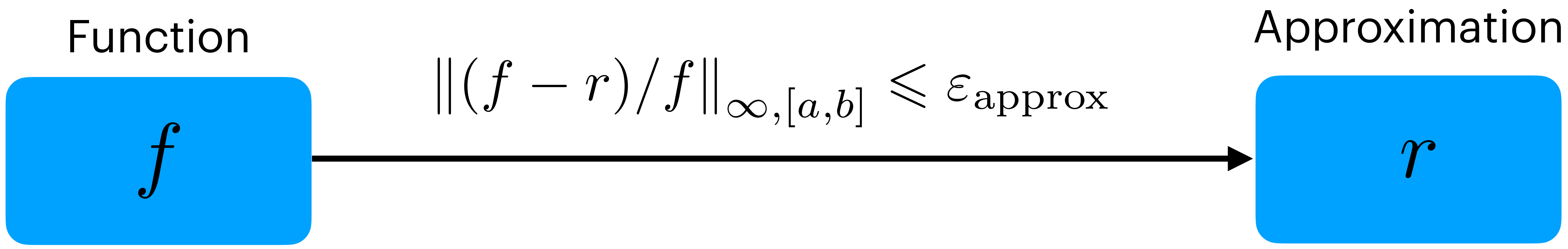
Towards Machine-Efficient Rational L^∞-Approximations of Mathematical Functions

Silviu-Ioan Filip, Univ Rennes, **Inria**, CNRS, IRISA
Joint work with **Nicolas Brisebarre**, CNRS, LIP

30th IEEE International Symposium on Computer Arithmetic
Portland, Oregon, USA. September 4-6, 2023

Building Mathematical Functions

Building Mathematical Functions



- relative error optimization on $[a, b]$:

L^{∞} norm $$\|g\|_{\infty,[a,b]} = \sup_{x \in [a,b]} |g(x)|$$

- evaluation error analyzed a posteriori

The Approximation: Polynomial vs Rational

Polynomials

$$r(x) = \sum_{i=1}^{n+1} p_i x^{i-1}$$

Rational Functions

$$r(x) = \frac{\sum_{i=1}^{m+1} p_i x^{i-1}}{\sum_{i=1}^{n+1} q_i x^{i-1}}$$

The Approximation: Polynomial vs Rational

Polynomials

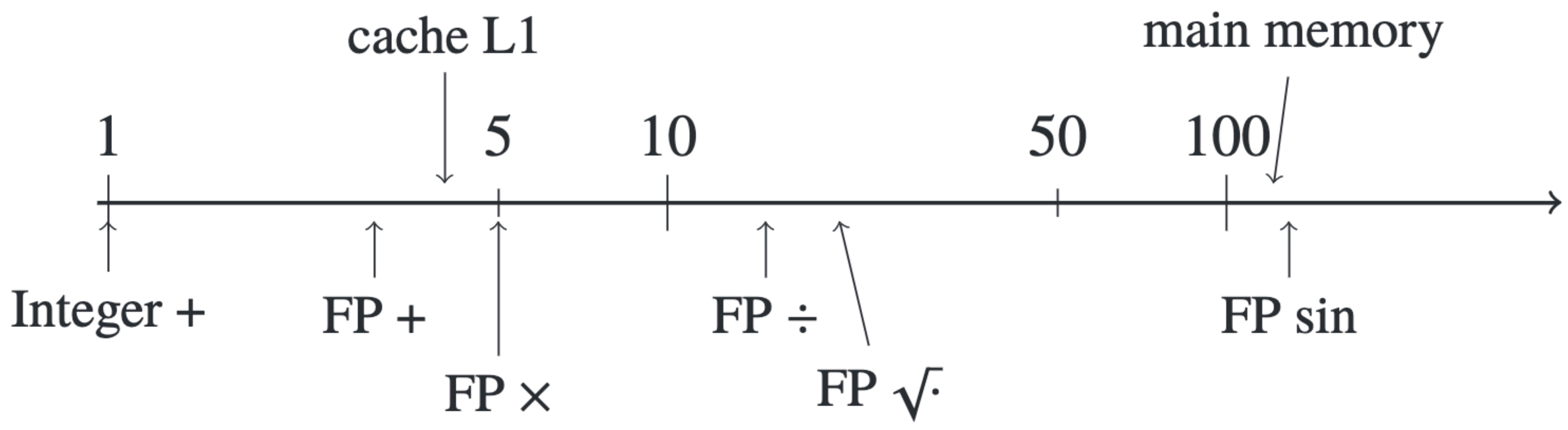
$$r(x) = \sum_{i=1}^{n+1} p_i x^{i-1}$$

☑ evaluation requires only $+$ and $\times$

Rational Functions

$$r(x) = \frac{\sum_{i=1}^{m+1} p_i x^{i-1}}{\sum_{i=1}^{n+1} q_i x^{i-1}}$$

☐ evaluation also requires $\div$



Typical current CPU latencies for FP operations in nb. of cycles (adapted from [1])

- FP division is **between three and ten times slower** than FP addition/multiplication

[1] Floating-point arithmetic, *S. Boldo and C.-P. Jeannerod and G. Melquiond and J.-M. Muller*, Acta Numerica, 32:203–290, 2023.

The Approximation: Polynomial vs Rational

Polynomials

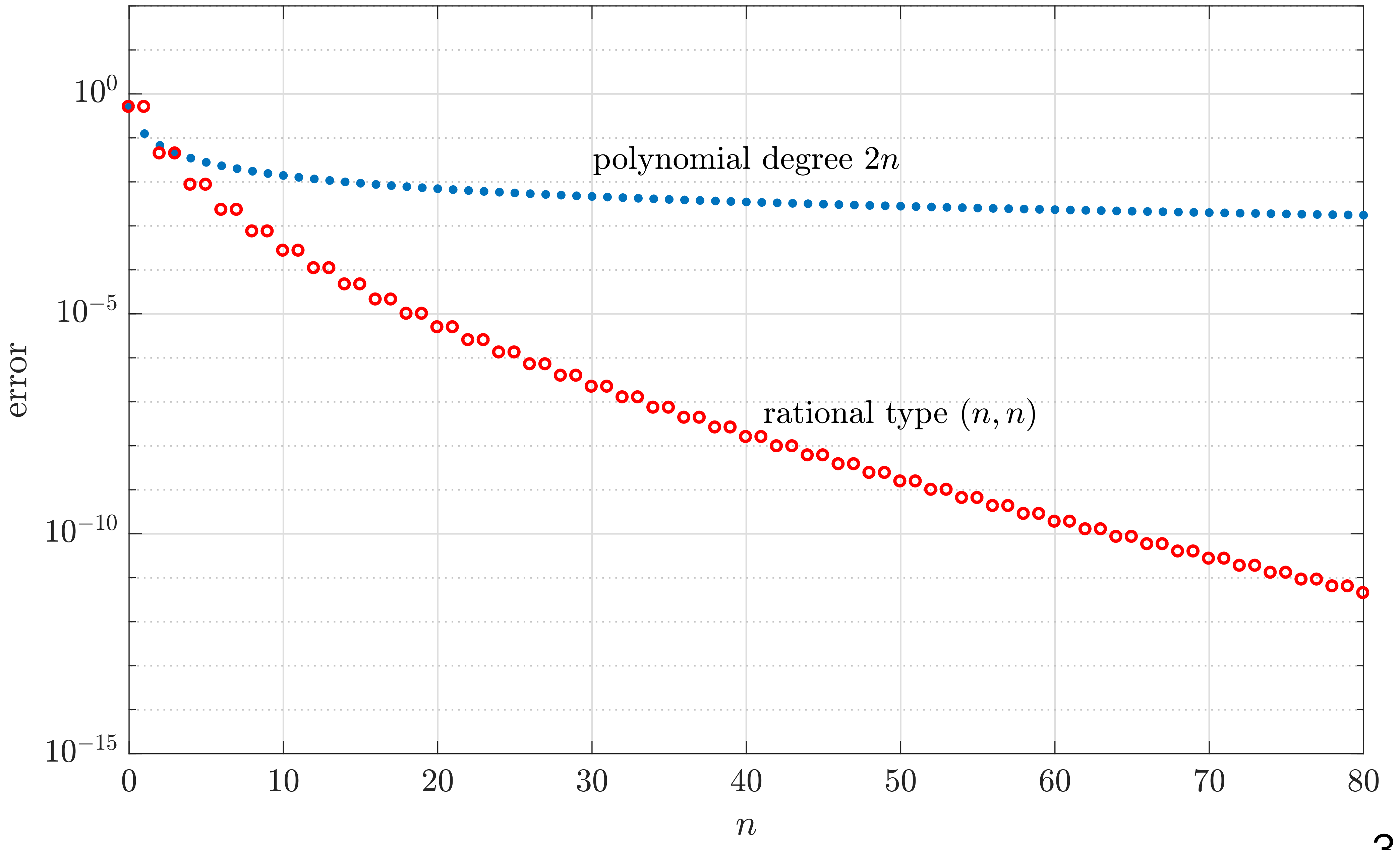
$$r(x) = \sum_{i=1}^{n+1} p_i x^{i-1}$$

- ☑ evaluation requires only $+$ and $\times$
- ☐ approximates well analytic functions

Rational Functions

$$r(x) = \frac{\sum_{i=1}^{m+1} p_i x^{i-1}}{\sum_{i=1}^{n+1} q_i x^{i-1}}$$

- ☐ evaluation also requires $\div$
- ☑ more general and powerful (e.g. near singularities)



A classic theoretical example: $f(x) = |x|, x \in [-1, 1]$

- asymptotic behavior:

$$E_{n,0}(f) \sim \beta/n, \qquad \beta = 0.2801\ldots \qquad [1]$$

$$E_{n,n}(f) \sim 8e^{-\pi\sqrt{n}}. \qquad [2]$$

A more practical `libm` example:

- special function implementations (*e.g.* the SPECFUN [3] package)

[1] On the Bernstein Conjecture in Approximation Theory, *R.S. Varga and A.J. Carpenter*, Constr. Approx., 1:333–348, 1985.

[2] Best Uniform Rational Approximation of $|x|$ on $[-1, +1]$, *H. Stahl*, Math. USSR Sbornik, 183:85–118, 1992.

[3] Algorithm 715: SPECFUN - A Portable FORTRAN Package Of Special Function Routines And Test Drivers, *W. J. Cody*, ACM TOMS, Vol. 19, No. 1, pp. 22–30, 1993.

The Approximation: Polynomial vs Rational

Polynomials

$$r(x) = \sum_{i=1}^{n+1} p_i x^{i-1}$$

- ☑ evaluation requires only $+$ and $\times$
- ☐ approximates well analytic functions
- ☑ easier to compute + powerful tools

Rational Functions

$$r(x) = \frac{\sum_{i=1}^{m+1} p_i x^{i-1}}{\sum_{i=1}^{n+1} q_i x^{i-1}}$$

- ☐ evaluation also requires $\div$
- ☑ more general and powerful (e.g. near singularities)
- ☐ harder to compute + less flexible tooling

Which one should I use?

Depends on the problem and hardware, but should **have access to powerful tools in both cases**

The Approximation Problems

Assumptions:

- $B := [a, b]$
- $\{\phi_i\}_{i=1}^{m}, \{\psi_i\}_{i=1}^{n} \subset \{1, x, x^2, \ldots\}$
- $\{p_i\}_{i=1}^{m}, \{q_i\}_{i=1}^{n}$ belong to target formats (*e.g.* float, double, double-double)

Polynomials

$$r(x) = \sum_{i=1}^{m} p_i \phi_i(x)$$

or

Rational Functions

$$r(x) = \frac{\sum_{i=1}^{m} p_i \phi_i(x)}{\sum_{i=1}^{n} q_i \psi_i(x)}$$

The Approximation Problems: $P_{\mathbb{F}}[B]$

Assumptions:

- $B := [a, b]$
- $\{\phi_i\}_{i=1}^{m}, \{\psi_i\}_{i=1}^{n} \subset \{1, x, x^2, \ldots\}$
- $\{p_i\}_{i=1}^{m}, \{q_i\}_{i=1}^{n}$ belong to target formats (*e.g.* float, double, double-double)

The problems:

$$P_{\mathbb{F}}[B] : \text{minimize} \left\{ \left\| \frac{f - r}{f} \right\|_{\infty, B} \;\middle|\; \begin{array}{l} \{p_i\}_{i=1}^{m}, \{q_i\}_{i=1}^{n} \text{ belong to} \\ \textit{target} \text{ floating-point formats} \end{array} \right\}$$

- semi-infinite mixed integer programming instance
  ➡ ***hard*** combinatorial problem (*e.g.* in polynomial setting [1])

Polynomials

$$r(x) = \sum_{i=1}^{m} p_i \phi_i(x)$$

or

Rational Functions

$$r(x) = \frac{\sum_{i=1}^{m} p_i \phi_i(x)}{\sum_{i=1}^{n} q_i \psi_i(x)}$$

[1] Computing Machine-Efficient Polynomial Approximations, *N. Brisebarre and J.-M. Muller and A. Tisserand*, ACM TOMS, Vol. 32, No. 2, pp. 236-256, 2006.

The Approximation Problems: $P_{\mathbb{R}}[B]$

Assumptions:

- $B := [a, b]$
- $\{\phi_i\}_{i=1}^{m}, \{\psi_i\}_{i=1}^{n} \subset \{1, x, x^2, \ldots\}$
- $\{p_i\}_{i=1}^{m}, \{q_i\}_{i=1}^{n}$ belong to target formats (*e.g.* float, double, double-double)

The problems:

$$P_{\mathbb{R}}[B] : \text{minimize} \left\{ \left\| \frac{f - r}{f} \right\|_{\infty, B} \middle| \begin{array}{l} \{p_i\}_{i=1}^{m}, \{q_i\}_{i=1}^{n} \\ \text{take values from } \mathbb{R} \end{array} \right\}$$

- real-coefficient relaxation with well-developed theory and algorithms
- in practice: multiple precision arithmetic

Polynomials

$$r(x) = \sum_{i=1}^{m} p_i \phi_i(x)$$

or

Rational Functions

$$r(x) = \frac{\sum_{i=1}^{m} p_i \phi_i(x)}{\sum_{i=1}^{n} q_i \psi_i(x)}$$

[1] Computing Machine-Efficient Polynomial Approximations, *N. Brisebarre and J.-M. Muller and A. Tisserand*, ACM TOMS, Vol. 32, No. 2, pp. 236-256, 2006.

The Approximation Problems: $P_{\mathbb{R}}[B]$

Assumptions:

- $B := [a, b]$
- $\{\phi_i\}_{i=1}^{m}, \{\psi_i\}_{i=1}^{n} \subset \{1, x, x^2, \dots\}$
- $\{p_i\}_{i=1}^{m}, \{q_i\}_{i=1}^{n}$ belong to target formats (*e.g.* float, double, double-double)

The problems:

$$P_{\mathbb{R}}[B] : \text{minimize} \left\{ \left\| \frac{f - r}{f} \right\|_{\infty, B} \middle| \begin{array}{l} \{p_i\}_{i=1}^{m}, \{q_i\}_{i=1}^{n} \\ \text{take values from } \mathbb{R} \end{array} \right\}$$

- real-coefficient relaxation with well-developed theory and algorithms
- in practice: multiple precision arithmetic
  - ➡ polynomial case (Sollya):
    - the `remez` command [2] & the `fpminimax` command [3]

$P_{\mathbb{R}}[B]$ solution $P_{\mathbb{F}}[B]$ heuristic

Polynomials

$$r(x) = \sum_{i=1}^{m} p_i \phi_i(x)$$

or

Rational Functions

$$r(x) = \frac{\sum_{i=1}^{m} p_i \phi_i(x)}{\sum_{i=1}^{n} q_i \psi_i(x)}$$

[1] Computing Machine-Efficient Polynomial Approximations, *N. Brisebarre and J.-M. Muller and A. Tisserand*, ACM TOMS, Vol. 32, No. 2, pp. 236-256, 2006.
[2] Sollya software tool: https://www.sollya.org/
[3] Efficient Polynomial L^{∞}-Approximations, *N. Brisebarre and S. Chevillard,* ARITH-18, 2008.

The Approximation Problems: $P_{\mathbb{R}}[B]$

Assumptions:

- $B := [a, b]$
- $\{\phi_i\}_{i=1}^{m}, \{\psi_i\}_{i=1}^{n} \subset \{1, x, x^2, \ldots\}$
- $\{p_i\}_{i=1}^{m}, \{q_i\}_{i=1}^{n}$ belong to target formats (*e.g.* float, double, double-double)

Polynomials

$$r(x) = \sum_{i=1}^{m} p_i \phi_i(x)$$

or

Rational Functions

$$r(x) = \frac{\sum_{i=1}^{m} p_i \phi_i(x)}{\sum_{i=1}^{n} q_i \psi_i(x)}$$

The problems:

$$P_{\mathbb{R}}[B] : \text{minimize} \left\{ \left\| \frac{f - r}{f} \right\|_{\infty, B} \middle| \begin{array}{l} \{p_i\}_{i=1}^{m}, \{q_i\}_{i=1}^{n} \\ \text{take values from } \mathbb{R} \end{array} \right\}$$

- real-coefficient relaxation with well-developed theory and algorithms
- in practice: multiple precision arithmetic
  - ➡ **polynomial** case (Sollya):
    - the `remez` command [2] & the `fpminimax` command [3]

$P_{\mathbb{R}}[B]$ solution $\qquad$ $P_{\mathbb{F}}[B]$ heuristic

Our goal:

Design similarly flexible alternatives to `remez` and `fpminimax` in the rational setting

[1] Computing Machine-Efficient Polynomial Approximations, *N. Brisebarre and J.-M. Muller and A. Tisserand*, ACM TOMS, Vol. 32, No. 2, pp. 236-256, 2006.
[2] Sollya software tool: https://www.sollya.org/
[3] Efficient Polynomial L^{∞}-Approximations, *N. Brisebarre and S. Chevillard,* ARITH-18, 2008.

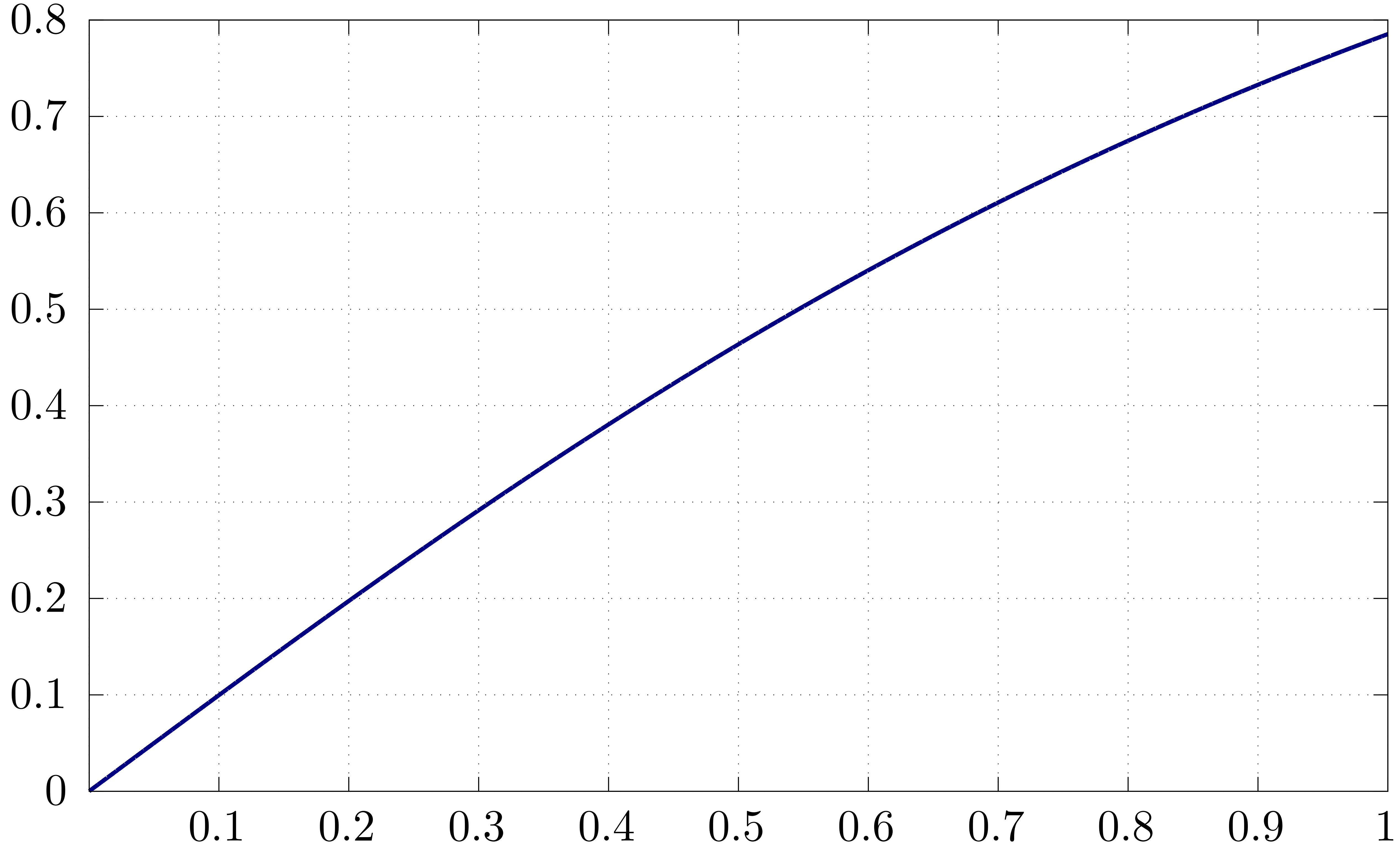
A Motivating Example: $\arctan$

- CORE-MATH [1] implementation of float $\arctan$

$$f(x) = \arctan(x), x \in B = [0.000127, 1]$$

$$r(x) := \frac{\sum_{i=1}^{7} p_i \phi_i(x)}{\sum_{i=1}^{7} q_i \psi_i(x)} = \frac{\sum_{i=1}^{7} p_i x^{2i-1}}{\sum_{i=1}^{7} q_i x^{2i-2}}$$

Goal: $P_{\mathbb{F}}[B]$ with double prec. coefficients



[1] The CORE-MATH Project, *A. Sibidanov and P. Zimmermann and S. Glondu,* ARITH-29, 2022.

