Root Finding by Similarity and Analogy: A Bracketing Method with a Data-Selected Decision Layer

V. Sitnikov

Abstract

We replace bisection's midpoint rule by a similarity rule — the step grows with the dissimilarity of $f(x)$ to 0 under a similarity S — and derive its contraction factor in closed form. It fixes, before any benchmark, a base rule: with the tent similarity and a three-point divided-difference scale it is a Newton step of Muller's order 1.839, clipped to the bracket and projected onto a budget of twice bisection whose multiplier is its one numerical constant. On the 154 problems of Alefeld, Potra and Shi it is level with TOMS 748 and 1.5% behind Chandrupatla's method; on wide brackets it is clearly behind. The gap is a decision, when to give the model up, and we take it by analogy: at a flat root the dimensionless configuration of the last three points recurs under zoom, at a simple root it tends to a line, and the analogous pair gives the order that says whether and where to split. This decision layer — about ten discrete rules selected on two development sets — gives A′. On 240 problems fixed before its code was frozen it needs 3656 evaluations against 5717 for Chandrupatla, 5535 for ModAB and 4488 with Lagouanelle's multiplicity estimate in place of the analogy; the lead lies in multiple and fractional roots, present in two of six families. With the multiplicity known, the transformation $\mathrm{sign}(f)|f|^{1/m}$ is 2.4 to 6.8 times cheaper on multiple roots; A′ adds an estimate of the order, within a guarantee of twice bisection.

1. Introduction

A bracketing root finder must decide, at every step, where inside the bracket to evaluate next. Bisection answers "in the middle", which is the correct answer when nothing is known. Every improvement since has answered "where a local model of f says the root is": a secant in false position, an inverse quadratic in Brent [2], a cubic in TOMS 748 [1], an interpolate–truncate–project cycle in ITP [4], an Anderson–Björck step behind a linearity check in ModAB [10, 20]. Each of those answers carries constants: Illinois [7] halves the stale ordinate, Brent accepts an interpolation subject to three empirical inequalities, ITP is governed by κ_1, κ_2, n_0.

This paper asks a different question. Suppose the only thing available is a *similarity function* — a map S from an outcome to $[0, 1]$, where 1 means "identical to what was wanted" and 0 means "not similar at all", in the sense used for parameter-free control in [6]. Can a root finder be built out of S alone, and how does it compare with methods built out of polynomial models and tuned constants?

The paper makes four claims, each measured.

1. A contraction law for the similarity step (§2) fixes a *base rule* — tent similarity, three-point scale, a projection safeguard — before any benchmark is consulted (§3); its order is Muller's.
2. Projecting the step onto the bisection budget, instead of replacing it, keeps the order and a worst case of twice bisection plus one step; the budget's multiplier is the base rule's one numerical constant, and a constant-free variant costs a measured price (§3.1, §5).
3. The base rule's gap on wide brackets is a decision — when to give the model up — and a similarity of the run to its own past takes it (§6). The resulting method, A′, is tested once

on a corpus fixed before its code was frozen, against simpler detectors put in its place, and against the classical transformation with the multiplicity known in advance (§7).
4. Every table uses one criterion, a sign change within 10^{-6} of the answer, because the benchmarks' published roots score six problems unsolved for every method (§4, §A.5).

The search that produced these results — readings of the similarity that did not pay, the decomposition of the wide-bracket gap, wall-clock time, and the rejected rules — is reported in Appendices A–C. The code, the corpora and the documents that recorded each test before its run are released (Appendix D and the availability statement).

---

2. The similarity step rule

Definition 1 (Similarity step). Let $[a, b]$ bracket a root, x be the current iterate and x_{far} the opposite end. Let $S : \mathbb{R} \to [0, 1]$ with $S(0) = 1$ be a similarity function and $\sigma > 0$ a scale. The next iterate is

$$x' \;=\; x \;+\; \bigl(1 - S(f(x)/\sigma)\bigr)\,(x_{\text{far}} - x). \tag{1}$$

The rule is exactly the informal statement "step further when the outcome is dissimilar to the expectation, and less far when it is similar". Bisection is the degenerate member $S \equiv \frac{1}{2}$: a similarity function that carries no information about f.

2.1 The two-point normalisation is false position

The most obvious way to make (1) scale-free is to normalise the similarity by the two ends of the bracket, $S(a) = |f(b)|/(|f(a)| + |f(b)|)$ and $S(b) = 1 - S(a)$, and to place the next iterate at the similarity-weighted average of the ends.

Proposition 1. For the ratio similarity $S(z) = 1/(1 + |z|)$ with $\sigma = |f(x_{\text{far}})|$, equivalently for the endpoint weights above,

$$S(a)\,a + S(b)\,b \;=\; \frac{af(b) - bf(a)}{f(b) - f(a)}.$$

That is the false-position formula, and it is specific to that similarity: on $f = x^2 - 2$ over $[1, 4]$, where false position gives 1.200000, $e^{-|z|}$ gives 1.206812, $1/(1 + z^2)$ 1.015228, $1/\sqrt{1 + z^2}$ 1.007624 and the truncated linear form 1.214286. Bracket-scaled similarity configurations take third, fourth and sixth place on Chandrupatla's set (Table 3), where false position solves 6 of 45.

Proposition 1'. It is, however, specific to the *bracket ends* rather than to the ratio form. Take the truncated linear similarity $S(z) = \max(0, 1 - |z|)$ and the divided difference over the two ends, $\sigma = |f[a, b]| \cdot |b - a| = |f(b) - f(a)| = |f(a)| + |f(b)|$. Then $1 - S = |f(a)|/(|f(a)| + |f(b)|)$, which is false position again, and the agreement is exact to the last bit.

Together the two say that the boundary of novelty runs through the words *most recent* in the definition of $\hat{f}'$ in §3: a divided difference over the bracket ends is false position, the same difference over the most recent iterates is the secant. In our measurements plain false position solves 127 of the 154 Alefeld–Potra–Shi problems, against 154 for bisection.

2.2 The contraction law

Write $e_n = x_n - r$ for the error, $D = x_{\text{far}} - r$, and let c be the slope of $1 - S$ at the origin, $1 - S(z) = c\,z + O(z^2)$.

Proposition 2 (Contraction law). For a simple root and e_n small, rule (1) satisfies

$$e_{n+1} \;=\; e_n \left(1 - \frac{c\,|f'(r)|\,D}{\sigma}\right) + O(e_n^2), \qquad \text{rate} = \left|1 - \frac{c\,|f'(r)|\,D}{\sigma}\right|. \tag{2}$$

Derivation. $|f(x_n)| = |f'(r)|\,|e_n| + O(e_n^2)$, hence $1 - S(f(x_n)/\sigma) = c\,|f'(r)|\,|e_n|/\sigma + O(e_n^2)$, and $x_{\text{far}} - x_n = D - e_n$. Substituting into (1) and discarding second-order terms gives (2). □

Table 1 checks (2) against measurement on $f(x) = x^2 - 2$ over $[0, 2]$ with $\sigma = |f(0)|$, for which $|f'(r)| = 2\sqrt{2}$, $D = 2 - \sqrt{2}$ and $\sigma = 2$.

Table 1. The contraction law against measurement, all rows at the *chosen* scale $\sigma = |f(x_0)|$. c is the slope of $1 - S$ at the origin and c_2 the coefficient of z^2 when $c = 0$.

$S(z)$	c	predicted	measured
$1/(1+\lvert z\rvert)$	1	rate 0.1716	0.1716
$e^{-\lvert z\rvert}$	1	rate 0.1716	0.1716
$1/(1+z^2)$	0, $c_2 = 1$	$n\,e_n \to 0.8536$	0.8503
e^{-z^2}	0, $c_2 = 1$	$n\,e_n \to 0.8536$	0.8498
$1/\sqrt{1+z^2}$	0, $c_2 = \frac{1}{2}$	$n\,e_n \to 1.7071$	1.6988
$\max(0, 1 - \lvert z\rvert)$	1	rate 0.1716	fails: stays at x_{far}

A similarity with $c = 0$ does not converge at a rate at all. Expanding $1 - S = c_2 z^2 + O(z^4)$ in the same way as (2) gives $e_{n+1} = e_n - c_2|f'(r)|^2|D|e_n^2/\sigma^2 + O(e_n^3)$, hence

$$e_n \;\simeq\; \frac{\sigma^2}{c_2\,|f'(r)|^2\,|D|} \cdot \frac{1}{n}, \tag{2a}$$

sublinear rather than stagnant. Reporting a "measured rate" for these rows would report the value of e_{n+1}/e_n at whatever n the window happened to end — it creeps to 1 monotonically (0.925 at $n = 10$, 0.991 at $n = 100$, 0.9998 at $n = 4000$) — so the prediction tested is the coefficient in (2a) instead.

The last row fails at once, and it matters because that similarity is the one this paper proposes: at $x_0 = 0$, $z = -1$, so $S = 0$, the step is the whole bracket and the iterate stays at x_{far} (error 0.5858 after one step and after two hundred). The failure belongs to the chosen scale, not to the similarity: under the derived scale of §3 the clip never engages on this problem, and the same function is the best of the nine (Table 4).

Corollary 1 (A chosen similarity is not enough). For fixed S and fixed σ the rate in (2) depends on $|f'(r)|$, D and σ, none of which is under the method's control. Convergence is at best linear and occurs only when the product lands in $(0, 2)$.

Empirically, with σ taken as $|f|$ at the starting point, five to eight of the eight standard problems fail outright for the six similarity functions tested. Those eight, used throughout as a quick set and never as evidence on their own, are: $x^2 - 2$ on $[0, 2]$; $\cos x - x$ on $[0, 1]$; $e^x - 2$ on $[0, 2]$; $x^3 - x - 2$ on $[1, 2]$; $\sin x - \frac{1}{2}$ on $[0, 1.5]$; $x^{10} - 1$ on $[0, 1.3]$, which stalls false position; $e^{-x}(x - 1)$ on $[-1, 40]$, whose right end is flat to 10^{-16}; and $\arctan x$ on $[-1, 3]$.

Corollary 2 (Missing the calibration in either direction costs). The rate degrades as the product $c\,|f'(r)|\,|D|/\sigma$ departs from 1, upwards or downwards. We claim no symmetry between the two directions, and the measurements do not support one: similarities with unbounded slope — $1 - \sqrt{|z|}$ has $c \approx 3.2 \cdot 10^3$ at the resolution used, $1 - |z|^{1/3}$ has $c \approx 4.6 \cdot 10^4$ — fail on five and six of the eight problems with a chosen scale, yet over the full benchmark (best over scales, Table

4) they finish at 9411 and 10874, ahead of all three $c = 0$ similarities at 13187, 13990 and 15272. Overshooting is the milder failure.

Corollary 3 (The scale must come from the data). The rate is zero exactly when $c\,|f'(r)|\,D/\sigma = 1$, i.e. $\sigma = c\,|f'(r)|\,D$. Substituting this σ back into (1) with $c = 1$ gives $x' - x = |f(x)|/|f'|$, the Newton step. The similarity rule is therefore superlinear precisely when its scale is the *local scale of the value axis*, and that scale is not available a priori: it must be read off the evaluations.

---

3. The method

Corollaries 1–3 fix every choice before any benchmark is consulted; Section 5.2 tests the prediction against the full grid. One number in the safeguard is not fixed by them, and §3.1 declares it.

Similarity: any with $c \neq 0$. The law makes the shape of S almost irrelevant and its slope at the origin decisive. We take the plainest member, the **truncated linear** or tent form $S(z) = \max(0, 1 - |z|)$. It is not the ratio form $1/(1 + |z|)$ of Proposition 1.

Scale: the finest data-derived one. We set $\sigma = |\hat{f}'| \cdot |x_{\text{far}} - x|$ with $\hat{f}'$ a divided difference of the **most recent evaluations** — not of the bracket ends, which by Proposition 1' would give false position. A divided difference is a computation, not a fitted coefficient: nothing is chosen, regressed or tuned. The two-point difference has slope error $O(h)$ and gives the secant order $\varphi \approx 1.618$; the three-point difference — the derivative at x_n of the parabola through the last three evaluations, $\hat{f}' = f[x_{n-1}, x_n] + f[x_{n-2}, x_{n-1}, x_n](x_n - x_{n-1})$ — has error $\frac{1}{6}f'''(\eta)\,(x_n - x_{n-1})(x_n - x_{n-2})$, that is $O(e_{n-1}e_{n-2})$. The Newton step with that derivative satisfies $e_{n+1} = \frac{f''}{2f'}e_n^2 + \frac{f'''}{6f'}\,e_n e_{n-1} e_{n-2} + \ldots$; the second term dominates, and the order is the root of $p^3 = p^2 + p + 1$, $p \approx 1.839$ — **Muller's order**, as it should be, since the step uses the same parabola as Muller's method, through its derivative rather than its root. Only where the parabola is exact ($f''' = 0$) does the order rise to Newton's 2. Carried to 10^{-2900} in 3000-digit arithmetic (`order_high_precision.py`), the computational order per step settles at 1.840 on $\cos x - x$, $e^x - 2$, $x^3 - x - 2$ and $\sin x - \frac{1}{2}$ and at 2.000 on $x^2 - 2$; the two-point scale settles at 1.618. In double precision a run takes three or four superlinear steps before it reaches the tolerance, too few to tell 1.84 from 2, and we report no order measured there.

