) region. Using $3(\tilde{t}^R_{\text{root}})^2 + \hat{b} \geq 0$ leads to $12(\tilde{t}^R_{\text{root}})^2 + \hat{b} \geq 11(\tilde{t}^R_{\text{root}})^2 + \frac{2\hat{b}}{3} > 8(\tilde{t}^R_{\text{root}})^2 + \frac{2\hat{b}}{3} > 0$ and thus $\frac{8(\tilde{t}^R_{\text{root}})^2+\frac{2\hat{b}}{3}}{12(\tilde{t}^R_{\text{root}})^2+\hat{b}} < 1$; in addition,

$$\frac{8(\tilde{t}^R_{\text{root}})^2 + \frac{2\hat{b}}{3}}{12(\tilde{t}^R_{\text{root}})^2 + \hat{b}} \geq \frac{5(\tilde{t}^R_{\text{root}})^2 - \frac{\hat{b}}{3}}{12(\tilde{t}^R_{\text{root}})^2 + \hat{b}} > \frac{5(\tilde{t}^R_{\text{root}})^2}{12(\tilde{t}^R_{\text{root}})^2} = \frac{5}{12} \tag{105}$$

again using $3(\tilde{t}^R_{\text{root}})^2 + \hat{b} \geq 0$. Equation 101 can be rewritten as

$$\frac{2\left(12(\tilde{t}^R_{\text{root}})^2 + \hat{b}\right)}{9\left(8(\tilde{t}^R_{\text{root}})^2 + \frac{2\hat{b}}{3}\right)}\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \tilde{b}} = \frac{-\tilde{t}^R_{\text{root}}}{\hat{b}^2}\left(\pm\sqrt{3(\tilde{t}^R_{\text{root}})^2 + \hat{b}}\right) \tag{106}$$

where the coefficient of $\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \tilde{b}}$ is bounded between $\frac{2}{9}$ and $\frac{8}{15}$. As discussed at the end of Section 10.3, choosing $\tilde{t}^R_{\text{root}} = \gamma(-\hat{b})^p$ with $\hat{b} < 0$ and $0 < p < \frac{1}{2}$ gives curves outside the green region as they approach the origin; on these curves,

$$\frac{-\tilde{t}^R_{\text{root}}}{\hat{b}^2}\left(\pm\sqrt{3(\tilde{t}^R_{\text{root}})^2 + \hat{b}}\right) \to \text{Sign}(\mp\gamma)\frac{\sqrt{3}(\tilde{t}^R_{\text{root}})^2}{\hat{b}^2} \to \text{Sign}(\mp\gamma)\infty \tag{107}$$

since $\hat{b} \to 0$ faster than $(\tilde{t}^R_{\text{root}})^2 \to 0$, i.e. $\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \tilde{b}} \to \text{Sign}(\mp\gamma)\infty$. Choosing a perturbed $\tilde{t}^R_{\text{root}} = \pm\sqrt{\frac{1}{3}(-\hat{b})(1 + 3\gamma^2\hat{b}^2)}$ in the blue/red regions respectively leads to

$$\hat{c} = \tilde{t}^R_{\text{root}}\left(8(\tilde{t}^R_{\text{root}})^2 + 2\hat{b}\right) = \pm 2\left(\frac{-\hat{b}}{3}\right)^{\frac{3}{2}}\left(1 + 12\gamma^2\hat{b}^2\right)\sqrt{1 + 3\gamma^2\hat{b}^2} \tag{108}$$

curves. When $\gamma = 0$, these are the boundary curves $\hat{c} = \pm 2\left(\frac{-\hat{b}}{3}\right)^{\frac{3}{2}}$; when $\gamma \neq 0$, the perturbed curves lie outside the green region. On these curves,

$$\frac{-\tilde{t}^R_{\text{root}}}{\hat{b}^2}\left(\pm\sqrt{3(\tilde{t}^R_{\text{root}})^2 + \hat{b}}\right) = \pm|\gamma|\sqrt{1 + 3\gamma^2\hat{b}^2} \to \pm|\gamma| \tag{109}$$

approaching the origin in the red region and

$$\frac{-\tilde{t}^R_{\text{root}}}{\hat{b}^2}\left(\pm\sqrt{3(\tilde{t}^R_{\text{root}})^2 + \hat{b}}\right) = \mp|\gamma|\sqrt{1 + 3\gamma^2\hat{b}^2} \to \mp|\gamma| \tag{110}$$

approaching the origin in the blue region. Thus, $\frac{\partial \tilde{t}^I_{\text{root}}}{\partial \tilde{b}}$ is a nonremovable singularity at the origin; however, it appears to be close to zero for subsequences very close to the red and blue curves (as expected).

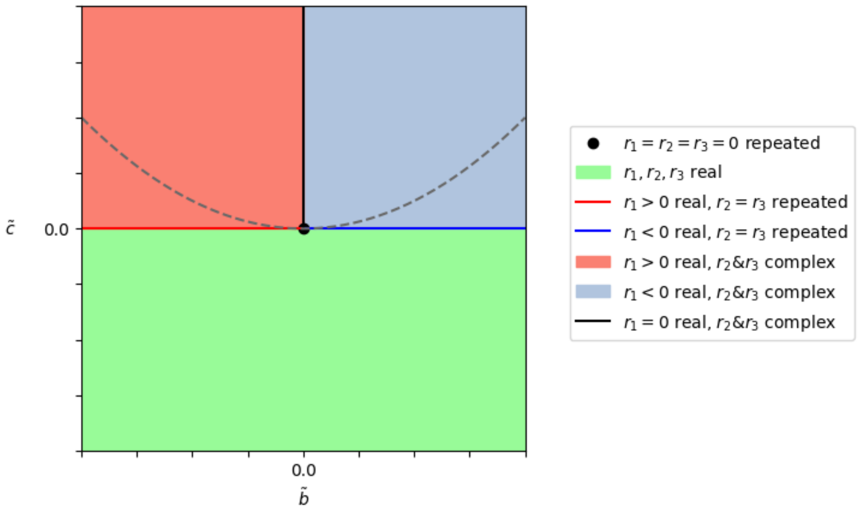


Figure 29: Root classification for the new reduced cubic in terms of $\tilde{b}$ and $\tilde{c}$. Note that the new reduced cubic only has critical points when $\tilde{c} \leq \frac{\tilde{b}^2}{4}$ (on and below the dotted line).

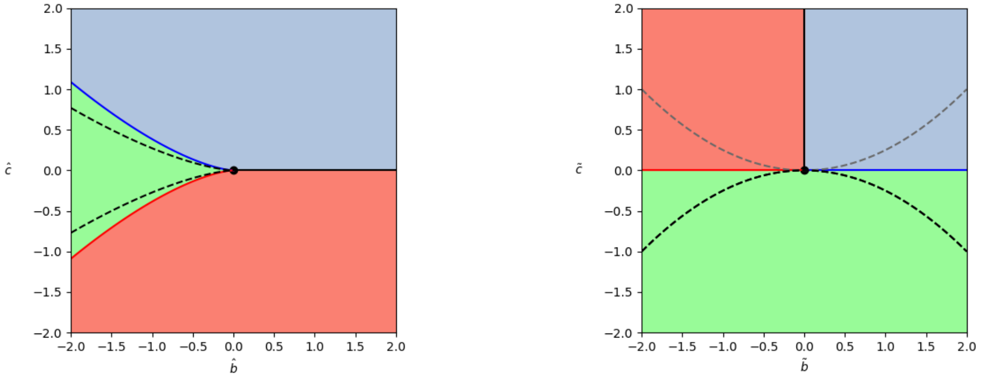


Figure 30: (Left) $\tilde{t}^{+}_{\text{root}} = \sqrt{\frac{2}{3}}(-\hat{b})^{\frac{1}{2}}$ on $\hat{c} = \sqrt{\frac{2}{27}}(-\hat{b})^{\frac{3}{2}}$, and $\tilde{t}^{-}_{\text{root}} = -\sqrt{\frac{2}{3}}(-\hat{b})^{\frac{1}{2}}$ on $\hat{c} = -\sqrt{\frac{2}{27}}(-\hat{b})^{\frac{3}{2}}$ (black dotted lines). (Right) $\tilde{t}^{+}_{\text{root}} = \sqrt{\frac{2}{3}}(-\hat{b})^{\frac{1}{2}}$ on $\tilde{c} = -\frac{\tilde{b}^2}{4}$ when $\tilde{b} > 0$, and $\tilde{t}^{-}_{\text{root}} = -\sqrt{\frac{2}{3}}(-\hat{b})^{\frac{1}{2}}$ on $\tilde{c} = -\frac{(-\tilde{b})^2}{4}$ when $\tilde{b} < 0$ (black dotted lines).

10.5. Roots in the New Canonical Form

In the $\tilde{c} < 0$ case, let

$$\theta_0 = \arctan 2\left(\sqrt{-\tilde{c}}, -\frac{\tilde{b}}{2}\right) \tag{111}$$

as illustrated in Figure 31 in order to define

$$\theta^- = \frac{1}{3}(\theta_0 + 2\pi) \in \left[\frac{2\pi}{3}, \pi\right] \qquad \text{with } \cos(\theta^-) \in \left[-1, -\frac{1}{2}\right] \tag{112a}$$

$$\theta^M = \frac{1}{3}(\theta_0 - 2\pi) \in \left[-\frac{2\pi}{3}, -\frac{\pi}{3}\right] \qquad \text{with } \cos(\theta^M) \in \left[-\frac{1}{2}, \frac{1}{2}\right] \tag{112b}$$

$$\theta^+ = \frac{1}{3}\theta_0 \in \left[0, \frac{\pi}{3}\right] \qquad \text{with } \cos(\theta^+) \in \left[\frac{1}{2}, 1\right] \tag{112c}$$

so that the roots are given by

$$\tilde{t}^R_{\text{root}} = 2\left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{1}{6}} \cos\theta \tag{113}$$

with $\theta = \theta^\pm$ for $\tilde{t}^\pm_{\text{root}}$ respectively and $\theta = \theta^M$ for $\tilde{t}^M_{\text{root}}$. Equation 113 can be verified via a $\cos^3\theta$ identity. Note that

$$3(\tilde{t}^R_{\text{root}})^2 + \hat{b} = 12\left(\frac{\tilde{b}^2}{4} - \tilde{c}\right)^{\frac{1}{3}} \left(\cos^2\theta - \frac{1}{4}\right) \tag{114}$$

which is nonnegative for $\tilde{t}^\pm_{\text{root}}$ and nonpositive for $\tilde{t}^M_{\text{root}}$.

In the $\tilde{c} > 0$ case, there is one real root and two complex conjugate roots. Let $\xi_S = \left(\frac{-\tilde{b}}{2} + \sqrt{\tilde{c}}\right)^{\frac{1}{3}}$ and $\xi_D = \left(\frac{-\tilde{b}}{2} - \sqrt{\tilde{c}}\right)^{\frac{1}{3}}$, so that the real root is $\tilde{t}^R_{\text{root}} = \xi_S + \xi_D$. Note that $3(\tilde{t}^R_{\text{root}})^2 + \hat{b} = 3\left(\xi_S^2 + \xi_S\xi_D + \xi_D^2\right) > 0$ unless $\xi_S = \xi_D = 0$, which is true if and only if $\tilde{b} = \tilde{c} = 0$ (i.e. when $\tilde{t}^R_{\text{root}} = 0$ is a triply repeated real root). The complex roots are

$$\begin{bmatrix} \tilde{t}^R_{\text{root}} \\ \tilde{t}^I_{\text{root}} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}(\xi_S + \xi_D) \\ \pm\frac{\sqrt{3}}{2}(\xi_S - \xi_D) \end{bmatrix} \tag{115}$$

noting that $\xi_S \neq \xi_D$.

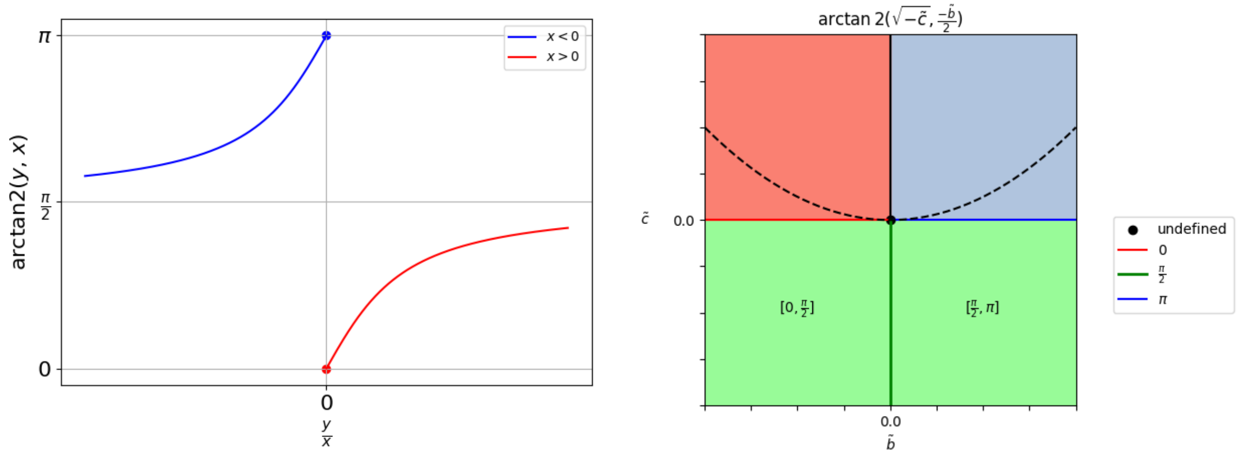


Figure 31: (Left) Since $\sqrt{-\tilde{c}} \geq 0$, only the $y \geq 0$ case is shown. The graph does not reflect the $x = 0$ case, which is: $\arctan 2(y, x) = \frac{\pi}{2}$ when $y > 0$, and $\arctan 2(y, x)$ is undefined when $y = 0$. The undefined case has $\tilde{b} = \tilde{c} = 0$ and thus triply repeated roots identically equal to 0. (Right) Values of $\arctan 2\left(\sqrt{-\tilde{c}}, \frac{-\tilde{b}}{2}\right)$ in the $\tilde{c} \leq 0$ region. Note the nonremovable singularity at the origin.