A Motivating Example: $\arctan$

- CORE-MATH [1] implementation of float $\arctan$

$$f(x) = \arctan(x), x \in B = [0.000127, 1]$$

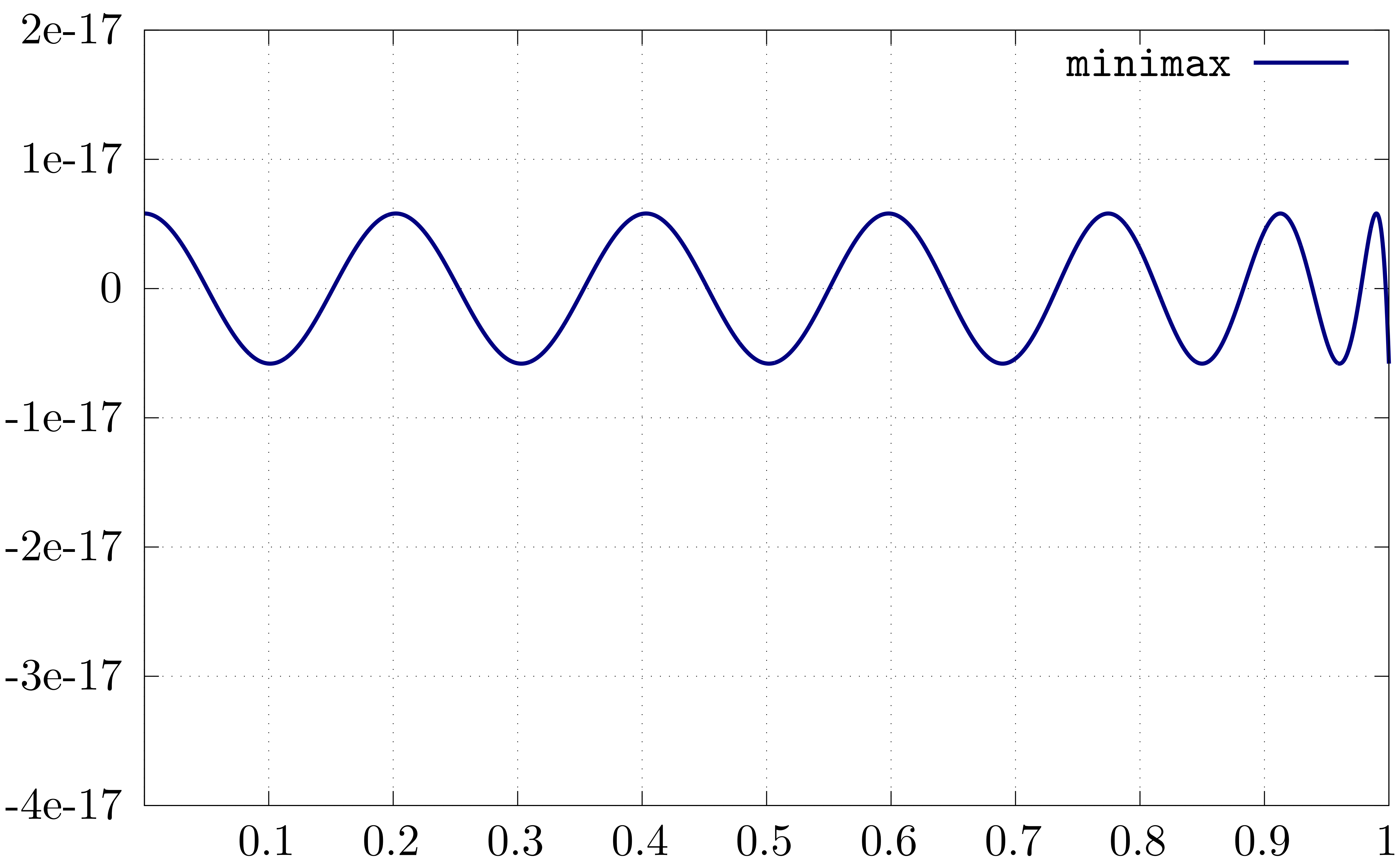
$$r(x) := \frac{\sum_{i=1}^{7} p_i \phi_i(x)}{\sum_{i=1}^{7} q_i \psi_i(x)} = \frac{\sum_{i=1}^{7} p_i x^{2i-1}}{\sum_{i=1}^{7} q_i x^{2i-2}}$$

Goal: $P_{\mathbb{F}}[B]$ with double prec. coefficients

- use our new `minimax` command to solve $P_{\mathbb{R}}[B]$

$$\|\varepsilon\|_{\infty,B} \approx 2^{-57.26}$$

$$\varepsilon(x) = (f(x) - r(x))/f(x)$$



What about a polynomial?

- at least a degree 20 $(m = 21, n = 1)$ approximation
- possible tradeoff: six additions & six multiplications for one division

[1] The CORE-MATH Project, *A. Sibidanov and P. Zimmermann and S. Glondu,* ARITH-29, 2022.

A Motivating Example: $\arctan$

- CORE-MATH [1] implementation of float $\arctan$

$$f(x) = \arctan(x), x \in B = [0.000127, 1]$$

$$r(x) := \frac{\sum_{i=1}^{7} p_i \phi_i(x)}{\sum_{i=1}^{7} q_i \psi_i(x)} = \frac{\sum_{i=1}^{7} p_i x^{2i-1}}{\sum_{i=1}^{7} q_i x^{2i-2}}$$

Goal: $P_{\mathbb{F}}[B]$ with double prec. coefficients

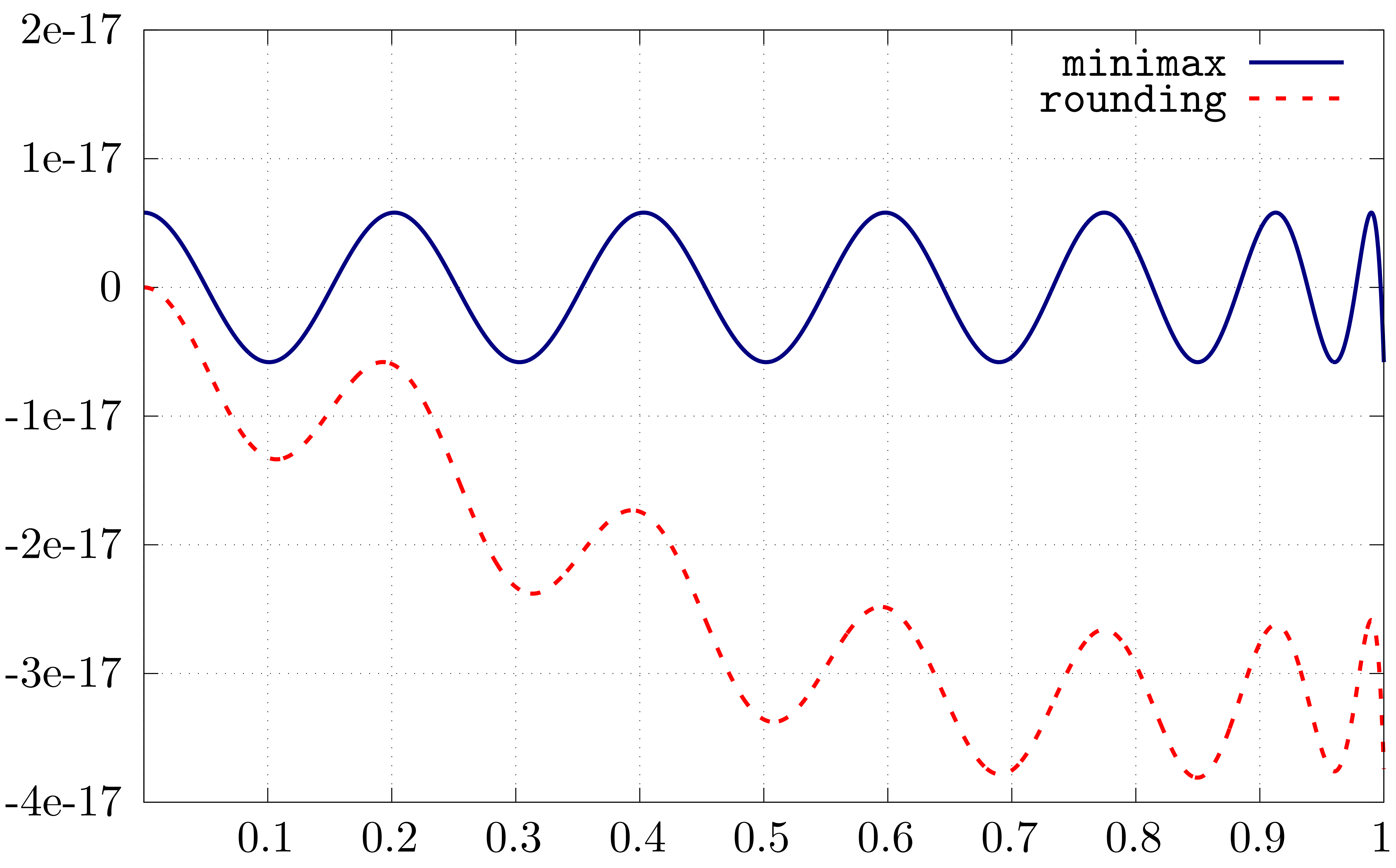
- use our new `minimax` command to solve $P_{\mathbb{R}}[B]$

$$\|\varepsilon\|_{\infty,B} \approx 2^{-57.26}$$

What happens if we round coeffs. to double prec. ?

$$\|\varepsilon\|_{\infty,B} \approx 2^{-54.54}$$

$\varepsilon(x) = (f(x) - r(x))/f(x)$



What about a polynomial?

- at least a degree 20 $(m = 21, n = 1)$ approximation
- possible tradeoff: six additions & six multiplications for one division

[1] The CORE-MATH Project, *A. Sibidanov and P. Zimmermann and S. Glondu,* ARITH-29, 2022.

A Motivating Example: $\arctan$

- CORE-MATH [1] implementation of float $\arctan$

$$f(x) = \arctan(x), x \in B = [0.000127, 1]$$

$$r(x) := \frac{\sum_{i=1}^{7} p_i \phi_i(x)}{\sum_{i=1}^{7} q_i \psi_i(x)} = \frac{\sum_{i=1}^{7} p_i x^{2i-1}}{\sum_{i=1}^{7} q_i x^{2i-2}}$$

Goal: $P_{\mathbb{F}}[B]$ with double prec. coefficients

- use our new `minimax` command to solve $P_{\mathbb{R}}[B]$

$$\|\varepsilon\|_{\infty,B} \approx 2^{-57.26}$$

What happens if we round coeffs. to double prec. ?

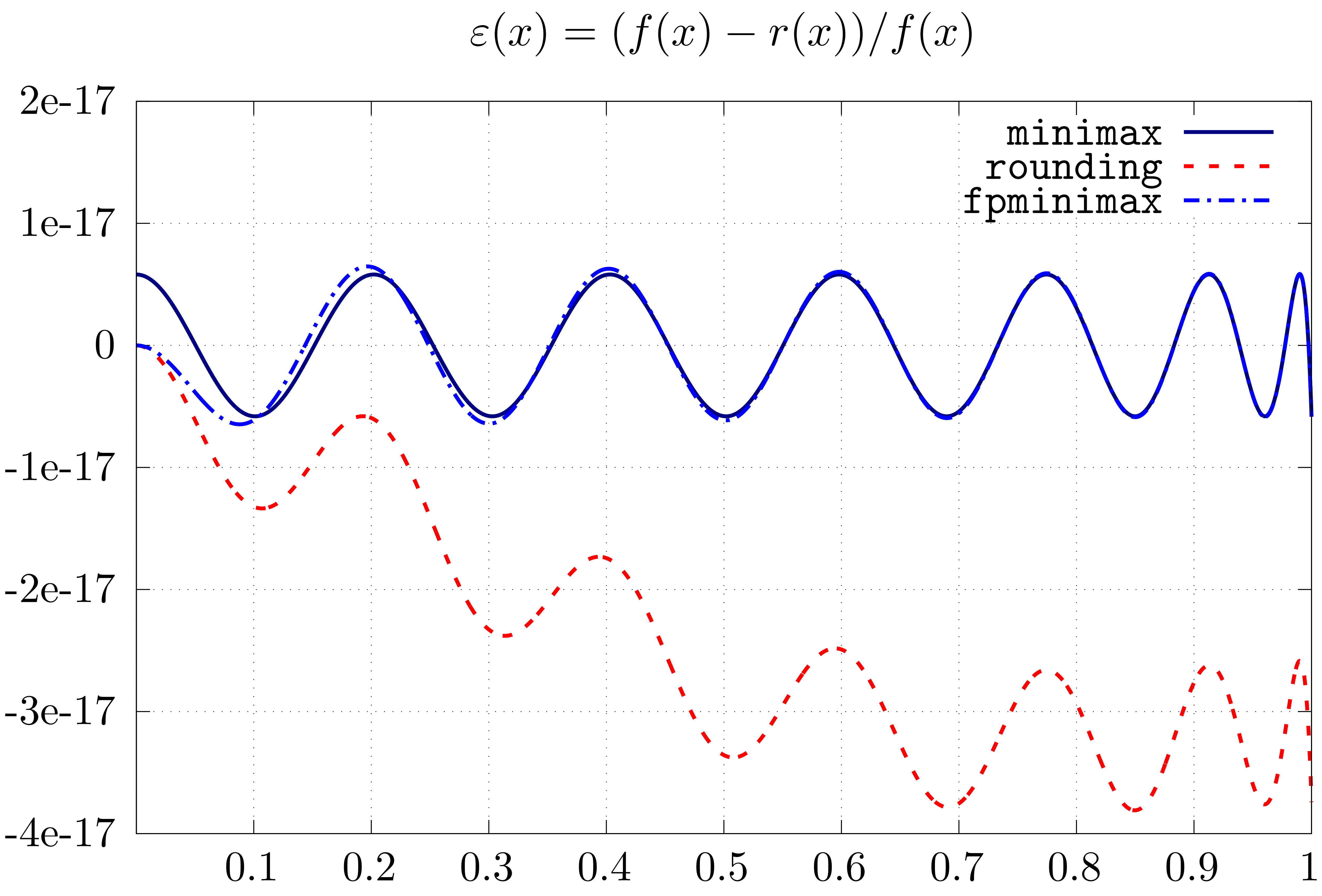
$$\|\varepsilon\|_{\infty,B} \approx 2^{-54.54}$$

- use our new `fpminimax` command to address $P_{\mathbb{F}}[B]$

$$\|\varepsilon\|_{\infty,B} \approx 2^{-57.09}$$

What about a polynomial?

- at least a degree 20 $(m = 21, n = 1)$ approximation
- possible tradeoff: six additions & six multiplications for one division



[1] The CORE-MATH Project, *A. Sibidanov and P. Zimmermann and S. Glondu,* ARITH-29, 2022.

Finding Solutions to $P_{\mathbb{R}}[B]$

$$P_{\mathbb{R}}[B] : \text{minimize} \left\{ \left\| \frac{f-r}{f} \right\|_{\infty,B} \middle| \begin{array}{c} \{p_i\}_{i=1}^{m}, \{q_i\}_{i=1}^{n} \\ \text{take values from } \mathbb{R} \end{array} \right\}$$

- find approximations with restricted denominators

$$\mathcal{R}_L(B) = \left\{ \frac{P}{Q} := \frac{p_1\phi_1 + \ldots + p_m\phi_m}{q_1\psi_1 + \ldots + q_n\psi_n} \middle| \begin{array}{c} Q \geqslant L > 0 \text{ on } B, \\ \max\limits_{1 \leqslant i \leqslant n} |q_i| = 1 \end{array} \right\}$$

Finding Solutions to $P_{\mathbb{R}}[B]$

$$P_{\mathbb{R}}[B] : \text{minimize} \left\{ \left\| \frac{f-r}{f} \right\|_{\infty,B} \middle| \begin{array}{l} \{p_i\}_{i=1}^m, \{q_i\}_{i=1}^n \\ \text{take values from } \mathbb{R} \end{array} \right\}$$

- find approximations with restricted denominators

$$\mathcal{R}_L(B) = \left\{ \frac{P}{Q} := \frac{p_1\phi_1 + \ldots + p_m\phi_m}{q_1\psi_1 + \ldots + q_n\psi_n} \middle| \begin{array}{l} Q \geqslant L > 0 \text{ on } B, \\ \max\limits_{1\leqslant i\leqslant n} |q_i| = 1 \end{array} \right\}$$

Why?

- solutions to $P_{\mathbb{R}}[B]$ always exist [1]
  - ➡ not necessarily true otherwise
- need normalizing condition: $\max\limits_{1\leqslant i\leqslant n} |q_i| = 1$
- limits dynamic range in denominator
- not such a strong constraint (by default, $L(x) = 10^{-20}$)

Desiderata:

- for *flexibility*, allow user specified bases $\{\phi_i\}_{i=1}^m, \{\psi_i\}_{i=1}^n$

[1] Uniform Approximation by Rational Functions Having Restricted Denominators, *E.H. Kaufman Jr. and G.D. Taylor*, J. Approx. Theory, Vol. 32, No. 1, pp. 9-26, 1981.

Finding Solutions to $P_{\mathbb{R}}[B]$

- a family of Generalized First Remez Algorithms [1]

Notation:

- minimal error $\mu(D) = \min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}$, $D \subseteq B$
- best approximation $r_D = \arg\min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}$

Step 1. $k \leftarrow 0$ and $D_0 \subseteq B$ finite set with $|D_0| \geqslant m + n$

Example:

$$f(x) = \exp(x + \sqrt{x})$$
$$B = [0.01, 1]$$

$$r(x) = \frac{p_1 + p_2 \exp(x) + p_3 \sin(x)}{q_1 + q_2 x + q_3 \cos(2x)}$$

[1] Modifications of the First Remez Algorithm, *R. Reemtsen*, SIAM Journal of Numerical Analysis, Vol. 27, No. 2, pp. 507–518, 1990.

Finding Solutions to $P_{\mathbb{R}}[B]$

- a family of **Generalized First Remez Algorithms** [1]

Notation:

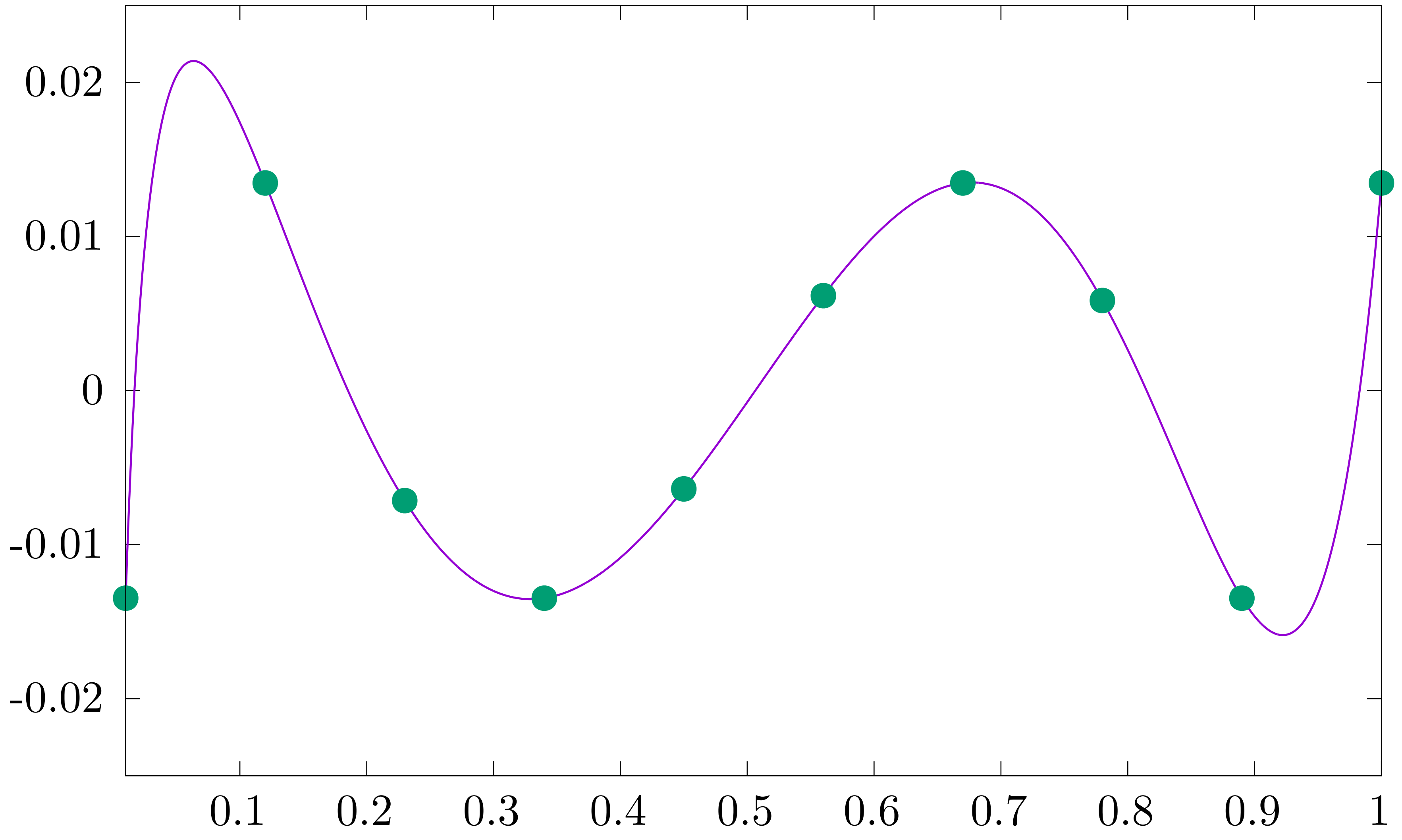
- minimal error $\mu(D) = \min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}$, $D \subseteq B$
- best approximation $r_D = \arg\min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}$

Step 1. $k \leftarrow 0$ and $D_0 \subseteq B$ finite set with $|D_0| \geqslant m + n$

Example:

$$f(x) = \exp(x + \sqrt{x})$$
$$B = [0.01, 1]$$

$$r(x) = \frac{p_1 + p_2 \exp(x) + p_3 \sin(x)}{q_1 + q_2 x + q_3 \cos(2x)}$$



Step 2. Find $\mu(D_k)$ and solution r_{D_k} to $P_{\mathbb{R}}[D_k]$

$e_k := f - r_{D_k}$

[1] Modifications of the First Remez Algorithm, *R. Reemtsen*, SIAM Journal of Numerical Analysis, Vol. 27, No. 2, pp. 507–518, 1990.

