

# Chaotic Convergence of Newton's method

Allen, Jont B.

Monday 17<sup>th</sup> July, 2023

## Abstract

### Problem statement:

In 1680 Newton proposed a algorithm for finding roots of polynomials. His method has since evolved, but the core concept remains intact. Here we briefly review this evolution, and consider the question of convergence. First, and most important, does his method always converge? From numerical experiments, it does converge for the vast initial guesses. Thus a key and important question: does the convergence depend on this initial guess? This question was carefully evaluated by Wilkinson, who studied conditions of sever divergence.<sup>1</sup> Second, if not, assuming it does converge, what are the necessary conditions for convergence. Here we investigate the conditions for convergence. In the following we assume a monic polynomial of degree  $N$ . The fundamental theorem of algebra states that every polynomial of degree  $N$  has  $N$  roots, which are typically complex  $s_n \in \mathbb{C}$ .

**Methods:** Newton's method is derived in Appendix A. Newton's method may be applied to any *complex analytic function*, thus holds for solutions of linear differential equations (Allen, 2020). The derivation is based on a Taylor series expansion in the Laplace frequency  $s = \sigma + j\omega$ . The convergence of Newton's method depends on the important concept known as the *Region of convergence* (RoC).

Because Newton's algorithm converges within the RoC for any complex analytic function, it converges to one of the roots of the polynomial when the nearest root  $s_r$  ( $s \in \mathbb{C}$ ) inside the RoC out to the nearest pole. This follows because every complex-analytic point on the complex plan has a region of convergence (Allen, 2020). We show that on the boundaries of the RoC regions, the method becomes hyper-sensitive to the initial condition (i.e., guess  $s_0$ ).

**Findings:** Under certain conditions, non-linear (NL) limit-cycles appear, resulting in a reduced rate of convergence to a root. Since Newton's method is inherently complex analytic (that is, linear and convergent), it is important to establish the source of this NL divergence. We show that this NL effect is due to violations of the Nyquist Sampling theorem, also known as *aliasing*. Aliasing is a well-known concept in discrete-time signal processing, due to the sampling a signal at less than twice its highest frequency. When a time signal is under-sampled, frequencies above the sampling frequency are shifted down in frequency. This is the definition of this nonlinear phenomena, known as aliasing, which follows from the reduced-sampling (Allen, 2020, p. 153,225).

Here the conditions and method for uniform convergence are explored. We propose a complex step-size  $\eta = ae^{j\phi}$  which we adaptively adjusted, greatly reducing, even removing the nonlinear effects of aliasing. Introducing this adaptive step-size ( $\eta$ ) is known as the *damped Newton's method* (Galántai, 2000, p. 25). In the limit as  $\eta \rightarrow 0$ , the NL aliasing is avoided. This naturally follows from the complex-analytic properties of Newton's method.

Under some special conditions, a slight change in the initial guess  $s_0$  ( $n = 0$ ) can result in the  $n + 1$  estimate of the root ( $s_{n+1}$ ) to cross an RoC boundary, resulting in NM to divert its initial path to a alternate root. When this happens, the change in the step  $\delta = s_{n+1} - s_n$  is unpredictable, and possibly even chaotic. It is this condition that is the source of a convergence instabilities due to aliasing, possibly leading to a limit cycle.

This chaotic behavior is a main topic of this document.

The  $N$  *regions of convergence* (RoC) are investigated. From the definition of the RoC, these contiguous naturally existing regions are defined over all  $s_0 \in \mathbb{C} < \infty$ . That is, every possible  $s_0$  comes from one of the  $N$  RoC regions. This naturally happens as  $\eta \rightarrow 0$ , since Newton's method is complex analytic (the step size is the ratio of two polynomials with different roots). The magnitude of the step-size  $a$  may also be reduced to avoid crossing the boundary between two RoCs. As the step-size is reduced, the trajectory naturally moves away from the poles and the smaller it is, the greater the effect. Thus reducing  $|\eta|$  is highly effective, but in theory, could slow the convergence. For this reason adaptively setting  $|\eta|$  is likely the optimum balance, to minimize the computation while avoiding NL aliasing, which results from any crossing of RoC boundaries.

Besides reducing  $|\eta|$ , one can modify its angle  $\phi$ , redirecting the trajectory away from any RoC boundary, to avoid crossings it. However we have not implemented this possible method.

**Conclusions:** We numerically demonstrate that reducing the step-size always results in a more stable convergence. The down side is that it always results in a sub-optimal convergence. It follows that a dynamic step size would be ideal, by slowly increasing the step-size until it fails, and then decreasing it until it converges. A balance of the two methods seems like a potential solution, but this remains unproven.

We show that when  $\eta = 1$ , depending on  $s_0$ , the solution can cross an RoC boundary (i.e., diverge). In such cases the target root ill change, resulting in a chaotic trajectory. Examples are provided. Depending critically on  $s_0$ , as long as the RoC region remains the same, every iteration converges.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Wilkinson's\\_polynomial](https://en.wikipedia.org/wiki/Wilkinson's_polynomial)

# 1 Introduction

Newton's method (NM) is a venerable complex-analytic mathematical algorithm for finding roots of any monic polynomial  $P_N(s)$ , where  $N$  is the degree and  $s = \sigma + \omega j \in \mathbb{C}$  is the Laplace frequency. However the convergence properties of NM are controversial<sup>2</sup> (Stewart, 2012, p. 347). In this report we shall investigate why such a controversy developed, and discuss how to assure convergence. In our experience, given some care, the method always converges to a root.

Every initial guess  $s_0$  on the plane of a complex analytic function is uniquely associated with one of the  $N$  roots of that function, which in turn are associated a unique region of convergence (RoC). This follows from the complex analytic property of a function (those that may be expanded in a complex-analytic Taylor series). When the trajectory of NM jumps to a different RoC, corresponding to a different root, it has been interpreted as a failure to convergence. What then happens in the examples presented here, the iteration still converges, but to a different root. NM contains properties that are similar to *dynamic analysis*, a mathematical science first introduced by Poincaré.

This question of the convergence of NM was recently explored in Allen (2020), where no instability or limit-cycles were observed. An explanation is due: Newton's method was modified by applying an *adaptive step-size*  $\eta$ , (Galántai, 2000, p. 25), a widely recognized contemporary technique in the engineering numerical analysis literature.<sup>3</sup>

A properly chosen adaptive step-size stabilizes the convergence, by forcing the convergence to remain in the target RoC. We show that its easy to detect when the divergence of the step, which should be monotonically decreasing. The onset of limit cycles are easily detected, and easily stabilized by reducing the magnitude of  $\eta$ .

Here we show that random jumps and limit-cycles can occur when the step-size  $\eta = 1$ . When the adaptive step-size  $\eta$  is sufficiently small, we show that the iteration always converges. The key here is to adaptively modify the magnitude of  $\eta$ , thereby constraining the trajectory to the initial RoC.

In Allen (2020, Fig. 3.2), two examples were provided using a fixed step-size ( $\eta = 0.5$ ) and a random initial guess. The details of the step-size used by Allen (2020) was not discussed. One of these figures is presented in Fig. 1 (**LEFT**).

While most of the curves seem to converge to a root, there are some small percentage (e.g., 1%) of cases where the trajectories take huge jumps to random locations in the complex plane. We shall show that these jumps occur when the trajectory approaches any of the poles of Newton's method, that is, at the roots of  $P'_N(s) = \frac{d}{ds}P_N(s)$ . They then always converge to a different root from that new initial starting value. Near a pole the step can be arbitrarily large, depending on how close the step comes to the pole (Boas, 1987). This be predicted if one carefully designs  $s_0$ , such that for a fixed  $\eta$ , the trajectory heads directly into a pole. We shall show that these poles are the source of these divergent points, which may be easily detected, as described below.

In Fig. 1 (**LEFT**), the five RoC regions are color coded, with each RoC region associated with one of the  $N$  roots. Due to the complex analytic nature of the RoCs, every point in the RoC is a valid initial condition ( $s_0$ ). However this is limited by the numerical accuracy of the computer. Also the convergence depends on the size of the steps, defined as  $s_{n+1} - s_n, s \in \mathbb{C}, n \in \mathbb{N}$ , which typically decreases in magnitude with  $n$ . An exception occurs if  $s_{n+1}$  approach one of the  $N - 1$  poles of NM, causing the step to abruptly diverge. The properties of this small subset of initial conditions depends critically on the *step size*  $|\eta|$ , which is a key topic of this article.