What the rule reduces to. With $S(z) = \max(0, 1 - |z|)$ one has $1 - S = \min(1, |z|)$, and (1) becomes

$$x' \;=\; x + \operatorname{sign}(x_{\text{far}} - x) \cdot \min\Big(\frac{|f(x)|}{|\hat{f}'|},\; |x_{\text{far}} - x|\Big), \tag{4}$$

a Newton step with a three-point divided-difference derivative, clipped to the bracket and then projected as in §3.1. That the tent turns the similarity step into Newton's is a property of the chosen similarity, and which S stands in the step is measured rather than assumed: in the proposed method, replacing the tent by any other function costs 19% to 425% (Table 7). The base rule is therefore not a new family of methods; what is ours is a derivation that fixes every choice before any benchmark is consulted, and a safeguard of ITP's projection type with ITP's truncation stage — and all three of its constants — removed. The proposed method builds on it in Section 6.

Algorithm. Given $[a, b]$ with $f(a)f(b) < 0$ and tolerance ε_x:

```
evaluate fa = f(a), fb = f(b)
A, B <- a, b                              the proven bracket
```

```
x  <- the endpoint with the smaller |f|;  xp <- the other;  xp2 <- none
n_max <- 2 * ceil(log2(|b-a| / xtol));   allow <- 2^n_max * xtol/2
while |B - A| > xtol:
    far   <- the end of [A,B] further from x
    f'    <- derivative at x of the parabola through (x,xp,xp2)
             (two-point difference while xp2 does not yet exist)
    sigma <- |f'| * |far - x|
    if sigma is 0 or not finite:  xn <- midpoint(A,B)
    else:
        xn <- x + min(1, |f(x)|/sigma) * (far - x)         # the step, eq (4)
        lo, hi <- min(A,B), max(A,B)
        xn <- clamp(xn, hi - allow, lo + allow)            # projection, eq (3)
    if xn outside (A,B) or already evaluated:  xn <- midpoint(A,B)
        (if the midpoint too was evaluated: return it)
    evaluate fn = f(xn);   allow <- allow / 2
    xp2, xp, x <- xp, x, xn
    update the bracket: whichever of A,B keeps the sign change
return midpoint(A,B)
```

Every quantity in it is either an evaluation of f, a divided difference of evaluations, or the halving that defines bisection — with one exception, the factor 2 in `n_max`, which §3.1 declares as the base rule's one numerical constant. The constant-free variant of §3.1 replaces the two `allow` lines by the earned credit of (3').

3.1 The safeguard: project, do not replace

An interpolation-only rule has no worst-case bound. The obvious safeguards *replace* the computed step with a midpoint whenever some progress test fails, and all of them are ruinous here, for a reason specific to a fast method: a step that collapses the *error* need not halve the *bracket*, so a bracket-based test fires on exactly the steps that were best. With the three-point scale, replacing on that test makes the convergence linear and takes the Alefeld–Potra–Shi total from 2634 to 7068 — nearly the cost of bisection (7338).

The remedy is to keep as much of the computed step as the budget allows. Let $n_{\max} = 2\lceil \log_2(w_0/\varepsilon_x) \rceil$ be twice the bisection step count for the initial width w_0 and x-tolerance ε_x, and let $\epsilon = \varepsilon_x/2$. At step j, with the bracket $[\ell, h]$, define the allowance and the projection

$$\alpha_j = 2^{\,n_{\max}-j}\epsilon, \qquad x \;=\; \mathrm{clamp}(x_{\mathrm{sim}},\; h-\alpha_j,\; \ell+\alpha_j). \tag{3}$$

The next bracket is at most $\max(x-\ell,\, h-x)$, so (3) gives $w_{j+1} \le \alpha_j$ directly, and induction bounds the run by $n_{\max}$ steps in exact arithmetic. In floating point the bracket can land exactly on the boundary, and the termination test then asks for one more; the guarantee stated and verified is therefore $2\lceil \log_2(w_0/\varepsilon_x) \rceil + 3$ evaluations.

The multiplier is the base rule's one numerical constant, and it is not neutral (`measure_budget_factor.py`, Linux):

Table 1a. The scheduled projection at budget multiplier c, i.e. $n_{\max} = \lceil c \cdot \log_2(w_0/\varepsilon_x) \rceil$. Total evaluations, failures charged.

c	APS-154	CHA-45	both
1	4625	2039	6664
1.25	2626	1822	4448

c	APS-154	CHA-45	both
1.5	2636	1991	4627
2	2634	2357	4991
3	2634	2936	5570
4	2634	3229	5863

On the canonical set the projection binds on one problem only once $c \geq 1.25$ — `aps.13.00`, whose f underflows to zero around the root (20, 30 and 28 evaluations at $c = 1.25$, 1.5 and 2). On the wide-bracket set the cost rises monotonically with c from 1.25 on, and 2 is not the best value; zero slack ($c = 1$) is not degenerate either (4625 and 2039, against bisection's 7338 and 2136). No member of the family is singled out by the bound: the multiplier is a free constant.

The budget earned rather than scheduled: a constant-free variant. The constant can be removed. Let $s_j = \log_2(w_j/w_{j+1})$ be the shrink a step delivers in bits — bisection delivers exactly 1 — and keep a credit

$$C \leftarrow \max(0,\ C + s_j - 1), \qquad \alpha_j = w_j\, 2^{\,C-1}, \qquad x = \mathrm{clamp}(x_{\mathrm{sim}},\ h - \alpha_j,\ \ell + \alpha_j), \qquad (3')$$

leaving the step alone while $C \geq 1$. A step may under-deliver only by what earlier steps banked, and since C never goes negative the shrinks over j steps sum to at least $j - C_0$: the run costs at most bisection plus $\lceil C_0 \rceil$ steps. The endowment C_0 is therefore the promise, not a setting. $C_0 = 0$ makes the allowance exactly half the bracket on the first step, which forces the midpoint and makes the method bisection; every $C_0 \in (0, 1]$ promises the same single extra step; and among those, 1 is the most permissive and — measured — the cheapest:

Table 1b. The earned projection at endowment C_0. Total evaluations, failures charged.

C_0	promise	APS-154	CHA-45	both
0.1	bisection + 1 step	5150	2097	7247
0.5	bisection + 1 step	4462	2037	6499
1	**bisection + 1 step**	**3377**	**2073**	**5450**
2	bisection + 2 steps	3502	1757	5259
5	bisection + 5 steps	2819	1666	4485

Within one promise the cheapest endowment is determined, so the constant-free variant is the earned projection with $C_0 = 1$; its guarantee, $\lceil \log_2(w_0/\varepsilon_x) \rceil + 4$ evaluations, is verified on all 199 problems. We nonetheless propose the scheduled projection with $c = 2$, fixed before any benchmark was consulted: it is 743 evaluations cheaper on the canonical set (2634 against 3377), the earned variant is cheaper on the wide-bracket set (2073 against 2357) — one declared numerical constant and the better canonical result, or none, a tighter bound and the better wide-bracket result.

Three remarks apply to both forms.

- **Guaranteed bounds of this kind are classical.** Bus and Dekker [21] gave bracketing methods whose worst case is four to five times bisection; the scheduled projection guarantees twice bisection plus one step, the earned one bisection plus one step.
- **This is ITP's projection stage without ITP's truncation stage.** All three of ITP's constants live in the truncation, whose purpose is to repair regula falsi; the similarity step needs no repair, so the truncation and every constant with it are dropped.

- **Writing the clamp in the form (3) rather than as a radius $\alpha_j - w_j/2$ about the midpoint is not cosmetic.** That radius is a difference of two nearly equal numbers; the cancellation made the computed radius too large by a relative $8.9 \cdot 10^{-5}$, which broke the invariant on eleven Chandrupatla problems.

---

4. Experimental protocol

Problem sets. The 154 problems of Alefeld, Potra and Shi [1] and the 45 of Chandrupatla [3], taken programmatically from SciPy's `scipy.optimize._tstutils` so that no bracket or root is transcribed by hand.

Cost model. Evaluations of f. A run counts as solved when it reports convergence and f changes sign within 10^{-6} of the returned point, or vanishes there — the bracketing notion of a root, the same criterion on every corpus of this paper; x-tolerance 10^{-12}; budget 200 evaluations. Every method, ours included, stops when its bracket is no wider than ε_x or when it evaluates an exact zero of the function it works on; none stops on a small $|f|$. Scoring against the published root instead, as the benchmarks' harness does, charges every method the full budget on six problems where f vanishes identically around the root in floating point (§A.5). Head-to-head counts (wins/ties/losses) are taken over every problem of a set with the same charging as the totals. *Failures are charged the full budget*, so no method can buy a low total by giving up early, and totals are reported over all problems rather than over the subset a method happened to solve. Rankings are by failures first, then by total.

Competitors. Where SciPy ships the method, the competitor is the library — Chandrupatla is `scipy.optimize.elementwise.find_root`, Brent `brentq`, TOMS 748 `toms748` — run with the stopping rule of every method (a bracket no wider than ε_x, the relative term at machine precision, no stopping on $|f|$) and counted by the same wrapper around f. ITP, which has no library implementation, is ours, run with $\epsilon = \varepsilon_x/2$ because it terminates on $b - a \le 2\epsilon$, checked against its own theorem ($n_{1/2} + n_0$ iterations at most, on all 154 problems for $n_0 \in \{0, 1, 2, 5\}$), and reported both at the authors' constants and as a *per-problem oracle* over a (κ_1, n_0) grid — not an achievable method, included so that it cannot be said to have been handicapped. Bisection and the false-position family are ours.

Reproducibility. All numbers come from a single Linux run. The test functions call `libm`, whose last digit differs between platforms; the same SciPy version scores 2635 for TOMS 748 on Windows and 2636 on Linux for the same set. Every number stated in this paper is checked against the run data by a script that is part of the test suite.

---

5. Results

The grid is nine similarity functions × four scales × seven safeguards = 252 configurations, plus twelve baselines: 264 in all. The base rule of Section 3 — tent similarity, three-point scale, projection safeguard with its multiplier declared — is one entry of that grid, fixed by the theory rather than chosen from the ranking; the constant-free earned variant of §3.1 is shown beside it. Chandrupatla's own method and Dekker's are among the baselines.

Table 2. Alefeld–Potra–Shi, 154 problems, out of 264 configurations. Places by failures first, then total.

#	configuration	tuned	solved	total evals	$> 2\times$
1	Chandrupatla	no	154/154	2595	0

#	configuration	tuned	solved	total evals	$> 2\times$
2	the same rule with no safeguard at all	no	154/154	2634	0
3	**base rule: tent, 3-point, projection ($c = 2$)**	**no**	154/154	**2634**	0
4	TOMS 748	yes	154/154	2636	0
5	Brent	yes	154/154	2707	0
11	ITP, per-problem oracle	yes	154/154	3068	0
15	Ridders [5]	no	154/154	3124	0
25	constant-free variant: earned projection	no	154/154	3377	0
29	ITP, recommended constants	yes	154/154	3767	0
73	bisection	no	154/154	7338	0
232	Dekker	no	153/154	3158	0
237	Illinois [7]	yes	153/154	7045	0
244	false position	no	127/154	10560	—

The base rule and the same rule without any safeguard tie at 2634: the projection never binds on this set. By total, Chandrupatla's method is first with 2595, 1.5% fewer; the base rule and TOMS 748 are level (2634 against 2636). Problem by problem the base rule wins against Chandrupatla 87/24/43 (wins/ties/losses; exact two-sided sign test $p = 1.4 \cdot 10^{-4}$), against TOMS 748 86/16/52, Brent 102/11/41, ITP at its recommended constants 132/3/19, the per-problem ITP oracle 121/4/29, and bisection 152/1/1. It trails Chandrupatla in total because its losses are the larger ones; by function family, the unit in which these problems are independent (§7), the comparison is 6/1/8 of 15 families, a tie. The constant-free variant goes 79/3/72 against Chandrupatla ($p = 0.63$), 78/9/67 against TOMS 748 and 79/5/70 against Brent. Dekker, Illinois and false position fail `aps.13.00` under either criterion and so keep their totals: they spend the whole budget without converging, Dekker and Illinois stalled on one side of the zero where $|f| < 10^{-62}$ but f does not change sign within 10^{-6}.