Finding Solutions to $P_{\mathbb{R}}[B]$

- a family of **Generalized First Remez Algorithms** [1]

Notation:

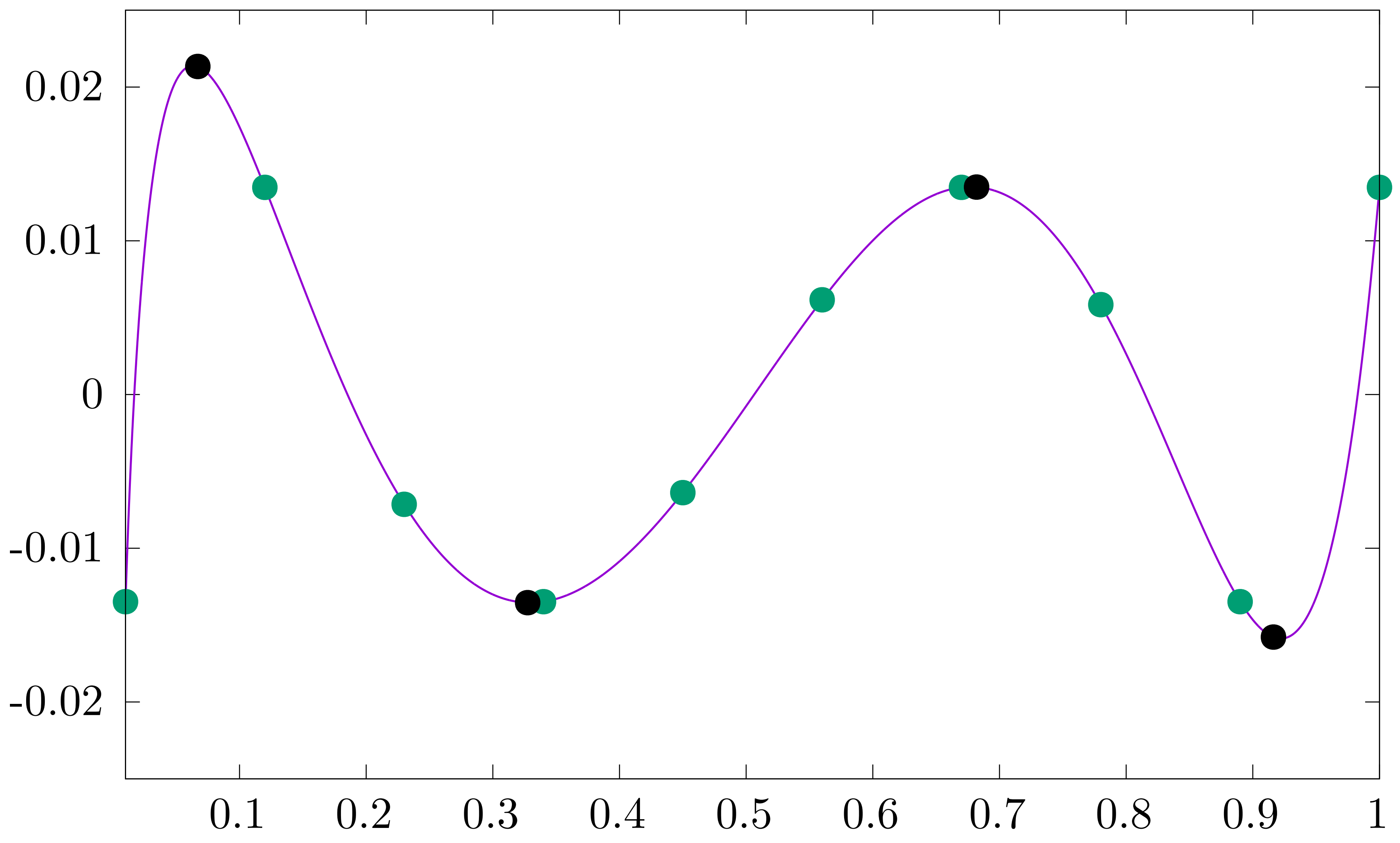
- minimal error $\mu(D) = \min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}\,, D \subseteq B$
- best approximation $r_D = \arg\min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}$

Step 1. $k \leftarrow 0$ and $D_0 \subseteq B$ finite set with $|D_0| \geqslant m + n$

Example:

$$f(x) = \exp(x + \sqrt{x})$$
$$B = [0.01, 1]$$

$$r(x) = \frac{p_1 + p_2 \exp(x) + p_3 \sin(x)}{q_1 + q_2 x + q_3 \cos(2x)}$$



Step 2. Find $\mu(D_k)$ and solution r_{D_k} to $P_{\mathbb{R}}[D_k]$

$$e_k := f - r_{D_k}$$

Step 3. Compute $E_k = \left\{ x \in B \,\middle|\, \begin{array}{l} x \text{ local extrema of } e_k \text{ over } B \\ \text{with } |e_k(x)| > \mu(D_k) \end{array} \right\}$

$$D_{k+1} := D_k \cup E_k$$

[1] Modifications of the First Remez Algorithm, *R. Reemtsen*, SIAM Journal of Numerical Analysis, Vol. 27, No. 2, pp. 507–518, 1990.

Finding Solutions to $P_{\mathbb{R}}[B]$

- a family of **Generalized First Remez Algorithms** [1]

Notation:

- minimal error $\mu(D) = \min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}$, $D \subseteq B$
- best approximation $r_D = \arg\min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}$

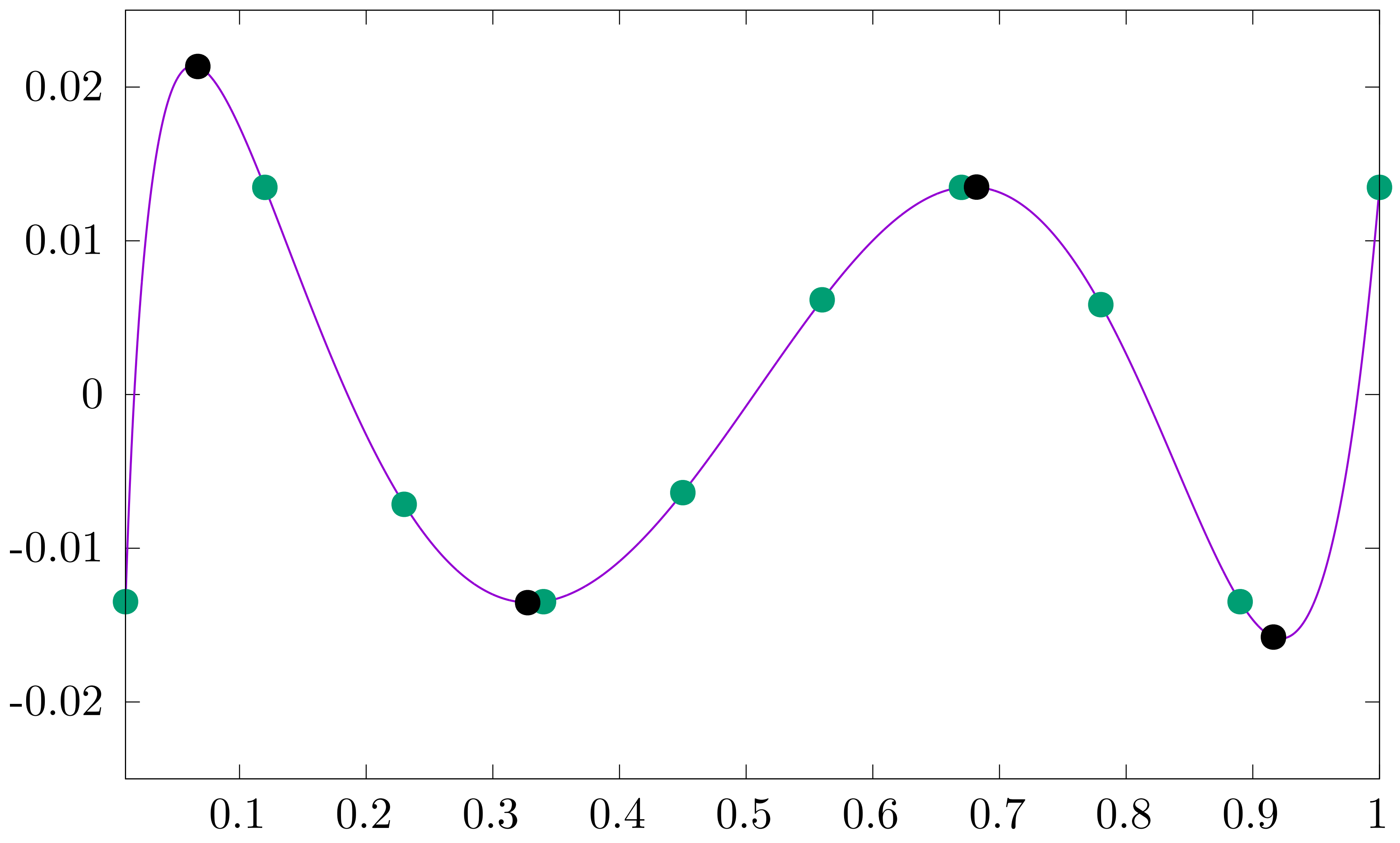
Step 1. $k \leftarrow 0$ and $D_0 \subseteq B$ finite set with $|D_0| \geqslant m + n$

Example:

$f(x) = \exp(x + \sqrt{x})$

$B = [0.01, 1]$

$$r(x) = \frac{p_1 + p_2 \exp(x) + p_3 \sin(x)}{q_1 + q_2 x + q_3 \cos(2x)}$$



Step 2. Find $\mu(D_k)$ and solution r_{D_k} to $P_{\mathbb{R}}[D_k]$

$$e_k := f - r_{D_k}$$

Step 3. Compute $E_k = \left\{ x \in B \,\middle|\, \begin{array}{l} x \text{ local extrema of } e_k \text{ over } B \\ \text{with } |e_k(x)| > \mu(D_k) \end{array} \right\}$

$$D_{k+1} := D_k \cup E_k$$

Step 4. $k \leftarrow k + 1$

[1] Modifications of the First Remez Algorithm, *R. Reemtsen*, SIAM Journal of Numerical Analysis, Vol. 27, No. 2, pp. 507–518, 1990.

Finding Solutions to $P_{\mathbb{R}}[B]$

- a family of **Generalized First Remez Algorithms** [1]

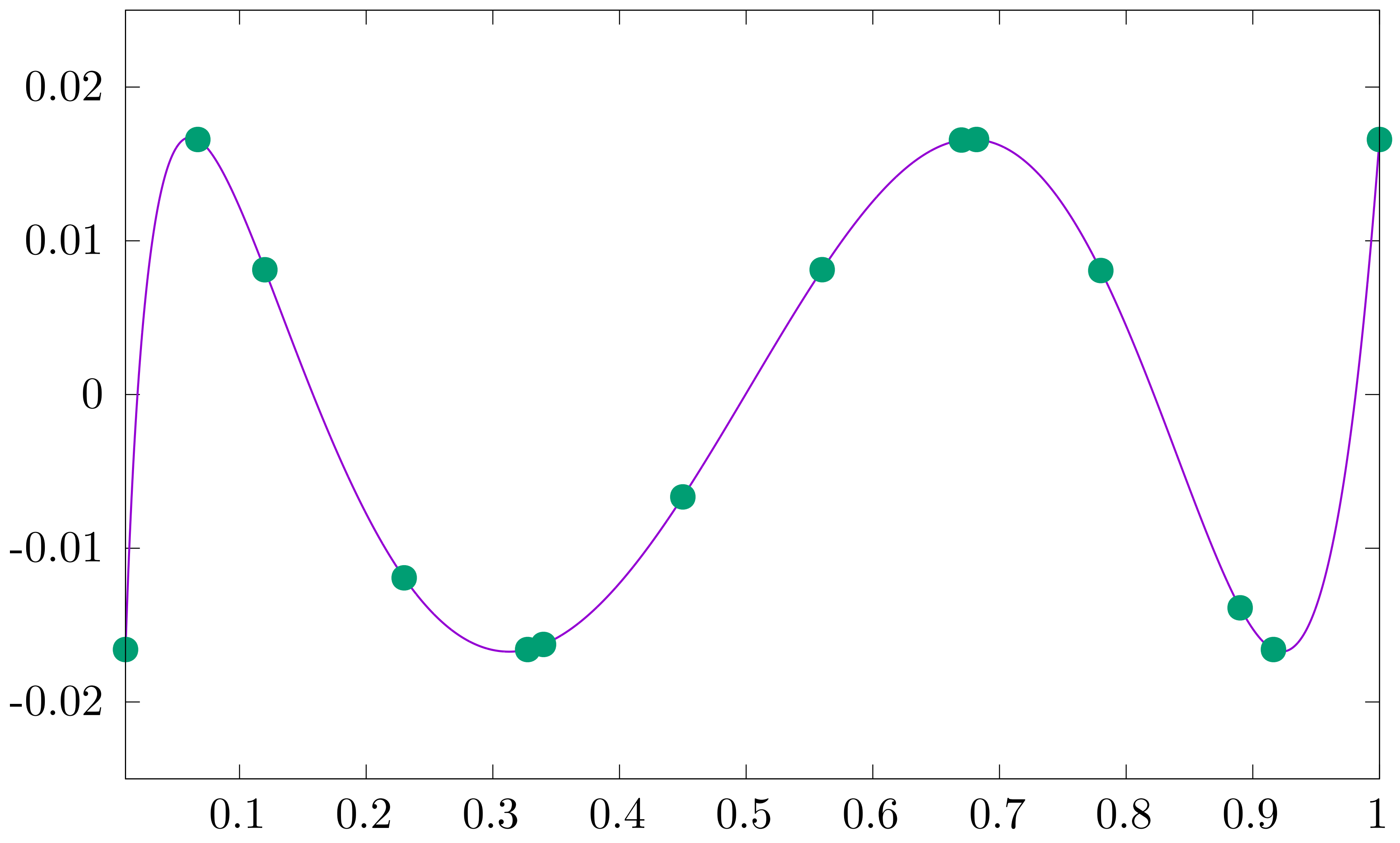
Notation:

- minimal error $\mu(D) = \min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}$, $D \subseteq B$
- best approximation $r_D = \arg\min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}$
- a maximum error point $x_k^* \in \arg\max_{x \in E_k} |e_k(x)|$

Example:

$$f(x) = \exp(x + \sqrt{x})$$
$$B = [0.01, 1]$$

$$r(x) = \frac{p_1 + p_2 \exp(x) + p_3 \sin(x)}{q_1 + q_2 x + q_3 \cos(2x)}$$



Step 1. $k \leftarrow 0$ and $D_0 \subseteq B$ finite set with $|D_0| \geqslant m + n$

do

Step 2. Find $\mu(D_k)$ and solution r_{D_k} to $P_{\mathbb{R}}[D_k]$

$e_k := f - r_{D_k}$

Step 3. Compute $E_k = \left\{ x \in B \,\middle|\, \begin{array}{l} x \text{ local extrema of } e_k \text{ over } B \\ \text{with } |e_k(x)| > \mu(D_k) \end{array} \right\}$

$D_{k+1} := D_k \cup E_k$

Step 4. $k \leftarrow k + 1$

while $\left(|e_{k-1}(x_{k-1}^*)| - \mu(D_{k-1})\right) / |e_{k-1}(x_{k-1}^*)| > 10^{-4}$

[1] Modifications of the First Remez Algorithm, *R. Reemtsen*, SIAM Journal of Numerical Analysis, Vol. 27, No. 2, pp. 507–518, 1990.

Finding Solutions to $P_{\mathbb{R}}[B]$

- a family of **Generalized First Remez Algorithms** [1]

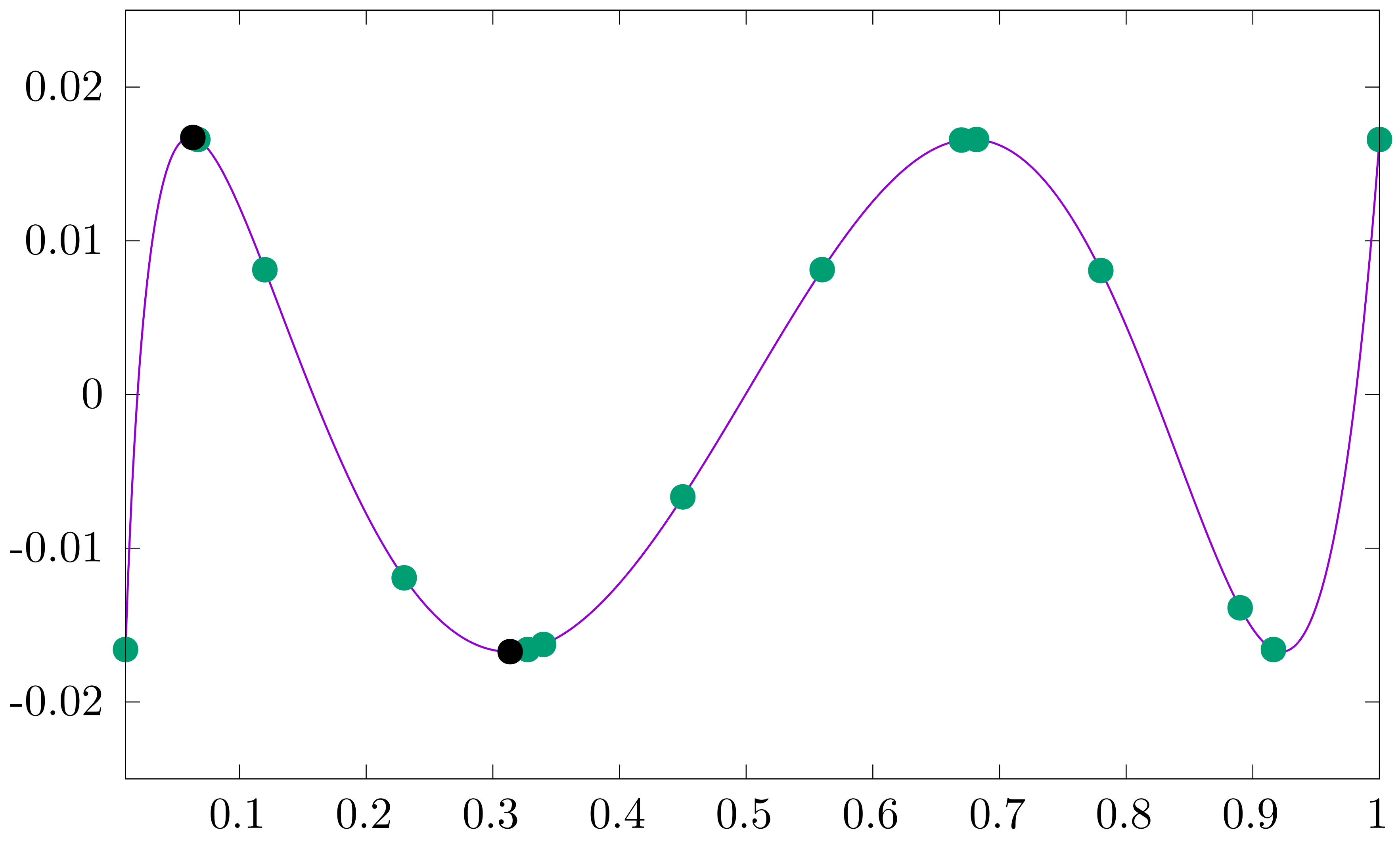
Notation:

- minimal error $\mu(D) = \min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty, D}$, $D \subseteq B$
- best approximation $r_D = \arg\min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty, D}$
- a maximum error point $x_k^* \in \arg\max_{x \in E_k} |e_k(x)|$

Example:

$$f(x) = \exp(x + \sqrt{x})$$
$$B = [0.01, 1]$$

$$r(x) = \frac{p_1 + p_2 \exp(x) + p_3 \sin(x)}{q_1 + q_2 x + q_3 \cos(2x)}$$



Step 1. $k \leftarrow 0$ and $D_0 \subseteq B$ finite set with $|D_0| \geqslant m + n$

do

Step 2. Find $\mu(D_k)$ and solution r_{D_k} to $P_{\mathbb{R}}[D_k]$

$e_k := f - r_{D_k}$

Step 3. Compute $E_k = \left\{ x \in B \,\middle|\, \begin{array}{l} x \text{ local extrema of } e_k \text{ over } B \\ \text{with } |e_k(x)| > \mu(D_k) \end{array} \right\}$

$D_{k+1} := D_k \cup E_k$

Step 4. $k \leftarrow k + 1$

while $\left(|e_{k-1}(x_{k-1}^*)| - \mu(D_{k-1})\right) / |e_{k-1}(x_{k-1}^*)| > 10^{-4}$

[1] Modifications of the First Remez Algorithm, *R. Reemtsen*, SIAM Journal of Numerical Analysis, Vol. 27, No. 2, pp. 507–518, 1990.

