Hyperbolic Interpolation and Iteration towards a Zero

§0: Abstract and Introduction

Given a real function $f(x)$ about which we know how to compute its value, we seek one of its *Zeros* z, a root of the equation $f(z) = 0$, starting from some first guess(es). This z should be the limit of a sequence of presumably improving guesses $x_{n+1} := Hf(x_n, x_{n-1}, x_{n-2})$ computed for $n = 0, 1, 2, 3, \ldots$ in turn by an *Iterating Function* Hf defined below. It will be compared with a few others, and its application to an eigenproblem will be analyzed in detail.

Several assertions, equations and inequalities will be left for diligent readers to reconfirm.

Some iterating functions are derived in §1 from zeros of interpolating functions that match $f(x)$ at two or three points. If a derivative or two of f can be computed too, confluent versions of those iterating functions become available; Newton's and Halley's are two of several offered in §2. How should an iterating function be chosen from the plethora available? Considerations relevant to that choice are explained in §3. Hyperbolic iterating functions seem apposite when f has poles sprinkled among its sought zeros. The close relation demonstrated in §4 between these iterating functions and *Möbius* (bilinear rational) functions helps reveal how a program's languid convergence may be misdiagnosed. This is why §5 chooses *Bi-Confluent Hyperbolic Iteration* to converge to a zero from both sides, thereby combatting both languor and roundoff except, as §6 observes, for slow convergence to a multiple zero. A zero *Hidden* by a pole too nearby in §7 can be found quickly provided it is first straddled. A practical application of the foregoing theory is the solution of the *Spectral* equation of an eigenvalue problem updated by a rank-1 perturbation; §§8 - 11 exhibit the details, including an error-analysis that assesses unavoidable uncertainties due to roundoff. Another application in §12 is another eigenproblem.

Contents

§1: Iterating Functions $Sf(\ldots)$, $Hf(\ldots)$ and $Mf(\ldots)$

If a function f is smooth enough it may be approximated well by the first several terms of its *Newton Interpolating Polynomial Series*

$$f(x) = f(x_0) + (x-x_0)\cdot\big(f^{\dagger}(x_0, x_{-1}) + (x-x_{-1})\cdot\big(f^{\dagger\dagger}(x_0, x_{-1}, x_{-2}) + (x-x_{-2})\cdot\big(f^{\dagger\dagger\dagger}\ldots\big)\big)\big)$$

in which a non-standard notation is being used for *Divided Differences*

$$f^{\dagger}(x, y) := (f(x)-f(y))/(x-y)\,,\quad f^{\dagger\dagger}(w, x, y) := (f^{\dagger}(w, x) - f^{\dagger}(x, y))/(w-y)\,,\quad f^{\dagger\dagger\dagger}\ldots\,.$$

The first two terms' linear polynomial vanishes when $x = Sf(x_0, x_{-1}) := x_0 - f(x_0)/f^{\dagger}(x_0, x_{-1})$. The iteration $x_{n+1} := Sf(x_n, x_{n-1})$ is called *Secant Iteration* and, if it converges to a *Simple* zero z of f (where $f(z) = 0 \ne f'(z)$), converges with *Order* $(1+\sqrt{5})/2 \approx 1.618$, or (rarely) faster. This means that if z, x_{-1} and x_0 agree in sufficiently many leading decimal digits, x_n agrees with z in at least an additional number of decimal digits that grows roughly proportional to $((1+\sqrt{5})/2)^n$ until roundoff in the computation of f *etc.* interferes with convergence.

If, unlike a polynomial, f resembles a rational function with poles scattered among its zeros, f may be better approximated by the first few levels of an *Interpolating Continued Fraction*

$$f(x) = f(x_0) + (x-x_0)/\big(f^{\#}(x_0, x_{-1}) + (x-x_{-1})/\big(f^{\#\#}(x_0, x_{-1}, x_{-2}) + (x-x_{-2})/\big(f^{\#\#\#}\ldots\big)\big)\big)$$

in which a non-standard notation is being used for non-standard *Reciprocal Divided Differences*

$$f^{\#}(x, y) := (x-y)/(f(x) - f(y))\,,\quad f^{\#\#}(w, x, y) := (x-y)/(f^{\#}(w, x) - f^{\#}(w, y))\,,\quad f^{\#\#\#}\ldots\,.$$

The first two levels' continued fraction interpolates (matches) $f(x)$ at $x = x_0$, x_{-1} and x_{-2} :

$$\begin{aligned} Yf(x; x_0, x_{-1}, x_{-2}) &:= f(x_0) + (x-x_0)/\big(f^{\#}(x_0, x_{-1}) + (x-x_{-1})/f^{\#\#}(x_0, x_{-1}, x_{-2})\big) \\ &= f(x_0) + (x-x_0)\cdot f^{\dagger}(x_0,x_{-1})/\big(1 - (x-x_{-1})\cdot f^{\dagger\dagger}(x_0,x_{-1},x_{-2})/f^{\dagger}(x_0,x_{-2})\big)\,. \end{aligned}$$

This interpolant $Yf(x; x_0, x_{-1}, x_{-2})$ vanishes when $x = Hf(x_0, x_{-1}, x_{-2})$ determined thus:

$$\begin{aligned} Hf(u, v, w) &:= u - f(u)\cdot\big(f^{\#}(u, v) + (u-v)/f^{\#\#}(u, v, w)\big)/\big(1 + f(u)/f^{\#\#}(u, v, w)\big) \\ &= u - f(u)/\big(f^{\dagger}(u, v) - f(v)\cdot f^{\dagger\dagger}(u, v, w)/f^{\dagger}(v, w)\big)\,. \end{aligned}$$

The iteration $x_{n+1} := Hf(x_n, x_{n-1}, x_{n-2})$ is called *Hyperbolic Iteration* and, if it converges to a simple zero z of f , converges with Order $\phi \approx 1.839$, the positive root of $\phi^3 = \phi^2 + \phi + 1$, or (rarely) faster. This follows from an important and tediously verifiable identity

$$\frac{Hf(w, x, y) - z}{(w-z)\cdot(x-z)\cdot(y-z)} = Rf(w, x, y) := \frac{f^{\dagger\dagger}(w, x, y)\cdot f^{\dagger\dagger}(x, y, z) - f^{\dagger}(x, y)\cdot f^{\dagger\dagger\dagger}(w, x, y, z)}{f^{\dagger}(w, x)\cdot f^{\dagger}(x, y) - f(x)\cdot f^{\dagger\dagger}(w, x, y)}$$

whose limit, as $w \to z$, $x \to z$ and $y \to z$, implies that as iterates $x_n \to z$ they satisfy

$$(x_{n+1} - z)/\big((x_n - z)\cdot(x_{n-1} - z)\cdot(x_{n-2} - z)\big) \to Rf(z, z, z) = \big(f''(z)^2/4 - f'(z)\cdot f'''(z)/6\big)/f'(z)^2\,,$$

whence a linear recurrence explains why $-\log(|x_n - z|)$ grows at least as fast as ϕ^n as $n \to +\infty$.

Both iterating functions have graphical interpretations. Sf uses a straight line, a secant, that cuts the graph of f twice to approximate it. Hf uses an hyperbola with vertical and horizontal asymptotes that cuts the graph of f thrice to approximate it. Its hyperbola resembles a straight line ever more widely as $f^{\#\#}$ approaches ∞ . ***Caution***: $f^{\#\#}(w, x, y) = f^{\#\#}(w, y, x) \ne f^{\#\#}(x, w, y)$.

Both our iterations can be applied with a complex analytic function $f(x)$ of a complex variable x to find a complex zero z . However, if $f(x)$ is real for all real x , then our iterations require complex initial guesses lest they never converge to a non-real zero z . This requirement can be circumvented by David Muller's iterating function $v = Mf(w, x, y)$ that is the root v nearest w of the quadratic equation $0 = f(w) + (v-w)\big(f^{\dagger}(w, x) + (v-x)\big(f^{\dagger\dagger}(w, x, y)\big)\big)$: $Mf(w,x,y) :=$ $w - 2f(w)/\big(f^{\dagger}(w,x)+(w-x)f^{\dagger\dagger}(w,x,y) \pm \sqrt{\big((f^{\dagger}(w,x)+(w-x)f^{\dagger\dagger}(w,x,y))^2 - 4f(w)f^{\dagger\dagger}(w,x,y) \big)} \big)$. Graphically, Mf uses a parabola that cuts the graph of f thrice to approximate it. Iterating functions Mf and Hf have the same Order $\phi \approx 1.839$ of convergence to a simple zero of f ; but Hf and Sf cannot be expected to converge to a double zero as fast as Mf can.

Note: Like $f^{\dagger}(\ldots)$ and $f^{\dagger\dagger}(\ldots)$, so do $Sf(\ldots)$, $Hf(\ldots)$ and $Rf(\ldots)$ disregard their arguments' order.

§2: Confluent Versions

If computing $f(x)$ and also its derivative $f'(x)$ simultaneously costs not much more time than computing $f(x)$ alone, confluent versions of the foregoing iterating functions become worth considering. Some of these confluent versions' arguments coincide; some divided differences are replaced by derivatives thus: $f^{\dagger}(x, x) = f'(x)$ and $f^{\dagger\dagger}(x, x, x) = f''(x)/2$. The confluent version of Secant Iteration is *Newton's Iteration* $x_{n+1} := Sf(x_n, x_n) = x_n - f(x_n)/f'(x_n)$ which, if convergent to a simple zero, converges with Order 2 or (rarely) faster. Hyperbolic Iteration has five confluent versions: $x_{n+1} := Hf(x_n, x_n, x_{n-1})$ … converges with Order $1 + \sqrt{2} \approx 2.414$; *Halley's Iteration* $x_{n+1} := Hf(x_n, x_n, x_n) = x_n - f(x_n)/\big(f'(x_n) - \frac{1}{2} f(x_n)\cdot f''(x_n)/f'(x_n) \big)$, if …, converges with Order 3 ; three more versions introduced below … converge with order ≥ 2 .