Table 3. Chandrupatla, 45 problems, brackets up to 10^{12} wide, out of 264 configurations.

#	configuration	tuned	solved	total evals	$> 2\times$
1	Chandrupatla	no	45/45	1505	0
2	ITP, per-problem oracle	yes	45/45	1675	0
3	similarity, $e^{-\lvert z\rvert}$, bracket scale, both guards	no	45/45	1778	0
4	similarity, $e^{-\lvert z\rvert}$, bracket scale, bracket guard	no	45/45	1848	0

#	configuration	tuned	solved	total evals	$> 2\times$
5	ITP, recommended constants	yes	45/45	1946	0
6	similarity, $e^{-\lvert z\rvert}$, bracket scale, earned projection	no	45/45	1952	0
9	constant-free variant: earned projection	no	45/45	2073	0
15	bisection	no	45/45	2136	0
49	**base rule: tent, 3-point, projection ($c = 2$)**	**no**	45/45	**2357**	0
158	TOMS 748	yes	45/45	3088	15
227	Brent	yes	43/45	3238	18
229	the same rule with no safeguard at all	no	39/45	3391	10

Here Chandrupatla's method is genuinely the stronger: it beats the base rule 29 problems to 11 ($p = 0.006$) and by 1505 against 2357 in total. The base rule beats Brent 34/4/7 and TOMS 748 33/1/11 — which exceed $2\times$ bisection, a bound they do not promise, on 18 and 15 problems — and loses to ITP 18/1/26 and to the ITP oracle 16/1/28. The wide-bracket set is where the multiplier matters (Table 1a) and where the constant-free variant is the cheaper form, 2073 against 2357. Over both sets the base rule needs 4991 and the constant-free variant 5450, against 4100 for Chandrupatla, 4743 for the ITP oracle, 5724 for TOMS 748 and 5945 for Brent.

5.1 What the guarantee costs

Against the same rule with no safeguard (Tables 2 and 3), the projection costs nothing on the canonical set and on the wide-bracket set solves six more problems for 30% fewer evaluations (2357 against 3391; unguarded, 10 runs break the bound). The earned projection, promising bisection plus one step, pays 743 evaluations (28%) on the canonical set and is the cheapest of the three on the wide-bracket set. That is the point of projecting rather than replacing: the two safeguards that replace cost 7068 and 7046 on Alefeld–Potra–Shi, against 2634.

5.2 The law across the whole grid

Corollaries 1–3 predict that the similarities with $c = 1$ come before those with $c = 0$ under a data-derived scale; they say nothing about the order within a group, nor about the functions with unbounded slope. For every data-derived scale the three similarities with $c = 1$ occupy the top three places of Table 4 and the three with $c = 0$ the bottom three (that $c = 0$ is slow is the unsurprising half: a step proportional to z^2 is damped quadratically). With the chosen scale the two groups are indistinguishable (14983 against 14228), as they should be.

Table 4. Mean total evaluations over both sets, step guard, grouped by the slope c of $1 - S$ at the origin.

scale	$c = 1$ (linear approach)	$c = 0$ (quadratic approach)
3-point derived	**5764**	21200

scale	$c = 1$ (linear approach)	$c = 0$ (quadratic approach)
2-point derived	5854	21089
bracket	9149	18015
start (chosen)	14983	14228

Ranked individually and best over scales, the similarities come out $1 - |z|$ (5422), $e^{-|z|}$ (5772), $1/(1 + |z|)$ (5993), $(1 + |z|)^{-2}$ (8455), $1 - \sqrt{|z|}$ (9411), $1 - |z|^{1/3}$ (10874), e^{-z^2} (13187), $1/(1 + z^2)$ (13990), $1/\sqrt{1 + z^2}$ (15272); the scales come out 3-point (5422), 2-point (5564), bracket (7972), start (13423). The group order and the order of the scales were predicted before the grid was run; the order inside each group is measured.

---

6. The flat regime, recognised by analogy

The base rule's gap on wide brackets is a decision, *when to give the model up* (§A.6): on the pure powers `fun3`–`fun6` the right move is to halve, Chandrupatla's method halves on up to all of its steps, and no reading of a single state measured takes that decision correctly. This section takes it by a similarity of the run to *its own past* — an analogy in Pólya's sense [8], two situations agreeing in more than one relation. Every rule below was recorded, with its criteria and stop rule, before its code and run (Appendix D); the rejected ones are in Appendix C.

6.1 The flag: a promise that was not kept

A tangent step on an approach of local order m lands at $e(1 - 1/m)$ and leaves the residual $r = |f_{\text{new}}/f_{\text{old}}| = (1 - 1/m)^m$ on the same side of the root. It beats a halving exactly when $|1 - 1/m| < \frac{1}{2}$, that is for $\frac{2}{3} < m < 2$. At the upper end $r = (1 - \frac{1}{2})^2 = \frac{1}{4}$. So a model step that did not cross the root and left $r \geq \frac{1}{4}$ is a *promise not kept*: the model is in the regime where a halving would have done better. The constant is bisection's $\frac{1}{2}$, carried through the formula; the contraction $e(1 - 1/m)$ is the one behind the classical multiplicity estimates from successive Newton iterates [14, 15], used here to set the point at which a model step has lost to a halving.

Acted upon step by step the flag does not pay: replacing each flagged step by a halving costs the canonical set 2634 → 2839, because a simple root far from convergence also breaks its promises (the flag rises at least once on 142 of the 154 canonical problems). Every refinement that reads the moment alone trades one set for the other (Appendix C, items 12–14).

6.2 The analogy: similarity of a run to its own past

What distinguishes a pure power from a simple root is not the shape of one configuration — `fun1` on a wide bracket looks as non-linear as x^9 — but what happens under zoom. A pure power $f = c(x - r)^m$ is similar to itself: zooming about the root by L scales $x - r$ by L and f by L^m, so the *dimensionless configuration* of the last three points,

$$\xi = \frac{x_k - x_{k-1}}{x_{k-2} - x_{k-1}}, \qquad \varphi = \frac{f_k - f_{k-1}}{f_{k-2} - f_{k-1}},$$

recurs as the iteration proceeds. At a simple root, zooming in makes f look like its tangent, and the configuration tends to the *linear prototype* $\varphi = \xi$ instead. Two prototypes, then: the line, and the run's own past. The judgment is ordinal — is the current (ξ, φ) nearer to some earlier configuration of this run than to the line? — with the plane's Euclidean distance and the perpendicular to the line, the one geometric choice in this section, declared as such.

Nearness of configurations is one relation. An analogy in the full sense needs a second, and the pair supplies it: if the analogous pair shrank x by L_x and f by L_f, the power law requires $L_f = L_x^m$, so the order is *read from the analogy itself*,

$$m = \frac{\ln L_f}{\ln L_x}. \tag{6}$$

Formula (6) is the log–log slope of a power law, with the shape of the computational order of convergence [18, 19]; what it compares is not consecutive iterates but a pair the similarity of their configurations has found. A real run's analogous pairs are only nearly homothetic and the reading scatters about the order, but the regime asks only whether it reaches 2, the order from which a tangent step loses to a halving (§6.1). On the traces of the base rule, at the moments the flag rises, the three conditions together hold on 3 of the 154 canonical problems, on all 20 flat problems, and on none of `fun2`, `fun8`, `fun9`.

The analogy belongs to the run, the flag to the moment. On a pure power the flag rises in the phases of the base rule's cycle whose configuration does *not* recur, so requiring both in one state delayed the entry to the thirteenth state on average. Once established, the analogy is remembered for the run and the regime is entered at the next unkept promise (eighth state on average).

The translation limit. The exponential is self-similar by *translation*, $e^{x-1} = e^x/e$: along its flank the step does not shrink while f does, so (6) is undefined or negative. That is the limit $m \to \infty$ of (6), steeper than any power, and it is read as $m = \infty$: when an analogous pair shows $L_f < 1 \le L_x$. It pays on high powers far from their root, which are locally exponential, $x^m = e^{m \ln x}$.

Exit. Inside the regime the step splits the bracket (§6.3). At every split the evaluated f is judged against two predictions: the line through the bracket ends, $F_A(1-t) + F_B t$ at the split fraction t, and 0. Nearer the line, the regime is left and the remembered analogy is cleared. A sticky halving without this exit cost the canonical set 6707; with it, 2935 — the exit is what makes a false entry cheap.

6.3 Where to split: the point from the order

Halving is the answer when nothing is known about the shape; inside the regime the order is known. For $|f| \propto |x - r|^m$ the ends of the bracket lie at distances in the ratio $|F_A|^{1/m} : |F_B|^{1/m}$ from the root, so the regime splits at

$$t = \frac{|F_A|^{1/m}}{|F_A|^{1/m} + |F_B|^{1/m}}, \tag{7}$$

which is exact for a pure power and becomes the midpoint as $m \to \infty$. It is false position applied to $\operatorname{sign}(f)|f|^{1/m}$ — the transformation that turns a root of multiplicity m into a simple one, classical since Schröder [16] and Traub [17]; what is new here is only where m comes from. With the multiplicity given in advance the transformation is far cheaper still (§7.1): the contribution is the estimate, and its price is measured there. (The order read from a failed promise, $(1 - 1/m)^m = r$, gives $m = \infty$ in 163 of 174 entries, because the base step is not a pure tangent.) With it the flat families of the wide-bracket set cost 743–847 evaluations against 1044 for pure bisection and 1068 for Chandrupatla.

6.4 Five more readings

The half the step promises. $S(z) \geq \frac{1}{2}$ means a step of at most half the way to the far end: the model promises the root in the half of the bracket *near* x. The line through the bracket ends puts it nearer the end with the smaller $|f|$. When the far end has the smaller $|f|$ the two disagree about the half, and the evaluation halves. On a steep flank — a slope dominated by $f(81) = 1.5 \cdot 10^{35}$ on `fun9.5` — this is the false promise "the root is here", and removing it alone takes the canonical set from 2634 to 2392. It is a relative of Brent's acceptance condition [2]: both refuse a model that points towards the worse end of the bracket.

The sign contradiction. The bracket says which way f must go between its ends; the last two points say which way it goes here. Opposite signs — the local picture contradicts the bracket — and the evaluation halves. It is the cause of the `fun9` well: a near-cancelling end reads the root as close while $|f|$ doubles for seventeen steps.

The repeated step pattern. Two successive moves the same way, of similar length, mean the iterate creeps along a geometric series whose remainder is step $\cdot\, s/(1-s)$; the step is lengthened by $1/(1-s)$, with s the similarity of successive step lengths, but only when the pattern of successive length ratios repeats with a period read from the data (the lag of highest mean similarity) and that similarity exceeds $\frac{1}{2}$.

The overshoot branch. §6.1's inequality has a second branch, $m < \frac{2}{3}$: a root of order below one, where the tangent step overshoots and the iterate oscillates about the root ($\sqrt{|x|}\,\mathrm{sign}\,x$: 84 evaluations of the base rule). The flag for it is the unkept promise *with* a sign change, $f_{\mathrm{new}}/f_{\mathrm{old}} \leq -\frac{1}{4}$, and an analogous pair reading $0 < m \leq \frac{2}{3}$; the split (7) is again exact for a pure power. At a root of order below one f is far from 0 on both sides and the exit fires falsely, so the memory of this branch is *not* cleared by the exit (clearing it was worse on the development corpora).

Borrowed: the tolerance step. Once the root is reached from one side the bracket's far end can stay far: on sin over $[3, 4]$ the base rule reaches $|f| \approx 10^{-16}$ in 7 evaluations and then halves the remaining $1.3 \cdot 10^{-7}$ down to the tolerance in 17 more. Dekker [9] and Brent [2] step at least the tolerance; A′ steps $\varepsilon_x/2$ towards the far end when the model's step is shorter, but only after the model has *kept* its previous promise ($r < \frac{1}{4}$). This element is borrowed.

6.5 The proposed method

The method this paper proposes, **A′**, is the base rule of §3 — tent similarity, three-point scale, projection with the declared multiplier $c = 2$ — with the regime of §6.2–6.3 (the analogy remembered, its translation limit, the split (7), the exit) and the five readings of §6.4. Its constants are bisection's $\frac{1}{2}$ and what follows from it: $\frac{1}{4} = (1 - \frac{1}{2})^2$ and the two branches $m \geq 2$, $m \leq \frac{2}{3}$ of $|1 - 1/m| > \frac{1}{2}$. Three choices are declared rather than derived: the multiplier c of the projection (§3.1), the Euclidean geometry of the (ξ, φ) plane, and the pattern test of the step lengthening (lags from 2 to half the history, threshold $\frac{1}{2}$). The projection carries the guarantee, $2\lceil \log_2(w_0/\varepsilon_x) \rceil + 3$ evaluations; it is verified with no violation on every corpus of this paper and under every safeguard of the grid.