11. Proposed Approach for the Cubic Equation

In our newly proposed reduced canonical form (see Section 10.4), equations 82 and 83 become

$$\tilde{f}(\tilde{t}_{\text{root}};\tilde{p}) = \begin{bmatrix} (\tilde{t}^R_{\text{root}})^3 - 3\tilde{t}^R_{\text{root}}(\tilde{t}^I_{\text{root}})^2 + \hat{b}\tilde{t}^R_{\text{root}} + \hat{c} \\ -(\tilde{t}^I_{\text{root}})^3 + 3(\tilde{t}^R_{\text{root}})^2\tilde{t}^I_{\text{root}} + \hat{b}\tilde{t}^I_{\text{root}} \end{bmatrix} = q^2 f(t_{\text{root}};\vec{p}) = \vec{0} \tag{116a}$$

$$\tilde{f}_{\theta_1}(\tilde{t}_{\text{root}};\tilde{p}) = \begin{bmatrix} 3(\tilde{t}^R_{\text{root}})^2 - 3(\tilde{t}^I_{\text{root}})^2 + \hat{b} & -6\tilde{t}^R_{\text{root}}\tilde{t}^I_{\text{root}} \\ 6\tilde{t}^R_{\text{root}}\tilde{t}^I_{\text{root}} & 3(\tilde{t}^R_{\text{root}})^2 - 3(\tilde{t}^I_{\text{root}})^2 + \hat{b} \end{bmatrix} = q f_{\theta_1}(t_{\text{root}};\vec{p}) \tag{116b}$$

$$\tilde{f}^{-1}_{\theta_1}(\tilde{t}_{\text{root}};\tilde{p}) = \frac{1}{s}\begin{bmatrix} 3(\tilde{t}^R_{\text{root}})^2 - 3(\tilde{t}^I_{\text{root}})^2 + \hat{b} & 6\tilde{t}^R_{\text{root}}\tilde{t}^I_{\text{root}} \\ -6\tilde{t}^R_{\text{root}}\tilde{t}^I_{\text{root}} & 3(\tilde{t}^R_{\text{root}})^2 - 3(\tilde{t}^I_{\text{root}})^2 + \hat{b} \end{bmatrix} \tag{116c}$$

$$\text{where } s = \left(3(\tilde{t}^R_{\text{root}})^2 - 3(\tilde{t}^I_{\text{root}})^2 + \hat{b}\right)^2 + \left(6\tilde{t}^R_{\text{root}}\tilde{t}^I_{\text{root}}\right)^2$$

$$\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}};\tilde{p}) = \tilde{f}_{\theta_2}(\tilde{t}_{\text{root}};\hat{p})\frac{\partial\hat{p}}{\partial\tilde{p}} = \frac{9}{\hat{b}^2}\begin{bmatrix} \tilde{t}^R_{\text{root}} & 1 \\ \tilde{t}^I_{\text{root}} & 0 \end{bmatrix}\begin{bmatrix} \frac{-\hat{c}}{2} & 1 \\ \frac{\hat{b}^2}{9} & 0 \end{bmatrix} \tag{116d}$$

where equation 116d utilizes equation 92f. Formally, $\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}};\tilde{p})$ and $\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}};\hat{p})$ are size 2x4 and $\frac{\partial\hat{p}}{\partial\tilde{p}}$ is size 4x4; however, $\tilde{f}(\tilde{t}_{\text{root}};\tilde{p})$ only requires the last two columns of $\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}};\tilde{p})$, and so only the last two columns are shown in equation 116d.

Similar to equation 68, $\tilde{f}_{\theta_1}(\tilde{t}_{\text{root}},\tilde{p})\frac{\partial\theta_1}{\partial\theta_2} = -\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}},\tilde{p})$ or $\tilde{f}_{\theta_1}(\tilde{t}_{\text{root}},\tilde{p})\frac{\partial\tilde{t}_{\text{root}}}{\partial\tilde{p}} = -\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}},\tilde{p})$; then, similar to equation 69b,

$$\tilde{f}_{\theta_1}(\tilde{t}_{\text{root}},\tilde{p})\frac{\partial\tilde{t}_{\text{root}}}{\partial\vec{p}} = \tilde{f}_{\theta_1}(\tilde{t}_{\text{root}},\tilde{p})\frac{\partial\tilde{t}_{\text{root}}}{\partial\tilde{p}}\frac{\partial\tilde{p}}{\partial\hat{p}}\frac{\partial\hat{p}}{\partial\vec{p}} = -\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}};\tilde{p})\frac{\partial\tilde{p}}{\partial\hat{p}}\frac{\partial\hat{p}}{\partial\vec{p}} = -\tilde{f}_{\theta_2}(\tilde{t}_{\text{root}};\hat{p})\frac{\partial\hat{p}}{\partial\vec{p}} \tag{117}$$

using equation 116d. Using equation 91 to expand the left hand side and inserting $\frac{\partial\hat{p}}{\partial\vec{p}}$ (see the text after equation 91) on the right hand side leads to

$$\tilde{f}_{\theta_1}(\tilde{t}_{\text{root}},\tilde{p})\left(q\frac{\partial t_{\text{root}}}{\partial\vec{p}} + \begin{bmatrix} t^R_{\text{root}} & \frac{1}{3} & 0 & 0 \\ t^I_{\text{root}} & 0 & 0 & 0 \end{bmatrix}\right) = -\begin{bmatrix} \tilde{t}^R_{\text{root}} & 1 \\ \tilde{t}^I_{\text{root}} & 0 \end{bmatrix}\begin{bmatrix} b & -\frac{2a}{3} & q & 0 \\ -\frac{ab}{3} + 2qc & \frac{2a^2}{9} - \frac{qb}{3} & -\frac{qa}{3} & q^2 \end{bmatrix} \tag{118a}$$

$$q\tilde{f}_{\theta_1}(\tilde{t}_{\text{root}},\tilde{p})\frac{\partial t_{\text{root}}}{\partial\vec{p}} = -q^2\begin{bmatrix} (t^R_{\text{root}})^3 - 3t^R_{\text{root}}(t^I_{\text{root}})^2 & (t^R_{\text{root}})^2 - (t^I_{\text{root}})^2 & t^R_{\text{root}} & 1 \\ 3(t^R_{\text{root}})^2 t^I_{\text{root}} - (t^I_{\text{root}})^3 & 2t^R_{\text{root}}t^I_{\text{root}} & t^I_{\text{root}} & 0 \end{bmatrix} \tag{118b}$$

$$\tilde{f}_{\theta_1}(\tilde{t}_{\text{root}},\tilde{p})\frac{\partial t_{\text{root}}}{\partial\vec{p}} = -q\begin{bmatrix} (t^R_{\text{root}})^3 - 3t^R_{\text{root}}(t^I_{\text{root}})^2 & (t^R_{\text{root}})^2 - (t^I_{\text{root}})^2 & t^R_{\text{root}} & 1 \\ 3(t^R_{\text{root}})^2 t^I_{\text{root}} - (t^I_{\text{root}})^3 & 2t^R_{\text{root}}t^I_{\text{root}} & t^I_{\text{root}} & 0 \end{bmatrix} \tag{118c}$$

where equation 82 was used to obtain the right hand side of equation 118b. Importantly, equation 118c was obtained from equation 118b by dividing by q, which allows for $q \to 0$ but not $q = 0$.

From equation 116c, one can obtain

$$\tilde{f}^{-1}_{\theta_1}(\tilde{t}_{\text{root}};\tilde{p}) = \begin{cases} \frac{1}{3(\tilde{t}^R_{\text{root}})^2+\hat{b}} I & \text{if } \tilde{t}^I_{\text{root}} = 0 \\ \frac{1}{2\tilde{t}^I_{\text{root}}}\frac{1}{(3\tilde{t}^R_{\text{root}})^2+(\tilde{t}^I_{\text{root}})^2}\begin{bmatrix} 3\tilde{t}^R_{\text{root}} & \tilde{t}^I_{\text{root}} \\ -\tilde{t}^I_{\text{root}} & 3\tilde{t}^R_{\text{root}} \end{bmatrix}\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} & \text{if } \tilde{t}^I_{\text{root}} \neq 0 \end{cases} \tag{119a}$$

$$\tilde{f}^{-1}_{\theta_1}(\tilde{t}_{\text{root}};\tilde{p}) = \begin{cases} \frac{1}{q}\frac{1}{3q(t^R_{\text{root}})^2+2at^R_{\text{root}}+b} I & \text{if } t^I_{\text{root}} = 0 \\ \frac{1}{q}\frac{1}{2t^I_{\text{root}}}\frac{1}{(3qt^R_{\text{root}}+a)^2+(qt^I_{\text{root}})^2}\begin{bmatrix} 3qt^R_{\text{root}} + a & qt^I_{\text{root}} \\ -qt^I_{\text{root}} & 3qt^R_{\text{root}} + a \end{bmatrix}\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} & \text{if } t^I_{\text{root}} \neq 0 \end{cases} \tag{119b}$$

noting that the $\frac{1}{q}$ will cancel with the q on the right hand side of equation 118c.

In the $\tilde{t}^I_{\text{root}} = 0$ case of equation 119a, the denominator is nonnegative for $\tilde{t}^{\pm}_{\text{root}}$ and nonpositive for $\tilde{t}^M_{\text{root}}$. As can be seen in equation 91, $\tilde{t}^M_{\text{root}}$ corresponds to t^M_{root} while $\tilde{t}^{\pm}_{\text{root}}$ correspond to the larger/smaller of $t^{\pm}_{\text{root}}$ (respectively) when $q > 0$ (and vice versa when $q < 0$); thus, the denominator in the $t^I_{\text{root}} = 0$ case in equation 119b is nonnegative for $t^{\pm}_{\text{root}}$ and nonpositive for t^M_{root}.

Remark 11.1: The derivative $3q(t^R_{root})^2+2at^R_{root}+b$ should be clamped to be positive for $t^\pm_{root}$ and negative for t^M_{root} when $q>0$ (and vice-versa when $q<0$). Recall that the cubic root solver determines a pseudo-sign for q when $q=0$. When there is one real root and two complex conjugate roots, the real root corresponds to either $t^\pm_{root}$ and is treated similarly.

Remark 11.2: In Remark 9.2, the signs for clamping do not change when a changes sign. This is because the standard definition of $t^\pm_{root}$ in equation 4 switches which of $t^\pm_{root}$ is smaller/larger when a changes sign (which seems like a poor convention given Remark 11.1; however, see Remark 11.5).

Remark 11.3: The sign of the denominator in the $t^I_{root}\neq 0$ case of equation 119b is straightforward based on the signs of q and t^I_{root}.

When $t^I_{\text{root}}=0$,

$$\frac{\partial t_{\text{root}}}{\partial \vec{p}}=\frac{-1}{3q(t^R_{\text{root}})^2+2at^R_{\text{root}}+b}\begin{bmatrix}(t^R_{\text{root}})^3 & (t^R_{\text{root}})^2 & t^R_{\text{root}} & 1\\ 0 & 0 & 0 & 0\end{bmatrix} \tag{120}$$

is obtained by substituting equation 119b into equation 118c. When $(t^R_{\text{root}})^3>1$, we factor $(t^R_{\text{root}})^3$ out into the numerator of the scalar multiplier so that each component of the vector is bounded by 1; then, the scalar multiplier is robustly evaluated (as discussed in the text after equation 72).

When $t^I_{\text{root}}\neq 0$,

$$\begin{aligned}\frac{\partial t_{\text{root}}}{\partial \vec{p}}=\frac{-1}{2t^I_{\text{root}}}\frac{1}{(3qt^R_{\text{root}}+a)^2+(qt^I_{\text{root}})^2}&\begin{bmatrix}3qt^R_{\text{root}}+a & qt^I_{\text{root}}\\ -qt^I_{\text{root}} & 3qt^R_{\text{root}}+a\end{bmatrix}\\ &\begin{bmatrix}3(t^R_{\text{root}})^2t^I_{\text{root}}-(t^I_{\text{root}})^3 & 2t^R_{\text{root}}t^I_{\text{root}} & t^I_{\text{root}} & 0\\ -(t^R_{\text{root}})^3+3t^R_{\text{root}}(t^I_{\text{root}})^2 & -(t^R_{\text{root}})^2+(t^I_{\text{root}})^2 & -t^R_{\text{root}} & -1\end{bmatrix}\end{aligned} \tag{121}$$

is obtained by substituting equation 119b in equation 118c. The 2x2 matrix needs some consideration. Note that q and a cannot both be zero; otherwise, the quadratic root solver utilized by the cubic root solver would not find complex roots.

Firstly, consider $qt^I_{\text{root}}=0$, which makes $q=0$ since $t^I_{\text{root}}\neq 0$; then, $a\neq 0$ and

$$\frac{1}{(3qt^R_{\text{root}}+a)^2+(qt^I_{\text{root}})^2}\begin{bmatrix}3qt^R_{\text{root}}+a & qt^I_{\text{root}}\\ -qt^I_{\text{root}} & 3qt^R_{\text{root}}+a\end{bmatrix}=\frac{1}{a}I \tag{122}$$

so that equation 121 becomes

$$\frac{\partial t_{\text{root}}}{\partial \vec{p}}=\frac{-1}{2at^I_{\text{root}}}\begin{bmatrix}3(t^R_{\text{root}})^2t^I_{\text{root}}-(t^I_{\text{root}})^3 & 2t^R_{\text{root}}t^I_{\text{root}} & t^I_{\text{root}} & 0\\ -(t^R_{\text{root}})^3+3t^R_{\text{root}}(t^I_{\text{root}})^2 & -(t^R_{\text{root}})^2+(t^I_{\text{root}})^2 & -t^R_{\text{root}} & -1\end{bmatrix} \tag{123}$$

which matches equation 72 except for an additional first column. For the top row, t^I_{root} is factored out front (where it cancels) leaving only $2a$ on the denominator. Note that the ±-sign is unnecessary for this (non-merging) real part of the complex root, and only the sign of a is required. The larger in magnitude between $2t^R_{\text{root}}$ and $3(t^R_{\text{root}})^2-(t^I_{\text{root}})^2$ is factored out front when it is larger than 1; in addition, it is more robust to consider $(\sqrt{3}t^R_{\text{root}}+t^I_{\text{root}})(\sqrt{3}t^R_{\text{root}}-t^I_{\text{root}})$ than $3(t^R_{\text{root}})^2-(t^I_{\text{root}})^2$ in the context of Remark 9.6. For the bottom row, the larger in magnitude between $t^R_{\text{root}}\left((t^R_{\text{root}})^2-3(t^I_{\text{root}})^2\right)$, $(t^R_{\text{root}})^2-(t^I_{\text{root}})^2$, and t^R_{root} is factored out front when it is larger than 1; in addition, it is more robust to consider $(t^R_{\text{root}}+\sqrt{3}t^I_{\text{root}})(t^R_{\text{root}}-\sqrt{3}t^I_{\text{root}})$ than $(t^R_{\text{root}})^2-3(t^I_{\text{root}})^2$.

Secondly, consider $3qt^R_{\text{root}}+a=0$, which makes $q\neq 0$ (and $qt^I_{\text{root}}\neq 0$) because $q=0$ would make $a=0$. From equation 91, $\tilde{t}^R_{\text{root}}=0$. From Remark 10.3, this corresponds to the black rays (not including the origin) in Figures 27 and 29. Equation 122 becomes

$$\frac{1}{(3qt^R_{\text{root}}+a)^2+(qt^I_{\text{root}})^2}\begin{bmatrix}3qt^R_{\text{root}}+a & qt^I_{\text{root}}\\ -qt^I_{\text{root}} & 3qt^R_{\text{root}}+a\end{bmatrix}=\frac{1}{qt^I_{\text{root}}}\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix} \tag{124}$$

and the $t^I_{\text{root}}\neq 0$ case of equation 119b becomes

$$\tilde{f}^{-1}_{\theta_1}(\tilde{t}_{\text{root}};\tilde{p})=\frac{1}{q}\frac{1}{2t^I_{\text{root}}}\frac{-1}{qt^I_{\text{root}}}I \tag{125}$$

so that equation 121 becomes

$$\frac{\partial t_{\text{root}}}{\partial \vec{p}} = \frac{1}{2q(t^I_{\text{root}})^2} \begin{bmatrix} (t^R_{\text{root}})^3 - 3t^R_{\text{root}}(t^I_{\text{root}})^2 & (t^R_{\text{root}})^2 - (t^I_{\text{root}})^2 & t^R_{\text{root}} & 1 \\ 3(t^R_{\text{root}})^2 t^I_{\text{root}} - (t^I_{\text{root}})^3 & 2t^R_{\text{root}} t^I_{\text{root}} & t^I_{\text{root}} & 0 \end{bmatrix} \tag{126}$$

where the top and bottom rows are treated similarly to equation 123. Note that the ± sign is unnecessary for the top row, which only contains $(t^I_{\text{root}})^2$.