Finding Solutions to $P_{\mathbb{R}}[B]$

- a family of **Generalized First Remez Algorithms** [1]

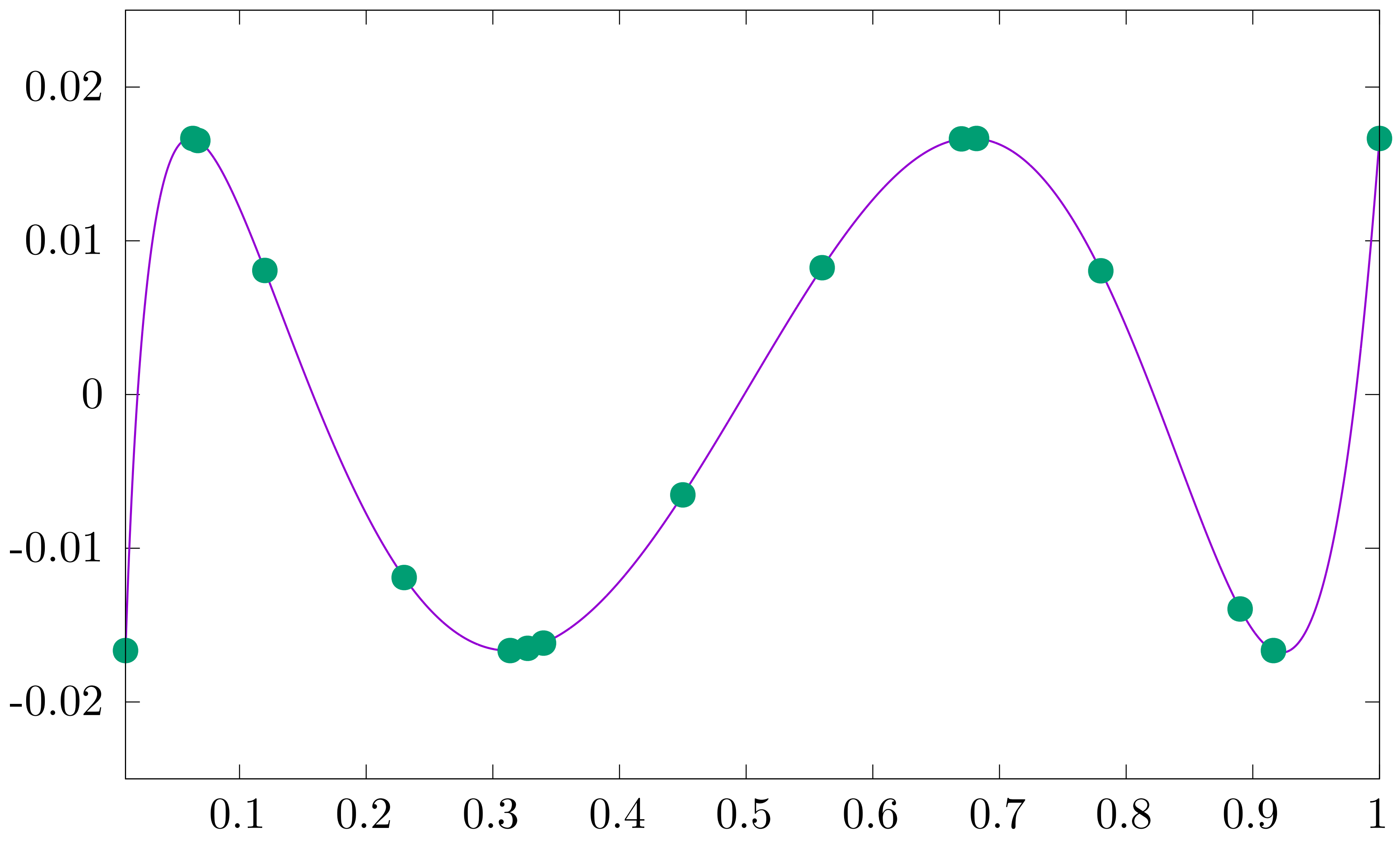
Notation:

- minimal error $\mu(D) = \min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty, D}$, $D \subseteq B$
- best approximation $r_D = \arg\min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty, D}$
- a maximum error point $x_k^* \in \arg\max_{x \in E_k} |e_k(x)|$

Example:

$$f(x) = \exp(x + \sqrt{x})$$
$$B = [0.01, 1]$$

$$r(x) = \frac{p_1 + p_2 \exp(x) + p_3 \sin(x)}{q_1 + q_2 x + q_3 \cos(2x)}$$



Step 1. $k \leftarrow 0$ and $D_0 \subseteq B$ finite set with $|D_0| \geqslant m + n$

do

Step 2. Find $\mu(D_k)$ and solution r_{D_k} to $P_{\mathbb{R}}[D_k]$

$$e_k := f - r_{D_k}$$

Step 3. Compute $E_k = \left\{ x \in B \,\middle|\, \begin{array}{l} x \text{ local extrema of } e_k \text{ over } B \\ \text{with } |e_k(x)| > \mu(D_k) \end{array} \right\}$

$$D_{k+1} := D_k \cup E_k$$

Step 4. $k \leftarrow k + 1$

while $\left(|e_{k-1}(x_{k-1}^*)| - \mu(D_{k-1})\right) / |e_{k-1}(x_{k-1}^*)| > 10^{-4}$

[1] Modifications of the First Remez Algorithm, *R. Reemtsen*, SIAM Journal of Numerical Analysis, Vol. 27, No. 2, pp. 507–518, 1990.

Finding Solutions to $P_{\mathbb{R}}[B]$

- a family of **Generalized First Remez Algorithms** [1]

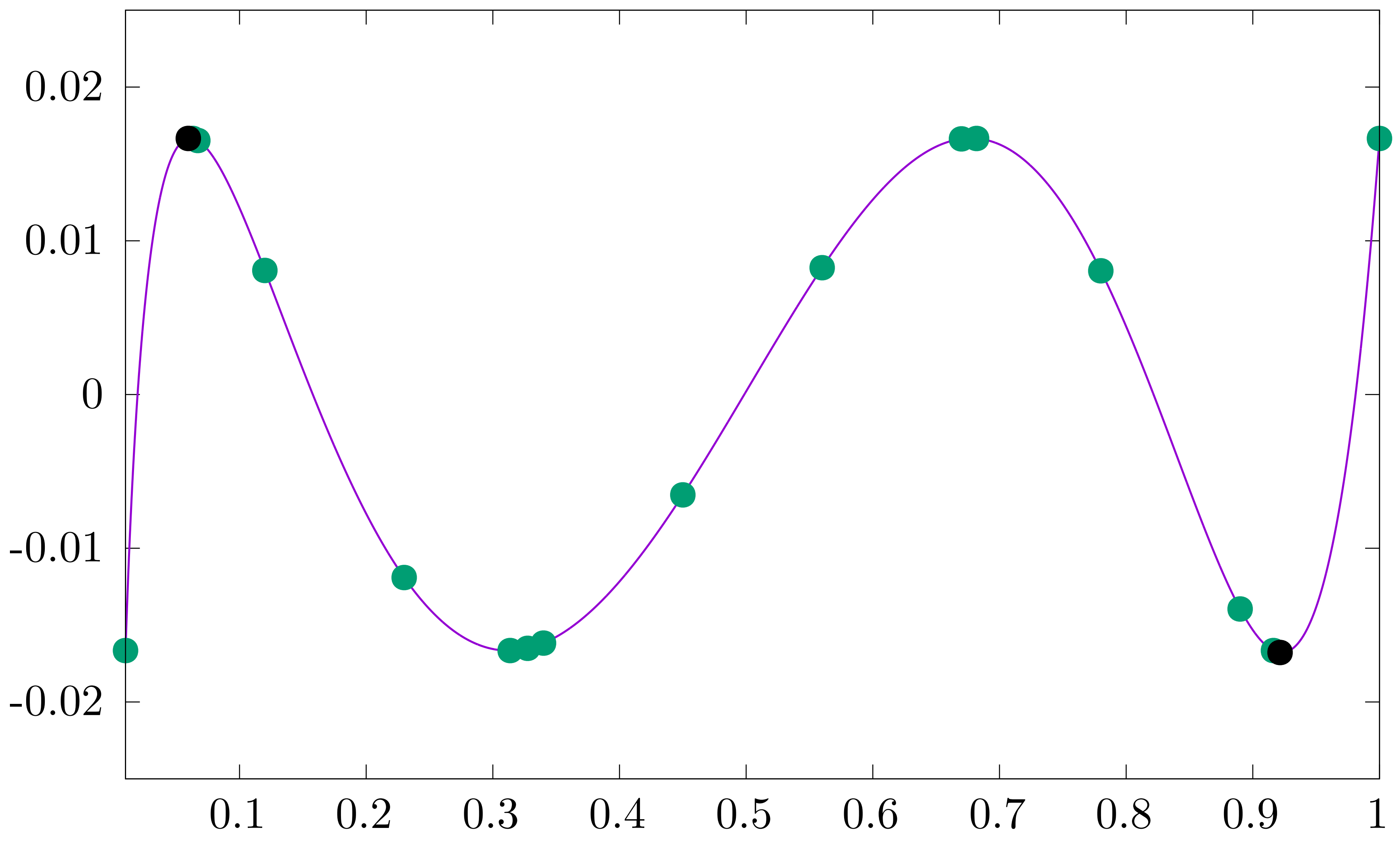
Notation:

- minimal error $\mu(D) = \min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}$, $D \subseteq B$
- best approximation $r_D = \arg\min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}$
- a maximum error point $x_k^* \in \arg\max_{x \in E_k} |e_k(x)|$

Example:

$$f(x) = \exp(x + \sqrt{x})$$
$$B = [0.01, 1]$$

$$r(x) = \frac{p_1 + p_2 \exp(x) + p_3 \sin(x)}{q_1 + q_2 x + q_3 \cos(2x)}$$



Step 1. $k \leftarrow 0$ and $D_0 \subseteq B$ finite set with $|D_0| \geqslant m + n$

do

Step 2. Find $\mu(D_k)$ and solution r_{D_k} to $P_{\mathbb{R}}[D_k]$

$e_k := f - r_{D_k}$

Step 3. Compute $E_k = \left\{ x \in B \,\middle|\, \begin{array}{l} x \text{ local extrema of } e_k \text{ over } B \\ \text{with } |e_k(x)| > \mu(D_k) \end{array} \right\}$

$D_{k+1} := D_k \cup E_k$

Step 4. $k \leftarrow k + 1$

while $\left(|e_{k-1}(x_{k-1}^*)| - \mu(D_{k-1})\right) / |e_{k-1}(x_{k-1}^*)| > 10^{-4}$

[1] Modifications of the First Remez Algorithm, *R. Reemtsen*, SIAM Journal of Numerical Analysis, Vol. 27, No. 2, pp. 507–518, 1990.

Finding Solutions to $P_{\mathbb{R}}[B]$

- a family of **Generalized First Remez Algorithms** [1]

Notation:

- minimal error $\mu(D) = \min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}$, $D \subseteq B$
- best approximation $r_D = \arg\min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}$
- a maximum error point $x_k^* \in \arg\max_{x \in E_k} |e_k(x)|$

Step 1. $k \leftarrow 0$ and $D_0 \subseteq B$ finite set with $|D_0| \geqslant m + n$

do

Step 2. Find $\mu(D_k)$ and solution r_{D_k} to $P_{\mathbb{R}}[D_k]$

$e_k := f - r_{D_k}$

Step 3. Compute $E_k = \left\{ x \in B \,\middle|\, \begin{array}{l} x \text{ local extrema of } e_k \text{ over } B \\ \text{with } |e_k(x)| > \mu(D_k) \end{array} \right\}$

$D_{k+1} := D_k \cup E_k$

Step 4. $k \leftarrow k + 1$

while $\left(|e_{k-1}(x_{k-1}^*)| - \mu(D_{k-1})\right) / |e_{k-1}(x_{k-1}^*)| > 10^{-4}$

- there are also **Second Remez Algorithms** [2]
  - potentially faster: *exchange* procedure ($|D_k| = m + n, \forall k \geqslant 0$)
  - sensitive to choice of D_0 and can fail to converge

[1] Modifications of the First Remez Algorithm, *R. Reemtsen*, SIAM Journal of Numerical Analysis, Vol. 27, No. 2, pp. 507–518, 1990.
[2] Introduction to Approximation Theory, *E. W. Cheney*, AMS Chelsea Pub., 1982.

Finding Solutions to $P_{\mathbb{R}}[B]$

- a family of Generalized First Remez Algorithms [1]

Notation:

- minimal error $\mu(D) = \min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty, D}$, $D \subseteq B$
- best approximation $r_D = \arg\min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty, D}$
- a maximum error point $x_k^* \in \arg\max_{x \in E_k} |e_k(x)|$

Efficient Algorithm & Implementation:

- r_{D_k} and $\mu(D_k)$: adaptive differential correction (ADC) [3, 4]

  Idea: *small* active subsets $S_k \subseteq D_k$
- E_k : Chebyshev-proxy root finding [5]

Step 1. $k \leftarrow 0$ and $D_0 \subseteq B$ finite set with $|D_0| \geqslant m + n$

Initialize active subset $S_0 \subseteq D_0$ s.t. $|S_0| = m + n$

do

Step 2. Find $\mu(D_k)$ and solution r_{D_k} to $P_{\mathbb{R}}[D_k]$ using ADC [4]

with starting active subset S_k

$e_k := f - r_{D_k}$

S_{k+1} is final active subset when solving $P_{\mathbb{R}}[D_k]$

Step 3. Compute $E_k = \left\{ x \in B \,\middle|\, \begin{array}{l} x \text{ local extrema of } e_k \text{ over } B \\ \text{with } |e_k(x)| > \mu(D_k) \end{array} \right\}$

$D_{k+1} := D_k \cup E_k$

Step 4. $k \leftarrow k + 1$

while $\left(|e_{k-1}(x_{k-1}^*)| - \mu(D_{k-1})\right) / |e_{k-1}(x_{k-1}^*)| > 10^{-4}$

[1] Modifications of the First Remez Algorithm, *R. Reemtsen*, SIAM Journal of Numerical Analysis, Vol. 27, No. 2, pp. 507–518, 1990.
[2] Introduction to Approximation Theory, *E. W. Cheney*, AMS Chelsea Pub., 1982.
[3] Uniform Approximation by Rational Functions Having Restricted Denominators, *E.H. Kaufman Jr., G.D. Taylor*, Journal of Approximation Theory, Vol. 32, No. 1, pp. 9–26, 1981.
[4] An Adaptive Differential-Correction Algorithm, E.H. Kaufman Jr. and S.F. McCormick and G.D. Taylor, Journal of Approximation Theory, Vol. 37, No. 3, pp. 197–211, 1983.
[5] Approximation Theory and Approximation Practice. Extended Edition, *L.N. Trefethen*, SIAM, 2019.

Finding Solutions to $P_{\mathbb{R}}[B]$

- a family of Generalized First Remez Algorithms [1]

Notation:

- minimal error $\mu(D) = \min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}$, $D \subseteq B$
- best approximation $r_D = \arg\min_{r \in \mathcal{R}_L(D)} \|f - r\|_{\infty,D}$
- a maximum error point $x_k^* \in \arg\max_{x \in E_k} |e_k(x)|$

Efficient Algorithm & Implementation:

- r_{D_k} **and** $\mu(D_k)$ **: adaptive differential correction (ADC)** [3, 4]
  Idea: *small* active subsets $S_k \subseteq D_k$
- E_k **: Chebyshev-proxy root finding** [5]

Step 1. $k \leftarrow 0$ and $D_0 \subseteq B$ finite set with $|D_0| \geqslant m + n$
Initialize active subset $S_0 \subseteq D_0$ s.t. $|S_0| = m + n$

do

Step 2. Find $\mu(D_k)$ and solution r_{D_k} to $P_{\mathbb{R}}[D_k]$ using ADC [4]
with starting active subset S_k
$e_k := f - r_{D_k}$
S_{k+1} is final active subset when solving $P_{\mathbb{R}}[D_k]$

Step 3. Compute $E_k = \left\{ x \in B \,\middle|\, \begin{array}{l} x \text{ local extrema of } e_k \text{ over } B \\ \text{with } |e_k(x)| > \mu(D_k) \end{array} \right\}$

$D_{k+1} := D_k \cup E_k$

Step 4. $k \leftarrow k + 1$

while $\left(|e_{k-1}(x_{k-1}^*)| - \mu(D_{k-1})\right) / |e_{k-1}(x_{k-1}^*)| > 10^{-4}$

[1] Modifications of the First Remez Algorithm, *R. Reemtsen*, SIAM Journal of Numerical Analysis, Vol. 27, No. 2, pp. 507–518, 1990.
[2] Introduction to Approximation Theory, *E. W. Cheney*, AMS Chelsea Pub., 1982.
[3] Uniform Approximation by Rational Functions Having Restricted Denominators, *E.H. Kaufman Jr., G.D. Taylor*, Journal of Approximation Theory, Vol. 32, No. 1, pp. 9–26, 1981.
[4] An Adaptive Differential-Correction Algorithm, E.H. Kaufman Jr. and S.F. McCormick and G.D. Taylor, Journal of Approximation Theory, Vol. 37, No. 3, pp. 197–211, 1983.
[5] Approximation Theory and Approximation Practice. Extended Edition, *L.N. Trefethen*, SIAM, 2019.

Finding Solutions to $P_{\mathbb{F}}[B]$

- we extend the polynomial `fpminimax` approach from Sollya [1] to the nonlinear rational setting

[1] Efficient Polynomial L^∞-Approximations, *N. Brisebarre and S. Chevillard,* ARITH-18, 2008.

Finding Solutions to $P_{\mathbb{F}}[B]$

- we extend the polynomial `fpminimax` approach from Sollya [1] to the nonlinear rational setting

How?

Step 1. from solution $r = P/Q \in \mathcal{R}_L(B)$ to $P_{\mathbb{R}}[B]$, up to normalizing and reordering, we want

$$\widehat{r}(x) = \frac{\sum_{i=1}^{m} \widehat{p}_i \phi_i(x)}{\psi_1(x) + \sum_{i=2}^{n} \widehat{q}_i \psi_i(x)}$$

s.t. $\{\widehat{p}_i\}_{i=1}^{m}$, $\{\widehat{q}_i\}_{i=2}^{n}$ are desired machine-coefficient values and $\|f - \widehat{r}\|_{\infty,B}$ is minimized

[1] Efficient Polynomial L^{∞}-Approximations, *N. Brisebarre and S. Chevillard,* ARITH-18, 2008.

Finding Solutions to $P_{\mathbb{F}}[B]$

- we extend the polynomial `fpminimax` approach from Sollya [1] to the nonlinear rational setting

How?

Step 1. from solution $r = P/Q \in \mathcal{R}_L(B)$ to $P_{\mathbb{R}}[B]$, up to normalizing and reordering, we want

$$\widehat{r}(x) = \frac{\sum_{i=1}^{m} \widehat{p}_i \phi_i(x)}{\psi_1(x) + \sum_{i=2}^{n} \widehat{q}_i \psi_i(x)}$$

s.t. $\{\widehat{p}_i\}_{i=1}^{m}, \{\widehat{q}_i\}_{i=2}^{n}$ are desired machine-coefficient values and $\|f - \widehat{r}\|_{\infty,B}$ is minimized

or equivalently

$$\widehat{r}(x) = \frac{\sum_{i=1}^{m} a_i \widehat{\phi}_i(x)}{\psi_1(x) + \sum_{i=2}^{n} b_i \widehat{\psi}_i(x)}$$

s.t. $\{a_i\}_{i=1}^{m}, \{b_i\}_{i=2}^{n}$ are integers and $\widehat{\phi}_i(x) = 2^{-u_i}\phi_i(x), \widehat{\psi}_i(x) = 2^{-v_i}\psi_i(x)$, where u_i, v_i are exponents of rounded coeffs. p_i, q_i of r

[1] Efficient Polynomial L^{∞}-Approximations, *N. Brisebarre and S. Chevillard,* ARITH-18, 2008.

Finding Solutions to $P_{\mathbb{F}}[B]$

- we extend the polynomial `fpminimax` approach from Sollya [1] to the nonlinear rational setting

How?

Step 2. choose $N_r \geqslant m+n-1$ distinct points $\{x_k\}_{k=1}^{N_r}$ from B + linearize problem

$$\sum_{i=1}^{m} a_i \underbrace{\begin{bmatrix} \widehat{\phi}_i(x_1) \\ \widehat{\phi}_i(x_2) \\ \vdots \\ \widehat{\phi}_i(x_{N_r}) \end{bmatrix}}_{\boldsymbol{\alpha}_i} + \sum_{i=2}^{n} b_i \underbrace{\begin{bmatrix} -r(x_1)\widehat{\psi}_i(x_1) \\ -r(x_2)\widehat{\psi}_i(x_2) \\ \vdots \\ -r(x_{N_r})\widehat{\psi}_i(x_{N_r}) \end{bmatrix}}_{\boldsymbol{\beta}_i} \simeq \underbrace{\begin{bmatrix} r(x_1)\psi_1(x_1) \\ r(x_2)\psi_1(x_2) \\ \vdots \\ r(x_{N_r})\psi_1(x_{N_r}) \end{bmatrix}}_{\boldsymbol{r}}$$

- we want to find integer a_i, b_i s.t. $\left\| \sum_{i=1}^{m} a_i \boldsymbol{\alpha}_i + \sum_{i=2}^{n} b_i \boldsymbol{\beta}_i - \boldsymbol{r} \right\|_\infty$ is minimized

[1] Efficient Polynomial L^∞-Approximations, *N. Brisebarre and S. Chevillard,* ARITH-18, 2008.

Finding Solutions to $P_{\mathbb{F}}[B]$

- we extend the polynomial `fpminimax` approach from Sollya [1] to the nonlinear rational setting

How?