§3: Choosing an Iterating Function

How shall we decide which version to choose? The decision must weigh three considerations listed here in order of difficulty: Cost, Vulnerability to Roundoff, and Appositeness.

Cost is easiest to assess when the accuracy desired is so high as will entail a large number n of iterations of Order $Ø > 1$. They will garner roughly $Ø^n$ correct significant digits at the cost of computing time $T := n\cdot Ç$ where Ç is the cost of one iteration. Ç is assumed the same for all iterations, as is the case when floating-point arithmetic affords just one precision, in which case the iterates' correct digits grow with time T roughly like $(Ø^{1/Ç})^T$. (Otherwise, if arithmetic's precision is variable, the last iteration is likely to cost more than all the others taken together.)

Thus $\mu := \log(Ø)/Ç$ is a rough *Figure-of-Merit*, the bigger the better, for an iterating function of Order Ø and cost Ç per iteration. A. Ostrowski proposed an assay like μ of an iteration's merit in the middle of the 20th century. μ is very rough because it disregards properties of f like values of its derivatives at its zero, so comparing Figures-of-Merit of two iterations may be misleading if they depend upon different derivatives, especially when, as we often hope will happen, adequate accuracy is attained with not very many iterations. We shall try not to be misled when we compare a few iterating functions' Figures-of-Merit.

Let $Ç_1$ be the cost of an iterating function that requires only one new value of f per iteration; let $Ç_2$ be the cost … of f' and f ; let $Ç_3$ be the cost … of f'', f' and f . With very rare exceptions, $Ç_1 < Ç_2 < Ç_3$; and normally $Ç_K < K\cdot Ç_1$. Ostrowski considered polynomials f of degrees so high that $Ç_K \approx K\cdot Ç_1$; these contribute the following table's last column.

Table 1: Iterations' Figures-of-Merit μ

Iteration	Order	Cost	μ	μ for polynomials f
Secant $x_{n+1} := Sf(x_n, x_{n-1})$	1.618	$Ç_1$	$0.4812 / Ç_1$	$0.4812 / Ç_1$
Newton's $x_{n+1} := Sf(x_n, x_n)$	2	$Ç_2$	$0.6931 / Ç_2$	$0.3466 / Ç_1$
Hyperbolic $x_{n+1} := Hf(x_n, x_{n-1}, x_{n-2})$	1.839	$Ç_1$	$0.6094 / Ç_1$	$0.6094 / Ç_1$
$\overline{\text{Confluent}}$ Hyperbolic $x_{n+1} := Hf(x_n, x_{n-1}, x_{n-1})$	2	$Ç_2$	$0.6931 / Ç_2$	$0.3466 / Ç_1$
Confluent Hyperbolic $x_{n+1} := Hf(x_n, x_n, x_{n-1})$	2.414	$Ç_2$	$0.8814 / Ç_2$	$0.4407 / Ç_1$
Bi-Confluent Hyperbolic (to be defined later)	3	$2Ç_2$	$0.5493 / Ç_2$	$0.2747 / Ç_1$
Halley's $x_{n+1} := Hf(x_n, x_n, x_n)$	3	$Ç_3$	$1.0986 / Ç_3$	$0.3662 / Ç_1$

Were Figures-of-Merit μ all that mattered, this table would enact a severe law of diminishing returns from iterations whose higher Order of convergence is achieved at the cost of computing derivatives. For instance, Newton's Iteration would be preferred over Secant Iteration only if f' added less than 44% to the cost of computing f alone. In fact, μ is not all that matters.

Vulnerability to roundoff matters too. Besides limiting how accurately a zero can be computed, roundoff in the computation of f complicates the decision to quit iterating. The last few iterates may waste time dithering in the zero's neighborhood unless roundoff's obscuration is estimated adequately by an error-analysis built into the computation of f. Instead of an error-analysis, an incorporation of f' accurate to a few significant digits helps an iteration to nearly minimize the magnitude of iterates' dithering, thus simplifying the program's decision to quit iterating.

Appositeness of an iterating function like $Sf(\dots)$ or $Hf(\dots)$ reflects how nearly its provenance accords with properties of f, and affects the iteration's behavior while not yet near the sought zero. For instance, if f has a pole near which an iterate may fall, Hyperbolic and Halley's Iterations escape from the pole's neighborhood much faster than Newton's or Secant Iteration can. We should consider also whether an iteration, perhaps modified slightly, can maintain a *Straddle*; this is a pair of iterates, not necessarily consecutive, between which f reverses sign an odd number of times. Newton's and Halley's Iterations require programs more complicated than the other iterations need to maintain a straddle when losing it risks losing the sought zero or converging to an undesired zero. Moreover, when Newton's iterates converge they almost always converge ultimately monotonically (*i.e.*, from one side), whereas the other iterations converge from both sides of a straddle for at least about as many functions f as not. When two-sided convergence is predictable it simplifies criteria for quitting the iteration; it shrinks nested straddles around a zero about as tightly as roundoff allows until it transgresses the last straddle.

The more is known about the properties of f, the better can an iterating function be chosen to take advantage of those properties and also, perhaps, overcome their vitiation by roundoff.

§4: Möbius Maps Commute with Hyperbolic Iterations

Four real constants a, b, c and d determine a *Möbius Function* $M(x) := (a\cdot x - b)/(c\cdot x + d)$, also called a "Linear Fractional" or "Bilinear Rational" function. Lest it degenerate into a constant, its constants must be constrained: $a\cdot d + b\cdot c \neq 0$. Then another Möbius function $W(\xi) := (b + d\cdot\xi)/(a - c\cdot\xi)$ turns out to be inverse to $M(x)$ in the sense that $W(M(x)) = x$ and $M(W(\xi)) = \xi$. These functions' range and domain is best construed as the circle obtained from the real axis after it has been closed by its incorporation of a single point at infinity thus:

$$M(-d/c) = \infty\ , \quad W(\infty) = -d/c\ , \quad M(\infty) = a/c \quad \text{and} \quad W(a/c) = \infty\ .$$

Any three distinct points p, q, r on this circle can be mapped by a suitably constructed Möbius map M to any other distinct three points $\pi = M(p)$, $\kappa = M(q)$, $\rho = M(r)$ resp. by solving a *Bilinear Cross-Ratio Equation*

$$(x-p)\cdot(q-r)\cdot(M(x)-\rho)\cdot(\kappa-\pi) = (M(x)-\pi)\cdot(\kappa-\rho)\cdot(x-r)\cdot(q-p)$$

for $M(x)$ after deleting every factor, if any, that contains ∞. Except across its pole $-d/c$, $M(x)$ is strictly monotonic, increasing if $a\cdot d + b\cdot c > 0$, else decreasing, because the divided difference $M^{\dagger}(x, y) = (a\cdot d + b\cdot c)/\big((c\cdot x+d)\cdot(c\cdot y+d)\big)$ has the same sign as $a\cdot d + b\cdot c$ has.

The set of hyperbolas and straight lines that interpolate $f(x)$ to determine Hyperbolic Iterating Functions $Hf(\ldots)$ is a set mapped to itself by $M(x)$. Consequently these *Commute* thus:

> Given functions $f(x)$ and $M(x)$, determine M's inverse $W(\xi)$, and define $\varphi(\xi) := \psi\cdot f(W(\xi))$ for any constant $\psi \neq 0$. Derive $H\varphi(\omega, \xi, \eta)$ from divided differences of φ just as $Hf(w, x, y)$ is derived from divided differences of f.
>
> Then $H\varphi(M(w), M(x), M(y)) = M(\,Hf(w, x, y)\,)$.

This means changing coordinates from x to $\xi := M(x)$ and from $f(x)$ to $\varphi(\xi) := \psi\cdot f(W(\xi))$ amounts to a Möbius Map that connects $\{x, f(x)\}$ and $\{\xi, \varphi(\xi)\}$ so tightly that Hyperbolic iterates $x_{n+1} = Hf(x_n, \ldots)$ starting from $x_0, \ldots$ and converging to the zero z of f are images of analogous Hyperbolic iterates $\xi_{n+1} = H\varphi(\xi_n, \ldots) = M(x_{n+1})$ starting from $\xi_0 = M(x_0), \ldots$ and converging to the zero $\zeta := M(z)$ of φ. Both sequences of iterates x_n and $\xi_n = M(x_n)$ must converge to their respective destinations at the same speed in the long run.

"In the long run we are all dead." — J.M. Keynes (1883 - 1946)

Long-running computation is what we wish to avoid. We prefer that iteration enter promptly a regime of fast *Superlinear* (Order > 1) convergence that we will terminate soon, as soon as iterates approach the sought zero z about as closely as roundoff allows. Easier said than done.

From an ill-fated start, an iteration can enter instead a regime of languid convergence, creeping through too many iterates before entering an ultimate regime of fast convergence. If a program could detect its embrace by languor the program could attempt an escape. Detection is hindered by an involuntary kind of conspiracy between roundoff and Möbius maps that can squeeze or stretch any interval through which many Hyperbolic iterates travel converging slowly to a zero outside it. In one coordinate system, say $\{\xi, \varphi(\xi)\}$, these iterates are separated widely enough for some regularity in their successive differences to suggest an extrapolation that may escape from languor. In another $\{x, f(x)\}$ coordinate system the iterates appear as crowded as if they were about as near their destination as roundoff allows though actually the destination is not that near at all. We desire that our root-finding program cope well with these phenomena.

§5: Infantile Greed, and Bi- or bi-Confluent Hyperbolic Iteration

We want it all, on the cheap, and soon. Sometimes we get it.

We wish to compute each zero of f at least about as accurately as the data and the arithmetic's precision deserve, but without performing the extra computation of error-analyses to estimate how much accuracy is deserved. And we wish not to wait while iterates grossly contaminated by roundoff dither. Consequently a root-finding program purporting to grant our wishes cannot rely upon an iterative method that will ultimately converge monotonically (from one side) lest short steps confound the program. It cannot distinguish numerous short steps taken to converge very slowly to an unknown destination far away, from short steps taken as iterates drift through a region around a zero where $|f(x)|$ is smaller than its computed value because of roundoff.