Finally, consider the case when both $qt^I_{\text{root}} \neq 0$ and $3qt^R_{\text{root}} + a \neq 0$. From equation 121, one can write

$$\frac{\partial t^R_{\text{root}}}{\partial \vec{p}} = \frac{-1}{2} \frac{1}{(3qt^R_{\text{root}} + a)^2 + (qt^I_{\text{root}})^2} \begin{bmatrix} (8qt^R_{\text{root}} + 3a)(t^R_{\text{root}})^2 - a(t^I_{\text{root}})^2 \\ 5q(t^R_{\text{root}})^2 + q(t^I_{\text{root}})^2 + 2at^R_{\text{root}} \\ 2qt^R_{\text{root}} + a \\ -q \end{bmatrix}^T \tag{127}$$

noting that the ±-sign is (again) unnecessary since only $(t^I_{\text{root}})^2$ appears. Since the fourth entry is nonzero, the largest entry can be robustly factored out front. Unlike equations 72, 120, 123, and 126, the parameters contribute to the direction of the vector; however, we have previously assumed that they can be treated as bounded in the discussion in Section 2. When $8qt^R_{\text{root}} + 3a$ and a have the same sign, it is more robust to consider

$$\text{Sign}(a) \left(\sqrt{|8qt^R_{\text{root}} + 3a|}\, t^R_{\text{root}} + \sqrt{|a|}\, t^I_{\text{root}} \right) \left(\sqrt{|8qt^R_{\text{root}} + 3a|}\, t^R_{\text{root}} - \sqrt{|a|}\, t^I_{\text{root}} \right) \tag{128}$$

than the first entry of the vector in equation 127. From equation 121, one can write

$$\begin{aligned} \frac{\partial t^I_{\text{root}}}{\partial \vec{p}} = \frac{1}{2t^I_{\text{root}}} & \frac{1}{(3qt^R_{\text{root}} + a)^2 + (qt^I_{\text{root}})^2} \\ & \begin{bmatrix} t^R_{\text{root}} \left(3qt^R_{\text{root}} + a\right)(t^R_{\text{root}})^2 - \left(q(t^I_{\text{root}})^2 + 6q(t^R_{\text{root}})^2 + 3at^R_{\text{root}}\right)(t^I_{\text{root}})^2 \\ (3qt^R_{\text{root}} + a)(t^R_{\text{root}})^2 - (qt^R_{\text{root}} + a)(t^I_{\text{root}})^2 \\ 3q(t^R_{\text{root}})^2 + q(t^I_{\text{root}})^2 + at^R_{\text{root}} \\ 3qt^R_{\text{root}} + a \end{bmatrix}^T \end{aligned} \tag{129}$$

noting that the fourth entry is nonzero; thus, the largest entry can be robustly factored out front. When $t^R_{\text{root}}\left(3qt^R_{\text{root}} + a\right)$ and $q(t^I_{\text{root}})^2 + 6q(t^R_{\text{root}})^2 + 3at^R_{\text{root}}$ have the same sign, it is more robust to consider

$$\begin{aligned} \text{Sign}\left(t^R_{\text{root}}\left(3qt^R_{\text{root}} + a\right)\right) & \left(\sqrt{\left|t^R_{\text{root}}\left(3qt^R_{\text{root}} + a\right)\right|}\, t^R_{\text{root}} + \sqrt{|q(t^I_{\text{root}})^2 + 6q(t^R_{\text{root}})^2 + 3at^R_{\text{root}}|}\, t^I_{\text{root}} \right) \\ & \left(\sqrt{\left|t^R_{\text{root}}\left(3qt^R_{\text{root}} + a\right)\right|}\, t^R_{\text{root}} - \sqrt{|q(t^I_{\text{root}})^2 + 6q(t^R_{\text{root}})^2 + 3at^R_{\text{root}}|}\, t^I_{\text{root}} \right) \end{aligned} \tag{130}$$

than the first entry of the vector in equation 129. Note that $6q(t^R_{\text{root}})^2$ could be equivalently moved (or partially moved) into the first square root as $-6q(t^I_{\text{root}})^2$; however, that requires carefully considering $3(t^R_{\text{root}})^2 - 6(t^I_{\text{root}})^2$ to help alleviate cancellation. When $3qt^R_{\text{root}} + a$ and $qt^R_{\text{root}} + a$ have the same sign, it is more robust to consider

$$\text{Sign}(3qt^R_{\text{root}} + a) \left(\sqrt{|3qt^R_{\text{root}} + a|}\, t^R_{\text{root}} + \sqrt{|qt^R_{\text{root}} + a|}\, t^I_{\text{root}} \right) \left(\sqrt{|3qt^R_{\text{root}} + a|}\, t^R_{\text{root}} - \sqrt{|qt^R_{\text{root}} + a|}\, t^I_{\text{root}} \right) \tag{131}$$

than the second entry of the vector in equation 129.

Remark 11.4: Equations 123 and 126 are probably fine alternatives to equations 127 and 129 whenever qt^I_{root} or $3qt^R_{root} + a$ is small.

11.1. Branch Selection

The root solver (in Section 10.1) robustly computes all three roots for any values of the parameters, and those values can be used in equations 120, 123, 126, 127, and 129 to robustly compute derivatives. Although the choice of objective function will typically be problem dependent, a few details related to branch selection are discussed here.

For the sake of exposition, consider the change of variables in Section 10.3 highlighted in Figure 27 in order to best parallel the discussion in Section 7. Plugging $\tilde{t}^R_{\text{root,L}}$ and $\tilde{t}^I_{\text{root,L}}$ into equation 82 (top) leads to $\hat{c} = -(\tilde{t}^R_{\text{root,L}})^3 + 3\tilde{t}^R_{\text{root,L}}(\tilde{t}^I_{\text{root,L}})^2 - \hat{b}\tilde{t}^R_{\text{root,L}}$, which can be substituted back into equation 82 (top) to obtain

$$\left(\tilde{t}^R_{\text{root}} - \tilde{t}^R_{\text{root,L}}\right)\left((\tilde{t}^R_{\text{root}})^2 + \tilde{t}^R_{\text{root}}\tilde{t}^R_{\text{root,L}} + (\tilde{t}^R_{\text{root,L}})^2 + \hat{b}\right) - 3\tilde{t}^R_{\text{root}}(\tilde{t}^I_{\text{root}})^2 + 3\tilde{t}^R_{\text{root,L}}(\tilde{t}^I_{\text{root,L}})^2 = 0 \tag{132}$$

after refactoring.

When the target root is real (with $\tilde{t}^I_{\text{root,L}} = 0$), the line $\hat{c} = -(\tilde{t}^R_{\text{root,L}})^3 - \hat{b}\tilde{t}^R_{\text{root,L}}$ describes the family of solutions (see Figure 32) and equation 132 reduces to

$$\left(\tilde{t}^R_{\text{root}} - \tilde{t}^R_{\text{root,L}}\right)\left((\tilde{t}^R_{\text{root}})^2 + \tilde{t}^R_{\text{root}}\tilde{t}^R_{\text{root,L}} + (\tilde{t}^R_{\text{root,L}})^2 + \hat{b}\right) - 3\tilde{t}^R_{\text{root}}(\tilde{t}^I_{\text{root}})^2 = 0 \tag{133}$$

which has a real root (with $\tilde{t}^I_{\text{root}} = 0$) of $\tilde{t}^R_{\text{root,L}}$ as expected. When there are three real roots, equation 133 dictates that the other two are given by $\hat{b} = -(\tilde{t}^R_{\text{root}})^2 - \tilde{t}^R_{\text{root}}\tilde{t}^R_{\text{root,L}} - (\tilde{t}^R_{\text{root,L}})^2$ as illustrated in Figure 33. When two of the roots are complex (with $\tilde{t}^I_{\text{root}} \neq 0$), equation 82 (bottom) gives $\hat{b} = (\tilde{t}^I_{\text{root}})^2 - 3(\tilde{t}^R_{\text{root}})^2$ and equation 133 becomes

$$-2\left(\tilde{t}^R_{\text{root}} + \frac{\tilde{t}^R_{\text{root,L}}}{2}\right)\left(\left(\tilde{t}^R_{\text{root}} - \tilde{t}^R_{\text{root,L}}\right)^2 + (\tilde{t}^I_{\text{root}})^2\right) = 0 \tag{134}$$

where $\tilde{t}^R_{\text{root}} = -\frac{\tilde{t}^R_{\text{root,L}}}{2}$ is the only solution; then, equation 82 (bottom) gives $(\tilde{t}^I_{\text{root}})^2 = \frac{3(\tilde{t}^R_{\text{root,L}})^2}{4} + \hat{b}$. Note that only $\tilde{t}^I_{\text{root}}$ depends on $\hat{b}$, while $\tilde{t}^R_{\text{root}}$ is always $-\frac{\tilde{t}^R_{\text{root,L}}}{2}$ as shown in Figure 33.

When the target root is complex (with $\tilde{t}^I_{\text{root,L}} \neq 0$), equation 82 (bottom) leads to a unique $\hat{b} = (\tilde{t}^I_{\text{root,L}})^2 - 3(\tilde{t}^R_{\text{root,L}})^2$; then, $\hat{c} = 2(\tilde{t}^R_{\text{root,L}})^3 + 2\tilde{t}^R_{\text{root,L}}(\tilde{t}^I_{\text{root,L}})^2$ is also unique. Eliminating $\hat{b}$ in equation 132 leads to

$$\left(\tilde{t}^R_{\text{root}} + 2\tilde{t}^R_{\text{root,L}}\right)\left(\left(\tilde{t}^R_{\text{root}} - \tilde{t}^R_{\text{root,L}}\right)^2 + (\tilde{t}^I_{\text{root,L}})^2\right) - 3\tilde{t}^R_{\text{root}}(\tilde{t}^I_{\text{root}})^2 = 0 \tag{135}$$

which has only one real root (with $\tilde{t}^I_{\text{root}} = 0$) of $-2\tilde{t}^R_{\text{root,L}}$. For the complex roots, equation 82 (bottom) gives $(\tilde{t}^I_{\text{root}})^2 = 3(\tilde{t}^R_{\text{root}})^2 + (\tilde{t}^I_{\text{root,L}})^2 - 3(\tilde{t}^R_{\text{root,L}})^2$. Substituting this into equation 135 gives

$$-2\left(\tilde{t}^R_{\text{root}} - \tilde{t}^R_{\text{root,L}}\right)\left(\left(2\tilde{t}^R_{\text{root}} + \tilde{t}^R_{\text{root,L}}\right)^2 + (\tilde{t}^I_{\text{root,L}})^2\right) = 0 \tag{136}$$

dictating a real part of $\tilde{t}^R_{\text{root}} = \tilde{t}^R_{\text{root,L}}$; then, equation 82 (bottom) gives $(\tilde{t}^I_{\text{root}})^2 = (\tilde{t}^I_{\text{root,L}})^2$.

Next, it is worth briefly commenting on the behavior as $q \to 0$ and switches sign. When a root near $\pm\infty$ changes sign (similar to the bottom right of Figure 21), t^M_{root} becomes one of the $t^{\pm}_{\text{root}}$ roots (and vice versa). As discussed in Remark 11.1, the clamping changes sign when q changes sign keeping the clamping consistent as these two roots change roles; however, one needs to account for this switching of roles when choosing roots for the objective function.

Remark 11.5: Given the need to account for this switching of roles (as $q \to 0$ and the root near $\pm\infty$ changes sign) when choosing roots for the objective function, the convention in equation 4 now seems prudent (as compared to Remark 11.2).

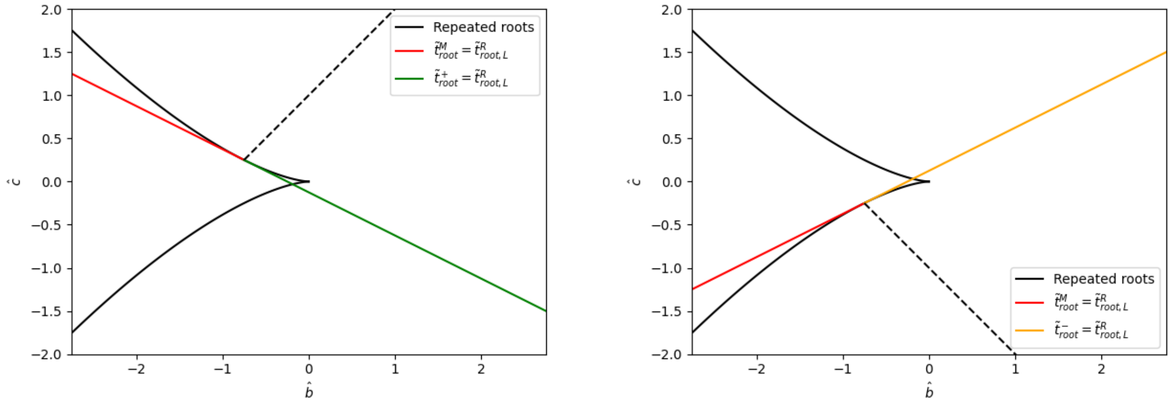


Figure 32: When the target root is real, the line $\hat{c} = -(\tilde{t}^R_{\text{root,L}})^3 - \hat{b}\tilde{t}^R_{\text{root,L}}$ describes the family of solutions. $\tilde{t}^R_{\text{root,L}} = .5$ is shown to the left (in red/green) and $\tilde{t}^R_{\text{root,L}} = -.5$ is shown to the right (in red/yellow). The dotted line is $\hat{c} = 8(\tilde{t}^R_{\text{root,L}})^3 + 2\tilde{t}^R_{\text{root,L}}\hat{b}$, which corresponds to erroneous complex-valued solutions with $\tilde{t}^R_{\text{root}} = \tilde{t}^R_{\text{root,L}}$ obtained when $\tilde{t}^I_{\text{root,L}}$ is not explicitly set to zero in the objective function (similar to the use of equation 58 in Figure 6).

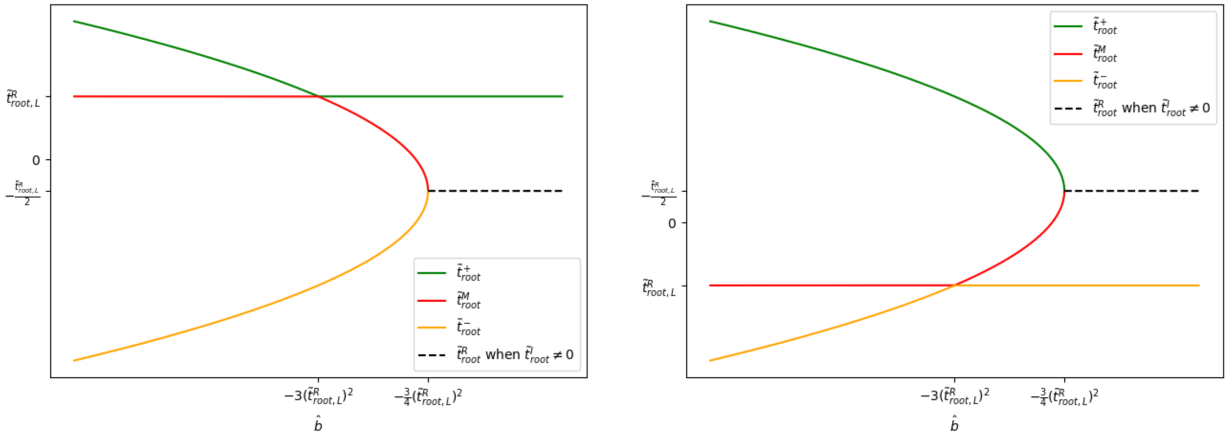


Figure 33: When there are three real roots, one is $\tilde{t}^R_{\text{root,L}}$ and the other two are specified by the parabola $\hat{b} = -(\tilde{t}^R_{\text{root}})^2 - \tilde{t}^R_{\text{root}}\tilde{t}^R_{\text{root,L}} - (\tilde{t}^R_{\text{root,L}})^2$. When two of the roots are complex, their real part is $\tilde{t}^R_{\text{root}} = -\frac{\tilde{t}^R_{\text{root,L}}}{2}$. (Left) $\tilde{t}^R_{\text{root,L}} > 0$ and $\tilde{t}^I_{\text{root,L}} = 0$. (Right) $\tilde{t}^R_{\text{root,L}} < 0$ and $\tilde{t}^I_{\text{root,L}} = 0$.