Counting numerical constants does not measure the method's freedom, however. Beyond these numbers A′ is a decision layer of about ten discrete rules — the flag, the analogy and its memory, the entry, the exit, the overshoot branch, the split, the step pattern, the half, the contradiction, the tolerance step — chosen on the two development sets among the many candidates of Appendix C. That selection is why the method was tested once on a corpus fixed before its code was frozen, and against simpler rules put in its place (§7). Algorithm 2 is the complete method; `simroot/pure.py` and the compiled `simbracket` implement exactly this and agree bit for bit.

The nearest-configuration search runs over the run's memory: $O(n)$ per step, $O(n^2)$ per run of n evaluations ($n \le 200$ here), arithmetic that matters only when f is cheap (Appendix B).

Algorithm 2 (A′). Notation of Algorithm 1; $S(z) = \max(0, 1 - |z|)$.

```
state:  bracket [A,B], FA, FB; points x, xp, xp2; values fx, fp, fp2
memory: configs (configurations), moves (displacements)
flags:  last_model, in_regime,
        analogy_seen (with order m_a), shallow_seen (with order m_s)
initialise as Algorithm 1; allow <- 2^n_max * xtol/2
while |B - A| > xtol:
    far <- end of [A,B] further from x; mid <- (A+B)/2; model <- false
    f' <- derivative at x of the parabola through (x, xp, xp2)
    sigma <- |f'| * |far - x|
    if sigma is 0 or not finite:  xn <- mid
    else:
        z <- |fx| / sigma
        # the flat regime (6.1-6.3)
        c <- (xi, phi, |x-xp|, |fx-fp|), last three points, if defined
        # unkept promise, and unkept promise with a crossing
        flag      <- not in_regime and last_model and fx/fp >= 1/4
        flag_over <- not in_regime and last_model and fx/fp <= -1/4
        # order (6) of the analogous pair.  q <- the configuration in
        # configs nearest to c (Euclidean in (xi, phi)); no m if c or
        # configs is empty or q is not nearer than the line phi = xi.
        # L_x <- |x-xp| / q.dx,  L_f <- |fx-fp| / q.df
        # m <- inf if L_f < 1 <= L_x (translation), else ln L_f / ln L_x
        m <- order of the analogous pair
        if m >= 2:        analogy_seen <- true;  m_a <- m
        if 0 < m <= 2/3:  shallow_seen <- true;  m_s <- m
        if flag and analogy_seen:         in_regime <- true
        elif flag_over and shallow_seen:  in_regime <- true; m_a <- m_s
        append c to configs
        if in_regime:  xn <- split point (7) with m = m_a
        elif x != xp and
             sign((fx-fp)/(x-xp)) != sign((FB-FA)/(B-A)):  xn <- mid
        elif S(z) >= 1/2 and |f(far)| < |fx|:  xn <- mid
        else:
            step <- (1 - S(z)) * (far - x)              # eq. (4)
            append x - xp to moves
            # step pattern: q_j <- |d_j|/|d_(j-1)| over the
            # successive nonzero moves d (k ratios);
            # rho(u,v) <- min(u,v)/max(u,v); repeats iff
            # max_{p=2..floor(k/2)} mean_j rho(q_j, q_(j-p)) > 1/2
            if repeats and step, x - xp point the same way:
                s <- min(|step|,|x-xp|) / max(|step|,|x-xp|)
                step <- step / (1 - s), at most |far - x| long
            if |step| < xtol/2 and last_model and |fx/fp| < 1/4:
                step <- sign(far - x) * xtol/2         # tolerance step
            xn <- x + step;  model <- true
        lo, hi <- min(A,B), max(A,B)
        xn <- clamp(xn, hi - allow, lo + allow)       # projection (3)
```

```
    if xn outside (A,B) or xn in {x, xp}:
        xn <- mid;  if mid in {x, xp}: return mid
    last_model <- model and xn is the model's own point
    fn <- f(xn)
    if in_regime:                                        # exit
        t <- (xn - A)/(B - A)
        if |fn - (FA(1-t) + FB t)| < |fn|:
            in_regime <- false;  analogy_seen <- false
    allow <- allow/2;  (xp2, xp, x) <- (xp, x, xn)
    keep the sign change in [A,B]
return midpoint(A,B)
```

Table 5. A′ on the two sets of §4 (failures charged 200). A′'s wins/ties/losses against each rival over all problems of the set, exact two-sided sign test.

	APS-154	CHA-45	both	A′ per problem, APS	A′ per problem, CHA
A′ (proposed)	**2259**	**1074**	**3333**	—	—
A′, earned budget (no free constant)	3417	1852	5269	—	—
base rule (§3)	2634	2357	4991	—	—
Chandrupatla	2595	1505	4100	103/20/31, $p = 3 \cdot 10^{-10}$	31/4/10, $p = 0.001$
TOMS 748	2636	3088	5724	127/6/21	42/2/1
Brent	2707	3238	5945	118/17/19	39/3/3
ITP, per-problem oracle	3068	1675	4743	126/7/21	35/1/9
ModAB [10, 20]	2726	1681	4407	118/10/26	34/4/7

A′ is 12.9% below Chandrupatla on the canonical set and 28.6% below on the wide-bracket set, and wins more problems than it loses against every rival on both. By function family, the unit in which these problems are independent (§7), it is ahead of Chandrupatla on 8 of the 15 canonical families (6 behind, one level) and on 7 of the 9 wide-bracket families — too few families for either count to be significant ($p = 0.79$, 0.18). The two it is behind on are `fun7`, 69 against 64, and `fun9`, 65 against 55.

Table 6. Leave-one-out ablation of A′ on all six corpora of §7 (uniform criterion, a sign change within 10^{-6}). Δ = total without the component − total with it; positive means the component earns its place.

component removed	APS	CHA	new-1	Roots.jl	applied	fourth	sum
the whole regime	−7	+1013	+1660	+324	+77	+2269	**+5336**
the split (7) (midpoint instead)	−5	+479	+667	+272	+30	+1672	**+3115**
overshoot branch	−6	0	−7	+16	+63	+893	+959

component removed	APS	CHA	new-1	Roots.jl	applied	fourth	sum
tolerance step (borrowed)	+130	+35	+97	+78	+135	−2	+473
the half the step promises	+249	+65	+82	−16	−5	+1	+376
translation limit	+31	+143	+100	−21	−2	+87	+338
repeated step pattern	−22	−6	+123	−30	−10	+34	+89
sign contradiction	+11	−12	+29	0	+1	−1	+28

Every component is positive in sum; the regime and the split carry the method. The step pattern (earns on two corpora of six) and the sign contradiction (+28 over 855 problems) are marginal — together under one per cent, 117 of 13032 evaluations — and are kept because the rule recorded in advance removes only a component that earns nowhere.

Table 7. The similarity in the step of A′: the same method with each of the nine similarity functions of §5.2 in place of the tent, everything else unchanged. Totals on the two sets of §4 and over the six corpora of §7; the guarantee holds with every function.

S in the step	APS + CHA	six corpora (855)
tent $1-\lvert z\rvert$ **(A′)**	**3333**	**13032**
$e^{-\lvert z\rvert}$	3965	14142
$1/(1+\lvert z\rvert)$	4424	14843
$(1+\lvert z\rvert)^{-2}$	7249	28500
$1-\sqrt{\lvert z\rvert}$	11613	45552
$1-\lvert z\rvert^{1/3}$	16094	63031
e^{-z^2}	17429	67701
$1/(1+z^2)$	17452	67394
$1/\sqrt{1+z^2}$	17487	67854

The nearest other function costs 19% more on the two sets and 9% more over the six corpora, and the functions that approach 1 quadratically cost five times more; the three functions with a linear approach come first and the three with a quadratic approach last, as the contraction law of §2.2 predicts. What this measures should be said plainly: with the tent the step is the undamped Newton step clipped to the bracket, and the other functions damp it, so the table shows that the undamped step is the better one — expected, and evidence for the choice of S within the rule rather than for the similarity framing as such.

6.6 What is known, and what is new

Several mechanisms of A′ coincide with classical ones, reached from similarity and analogy and recognised afterwards:

in A′	classical counterpart
the three-point order estimate of §A.3	Lagouanelle's multiplicity estimate $m = f'^2/(f'^2 - ff'')$ [13]

in A′	classical counterpart
the contraction $e(1-1/m)$ behind the flag of §6.1	multiplicity estimates from successive Newton iterates [14, 15]
the order (6) of an analogous pair	the log–log slope of a power law; the (approximate) computational order of convergence [18, 19]
the split (7)	false position on $\operatorname{sign}(f)\lvert f\rvert^{1/m}$ [16, 17]; inside a bracket, with the multiplicity given, the SFRFm method of [22]
the dimensionless configuration (ξ, φ) of §6.2	the ratios of Chandrupatla's acceptance band [3]; every scaled regula falsi written in two dimensionless parameters [22]
the half the step promises	a relative of Brent's acceptance condition [2]
the tolerance step	Dekker [9], Brent [2] — borrowed and named as such
the exit judgment of §6.2 (the value at a split against the line)	ModAB's linearity check at the midpoint [10, 20], $\lvert y_m - y_3\rvert < k\,(\lvert y_3\rvert + \lvert y_m\rvert)$ with k from a symmetry factor; ours compares two predictions, the line and 0, without a factor

What we found no counterpart for, in the literature we could reach, is the part that carries the gain in Table 6: the *decision* layer. A bracketing method that recognises the regime in which its model has lost by the similarity of the run to its own past — the recurrence of a dimensionless configuration against the linear prototype — reads the order from the analogous pair itself, distinguishes similarity by scaling from similarity by translation, and leaves the regime by a similarity judgment at a point the model did not choose; and a step whose length is set by a similarity function whose form, measured, is decisive. The methods for multiple roots we found are single-point iterations without a bracket, with the multiplicity assumed, estimated from derivatives, or removed by a transformation ([17, 23]; Roots.jl's `Order1B` and `Order2B` [12]); inside a bracket, [22] needs the multiplicity given and recognises a multiple root only by the number of evaluations spent, against a third of bisection's; Chandrupatla takes the decision to halve by an empirical acceptance band [3]. §7.1 tests this against the detectors a reader would try first and against the classical transformation with the multiplicity known or estimated, and the outcome is sharper than the claim: the order read by analogy transfers to the classical frame and works better there than in A′'s own split, while A′'s exit is what keeps simple roots cheap.

---

7. Out-of-sample evidence

Every decision above was taken on the two sets of §4. Recording each decision in advance protects the decision but not the chain: the final configuration — a decision layer of about ten discrete rules, chosen among the many candidates listed in Appendix C — was selected on the same 199 problems. Four further corpora were therefore used, each fixed and committed before any method was run on it, each with its criteria and stop rule written down first.

1. **New corpus 1** — 206 problems: SciPy's `original` set (6) and five generated classes of 40 (seed 20260921): smooth and transcendental simple roots, multiple roots $\lvert x-r\rvert^p$ with $p \sim U(2, 20)$, exponential flanks, near-cancelling terms.
2. **Roots.jl / ModAB** — the bracketing benchmark of Roots.jl [12] used by ModAB's authors [10]: 91 problems, 13 of which coincide with APS or Chandrupatla functions and are excluded, leaving 78.
3. **Applied and GSL** — the 10 bracketed problems of the GNU Scientific Library's test suite [11] and 122 equations from physics and engineering on grids fixed beforehand (Kepler,

Colebrook–White, Lambert's W, Planck/Wien, van der Waals, internal rate of return, the finite square well).

4. **Corpus 4** — 240 problems in six new families (seed 20260922): random polynomials with a simple root, polynomials with a multiple root, fractional powers $|x - r|^p$ with $p \sim U(0.15, 0.95)$, sums of exponentials and sines, compositions tanh, arctan, $\log(1 + q^2)$ of a cubic, and functions odd about the root. The code of A′ was committed *before* this corpus was run, and nothing was changed after. The families were chosen by the author, who knew by then where the method is strong: two of the six (multiple roots, fractional powers) are the regime §6 was built for.

A root must carry a sign change to be bracketed, so the multiple roots in these corpora have odd multiplicity (3, 5, 7, 9 in corpus 4) or the form $\mathrm{sign}(x - r)|x - r|^p$ (corpora 1 and 4).

ModAB, the most recent bracketing method we know of, is run from its authors' package (`pymodab` 1.0.9) with the stopping rule and the counting of every other method. The criterion, tolerance, budget and charging are those of §4.

