W. Kahan, Prof. *Emeritus* (Retired)

Math. Dept., and E.E. & Computer Science Dept.
University of California at Berkeley

Abstract
These course notes concern the solution of one real equation f(z) = 0 for one real root z , also called a real *zero* z of function f(x) . They *supplement*, not *supplant*, textbooks and deal mathematically with troublesome practical details not discussed in my reprint [1979"] about a calculator's [SOLVE] key, which should be read first; it offers easy–to–read advice about real root-finding in general to anyone who wishes merely to use a root–finder to solve an equation in hand. These course notes are harder to read; intended for the would–be designer of a root–finder, they exercise what undergraduates may learn about Real Analysis from texts like Bartle [1976]. Collected here are proofs, mostly short, for mathematical phenomena, some little known, worth knowing during the design of robust and rapid root-finders.

Almost all Numerical Analysis texts cover the solution of one real equation f(z) = 0 for one real root z by a variety of iterative algorithms, like $x \to U(x)$ for some function U that has z = U(z) as a fixed-point. The best known iteration is Newton's: $x \to x - f(x)/f'(x)$. Another is Secant iteration: pair $\{x, y\} \to \{w, x\}$ where $w := x - f(x)\cdot(x-y)/(f(x) - f(y))$. But no text I know mentions some of the most interesting questions:

- Is some simple Combinatorial (Homeomorphically invariant) condition both Necessary and Sufficient for convergence of $x \to U(x)$? (Yes; §5)
- Is that condition relevant to the design of root-finding software? (Yes; §6)
- Do other iterations $x \to U(x)$ besides Newton's exist? (Not really; §3)
- Must there be a neighborhood of z within which Newton's iteration converges if f'(x) and x - f(x)/f'(x) are both continuous? (Maybe Not; §7)
- Do useful conditions less restrictive than Convexity suffice Globally for the convergence of Newton's and Secant iteration? (Yes; §8)
- Why are these less restrictive conditions not Projective Invariants, as are Convexity and the convergence of Newton's and Secant iterations? (I don't know; §A3)
- Is slow convergence to a multiple root worth accelerating? (Probably not; §7)
- Can slow convergence from afar be accelerated with no risk of overshooting and thus losing the desired root? (In certain common cases, Yes; §10)
- When should iteration be stopped? (*Not* for the reasons usually cited; §6)
- Which of Newton's and Secant iterations converges faster? (Depends; §7)
- Which of Newton's and Secant iterations converges from a wider range of initial guesses at z ? (Secant, unless z has even multiplicity; §9)

Therefore, Why Use Tangents When Secants Will Do?

- Have *all* the foregoing answers been *proved* ? Yes. Most were proved in the 1960s and 1970s [1979'], and influenced the design of the [SOLVE] key on Hewlett-Packard Calculators.

Contents

Acrobat™ Reader PDF files:
http://www.cs.berkeley.edu/~wkahan/Math128/RealRoots.pdf
…/SOLVEkey.pdf
…/Cubic.pdf

§1. Overview

Before a real root z of an equation "f(z) = 0" can be found, six questions demand attention:

«1» Which equation?
Infinitely many equations, some far easier to solve than others, have the same root z .

«2» What method?
Usually an iterative method must be chosen; there are infinitely many of them too.

«3» Where should the search for a root begin?
A global theory of the iteration's convergence helps compensate for a vague guess at z .

«4» How fast can the iteration be expected to converge?
A local theory helps here. Convergence much slower than expected is ominous.

«5» When should iteration be stopped?
Error-analysis helps here. And the possibility that no z exists may have to be faced.

«6» How will the root's accuracy be assessed?
Error-analysis is indispensable here, and it can be done in more than one way.

The questions are not entirely independent, nor can they always be answered in order. If question «2» is answered by some available software that contains its own root–finder, the method it uses should influence the answer to question «1». Question «5» may depend upon question «6», which may be easier to answer after z has been found. Anyway, these questions do not have tidy answers. Instead, the following notes answer questions that resemble the foregoing six, and the reader must decide whether available answers pertain well enough to his own questions.

Different contexts may call for different answers. Two contexts are worth distinguishing during the design of root–finding software: General–purpose root–finders have to be designed without knowing the equations they will be asked to solve; special–purpose root–finders are designed to solve one equation "F(z, p) = 0" for a root z = z(p) regarded as a function of the parameter(s) p over some preassigned range. General–purpose root–finders must be robust above all; they cope with very diverse equations and with poor first guesses at roots that need not be unique or, in other cases, need not exist; speed matters only because a root–finder that runs too slowly will be abandoned by impatient users before it finds a root. Speed is the reason for a special–purpose root–finder's existence, and to that end it exploits every advantage that mathematical analysis can wrest from the given expression F(x, p) . Applicability to many such special cases justifies the inclusion of much of the theory presented in these notes.

Root–finders are almost always iterative; they generate a sequence of approximations intended to converge to a desired root. For reasons outlined in §2, §3 gives the infinite variety of iterative methods short shrift. Whereas textbooks concentrate mostly upon questions of local convergence answerable often by appeals to Taylor series, these notes concentrate mostly upon questions of global convergence. Does "global" convergence theory differ from "local"? It's a distinction with a small difference: Local theories touched in §3 and §4 describe what happens, and how fast, in every sufficiently small neighborhood of a root; this kind of theory applies to practically

all cases. A global convergence theory provides ways to tell whether a root exists, whether an iteration will converge to it from afar, and whether slow convergence from afar can be sped up without jeopardizing convergence to the desired root; these questions have usable answers only in special cases. The special cases discussed in these notes arise often enough to make their study generally worthwhile.

Most iterations discussed in these notes have the form $x_{n+1} := U(x_n)$, which may seem very general but isn't really; there is a sense (see Thesis 3.1 below) in which every such scalar (not vector) iteration is really Newton's iteration in disguise. Textbooks and our §4 treat iterations whose U is a *Contraction*: ($|U'| < 1$) throughout some domain supposed to contain the desired root and all but finitely many initial iterates. Finding that domain can be as hard as finding the root, and futile too because Contraction in a *wide* domain surrounding the root is a condition merely sufficient, not necessary, for convergence. There is a conceptually simpler combinatorial condition necessary and sufficient for convergence from every starting point in a wide domain; see Sharkovsky's No–Swap Theorem 5.1 below. This theorem provides an invaluable "One–Sided" criterion by which to decide when a program must intervene to force an iteration to converge. That decision may be necessitated by the intrusion of rounding errors whose worst effects can be avoided only by using appropriate criteria to stop the iteration. Such criteria and other software issues are discussed at length in §6.

Newton's iteration $x_{n+1} := x_n - f(x_n)/f'(x_n)$ and Secant iteration $x_{n+1} := x_n - f(x_n)/f^\dagger(x_n, x_{n-1})$ are treated next; here $f^\dagger$ is a *First Divided Difference* whose analogy with the first derivative f' is explained below in Appendix A1 on Divided Differences. Both iterations have such similar local convergence properties that they are treated together in Theorems 7.4, 7.5 and 7.6. The weakest known global conditions sufficient for convergence are named in Theorem 8.2 and Corollary 8.3; roughly speaking, they require that |f'| not vary too much. (A connection with the classical theory of Functions of Bounded Variation is covered in Appendix A2.) Both iterations have similar global convergence properties because those properties are invariants of certain plane Projective Maps that are the subject of yet another Appendix A3. Unfortunately, the aforementioned weakest known global conditions sufficient for convergence are not invariant under projective maps; to find usable weaker invariant conditions remains an open problem.

The projective invariance of Newton's and Secant iteration is the source of an astonishing Theorem 9.2 which says, roughly, that if f reverses sign wherever it vanishes in some interval, and if Newton's iteration converges within that interval from every starting point therein, then Secant iteration converges too from every two starting points in that interval. Of course, they converge then to the unique zero of f in the interval. This theorem has no converse; Secant iteration can converge but not Newton's. The discovery of this theorem over thirty years ago had a profound effect upon the design of root–finders built into Hewlett–Packard calculators.

Slow convergence of Newton's and Secant iteration to a multiple root is a problem that has received more attention in the literature than it deserves in the light of Theorem 7.6, which is too little known. This theorem provides good reasons to expect computed values of f(x) to drop below the noise due to roundoff, or else below the underflow threshold, rapidly no matter how slowly iterates x converge, so iteration cannot be arbitrarily prolonged. Convergence slowly

from afar to a simple root that appears, from afar, to belong to a tight cluster of roots is a problem deserving more attention. The problem is not how to accelerate the iteration, but how not to accelerate it too far beyond the desired root. In cases covered by Theorem 10.1 the problem has a simple solution that roughly halves the number of Newton iterations when they converge slowly. A similar solution works for Secant iteration but the details of its proof are incomplete.

I have tried to prove every unobvious unattributed assertion in these notes. The proofs are as brief as I could make them, and not merely by leaving steps out. Still, the proofs should be skipped on first reading; to make doing so easier, each proof is terminated by END OF PROOF. To ease the location of this document's sections, theorems, lemmas, corollaries, examples, ..., they will be numbered consecutively when the notes are complete.

Yet to be transcribed are sections about finding all real zeros of a polynomial, all zeros of a real cubic, error bounds for computed zeros, and running error bounds for computed values of a polynomial. Meanwhile the author will welcome corrections and suggestions, especially for shorter and more perspicuous proofs.

§2. Three Phases of a Search

Root-finding software invoked to solve " f(z) = 0 " seeks a root z by employing a procedure generally more complicated than the mere iteration of some formula x := ... until it converges. Watching such software at work, when it works, we can usually discern three phases:

Phase 1 : **Flailing**

Initial iterates x approximate the desired root z poorly. They may move towards it, or wander, or jump about as if at random, but they do not converge rapidly.

Phase 2 : **Converging**

Differences between successive iterates x dwindle,— rapidly, we hope.

Phase 3 : **Dithering**

Indistinguishable from Flailing except that different iterates x differ much less from a root and may (very nearly) repeat. Dithering is due entirely to roundoff.

Dithering is a symptom of an attempt to solve " f(z) = 0 " more accurately than roundoff allows. Ultimately accuracy is limited by what roundoff contributes unavoidably to the computed values of f(x) . Accuracy much worse than that should be blamed upon an inept implementation of the iteration formula x := ... or upon some other defect in the software, or else upon intentional premature termination of the iteration because its accuracy was judged adequate. Judgments like this posit the existence of a trustworthy error estimate, which is a nontrivial requirement. It looks easy at first; the possession of a *Straddle* *,— two iterates $x_«$ and $x_»$ where $f(x_«)\,f(x_») < 0$,— suffices (if f is continuous) to locate a root z between them with an error less than $|x_« - x_»|$. However the purchase of a sufficiently close straddle may cost almost twice as much computation as a simple iteration x := ... that converges from one side, unless error analysis can be brought to bear. Error analysis will be discussed at length later; without it, dithering could waste a lot of time.

Converging is what we hope the chosen iteration does quickly, and usually it does; and when it does, the search for a zero can spend relatively little time in Phase 2. Why then is so much of the literature about numerical methods concerned with this phase? Perhaps because it is the easiest phase to analyze. Ultimately superlinear (fast) convergence is rarely difficult to accomplish, as we shall see; Newton's iteration usually converges quadratically. Convergence faster than that is an interesting topic omitted from these notes because it reduces only the time spent in Phase 2; higher order convergence is worth its higher cost only if extremely high accuracy is sought.

We shall devote more consideration than usual to Phase 1 because it is the least understood and potentially most costly. A long time spent flailing is a symptom of a mismatch between the given equation " f(z) = 0 " and the root–finder chosen to solve it.

---

*Footnote: A *Straddle* is to the Navy what a *Bracket* is to the Army;— a pair of shots fired one beyond and the other short of a target to intimidate it or to gauge its range. But "Straddle" and "Bracket" have distinct meanings in these course notes.

§3. Models and Methods

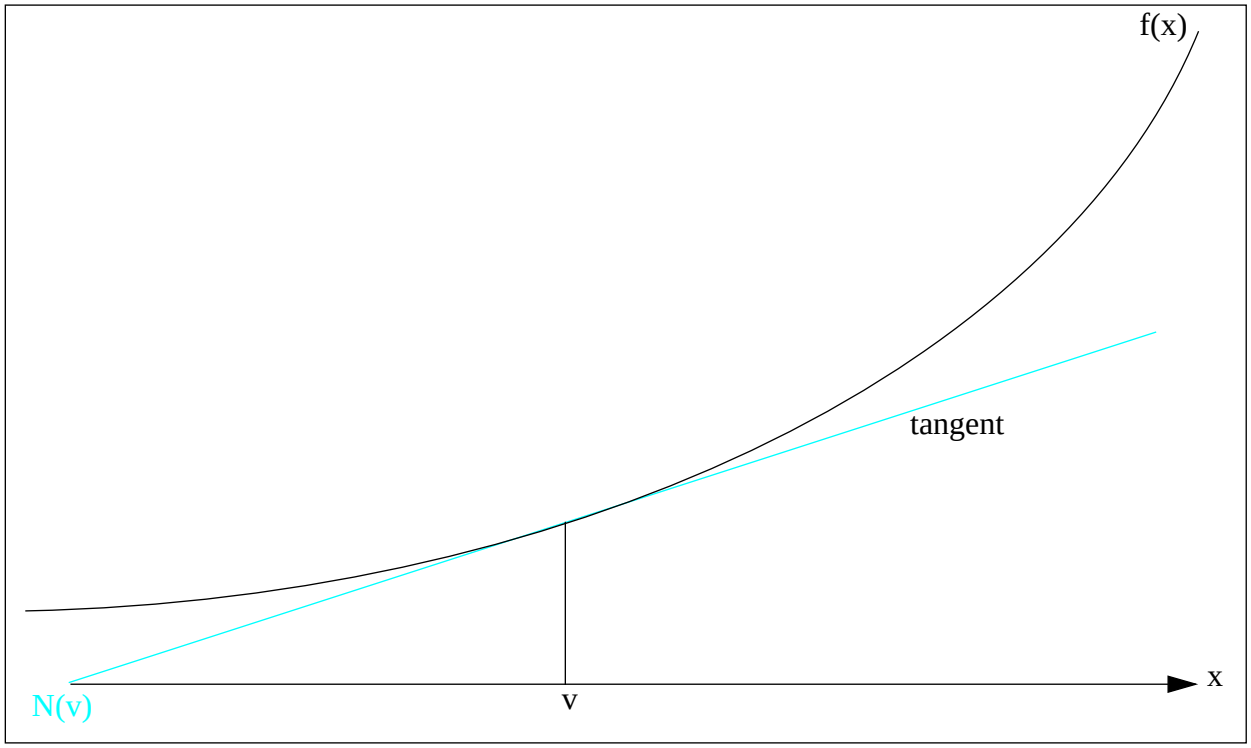
Every iterative method for solving "$f(z) = 0$" is motivated by a *model*; this is a family of easily solved equations from which is drawn a sequence of ever better approximations to the given equation over a sequence of ever narrowing intervals around the desired root z. For example, the given equation may be rewritten as an equivalent equation $g(z) = h(z)$ with the same root z but with $h(x)$ slowly varying (approximately constant) when x is near z, and with $g(x)$ easily inverted. The last equation is turned into an iteration by solving $g(x_{n+1}) = h(x_n)$ for each new approximation x_{n+1} to replace the previous approximation x_n to z. When $h'(x)/g'(x)$ is continuous and $|h'(z)/g'(z)| < 1$, the iteration can easily be proved to converge to z from any initial x_0 close enough to z. (Look at $(x_{n+1}-z)/(x_n-z) = h^\dagger(x_n,z)/g^\dagger(x_{n+1},z)$ as $x_n \to z$; here $h^\dagger$ is a divided difference analogous to the derivative h' and explained in Appendix A1.)

For instance take the equation $3e^z = e^3z$. It can be "solved" for $z = 3 + \ln(z/3)$ to construct an iteration $x_{n+1} := 3 + \ln(x_n/3)$, or for $z = 3e^{z-3}$ to construct an iteration $x_{n+1} := 3\exp(x_n - 3)$. Each iteration is attracted to a different root z. (Find them! Why are there no more roots?)

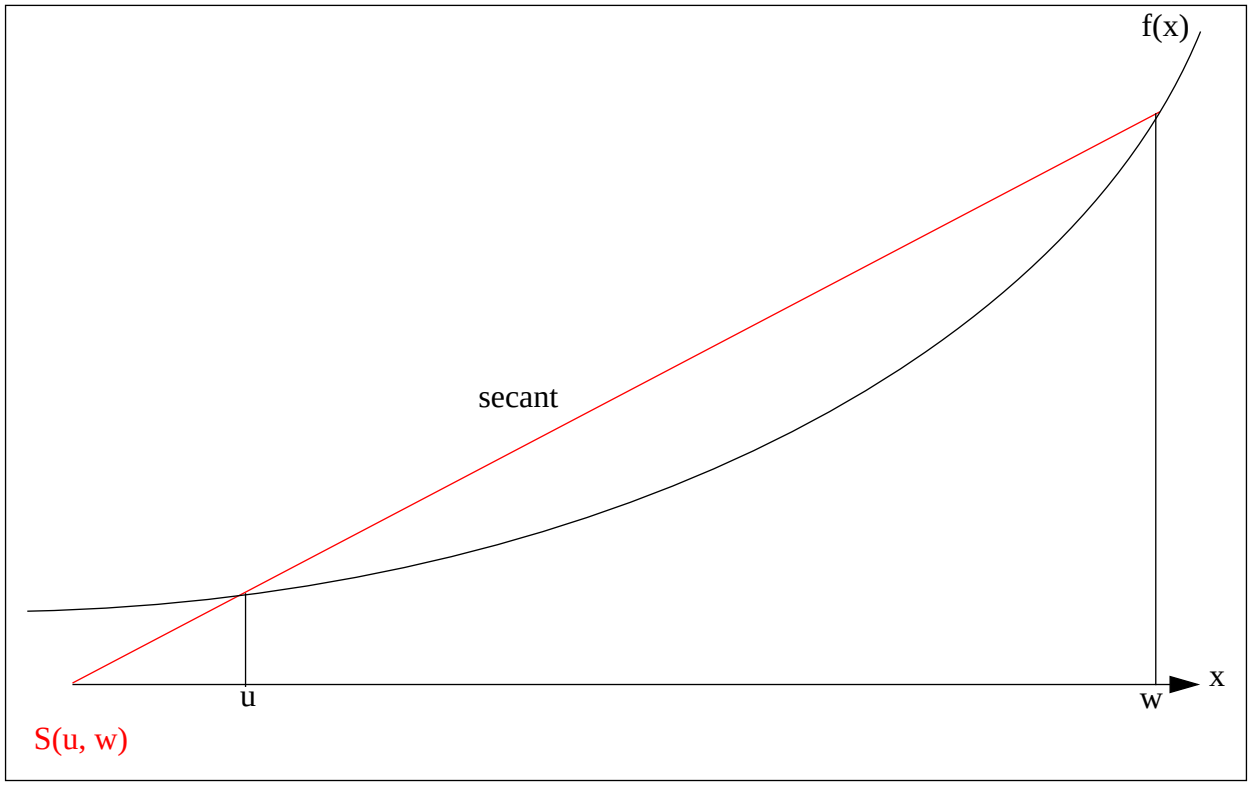
More generally, a given equation "$f(z) = 0$" may be rewritten "$g_n(z) = h_n(z)$" in a way that can change with every iteration that solves $g_n(x_{n+1}) = h_n(x_n)$ for x_{n+1}, and can depend also upon previous iterates $x_{n-1}, x_{n-2}, \ldots$. These dependencies are motivated by a model all the same, but now reinterpreted as a family of convenient curves from which is drawn a sequence of ever better approximations to the graph of the given function f over a sequence of ever narrowing intervals around the desired root z. The wider the interval over which f resembles a member of that family, and the closer the resemblance, the faster the iteration derived from the model converges.

A substantial body of theory connects the qualities of a model to the ultimate speed of the derived iteration's convergence; see Traub [1964] or Ostrowski [1973]. Like most of today's texts on Numerical Analysis, these notes draw little more from that theory than two items of terminology: *Rate* and *Order* are measures of the ultimate speed with which a sequence $x_1, x_2, x_3, \ldots, x_n, \ldots$ may converge to its limit z as $n \to \infty$. Its Rate := lim inf $-\ln(|x_n - z|)/n$, and its Order := lim inf $(-\ln(|x_n - z|))^{1/n}$. *Linear* convergence has Order = 1 and a positive finite Rate, which means the number of digits to which x_n and z agree grows ultimately linearly with n; slower than linear convergence is almost intolerable. For most practical purposes we expect *Superlinear* convergence with Rate = $+\infty$ and Order > 1, which means that ultimately each iteration multiplies the number of agreeing digits by Order on average.

Here are examples: Newton's iteration $x_{n+1} := x_n - f(x_n)/f'(x_n)$ approximates the graph of f by its tangent T_n at a point $(x_n, f(x_n))$ that the iteration tries to move closer to $(z, 0)$ by moving the point of tangency to the point $(x_{n+1}, f(x_{n+1}))$ on the graph above T_n's intersection with the x–axis. Convergence is typically *Quadratic* (Order = 2). Similarly, the *Secant* iteration $x_{n+1} := x_n - f(x_n)/f^\dagger(x_n, x_{n-1}) = x_n - f(x_n)(x_n - x_{n-1})/(f(x_n) - f(x_{n-1}))$ approximates the graph of f by its secant through two points $(x_n, f(x_n))$ and $(x_{n-1}, f(x_{n-1}))$, and replaces the latter by the point $(x_{n+1}, f(x_{n+1}))$ above where the secant cuts the x–axis. The iteration's Order ≈ 1.618 typically.



Newton's: $x_{n+1} := N(x_n)$ where $N(v) := v - f(v)/f'(v)$.



Secant: $x_{n+1} := S(x_n, x_{n-1})$ where $S(u, w) := u - f(u)(u - w)/(f(u) - f(w))$.

David Muller's method fits a parabola through three points on the graph of f, and replaces one of them by a point on the graph above the nearer intersection of the parabola with the horizontal axis. An hyperbola with vertical and horizontal asymptotes can also be fitted through three points on the graph of f, and provides an iteration simpler than Muller's and better suited to finding a simple zero z close to a pole of f. (A *pole* of f is an argument $\hat{o}$ at which $f(\hat{o}) = \infty$.) The hyperbola is the graph of $\mu(x - x_{n+1})/(x - \tilde{o})$ for constants μ, $\tilde{o}$, x_{n+1} chosen by making that expression interpolate (match) $f(x)$ at three consecutive iterates x_n, x_{n-1}, x_{n-2}. Both these iterations converge typically at Order ≈ 1.839.

Given two iterates $x_«$ and $x_»$ that straddle a sign-change of f because $f(x_«) f(x_») < 0$, we may well wish to continue the iteration in such a way that straddling persists even if preserving it slows convergence. The simplest way is *Binary Chop*; this method models f by a step-function that disregards everything about f but its sign, and in each iteration replaces either $x_«$ or $x_»$ by $x_V := (x_«+x_»)/2$ according to $\text{sign}(f(x_V))$ so that straddling persists. *Regula Falsi* differs from Binary Chop only by determining x_V as the place where a secant through $(x_«, f(x_«))$ and $(x_», f(x_»))$ cuts the horizontal axis. Both methods usually converge linearly, too slowly. Regula Falsi can converge arbitrarily slower than Binary Chop when the graph of f is more nearly L-shaped than straight, so D. Wheeler's method (see program F2 in Wilkes *et al.* [1951]) speeds up Regula Falsi by halving whichever of $f(x_«)$ or $f(x_»)$ has not been supplanted after two iterations. C.J.F. Ridder's method, promoted by W.H. Press *et al.* [1994], chooses μ, $ß$ and x_Δ to make the expression $L(x) := \mu(x - x_\Delta)e^{ßx}$ interpolate $f(x)$ at $x_«$, $x_V := (x_«+x_»)/2$ and $x_»$, and then retains whichever pair of $x_«$, x_V, x_Δ, $x_»$ most closely straddles the sign-change of f. (One of the pair is always x_Δ.) This method is plausible when the graph of f may be very nearly L-shaped but not necessarily monotonic. Ridder's and Wheeler's methods usually converge superlinearly; for the latter see Dahlquist *et al.* [1974].

Vastly many more models and iterative methods have been published. Do we need all of them? Perhaps not; most of them converge superlinearly, so they spend similar small numbers of iterations in Phase 2. Reducing these small numbers by increasing the Order of convergence is relatively straightforward if enough derivatives of f are available. For instance, convergence (typically) at Order = 3 is obtained by fitting osculatory hyperbolas instead of tangents to the graph of f to derive Halley's iteration $x_{n+1} := x_n - 2f(x_n)/\big(2f'(x_n) - f''(x_n)\, f(x_n)/f'(x_n)\big)$.

Widening the range of initial guesses from which convergence will follow is harder but worth a try when dawdling in Phase 1 indicates a mismatch between the model and the equation to be solved. Acquaintance with many models improves our prospects of finding one that matches the given equation well. Alternatively, when possession of a software package implies the use of its root–finder, awareness of the model(s) that motivated its root–finder may suggest how to recast equations so as to match its model(s) better. Because all models include the straight line graph of a linear equation as a special or limiting case, equations $f(z) = 0$ incur fewer iterations in Phases 1 and 2 according as f is more nearly linear over a wider range around z. This observation motivates attempts to recast a given equation into an equivalent but more nearly linear form. A successful attempt will be described below after Theorem 8.2.

The two motivations, one to fit a model as closely to the equation as is practical, the other to linearize the equation as nearly and widely as possible, may become indistinguishable in the final analysis of a real (or complex) root–finder's performance. Here is a reason for thinking so:

> **Thesis 3.1: Newton's Iteration is Ubiquitous**
> Suppose that U is differentiable throughout some neighborhood Ω of a root z of the given equation " $f(z) = 0$ ". If the iteration $x_{n+1} := U(x_n)$ converges in Ω to z from every starting point x_0 in Ω, then this iteration is Newton's iteration applied to some equation " $g(z) = 0$ " equivalent on Ω to the given equation; in other words, $U(x) = x - g(x)/g'(x)$, and $g(x) \to 0$ in Ω only as $x \to z$.

Defense: $g(x) = \pm\exp(\int dx/(x - U(x)))$ with a "constant" of integration that may jump when x passes from one side of z to the other, reflecting the fact that U is unchanged when $g(x)$ is replaced by, say, $-3g(x)$ for all x on one side of z. (There is no need for $g'(z)$ to exist since it need not be computed when $g(z) = 0$; however the jump in the "constant" of integration can often be so chosen that $g'(x)$ is continuous as x passes through z.) The iteration's convergence in Ω to z alone implies first that $x - U(x)$ vanishes only at $x = z$ in Ω, and then that $x - U(x)$ has the same sign as $x - z$. (The opposite sign would compel the iteration to flee from z.) Therefore the integral decreases monotonically as x approaches z from either side. To complete the defense we shall infer from the differentiability of U that the integral descends to $-\infty$, implying that $g(x) \to 0$ as $x \to z$ as claimed.

For the sake of simpler arithmetic, shift the origin to make $z = 0$ and write Ω' for what remains of Ω when 0 is removed from it. This makes $U(x)/x < 1$ at all x in Ω'. Since $U'(0)$ exists, there also must exist some constant $1 - 1/C < U(x)/x < 1$ for all x in Ω', whence it follows that the integral $\int dx/(x - U(x)) <$ (another constant) $+ C\int dx/x \to -\infty$ as $x \to 0$ in Ω' from one side or the other. END OF DEFENSE.