11.2. Examples

In this section, we show the efficacy for our proposed approach using the root solver from Section 10.1 and equations 120, 123, 126, 127, and 129 to robustly compute the derivatives. The sign of the denominator of each scalar multiplier is chosen according to Remark 11.1 (and Remark 11.3). Equation 73 is used as the objective function, and Adam was used for the optimization. Each example shows the parameter updates in both reduced spaces (see Figures 27 and 29). In our newly proposed canonical form, the family of acceptable solutions is cubic instead of linear. Since the family of acceptable solutions varies as the parameters vary, we (once again) only plot it for the last iteration.

The roots are shown in the non-reduced space. When there are three real roots, the largest is labeled t^+_{root} and colored green, the smallest is labeled t^-_{root} and colored yellow, and the middle root is labeled t^M_{root} and colored red. When two of the roots merge to become complex, the real part of the complex roots is represented by a black dotted line. The remaining real root changes its label/color (from t^+_{root}/green to t^-_{root}/yellow or vice versa) whenever $\hat{c}$ (equal to $\tilde{b}$) changes sign, indicating whether the parameter updates are in the red or blue shaded regions of Figures 27 and 29. If a parameter update lands exactly on the black ray, we arbitrarily use t^+_{root}/green for the label/color. The imaginary parts of the complex roots are colored using the two colors (green/red or red/yellow) not used by the remaining real root.

In each example, we initially choose t^+_{root} as the branch under consideration; however, we switch to t^M_{root}, or to t^-_{root}, or to one of the complex roots when appropriate in order to ensure that we are always using the same root. As discussed at the end of Section 11.1, such a switch is necessary whenever a root near $\pm\infty$ changes sign. It is also necessary when $\hat{c}$ (equal to $\tilde{b}$) changes sign while utilizing the single real root (with the other two roots complex) in the objective function. When the root being utilized merges (switching from being real to complex or vice versa), either of the two new branches may be selected (unless the target root is complex, in which case the sign of the imaginary part matters); however, we always map the larger/smaller real root to the complex root with positive/negative imaginary part for consistency (noting that merging roots can be considered as an independent quadratic factor along the lines of equations 76 and 77).

We first consider an example similar to that shown in Figure 5. Setting $dq = da = 0$ and holding $q = 1$ and $a = 0$ removes the first two columns of $\frac{\partial t_{\text{root}}}{\partial \vec{p}}$. This leads to $\tilde{t} = qt + \frac{a}{3} = t$, $\tilde{b} = b$, and $\hat{c} = c$ for the change of variables in the beginning of Section 10.4. Figures 34 and 35 show the results obtained with an initial guess of $[q, a, b, c]^T = [1, 0, -7.1, 6]^T$ choosing $M = 1000$ and $M = .1$ for the robust division (respectively). In Figure 36, q and a are allowed to vary. These examples (as well as many others, omitted for brevity) exhibit the behavior one would expect given the prior discussions in the paper. Typically, we choose $M = 1000$ for the robust division, but smaller values behave as expected (including the occasional lack of convergence illustrated in Figures 17 and 18). In the subsequent examples, we demonstrate robustness with regard to various degeneracies.

Figure 37 starts out with $t^R_{\text{root}} = 0$ as a triply repeated root, using $[q, a, b, c]^T = [1, 0, 0, 0]^T$. Tables 9 and 10 show the first few iterations. The cubic iterative solver finds one root, and the quadratic solver finds the other two. Equation 120 is used to compute the derivatives. The scalar out front is computed with robust division, and the exactly-zero denominator is clamped to be positive according to Remark 11.1. This leads to $\frac{\partial t^+_{\text{root}}}{\partial c} = -1000$ and $\frac{\partial L}{\partial c} = \frac{\partial L}{\partial t^+_{\text{root}}} \frac{\partial t^+_{\text{root}}}{\partial c} = (0 - \frac{1}{2})(-1000) = 500$, while all the other derivatives are identically zero. As can be seen in Figure 37 and Table 10, two of the roots then become complex as the single real root meanders towards a valid solution.

Figures 38 and 39 along with Tables 11, 12, 13, and 14 address $q = 0$ with $a \neq 0$, where two roots are bounded and one root is $\pm\infty$. Figure 38 starts with $t^2 - 1 = 0$, using $[q, a, b, c]^T = [0, 1, 0, -1]^T$. First, the quadratic root solver finds two roots; then, the cubic solver finds a root at -10^{150} since $a > 0$. Equation 120 is used to compute the derivatives. Since $t^+_{\text{root}} = 1$, the scalar out front is $-\frac{1}{2}$, and all the derivatives are also $-\frac{1}{2}$. Since $\frac{\partial L}{\partial t^+_{\text{root}}} = 1 - \frac{1}{2} = \frac{1}{2}$, all of the derivatives of L are $-\frac{1}{4}$. Figure 39 starts with $-t^2 + 1 = 0$, using $[q, a, b, c]^T = [0, -1, 0, 1]^T$. First, the quadratic root solver finds two roots; then, the cubic solver finds a root at 10^{150} since $a < 0$. Equation 120 is used to compute the derivatives. This leads to $\frac{\partial t^+_{\text{root}}}{\partial q} = -1000$ because of clamping. Since $\frac{\partial L}{\partial t^+_{\text{root}}} = 10^{150} - \frac{1}{2} \approx 10^{150}$, $\frac{\partial L}{\partial q} = \frac{\partial L}{\partial t^+_{\text{root}}} \frac{\partial t^+_{\text{root}}}{\partial q} = 10^{150}(-1000) = -10^{153}$. Note that $\frac{\partial L}{\partial a}$ is 10^{150} times smaller, but still changes a as much as $\frac{\partial L}{\partial q}$ changes q because of the way Adam works. As can be seen in the figure, t^+_{root} is quickly dragged downwards towards $t^R_{\text{root,L}} = \frac{1}{2}$. Even after merging with t^M_{root} to become a complex root (in iteration 3), the tracked root continues making progress and eventually converges. Note that in iterations 26-31, the single real root switches from being t^-_{root} to t^+_{root} as $\hat{c}$ (equal to $\tilde{b}$) changes sign; however, this has no effect on the complex root being tracked (the tracked complex root switches to being t^+_{root} at iteration 41).

Figures 40 and 41 along with Tables 15, 16, 17, and 18 address $q = a = 0$ with $b \neq 0$, where one root is bounded and two roots are $\pm\infty$. Figure 40 starts with $7.1t + 6 = 0$, using $[q, a, b, c]^T = [0, 0, 7.1, 6]^T$. First, the quadratic root solver finds two roots, while choosing a pseudo-sign of $a > 0$ to obtain -10^{150} for the unbounded root; then, the cubic solver also finds a root at -10^{150} using $a > 0$. Equation 120 is used to compute the derivatives. Since t^+_{root} is not one of the unbounded roots, the derivatives are finite. Note that t^+_{root} immediately merges with t^M_{root} to become complex, but later unmerges (at iteration 87) as it converges to the desired solution. Figure 41 starts with $-7.1t + 6 = 0$, using $[q, a, b, c]^T = [0, 0, -7.1, 6]^T$. First, the quadratic root solver finds two roots, while choosing a pseudo-sign of $a > 0$ to obtain 10^{150} for the unbounded root; then, the cubic solver finds a root at -10^{150} using $a > 0$. Equation 120 is used to compute the derivatives. This leads to $\frac{\partial t^+_{\text{root}}}{\partial q} = -1000$ because of clamping. Since $\frac{\partial L}{\partial t^+_{\text{root}}} = 10^{150} - \frac{1}{2} \approx 10^{150}$, $\frac{\partial L}{\partial q} = \frac{\partial L}{\partial t^+_{\text{root}}} \frac{\partial t^+_{\text{root}}}{\partial q} = 10^{150}(-1000) = -10^{153}$. As can be seen in the figure, t^+_{root} is quickly dragged downwards towards $t^R_{\text{root,L}} = \frac{1}{2}$ and eventually converges.

Figures 42 and 43 along with Tables 19, 20, 21, and 22 address $q = a = b = 0$ with $c \neq 0$, where all three roots are $\pm\infty$. Figure 42 starts with $[q, a, b, c]^T = [0, 0, 0, 6]^T$. First, the quadratic root solver finds roots at $\pm 10^{150}$ choosing a pseudo-sign of $a < 0$; then, the cubic solver finds a second root at 10^{150} using $a < 0$. Equation 120 leads to $\frac{\partial t^+_{\text{root}}}{\partial q} = -1000$ because of clamping, and $\frac{\partial L}{\partial q} = -10^{153}$. Figure 43 starts with $[q, a, b, c]^T = [0, 0, 0, -6]^T$. First, the quadratic root solver finds roots at $\pm 10^{150}$ choosing a pseudo-sign of $a > 0$; then, the cubic solver finds a second root at -10^{150} using $a > 0$. Equation 120 leads to $\frac{\partial t^+_{\text{root}}}{\partial q} = -1000$ because of clamping, and $\frac{\partial L}{\partial q} = -10^{153}$.

Figure 44 along with Tables 23 and 24 address $q = a = b = c = 0$, where all three roots are chosen to be identically zero by the quadratic/cubic root solvers. Equation 120 leads to $\frac{\partial t^+_{\text{root}}}{\partial c} = -1000$ as the only nonzero derivative. Since $\frac{\partial L}{\partial t^+_{\text{root}}} = 0 - \frac{1}{2} = -\frac{1}{2}$, $\frac{\partial L}{\partial c} = -\frac{1}{2}(-1000) = 500$; then, c subsequently becomes negative and the iterations continue similar to Figure 43 (and Tables 21 and 22).

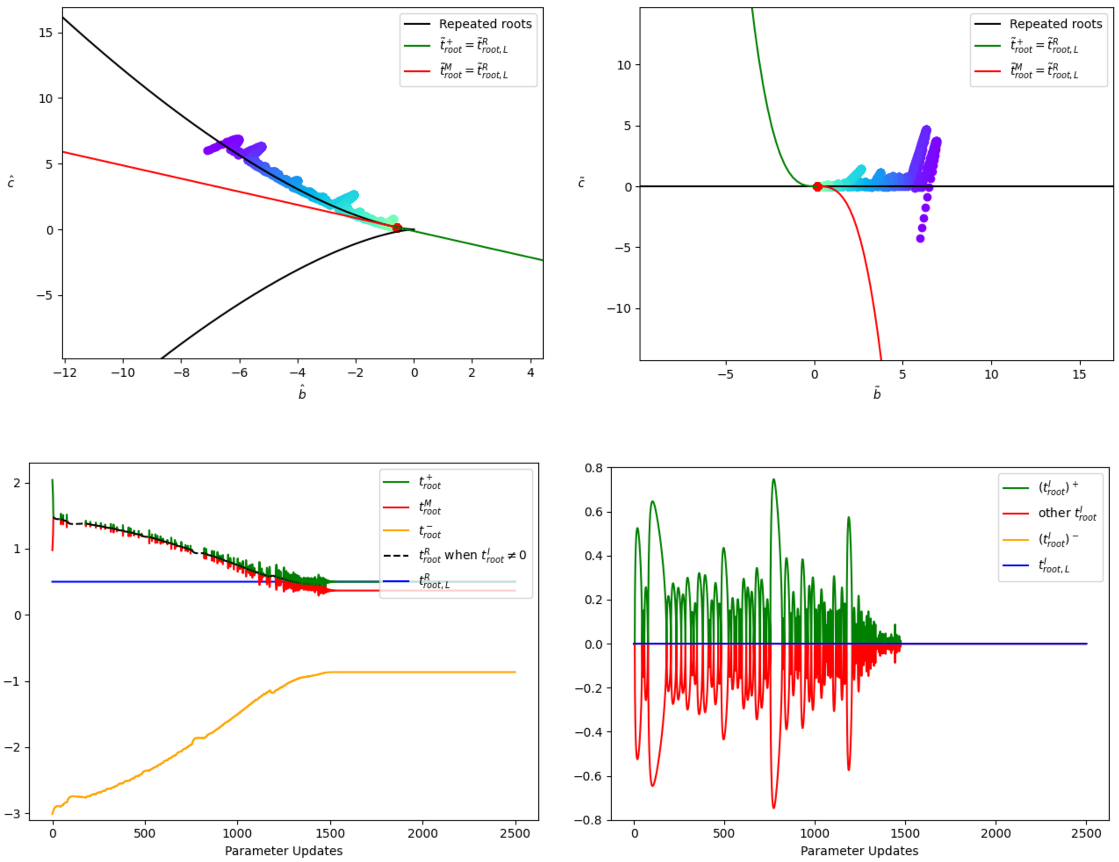


Figure 34: Cubic equation with $q = 1$ and $a = 0$ held fixed. $M = 1000$ is used for the robust division.

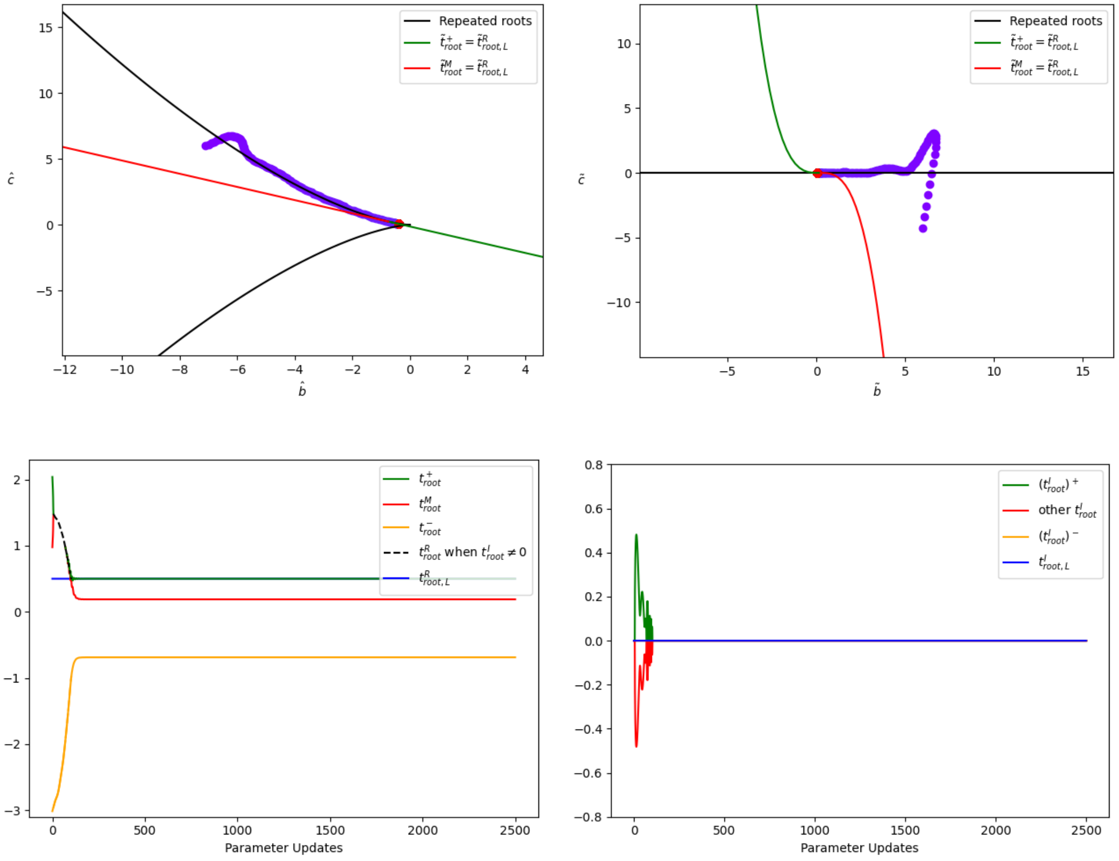


Figure 35: Same as Figure 34, except $M = .1$ is used in the robust division.

