Tales from the coal face

Industry-grade function approximation

Peter Jäckel

VTB Europe SE

October 2019

Outline

Why

- Financial calculations often result in "closed-form" solutions that involve special functions such as $\mathrm{erf}()$, $\mathrm{erfc}()$, or $\mathrm{erfcx}()$, and many others.
- *However*, on a computer, all maths has to be reduced to *simple additions*. *Even multiplications* (done by the aid of the logarithmic scale).
- *Speed is important*.
  Even for the inverse cumulative normal function $\Phi^{-1}(\cdot)$, some firms developed in-house versions that use CPU-specific low-level vector floating point instructions (mainly for their Monte Carlo engine).
- Some software providers' business is based on the replacement of computationally intensive evaluations by (barycentric Chebyshev) interpolation(!).
- We also find many inverse problems, where nonlinear equations must be solved, thus implicitly defining new (special) functions, e.g., (the reduced form of) implied normal volatility [Jäc17].

➡ *Parameter-dependent problems*

The need for speed: Implied Black volatility.

- A well known tier 1 bank identified that a significant portion of their server farm time was spent in the calculation of Black implied volatility.
- They engaged a team of academics to find an efficient implied Black volatility algorithm.
-  (plug warning...)

I recommend "Let's Be Rational" [Jäc15] which does this in

less than 700 nano-seconds

on an i5-7200U (fanless notebook CPU) for all possible parameter input values to full *attainable*[1] standard 64-bit floating point precision.

[1]Note that the input values may not admit the inference of all mantissa digits due to intrinsic loss of precision such as is the case for in-the-money options.

The need for speed: Strike-from-delta-with-premium in FX.

- In FX, for many currency pairs [RW10], implied volatility $\hat{\sigma}$ is quoted over *delta-with-premium*[2]

$$\tilde{\Delta} := \Delta - B(F, K, \hat{\sigma}, T, \theta)/F \tag{2.1}$$

where $B(F, K, \sigma, \theta)$ is the Black formula and $\theta := \pm 1$ for calls/puts.

- In actual calculations, we need to know what strike K is implied by the given volatility and delta-with-premium $\tilde{\Delta}$.

[2]We assume forward deltas which can be imputed from spot deltas where required.

- After reduction to convenient form, this means we need to solve

$$f(y;\alpha) = \ln|2 \cdot \tilde{\Delta}| + \frac{\alpha^2}{2} \tag{2.2}$$

with

$$f(y;\alpha) := \ln(2 \cdot \Phi(-y)) + \alpha \cdot y \tag{2.3}$$

$$\alpha := \theta \cdot \hat{\sigma} \cdot \sqrt{T} \tag{2.4}$$

$$y := \ln(K/F)/\alpha + \alpha/2 \tag{2.5}$$

for y, and then set

$$K = F \cdot \mathrm{e}^{\alpha y - \frac{\alpha^2}{2}} \, . \tag{2.6}$$

- Conventional wisdom resorts to iterative solvers here: often, the Brent algorithm[3] is used in this context.
- *We can do better* (: to be published...).

[3]which should only ever be a method of last resort

➡ Simple classic methods

- Everybody knows Taylor expansions.

$$f(x) = f(x_0) + f'(x_0)\cdot(x-x_0) + f''(x_0)\cdot\frac{(x-x_0)^2}{2} + f'''(x_0)\cdot\frac{(x-x_0)^3}{6} + \ldots \tag{3.1}$$

- Usefully, and arguably less well known, when we need the inverse of

$$y = f(x)\ , \tag{3.2}$$

near x_0, we can use the Lagrange inversion theorem:

$$x = x_0 + \sum_{n=1}^{\infty} \frac{g_n}{n!}(y - f(x_0))^n \tag{3.3}$$

$$g_n = \lim_{x\to x_0}\left[\frac{\mathrm{d}^{n-1}}{\mathrm{d}x^{n-1}}\left(\frac{x-x_0}{f(x)-f(x_0)}\right)^n\right] \tag{3.4}$$

$$\Rightarrow \quad g_1 = \frac{1}{f'(x_0)}\ , \quad g_2 = -\frac{f''(x_0)}{f'(x_0)^3}\ , \quad g_3 = \frac{3f''(x_0)^2 - f'(x_0)f'''(x_0)}{f'(x_0)^5}\ , \quad \ldots$$

See also [AS84, Formula 3.6.25] for the first 7 explicit terms.

Consider a function that permits a Taylor expansion[4]

$$f(x) \ = \ \sum_{i=0}^{\infty} a_i x^i \tag{3.5}$$

with positive convergence radius (remember $\mathrm{e}^{-\frac{1}{x^2}}$?).

We usually[5] find that $f(x)$, instead of by its N^{th} order Taylor expansion

$$f(x) \ \approx \ \sum_{i=0}^{N} a_i x^i\ , \tag{3.6}$$

is better approximated by a Padé$_{(m,n)}$ expansion

$$R_{(m,n)}(x) \ := \ \frac{\sum_{j=0}^{m} p_j x^j}{1 + \sum_{k=1}^{n} q_k x^k}\ , \tag{3.7}$$

for at least one pair of m and n such that $m + n = N$.

Rational functions of order N are a richer set than polynomials of order N.

[4] not necessarily around zero — assumed here without loss of generality

[5] There is no guarantee here but in practice this tends to be the case.

Householder's method [Hou70; Wik19] is designed as an iterative procedure to solve $f(x) = 0$ for x via

$$x_{n+1} \;=\; x_n + \mathrm{HH}_d(x_n) \tag{3.8}$$

with

$$\mathrm{HH}_d(x) \;:=\; d \cdot \frac{g^{(d-1)}(x)}{g^{(d)}(x)} \qquad \text{and} \qquad g(x) \;:=\; \frac{1}{f(x)}\,. \tag{3.9}$$

It has convergence order $d + 1$.

With $\nu := -f(x)/f'(x)$ and $h_k := f^{(k)}(x)/f'(x)$, we have:-

$$\mathrm{HH}_1 = \nu \qquad \text{/* Newton's method */} \tag{3.10}$$

$$\mathrm{HH}_2 = \tfrac{\nu}{1+h_2\nu/2} \qquad \text{/* Halley's method */} \tag{3.11}$$

$$\mathrm{HH}_3 = \tfrac{\nu(1+h_2\nu/2)}{1+\nu(h_2+h_3\nu/6)} \tag{3.12}$$

$$\mathrm{HH}_4 = \tfrac{\nu(1+\nu(h_2+h_3\nu/6))}{1+\nu(3h_2/2+\nu(h_2^2/4+h_3/3+h_4\nu/24))} \tag{3.13}$$

With ν being the increment of the standard Newton method, the HH_d step (for $d > 1$) has the form of an $R_{(d-1),(d-1)}(\nu)$ rational function of ν.

This is by design!

(symmetric rational function approximations of
the form $R_{m,n}(\cdot)$ where $m = n$ typically work best)

The method is immensely powerful but, sadly, rarely used in financial maths.

An often overlooked use of **Householder's method** is that it represents

a rational function approximation for the inverse of any function $f(x)$

in the vicinity of any desired expansion centre x_c:

$$x(y) \;\approx\; x_c + \mathrm{HH}_d(x_c) \tag{3.14}$$

with

$$\nu := \tfrac{y-f(x_c)}{f'(x_c)} \quad \text{and} \quad h_k := \tfrac{f^{(k)}(x_c)}{f'(x_c)}\,. \tag{3.15}$$

➡ *General Caveats*

Polynomial versus rational approximations

In all that follows:

We (at most) ever assume that the approximand $f(x)$ is *continuous*, i.e., $f(x) \in \mathcal{C}^0$, not more.

Polynomial versus rational function approximations

- Much is being advertised that polynomials (can) have *spectral* convergence properties, i.e., like e^{-n}.
- However, *this is at most true for smooth functions!*

A kinky tale

Take the function

$$f(x) := |x| \tag{4.1}$$

as the archetype for any function with a *kink*.

Polynomial approximants for $|x|$ of degree n converge [Lip09; New64]

worse than $1/n$.

NOT EVEN LINEAR!

In contrast, there are rational approximations [Akh29; New64] that

converge better than $3 \cdot \mathrm{e}^{-\sqrt{n}}$.

Similar results exist for other functions where polynomials converge slowly.

There are even efficient rational approximations[6] for

$$\mathbf{sign}(x)$$

which is not even continuous!

Many of our financial functions have a kink in some limit, e.g., when $t = T$.

➡ *Relative vs absolute error*

[6]First shown in *1877* by Егор Иванович Золотарёв (Yegor Ivanovich Zolotarev).

- The literature on function approximation typically starts with a focus on the *absolute error* of the approximation $\tilde{f}(x)$ for the target function $f(x)$

$$|\tilde{f}(x) - f(x)| \,. \tag{4.2}$$

- I advise against ever even bothering with the absolute error.
- Financial mathematics functions are typically implemented in the format of *floating point numbers*. These have a *mantissa* and an *exponent*.
- Accuracy in floating point representation is by convention and for practical reasons expressed in the *number of significant digits in the mantissa*.
- This means we need to use the ***relative accuracy***

$$\left|\frac{\tilde{f}(x)}{f(x)} - 1\right| \tag{4.3}$$

  for any benchmarking purposes.

➡ *Handling Zero*

- Obviously, this means that we must ensure that any approximation $\tilde{f}(x)$ is zero ***exactly*** wherever $f(x)$ is zero.

- Easy: instead of $f(x)$, we approximate

$$g(x) \;:=\; \frac{f(x)}{(x - x_0)^\mu} \tag{4.4}$$

  by some $\tilde{g}(x)$, where μ is the multiplicity of the root x_0, and set

$$\tilde{f}(x) \;:=\; (x - x_0)^\mu \cdot \tilde{g}(x) \,. \tag{4.5}$$

- When multiple roots exist, most likely, the respective interval is subdivided and each root treated separately by its own localized approximation.

➡ *Some formal background*

The alternation theorem of de la Vallée-Poussin [Val10; Enc17]

Given a continuous function $f(x)$ on $x \in [a,b]$, an n-th order polynomial approximant $\tilde{P}_n(x)$, and some successive points $x_0 < x_1 < \cdots < x_{n+1}$ on $[a,b]$ at which the errors $\tilde{\Delta}_i := f(x_i) - \tilde{P}_n(x_i)$ alternate in sign,

$$\tilde{\Delta}_i \cdot \tilde{\Delta}_{i+1} \; < \; 0 \quad \forall\, i = 0, \ldots, n\,, \tag{5.1}$$

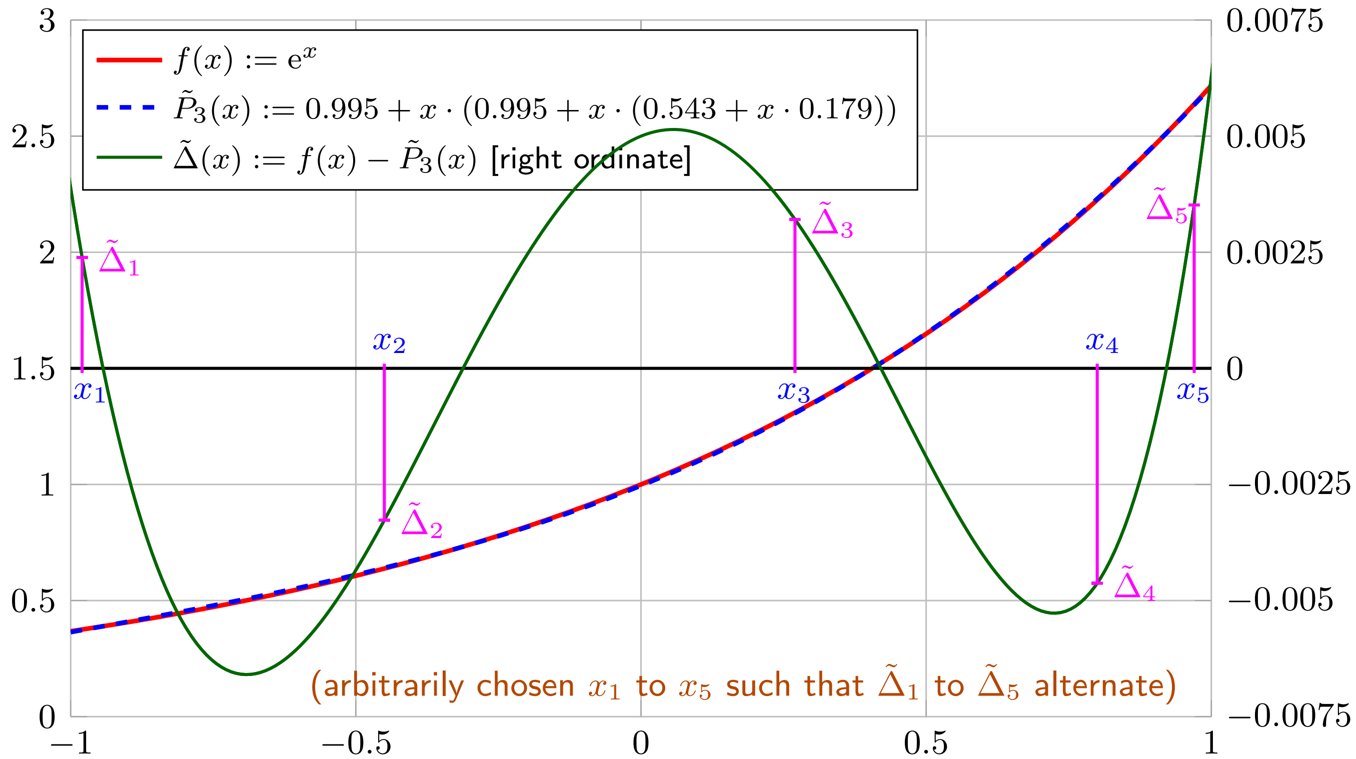
then the maximum absolute error of *any* n-th order polynomial approximant $P_n(x)$ is at least as large as the smallest of the $|\tilde{\Delta}_i|$, i.e.,

$$\inf_{P_n} \left\{ \max_{x\in[a,b]} |f(x) - P_n(x)| \right\} \quad \geq \quad \min\left\{ |\tilde{\Delta}_0|, \ldots, |\tilde{\Delta}_{n+1}| \right\} \,. \tag{5.2}$$

A simple proof for its generalization to rational functions is given in [Lit01].

The Chebyshev (equioscillation) theorem

The equality in (5.2) holds *iff* $\tilde{P}_n(x)$ is the polynomial of best approximation.



For ***all possible*** $P_3(x)$:

$$\boxed{\max(|f(x) - P_3(x)|) \quad \geq \quad \min_{1\leq i\leq 5}\{|\tilde{\Delta}_i|\} \quad !}$$

Chebyshev polynomials

$$T_0(x) = 1 \tag{5.3}$$

$$T_1(x) = x \tag{5.4}$$

$$T_2(x) = 2x^2 - 1 \tag{5.5}$$

$$T_3(x) = 4x^3 - 3x \tag{5.6}$$

$$T_n(x) = \cos(n \arccos(x)) \tag{5.7}$$

Recursion

$$T_n(x) = 2x \cdot T_{n-1}(x) - T_{n-2}(x) \tag{5.8}$$

Product rule (this will be important later)

$$T_m(x)T_n(x) = \tfrac{1}{2}\left[T_{m+n}(x) + T_{|m-n|}(x)\right] \tag{5.9}$$

Orthogonality

$$\langle T_m, T_n \rangle_C = \delta_{nm} \cdot \frac{\pi}{\aleph_m} \tag{5.10}$$

where

$$\aleph_m \equiv (m = 0 \;?\; 1 : 2) \tag{5.11}$$

and

$$\langle f, g \rangle_C \equiv \int_{-1}^{1} \frac{f(x) \cdot g(x)}{\sqrt{1 - x^2}} \,\mathrm{d}x \tag{5.12}$$

Chebyshev series

If $f(x)$ has finite Chebyshev norm $||f||_C := \sqrt{\langle f, f \rangle_C}$, then,

$$f(x) = \sum_{i=0}^{\infty} \hat{c}_i T_i(x) \tag{5.13}$$

on $x \in [-1, 1]$ with

$$\hat{c}_i = \frac{\aleph_i}{\pi} \cdot \langle f, T_i \rangle_C \;. \tag{5.14}$$

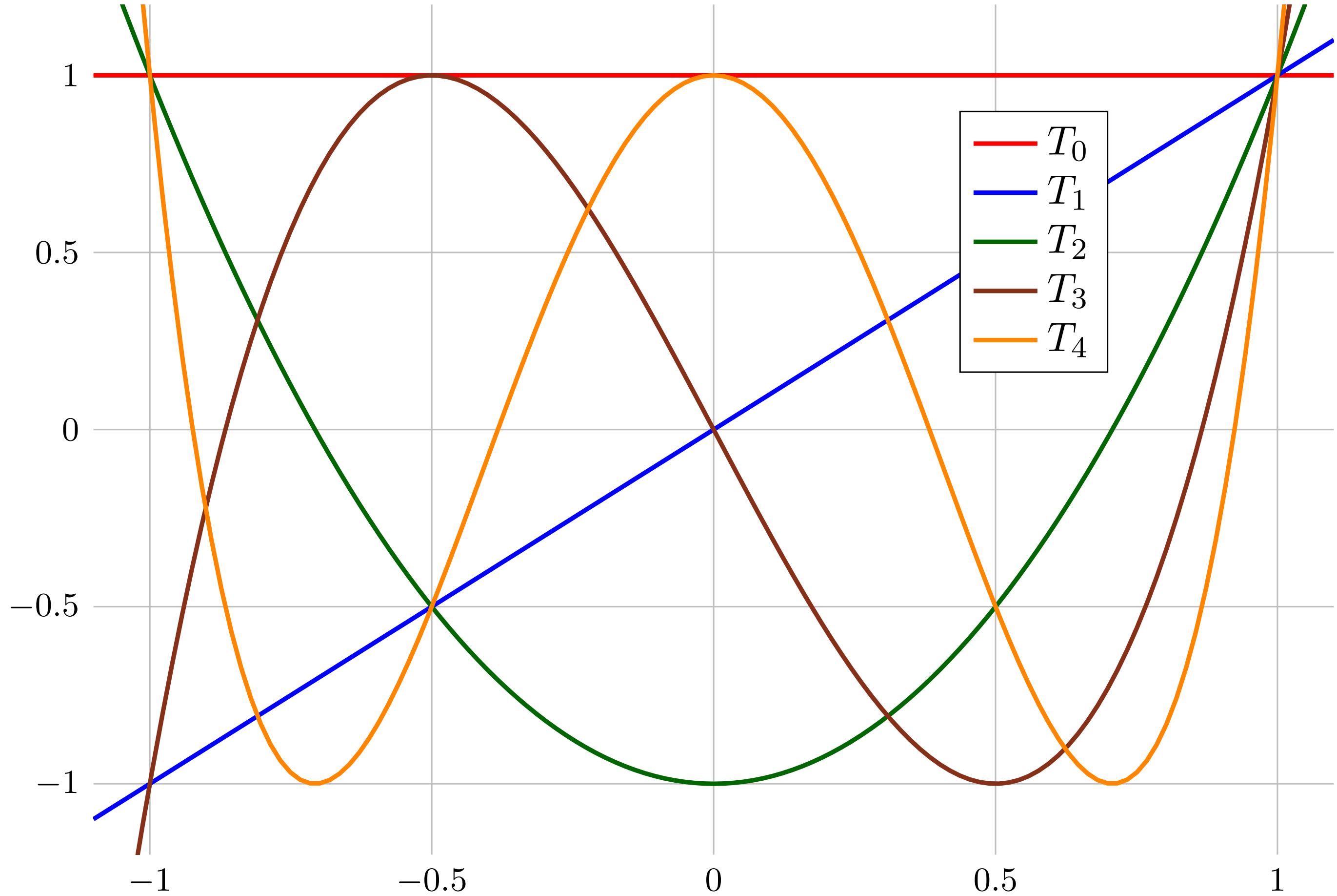
Discrete orthogonality

On the roots of $T_N(x)$, also called *"Chebyshev nodes"*, given by

$$x_k = \cos\left(\frac{(k-\frac{1}{2})\cdot\pi}{N}\right) \quad \forall \;\; k = 1, 2, ..., N \;, \tag{5.15}$$

we have for $i, j < N$, with $\aleph_i \equiv (i = 0 \;?\; 1 : 2)$,

$$\sum_{k=1}^{N} T_i(x_k) \cdot T_j(x_k) = \delta_{ij} \cdot \frac{N}{\aleph_i} \;. \tag{5.16}$$



Cheney's theorem

Let $f(x)$ be a function integrable on [-1, 1]. If

$$\langle f, T_i \rangle_C \;=\; 0 \quad \forall\; i = 0, \ldots, N\;, \tag{5.17}$$

then $f(x)$ either changes its sign in [-1, 1] at least $N+1$ times or vanishes almost everywhere.

This means that any function of the form

$$\sum_{i=n+1}^{\infty} c_i T_i(x) \tag{5.18}$$

features the alternation (5.1) condition in the de la Vallée-Poussin theorem.