Without computing a modest overestimate (like the Error-Analysis in §11 of §8's example) of roundoff's contribution to f, the only way to be about as sure as roundoff allows of a zero's location is to straddle it about as tightly as roundoff allows. To this end a root-finder's iterative method must, after it finds a straddle, maintain it through a nested sequence of tighter straddles that shrink onto a zero inside them until roundoff thwarts the method. This kind of method is the kind we will prefer. However, what "thwarts" means here entails subtleties.

Roundoff cannot thwart an iterative method, if it samples *only* computed values of f (not its derivative nor any other information about it), until the straddle's ends coincide or are adjacent floating-point numbers. To reach this state, if its precision overreaches the accuracy the data deserve, often takes longer than we wish to wait. We will prefer some other method.

Roundoff can thwart an iterative method that would, in the absence of roundoff, *always* shrink a straddle and ultimately shrink it fast (superlinearly). Roundoff thwarts the method when, as computed, an iteration-step would lose the straddle or not shrink it. Then is the time to quit the method. It must depend upon information, like a computable derivative f' and/or assumptions about the smoothness and monotonicity of f, beyond merely how to compute its values. And the method should not so exaggerate the effects of roundoff that iteration is stopped before the sought zero has been located at least about as accurately as it deserves. Such a method can be expected to succeed only under favorable circumstances. Here are two such methods:

Bi- and *bi-Confluent Hyperbolic Iterations* apply the iteration function computed as

$$\mathrm{H}f(y,\ x,x) = \mathrm{H}f(x,x,\ y) := x - f(x)/\big(f'(x) - f(x)\cdot f^{\dagger\dagger}(x,x,y)/f^{\dagger}(x,y)\big)$$

to both sides of a straddle. Suppose u and v straddle f's sought simple zero z but no other zero nor singularity nor zero of f'. Therefore f is strictly monotonic within the straddle, so $f(u)\cdot f(v) < 0$, and $f'(u)$, $f^{\dagger}(u,v)$ and $f'(v)$ all have the same nonzero sign. Computing both

$$t := \mathrm{H}f(u,u,\ v) \quad\text{and}\quad w := \mathrm{H}f(u,\ v,v)$$

will cost very little more than computing either, and both will fall into the straddle between u and v. Appropriate replacement(s) of u and/or v by t and/or w will maintain and shrink the straddle. How shall "Appropriate" be determined? Putting this question's answer into effect constitutes a Bi- or bi-Confluent Hyperbolic Iteration-step.

The obvious answer comes from the computation of both $f(t)$ and $f(w)$, together with $f'(t)$ and $f'(w)$ for use in the next iteration. The computation is wasteful only if $f(t)\cdot f(w) > 0$, in which case either $\{t, f(t), f'(t)\}$ or $\{w, f(w), f'(w)\}$ will be discarded and the other made part of a new straddle nested in and noticeably narrower than the given straddle $\{u, v\}$. Waste like this cannot afflict very many consecutive Bi-Confluent Hyperbolic iteration-steps; ultimately, as is explained hereunder, waste is inhibited by the influence of the ratio $\mathrm{R}f(\dots)$ defined above in §1's tediously verifiable identity, namely …

$$(t-z)/\big((v-z)\cdot(u-z)^2\big) = \mathrm{R}f(u,u, v) = \mathrm{R}f(v, u,u) =$$
$$= \big(f^{\dagger\dagger}(u,u, v)\cdot f^{\dagger\dagger}(u,u, z) - f'(u)\cdot f^{\dagger\dagger\dagger}(u,u, v, z)\big)/\big(f^{\dagger}(u, v)\cdot f'(u) - f(u)\cdot f^{\dagger\dagger}(u,u, v)\big) ;$$

$$(w-z)/\big((u-z)\cdot(v-z)^2\big) = \mathrm{R}f(u, v,v) =$$
$$= \big(f^{\dagger\dagger}(u, v,v)\cdot f^{\dagger\dagger}(v,v, z) - f'(v)\cdot f^{\dagger\dagger\dagger}(u, v,v, z)\big)/\big(f^{\dagger}(u, v)\cdot f'(v) - f(v)\cdot f^{\dagger\dagger}(u, v,v)\big) .$$

Though important, $\mathrm{R}f(\dots)$ is not intended to be computed during the iteration.

$\mathrm{R}f(\dots)$ cannot be infinite; otherwise t and/or w would be thrown out of the straddle, contrary to the assumed monotonicity of f . Therefore $\mathrm{R}f$, like f , has no singularity so long as all its arguments stay within the given straddle. As u and v converge to their respective limits (not yet proved to be coincident) in the course of iteration, $\mathrm{R}f(u,u, v)$ and $\mathrm{R}f(u, v,v)$ converge to their respective finite limits, as do $\mathrm{H}f(u,u, v)$ and $\mathrm{H}f(u, v,v)$ to limits each of which must coincide with a limit of u or of v . Because the assumed strict monotonicity of f bounds $f^{\dagger}$ away from zero, the convergence of $\mathrm{H}f$ to the same limit as one of its arguments (regardless of their order) implies that f converges to zero there; consequently iterates u and v converge to z from opposite sides, and both values of $\mathrm{R}f(\dots)$ converge to $\mathrm{R}f(z,z,z)$.

If $\mathrm{R}f(z,z,z) = 0$, iterates will converge to z so exceptionally fast that the iterates computed will be too few for anyone to care whether some were wasteful in the sense described above.

If $\mathrm{R}f(z,z,z) \neq 0$, its sign will ultimately be matched by the signs of $\mathrm{R}f(u,u, v)$ and $\mathrm{R}f(u, v,v)$, whereupon z will lie between t and w both within the straddle; neither need be wasted. That nonzero sign determines how z and the Bi-Confluent Hyperbolic iterates will be ordered:

When all $\mathrm{R}f(\dots) > 0$, either $u < w < z < t < v$ or $v < t < z < w < u$.

When all $\mathrm{R}f(\dots) < 0$, either $u < t < z < w < v$ or $v < w < z < t < u$.

Either way the iteration produces a sequence of ultimately strictly nested straddles whose ends converge to z with the same Order 3 as Halley's Iteration's. However, unless we value a tight straddle more than a one-sided but usually no worse approximation to a sought zero, the Figure of Merit $\mu \approx 0.5493/Ç_2$ since each Bi-Confluent Hyperbolic iteration-step performs nearly twice as much arithmetic as a merely Confluent one.

This cost can be cut back if $\mathrm{R}f(\dots)$ always has the same nonzero sign deduced in advance from properties of f ; such functions will appear in §8 and §12. Knowledge of that sign permits values of $t := \mathrm{H}f(u,u, v)$ and $w := \mathrm{H}f(u, v,v)$ computed from the ends of a straddle to be put in order with z according to the inequalities displayed above before $f(t)$ or $f(w)$ is computed. Of two nested straddles, each with one end at u or v and the other at t or w , the narrower provides the next *bi-Confluent Hyperbolic* iteration-step. It requires only either $\{f(t), f'(t)\}$ or else $\{f(w), f'(w)\}$ to be computed. Ultimately this iteration converges alternatingly with Order $1 + \sqrt{2} \approx 2.414$ if $\mathrm{R}f(\dots) > 0$, or else Order $\sqrt{(2 + \sqrt{5})} \approx 2.058$ if $\mathrm{R}f(\dots) < 0$.

§6: Linear Convergence to a Multiple Zero

Occasionally an iteration converges far slower than is consistent with its Order of convergence exhibited in Table 1. If we wish to preclude such languor we shall have to understand what can cause it. One possible cause is a violation of the assumption that the sought zero z is Simple.

Functions with multiple zeros are rare, the more so the higher is the multiplicity. On the other hand, functions f with clusters of zeros well separated from the function's other zeros and poles are commonplace. From afar (far enough from both the cluster and all other zeros and poles) the cluster resembles a multiple zero closely enough to retard an iteration's convergence towards a sought zero z until iterates come sufficiently nearer to it than to all the cluster's other zeros. While retarded the iteration's behavior often exposes the cause of languor thus:

While convergence is languid it is roughly *Linear*, which means the ratio $(x_{n+1} - z)/(x_n - z)$ of successive errors approximates a value $\rho \neq 0$ dependent upon only the number m of zeros in the cluster and the choice $Sf(\dots)$ or $Hf(\dots)$ of iterating function. This ρ is the ratio's limiting value as iterates approach any function's zero z of actual multiplicity $m \geq 2$. How ρ depends upon m and that choice of iterating function is summarized below without proofs which, when they exist, can be extremely tedious. Tabulated below for each choice $Sf(\dots)$ and $Hf(\dots)$ is the polynomial of which ρ is its sole zero strictly between 0 and 1. When m is big, $1 - \rho$ is of the order of $1/m$. Tabulated too is the value ρ takes when $m = 2$, which is by far the most common multiplicity m exceeding 1.