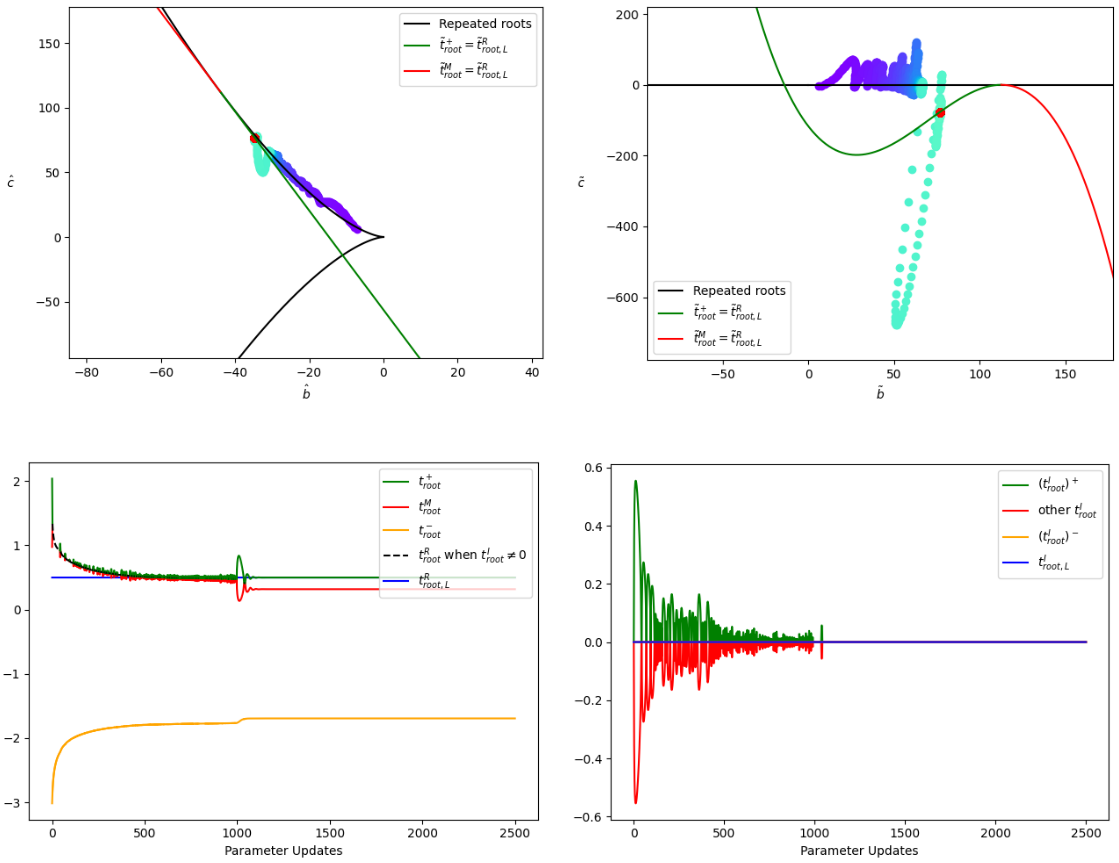


Figure 36: Same as Figure 34, except q and a are allowed to vary.

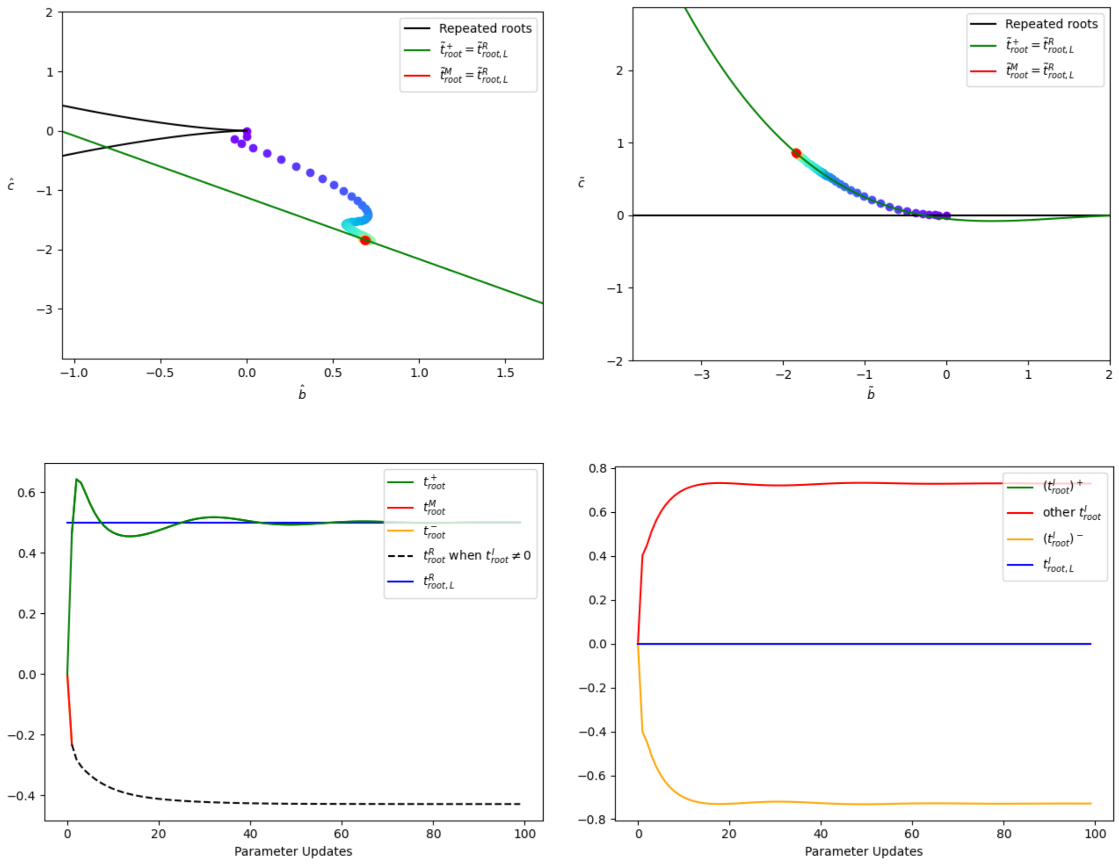


Figure 37: This example starts out with $t^R_{\text{root}} = 0$ as a triply repeated root, using $[q, a, b, c]^T = [1, 0, 0, 0]^T$. $M = 1000$ is used for the robust division.

Itr.	q	a	b	c	$\frac{\partial L}{\partial q}$	$\frac{\partial L}{\partial a}$	$\frac{\partial L}{\partial b}$	$\frac{\partial L}{\partial c}$
0	1.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	5.00e+02
1	1.00e+00	0.00e+00	0.00e+00	-1.00e-01	5.55e-03	1.19e-02	2.57e-02	5.55e-02
2	9.26e-01	-7.44e-02	-7.44e-02	-1.67e-01	-3.87e-02	-6.03e-02	-9.38e-02	-1.46e-01

Table 9: Starting with $[q, a, b, c]^T = [1, 0, 0, 0]^T$

Itr.	$\tilde{c}$	Real Roots			Complex Roots	
		t^+_{root}	t^M_{root}	t^-_{root}	t^R_{root}	$\lvert t^I_{\text{root}} \rvert$
0	0.00e+00	0.00e+00	0.00e+00	0.00e+00	N/A	N/A
1	2.50e-03	4.64e-01	N/A	N/A	-2.32e-01	4.02e-01
2	5.23e-03	6.43e-01	N/A	N/A	-2.81e-01	4.49e-01

Table 10: Starting with $[q, a, b, c]^T = [1, 0, 0, 0]^T$

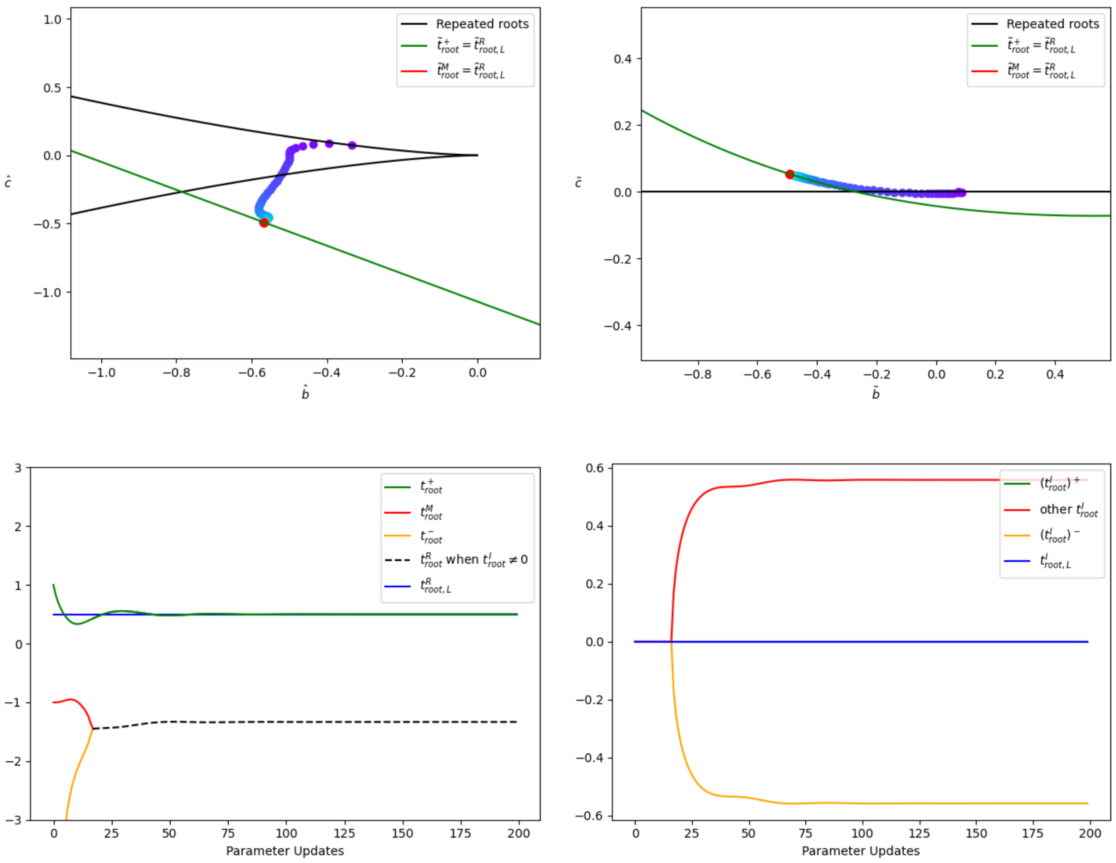


Figure 38: This example starts with $[q, a, b, c]^T = [0, 1, 0, -1]^T$. $M = 1000$ is used for the robust division.

Itr.	q	a	b	c	$\frac{\partial L}{\partial q}$	$\frac{\partial L}{\partial a}$	$\frac{\partial L}{\partial b}$	$\frac{\partial L}{\partial c}$
0	0.00e+00	1.00e+00	0.00e+00	-1.00e+00	-2.50e-01	-2.50e-01	-2.50e-01	-2.50e-01
1	1.00e-01	1.10e+00	1.00e-01	-9.00e-01	-8.89e-02	-1.07e-01	-1.29e-01	-1.55e-01
2	1.88e-01	1.19e+00	1.94e-01	-8.04e-01	-3.45e-02	-4.86e-02	-6.85e-02	-9.66e-02

Table 11: Starting with $[q, a, b, c]^T = [0, 1, 0, -1]^T$

Itr.	$\tilde{c}$	Real Roots			Complex Roots	
		t^+_{root}	t^M_{root}	t^-_{root}	t^R_{root}	$\lvert t^I_{\text{root}} \rvert$
0	0.00e+00	1.00e+00	-1.00e+00	-1.00e+150	N/A	N/A
1	-4.08e-04	8.31e-01	-1.00e+00	-1.08e+01	N/A	N/A
2	-1.39e-03	7.09e-01	-9.97e-01	-6.04e+00	N/A	N/A

Table 12: Starting with $[q, a, b, c]^T = [0, 1, 0, -1]^T$

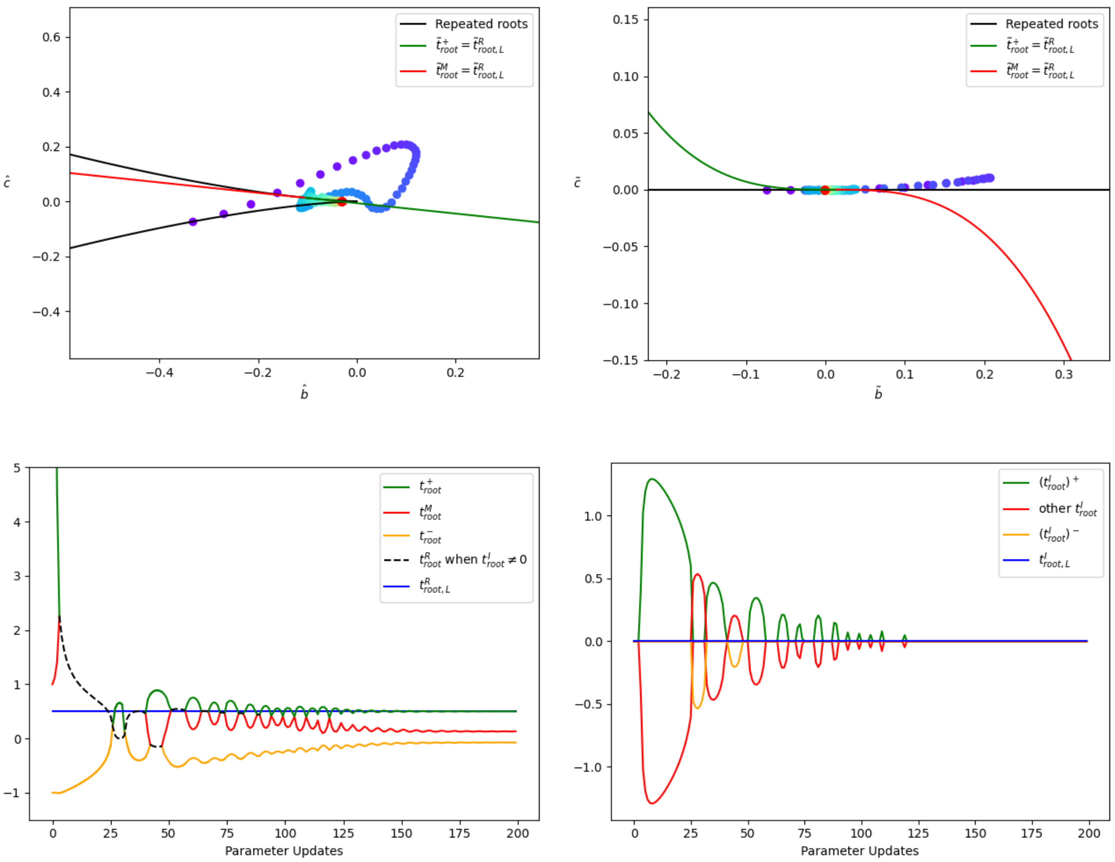


Figure 39: This example starts with $[q, a, b, c]^T = [0, -1, 0, 1]^T$. $M = 1000$ is used for the robust division.