Alternation of the truncated Chebyshev series

Given the Chebyshev expansion of a function $f(x)$

$$f(x) = \textstyle\sum_{i=0}^{\infty} \hat{c}_i T_i(x)\;, \tag{5.19}$$

and its n-th order truncated Chebyshev approximant

$$P_n(x) := \textstyle\sum_{i=0}^{n} \hat{c}_i T_i(x)\;, \tag{5.20}$$

the resulting error function

$$\Delta_n(x) := f(x) - P_n(x) \tag{5.21}$$

satisfies the Cheney condition (5.17). It follows that either $\Delta_n(x)$ is identically zero or has $n+1$ sign changes. This means that alternation is present in the sense of de la Vallée-Poussin, i.e.,

"*the approximant $P_n(x)$ is close to the best one*" [Lit01].

➡ *More sophisticated methods*

- Recall the consequence of Cheney's theorem:

  a truncated Chebyshev series is close to an optimal polynomial.

- This led to the idea to convert any

  truncated Taylor series

  into an

  even more truncated Chebyshev series.

- This is called the

  "*Economization of power series*".

- Develop some power series for $f(x)$ to some order n:

$$f(x) \approx \sum_{i=0}^{n} a_i x^i \ . \tag{6.1}$$

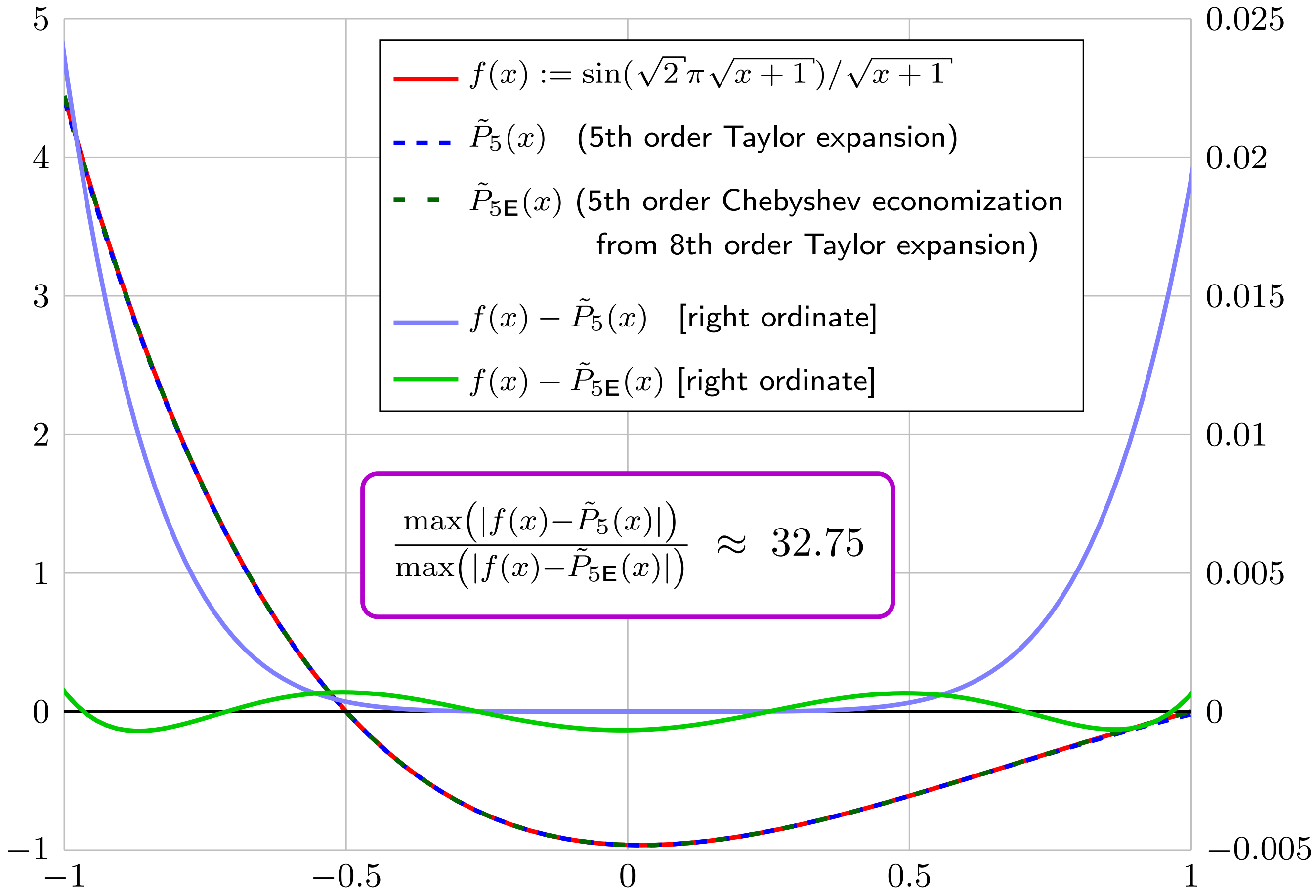
- Convert it into the equivalent Chebyshev series

$$\sum_{i=0}^{n} c_i T_i(x) \ = \ \sum_{i=0}^{n} a_i x^i \ . \tag{6.2}$$

- Truncate the Chebyshev series at some order $m < n$.
- Re-express as a polynomial, e.g., in Horner form for efficient evaluation:

$$\sum_{i=0}^{m} c_i T_i(x) \ = \ c_0 + x \cdot (c_1 + x \cdot (c_2 + \cdots (c_{m-1} + x \cdot c_m) \cdots)) \tag{6.3}$$

- *Miraculously*, the Chebyshev expansion to order m is *more accurate* than the power series to order m.



- Pretty, huh?
- In practice, however, *I have yet to find an actual use case of*

  power series economization.
- This is not because the method has no merit.
- It is because *we can easily do much better!*

Fitting/approximating/quadratures

By virtue of the discrete orthogonality (5.16), the Chebyshev approximation

$$\tilde{f}_N(x) \; = \; \sum_{i=0}^{N-1} c_i T_i(x) \tag{6.4}$$

satisfies $\tilde{f}_N(x_k) = f(x_k)$ on the roots of $T_N(x)$, i.e., on

$$x_k = \cos\left(\tfrac{(k-\frac{1}{2})\cdot\pi}{N}\right) \quad \forall \;\; k = 1, 2, ..., N \; , \tag{6.5}$$

where, with $\aleph_i \equiv (i = 0 \; ? \; 1 : 2)$,

$$\begin{aligned} c_i \; &= \; \frac{\aleph_i}{N} \sum_{k=1}^{N} f(x_k) \cdot T_i(x_k) \qquad (6.6) \\ &= \; \frac{\aleph_i}{N} \sum_{k=1}^{N} f(x_k) \cdot \cos\left(\tfrac{i\cdot(k-\frac{1}{2})\cdot\pi}{N}\right) \; . \end{aligned}$$

Note that the N coefficients c_i of the N-point Chebyshev fit given by (6.6)

are not the same as the $\hat{c}_i$ of the true Chebyshev expansion (5.14).

For practical purposes, however, the difference is negligible, meaning

$$c_i \; \approx \; \hat{c}_i \; . \tag{6.7}$$

What's more, the fact that the fit is exact on the N Chebyshev nodes guarantees that the error of the fit alternates $N-1$ times (unless the fit is already exact in which case the error is zero everywhere).

The discrete Chebyshev fit is easy to compute and works extremely well.

This is why we never really need to use the *economization* technique.

Given a (possibly approximate) Chebyshev expansion of a function $f(x)$ on [-1,1] to whichever order we may require,

$$f(x) = \sum_{l=0}^{\infty} c_l T_l(x) \tag{6.8}$$

in analogy to the construction of a Padé expansion from a Taylor expansion, we can also form rational function approximation based on the Chebyshev basis set:

$$R_{(m,n)}(x) = \frac{P_m(x)}{Q_n(x)} \tag{6.9}$$

$$P_m(x) = \sum_{j=0}^{m} p_j T_j(x) \tag{6.10}$$

$$Q_n(x) = 1 + \sum_{k=1}^{n} q_k T_k(x) \tag{6.11}$$

A true Chebyshev-Padé[7] expansion $R_{(m,n)}(x)$ of $f(x)$ must have the same Chebyshev series coefficients as $f(x)$ to order $N = m + n$, i.e.,

$$\left\langle \left(f - \tfrac{P_m}{Q_n} \right), T_i \right\rangle_C = 0 \qquad \forall \;\; i = 0, \ldots, N = m + n \;. \tag{6.12}$$

A solution $R_{(m,n)}(x) = \frac{P_m(x)}{Q_n(x)}$ to (6.12) is also called a

Nonlinear Chebyshev-Padé approximation[8].

It is guaranteed to satisfy the Cheney condition (5.17), and thus to alternate.

> *Thus, there are reasons to assume that nonlinear Chebyshev-Padé approximants are close to the best ones in the sense of the absolute error* [Lit01].

One small snag is ...

> *that the nonlinear Chebyshev-Padé approximant does not always exist...*

[7] Some authors call it "Chebyshev-Padé" and some call it "Padé-Chebyshev".
[8] Also known as *Clenshaw-Lord approximant* on behalf of their algorithm in [CL74].

In contrast, the linearized version of the orthogonality condition given by

$$\langle (f \cdot Q_n - P_m), T_i \rangle_C = 0 \qquad \forall \ i = 0, \ldots, N = m + n \tag{6.13}$$

always admits a solution for P_m and Q_n.

What's more,

> *nonlinear Padé-Chebyshev approximants (in comparison with the linear ones) have, as a rule, a somewhat smaller absolute errors* [sic], *but* ***can have larger relative errors*** [Lit01]

and we really want a minimal *relative* error!

A solution to (6.13) is also called a *Linear Chebyshev-Padé approximation*[9].

Since, in practice, I never seek a true analytical Chebyshev-Padé expansion anyway, I have always found the linear approximant perfectly adequate to work with.

[9] Also known as *Maehly approximant* for its first publication in [Mae60], and referred to as *cross-multiplied linear Padé–Chebyshev approximation* in [Lit01] for obvious reasons.

The coefficients of P_m and Q_m are thus to be computed from

$$\langle G \,,\, T_i \rangle_C = 0 \tag{6.14}$$

with

$$G := \sum_{j=0}^{m} p_j T_j - \Big(1 + \sum_{k=1}^{n} q_k T_k\Big) \cdot \sum_{l=0}^{\infty} c_l T_l \tag{6.15}$$

for all $i = 0, \ldots, N = m + n$. Expanding G, we obtain

$$G = \sum_{j=0}^{m} p_j T_j - \sum_{k=1}^{n} q_k T_k \cdot \Big(c_0 + \sum_{l=1}^{\infty} c_l T_l\Big) - \sum_{l=0}^{\infty} c_l T_l \tag{6.16}$$

$$= \sum_{j=0}^{m} p_j T_j - \sum_{k=1}^{n} q_k c_0 T_k - \sum_{k=1}^{n} \sum_{l=1}^{\infty} q_k c_l T_k T_l - \sum_{l=0}^{\infty} c_l T_l \tag{6.17}$$

$$= \sum_{j=0}^{m} p_j T_j - \sum_{k=1}^{n} q_k c_0 T_k - \tfrac{1}{2} \sum_{k,l=1}^{n,\infty} q_k c_l \left[T_{k+l} + T_{|k-l|} \right] - \sum_{l=0}^{\infty} c_l T_l \tag{6.18}$$

For $i \geq 0$ and $k > 0$, we have the generic equality

$$\sum_{l=1}^{\infty} c_l \cdot \left(\delta_{i\,(k+l)} + \delta_{i\,|k-l|}\right) \;=\; c_{k+i} + c_{|k-i|} \cdot (1 - \delta_{ik}) \cdot \mathbf{1}_{\{i>0\}}\,, \qquad (6.19)$$

and with this we obtain from (6.14) and (6.18) the linear system

$$\boxed{p_i \cdot \mathbf{1}_{\{i\leq m\}} - \sum_{k=1}^{n} \frac{q_k}{2} \cdot \Big(c_{k+i} + c_{|k-i|} \cdot (1 + \delta_{ik}) \cdot \mathbf{1}_{\{i>0\}}\Big) \;=\; c_i} \qquad (6.20)$$

for $i = 0, \ldots, N = m + n$.

Note that the occurrence of the coefficient c_{k+i} means[10] that *we must have the coefficients c_l* of $f(x) = \sum_{l=0}^{\infty} c_l T_l(x)$ *up to order $m + 2n$*.

[10] Whilst this is known in the literature [Lit01, page 26], it is still sometimes missed.

- We have seen how easy it is to convert a Chebyshev expansion (to order $m + 2n$) into a Chebyshev-Padé approximant.
- In practice, we can simply use a discrete Chebyshev fit instead of a precise analytical Chebyshev expansion.
- The real good news is that (virtually) all our analytical understanding for polynomial approximations (de la Vallée-Poussin, Cheney, the Chebyshev equioscillation theorem, etc.), subject to certain conditions,

also holds for rational approximations!

➡ *the Remez algorithm*

Based on the above, *in 1934*, **Евгений Яковлевич Ремез**[11], published an algorithm for the computation of the best possible (polynomial) function approximation [Rem34].

A good reference introduction comes with the Boost C++ library [Boo].

In a nutshell (here for polynomials, but it holds for rational functions):-

- Select a finite interval $[a, b]$ on which a function $f(x)$ is to be approximated. We assume this interval to be $[-1, 1]$ (which can always be achieved by virtue of an affine transformation).

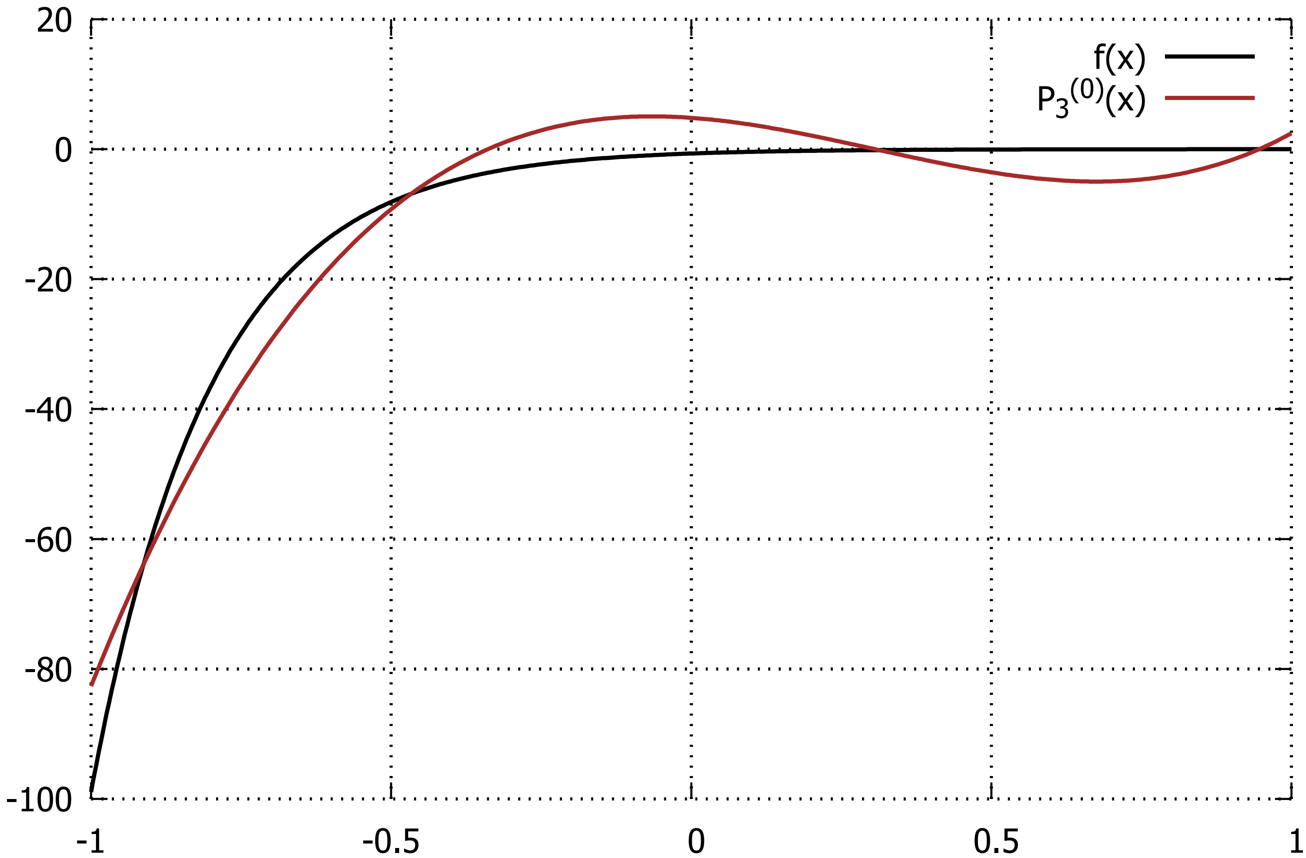
- Find an initial approximant $P_N^{(0)}(x)$ of order N and a sequence of $N+2$ *Chebyshev reference points*

$$x_i^{(0)} \in [-1, 1], \quad i = 0, \ldots, N$$

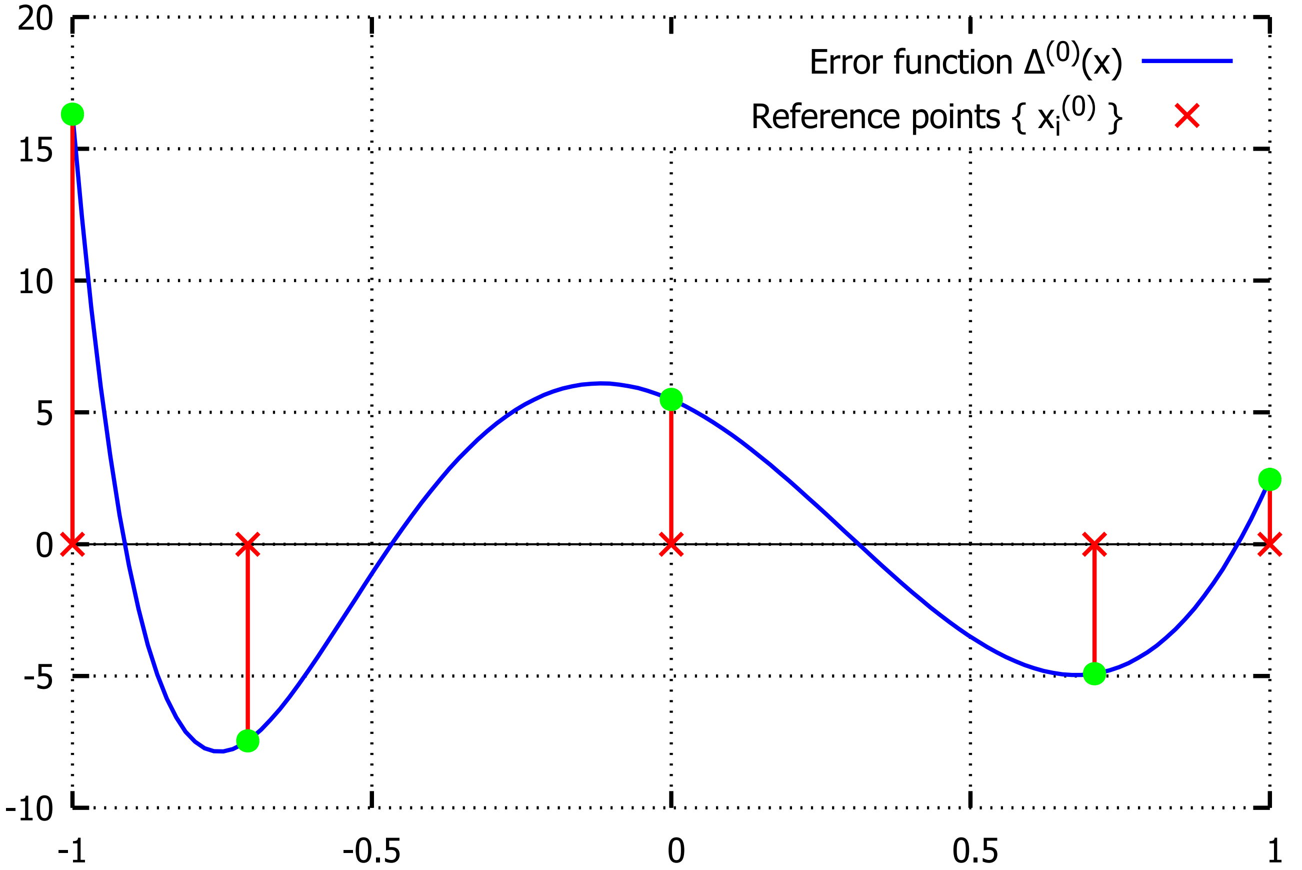
such that the error $\Delta^{(0)}(x) := P_N^{(0)}(x) - f(x)$ ***alternates*** over the x_i.

[11] Evgeny Yakovlevich Remez

Example: target function and initial polynomial $P_3^{(0)}(x)$.



Error function and reference points:



The *minimax polynomial*[12] has the property that it

alternates with equal amplitude at its extrema in the interval[13].

This, directly, is a nonlinear objective of significant complexity.

The Remez algorithm breaks this into an

iteration between two easier tasks.

[12] another name for the equioscillatory best fit

[13] Note that the "extrema" locations (usually) include the end points of the interval.

At iteration k:-

1. Given a set of *fixed* reference points $\{x_i^{(k-1)}\}$, find the coefficients of the polynomial $P_N^{(k)}(x)$ such that the error function $\Delta^{(k)}(x) := P_N^{(k)}(x) - f(x)$

   alternates with ***equal amplitude at the fixed reference points***.