(What if U were merely continuous instead of differentiable? Then g could be discontinuous at z like $g(x) :=$ (if $x \geq 0$ then $(1 + \sqrt{x})^2$ else x^2) . In general then, must $g(z+)\cdot g(z-) = 0$?)

Don't read too much significance into Thesis 3.1 . It does suggests that an iteration, derived from a family of curves that osculate (match tangent and curvature of) the graph of f more closely than tangents do, could equivalently have been derived as Newton's iteration applied to a function g whose graph is more nearly linear than the graph of f around the zero z that g and f have in common. For instance, Halley's third order iteration above is Newton's applied to $g(x) := f(x)/\sqrt{(|f'(x)|)}$. But Thesis 3.1 does not say which derivation will be the more convenient.

Thesis 3.1 implies that most of these notes will never generalize to the iterative solution of systems of equations nor to multi–point iterations. " $\mathbf{U(x)} = \mathbf{x} - \mathbf{g'(x)}^{-1}\mathbf{g(x)}$ " generally cannot be solved for a vector–valued function $\mathbf{g}$ of a vector $\mathbf{x}$. Iterations $x_{n+1} := U(x_n, x_{n-1}, \ldots, x_{n-k})$ generally do not behave like Newton's if $k \geq 1$, so Theorem 9.2 will come as a surprise.

§4. "Global" Convergence Theory from Textbooks

The behavior of iterations $x_{n+1} := U(x_n)$, also called Discrete Dynamical Systems, has become much better understood over the past few decades. Iterations $x_{n+1} := U(x_n, x_{n-1}, x_{n-2}, ..., x_{n-k})$ fall under the same rubric when rewritten as vector iterations $\mathbf{x}_{n+1} := \mathbf{U}(\mathbf{x}_n)$ in which the vector $\mathbf{x}_n = [x_n, x_{n-1}, x_{n-2}, ..., x_{n-k}]$. Although large values of k promise ultimately faster convergence, they offer little advantage because "ultimately" need not arrive much sooner than adequate accuracy would be achieved by simpler means. Anyway, so much less is known about the global behavior of iterations with $k \geq 1$ that we shall keep $k = 0$ except when discussing the Secant iteration, whose $k = 1$. And almost all variables will be kept real.

Presumably the roots z of the given equation "$f(z) = 0$" are roots of the equation "$z = U(z)$" too, so the desired roots lie among the *fixed–points* of U if any exist. The existence of fixed–points, some of which may be spurious because they are not roots, is a nontrivial issue. For example, the fixed–points of Newton's iteration, for which $U(x) := x - f(x)/f'(x)$, include the poles of f' as well as those zeros z of f at which $f'(z) \neq 0$, plus those zeros of both f and f' at which a justifiable redefinition of U sets $U(z) := z$. (Justification will be tendered later.) Fortunately poles are *repulsive* and zeros are usually *attractive* fixed–points of Newton's iteration; in general ...

- A fixed–point $z = U(z)$ is called "Attractive" if it belongs to some non–degenerate interval Ω from whose every other point x_0 the iteration $x_{n+1} := U(x_n)$ converges to z, though some early iterates may stray outside Ω before later iterates converge.
- A fixed–point $z = U(z)$ is called "Repulsive" if it belongs to some non–degenerate interval Ω throughout which $|U(U(x)) - z| > |x-z|$ when $x \neq z$; then, if Ω contains only every other iterate $x_{n+1} := U(x_n)$, consecutive iterates in Ω still move away from z.

A fixed–point can be both attractive (from one side) and repulsive (from the other), as are all the nonzero fixed–points of $U(x) = x \sin^2(1/x)$. Its fixed–point $z = 0$ is neither attractive nor repulsive. So is the zero of $\sqrt{|x|}$ to Newton's iteration; the zero of $1/\ln(|x|)$ is repulsive.

Global convergence theory is concerned with the existence of attractive fixed–points. In general, the best known conditions sufficient for at least one fixed–point to exist figure in the following ...

> **Lemma 4.1:** If U maps a closed interval Ω continuously into itself, then Ω contains at least one fixed–point $z = U(z)$.

Proof: If neither endpoint of Ω is a fixed–point of U then it maps each endpoint elsewhere into Ω, in which case they constitute a *Straddle* for the equation "$U(z) - z = 0$". END OF PROOF.

(Ω must include its two endpoints lest the fixed point lie not in Ω but on its boundary. If Ω is infinite it must include its endpoint(s) at $+\infty$ and/or $-\infty$, and the continuity of U there must be understood in an appropriate sense: U is deemed continuous at $+\infty$ if either of $U(1/w)$ and $1/U(1/w)$ approaches a finite limit as $w \to 0+$. Similarly for $-\infty$. And Ω must have distinct endpoints; the lemma may be rendered inapplicable if $+\infty$ and $-\infty$ are declared equal, thereby turning Ω topologically into a circle O that can be mapped continuously to itself by a rotation

without a fixed–point. Lemma 4.1 is a special case of the Brouwer/Schauder theorem valid for compact convex regions Ω in spaces of arbitrarily high, even infinite, dimension.)

Do not misconstrue the interval Ω as something introduced merely for the sake of additional inessential generality. Such a misapprehension could arise from the observation that U may be extended continuously to the whole real axis, and without introducing any new finite fixed–point, by declaring U' := 1 or else U' := 0 in each interval outside Ω. However, generality is not the motive for not thus dispatching Ω. It is essential to the following theory because U will be assumed to satisfy convergence conditions that need not be satisfied everywhere in general, yet they must be satisfied in an interval Ω wide enough to support useful inferences.

The foregoing lemma is easier to prove than apply because, given U and Ω, the confirmation that $U(\Omega)$ is contained in Ω is tantamount to an assertion about the extrema of U in Ω. Why should such an assertion cost much less computation than the location of a fixed point? Besides, the mere existence of fixed points cannot ensure that the iteration $x_{n+1} := U(x_n)$ will converge to any of them. For example, in $-1 \le x \le 1$, $U(x) := \sin(\pi x)$ has three fixed–points $z = 0$ and $z = \pm 0.736484448...$, all repulsive; $U(U(x))$ has seven therein, all repulsive; iteration cannot converge to any of them except by an unlikely accident. In general, if we desired no more than to find a fixed–point whose existence is guaranteed by lemma 4.1's hypotheses, we should proceed from those hypotheses to the construction of a fixed–point by Binary Chop guided in accordance with the following now obvious ...

> **Corollary 4.2:** If U maps a closed interval Ω continuously into itself, and if x in Ω is not a fixed–point of U, then there is at least one fixed–point $z = U(z)$ in Ω on the same side of x as $U(x)$.

It makes Binary Chop foolproof. But such is not our purpose now. Our purpose is to investigate whether and how the iteration $x_{n+1} := U(x_n)$ converges. (Faster than Binary Chop, we hope.)

The best known conditions sufficient for this iteration to converge require U to be a ...

***Contraction*:** $|U(x) - U(y)| < |x - y|$ for all distinct x and y in some interval Ω.

Contraction U must be continuous, if not differentiable with $|U'| < 1$ almost everywhere in Ω; and its interval Ω can contain at most one fixed–point $z = U(z)$. (Can you see why?)

> **Lemma 4.3:** If U contracts Ω into itself then the iteration $x_{n+1} := U(x_n)$ must converge in Ω to the fixed–point $z = U(z)$ from every initial guess x_0 in Ω, and both errors $|x_n - z|$ and steps $|x_{n+1} - x_n|$ shrink monotonically as n increases.

Proof outline: Contraction U shrinks $|x_n - z|$ monotonically, so iterates have one or two points of accumulation v and w. If different they would have to be swapped by U, thereby satisfying $0 < |v - w| = |U(w) - U(v)| < |w - v|$ paradoxically; instead, $v = w = z$. END OF PROOF.

But a contraction might contract no interval into itself; $\ln(x)$ for $x \ge 1$ is an example. Under what conditions can we ascertain that an interval Ω is contracted into itself? Conditions typical of the kind that appear in textbooks appear in the following lemmas:

Lemma 4.4: Suppose $-1 < U' \le 0$ throughout an interval Ω that includes both x_0 and $x_1 := U(x_0)$; then the iteration $x_{n+1} := U(x_n)$ converges in Ω to the one fixed–point $z = U(z)$ therein. Convergence is alternating with diminishing steps $|x_{n+1} - x_n|$. (Proof is left to the reader.)

Lemma 4.5: Suppose $0 \le U' \le \mu < 1$ for a positive constant μ throughout an interval Ω that includes both x_0 and $(U(x_0) - \mu x_0)/(1 - \mu)$; then the iteration $x_{n+1} := U(x_n)$ converges monotonically to the unique fixed–point $z = U(z)$ in Ω with diminishing steps $|x_{n+1} - x_n|$. (Proof is left to the reader.)

Lemma 4.6: Suppose $-1 < U' \le \mu < 1$ for a positive constant μ throughout an interval Ω that includes both x_0 and $X(x_0) := (U(x_0) - \mu x_0)/(1 - \mu)$; then the iteration $x_{n+1} := U(x_n)$ converges with decreasing error $|x_n - z|$ and diminishing steps $|x_{n+1} - x_n|$ to the unique fixed– point $z = U(z)$ in Ω.

Proof: Since U is a contraction on Ω, the fixed–point $z = U(z)$ is unique if it exists in Ω. That z does exist in Ω between x_0 and $X := (x_1 - \mu x_0)/(1 - \mu)$ follows from the observation that $(x_1 - U(X))/(x_0 - X) = (U(x_0) - U(X))/(x_0 - X) \le \mu$ provided $x_1 \ne x_0 \ne z$; that implies that $(X - U(X))/(x_0 - U(x_0)) \le 0$ and therefore $x - U(x)$ changes sign at some $x = z$ between X and x_0. In fact z lies between X and $(x_0 + x_1)/2$ since $(x_1 - z)/(x_0 - z) = (U(x_0) - U(z))/(x_0 - z) > -1$; consequently $(z - (x_0 + x_1)/2)/(z - x_0) > 0$, which implies that $z - (x_0 + x_1)/2$ has the same sign as $z - x_0$, which has the same sign as $X - x_0$. To complete the proof we shall show that U contracts a subinterval of Ω including x_0 into itself, and then invoke Lemma 4.3.

To simplify the argument suppose that $x_0 < x_1$; otherwise reverse the signs of x and U. Now we have $x_0 < (x_0 + x_1)/2 < z \le X = (x_1 - \mu x_0)/(1 - \mu)$. Set $w := z + (1 - \mu)(X - z)/(1 + \mu)$ and $v := z - (1 - \mu)(X - z)/(1 + \mu) = x_0 + (2z - x_0 - x_1)/(1 + \mu)$; evidently $x_0 < v < z < w < X$. Now we shall confirm that $U(x)$ contracts the subinterval $x_0 \le x \le w$ into itself. First we obtain upper bounds for $U(x)$:

When $x_0 \le x \le v$, $U(x) \le U(x_0) + \mu(x - x_0) \le x_1 + \mu(v - x_0) = w$;
when $v \le x \le z$, $U(x) \le U(z) - (x - z) = 2z - x \le 2z - v = w$;
when $z < x \le w$, $U(x) \le U(z) + \mu(x - z) < z + (x - z) \le w$.

Next we obtain lower bounds for $U(x)$:

When $x_0 \le x < z$, $U(x) > x \ge x_0$;
when $z < x \le w$, $U(x) > U(z) - (x - z) = 2z - x \ge 2z - w = v > x_0$.

Evidently $x_0 < U(x) \le w$ too when $x_0 \le x \le w$, as claimed. END OF PROOF.

Lemma 4.6 is nearly the most general of its kind, and yet often too difficult to apply. Difficulty arises from the possibility that μ and the minimum width $|x_1 - x_0|/(1 - \mu)$ of Ω may chase after each other. For example, given x_0 and $x_1 := U(x_0)$ and $U'(x_0) < 1$, we have to make a guess at Ω at least as wide as $|x_1 - x_0|/(1 - U'(x_0))$; then somehow we must estimate the range of $U'(\Omega)$ hoping it will be narrow enough to satisfy a lemma's requirements. But if that estimated range is

too wide, say if $\mu \geq U'(\Omega)$ is so big that $(x_1 - \mu x_0)/(1 - \mu)$ lies beyond Ω, we must widen Ω to include this point, thereby perhaps increasing μ and forcing Ω to be widened again, and so on. This can go on forever for examples like $U(x) := \sqrt{(1 + x^2)} - 1/(z + \sqrt{(1 + z^2)})$ when $0 < x_0 < z - 1$ although its iteration always converges. The chase need never end because the lemmas' requirements that $-1 < U' \leq \mu < 1$ in Ω merely suffice for convergence; they are not necessary. For example, iterates converge from every x_0 to $z = 0$ for $U(x) := -\arctan(x)$ with $U'(z) = -1$, and for $U(x) := x - \tanh^3(x)$ with $U'(z) = 1$, though both examples converge *sublinearly* (*i.e.*, extremely slowly): $|x_n - z| = O(1/\sqrt{n})$.

The foregoing three lemmas are really local convergence theorems posing as global. They are applicable only in a sufficiently small neighborhood Ω of a fixed–point $z = U(z)$ at which $|U'(z)| < 1$, in which case $|x_n - z|$ ultimately decreases with every iteration, converging to zero linearly like $|U'(z)|^n$ or superlinearly if $U'(z) = 0$. However, finding a neighborhood to which a lemma above is applicable can be almost as hard as finding z. Besides, convergence can occur without ultimate monotonic decline in $|x_n - z|$, as when $U(x) := e^{-x} - 1$; for this example the iteration converges to $z = 0$ alternatingly, sublinearly and invariably, as we'll see in Ex. 5.3 .

Apparently the "global" theory of iterations' convergence presented in most textbooks answers questions that the designers of root–finding software are unlikely to ask, much less answer.

§5. Global Convergence Theory

What pattern of behavior distinguishes convergent iterations from the others ?
This question matters to software designers because, by mimicking this pattern in our root–finding software, we hope to enhance its prospects for success. The pattern is slightly more complicated than a monotonic decline in $|x_n - z|$ as n increases. To suppress superfluous complexity we shall try to describe only the pattern's essentials. What is essential? It is whatever persists after inessential changes of variables, *i.e.* after homeomorphisms.

Consider any change from x to a new variable $X = \mathbf{X}(x)$ which is continuous and invertible, and therefore monotonic, on the domain Ω of x ; we shall let $x = \mathbf{x}(X)$ denote the inverse change of variable, also continuous and monotonic on its domain $\mathbf{X}(\Omega)$. Usually both changes of variable shall be differentiable too, in which case $\mathbf{X}'(x)$ and $\mathbf{x}'(X) = 1/\mathbf{X}'(\mathbf{x}(X))$ must keep the same constant nonzero sign inside their domains. $U(x)$ changes into $H(X) := \mathbf{X}(U(\mathbf{x}(X)))$. If the iteration $x_{n+1} := U(x_n)$ converges from x_0 to $z = U(z)$, we expect $X_{n+1} := H(X_n)$ to converge too from $X_0 := \mathbf{X}(x_0)$ to $Z := \mathbf{X}(z) = H(Z)$, though divergence either to $+\infty$ or to $-\infty$ may have to be redefined as "convergence to infinity" in case z is an infinite endpoint of Ω , or Z an infinite endpoint of $\mathbf{X}(\Omega)$.

Besides fixed–points and convergence, what qualities must each of U and H inherit from the other independently of $\mathbf{X}$?

- Continuity
- Separation: x lies between $U(x)$ and $U(U(x))$ if and only if $X := \mathbf{X}(x)$ lies between $H(X)$ and $H(H(X))$.
- Differentiability: $H'(X) = \mathbf{X}'(U(\mathbf{x}(X)))\, U'(\mathbf{x}(X))\, \mathbf{x}'(X)$ if all derivatives are finite.

When they exist, both derivatives $H'(X)$ and $U'(\mathbf{x}(X))$ have the same sign but they usually have different values except at *Stationary Points* (where both derivatives vanish) and at fixed–points: Whenever $z = U(z)$ and consequently $Z := \mathbf{X}(z) = H(Z)$ then also $H'(Z) = U'(z)$. Then, if both fixed–points z and Z are finite and if the respective iterations $x_{n+1} := U(x_n)$ and $X_{n+1} := H(X_n)$ converge to them, both converge at the same *Rate* $:= \lim\inf_{n \to \infty} \ln(|x_n - z|^{-1/n}) = -\ln|U'(z)| \geq 0$. Sublinear convergence has Rate zero; linear convergence has a positive Rate. And when this Rate is infinite then both iterations may be shown to converge with the same superlinear *Order* $:= \lim\inf_{n \to \infty} (-\ln|x_n - z|)^{1/n} \geq 1$; higher Order implies ultimately faster convergence.

Like the foregoing qualities, conditions for convergence should ideally be inheritable by each of U and H from the other. By this criterion typical textbook conditions, like the uninheritable bounds upon U' in lemmas 4.4 to 4.6 above, are not ideal. Ideal conditions follow.

> **Theorem 5.1: Sharkovsky's No-Swap Theorem**
> Suppose U maps a closed interval Ω continuously into itself; then the iteration $x_{n+1} := U(x_n)$ converges to some fixed–point $z = U(z)$ from every x_0 in Ω if and only if these four conditions, each of which implies all the others, hold throughout Ω :
>
> **No-Swap Condition:** U exchanges no two distinct points of Ω ; in other words, if $U(U(x)) = x$ in Ω then $U(x) = x$ too.

No Separation Condition: No x in Ω can lie strictly between $U(x)$ and $U(U(x))$; in other words, if $(x - U(x))(x - U(U(x))) \le 0$ then $U(x) = x$.

No Crossover Condition: If $U(x) \le y \le x \le U(y)$ in Ω then $U(x) = y = x = U(y)$.

One-Sided Condition: If $x_1 := U(x_0) \ne x_0$ in Ω then all subsequent iterates $x_{n+1} := U(x_n)$ also differ from x_0 and lie on the same side of it as does x_1 . (Compare Corollary 4.2 above.)

These conditions have been rediscovered several times since they were first established by A.N. Sharkovsky [1964, 1965]. The proof that each implies all others is too long to reproduce fully here but elementary enough to leave to the diligent reader helped by the following suggestions:

Think of $\Omega x \Omega$ as a square whose lower–left–to–upper–right diagonal is touched or crossed at every fixed–point by the graph of U , which enters the square through its left side and exits through its right. That graph and its reflection in the diagonal touch or cross nowhere else when the No–Swap condition holds. When the No Separation condition is violated, all attempts to draw both graphs must violate the No–Swap condition too. Similarly for the No Crossover condition; therefore these three are equivalent conditions. The One–Sided condition obviously implies No Separation; and a violation of One–Sidedness can be shown soon to violate No Crossover too. Thus all four named conditions are equivalent to each other though not yet proved equivalent to convergence from every starting point in Ω ; that proof follows the next lemma.

Besides pertaining to an iterating function U , the One–Sided condition is satisfied by any sequence $\{ x_0, x_1, x_2, x_3, ... \}$, regardless of its provenance, whose every member x_n lies on the same side of all subsequent members x_{n+m} with $m > 0$. In other words, that sequence is One–Sided just when, first, if any two members are equal so are all members between and after them, and secondly, for every integer $n \ge 0$, no members of the sequence of differences $\{ x_{n+1} - x_n,\ x_{n+2} - x_n,\ x_{n+3} - x_n,\ ... \}$ have opposite (non-zero) signs. Note that every subsequence of a One–Sided sequence is One–Sided too. Some One–Sided sequences are *Ultimately Monotonic* in the sense that all but finitely many differences $x_{n+1} - x_n$ have the same sign; such sequences obviously converge, perhaps to infinity. Other One–Sided sequences are the subject of the next lemma:

Lemma 5.2: The No-Man's-Land Lemma
If the One–Sided sequence $\{ x_0, x_1, x_2, x_3, ... \}$ is *not* ultimately monotonic then it can be partitioned into two disjoint infinite subsequences, one of which ascends strictly monotonically to a limit no larger than the limit to which the other descends strictly monotonically; if these limits differ, the gap between them is a no–man's–land containing no member of this sequence.

Proof outlined: The ascending subsequence consists of those $x_n < x_{n+1}$, and the descending subsequence consists of those $x_n > x_{n+1}$. For instance, if x_m is a local maximum and x_j the subsequent local minimum in the sequence, whereupon

$$... \; x_{m-1} < x_m > x_{m+1} > ... > x_{j-1} > x_j < x_{j+1} \; ... \qquad (m < j) ,$$

then x_{m-1} and x_j are consecutive members of the ascending subsequence (note that One–Sidedness implies $x_{m-1} < x_j$) while $x_m, x_{m+1}, ..., x_{j-1}$ are consecutive members of the descending subsequence. It soon follows that each subsequence is strictly monotonic and bounded by the other. END OF PROOF.

Return to the proof of Sharkovsky's No-Swap theorem; suppose U satisfies the four named conditions of his theorem on Ω . Then the iteration $x_{n+1} := U(x_n)$ generates a One–Sided sequence. If it did not converge then, according to the no–man's–land lemma, it would have two points of accumulation with no iterate between them; and then because U is continuous it would swap them, contrary to the No–Swap condition. Therefore the iteration does converge.

I am indebted to the late Prof. Rufus Bowen for pointing out Sharkovsky's work. It answers easily many convergence questions that would be awkward without it. Here are two examples:

Example 5.3: Suppose $U(x) := e^{-x} - 1$ and Ω is the whole real axis; the iteration $x_{n+1} := U(x_n)$ converges to $z = 0$ from every starting point because $U' < 0$ (so U has just one fixed–point) and U cannot swap two points in Ω . No–Swap follows from the fact that the graphs of U and its inverse intersect just once, which follows from the fact that $e^{-x} - 1 + \ln(1+x)$ cannot vanish if $-1 < x \neq 0$, which follows after differentiation from $e^x > 1+x$. Convergence is alternating because $U'(0) = -1 < 0$, and $x_n = O(\sqrt{6/n})$ because $U(U(x)) = x - x^3/6 + ...$. END EX. 5.3.

Example 5.4: Suppose f is a rational function with simple real interlacing zeros and poles, one of them a pole at ∞ . An instance is $f(x) := p(x)/p'(x)$ where p(x) is a polynomial all of whose zeros are real. Another instance is $f(x) := \det(xI - A)/\det(xI - \hat{A}) = \prod_i (x - z_i)/\prod_j (x - \hat{o}_j)$ in which A is an hermitian matrix, $\hat{A}$ is obtained from it by striking off its last row and column, and the I 's are identity matrices; the zeros z_i lie among the eigenvalues of A , and the poles $\hat{o}_j$ are the distinct eigenvalues of $\hat{A}$ that are not also eigenvalues of A . That they interlace, i.e.,

$$z_0 < \hat{o}_1 < z_1 < \hat{o}_2 < z_2 < ... < \hat{o}_K < z_K \; ,$$

is a well–known theorem attributed to Cauchy. We do not know the zeros z_i but, like Y. Saad [1974], propose to compute them by running Newton's iteration $x_{n+1} := x_n - f(x_n)/f'(x_n)$. Does it converge? If so, to what? These are thorny questions, considering how spiky is the graph of f , and yet Newton's iteration can be proved to converge to some zero z_i from every real starting value except a countable nowhere–dense set of starting values from which the iteration must converge accidentally (after finitely many steps) to a pole $\hat{o}_j$. The proof outlined below is extracted from one first presented in my report [1979'].

For the proof's sake express f in the forms $f(x) = x - ß - \sum_j w_j/(x-\hat{o}_j) = 1/\sum_i v_i/(x-z_i)$ in which the coefficients ß , w_j and v_i are determined as sums, products and quotients of differences among the zeros z_i and poles $\hat{o}_j$ by matching the behavior of f(x) as x approaches each pole or zero. By counting negative differences we find every $w_j > 0$ and every $v_i > 0$, and by matching behavior at ∞ we find $\sum_i v_i = 1$. Newton's iterating function now takes the forms

$$N(x) := x - f(x)/f'(x) \quad \text{except at poles } \hat{o}_j \text{ of } f,$$
$$= (ß + \sum_j (2x-\hat{o}_j)w_j/(x-\hat{o}_j)^2)/(1 + \sum_j w_j/(x-\hat{o}_j)^2) \quad \text{if no } \hat{o}_j = x,$$
$$= (\sum_i z_i v_i/(x-z_i)^2)/\sum_i v_i/(x-z_i)^2 \quad \text{if no } z_i = x .$$

From these we infer easily that N maps the whole real axis continuously into an interval whose endpoints are the outermost zeros z_0 and z_K; and every zero z_i is a strongly attractive fixed–point of N because $N'(z_i) = 0$, and every pole $\hat{o}_j$ is a strongly repulsive fixed–point because $N'(\hat{o}_j) = 2$; and N has no more fixed–points. To conclude that the iteration always converges (almost always to a zero z_i) we have to confirm that N cannot swap two points. If N did swap x and $y \neq x$, the equations $y = N(x)$ and $x = N(y)$ could be turned into $\sum_i v_i(y-z_i)/(x-z_i)^2 = 0$ and $\sum_i v_i(x-z_i)/(y-z_i)^2 = 0$ which, when subtracted and divided by $y-x$, would simplify to $0 = \sum_i v_i((x-z_i)^{-2} + (x-z_i)^{-1}(y-z_i)^{-1} + (y-z_i)^{-2}) > 0$, which is impossible. END EX. 5.4.

The foregoing example is an instance of a general algebraic decision procedure based upon Sharkovsky's No–Swap theorem:

> Suppose an interval Ω and a *rational* function U are given. Then the question "Does the iteration $x_{n+1} := U(x_n)$ converge in Ω from every initial x_0 in Ω ?" can be decided by performing finitely many rational operations without solving any nonlinear polynomial equation.

U satisfies the No–Swap condition if and only if the simplified form of the rational function

$$1 + (U(U(x)) - U(x))/(U(x) - x)$$

has no zeros in Ω which are not also zeros of $U(x) - x$. This can be tested by removing common divisors from certain polynomials and then counting their sign–changes in Ω by computing Sturm sequences. Whether U maps Ω continuously into itself can also be determined from certain polynomials' sign–changes in Ω counted by computing Sturm sequences. The details were worked out by R.J. Fateman [1977] in a program written to run on the computerized algebra system MACSYMA. The procedure is practical only on a fairly big computer because some of the polynomials in question can have large degrees, as large as the square of the degree of the numerator or denominator of U.

Sharkovsky's No–Swap theorem is the simplest of a family of relationships he discovered for the properties of the fixed-points $z_k = U^{[k]}(z_k)$ of a continuous iterating function U and of its compounds

$$U^{[k]}(x) := U(U(U(\ldots U(x)\ldots))) \quad k \text{ times.}$$

For instance, if $U^{[3]}$ has a fixed-point that is not a fixed-point of U, then for every integer $k > 1$ there are fixed points of $U^{[k]}$ that are not fixed-points of $U^{[m]}$ for any divisor m of k. For an elementary treatment of Sharkovsky's relationships see Huang [1992]. For a brief discussion of these and related results and other proofs, see Misiurewicz [1997].

§6. A One-Sided Contribution to Software Strategy

Suppose an iterating function U has been chosen because its fixed–point(s) $z = U(z)$ coincide(s) with the root(s) of a given equation to be solved, and because the iteration $x_{n+1} := U(x_n)$ is expected to converge to a root quickly. When should this iteration be stopped or amended?

- When it appears to have converged well enough, or about as well as it ever will.
- When it will converge too slowly.
- When it will not converge.