Table 2: Iterations' Linear Convergence Ratio ρ to a Zero of Multiplicity m

Iteration	Polynomial	ρ at $m = 2$
Secant $x_{n+1} := Sf(x_n, x_{n-1})$	$\rho^m + \rho^{m-1} - 1$	0.618034
Newton's $x_{n+1} := Sf(x_n, x_n)$	$m \cdot \rho - m + 1$	0.5
Hyperbolic $x_{n+1} := Hf(x_n, x_{n-1}, x_{n-2})$	$(\rho^2 + \rho + 1) \cdot \rho^{m-1} - 1$	0.543689
$\overline{\text{Confluent}}$ Hyperbolic $x_{n+1} := Hf(x_n, x_{n-1}, x_{n-1})$	$m \cdot (\rho^m + \rho^{m-1}) - (\rho^m - 1)/(\rho - 1)$	0.5
Confluent Hyperbolic $x_{n+1} := Hf(x_n, x_n, x_{n-1})$	$\rho^m + 2\rho \cdot (\rho^{m-1} - 1)/(\rho - 1) - m + 1$	0.414214
Halley's $x_{n+1} := Hf(x_n, x_n, x_n)$	$(m+1) \cdot \rho - m + 1$	0.333333

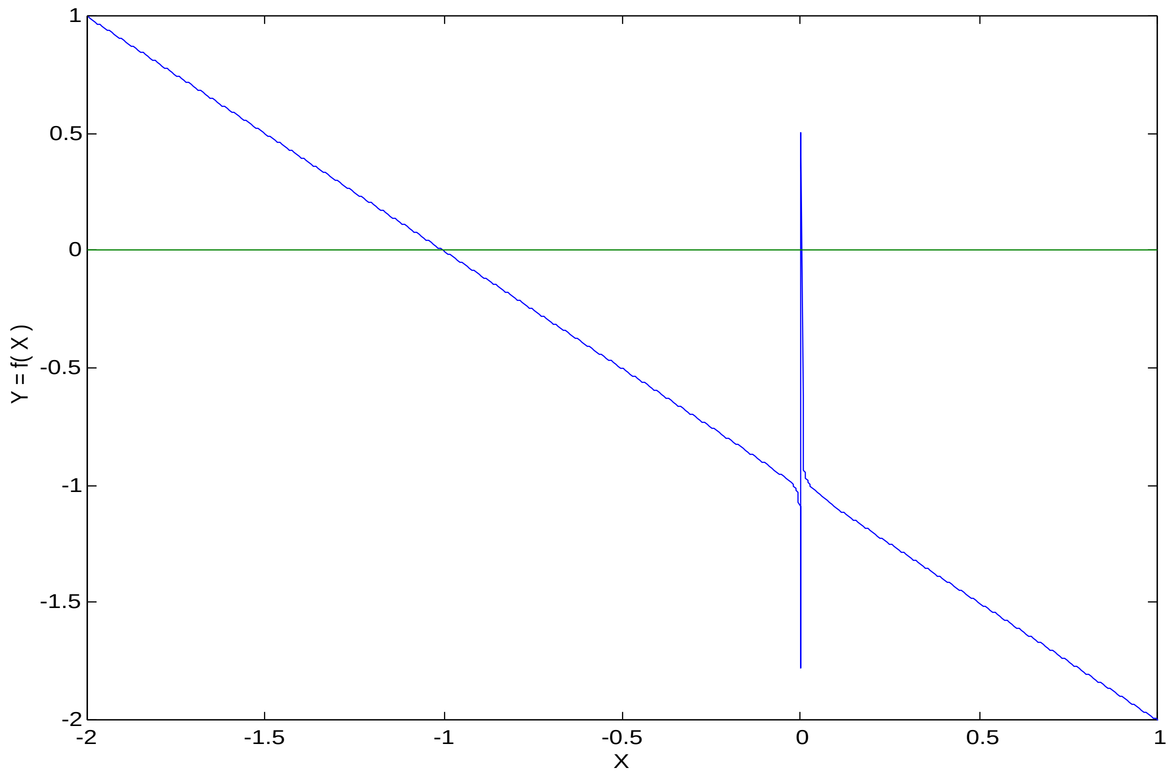
Unfortunately, languid convergence can be caused by other than clustered or multiple zeros.

Table 2 omits Bi-Confluent Hyperbolic Iteration because it requires a straddle unobtainable for a double root. Whether and how to accelerate languid convergence to a cluster are discussed in §7 and §10 of my *Lecture Notes on Real Root-Finding* posted at <www.cs.berkeley.edu/~wkahan/Math128/RealRoots.pdf>.

§7: Languid Convergence to a Hidden Zero

Call z a *Hidden Zero* of f when it has a pole so close to the zero that they almost cancel out. An example $f(x) := (z - x) \cdot (1 + 1/x)$ has a pole at $x = 0$ that almost cancels the zero z if it is tiny enough. As $z \to 0+$ the graph of $y = f(x)$ approaches the union of two straight lines, one the graph of $y = -(x + 1)$, the other a vertical line along the y-axis, as shown hereunder:

$f(x) = (z - x)\cdot(1 + 1/x)$, $z = 0.0005$



More generally, a plotted graph of a function f with a zero Hidden by a simple pole nearby will fail (except by accident) to reveal the zero unless the plotted points are dense enough to reveal that a nearly straight vertical line crosses a graph otherwise almost unexceptional.

Hyperbolic iterations appear called for to cope with the pole adjacent to a sought Hidden zero, though it is unlikely to be found by these or any other iterations unless they start from a straddle and maintain it. Halley's iteration begun between the adjacent pole and zero converges well unless an iterate jumps a little too far past the sought zero, whereupon subsequent iterates will go elsewhere. A program can inhibit Halley's divagation by including *ad hoc* expedients like retractions at the risk of needing so many of them that they retard convergence.

No *ad hoc* expedients are needed for the other two confluent Hyperbolic iterations to maintain a straddle. They do so whenever $f(x)$ is strictly monotonic within it: If $f(u)\cdot f(v) < 0$ and all three of $f'(u)$, $f^{\dagger}(u,v)$ and $f'(v)$ have the same sign, then $Hf(u,u,v)$ and $Hf(u,v,v)$ both lie strictly between u and v which, after the appropriate replacement(s), give way to a tighter straddle. Nested straddles will ultimately confine the sought zero as tightly as roundoff allows. This desirable behavior comes at a price: Occasionally these iterations can tarry in a regime of languid convergence. The example f graphed above will illustrate how this languor can occur:

The Confluent Hyperbolic iterating function in question, namely

$$Hf(y, x,x) = Hf(x,x, y) := x - f(x)/\big(f'(x) - f(x)\cdot f^{\dagger\dagger}(x,x,y)/f^{\dagger}(x,y) \big) ,$$

will be applied from each side to the foregoing example's function $f(x) := (z - x)\cdot(1 + 1/x)$ with $0 < z << 1$; but we must assume that no more is known about z than a straddle $0 < u < z < v$.

Halley's iterates Hf(x,x,x) jump over z . From the far right Hf(v,v,v) jumps beyond the pole too; $Hf(v,v,v) < 0$ unless $z < v < (\sqrt[3]{z})^2/(1 - \sqrt[3]{z})$, so Hf(v,v,v) must be retracted unless v on the right of z is extremely near it. From the near left, from u between z and the pole,

$$0 < u < z < Hf(u,u,u) = z + (z–u)^3/\big(u^3 + 3u·z + z·(1–z)\big) < Hf(0,0,0) = z + z^2/(1–z) ,$$

which is excellent. If Halley's iteration starts from the near left it will converge alternatingly and rapidly for this simple example. Other more complicated examples' first iterates can jump from between z and the pole far enough past z that their second iterates jump back beyond the pole and must be retracted lest the straddle be lost. Hidden zeros tend to hide from Halley.

Both Confluent iterates t := Hf(u,u, v) and w := Hf(u, v,v) maintain the straddle; moreover neither jumps past z for this simple example whose $Rf(z,z,z) \approx –1/z < 0$. Keeping in mind that $0 < z << 1$ and $0 < u < z < v$, we find that, coming from v far on the right,

$$v > Hf(u, v,v) = z + (1 – u/z)·(v–z)^2/(1 + 2v + v^2·u/z + u – z)$$
$$\approx v·(1 – u/z)/(2 + v·u/z + 1/v) \quad \text{when } v >> \sqrt{z} .$$

This means that iteration solely from the right, starting afar from $v >> \sqrt{z}$, shrinks the straddle repeatedly by factors somewhat smaller than $(1 – \sqrt{u/z})/2$ until abruptly convergence becomes quadratic. Coming from u on the near left,

$$u < Hf(u,u, v) = z – (z–u)^2·(v–z)/\big((z+u^2)·v + (1 – z + 2u)·z\big)$$
$$\approx z·\big(1 – (1 – u/z)^2·v/(v+1) \big) \quad \text{when } v >> z ,$$

which implies that iteration solely from the left converges quadratically up to the tiny hidden zero z without ever shrinking the straddle much.

Does this example's behavior adumbrate what happens with Hidden zeros generally?

If so, it bodes ill for bi-Confluent Hyperbolic Iteration because its convergence is retarded by its greedy choice of the narrower straddle. Bi-Confluent Hyperbolic Iteration converges to a Hidden zero so much faster that it seems preferable over all other iterative methods whenever a Hidden zero is likely to hide among those that are sought. This preference persists despite that computing both {t, *ƒ*(t), *ƒ*'(t)} and {w, *ƒ*(w), *ƒ*'(w)} seems costly, or at least wasteful of the one not retained. Because both characterize *ƒ*(…) from the same data, computing both costs appreciably less than twice the cost of computing either on most of today's computers, which take longer to fetch data from memory than to perform pipelined arithmetic upon them.

However, Bi-Confluent Hyperbolic Iteration converges to a Hidden zero only after it has been straddled. How much should be known about a function *ƒ* for its Hidden zeros to be straddled more reliably than by chance? At least the poles of *ƒ* that may hide its Hidden zeros should be easy to find. Therefore the question that deserves to be considered is actually …

Are there functions *ƒ* whose Hidden zeros are hidden only by poles easy to find?

Such functions will appear in §8 and §12. They were not contrived. They are not artificial.

"Man is a tool-using animal." — Thomas Carlyle (1795 - 1881).
This explains why a man holding a hammer must find nails to hit.

