Approximation Theory and Proof Assistants: Certified Computations

Nicolas Brisebarre and Damien Pous

Master 2 Informatique Fondamentale
École Normale Supérieure de Lyon, 2020-2021

Chapter 4. Interval Arithmetic, Interval Analysis

Floating Point (FP) Arithmetic

Given

$$\begin{cases} \text{a radix} & \beta \geqslant 2, \\ \text{a precision} & p \geqslant 1, \\ \text{a set of exponents} & E_{\text{min}}, \cdots, E_{\text{max}}. \end{cases}$$

A finite FP number x is represented by 2 integers:

- integer mantissa : M, $\beta^{p-1} \leqslant |M| \leqslant \beta^p - 1$;
- exponent E, $E_{\text{min}} \leqslant E \leqslant E_{\text{max}}$

such that

$$x = \frac{M}{\beta^{p-1}} \times \beta^E.$$

We assume binary FP arithmetic (that is to say $\beta = 2$.)
We denote $\mathcal{F}_p$ the corresponding set of FP numbers.
Multiple-precision FP arithmetic: we let p and E vary.

IEEE Precisions

See http://en.wikipedia.org/wiki/IEEE_floating_point

	precision	minimal exponent	maximal exponent
single (binary 32)	24	−126	127
double (binary 64)	53	−1022	1023
extended double	64	−16382	16383
quadruple (binary 128)	113	−16382	16383

IEEE Rounding Modes

The result of an arithmetic operation whose input values belong to $\mathcal{F}_p$ may not belong to $\mathcal{F}_p$ (in general it does not): the result must be rounded.

IEEE Rounding Modes

The result of an arithmetic operation whose input values belong to $\mathcal{F}_p$ may not belong to $\mathcal{F}_p$ (in general it does not): the result must be rounded.

IEEE standard defines 4 different rounding modes:

- rounding towards $+\infty$, or upwards: $\circ_u(x) = \min\{y \in \mathcal{F}_p : y \geqslant x\}$;

IEEE Rounding Modes

The result of an arithmetic operation whose input values belong to $\mathcal{F}_p$ may not belong to $\mathcal{F}_p$ (in general it does not): the result must be rounded.

IEEE standard defines 4 different rounding modes:

- rounding towards $+\infty$, or upwards: $\circ_u(x) = \min\{y \in \mathcal{F}_p : y \geqslant x\}$;
- rounding towards $-\infty$, or downwards: $\circ_d(x) = \max\{y \in \mathcal{F}_p : y \leqslant x\}$;

IEEE Rounding Modes

The result of an arithmetic operation whose input values belong to $\mathcal{F}_p$ may not belong to $\mathcal{F}_p$ (in general it does not): the result must be rounded.

IEEE standard defines 4 different rounding modes:

- rounding towards $+\infty$, or upwards: $\circ_u(x) = \min\{y \in \mathcal{F}_p : y \geqslant x\}$;
- rounding towards $-\infty$, or downwards: $\circ_d(x) = \max\{y \in \mathcal{F}_p : y \leqslant x\}$;
- rounding towards 0: $\circ_z(x) := \circ_u(x)$ if $x < 0$, and to $\circ_d(x)$ otherwise;

IEEE Rounding Modes

The result of an arithmetic operation whose input values belong to $\mathcal{F}_p$ may not belong to $\mathcal{F}_p$ (in general it does not): the result must be rounded.

IEEE standard defines 4 different rounding modes:

- rounding towards $+\infty$, or upwards: $\circ_u(x) = \min\{y \in \mathcal{F}_p : y \geqslant x\}$;
- rounding towards $-\infty$, or downwards: $\circ_d(x) = \max\{y \in \mathcal{F}_p : y \leqslant x\}$;
- rounding towards 0: $\circ_z(x) := \circ_u(x)$ if $x < 0$, and to $\circ_d(x)$ otherwise;
- rounding to the nearest even: $\circ_n(x)$ is the element of $\mathcal{F}_p$ that is closest to x. If x is exactly halfway between two consecutive elements of $\mathcal{F}_p$, $\circ_n(x)$ is the one for which the integral significand j is an even number.

IEEE Rounding Modes

The result of an arithmetic operation whose input values belong to $\mathcal{F}_p$ may not belong to $\mathcal{F}_p$ (in general it does not): the result must be rounded.