Itr.	q	a	b	c	$\frac{\partial L}{\partial q}$	$\frac{\partial L}{\partial a}$	$\frac{\partial L}{\partial b}$	$\frac{\partial L}{\partial c}$
0	0.00e+00	-1.00e+00	0.00e+00	1.00e+00	-1.00e+153	-1.00e+03	-1.00e-147	0.00e+00
1	1.00e-01	-9.00e-01	1.00e-140	1.00e+00	-7.65e+02	-8.62e+01	-9.71e+00	-1.09e+00
2	1.67e-01	-8.27e-01	7.44e-02	1.07e+00	-1.31e+02	-2.88e+01	-6.34e+00	-1.40e+00

Table 13: Starting with $[q, a, b, c]^T = [0, -1, 0, 1]^T$

Itr.	$\tilde{c}$	Real Roots			Complex Roots	
		t^+_{root}	t^M_{root}	t^-_{root}	t^R_{root}	$\lvert t^I_{\text{root}} \rvert$
0	0.00e+00	1.00e+150	1.00e+00	-1.00e+00	N/A	N/A
1	-2.45e-04	8.87e+00	1.13e+00	-1.00e+00	N/A	N/A
2	-3.52e-04	4.54e+00	1.41e+00	-1.00e+00	N/A	N/A

Table 14: Starting with $[q, a, b, c]^T = [0, -1, 0, 1]^T$

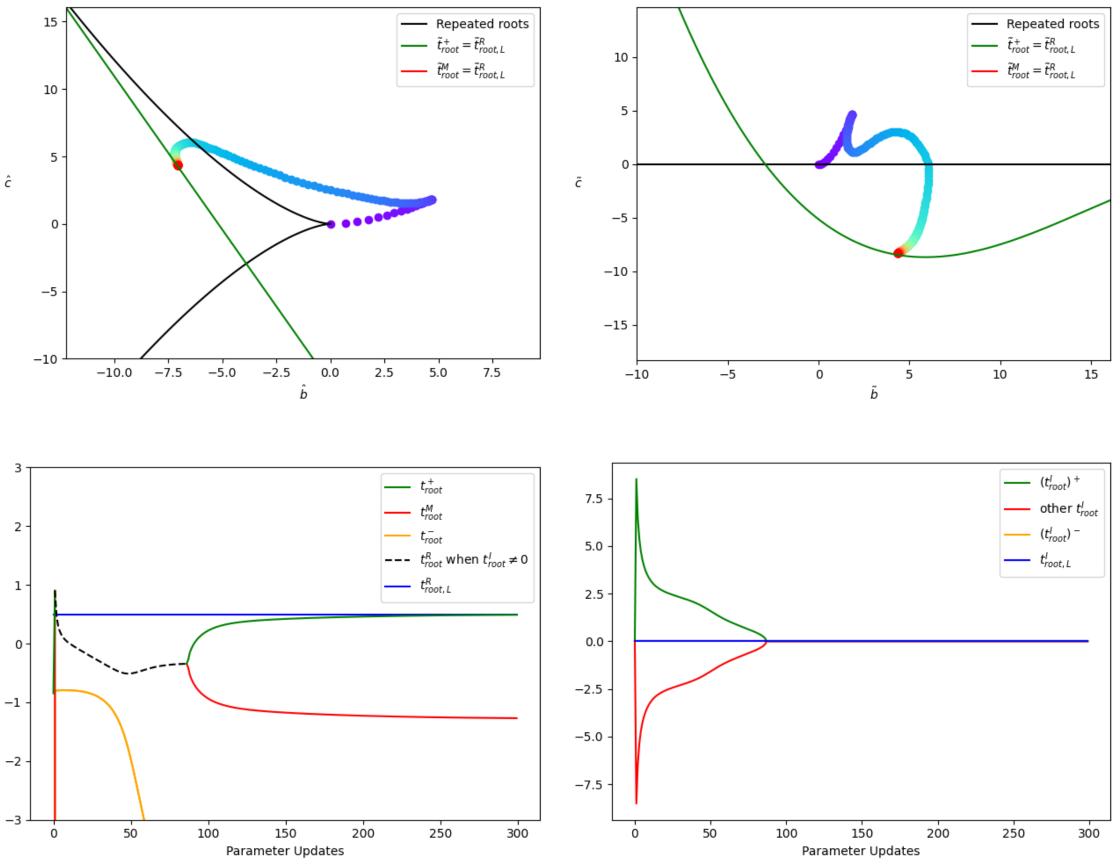


Figure 40: This example starts with $[q, a, b, c]^T = [0, 0, 7.1, 6]^T$. $M = 1000$ is used for the robust division.

Itr.	q	a	b	c	$\frac{\partial L}{\partial q}$	$\frac{\partial L}{\partial a}$	$\frac{\partial L}{\partial b}$	$\frac{\partial L}{\partial c}$
0	0.00e+00	0.00e+00	7.10e+00	6.00e+00	-1.14e-01	1.35e-01	-1.60e-01	1.89e-01
1	1.00e-01	-1.00e-01	7.20e+00	5.90e+00	-3.61e+02	-2.37e+00	4.89e+00	1.39e-01
2	1.74e-01	-2.95e-02	7.13e+00	5.80e+00	-1.17e+02	-8.03e-01	2.80e+00	8.44e-02

Table 15: Starting with $[q, a, b, c]^T = [0, 0, 7.1, 6]^T$

Itr.	$\tilde{c}$	Real Roots			Complex Roots	
		t^+_{root}	t^M_{root}	t^-_{root}	t^R_{root}	$\lvert t^I_{\text{root}}\rvert$
0	0.00e+00	-8.45e-01	-1.00e+150	-1.00e+150	N/A	N/A
1	1.54e-02	N/A	N/A	-8.03e-01	9.02e-01	8.52e+00
2	8.00e-02	N/A	N/A	-7.99e-01	4.84e-01	6.43e+00

Table 16: Starting with $[q, a, b, c]^T = [0, 0, 7.1, 6]^T$

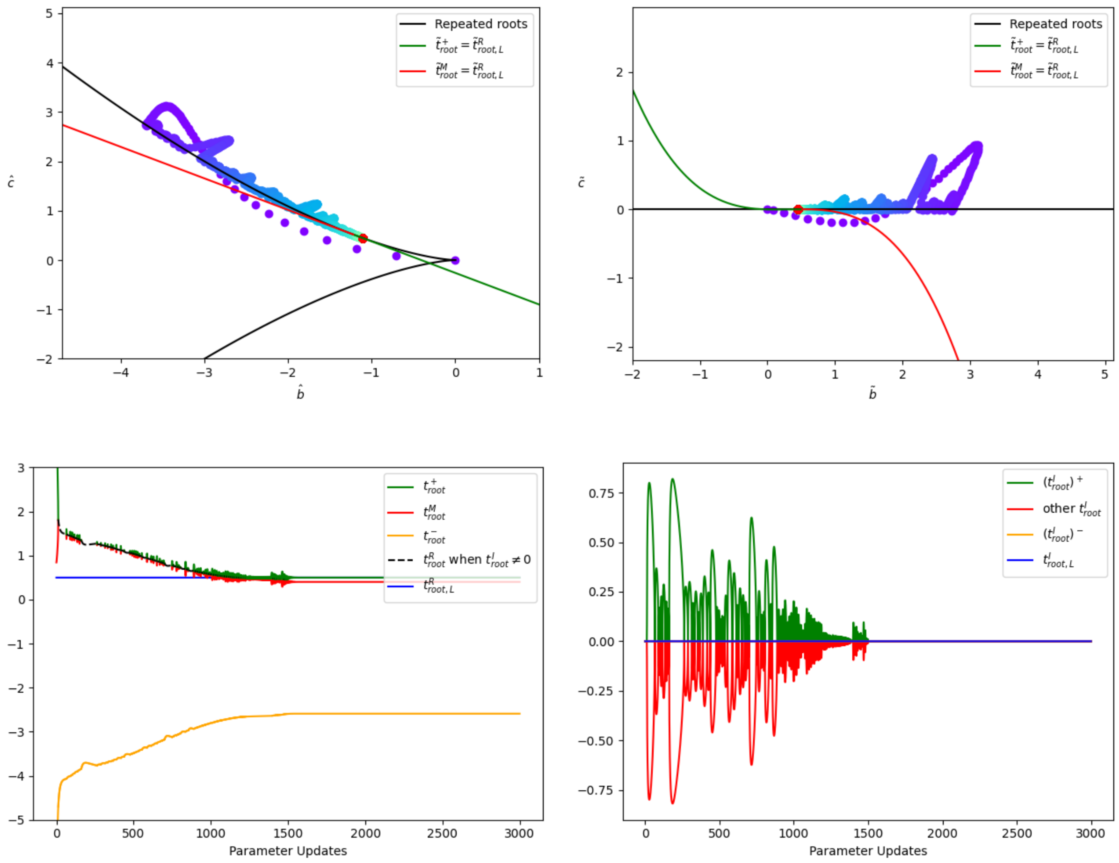


Figure 41: This example starts with $[q, a, b, c]^T = [0, 0, -7.1, 6]^T$. $M = 1000$ is used for the robust division.

Itr.	q	a	b	c	$\frac{\partial L}{\partial q}$	$\frac{\partial L}{\partial a}$	$\frac{\partial L}{\partial b}$	$\frac{\partial L}{\partial c}$
0	0.00e+00	0.00e+00	-7.10e+00	6.00e+00	-1.00e+153	-1.00e+03	-1.00e-147	0.00e+00
1	1.00e-01	1.00e-01	-7.10e+00	6.00e+00	-2.60e+02	-3.49e+01	-4.69e+00	-6.30e-01
2	1.67e-01	1.70e-01	-7.03e+00	6.07e+00	-8.28e+01	-1.51e+01	-2.77e+00	-5.07e-01

Table 17: Starting with $[q, a, b, c]^T = [0, 0, -7.1, 6]^T$

Itr.	$\tilde{c}$	Real Roots			Complex Roots	
		t^{+}_{root}	t^{M}_{root}	t^{-}_{root}	t^{R}_{root}	$\lvert t^{I}_{\text{root}} \rvert$
0	0.00e+00	1.00e+150	8.45e-01	-1.00e+150	N/A	N/A
1	-1.17e-02	7.45e+00	8.65e-01	-9.31e+00	N/A	N/A
2	-4.74e-02	5.46e+00	9.02e-01	-7.38e+00	N/A	N/A

Table 18: Starting with $[q, a, b, c]^T = [0, 0, -7.1, 6]^T$

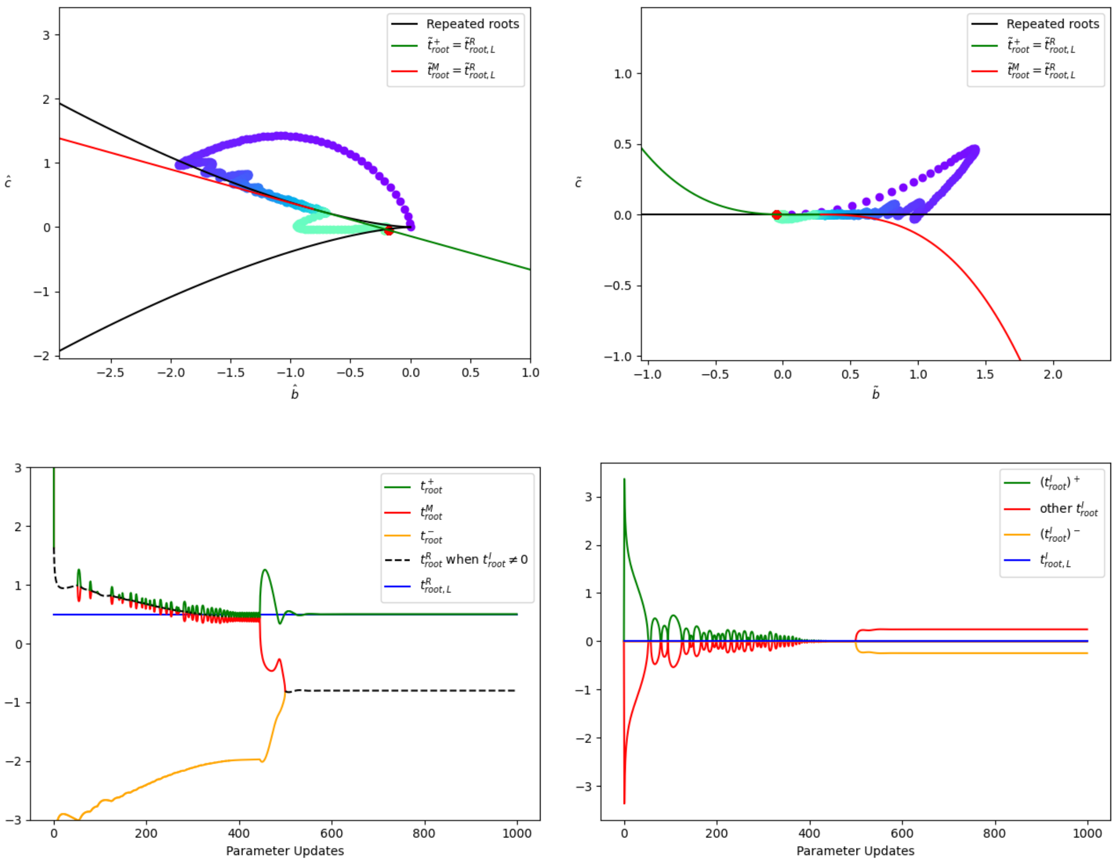


Figure 42: This example starts with $[q, a, b, c]^T = [0, 0, 0, 6]^T$. $M = 1000$ is used for the robust division.

Itr.	q	a	b	c	$\frac{\partial L}{\partial q}$	$\frac{\partial L}{\partial a}$	$\frac{\partial L}{\partial b}$	$\frac{\partial L}{\partial c}$
0	0.00e+00	0.00e+00	0.00e+00	6.00e+00	-1.00e+153	-1.00e+03	-1.00e-147	0.00e+00
1	1.00e-01	1.00e-01	1.00e-140	6.00e+00	-4.07e+01	-4.96e+00	1.74e+00	7.61e-01
2	1.67e-01	1.67e-01	-7.44e-02	5.93e+00	-1.61e+01	-2.31e+00	1.03e+00	5.31e-01

Table 19: Starting with $[q, a, b, c]^T = [0, 0, 0, 6]^T$

Itr.	$\tilde{c}$	Real Roots			Complex Roots	
		t^{+}_{root}	t^{M}_{root}	t^{-}_{root}	t^{R}_{root}	$\lvert t^{I}_{\text{root}}\rvert$
0	0.00e+00	1.00e+150	1.00e+150	-1.00e+150	N/A	N/A
1	9.02e-04	N/A	N/A	-4.28e+00	1.64e+00	3.37e+00
2	6.91e-03	N/A	N/A	-3.71e+00	1.35e+00	2.78e+00

Table 20: Starting with $[q, a, b, c]^T = [0, 0, 0, 6]^T$

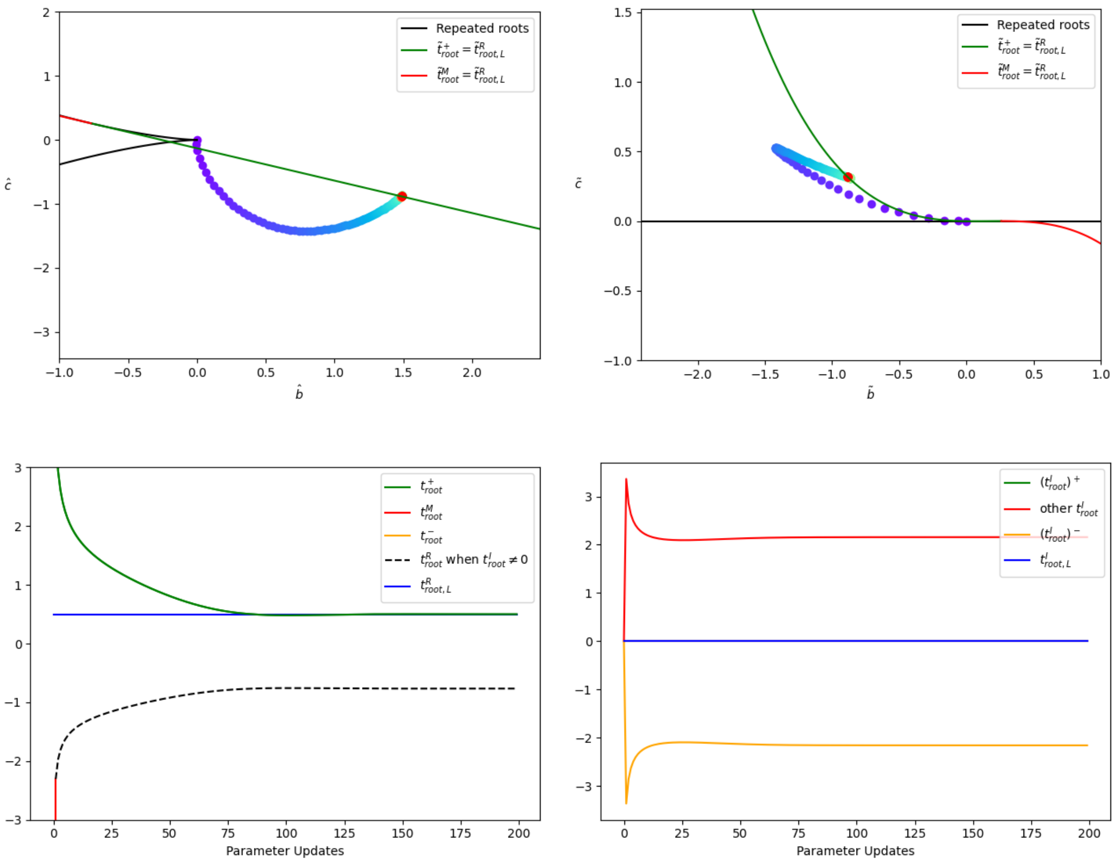


Figure 43: This example starts with $[q, a, b, c]^T = [0, 0, 0, -6]^T$. $M = 1000$ is used for the robust division.