   This means, we have to solve the linear system of $N+2$ equations

$$\boxed{\sum_{j=0}^{N} p_j^{(k)} T_j(x_i^{(k-1)}) \;-\; f(x_i^{(k-1)}) \;=\; (-1)^i E^{(k)}} \tag{7.1}$$

   for the $N+1$ coefficients $\{p_j^{(k)}\}$ and the amplitude $E^{(k)}$.

2. Having established the new error function

$$\Delta^{(k)}(x) \;:=\; \sum_{j=0}^{N} p_j^{(k)} T_j(x) \;-\; f(x)\;, \tag{7.2}$$

   move the reference points:-

   - in the Remez-I algorithm, just one point is moved[14];
   - in the Remez-II algorithm, *all are moved to the extrema of* $\Delta^{(k)}(x)$;

   to become the new set of reference points $\{x_i^{(k)}\}$.

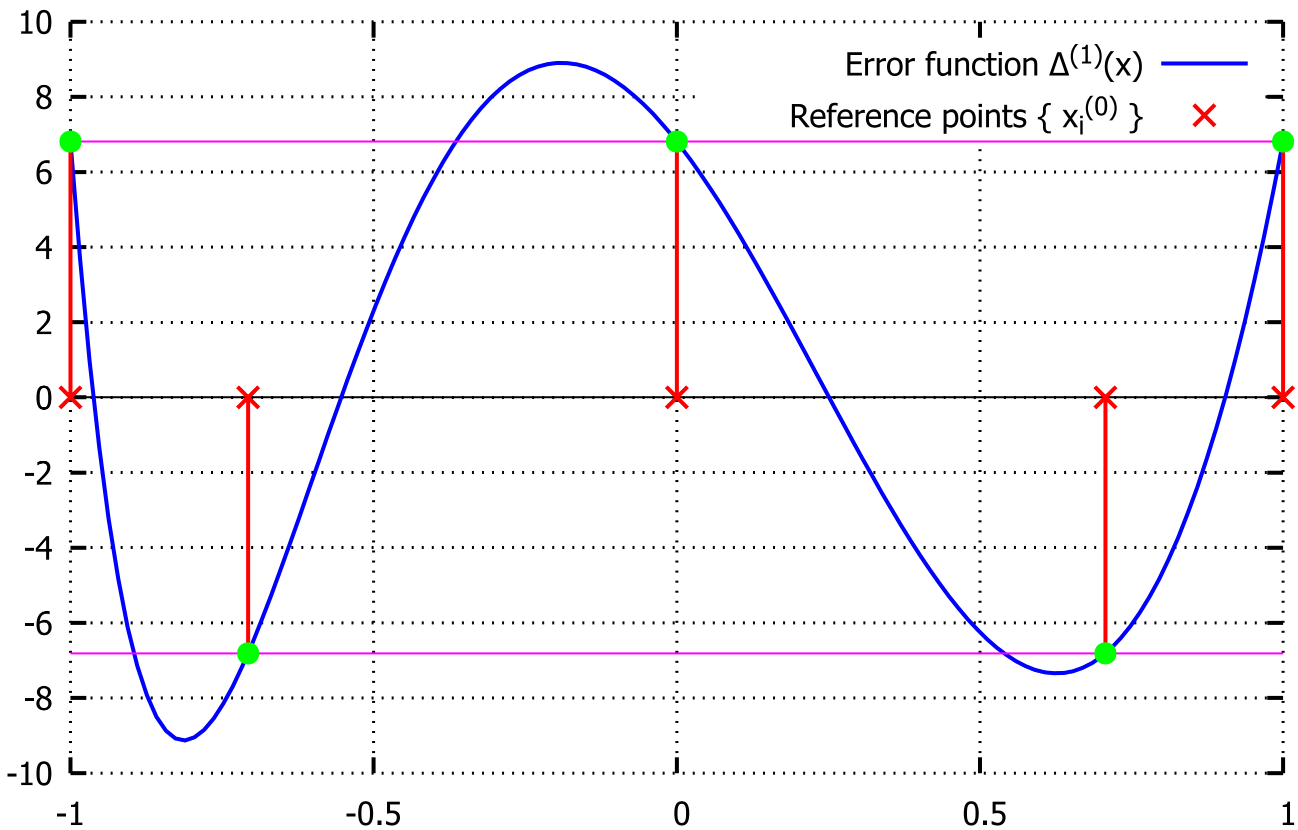
[14] in practice, it seems that Remez-I is no longer in use

Caveats:-

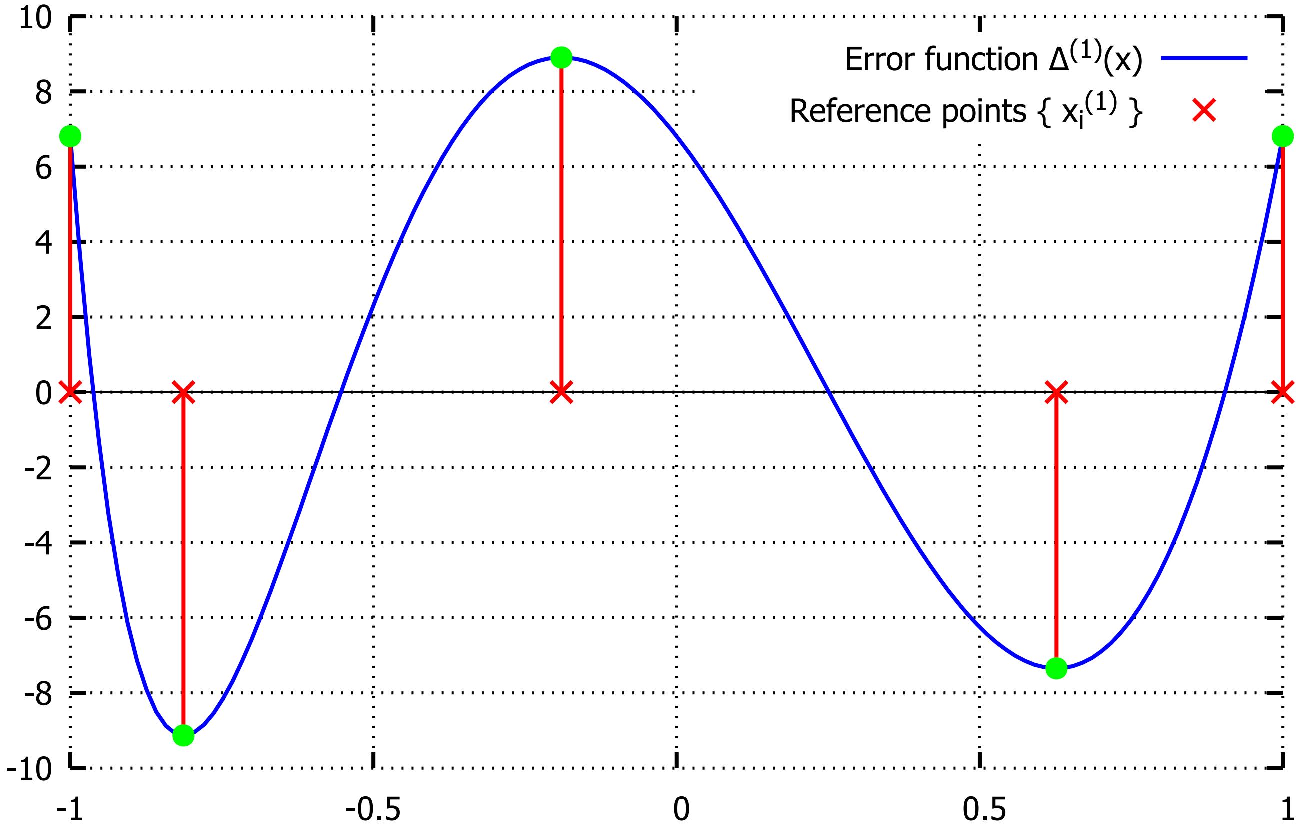
- Most of the complexity lies in step 2 of each Remez iteration, and the following various required checks.
- In the more general *rational approximation* case, the solution found in step 1 may have poles inside the interval. *That's bad.*[15]
- In the more general *transformed* case, the solution found in step 1 may have more or fewer error extrema than that of the previous iteration.
- The error function may also have zero slope points which are not extrema.
- Much else can go wrong. It is a ***black art***.
  See the Boost C++ library's [Boo] "*checklist*" on the Remez method.
- It is not for the fainthearted, but perseverance pays off.

[15] This is often caused by insufficient precision, else usually can only be helped by increasing N.

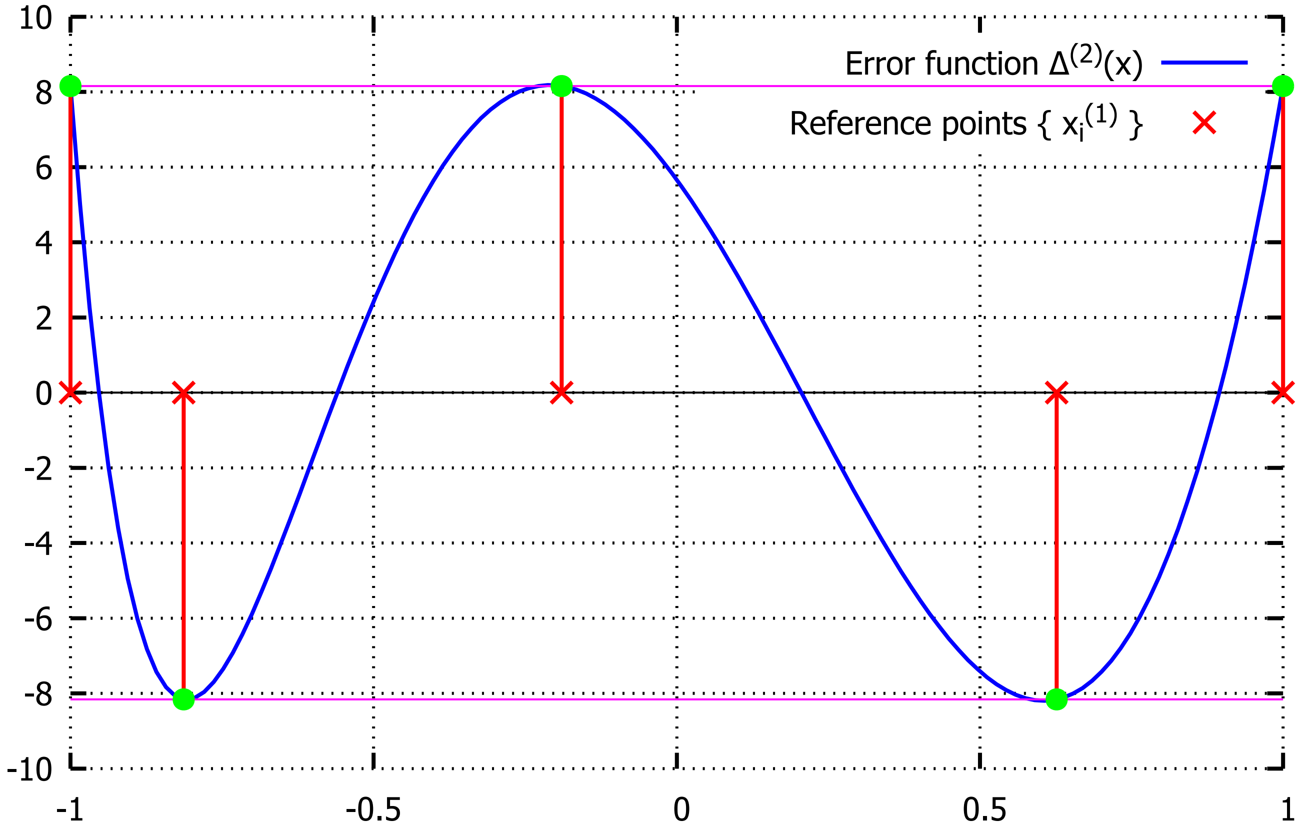
Back to our example. Iteration 1, step 1, after solving (7.1):



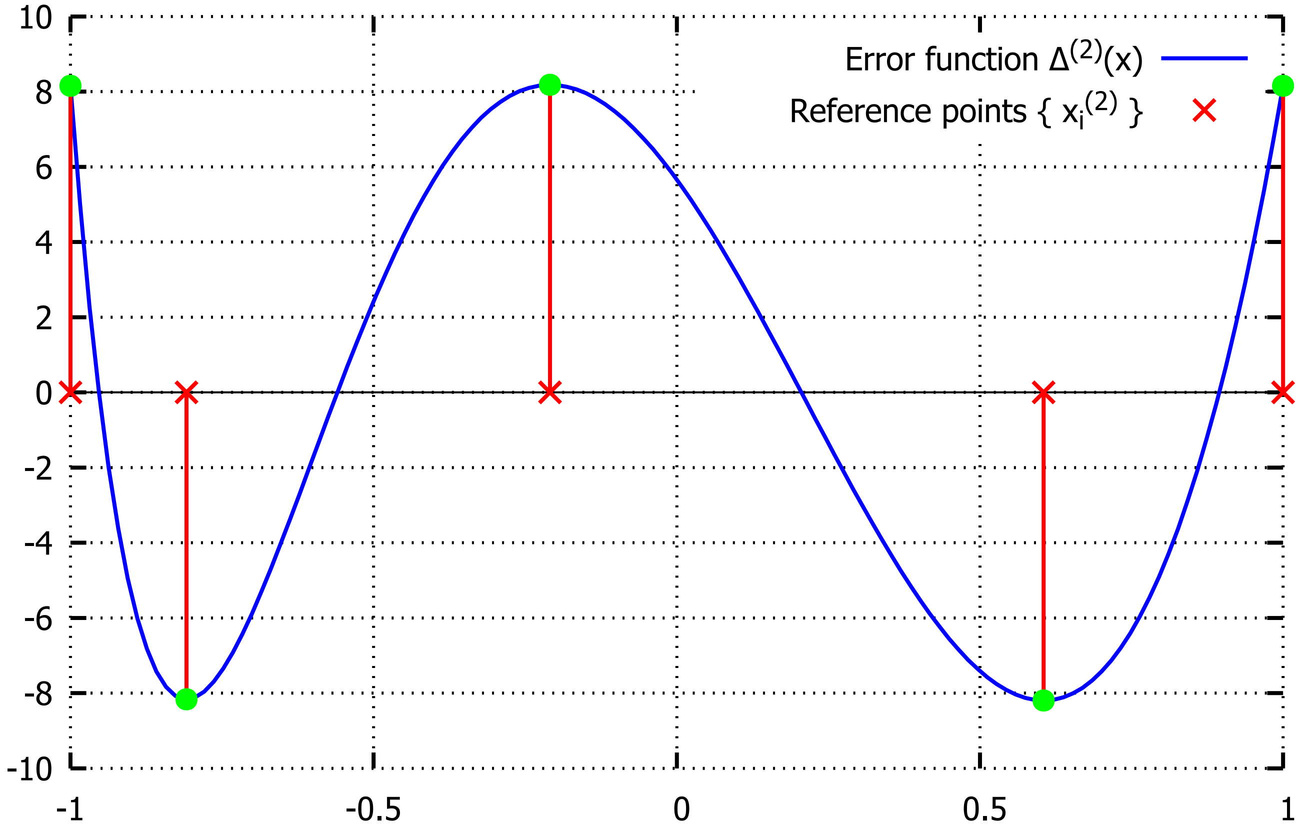
Iteration 1, step 2, upon moving the reference points:



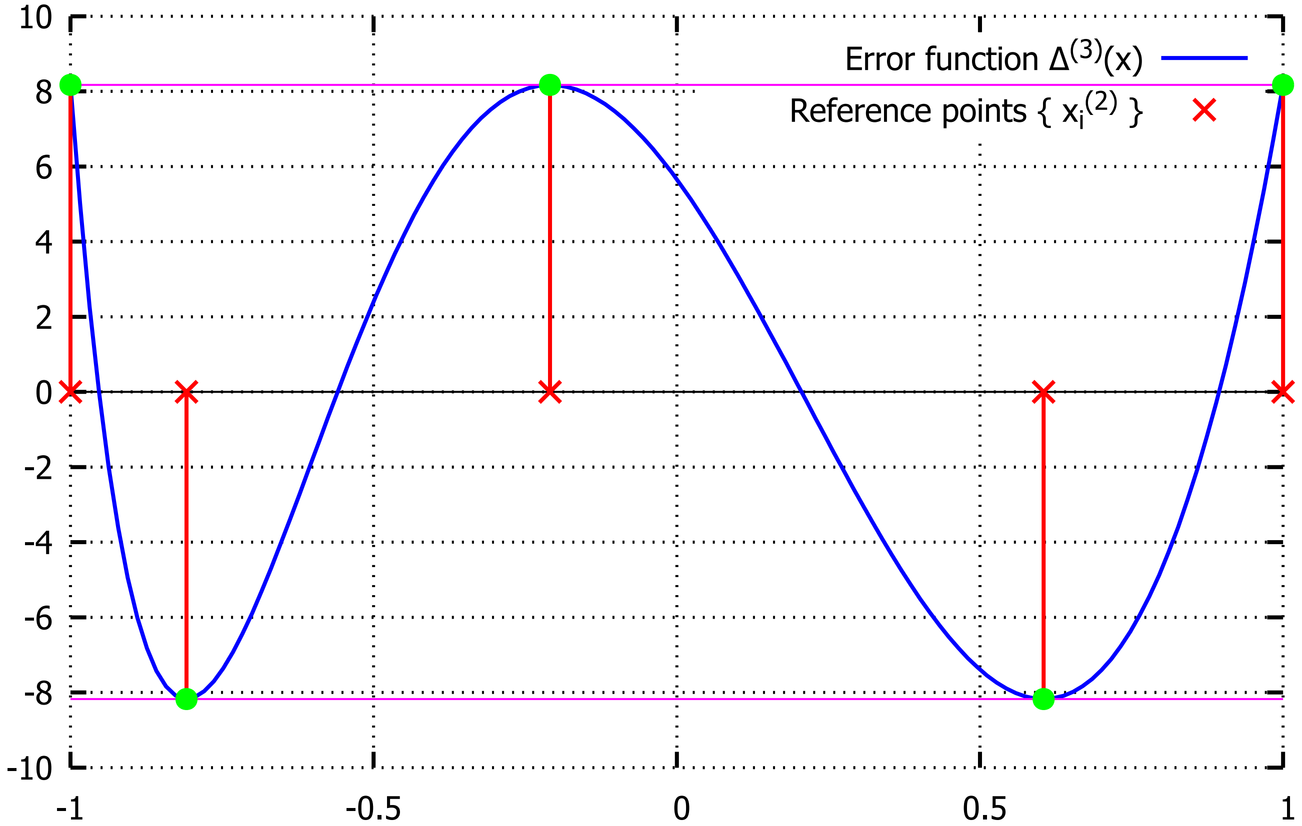
Iteration 2, step 1, after solving (7.1):



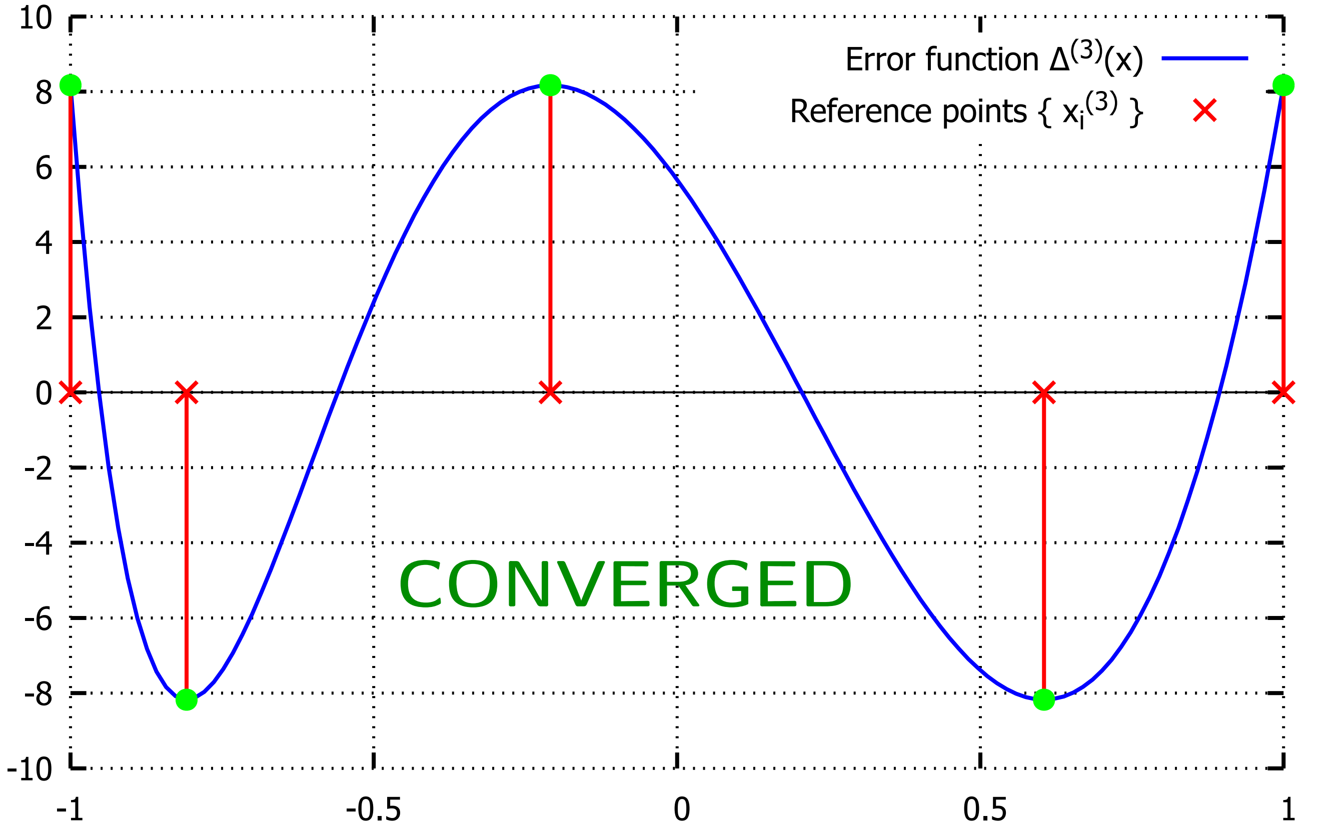
Iteration 2, step 2, upon moving the reference points:



Iteration 3, step 1, after solving (7.1):



Iteration 3, step 2, upon moving the reference points:



➡ *Rational Remez*

For a *rational function* approximation $R_{(m,n)}(x)$, the equi-alternation equation (7.1) of the Remez algorithm step 1 becomes

$$\frac{\sum_{j=0}^{m} p_j^{(k)} T_j(x_i^{(k-1)})}{1+\sum_{l=0}^{n} q_l^{(k)} T_l(x_i^{(k-1)})} \;-\; f(x_i^{(k-1)}) \;=\; (-1)^i E^{(k)} \tag{7.3}$$

which, assuming no poles of the approximant on [-1,1], turns into

$$\sum_{j=0}^{m} p_j^{(k)} T_j(x_i^{(k-1)}) \;=\; \Big(1+\sum_{l=0}^{n} q_l^{(k)} T_l(x_i^{(k-1)})\Big)\cdot\Big(f(x_i^{(k-1)})+(-1)^i E^{(k)}\Big)\,. \tag{7.4}$$

Note that this equation is ***no longer linear*** in $\{p_j^{(k)}, q_l^{(k)}, E^{(k)}\}$.