How can non-convergence be predicted? A portent, at least when U is continuous, is a violation of the One–Sided condition in Sharkovsky's No–Swap theorem. That condition is the only one of the theorem's conditions that software can check: Until One–Sidedness fails, or until so many iterations have been executed as must arouse suspicions that convergence will be too slow, the software has no better option than to persist in the chosen iteration $x_{n+1} := U(x_n)$. How can software detect slow convergence or a failure of One–Sidedness? The answer to this question, at least for continuous iterating functions U, is *Brackets*.

A *Bracket* is an ordered pair $\{x_«, x_»\}$ of arguments, normally both iterates, between which all subsequent iterates must lie if they are to constitute a One–Sided sequence. A bracket is usually a straddle but this is not obligatory; $U(x) - x$ need not take opposite signs at the ends of a bracket. Initially, $x_«$ and $x_»$ are set to the endpoints, possibly infinite, of the interval Ω in which a fixed–point of U is being sought. Subsequently, as suggested by the no–man's–land lemma, $x_«$ is the most recent of any iterates x_n that satisfied $x_n < U(x_n)$, and $x_»$ is the most recent x_n that satisfied $U(x_n) < x_n$, if any. Consequently, once a bracket becomes a straddle it stays a straddle. Normally every iteration narrows the bracket by moving one end closer to the other. Normally at least one end of the bracket converges monotonically to the sought fixed–point of U.

Software must cope with whatever abnormal behavior a bracket exposes. For instance, bracket $\{x_«, x_»\}$ need not be a straddle; $U(x_«) - x_«$ and $U(x_») - x_»$ may have the same sign at first because U does not map Ω into itself, and later perhaps because Ω contains no fixed–point of U or more than one. A new iterate $U(x_n)$ may stray outside the current bracket perhaps because x_n is too close to a strongly repulsive fixed–point, or perhaps because U violates the No–Swap condition, or because U does not map Ω into itself. Normal behavior, consistent with the no–man's–land lemma, may require software intervention too if the width of the bracket does not shrink fast enough, as may happen because convergence is alternating but very slow, or because both ends of the brachet are converging to different limits swapped by U, or because one end stopped moving after the iteration's convergence became monotonic.

Tactics can be chosen to cope with aberrations only after they have been diagnosed. For instance, splitting the difference (as in Binary Chop) copes well with alternating slow (non)convergence; a better expedient is Steffenson's, which is tantamount to one step of Secant Iteration to solve $U(z) - z = 0$. Occasional difference extension (extrapolation) helps to accelerate monotonic but slow behavior; a way to do it is Aitken's Δ^2 Process, which takes $U(x) \approx z + (x - z)\mu$ to be an approximate model for unknown constants z and μ determined from three consecutive iterates: $z \approx z_n := x_{n+1} - (x_{n+1} - x_n)^2/(x_{n+1} - 2x_n + x_{n-1})$. Such expedients afford software the possibility

of extracting tolerable rates of convergence from iterations that would otherwise converge too slowly or not at all. The programmer's options and the occasions that call for them would bewilder but for diagnostic information furnished by brackets and by Sharkovsky's theorem, at least when U is continuous.

Diagnosis is complicated when U may be discontinuous. Then a straddle may enclose a jump or pole instead of a root or fixed–point. Reasons to doubt whether a pole can always be distinguished from a root by solely numerical means will be presented later (Ex. 6.3).

Diagnosis is interesting also when U(x) may be undefined for some arguments x . What should software do if an attempt to compute $U(x_n)$ produces instead an error–indication like "INVALID OPERATION" ? In the past that has served as an excuse to abandon computation, but nowadays the temptation to quit should be resisted. Unless it is trapped, an "Invalid" operation like 0/0 or $\sqrt{-3}$ on most computers to-day will produce a *NaN* , and subsequent arithmetic operations upon it will almost all propagate it. It can be detected because the predicate "*NaN* = *NaN*" is False; this ostensible paradox merely confirms that *NaN* is *N*ot *a N*umber. Consequently, when $U(x_n)$ turns out to be *NaN* instead of a number the appropriate inference is that x_n has fallen outside U 's domain. The appropriate response is to supplant x_n by something else closer to x_{n-1} and therefore, presumably, inside U 's domain. Then computation can be resumed.

A policy of continued computation past an invalid operation may seem reckless, and sometimes it is. However the opposite policy, that abandons computation after any "Invalid" operation, is tantamount to abandoning the search for an equation's root merely because the computer signaled

"Look elsewhere for what you seek."

That policy of abandonment frustrates software users who wish to solve an equation without first ascertaining the boundary of its domain. Why should its domain be much more obvious than the equation's root? Except for examples contrived for classroom purposes, an equation's domain is generally found by an exploration that resembles the search for a root. Combining both searches by forgiving "Invalid" operations makes more sense than abandonment does.

Searching continued past "Invalid" operations is now the policy built into the [SOLVE] keys on Hewlett-Packard calculators starting with the hp-18C *Business Consultant* and the hp-28C ; see McClellan [1987]. Consequently they can be used with far less fuss than other unforgiving software requires to solve difficult equations. Here is my favorite example:

Example 6.1: We wish to decide whether the equation $(\tan(z) - \arcsin(z))/z^4 = 0$ has a *positive* root z or not. Unforgiving software will fail to find it despite repeated attempts each of which starts, say, Newton's iteration $x_{n+1} := N(x_n)$, whose iterating function is

$$N(x) := x + 1/\left(4/x - \left(1 + \tan^2(x) - 1/\sqrt{((1-x)(1+x))} \right)/(\tan(x) - \arcsin(x)) \right) ,$$

from small positive initial guesses like $x_0 = 0.1$. For the sake of realism we must pretend not to know that the equation's domain is the interval $0 < x \le 1$. Whatever its domain, the iteration behaves as if doomed to move through it from left to right and escape. ($N(x) > 1$ whenever $0.46137 < x < 0.99964$.) A few such escapes followed by "Invalid" operations suggest fairly persuasively that no positive root z exists, but in fact $z = 0.9999060...$. From random initial guesses x_0 scattered uniformly between 0 and 1 , Newton's iteration is more than 1000 times

more likely to encounter an "Invalid" operation than to converge to this z. Despite these odds, the hp-28C solves this equation quickly (by means of a modified Secant iteration) from any initial guess(es) between 0 and 1, thereby vindicating a policy of continued computation past forgiven "Invalid" operations. (Some recent *Casio* calculators appear to do likewise.)

Sharkovsky's No–Swap theorem contributes more than a convergence criterion to the strategy and theory of iteration. It changes our attitudes. Rather than focus exclusively upon conditions sufficient for convergence, we also make use of criteria that tell us when an iteration may not converge unless we do something more than merely iterate. As we pursue this line of thought, we come to understand why successful root-finding software need not always find a root, especially if none exists. Satisfactory software should almost always find a root if any are to be found, and usually find it fast, and come to a conclusion soon if a root is not going to be found. Deemed unsatisfactory are indecisive iterations that meander interminably. Our foray into iteration theory is a search for conditions under which an iteration won't meander. We'll find some later in §8.

What if the object sought is nowhere to be found? Root–finding software can cope with this possibility by finding something other than a root, provided the substitution is made manifest to the user of the software. An obvious candidate to supplant a zero of f that cannot be found is a local minimum of $|f|$. However this substitution poses two challenges, one for the designer of the software and one for its user. The designer must devise an algorithm whose efficiency is not too much degraded by the necessity to switch, sometimes repeatedly, between two tasks:

seeking a nonzero minimum, and
seeking a zero.

After the software has found one, the user may be unable to decide which of the two has been found in some cases.

Example 6.2:

$f(x) := (x - (7 - (x - (7 - x))))^2$ and $f'(x) = 6(x - (7 - (x - (7 - x))))$ (DON'T REMOVE PARENTHESES !)

will be calculated exactly (unblemished by roundoff) on every computer or calculator built in the Western world for all x close enough to $14/3 = 4.666\ldots$, and therefore neither calculated value can vanish when computed in floating–point arithmetic since $14/3$ is not a floating–point number on any of those machines. Consequently, if $æ := 1.000\ldots001 - 1$ is a small positive number like roundoff in numbers near 1, no way exists to distinguish f and its derivative from $f + æ^4$ and its derivative using only their values computed in floating–point arithmetic. In other words, software that finds a positive local minimum of $|f|$ instead of a double zero deserves no opprobrium if it cannot tell which it has found from numerical values alone.

Discriminating between a pole and a zero across which a function changes sign can be difficult too in certain very rare cases like …

Example 6.3: The computed values of $f(x) := 1/(x - (7 - (x - (7 - x))))$ and of $F(x) := 1/\big((x - (7 - (x - (7 - x)))) + æ^4/(x - (7 - (x - (7 - x)))) \big)$ are the same everywhere although f has a pole and F a zero at $x = 14/3$.

Despite a few ambiguous cases, root–finding software can describe its find to its user sufficiently well to make the attempt worthwhile. The software can deliver its latest bracket $\{x_«, x_»\}$ and, to help the user interpret it, an indicator that points to one of the following cases:

•A zero $z = x_« = x_»$ has been found because the computed $f(z) = 0$.

•A sign–reversal has been found; $x_«$ and $x_»$ differ only in their last significant digits and $f(x_«)f(x_») < 0$. Three sub–cases have to be distinguished:
- •Probably a zero since $|f(x)|$ grows as $(x-x_«)(x-x_»)$ increases from 0 .
- •Probably a pole since $|f(x)|$ drops as $(x-x_«)(x-x_»)$ increases from 0 .
- •Otherwise probably a jump discontinuity.

•A local minimum of $|f(x)|$ has been found. Three sub–cases have to be distinguished:
- •Probably a double zero since $|f(x)|$ grows rapidly as $(x-x_«)(x-x_»)$ increases from 0 .
- •Apparently $f(x)$ is a nonzero constant when x is near or between $x_«$ and $x_»$.
- •Otherwise probably a nonzero local minimum of $|f(x)|$ at some x near $x_«$ and $x_»$.

Good root–finding software, able to present all those possibilities to its users without violating Albert Einstein's maxim, that

" Everything should be made as simple as possible, but not simpler ",

has to be more complicated to use than any single user might like, and harder to design than most programmers will like. Well–designed software is parsimonious, uncluttered by extraneous inputs and outputs. The necessary outputs, as we have seen, are now obvious:

- The latest bracket $\{x_«, x_»\}$ found in lieu of a zero and, to help interpret it,
- An integer indicator for use in an indexed branch or `Case` statement.

The inputs needed by good root–finding software are unobvious because equations to be solved are so diverse. Equations are like *canapés*; after one comes another. Often the equation to be solved has the form $f(z, p) = 0$ with a parameter p that will take several values for each of which a root $z(p)$ has to be computed. For some equations the derivative $\partial f(x, p)/\partial x$ is easy to compute, for others difficult. Often the equation has more than one root; some users seek all the roots; other users wish to avoid all but one root. Sometimes high accuracy is desired; often not. Only a cluttered menu can cater to all tastes. To promote parsimony I offer here my suggested list of inputs to good root–finding software:

- The name of the program that computes either $f(x, p)$ or else $f(x, p) / \partial f(x, p)/\partial x$.
- One or two initial guesses x_0 , x_1 to start the search for a root z of $f(z, p) = 0$.
- An initial bracket $\{x_«, x_»\}$ to constrain that search. (It can be $\{-\infty, +\infty\}$.)
- A place for (optional) parameter(s) p to be passed to the named program $f(\ldots)$.

Initial guesses are essential inputs even if brackets are supplied because, for example, when a root $z(p)$ is plotted as a function of a slowly changing parameter p the old value of $z(p)$ is often a good first guess at the new $z(p)$. The program that provides initial guesses should be able to find a record (in `SAVEd` or `static` variables) of the old p and $z(p)$ for use when the new p is not too different; $\partial z/\partial p = -(\partial f/\partial p)/(\partial f/\partial x)|_{x = z}$ usually helps too.

In programming languages that allow argument lists of variable lengths, the parameter(s) p can be the root–finder's last argument(s) and then can be passed *verbatim* to the named program f(…) as its last argument(s), thereby avoiding unnecessary prejudice against parameters of mixed types (arrays, lists, pointers, procedures, strings, integers, floating–point numbers, …). The conveyance of optional parameter(s) p has to be fast, not encumbered by excessive overheads to de–reference p , because the root–finder invokes f(…) many times for each computation of z(p) . This kind of computation, either the inversion of a given function f(z) = p or the conversion of an implicit definition f(z, p) = 0 to an ostensibly explicit invocation of the solution z(p) , is the root–finder's most frequent application, and deserves software engineers' attention.

Conspicuous omissions from my list deserve explanation. The list includes no upper limit upon the number of iterations. There are three reasons to omit it. First, such a limit is difficult to choose; might the search have succeeded had it been allowed two more iterations? Second, abandoning a search prematurely may be justified after the expiry of some preassigned quantum of time worth more than the root being sought; but f(…) can take longer to compute for some arguments than for others, so a stopping criterion should count clock–ticks, not iterations. Third, by using brackets, good software need never get stuck in an interminable sequence of iterations; besides, as we shall see in the course of developing the theory below, well–designed software can practically always ensure that f(…) becomes negligible after a moderate number of iterations no matter how slowly they converge. By stopping after f(…) becomes negligible, or else after the clock runs out, we can can omit iteration counts from our stopping criteria.

Also conspicuously absent from my list of inputs are two tolerances to serve in stopping criteria, one for the negligibility of f(…) and a second for the negligibility of the difference between consecutive iterates. Such tolerances will be chosen cavalierly if they must be constants chosen in advance. Chosen properly, they generally depend upon the same arguments as f(…) depends upon; therefore these tolerances should be computed inside the program that computes f(…) .

Example 6.4: Consider

$$f(x) := ((((((((((x-12)x+66)x-220)x+495)x-792)x+924)x-792)x+495)x-220)x+66)x-12)x+1$$

and pretend not to notice that this is an unfortunate way to compute $(x-1)^{12}$. Error analysis reveals that the difference, due to roundoff, between f(x) and its computed value must be smaller than roughly $\Delta f(x) := 12\,|x|(|x| + 1)^{11}\,æ$ but not often enormously smaller. Here $æ := 1.000\ldots001 - 1$ is the roundoff threshold for the computer's floating–point arithmetic; typically $æ = 1/2^{52} = 2.22/10^{16}$ for 8-byte floating–point. For arguments x near the zero z = 1 of f , its error bound $\Delta f \approx 5.5/10^{12}$ is not enormously bigger than observed errors almost as big as $2/10^{13}$ in computed values of f . How can someone be expected to guess either constant $5.5/10^{12}$ or $2/10^{13}$ in advance?

Computing (or guessing) a tolerance Δf(…) for the negligibility of f(…) within the program that computes f(…) lets Δf(…) serve in a simple way to stop the search for a zero as soon as f(…) becomes negligible:

Whenever the computed f would be no bigger than Δf , return 0 in place of f .

This immediately stops the root–finder at what it thinks is a zero. Techniques for computing Δf include *Running Error–Analysis* and *Interval Arithmetic*, both described in a text by Higham [2002] with ample references to the literature. These techniques can add considerably to the time needed to compute f alone, so they should not be employed indiscriminately. The subprogram

can record (in `SAVEd` or `static` variables) the last few arguments at which f was computed, and compute Δf only at new arguments close enough to an old one that stopping is a plausible possibility. In any event, iterations prolonged much beyond the time when $|f| \le \Delta f$ will waste time dithering, so computing Δf too often may waste less time than not computing Δf at all.

My list of inputs also omits a tolerance Δx for the difference between consecutive iterates, or the width $|x_{\text{»}}-x_{\text{«}}|$ of a bracket or straddle, because the use of Δx to stop iteration lends itself too easily to misinterpretation. The clear intention is to stop when the iteration has come within $\pm\Delta x$ of the desired zero z , and that is what happens when convergence is so fast (as it usually is) that $|x_{n+1} - x_n|$ is rather bigger than $|x_{n+1} - z|$; but then little is gained by stopping the iteration before $|f| \le \Delta f$. Only when convergence is slow can Δx be used to stop iteration in time to save much time, but then this stopping criterion becomes treacherous. If convergence is slow because z is a multiple zero (see §10 on Accelerated Convergence to Clustered Zeros below) then $|x_{n+1} - x_n|$ can stay arbitrarily smaller than $|x_{n+1} - z|$ even though $|f(x_{n+1})|$ usually plunges below any practical threshold Δf fairly soon (see Theorem 7.6); then not much is gained by stopping sooner, say when $|x_{n+1} - x_n| \le \Delta x$, beyond the illusion that $|x_{n+1} - z| \le \Delta x$ too. If roundoff interferes severely with convergence, not even a straddle $\{x_{\text{»}}, x_{\text{«}}\}$ can be trusted to contain z , not even approximately.

Recall Example 6.4, $f(x) := (\dots)x + 1 = (x-1)^{12}$ above. The uncertainty $\pm\Delta f$ in f propagates into an uncertainty $\pm(\Delta f)^{1/12}$ in the computed zero $z \approx 1$; for 8-byte floating–point arithmetic carrying the equivalent of about 15 sig. dec., the computed z is uncertain in its second decimal after the point. In fact, root–finders frequently stop with a straddle $\{x_{\text{»}}, x_{\text{«}}\}$ whose ends differ only in their 13th decimal or beyond but which both differ from 1 by more than 0.07 . How could a tolerance Δx be chosen meaningfully in a case like this?

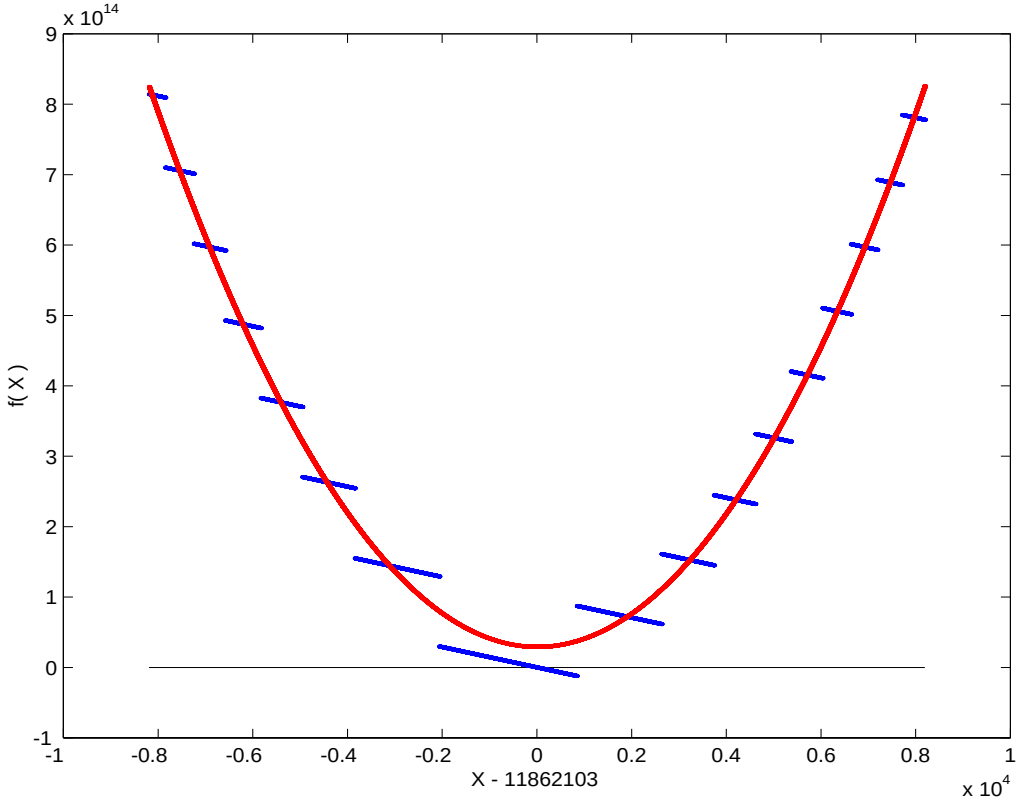
Generally, an appraisal of uncertainty in a computed zero z of f begins with an estimate of uncertainty Δf in the computed value of f . After that, uncertainty in z is either trivial or very difficult to ascertain; see Higham [2002]. Including a tolerance Δx among the root–finder's inputs to stop iteration sooner deceives users too often while contributing little to speed and less to error–analysis, in my experience, so I have omitted it from my root–finding software. Other programmers think otherwise. Rather than argue longer here about where (outside the root–finder) error–analysis should play its rôle, I prefer to develop root–finding iterations that find roots fast enough to render early termination (before $|f| \le \Delta f$) of the iteration uninteresting.

Still, if an iteration's convergence is normally superlinear and never worse than linear, here is a strategy that may save an iteration or two if monotonic convergence shrinks brackets too slowly:

Suppose *Difference Quotients* $(x_{k+1} - x_k)/(x_k - x_{k-1}) \to 0$ as $k \to \infty$. Provided (while roundoff is insignificant) these quotients will constitute a decreasing sequence as $x_k \to z$, after $L := (x_{k+1} - x_k)/(x_k - x_{k-1}) < 1$ we can soon deduce that the error $|z - x_{k+1}| \le |x_{k+1} - x_k|\cdot L/(1–L)$. Therefore iteration can be stopped after at least two or three consecutive difference quotients, all less than 1 , have strictly decreased to a latest difference quotient L small enough that $|x_{k+1} - x_k|\cdot L/(1–L) < \Delta x$. If this ever happens, x_{k+1} can be delivered with a reasonable expectation that $|x_{k+1} - z| < \Delta x$, and without having to compute $f(x_{k+1})$ though checking that it is negligible would be prudent. Don't omit the divisor (1–L) lest iteration be stopped far too soon when convergence is slowly slowing.

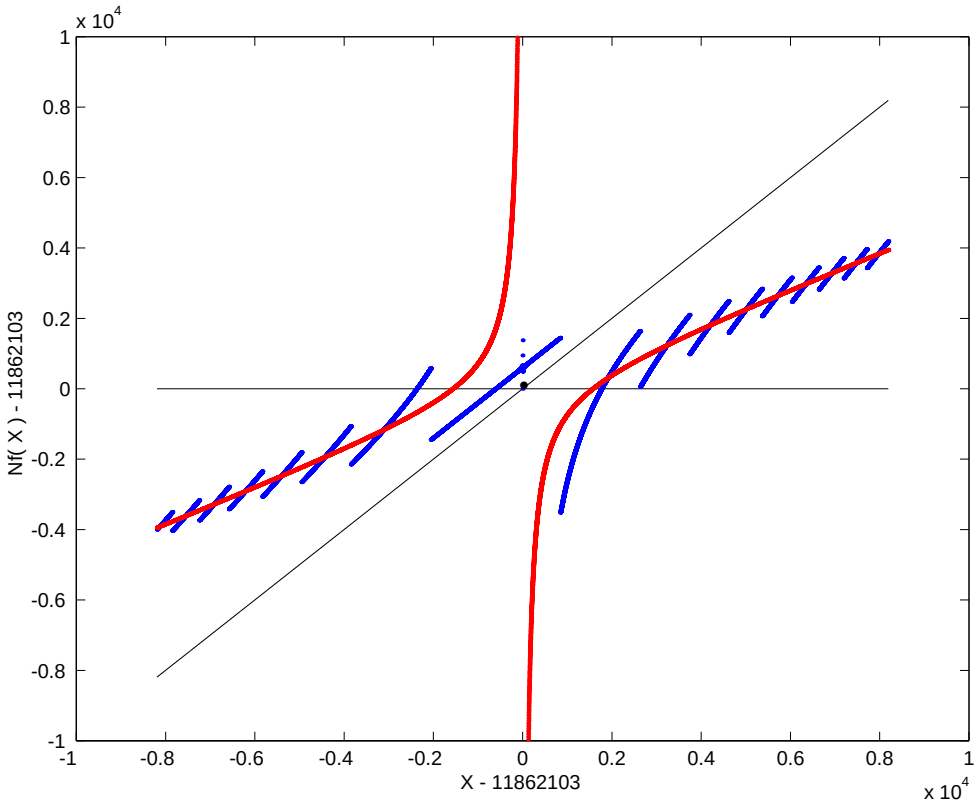
In the foregoing discussions of strategies for root-finding software, the avoidance of dithering and other unpleasant consequences of roundoff has been a *desideratum* achieved by stopping an iteration before it can be deflected intolerably by roundoff. Such a stopping criterion entails an error-analysis, either rigorous or approximate. "Perform an error-analysis!" is *Counsel of Perfection* (*cf. Matthew* 19:21 and other early ecclesiastical writings) impractical for most users of numerical software. Instead they are likely to run a root-finder until it stops with a result of unknowable accuracy as good as unknown roundoff has allowed. Therefore, besides the short list of inputs recommended above for a good root-finder, it must manage brackets and straddles well enough to cope not much slower than Binary Chop with the raggedness of roundoff. Here is a simple example that arose in one of my own computations; it was not contrived.

Example 6.5: Let cubic polynomial $f(x) := ((x - b)\cdot x + g)\cdot x + h$ for coefficients $b := 23722988$, $g := 16770435\cdot 2^{23}$, $h := 9968105\cdot 2^{34}$, all represented exactly in the 24-sig.bit floating-point arithmetic that will be used for all this example's computations. The computed value of $f(z)$ vanishes at $z := 11862103$; and the computed value of $f(11862945)\cdot f(11862946) < 0$. The jagged graph below exhibits values of $f(x)$ computed at 16385 consecutive 24-sig.bit floating-point integers x centered around z. The smooth nearly parabolic graph exhibits $f(x)$ exactly.



As secant iterations converge to one of the real "roots" that $f(x)$ should not have, two closely spaced iterates may send a third far to the right unless inhibited by a bracket or straddle. How should such an inhibition be implemented? An unlikely straddle, if available, can be Binary Chopped; this is what Wilkins & Gu [2003] recommend after any five iterations fail to halve the straddle's width or a new iterate fails to halve the previous sample of $|f(x)|$. More likely is a bracket that does not straddle; it will require a treatment more complicated than Binary Chop.

Usually roundoff degrades Newton's iteration less than secant iteration. The jagged graph below exhibits $Nf(x) := x - f(x)/f'(x)$ computed in 24-sig.bit floating-point at the same 16385 values x as before. The smooth nearly hyperbolic graph exhibits Nf(x) uncontaminated by roundoff.



When iteration starts from the far right, accelerated by the procedure mentioned after Corollary 10.4, brackets soon turn into straddles that are Binary Chopped to inhibit iterates that converge almost always to the 24-sig.bit adjacent pair [11862945, 11862946] . Actually the cubic f(x) has only one real zero $z \approx -1217.051909940\ldots$ found quickly if iteration starts from the left.

§7. Local Behavior of Newton's and Secant Iterations

The two best–known iterations for solving a given equation $f(z) = 0$ come from approximations to the graph of f by linear graphs, one a tangent and the other a secant. They have the following iterating functions:

$N(x) := x - f(x)/f'(x)$ for Newton's iteration $x_{n+1} := N(x_n)$, and

$S(x, y) := x - f(x)/f^\dagger(x, y) = S(y, x)$ for Secant iteration $x_{n+1} := S(x_n, x_{n-1})$.

See the Appendix on Divided Differences for an explanation of the *first Divided Difference*

$$f^\dagger(x, y) := \begin{cases} (f(x) - f(y))/(x - y) & \text{if } y \neq x, \\ f'(x) & \text{if } y = x. \end{cases}$$

Programmers can handle 0/0 in these formulas by stopping both iterations as soon as $f(x_n) = 0$, and otherwise by perturbing x_n slightly whenever $x_n = x_{n-1}$ during Secant iteration. For a mathematician the limiting value $S(x, x) = N(x)$ is the obvious expedient. Not so obvious is how to redefine $N(z)$ when $f'(z) = f(z) = 0$ because then $N(x)$ might oscillate too wildly to approach a limit as x approaches z, as happens for the example $f(x) := \int_0^x t \sin^2(1/t)\, dt$. None the less, redefining $N(z) := z$ whenever $f(z) = 0$ can be justified by the next lemma:

> **Lemma 7.1:** Suppose f' is finite throughout some neighborhood of a zero z of f, and $N(x)$ approaches a limit as $x \to z$. Then $N(x) \to z$; therefore defining $N(z) := z$ conserves the continuity of $N(x)$ near z whenever possible.