§8: Example: A *Spectral* Equation (also called a *Secular* Equation)

Here is an example of an equation $f(z) = 0$ which Hyperbolic Iteration is apposite to solve:

Suppose we seek all K eigenvalues ξ_j of a real symmetric matrix $X := V + \alpha\cdot\mathbf{c}\cdot\mathbf{c}'$ differing by a matrix of rank 1 from another real symmetric matrix V whose eigensystem is given; given also are scalar $\alpha \neq 0$ and column $\mathbf{c} \neq \mathbf{o}$. The eigenvalues sought are the zeros $\xi = \xi_j$ of the *Characteristic Polynomial* $\det(\xi\cdot I - X)$, which turns out to cost much more to compute than does the ratio $\det(\xi\cdot I - X)/\det(\xi\cdot I - V)$ of characteristic polynomials. Consequently our task reduces to the numerical computation of all roots ξ_j of a *Spectral Equation* $f(\xi) = 0$ whose $f(\xi) := \sum_k \gamma_k^2/(\xi - \lambda_k) - 1/\alpha$ with given values of $\alpha \neq 0$, $\gamma_k \neq 0$ and eigenvalues λ_k of V all distinct and sorted so that $\lambda_k < \lambda_{k+1}$. Each root ξ_j is known to lie in an open interval Ξ_j one of whose endpoints is λ_j; the other is the nearer of $\lambda_j + \alpha\cdot\sum_k \gamma_k^2$ and $\lambda_{j+\mathrm{sign}(\alpha)}$ (when it exists). Every root is Simple since the derivative $f'(\xi) = -\sum_k \gamma_k^2/(\xi - \lambda_k)^2 < 0$; moreover $f'(\xi_j) < -1/(\alpha^2\cdot\sum_k \gamma_k^2)$, so every root ξ_j is determined sharply by the Spectral Equation's f. Though no root is repeated, some can be Hidden in the sense explored in §7 above.

Alas, the given data can come almost arbitrarily close to pathological: α can be arbitrarily tiny or big; distinct values λ_j can agree in all but their last digits; and coefficients γ_k^2 can vary arbitrarily widely unless each γ_k tinier than roundoff in $\gamma_{k\pm1}$ is reset to zero and *deflated* out. In consequence of a near-pathology, an interval Ξ_j can be arbitrarily narrow, or its ξ_j can be arbitrarily nearly Hidden. Pathology and roundoff conspire sometimes to render a numerical computation unreliable. The conspiracy can usually be thwarted by artful origin-shifts, but these are a tiresome story for another document.

Divisions dominate the cost of computing $f(\xi)$ on computers that lack a pipeline dedicated to divisions. Whatever that cost, it is less than doubled by the cost of computing the derivative $f'(\xi)$ too. However, because ∞ is an attractive fixed-point for Newton's and Secant iterating functions $Sf(\dots)$, these are ill-suited to solving our Spectral Equation unless started close enough to a sought zero. Confluent Hyperbolic Iterations can generate iterates that converge to ξ_j rapidly from both sides when started from a straddle in Ξ_j, as we shall see. The second derivative $f''(\xi)$ would add no more than $f'(\xi)$ did to each iteration's cost and would enable Halley's Iteration; but this must be started close enough to the sought zero or else occasionally inhibited by retractions of iterates that would escape from the current straddle. Flurries of these retractions, by delaying convergence to some zeros, especially Hidden zeros, can complicate load-balancing severely enough to impede parallel computation, for which all eigenvalues are otherwise eminently eligible since each is computable with no reference to any other.

§9: How Straddling is Maintained

Subscript j will be dropped from ξ_j and Ξ_j to allay notational clutter in the explanation here. Now $f(\xi) = 0$ and ξ lies in an open interval Ξ either between two adjacent poles λ_k of f or else outside all poles. How do the properties of the Spectral Equation's f and of Hf combine to maintain straddling? Suppose Ξ contains three points w, x and y, maybe not all distinct, of which two straddle ξ. These two straddle $v := Hf(w, x, y)$ too because $f' < 0$ and $f^\dagger < 0$.

On which side of ξ will v lie? An answer comes out of a close examination of f and $\mathrm{H}f$. Start by dissecting $\mathrm{R}f(w, x, y) := \big(\mathrm{H}f(w, x, y) - \xi\big)/\big((w-\xi)\cdot(x-\xi)\cdot(y-\xi)\big)$ with the aid of the tediously verifiable identity in §1 to find for $\mathrm{R}f(w, x, y) =: -\mathrm{Rn}(w, x, y)/\mathrm{Rd}(w, x, y)$ that

$$\mathrm{Rn}(w, x, y) := f^{\dagger}(x, y)\cdot f^{\dagger\dagger\dagger}(w, x, y, \xi) - f^{\dagger\dagger}(w, x, y)\cdot f^{\dagger\dagger}(x, y, \xi)$$
$$= \textstyle\sum_i \sum_{k>i} \gamma_i^2\cdot\gamma_k^2\cdot(\lambda_i-\lambda_k)^2/\big((\xi-\lambda_i)\cdot(w-\lambda_i)\cdot(x-\lambda_i)\cdot(y-\lambda_i)\cdot(\xi-\lambda_k)\cdot(w-\lambda_k)\cdot(x-\lambda_k)\cdot(y-\lambda_k)\big)$$
$$> 0 \quad \text{and}$$
$$\mathrm{Rd}(w, x, y) := f^{\dagger}(w, x)\cdot f^{\dagger}(x, y) - (x-\xi)\cdot f^{\dagger}(x, \xi)\cdot f^{\dagger\dagger}(w, x, y) > 0 .$$

The last rather unobvious inequality follows from our supposition above about a straddle plus three observations: First, $\mathrm{Rd}(w, \xi, y) > 0$. Second, $\mathrm{Rd}(w, x, y)$ is a continuous function of its arguments and independent of their order. Third, while two of w, x and y in Ξ straddle ξ, $\mathrm{Rd}(w, x, y) \ne 0$; otherwise $v := \mathrm{H}f(w, x, y)$ could be thrown out of Ξ instead of falling inside the straddle along with ξ. Taken together those inequalities about Rn and Rd imply that

$$(v-\xi)/\big((w-\xi)\cdot(x-\xi)\cdot(y-\xi)\big) = \mathrm{R}f(w, x, y) < 0 .$$

This inequality $\mathrm{R}f(w, x, y) < 0$ and the prerequisites for its validity are important. These are …

- w, x and y (not necessarily all distinct) lie in Ξ along with ξ, and
- Either two of w, x and y straddle ξ, or it is close enough to one of them.

(Amply many examples dissatisfy the second prerequisite and throw $\mathrm{H}f(w, x, y)$ out of Ξ .)

The first of two inferences from the inequality $\mathrm{R}f < 0$ is that, so long as both its prerequisites hold for *all* the iterates x_n of a (perhaps Confluent) Hyperbolic Iteration, a persistent pattern of signs of $x_n - \xi$ determines the maximum number $\overline{m}$ of consecutive iterates that can lie on the same side of ξ. Here is a tabulation of $\overline{m}$ and that pattern:

Hyperbolic Iterations		Fig.-Merit μ	$\overline{m}$	Sign($x_n - \xi$) Patterns
Hyperbolic	$x_{n+1} := \mathrm{H}f(x_n, x_{n-1}, x_{n-2})$	0.6094 / Ç$_1$	3	… + + + – + + + – … or … – – – + – – – + …
$\overline{\text{Confluent}}$ Hyperbolic	$x_{n+1} := \mathrm{H}f(x_{n-1}, x_{n-1}, x_n)$	0.6931 / Ç$_2$	1	… + – + – + – + – …
Confluent Hyperbolic	$x_{n+1} := \mathrm{H}f(x_n, x_n, x_{n-1})$	0.8814 / Ç$_2$	2	… + + – – + + – – …
Halley’s	$x_{n+1} := \mathrm{H}f(x_n, x_n, x_n)$	1.0986 / Ç$_3$	1	… + – + – + – + – …

This table explains how Hyperbolic iterates that *all* satisfy both prerequisites straddle ξ and converge to it from both sides until roundoff interferes. $\overline{\text{Confluent}}$ Hyperbolic Iteration always satisfies both prerequisites if started from a straddle, but converges a little slower than do the others. Their faster convergence could be enjoyed if we could answer an interesting question:

How can we predict from some few of a chosen Iteration’s iterates whether
all subsequent iterates will satisfy both of the prerequisites that keep $\mathrm{R}f < 0$?

Prediction is obvious only for $\overline{\text{Confluent}}$ Hyperbolic Iteration only if it starts from a straddle.

For other Hyperbolic Iterations, prediction may become practicable if the following conjecture is true:

Provided w, x and y satisfy both prerequisites for $\mathrm{R}f(w, x, y) < 0$,
whenever any one of them moves towards ξ, so does $v := \mathrm{H}f(w, x, y)$.

Until this conjecture is decided I am inclined to eschew non-Confluent Hyperbolic and Halley’s Iterations.

A second inference from the inequality $\mathrm{R}f < 0$ is that since $\mathrm{R}f(u,u,v) < 0$ and $\mathrm{R}f(u,v,v) < 0$ whenever u and v straddle ξ, then we can predict $\mathrm{sign}(f(\mathrm{H}f(u,u,v))) = \mathrm{sign}(f(u))$ and $\mathrm{sign}(f(\mathrm{H}f(u,v,v))) = \mathrm{sign}(f(v))$. These predictions figure in straddle-maintaining Bi-Confluent Hyperbolic Iterations that converge to a Hidden zero ξ never slower than quadratically, we hope, and converge ultimately with Order 3 and Figure-of-Merit $\mu \ge 0.5493/Ç_2$.

Of course, the program must allow for predictions flouted by roundoff. It must terminate the iteration when, because of roundoff, neither prospective computed replacement $t := \mathrm{H}f(u,u,v)$ for u nor $w := \mathrm{H}f(u,v,v)$ for v could narrow the straddle. And the program has to find an initial straddle to initiate Bi-Confluent Hyperbolic Iteration.