For most initial guess  $s_0 \in \mathbb{C}$  the iteration simply converges to a root, independent of  $|\eta| \leq 1$ . In fact for most starting values the solution converges for  $|\eta| = 1$ . However for  $s_0$  values near the RoC boundary between two roots, the dependence is highly dependent on  $|\eta|$ , and can even be chaotic. This happens when  $s_0$  defines a path that heads directly for a pole. In these cases the trajectory will be hypersensitive to both  $s_0$  and  $|\eta|$ . The RoC regions are well defined non-overlapping complex-valued analytic regions. The convergence of NM critically depends on the step-size when  $s_0$  is close to the RoC boundary. Even when  $|\eta| \ll 1$ , the convergence can become nonlinear (NL), causing the iteration to wildly diverge. These observations are supported by several detailed numerical examples.

| Figure                | $P_N(s) =$   | $\Re s_r$                        | $\Im s_r$                    |
|-----------------------|--|----------------------------------|------------------------------|
| Fig. 1a; <b>LEFT</b>  | $s^5 - (13 + 0.5j)s^4 + (66.2 + 5j)s^3 - (164.2 + 20.12j)s^2 + (195 + 4.7j)s + 87.5 + 31.2j$ | [4, 3, 3, 2, 1]                  | [1, -2, 2, -1, 1]/2          |
| Fig. 1b; <b>RIGHT</b> | $s^5 - s - 1 \leftrightarrow [1\ 0\ 0\ 0\ -1\ -1]$   | [1.17, 0.18, 0.18, -0.76, -0.76] | [0, 1.08 -1.08, 0.35, -0.35] |

Table 1: Properties of the polynomials for each figure. Figures 2-4 are the same as Fig. 1b (rounded). The polynomial coefficients must be wrong because the roots are not symmetrical. The **LEFT** is from `./M/fig3dot2.m` [FIGS/NewtonJPD2.eps](#) while **RIGHT** is from `Demo-JPD/NewtonMS.m` [FIGS/NewtonJPD2.pdf](#)

For example, in Fig. 1 (**LEFT**), the red region corresponding to the root at  $(2.0 - 0.5j)$  has a long narrow "RoC stream" for initial guesses  $s_0$  south-east of  $(4.5 - 1j)$ . There is a second narrow neighboring related parallel (green) stream just north of the red stream, for  $s_0$  values at  $(4.5 - 1.0j)$ , which are the RoC for the root at  $(3 + 1j)$ .

ERROR: The two  $s_0$  are the same.

<sup>2</sup>[https://en.wikipedia.org/wiki/Newton's\\_method#Failure\\_of\\_the\\_method\\_to\\_converge\\_to\\_the\\_root](https://en.wikipedia.org/wiki/Newton's_method#Failure_of_the_method_to_converge_to_the_root)

<sup>3</sup>[https://en.wikipedia.org/wiki/Adaptive\\_step\\_size](https://en.wikipedia.org/wiki/Adaptive_step_size)

While this may seem obvious given this figure, I am not aware of any discussion of such distortion of the RoC's. Presently I do not know of any method for predict this remapping of the RoC regions. The conditions for Fig. 1 are provided in Table 1.

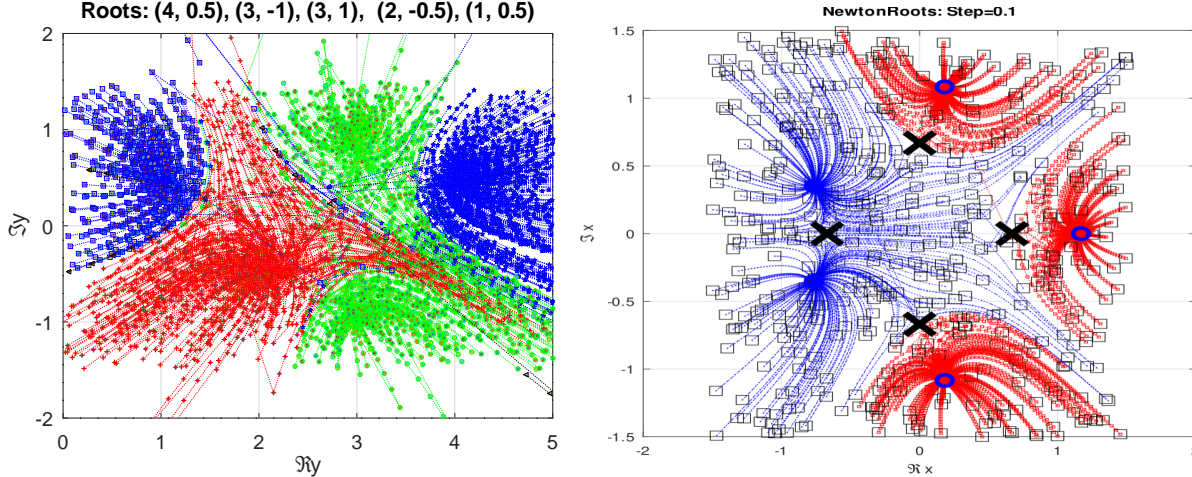


Figure 1: Example LEFT: 1a This figure is taken from Allen (2020, p. 78). It is a plot of a thousand trajectories for Newton's method, with a random initial guess, taken from the complex plane, between  $[0,5]$  along the real axis and  $\pm 1.5$  along the imaginary axis, for a polynomial having  $N = 5$  complex coefficients. Note the long straight lines that occasionally appear in the figure. Example RIGHT: 1b shows the poles and zeros of the polynomial having coefficients  $C = [1, 0, 0, 0, -1, -1]$  with random starting points, for 200 iterations of Newton's Method. In this case the roots (o) and poles (x) are superimposed on top of the trajectories of Newton's method. A step-size of  $\eta = 0.1$  is used to reduce the NL aliasing. RIGHT: Fix label font-size; Need to add roots in title. LEFT: Add o,x.

## 1.1 Convergence of Newton's method

Given a monic polynomial of degree  $N \in \mathbb{N}$  as  $P_N(s) = s^N + c_{N-1}s^{N-1} + c_{N-2}s^{N-2} + \dots + c_0$ , and its derivative define  $P'_N(s) \equiv dP(s)/ds$ , we may define Newton's method.

Newton's formula is derived in Appendix A as an iteration in integer  $n \in \mathbb{N}$  over complex frequency values  $s_n \in \mathbb{C}$ , modified with an adaptive step-size  $\eta \in \mathbb{C}$  ( $|\eta| < 1$ )

$$s_{n+1} = s_n - \eta \frac{P_N(s_n)}{P'_N(s)} \quad (1.1)$$

With a little algebra,

$$\frac{s_{n+1} - s_n}{\eta} = -\frac{P_N(s_n)}{P'_N(s)}.$$

Taking the limit  $\eta \rightarrow 0$  reveals the complex-analytic expression

$$\frac{ds}{d\eta} \equiv \lim_{\eta \rightarrow 0} \left( \frac{s_{n+1} - s_n}{\eta} \right) = -\frac{P_N(s)}{P'_N(s)}. \quad (1.2)$$

The right hand side is the reciprocal of the log-derivative of  $P_n(s)$ , and the left hand side is the pre-limit form of the slope of the Laplace frequency ( $s = \sigma + j\omega$ ) wrt  $\eta$ . Because the  $\ln(\cdot)$  function is complex analytic, the log-derivative of polynomial  $P_N(s)$ , namely  $\frac{d}{ds} \ln P_N(s) \equiv \frac{1}{P_N(s)} \frac{dP_N}{ds} = P'_N(s)/P_N(s)$ , must also be complex-analytic.

When expressed as a complex-analytic function, we note that the expression, which is a limit form of Newton's method, always converges for sufficiently small  $\eta$ .

**In summary:** Determining the RoCs for NM by analytic methods seems difficult, since the function  $S_N(s_n)$  has poles, confounding the locations of the RoC regions. Based on Fig. 1, they are complicated. If  $s_n$  approaches one of these the poles, the update can become arbitrarily large, depending on how close  $s_n$  is to the pole. If the step-size  $S_N(s_n)$  is within the RoC, this does not seem occur. When the value of  $s_{n+1}$  falls outside the RoC there can be an arbitrary increase in step-size. Normally this will not happen, since when  $s_n$  approaches a pole, since  $s_{n+1}$  is naturally "pushed" away from the pole, as may be seen in both panels. Newton's method then becomes

$$s_{n+1} = s_n - \eta \frac{1}{5} \frac{s_n^5 - s_n - 1}{s_n^4 - 1/5}. \quad (1.3)$$

## 2 Examples of Newton's method

**Example 1:** We start with the monic polynomial of Example 1b, Fig. 1 (RIGHT), for  $N = 5$ ,

$$P_5(s) = s^5 - s - 1. \quad (2.1)$$

In this case monic  $\frac{1}{5}P'(s) = s^4 - 1/5$ , has four roots, shown as black bold X symbols. These are the poles. The black bold circles are the zeros we seek.