It is, however, only mildly so (bilinear, to be precise).

Alas, *"the method usually adopted to solve these equations is an iterative one"* [Boo], i.e., to iterate over ℓ, each time solving the linear system [Hen63]

$$\sum_{j=0}^{m} \tau_{ij} \cdot p_j^{(k)(\ell)} - \omega_i^{(\ell-1)} \cdot \sum_{j=0}^{n} \tau_{ij} \cdot q_j^{(k)(\ell)} + (-1)^i \cdot E^{(k)(\ell)} = f_i \qquad (7.5)$$

with

$$\tau_{ij} := T_j(x_i^{(k-1)}) , \quad f_i := f(x_i^{(k-1)}) \quad \omega_i^{(\ell-1)} := f_i + (-1)^i \cdot E^{(k)(\ell-1)} , \qquad (7.6)$$

and $E^{(k)(0)}$ initialized to the previous average error.

I would recommend against this.

As discussed by Henderson in 1963 [Hen63], this simple *fixed point iteration* method may not even be contractive. It depends on the target function whether it converges towards your solution or diverges away from it. It may work in practice. However, ...

The bilinear form of the full system (7.4) makes it particularly suitable for the standard (Newton-)Raphson method.

Instead, simply use (Newton-)Raphson to solve (7.4).

➡ Nonlinear Remez

Nonlinear Remez

It is not uncommon for the approximand function to vary *significantly* in magnitude on the approximation interval.

As a result, homogenization of the absolute error can result in a significant loss of precision in the *relative* error.

To compensate for this, [Rem34] added a weight function $q(x)$:

> **2. Более простой вариант вычислительного оформления основного алгорифма в случае взвешенного полиномиального приближения.** В этом случае чебышевская задача типа (255) представляется в несколько трансформированном виде
>
> $$\sup\,[q(x) \cdot |f(x) - P_n(x;\,K)|] = L = L(K_0,\,K_1,\,\ldots,\,K_n) = \min!, \qquad (264)$$

When $q(x) \equiv \frac{1}{f(x)}$, this means we are aiming to minimize the relative error of the approximant versus the target function.

In practice, we often use auxiliary functions[16] to carry part of the burden of the approximation, e.g.,

- the exponential function e^y,
- or $\sqrt{y}$,
- or simply $x \cdot y$ (when we had to remove a root of the target function).

➡ In general, I allow for a transformation from the original target function $f(x)$ to a *reduced problem function $g(x)$* via a suitably chosen, possibly nonlinear, transformation $\theta(x, y)$ such that

$$f(x) \;\equiv\; \theta(x,y)\big|_{y=g(x)} \tag{8.1}$$

and then seek an approximant $R_{(m,n)}$ for $g(x)$.

The purpose of the transformation is to render the reduced approximand $g(x)$ as close to linear as possible.

➡ This is where ***analytics*** come in, e.g., asymptotic expansions, etc.

[16] usually somehow *built-in*, i.e., highly efficiently implemented

- In order to minimize the relative error of the original target function, we seek the approximant $R_{(m,n)}(x)$ that makes

$$\Delta(x) \;:=\; \frac{\theta(x,y)\big|_{y=R_{(m,n)}(x)}}{\theta(x,y)\big|_{y=g(x)}} - 1 \tag{8.2}$$

equi-oscillatory in its extrema on $[-1, 1]$.

- *Most of the time, $m = n$ gives the best approximation.*
- However, for a specific (relative) target accuracy, on occasions, I have also used a numerator polynomial of higher degree than the denominator.
- This yields the equi-alternation rule of the Remez algorithm step 1 as

$$\boxed{\theta\big(x_i, \tfrac{P_m(x_i)}{Q_n(x_i)}\big) \;-\; \theta\big(x_i, g(x_i)\big) \cdot \big(1 + (-1)^i \cdot E\big) \;=\; 0\,.} \tag{8.3}$$

- **We can still use (Newton-)Raphson to solve this!**
- Note that the nonlinearity of $\theta(x, y)$ can cause $\Delta(x)$ not to have the expected number of extrema, or move the outmost extrema inwards from the boundaries, etc., etc.

Nonlinear Remez: Step 1 (of the k-th iteration)

Dropping all mentioning of outer Remez algorithm iteration count (i.e., k), define the full solution coefficient vector $\boldsymbol{c}$ as

$$\boldsymbol{c} \;:=\; (\boldsymbol{p}^\top, \boldsymbol{q}^\top, E)^\top \tag{8.4}$$

with

$$\boldsymbol{p} \in \mathbb{R}^{m+1}\,, \quad \boldsymbol{q} \in \mathbb{R}^{n}\,, \quad \text{and thus} \quad \boldsymbol{c} \in \mathbb{R}^{m+n+2}\,,$$

and the i-th element of the objective function $\boldsymbol{\gamma}(\boldsymbol{c})$ of the equi-alternation condition as the left hand side of (8.3), i.e., as

$$\boxed{\gamma_i(\boldsymbol{c}) \quad := \quad \theta(x_i, \tfrac{P_i}{Q_i}) \;-\; \theta(x_i, g_i)\cdot\big(1+(-1)^i\cdot E\big)} \tag{8.5}$$

where

$$P_i := \sum_{j=0}^{m} \tau_{ij}\cdot p_j\,, \quad Q_i := 1 + \sum_{j=1}^{n} \tau_{ij}\cdot q_j\,, \quad g_i := g(x_i)\,, \quad \tau_{ij} := T_j(x_i)\,.$$

The (Newton-)Raphson solution of the nonlinear system (8.5) is to iterate over ℓ, solving the linear system

$$J(\boldsymbol{c}^{(\ell)})\cdot\big(\boldsymbol{c}^{(\ell+1)} - \boldsymbol{c}^{(\ell)}\big) \;=\; \gamma(\boldsymbol{c}^{(\ell)}) \tag{8.6}$$

for $\boldsymbol{c}^{(\ell+1)}$ in each iteration with the Jacobian elements

$$J_{il}(\boldsymbol{c}) \;\equiv\; \frac{\partial\gamma_i(\boldsymbol{c})}{c_l} \tag{8.7}$$

given by

$$J_{il} \;=\; \begin{cases} \theta_y(x_i, \frac{P_i}{Q_i})\cdot\frac{\tau_{il}}{Q_i} & \text{for} \quad l = 0,\ldots,m \\ -\theta_y(x_i, \frac{P_i}{Q_i})\cdot\frac{P_i}{Q_i}\cdot\frac{\tau_{i(l-m)}}{Q_i} & \text{for} \quad l = m+1,\ldots,m+n \\ -\theta_y(x_i, g_i)\cdot\frac{P_i}{Q_i}\cdot(-1)^i & \text{for} \quad l = m+n+1 \end{cases} \tag{8.8}$$

with $\theta_x(\xi,\eta) \equiv \frac{\partial\theta(x,y)}{\partial x}\Big|_{\substack{x\,=\,\xi\\ y\,=\,\eta}}$ and $\theta_y(\xi,\eta) \equiv \frac{\partial\theta(x,y)}{\partial y}\Big|_{\substack{x\,=\,\xi\\ y\,=\,\eta}}$.

This isn't too bad!

Step 2

Everyone seems to have their own preferred way to move the reference points.

Here's mine[17].

- In order to find the extrema, I compute the derivative

$$\Delta'(x) \equiv \frac{\partial \Delta(x)}{\partial x} \tag{8.9}$$

of (8.2) analytically, retaining only the numerator

$$\begin{aligned} &\Big[\theta_x(x, \tfrac{P}{Q}) \cdot Q^2 + \theta_y(x, \tfrac{P}{Q}) \cdot \big(P' \cdot Q - P \cdot Q'\big)\Big] \cdot \theta(x, g) \\ &\qquad - \theta(x, \tfrac{P}{Q}) \cdot Q^2 \cdot \big[\theta_x(x, g) + \theta_y(x, g) \cdot g'\big] \ , \end{aligned} \tag{8.10}$$

[17] I tend to use (wx)Maxima (with the 64-bit Steel Bank Common Lisp runtime) since it is free, if somewhat more cumbersome than commercial alternatives, using its `bigfloat` data type with a precision of no less than 50 decimal digits.

- sub-sample [typically 4 times] between the reference nodes $\{x_i\}$,
- evaluate $\Delta'(x)$ at all [typically $4 \cdot (m+n) + 5$] sampling points,
- identify all brackets of roots of $\Delta'(x)$,
- and invoke a robust[18] root finder on $\Delta'(x)$ in each identified bracket.
- Then, move the reference points to the found roots.

➡ *Remember the Caveats mentioned on page 43!*

[18] Having experimented with a range of the root finders built into (wx)Maxima, I have converged to using a simple secant solver hand-written directly in Maxima itself.

➜ *Examples*

- Say, we wish to approximate

$$\sqrt{1+x}-1$$

for small x to avoid the otherwise inevitable catastrophic loss of precision near zero as was required in Genz's algorithm for the bivariate cumulative normal function (see my presentation on composite options [Jäc18; Jäc19]).

- We choose the applicable interval to be $[-\frac{1}{8},\frac{1}{8}]$: we set $x = x'/8$, drop the primes, and define

$$f(x) \;:=\; \sqrt{1+\tfrac{x}{8}}-1 \tag{9.1}$$

for x now to be in the standard Chebyshev interval [-1,1].

Based on the Taylor expansion

$$f(x) = \frac{x}{16} - \frac{x^2}{512} + \mathcal{O}(x^3)$$

we choose

$$\theta(x,g) \;:=\; x\cdot\left(\tfrac{1}{16} - x\cdot g\right) \tag{9.2}$$

which removes the root at zero.

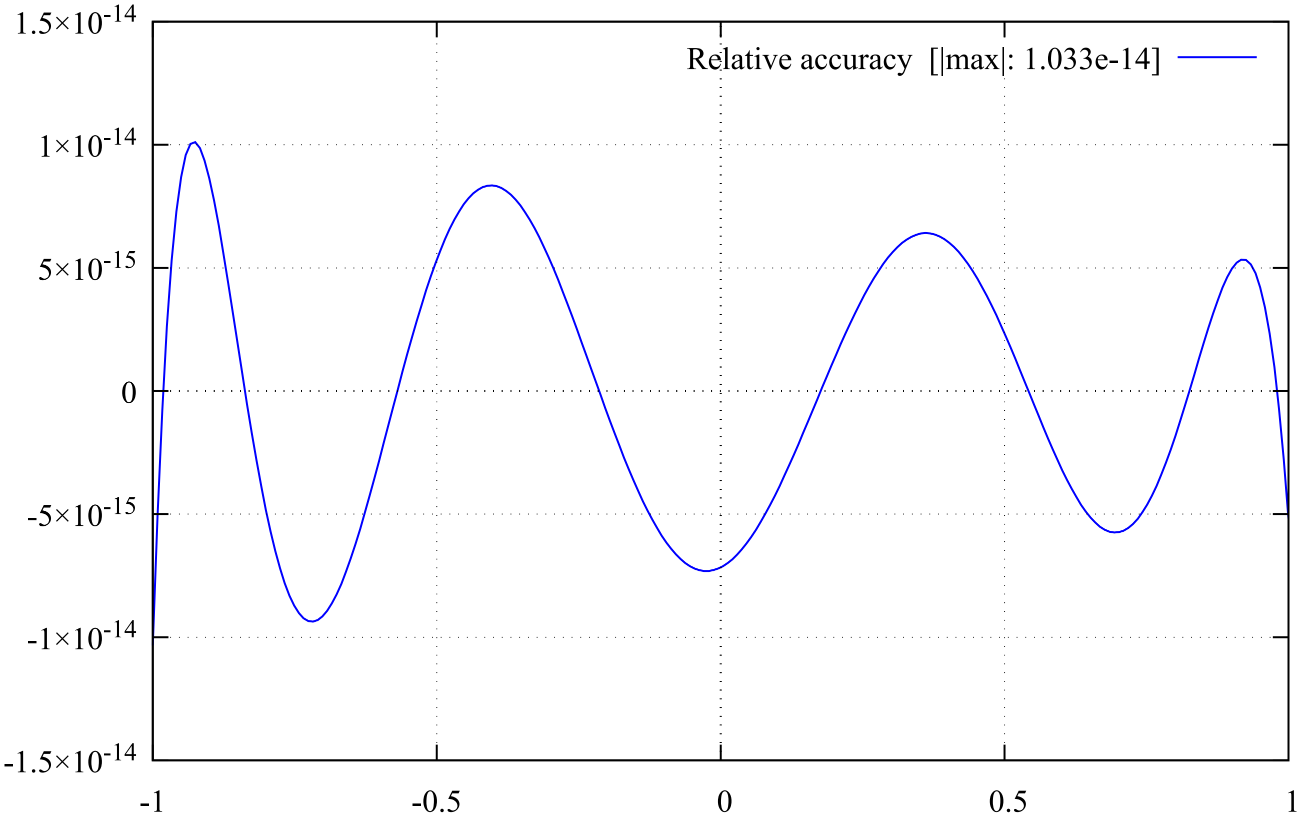
This of course means that, due to $f(x) \equiv \theta(x, g(x))$, we have implicitly defined

$$g(x) \;=\; \frac{1}{x}\cdot\left({}^1\!/_{16} - \left(\sqrt{1+{}^x\!/_8}-1\right)/x\right)\;. \tag{9.3}$$

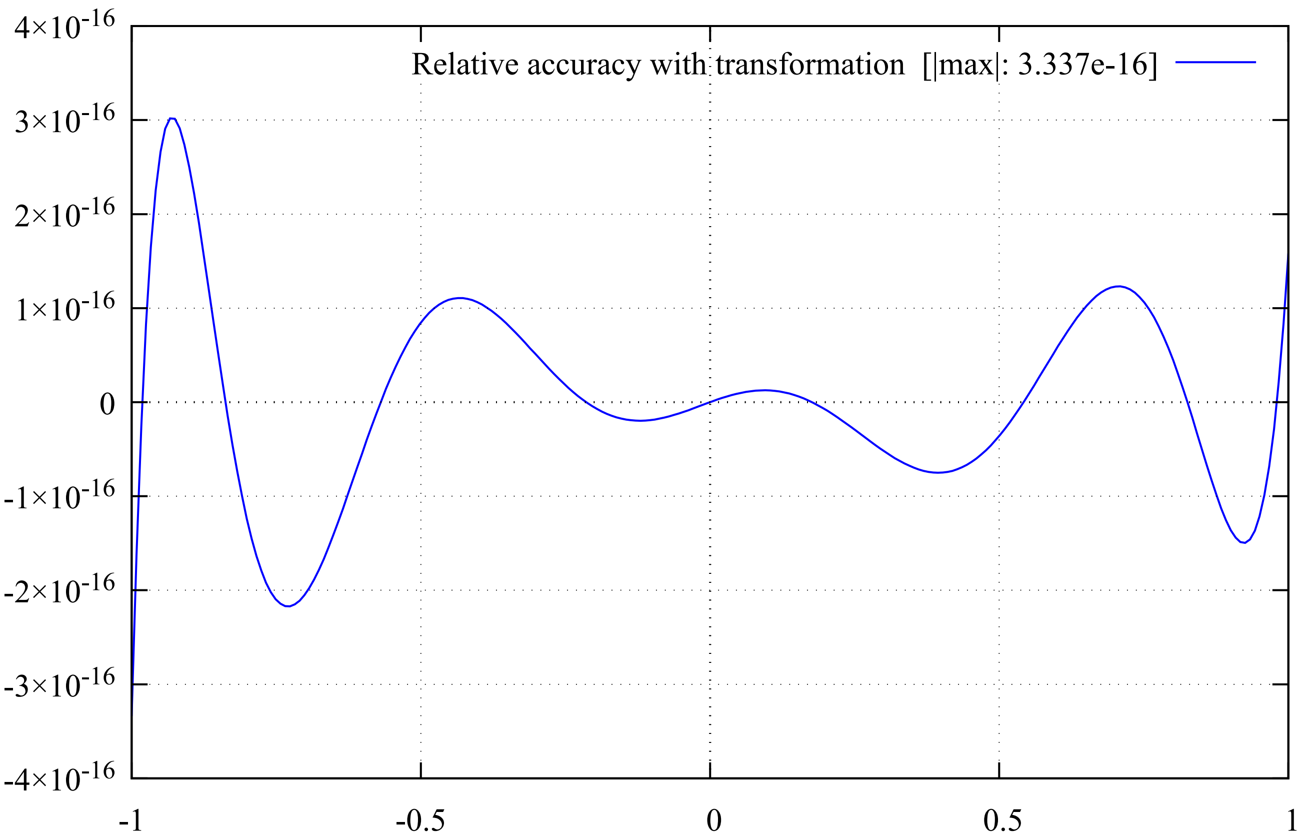
We seek a rational approximation $\tilde{g}(x)$ of order $R_{(3,4)}$ for $g(x)$.

We set the initial guess $\tilde{g}^{(0)}(x)$ to the linear Chebyshev-Padé approximation.