IEEE standard defines 4 different rounding modes:

- rounding towards $+\infty$, or upwards: $\circ_u(x) = \min\{y \in \mathcal{F}_p : y \geqslant x\}$;
- rounding towards $-\infty$, or downwards: $\circ_d(x) = \max\{y \in \mathcal{F}_p : y \leqslant x\}$;
- rounding towards 0: $\circ_z(x) := \circ_u(x)$ if $x < 0$, and to $\circ_d(x)$ otherwise;
- rounding to the nearest even: $\circ_n(x)$ is the element of $\mathcal{F}_p$ that is closest to x. If x is exactly halfway between two consecutive elements of $\mathcal{F}_p$, $\circ_n(x)$ is the one for which the integral significand j is an even number.

The first three rounding modes are called directed rounding modes.

Chapter 4. Interval Arithmetic, Interval Analysis, Rigorous Polynomial Approximations

Interval Arithmetic is "an arithmetic for inequalities".

Interval Arithmetic is "an arithmetic for inequalities".
Assume for instance that we know that $5 \leqslant a \leqslant 6$ and $10 \leqslant b \leqslant 11$: then of course $50 \leqslant ab \leqslant 66$. We will define a product of real intervals such that

$$[5,6] \times [10,11] = [50,66]$$

that allows for such reasoning.

Interval Arithmetic is "an arithmetic for inequalities".
Assume for instance that we know that $5 \leqslant a \leqslant 6$ and $10 \leqslant b \leqslant 11$: then of course $50 \leqslant ab \leqslant 66$. We will define a product of real intervals such that

$$[5, 6] \times [10, 11] = [50, 66]$$

that allows for such reasoning.

In double precision, compute $x_{k+1} = (x_k)^2$ where $x_0 = 1 - 10^{-19}$.

Chapter 4. Interval Arithmetic, Interval Analysis, Rigorous Polynomial Approximations

Interval Arithmetic is "an arithmetic for inequalities".
Assume for instance that we know that $5 \leqslant a \leqslant 6$ and $10 \leqslant b \leqslant 11$: then of course $50 \leqslant ab \leqslant 66$. We will define a product of real intervals such that

$$[5,6] \times [10,11] = [50,66]$$

that allows for such reasoning.

In double precision, compute $x_{k+1} = (x_k)^2$ where $x_0 = 1 - 10^{-19}$.

Another need for interval arithmetic comes from the roundoff errors that occur when working with finite precision numbers.

Notable applications of interval arithmetic to bring rigor to numerical computations performed on a computer include:

- T. Hales' proof of Kepler's conjecture (see `http://code.google.com/p/flyspeck/`),
- W. Tucker's solution of Smale's 14th problem (see `http://www2.math.uu.se/~warwick/main/thesis.html` and also `http://paulbourke.net/fractals/lorenz/`).

Numerous additional interesting information on the website `http://www.cs.utep.edu/interval-comp/`.

Chapter 4. Interval Arithmetic, Interval Analysis, Rigorous Polynomial Approximations

In this course, we are interested in the use of interval arithmetic for mathematical function evaluation purpose.

In this course, we are interested in the use of interval arithmetic for mathematical function evaluation purpose.
Given $\varepsilon > 0$ and $f : [a, b] \to \mathbb{R}$, we would like to make sure that the evaluation $\widehat{f(x)}$ of f at any value $x \in [a, b]$ is such that

$$|\widehat{f(x)} - f(x)| \leqslant \varepsilon.$$

In this course, we are interested in the use of interval arithmetic for mathematical function evaluation purpose.
Given $\varepsilon > 0$ and $f : [a, b] \to \mathbb{R}$, we would like to make sure that the evaluation $\widehat{f(x)}$ of f at any value $x \in [a, b]$ is such that

$$|\widehat{f(x)} - f(x)| \leqslant \varepsilon.$$

Note that, in practice, one commonly uses relative error $\left|1 - \frac{f(x)}{\widehat{f(x)}}\right|$ rather than absolute error $|\widehat{f(x)} - f(x)|$.

We focus on the absolute error case for the sake of clarity.

To perform the evaluation, we replace f by a polynomial p. Then we evaluate p, and $\widehat{f(x)} = \circ(p(x))$, where $\circ$ is the active rounding mode.

To perform the evaluation, we replace f by a polynomial p. Then we evaluate p, and $\widehat{f(x)} = \circ(p(x))$, where $\circ$ is the active rounding mode.