As shown in Example 1b, Fig. 1 (RIGHT), given the initial guess  $s_0$  and step-size  $\eta$ , as  $n \rightarrow \infty$ ,  $s_n$  approaches a unique root  $s_r$ . Complex  $s_{n+1}$  is the  $n+1$  the estimate of the root given the  $n$  estimate  $s_n$ , defined by Eq. 1.1.

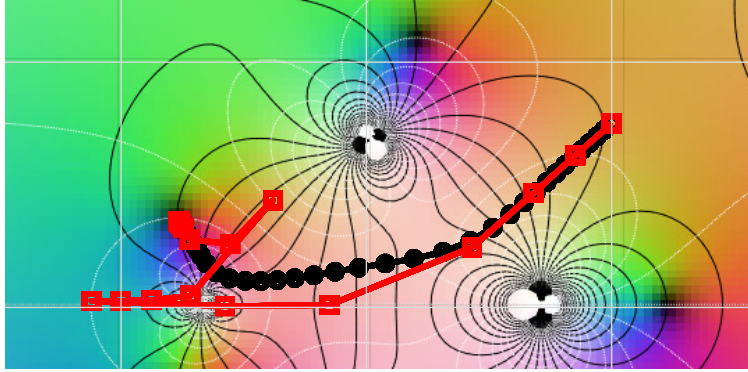


Figure 2: This is a zoomed-in version of Example 1b (RIGHT), presented as a colored plot (Allen, 2020, p. 168) of  $S_N(s)$ , for  $P_5(s) = s^5 - s - 1$  ( $[1, 0, 0, 0, -1, -1]$ ). The magnitude of each Newton-step  $S_N(s)$  is coded by the brightness, and the phase ( $\angle(L_N(s))$ ) by the color (hue). Thus dark regions are zeros (roots of  $P_n(s)$ ) while the white regions are poles of  $S_N(s)$  (roots of  $P'(s)$ ). Two trajectories of Newton's method are shown, as the black circles and red squares. The initial value for both cases is  $s_0 = 1 + 0.75j$ . The black circles correspond to  $\eta = 0.1$ , while the red squares, for  $\eta = 0.5$ , form a brief limit cycle. It is clear that the limit cycle depends on the step-size  $\eta$ , and becomes nonlinear when  $\eta = 1$ . With the smaller step-size ( $\eta = 0.1$ ) the recursion remains complex analytic, thus is linear, and converges to root  $s_3 = -0.765 + 0.3525j$ . The vertical white lines are at  $\{-1.0, 0, 1.0\}$ , and the horizontal lines are at  $\{0, 1.0\}$ . The polynomial coefficients are  $P_5(s) = [1, 0, 0, 0, -1, -1]$  having roots  $s_r = [1.167, 0.181 \pm 1.084j], -0.765 \mp 0.3525j$ . The real poles (roots of  $P'_5(s_r) = 0$ ) are  $\pm 1/5^{0.25} \approx \pm 0.669$ , while the imaginary poles are  $\pm 0.669j$ .

The example shown in Fig. 2 is a zoomed-in version of Example 1b, Fig. 1 (RIGHT). These conditions, which depend on both the precise value of the initial guess  $s_0$  and the step size  $\eta$ , result in a limit-cycle. When the step-size is  $\eta = 0.5$  (red squares),  $s_{n+1}$  over-shoots the pole, resulting in a limit cycle. When the step-size is reduced to  $\eta = 0.1$  (black circles), the trajectory is stable (avoids the pole) and converges smoothly to the complex-valued zero at  $0.181 + 1.084j$ .

### 2.1 Discussion of Example 1b of Fig. 2

To study the convergence and limit-cycles it is helpful to use numerical methods and look at specific step-sizes and starting points, where that the trajectory approaches a pole (root of  $P'_n(s)$ ).

In practice, once near a root, it takes only a few steps to converge. In experimental trials, fewer than 10 steps can give double-precision floating-point machine accuracy.

As discussed in the figure caption, if  $s_n$  is close to a root  $s_r$  of  $P'_N$  (i.e., a pole), the recursion dramatically fails, because the step becomes arbitrarily large, forcing the next trial to a random location in the  $s$  plane, denote  $\tilde{s}_r$ . In such cases the solution typically converges to a different root (RoC). It is not difficult to detect these large random steps by monitoring  $|s_{n+1} - s_n|$ , which must be monotonically decreasing with increasing  $n$ .

Figure 2 shows two paths with the same initial condition  $s_0 = 1 + 0.75j$  but different step-sizes. The utility of the reduced step-size is clear from Fig. 2.

If we start the iteration with the larger step size, the path develops into a NL limit-cycle near the pole at  $-0.669$ . It is a combination of the large steps and the proximity to the real pole that results in the nonlinear limit-cycle. On the 10 step it comes out of the limit cycle, and after 10 more steps, has converged to the root at  $-0.766 + 0.3525j$ . When the step-size  $\eta$  is reduced from 0.5 to 0.1, as the path approaches the positive pole, it moves away, avoid the limit-cycle. With steps sizes of 0.2, 0.3 it becomes captured by the pole. The black circles provide an example of a smooth analytic trajectory, while the red-squares is chaotic.

With the step-size of 0.9 (not shown), the trajectory is similar to that of 0.5, but after 5 steps it is in well within a different RoC, corresponding to the zero at  $-0.8 - 0.3j$ . After 20 steps, the error is less than 1%.

In summary, it still converges, but much more slowly, since the NL becomes greater. Thus the convergence time may be a crude quantitative measure of the NL. The smoothness (e.g., spectral properties) of the trajectory may be more appropriate. This NL result is due to the reduced sample step size, is known as *aliasing*.

If one assumes that the initial guess is real ( $s_0 \in \mathbb{R}$ ) Octave/Matlab evaluates the polynomial using real arithmetic, forcing the estimate  $s_{n+1} \in \mathbb{R}$  (Allen, 2020). Thus the iteration cannot converge when  $s_r \in \mathbb{R}$  and  $s_r \in \mathbb{C}$ .

**Example 2:** This example is for  $P_3(x) = x^3 - x + 1$  ( $C = [1, 0, -1, 1]$ ), an example where Newton's method appears to fail. This example has two imaginary roots,  $0.66236 \pm 0.56228j$ , and a real root  $-1.32472$ . If the initial guess is taken to be  $s_0 = 1$ , the recursion proceeds using real arithmetic (Matlab and Octave). Due to the restriction that the computation is real, the solution is forced to the real line, where it limit cycles between 1.155 and 0.694.

If  $x_o = j$ , the solution converges in 3 steps to the upper complex root  $x_3(3) = 0.639379 + 0.509792j$  If one starts the iteration with an imaginary component at  $1 + j10^{-6}$ , the iteration converges to the imaginary root in 13 steps.<sup>4</sup>

verify!

**In summary:** Roots  $s_r \in \mathbb{C}$  may be found by a recursion that denotes a sequence  $s_n \rightarrow s_r \in \mathbb{C}$ ,  $n \in \mathbb{N}$ , such that  $P_N(s_n) \rightarrow 0$  as  $n \rightarrow \infty$ . As shown in Fig. 2, solving for  $s_{n+1}$  using Eq. A.3 always gives one of the roots, due to the analytic behavior of the complex logarithmic derivative  $P'_N/P_N = d \ln(P_n(s))$ .

When there are no limit cycles, with every step,  $s_{n+1}$  is closer to the root, finally converging to the root in the limit. As it comes closer, the linearity assumption becomes more accurate, resulting in a better approximation and thus a faster convergence.

Equation A.3 depends on the reciprocal of the log-derivative  $L(s) \equiv d \log P(s)/ds = P'(s)/P(s)$ ,  $\mathcal{P} \in \mathbb{C}$ .

## 2.2 Newton's method applied to functions other than polynomials

Proved:  
Monday 17<sup>th</sup> July, 2023

**Example 2:** Example of Plank's formula for Black Body radiation.

Planks famous BB radiation formula is (Kuhn, 1978; Allen, 2020)

$$S(\nu) = \frac{\nu^3}{e^{h\nu/kT} - 1}. \quad (2.2)$$

In this historically important example, because the function is real (it is not complex analytic), the spectrum only has one pole, at  $\nu = 0$ . This formula is known to match the experimental data of the smoothed (non-analytic) black-body power spectrum (Haar, 2016).