Itr.	q	a	b	c	$\frac{\partial L}{\partial q}$	$\frac{\partial L}{\partial a}$	$\frac{\partial L}{\partial b}$	$\frac{\partial L}{\partial c}$
0	0.00e+00	0.00e+00	0.00e+00	-6.00e+00	-1.00e+153	-1.00e+03	-1.00e-147	0.00e+00
1	1.00e-01	1.00e-01	1.00e-140	-6.00e+00	-3.16e+01	-8.75e+00	-2.42e+00	-6.72e-01
2	1.67e-01	1.68e-01	7.44e-02	-5.93e+00	-1.15e+01	-3.92e+00	-1.33e+00	-4.52e-01

Table 21: Starting with $[q, a, b, c]^T = [0, 0, 0, -6]^T$

Itr.	$\tilde{c}$	Real Roots			Complex Roots	
		t^{+}_{root}	t^{M}_{root}	t^{-}_{root}	t^{R}_{root}	$\lvert t^{I}_{\text{root}} \rvert$
0	0.00e+00	1.00e+150	-1.00e+150	-1.00e+150	N/A	N/A
1	8.98e-04	3.61e+00	N/A	N/A	-2.30e+00	3.36e+00
2	6.86e-03	2.94e+00	N/A	N/A	-1.97e+00	2.86e+00

Table 22: Starting with $[q, a, b, c]^T = [0, 0, 0, -6]^T$

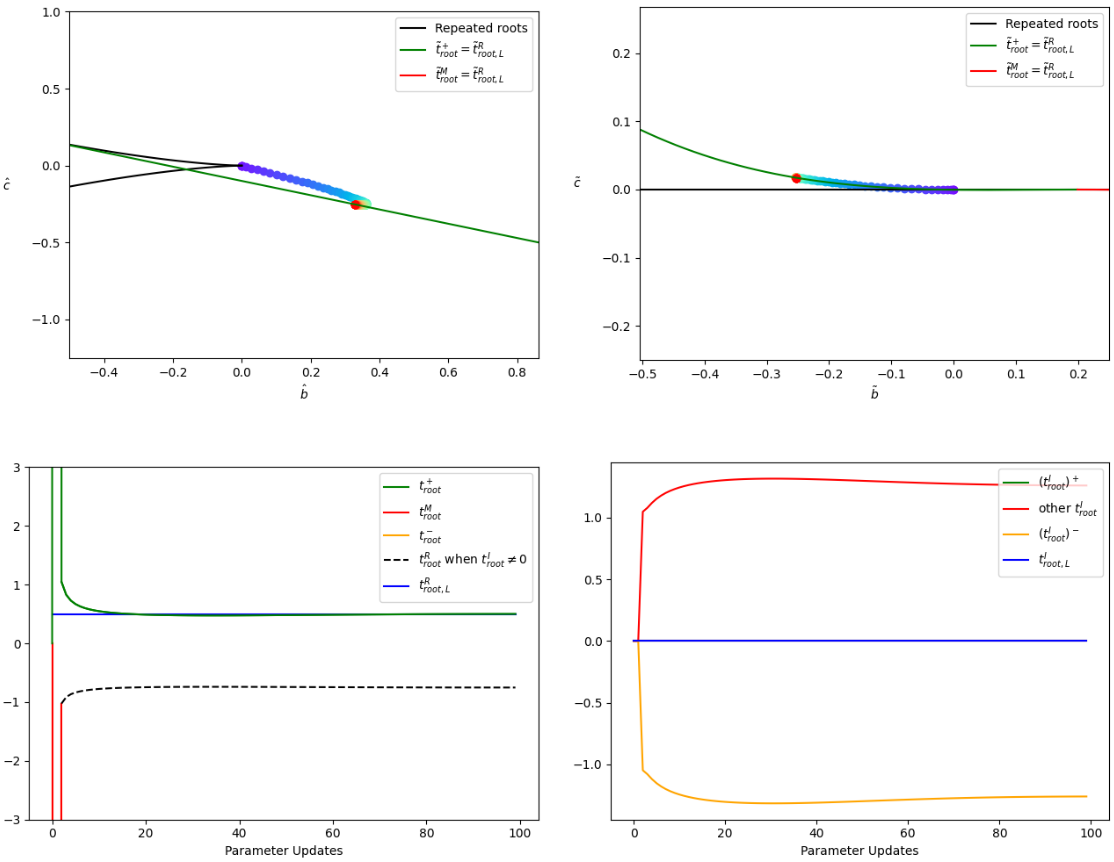


Figure 44: This example starts out with $t^R_{\text{root}} = 0$ as a triply repeated root, using $[q, a, b, c]^T = [0, 0, 0, 0]^T$. $M = 1000$ is used for the robust division.

Itr.	q	a	b	c	$\frac{\partial L}{\partial q}$	$\frac{\partial L}{\partial a}$	$\frac{\partial L}{\partial b}$	$\frac{\partial L}{\partial c}$
0	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	5.00e+02
1	0.00e+00	0.00e+00	0.00e+00	-1.00e-01	-1.00e+153	-1.00e+03	-1.00e-147	0.00e+00
2	7.44e-02	7.44e-02	5.26e-141	-1.67e-01	-1.57e+00	-1.50e+00	-1.43e+00	-1.37e+00

Table 23: Starting with $[q, a, b, c]^T = [0, 0, 0, 0]^T$

Itr.	$\tilde{c}$	Real Roots			Complex Roots	
		t^{+}_{root}	t^{M}_{root}	t^{-}_{root}	t^{R}_{root}	$\lvert t^{I}_{\text{root}} \rvert$
0	0.00e+00	0.00e+00	0.00e+00	0.00e+00	N/A	N/A
1	0.00e+00	1.00e+150	-1.00e+150	-1.00e+150	N/A	N/A
2	2.00e-07	1.05e+00	N/A	N/A	-1.02e+00	1.05e+00

Table 24: Starting with $[q, a, b, c]^T = [0, 0, 0, 0]^T$

12. Conclusions and Future Work

For general polynomials, one could treat any two real roots or complex conjugate pairs as a quadratic factor. Subsequently, the polynomial can be written as the quadratic factor times the remaining factor; moreover, this can be done even when the roots of the quadratic factor have only been found numerically via an iterative solver. Once a polynomial has been written with all merging, potentially-merging, or close-to-merging pairs of roots in quadratic factors, the product rule can be used to isolate the derivative of each quadratic factor for further consideration along the lines discussed in this paper. We leave this as future work. Treating general polynomials would be interesting, since many have used polynomials as approximations to real-world events. For example, in a pursuer/evader scenario, trajectories can be approximated by polynomials with the pursuer winning when there is a real-valued root to the difference between the polynomials in the allotted time (and the evader winning when the roots stay complex for the entire time). Importantly, a competitive scenario would be played out near/crossing the numerically-sensitive boundary between real and complex roots. In the context of differentiable game theory, one could formulate polynomial approximations of strategies with a real-world event occurring when the roots are real and not occurring when the roots are complex.

We briefly summarize some major points here: Backpropagation through Newton's method is untenable, and there are many cases where it does not work (see Remark 6.1); thus, we utilize implicit differentiation. In the repeated root case, implicit differentiation yields a coefficient matrix (to the desired derivatives) which is identically equal to zero. Although numerically untenable, we analytically capture the behavior of the unbounded derivatives and use the results to formulate a robust numerical method for ascertaining search directions; notably, only the search directions (not the individual derivatives) are required for optimization. Our approach treats every possible degeneracy (as elucidated somewhat in the examples section), and we provide detailed remarks discussing how to precisely implement various formulas that inherently require L'Hopital's rule in one form or another.

13. Acknowledgments

Research supported in part by ONR N00014-19-1-2285 and ONR N00014-21-1-2771. DJ is supported in part by a Stanford Graduate Fellowship. RF would like to acknowledge his first PhD advisor, Charles Lange (1942-1993), for introducing him to [85].

References

[1] H. Lee, I. S. Kang, Neural algorithm for solving differential equations, Journal of Computational Physics 91 (1990) 110–131.

[2] J. Sirignano, K. Spiliopoulos, Dgm: A deep learning algorithm for solving partial differential equations, Journal of computational physics 375 (2018) 1339–1364.

[3] M. Raissi, G. E. Karniadakis, Hidden physics models: Machine learning of nonlinear partial differential equations, Journal of Computational Physics 357 (2018) 125–141.

[4] R. K. Tripathy, I. Bilionis, Deep uq: Learning deep neural network surrogate models for high dimensional uncertainty quantification, Journal of computational physics 375 (2018) 565–588.

[5] N. Winovich, K. Ramani, G. Lin, Convpde-uq: Convolutional neural networks with quantified uncertainty for heterogeneous elliptic partial differential equations on varied domains, Journal of Computational Physics 394 (2019) 263–279.

[6] M. Raissi, P. Perdikaris, G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics 378 (2019) 686–707.

[7] J. Berg, K. Nyström, Data-driven discovery of pdes in complex datasets, Journal of Computational Physics 384 (2019) 239–252.

[8] N. Dal Santo, S. Deparis, L. Pegolotti, Data driven approximation of parametrized pdes by reduced basis and neural networks, Journal of Computational Physics (2020) 109550.

[9] J. Magiera, D. Ray, J. S. Hesthaven, C. Rohde, Constraint-aware neural networks for riemann problems, Journal of Computational Physics 409 (2020) 109345.

[10] D. Z. Huang, K. Xu, C. Farhat, E. Darve, Learning constitutive relations from indirect observations using deep neural networks, Journal of Computational Physics (2020) 109491.

[11] A. D. Jagtap, K. Kawaguchi, G. E. Karniadakis, Adaptive activation functions accelerate convergence in deep and physics-informed neural networks, Journal of Computational Physics 404 (2020) 109136.

[12] Z. Geng, D. Johnson, R. Fedkiw, Coercing machine learning to output physically accurate results, Journal of Computational Physics 406 (2020) 109099.

[13] O. Alund, G. Iaccarino, J. Nordstrom, Learning to differentiate, Journal of Computational Physics 424 (2021) 109873.

[14] JCP Special Issue on Machine Learning for Physical Systems, Journal of Computational Physics, Guest edited by George E. Karniadakis and Jan Hesthaven (2020).

[15] R. Fletcher, M. J. Powell, A rapidly convergent descent method for minimization, The computer journal 6 (1963) 163–168.

[16] C. G. Broyden, A class of methods for solving nonlinear simultaneous equations, Mathematics of computation 19 (1965) 577–593.

[17] C. G. Broyden, Quasi-newton methods and their application to function minimisation, Mathematics of Computation 21 (1967) 368–381.

[18] C. Broyden, A new double-rank minimisation algorithm. preliminary report, in: Notices of the American Mathematical Society, volume 16, AMER MATHEMATICAL SOC 201 CHARLES ST, PROVIDENCE, RI 02940-2213, 1969, p. 670.

[19] R. Fletcher, A new approach to variable metric algorithms, The computer journal 13 (1970) 317–322.

[20] D. Goldfarb, A family of variable-metric methods derived by variational means, Mathematics of computation 24 (1970) 23–26.

[21] D. F. Shanno, Conditioning of quasi-newton methods for function minimization, Mathematics of computation 24 (1970) 647–656.

[22] J. Nocedal, Updating quasi-newton matrices with limited storage, Mathematics of computation 35 (1980) 773–782.

[23] D. C. Liu, J. Nocedal, On the limited memory bfgs method for large scale optimization, Mathematical programming 45 (1989) 503–528.

[24] W. C. Davidon, Variable metric method for minimization, SIAM Journal on Optimization 1 (1991) 1–17.

[25] Q. V. Le, J. Ngiam, A. Coates, A. Lahiri, B. Prochnow, A. Y. Ng, On optimization methods for deep learning, in: Proceedings of the 28th International Conference on International Conference on Machine Learning, 2011, pp. 265–272.

[26] H. Robbins, S. Monro, A stochastic approximation method, The Annals of Mathematical Statistics (1951) 400–407.

[27] J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization., Journal of machine learning research 12 (2011).

[28] T. Tieleman, G. Hinton, Lecture 6.5: rmsprop: Divide the gradient by a running average of its recent magnitude, Coursera: Neural Networks for machine learning 4 (2012) 26–31.

[29] M. D. Zeiler, Adadelta: an adaptive learning rate method, arXiv preprint arXiv:1212.5701 (2012).

[30] N. Qian, On the momentum term in gradient descent learning algorithms, Neural networks 12 (1999) 145–151.

[31] Y. E. Nesterov, A method for solving the convex programming problem with convergence rate $o(1/k^2)$, in: Doklady Akademii nauk SSSR, volume 269, 1983, pp. 543–547.

[32] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[33] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, in: Advances in neural information processing systems, 2019, pp. 8026–8037.

[34] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467 (2016).

[35] R. Collobert, K. Kavukcuoglu, C. Farabet, Torch7: A matlab-like environment for machine learning, in: BigLearn, NIPS Workshop, 2011.

[36] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: Proceedings of the 22nd ACM international conference on Multimedia, 2014, pp. 675–678.

[37] R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, A. Belopolsky, et al., Theano: A python framework for fast computation of mathematical expressions, arXiv (2016) arXiv–1605.

[38] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, Q. Zhang, JAX: composable transformations of Python+NumPy programs, 2018. URL: `http://github.com/google/jax`.

[39] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind, Automatic differentiation in machine learning: a survey, The Journal of Machine Learning Research 18 (2017) 5595–5637.

[40] J. Schmidhuber, Deep learning in neural networks: An overview, Neural networks 61 (2015) 85–117.

[41] M. T. Heath, Scientific Computing: An Introductory Survey, Revised Second Edition, SIAM, 2018.

[42] I. Harari, U. Albocher, Computation of eigenvalues of a real, symmetric 3× 3 matrix with particular reference to the pernicious case of two nearly equal eigenvalues, International Journal for Numerical Methods in Engineering 124 (2023) 1089–1110.

[43] G. C. di Fagnano, Produzioni matematiche del conte Giulio Carlo di Fagnano, marchese de'Toschi, e di Sant'Onorio, nobile romano, e patrizio senogagliese..., volume 1, Nella stamperia Gavelliana, 1750.

[44] R. Bridson, R. Fedkiw, J. Anderson, Robust treatment of collisions, contact and friction for cloth animation, in: Proceedings of the 29th annual conference on Computer graphics and interactive techniques, 2002, pp. 594–603.