§10: Where to Find the Initial Straddle

Bi-Confluent Hyperbolic Iteration needs initial points u and v in Ξ_j and on opposite sides of ξ_j. Toward this end, closed-form solutions of the Spectral Equations of 2-by-2 matrices X will be introduced: Given values of just $\alpha \ne 0$, $\gamma_1^2 > 0$, $\gamma_2^2 > 0$, and $\lambda_1 \ne \lambda_2$, the two zeros $\zeta_{\dots}$ of $\gamma_1^2/(\zeta-\lambda_1) + \gamma_2^2/(\zeta-\lambda_2) - 1/\alpha$ are the functions …

$$\zeta_O(\alpha, \gamma_1^2, \lambda_1, \gamma_2^2, \lambda_2) := (\lambda_1+\lambda_2+\alpha\cdot(\gamma_1^2+\gamma_2^2) + \mathrm{sign}(\alpha)\cdot\sqrt{\Delta})/2 \quad \text{and}$$
$$\zeta_B(\alpha, \gamma_1^2, \lambda_1, \gamma_2^2, \lambda_2) := (\lambda_1+\lambda_2+\alpha\cdot(\gamma_1^2+\gamma_2^2) - \mathrm{sign}(\alpha)\cdot\sqrt{\Delta})/2 \quad \text{wherein}$$
$$\text{discriminant} \quad \Delta := (\lambda_1-\lambda_2+\alpha\cdot(\gamma_1^2-\gamma_2^2))^2 + 4\alpha^2\cdot\gamma_1^2\cdot\gamma_2^2 > 0\,.$$

The names "ζ_O" and "ζ_B" have been so chosen because ζ_B lies strictly *Between* λ_1 and λ_2 whereas ζ_O lies strictly *Outside* them. When ζ_O and ζ_B have very different magnitudes the smaller is best computed after the bigger from the identity $\zeta_O\cdot\zeta_B = \lambda_1\cdot\lambda_2 + \alpha\cdot(\lambda_1\cdot\gamma_2^2 + \lambda_2\cdot\gamma_1^2)$ to lessen contamination by roundoff. What follows uses both functions $\zeta_O(\dots)$ and $\zeta_B(\dots)$.

Back to the original task: Bi-Confluent Hyperbolic Iteration needs two initial iterates u and v in Ξ_j but on opposite sides of an eigenvalue ξ_j of X, whose dimension now exceeds 2. Two Cases (B and O) shall be distinguished according to whether Ξ_j lies Between λ_j and $\lambda_{j+\mathrm{sign}(\alpha)}$, or else Outside the narrowest interval containing all the eigenvalues $\lambda_{\dots}$ of V.

Case B: When Ξ_j lies Between λ_j and $\lambda_{j+\mathrm{sign}(\alpha)}$.
This is the more common Case. It entails the computation of two nonempty sums-of-squares

$$\sigma_{«} := \sum(\gamma_k^2 \text{ over indices } k \text{ for which } \alpha\cdot(k-j) \le 0) \quad \text{and}$$
$$\sigma_{»} := \sum(\gamma_k^2 \text{ over indices } k \text{ for which } \alpha\cdot(k-j) > 0)\,.$$

In other words, $\sigma_{«}$ sums γ_k^2 for all indices k on the same side of $j+\mathrm{sign}(\alpha)$ as is j, and $\sigma_{»}$ sums γ_k^2 for all indices k on the same side of j as is $j+\mathrm{sign}(\alpha)$. Then ξ_j turns out to lie strictly between $\zeta_B(\alpha, \sigma_{«}, \lambda_j, \gamma_{j+\mathrm{sign}(\alpha)}{}^2, \lambda_{j+\mathrm{sign}(\alpha)})$ and $\zeta_B(\alpha, \gamma_j^2, \lambda_j, \sigma_{»}, \lambda_{j+\mathrm{sign}(\alpha)})$, both of which fall inside Ξ_j too.

Case O: When Ξ_j lies Outside the narrowest interval containing all eigenvalues $\lambda_{\dots}$ of V. This Case arises only when λ_j is the least (if $\alpha < 0$) or largest (if $\alpha > 0$) eigenvalue of V. Let $\sigma := \sum_{k\neq j} \gamma_k^2$; this sums γ_k^2 for all indices k with λ_k on the side of λ_j opposite ξ_j, which lies strictly between $\zeta_O(\alpha, \gamma_j^2, \lambda_j, \gamma_{j-\mathrm{sign}(\alpha)}^2, \lambda_{j-\mathrm{sign}(\alpha)})$ and $\zeta_O(\alpha, \gamma_j^2, \lambda_j, \sigma, \lambda_{j-\mathrm{sign}(\alpha)})$. Both of these also fall inside Ξ_j, which is the interval strictly between λ_j and $\lambda_j + \alpha\cdot\sum_k \gamma_k^2$.

. . . .

The relevant Case provides two estimates $\zeta_{\dots}(\dots)$ inside Ξ_j that straddle ξ_j. Neither of the two need be close to ξ_j, especially if $\gamma_{j+\mathrm{sign}(\alpha)}^2$ and γ_j^2 are tiny compared with $\sum_k \gamma_k^2$, in which case each estimate $\zeta_{\dots}(\dots)$ may lie so close to its end of Ξ_j that too many Hyperbolic Iterations will be squandered to get all iterates far enough away from those ends' poles that fast convergence can commence. To defend against this contingency, form a third estimate ζ from the average of the two estimates $\zeta_{\dots}(\dots)$ and, after examining the sign of $f(\zeta)$, keep ζ and whichever of the two estimates $\zeta\dots(\dots)$ lies on the other side of ξ_j to serve as the two initial iterates u and v. Subsequent Bi-Confluent Hyperbolic Iterations will maintain the straddle and shrink it rapidly onto ξ_j, even if it is Hidden, until roundoff retards further shrinkage. The iteration's stopping criterion takes roundoff into account without mentioning it explicitly:

Unless $f(u) = 0$ or $f(v) = 0$, in which case there is no need for more,
end iteration as soon as neither computed $Hf(u,u, v)$ nor $H(u, v,v)$
falls strictly between u and v, and then assign to ξ_j whichever
of these two minimizes the computed $|f(\xi_j)|$.

§11: Error Analysis of the Roots of the Spectral Equation $f(\xi) = 0$

How uncertain is a computed root ξ_J because of roundoff? Uncertainty is engendered almost entirely by roundoff's contribution to the computed value of $f(\xi) := \sum_k \gamma_k^2/(\xi - \lambda_k) - 1/\alpha$.

Recall that its data are $\alpha \neq 0$ and K values $\gamma_k \neq 0$, and K values λ_k all distinct and sorted so $\lambda_k < \lambda_{k+1}$. Each root ξ_J is known to lie in an open interval Ξ_J one of whose endpoints is λ_J; the other is the nearer of $\lambda_{J+\mathrm{sign}(\alpha)}$ and $\lambda_J + \alpha\cdot\sum_k \gamma_k^2$ subject to the understanding that $\lambda_0 := -\infty$ and $\lambda_{K+1} := +\infty$. Every root ξ_J is Simple because the derivative $f'(\xi) = -\sum_k \gamma_k^2/(\xi - \lambda_k)^2 < 0$; moreover $f'(\xi_J) < -1/(\alpha^2\cdot\sum_k \gamma_k^2)$, so every root ξ_J is determined sharply by the Spectral Equation's f. In fact, f is monotone non-increasing in each Ξ_J despite roundoff.

Alas, the given data can come almost arbitrarily close to pathological: α can be arbitrarily tiny or big; distinct values λ_k may agree in all but their last digits; and the coefficients γ_k^2 can range almost arbitrarily widely unless each γ_k tinier than roundoff in $\gamma_{k\pm1}$ is reset to zero and *Deflated* out. In consequence of a near-pathology, an interval Ξ_J can be arbitrarily narrow, or its ξ_J can come arbitrarily close to either endpoint where a pole may Hide the zero. Pathology and roundoff can conspire to render the numerical computation of ξ_J unreliable.

The analyses hereunder will help anyone who wishes to assess a computed zero's uncertainty due to roundoff and the effectiveness of an attempt to combat it by *Deflation* (described below).

A reasonable bound upon roundoff's contribution is $e(\xi) := æ\cdot\sum_k \gamma_k^2/|\xi - \lambda_k|$ wherein æ is a modest multiple of a roundoff threshold $1.000\ldots001 - 1$ like MATLAB's `eps` . This "modest multiple" varies with matrix X 's dimension K , which is the number of terms in $\sum_k$. The multiple depends on how summation is programmed and runs from 1 through $\log_2(K)$ to K ; roughly $\sqrt{K}$ is probably satisfactory. The consequent uncertainty in any computed value of ξ at which $f(\xi) \approx 0$ is about $ü(\xi) := e(\xi)/|f'(\xi)| = æ\cdot(\sum_k \gamma_k^2/|\xi - \lambda_k|)/\sum_k \gamma_k^2/(\xi - \lambda_k)^2$. In this formula α figures indirectly by virtue of its influence upon ξ . Thus $ü(\xi_J)/æ$ is a computable positively weighted harmonic mean of all the gaps $|\xi_J - \lambda_k|$ between ξ_J and the poles λ_k .

We can tolerate this much uncertainty ü easily if *absolute* error in ξ_J is our sole concern.

However, in order to compute every eigenvector $\mathbf{x}_J$ of X accurately enough from the simple formula $\mathbf{x}_J := \text{Column}(\{\gamma_k/(\xi_J - \lambda_k)\})$, all the gaps' *relative uncertainties* must be kept small. To this end we might endeavor to keep both $ü(\xi_J)/(æ\cdot|\xi_J - \lambda_j|)$ and $ü(\xi_J)/(æ\cdot|\xi_J - \lambda_{J+\text{sign}(\alpha)}|)$ small enough.

This endeavor must fail occasionally when ξ_J is too nearly Hidden by a pole $\lambda_{\ldots}$.