Step 2. choose $N_r \geqslant m+n-1$ distinct points $\{x_k\}_{k=1}^{N_r}$ from B + linearize problem

$$\sum_{i=1}^{m} a_i \underbrace{\begin{bmatrix} \widehat{\phi}_i(x_1) \\ \widehat{\phi}_i(x_2) \\ \vdots \\ \widehat{\phi}_i(x_{N_r}) \end{bmatrix}}_{\boldsymbol{\alpha}_i} + \sum_{i=2}^{n} b_i \underbrace{\begin{bmatrix} -r(x_1)\widehat{\psi}_i(x_1) \\ -r(x_2)\widehat{\psi}_i(x_2) \\ \vdots \\ -r(x_{N_r})\widehat{\psi}_i(x_{N_r}) \end{bmatrix}}_{\boldsymbol{\beta}_i} \simeq \underbrace{\begin{bmatrix} r(x_1)\psi_1(x_1) \\ r(x_2)\psi_1(x_2) \\ \vdots \\ r(x_{N_r})\psi_1(x_{N_r}) \end{bmatrix}}_{\boldsymbol{r}}$$

- we want to find integer a_i, b_i s.t. $\left\| \sum_{i=1}^{m} a_i \boldsymbol{\alpha}_i + \sum_{i=2}^{n} b_i \boldsymbol{\beta}_i - \boldsymbol{r} \right\|_\infty$ is minimized **TOO DIFFICULT**
- search for integer a_i, b_i s.t. $\left\| \sum_{i=1}^{m} a_i \boldsymbol{\alpha}_i + \sum_{i=2}^{n} b_i \boldsymbol{\beta}_i - \boldsymbol{r} \right\|_2$ is *approximately* minimized

[1] Efficient Polynomial L^∞-Approximations, *N. Brisebarre and S. Chevillard,* ARITH-18, 2008.

Finding Solutions to $P_{\mathbb{F}}[B]$

- we extend the polynomial `fpminimax` approach from Sollya [1] to the nonlinear rational setting

Challenges

What is the best normalization choice?

$$\widehat{r}(x) = \frac{\sum_{i=1}^{m} \widehat{p}_i \phi_i(x)}{\psi_1(x) + \sum_{i=2}^{n} \widehat{q}_i \psi_i(x)}$$

Heuristic: sweep through $[1, 2)$ binade (128 different values)

[1] Efficient Polynomial L^{∞}-Approximations, *N. Brisebarre and S. Chevillard,* ARITH-18, 2008.

Finding Solutions to $P_{\mathbb{F}}[B]$

- we extend the polynomial `fpminimax` approach from Sollya [1] to the nonlinear rational setting

Challenges

What is the best normalization choice?

$$\widehat{r}(x) = \frac{\sum_{i=1}^{m} \widehat{p}_i \phi_i(x)}{\psi_1(x) + \sum_{i=2}^{n} \widehat{q}_i \psi_i(x)}$$

Heuristic: sweep through $[1, 2)$ binade (128 different values)

How many and which discretization nodes?

- in polynomial case, nb. of points close to the degree [1] (*i.e.*, zeros of $f - r$ or Chebyshev nodes)
- rational case: can lead to *spurious poles* inside B (*e.g.* Froissart doublets)
  - ➡ larger nb. of points helps: $N_r = 10(m+n)$ points distributed following zeros of $f - r$

[1] Efficient Polynomial L^∞-Approximations, *N. Brisebarre and S. Chevillard,* ARITH-18, 2008.

A Motivating Example: $\arctan$

- CORE-MATH [1] implementation of float $\arctan$

$$f(x) = \arctan(x), x \in B = [0.000127, 1]$$

$$r(x) := \frac{\sum_{i=1}^{7} p_i \phi_i(x)}{\sum_{i=1}^{7} q_i \psi_i(x)} = \frac{\sum_{i=1}^{7} p_i x^{2i-1}}{\sum_{i=1}^{7} q_i x^{2i-2}}$$

Goal: $P_{\mathbb{F}}[B]$ with double prec. coefficients

- use our new `minimax` command to solve $P_{\mathbb{R}}[B]$

$$\|\varepsilon\|_{\infty,B} \approx 2^{-57.26}$$

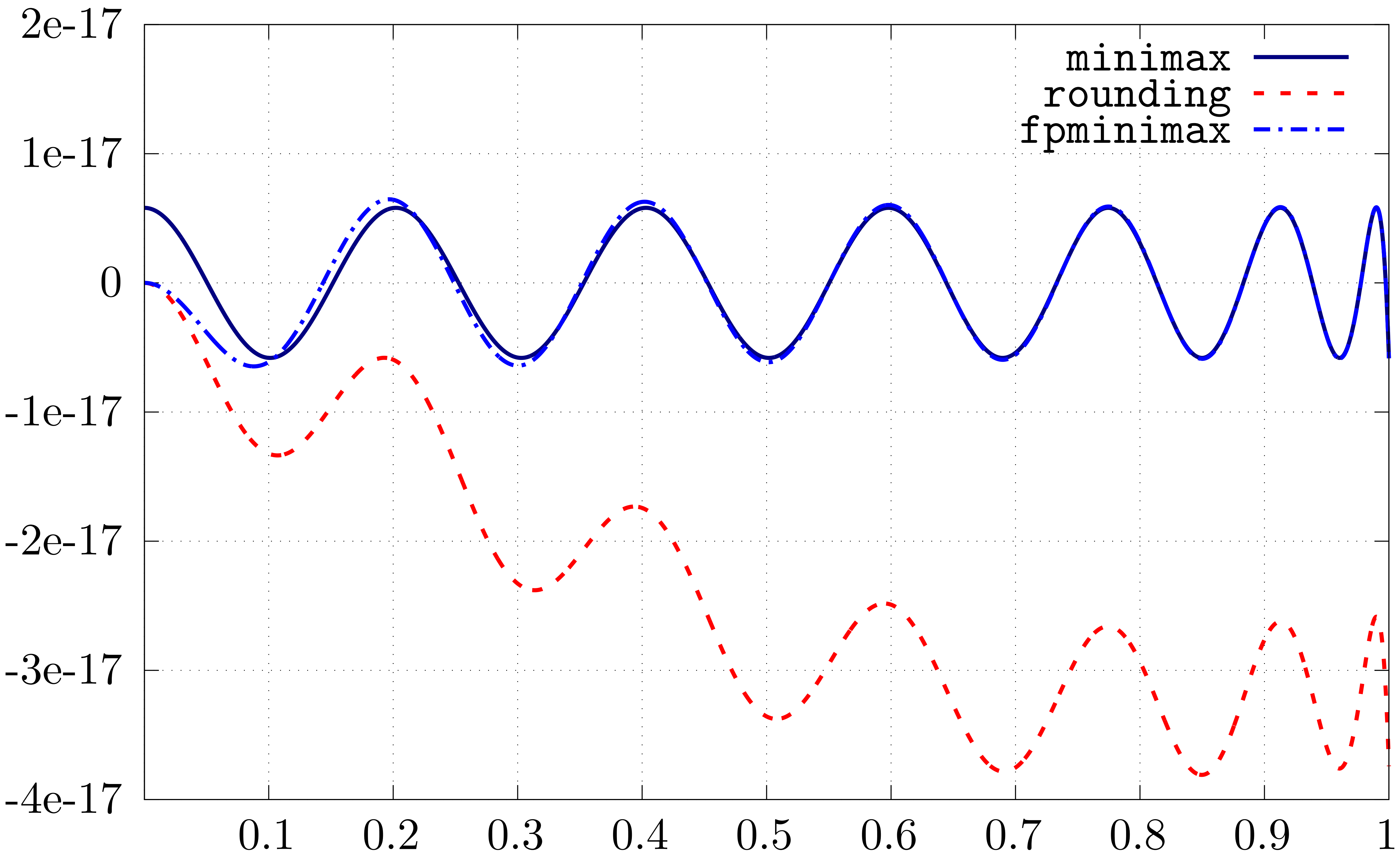
What happens if we round coeffs. to double prec. ?

$$\|\varepsilon\|_{\infty,B} \approx 2^{-54.54}$$

- use our new `fpminimax` command to address $P_{\mathbb{F}}[B]$

$$\|\varepsilon\|_{\infty,B} \approx 2^{-57.09}$$

$$\varepsilon(x) = (f(x) - r(x))/f(x)$$



[1] The CORE-MATH Project, *A. Sibidanov and P. Zimmermann and S. Glondu,* ARITH-29, 2022.

A Motivating Example: arctan

- CORE-MATH [1] implementation of float arctan

$$f(x) = \arctan(x), x \in B = [0.000127, 1]$$

$$r(x) := \frac{\sum_{i=1}^{7} p_i \phi_i(x)}{\sum_{i=1}^{7} q_i \psi_i(x)} = \frac{\sum_{i=1}^{7} p_i x^{2i-1}}{\sum_{i=1}^{7} q_i x^{2i-2}}$$

Goal: $P_{\mathbb{F}}[B]$ with double prec. coefficients

- use our new `minimax` command to solve $P_{\mathbb{R}}[B]$

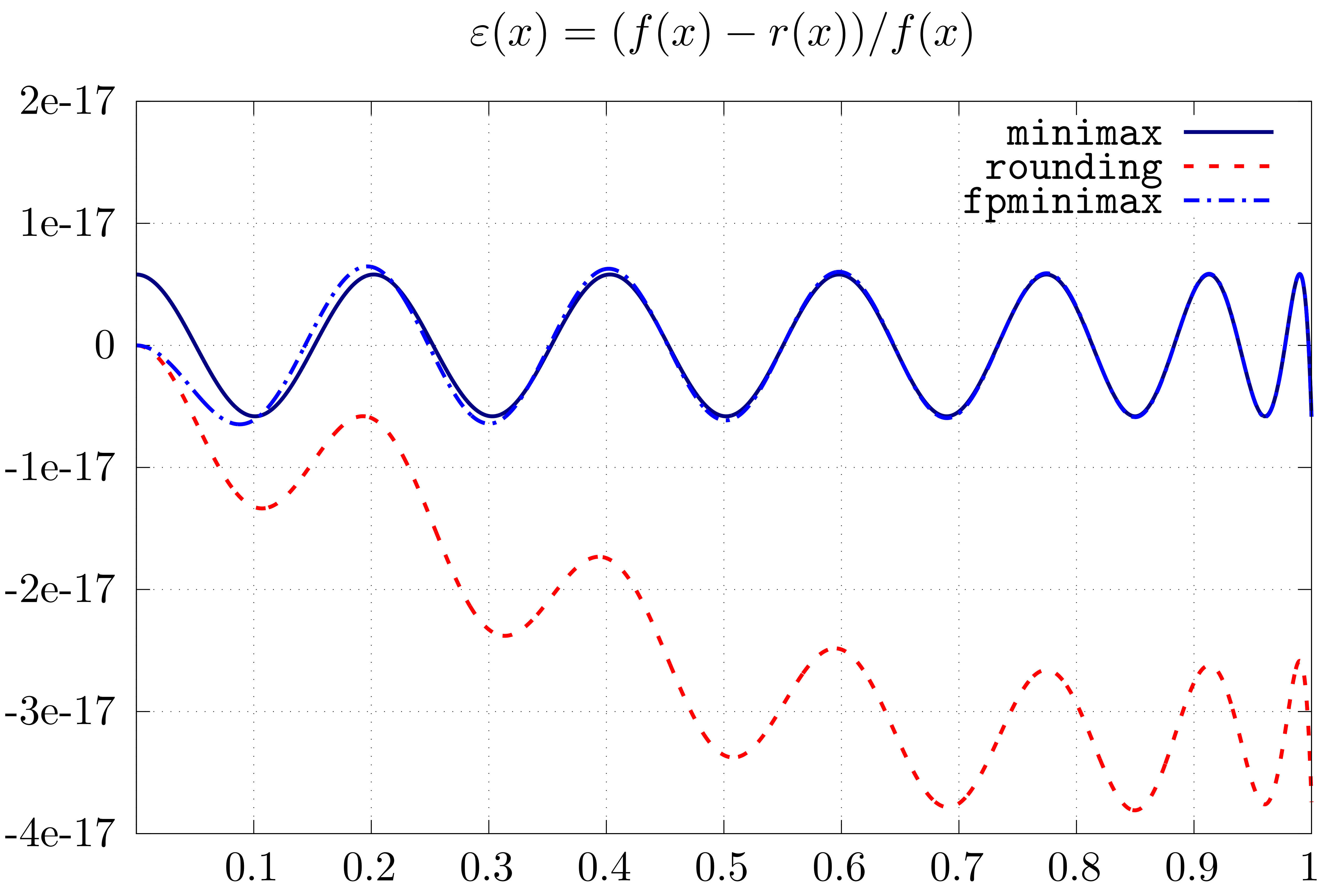
$$\|\varepsilon\|_{\infty,B} \approx 2^{-57.26}$$

What happens if we round coeffs. to double prec. ?

$$\|\varepsilon\|_{\infty,B} \approx 2^{-54.54}$$

- use our new `fpminimax` command to address $P_{\mathbb{F}}[B]$

$$\|\varepsilon\|_{\infty,B} \approx 2^{-57.09}$$



Problem:

- correct rounding not ensured with round to nearest and $x =$ `0x1.1ad646p-4`

[1] The CORE-MATH Project, *A. Sibidanov and P. Zimmermann and S. Glondu,* ARITH-29, 2022.

A Motivating Example: $\arctan$

- CORE-MATH [1] implementation of float $\arctan$

$$f(x) = \arctan(x), x \in B = [0.000127, 1]$$

$$r(x) := \frac{\sum_{i=1}^{7} p_i \phi_i(x)}{\sum_{i=1}^{7} q_i \psi_i(x)} = \frac{\sum_{i=1}^{7} p_i x^{2i-1}}{\sum_{i=1}^{7} q_i x^{2i-2}}$$

Goal: $P_{\mathbb{F}}[B]$ with double prec. coefficients

- use our new `minimax` command to solve $P_{\mathbb{R}}[B]$

$$\|\varepsilon\|_{\infty,B} \approx 2^{-57.26}$$

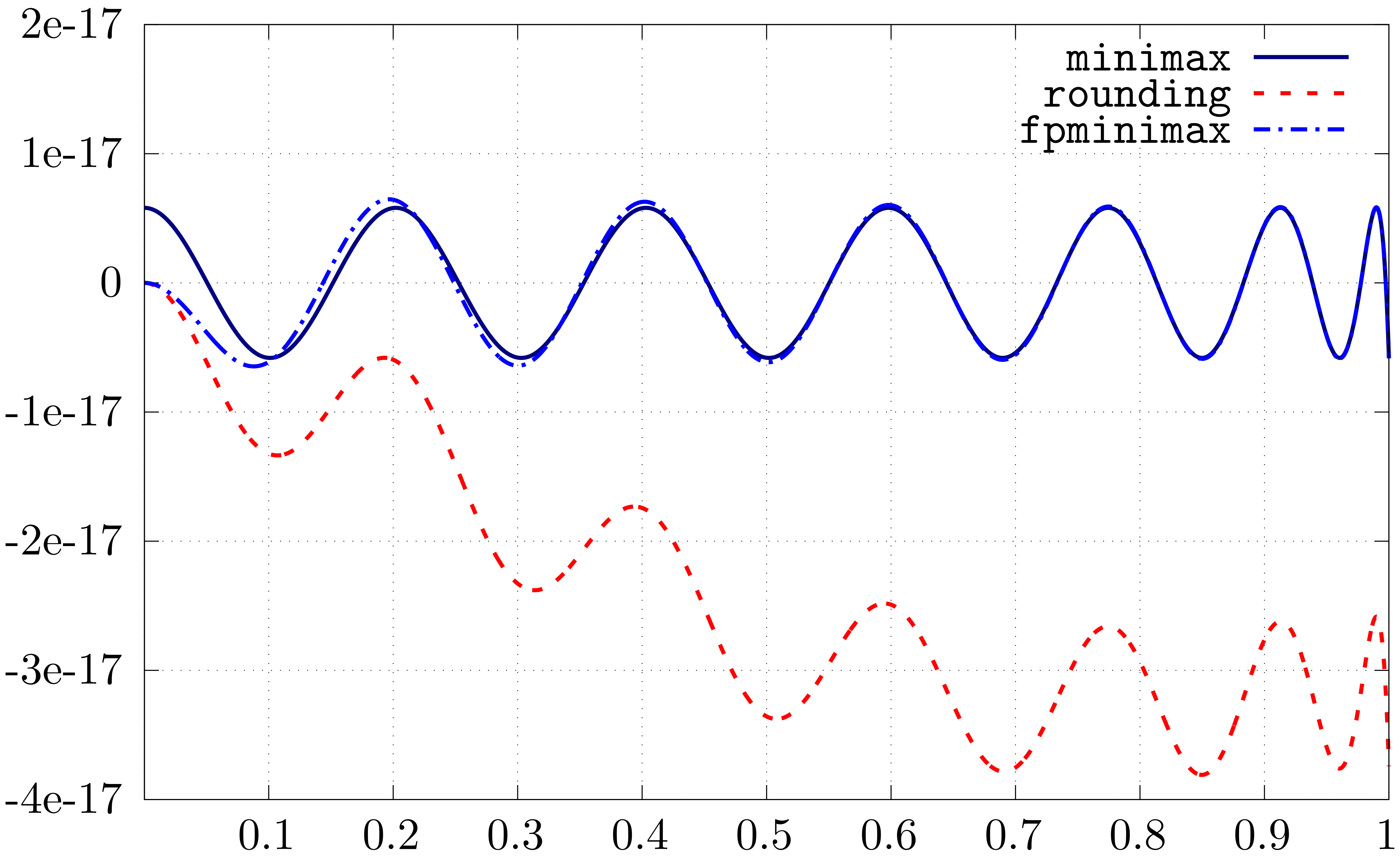
What happens if we round coeffs. to double prec. ?

$$\|\varepsilon\|_{\infty,B} \approx 2^{-54.54}$$

- use our new `fpminimax` command to address $P_{\mathbb{F}}[B]$

$$\|\varepsilon\|_{\infty,B} \approx 2^{-57.09}$$

$$\varepsilon(x) = (f(x) - r(x))/f(x)$$



Problem:

- correct rounding not ensured with round to nearest and $x =$ `0x1.1ad646p-4`

Solution: normalization search resolves issue, resulting in $\|\varepsilon\|_{\infty,B} \approx 2^{-57.10}$

[1] The CORE-MATH Project, *A. Sibidanov and P. Zimmermann and S. Glondu,* ARITH-29, 2022.

Another Example: Inverse Langevin Function

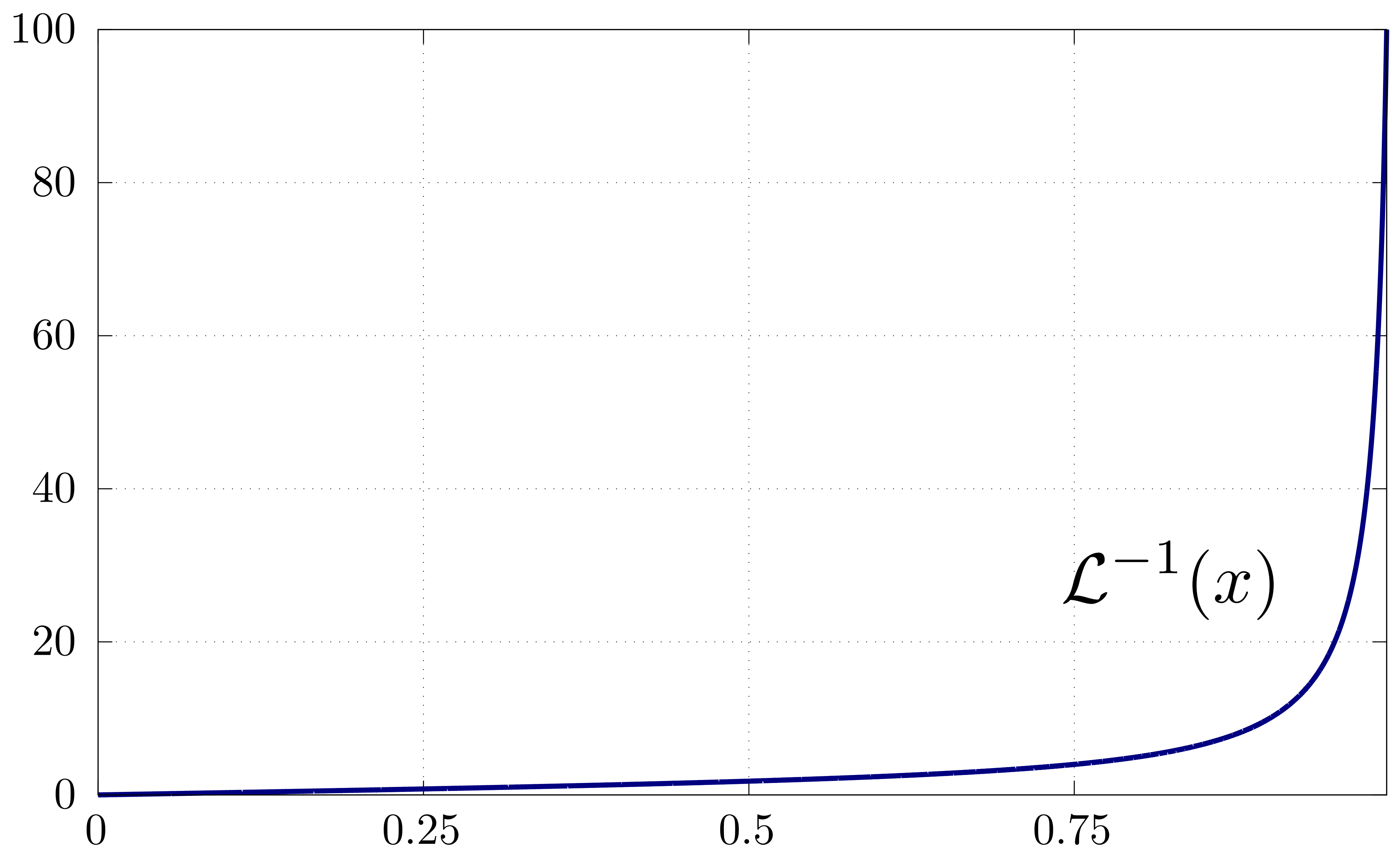
- the Langevin function:

$$y = \mathcal{L}(x) = \coth(x) - 1/x$$

- use cases:
  - ➡ polymer science
  - ➡ magnetism
  - ➡ biomechanics

Challenge: $\mathcal{L}^{-1}$ has no closed form representation

- many low accuracy approximations (2-4 digits)
- need for more accuracy [1] (*e.g.* 12-13 digits)



[1] Effect of the inverse Langevin approximation on the solution of the Fokker-Planck equation of non-linear dilute polymer, *A. Ammar,* Journal of Non-Newtonian Fluid Mechanics, Vol. 231, pp. 153–163, 2016.