If we replace the real frequency  $\nu$  with the negative Laplace frequency  $-s = -\sigma - \omega j$ , Eq. 2.2 becomes

$$S(-s) = \frac{-s^3}{e^{-\hbar s/kT} - 1}, \quad (2.3)$$

which is complex analytic, thus has a causal inverse Laplace transform. To use Newton's method we must compute NM update  $L(s_n)$  where  $s_n$  is the present estimate of the root, defined as the the reciprocal of the logarithmic derivative (see derivation in Apdx. A). Taking the log followed by its derivative wrt  $s \in \mathbb{C}$ , gives

Needs a careful verification

$$\begin{aligned} \frac{1}{L(s)} &\equiv \frac{d}{ds} \ln S(-s) \\ &= \frac{d}{ds} [-3 \ln s + \ln(e^{-\hbar s/kT} - 1)], \\ &= -\frac{3}{s} - \frac{\hbar}{kT} \cdot \frac{e^{-\hbar s/kT}}{e^{-\hbar s/kT} - 1}, \end{aligned}$$

or as reported by Arbab (personal communication)

$$\frac{1}{L(s)} = -\frac{3}{s} - \frac{\hbar}{kT} \cdot \frac{1}{1 - e^{\hbar s/kT}}. \quad (2.4)$$

The inverse  $\mathcal{L}\mathcal{T}$  of the causal expression is (Allen, 2020, p. 321)

$$-3u(t) - \frac{\hbar}{kT} \sum_{n=1}^{\infty} \delta(t - n \frac{\hbar}{kT}) \leftrightarrow -\frac{3}{s} - \frac{\hbar}{kT} \cdot \frac{e^{-\hbar s/kT}}{1 - e^{-\hbar s/kT}}. \quad (2.5)$$

Thus there is a first order pole at  $s = 0$  and poles at  $h\nu_n/kT = 2\pi n$  for  $n \in \mathbb{N}$ . The discrete frequencies account for the eigen-modes in the black body radiation, as discussed by Kuhn, Plank and Einstein (Haar, 2016).

The most important result here is that Eq. 2.3 is causal, since it has a causal inverse  $\mathcal{L}\mathcal{T}$  (Allen, 2020). The application of NM to Plank's famous formula can be used to make it complex analytic, by replace  $\nu$  with the Laplace

<sup>4</sup>Octave program: ./DEMO-MATHINSIGHT/NEWTONEXP1.M

frequency  $s = 2\pi\nu_n j$  and  $h$  by  $\hbar$ , making the real frequency  $\nu$  complex analytic. It is well established that complex analytic functions of the Laplace frequency  $s$  are *causal* (zero for negative time). In the case of Eq. 2.3,  $S(-s)$  is causal, due to the Laplace transform relation of the exponent (Allen, 2020)

$$\delta(t - \tau_o) \leftrightarrow e^{-s\tau_o}.$$

Here the time delay  $\tau_o = \hbar/kT = (6.63/2\pi k) \cdot 10^{-11}$  [s], 6,280 [GHz],  $\lambda \approx \frac{\pi}{2} 10^{-11}$  [m], or  $\frac{\pi}{20}$  [Å], and  $T^\circ$  [K] is the temperature.

Newton's method uses the reciprocal of  $L(s)$  to find the  $s_r$  ( $S(s_r) = \infty$ ), given by

$$N(s_r) = 1 - e^{-\hbar s_r/kT} = 0. \quad (2.6)$$

There are an infinite number of such roots, since the roots are  $\hbar s_r/kT \approx 2\pi j$ . These poles are the missing discrete spectral lines (atomic resonances), required by quantum mechanics.

Applying Newton's method gives

$$x_{n+1} = x_n - \frac{e^x - 2}{e^x} = s_n - (1 - 2e^{-s_n}).$$

Since  $e^x$  is entire, there are no convergence issues.<sup>5</sup> Since  $x \in \mathbb{C}$ , the imaginary part quickly decays to zero, and depending on the starting condition, approaches one of the infinite number of solutions, within a few steps.

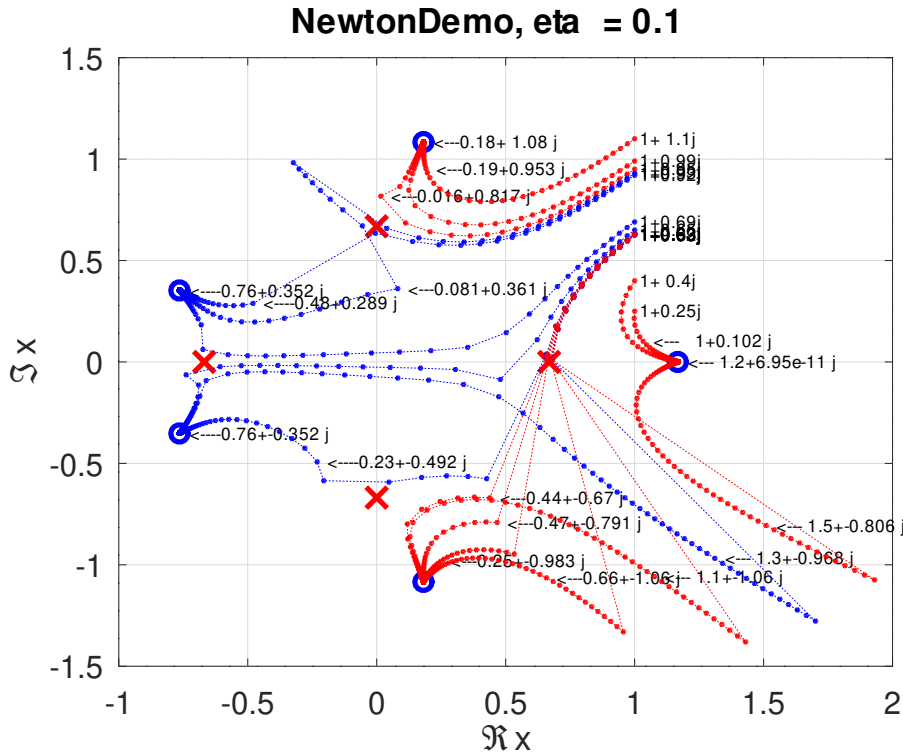


Figure 3: This numerical experiment for polynomial coefficients  $[1, 0, 0, 0, -1, -1]$  (the same as shown on the right panel of Fig. 1) having a step-size of 0.1), reveals the inner workings of Newton's method. We number the roots counter-clockwise from 1-5, with  $s_1 = 0.18123 + 1.08395j$  and  $s_5 = 1.2$ . Seventeen different starting values ( $s_0$ ) have been carefully chosen, to determine the RoCs associated with each  $s_0$ . All the starting values are of the form  $s_0 = 1 + j\beta$ , where each  $\beta$  and the converged root are indexed in Table 2. The root index goes from 1 to 5, counting counter clockwise from the northern most root. The scattering angle is determined by the residue of the scattering pole. Each curve is labeled twice, once at the starting point and once at another point somewhere along the trajectory. The carefully evaluated case is for starting points between  $1 + 0.62j$  and  $1 + 0.5999j$ , which converge to dramatically different RoCs due to squarely hitting the positive real pole at 0.6 ( $s_0 = 1 + j 0.6 \pm 0.001$ ). [NewtonDemo-5thOrder](#); [Code: ./M/NEWTONDEMO.M](#)

### 2.3 Example 3:

The effect of  $s_0$  is shown in greater detail in Example 3, as shown in Fig. 3. When the value to  $s_0$  is finely tuned, such that the trajectory intercepts a pole, a NL limit-cycle will result.

The Gauss-Lucas theorem<sup>6</sup> comes into play at this point (Allen, 2020). This theorem says that the convex hull of the roots of a polynomial bound the roots of its derivative. This theorem is relevant to the convergence of Newton's method. Galántai (2000) has 75 relevant citations, many citing the same problems addressed here. The key to avoiding the troublesome limit-cycles is to detect them, and then reduce the step-size.