Proof: If necessary, shrink the neighborhood around z to exclude any other point at which N is undefined or infinite; then this neighborhood excludes every zero of f' except perhaps z, and by Rolle's theorem excludes also every zero of f other than z. Consequently the derivative $(\ln|f(x)|)' = f'(x)/f(x) = 1/(x-N(x))$ must be finite throughout this neighborhood except at $x = z$. Therefore $\ln|f|$ is eligible for an application of the Mean Value Theorem of the Differential Calculus to its first divided difference: for any distinct v and w on the same side of z in this neighborhood, some x between v and w must satisfy

$$\ln(f(v)/f(w))/(v-w) = (\ln|f(v)| - \ln|f(w)|)/(v - w) = f'(x)/f(x) = 1/(x - N(x)) .$$

Now suppose for the sake of argument that $N(x) \to L \neq z$ as $x \to z$; we shall infer a contradiction: For all distinct v and w close enough to z (and much closer to z than L is), but not separated by z, we would find $\ln(f(v)/f(w))/(v-w) = 1/(x - N(x)) \approx 1/(z-L)$ at some x between v and w. The last approximation could be kept as close as we please by keeping v and w close enough to z. But then, by fixing one of v and w and letting the other tend to z, we would infer that $\ln|f(z)|$ is finite, so z could not be a zero of f. But it is; therefore $L = z$. END OF PROOF.

Now that $N(x)$ and $S(x, y)$ are defined properly, and practically always continuous around the zero z, we turn to their local convergence properties. Their convergence to a simple zero z is typified by their behavior when $f(x) = (x-z)/(x-\hat{o}) \neq 1$; for this example $N(x) = z + (x-z)^2/(\hat{o}-z)$ and $S(x,y) = z + (x-z)(y-z)/(\hat{o}-z)$. Simple computations confirm first that Newton's iteration $(x_{n+1}-z)/(\hat{o}-z) = ((x_n-z)/(\hat{o}-z))^2$ converges quadratically to z from every x_0 closer to z than the pole $\hat{o}$ is, and second that Secant iteration $(x_{n+1}-z)/(\hat{o}-z) = ((x_n-z)/(\hat{o}-z))\,((x_{n-1}-z)/(\hat{o}-z))$ converges at order $(1+\sqrt{5})/2$ to z from a wider range of starting iterates x_0 and x_1 satisfying

$|x_0-z|^{3-\sqrt{5}}\cdot|x_1-z|^{\sqrt{5}-1} < (\hat{o}-z)^2$. Orders of convergence different from these are uncommon for the functions f typically encountered in practice, as we shall see.

Typical or not, these iterations' local convergence to a zero z depends upon how f behaves in the neighborhood of z . What kind of behavior guarantees convergence? The graph of f has to resemble its tangents or secants closely enough in the sense that fluctuations in the derivative f' have to stay sufficiently small compared with f' . How small is " sufficiently small " ? It's not obvious yet. The first hypotheses that come to mind do not suffice:

> **Non–Theorem 7.2:** Suppose f'(x) and N(x) := x - f(x)/f'(x) are continuous at every x in some open neighborhood Ω of a zero z of f . Then it seems at least plausible that Newton's iteration $x_{n+1} := N(x_n)$ should converge to z from every initial x_0 in Ω close enough to z ; but it ain't necessarily so if f'(z) = 0 .

Counter–Example 7.2: A function f(x) will be contrived with these properties: f'(x) and N(x) are continuous everywhere, $f'(x) > 0$ for all $x \neq 0$, and $z = f(z) = f'(z) = N(z) = 0$. However, around z every open neighborhood Ω , no matter how small, contains infinitely many closed subintervals all of positive width from each of which Newton's iteration tends to two-cycle, jumping back and forth across z forever instead of converging to z .

The construction of this perverse f begins with an integer-valued step–function

$$k(x) := \text{IntegerNearest}(-\ln(|x|)/\ln(2)) = \text{IntegerNearest}(-\log_2(|x|)) ,$$

and a quartic polynomial

$$q(x) := 1+x + (13 + 9\sqrt{2})(x-1)^3 + (1 + 3/\sqrt{8})(x-1)^4$$

monotone increasing over $1/\sqrt{2} \le x \le \sqrt{2}$. This q meets the following specifications:

$q(1) = 2$, $q'(1) = 1$, $q''(1) = 0$, $q(\sqrt{2}) = 15/\sqrt{8} - 1 = 4q(1/\sqrt{2})$, $q'(\sqrt{2}) = 12 + \sqrt{8} = 2q'(1/\sqrt{2})$.

Note that $1/\sqrt{2} \le 2^{k(x)}|x| \le \sqrt{2}$; note too that k(x) is ambiguous when $\log_2|x|$ is a half–integer, but then either choice $k = -\log_2|x| \pm 1/2$ is acceptable. Finally define f(0) := f'(0) := 0 and

$$f(x) := \text{sign}(x)\, q(2^{k(x)}|x|)/4^{k(x)} \quad \text{for} \quad x \neq 0 .$$

The continuity of f(x) and of $f'(x) = q'(2^{k(x)}|x|)/2^{k(x)}$ are easily confirmed along with the identities $f(x) = -f(-x) = f(2^{k(x)}|x|)/4^{k(x)}$ and $f'(x) = f'(-x) = f'(2^{k(x)}|x|)/2^{k(x)} > 0$ for $x \neq 0$. The ranges of values taken by $|f(x)|/x^2$ and by $f'(x)/|x|$ over all $x \neq 0$ are the same respectively as the ranges of $q(x)/x^2$ and $q'(x)/x$ over $1/\sqrt{2} \le x \le \sqrt{2}$, so as $x \to 0$ we find $|f(x)| \le 3x^2$ and $f'(x) \le 12|x|$, confirming continuity at x = 0 . And $N(x) = x - f(x)/f'(x) = N(2^{k(x)}|x|)/2^{k(x)}$ is continuous there too because $|N(x)| \le 2|x|$ similarly.

The design of f(x) ensures that N(x) = -x and N'(x) = 0 whenever $x = \pm 2^k$ for every integer k ; moreover $2^k\cdot 0.9935 < |N(x)| < 2^k\cdot 1.0064$ whenever $2^k\cdot 0.9935 < |x| < 2^k\cdot 1.0064$, so from any x_0 in those intervals Newton's iteration tends rapidly to a two-cycle $+2^k \leftrightarrow -2^k$, as claimed. Numerical experiments suggest that such a two–cycle, though with a large negative k , is the likeliest outcome of iteration from a randomly chosen x_0 . END OF COUNTER–EXAMPLE.

In general, the convergence of Newton's and Secant iterations cannot be taken for granted. Their local convergence depends upon whether, as $x \to z$ and $y \to z$, the limiting values of certain first divided differences like

$$N^\dagger(x,z) := (N(x) - N(z))/(x-z) = (N(x) - z)/(x-z) \to N'(z) \quad \text{and}$$
$$S^\dagger(\{x,z\},y) := (S(x,y) - S(z,y))/(x-z) = (S(x,y) - z)/(x-z) \to \partial S(x, y)/\partial x \,|_{x=y=z}$$

exist and are small enough. In particular, convergence is superlinear if these derivatives vanish, because then $|x_{n+1} - z|/|x_n - z| \to 0$ as the iterations converge; also the Order of convergence depends then upon whether limiting values exist for certain second divided differences

$$N^{\dagger\dagger}(x,z,z) := (N^\dagger(x,z) - N'(z))/(x-z) = (N(x) - z)/(x-z)^2 \quad \text{and}$$
$$S^{\dagger\dagger}(\{x,z\},\{y,z\}) := (S^\dagger(\{x,z\},y) - S^\dagger(\{x,z\},z))/(y-z) = (S(x,y) - z)/((x-z)(y-z))$$

from which bounds for quotients $|x_{n+1} - z|/|x_n - z|^2$ and $|x_{n+1} - z|/|(x_n - z)(x_{n-1} - z)|$ respectively can be obtained. Such bounds will be obtained from first and second derivatives and divided differences of f by invoking recondite identities like …

> **Identities 7.3:** $f(S(u, w)) \equiv (S(u, w) - u)\,(S(u, w) - w)\, f^{\dagger\dagger}(S(u, w), u, w)$. This includes the limiting case $f(N(v)) \equiv (N(v) - v)^2\, f^{\dagger\dagger}(N(v), v, v)$. Taking $f(z) = 0$ into account yields the identity $(S(x,y) - z)/((x-z)(y-z)) \equiv f^{\dagger\dagger}(x,y,z)/f^\dagger(x,y)$ and its limiting case $(N(x) - z)/(x-z)^2 \equiv f^{\dagger\dagger}(x,x,z)/f'(x)$.

The identities' proofs are entirely mechanical and left to readers who have reviewed the notation and formulas in the first two pages of the Appendix on Divided Differences.

Conditions sufficient locally for convergence have been found in two ancient theorems of which at least one applies in almost all practical situations. The first theorem is as old as Taylor series:

> **Theorem 7.4:** Suppose f' is continuous throughout some neighborhood Ω of a zero z of f at which $f'(z) \neq 0$. Then $N'(z) = 0$; therefore Newton's iteration converges superlinearly to z from every initial x_0 close enough to z. Similarly Secant iteration converges superlinearly to z from every initial x_0 and x_1 close enough to z. If f" exists and is bounded throughout Ω then $N''(z) = f''(z)/f'(z)$ and the convergence of Newton's iteration is at least quadratic (Order = 2), and the convergence of Secant iteration has Order at least $(1 + \sqrt{5})/2 = 1.618\ldots$.

Proof: As $u \to z$ and $w \to z$ independently the continuity of f' carries $f^\dagger(u, w) \to f'(z)$. Consequently $(N(x) - z)/(x-z) = (f'(x) - f^\dagger(x, z))/f'(x) \to 0/f'(z) = 0$ as $x \to z$ and so $N'(z) = 0$ as claimed, whence Newton's iteration converges superlinearly. Similar reasoning shows that $(S(x,y) - z)/(x-z) = (f^\dagger(x,y) - f^\dagger(x,z))/f^\dagger(x,y) \to 0$ as $x \to z$ and $y \to z$, so Secant iteration converges superlinearly too.

When f" exists and is bounded, some constant $C > |\frac{1}{2} f''(x)/f'(z)|$ throughout Ω. Therefore $(N(x) - z)/(x-z)^2 = f^{\dagger\dagger}(x,x,z)/f'(x)$ lies between $\pm C$ for all x close enough to z and therefore

$|(x_{n+1} - z)/(x_n - z)^2| < C$ if x_0 is close enough to z; convergence is at least quadratic as claimed. And $(S(x,y) - z)/((x-z)(y-z)) = f^{\dagger\dagger}(x,y,z)/f^{\dagger}(x,y)$ also lies between $\pm C$ for all x and y close enough to z, so $|(x_{n+1} - z)/((x_n - z)(x_{n-1} - z))| < C$ if x_0 and x_1 are close enough to z, thereby vindicating the claimed Order of convergence; here is an outline of how that works (*cf.* Ostrowski [1966], or Dahlquist *et al.* [1974], or Vianello & Zanello [1992].):

For that constant $C > |\frac{1}{2} f''(x)/f'(z)|$ throughout Ω let $D_n := -\ln|C (x_n - z)|$; then the Secant iteration's $|(x_{n+1} - z)/((x_n - z)(x_{n-1} - z))| < C$ means that $D_{n+1} > D_n + D_{n-1} > 0$ if x_0 and x_1 are close enough to z. Next, $D_{n+1} > F_n D_1 + F_{n-1} D_0$ by induction where the Fibonacci numbers $F_n = F_{n-1} + F_{n-2} = (Ç^{n+1} - (-Ç)^{-n-1})/(Ç + 1/Ç)$ for another constant $Ç := (1 + \sqrt{5})/2 = 1 + 1/Ç$. Thus D_n approaches $+\infty$ at least as fast as some multiple of $Ç^n$. END OF PROOF.

(Continuity of f' in Theorem 7.4 cannot be replaced by mere existence of f' and its consequent *Darboux Continuity* lest N oscillate violently for examples like $f(x) := \int_0^x \sin^2(1/t)dt$ whose $f(0) = 0$ and $f'(0) = 1/2$. In general a function, perhaps too wildly oscillatory to be continuous, is called "Darboux Continuous" if, among the values it takes on every closed subinterval of its domain, lie all values between those taken at that subinterval's ends. Every derivative has that property. For more about Darboux Continuity see Bruckner and Ceder [1965].)

The ultimate speeds of convergence of Newton's and Secant iteration should not be compared by considering only their orders of convergence. As many a textbook points out nowadays, the two iterations yield correct decimal digits ultimately at about the same rate if the computation of the derivative f' too adds about 44% to the time taken to compute f alone. If f' costs much more than that, Secant iteration goes faster in the likeliest cases. But Theorem 7.4 says nothing about the iterations' speeds when $f'(z) = 0$, in which case a different approach is needed.

> **Theorem 7.5:** Suppose $|f'(x)|$ increases as x moves away from z through some neighborhood Ω on one side of a zero z of f. Then $0 < (N(x) - z)/(x-z) < 1$ and so Newton's iteration converges monotonically to z from every initial x_0 in Ω. Similarly $0 < (S(x,y) - z)/(x-z) < 1$ for all x and y in Ω and so Secant iteration converges monotonically to z from every initial x_0 and x_1 in Ω.

In other words, this theorem's hypothesis is that the graph of $f(x)$ is convex towards the x–axis as is the case, for example, when $f''f > 0$ inside Ω. Theorems like this appear in many texts, for instance Ostrowski [1960 *et seq.*] ch. 9 and 10, and Dahlquist *et al.* [1974] p. 225. Texts written in France attribute theorems like this to Dandelin and/or Fourier, as if it had not been geometrically obvious before them. Let the reader compare the limpidity of his own proof–by–pictures with the turgidity that follows.

Proof: Regardless of whether $f'(z) = 0$, the growth of $|f'(x)|$ as x moves away from z implies that $f'(x)$ can't reverse sign, and therefore $0 < (f'(x) - f'(y))/f'(x) < 1$ at that y strictly between z and x where $f'(y) = f^{\dagger}(x,z)$. Therefore $0 < (N(x) - z)/(x-z) = (f'(x)-f^{\dagger}(x, z))/f'(x) < 1$ for every $x \neq z$ in Ω. This implies that the iteration $x_{n+1} := N(x_n)$ converges monotonically to a

limit between z and x_0 inclusive from every initial x_0 in Ω. Where is that limit? Since $f(x_n)/f'(x_n) = x_n - x_{n+1} \to 0$ and $|f'(x_n)| \le |f'(x_0)|$, so does $f(x_n) \to 0$, whence $x_n \to z$ as claimed. Similarly $(S(x,y) - z)/((x-z)(y-z)) = f^{\dagger\dagger}(x,y,z)/f^{\dagger}(x,y)$ for all x and y in Ω from one of Identities 7.3; what we do with this depends upon which of x and y lies closer to z. If x lies strictly between y and z then $(S(x,y) - z)/(x-z) = (f^{\dagger}(y,x) - f^{\dagger}(x,z))/f^{\dagger}(y,x)$; if y lies strictly between x and z then $(S(x,y) - z)/(x-z) = ((y-z)/(x-z))\,(f^{\dagger}(x,y) - f^{\dagger}(y,z))/f^{\dagger}(x,y)$. Either way the quotient in question lies strictly between 0 and 1, so the Secant iteration's $x_{n+1} := S(x_n, x_{n-1})$ converges monotonically to some limit between z and the closer of any two starting iterates x_0 and x_1 in Ω. Where is that limit? Since $f(x_n)/f^{\dagger}(x_n, x_{n-1}) = x_n - x_{n+1} \to 0$ and $|f^{\dagger}(x_n, x_{n-1})| \le |f'(x_0)|$, so does $f(x_n) \to 0$, whence $x_n \to z$ as claimed. END OF PROOF.

For practical purposes Theorems 7.4 and 7.5 tell us to expect Newton's and Secant iteration to converge *ultimately* superlinearly or monotonically or both if started close enough to z. Alas, the speed of convergence is not mentioned in Theorem 7.5, and for good reason; its convexity hypothesis is compatible with arbitrarily slow convergence. For example, when $f(x) = |x|^m$ for any constant $m > 1$, Newton's iteration yields $x_n = (1 - 1/m)^n x_0$ convergent arbitrarily slowly for m big enough; however $f(x_n)/f(x_0) = (1 - 1/m)^{mn} < e^{-n}$ tends to 0 quickly. When m is a negative constant tiny enough, $f(x_n)/f(x_0)$ tends to 0 arbitrarily slowly although x_n diverges to $z = \infty$ quickly. Both x_n and $f(x_n)$ converge arbitrarily slowly if m exceeds 1/2 by little enough, but then the convexity hypothesis is violated. What light do these examples shed upon the general case ? The case $m > 1$ turns out to be typical of what happens when the graph of $f(x)$ is convex towards the x-axis and x_n converges to a finite zero z of both f and f' :

> **Theorem 7.6:** Under the convexity hypothesis of Theorem 7.5, the iterates x_n may converge to z arbitrarily slowly, though monotonically; but $f(x_n)$ tends monotonically to 0 at least so fast that $\sum_n (2^n f(x_n))^2 \le f(x_0)^2 (x_0 - z)/(x_0 - x_1)$.

(The " 2 " in " $2^n f(x_n)$ " cannot be replaced by a bigger constant since $f(x_{n-1})/f(x_n) \to 2$ when Secant iteration is applied to the example $f(x) := x \exp(-1/x)$ with $x_0 > x_1 > 0$. An example $f(x)$ that justifies " 2 " for Newton's iteration is too complicated to be worth reproducing here though " e " can be used instead of " 2 " for all infinitely differentiable examples f .)

Proof: For definiteness restrict attention to nonnegative functions $f(x)$ and $f'(x)$ increasing over an interval $z \le x \le x_0 > z$, and for Secant iteration suppose too that x_1 lies inside that interval. Theorem 7.5 implies $z < x_{n+1} < x_n$, $0 = f(z) < f(x_{n+1}) < f(x_n)$ and $0 \le f'(z) < f'(x_{n+1}) < f'(x_n)$ without constraining the rapidity with which $x_n \to z$. Given any such sequence x_n convergent monotonically downwards to z, no matter how slowly convergent, do convex functions $f(x)$ exist from which Newton's or Secant iteration would have generated that sequence of iterates? To answer this question, a sequence of values f_n and f'_n will be derived from x_n, and then a continuously once differentiable convex function $f(x)$ satisfying $f(x_n) = f_n$ and $f'(x_n) = f'_n$ will

be constructed out of parabolic arcs; this is a function from which Newton's or Secant iteration generates the given sequence of iterates x_n .

Consider Secant iteration first because it is easier. The values f_n will have to satisfy $x_{n+1} = x_n - (x_n - x_{n-1})f_n/(f_n - f_{n-1})$, which fixes $f_n := f_{n-1}\,(x_n - x_{n+1})/(x_{n-1} - x_{n+1})$ recursively for n = 1, 2, 3, ... starting from any arbitrarily chosen $f_0 > 0$. Since $x_0 > x_1 > x_2 > \ldots > x_n > x_{n+1}$, also $f_0 > f_1 > f_2 > \ldots > f_n > f_{n+1} > 0$, and less obviously
$0 < (f_n - f_{n+1})/(x_n - x_{n+1}) = ((x_n - x_{n+1})/(x_n - x_{n+2}))((f_{n-1} - f_n)/(x_{n-1} - x_n)) < (f_{n-1} - f_n)/(x_{n-1} - x_n)$.
Therefore leeway exists to choose a positive descending sequence of values f'_n satisfying $(f_n - f_{n+1})/(x_n - x_{n+1}) < f'_n < (f_{n-1} - f_n)/(x_{n-1} - x_n) < f'_{n-1}$ for n = 1, 2, 3, ... After choices for all values f_n and f'_n have been assigned, f(x) is defined in each subinterval $x_n \le x \le x_{n-1}$ as the function whose graph is a convex parabolic arc subject to the constraints $f(x_n) = f_n < f(x_{n-1}) = f_{n-1}$ and $f'(x_n) = f'_n < f'(x_{n-1}) = f'_{n-1}$. The existence of this parabola (its axis need not be vertical) is the gist of Lemma A4.1 in Appendix A4: Parabolas. The triangle QRS in that lemma has Q at (x_n, f_n) , R at (x_{n-1}, f_{n-1}) , and sides QS and RS with slopes f'_n and f'_{n-1} respectively. The arc lies inside the triangle and joins Q to R . Taken together, all such arcs make up the graph of a function f(x) over the interval $z < x \le x_0$. This f(x) is convex and continuously once (but not likely twice) differentiable. What remains to be proved is that this $f(x) \to 0$ as $x \to z$; it will be proved later.

A different f(x) is needed for Newton's iteration, whose descending iterates x_n determine all quotients $f_n/f'_n = x_n - x_{n+1} > 0$ but leave the values f_n and f'_n partially arbitrary. Let us choose any positive f_0 and any positive $f_n < f_{n-1}(x_n - x_{n+1})/(x_{n-1} - x_{n+1})$ recursively for n = 1, 2, 3, ... , thereby determining also $f'_n := f_n/(x_n - x_{n+1})$. Obviously $0 < f_n < f_{n-1}$; less obviously

$$\begin{aligned} 0 < (f_n - f_{n+1})/(x_n - x_{n+1}) &= (1 - f_{n+1}/f_n)\, f'_n \\ &< f'_n = f_n/(x_n - x_{n+1}) \\ &< (f_{n-1} - f_n)/(x_{n-1} - x_n) \quad . \end{aligned}$$

Next define f(x) in each subinterval $x_n \le x \le x_{n-1}$ to be the function whose graph is a convex parabolic arc subject to the constraints $f(x_n) = f_n < f(x_{n-1}) = f_{n-1}$ and $f'(x_n) = f'_n < f'(x_{n-1}) = f'_{n-1}$ as before. Once again, all such arcs make up the graph of a convex and continuously once (but not likely twice) differentiable function f(x) over the interval $z < x \le x_0$. What remains to be proved is that this $f(x) \to 0$ as $x \to z$.

What remains to be proved, not just for the functions f constructed above but for every f that satisfies the theorem's convexity hypothesis, is that the values $f(x_n)$ tend to f(z) = 0 faster than the terms of a geometric progression with common ratio 1/2 . Attention is still restricted to nonnegative functions f(x) and f'(x) increasing over the finite interval $z \le x \le x_0 > z$; and for Secant iteration x_1 lies inside that interval. Now the abbreviations $f_n = f(x_n)$ and $f'_n = f'(x_n)$ stand for values computed during the iteration and, because $z < x_n < x_{n-1}$, they satisfy both $0 < f_n < f_{n-1}$ and $0 < f'_n < (f_{n-1} - f_n)/(x_{n-1} - x_n) < f'_{n-1}$, the latter because f' is increasing.

Consider Secant iteration $x_{n+1} = x_n - (x_n - x_{n-1})f_n/(f_n - f_{n-1})$ first because it is easier. It has $2f_n/f_{n-1} = 2/(1 + (x_{n-1} - x_n)/(x_n - x_{n+1})) \leq \sqrt{((x_n - x_{n+1})/(x_{n-1} - x_n))}$. Next, Newton's iteration $x_{n+1} = x_n - f_n/f'_n$ has $f_n/(x_n - x_{n+1}) = f'_n < (f_{n-1} - f_n)/(x_{n-1} - x_n)$, from which follows again $2f_n/f_{n-1} < 2/(1 + (x_{n-1} - x_n)/(x_n - x_{n+1})) \leq \sqrt{((x_n - x_{n+1})/(x_{n-1} - x_n))}$. For both iterations, repeated multiplication implies $(2^n f_n/f_0)^2 \leq (x_n - x_{n+1})/(x_0 - x_1)$; now sum over n. END OF PROOF.

Theorems 7.4, 7.5 and 7.6 are best regarded as contributions to local convergence theory since they say too little about convergence from afar. Monotonic convergence is what disqualifies the global pretensions of the latter two although their convexity hypothesis might hold in a wide neighborhood Ω. More often the first few (if not all) iterates of a convergent iteration approach z non–monotonically, in which cases the convexity hypothesis can hold in at most a bounded domain. Therefore theorems 7.5 and 7.6, unable to discriminate between non–monotonic convergence and interminable meandering, are too often applicable only locally.

For example, if f is a cubic polynomial monotonic over a non–finite (including $+\infty$ or $-\infty$, or both) interval Ω but not convex thereon, the iterations cannot meander in Ω but will either escape from it or converge to a zero of f therein; this follows from Theorem 8.2 below, not from theorems 7.5 and 7.6 above. On the other hand, if f is a quintic polynomial monotonic over a non–finite interval Ω but not convex thereon, Newton's iteration can meander in Ω forever; $f(x) := 5x^5 - 18x^3 + 45x$ is an instance with all the real axis for Ω and with $f' \geq 15.84$, but alternate iterates x_n approach +1 and -1 if ever $1 \leq |x_n| < 1.076570927$.

What distinguishes monotonic cubics from other monotonic polynomials? The distinction will become clear later when we deduce Theorem 8.2 from hypotheses that are the weakest and thus most widely applicable conditions now known to suffice for convergence.

§8. Sum–Topped Functions

A function *Sum–Topped* on an interval Ω is by definition a function $q(x)$ that lies between 0 and $q(u) + q(w)$ inclusive throughout *every* closed subinterval $u \le x \le w$ of Ω. The label "Sum–Topped" has been born out of desperation for lack of a better label. Also lacking is a neat characterization of sum–topped functions. Some of their properties are obvious; for instance, functions sum–topped on an interval Ω are also sum–topped on every subinterval of Ω, but not *vice versa*. If q is sum–topped on Ω so is μq for every real constant μ, positive or negative. Monotonic functions that do not reverse sign are sum–topped; and non–monotonic sum–topped functions exist too. Here are some examples (plot them!) to illustrate their diversity:

Any quadratic q on an interval none of which falls between two simple zeros of q ;

$3 + \cos(e^x)$ on the whole real x-axis ;

$|x - \sin(x)|$ on the whole real x-axis ;

$1/(1 + x^2)$ on the interval $-1/w \le x \le w$ for any $w > 0$; and

$2\cos(x) + x - \mu$ on the positive real x-axis for any constant $\mu \le 2 + 2\pi/3 - 2\sqrt{3} = 0.63\ldots$.

Some properties of sum–topped functions are almost obvious:

> **Lemma 8.1:** A function q sum–topped on Ω cannot reverse sign (by taking both positive and negative values) therein; and if $q(z) = 0$ at some z in Ω then $|q(x)|$ is a non–decreasing function of $|x-z|$ while x is in Ω.

Proof: If $q(u)\,q(w) \le 0$ for some u and w in Ω then, since $(q(u) + q(w) - q(x))\cdot q(x) \ge 0$ for all x between u and w inclusive, setting $x = w$ implies that $q(u)\,q(w) = 0$. And if $q(z) = 0$ at some z in Ω then, because $q(y)$ must lie between 0 and $q(z) + q(x) = q(x)$ for all y between z and x, we infer that $0 \le q(y)/q(x) \le 1$ if $q(x) \ne 0$. Therefore $q(x)$ may vanish throughout some closed subinterval of Ω but must then become nonzero and monotonic as x departs from that subinterval. END OF PROOF.