Another Example: Inverse Langevin Function

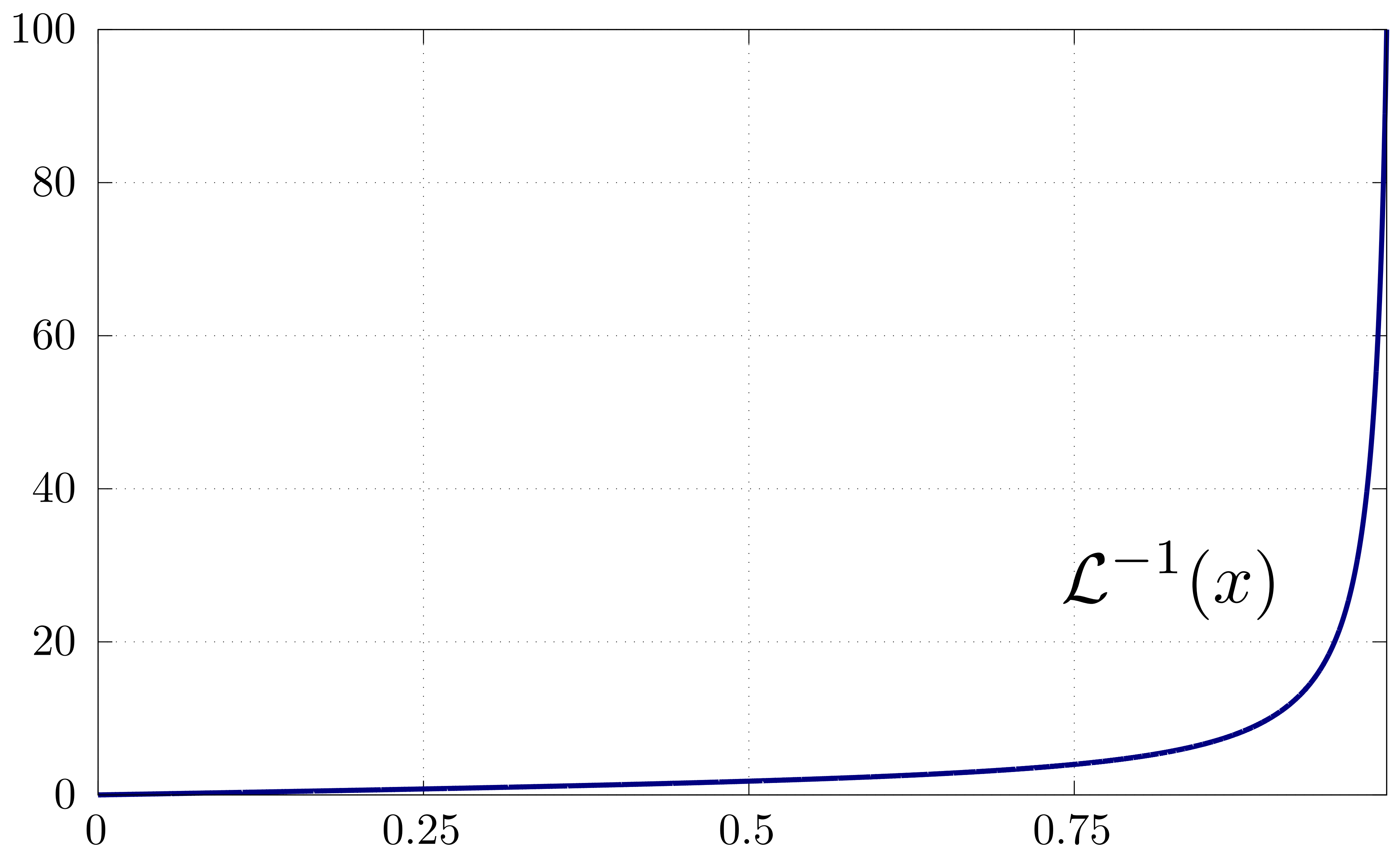
- the Langevin function:

$$y = \mathcal{L}(x) = \coth(x) - 1/x$$

- use cases:
  - ➡ polymer science
  - ➡ magnetism
  - ➡ biomechanics

Challenge: $\mathcal{L}^{-1}$ has no closed form representation

- many low accuracy approximations (2-4 digits)
- need for more accuracy [1] (*e.g.* 12-13 digits)



Properties of $\mathcal{L}^{-1}$:

- simple pole at $x = 1$
- $\mathrm{Res}(\mathcal{L}^{-1}, 1) = \lim_{x \to 1} (1 - x)\mathcal{L}^{-1}(x) = 1$
- Taylor expansion around $x = 0$ with radius $R = 0.904643\ldots$ [2]
- infinite nb. of complex singularities with modulus 1

$$\mathcal{L}^{-1}(x) = 3x + \frac{9}{5}x^3 + \frac{297}{175}x^5 + \frac{1539}{875}x^7 + \cdots$$

[1] Effect of the inverse Langevin approximation on the solution of the Fokker-Planck equation of non-linear dilute polymer, *A. Ammar,* Journal of Non-Newtonian Fluid Mechanics, Vol. 231, pp. 153–163, 2016.
[2] On the complex singularities of the inverse Langevin function, S.R. Rickaby and N.H. Scott, IMA Journal of Numerical Analysis, Vol. 83, No. 6, pp. 1092–1116, 2018.

Another Example: Inverse Langevin Function

- we can focus on approximating

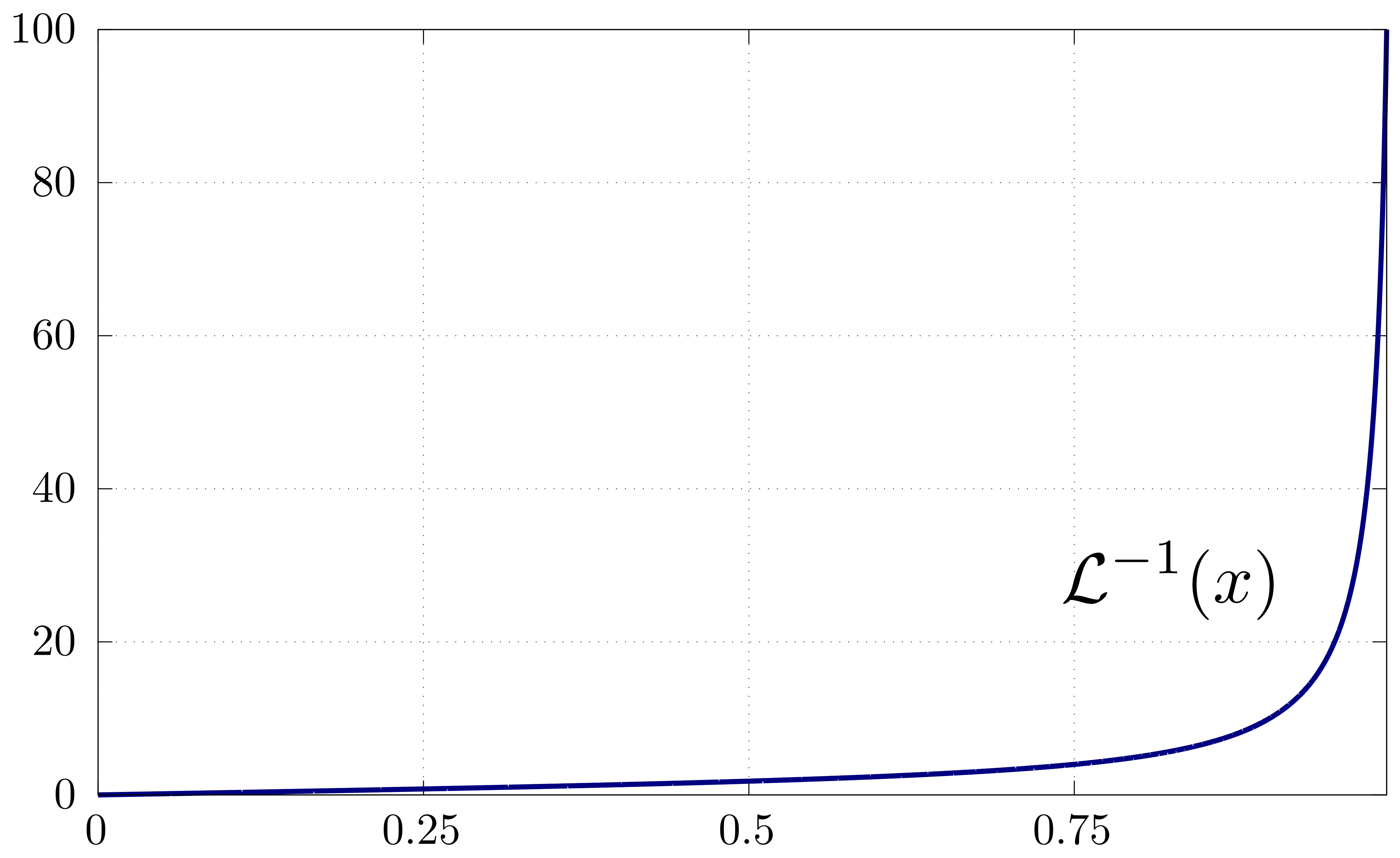
$$f(x) = \begin{cases} \frac{\mathcal{L}^{-1}(x)(1-x)}{x}, & x \in (0,1), \\ 3, & x = 0, \\ 1, & x = 1. \end{cases}$$

using approximations from the sets

$$\mathcal{R}_{m,n} = \left\{ r(x) := \frac{\sum_{i=1}^{m+1} p_i x^{i-1}}{\sum_{i=1}^{n+1} q_i x^{i-1}} \right\}$$

and

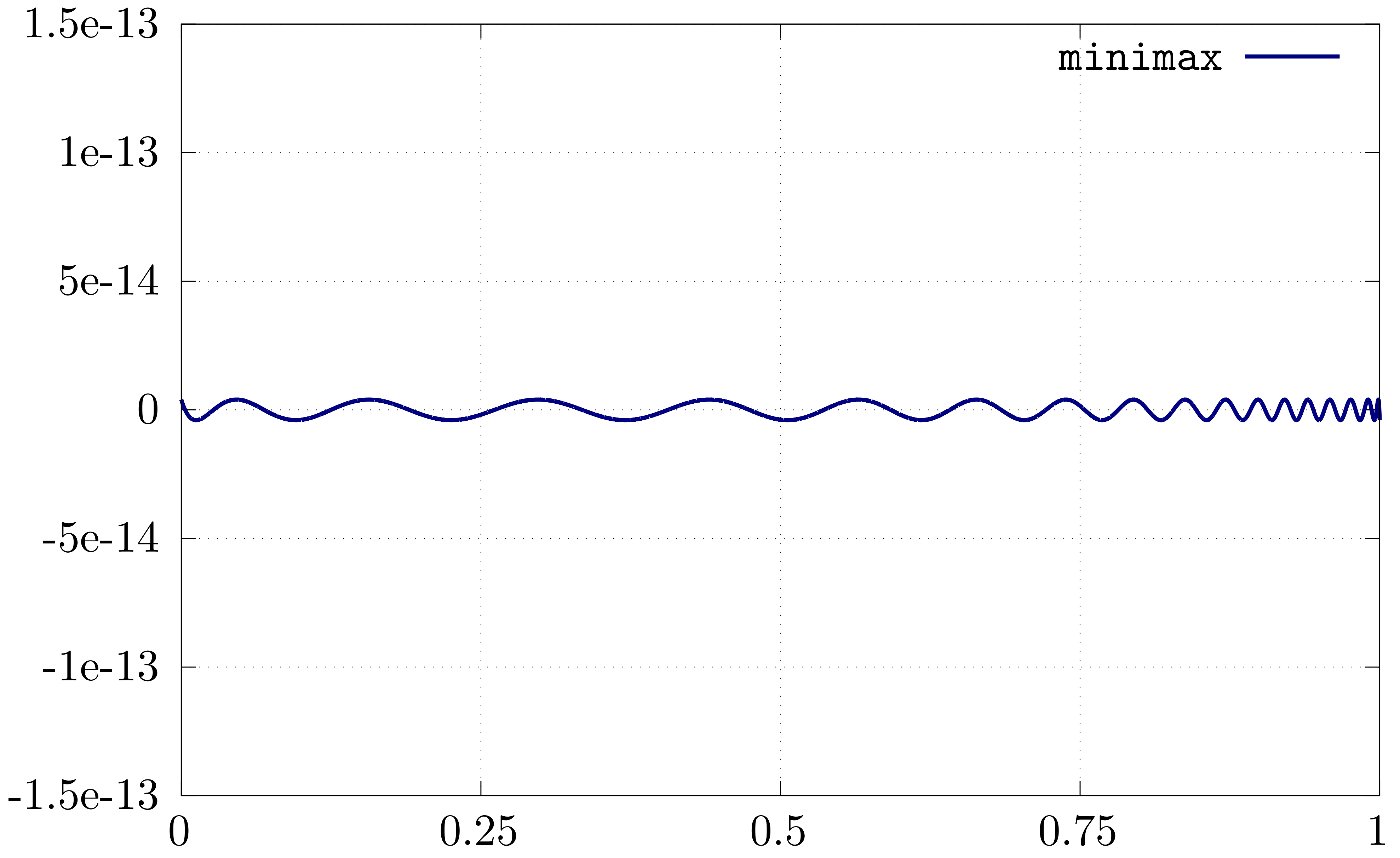
$$\mathcal{J}_{m,n} = \left\{ r(x) := \frac{\sum_{i=1}^{m+1} p_i x^{2i-2}}{\sum_{i=1}^{n+1} q_i x^{i-1}} \right\}$$



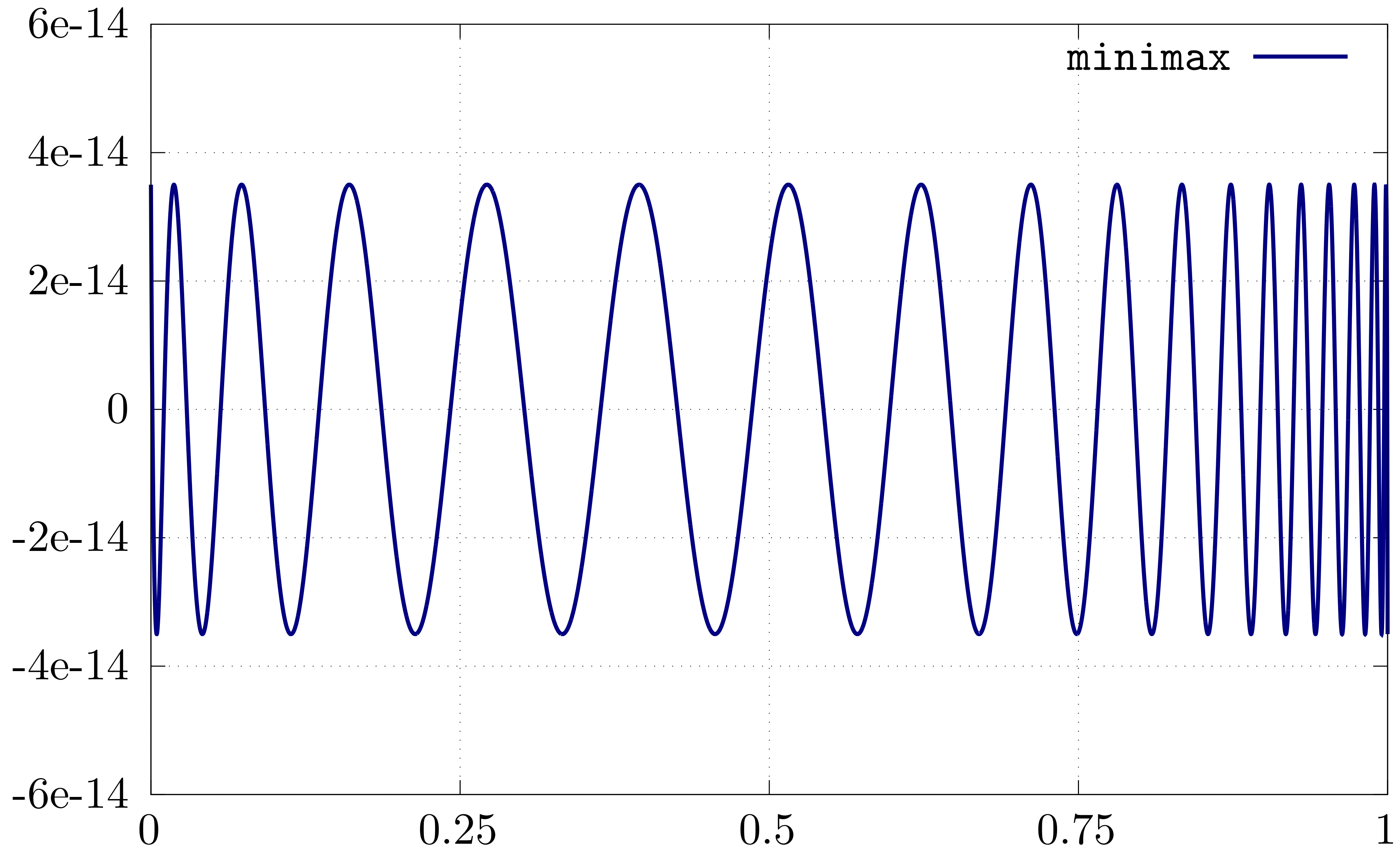
[1] Effect of the inverse Langevin approximation on the solution of the Fokker-Planck equation of non-linear dilute polymer, *A. Ammar,* Journal of Non-Newtonian Fluid Mechanics, Vol. 231, pp. 153–163, 2016.
[2] On the complex singularities of the inverse Langevin function, S.R. Rickaby and N.H. Scott, IMA Journal of Numerical Analysis, Vol. 83, No. 6, pp. 1092–1116, 2018.

Another Example: Inverse Langevin Function

$\mathcal{R}_{17,17}$



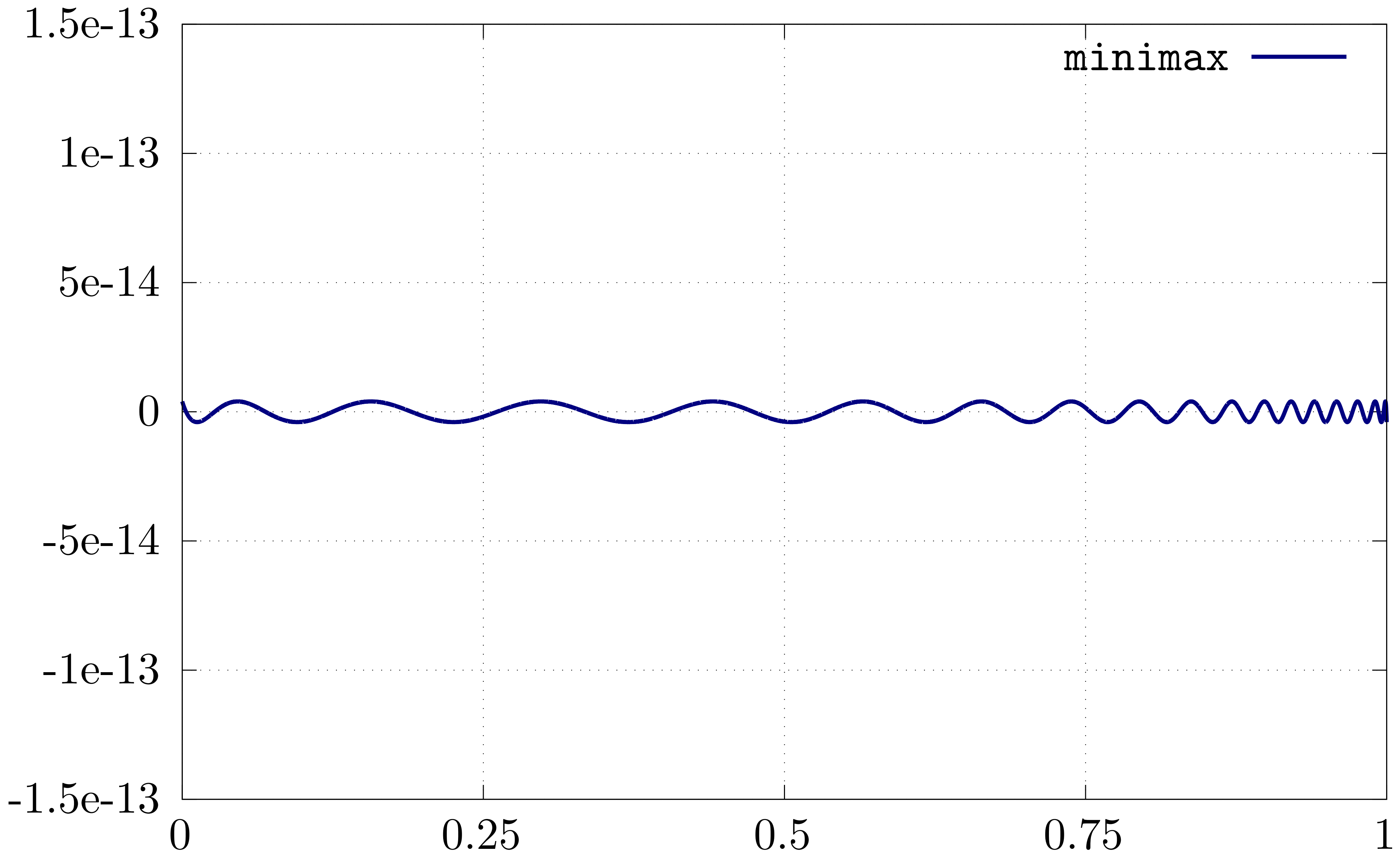
$\mathcal{J}_{17,17}$



- `minimax` error $4.0 \cdot 10^{-15}$
- `minimax` error $3.5 \cdot 10^{-14}$
- need degree $m > 70$ polynomial approx. for error $< 10^{-14}$