<sup>5</sup><https://www.quantamagazine.org/how-mathematicians-make-sense-of-chaos-20220302/>

<sup>6</sup>[https://en.wikipedia.org/wiki/Gauss-Lucas\\_theorem](https://en.wikipedia.org/wiki/Gauss-Lucas_theorem)

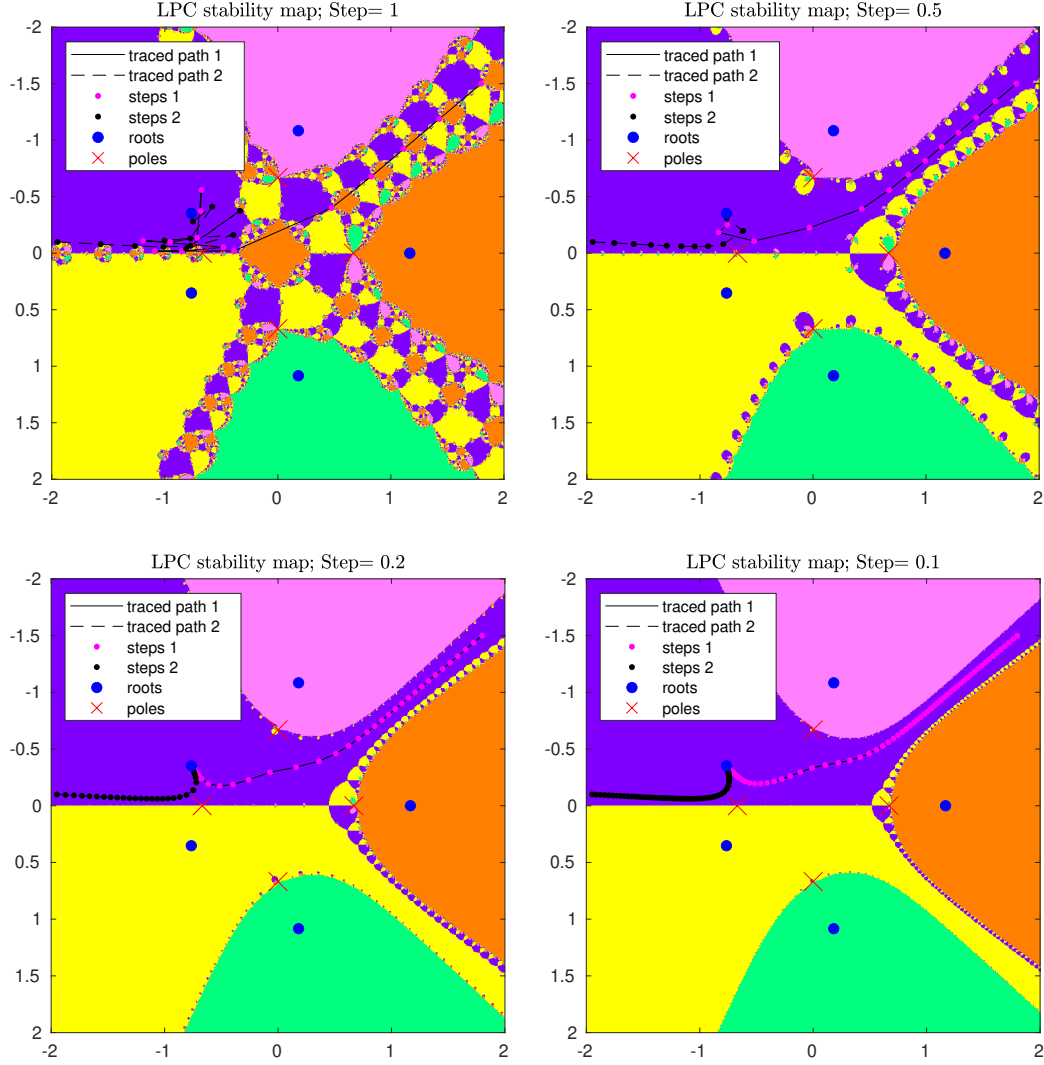


Figure 4: Four colorized plots for  $P_N = [1, 0, 0, 0, -1, -1]$  showing the  $N = 5$  regions of convergence and two trajectories, for  $s_0 = 1.8 - 1.5j$  and  $-1.95 - 0.1j$ . The four step-sizes are  $\eta = \{1.0, 0.5, 0.2, 0.1\}$  (note that the imaginary axis is reversed). The fractal regions reside on the RoC boundaries, the sizes of which depend on the step-size, with the step-size of  $\eta = 1.0$  (Upper-Left) resulting in large fractal regions. Reducing the step-size to  $\eta = 1/2$  dramatically reduces the fractal regions. For  $\eta = 0.1$  they almost disappear, except at  $0.5 + 0j$ . In the dark RoC (purple) corresponding to root  $-0.76 - 0.352j$ , two trajectories are shown. For the step-size of 1, a limit cycle is seen, for both trajectories. For the other step-sizes  $[0.5, 0.2, 0.1]$ , there are no limit cycles. As the trajectories approach the negative real pole, labeled as the red  $\times$ , they head for the root at  $-0.76 - .352j$ . In summary: 1) limit cycles are wasted steps, easily fixed by reducing the step-size. 2) Given a smaller step-size, the fractal regions shrink, but never totally disappear. 3) Detecting a limit cycle is easy because the path reverses (oscillates). An obvious method for avoiding limit cycles is to detect that the boundary has been crossed, corresponding to a different root, and restart the iteration with a reduced step-size, at step  $s_n$  (or  $s_{n-1}$ ). [fig:LPC1](#)

The following quote is from Galántai (2000, p. 39):

The possibility that a small change in  $s_0$  can cause a drastic change in convergence indicates the nasty nature of the convergence problem. The set of divergence points of the Newton method is best described for real polynomials.

As demonstrated in Fig. 3, we agree with Galántai’s first point. His second remains open. Is a “real polynomial” one with real coefficients or real roots?

For example in Fig. 1, the red “stream” corresponding to the root near  $(2 - 0.5j)$  has an interesting long narrow “RoC-stream,” converging from the lower-right quadrant, first seen at  $(4.5 - 1j)$ . There is a second neighboring parallel (green) RoC-stream just north of the red stream, first seen at  $(4.5 - 0.9j)$ .

I am not aware of any discussion in the literature of this distortion of the RoC regions, bound to Newton’s method. Presently I know of no way to predict the conformal remapping of the RoC regions for NM, other than tracking them, as done here. It seems likely that methods for doing must exist using modern analysis techniques, as discussed in Appendix B.

Table 2: *Table of starting values  $s_0 = 1 + \beta j$  use in Fig. 3, along with the RoC targeted root index, defined as #1 for the real root at  $0.2^{1/4}$ . Root #1 converges from  $s_0 = 0 + 1.25j$ , Root #2 is defined by counting counter-clockwise from #1, at  $0.18 + 1.08j$ , starting fro  $s_0 = 1 + 0.69j$ . Root #3 also converges from three values of  $\beta$ . Root #5 is the most carefully explored, starting from  $1 + \beta j$ . It is shown to converge to roots 1, 3, 4, 5, but not 2, which is reachable from very selective values of  $\beta$ . For other choices of  $\beta_0$ , all 5 roots can be reached, as shown in Fig. 3 for  $\eta = 1$ .*

[./M/ZvizDemo.m.](#)

| $\beta$          | root |
|------------------|------|
| 0.25, 0.4        | #1   |
| 0.95, 0.99, 1.1  | #2   |
| 0.69, 0.92, 0.93 | #3   |
| 0.65, 0.632      | #4   |
| 0.63, 0.631      | #5   |

In the example of Fig. 3,

$$s_{n+1} = s_n - \frac{0.1}{5} \cdot \frac{s_n^5 - s_n - 1}{s_n^4 - 1/5},$$

for 17 carefully chosen initial guess  $s_0 \in \mathbb{C}$ . For readability, each trajectory is color-coded either red or blue.

**Nonlinear limit cycles:** It is well documented that limit cycles are nonlinear. Newton’s method on the other hand is a linear recursion equation, with poles and zeros in the complex plane. The research question is “*Why does the complex-analytic linear equation become nonlinear?*” We show how the these NL limit-cycles may be easily avoided by removing (linearizing) the NL recursion once it is detected.

The suggested procedure will result in a net convergence speed-up, because the NL limit-cycle adds meandering rambling NL steps to the recursion. Thus removing the NL behavior will speedup the iteration. Initially these appear to be conflicting requirements. We seek to convince the reader that there is no conflict, once the behavior is explained.

**Ratios of monics as NM:** It can be notationally useful to define the *step*  $S_N(s)$  as the ratio of the two monic polynomials

$$S_N(s_n) = \frac{s_n^N + c_{N-1}s_n^{N-1} + \dots + c_0}{s_n^{N-1} + \frac{N-1}{N}c_{N-1}s_n^{N-2} \dots + \frac{1}{N}c_1} = \frac{1}{N} \frac{P_N(s_n)}{P'_N(s_n)}. \quad (2.7)$$

As before, the poles and zeros of the step-size are both unmodified and simplified by the use of monic ratios, and it reduces the step-size by  $1/N$ , which can dramatically improve the convergence for the important case of large  $N$ , as demonstrated graphically in Fig. 2. When  $S_N(s)$  is expanded as a *partial fraction expansion*, the RoC regions for this case are circles about the poles (the roots of monic  $P'(s)$ ), having a radius out to the nearest pole.

We would like to determine how are the RoC regions of Fig. 1 map to the  $s$  plane. By reducing the step-size  $\eta$  from 0.5 to 0.1, the black circles smoothly converged. For the larger step size, the trajectory of red squares in Fig. 2 limit cycle. This natural reduction in step-size by  $N$ , due to expressing the step-size as the ratio of monics, is especially important when mapping out the RoC regions for large values of  $N$  for Eq. 1.1. Given  $s_n$ , everything on the right is known; thus when  $s_n$  is within the RoC,  $s_{n+1}$  will converge to a unique root of  $P_N(s)$  as  $n \rightarrow \infty$ .