The relative error of $\tilde{g}^{(0)}(x)$, i.e., $(\frac{\tilde{g}^{(0)}(x)}{g(x)}-1)$:

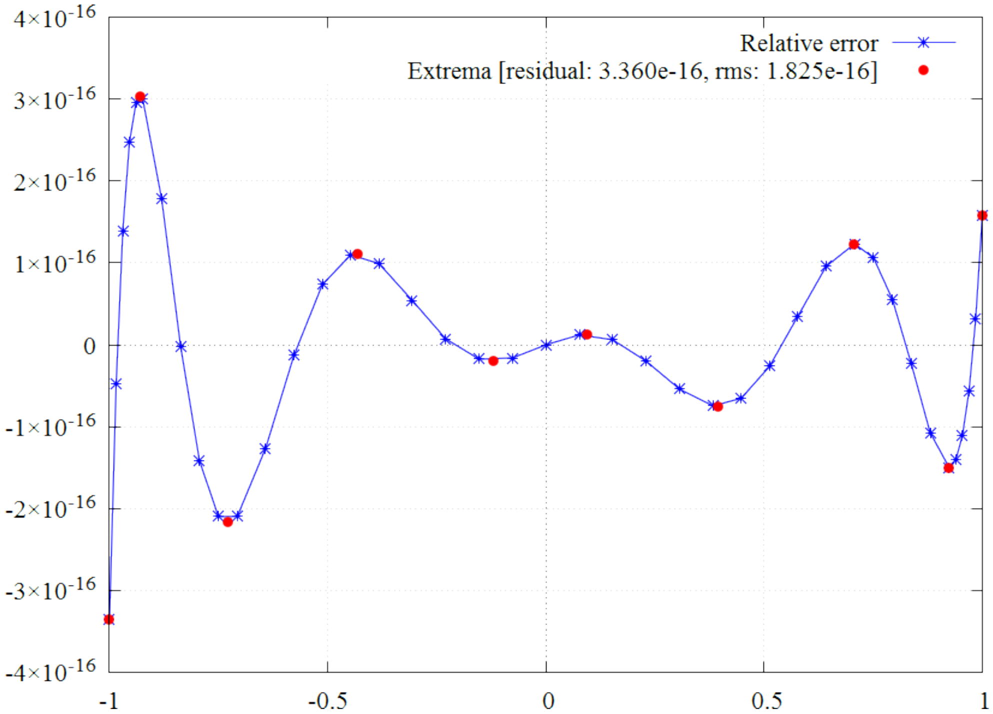


The transformed relative error function $\Delta^{(0)}(x) \equiv \frac{\theta(x,\tilde{g}^{(0)}(x))}{f(x)} - 1$:



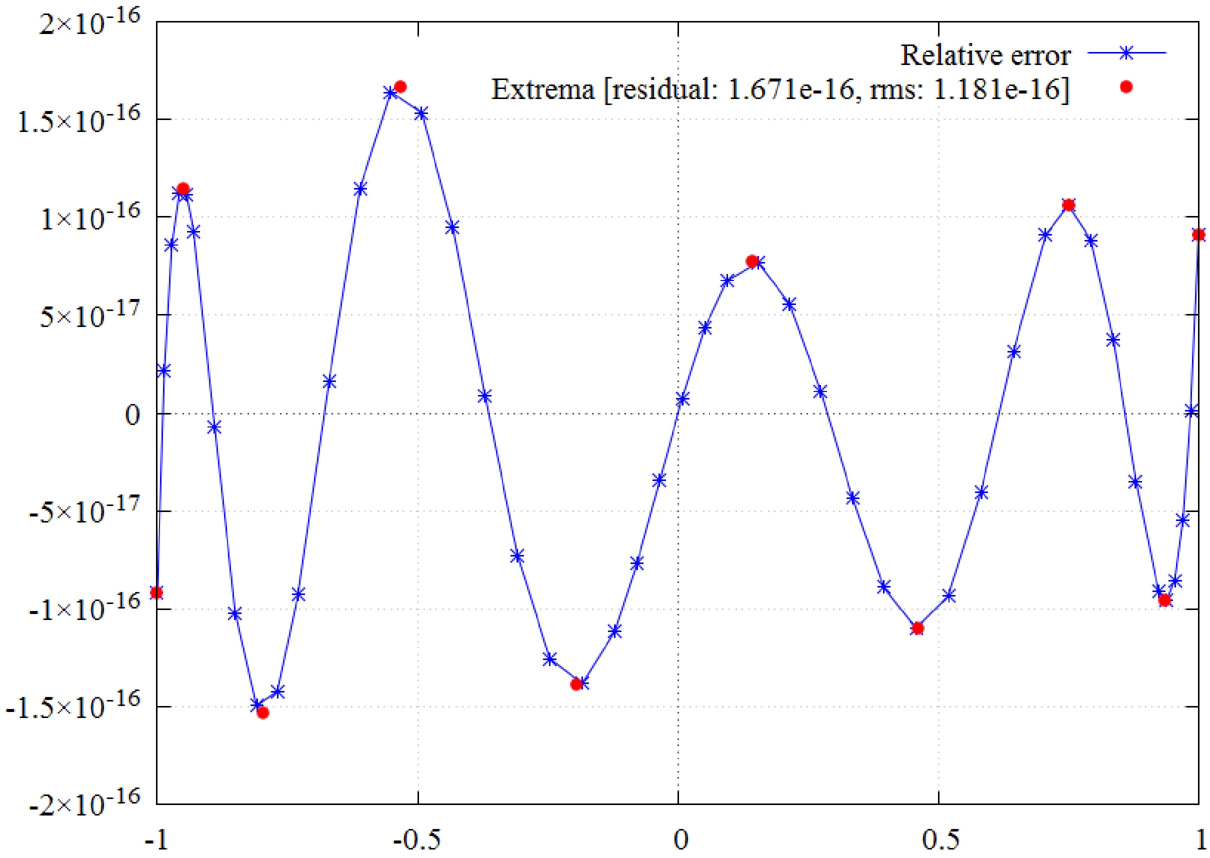
We initialise the reference points to the extrema of $\Delta^{(0)}(x)$.

Found 8 brackets of interior (local) extrema.
[0.026 seconds, completed at 2019−08−21 15:26:19+02:00]
Finding precise location of extrema ...
[0.048 seconds, completed at 2019−08−21 15:26:19+02:00]
Plotting ...



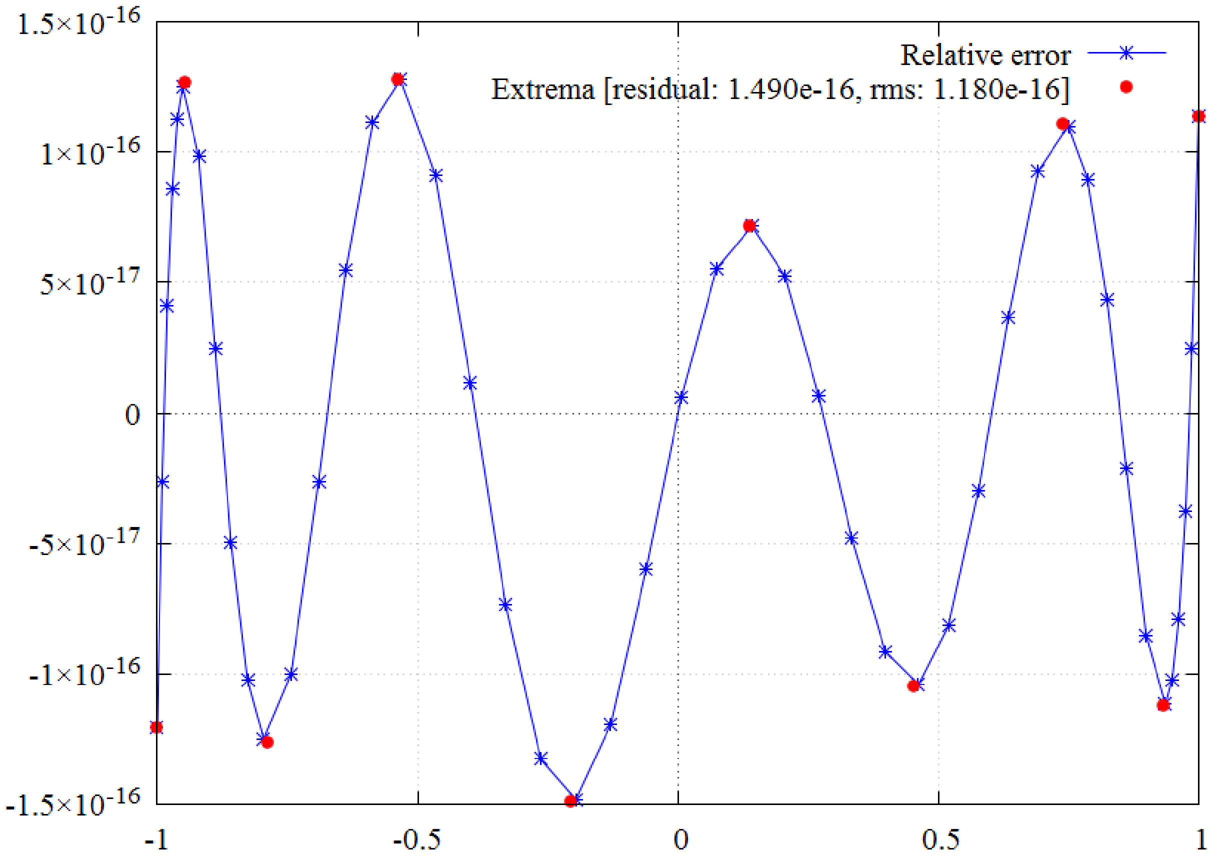
Iteration 1:
============
Setting up nonlinear system and its local linearization ...
The nonlinear system comprises 10 equations for 9 variables and is thus overdetermined.
Attempting iteration of least squares solutions of overdetermined system J·Δx = −r
via solution of (J^T·J)·Δx = −J^T·r in each step.
[0.006 seconds, completed at 2019−08−21 17:41:01+02:00]
Nonlinear system RMS error: 1.031e−16 ==(6 iterations)==> 1.000e−17
[0.11 seconds, completed at 2019−08−21 17:41:01+02:00]
Checking for singularities ...
No interior singularities found.
Found 8 brackets of interior (local) extrema.
[0.03 seconds, completed at 2019−08−21 17:41:01+02:00]
Finding precise location of extrema ...
[0.04 seconds, completed at 2019−08−21 17:41:01+02:00]
Plotting ...

$\Delta^{(1)}(x)$:

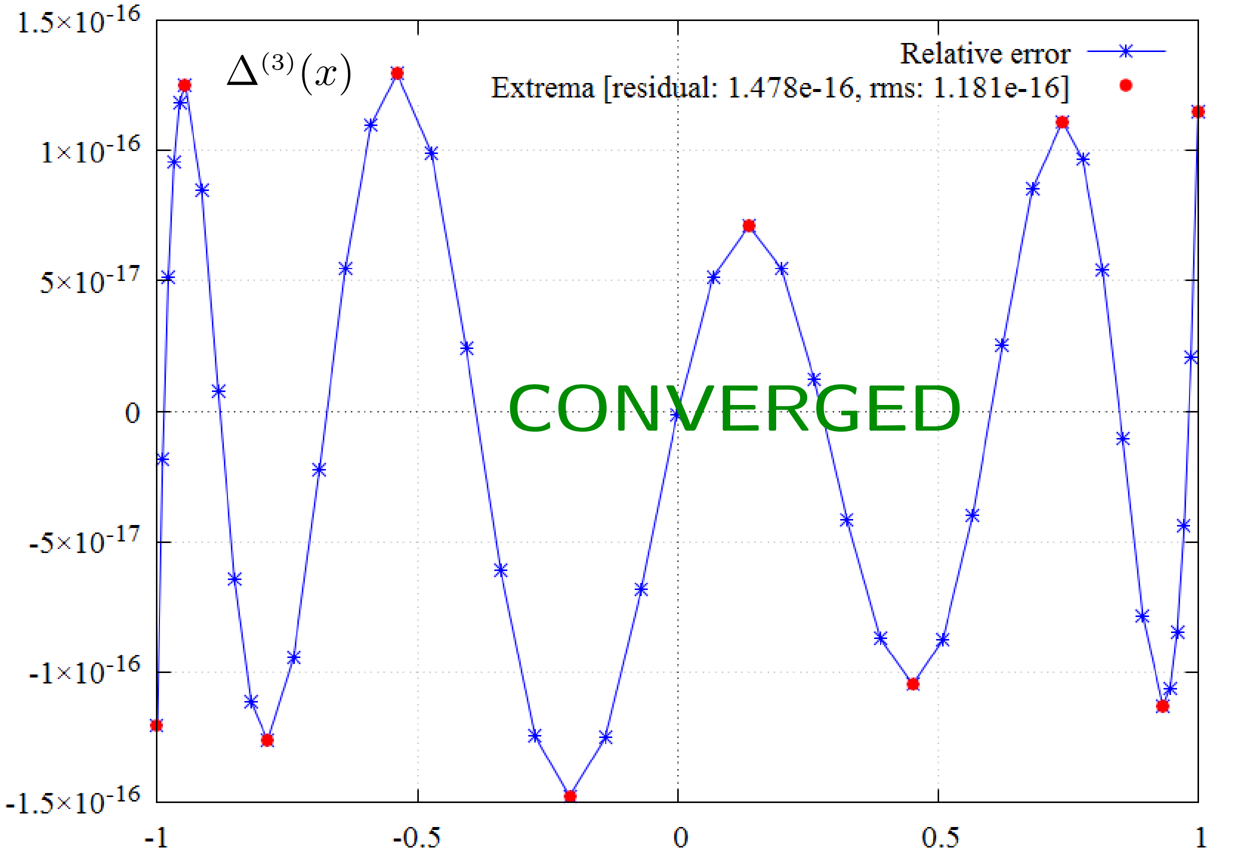


Iteration 2:
============
Setting up nonlinear system and its local linearization ...
The nonlinear system comprises 10 equations for 9 variables and is thus overdetermined.
Attempting iteration of least squares solutions of overdetermined system J·Δx = −r
via solution of (J^T·J)·Δx = −J^T·r in each step.
[0.006 seconds, completed at 2019−08−21 17:41:01+02:00]
Nonlinear system RMS error: 2.774e−17 ==(6 iterations)==> 1.882e−17
[0.106 seconds, completed at 2019−08−21 17:41:01+02:00]
Checking for singularities ...
No interior singularities found.
Found 8 brackets of interior (local) extrema.
[0.029 seconds, completed at 2019−08−21 17:41:02+02:00]
Finding precise location of extrema ...
[0.041 seconds, completed at 2019−08−21 17:41:02+02:00]
Plotting ...

$\Delta^{(2)}(x)$:

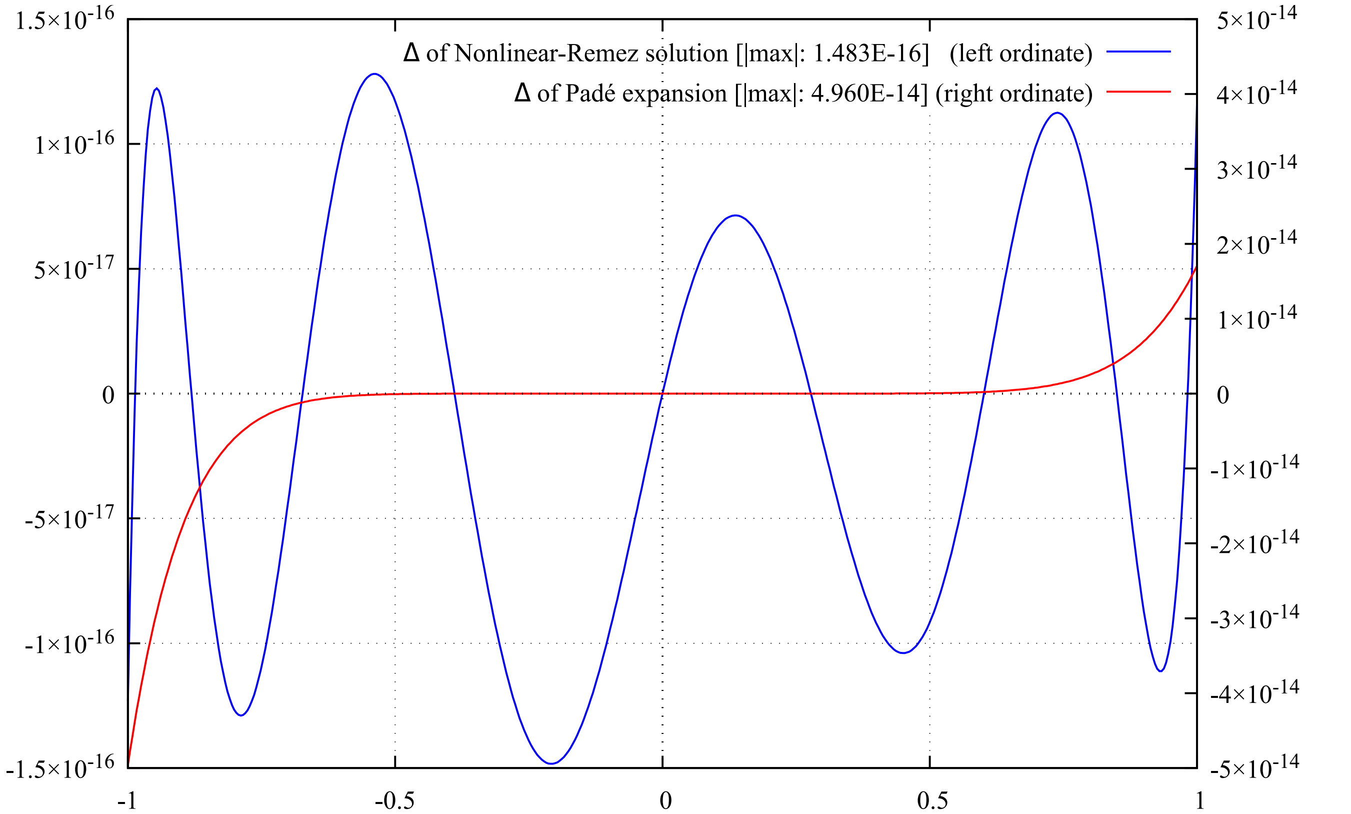


Iteration 3:
============
Setting up nonlinear system and its local linearization ...
The nonlinear system comprises 10 equations for 9 variables and is thus overdetermined.
Attempting iteration of least squares solutions of overdetermined system J·Δx = −r
via solution of (J^T·J)·Δx = −J^T·r in each step.
[0.007 seconds, completed at 2019−08−21 17:41:02+02:00]
Nonlinear system RMS error: 1.898e−17 ==(6 iterations)==> 1.896e−17
[0.107 seconds, completed at 2019−08−21 17:41:02+02:00]
Checking for singularities ...
No interior singularities found.
Found 8 brackets of interior (local) extrema.
[0.03 seconds, completed at 2019−08−21 17:41:02+02:00]
Finding precise location of extrema ...
[0.039 seconds, completed at 2019−08−21 17:41:02+02:00]
Plotting ...



$$\tilde{g}^{(3)}(x) = \frac{1.953124999999942E{-}3+x\cdot(3.671668869161992E{-}4+x\cdot(1.918897399977124E{-}5+x\cdot 2.409673162592619E{-}7))}{1+x\cdot(0.2504894461010475+x\cdot(0.0205975325688011+x\cdot(6.148741470367674E{-}4+x\cdot 4.822062201867934E{-}6)))}$$

Compare this with a $(3,4)$-Padé approximation (300 times larger error):



➡ $\mathrm{lnerfcx}^{-1}(\cdot)$

Recall

$$\Phi(z) \;=\; \tfrac{1}{2}\mathrm{erfc}(-\tfrac{z}{\sqrt{2}}) \tag{9.4}$$

and *the scaled complementary error function*

$$\mathrm{erfcx}(z) \;=\; \mathrm{e}^{z^2}\cdot\mathrm{erfc}(z)\;. \tag{9.5}$$

which diminishes like $1/z$ for $z \to +\infty$.

Now define

$$\mathrm{lnerfcx}(z) \;:=\; \ln(\mathrm{erfcx}(z))\;. \tag{9.6}$$

- We seek an approximation for the

  inverse logarithmic scaled complementary error function

$$\mathrm{lnerfcx}^{-1}(\cdot)\;. \tag{9.7}$$

Analytics

We make heavy use of [AS84, p. 26.2.12]:

$$\Phi(z) \;\approx\; -\varphi(z)\cdot\left(\tfrac{1}{z}-\tfrac{1}{z^3}+\ldots\right) \quad \text{for} \quad z\to-\infty\;. \tag{26.2.12}$$

We define

$$\lambda := \mathrm{lnerfcx}(z) \tag{9.8}$$

as the *input* value for our inverse function, and we wish to compute

$$z(\lambda) \;\equiv\; \mathrm{lnerfcx}^{-1}(\lambda)\;. \tag{9.9}$$

We obtain for:-

- $z \to -\infty$ (and thus $\lambda \to +\infty$):

$$\lambda \approx \ln(2) + z^2 + \frac{1}{\sqrt{4\pi}} \cdot \frac{e^{-z^2}}{z} \tag{9.10}$$

$$z \approx -\tfrac{1}{\sqrt{\omega}} \cdot \left(1 + e^{-\frac{1}{\omega}\cdot(1-\frac{3}{2}\omega\ln\omega)}/\sqrt{16\pi}\right) \tag{9.11}$$

with

$$\boxed{\omega := \tfrac{1}{\lambda-\ln(2)}\ .} \tag{9.12}$$

For the inverse function in this region, we choose

$$\theta(\omega,\zeta) := -\tfrac{1}{\sqrt{\omega}} \cdot \left(1 + e^{-\zeta/\omega}/\sqrt{16\pi}\right) \tag{9.13}$$

and seek a rational function approximation for $\zeta = \zeta(\omega)$.

- For z near zero (and thus λ also near zero):

$$\lambda = -\tfrac{2}{\sqrt{\pi}} \cdot z + (1 - \tfrac{2}{\pi}) \cdot z^2 + \mathcal{O}(z^3) \tag{9.14}$$

$$z = -\tfrac{\sqrt{\pi}}{2} \cdot \lambda + \tfrac{\sqrt{\pi}}{4} \cdot (\tfrac{\pi}{2} - 1) \cdot \lambda^2 + \mathcal{O}(\lambda^3) \tag{9.15}$$

Here, we choose

$$\theta(\lambda,\gamma) := \lambda \cdot (-\tfrac{\sqrt{\pi}}{2} + \lambda \cdot \gamma) \tag{9.16}$$

and find a rational function approximation for $\gamma = \gamma(\lambda)$.

- In the limit $z \to +\infty$, and $\boxed{y := e^{\lambda} \to 0}$:

$$y \approx \tfrac{1}{\sqrt{\pi}} \cdot \tfrac{1}{z} \cdot \left(1 - \tfrac{1}{2} \cdot \tfrac{1}{z^2} + \tfrac{3}{4} \cdot \tfrac{1}{z^4} + \mathcal{O}(z^{-6})\right) \tag{9.17}$$

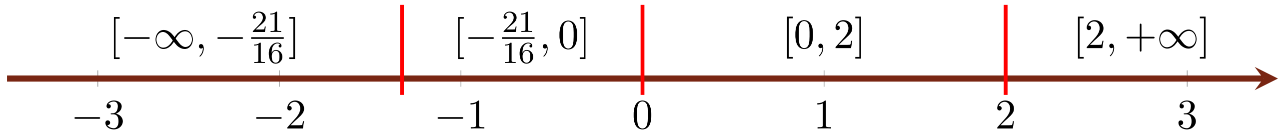
$$\frac{1}{z} \approx \sqrt{\pi} \cdot y \cdot \left(1 + \tfrac{\pi}{2} \cdot y^2 + \underbrace{\mathcal{O}(y^6)}_{\text{yes, really, due to cancellation of } \mathcal{O}(y^4)!}\right) \tag{9.18}$$

We choose

$$\theta(y,R) := \tfrac{1}{\sqrt{\pi}\cdot y} \cdot (1 - y^2 \cdot R) \tag{9.19}$$

and seek a rational function approximation for $R = R(\mu)$ with $\boxed{\mu := y^2}$.