There are two sources of error:

- *approximation error*: let η_1 be an upper bound for $\|f - p\|_\infty$,
- *rounding error*: let η_2 be an upper bound for the error $|p(x) - \circ(p(x))|$,

we have to guarantee that $\eta_1 + \eta_2 \leqslant \varepsilon$.

To perform the evaluation, we replace f by a polynomial p. Then we evaluate p, and $\widehat{f(x)} = \circ(p(x))$, where $\circ$ is the active rounding mode.

There are two sources of error:

- *approximation error*: let η_1 be an upper bound for $\|f - p\|_\infty$,
- *rounding error*: let η_2 be an upper bound for the error $|p(x) - \circ(p(x))|$,

we have to guarantee that $\eta_1 + \eta_2 \leqslant \varepsilon$.

In this course: tools that help to establish rigorous approximation error.

Regarding rounding errors, G.Melquiond has developed formal proof tools (in Coq) which address this issue (see `http://gappa.gforge.inria.fr/`).

4.1. Interval arithmetic

Definition

(Real interval.) Let $\bar{x}, \underline{x} \in \mathbb{R}$, $\bar{x} \leqslant \underline{x}$*. We define the interval*

$$X = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} : \underline{x} \leqslant x \leqslant \bar{x}\}.$$

The real numbers $\underline{x}$ *and* $\bar{x}$ *are called the endpoints of the interval,* $\underline{x}$ *is its minimum,* $\bar{x}$ *its maximum. The set of all real intervals will be denoted* $\mathbb{IR}$*.*

4.1. Interval arithmetic

Definition

(Real interval.) Let $\bar{x}, \underline{x} \in \mathbb{R}$*,* $\bar{x} \leqslant \underline{x}$*. We define the interval*

$$X = [\underline{x}, \bar{x}] = \{x \in \mathbb{R} : \underline{x} \leqslant x \leqslant \bar{x}\}.$$

The real numbers $\underline{x}$ *and* $\bar{x}$ *are called the endpoints of the interval,* $\underline{x}$ *is its minimum,* $\bar{x}$ *its maximum. The set of all real intervals will be denoted* $\mathbb{IR}$*.*

Definition

Let $x \in \mathbb{IR}$*. The width of* x *is denoted* $w(x) = \bar{x} - \underline{x}$*. We also define the center*

$$\mathrm{mid}(x) = \frac{\underline{x} + \bar{x}}{2},$$

and the radius $\mathrm{rad}(x) = \frac{1}{2} w(x)$*.*

4.1. Interval arithmetic

Remark

It is common in the litterature to encounter the notation $(\mathrm{mid}\,(x)\,, \mathrm{rad}\,(x)) = \{x \in \mathbb{R} : |x - \mathrm{mid}\,(x)| \leqslant \mathrm{rad}\,(x)\}$.

This mid-rad representation is the basis of the so called Ball Arithmetic, cf. the excellent software Arb `http://arblib.org/`.

4.1. Interval arithmetic

Remark

It is common in the litterature to encounter the notation
$(\mathrm{mid}\,(x)\,,\mathrm{rad}\,(x)) = \{x \in \mathbb{R} : |x - \mathrm{mid}\,(x)| \leqslant \mathrm{rad}\,(x)\}$.

This mid-rad representation is the basis of the so called Ball Arithmetic, cf. the excellent software Arb `http://arblib.org/`.

Definition

A point (or degenerate, or thin) interval is one of the form $[x, x]$*, also denoted* $[x]$.

4.1.1. Operations on intervals

We now define basic arithmetic operations on intervals. As you will see, monotonicity plays an essential role for obtaining sharp enclosures.

Definition

Let $X, Y \in \mathbb{IR}$. *Let* $* \in \{+, -, \times, /\}$. *We denote*

$$X * Y = \{x * y; x \in X, y \in Y\}$$

where, if $* = /$, *we assume that* $0 \notin Y$.