For sufficiently small step-size, the roots of Eq. 2.7 are the solution to a linear difference equation, the simplest example being (Galántai, 2000)

$$s_{n+1} = s_n - \frac{\eta}{N} S_N(s_n). \quad (2.8)$$

By introduced an *adaptive step-size gain*  $|\eta| < 1 \in \mathbb{C}$ , which can linearize the iteration when  $s_n$  is in the neighborhood of a pole.

Near any pole, the step-size  $|S_N(s_n)|$  can become arbitrary large, introducing non-linearity into the iteration.

## 2.4 Limits on the use of small $\eta$

We shall show that reducing the step-size is not a panacea. While it dramatically reduces the probability of limit-cycles, it cannot entirely remove them. When  $s_0$  lies “exactly” on the boundary between two RoC’s, it is not clear what will happen. One likely outcome is that the trajectory will come close to the pole where it will “leap” from from that location, forcing a restarting of the iteration. It seems intuitively obvious that the probability of this repeating is incredibly small, likely even zero (but surely never zero). However the RoC condition (Eq. A.4) will always fail if the value of  $s_{n+1}$  crosses an RoC boundary, stepping into a different RoC region, forcing a new target root. This is likely to occur when  $s_n$  is close to the RoC boundary. Fortunately this condition is easily detected by monitoring the magnitude of the step-size, which must always decrease, due to the RoC condition (Eq. A.4). The concept of “exact” depends on the computer software, not the mathematics.

These RoC regions are labeled by different colors in Fig. 1, each of which is centered on one of the five complex roots.

Rational complex analytic functions, such as  $S_N(s)$ , consisting of poles and zeros, which may be expanded into partial fractions, have RoCs that are circles in the complex plane, centered on the poles, with an RoC out to the nearest pole. The RoC’s of Fig. 1 are obviously quite different. The circle to the nearest pole rule only specifies the limit, but not the entire RoC. Within the RoC circle (Eq. A.4) holds, and Newton’s method is guaranteed to converge. The remaining RoC region is unspecified.

To solve the equation we can form the partial fraction expansion (residue expansion) about the  $N$  roots  $s_p$  of  $P'(s_p) = 0$ , which are the poles of  $S_N(s)$ , and solve for the eigen-vectors, which form the basis for the linear solution of Eq. A.3. However this then requires the roots of  $P'(s)$ , which are unknown. More about this approach in [Sec. B.1.1](#).

## 2.5 Source of the limit-cycles

Based on the examples of Figs. 2 and 4, it is clear that the NL limit-cycles are critically dependent on the step-size, and are significantly reduced for a smaller step-size ( $\eta = 0.1$ ). However by taking a slightly different initial value  $s_0$ , the limit-cycle problem can return, if the modified trajectory comes closer to the pole.

Limit-cycles can occur if the initial condition  $s_0$  is close to an RoC boundary, or even worse, on top of (i.e., *very* close to) a root of  $P'(s)$  (a pole). The natural convergence factor  $1/N$  in Eq. 2.8 can improve the convergence, but it will not remove the problem when the trajectory comes close to a pole. This is the case of these “rare” (seeming-random) circumstance, which depend on  $s_0$  and  $\eta$ .

While both trajectories start at  $s_0$ , the influenced by the two poles at  $\pm 0.669j$ , is much greater for the larger step-size (red squares). The smaller step-size (black-circles) result in a linearized trajectory, which drifts smoothly away from both real poles, eventually converging on the zero of  $S_N$  at  $-0.766 + 0.352j$ . The large step-size (red square) trajectory jumps erratically over the pole at  $-0.766 + 0.35j$ , resulting in a nonlinear limit-cycle oscillation.

# 3 Summary and Discussion

## 3.1 The role of the step-size

In the derivation of NM, we modified Eq. A.3 by introducing a step-size  $\eta < 1$ , we obtained Eq. 2.8. The effect of the reduced step-size is to force the trajectory to be more sensitive to the influence of the poles, rather than stepping over them. The addition of the step size is an important modification to Newton’s method. The smaller step-size can eliminate the nonlinear limit-cycles, as seen in the example of Fig. 4.

When the initial value for the iteration  $s_0$  is close to the cross-over of two RoCs,  $s_n \rightarrow s_{n+1}$  can cross over an RoC boundary, changing the limit point (root it converges to). A limit cycle can happen when  $s_n$  comes close to one of the poles of  $S_N(s_n)$ . At a pole, the value of  $S_N$  can become arbitrary large, causing the unmodified ( $\eta = 1$ ) update  $S_N = s_{n+1} - s_n$  to fail to satisfy the required RoC convergence condition (Eq. A.4).

One strategy for detecting the pole is to look at the magnitude of the step ( $|\eta|$ ). If  $|\hat{s}_{n+1} - s_n| > 1$ , the RoC condition has failed. The step must then be reverted back to  $s_n$ , and the step-size reduced, and  $s_{n+1}$  recomputed. This then repeated until the RoC condition ( $|s_n| > |s_{n+1}|$ ), thus avoiding a possible limit cycle.

Based on our numerical results, the addition of the convergence factor  $\eta$  seems unnecessary when the the initial value is well within the RoC, as required by Eq. A.4. The main question is when (and why) the limit-cycles are created with Newton’s method. This question is at least partial explored in the example of Fig. 2. As long as the RoC condition is maintained, each step will progress closer to a root, and in the limit, as  $n \rightarrow \infty$ ,

$$\frac{P_N(s_n)}{P'_N(s_n)} \rightarrow 0, \tag{3.1}$$

since  $s_n \rightarrow s_r$  as  $n \rightarrow \infty$ .

We don’t understand many observations in science (math and physics), which with some analysis, can eventually be explained. It is the *reductionist* method in science that explains the success of the scientific method. This might be viewed as a form of evolution: success begets more success, while failure eventually dies off, perhaps slowly.

The process of systematically exploring these seemingly tiny discrepancy, almost always leads to new knowledge. Seeking out these idiosyncratic inconsistencies and trying to explain them is at the heart of the scientific method. When a problem is longstanding and considered fundamental, its resolution can even lead to a paradigm shift. Not surprisingly such deep insights are rarely welcomed by the scientific community, rather they are viewed with great skepticism. This can be good when it doesn’t go on for 50 years.

The problem of finding roots using Newton’s method is an excellent example. It is a case that can be explained only after careful thought and iterative analysis. I feel we are either close to that understanding, or it has been explained clearly enough that the debate can be stopped, and final conclusions may be reached. However, realize that there is no “final.”

Limit cycles do exist in Newton’s method, but in my view, they are due to under-sampling the complex plane. This is an example of aliasing, in the Nyquist sense, (Allen, 2020, p. 153,262). An under-sampled process becomes nonlinear when the “high frequencies” alias into the “base-band” frequencies. This nonlinear effect is easily removed by increasing the sampling rate above the Nyquist sampling frequency, defined as twice the highest frequency in the signal. While that concept is not clear in the context of Newton’s method, it can explain limit-cycles, and slightly (2x-3x) increasing the computation, by decreasing the step-size  $\eta$ , the aliasing may be brought under control, and the problem becomes linear and well behaved. The onset of aliasing is easily detected. This leads to a well know method in signal processing called the *adaptive step-size*, which has been successfully applied in many engineering problems. It is, I believe, well understood and characterized in terms of aliasing (Allen and Sondhi, 1979; Rinzel and Miller, 1980, Sec. V, p. 126).

**The linear prediction algorithm:** An interesting alternative to stabilize NM is to use the *linear prediction* method, a causal recursion method invented in the 1940’s (Vaidyanathan, 2007). It seems likely to me that the use of Linear Prediction (LP) could greatly improve the convergence properties of NM. The down side is that the LP method assume the step-size only has poles, which in our case is clearly not true. The zeros of  $P_N(s)$  bias the estimate in a negative manner. However when the trajectory steps near a pole, the LP algorithm should fit the data extremely well, thus removing the influence of the pole. This approach could be especially effective if there were several poles in proximity.

The LP method uses a least squares minimization of the residual error, given an adaptively determined zeros, to cancel any poles. Research has shown that LP is highly effective at identifying and removing poles in a time series (Vaidyanathan, 2007). The added computational overload is likely small.