Before the failure is explained some simplifying assumptions and abbreviations will be invoked: First, for any given J , let Î be the subscript of the pole $\lambda_{Î}$ nearest ξ_J so that either Î = J or Î = J + sign(α) . Next assign abbreviations …

$$c_k := \gamma_k^2/\gamma_{Î}^2 > 0\ ;\quad r_k(\xi_J) := |\xi_J - \lambda_{Î}|/|\xi_J - \lambda_k|\ ;\quad \textstyle\sum' := \sum_{k\neq Î}\ ;\ \text{and}\quad q(\xi_J) := ü(\xi_J)/(æ\cdot|\xi_J - \lambda_{Î}|)\ .$$

Now $æ\cdot q(\xi_J)$ is the *relative uncertainty* in the smallest gap $\xi_J - \lambda_{Î}$. Drop (ξ_J) to get simply $q = (1 + \sum' c_k\cdot r_k)/(1 + \sum' c_k\cdot r_k^2)$. Note that $0 < r_k \le r_{Î} = 1$ for all k ; consequently $q > 1$.

How big can q *not* be?

In the absence of constraints upon the data α , $\{\lambda_k\}$ and $\{\gamma_k\}$ beyond those assumed so far, q can be arbitrarily bigger than 1 , as happen to a small K-by-K matrix example with

$$K = 3\ ,\quad J = 2\ ,\quad \lambda_2 = 0\ ,\quad \alpha = \gamma_2 = \xi_2 = 1\ ,\quad \lambda_1 = 1 - 2q\ ,\quad \lambda_3 = 1 + 2q\ ,\quad \gamma_1 = \gamma_3 = \sqrt{2q\cdot(q-1)}\ .$$

This small example's uncertainty $ü(\xi_2) = q\cdot æ\cdot|\xi_2 - \lambda_2|$ for any chosen q no matter how big.

More generally, for arbitrary dimensions and data,

$$q = ü(\xi_J)/(æ\cdot|\xi_J - \lambda_{Î}|) \le \tfrac{1}{2}(1 + \sqrt{(\textstyle\sum_k \gamma_k^2)}/|\gamma_{Î}|)\ ,$$

and the upper bound is achievable. This inequality's proof starts from Cauchy's inequality, which implies

$$q = (1 + \textstyle\sum' c_k\cdot r_k)/(1 + \sum' c_k\cdot r_k^2) \le \bar{q} := (1 + \sqrt{(\sum' c_k\cdot\sum' c_k\cdot r_k^2)})/(1 + \sum' c_k\cdot r_k^2)\ ,$$

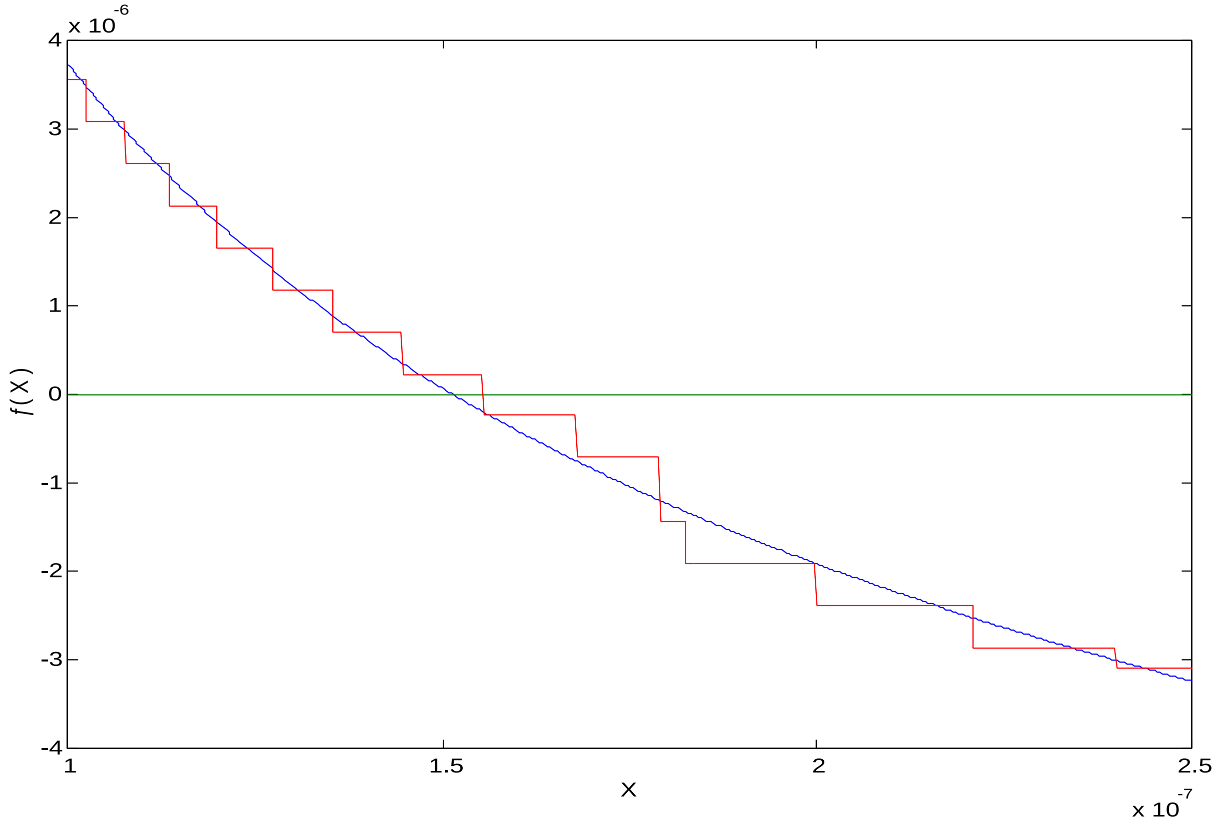
with equality just when all numbers r_k are the same for all $k \ne Î$. Anyway, $\bar{q} \ge 1$ since all such $r_k \le 1$. Now introduce temporary abbreviations $ç := \sqrt{(\sum' c_k)}$ and $\theta := \sqrt{(\sum' c_k\cdot r_k^2)}$ to simplify $\bar{q} := (1 + ç\cdot\theta)/(1 + \theta^2)$ and the confirmation that

$$1 + ç^2 - (2\bar{q} - 1)^2 = (ç\cdot\theta^2 + 2\theta - ç)^2/(1 + \theta^2)^2 \ge 0\ ,$$

so that $\bar{q} \le (1 + \sqrt{(1 + ç^2)})/2$ with equality just when $2θ = ç/\bar{q}$. Work backward to obtain the foregoing more general upper bound upon q . It can be arbitrarily big if $|γ_{\hat{I}}|/\sqrt{(\sum_k γ_k^2)}$ is tiny enough, and then the endeavor described above can fail and often does, alas; then roundoff can deflect some eigenvectors, if computed from the simple formula above, through angles roughly as big as the gaps' worst relative errors.

. . . .

The rounding-error-bounds $ü(ξ_J)$ and $æ·q(ξ_J)$ cost little to compute because they come from quantities already computed for each iteration. The bounds are approachable too, occasionally bigger than actual errors by less than an order of magnitude. However the bounds tend to gross pessimism, the more so as dimension K increases. They are pessimistic, on computers that conform to IEEE Standard 754 for floating-point arithmetic, because rounding errors resemble independent random variables with mean zero in so far as they tend to cancel partially as they accumulate. But roundoff is not random on these computers; consequently computed values of $f(x)$ are monotone non-increasing between poles. Here is a snapshot comparing values of a typical Spectral $f(x)$ computed with 4-byte floating-point versus 8-byte:



This staircase graph is typical, though the sizes of horizontal steps and vertical risers are often far more diverse. Were roundoff random or biased, the graph of f would appear ragged or shifted, and its zeros $ξ_J$ would need a trickier program and noticeably longer to be computed "at least about as accurately as the data and the arithmetic's precision deserve".

. . . .

Evidently roundoff can debase the relative accuracy of a computed ξ_J severely only when $|\gamma_{\hat{I}}|/\sqrt{(\sum_k \gamma_k^2)}$ is too tiny. Then, in an attempt to circumvent debasement, we could try …

> *Deflation*: As if $\gamma_{\hat{I}} = 0$, remove it and $\lambda_{\hat{I}}$ from the data temporarily;
> restore them after all K–1 other roots ξ_k have been computed.

Deflation is motivated by the observation that one of the Spectral equation's roots $\xi_J \to \lambda_{\hat{I}}$ as $\gamma_{\hat{I}} \to 0$ while all other data is held fixed. Two questions demand attention:

Which root $\xi_J \to \lambda_{\hat{I}}$? How big is Deflation's error $|\xi_J - \lambda_{\hat{I}}|$?

Their answers will tell us how small $|\gamma_{\hat{I}}|$ must be to render Deflation's error (accepting $\lambda_{\hat{I}}$ in place of ξ_J) tolerable, and where to restore $\lambda_{\hat{I}}$ in sorted order among the other roots ξ_k .

We know already that either $J = \hat{I}$ or $J = \hat{I} - \mathrm{sign}(\alpha)$. Assume for now that we know which; the appropriate choice(s) will emerge later. And assume $\lambda_{\hat{I}}$ is not too near any other $\lambda_{\dots}$.

At first sight the error $|\xi_J - \lambda_{\hat{I}}|$ might be expected to approach zero like $\gamma_{\hat{I}}^2$ since this is the only way $\gamma_{\hat{I}}$ appears in the Spectral equation $f(\xi_J) := \sum_k \gamma_k^2/(\xi_J - \lambda_k) - 1/\alpha = 0$. Usually the error does behave that way; but whenever $\lambda_{\hat{I}}$ is a kind of double root the error approaches zero much slower, like $|\gamma_{\hat{I}}|$, instead. Here is how that can happen:

Abbreviations $\sum' := \sum_{k \ne \hat{I}}$, $f_{\hat{I}}(\xi) := \sum' \gamma_k^2/(\xi - \lambda_k) - 1/\alpha$, $f_{\hat{I}} := f_{\hat{I}}(\lambda_{\hat{I}})$, $f_{\hat{I}}^\dagger := f_{\hat{I}}^\dagger(\xi_J, \lambda_{\hat{I}})$ and $f_{\hat{I}}' := f_{\hat{I}}'(\lambda_{\hat{I}}) = f_{\hat{I}}^\dagger(\lambda_{\hat{I}}, \lambda_{\hat{I}}) < 0$ will help reduce clutter. Note: $f_{\hat{I}}$ and $f_{\hat{I}}'$ are computable.