In the light of this lemma, the unobviously sum–topped functions q on Ω are the non–monotonic ones that retain the same nonzero sign throughout; suppose $q > 0$ to simplify the following exposition. Whether a continuous non–monotonic q is sum–topped is determined solely by the values it achieves at its local extrema (maxima and minima) in Ω. Suppose all its local minima are $q_j := q(v_j) > 0$ and all its local maxima strictly inside Ω are $Q_i := q(m_i)$ for $v_0 < m_1 < v_1 < m_2 < \ldots < m_K < v_K$ all in Ω. Then q is sum–topped if and only if every $Q_i \le \min_{j<i} q_j + \min_{j \ge i} q_j$. This decision procedure can be inconvenient, as it is for large K, and gets worse when q has infinitely many extrema or is discontinuous. Among sum–topped functions the easiest to recognize are those of *Restrained Variation,* which are explained below in appendix A2. Before digressing to that explanation, let us see how sum–topped functions figure in Newton's and Secant iteration:

> **Theorem 8.2: A Sum-Topped Derivative**
> Suppose f' is continuous and sum–topped throughout a closed interval Ω. Then Newton's iteration $x_{n+1} := x_n - f(x_n)/f'(x_n)$, started from any x_0 in Ω, either converges in Ω to the zero z of f or leaves Ω; the iteration cannot meander in Ω endlessly.

Proof: At the cost perhaps of replacing f by -f , we may assume that $f' \geq 0$ throughout Ω since Lemma 8.1 prevents f' from reversing sign. Newton's iterating function $N(x) := x - f(x)/f'(x)$ is therefore continuous except possibly where $f'(x)$ vanishes. This possibility must be dispatched first; later we shall deal with cases in which f' never vanishes in Ω .

The theorem's hypotheses allow f and/or f' to vanish at most once in Ω . To see why, observe first that f' must vanish between any two distinct zeros of f . Next suppose $f'(u) = 0$. Then Lemma 8.1 implies that there must be some closed subinterval $ù \leq x \leq ú$ of Ω throughout which $f'(x) = 0$ and $f(ù) = f(x) = f(ú)$ while $ù \leq x \leq ú$; but

f'(x) is positive and decreasing, and $f(x) < f(ù)$, while $x < ù$ in Ω ; and

f'(x) is positive and increasing, and $f(x) > f(ú)$, while $x > ú$ in Ω .

Only in the subinterval (it may be a single point u) need N be redefined:

•° Wherever $f'(z) = 0 = f(z)$ define $N(z) := z$.

•' Wherever $f'(u) = 0 \neq f(u)$ define $N(u) := -\text{sign}(f(u)) \infty$ as if $f'(u) = +0$.

At most one of these two cases can arise. In the first case (•°), Theorem 7.5 guarantees the convergence of Newton's iteration to an endpoint of the subinterval of Ω wherein $f(z) = 0$. The same theorem dispatches the second case (•') too because, so long as $f(x_n)$ has the same nonzero sign as f(u) , iteration must move monotonically in the direction that decreases |f| until one of the following three eventualities occurs:

i) An iterate escapes from Ω , perhaps by jumping to $\pm\infty$, or else

ii) Iterates stay in Ω and "converge" monotonically to $+\infty$ or $-\infty$ in Ω , or else

iii) An $f(x_n)$ reverses sign and subsequent iterates reverse course and converge to z .

Only eventuality (iii) delivers a finite root z of $f(z) = 0$ in Ω , and $f'(z) \neq 0$ there. Whether eventuality (ii) delivers a root depends upon whether the limit to which |f(x)| declines, as x approaches that infinite endpoint of Ω at which f' vanishes, is zero.

Eventuality (i) must arise also when neither f nor f' vanishes in Ω since then too the iteration must move monotonically in a direction that decreases |f| .

Now only one case is left to consider: Suppose henceforth that $f' > 0$ throughout Ω and $f(z) = 0 < f'(z)$ at some z in Ω . Now N must be continuous in Ω and its sole fixed–point therein is $z = N(z)$. If finitely many iterates lie on one side of z and infinitely many on the other side in Ω , then the iteration must converge ultimately monotonically because, except for finitely many initial iterates, every subsequent iteration with $x_n \neq z$ maintains $0 \leq (x_{n+1} - z)/(x_n - z) < 1$ and $0 \leq f(x_{n+1})/f(x_n) < 1$, as is easily confirmed; of course the iteration converges to z . But if the iteration neither escaped from Ω nor converged to z , as we shall assume henceforth for the sake of argument by contradiction, infinitely many iterates would have to fall on both sides of z , which would have to lie strictly inside Ω . We shall complete the proof of theorem 8.2 by demonstrating that its hypotheses are not consistent with the last assumption.

By virtue of Theorem 7.4, the iterates could not come arbitrarily close to z ; they would all have to stay at least some positive distance away from z . Let u and w be the iteration's points of accumulation nearest z on both sides; say $u < z < w$. Then every open neighborhood of u would contain infinitely many iterates, as would every open neighborhood of w , but any closed interval strictly between u and w could contain at most finitely many iterates. Since N(u)

would be another point of accumulation and $N(u) > u$, we would find $N(u) \geq w$ too. Similarly for $N(w) \leq u$. Let's scrutinize the last two inequalities; they would imply respectively that

$$0 < (w-u)f'(u) \leq -f(u) \quad \text{and} \quad 0 < (w-u)f'(w) \leq f(w) \ .$$

Adding them would produce

$$(w-u)(f'(u) + f'(w)) \ \leq \ f(w) - f(u) \ = \ \int_u^w f'(x)dx$$

which simplifies to

$$0 \geq \ \int_u^w (f'(u) + f'(w) - f'(x))dx \ .$$

But the theorem's hypotheses force the integrand to be nonnegative and continuous, so it would have to vanish at every x between u and w inclusive, which would force $f'(u) = f'(w) = 0$ contrary to the supposition $f' > 0$ made when this case began to be considered. END OF PROOF.

In showing that N cannot swap distinct points u and w of Ω, the foregoing proof resembles an application of Sharkovsky's No-Swap Theorem, but the resemblance is superficial for two reasons. First, the theorem's hypotheses merely suffice for its conclusion; they are not necessary. Second, N was not required to map Ω to itself; determining whether such a requirement has been fulfilled can be harder than solving the given equation $f(z) = 0$. An easier expedient is to incorporate whatever may be known about f and Ω into a bracketing procedure that decides whether an excursion out of Ω should stop the iteration or be returned to Ω. After that the only hazard to prevent is the possibility that, left alone, the iteration may meander in Ω forever. This hazard is precluded if f' is sum–topped but, as we have seen just before Theorem 8.2, deciding whether f' is sum–topped can be inconvenient. Fortunately, some oft–encountered sum–topped configurations are easy to recognize:

Corollary 8.3: A Weak Convexity Condition
Suppose $f = g-h$ is a differentiable difference between two convex functions, one non–decreasing and the other non-increasing, throughout a closed interval Ω. Then Newton's iteration $x_{n+1} := x_n - f(x_n)/f'(x_n)$, started from any x_0 in Ω, either converges in Ω to the zero z of f or leaves Ω; the iteration cannot meander in Ω endlessly.

Proof: See Corollary A2.3 in Appendix A2: *Functions of Restricted Variation* ; apparently f' is one of those, and therefore continuous and sum–topped over Ω. Therefore Theorem 8.2 applies. END OF PROOF.

Since f determines neither Ω nor the splitting $g-h = f$ uniquely, arbitraryness can complicate the application of Corollary 8.3. Take the (admittedly contrived) example $f(x) := \arctan(x)$, for which Newton's iteration converges to $z = 0$ from any x_0 strictly between the points ± 1.3917452 swapped by N, but diverges otherwise. These points cannot serve as endpoints for Ω in Corollary 8.3; indeed, no Ω that includes both points ± 1 in its interior can sustain a splitting $g-h = f$ satisfying the theorem's requirements because $f'(0)$ is too big for f' to satisfy the sum–topped condition

" $0 \leq f'(v)/(f'(u) + f'(w)) \leq 1$ whenever v lies between u and w both in Ω "

that every splittable f must satisfy. On the other hand, for every $L > 0$ the interval $\Omega := [-L, 1/L]$ sustains such a splitting thus:

$g(x) := \arctan(x) - x/(1 + L^2)$ for $-L \le x \le 0$,
$:= x/(1 + 1/L^2)$ for $0 \le x \le 1/L$; and
$h(x) := g(x) - \arctan(x)$.

But then N maps Ω to itself only when $0.86033359 \le L \le 1.1623398$. Otherwise the iteration may escape from Ω ; and after that it may come back and converge, or else diverge, according to whether it started between ± 1.3917452 or not.

In general, the weak conditions in Theorem 8.2 and Corollary 8.3 are not necessary for convergence but are at best sufficient. Their virtue is their ease of application compared with attempts to apply Sharkovsky's No–Swap theorem to N .

Example 8.4: Corollary 8.3 was discovered first, before Theorem 8.2, in 1976 while I was helping Dr. D.W. Harms and R.E. Martin to design a financial calculator (see Martin [1977]). The equation $f(z) = 0$ to be solved for a positive root

$z = 1 + (\text{interest rate})$ or $z = 1 - (\text{discount rate})$

was put into the theorem's partitioned form $f = g-h$ thus:

$$f(x) = (C_m x^m + ... + C_3 x^3 + C_2 x^2 + C_1 x) - (c_0 + c_1/x + c_2/x^2 + c_3/x^3 + ... + c_k/x^k)$$

with nonnegative coefficients $C_{...}$ and $c_{...}$ representing cash flows, perhaps investments and returns, or borrowings and repayments. Ω was the positive real axis and was mapped to itself by Newton's iterating function N for this f . However, because m and k could be huge (many thousands), a complicated initial guess x_0 had to be contrived to prevent instances of intolerably deferred convergence. The complexity of x_0 cast a shadow over the design's integrity.

R. Carone and I got rid of that complexity when we worked on the hp-12C financial calculator introduced in 1982 (and still selling over thirty years later). It solves a different but equivalent equation $f(z) = 0$ for its real root

$z = \ln(1 + (\text{interest rate}))$ or $z = \ln(1 - (\text{discount rate}))$.

The partitioned form $f = g-h$ required for Theorem 8.2 is obtained thus:

$$f(x) = \ln(C_m e^{mx} + ... + C_3 e^{3x} + C_2 e^{2x} + C_1 e^{x}) - \ln(c_0 + c_1 e^{-x} + c_2 e^{-2x} + c_3 e^{-3x} + ... + c_k e^{-kx})$$

with the same coefficients as before. The convexity of g and h is less obvious than before. Ω is all the real axis. Because this $f(x)$ is so nearly linear when $|x|$ is big, the iteration's dependence upon the initial guess x_0 has become so mild that a crude guess provably suffices. END EX. 8.4.

The hypotheses of Theorem 8.2 and Corollary 8.3 are the weakest global conditions known to be sufficient to prevent Newton's iteration from meandering forever. Their hypotheses suffice also to prevent Secant iteration from meandering, as we shall see in §9.

§9. The Projective Connection between Newton's and Secant Iterations

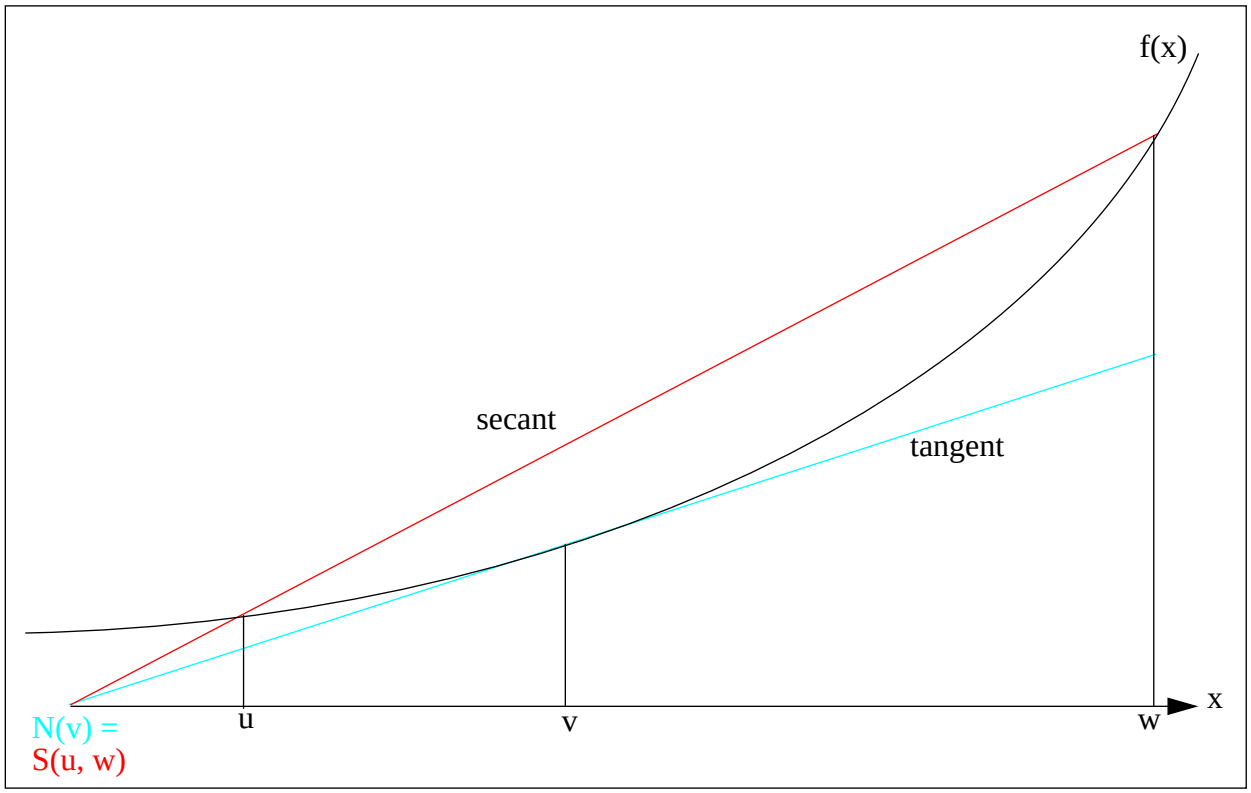
Before proving that Theorem 8.2 and Corollary 8.3 above apply as well to Secant as to Newton's iteration, we explore a connection that the reader may have anticipated: roughly, …

If Newton's iteration converges to a simple zero of f , so does Secant iteration.

This connection grows out of the straight lines, the tangents and secants, that figure in both iterations. The straightness of both kinds of lines is preserved by a family of *Projective Maps* of the plane to itself; consequently both iterations' convergence is invariant under these maps, as is the convexity hypothesis in Theorems 7.5 and 7.6 above. See Appendix A3: Projective Images, and especially Lemma A3.2, for details of which very few will figure directly in what follows.

Lemma 9.1: An Intermediate Value

If $S(u, w) := u - f(u)(u - w)/(f(u) - f(w))$ does not lie between u and w , *i.e.* if $f(u)f(w) > 0$, and if $f'(x)$ is finite throughout $u \le x \le w$, then at some v strictly between u and w either $N(v) := v - f(v)/f'(v) = S(u, w)$ or $f(v) = f'(v) = 0$.



Proof: There is a trivial case when $u = v = w$ and $S(u, w) := N(v)$. A different special case can arise with $f(u) = f(w) \neq 0$; in this case $S(u, w) = \infty = N(v)$ at some v strictly between u and w where Rolle's theorem implies $f'(v) = 0$. The lemma generalizes this special case. For finite $s := S(u, w)$ the proof is constructed from a projective map that preserves u and w but pushes s off to ∞ . Then, like scaffolding under a newly built bridge, the projective map is removed to leave only a slender proof standing.

Let $Ø(x) := f(x)/(s-x)$. Since s does not lie between u and $w > u$, $Ø(x)$ and $Ø'(x)$ are finite throughout $u \le x \le w$. And $Ø(u) = Ø(w)$ because of how s was defined, so Rolle's theorem implies $Ø'(v) = 0$ at some v strictly between u and w . $Ø'(v) = f'(v)/(s-v) + f(v)/(s-v)^2 = 0$ implies that this v is where either $N(v) = s$ or $f(v) = f'(v) = 0$. END OF PROOF.

Digression: Where between u and w may the lemma's v fall? In general v need not be unique. However, in the special case that sign(f"(x)) stays constant throughout $u < x < w$, the aforementioned projective map can be used to show that the equation $N(v) = S(u, w)$ has just one root v between u and w; and then the smaller the variation of $\log|f"(x)|$, the more closely can v be located. Its location is obtained from the first of Identities 7.3:

$$f(S(u,w)) \equiv (S(u,w) - u)\,(S(u,w) - w)\,f^{\dagger\dagger}(S(u,w), u, w) \quad \text{and} \quad f(N(v)) \equiv (N(v) - v)^2\,f^{\dagger\dagger}(N(v), v, v)\,.$$

When $f^{\dagger\dagger} \approx f"/2$ is nearly constant, so that f is nearly a quadratic polynomial, combining these identities with the equation $N(v) = S(u, w)$ of Lemma 9.1 implies that its root v is

$$v = S(u, w) + \sqrt{\big(}\,(u - S(u, w))\,(w - S(u, w))\,f^{\dagger\dagger}(S(u, w), u, w)/f^{\dagger\dagger}(N(v), v, v)\,\big)\,\mathrm{sign}(u - S(u, w))$$
$$\approx S(u, w) + \sqrt{\big(}\,(u - S(u, w))\,(w - S(u, w))\,\big)\,\mathrm{sign}(u - S(u, w))\,.$$

END OF DIGRESSION.

Lemma 9.1 joins Newton's iterating function N and the Secant's S by a bridge that breaks only over a zero of f across which f does not reverse sign; otherwise the bridge bears a big load:

> **Theorem 9.2:** Suppose f' and N are continuous throughout a closed finite interval Ω strictly inside which f does not vanish without reversing sign there too. If Newton's iteration converges in Ω from every initial x_0 in Ω, then it converges to the sole zero z of f in Ω, and Secant iteration also converges in Ω to z from every two starting points x_0 and x_1 in Ω.

That Newton's iteration always converges within Ω is an essential assumption independent of the others; see Non–Theorem 7.2 above. Unless z is an endpoint of Ω, the assumption that f reverses sign across its zero z is essential; otherwise two consecutive Secant iterates astride z could send a third to ∞. The assumption that N is continuous is essential too; otherwise, as Example A3.3 shows, the theorem's "converges" would have to be replaced by a complicated assertion about convergent subsequences of iterates like the one in my report [1979']. This theorem was discovered in 1977 in time to affect decisions made during the design of the root–finder behind the [SOLVE] key on Hewlett-Packard hand-held calculators beginning with the hp-34C described in my reprint [1979"]. The proof is long but, because it cannot now be found elsewhere, it is presented here despite its length.

Proof of Theorem: Because N maps Ω continuously into itself (otherwise Newton's iteration could escape from Ω) it must contain at least one fixed–point $z = N(z)$, which has to be a zero of f. This zero z cannot be a subinterval of Ω because f reverses sign at z. Another zero is ruled out by Rolle's theorem, which would imply a point between them where f' would vanish and N would jump out of Ω to ∞. In fact, f' cannot vanish in Ω except perhaps at z; elsewhere f is strictly monotonic in Ω. At the possible cost of replacing f by -f, we may assume that f is strictly increasing throughout Ω. Finally, N satisfies all four conditions that U satisfies in Sharkovsky's No–Swap Theorem 5.1 above. These conditions will figure at several places in the rest of the proof, which is presented below as a sequence of shorter propositions.

•Proposition 9.3: All Secant iterates $x_{n+1} := S(x_n, x_{n-1})$ stay in Ω.
This follows from Intermediate Value Lemma 9.1 above and the assumption that N stays in Ω.

•Proposition 9.4: For $n \geq 1$ we might as well assume that every $x_{n+1} \neq x_n \neq z$.
Otherwise nothing would be left to prove, because $x_{n+1} := S(x_n, x_{n-1}) = x_n$ if and only if $f(x_n) = 0$ and hence $x_{n+1} = x_n = z$. The possibility that $x_1 = x_0$ and $x_2 = N(x_1)$ is harmless.

•Proposition 9.5: If a subsequence of differences $x_{n+1} - x_n \to 0$ then, for every integer $k \geq 0$ fixed in advance, the corresponding subsequence $x_{n+k} \to z$.

Since divided difference $f^\dagger$ lies between the minimum and maximum values taken by derivative f' on Ω, the corresponding subsequence $f(x_n) = (x_{n+1} - x_n)\, f^\dagger(x_n, x_{n-1}) \to 0$, and therefore $x_n \to z$ since f is strictly increasing, and then $x_{n+1} \to z$ too. For each n in the subsequence, Intermediate Value Lemma 9.1 implies that some y_n exists between x_n and x_{n+1} satisfying either $x_{n+2} = y_n$ or $x_{n+2} = N(y_n)$; either way, the subsequence $x_{n+2} \to z$ too because N is continuous. Repeat as often as necessary to infer that $x_{n+k} \to z$. (Does the continuity of N then imply by itself that all $x_{n+k} \to z$ too no matter how k varies with n ?)

Definitions:

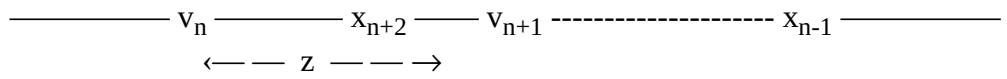
•A *Variance* is an iterate $v_n := S(x_{n-1}, x_{n-2})$ for which $f(x_{n-1})/f(v_n) < 0$, and then both z and $x_{n+1} := S(v_n, x_{n-1})$ must lie strictly between v_n and x_{n-1}.

•A *Permanence* is an iterate $p_n := S(x_{n-1}, x_{n-2})$ for which $f(x_{n-1})/f(p_n) > 1$, and then both z and $x_{n+1} := S(p_n, x_{n-1})$ must lie strictly on the side of p_n opposite from x_{n-1}.

•The *Wraith* of Permanence p_n is its nearest solution w_n of $N(w_n) = x_{n+1}$ strictly between p_n and x_{n-1}; the existence of w_n is assured by Intermediate Value Lemma 9.1.

•Proposition 9.6: For $n \geq 2$ every iterate $x_n := S(x_{n-1}, x_{n-2})$ is a Permanence or a Variance. The possibility that $0 < f(x_{n-1})/f(x_n) < 1$ is ruled out by the strictly increasing nature of f as follows: For the sake of argument suppose $0 < f(x_{n-1}) < f(x_n)$. This supposition would imply $z < x_{n-1} < x_n$, since f is increasing, and then $f^\dagger(x_{n-1}, x_{n-2}) = f(x_{n-1})/(x_{n-1} - x_n) < 0$, which is contradictory. The other impossibility $0 > f(x_{n-1}) > f(x_n)$ is dispatched similarly. Therefore every iterate x_n can be renamed either p_n or v_n.

•Proposition 9.7: If two consecutive iterates $v_n := S(x_{n-1}, x_{n-2})$ and $v_{n+1} := S(v_n, x_{n-1})$ are both Variances, then v_{n+1} lies strictly between v_n and x_{n-1}, and then both x_{n+2} and z lie strictly between v_{n+1} and v_n, and also $(v_n - x_{n-1})/(x_{n+2} - v_{n+1}) > 4$.



Only the last inequality requires unobvious confirmation. The definition of Variance implies that $f(x_{n-1})/f(v_n) < 0$ and $f(v_n)/f(v_{n+1}) < 0$, so $\text{sign}(f(v_{n+1})) = \text{sign}(f(x_{n-1}))$ and then, since f is monotonic, $f(x_{n-1})/f(v_{n+1}) > 1$ because v_{n+1} is closer to z than x_{n-1} is. Consequently

$$(v_n - x_{n-1})/(x_{n+2} - v_{n+1}) = \big(-(v_{n+1} - v_n)(f(v_n) - f(x_{n-1}))/f(v_n) \big) / \big(-f(v_{n+1})/f^\dagger(v_{n+1}, v_n) \big)$$
$$= (f(v_n) - f(x_{n-1}))(f(v_{n+1}) - f(v_n)) / \big(f(v_n) f(v_{n+1}) \big) \quad \ldots$$

$$= 1 - f(v_n)/f(v_{n+1}) - f(x_{n-1})/f(v_n) + f(x_{n-1})/f(v_{n+1})$$
$$\geq 1 + 2\sqrt{(f(x_{n-1})/f(v_{n+1}))} + f(x_{n-1})/f(v_{n+1}) > 4 \text{ as claimed.}$$

•Proposition 9.8: If, among the Secant iterates x_n, at most finitely many are Variances, or if at most finitely many are Permanences, then the iteration converges to z.

If all but the first finitely many iterates are Permanences they must converge monotonically in Ω to something, and it must be z by Proposition 9.5. If all but the first finitely many iterates are Variances then the subsequences $\{x_{2n}\}$ and $\{x_{2n+1}\}$ must ultimately converge monotonically in opposite directions with $|x_{2n} - x_{2n\pm1}| \to 0$ at least as fast as $1/4^n$, thanks to Proposition 9.7, so the iteration converges to z as claimed.

Henceforth only those sequences $\{x_n\}$ containing infinitely many Permanences and infinitely many Variances need be considered. Think of Permanences as punctuation marks separating strings of consecutive Variances. What matters most about such a string is whether its length is even or odd. Even lengths (including 0) will be treated first.

•Proposition 9.9: If a Permanence p_n is followed by an even number $2k \geq 0$ of consecutive Variances $v_{n+1}, v_{n+2}, \ldots, v_{n+2k}$ before the next Permanence p_{n+2k+1}, then the numbers

$$x_{n-1}, w_{n-1}, p_n, v_{n+2}, v_{n+4}, \ldots, v_{n+2k}, w_{n+2k+1}, p_{n+2k+1}, z, v_{n+2k-1}, \ldots, v_{n+3}, v_{n+1}$$

are exhibited here in strictly monotonic order (perhaps reversed).

If $2k = 0$ then $x_{n-1}, w_n, p_n, w_{n+1}$ and p_{n+1} lie on the same side of z. If $2k = 2$ only v_{n+1} lies on the side of z opposite the other four iterates and two Wraiths. For $2k \geq 2$ this proposition follows from Proposition 9.7.

•Proposition 9.10: If at most finitely many strings of Variances have odd lengths, the iterates x_n converge to z.

Discard as many of the earliest iterates as necessary, and renumber the rest, to obtain a sequence of iterates $x_{n+1} := S(x_n, x_{n-1})$ in which no string of Variances has odd length. Proposition 9.9 implies that the Permanences and their immediately antecedent iterates constitute a monotonic subsequence bounded by z. In other words, if the successive Permanences are $p_{n1}, p_{n2}, p_{n3}, \ldots$ then $x_{n1-1}, p_{n1}, x_{n2-1}, p_{n2}, x_{n3-1}, p_{n3}, \ldots, z$ are exhibited here in monotonic order, but perhaps not strictly so. This subsequence of iterates must converge and, by Proposition 9.5, it must converge to z. Recall now the Permanences' Wraiths; for $j = 1, 2, 3, \ldots$ each Wraith w_{nj} lies between x_{nj-1} and p_{nj} and satisfies $N(w_{nj}) = x_{nj+1}$. Evidently the Wraiths converge to z and, since N is continuous, so must the subsequence of iterates $x_{n1+1}, x_{n2+1}, x_{n3+1}, \ldots$. Among these lie all the initial Variances in strings of consecutive Variances, each string having nonzero even length. With the aid of Proposition 9.9 again we conclude that the Variances converge to z too.