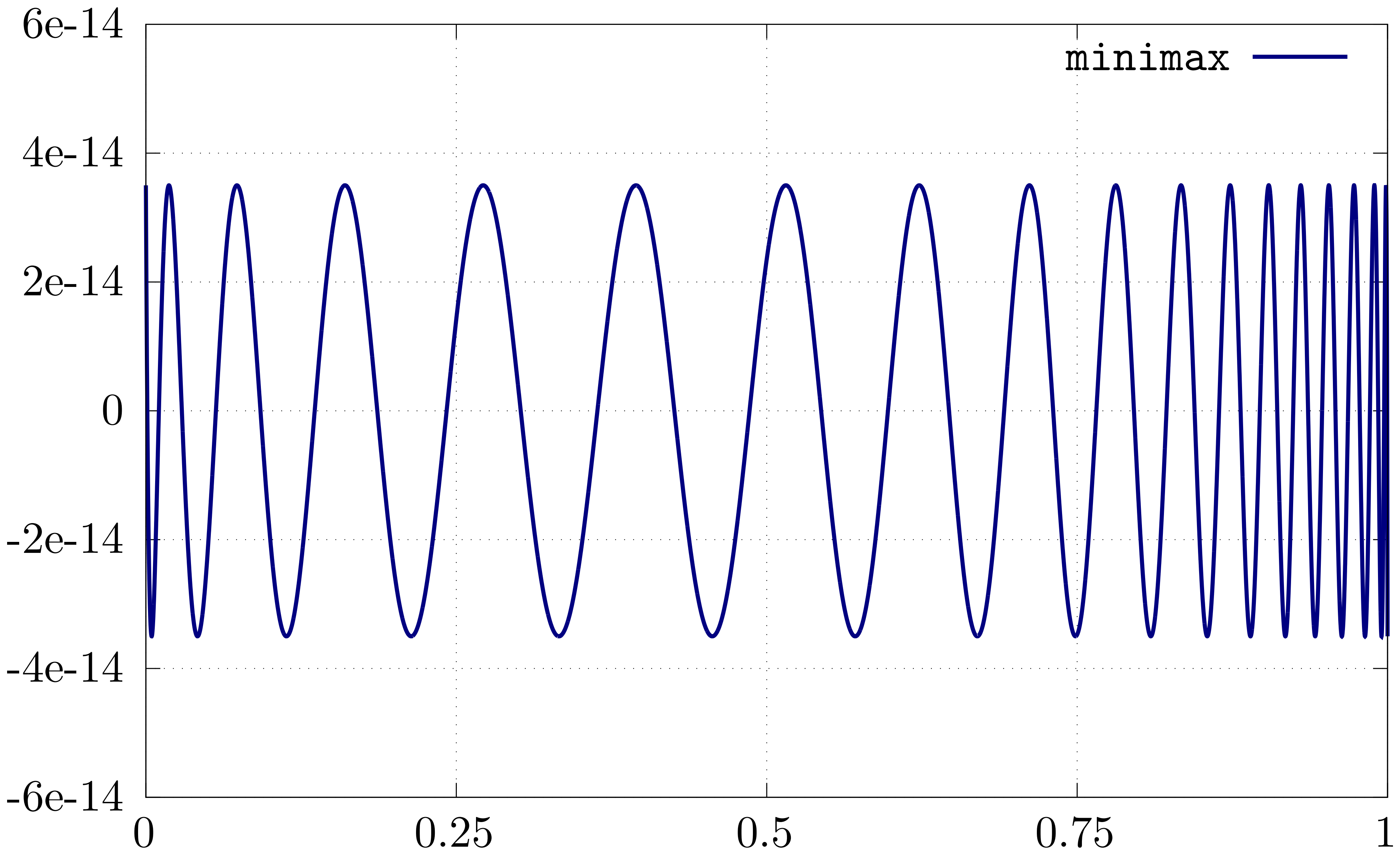
Another Example: Inverse Langevin Function

$\mathcal{R}_{17,17}$



- minimax error $4.0 \cdot 10^{-15}$
- rounded coeff. (double) error $1.14 \cdot 10^{-2}$

$\mathcal{J}_{17,17}$

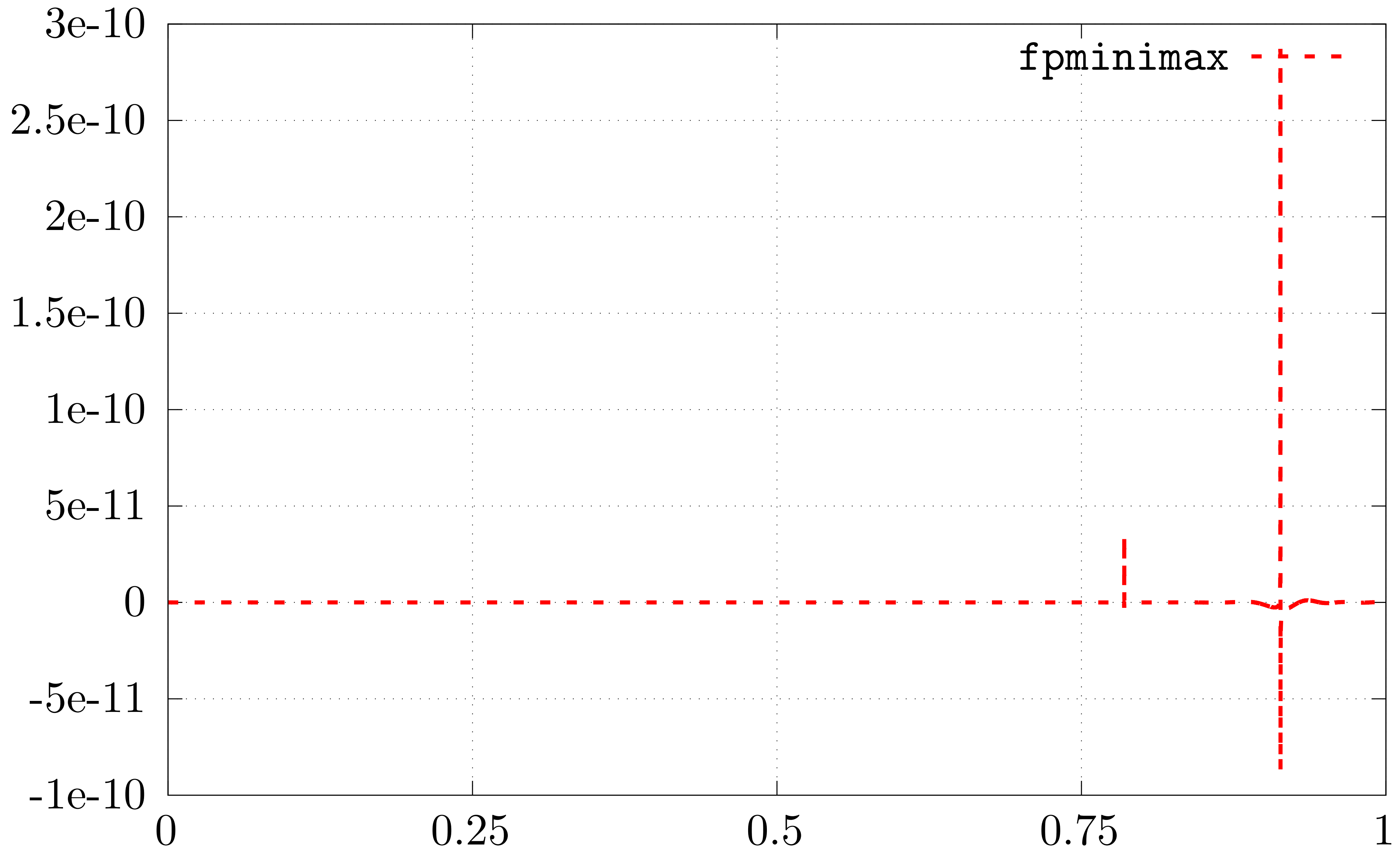


- minimax error $3.5 \cdot 10^{-14}$
- rounded coeff. (double) error $1.07 \cdot 10^{-9}$

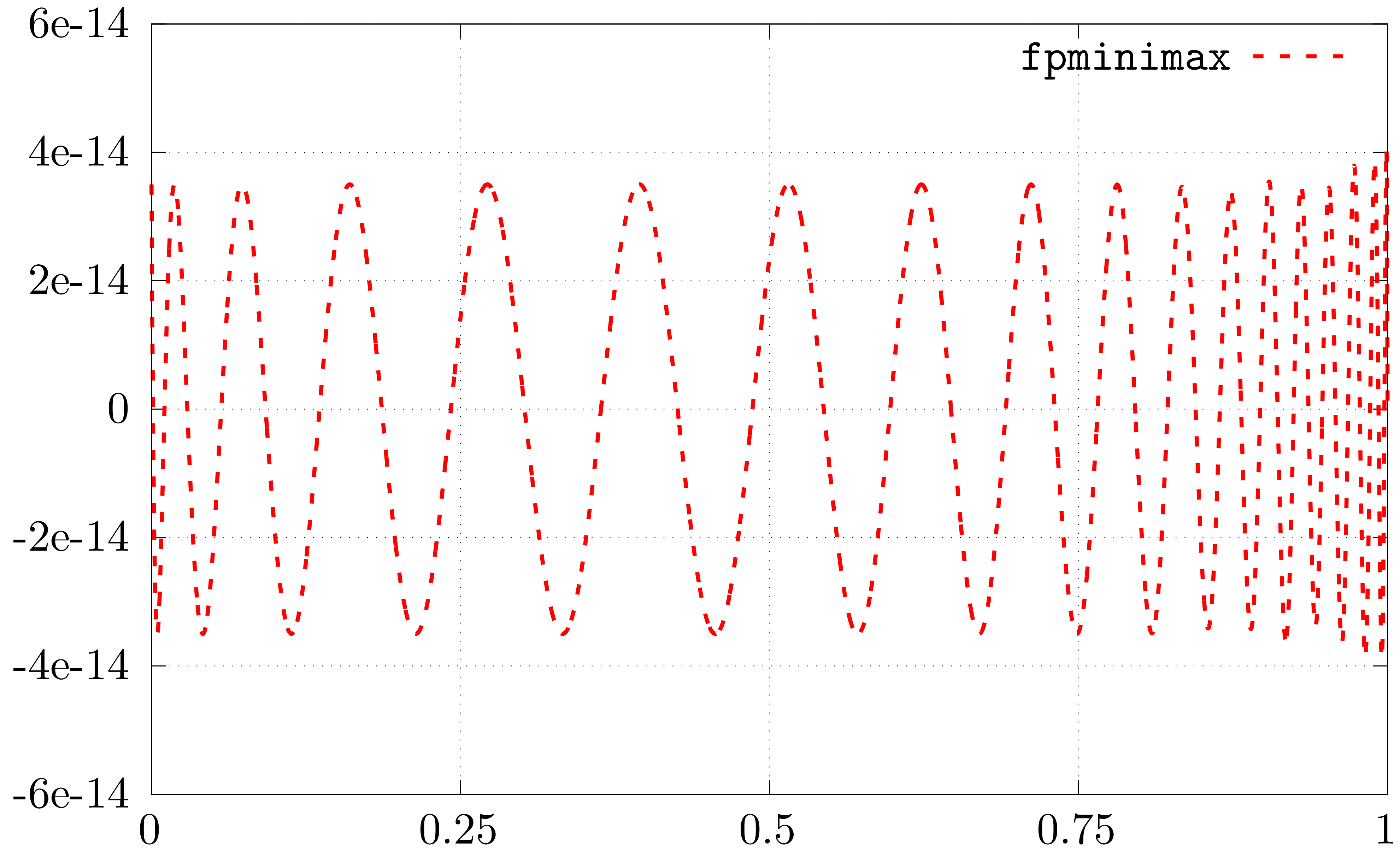
- sensitive to coefficient perturbations

Another Example: Inverse Langevin Function

$\mathcal{R}_{17,17}$



$\mathcal{J}_{17,17}$



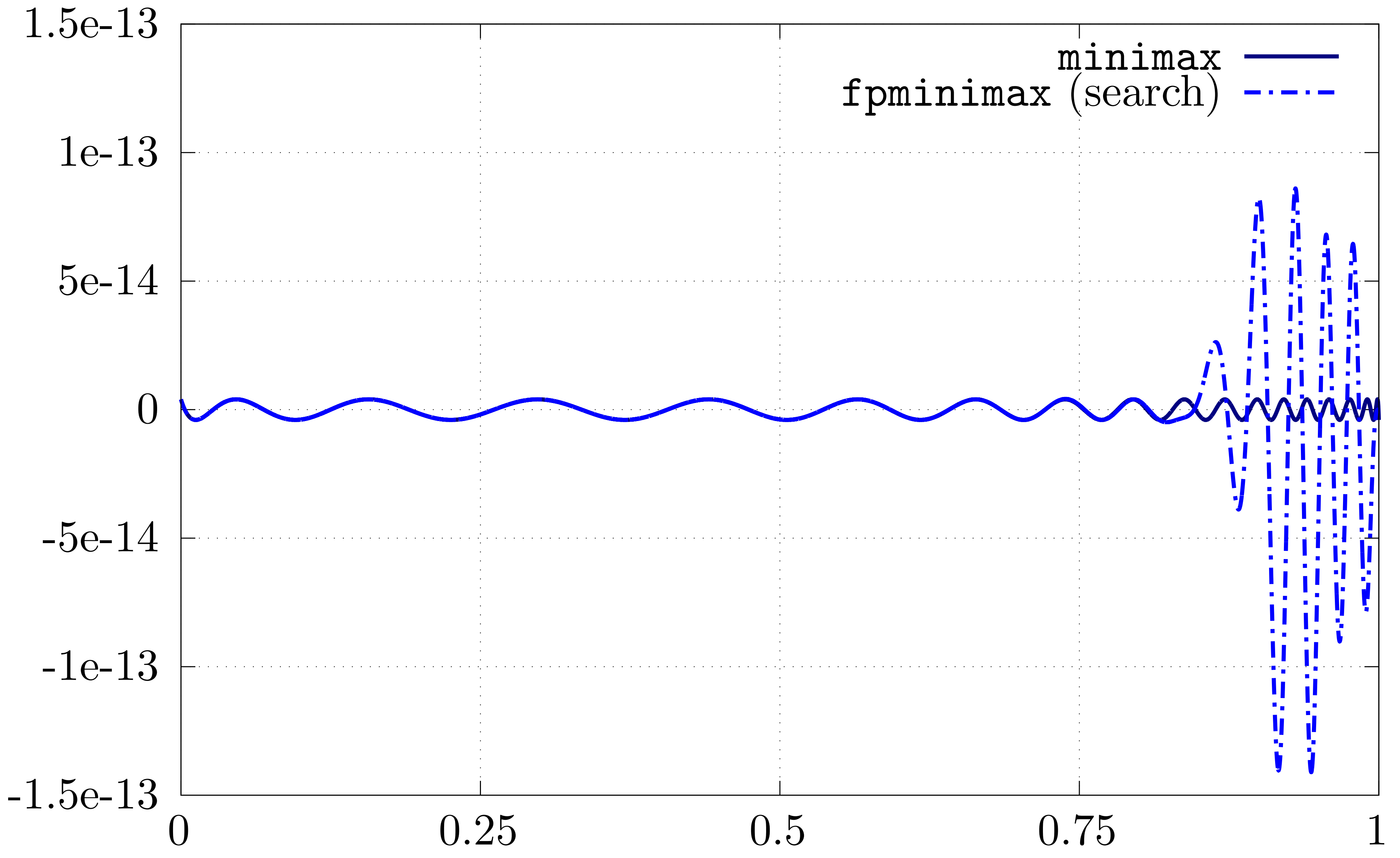
- `minimax` error $4.0 \cdot 10^{-15}$
- rounded coeff. (`double`) error $1.14 \cdot 10^{-2}$
- `fpminimax` introduces two spurious poles, but...

- sensitive to coefficient perturbations

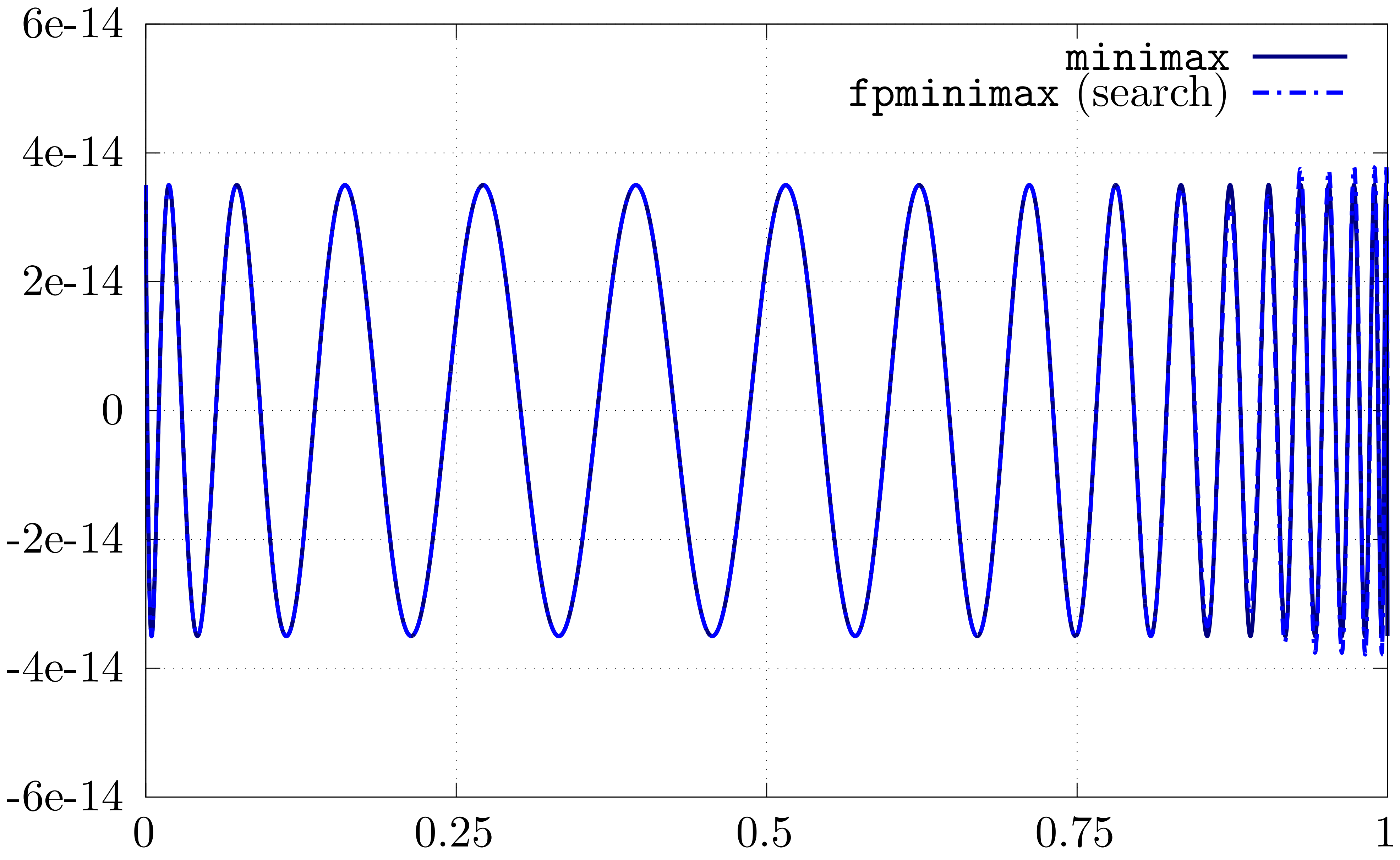
- `minimax` error $3.5 \cdot 10^{-14}$
- rounded coeff. (`double`) error $1.07 \cdot 10^{-9}$
- `fpminimax` recovers lost accuracy, with error $4.05 \cdot 10^{-14}$

Another Example: Inverse Langevin Function

$\mathcal{R}_{17,17}$



$\mathcal{J}_{17,17}$

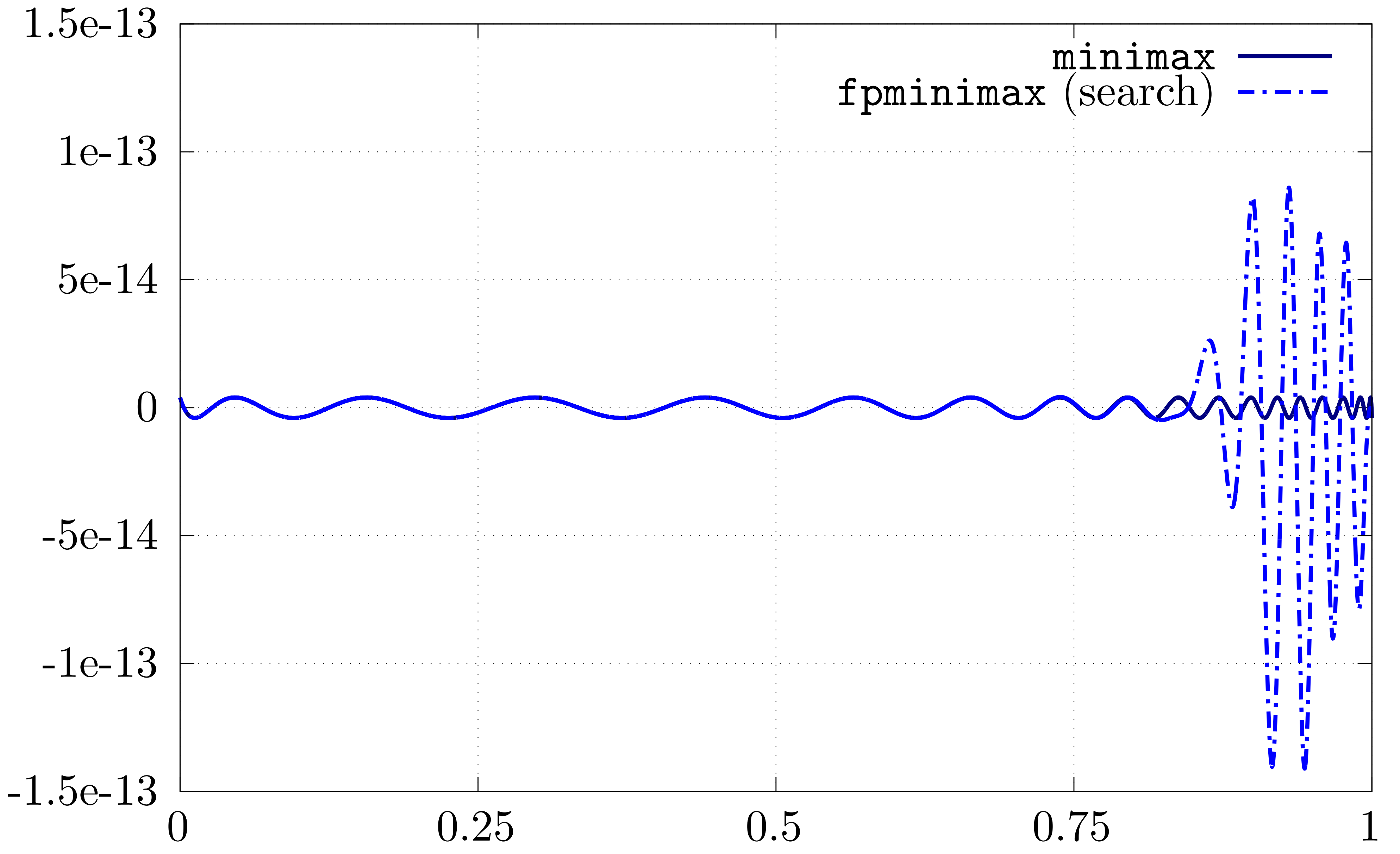


- `minimax` error $4.0 \cdot 10^{-15}$
- rounded coeff. (`double`) error $1.14 \cdot 10^{-2}$
- `fpminimax` introduces two spurious poles, but...
- ... normalization search removes them, error $1.15 \cdot 10^{-13}$
- sensitive to coefficient perturbations