Now $0 = f(\xi_J) = \gamma_{\hat{I}}^2/(\xi_J - \lambda_{\hat{I}}) + f_{\hat{I}}(\xi_J) = \gamma_{\hat{I}}^2/(\xi_J - \lambda_{\hat{I}}) + f_{\hat{I}}(\lambda_{\hat{I}}) + (\xi_J - \lambda_{\hat{I}})\cdot f_{\hat{I}}^\dagger(\xi_J, \lambda_{\hat{I}})$, whence comes $\gamma_{\hat{I}}^2/(\xi_J - \lambda_{\hat{I}}) + f_{\hat{I}} + (\xi_J - \lambda_{\hat{I}})\cdot f_{\hat{I}}^\dagger = 0$. From this equation's two solutions $(\xi_J - \lambda_{\hat{I}})$ we select the one conforming to our assumption that $|\gamma_{\hat{I}}|$ is tiny enough that the limits, namely $|\xi_J - \lambda_{\hat{I}}| \to 0$ and $f_{\hat{I}}^\dagger \to f_{\hat{I}}' < 0$ as $\gamma_{\hat{I}} \to 0$, are very near. When $f_{\hat{I}} \ne 0$ our selection must be

$$\xi_J - \lambda_{\hat{I}} = -2\gamma_{\hat{I}}^2/\big(\,\mathrm{sign}(f_{\hat{I}})\cdot\sqrt{(f_{\hat{I}}^2 - 4\gamma_{\hat{I}}^2\cdot f_{\hat{I}}^\dagger)} + f_{\hat{I}}\,\big) \approx -2\gamma_{\hat{I}}^2/\big(\,\mathrm{sign}(f_{\hat{I}})\cdot\sqrt{(f_{\hat{I}}^2 - 4\gamma_{\hat{I}}^2\cdot f_{\hat{I}}')} + f_{\hat{I}}\,\big) ;$$

and then $J = \hat{I} - \big(\mathrm{sign}(f_{\hat{I}}) + \mathrm{sign}(\alpha)\big)/2$.

When $f_{\hat{I}} = 0$ (then $\hat{I} \ne \mathrm{sign}(\alpha)$ and $\hat{I} \ne K+1+\mathrm{sign}(\alpha)$) our selection becomes ambiguous:

$$\xi_J - \lambda_{\hat{I}} = \pm|\gamma_{\hat{I}}|/\sqrt{(-f_{\hat{I}}^\dagger)} \approx \pm|\gamma_{\hat{I}}|/\sqrt{(-f_{\hat{I}}')} \quad \text{for} \quad J = \hat{I} + \big(\pm 1 - \mathrm{sign}(\alpha)\big)/2 \quad \text{resp.}$$

This ambiguity is not a defect in our analysis, but a characteristic of the dispersal of a double root's fragments by a perturbation of its equation. $\lambda_{\hat{I}}$ is the double root, half Hidden, of the Secular equation when $\gamma_{\hat{I}}$ is infinitesimal. In other words, $\lambda_{\hat{I}} = \lambda_{\hat{I}-\mathrm{sign}(\alpha)}$ turns out to be a double eigenvalue of matrix $X := V + \alpha\cdot\mathbf{c}\cdot\mathbf{c}'$ when $\gamma_{\hat{I}} = 0$. A small example with $\hat{I} = 2$ has

$$\gamma_1 := \lambda_1 := -2 , \quad \lambda_2 := 0 , \quad \gamma_3 := \lambda_3 := \alpha := 1 , \quad \text{and tiny variable } \gamma_2 := \gamma$$

and produces these series expansions for the three eigenvalues of X :

$\xi_1 = -|\gamma|/\sqrt{2} - \gamma^2/16 + 29\cdot|\gamma|^3/(256\sqrt{2}) + \dots$,
$\xi_2 = +|\gamma|/\sqrt{2} - \gamma^2/16 - 29\cdot|\gamma|^3/(256\sqrt{2}) + \dots$, and
$\xi_3 = 4 + 9\cdot\gamma^2/8 - 81\cdot\gamma^6/8192 + \dots$, all for $|\gamma| < 1.461\dots$.

Generally, when $\lambda_{\hat{I}}$ is far enough from all other $\lambda_{\dots}$s, the foregoing approximations to $\xi_J - \lambda_{\hat{I}}$ permit Deflation's error to be deemed tolerable or not. The situation gets more complicated otherwise, when $\lambda_{\hat{I}}$ belongs to a tight cluster of values $\lambda_{\dots}$, since $f_{\hat{I}}'$ may approximate $f_{\hat{I}}^{\dagger}$ poorly then. A crude but fully general overestimate of the effect upon all X's eigenvalues of Deflation comes from recognizing it as a Rank-2 perturbation of X that subtracts from all its eigenvalues amounts between $\alpha\cdot(\,\gamma_{\hat{I}} \pm \sqrt{(\gamma_{\hat{I}}^2 + 4\sum_{k\neq\hat{I}}\gamma_k^2)}\,)\cdot\gamma_{\hat{I}}/2$, most of them much tinier.

Whether with Deflation or without, whether assessed or not, the accuracies of all computed eigenvalues can be at least about as good as the data and arithmetic's precision deserve, and yet some gaps $\xi_J - \lambda_k$ can be too inaccurate for reliable computation of all X's eigenvectors from the simple formula $\mathbf{x}_J = \text{Column}(\{\gamma_k/(\xi_J - \lambda_k)\})$. To compute every eigenvector of X reliably without extra-precise arithmetic requires unobvious formulas for them and for eigenvalues obtained accurately enough with the aid of artful origin shifts; these are a story for another day.

§12: Another Example, a Tridiagonal Eigenvalue Problem

Suppose T is a K-by-K symmetric tridiagonal matrix whose eigenvalues τ_j are sought. These are the K zeros of $\det(T - \tau\cdot I)$ but it is harder to compute than $\pi(\tau) := \det(T - \tau\cdot I)/\det(\overline{T} - \tau\cdot I)$, where $\overline{T}$ is obtained from T by striking off its last row and column, after floating-point Over/Underflow is taken into account. In fact, $\pi(\tau)$ is the last diagonal element of the upper-triangular factor U of $T - \tau\cdot I = L\cdot U$ *without* pivotal exchanges; L is lower-triangular with a diagonal whose every element is 1 . The K poles of π are at ∞ and the eigenvalues of $\overline{T}$, and interlace the zeros of π. The count of zeros of π less than τ is the same as the count of negative diagonal elements of U ; a count of the signs of all but the last diagonal element of U counts finite poles of π less than τ , locating them implicitly. Between poles, $\pi'(\mu) < -1$ and computed values of $\pi(\mu)$ are monotone non-increasing despite roundoff.

Hyperbolic Iterating Functions $H\pi(\dots)$ act much as $Hf(\dots)$ do for the Spectral equation's f because a Möbius Map (§4) that takes the infinite pole of π to a finite location turns π into a rational function f with K finite zeros interspersed among K finite poles between which f is strictly decreasing. However, because the poles of π are not explicit, straddles of Hidden zeros of π have to be found by a process of *Binary Chop* acting on the aforementioned counts. On the other hand, Deflation, accomplished by zeroing out negligible off-diagonal elements of T , is simpler to manage for π than for f . So is roundoff. Another story for another day.

§13: Further Reading

For more about the theory of equation-solving see books by J.F. Traub, by A.M. Ostrowski, and by A.S. Householder, all cited in my extensive *Lecture Notes on Real Root-Finding* posted at <www.cs.berkeley.edu/~wkahan/Math128/RealRoots.pdf>. These notes also apply *Projective* maps built out of Möbius maps to Secant and Newton’s Iterations. Properties of Möbius maps are explored in my lecture notes posted at <…/Math185/Mobius.pdf>. For the properties of eigenvalues of real symmetric matrices see B.N. Parlett’s book *The Symmetric Eigenvalue Problem* (1998, Classics in Applied Mathematics #20) Soc. for Indust. & Appl. Math., Philadelphia. Roundoff is treated at length in N.J. Higham’s book *Accuracy and Stability of Numerical Algorithms* 2d ed. (2002) Soc. Indust. & Appl. Math., Philadelphia. For the tiresome details about unobvious eigenvector formulas and “artful origin shifts” see my notes on *Rank 1 Updates of a Symmetric Eigenproblem* to be (WHEN?) posted at <…/SymUpdt1.pdf>, and the MATLAB programs therein. The unobvious eigenvector formulas come from Ming Gu and Stanley C. Eisenstat (1994) “A Stable and Efficient Algorithm for the Rank-One Modification of the Symmetric Eigenproblem” pp. 1266-1276 of *SIAM J. Matrix Anal. Appl.* **15**. For the currently best alternative to Bi-Confluent Hyperbolic Iteration see Ren-Cang Li (1994) *Solving Secular Equations Stably and Efficiently,* EECS Comp. Sci. Divn. Tech. Rept. No. UCB//CSD-94-851, Univ. of Calif. @ Berkeley. His scheme supplanted an older one, in “Rank-one modication of the symmetric eigenproblem” by J.R. Bunch, Ch.P. Nielsen, and D.C. Sorensen (1978) on pp. 31-48 of *Numer. Math.* **31**, that converged to each zero monotonically, and to Hidden zeros far too slowly.

Prof. W. Kahan
Mathematics Dept. and
E.E. & Computer Sci. Dept.
University of California
Berkeley CA 94720
<www.cs.berkeley.edu/~wkahan>