(Were N not continuous, the Variances might not all converge to z; see Example A3.3.)

Only the possibility that infinitely many strings of Variances have odd lengths remains to be addressed to complete the proof of Theorem 9.2. For this purpose we introduce three more …

Definitions:

•A *Scout* s_n is a Permanence followed by a string of Variances of odd length.

•A *Guard* g_{n+1} is the first Variance following (in the sequence of iterates) a Scout s_n .

•A *Convoy* is the set of Wraiths belonging to the Permanences that come after (in the sequence of iterates) a Guard but not after its subsequent Scout.

For instance, if Scout s_n is followed by 2k+1 Variances $g_{n+1}, v_{n+2}, \ldots, v_{n+2k+1}$ followed by Permanence p_{n+2k+2} , then the numbers

$$x_{n-1},\ w_n,\ s_n,\ v_{n+2},\ \ldots,\ v_{n+2k},\quad z\ ,\quad p_{n+2k+2},\ w_{n+2k+2},\ v_{n+2k+1},\ \ldots,\ v_{n+3},\ g_{n+1}$$

appear here in monotonic order, according to the definitions of Permanences, Variances and Wraiths. (If 2k+1 = 1 then $v_{n+2}, \ldots, v_{n+2k}$ do not appear here.) Only the desired zero z and the Wraiths w_n and w_{n+1} are not iterates. That last Permanence p_{n+2k+2} might be a Scout too, or it might not. We have to confirm next that every Wraith belongs to a Convoy escorted by a Scout ranging ahead of it and a Guard bringing up the rear, and that alternate Convoys approach z from opposite sides.

•Proposition 9.11: In the sequence of iterates, suppose s_n and s_m are consecutive Scouts with $m > n$. Then $m \geq n+2$ and the numbers $x_{n-1},\ w_n,\ g_{m+1},\ z\ ,\ s_m,\ w_m,\ g_{n+1}$ appear here in monotonic order; and the Convoy of Wraiths w_j for $n < j \leq m$ lie numerically between Guard g_{n+1} and the next Scout s_m on the other side of which lie first z and then g_{m+1} and then w_n .

In the sequence of iterates, Scout s_n is followed by some odd number 2k+1 of Variances $g_{n+1}, v_{n+2}, \ldots, v_{n+2k+1}$ followed by Permanence p_{n+2k+2} followed perhaps by more strings of Variances of even lengths separated by Permanences up to the Permanence–and–Scout s_m followed by an odd number of Variances $g_{m+1}, \ldots$. How are all these numbers ordered numerically? It is easy to verify that

$$x_{n-1},\ w_n,\ s_n,\ v_{n+2},\ \ldots,\ v_{n+2k},\quad z\ ,\quad s_m,\ w_m,\ p_{n+2k+2},\ w_{n+2k+2},\ v_{n+2k+1}\ ,\ \ldots,\ g_{n+1}$$

appear here in monotonic order except that if $m = n+2k+2$ then p_{n+2k+2} and w_{n+2k+2} are redundant and should be dropped. If $m > n+2k+2$ then every string of Variances between (in the sequence of iterates) p_{n+2k+2} and s_m has even length, so Proposition 9.9 ensures that every Permanence after (in the sequence of iterates) g_{n+1} but not after s_m has its Wraith strictly between the Guard g_{n+1} and the Scout s_m of this Convoy of Wraiths all on the side of s_m opposite z . Moreover w_m is this Convoy's Wraith nearest z . The Guard g_{m+1} following Scout s_m falls somewhere on the other side of z ; where? Here Intermediate Value Lemma 9.1 combines with Sharkovsky's No–Swap Theorem 5.1 to explain why this new Guard g_{m+1} must come between z and the previous Convoy's Wraith w_n nearest z . If that were not so, if g_{m+1} fell on the side of w_n opposite z , then the numbers

$$g_{m+1} = N(w_m),\ w_n,\ \ z\ ,\ w_m,\ g_{n+1} = N(w_n)$$

would appear here in monotonic order and violate the No Crossover Condition that N must satisfy if Newton's iteration is to converge to z from every x_0 in Ω .

Thus has every claim in Proposition 9.11 been vindicated, and without saying which of g_{m+1} and s_n lies between the other and z; it is impossible to say, and does not matter. What matters is that alternate Convoys of Wraiths proceed monotonically towards z from opposite sides; on each side every Convoy is separated by its Guard from the preceding Convoy on the same side.

•Proposition 9.12: If the sequence of Secant iterates contains infinitely many Guards then the Secant iteration converges to the desired zero z.

Let $g_« \le z$ be the least upper bound for those Convoys and their Guards less than z; they constitute a subsequence of iterates and Wraiths converging monotonically upward to $g_«$. Similarly for greatest lower bound $g_» \ge z$. For every guard $g_n > g_»$ there is a Wraith $w_n < g_«$ for which $g_n = N(w_n)$; as the subsequence $w_n \to g_«-$ the corresponding subsequence $g_n = N(w_n) \to g_»+$ and so, because N is continuous, $g_» = N(g_«)$. Similarly $N(g_») = g_«$. Now the No Swap Condition satisfied by N implies that $g_» = g_« = z$. Then all the Wraiths must converge to z, pushing their Permanences (including the Scouts) ahead of them to converge to z also. Then Permanences and Guards squeeze the rest of the Variances to converge too.

Propositions 9.8, 9.10 and 9.12 leave no alternative but convergence for the Secant iteration and hence prove Theorem 9.2. END OF PROOF.

Note that Theorem 9.2 just proved has no converse; in many situations Secant iteration converges from all starting points but Newton's does not. $f(x) := 5x^5 - 18x^3 + 45x$ is a strongly monotonic ($f' > 15.84$) example for which Secant iteration always converges but Newton's iteration gets trapped when $1 \le |x_n| < 1.076570927$, as we have already seen after Theorem 7.6. Proposition 9.7 prevents Secant iteration from meandering in this example.

Another example is $f(x) := \arctan(x)$ discussed after Corollary 8.3, where we saw that Newton's iteration converges if started between ± 1.3917452 but diverges otherwise. Apparently Secant iteration converges if started anywhere in a wider interval between about ± 2.25, but can cycle on four points $x_{4n} = 4.75048222$, $x_{4n+1} = 1.12143673$, $x_{4n+2} = -x_{4n}$ and $x_{4n+3} = -x_{4n+1}$, and certainly diverges from starting points both greater than about 2.5.

Theorem 9.2 shows how slightly an ability to solve $f(z) = 0$ depends upon the computability of the derivative $f'(x)$. This is not to say that Secant iteration obsoletes Newton's. Instead the theorem simplifies the choice between them. Secant iteration is preferable to Newton's when …

- •Computing the derivative f' adds more than about 44% to the cost of computing f, and
- •The desired zero z is one across which f reverses sign, and
- •The desired accuracy requires at least several iterations, and
- •The contribution of roundoff to f is not so bad that its effect has to be minimized.

The last consideration arises out of Secant iteration's greater susceptibility than Newton's to roundoff, especially if its contribution has been seriously underestimated. If roundoff has been assessed reasonably well, and if iteration can be stopped as soon as the computed value of $|f|$ drops below or near its uncertainty due to roundoff, that last consideration becomes unimportant. Anyway, the global convergence properties of the two iterations rarely provide a strong reason to prefer one over the other.

Finally, Theorem 9.2 also contributes to Theorem 8.2 another corollary that is easy to prove:

> **Corollary 9.13:** Suppose f' is continuous and sum–topped throughout a closed interval Ω; or suppose $f = g - h$ is a differentiable difference between two convex functions, one non–decreasing and the other non-increasing, throughout a closed interval Ω. Then Secant iteration $x_{n+1} := x_n - f(x_n)/f^\dagger(x_n, x_{n-1})$, started from any x_0 and x_1 in Ω, either converges in Ω to the zero z of f or leaves Ω; the iteration cannot meander in Ω endlessly.

§10. Accelerated Convergence to a Zero in a Cluster

Where do multiple zeros come from? They would be extremely rare if the equations we solve were chosen at random; multiple zeros z imply an unlikely coincidence $f'(z) = f(z) = 0$. Since they are not so rare, their sources must be systematic. One such source is optimization. Suppose we wish to minimize the largest root $z(p)$ of an equation "$F(p, z) = 0$" containing a parameter p. Values p at which $dz/dp = 0$ are candidates, but they need not yield the desired minimum. It may occur when the two largest roots coincide, as is the case for $F(p, z) = z^2 - (p + 1/p)z + 1$; its optimal $p = 1$. For near–optimal values of p the two largest roots nearly coincide.

Where else may clustered zeros come from? Consider an analytic function $f(x)$ with several real and complex zeros $z_1, z_2, \ldots, z_m$ inside a region $ç$ in the complex plane, and suppose that $ç$ lies deep inside a far larger region $Ç$ that contains no other zeros nor singularities of f. Let the average of those m zeros be $\mu := \sum_j z_j /m$; then $f(x)$ and its derivative must closely resemble another analytic function $f(x)(x - \mu)^m/\prod_j (x - z_j)$ and its derivative at all x in $Ç$ far enough from $ç$. For all such x their respective Newton's iterating functions $N(x) := x - 1/\big(f'(x)/f(x)\big)$ and

$$x - 1/\big(f'(x)/f(x) - \sum_j (\mu - z_j)^2/(x - \mu)^3 + O(x - \mu)^{-4} \big)$$

must resemble each other closely too. In other words, to Newton's iterating function, any collection of several zeros may appear, from far enough away, like clustered zeros practically indistinguishable at that distance from a multiple zero. We have seen already, before and during Theorem 7.6, that convergence to a multiple zero can be slow. Consequently we should expect convergence to a cluster from afar to be retarded too. Usually it is retarded, but not always.

Take $f(x) := 3e^x - e^3 x$ for example. All its zeros are simple. Two of them, $z = 0.17856\ldots$ and $Z = 3$, are real; but infinitely many are complex falling not far from $2 + \ln(2k\pi) \pm (2k + 1/2)\pi\imath$ for positive integers k. From any $x_0 < 1$, Newton's iteration $x_{n+1} := x_n - f(x_n)/f'(x_n)$ converges to z almost immediately because $z - 0.003 < x_2 < z$ no matter how huge (and negative) x_0 is. From any big $x_0 > 2{\cdot}Z$, Newton's iteration converges to Z slowly at first, taking about x_0 iterations to get between Z and $Z + 0.001$ because $x_{n+1} \approx x_n - 1$ for a while. Thus, from far away on the positive (but not the negative) real axis, z and Z look to Newton's iteration like roots of infinite multiplicity towards which it must move very slowly. A simple way to cure this lethargy is to replace $f(x)$ by $x - 3 - \ln(x/3)$, which has the same real zeros but none complex.

In general lethargic convergence has no simple cure. And, when found, a cure rarely saves much time. No matter how slowly Newton's or Secant iterates x_n converge, usually $2^n f(x_n) \to 0$ because of Theorem 7.6. Then $f(x_n) \to 0$ so fast that it must soon fall below the threshold of rounding error noise in f, or else below the computer's Underflow threshold. Since the amount of time that can be saved is usually limited, no cure for lethargic convergence is worthwhile if it adds much to the cost of Newton's or Secant iteration; nor is a cure satisfactory if it spawns disagreeable consequences like convergence to an undesired zero.

When the multiplicity $m > 1$ of a desired zero z of f is known, superlinear convergence can be achieved by applying Newton's or Secant iteration to $|f|^{1/m}\,\mathrm{sign}(f)$ instead of f; then Newton's iteration takes the form $x_{n+1} := x_n - m\, f(x_n)/f'(x_n)$. However z is usually computable more

accurately as a simple zero of the derivative $f^{[m-1]}$, if this can be computed. (Could m be known but not $f^{[m-1]}$? Perhaps that's why an $m > 1$ is usually unknown.) An unknown m known to exceed 1 is probably 2 since larger multiplicities are extremely unlikely.

An unknown multiplicity can be estimated. For instance, if z is a zero of f with multiplicity m then $1/(1/(\ln|f(x)|)')' \to m$ as $x \to z$. This appears to require the computation of f" but that can be circumvented by introducing a multiplicity estimate m_n into an accelerated version of Newton's iteration thus:

$$m_n := \text{Max}\{\, 1, \ \text{Integer Nearest } (x_n - x_{n-1})/\big(f(x_n)/f'(x_n) - f(x_{n-1})/f'(x_{n-1}) \big) \,\} ;$$
$$x_{n+1} := x_n - m_n\, f(x_n)/f'(x_n) \ .$$

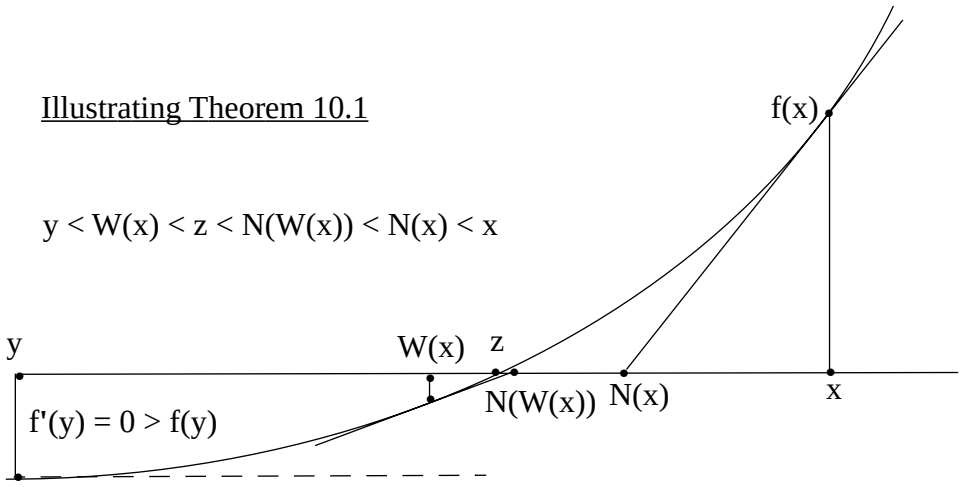
When x_n converges to a zero z of an analytic function f(x) it converges at least quadratically and m_n converges to the zero's multiplicity, which must be an integer. This convergence is faster than if Secant iteration had been applied to f(x)/f'(x) of which z is a simple zero. From far enough away, however, a cluster of zeros (complex as well as real) of f can appear so much like a multiple zero to Newton's iteration that m_n may actually approximate the number of zeros in the cluster. Only if and when iterates approach z can its own lower multiplicity m become manifest. Alas, the first few accelerated iteration steps can overshoot the zero nearest the starting point too easily, after which subsequent iterates may diverge or converge to a zero other than the one desired, especially if an extremal real zero was desired.

Take $f(x) := 3e^x - e^3 x$ for example again. Starting from $x_0 > 5$, the foregoing acceleration scheme practically always skips over the larger zero $Z = 3$ and converges to the smaller zero $z \approx 0.17856$. In general no way is known to moderate the growth of m_n so as to prevent this kind of undesired overshoot in all cases.

There is a special but common case that can be accelerated modestly without overshoot. Define

$N(x) := x - f(x)/f'(x)$ (Newton's iteration function) and
$W(x) := x - 2\, f(x)/f'(x)$ (Doubled-Newton's iteration function).

This W(x) can be iterated with little harm from overshoot in the following circumstances:



Theorem 10.1: Suppose that $f'(y) = 0 \geq f(y)$ at the left-hand end of a closed finite interval $y \leq x \leq x_0$ inside which $f''(x)$ is a positive nondecreasing function; also assume $f(x_0) > 0$. Then, in that interval, ...
1) Equation "$f(z) = 0$" has just one root $z \geq y$; and $W(x) < N(x)$ when $x > z$.
2) $N(x) > z$ if $y < x \neq z$, and then $W(x) > y$ unless $W(x) = y = z$.
3) If $x > z$ then $z < N(W(x)) \leq N(x)$, with equality only when $f''' \equiv 0$.

Starting from $x_0 > z$ this theorem motivates the following procedure:

Iterate $x_n := W(x_{n-1})$, thereby descending faster than Newton's iteration would, until $x_J \leq z$ (detected when $x_{J+1} \geq x_J$); then if $x_{J+1} > x_J$ replace x_{J+1} by $N(x_J)$ and continue Newton's.

The theorem's clause (3) guarantees that the last retained W-iterate $x_J := W(x_{J-1})$, after which iteration reverts to $x_{n+1} := N(x_n)$, cannot jump beyond z so close to y that the next Newton iterate x_{J+1} would jump way back behind x_{J-1}. On the contrary, x_{J+1} comes closer to z than $N(x_{J-1})$ would have come. An example will illustrate the procedure after the theorem's proof.

Proof of Theorem 10.1:
1) As $x - y$ increases through positive values, so does $f'(x)$ because $f'' > 0$. Therefore at most one root $z \geq y$ can exist in the given interval; $f(x)$ increases through 0 to a positive value $f(x_0)$ as x increases from y to x_0, so $z \geq y$ does exist in the interval. And then obviously $N(x) - W(x) = f(x)/f'(x) > 0$ for all $x > z$ therein.

2) $N'(x) = f(x)f''(x)/f'(x)^2$ has the same sign as $x - z$ if $x > y$; therefore $N(x)$ descends to its minimum value $N(z) = z$ as $x \to z$ from either side. A nondecreasing derivative is a continuous and therefore integrable derivative, and $f''(x)$ is nondecreasing as x increases beyond y, so

$$0 \leq \int_y^x \int_y^\tau (f''(\tau) - f''(\sigma)) d\sigma \, d\tau = (x - y)f'(x) - 2f(x) + 2f(y) .$$

This implies $W(x) \geq y - 2f(y)/f'(x) \geq y$ too with strict inequality unless $y = z$, in which special case Theorem 7.5 above implies $N(x) \to z+$ and $W(x) \to z+$ as $x \to z+$. In the further special case of a quadratic f (constant $f'' > 0$) with a double zero $y = z$ we find $W(x) \equiv z$.

3) When $z \leq W(x) < x$, inequality $N(W(x)) \leq N(x)$ is now obvious; but a proof is harder when $y < W(x) < z < x$. The proof might be easier if $N(x) - N(W(x)) = f(x)/f'(x) + f(W(x))/f'(W(x))$ increased monotonically, but it needn't; for example, try $f(x) = x^2 + (x/2)^{24} - 1 - 1/2^{24}$. Worse, $N(x) - N(W(x))$ vanishes like $O(x - z)^3$, so three differentiations (or integrations) would be needed to infer the desired inequality directly from the hypothesis $f''' \geq 0$. We shall simplify the work a little by proving that $(N(x) - N(W(x))) \cdot f'(W(x)) = f'(W(x)) \cdot f(x)/f'(x) + f(W(x)) \geq 0$.

To exploit the symmetry of $W(x)$ and x about $N(x)$, let's use abbreviations $q := f(x)/f'(x)$, $n := N(x) = x - q$, and $w := W(x) = n - q$; then $f''(n + \sigma) - f''(n - \sigma) \geq 0$ when $0 \leq \sigma \leq q$ because f'' is nondecreasing. Integrate twice to get

$$0 \leq \int_0^q \int_\tau^q (f''(n + \sigma) - f''(n - \sigma)) d\sigma \, d\tau = (f'(x) + f'(w)) \cdot q - f(x) + f(w) ,$$

which simplifies to the last inequality of the previous paragraph. This inequality becomes equality just when f is quadratic (with constant $f'' > 0$). END of PROOF.

I first proved theorem 10.1 in the early 1960's, but rather differently, for the special case of a polynomial P(x) whose (at least two) zeros are all real, the largest being z ; one of f = P or f = -P can easily be shown to satisfy the hypotheses of this theorem. A proof for this polynomial case can be found in Stoer & Bulirsch [1980]; for this case, parts 1) and 2) of the theorem had been quoted by Jim Wilkinson [1965], who had learned them from Hans Maehly as well as me. Soon afterwards Werner Greub liberated the whole theorem from polynomials by suggesting that the crucial hypothesis was merely $f'''(x) \geq 0$, from which the foregoing proof evolved.

The procedure described before the proof locates the largest real zero of a polynomial whose other zeros, real and complex, all have lesser real parts. It locates the largest real zero (Z = 3) of examples like $f(x) = 3e^x - e^3 x$ discussed above, usually faster than would Newton's iteration all the way. Theorem 10.1 provides a guarantee that the doubled iteration $x_{n+1} = W(x_n)$ cannot overshoot the desired zero z so far as would lose more than one iteration–step after reversion to Newton's. Except for that one step that overshoots z , the iterates of W starting from $x_0 > z$ approach z faster than correspondingly numbered iterates of N would because $N'(x) > 0$ for all $x > z$ (see the proof of (2) above).

How much faster do iterates of W descend than iterates of N would? Since $W(x) \leq N(N(x))$ at x close enough to $z \neq y$ and usually at all $x > z$ in the interval, W usually descends at least twice as fast as N until z is overshot. It happens for $f(x) := e^x$ whose $W(x) \cong N(N(x)) \cong x - 2$ and $-\infty = y = z < x < x_0 < +\infty$. But not always; $f(x) := x/(1-x)$ in the interval $z = 0 < x < x_0 < 1$ behaves differently because its $W(x) > N(N(x))$ when $1 > x > 2/(1 + \sqrt{5}) \approx 0.618$. More nearly typical is example $f(x) := 3e^x - e^3 x$ for which iterates descend to Z = 3 from $x_0 = 8$ thus:

Table 1: For $f(x) := 3e^x - e^3 x$

	Iterates of N	Iterates of W
x_0	8	8
x_1	7.015757	6.031524
x_2	6.052129	4.195981
x_3	5.132988	2.912537
x_4	4.302929	3.006191
x_5	3.631900	3.000029
x_6	3.198687	3
x_7	3.025447	
x_8	3.000476	
x_9	3.000000	

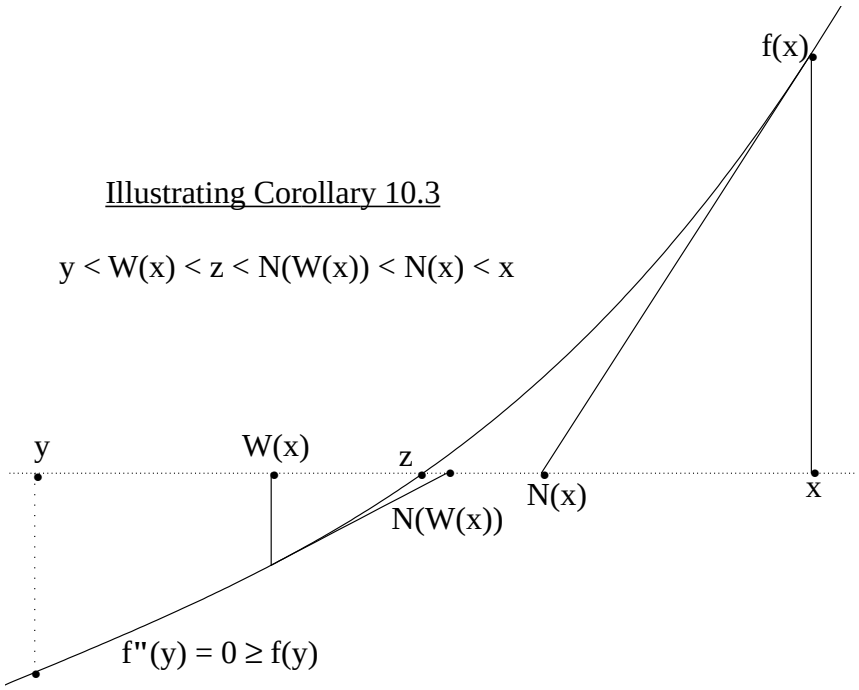
The doubled iteration $x_{n+1} := W(x_n)$ can still converge arbitrarily slowly to a highly multiple zero; but its values $f(x_n)$ tend to zero usually more than twice as fast as Newton's would, and always at least twice as fast as Theorem 7.6 described:

Corollary 10.2: Assume the hypotheses of Theorem 10.1 again, and suppose also that the procedure described just after it is followed. Then either the iteration ultimately reverts to Newton's and converges quadratically, or else the doubled iteration $x_n := W(x_{n-1})$ converges monotonically, though slowly, and $f(x_n)$ tends monotonically to 0 at least so fast that $\sum_n 4^n |f(x_n)| \le |f(x_0)|\,(x_0 - z)/(x_0 - x_1)$.

Proof: Either $y < z$ or $y = z$. If $y < z$ then $f'(z) \neq 0$ and ultimately $x_n := W(x_{n-1})$ falls upon z and stops or falls between z and y. In the latter event the iteration reverts to Newton's iteration which, after stepping backward once to $N(x_n)$ between z and $N(x_{n-1})$, converges monotonically according to Theorem 7.5, and quadratically according to Theorem 7.4.

If $y = z$ then $f'(z) = f(z) = 0$ and the doubled iteration $x_{n+1} := W(x_n)$ converges monotonically towards z. This iteration is the same as Newton's applied to solve the equation $\sqrt{f(z)} = 0$. Is $\sqrt{f}$ convex? To find out consider the Riemann–Stieltjes integral $\int f\,df''$, which exists since f'' is nondecreasing. If $x > z$ then $0 \le 2\int_z^x f(\tau)\,df''(\tau) = 2f''(x)\cdot f(x) - (f'(x))^2 = 4(\sqrt{f(x)})^3\cdot(\sqrt{f(x)})''$. Therefore $\sqrt{f}$ satisfies the convexity hypothesis f satisfied in Theorem 7.6, whence follows its conclusion for $\sqrt{f}$, which is this Corollary's inequality. END OF PROOF.

What if f' never vanishes, or whether f' ever vanishes is unknown? So long as W. Greub's hypothesis $f''' \ge 0$ holds, the doubled Newton iteration $x_{n+1} := W(x_n)$ deserves to be tried:



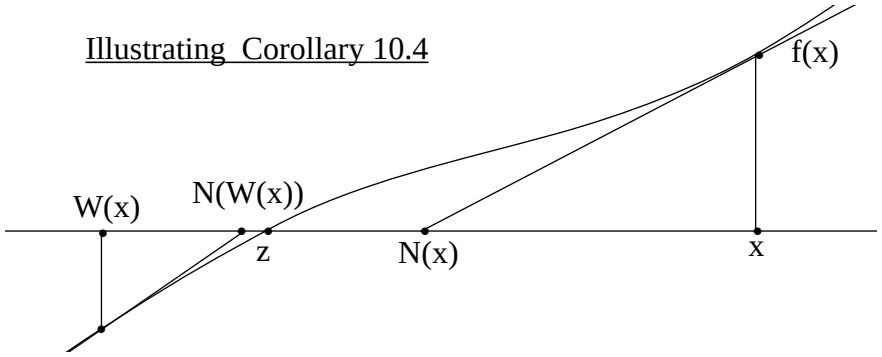
Corollary 10.3: Redefine y in Theorem 10.1 to satisfy $f''(y) = 0$ and $f(y) < 0$, leaving all else unchanged. Then all its three inferences 1), 2) and 3) persist except if $W(x) < y$, in which case $W(x) > y - (x-y)$ is all that can be inferred.

Proof: Almost the same as for Theorem 10.1. END OF PROOF.

For example take $f(x) := (x+1)^3 + p\cdot x - 1$ with a parameter $p > 0$ and initial $x_0 := 100$.