We split the domain $z \in [-\infty, \infty]$ into four intervals:



NOTE: for the purpose of the Nonlinear-Remez algorithm, we map all (respectively transformed) intervals to $[-1, 1]$ via an affine abscissa mapping.

We start with the leftmost region with $\boxed{\omega := 1/(\lambda - \ln 2)}$ where

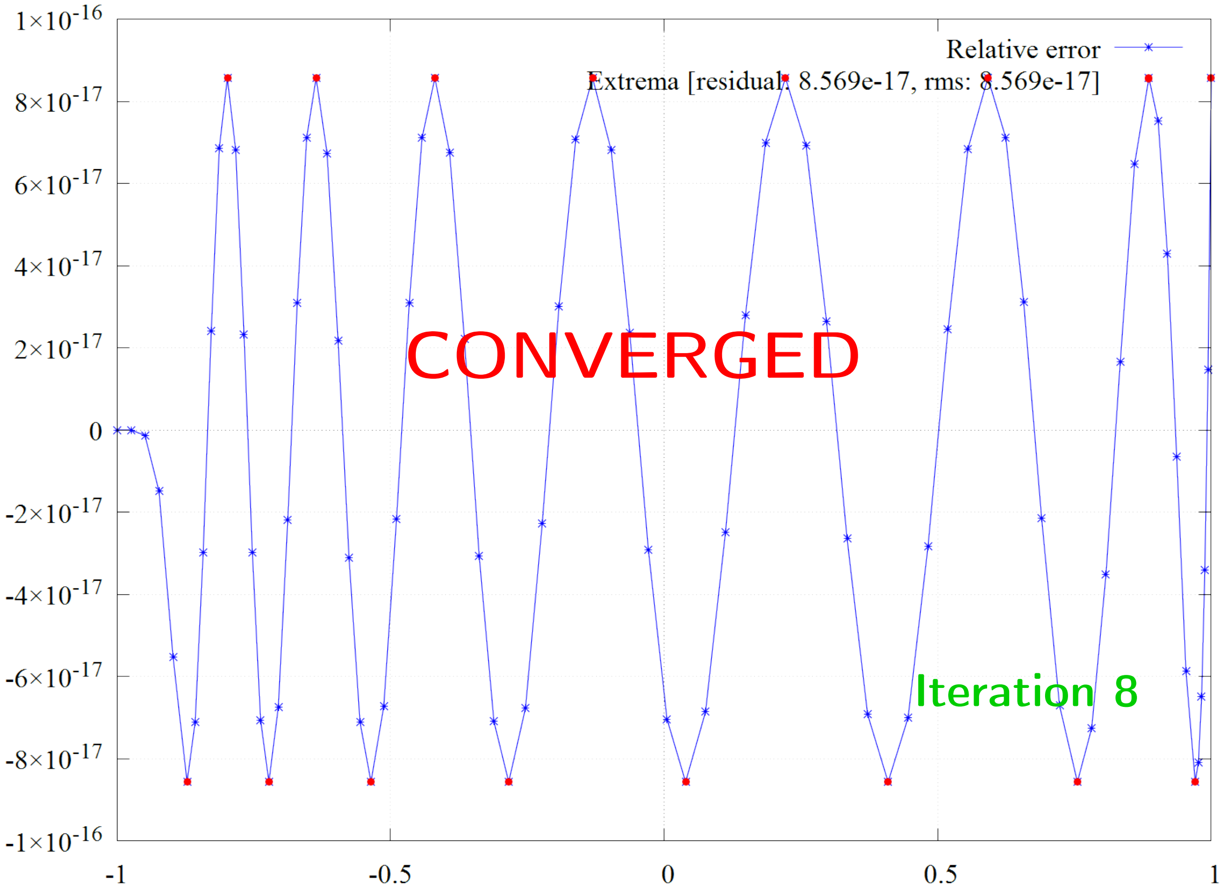
$$z \in [-\infty, -1.3125] \Leftrightarrow \lambda \in [+\infty, 2.383574] \Leftrightarrow \omega \in [0, 0.591567]$$

and $z = z(\lambda) = \theta(\omega, \zeta(\omega))$ given by

$$\theta(\omega, \zeta) \;:=\; -\tfrac{1}{\sqrt{\omega}} \cdot \left(1 + \mathrm{e}^{-\zeta/\omega} / \sqrt{16\pi}\right) \tag{9.13}$$

to find an $R_{(7,7)}$ approximant for $\zeta(\omega)$.

Note: ***this is the most difficult segment.***



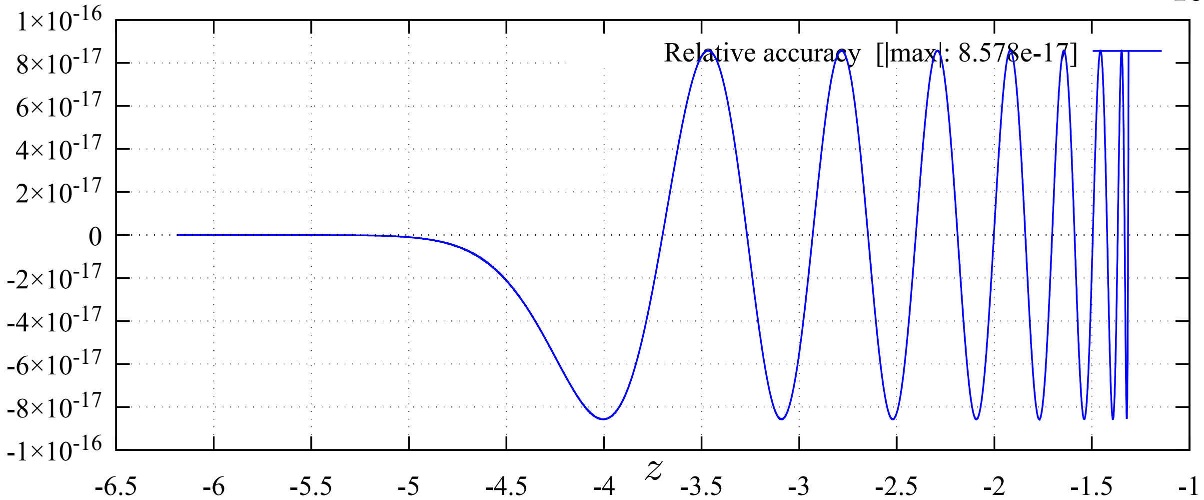
Far left z region

The result is $\zeta(\omega) = \frac{\sum_{j=0}^{7} p_j \omega^j}{\sum_{j=0}^{7} q_j \omega^j}$ with

j	p_j	q_j
0	1.0095166315027250	1
1	5.2198808595043985E1	4.4778820521335590E1
2	6.6489488281084671E2	4.7213770185171718E2
3	5.7172982928421216E3	4.1649570148555720E3
4	2.7640276859915855E4	1.5305272267032917E4
5	3.1445869626862596E4	1.5339137807482323E4
6	−6.2956788351087912E3	3.0196766794574674E3
7	−7.3239092097224373E3	−5.8212790910641370E1

(9.20)

with net accuracy of $\mathrm{lnerfcx}^{-1}(\cdot)$ better than 8.6E−17 on $z \in [-\infty, -\frac{21}{16}]$.



Next, the centre-left region where

$$z \in [-1.3125, 0] \quad \Leftrightarrow \quad \lambda \in [2.383574, 0]$$

(note the inverted bounds for λ) and $z = z(\lambda) = \theta(\lambda, \gamma(\lambda))$ is given by

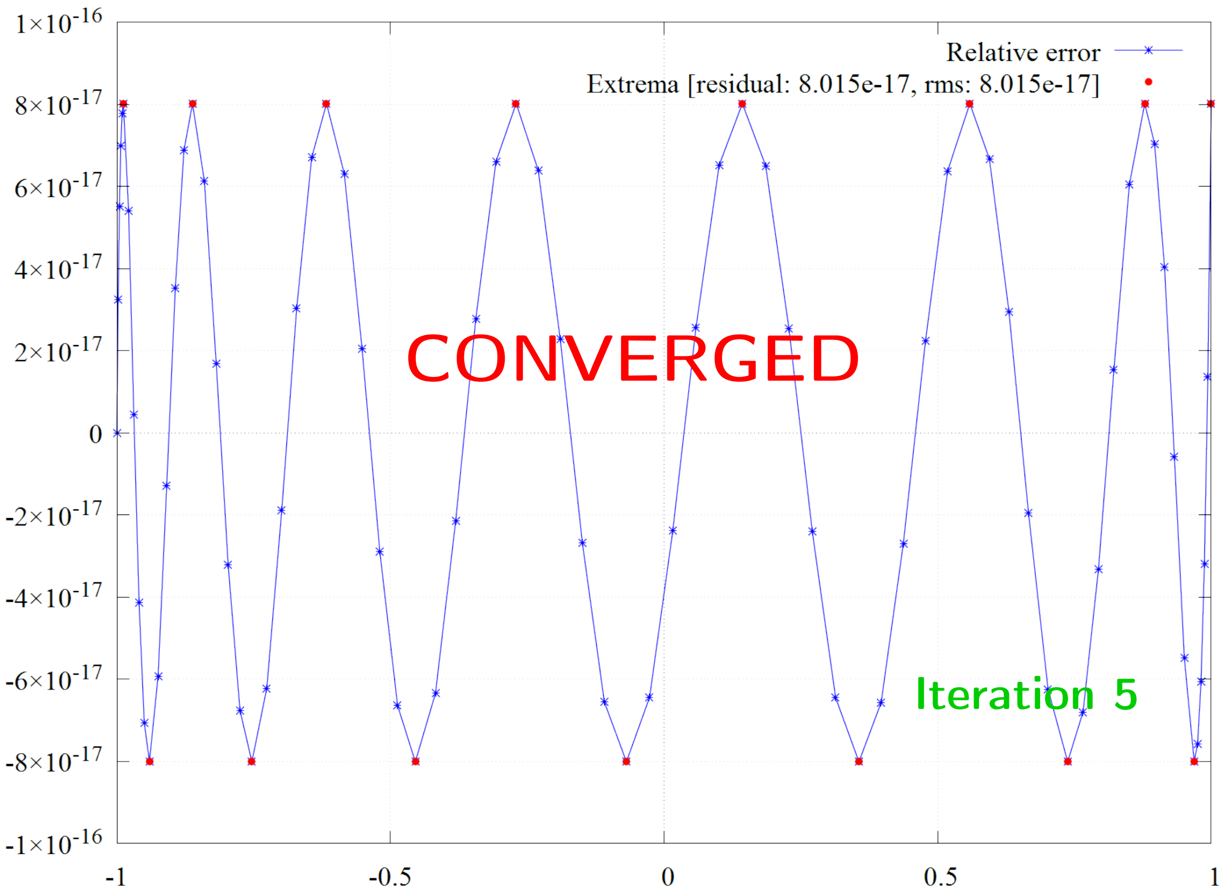
$$\theta(\lambda, \gamma) \;:=\; \lambda \cdot (-\tfrac{\sqrt{\pi}}{2} + \lambda \cdot \gamma) \tag{9.16}$$

We optimize an $R_{(6,7)}$ approximant for $\gamma(\lambda)$ using our Nonlinear-Remez algorithm (which is much faster in this region):

Nonlinear−Remez initialized

[**0.015**]
Found 14 brackets of interior (local) extrema.
[0.154 seconds, completed at 2019−08−26 15:20:04+02:00]
Finding precise location of extrema ...
[0.218 seconds, completed at 2019−08−26 15:20:04+02:00]
Interpreting leftmost (local) extremum as replacement of left edge maximum.
Plotting ...

Centre-left z region



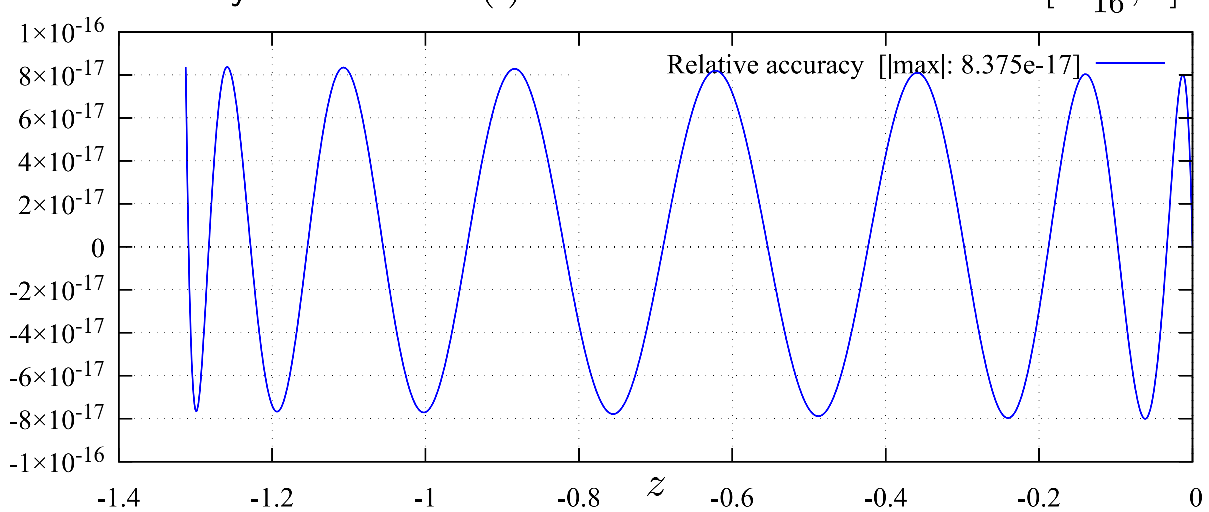
Centre-left z region

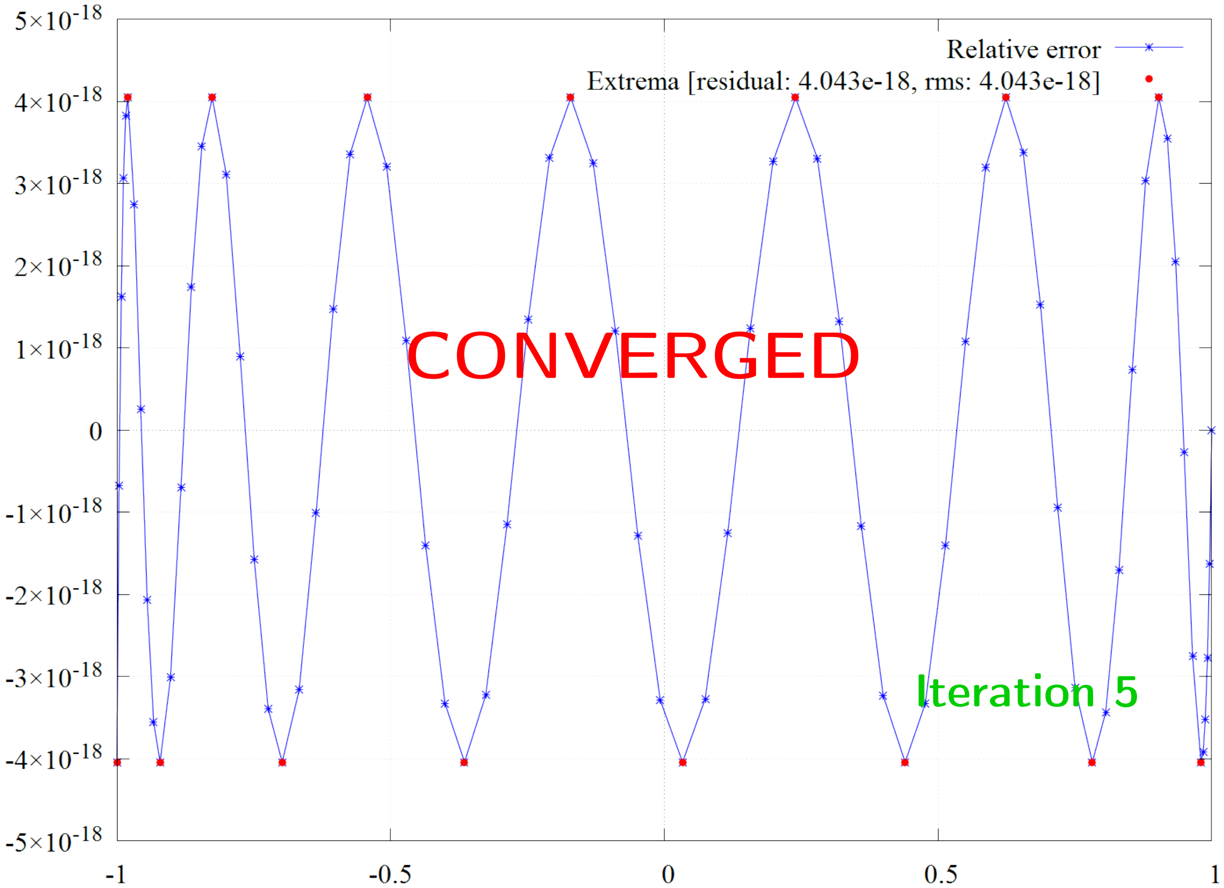
The result is $\gamma(\lambda) = \frac{\sum_{j=0}^{6} p_j \lambda^j}{\sum_{j=0}^{7} q_j \lambda^j}$ with

j	p_j	q_j
0	2.5292753687757275E−1	1
1	2.1452708953922655E−1	1.16832628651416550
2	1.2024361368623330E−1	7.5851282175746919E−1
3	4.1558800861885828E−2	3.2261336129477141E−1
4	9.6174529928918345E−3	9.3691654909889110E−2
5	1.1691581040156436E−3	1.7501961671284489E−2
6	4.9817716885778323E−5	1.7797521724949996E−3
7		6.0924354073989308E−5

(9.21)

with net accuracy of $\mathrm{lnerfcx}^{-1}(\cdot)$ better than 8.4E−17 on $z \in [-\frac{21}{16}, 0]$.





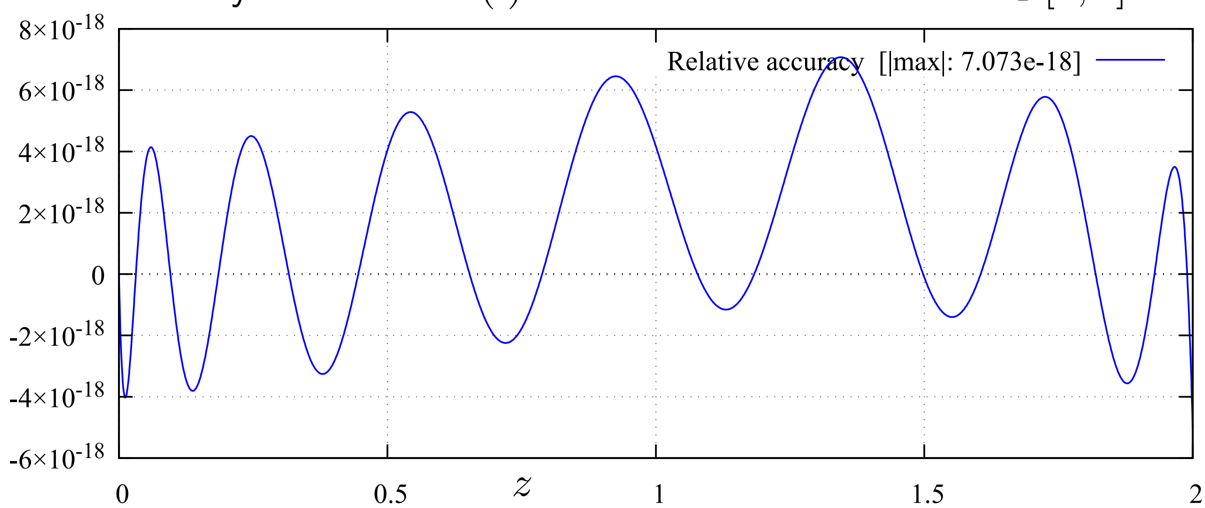
Centre-right z region

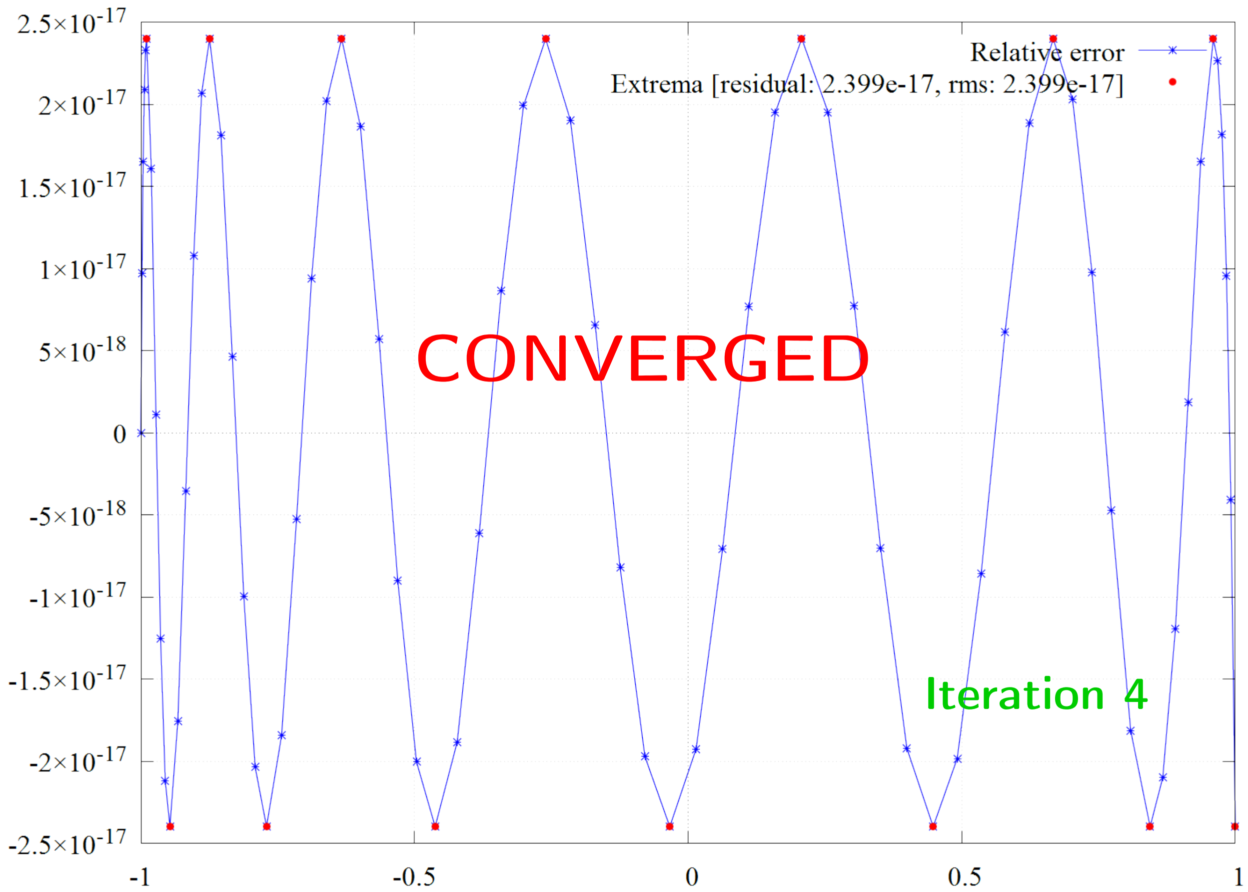
The result is $\gamma(\lambda) = \dfrac{\sum_{j=0}^{6} p_j \lambda^j}{\sum_{j=0}^{7} q_j \lambda^j}$ with

j	p_j	q_j
0	2.5292753687758385E−1	1
1	2.5303077883189806E−1	1.3205583833773353
2	1.3802754023765619E−1	8.7756229564117284E−1
3	4.3218642036409986E−2	3.5344645149617066E−1
4	8.5850273448102187E−3	9.1959520651133049E−2
5	9.4705311261385956E−4	1.5240338041796014E−2
6	5.1048163821843303E−5	1.4766888946846876E−3
7		6.4213512630622214E−5

(9.22)

with net accuracy of $\mathrm{lnerfcx}^{-1}(\cdot)$ better than 7.1E−18 on $z \in [0,2]$.