The variant carried to corpora 1–3 was A′ without the tolerance step and the overshoot branch. It transferred to corpus 1 by every criterion recorded in advance (per problem 105/34/67 against Chandrupatla, $p = 0.005$), did *not* transfer to Roots.jl (38/10/30, $p = 0.4$) and lost on the applied corpus; the two readings that closed those gaps were developed on the seen corpora and tested once, on corpus 4.

Table 8. A′ against the library rivals on all six corpora (uniform criterion): total evaluations; in brackets the runs that break the 2×-bisection bound.

corpus	A′	Chandrupatla	ModAB	ITP	ITP oracle	Brent	TOMS 748
APS (154)	**2259** (0)	2595 (0)	2726 (0)	3767 (0)	3068 (0)	2707 (0)	2636 (0)
Chandrupatla (45)	**1074** (0)	1505 (0)	1681 (0)	1946 (0)	1675 (0)	3238 (18)	3088 (15)
new corpus 1 (206)	**3290** (0)	3716 (0)	3809 (1)	5674 (0)	4707 (0)	6395 (38)	5746 (22)
Roots.jl (78)	**1371** (0)	1566 (0)	1446 (0)	2161 (0)	1629 (0)	2455 (11)	2339 (9)
applied + GSL (132)	1382 (0)	1382 (0)	1460 (0)	3245 (0)	2386 (0)	1503 (1)	1756 (1)
corpus 4 (240), the one test	**3656** (0)	5717 (0)	5535 (8)	5721 (0)	4832 (0)	8267 (50)	8084 (37)
all 855	**13032**	16481	16657	22514	18297	24565	23649

Table 8b. The same runs problem by problem: A′'s wins/ties/losses against Chandrupatla and against ModAB (exact two-sided sign test).

corpus	A′ vs Chandrupatla	A′ vs ModAB
APS (154)	103/20/31, $p = 3 \cdot 10^{-10}$	118/10/26
Chandrupatla (45)	31/4/10, $p = 0.001$	34/4/7
new corpus 1 (206)	108/33/65, $p = 0.001$	125/26/55
Roots.jl (78)	44/14/20, $p = 0.004$	47/11/20
applied + GSL (132)	48/52/32, $p = 0.09$	76/35/21
corpus 4 (240), the one test	**177/35/28,** $p = 10^{-27}$	**183/23/34**

Only the last line of corpora is evidence for A′ in the strict sense; the first five were seen while it was developed and are reported as a description. ITP, at its recommended constants and as the

per-problem oracle, keeps its own bound of bisection plus n_0 everywhere and is behind A′ on every corpus.

Independence. A sign test counts problems as independent draws, and not all of them are. In the two canonical sets one function appears with several brackets or parameters, so the independent unit is the function family (15 in APS, 9 in CHA); at that level the counts of Table 5 are not significant, because there are too few families. In corpora 1 and 4 every problem is drawn independently from its family's generator, so the per-problem test is valid — for the population those generators define, which the author chose (above). On corpus 4 A′ is 36% below Chandrupatla and wins 177 problems to 28. Without the two classes aimed at by the last two readings it is 2403 against 3569, per problem 107/31/22 ($p = 10^{-14}$); on the simple-root classes alone, 737 against 747 (−1.3%), per problem 43/22/15 ($p = 3 \cdot 10^{-4}$).

Table 9. Corpus 4 by class: totals of A′ / Chandrupatla, per problem wins/ties/losses.

class	A′	Chandrupatla	per problem
random polynomial, simple root	433	419	20/8/12
polynomial, multiple root	1163	1805	39/1/0
fractional power, $p < 1$	924	1784	37/1/2
sums of exponentials and sines	304	328	23/14/3
tanh, arctan, $\log(1 + q^2)$ of a cubic	503	1017	25/8/7
odd about the root	329	364	33/3/4

What the evidence supports, and what it does not. Against Brent and TOMS 748 A′ is ahead on every corpus, by 8% to 67% in total, and it stays within its own bound of twice bisection on all 855 problems, a bound that Brent exceeds on 118 and TOMS 748 on 84 — neither promises it. Against ModAB it is ahead on every corpus, in total and problem by problem ($p \leq 0.0013$) — on corpus 4 by 3656 to 5535 and 183 problems to 34 ($p = 7 \cdot 10^{-26}$), on all 91 problems of ModAB's own benchmark by 1507 to 1588 (56/13/22, $p = 1.5 \cdot 10^{-4}$); ModAB, which does not promise the bound either, exceeds it on 9 problems. Against Chandrupatla it is ahead in total on five corpora and level on the sixth. The lead is large on multiple, fractional and nearly flat roots and small on simple roots. It is behind on `fun9` (65 against 55), the near-cancelling class of corpus 1 (491 against 391) and random polynomials with a simple root in total (433 against 419), and level on the smooth equations of the applied corpus.

7.1 Controls: simpler detectors, and the multiplicity known in advance

Two controls ask whether the gain of §6 comes from the recognition by analogy or only from the split (7) with any reasonable detector. Both were recorded, with what the paper would say under every outcome, before the run (`sweep_controls.py`).

Other detectors in A′. A′ is run with only its detector swapped; the memory, the split, the exit and the five readings of §6.4 are unchanged. (a) *Lagouanelle's estimate* $m_L = 1/(1 - \theta)$, $\theta = ff''/f'^2$ from the parabola through the last three points ($m_L = \infty$ for $\theta \geq 1$), with the same memory — once seen, held until the exit, a hysteresis — entering at $m_L \geq 2$ and at $0 < m_L \leq \frac{2}{3}$ and splitting at m_L. (b) *Chandrupatla's acceptance band*, entering when the band rejects the current triple and splitting at the current m_L. (c) *No detector*: the regime is entered at every unkept promise.

Table 10. A′ against its detector swapped: totals on the two canonical sets, on corpus 4 and over the six corpora of §7 (uniform criterion), and A′'s wins/ties/losses on corpus 4 with the exact sign test. No variant breaks the bound.

detector	APS	CHA	corpus 4	six corpora	A′ on corpus 4	p
analogy (A′)	2259	**1074**	**3656**	**13032**	—	
Lagouanelle, with memory	**2235**	1437	4488	15171	84/121/35	$8 \cdot 10^{-6}$
Chandrupatla's band	2682	2526	6236	20668	110/96/34	$1.5 \cdot 10^{-10}$
none	2693	2526	6165	20749	113/87/40	$3 \cdot 10^{-9}$

On corpus 4, the one test, A′ beats all three in total and problem by problem, as the criterion recorded in advance required. The competent alternative is Lagouanelle's estimate: it is slightly better on the canonical set (2235 against 2259), and on corpus 4 A′'s lead over it lies in the multiple-root and fractional-power families (1163 against 1739, 924 against 1207); it is behind on $\log(1+q^2)$ of a cubic (503 against 477) and level on the rest. The band and the regime without a detector are worse than no regime at all. The decision layer by analogy therefore earns its place against the detectors a reader would try first.

The multiplicity known in advance. The second control asks how good the classical transformation is when its multiplicity is given. Library Chandrupatla and Brent are run on $g = \operatorname{sign}(f)\,|f|^{1/m^*}$, with m^* the order read at the known root — the slope of $\ln|f|$ against $\ln|x-r|$ on both sides at $h = 10^{-5}$, rounded to 0.01, and 1 where either side shows no zero (a jump). It is not an achievable method but an upper bound. On the multiple and fractional roots, $m^* \geq 2$ or $m^* \leq \frac{2}{3}$ — the branches of §6.1 where a tangent step loses to a halving:

Table 11. Multiple and fractional roots with the multiplicity given: totals, failures charged. A′ wins none of the 148 problems.

corpus	problems	A′	Chandrupatla on g	Brent on g	Chandrupatla on f
Chandrupatla	20	654	96	66	1068
new corpus 1	40	1229	358	337	1792
corpus 4	88	2159	901	844	4151

With the multiplicity known the transformation is 2.4 to 6.8 times cheaper than A′. Every method, A′ included, stops on an exact zero of the function it evaluates, and on the pure powers of the Chandrupatla set g is exactly linear, so the secant lands on the root: 15 of Brent's 20 runs and 4 of Chandrupatla's end that way there (A′: 2). On new corpus 1 no transformed run ends on a zero, and the transformation is still 3.4 times cheaper. A′ is far ahead of Chandrupatla on f itself, which does not know the multiplicity either. On the remaining problems the transformed and untransformed runs differ little and A′ is close to them: ahead on the canonical sets (2259 against 2640, 420 against 437 for Chandrupatla on g), behind on corpora 1 and 4 (2061 against 1924, 1497 against 1427). So on multiple roots the contribution of A′ is the estimate of the order without knowing it, and the gap to Table 11 is the price of that estimate — large, and the clearest direction for improving the method.

The multiplicity estimated during the run. Between A′ and the oracle lies the natural classical method: Chandrupatla's interpolation on $g = \operatorname{sign}(f)|f|^{1/\hat{m}}$ with $\hat{m}$ estimated during the run. Our implementation of Chandrupatla's method (identical to the library on all 199 problems, and to itself when $\hat{m} \equiv 1$) keeps the true f at its three points and reads $G(v) = \operatorname{sign}(v)|v|^{1/\hat{m}}$ only in the acceptance band and the interpolation; $\hat{m}$ starts at 1 and takes an order read from the last three evaluations whenever that order lies on a branch of §6.1 ($m \geq 2$ or $0 < m \leq \frac{2}{3}$), without reset — the hysteresis of the controls above. The order is read either by Lagouanelle's estimate or by A′'s own analogy (recorded in advance, `sweep_estimated.py`):

Table 12. Chandrupatla on the transformed function: the multiplicity estimated during the run, against A′, the oracle of Table 11 and Chandrupatla on f. Totals, failures charged; the 148 multiple and fractional roots of Table 11, and the simple-root canonical set. The estimated forms have no exit: an order once taken is held to the end of the run. A′ leaves its regime by the exit of §6.2.

method	corpus 4	six corpora	multiple/fractional (148)	APS
A′	**3656**	**13032**	4042	**2259**
Chandrupatla on g, $\hat{m}$ by analogy	3806	15733	2809	3453
Chandrupatla on g, $\hat{m}$ by Lagouanelle	7526	30991	6041	7215
the same, branch $m \geq 2$ only (post hoc)	5568	18372	6540	2960
Chandrupatla on g, m^* known (Table 11)	—	—	**1355**	—
Chandrupatla on f	5717	16481	7011	2595

A′ is ahead of both estimated forms on corpus 4, in total and problem by problem (against the analogy form 126/38/76, $p = 5 \cdot 10^{-4}$), and over the six corpora. The two estimators behave differently in this frame. Lagouanelle's reads the parabola through Chandrupatla's points, which alternate sides of the root; there it reports a fractional order on 115 of the 154 simple roots of the canonical set, holds it, and turns the method into little more than bisection. Restricted after the fact to the branch $m \geq 2$ (post hoc, `POSTHOC_lag_branch_2026-09-22.md`), it no longer breaks, but in this frame it hardly sees a multiple root either: 6540 on the 148 multiple and fractional roots against 7011 for Chandrupatla on f, and 5568 on corpus 4, with A′ ahead 178/34/28. The analogy's order is the better one here too, and on the multiple and fractional roots it does better inside Chandrupatla's interpolation than inside A′'s split: it closes 59%, 58% and 31% of the gap from A′ to the oracle on the Chandrupatla set, corpus 1 and corpus 4 (322 against 654, 721 against 1229, 1766 against 2159). It pays on simple roots, where it has no exit to take a false reading back (3453 against 2595 on the canonical set). Two conclusions follow. First, the order read by analogy transfers to the classical frame and works better there than in A′'s own split: on the multiple and fractional roots Chandrupatla's interpolation with that order beats A′ itself (2809 against 4042). Second, A′'s decision layer as a whole — the detector together with its exit — earns its place: without an exit the transferred detector loses more on simple roots than it gains on multiple ones, and A′ stays ahead on corpus 4 and over the six corpora.

8. Conclusion

A root finder can be derived from a similarity function alone, and the derivation shows where its content sits: not in the shape of S, whose slope at the origin decides which group of functions comes first, but in the *scale* against which similarity is judged — chosen, the method is linear and brittle; taken from the bracket, it is false position; read off the most recent evaluations, it is Newton's step with Muller's order. The safeguard must project the step rather than replace it; the projection's multiplier is a free constant, declared, and a budget the steps earn removes it at a measured price.

The base rule is level with TOMS 748 and 1.5% behind Chandrupatla's method on the canonical set, and clearly behind Chandrupatla on wide brackets (1505 against 2357); the cause is not the order of the model but the decision to give it up.