Table 2: For $f(x) := (x+1)^3 + p\cdot x - 1$, $f''(-1) = f(0) = z = 0$.

	p = 6		p = 300	
	Iterates of N	Iterates of W	Iterates of N	Iterates of W
x_0	100	100	100	100
x_1	66.3204	32.6407	65.6894	31.3788
x_2	43.8610	10.1386	42.5040	6.20835
x_3	28.8788	2.51426	26.5975	-3.60146
x_4	18.8773	-0.156432	15.3795	-0.170196
x_5	12.1906	0.00808341	7.22709	0.00025504
x_6	7.70709	0.00002178	1.81222	0.000000001
x_7	4.68552	0.00000000	0.067204	0
x_8	2.63747	0	0.00004666	
x_9	1.25975		0.000000000	
x_{10}	0.410861		0	
x_{11}	0.0538886			
x_{12}	0.00096709			
x_{13}	0.000000312			
x_{14}	0.000000000			



Corollary 10.4: Suppose $f' > 0$ and f'' is nondecreasing throughout an interval Ω wide enough to contain z, where $f(z) = 0$, as well as x, $N(x)$ and $W(x)$ for every x in Ω. Then $N(x)$ must lie between x and $N(W(x))$, though z may lie anywhere on the same side of x as $N(x)$, including perhaps between them.

Proof: This corollary's hypotheses apply to $-f(-x)$ as well as to $f(x)$, so the assumptions that $x > z$ and $f(x) > 0$ simplify the proof without loss of generality. Then the double integral at the end of Theorem 10.1's proof proves that $N(x) \geq N(W(x))$, as claimed. END OF PROOF.

The corollaries above motivate the following procedure whenever $\pm f$ satisfies their assumptions:

> Whenever $f(x_n)/f(x_{n-1})$ is not small, say whenever $f(x_n)/f(x_{n-1}) > 0.1$, compute $x_{n+1} := W(x_n)$ instead of $N(x_n)$ unless doing so would escape from a straddle known to enclose only one zero of f.

The doubled Newton iteration $x_{n+1} := W(x_n)$ works so well in the circumstances for which it was intended that it encourages us to consider a doubled Secant iteration too. There are two ways to double the Secant iterating function

$$S(x, X) := x - f(x)/f^\dagger(x, X) = x - f(x)\cdot(X-x)/(f(X) - f(x)) .$$

One is the obvious way:

$$\$(x, X) := x - 2\cdot f(x)/f^\dagger(x, X) .$$

The unobvious way applies Secant iteration to the equation $\sqrt{f} = 0$ to get an iterating function

$$R(x, X) := x - (X-x)/(\sqrt{(f(X)/f(x))} - 1) .$$

Both doubled iterations work. The latter is faster because, if $z < x < X$ and so $0 < f(x) < f(X)$, then $R(x, X) < \$(x, X) < S(x, X)$, as is easy to verify. Therefore we concentrate upon $R(\ldots)$.

Suppose the hypotheses of Theorem 10.1 are in force:

> $f'(y) = 0 \geq f(y)$ at the left-hand end of a finite interval $y \leq x \leq x_0$ throughout which f'' is a positive nondecreasing function; also assume $f(x_0) > 0$, so the equation "$f(z) = 0$" has just one root $z \geq y$ in that interval.

The procedure that follows that theorem is now supplanted by this:

> Starting from $x_0 > x_1 > z$, …, iterate $x_n := R(x_{n-1}, x_{n-2})$, thereby descending faster than Secant iteration would, until $x_n < z$ (detected when $f(x_{n-1})/f(x_n) < 0$), and then revert to $x_{n+1} := S(x_n, x_{n-1})$.

Once again, as in Theorem 10.1 part 3), we seek reassurance that the last doubled–iterate x_n cannot overshoot z so far as might set subsequent Secant iterates back behind $S(x_{n-1}, x_{n-2})$.

> **Conjecture 10.5:** Assume the hypotheses of Theorem 10.1 again, and also the definitions of S and R; then $S(S(R(u, w), u), R(u, w)) \leq S(u, w)$ if $z < u < w$.

Discussion: Intermediate Value Lemma 9.1 lets us define $v(u, w)$ to lie strictly between u and w and satisfy $N(v(u, w)) = S(u, w) > z$ whenever either $y \leq w < u < z$ or $z < u < w$. This $v(\ldots)$ is defined uniquely because $N(x)$ is monotone decreasing when $y \leq x \leq z$, increasing when $z \leq x$.

THERE ARE NUMEROUS DETAILS STILL TO BE SUPPLIED HERE.
??

Corollaries:

Convergence of $f(x_n)$ to 0 is faster than $1/3^n$ for \$ or $1/4^n$ for R.

Example:

Table 3: For $f(x) := 3e^x - e^3 x$

	Iterates of S	Iterates of R
x_0	9	9
x_1	8	8
x_2	7.427732	6.479176
x_3	6.706504	5.205631
x_4	6.057988	4.250802
x_5	5.407616	3.166181
x_6	4.800048	2.812781
x_7	4.243171	2.976582
x_8	3.766543	3.003603
x_9	3.396594	2.999936
x_{10}	3.154697	3.000000
x_{11}	3.037465	3
x_{12}	3.004024	
x_{13}	3.000111	
x_{14}	3.000000	

??

Work still to be rewritten out:

What to do when the search for a zero of f encounters a value of x outsude the domain of f ? See pp. 23-5 of <www.eecs.berkeley.edu/~wkahan/Math128/LecRlRtF.pdf> .

§11. All Real Zeros of a Real Polynomial. Finding only real zeros of a real polynomial of high degree is applicable to Tarsky resolution of rational (in)equalities, geometrical computation, construction of numerical ODE formulas. Sturm Sequences (Turnbull [1952]) are costly to compute or vulnerable to roundoff or both. A better way using Rolle's Theorem and running error–bounds is attractive when the real zeros are far fewer than the polynomial's degree, as is usually the case.

§12. Zeros of a Real Cubic. How to find the zeros of a real cubic quickly and accurately using Newton's iteration from an artfully chosen starting guess.

§13. Error Bounds for Computed Roots using §A5: Running Error Bounds

§ççç. Conclusion:
These notes were written at the behest of two mathematicians who inhabit my body. The pure mathematician savors surprises. The applied mathematician tries to avoid them by predicting how computational procedures will behave. Both mathematicians rejoice when they prove a procedure to be surprisingly predictable. But the latter's joy must be short-lived for two reasons. First, compared with the procedures they explain, our proofs are too long; they augur ill for our understanding of more complicated procedures. Second, more complicated procedures will arise inevitably from attempts to circumvent limitations in the simple procedures we have come to understand at last. Thus, these notes contain the seeds of their own obsolescence.

We say "mature" when we wish to avoid the pejorative "obsolescent". The material in these notes will soon be mature if it isn't already. The corresponding material in most textbooks is too mature. Bringing textbooks up to date is a formidable challenge compounded by limitations upon space and time, both the author's and the readers'. Until a brave author rises to this challenge, the burden of these notes will continue to be added to my students' load. They and I pray that their load will be lightened soon.

Surely Sharkovsky's Theorem 5.1 deserves to appear in texts. So does Corollary 8.3 and an example of its application, if not also Theorem 8.2, because they suggest how to reformulate equations to make them easier to solve by Newton's and Secant iteration. Theorem 9.2 deserves at least a footnote, more if someone finds a shorter proof, because it justifies the use of Secant iteration instead of Newton's. Error analysis, dull but necessary, deserves more space in texts too; without it, who can tell when to quit iterating or how much the result is worth?

§A1. Appendix: Divided Differences Briefly

This topic is discussed at length in Numerical Analysis texts like Conte & de Boor [1980], but usually in the context of *Interpolation* and always in a different notation. For an ancient subject the persistence of diverse notations suggests that none are satisfactory and licenses us to introduce another notation more nearly analogous to a widely used notation for derivatives. Inspired by formulas attributed to Hermite, we define for any sufficiently smoothly differentiable function f(x) its *First Divided Difference*

$$f^{\dagger}(u, w) := \int_0^1 f'(u + (w-u)t) \, dt$$

and its *Second Divided Difference*

$$f^{\dagger\dagger}(u, v, w) := \int_0^1 \int_0^t f''(u + (v-u)t + (w-v)s) \, ds \, dt .$$

For positive integers k generally, the k^{th} divided difference is the uniformly weighted average of the k^{th} derivative over a simplex, the convex hull of k+1 arguments, then divided by k! . However k > 2 will not be needed in these notes. In general the argument x of f(x) could be a vector but in these notes it will almost always be a real scalar. Then that simplex, the convex hull of the k+1 real arguments, degenerates into an interval of the real x–axis over which the k^{th} divided difference becomes a positively (not necessarily uniformly) weighted average of the k^{th} derivative divided by k! . For instance, if u < v < w then it follows that

$$f^{\dagger\dagger}(u, v, w) = \left(\int_u^v (t-u) f''(t) \, dt /(v-u) + \int_v^w (w-t) f''(t) \, dt /(w-v) \right) /(w-u) .$$

Because it is an average, the k^{th} divided difference lies between the largest and least values taken in that interval by the k^{th} derivative divided by k! . This *Mean Value* property figures in nearly all applications of divided differences in these lecture notes. Divided differences turn up elsewhere as coefficients in *Newton's Interpolating Polynomials*, which see below, or during root–finding or optimization, or when differential equations are solved using finite differences.

Because the argument x of f(x) is a scalar, the foregoing integrals can always be " simplified " into expressions with no integral signs. For instance,

$$\begin{aligned} f^{\dagger}(u, w) &= (f(u) - f(w))/(u - w) && \text{if } w \neq u , \\ &= f'(u) && \text{if } w = u , \\ &= f^{\dagger}(w, u) && \text{(arguments' order doesn't matter)} \\ &= f'(v) && \text{at some } v \text{ strictly between } u \text{ and } w \text{ if they are unequal.} \end{aligned}$$

The first two equations above constitute an alternative definition of $f^{\dagger}$ in so far as they describe it independently of whether f' exists strictly between u and w ; and then the last equation turns out to be valid so long as f'(x) does exist at every x strictly between u and w , and f(x) is continuous at u and w , even if f' is not integrable. Similarly the next two lines describe or alternatively define $f^{\dagger\dagger}$ independently of whether f" exists:

$$\begin{aligned} f^{\dagger\dagger}(u,v,w) &= (f^{\dagger}(u,v) - f^{\dagger}(v,w))/(u-w) && \text{if } w \neq u , \\ &= \partial f^{\dagger}(u,v)/\partial u = (f'(u) - f^{\dagger}(u,v))/(u-v) && \text{if } w = u \neq v \\ &= f(u)/((u-v)(u-w)) + f(v)/((v-w)(v-u)) + f(w)/((w-u)(w-v)) && \text{if } u \neq v \neq w \neq u \\ &= f^{\dagger\dagger}(v,w,u) = f^{\dagger\dagger}(u,w,v) = \cdots && \text{(arguments' order doesn't matter)} \\ &= \tfrac{1}{2} f''(y) && \text{at some } y \text{ between } \min\{u,v,w\} \text{ and } \max\{u,v,w\} \text{ if } f'' \text{ exists ... .} \end{aligned}$$

(Don't confuse $f^{\dagger\dagger}(u, v, w)$ with $f'^{\dagger}(u, w) = (f'(u) - f'(w))/(u-w) = f^{\dagger\dagger}(u, u, w) + f^{\dagger\dagger}(u, w, w)$.)

Strictly speaking, we should write $f^{\dagger}(\{u, w\})$ instead of $f^{\dagger}(u, w)$ because it is best construed as a function of an unordered pair $\{u, w\}$ that replaces the single argument x of $f(x)$. Similarly we should write $f^{\dagger\dagger}(\{u, v, w\})$ instead of $f^{\dagger\dagger}(u, v, w)$. The extra braces $\{\ldots\}$ are superfluous in divided differences of functions of one argument, but a necessary nuisance in partial divided differences of functions of more than one argument. For instance, given any function $g(x, y)$ of two scalar arguments, we must distinguish $g^{\dagger}(\{u,w\}, y) := (g(u,y) - g(w,y))/(u-w)$ from $g^{\dagger}(x, \{u,w\}) := (g(x,u) - g(x,w))/(u-w)$ by our placement of the braces to show which argument was split into a pair; $\partial g(x,y)/\partial x = g^{\dagger}(\{x,x\}, y)$ and $\partial g(x,y)/\partial y = g^{\dagger}(x, \{y,y\})$ are distinguished by the same imperative. Similarly the mixed partial divided difference

$$g^{\dagger\dagger}(\{t,u\}, \{v,w\}) := (g(t,v) - g(u,v) - g(t,w) + g(u,w))/((t-u)(v-w))$$

has to be distinguished from

$$g^{\dagger\dagger}(t, \{u,v,w\}) := g(t,u)/((u-v)(u-w)) + g(t,v)/((v-w)(v-u)) + g(t,w)/((w-u)(w-v))$$

much as we distinguish $\partial^2 g/\partial x\partial y = g^{\dagger\dagger}(\{x,x\}, \{y,y\})$ from $\partial^2 g/\partial y^2 = 2g^{\dagger\dagger}(x, \{y,y,y\})$. (The factor 2 will be vindicated in a moment; and if discontinuity invalidates $\partial^2 g/\partial x\partial y = \partial^2 g/\partial y\partial x$ it may render $g^{\dagger\dagger}(\{x,x\}, \{y,y\})$ ambiguously dependent upon the order of limiting processes.)

Return to functions $f(x)$ of one argument. A composed function $f(x) = h(p(x))$ has a derivative $f'(x) = h'(p(x))\, p'(x)$ derived from a *Chain Rule* that works analogously for divided difference $f^{\dagger}(\{u,w\}) = h^{\dagger}(\{p(u), p(w)\})\, p^{\dagger}(\{u,w\})$. And, just as derivatives compound to form higher order derivatives like $f''(x) = (f'(x))'$, divided difference operations compound to form higher order divided differences. For instance, the alternative definition of $f^{\dagger\dagger}$ above amounts to

$$f^{\dagger\dagger}(\{u, v, w\}) = f^{\dagger\dagger}(\{\{u,v\}, w\}) = f^{\dagger\dagger}(\{u, \{v,w\}\}) = f^{\dagger\dagger}(\{v, \{u,w\}\}) ;$$

in other words, every second divided difference is a first divided difference of a first divided difference in as many as three ways. Since derivatives are limiting values of divided differences,

$$\partial f^{\dagger}(\{u,w\})/\partial u = f^{\dagger\dagger}(\{\{u,u\},w\}) = f^{\dagger\dagger}(\{u,u,w\}) \quad \text{and} \quad \partial f^{\dagger}(\{u,w\})/\partial w = f^{\dagger\dagger}(\{u,w,w\})$$

provided the derivatives in question exist. Setting $u = v = w$ vindicates the factor 2 in

$$f''(v) = df^{\dagger}(\{v,v\})/dv = f^{\dagger\dagger}(\{\{v,v\},v\}) + f^{\dagger\dagger}(\{v,\{v,v\}\}) = 2f^{\dagger\dagger}(\{v,v,v\}) .$$

Like differentiation, divided differencing maps certain families of functions into themselves. Divided differences of polynomials are polynomials, albeit with more arguments. Divided differences of rational functions of scalar arguments are rational. Likewise algebraic. Irrational algebraic functions are handled by implicit divided differencing just like implicit differentiation, and derived in the same way from the Chain Rule. With the aid of that rule, any algorithm that computes an algebraic function $f(x)$ can be expanded mechanically into a similar algorithm that computes divided difference $f^{\dagger}(u, w) = (f(u) - f(w))/(u-w)$ at almost the same cost as computing $f(u)$ and $f(w)$ but without ever dividing by $u-w$. A simple example is $\sqrt{}^{\dagger}(u, w) = 1/(\sqrt{u} + \sqrt{w})$. Ideally such expansions should be performed on request by computerized algebra software like Derive, Macsyma, Maple and Mathematica, which ought to manipulate divided differences as well as derivatives, but they don't. Consequently the computing public remains largely unable to exploit a valuable but little known application of divided differences, namely the suppression of numerical instability attributable to systematic cancellation.

Many a numerical computation turns out to be the computation of a divided difference in disguise. Attempts to compute $f^\dagger(u, w)$ naively from the obvious formula $(f(u) - f(w))/(u-w)$ can be thwarted by roundoff and then cancellation when u is too near w. If nonzero, the divisor $u - w$ is no problem because its cancellation occurs without error. But the computed value of $f(u)$ is generally rounded to, say, $f(u) + \Delta f(u)$, and therefore the value computed naively for $f^\dagger(u, w)$ when $f(u) - f(w)$ would mostly cancel turns out to be $f^\dagger(u, w) + (\Delta f(u) - \Delta f(w))/(u - w)$ instead, overwhelmed by the last quotient if $u - w$ barely exceeds roundoff. For example consider the solution of a quadratic equation $Az^2 - 2Bz + C = 0$. Its solutions z are $(B \pm \sqrt{}(B^2 - AC))/A$. The solution of smaller magnitude, $z := C\,\mathrm{sign}(B)\,\sqrt{}^\dagger(B^2, B^2 - AC)$, is vulnerable to roundoff and cancellation when $|AC| << B^2$ unless the divided difference $\sqrt{}^\dagger$ is " expanded " as was mentioned above to yield $z = C/(B + \mathrm{sign}(B)\sqrt{}(B^2 - AC))$, which stays accurate if $|AC| << B^2$.

Sometimes the accuracy of transcendental expressions can be insulated from cancellation with the aid of ancient formulas motivated by divided differences. For example, $(\tan(u) - \tan(w))/(u - w)$ is best computed from the formula $\tan^\dagger(u, w) = (1 + \tan(u)\tan(w))\tan^\dagger(u - w, 0)$ when u nearly equals w. Sometimes an inverse divided difference can render cancellation harmless. For instance, because $\ln^\dagger(v, 1) = \ln(v)/(v - 1)$ does not suffer from cancellation when v nearly equals 1, the computation of $\exp^\dagger(u, 0) = (\exp(u) - 1)/u$ can be protected from cancellation in the numerator by the use of the formula $\exp^\dagger(u, 0) = 1/\ln^\dagger(\exp(u), 1)$ instead. These transcendental examples work because they exploit the few occasions when transcendental functions take simple rational values at rational arguments.

In general, transcendental functions afflict divided differences but not derivatives in two ways. First, many transcendental functions have simple (perhaps algebraic) derivatives but no simple " expanded " divided differences undefiled by cancellation. For example, $d^2 \ln(v)/dv^2 = -1/v^2$; but no known simple finite formula for $\ln^{\dagger\dagger}(u, v, w)$ stays accurate no matter how u, v and w approach each other. Secondly, the divided difference of a non–polynomial rational function of a *vector* argument generally involves logarithms and/or arctangents. For example, let column vectors $\mathbf{x} := \begin{bmatrix} y \\ z \end{bmatrix}$ and $\mathbf{u} := \begin{bmatrix} v \\ w \end{bmatrix}$, and let $f(\mathbf{x}) := y/z$; then its derivative $f'(\mathbf{x}) = [1/z,\ -y/z^2]$ is a rational row vector but Hermite's formula for its first divided difference yields a transcendental

$$f^\dagger(\mathbf{x}, \mathbf{u}) = [\ \ln^\dagger(z,w),\qquad \tfrac{1}{2}(y-v)\big(\ln^{\dagger\dagger}(z,z,w) - \ln^{\dagger\dagger}(z,w,w)\big) - \tfrac{1}{2}(y+v)/(zw)\ ]\ .$$

~ ~

Newton's Interpolating Polynomials approximate functions of scalar or vector arguments:

$$\begin{aligned} f(x) &:= f(u) + f^\dagger(u,x)(x-u)\ , \\ &= f(u) + \big(\ f^\dagger(u,v) + f^{\dagger\dagger}(u,v,x)(x-v)\ \big)(x-u)\ , \\ &= f(u) + \big(\ f^\dagger(u,v) + \big(\ f^{\dagger\dagger}(u,v,w) + f^{\dagger\dagger\dagger}(u,v,w,x)(x-w)\ \big)(x-v)\ \big)(x-u)\ ,\ \ \ldots\ \text{etc.} \end{aligned}$$

The polynomial in x obtained by substituting 0 for $f^{\dagger\dagger\dagger}$ interpolates (matches) $f(x)$ at $x = u$, $x = v$ and $x = w$; elsewhere it differs from $f(x)$ by a *remainder* term $f'''(y)(x-w)(x-v)(x-u)/6$ in which y falls somewhere inside the convex hull of $\{ u, v, w, x \}$. Interpolation is *osculatory* if two of u, v, w coincide. This polynomial's degree is minimal only for a scalar argument x.

§A2. Appendix: Functions of Restrained Variation

This digression concerns a way to sum an undulating function's fluctuations. The *Total Variation* of a real function $Q(x)$ over a closed finite interval $u \le x \le w$ is defined to be

$$V_u^w Q := \int_u^w |dQ(x)| = \int_u^w |Q'(x)|dx$$

though the last equation is valid only if $|Q'(x)|$ exists and is integrable. In general, a function whose Total Variation over some interval is finite is called " a function of Bounded Variation " thereon. Such functions figure in Measure Theory, Stieltjes integrals, and Fourier series. They can have none but jump discontinuities, and at most countably many of these (Bartle [1976]). In particular, a derivative of bounded variation must be a continuous derivative.

Obviously $V_u^w Q \ge |Q(w) - Q(u)|$, with equality just when Q is monotonic. Where $Q(x)$ is continuous, so is $V_u^x Q$. Wherever $Q(x)$ jumps, so do $V_u^x Q$ and $V_x^w Q$, and by the same amount, but the former always increases and the latter always decreases as x increases. Hence Total Variation is *Additive* over abutting sub–intervals: if $u \le x \le w$ then $V_u^x + V_x^w = V_u^w$. It is a *Semi–Norm* because it satisfies the *Triangle Inequality* $0 \le V_u^w (P \pm Q) \le V_u^w P + V_u^w Q$.

If $V_u^w Q < \infty$ and $u \le x \le w$ then $Q(x)$ admits infinitely many *Splittings* into a difference $Q = P - M$ between two non–decreasing functions $P(x) := \frac{1}{2}(R(x) + Q(x) + V_u^x Q - \frac{1}{2} V_u^w Q)$ and $M(x) := \frac{1}{2}(R(x) - Q(x) - V_x^w Q + \frac{1}{2} V_u^w Q)$ in which R can be *any* non–decreasing function. Conversely, any non–decreasing P and M determine both $Q := P - M$ of bounded variation and the function $R(x) := \frac{1}{2}(P(x) + M(x) - V_u^x (P - M)) + \frac{1}{2}(P(x) + M(x) + V_x^w (P - M))$ that appears in Q's splitting; this R is non–decreasing because $P + M$ varies faster than $P - M$.

If Q and R are continuous, so are P and M, and *vice-versa*. If Q and R have integrable derivatives, so do P and M, and *vice-versa*. But when Q' is so violently oscillatory that $V_u^w Q = +\infty$ then Q is unsplittable, as are examples like $Q(x) = x^2 \cos(1/x^2)$ around 0.

Among functions Q of bounded variation, the ones that will interest us have a splitting $Q = P - M$ that is special because all three of Q, P and $-M$ have the same sign and keep it throughout the interval $u \le x \le w$. We shall call such a function Q " a function of Restrained Variation."

> **Lemma A2.1: A Function of Restrained Variation**
> Q can be split into a difference $Q = P - M$ between two non–decreasing functions P and M, one non–negative and the other non–positive, throughout the closed finite interval $u \le x \le w$ if and only if $V_u^w Q \le |Q(u) + Q(w)|$.

Proof: If necessary, replace Q by $-Q$ to get $Q \ge 0$. If $r := Q(u) + Q(w) - V_u^w Q \ge 0$ then choose any non–decreasing $R \ge 0$ and $P(u) \ge 0$ and $M(w) \le 0$ subject only to the constraint $2P(u) - 2M(w) + R(w) - R(u) = r$, and construct functions

$$P(x) := P(u) + \tfrac{1}{2}(V_u^x Q + Q(x) - Q(u) + R(x) - R(u)) \ge 0 , \quad \text{and}$$

$$M(x) := M(w) - \tfrac{1}{2}(V_x^w Q + Q(x) - Q(w) + R(w) - R(x)) \le 0 ;$$

evidently they are non–decreasing and satisfy $P(x) - M(x) = Q(x)$ too as desired. If $r = 0$ this splitting is determined uniquely with $P(u) = M(w) = R(x) - R(u) = R(w) - R(x) = 0$. On the other hand, if a splitting $Q = P - M$ already exists with non–decreasing P and M and $P \ge 0 \ge M$, then this splitting also determines the non–decreasing $R(x) := P(x) + M(x) - \frac{1}{2} V_u^x Q + \frac{1}{2} V_x^w Q$ as explained before the lemma; therefore $0 \le R(w) - R(u) = P(w) + M(w) - P(u) - M(u) - V_u^w Q$ whence $V_u^w Q \le Q(w) + 2M(w) - 2P(u) + Q(u) \le Q(u) + Q(w)$ as claimed. END OF PROOF.

What are here called " functions of restrained variation " are also called " tame " by Aharoni *et al*. [1992], who characterized them by means of a discretized version of the foregoing lemma, which now shortens the proof of their characterization:

Lemma A2.2: Tame Functions (Aharoni *et al*. [1992])
$Q(x)$ is a nonnegative function of restrained variation over the interval $u \le x \le w$ if and only if $Q(x_0) - Q(x_1) + Q(x_2) - \dots - Q(x_{2k-1}) + Q(x_{2k}) \ge 0$ for every integer $k \ge 0$ whenever $u \le x_0 \le x_1 \le x_2 \le \dots \le x_{2k-1} \le x_{2k} \le w$.

Proof: If $Q = P - M$ for some non–decreasing $P \ge 0$ and $M \le 0$, then every alternating sum $Q(x_0) + \sum_{j=1}^{2k} (-1)^j Q(x_j) = P(x_0) + \sum_{j=1}^{k} (P(x_{2j}) - P(x_{2j-1})) + \sum_{j=0}^{k-1} (M(x_{2j+1}) - M(x_{2j})) - M(x_{2k})$ is nonnegative term–by–term, which confirms the lemma's " only if " part. Except for setting $k = 0$ to prove $Q \ge 0$, the "if" part is harder to prove. Its proof is easier when $Q(x)$ takes its locally extreme values at only finitely many points in the interval $u \le x \le w$, including its endpoints among them. Then we assign $x_0 := u$, $x_{2k} := w$, and for $0 < j \le k$ we set all other x_{2j} to be all consecutive points where $Q_{2j} := Q(x_{2j})$ is locally minimal, and x_{2j-1} to be all consecutive points where $Q_{2j-1} := Q(x_{2j-1})$ is locally maximal; these points interlace, including possibly $x_1 = u$ if $Q_0 = Q(u)$ is locally maximal and/or $x_{2k-1} = w$ if $Q_{2k} = Q(w)$ is locally maximal. Because the lemma's alternating sums are all nonnegative, we soon find that $Q(u) + Q(w) \ge (Q_1 - Q_0) + (Q_1 - Q_2) + (Q_3 - Q_2) + \dots + (Q_{2k-1} - Q_{2k-2}) + (Q_{2k-1} - Q_{2k}) = V_u^w Q$. Applying Lemma A2.1 completes the proof for the case when Q has just finitely many extrema. When Q has infinitely many extrema the last equation is invalid but salvaged by taking its left–hand side's supremum over all partitions $u = x_0 \le x_1 \le x_2 \le \dots \le x_{2k-1} \le x_{2k} = w$. END OF PROOF.

Restrained variation has only one consequence significant for Newton's or Secant iterations; it is the following corollary, whose now nearly obvious proof is left to the reader:

Corollary A2.3: A function Q of restrained variation over an interval Ω is also of restrained variation over every subinterval of Ω, and is sum–topped thereon.

(" Sum–topped " is case $k = 1$ of Lemma A2.2.)