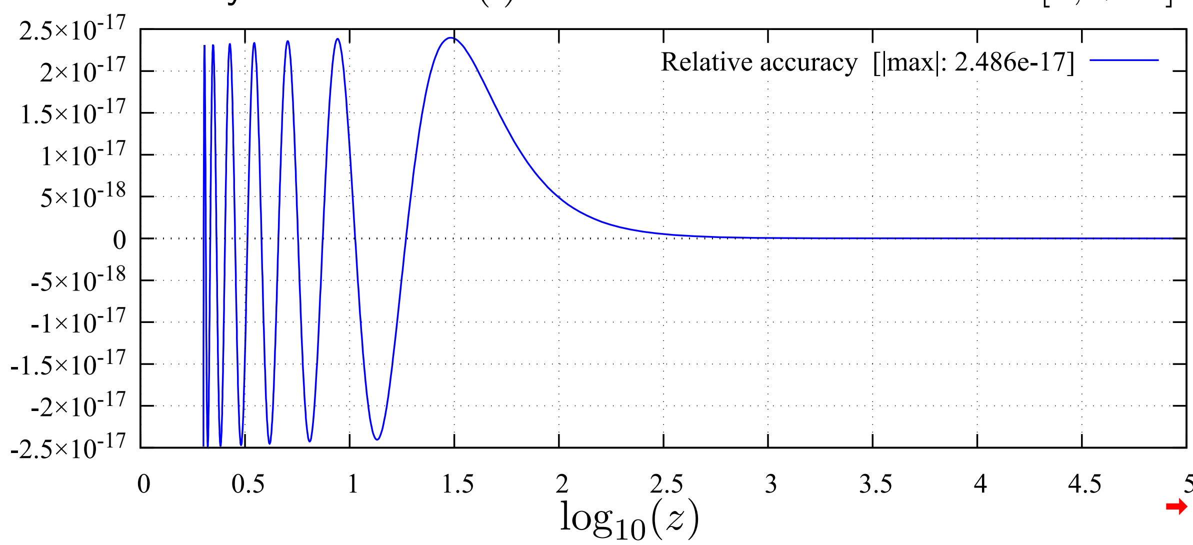
Far right z region

The result is $R(\mu) = \frac{\sum_{j=0}^{6} p_j \mu^j}{\sum_{j=0}^{6} q_j \mu^j}$ where $z(\lambda) = \frac{1 - y^2 \cdot R(y^2)}{y \cdot \sqrt{\pi}}$ with $y := \mathrm{e}^{\lambda}$ and

j	p_j	q_j
0	1.5707963267947315	1
1	1.2913020647185905E2	8.3777638976466067E1
2	3.6369729625814331E3	2.4370969028801017E3
3	4.2747204469378166E4	3.0319692409643002E4
4	2.0338989025588868E5	1.6031208819752618E5
5	3.1425713522153701E5	3.0823996502627300E5
6	7.3507808703235503E4	1.3524524137416423E5

(9.23)

with net accuracy of $\text{lnerfcx}^{-1}(\cdot)$ better than 2.5E−17 on $z \in [2, +\infty]$.



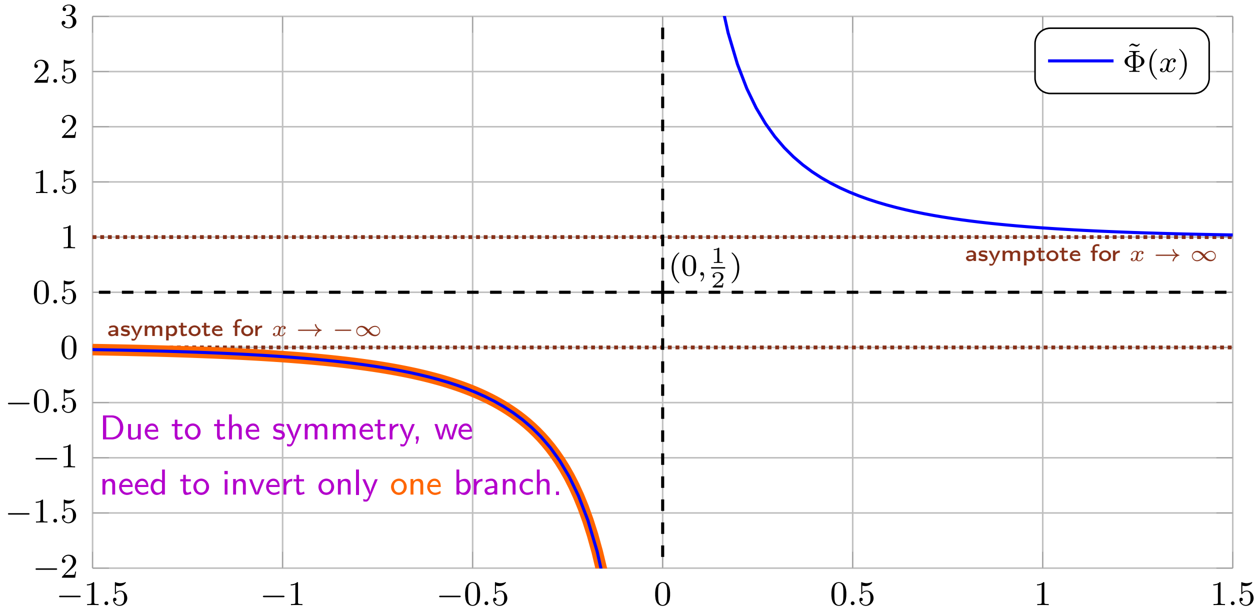
$\rightarrow \tilde{\Phi}^{-1}(\cdot)$

As explained, e.g., in [Jäc17], computing the implied *Bachelier* (aka *normal*) volatility reduces to the computation of the inverse of

$$\tilde{\Phi}(x) \; := \; \Phi(x) + \frac{\varphi(x)}{x} \; . \tag{9.24}$$

$\boxed{\tilde{\Phi}(x) \text{ is odd around } (0, \tfrac{1}{2}).}$

$\boxed{\tilde{\Phi}(x) \text{ diverges like } \tfrac{1}{x} \text{ for } x \to 0.}$



For $-9/4 < x < 0$, we define

$$g := \frac{1}{\tilde{\Phi} - \frac{1}{2}} \tag{9.25}$$

which is linear in x in lowest order near $x \to 0$:

$$g = \sqrt{2\pi} \cdot x \cdot \left[1 - \tfrac{1}{2}x^2 + \mathcal{O}(x^4)\right] \tag{9.26}$$

Standard use of the Lagrange inversion theorem gives us

$$x = \frac{g}{\sqrt{2\pi}} \cdot (1 + \tfrac{1}{4\pi}g^2 + \mathcal{O}(g^4)) \tag{9.27}$$

and hence we choose to set

$$x = g \cdot (\tfrac{1}{\sqrt{2\pi}} + g^2 \cdot \xi(g^2)) \tag{9.28}$$

and seek a (3,3)-rational function approximation for $\xi(g^2)$.

For $x_{-\infty} \leq x < -9/4$, with $x_{-\infty} \approx -38.28$ chosen such that

$$\tilde\Phi(x_{-\infty}) \;=\; -\text{DBL\_EPSILON}\cdot\text{DBL\_MIN} \;\approx\; 4.9\cdot 10^{-324}\;, \tag{9.29}$$

we once again make use of [AS84, p. 26.2.12], define

$$\eta := \sqrt{-\ln(-\tilde\Phi)}\;, \tag{9.30}$$

set

$$x = \chi(\eta)\;, \tag{9.31}$$

and seek a (3,3)-rational approximation for $\chi(\eta)$.

In both sub-intervals,

we improve the respective approximation with *one* HH_3 *polishing* step.

NOTE: the Householder-3 method is of 4-th order convergence.

Thus, a relative error[19] of magnitude ε will be reduced to $\mathcal{O}(\varepsilon^4)$.

[19] within the convergence radius of the method

Recall that the Householder-3 method to find a root of $f(x)$, is given by

$$x_{n+1} \;=\; x_n + \mathrm{HH}_3(x_n) \tag{3.8}$$

where

$$\mathrm{HH}_3 \;=\; \frac{\nu(1+h_2\nu/2)}{1+\nu(h_2+h_3\nu/6)} \tag{3.12}$$

with $\nu := -f(x)/f'(x)$ and $h_k := f^{(k)}(x)/f'(x)$.

By good fortune, the HH_3 increment can be simplified to

$$\mathrm{HH}_3(x) \;=\; \frac{3qx^2\cdot(2-qx\cdot(2+x^2))}{6+qx\cdot(-12+x\cdot(6q+x\cdot(-6+qx\cdot(3+x^2))))} \tag{9.32}$$

with

$$q \;:=\; \frac{\tilde\Phi(x)-\tilde\Phi^*}{\varphi(x)}\;. \tag{9.33}$$

where $\tilde\Phi^*$ represents the value for which we seek x such that $\tilde\Phi(x) = \tilde\Phi^*$.

There is one ***major*** snag with this approach:

the relative error profile of our initial approximation for x will be

enormously distorted by the HH_3 step!

In other words, once we have optimised $\xi(g^2)$ and $\chi(\eta)$ to minimize the relative error of x as given by (9.28) for the inner and by (9.31) for the outer branch ($x < -9/4$), the relative error of the HH_3-polished value of x will display a very different, *undesirable*, profile.

This is where the Nonlinear-Remez algorithm can show its full power!

➡ *We include the HH_3 step*

in the definition of the nonlinear transformation function $\theta(\cdot,\cdot)$.

In formulae, for the optimisation of the rational function

$$\chi(\eta) \quad \text{for} \quad x < -{}^9\!/\!_4\,,$$

and

$$\xi(g^2) \quad \text{on} \quad x \in [-{}^9\!/\!_4, 0]\,,$$

we define the auxiliary (pre-polishing) quantities

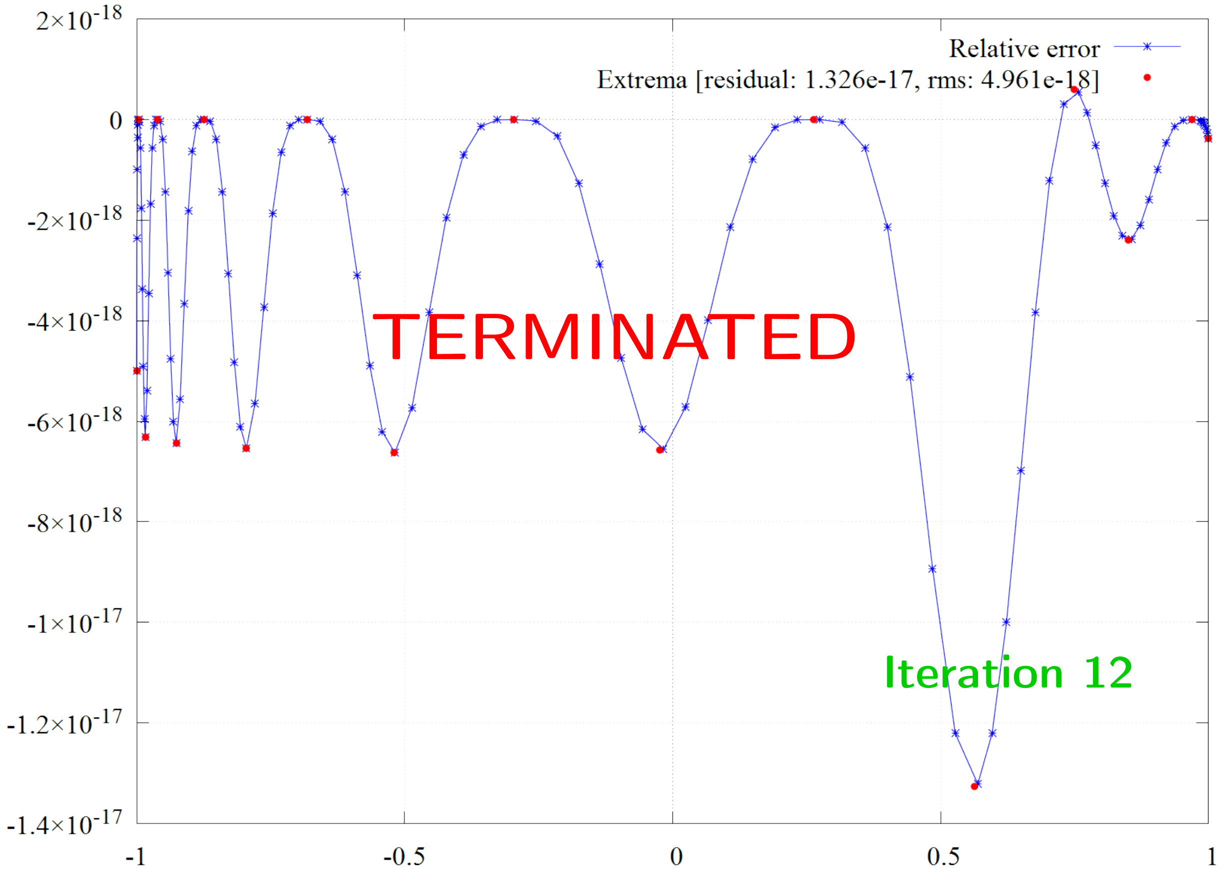
$$\begin{aligned} \bar{x}(\eta,\chi) &:= \chi && \text{when} \quad x < -{}^9\!/\!_4 \quad , \\ \bar{x}(g,\xi) &:= g\cdot\left(\tfrac{1}{\sqrt{2\pi}} + g^2\cdot\xi\right) && \text{else} \quad , \end{aligned} \tag{9.34}$$

and then use the nonlinear transformation functions

$$\begin{aligned} \theta(\eta,\chi) &:= \bar{x}(\eta,\chi) + \mathrm{HH}_3(\bar{x}(\eta,\chi)) && \text{when} \quad x < -{}^9\!/\!_4 \quad , \\ \theta(g,\xi) &:= \bar{x}(g,\xi) + \mathrm{HH}_3(\bar{x}(g,\xi)) && \text{else} \quad . \end{aligned} \tag{9.35}$$

Disclosure:

In practice, the optimisation had to be broken into stages to achieve convergence, namely, first optimising with HH_1, then using that result for an optimisation with HH_2, and so on. In the following, we only show the final stage with HH_3.



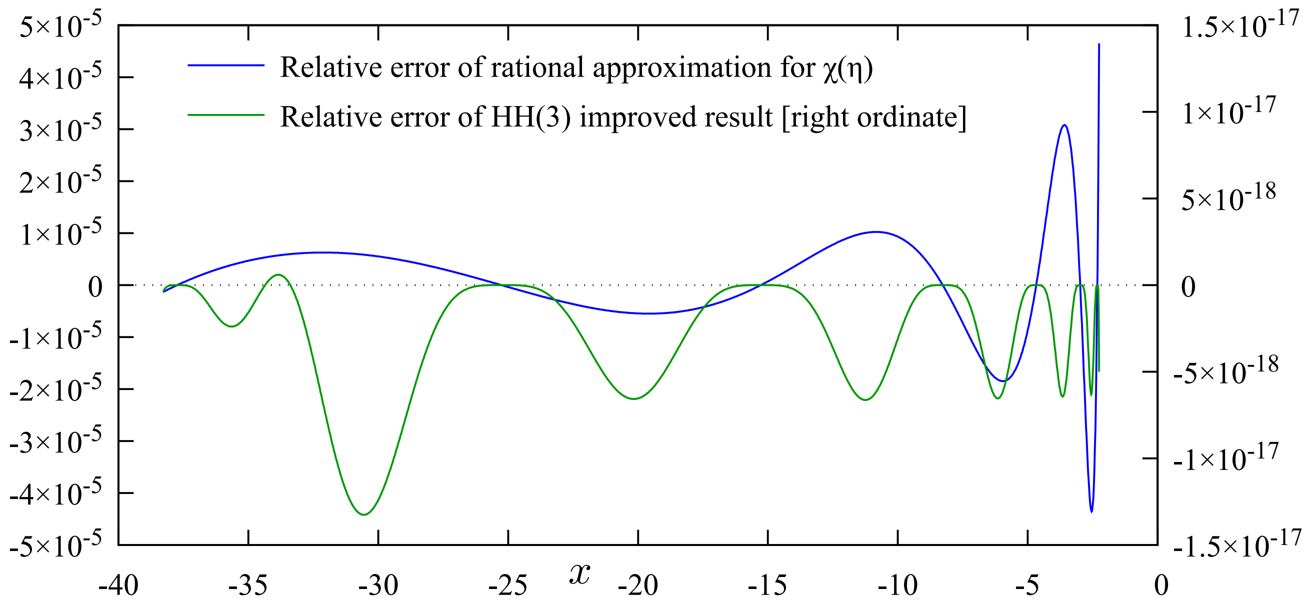
$x \in [x_{-\infty}, -9/4]$

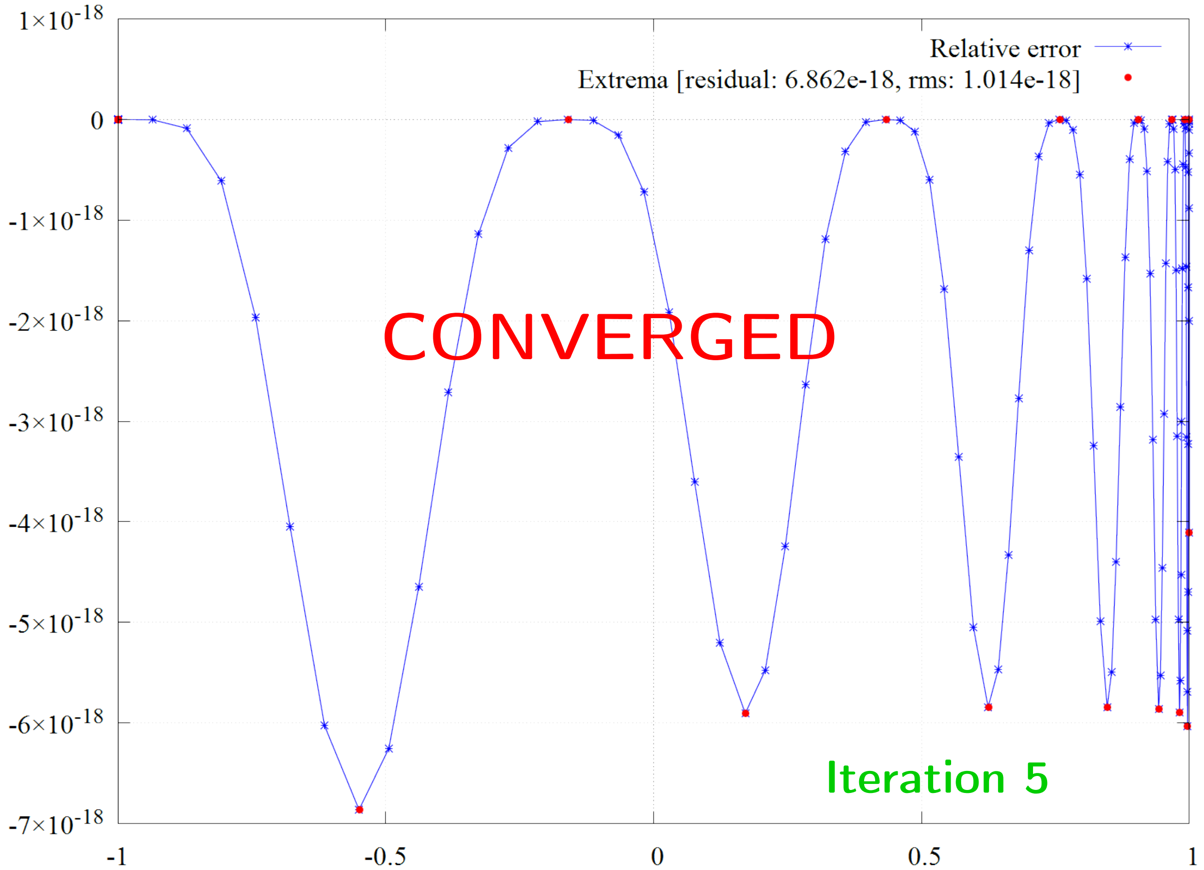
We obtain $\chi(\eta) = \dfrac{\sum_{j=0}^{3} p_j \eta^j}{\sum_{j=0}^{3} q_j \eta^j}$ where

j	p_j	q_j
0	9.7738561392	1
1	−9.9078818367	−0.66464958035
2	0.58896177476	−1.5675449008
3	2.225343956	−7.0388793356E−5

(9.36)

with net accuracy of $\tilde{\Phi}^{-1}(\cdot)$ better than 1.4E−17 on $x \in [x_{-\infty}, -9/4]$.