That decision is a similarity too, of the run to its own past: a pure power repeats its configuration under zoom, an exponential under translation, a simple root tends to the line, and the analogous pair reads the order that decides between the model and a split. The method built on it, A′, is ahead of Chandrupatla on both canonical sets, and on a corpus fixed before its code was frozen it needs 36% fewer evaluations and wins 177 problems to 28. Its detector, the order read by analogy, beats Lagouanelle's estimate, Chandrupatla's band and no detector put in its place, and it transfers: inside Chandrupatla's interpolation on the transformed function it does better on multiple roots than A′'s own split, while only A′'s exit keeps simple roots cheap; with the multiplicity known in advance the classical transformation is 2.4 to 6.8 times cheaper on multiple roots, which measures what estimating the order costs. The claim the measurements support is a lead over Brent, TOMS 748, ITP and ModAB everywhere, within a bound of twice bisection that Brent, TOMS 748 and ModAB do not promise and exceed, and over Chandrupatla a lead that is large where roots are multiple, fractional or nearly flat and small where they are simple — not superiority by the same margin everywhere. `fun9`, near-cancelling terms and smooth applied equations are where it is behind or level.

Future work

The first is the price of estimating the order. With the multiplicity given, the transformation of §6.3 is 2.4 to 6.8 times cheaper than A′ on multiple and fractional roots (Table 11): closing that gap is open.

The second is the steep flank. Where the loss remains — `fun9`, the near-cancelling class, 4 − tan x at its pole — the base step approaches along a flank far steeper than its model, and the half contradiction of §6.4 removes only part of the cost. The third is the one geometric choice of §6.2, the Euclidean metric of the (ξ, φ) plane: a metric derived from the configuration's own sensitivity would remove it.

Code and data availability

Everything behind this paper is released with it: the reference implementation of the base rule and of A′ (`simroot/pure.py`); `simbracket`, a compiled library of A′ (Cython; wheels for Linux and Windows, Python 3.10 or later) whose results equal the reference bit for bit on all 868 problems of the six corpora; the corpora as code (`corpus_*.py`); every sweep script with its results file; the advance records with the commits of Appendix D; and `verify_article.py`, which recomputes or re-reads every number stated in this paper and is part of the test suite. All of it is released under the Apache License 2.0. All numbers come from one Linux machine (§4). Archive (code, data and history): https://doi.org/10.5281/zenodo.22894337 (DOI 10.5281/zenodo.22894337).

Appendix A. Where the evaluations go, and where the similarity is read

A.1 Where the evaluations go

Splitting each run at the geometric mean of the initial width and the attainable tolerance — the log-midpoint of the journey, at which bisection by construction has spent half its evaluations — separates the *approach* from the *endgame*. On Alefeld–Potra–Shi the two-point rule spends 2235 on the approach and 668 on the endgame against Brent's 2175 and 515: its deficit is

almost entirely the endgame, i.e. the order, which is what motivated the three-point scale. On Chandrupatla it spends 1068 and 1161 against Brent's 1343 and 1358, winning both phases.

Two natural explanations of the Chandrupatla advantage were tested and *refuted*. Brent discards 53.3% of its interpolations to its safeguard on the set it wins and only 34.0% on the set it loses, so safeguard rate is not the mechanism; and on Chandrupatla Brent is the bolder of the two — 56.3% of its probes land in the outer tenths of the bracket against our 41.8% — so step length is not the mechanism either.

The mechanism is step *quality*. Reconstructing each method's bracket from the signs it observed, the median probe removes 0.500 of the bracket for every method on Alefeld–Potra–Shi, but on Chandrupatla it removes 0.275 for the similarity rule against 0.058 for Brent and 0.068 for TOMS 748. The bracket here is reconstructed from *every* sign the method has observed — the narrowest interval the evaluations prove, not the one the method keeps; Brent and TOMS 748 always probe inside their own bracket, but 69 and 141 of their probes fall outside this narrowest one and remove nothing from it, and those probes are left out of the median. High-order interpolation is a wager that f is locally polynomial. When the wager pays, a high order wins the endgame; when it fails, the probe removes little. The similarity rule makes the weakest available assumption — a single local slope — and degrades gracefully.

A.2 Where the similarity is read

Everything above reads the similarity in one place: the *length* of the step, $(1 - S(z))$ times the distance to the far end. But a bracketing iteration takes several decisions per step, and the similarity was not consulted in any of the others: which end counts as far (decided by distance), which points the local slope is estimated from (decided by recency), and whether the step is worth taking at all (decided after the fact, by the safeguard). Each of those is a place where a similarity can be read, and the placement turns out to matter more than the form of S.

Three readings were added to the base rule of §3 (tent similarity, three-point scale), each parameter-free and each measured on the protocol of §4; every number below is re-checked against its own sweep file by `verify_article.py`, and every configuration in Table A1 keeps the $2\lceil \log_2(w_0/\varepsilon_x) \rceil + 3$ guarantee (0 violations) and solves all 199 problems.

Where a safeguard other than the projection of §3.1 appears, it is one of the grid's. *safe* is the projection preceded by a step-length test — a step not at most half the step taken two evaluations earlier is replaced by a halving — with the projection still carrying the bound. *credit* replaces the scheduled allowance of (3) by one earned step by step: a surplus is banked whenever a step shrinks the bracket by more than half and drawn on when it shrinks it by less, which bounds the run by bisection plus one step. *bracket* and *both* are the replacing safeguards of §3.1, which keep a bound at the price of the order; *step* alone and *none* carry no bound.

1. **Trust before the step.** The step trusts a local slope, so the question of trust is whether that slope has settled: if two consecutive estimates have opposite signs the local model is not established, and the evaluation is spent halving instead. A sign is compared, not a magnitude.
2. **Which points the model uses.** For a linear f the ratio of positions $\xi = (x - u)/(v - u)$ and the ratio of values $\varphi = (f(x) - f(u))/(f(v) - f(u))$ coincide, so $|\varphi - \xi|$ measures how unlike a linear configuration a triple is. Among the declared candidate triples the one minimising it is *selected* — an argmin over what is already in hand, with no threshold.
3. **Which end is far.** Read by dissimilarity instead of distance: the end whose $|f|$ is larger is the one less similar to the target.

Table A1. Reading the similarity in one more place is worth more than changing S. Evaluations of f; failures charged the full budget.

configuration (all with the guarantee, 199/199)	APS-154	CHA-45	both
base rule: tent, 3-point, projection $c = 2$ (§3)	2634	2357	4991
constant-free variant: earned projection (§3.1)	3377	2073	5450
(1) trust before the step, projection	2663	2170	4833
(1)+(2) points chosen by shape, *safe*	**2337**	2177	4514
(1)+(3) far end by dissimilarity, projection	2539	**1864**	**4403**
(1)+(2)+(3), *safe*	2695	1928	4623
Chandrupatla	2595	1505	4100
TOMS 748	2636	3088	5724
ITP, per-problem oracle	3068	1675	4743

Two results here are worth stating plainly. First, the canonical set improves from 2634 to **2337**, which is below every method measured, including Chandrupatla's 2595, with the worst-case guarantee intact. It hardly feels the declared multiplier of §3.1: 2337 for every multiplier from 1.5 to 4, 2351 at 1.25 (`measure_records_vs_multiplier.py`). Second, on the two sets together the 4403 of reading (3) passes ITP's oracle (4743) but not Chandrupatla (4100): the remaining gap is entirely on the wide-bracket families, where Chandrupatla needs 1505 against our 1752 (reading (3) under *safe*).

The readings do not compose. Taken together, (2) and (3) give 2695 and 1928 — worse than either alone on the canonical set, and on the wide-bracket set worse than (3) alone. Two selections that are each an improvement interfere, which is a reason to measure composition rather than assume it.

Which similarity stands in the step decides whether a reading pays. The readings above, and the decision *defect* of §A.4, compare quantities directly — an ordinal judgment of which is more similar, in which every decreasing S would decide alike. Their gains nevertheless depend on the S that sets the step (`audit_similarity_places.py`, on the same sweeps): on the canonical set the triple chosen by shape gains 493 evaluations with the tent in the step and loses 250 with $1/\sqrt{1+z^2}$; *defect* gains 283 with the tent and loses 215 with $1-|z|^{1/3}$. The records of this section are the pairing of the tent step with these judgments, not the judgments alone.

Under the constant-free earned projection the same readings gain little or lose. Reading (1) gives 3316 and 2050, (1)+(2) 3564 and 2074, (1)+(3) 3666 and 2052, all three together 4163 and 2087, against 3377 and 2073 for the constant-free variant itself: the earned budget binds early on the canonical problems, and what the readings buy under the projection is mostly room that the earned budget does not grant.

A.3 The similarity function belongs to the place, not to the method

Everything above varies *where* the similarity is read while keeping one function S for the whole method. That is itself an assumption, and it is measurable. Proposition 1 says that the ratio similarity scaled by the far end is false position; what it does not say is *why* false position stalls. Write $|f| \sim Ke^m$ near the current point, $e = |x - r|$. Then the residual ratio against the far end needs no slope at all,

$$z \;=\; \frac{|f(x)|}{|f(x_{\text{far}})|} \;=\; \Big(\frac{e}{E}\Big)^m, \qquad \text{so} \qquad z^{1/m} \;=\; \frac{e}{E}, \tag{5}$$

and under the ratio similarity $S = 1/(1 + |z|)$ the step fraction is $z/(1 + z)$, which is exactly inverse linear interpolation. False position therefore reads a dimensionless ratio *at order one* on an approach of order m: its three-century-old stall is a mismatch between the similarity function and the place it is read in. The exponent $1/m$ is not a constant — $m = 1/(1 - ff''/f'^2)$, Lagouanelle's estimate of the multiplicity [13] with the derivatives replaced by divided differences, comes from three evaluations already paid for, as in item 5 of Appendix C, and is 1 whenever the power model does not apply.

Measured over the grid, the correction behaves as that reading predicts and not as a universal improvement. Paired inside each (function, safeguard) cell of the far-end reading it is cheaper in 33 of 63 cells — indistinguishable from a coin. But in the one pairing where that reading *is* exact inverse interpolation, the ratio similarity, it is cheaper under all seven safeguards:

Table A2. The measured order in the step place, on the far-end reading with the ratio similarity $1/(1 + |z|)$. Cheaper in 7 of 7 safeguards; in 33 of 63 cells over all nine functions.

safeguard	APS, $m = 1$	APS, measured m	CHA, $m = 1$	CHA, measured m
both	4331	**3485**	2469	**2306**
bracket	4418	**3352**	2671	**2428**
credit	4999	**4495**	2064	**1933**
projection	6783	**6176**	4057	**3751**
safe	5803	**5040**	3218	**3032**
step	6251	**5362**	3371	**3146**
none	10676	**9855**	8265	**7424**

Two limits must be stated with it. The estimator is *biased low*: on a pure cube, with no noise at all, it returns 2.830 rather than 3, because a parabola through three points carries only the second difference. The bias is towards $m = 1$, that is towards the uncorrected reading, which is the harmless direction, but it is a bias. And the correction does not rescue the far-end reading as a method: the best cell it reaches on the wide-bracket set is 1933, against 1752 for reading (3) above and 1505 for Chandrupatla. In the *slope* reading of §3 the same exponent is harmful, 2634 → 2703, which is the same statement read from the other side — a function that does not belong to a place does not become right by being applied there.

A.4 The decision place

Three places have now been measured: the step length, the far end, and the points the model is built on. The fourth is the *decision* — whether the local model deserves this evaluation or halving is the better use of it. Chandrupatla's method is strong exactly there: it refuses its model on most steps by the acceptance band $1 - \sqrt{1 - \xi} < \varphi < \sqrt{\xi}$, with ξ and φ the ratios of positions and of values defined above. The rule of §3 has no similarity in this place at all: a sign gate and a projection that repairs *after* the step.

Three readings of that place were measured, each on quantities already in hand and each without a constant. Two are ours: refuse when the configuration is less like a linear one than the iterate is off-centre, $|\varphi - \xi| > \min(\xi, 1 - \xi)$ ("defect"); and refuse when the configuration's similarity to linear falls below the iterate's similarity to the root, $S(|\varphi - \xi|) < S(z)$, with the *same* S in both places ("simpair"). The third is *Chandrupatla's own band*, placed inside our step rule. It is not a proposal; it is an instrument, and it answers whether the remaining gap lives in this place.

In total, no. With Chandrupatla's band in our step, the wide-bracket set costs 2101 under the projection and 1942 under *credit*, against that method's own 1505, while the canonical set rises from 2634 to 5242. The band is not idle — it refuses 63.5% of steps — so this is a decision that was taken and did not pay *in total*. Per family it did pay, and where: it closes almost the whole

gap on the flat families and gives back more than that on the families where our step is good. §A.6 takes this apart; the short form is that the decision place does hold most of the gap, but no reading measured so far makes that decision correctly for *our* step.