# Appendix

## A Derivation of Newton's method

Consider the polynomial  $P_N(s)$ , with  $s, s_r, c_n \in \mathbb{C}$  and  $N \in \mathbb{N}$ :

$$P_N(s) = c_N(s - s_r)^N + c_{N-1}(s - s_r)^{N-1} + \cdots + c_1(s - s_r) + c_0, \quad (\text{A.1})$$

where Taylor's formula is used to determine the coefficient vector  $\mathbf{C} = [c_N, c_{N-1}, \dots, c_0]^T$

$$c_k = \frac{1}{k!} \left. \frac{d^k}{ds^k} P_N(s) \right|_{s=s_r}. \quad (\text{A.2})$$

Here  $s = \sigma + j\omega$  is called the *Laplace frequency*, as defined by the Laplace transform (Allen, 2020). Depending on physical considerations, the coefficients  $c_n$  may be real or complex.

If our initial guess for the root  $s_1$  is close to a root  $s_r$   $|(s_1 - s_0)^k| \ll |(s_1 - s_0)|$  for  $k \geq 2 \in \mathbb{N}$ , (i.e.,  $\epsilon_n = s_1 - s_0$  is within its RoC), we may truncate Eq. A.2 to its linear term  $c_1 = \left. \frac{dP_N(s)}{ds} \right|_{s_n}$ , resulting in the approximation

$$\begin{aligned} P_N(s_1) &\approx (s_1 - s_0) \left. \frac{d}{ds} P_N(s) \right|_{s_0} + P_N(s_0) \\ &= (s_1 - s_0) P'_N(s_0) + P_N(s_0). \end{aligned}$$

Here  $P'_N(s)$  is shorthand for  $dP_N(s)/ds$ .

Replacing  $s_0$  by  $s_n$  and  $s_1$  with  $s_{n+1}$ , and letting  $s_{n+1} \rightarrow s_r$ , the LHS goes to zero, giving

$$(s_n - s_{n+1}) P'_N(s_n) + P_N(s_n) = \cancel{P_N(s_{n+1})} \rightarrow 0$$

Solving for  $s_{n+1}$  gives Newton's method:

$$s_{n+1} = s_n - \frac{P_N(s_n)}{P'_N(s_n)}. \quad (\text{A.3})$$

Starting from  $n = 0$  everything on the right is known, and  $s_{n+1} \rightarrow s_r$ , as  $n \rightarrow \infty$ . The difference  $|s_{n+1} - s_n|$  is useful in detecting the stopping rule. When the error  $\epsilon_n = s_{n+1} - s_n = 0$ , the numerical-based lower bound has been reached, and the iteration naturally terminates.

Importantly, if  $s_n$  approaches a root of  $P'(s)$ , the denominator can become arbitrarily large, resulting in a restart of the entire procedure. We shall demonstrate that this is the source of the chaotic lack of convergence.

On the other hand, if any guess for the root  $s_n$  is close to a root of  $P_N(s_r \pm \epsilon)$ , (i.e.,  $\epsilon = s_n - s_r$  is within the RoC), then

$$|(s_n - s_r)^k| \ll |(s_n - s_r)| \quad (\text{A.4})$$

for  $k \geq 2 \in \mathbb{N}$ . This *complex analytic linearization step* is the key to Newton's method. *It will only be true if the difference equation remains linear, which requires the RoC condition Eq. A.4.*<sup>7</sup> Namely Newton's method is a linear approximation that critically depends on the RoC condition (Eq. A.4).

Add Balakrishnan ref.

## B A proof of convergence

We define a monic polynomial  $P_N(s) = s^N + \sum_{n=N-1}^0 c_n s^n$  (i.e.,  $c_N = 1$ ) having roots  $r_k \in \mathbb{C}$ , such that  $P_N(r_k) = 0$  where  $s = \sigma + j\omega$  is called the Laplace frequency. We also define the *adaptive step-size*  $\eta = |\eta| e^{j\angle\eta}$  (see Eq. A.3), or in terms of polar coordinates  $a, \phi \in \mathbb{R}$ ,  $a \equiv |\eta| < 1$ , and  $\phi \equiv \angle\eta$  where  $0 \leq \phi \leq 2\pi$ . Here we present the hypothesis we wish to prove true, and outline a proof, which can be verified numerically, but only within the limits of the computers numerical accuracy. Thus we have only proved the hypothesis for IEEE-754 floating point numerics (Allen, 2020, p. 26), namely rational roots  $r_k \in \mathbb{Q}$  which is defined in Octave/Matlab as  $eps = 2.2204 \times 10^{-16}$ , which is the smallest floating point number on Intel computers (IEEE-754).

However we believe it to be true within these numerical limits (rational roots) ( $r_k \in \mathbb{F} \equiv n/m$ ), where  $n, m \in \mathbb{N}$ ,  $P_n(s) = \prod_{k=1}^N (s - r_k)$  and  $P_N(r_k) = 0$ . We seek a generalization for irrational roots  $r_k \in \mathbb{I}$ .

<sup>7</sup>Given some basic algebra, Eq. A.4 is equivalent to  $|\frac{s_n}{s_r}| < 1$ , which follows from the triangle inequality for Hilbert vectors  $s_r - s_n$  and  $s_r - s_{n+1}$  within the RoC (Allen, 2020, p. 130). This needs to be verified, not simply stated. Specifically, is  $\|s/s_n\| < \|s_n/s_{n-1}\|$ ? For example, is  $s_{n+1} - 2s_n + s_{n-1}$  "small"? See *Mathematical physics*, Balakrishnan (2020), page 508, Eq. 22.32, on the Ratio test for the RoC of any analytic function. This should to be tested numerically, for verification.

### Theorem 1

Given a monic polynomial  $P_N(s)$  we apply Newton's method (Eq. 2.8) starting with  $n = 0$

$$\begin{aligned} s_{n+1} &= s_n - \frac{\eta}{N} S_N(s_n) \\ n &= n + 1, \end{aligned} \quad (\text{B.1})$$

and an arbitrary starting point  $s_0 \in \mathbb{C}$ , including the point at  $s_0 \rightarrow \infty$ . We shall prove that every  $s_0$  lines within a unique *Region of convergence* (RoC). There are  $N$  of these regions, one for each root  $r_k$ .

This follows from the basic property of every complex analytic function, that its inverse must also be complex analytic. For the case in point, the inverse of  $P_5(s)$  must have  $N$  branch cuts, with  $N$  RoC's. It cannot not have 6 RoCs, since every point in  $s$  must fall in one of these 6 RoCs, shown by the color codes. The only exceptions are the results of numerical errors. The boundaries of the RoC are on the branch-cuts of  $s(n, \phi) = P_5^{-1}(\phi)$ . Here we set  $|\eta| = 1$ . The step-size is  $\eta/N$  because both  $P_N$  and  $P'_N$  are monics.

After the first step we have  $s_1 = s_0 - \frac{1}{N} S_N(s_0)$ . If  $|s_1/s_0| < 1$  we have satisfy the RoC assumption built into Newton's method, thus we have moved closer to the root  $r_k$  associated with  $s_0$ . Thus  $s_1$  is in the same RoC as  $s_0$ . In this case we must repeat this calculation for  $n > 1$ , and at each step we must verify that the RoC condition ( $|s_{n+1}/s_n| < 1$ ) holds.

If the RoC condition fails for  $n$ , we must repeating Eq.B.1, restarting from  $s_n$ , the last point inside the RoC. At this point we have three options:

1. reduce the step-size  $a_n$ , and recompute  $s_{n+1}$  using Eq. B.1
2. change  $\phi_n \equiv \angle \eta_n$ , leaving  $a_n$  the same.
3. vary both  $a_n < 1$  and  $\phi_n$ .

This third more aggressive option is to vary both  $a$  and  $\phi$  to minimize  $|\frac{\eta}{N} L(s_n)| < 1$ . As the angle  $\phi_n = \angle \eta_n$  is varied, the RoC condition should vary from greater than 1 at  $\phi_{n+1}$  (because with  $\phi_{n+1}$ , the RoC condition failed) to less than 1. The locus of such points such that  $|\eta| = 1$  identifies the RoC boundary.

This appears to be a variant on the conjugate-gradient method for Newton's Method. It seems likely (but is presently untested), that the best approach is to vary  $\phi$  to minimize  $|s_{n+1}/s_n| < 1$ , and then step in that direction. Such an approach will, we believe, give a faster convergence, because the RoC condition will be smaller. Varying the angle eliminates the need to reduce  $|\eta_n| < 1$ , yet improving the rate of convergence.

Our claim is, that by changing the angle of the step-size, we will find a range of angles that optimize the RoC condition (find the smallest ratio with respect to  $\phi$ ).

Moving in the direction that  $|s_{n+1}/s_n| \leq 1$  delineates the RoC boundary. The prediction is, that by moving in the direction where RoC is 1, will trace out the RoC boundary region. Crossing the boundary and starting with  $s_{n+1}$ , Eq. B.1 should converge to a different root.

Numerically this is easily demonstrated, providing an empirical proof of our hypothesis (theorem).