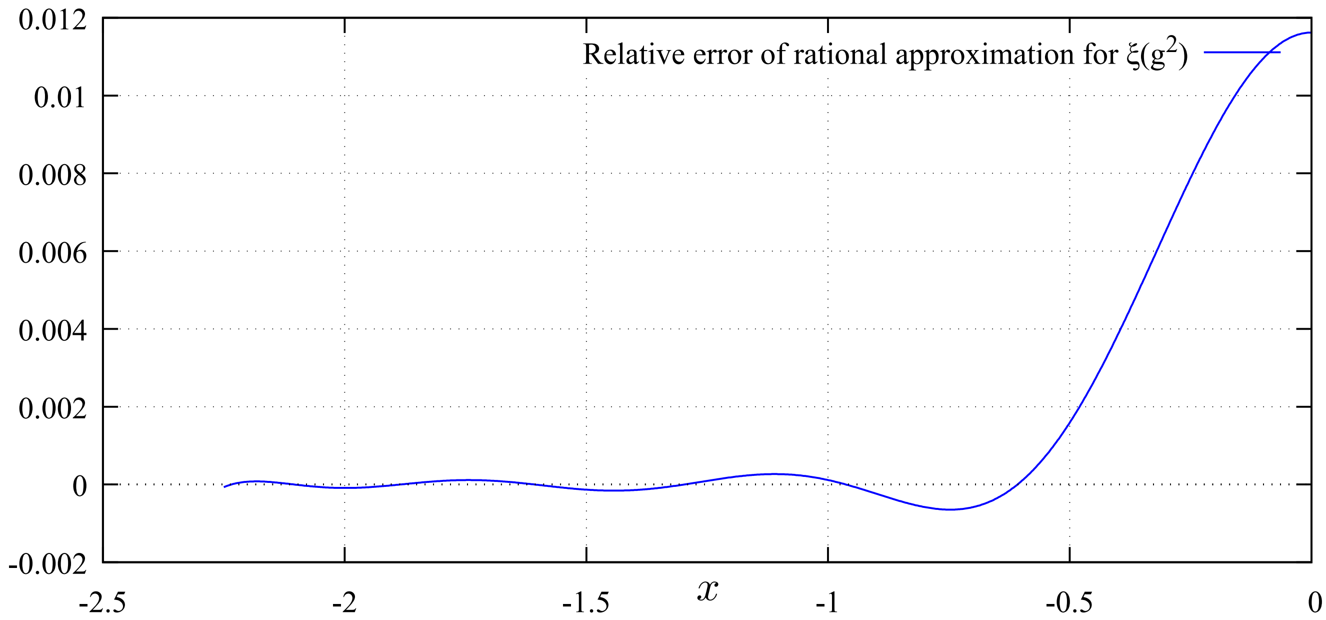
$x \in [-9/4, 0)$

Finally, we have $\xi(g^2) = \dfrac{\sum_{j=0}^{3} p_j g^{2j}}{\sum_{j=0}^{3} q_j g^{2j}}$ where

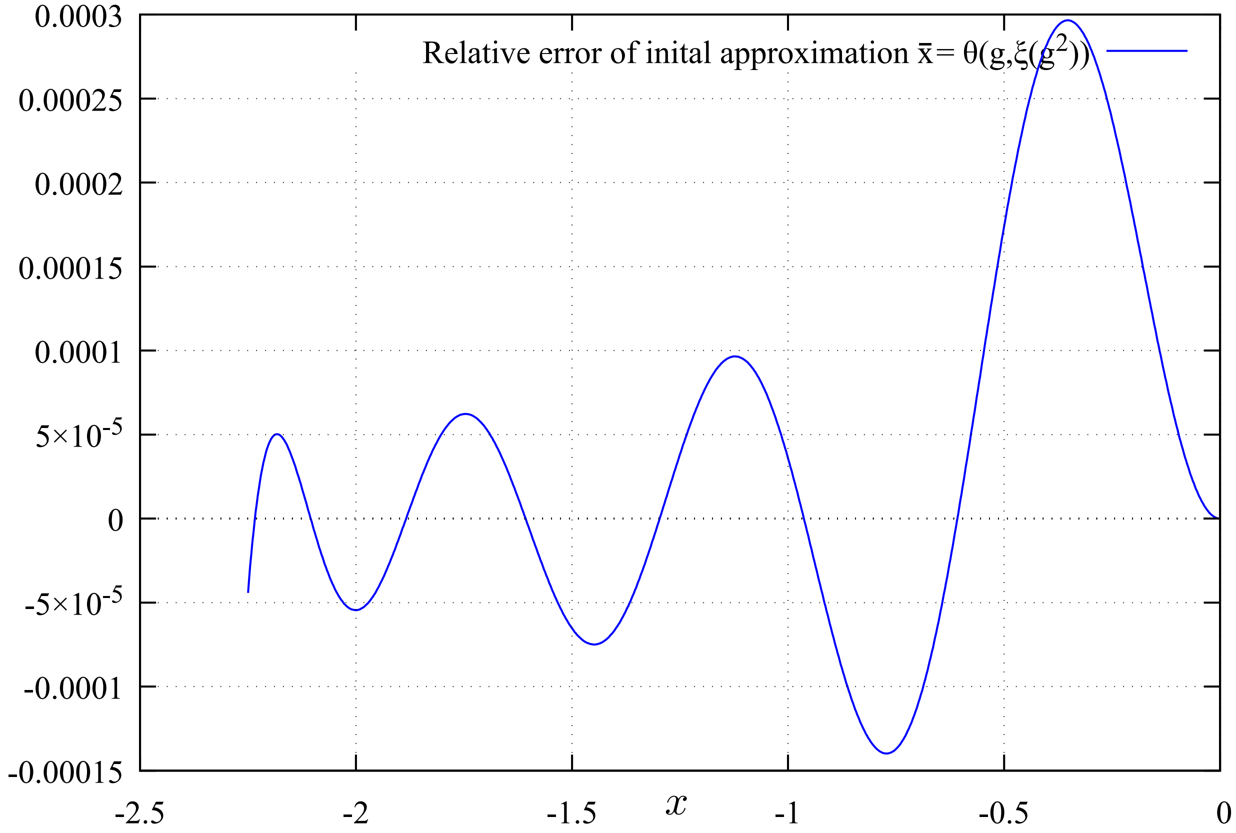
j	p_j	q_j
0	0.0321157192137	1
1	−0.0169727106645	−0.663608954938
2	2.62178706002E−3	0.145308053672
3	− 9.61687028942E−5	−0.010475324293

. (9.37)

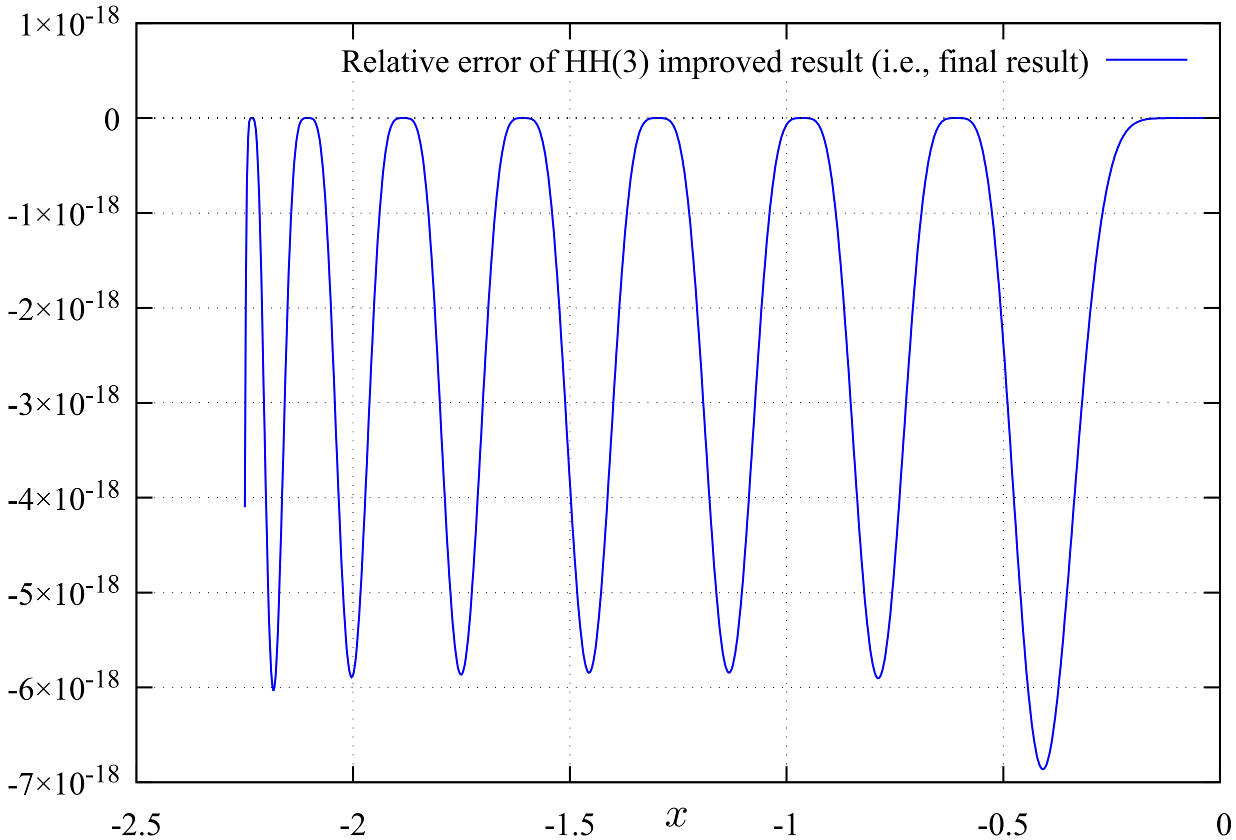
The accuracy of $\xi(g^2)$ is no better than 10^{-2} and has a peculiar profile!



The accuracy of the initial approximation $\bar{x}$ is also uneven, as predicted.



The accuracy of the final HH_3-polished solution, however, is almost equi-oscillatory, as desired, albeit [mostly] unidirectional (for HH_k with odd k).



And that's how the magic was done. Below the final formulae in [Jäc17].

We assume $\tilde{\Phi}^* < 0$ by conditioning on out-of-the money options and solve $\tilde{\Phi}(x^*) = \tilde{\Phi}^*$.

If $\tilde{\Phi}^* < \tilde{\Phi}^*_C$ with $\tilde{\Phi}^*_C := \tilde{\Phi}(-9/4) \approx -0.001882039271$, set

$$g := 1/(\tilde{\Phi}^* - {}^1\!/\!_2) \tag{2.1}$$

$$\bar{\xi} := \frac{0.032114372355 - g^2 \cdot (0.016969777977 - g^2 \cdot (2.6207332461\text{E}{-3} - 9.6066952861\text{E}{-5} \cdot g^2))}{1 - g^2 \cdot (0.6635646938 - g^2 \cdot (0.14528712196 - 0.010472855461 \cdot g^2))} \tag{2.2}$$

$$\bar{x} := g \cdot \left(\tfrac{1}{\sqrt{2\pi}} + \bar{\xi} \cdot g^2\right) \tag{2.3}$$

else set

$$h := \sqrt{-\ln(-\tilde{\Phi}^*)} \tag{2.4}$$

$$\bar{x} := \frac{9.4883409779 - h \cdot (9.6320903635 - h \cdot (0.58556997323 + 2.1464093351 \cdot h))}{1 - h \cdot (0.65174820867 + h \cdot (1.5120247828 + 6.6437847132\text{E}{-5} \cdot h))} \tag{2.5}$$

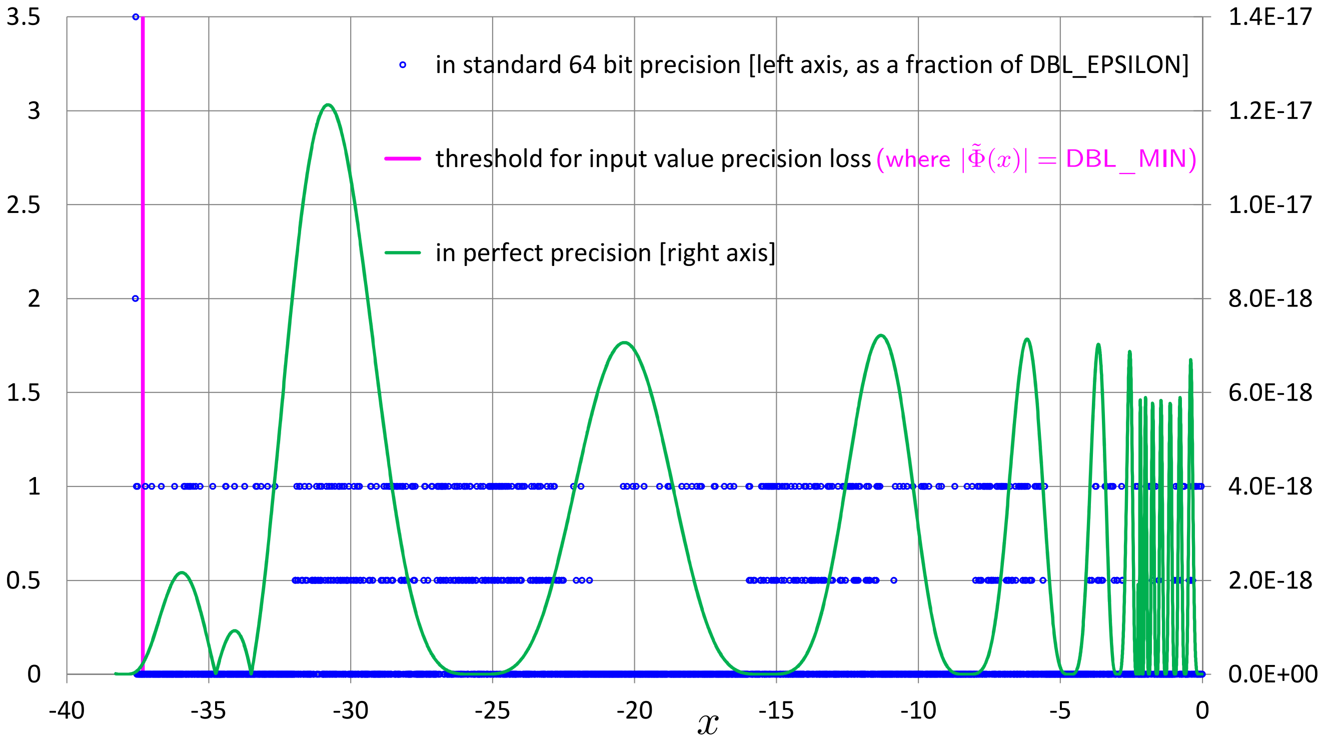
Then,

$$x^* := \bar{x} + \frac{3q\bar{x}^2 \cdot (2 - q\bar{x} \cdot (2 + \bar{x}^2))}{6 + q\bar{x} \cdot (-12 + \bar{x} \cdot (6q + \bar{x} \cdot (-6 + q\bar{x} \cdot (3 + \bar{x}^2))))} \tag{2.6}$$

with

$$q := \frac{\tilde{\Phi}(\bar{x}) - \tilde{\Phi}^*}{\varphi(\bar{x})} . \tag{2.7}$$

Those formulae for $\tilde{\Phi}^{-1}(\cdot)$ give a net relative accuracy better than 1.3E−17.



[Akh29] N.I. Akhiezer. 'On a problem of E. I. Zolotarev'. In: *Izv. Akad. Nauk, Otd. Fiz. Mat. Nauk* (1929). (in Russian), pp. 919–931.

[AS84] M. Abramowitz and I.A. Stegun. *Pocketbook of Mathematical Functions*. ISBN 3-87144-818-4. Harri Deutsch, 1984.

[Boo] Boost. *The Boost C++ library: the Remez Method*. www.boost.org/doc/libs/1_70_0/libs/math/doc/html/math_toolkit/remez.html.

[CL74] C.K. and Clenshaw K. Lord. 'Rational approximations from Chebyshev series'. In: *Studies in Numer-ical Analysis (B.K.P.Scaife, ed.)* Academic Press, London and New York, 1974, pp. 95–113.

[Enc17] Encyclopedia of Mathematics. *De la Vallée-Poussin alternation theorem*. 2017.

[Hen63] J.R. Henderson. 'Direct Chebyshev approximation'. open.library.ubc.ca/collections/ubctheses/831/items/1.0080568. MA thesis. University of British Columbia, 1963.

[Hou70] A. S. Householder. *The Numerical Treatment of a Single Nonlinear Equation*. McGraw-Hill New York, 1970.

[Jäc15] P. Jäckel. 'Let's be rational'. In: *Wilmott Magazine* (Jan. 2015). First published in November 2013 at www.jaeckel.org, pp. 40–53.

[Jäc17] P. Jäckel. 'Implied Normal Volatility'. In: *Wilmott Magazine* (Mar. 2017), pp. 52–54. *Please download the correct version from* www.jaeckel.org *since the magazine contains typesetting errors in the formulae*.

[Jäc18] P. Jäckel. *Composite option valuation with smiles*. WBS 14th Fixed Income Conference, Nice. Sept. 2018.

[Jäc19] P. Jäckel. *Composite option valuation with smiles*. Quantminds Conference, Vienna. May 2019.

[Lip09] R.J. Lipton. *The Surprising Power of Rational Functions*. https://rjlipton.wordpress.com/2009/10/07/the-surprising-power-of-rational-functions. Oct. 2009.

[Lit01] G. Litvinov. 'Approximate construction of rational approximations and the effect of error autocorrection. Applications'. In: *Russian J. Math. Phys.* 1 (Feb. 2001).

[Lit90] G. L. Litvinov. 'Approximate construction of rational approximations and an effect of error autocorrection'. In: *Mathematics and modeling, NIVTs AN SSSR, Puschino*. 1990, pp. 99–141.

[Mae60] H.J. Maehly. 'Rational approximations for transcendental functions'. In: *UNESCO, Butterworths, London*. Academic Press, London and New York, 1960, pp. 57–62.

[New64] D.J. Newman. 'Rational approximation to $|x|$.'. In: *Michigan Math. J.* 11.1 (Mar. 1964), pp. 11–14. URL: https://doi.org/10.1307/mmj/1028999029.

[Rem34] E.Ya. Remez. 'Sur le calcul effectif des polynômes d'approximation de Tschebyscheff'. In: *Compt. Rend. Acade. Sc.* 199 (Jan. 1934), pp. 337–340.

[RW10] D. Reiswich and U. Wystup. 'A Guide to FX Options Quoting Conventions'. In: *The Journal of Derivatives* 18 (Nov. 2010), pp. 58–68. DOI: 10.3905/jod.2010.18.2.058.

[Val10] Ch.J. de la Vallée-Poussin. 'Sur les polynômes d'approximation et la répresentation approchée d'un angle'. In: *Bull. Acad. Belg.* 12 (1910), pp. 808–845.

[Wik19] Wikipedia. *Householder's method*. 2019.

[Рем57] Е.Я. Ремез. *Общие вычислительные методы Чебышевского приближения*. Изд-во АН УкрССР, 1957.