Our own readings do not close it either (best 1941 on the wide-bracket set), but "defect" gives the second-best canonical total: **2351** under the projection of the base rule, next to the 2337 of reading (2) and below every method measured, with the guarantee intact, and between 2343 and 2353 for every multiplier from 1.25 to 4. Under the earned projection it gives 3354 and 2002 (5356 against 5450), the best of the readings of this section without a constant; the flat regime of §6 does better still. It refuses only 6.6% of steps, the fewest of the three. Over both sets it costs 4697 against the 4403 of reading (3), because it gives back on the wide brackets (2346) what it gains on the canonical problems — so it is a second point on the same trade, not a replacement.

A.5 Why the criterion is a sign change: six problems the published root cannot score

Chasing the remaining gap to Chandrupatla led to a property of the benchmarks themselves. On the five problems of the wide-bracket family `fun7`, and on the single canonical problem `aps.13.00`, the declared root is 0 while f vanishes *identically* on a neighbourhood of it in floating point: $f(0 \pm t) = 0$ for every t up to at least 3.2×10^{-2}, because the exponential factor underflows — at the returned points $|f(x)|$ is below the smallest normal double. Every method therefore stops after 8 to 28 evaluations at a point where $f(x) = 0$ *exactly* — a genuine root — and every method is then scored as having failed, because the harness asks for $|x - \text{root}| \le 10^{-6}$ while the returned point sits up to 2.8×10^{-2} away.

The consequence is arithmetic. Each such problem charges the full budget of 200 evaluations to every method, so about 1150 evaluations of every two-set total under the harness — ours and every competitor's — are this artefact, and the "153 of 154" and "40 of 45" that every method reports under the harness are statements about the scoring rather than about the methods. This is why every number in this paper uses the bracketing criterion of §4, a sign change within 10^{-6} of the returned point or $f = 0$ there: under it Chandrupatla, TOMS 748, ITP, bisection and every configuration of Table A1 solve all 199, Brent 197, Dekker and Illinois 178 and 177.

A.6 Where the wide-bracket limit actually is

The remaining gap looks diffuse — per family +57, +49, +42, +39, +30 and smaller — but part of it has one address. The `fun1` family is the same function with brackets of growing width, and our cost against Chandrupatla's across them is 7/8, 12/12, 19/15, 59/24, 63/44: we tie or win while the bracket is narrow and lose progressively as it widens. The endgame is not the cause; the cost is incurred far from the root: 59 evaluations on a 2×10^4-wide bracket is what bisection itself needs ($\lceil \log_2(2 \times 10^{16}) \rceil \approx 54$), i.e. there we advance by about one halving per evaluation while Chandrupatla advances by about 2.3.

It is not the guarantee that caps us. (The five costs above are under the *safe* safeguard; under the projection of the base rule they are 7, 12, 30, 46, 57.) With the safeguard removed entirely the two widest brackets, `fun1.4` and `fun1.5`, cost 46 and 57: better than bisection's 57 and 77, still far from Chandrupatla's 24 and 44. Unguarded, the rule is worse than bisection elsewhere — 61 and 68 against 45 on the flat problems `fun3.1` and `fun5.1` — which is what the guarantee is for.

Nor is it that the step fails to be relative to the bracket. With the tent similarity $1 - S = \min(1, |f|/(|\hat{f}'|D))$, so by (4) the step is exactly $\min(D, |f|/|\hat{f}'|)$: a *fraction* of the distance to the far end, and the similarity is that fraction. Nor does the iterate creep along one side

without moving the proven bracket: measured probe by probe, 91.5% of our evaluations on the 2×10^4-wide bracket do shrink the bracket, slightly *more* often than Chandrupatla's 87.5%.

What differs on `fun1` is how much each probe removes. Ours shrinks the bracket by 0.93 halvings per evaluation there, Chandrupatla's by 2.30. On a wide bracket of a convex function the Newton step systematically undershoots, because a linear model of the value axis cannot see the curvature that separates the iterate from the root; a model that carries the curvature — inverse quadratic interpolation — lands far closer.

That mechanism is right for `fun1` and *wrong as a statement about the whole gap.* Replaying our own trajectories on the 35 problems where we lose, and placing both models on the *same* evaluations — one extra evaluation each, from the same state — splits the 262 evaluations of the gap in two. On `fun1` (58 of them, about a fifth) inverse quadratic interpolation from our own three points removes far more of the bracket than our step did: 10.37 halvings against 0.62 on the 2×10^4-wide bracket. There the gap is the order of the model, and the reason is structural: our step $x + (1 - S)(x_{\text{far}} - x) = S\,x + (1 - S)\,x_{\text{far}}$ is a convex combination of two points, while the quadratic interpolant's Lagrange weights at 0 always include exactly one *negative* weight when the triple brackets the root (verified on all 431 non-degenerate states; in the other three the negative weight underflowed to -0). A similarity in $[0, 1]$ cannot produce a negative weight — Proposition 1 extended from two points to three.

But 172 of the 262 — two thirds — sit on `fun3`–`fun6`, which are pure powers of order 3, 5, 9 and 19, and there *no* interpolating model works: our step removes 0.41–0.75 halvings of the bracket per evaluation, inverse quadratic interpolation from the same points only 0.01–0.17. Chandrupatla's method does not win those families with its model — its traced run bisects 92, 96, 100 and 100% of its steps. We lose them because our step is slower than bisection there and our guarantee, at most twice bisection, licenses exactly that. Its band inside our step confirms it family by family: under the projection it takes `fun4`, `fun5`, `fun6` from 573, 456, 456 to 296, 239, 228 — against its own 297, 233, 233 — and costs +164, +111, +170, +144 on `fun1`, `fun2`, `fun8`, `fun9`, where it refuses steps that were good.

So most of what remains is not the order of the model but the decision *when to give the model up*, and it is open because every detector measured so far confounds "the model is useless here" with something else: the band with "the configuration is not shaped for inverse quadratic interpolation", bracket progress with "the step has not moved the bracket yet" (4258 on the canonical set, Appendix C), and the local order with nothing at all — $m \geq 2$ is exactly the condition under which a tangent step loses to a halving, since it lands at $e(1 - 1/m)$, and it is clean (0.3% of canonical states) but blind: it reports $m \geq 2$ on none of the states of families whose order is by construction 3 to 19. For the `fun1` part, three parameter-free ways round the model order have been measured and all three fail in the same way, buying the wide brackets at the price of the canonical set: a fixed fraction of the bracket (the cube-root similarity), a logarithmic change of variable, and the curvature-corrected scale of Appendix C item 5. The fourth way is to model the curvature directly, which is inverse quadratic interpolation — a different method rather than another reading of this one. Section 6 shows that the decision is reachable without a new model: by a similarity of the run to its own past.

On the wide-bracket set our shape reading (2177) is *worse* than bisection (2136); only the dissimilarity reading (1752 under *safe*) is clearly ahead, which is the same statement seen from the other end: where the step cannot march, the reading that helps least is the one that only improves the local model.

Every statement in §§A.5–A.6 is reproduced by `diag_fun7_root.py`, `diag_cha_gap.py`, `diag_who_caps_us.py`, `diag_model_order.py`, `diag_model_order_checks.py`, `diag_flat_detectable.py` and `diag_band_by_family.py`, whose outputs are kept beside them.

Appendix B. Wall-clock time, and why it is not the cost model

Evaluations of f are the cost model because in any real problem f is the expensive part. Wall-clock time measures something narrower — the method's own arithmetic per step — and it is implementation- and language-bound. We report it because a reader will ask, and because what it shows is instructive.

On the 199 problems as they stand, every f being a formula of a few operations, the base rule is the **slowest** of the pure-Python methods compared, at 32.8 ms against 18.4 for Brent, 21.1 for ITP, 14.2 for Ridders and 22.9 for bisection — while using the fewest evaluations of all of them, 4991 against 5932, 5675, 6421 and 9474. Bisection, needing nearly twice the evaluations, finishes in two thirds of our time. (These are the evaluations the timing loop actually spent; for the base rule and bisection, which solve every problem, they equal the totals of §5; Brent here is our pure-Python implementation, which is what is timed.)

That inversion is the point. Making f progressively more expensive by a controlled amount of arithmetic and timing the sweep directly — nothing subtracted, nothing modelled — the base rule overtakes ITP once one call to f costs about 9 µs and Brent at about 20 µs, which on this machine is of the order of a hundred arithmetic operations. Below that the ranking is decided by per-step overhead; above it, by evaluations.

Three caveats, all of them limitations of the measurement rather than of the methods. Timings vary by tens of percent between runs even on an idle machine. An attempt to separate our per-step cost into "algorithm" and "Python call overhead", by timing a flattened rewrite of the same rule, produced a variant that diverged from the original on 42 of the 199 problems and is not quoted. Nothing is subtracted: estimating a per-evaluation overhead by subtracting the measured cost of f returns negative overheads for four of six methods.

Appendix C. Negative results

1. **The prototype-aggregation form of the method loses.** A richer construction — prototypes over *pairs* of evaluated points, softmax weights over the whole memory, similarity on both the value and the abscissa axis — leads on eight hand-picked problems but loses decisively on the real set: 0 wins, 6 ties and 147 losses against Brent. The simplest form of the idea won.

2. **Every safeguard that replaces the step destroys the order.** Bracket-based and step-based replacement both make the three-point rule converge linearly and cost 7068 and 7046 evaluations on Alefeld–Potra–Shi. Only projection preserves both. We record this because it is the trap the construction naturally falls into.

3. **Zero slack in the projection is not a degenerate case.** A worst case of exactly bisection might be expected to leave the projection no freedom and make the guard behave as bisection. Measured (Table 1a), multiplier 1 costs 4625 and 2039, against bisection's 7338 and 2136 — and on the wide-bracket set it beats the multiplier of 2. It is the most expensive multiplier on the canonical set.

4. **Regimes exist, at the level of a function family, and cannot be exploited by a switch.** The cube-root similarity beats the base rule on all five members of Chandrupatla's `fun4` — 32/32/40/47/59 against 89/93/113/125/153, a factor of 2.6 — and consistency across a family at different parameters is the signature of structure rather than luck. But the same similarity over the whole Chandrupatla set costs 3204 against 2357, so a switch

chosen per benchmark is worse than no switch at all: 5838 against 4991 over both sets. The scale axis offers only 5.3% headroom even to an oracle handed the best scale per problem, two thirds of it inside a single problem family, and the winners are not separated by anything the method can see before it commits — the derived scale wins on bracket widths from 10^0 to 10^{12}.

5. **The phase detector: derivation sound, estimator not.** Redoing the contraction law without the assumption $|f| \approx |f'(r)|e$ — taking $|f| \approx Ke^m$ instead — does produce the march exponent on its own, $p = L/(L - \ln m)$ with $L = \ln(e/D)$, giving $p = 1$ at a simple root and dropping to 0.50 and 0.23 as the approach flattens, which is the neighbourhood of the cube root that wins those families. Substituted back it collapses to the multiplier m, measurable as $m = 1/(1 - \theta)$ with $\theta = ff''/(f')^2$ from three divided differences and no constant — which is Lagouanelle's multiplicity estimate [13]. It improves the eight standard problems (82 against 84) — and loses on the full benchmark, 5336 against 4991. The estimator is the failure: on Alefeld–Potra–Shi, where almost every root is simple and the truth is $m = 1$, it reports $m > 1$ on 27.9% of steps with excursions to $m = 34.8$, against 35.8% on the flat families. 27.9 against 35.8 is no separation. A cruder variant, correcting by the fraction of its own promise the previous step kept, fails everything: the convergence becomes sublinear and the standard problems double to 161.

6. **The harness scores one canonical problem unsolved for every method**, `aps.13.00`, $f(x) = xe^{-1/x^2}$, all of whose derivatives vanish at the root and which underflows to exactly zero throughout $|x| < 0.086$. Every method returns a point where $f(x) = 0$ exactly — a genuine root — which the harness scores as a failure because it lies more than 10^{-6} from the declared root. With the five `fun7` problems of §A.5 it is why this paper scores by a sign change.

7. **The symmetric ratio similarity degenerates.** For $S(x, y) = \min(|x|, |y|)/\max(|x|, |y|)$ with y the target 0, $S \equiv 0$; the only usable reading is $1 - |f|/\sigma$, which saturates and stagnates (Table 1, last row).

8. **Placement beats form, and the wrong placement is expensive.** The same "refuse before the step" idea as reading (1) of §A.2, with refusal decided by *promised bracket progress* instead of by the shape of the model, costs 4258 on the canonical set against 2337 — a difference of 1921 evaluations for one choice of criterion. A secant step routinely collapses the error while barely moving the proven bracket, so bracket progress rejects steps that were excellent; this is the same mechanism that makes the bracket guard of §3.1 cost 7068.