[45] D. H. Bailey, High-precision computation and mathematical physics (2009).

[46] J. R. Shewchuk, Adaptive precision floating-point arithmetic and fast robust geometric predicates, Discrete & Computational Geometry 18 (1997) 305–363.

[47] D. Johnson, T. Maxfield, Y. Jin, R. Fedkiw, Software-based automatic differentiation is flawed, 2023. `arXiv:2305.03863`.

[48] R. J. LeVeque, R. J. Leveque, Numerical methods for conservation laws, volume 3, Springer, 1992.

[49] E. F. Toro, Riemann solvers and numerical methods for fluid dynamics: a practical introduction, Springer Science & Business Media, 2013.

[50] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, ii, in: Upwind and High-Resolution Schemes, Springer, 1989, pp. 328–374.

[51] S. Osher, R. Fedkiw, K. Piechor, Level set methods and dynamic implicit surfaces, Appl. Mech. Rev. 57 (2004) B15–B15.

[52] R. P. Fedkiw, T. Aslam, B. Merriman, S. Osher, et al., A non-oscillatory eulerian approach to interfaces in multimaterial flows (the ghost fluid method), Journal of computational physics 152 (1999) 457–492.

[53] X.-D. Liu, R. P. Fedkiw, M. Kang, A boundary condition capturing method for poisson's equation on irregular domains, Journal of computational Physics 160 (2000) 151–178.

[54] M. Kang, R. P. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, Journal of Scientific Computing 15 (2000) 323–360.

[55] Z. Li, K. Ito, The immersed interface method: numerical solutions of PDEs involving interfaces and irregular domains, SIAM, 2006.

[56] N. Moës, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, International journal for numerical methods in engineering 46 (1999) 131–150.

[57] A. Jameson, Time dependent calculations using multigrid, with applications to unsteady flows past airfoils and wings, in: 10th Computational Fluid Dynamics Conference, 1991, p. 1596.

[58] A. Belov, L. Martinelli, A. Jameson, A new implicit algorithm with multigrid for unsteady incompressible flow calculations, in: 33rd Aerospace sciences meeting and exhibit, 1995, p. 49.

[59] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted eno schemes, Journal of computational physics 126 (1996) 202–228.

[60] T. Belytschko, K. Mish, Computability in non-linear solid mechanics, International Journal for Numerical Methods in Engineering 52 (2001) 3–21.

[61] S. Y. Kadioglu, M. Sussman, S. Osher, J. P. Wright, M. Kang, A second order primitive preconditioner for solving all speed multi-phase flows, Journal of computational physics 209 (2005) 477–503.

[62] J. Teran, E. Sifakis, G. Irving, R. Fedkiw, Robust quasistatic finite elements and flesh simulation, in: Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA '05, ACM, New York, NY, USA, 2005, pp. 181–190.

[63] N. Kwatra, J. Su, J. T. Grétarsson, R. Fedkiw, A method for avoiding the acoustic time step restriction in compressible flow, Journal of Computational Physics 228 (2009) 4146–4161.

[64] T. J. Hughes, The finite element method: linear static and dynamic finite element analysis, Courier Corporation, 2012.

[65] R. De Borst, M. A. Crisfield, J. J. Remmers, C. V. Verhoosel, Nonlinear finite element analysis of solids and structures, John Wiley & Sons, 2012.

[66] X.-D. Liu, S. Osher, T. Chan, et al., Weighted essentially non-oscillatory schemes, Journal of computational physics 115 (1994) 200–212.

[67] T. Belytschko, W. K. Liu, B. Moran, K. Elkhodary, Nonlinear finite elements for continua and structures, John wiley & sons, 2013.

[68] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes, Acta Numerica (2020) 1–63.

[69] Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, F. Durand, Difftaichi: Differentiable programming for physical simulation, arXiv preprint arXiv:1910.00935 (2019).

[70] J. Zhuang, N. Dvornek, X. Li, S. Tatikonda, X. Papademetris, J. Duncan, Adaptive checkpoint adjoint method for gradient estimation in neural ode, arXiv preprint arXiv:2006.02493 (2020).

[71] H. J. Suh, M. Simchowitz, K. Zhang, R. Tedrake, Do differentiable simulators give better policy gradients?, in: International Conference on Machine Learning, PMLR, 2022, pp. 20668–20696.

[72] L. Metz, C. D. Freeman, S. S. Schoenholz, T. Kachman, Gradients are not all you need, arXiv preprint arXiv:2111.05803 (2021).

[73] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, The journal of machine learning research 15 (2014) 1929–1958.

[74] D. E. Stewart, Dynamics with Inequalities: impacts and hard constraints, SIAM, 2011.

[75] D. E. Stewart, J. C. Trinkle, An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction, International Journal for Numerical Methods in Engineering 39 (1996) 2673–2691.

[76] Z. Ferguson, M. Li, T. Schneider, F. Gil-Ureta, T. Langlois, C. Jiang, D. Zorin, D. M. Kaufman, D. Panozzo, Intersection-free rigid body dynamics, ACM Transactions on Graphics 40 (2021) 183.

[77] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, J. Z. Kolter, Differentiable convex optimization layers, in: Advances in neural information processing systems, 2019, pp. 9562–9574.

[78] Z. Kolter, D. Duvenaud, M. Johnson, Deep implicit layers - neural odes, deep equilibirum models, and beyond, NeurIPS Tutorial (2020).

[79] J. Liang, M. Lin, V. Koltun, Differentiable cloth simulation for inverse problems, in: Advances in Neural Information Processing Systems, 2019, pp. 772–781.

[80] Y.-L. Qiao, J. Liang, V. Koltun, M. C. Lin, Scalable differentiable physics for learning and control, arXiv preprint arXiv:2007.02168 (2020).

[81] J. Bolte, T. Le, E. Pauwels, T. Silveti-Falls, Nonsmooth implicit differentiation for machine-learning and optimization, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), Advances in Neural Information Processing Systems, volume 34, Curran Associates, Inc., 2021, pp. 13537–13549. URL: `https://proceedings.neurips.cc/paper/2021/file/70afbf2259b4449d8ae1429e054df1b1-Paper.pdf`.

[82] J. Teran, S. Blemker, V. N. T. Hing, R. Fedkiw, Finite volume methods for the simulation of skeletal muscle, in: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation, Citeseer, 2003, pp. 68–74.

[83] C. C. Paige, M. A. Saunders, Solution of sparse indefinite systems of linear equations, SIAM journal on numerical analysis 12 (1975) 617–629.

[84] C. Lanczos, An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, United States Governm. Press Office Los Angeles, CA, 1950.

[85] C. M. Bender, S. A. Orszag, Advanced mathematical methods for scientists and engineers (International Series in Pure and Applied Mathematics), McGraw-Hill, 1978.

Appendix A: Adam Oscillations Near Convergence

In this paper, we took an extremely thorough approach to the numerical examples (both those illustrated in the paper and those omitted for brevity) in order to provide convincing evidence for some of our (not so obvious) claims. In doing so, we discovered a peculiarity with Adam optimization that does not seem to be addressed in the literature in spite of its extreme popularity for training neural networks. The Adam update contains a division of a so-called first moment by a so-called second moment, where both moments are calculated by averaging new information with older information (as is typical for momentum-style optimization methods). Typically, the default parameters (for Adam) put more weight on the new information when including it into the first moment than when including it into the second moment. This allows a disturbance to increase the numerator faster than the denominator. Under normal circumstances, this works well; however, when both the numerator and denominator are very small (as occurs during convergence), this can lead to the numerator growing faster than the denominator (creating oscillations that disturb the convergence).

We illustrate this in Figure 45, which is the example from Figure 5 run for a longer number of parameter updates. The value of L jumps significantly after about 2500 parameter updates, and both $\frac{\partial L}{\partial \hat{b}}$ and $\frac{\partial L}{\partial \hat{c}}$ return to having relatively large magnitudes. Note that $\tilde{c}$ remains bounded away from zero during the disturbance (and thus is not the cause for this behavior). In this particular example, the solution is not too adversely affected, but we have observed other examples that are more dramatic.

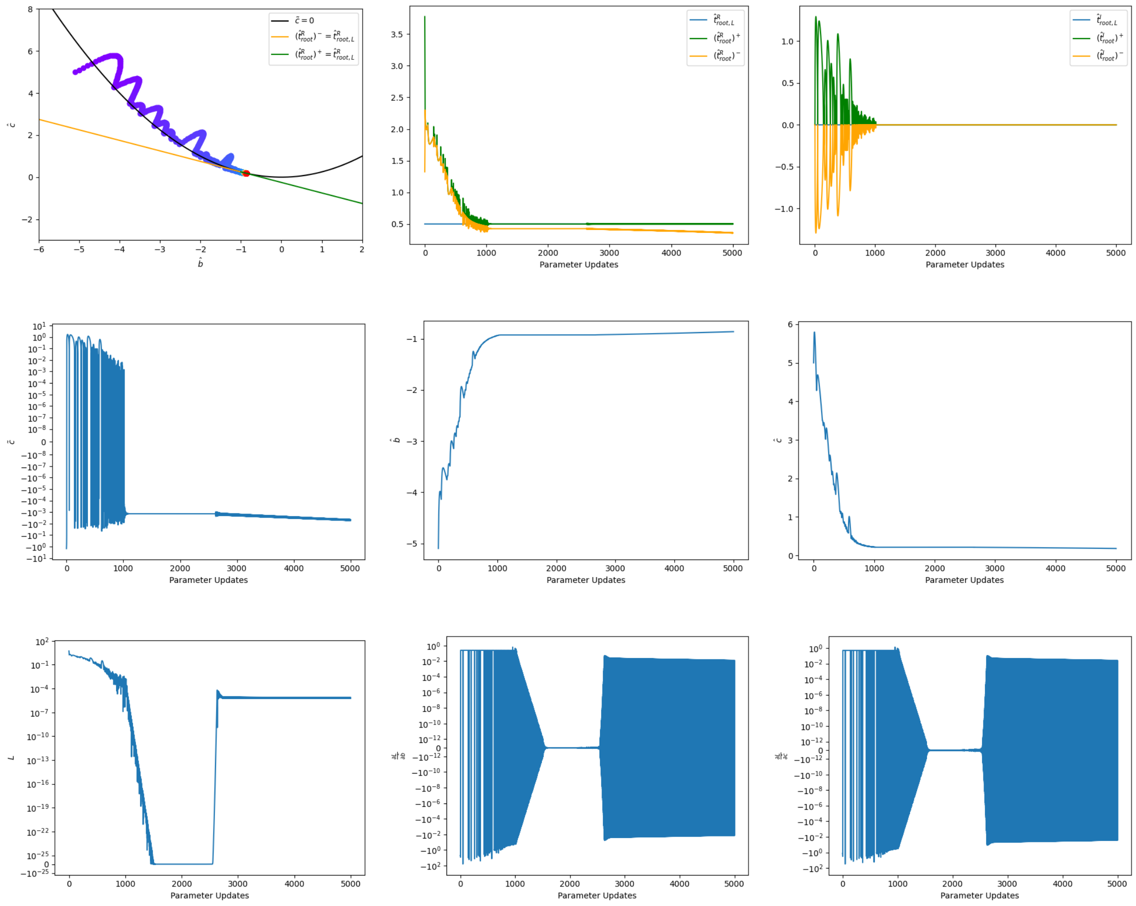


Figure 45: Adam Oscillations Near Convergence.

Appendix B: Failure of L-BFGS

Although we focused on the use of Adam, one might also consider so-called second order optimization methods that seek to estimate Hessians; however, given the various issues addressed in the paper, this seems significantly more difficult. Figure 46 shows the results obtained using the default implementation of L-BFGS in Pytorch [33] on the examples shown in Figures 4, 5, and 10. For the example from Figure 4, L-BFGS performs better than Adam (as expected in this simple case). For the examples from Figures 5 and 10 where large gradients near the $\tilde{c} = 0$ parabola are problematic, L-BFGS fails to converge. For the example from Figure 5, L-BFGS starts out moving in the right direction, but the Hessian approximation deteriorates to the point where the iterates eventually stall out and even start to move in the wrong direction. For the example from Figure 10, the initial large gradient causes L-BFGS to jump into the complex region where it gets stuck (even after 200,000 iterations). Recall from Remark 8.1 and equation 62 that the gradient is constant in this region, meaning that L-BFGS is not getting any new information (which is needed to update its current poor approximation of the Hessian).

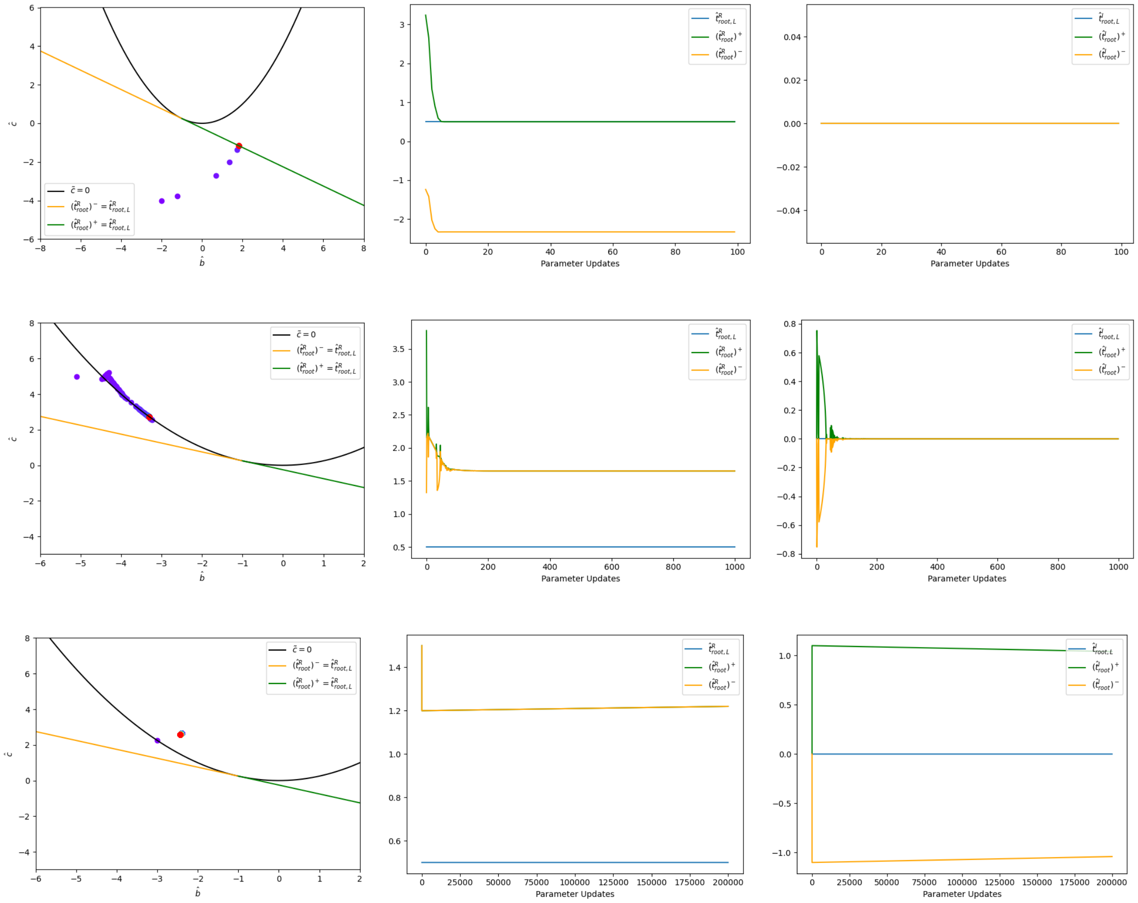


Figure 46: The results obtained using the default implementation of L-BFGS in Pytorch on the examples from Figures 4 (top row), 5 (middle row), and 10 (bottom row).