This corollary's converse is false: A function can be of restrained variation over two abutting intervals and yet not over their union. A function can be sum–topped but not of restrained variation; $Q(x) := 3 + \cos(x)$ is an example over any interval wider than 2π. But a sum–topped unimodal function is of restrained variation. (A function *unimodal* over an interval Ω has at most one extremum, maximum or minimum, strictly inside Ω .)

Our interest in functions of restrained variation is now mainly historical. In the late 1970s they were the first non–monotonic functions to be recognized as sum–topped; and in practice they are still easier to recognize as such from their splittings than are most other sum–topped functions. Their relevance to Newton's and Secant iteration is apparent in Corollary 8.3.

§A3. Appendix: Projective Images

The redefinition $S(x, x) := N(x)$ connects Newton's iteration $x_{n+1} := N(x_n) := x_n - f(x_n)/f'(x_n)$ to Secant iteration $x_{n+1} := S(x_n, x_{n-1}) := x_n - f(x_n)/f^\dagger(x_n, x_{n-1})$, but not so tightly as they are connected by a shared family of invariants under certain *Projective* transformations. In general, plane projective transformations are those that map straight lines to straight lines. Thus they map tangents to tangents and secants to secants, which is why some of them are pertinent to Newton's and Secant iteration. The pertinent ones constitute a four–parameter family of projective maps each of which takes a pair $\{x, f(x)\}$ to a pair $\{X, F(X)\}$ in such a way that both *Projective Images* $f(x)$ and $F(X)$ are linear functions of their respective arguments, or else neither are. Each of these maps is determined by the values of four constants ß, μ, b and m chosen almost arbitrarily subject to two inequality constraints:

Constraint I: $ç := ßm + bμ \neq 0$, and

Constraint II: b/m does not lie strictly inside the interval Ω in which we seek a zero z of f.

After these constants have been chosen, the projective map $\{x, f(x)\} \Rightarrow \{X, F(X)\}$ and its inverse $\{x, f(x)\} \Leftarrow \{X, F(X)\}$ are defined thus:

$X = \mathbf{X}(x) := (μx + ß)/(b - mx)$, $F(X) := f(\mathbf{x}(X))/(b - m\mathbf{x}(X)) = f(\mathbf{x}(X))(μ + mX)/ç$,

$x = \mathbf{x}(X) := (bX - ß)/(μ + mX)$, $f(x) = F(\mathbf{X}(x))(b - mx) = F(\mathbf{X}(x))ç/(μ + m\mathbf{X}(x))$.

In the last two lines the last equation is derived from the first, which is a *Möbius* (Bilinear–Rational) transformation, with the aid of a valuable identity

$$(b - mx)(μ + mX) = ç \neq 0 .$$

It and Constraint II prevent $b - mx$ from reversing sign while x runs through Ω, and prevent $μ + mX$ from reversing sign while X runs through the interval $\mathbf{X}(Ω)$. Whether this Möbius map preserves or reverses order in those intervals depends upon the sign of ç in Constraint I because the same sign turns up in

$$dX/dx = \mathbf{X}'(x) = ç/(b - mx)^2 \quad \text{and} \quad dx/dX = \mathbf{x}'(X) = ç/(μ + mX)^2 .$$

What do projective images F and f have in common? F has as many zeros strictly inside $\mathbf{X}(Ω)$ as f has strictly inside Ω. (A zero at an end of an interval can evaporate if that end is mapped to ∞; for example consider $f(x) := x$ and $X(x) := -1/x$ for $x \geq 0$, whence $F(X) = -1$ for $X \leq 0$.) Similarly, F and f have the same number of poles strictly inside their intervals. Therein F also has as many *Inflexion–points* (where $F'' = 0$) and *Notches* (where $F'' = ∞$) as f has since $F''(X) = f''(\mathbf{x}(X))ç/(μ + mX)^3$. Other properties F and f share are less obvious.

Under composition, the projective transformations form a *non–Abelian* (non–commutative) *Group* isomorphic to the multiplicative group of nonsingular 2-by-2 matrices. In other words, suppose $\mathbf{X}_j(x) := (μ_j x + ß_j)/(b_j - m_j x)$ for $j = 1, 2, 3$ are the Möbius parts of three projective transformations of which the third is composed from the first and second: $\mathbf{X}_3(x) = \mathbf{X}_2(\mathbf{X}_1(x))$;

then $\begin{bmatrix} b_3 & -m_3 \\ ß_3 & μ_3 \end{bmatrix} = \begin{bmatrix} b_2 & -m_2 \\ ß_2 & μ_2 \end{bmatrix} \cdot \begin{bmatrix} b_1 & -m_1 \\ ß_1 & μ_1 \end{bmatrix}$ and $ç_3 = \det\left(\begin{bmatrix} b_3 & -m_3 \\ ß_3 & μ_3 \end{bmatrix} \right) = ç_2\, ç_1 \neq 0$. In this isomorphism the projective map associated with the constants $\{ß, μ, b, m, ç = bμ + ßm\}$ has an inverse that must be associated with constants respectively $\{-ß/ç, b/ç, μ/ç, -m/ç, 1/ç\}$. Every projective map

can be decomposed into a sequence of at most five maps each selected from a subgroup listed in the following table:

Table 4: Subgroups of Projective Maps {x, f} ‹=› {X, F}

Subgroup Name	**X**(x)	**x**(X)	F(X)	ß	µ	b	m	ç
Scaling	x	X	f(X)/b	0	b	b ≠ 0	0	b^2
Dilation	µ x	X/µ	f(X/µ)	0	µ ≠ 0	1	0	µ
Translation	x + ß	X - ß	f(X-ß)	ß	1	1	0	1
Reciprocal*	1/x	1/X	X f(1/X)	1	0	0	-1	-1

(* The Reciprocal subgroup has two elements including an Identity that changes nothing.)

Often the easiest way to prove an assertion true for all projective maps is to prove it for each subgroup separately and then infer it for their compositions. Often the assertion is unobvious only for the Reciprocal subgroup. Such is the case for the next two assertions:

> **Lemma A3.1:** $Æf := ((f^3 f'')')^2 /(f^5 (f'')^3) = (3 f' f'' + f f''')^2 /(f (f'')^3)$ is invariant for projective maps {x, f(x)} ‹=› {X, F(X)} of nonlinear functions; in other words, nonlinear projective images f(x) and F(X) satisfy Æf(x) = ÆF(X) after substitution of the projective map's Möbius part, say X = **X**(x) .
>
> **Lemma A3.2:** Newton's iterating function **N**f(x) := x - f(x)/f'(x) and Secant iterating function **S**f(x, y) := x - f(x)(x-y)/(f(x) - f(y)) are constructed from f by operators **N** and **S** that commute with projective maps {x, f(x)} ‹=› {X, F(X)} ; in other words, **N**F(**X**(x)) = **X**(**N**f(x)) and **S**F(**X**(x), **X**(y)) = **X**(**S**f(x, y)) wherein X = **X**(x) is the projective map's Möbius part.

The tedious but easy proof of both lemmas is left to the reader. For example, a Negative–Reciprocal projective map {x, f(x)} ‹=› {X, F(X)} defined by X := -1/x and F(X) := Xf(-1/X) has **N**F(X) = X - F(X)/F'(X) = X - Xf(-1/X)/(f(-1/X) + f'(-1/X)/X) = -1/**N**f(x) as claimed in Lemma A3.2. It implies that whether the iterations converge or meander is another invariant of projective maps and motivates us to learn more about them.

The Möbius part of a projective map is determined by what it does to any three distinct values u, v, w of x . It must map them to some three distinct values U, V, W respectively of X , and *vice versa*. It can be constructed from these triples by solving a bilinear *Cross Ratio* equation like

$$(x - u)(v - w)(X - W)(V - U) = (X - U)(V - W)(x - w)(v - u)$$

for either X = **X**(x) or x = **x**(X) , thereby determining the constants ß, µ, b and m except for a common factor. (One member of the triple {u, v, w} can be ∞ if the cross–ratio equation is replaced by an appropriate limit; similarly for {U, V, W} .) The sign of ç = ßm + bµ , which determines whether the Möbius transformation preserves or reverses order, is the same as the sign of

$$(u-v)(v-w)(w-u)/((U-V)(V-W)(W-U)) = ((b - mu)(b - mv)(b - mw))^2/ç^3$$

if both triples {u, v, w} and {U, V, W} are entirely finite. Moreover Constraint II, " b/m does not lie strictly inside the interval Ω ", which ensures that $\mathbf{X}(\Omega)$ be an interval too, requires both triples {u, v, w} and {U, V, W} to have either the same or the opposite linear order whenever u, v and w lie in Ω or, equivalently, whenever U, V and W lie in $\mathbf{X}(\Omega)$.

Many a theoretical problem is simplified by a projective map that transforms a finite interval Ω into a semi–infinite $\mathbf{X}(\Omega)$. One example is the proof of the Intermediate Value Lemma 9.1. Here is another example designed to show why the continuity of Newton's iterating function N is an hypothesis necessary for the conclusion of Theorem 9.2.

•Example A3.3: Twice differentiable function f will strictly increase throughout a finite interval Ω . From any starting point in Ω Newton's iteration $x_{n+1} := N(x_n)$ will always converge in Ω to a zero z of f . On the other hand, Secant iteration won't converge from some starting points in Ω . This ostensible violation of Theorem 9.2 merely violates one of its hypotheses; this example's N is discontinuous at z . This example f(x) is the projective image of a simpler F(X) constructed over the semi–infinite interval $-\infty \le X \le X_0 := 1/\ln(2) = 1.442695$ thus:

For n = 0, 1, 2, 3, … in turn let $X_{3n} := 1/\ln(n+2)$, $X_{3n+1} := (X_{3n} + X_{3n+3})/2$, $X_{3n+2} := -\infty$. Evidently $X_0 \ge X_{3n} > X_{3n+1} > X_{3n+3} > 0$ and $X_{3n} - 2X_{3n+3} + X_{3n+6} > 0$. Next define

$$\begin{aligned} F(X) &:= X \exp(1/X) && \text{if } X < 0 , \\ &:= 0 && \text{if } X = 0 , \\ &:= (X_{3n} - X_{3n+3})/2 && \text{if } X_{3n+1} \le X \le X_{3n} , \text{ and} \\ &:= p_n + q_n\, T((2X - X_{3n+1} - X_{3n+3})/(X_{3n+1} - X_{3n+3})) && \text{if } X_{3n+3} \le X \le X_{3n+1} , \end{aligned}$$

where

$$\begin{aligned} p_n &:= (F(X_{3n+1}) + F(X_{3n+3}))/2 = (X_{3n} - X_{3n+6})/4 > 0 , \\ q_n &:= (F(X_{3n+1}) - F(X_{3n+3}))/2 = (X_{3n} - 2X_{3n+3} + X_{3n+6})/4 > 0 , \text{ and} \\ T(t) &:= \tanh(\tan(\pi t/2)) \quad \text{if } -1 < t < 1 , \\ &:= \operatorname{sign}(t) \quad \text{otherwise.} \end{aligned}$$

T is infinitely differentiable with $T'(t) > 0$ for $-1 < t < 1$ and $T' = 0$ otherwise; $T(\pm 1) = \pm 1$. Consequently the graph of F(X) over $0 < X \le X_0$ looks like a rising staircase with rounded corners and risers and treads that shrink to zero as $X \to 0+$. Between subintervals over which F is constant are subintervals $X_{3n+3} < X < X_{3n+1}$ over which $F'(X) > 0$ and F(X) increases monotonically from $F(X_{3n+3}) = (X_{3n+3} - X_{3n+6})/2$ to $F(X_{3n+1}) = F(X_{3n}) = (X_{3n} - X_{3n+3})/2$ as X increases. In the middle of each such subinterval the derivative F' rises to its local maximum $(\pi/2)(X_{3n} - 2X_{3n+3} + X_{3n+6})/(X_{3n} - X_{3n+3})$, which approaches 0 roughly like 1/n as $n \to +\infty$. Consequently $F(X) \to 0+$ and $F'(X) \to 0+$ roughly like $\exp(-1/X)$ or faster as $X \to 0+$. It soon follows that F(X) is twice differentiable wherever it is defined, namely $-\infty \le X \le X_0$.

The completed definition of Newton's iterating function $\mathbf{N}F(X) := X - F(X)/F'(X)$, including $\mathbf{N}F(0) := 0$, $\mathbf{N}F(-\infty) := -1$, and $\mathbf{N}F(X) := -\infty$ when $X_{3n+1} \le X \le X_{3n}$, remains discontinuous at 0+ because $\mathbf{N}F(X)$ runs from $-\infty$ up to a small positive value and back to $-\infty$ as X runs through each subinterval $X_{3n+3} \le X \le X_{3n}$. None the less, Newton's iteration converges to Z = 0 ultimately monotonically and usually slowly from every starting iterate in the domain of F . But Secant iteration need not converge.

The completed definition of the Secant iterating function $\mathbf{S}F(X, Y) := X - F(X)/F^{\dagger}(X, Y)$ has to include the limiting values $\mathbf{S}F(X, X) := \mathbf{N}F(X)$, $\mathbf{S}F(X, -\infty) := X - F(X)$, and $\mathbf{S}F(X,Y) := -\infty$ when $F(X)-F(Y) = 0 \neq X-Y$. Then $X_{n+1} = \mathbf{S}(X_n, X_{n-1})$ for $n = 1, 2, 3, \ldots$ by design. Starting from X_0 and X_1, every third Secant iterate $X_{3n+2} = -\infty$; thus Secant iteration does not converge although the subsequence of finite Secant iterates converges slowly to $Z = 0$.

To transform the semi–infinite interval $-\infty \le X \le X_0$ into a finite interval $-1 \le x \le 2.5887$ set $x = \mathbf{x}(X) := X/(2-X)$, or $X = \mathbf{X}(x) = 2x/(1+x)$, and $f(x) := (1+x)F(\mathbf{X}(x))$. This projective map turns F into a twice differentiable strictly increasing function f while preserving the iterations' (non)convergence; Newton's iteration converges to $z = 0$ from every starting iterate between -1 and $x_0 := \mathbf{x}(X_0)$ but, starting from x_0 and $x_1 := \mathbf{x}(X_1)$, every third Secant iterate $x_{3n+2} = \mathbf{x}(X_{3n+2}) = -1$. Thus Secant iteration need not converge, though I have proved [1979'] that a subsequence of its iterates always imitates Newton's by converging to z. END EX. A3.3.

• • •

Inverse to the problem of constructing a projective map is the problem of detecting one. Given $f(x)$ and $F(X)$, what test reveals whether they are projective images of each other? An easy test works if they have at least three (but not too many) of the following special points:

zeros, poles, inflexion–points, notches.

For instance, suppose the triple $\{u, v, w\}$ includes one zero and two inflexion–points of f, and $\{U, V, W\}$ does likewise respectively for F; then solving the cross–ratio equation above determines a prospective Möbius transformation $X = \mathbf{X}(x)$ that passes the test if $f(x)/F(\mathbf{X}(x))$ is a linear function, namely $(b - mx)$. If this $\mathbf{X}(x)$ fails the test, all other matching triples of consecutive special points have to be tried and fail too before f and F can be deemed not to be projective images; this is why we hope f and F have not too many special points.

Another test can be fashioned out of Lemma A3.1's projective differential invariant

$$Æf := (3 f'' f' + f''' f)^2 / ((f'')^3 f) \quad .$$

After the substitution $X = \mathbf{X}(x)$ of their Möbius transformation, nonlinear projective images $f(x)$ and $F(X)$ must satisfy $Æf(x) = ÆF(X)$. Conversely, if the equation $Æf(x) = ÆF(X)$ is satisfiable by a Möbius transformation $X = \mathbf{X}(x)$ for which $f(x)^3 f''(x)/(F(\mathbf{X}(x))^3 F''(\mathbf{X}(x)))$ simplifies to a positive constant ($ç^2$) and $f(x)/F(\mathbf{X}(x))$ simplifies to a linear function ($b - mx$), then $f(x)$ and $F(X)$ are projective images. For example $Æf(x) = 12 - 4\ln(x) - 9/\ln(x)$ and $ÆF(X) = 12 + 4\ln(X) + 9/\ln(X)$ when $f(x) = \ln(x)$ and $F(X) = X\ln(X)$, so the equation $Æf(x) = ÆF(X)$ has two solutions $X = \mathbf{X}(x)$ of which only one is a Möbius transformation $\mathbf{X}(x) = 1/x$; next $ç^2 = f(x)^3 f''(x)/(F(\mathbf{X}(x))^3 F''(\mathbf{X}(x))) = 1$ and $(b - mx) = f(x)/F(\mathbf{X}(x)) = -x$, whence $\mu = b = 0$ and $ß = -m = ç = -1$ in the projective map $\{x, f(x)\} \leftrightarrow \{X, F(X)\}$. For another example, the projective map $\{x, (x-1)/x)\} \leftrightarrow \{X, X/(1-X)\}$ can have either of two Möbius parts, either $\mathbf{X}(x) = (x-1)/(-1)$ and $ç = -1$, or $\mathbf{X}(x) = 1/(1-x)$ and $ç = 1$.

This test is complicated slightly by the possibility that infinitely many Möbius transformations may be compatible with a given pair of of projective images. For instance, $Æf = ÆF = 16$ when $f(x) = \exp(x)$ and $F(X) = X\exp(-1/X)$, and then the equation $Æf(x) = ÆF(X)$ is satisfied by all

Möbius transformations $X = \mathbf{X}(x)$; but $ç^2 = f(x)^3 f"(x)/(F(X)^3 F"(X)) = \exp(4x + 4/X)$ is a constant only if $x + 1/X = ô$ is a constant, so the only compatible Möbius transformations are $\mathbf{X}(x) = 1/(ô - x)$, whereupon $(b - mx) = f(x)/F(\mathbf{X}(x)) = e^{ô} (ô - x)$, whence $ß = m = e^{ô}$, $µ = 0$, $b = ôe^{ô}$ and $ç = e^{2ô}$ in projective maps $\{x, f(x)\} ‹=› \{X, F(X)\}$ wherein $ô$ is a parameter. Another one–parameter family of projective maps with Möbius part $\mathbf{X}(x) = \text{constant}/x \neq 0$ has projective images $f(x) = x^k$ and $F(X) = X^{1-k}$ and $Æf = ÆF = 16/(1-1/(2k-1)^2)$ for any constant k . I know no other one–parameter family, nor other projective images with constant $Æf$.

We have seen that $Æf$, and the convergence of Newton's and Secant iterations applied to solve $f(z) = 0$, are invariants of projective maps. Are they related? Is there some condition that $Æf$ can satisfy in an interval $Ω$ to prevent the iterations from meandering in $Ω$ forever? Because $f^3 f" = ç^2 F^3 F"$, another invariant is the sign of $ff"$ if it is constant; it figures in Theorem 7.5. Otherwise monotonicity is not a projective invariant, so neither are Theorem 8.2 nor Corollary 8.3; do invariant versions of them exist?

§A4. Appendix: Parabolas

This Appendix is provided for students who have taken a course on Cartesian Geometry in High School but not yet in College.

> **Lemma A4.1:** Let any nondegenerate triangle's vertices be {Q, R, S}; then one parabola Ç passes through Q and R and is tangent there to sides QS and RS.

Proof: In Cartesian (x, y)–coordinates let the triangle's sides have equations $ax + by + c = 0$ for QS, $Ax + By + C = 0$ for RS, and $ex + fy + g = 0$ for QR. Then for any μ define

$$H_\mu(x, y) := (ax + by + c)\cdot(Ax + By + C) - \mu\cdot(ex + fy + g)^2 .$$

For every choice of the constant μ, the equation $H_\mu(x, y) = 0$ is the equation of a *Conic Section* $Ç_\mu$ (an ellipse, parabola, hyperbola, or pair of straight lines) that passes through Q and R. The set of all such conics $Ç_\mu$ is called a *Pencil* of conics. Every $Ç_\mu$ passes through Q because $(ax + by + c) = (ex + fy + g) = 0$ at Q; similarly $Ç_\mu$ passes through R. Therefore no $Ç_\mu$ degenerates into a single point nor the empty set. The differential

$$dH_\mu(x, y) = (ax + by + c)\cdot(Adx + Bdy) + (Ax + By + C)\cdot(adx + b\,dy) - 2\mu\cdot(ex + fy + g)\cdot(edx + f\,dy)$$

must vanish along $Ç_\mu$; this means that if (x, y) lies on $Ç_\mu$ because $H_\mu(x, y) = 0$, then (dx, dy) points along the tangent to $Ç_\mu$ at (x, y) when $dH_\mu(x, y) = 0$ too. At Q,

$$dH_\mu(x, y) = 0 + (Ax + By + C)\cdot(adx + b\,dy) - 0 = 0 \quad \text{but} \quad (Ax + By + C) \neq 0 ,$$

so $a\,dx + b\,dy = 0$, which means that the tangent to $Ç_\mu$ at Q is parallel to QS; therefore QS is tangent to $Ç_\mu$ at Q. Similarly RS is tangent to $Ç_\mu$ at R.

The next step is to select the lemma's parabola $Ç = Ç_\mu$ from the pencil of conics by choosing the appropriate value for μ. For this purpose $H_\mu(x, y)$ must be expanded:

$$H_\mu(x, y) = (aA - \mu e^2)\cdot x^2 + (aB + bA - 2\mu ef)\cdot xy + (bB - \mu f^2)\cdot y^2 + (\text{terms linear in } x \text{ and } y) .$$

Its *Discriminant*

$$(aB + bA - 2\mu ef)^2 - 4(aA - \mu e^2)\cdot(bB - \mu f^2) = (aB - bA)^2 + 4\mu\cdot(be - af)\cdot(Be - Af)$$

vanishes just when μ takes the finite nonzero value

$$\mu := -(aB - bA)^2 / (4(be - af)\cdot(Be - Af)) .$$

It is finite and nonzero because no two sides of the triangle QRS are parallel, so no factor of μ can vanish. With this choice for μ the vanished discriminant implies that

$$H_\mu(x, y) = \pm(\text{other terms linear in } x \text{ and } y)^2 + (\text{terms linear in } x \text{ and } y) ,$$

so "$H_\mu(x, y) = 0$" is the equation of either a pair of parallel straight lines or a parabola. The pair is ruled out by the intersection of its tangents QS and RS, so $Ç_\mu$ is a parabola. END OF PROOF.

The parabola is a convex curve because it lies entirely on one side of its every tangent, as can be verified easily. The triangle is a convex figure too; and its side QR lies inside the parabola. Therefore an arc of the lemma's parabola Ç stays inside QRS as the arc runs from Q to R. This parabola figures in the proof of Theorem 7.6.

§C. Citations

R. Aharoni, A. Regueros, J. Prashker & D. Mahalel [1992] "A Global Convergence Theorem Result for the One-Dimensional Newton-Raphson and Secant Methods with an Application to Equilibrium Assignment" 17 pp. draft stamped "DEC 03 1992" from the Mathematics & Civil Engineering departments, Technion, Haifa, Israel

R.G. Bartle [1976]*The Elements of Real Analysis* 2d ed., Wiley, New York; pp. 225-227.

A.M. Bruckner & J.G. Ceder [1965] "Darboux Continuity" *Jahresbericht der Deutschen Mathem.–Verein* **67** I. Abt., ss. 93-117

S.D. Conte & C. de Boor [1980] *Elementary Numerical Analysis, an Algorithmic Approach* 3d ed., McGraw-Hill, New York; pp. 62-71.

G. Dahlquist, Å. Björck & N. Anderson [1974] *Numerical Analysis*, Prentice–Hall, New Jersey; pp. 227-232.

R.J. Fateman [1977] "An Algorithm for Deciding the Convergence of the Rational Iteration $x_{n+1} = f(x_n)$ " in *ACM Trans. Math. Software* **3** pp. 272–278.

N.J. Higham [2002] *Accuracy and Stability of Numerical Algorithms* 2d ed., Soc. Indust. & Appl. Math., Philadelphia.

X-C. Huang [1992] "From Intermediate Value Theorem to Chaos" *Mathematics Magazine* 65 #2 (April 1992) pp. 91-103.

W. Kahan [1979'] "No Period Two Implies Convergence, or Why Use Tangents when Secants Will Do?" 69 pp. Memorandum No. UCB/ERL M79/61, 10 Oct. 1979, Electronics Research Lab., Univ. of Calif. @ Berkeley, CA 94720. (Out of print.)

W. Kahan [1979"] "Personal Calculator Has Key to Solve Any Equation f(x) = 0." *Hewlett–Packard Journal* **30** #12 (Dec. 1979) pp. 20–26. The calculator was the hp-34C. A scanned copy is at http://www.cs.berkeley.edu/~wkahan/Math128/SOLVEkey.pdf .

R.E. Martin [1977] "Printing Financial Calculator Sets New Standards for Accuracy and Capability" *Hewlett–Packard Journal* **29** #2 (Oct. 1977) pp. 22–28. The calculator was the hp-92.

P.J. McClellan [1987] "An Equation Solver for a Handheld Calculator" *Hewlett–Packard Journal* **38** #8 (Aug. 1987) pp. 30–34. There were two calculators, the hp-18C and -28C.

M. Misiurewicz [1997] "Remarks on Sharkovsky's Theorem" in *Amer. Math. Monthly* **104** #9 (Nov. 1997) pp. 846-7.

A. Ostrowski [1960] *Solution of Equations and Systems of Equations*, and 2nd ed. [1966] and 3rd ed. [1973], Academic Press, New York; ch. 3.

W.H, Press, S.A. Teukolsky, W.T. Vetterling & B.P. Flannery [1994] *Numerical Recipes in Fortran – the Art of Scientific Computing*, 2d ed. corrected, Cambridge Univ. Press; p. 351.

Y. Saad [1974] "Shifts of Origin for the QR Algorithm" in *Information Processing 74 — Proc. IFIP Congress of Aug. 5-10, 1974, in Stockholm*, pp. 527–531, North Holland Publ., Amsterdam.

A.N. Sharkovsky [1964] "Co-existence of cycles of a continuous mapping of the line into itself" (Russian with English summary) *Ukrain. Math. Zh.* **16** no. 1, pp. 61–71; *Math. Rev.* **28** (1964) #3121. Translated in *Internat'l J'l Bifurc. Chaos Appl. Sci. Engrg.* **5** (1995) pp. 1263-1273.

A.N. Sharkovsky [1965] "On cycles and the structure of a continuous mapping" (Russian) *Ukrain. Math. Zh.* **17** no. 3, pp. 104–111; *Math. Rev.* **32** (1966) #4213.

J. Stoer & R. Bulirsch [1980] *Introduction to Numerical Analysis* translated from the German versions of 1972 and 1976 by R. Bartels, W. Gautschi & C. Witzgall, Springer–Verlag, New York; pp. 274–277.

J.F. Traub [1964] *Iterative Methods for the Solution of Equations*, Prentice-Hall, New Jersey.

H.W. Turnbull [1952] *Theory of Equations* 5th ed., Oliver & Boyd, Edinburgh; pp. 103-107.

M. Vianello & R. Zanovello [1992] "On the Superlinear Convergence of the Secant Method" pp. 758-761 *Amer. Math. Monthly* **99** #8 (Oct.'92) A long proof from minimal hypotheses.

M.V. Wilkes, D.J. Wheeler & S. Gill [1951] *The Preparation of Programs for an Electronic Digital Computer*, Addison Wesley, Cambridge, Mass.; pp. 84-85 and 130-132. The computer was the EDSAC at Cambridge University, England.

G. Wilkins & M. Gu [2013] "A modified Brent's method for finding zeros of functions" in *Numerische Mathematik* **123** pp. 177-188

J.H. Wilkinson [1965] *The Algebraic Eigenproblem*, Oxford Univ. Press; p. 480.