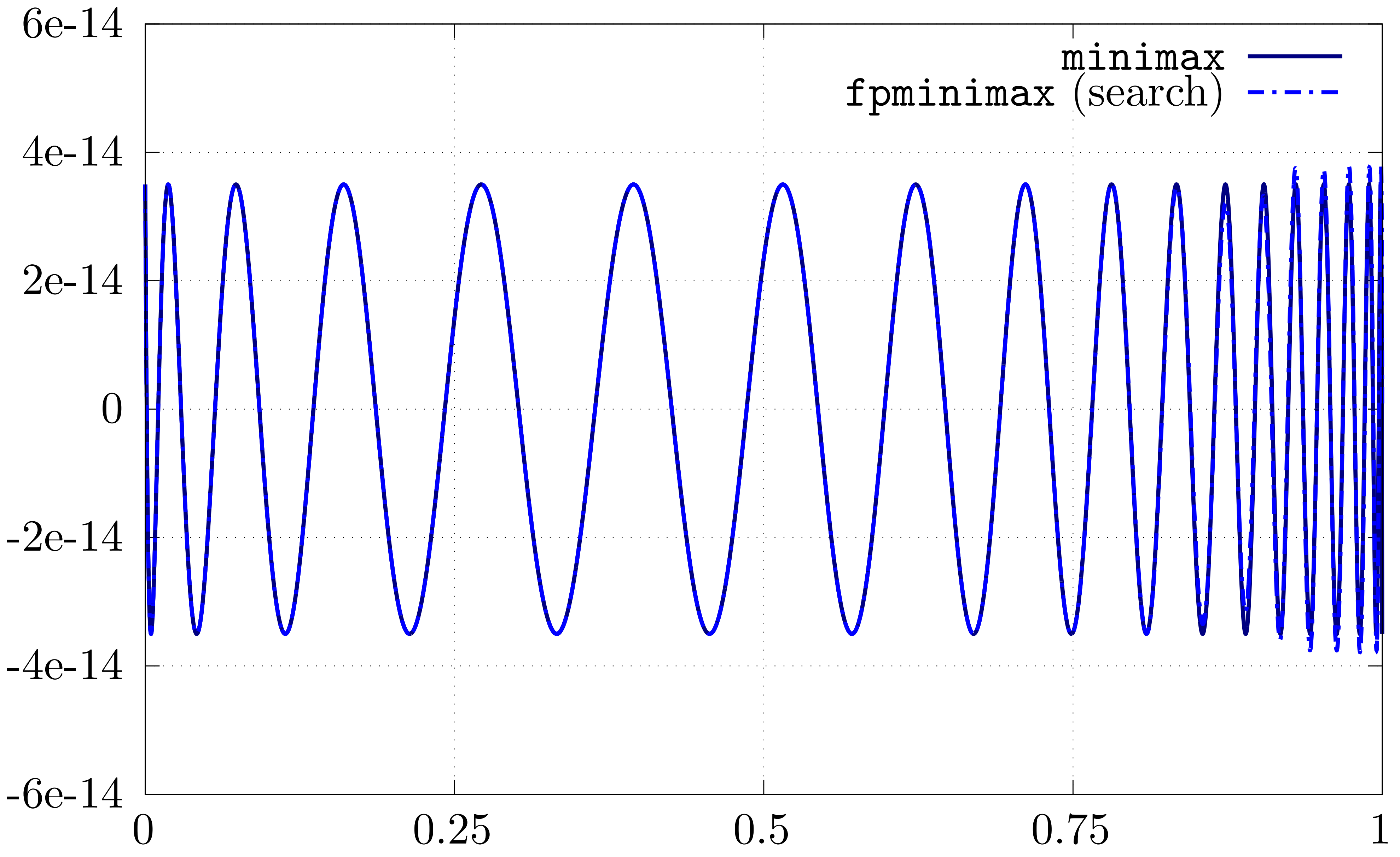
- `minimax` error $3.5 \cdot 10^{-14}$
- rounded coeff. (`double`) error $1.07 \cdot 10^{-9}$
- `fpminimax` recovers lost accuracy, with error $4.05 \cdot 10^{-14}$
- normalization search reduces error to $3.81 \cdot 10^{-14}$

Another Example: Inverse Langevin Function

$\mathcal{R}_{17,17}$



$\mathcal{J}_{17,17}$



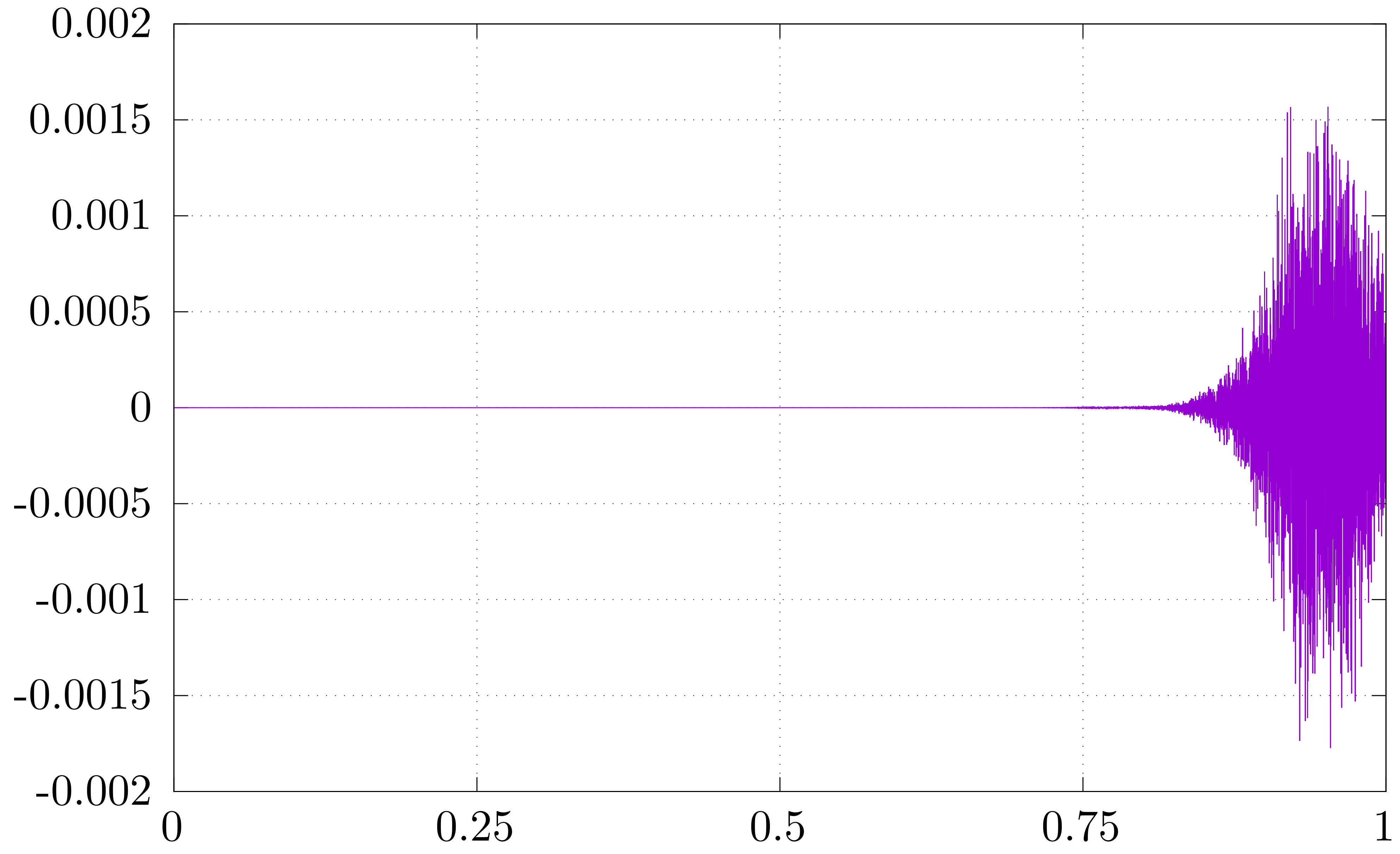
- `minimax` error $4.0 \cdot 10^{-15}$
- rounded coeff. (`double`) error $1.14 \cdot 10^{-2}$
- `fpminimax` introduces two spurious poles, but...
- ... normalization search removes them, error $1.15 \cdot 10^{-13}$
- sensitive to coefficient perturbations

- `minimax` error $3.5 \cdot 10^{-14}$
- rounded coeff. (`double`) error $1.07 \cdot 10^{-9}$
- `fpminimax` recovers lost accuracy, with error $4.05 \cdot 10^{-14}$
- normalization search reduces error to $3.81 \cdot 10^{-14}$

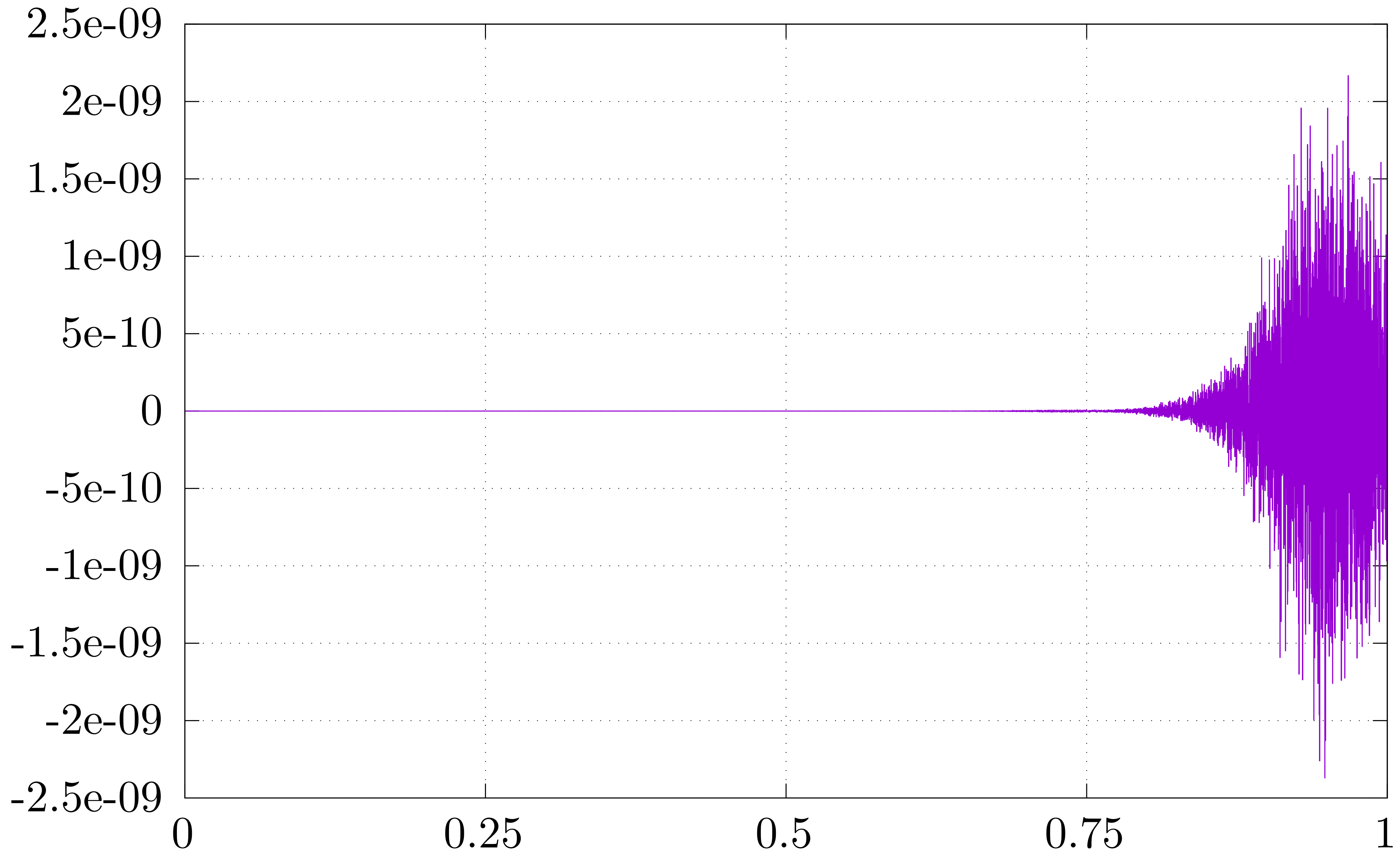
What about evaluation errors?

Another Example: Inverse Langevin Function

$\mathcal{R}_{17,17}$



$\mathcal{J}_{17,17}$



- implementation using Horner scheme with addition and multiplication
- sensitivity/ill-conditioning present close to 1

Another Example: Inverse Langevin Function

What can we do?

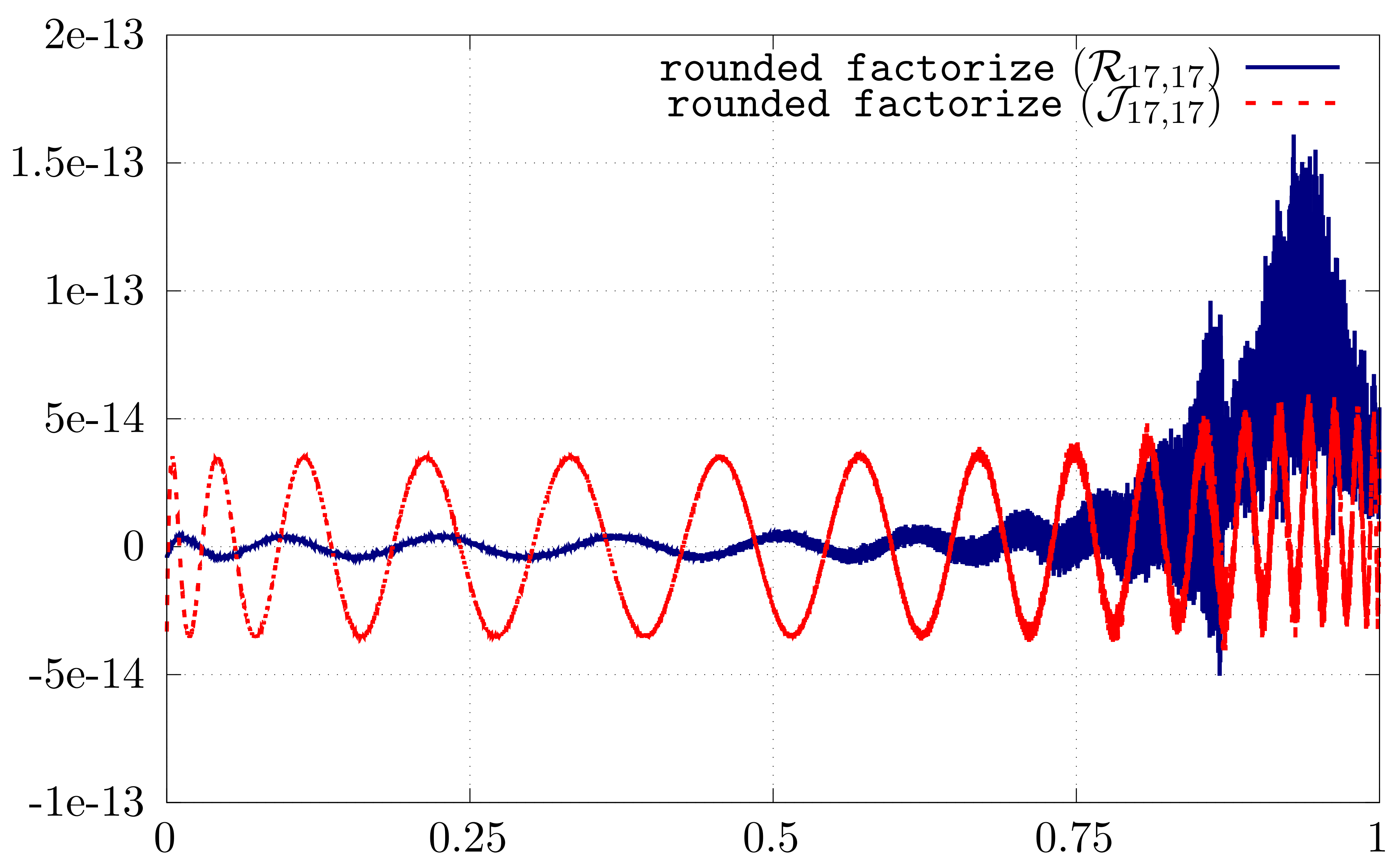
- higher precision coefficients + arithmetic (e.g. double-`double`)
- interval subdivision + polynomial approximations
- use better-conditioned representations:
  - barycentric form [1, 2]:

$$r(x) = \frac{\sum_{i=1}^{m} \frac{p_i}{x - x_i}}{\sum_{i=1}^{m} \frac{q_i}{x - x_i}}$$

  - **factorized representation of numerator and denominator**

    eight degree two factors + one degree one
    in numerator/denominator

  - ...
- ...



[1] The AAA algorithm for rational approximation, *Y. Nakatsukasa and O. Sète and L.N. Trefethen,* SIAM Journal of Scientific Computing, Vol. 40, No. 3, pp. A1494—A1522, 2018.
[2] Rational Minimax Approximation via Adaptive Barycentric Representations, S.-I. Filip and Y. Nakatsukasa and L.N. Trefethen, SIAM Journal of Scientific Computing, Vol. 40, No. 4, pp. A2427—A2455, 2018.

Summary

- introduce generalized rational approximation algorithms

$$P_{\mathbb{R}}[B]\text{: } \texttt{minimax} \qquad\qquad P_{\mathbb{F}}[B]\text{: } \texttt{fpminimax}$$

- C++ implementation (library + standalone executable)
- eases exploration of polynomial vs rational in `libm` design contexts
  - ➡ regarding latency and throughput of division (from [1]):

This must be taken into account ... when hesitating between ... a polynomial or rational function. There is a chicken-or-egg issue here: ... programmers ... tend to avoid using [division], and since it is less used, computer manufacturers do not make the necessary efforts to significantly accelerate it.

Repository link:

[1] Floating-point arithmetic, *S. Boldo and C.-P. Jeannerod and G. Melquiond and J.-M. Muller*, Acta Numerica, 32:203–290, 2023.

Summary

- introduce generalized rational approximation algorithms

$$P_{\mathbb{R}}[B]\text{: } \texttt{minimax} \qquad\qquad P_{\mathbb{F}}[B]\text{: } \texttt{fpminimax}$$

- C++ implementation (library + standalone executable)
- eases exploration of **polynomial vs rational** in `libm` design contexts
  - ➡ regarding latency and throughput of division (from [1]):

This must be taken into account … when hesitating between … a polynomial or rational function. There is a chicken-or-egg issue here: … programmers … tend to avoid using [division], and since it is less used, computer manufacturers do not make the necessary efforts to significantly accelerate it.

TODOs

- possible integration into Sollya
- optimize normalization factor search procedure
- explore & optimize different rewritings of r
- multivariate approximation problems
- …

Repository link:

[1] Floating-point arithmetic, *S. Boldo and C.-P. Jeannerod and G. Melquiond and J.-M. Muller*, Acta Numerica, 32:203–290, 2023.

Summary

- introduce generalized rational approximation algorithms

$$P_{\mathbb{R}}[B]\text{: } \texttt{minimax} \qquad P_{\mathbb{F}}[B]\text{: } \texttt{fpminimax}$$

- C++ implementation (library + standalone executable)
- eases exploration of polynomial vs rational in `libm` design contexts
  - ➡ regarding latency and throughput of division (from [1]):

This must be taken into account … when hesitating between … a polynomial or rational function. There is a chicken-or-egg issue here: … programmers … tend to avoid using [division], and since it is less used, computer manufacturers do not make the necessary efforts to significantly accelerate it.

TODOs

- possible integration into Sollya
- optimize normalization factor search procedure
- explore & optimize different rewritings of r
- multivariate approximation problems
- …

Repository link:

Thank you! Questions?

[1] Floating-point arithmetic, *S. Boldo and C.-P. Jeannerod and G. Melquiond and J.-M. Muller*, Acta Numerica, 32:203–290, 2023.