9. **Making the scale independent of the bracket does not help either.** The natural objection to the derived scale is that it defeats the very principle of a similarity step: with $\sigma = |\hat{f}'|D$ the normalised residual is $|f|/(|\hat{f}'|D)$, which on the 2×10^4-wide bracket of $x^3 - 2x - 5$ at $x = 10^4$ equals 0.167, so $S = 0.833$ — "similar" — and the step is small although the root is some five thousand step-lengths away. The similarity is reading what fraction of the *bracket* the Newton step is, not how far the iterate is from the root. Selecting the scale between the two quantities already computed, $\sigma = \min(|\hat{f}'|D, \max(|f(A)|, |f(B)|))$ — no new constant, a minimum over what is in hand — does give a full-bracket step far away and Newton's step near the root, and on eight hand-picked wide problems it improves 403 evaluations to 390. Over the whole grid it is worse on both sets: its best cell is 3101 on the canonical set against 2539 for the best uncapped cell of the same grid, and 1766 on the wide-bracket set against 1752.

10. **Reading the similarity at the measured order does not cure false position.** Since false position's stall is a similarity read at order one (§A.3), the obvious hope is that reading it at the measured order is the parameter-free version of the classical fix. It

is not. Unguarded, the correction takes the far-end reading from 127/154 and 10676 to 129/154 and 9855 — two problems and eight per cent — where Illinois [7], at the price of one constant (halve the stale ordinate), reaches 153/154 and 7045. A sixth of the gap is not the gap. The correction is real and it is in the right place; it is simply much smaller than a constant that was chosen for exactly this failure. *Our* 10676 and the 10560 of Table 2 are not the same algorithm — our iterate starts at the midpoint of the bracket, plain false position at an end — and the two must not be differenced.

11. **A logarithmic change of variable does not buy the wide brackets.** Working in $u = \operatorname{sign}(x-c)\log(1+|x-c|/\varepsilon_x)$ makes the step multiplicative, which looked like the right diagnosis of the flat families — for the same similarity, scale and safeguard ($(1+|z|)^{-2}$, two-point, *safe*) the wide-bracket set improves from 2759 to 2598 — but the canonical set gets worse, 5635 to 6749, against the 2634 of the base rule. A march needs a *ratio* of distances while an identity on narrow brackets needs an *absolute* unit, and no parameter-free unit provides both. §A.6 shows the diagnosis itself was wrong: on the flat families the right move is to halve, not to march.

12. **Reading the unkept promise one step at a time does not pay.** The flag of §6.1 sees the flat regime, but a halving in place of each flagged step takes the canonical set from 2634 to 2839 and the wide-bracket set only from 2357 to 2243: a simple root far from convergence breaks its promises too. Requiring the promise to fail twice in a row keeps the canonical set (2613) but fires one step late, before the crossing, and replaces the most useful step of the cycle (wide brackets 2388); requiring a creep in one direction as well gives 2688 and 2399.

13. **A halving that sticks once begun wins the flat families and loses everything else.** Without an exit the flat families fall to 1057 — pure bisection's level — and the canonical set rises to 6707, because a false entry now lasts to the end of the run. Splitting at the order read from the failed promise, $(1-1/m)^m = r$, changes nothing: that order is $m = \infty$ in 163 of 174 entries (6591).

14. **The model's own point cannot tell a flat root from a simple one.** Judging the evaluated point against the line through the bracket ends and against 0 — the exit judgment of §6.2 — also at *entry* gives 2820 and 1532 against 2935 and 1546 for the exit alone: the model aims at the root, so f is small there by construction and looks flat at a simple root as readily as at a power. The same judgment works at a split, which the model did not choose.

15. **The decision reading "defect" does not transfer.** Combined with the regime it was the best configuration on the two development sets (§A.4's defect reading with the flat families of §6), and it passed every criterion recorded for it there. On new corpus 1 it did not: per problem 95/33/78 against Chandrupatla ($p = 0.22$), losing the class of exponential flanks 9 to 26. It is therefore not part of A′.

16. **A component rejected alone can pay in combination.** Remembering the analogy for the run was rejected by its own advance record, under the harness scoring of the time (canonical set 2814 against 2806 without the regime; under the sign-change criterion the two are level, 2633 against 2634), and with the split of §6.3 it is the better form (with the step pattern, wide brackets 1250 against 1388 for the analogy read in one state). Composition was measured, not assumed, throughout — which is also why §A.2's readings (2) and (3), each an improvement, are not combined.

Appendix D. Advance records and the frozen code

Every rule of §6 and every test of §7 was written down, with its criteria and stop rule, before its code and its run. Table D1 gives, for each document, the commit that first added it and the commit that first added the results its run wrote, both read from git by `check_prereg_order.py`: 25 of the 30 were committed before their results; four were committed together with them, so git cannot order them; one was decided on its smoke run and has no results file. We call these documents advance records, not pre-registrations: they live in the author's repository, and their order relative to the runs is attested only by the author's git history, whose timestamps the author sets; no external registry was used. A deposit with an external timestamp (Zenodo, OSF) before a run would remove that limitation for future tests. Before publication the history was rewritten once, to remove infrastructure details (a server address); the commits cited here are those of the published history, in which `check_prereg_order.py` recomputes Table D1. The code of A′ was committed (`90adc15`) after corpus 4 was defined (`f9b8f3a`) and before its results (`5a15ad0`); `simroot/pure.py` did not change in between.

Table D1. Advance records, in the order they were written.

advance record	commit	results file	commit	order
logarithmic_2026-09-20	8b043e5	sweep_log.json	b0f1805	before
prestep_gate_2026-09-21	3a7809a	sweep_gate.json	b0170f5	before
slope_gate_2026-09-21	4bb932c	sweep_slope.json	4bb932c	same commit
points_and_direction_-2026-09-21	77ae329	sweep_points.json	b73f15f	before
similarity_per_slot_-2026-09-21	7cdaad2	sweep_order.json	7cdaad2	same commit
scale_selection_-2026-09-21	a8594d9	sweep_cap.json	a8594d9	same commit
decision_place_-2026-09-21	10e536d	sweep_decision.json	10e536d	same commit
step_extrapolation_-2026-09-21	4c10d18	results_diag_step_-extrapolation.txt	767e493	before
extrapolate_in_method_-2026-09-21	02c6259	sweep_extrap.json	2cd0925	before
extrapolation_gates_-2026-09-21	a0275fc	sweep_extrap2.json	539654c	before
empty_step_2026-09-21	faae645	results_diag_empty_step.txt	5855bb8	before
outcome_memory_-2026-09-21	aaf8d06	results_diag_outcome_-memory.txt	cff62c5	before
flat_regime_2026-09-21	1681f28	sweep_flat.json	fa3e02c	before
flat_twice_2026-09-21	8f19bca	sweep_flat2.json	de719aa	before
flat_creep_2026-09-21	84158b3	sweep_flatcreep.json	3b5ebe5	before
flat_stick_2026-09-21	d3cd63f	sweep_flatstick.json	cf0fc6e	before
flat_exit_2026-09-21	3018ed1	sweep_flatexit.json	971f43b	before
flat_judge_2026-09-21	d0bb3cd	sweep_flatjudge.json	f2d597c	before
flat_analogy_2026-09-21	4e03643	sweep_flatanalog.json	971f43b	before
flat_analogy_mem_-2026-09-21	94f7d47	–	–	decided on the smoke run, no results file
defect_contra_flat_-2026-09-21	8f9ae50	sweep_defcontra.json	ed5a176	before
flat_order_point_-2026-09-21	97745a2	sweep_orderpt.json	2f52cf5	before
flat_exp_limit_-2026-09-21	a3a91a0	sweep_explimit.json	574ba85	before

advance record	commit	results file	commit	order
`half_contra_2026-09-21`	`4ad04ca`	`sweep_half.json`	`5f9d99b`	before
`out_of_sample_-2026-09-21`	`e576b4c`	`sweep_oos.json`	`08e3b57`	before
`roots_jl_2026-09-21`	`be59001`	`sweep_rootsjl.json`	`84b8857`	before
`applied_2026-09-21`	`9754df0`	`sweep_applied.json`	`4924e99`	before
`close_sub_2026-09-21`	`f9b8f3a`	`sweep_fourth_fourth.json`	`5a15ad0`	before
`modab_2026-09-21`	`1663066`	`sweep_modab.json`	`03ab52f`	before
`controls_2026-09-22`	`911f370`	`sweep_controls.json`	`e154d3a`	before

References

1. G. E. Alefeld, F. A. Potra and Y. Shi, "Algorithm 748: Enclosing zeros of continuous functions," *ACM Trans. Math. Softw.*, vol. 21, no. 3, pp. 327–344, 1995.
2. R. P. Brent, *Algorithms for Minimization without Derivatives.* Englewood Cliffs, NJ: Prentice-Hall, 1973.
3. T. R. Chandrupatla, "A new hybrid quadratic/bisection algorithm for finding the zero of a nonlinear function without using derivatives," *Adv. Eng. Software*, vol. 28, no. 3, pp. 145–149, 1997.
4. I. F. D. Oliveira and R. H. C. Takahashi, "An enhancement of the bisection method average performance preserving minmax optimality," *ACM Trans. Math. Softw.*, vol. 47, no. 1, pp. 1–24, 2021.
5. C. Ridders, "A new algorithm for computing a single root of a real continuous function," *IEEE Trans. Circuits Syst.*, vol. 26, no. 11, pp. 979–980, 1979.
6. V. Sitnikov, "Parameter-free similarity control," preprint, 2026.
7. M. Dowell and P. Jarratt, "A modified regula falsi method for computing the root of an equation," *BIT*, vol. 11, pp. 168–174, 1971.
8. G. Pólya, *How to Solve It.* Princeton, NJ: Princeton University Press, 1945.
9. T. J. Dekker, "Finding a zero by means of successive linear interpolation," in B. Dejon and P. Henrici (eds.), *Constructive Aspects of the Fundamental Theorem of Algebra.* London: Wiley-Interscience, 1969, pp. 37–51.
10. N. Ganchovski and A. Traykov, "Modified Anderson-Björck's method for solving non-linear equations in structural mechanics," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1276, 012010, 2023.
11. M. Galassi et al., *GNU Scientific Library Reference Manual*, 3rd ed. Network Theory, 2009; root-finding test suite `roots/test.c`.
12. J. Verzani et al., *Roots.jl*, Julia package; bracketing benchmark `test/benchmark-bracketing-methods.jl`, commit cf1d256, 2026.
13. J. L. Lagouanelle, "Sur une méthode de calcul de l'ordre de multiplicité des zéros d'un polynôme," *C. R. Acad. Sci. Paris Sér. A*, vol. 262, pp. 626–627, 1966.
14. L. B. Rall, "Convergence of the Newton process to multiple solutions," *Numer. Math.*, vol. 9, pp. 23–37, 1966.
15. E. Hansen and M. Patrick, "Estimating the multiplicity of a root," *Numer. Math.*, vol. 27, pp. 121–131, 1976.
16. E. Schröder, "Ueber unendlich viele Algorithmen zur Auflösung der Gleichungen," *Math. Ann.*, vol. 2, pp. 317–365, 1870.
17. J. F. Traub, *Iterative Methods for the Solution of Equations.* Englewood Cliffs, NJ: Prentice-Hall, 1964.
18. S. Weerakoon and T. G. I. Fernando, "A variant of Newton's method with accelerated third-order convergence," *Appl. Math. Lett.*, vol. 13, pp. 87–93, 2000.

19. A. Cordero and J. R. Torregrosa, "Variants of Newton's method using fifth-order quadrature formulas," *Appl. Math. Comput.*, vol. 190, pp. 686–698, 2007.
20. N. Ganchovski, O. Smith, C. Rackauckas, L. Tomov and A. Traykov, "Improvements to the Modified Anderson–Björck (modAB) root-finding algorithm," *Algorithms*, vol. 19, 332, 2026; package `pymodab` 1.0.9.
21. J. C. P. Bus and T. J. Dekker, "Two efficient algorithms with guaranteed convergence for finding a zero of a function," *ACM Trans. Math. Softw.*, vol. 1, no. 4, pp. 330–345, 1975.
22. J. M. Fernández-Díaz and C. O. Menéndez-Pérez, "A common framework for modified Regula Falsi methods and new methods of this kind," *Math. Comput. Simul.*, vol. 205, pp. 678–696, 2023.
23. A. Cordero, B. Neta and J. R. Torregrosa, "Memorizing Schröder's method as an efficient strategy for estimating roots of unknown multiplicity," *Mathematics*, vol. 9, 2570, 2021.