The RoC regions may be seen in Fig. 4 as the different colored regions. The long dashed lines indicate when the trajectory jumped from on RoC to another due to coming under the influence of a pole of  $S_N(s)$  (zero of  $P'(s)$ ).

## B.1 The inverse Laplace transform of $L_N(s)$

To find the exact solution to Newton's method we must take the inverse Laplace transform of Eq. 2.7

$$\begin{aligned} \vec{\varrho}_N(t) &= \frac{1}{2\pi j} \oint \frac{P_N(s_n)}{P'_N(s)} e^{st} ds \\ &\leftrightarrow \frac{1}{2\pi j} \sum_{p=1}^{N-1} R_p e^{\lambda_p t} u(t). \end{aligned}$$

Here  $u(t)$  is the Heaviside step function,  $R_p$  are the residues and  $\lambda_p$  are the roots of  $P'_N(\lambda_p) = 0$ .

Each term in  $\varrho_N(t)$  requires an initial condition, such as  $\varrho(0) = 0$ . To evaluate the inverse Laplace transform, one must use Cauchy's integral theorem, following the residue expansion of  $S_N(s)$  under the integral sign.

As discussed in Sec. 1.1

$$ds \equiv (s_{n+1} - s_n) = - \lim_{\eta \rightarrow 0} \eta \left( \frac{1}{N} \frac{P_N(s_n)}{P'_N(s_n)} \right) \quad (\text{B.2})$$

thus

$$\frac{d}{ds} S_N(s) = \sum_{n=1}^{N-1} \frac{R_n}{s - s_p^n} \leftrightarrow \sum_{n=1}^{N-1} R_n e^{-s_p^n t}. \quad (\text{B.3})$$

Here  $s_p^n$  is the  $n$  power of the  $p$ th root of  $P'(s)$ , and  $\leftrightarrow$  indicates the inverse Laplace transform.

The inverse Laplace transform of this residue expansion is the solution to this linear difference equation. An alternative is to seek the inverse  $z$  transform, which gives the exact solution to the linear difference equation. In both cases the eigen-values are the  $N - 1$  roots of  $P'(s_p) = 0$ .

To complete the proof of the failure to converge, one may take the inverse Z transform of Eq. B.3. Under certain conditions, as  $\eta \rightarrow 1$  the imaginary poles can become real, at which point they are no longer conjugate pairs. When this happens, the pole pair split and one pole goes to 0 and the other to  $\infty$ . This is the condition which results in an unstable limit cycle (Allen and Sondhi, 1979, Sec. V, p. 126).

### B.1.1 Use of the companion matrix method:

This seems like very little progress, since we don't know the roots of  $P'_N(\lambda_p) = 0$ , thus we cannot form the partial fraction expansion.

However we do know the coefficients of both  $P_N(s)$  and  $P'_N(s)$ . Given the coefficients of a polynomial  $P'_n(s)$ , we may form its companion matrix, which by definition, has the same eigen-values as the roots of monic polynomial  $P'_N(s)$   $C'_N$  (Allen, 2020).

Proceeding with this approach, based on Eq. 1.2, we form the vector relation for  $s_k(\eta)$

$$\frac{d}{d\eta} \vec{q}_N(t) = e^{\eta C'_N} \mathbf{R}_p. \quad (\text{B.4})$$

This method is demonstrated via the following example:

$$P_4(s) = \frac{1}{4}s^4 - \frac{1}{3}s^3 - \frac{1}{2}s^2 - s - c_0 \leftrightarrow [1/4, -1/3, -1/2, -1, -c_0].$$

Thus  $P'_4 = s^3 - s^2 - s - 1 \leftrightarrow [1, -1, -1, -1]$ . The companion matrix for  $P'_4(s)$  is

$$\mathbf{E} = C'_4 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}_{3 \times 3},$$

which has the same eigen-values  $\lambda_p$  of the monic polynomial  $P'_4(s)$  [ $\lambda_p = 1.83929, -0.41964 \pm j0.60629$ ]. This has one unstable real pole and two stable poles with radian frequencies  $\pm 0.60629$  [rad/s].

The residues for this example are  $R_p = [-0.68437, 0.17552 \pm j0.19742j]$  and the poles are  $\lambda_k = [1.83929, -0.41964 \pm j0.60629j]$ .

It follows that if the initial conditions for each root is 1 then

$$\vec{q}_3(\eta) = \frac{1}{3} \sum_1^3 e^{\eta C'_4} = \frac{1}{3} \sum_1^3 e^{\mathbf{E} \begin{bmatrix} \eta \lambda_1 & 0 & 0 \\ 0 & \eta \lambda_2 & 0 \\ 0 & 0 & \eta \lambda_3 \end{bmatrix} \mathbf{E}^{-1}}$$

Recall that the matrix eigen equation is  $\vec{q}_3(t) \mathbf{E} = t \mathbf{E} \Gamma$  (Allen, 2020, p. 305), thus

$$\vec{q}_3(\eta) = e^{\mathbf{E} \eta \Gamma \mathbf{E}^{-1}} = e^{\mathbf{E}} \cdot e^{\eta \Gamma} \cdot e^{\mathbf{E}^{-1}}.$$

Importantly, it is still necessary to use a numeric method to find the  $N - 1$  eigenvalues of the matrix  $C'$ . The above is an interesting description of the solution for  $\vec{q}_3(\eta)$ , but of questionable value since it requires the computation  $\lambda_n$  which is almost the problem we started with for  $C_4$ . Thus we still need to find the roots of the characteristic polynomial, which are the 3 roots of  $P'_3(\lambda_n) = 0$ . Thus it is minor simplification of the original approach, but it seems to not be useful.

**Summary:** By way of summary, the eigen-matrix  $\mathbf{E}$  and its inverse  $\mathbf{E}^{-1}$  play a major role in the eigen analysis mechanics. Since  $\vec{q}_3 \mathbf{E} = \mathbf{E} \Gamma$ , it follows that  $\vec{q}_3 = \mathbf{E} \Gamma \mathbf{E}^{-1}$ . Thus  $\Gamma \mathbf{E}^{-1}$  rotates the eigenvalues off the diagonal such that when operated on by  $\mathbf{E}$  they equal  $\vec{q}_3$ .

For reference

$$\Gamma C'_3 = \begin{bmatrix} \lambda_1, 0, 0 \\ 0, \lambda_2, 0 \\ 0, 0, \lambda_3 \end{bmatrix} \begin{bmatrix} -1, 1, 0 \\ -1, 0, 1 \\ 1, 0, 0 \end{bmatrix} = \begin{bmatrix} -\lambda_1, \lambda_1, 0 \\ -\lambda_2, 0, -\lambda_2 \\ \lambda_3, 0, 0 \end{bmatrix}$$

**On last point:** Mention the importance of the partial fraction expansion, which raises  $P'_N(s)$  as the characteristic polynomial of the system being analyzed. From this view,  $P'(s)$  has a more important role than  $P(s)$ . Perhaps this is critically important, but was not properly noted.

An famous example is the case of  $f(s) = e^{3s} + 2s \cos(s) - 1$ , mentioned in *The ZerSol Library* by Ivan B. Ivanov 2012.

A great example of a divergence of NM when  $f(x) = \tan^{-1}(x), x \in \mathbb{R}$ .<sup>8</sup>

## References

- Allen, J. B. (2020). *An Invitation to Mathematical Physics and Its History*. Springer, New York, New Delhi.
- Allen, J. B. and Sondhi, M. M. (1979). Cochlear macromechanics: Time-domain solutions. *Journal of the Acoustical Society of America*, 66(1):120–132.
- Boas, R. (1987). *Invitation to Complex Analysis*. The Random House, New York.
- Galántai, A. (2000). The theory of Newton's method. *Journal of Computational and Applied Mathematics*, 124(1-2):25–44.
- Haar, D. (2016). *The Old Quantum Theory*. Elsevier Science.
- Kuhn, T. (1978). *Black-Body Theory and the Quantum Discontinuity, 1894-1912; (371 pages)*. Oxford University Press (1978) & University of Chicago Press (1987), Oxford, Oxfordshire; Chicago, IL.
- Rinzel, J. and Miller, R. N. (1980). Numerical calculation of stable and unstable periodic solutions to the Hodgkin-Huxley equations. *Mathematical Biosciences*, 49(1):27–59.
- Stewart, J. (2012). *Essential Calculus: Early Transcendentals*. Cengage Learning, Boston, MA.  
<https://books.google.com/books?id=AcQJAAAAQBAJ>.
- Vaidyanathan, P. P. (2007). The theory of linear prediction. *Synthesis lectures on signal processing*, 2(1):1–184.

<sup>8</sup><https://web.archive.org/web/20120615221147/http://math.fullerton.edu/mathews/a2001/Animations/RootFinding/NewtonMethod/Newtonff.html>