4.1.1. Operations on intervals

Proposition

We can compute the $X * Y$ above using formulae such as

$$
\begin{aligned}
[\underline{x}, \bar{x}] + [\underline{y}, \bar{y}] &= [\underline{x} + \underline{y}, \bar{x} + \bar{y}]\,, \\
[\underline{x}, \bar{x}] - [\underline{y}, \bar{y}] &= [\underline{x} - \bar{y}, \bar{x} - \underline{y}]\,, \\
[\underline{x}, \bar{x}] \times [\underline{y}, \bar{y}] &= \left[\min\left(\underline{x}\cdot\underline{y}, \underline{x}\cdot\bar{y}, \bar{x}\cdot\underline{y}, \bar{x}\cdot\bar{y}\right), \max\left(\underline{x}\cdot\underline{y}, \underline{x}\cdot\bar{y}, \bar{x}\cdot\underline{y}, \bar{x}\cdot\bar{y}\right)\right], \\
[\underline{x}, \bar{x}] \,/\, [\underline{y}, \bar{y}] &= [\underline{x}, \bar{x}] \times \left[\frac{1}{\bar{y}}, \frac{1}{\underline{y}}\right] \quad \text{if } 0 \notin Y,
\end{aligned}
$$

which depend only on the endpoints.

Proof.

Exercise. □

4.1.1. Operations on intervals

Remark

Note that, in $\mathbb{IR}$*, the operations* $+$ *and* $\times$ *are associative and commutative.*

Remark

In practice, multiplication (hence division) can be made more efficient (check the signs of the endpoints).

4.1.1. Operations on intervals

Proposition

1. *Interval subtraction is not the inverse of addition.*
2. *Interval division is not the inverse of multiplication.*
3. *Interval multiplication of an interval with itself is not equivalent to "squaring the interval": if* $\underline{x} < 0 < \bar{x}$*,*

$$[\underline{x}, \bar{x}] \times [\underline{x}, \bar{x}] \neq \left[0, \max\left(\underline{x}^2, \bar{x}^2\right)\right].$$

4. *Interval multiplication is sub-distributive wrt addition: for all* $X, Y, Z \in \mathbb{IR}$*, we have*

$$X \times (Y + Z) \subset X \times Y + X \times Z.$$

5. *For all* $X \in \mathbb{IR}$*, we have* $X + [0] = X$ *and* $[0] \times X = [0]$*.*

Proof.

Exercise. □

4.1.1. Operations on intervals

A straightforward yet quite useful statement is the following.

Lemma

(Inclusion isotonicity) If $X \subset X', Y \subset Y', * \in \{+, -, \times, /\}$*, then*

$$X * Y \subset X' * Y'.$$

For division, we assume that $0 \notin Y'$*.*

Proof.

Obvious from Definition . □

4.1.2. Floating-point interval arithmetic

When it comes to implementing interval arithmetic on a computer, we no longer work over $\mathbb{R}$, but in most cases with floating-point numbers.

4.1.2. Floating-point interval arithmetic

When it comes to implementing interval arithmetic on a computer, we no longer work over $\mathbb{R}$, but in most cases with floating-point numbers.

Let $\mathcal{F}$ be the set of machine numbers we are working with. Then we denote

$$\mathbb{I}\mathcal{F} = \{[\underline{x}, \bar{x}] : \underline{x}, \bar{x} \in \mathcal{F}\}.$$

4.1.2. Floating-point interval arithmetic

When it comes to implementing interval arithmetic on a computer, we no longer work over $\mathbb{R}$, but in most cases with floating-point numbers.

Let $\mathcal{F}$ be the set of machine numbers we are working with. Then we denote

$$\mathbb{I}\mathcal{F} = \{[\underline{x}, \bar{x}] : \underline{x}, \bar{x} \in \mathcal{F}\}.$$

Of course the set of floating-point numbers is not arithmetically closed (e.g., the sum of two floating-point numbers is not always a floating-point number).

4.1.2. Floating-point interval arithmetic

When it comes to implementing interval arithmetic on a computer, we no longer work over $\mathbb{R}$, but in most cases with floating-point numbers.

Let $\mathcal{F}$ be the set of machine numbers we are working with. Then we denote

$$\mathbb{I}\mathcal{F} = \{[\underline{x}, \bar{x}] : \underline{x}, \bar{x} \in \mathcal{F}\}.$$

Of course the set of floating-point numbers is not arithmetically closed (e.g., the sum of two floating-point numbers is not always a floating-point number).

When we perform arithmetic operations on intervals in $\mathbb{I}\mathcal{F}$, we need to make sure to "round the resulting interval outwards" in order to guarantee that it contains the "true result".

4.1.2. Floating-point interval arithmetic

For $X, Y \in \mathbb{IF}$, we set

$$
\begin{aligned}
X + Y &= \left[\triangledown\left(\underline{x} + \underline{y}\right), \vartriangle\left(\bar{x} + \bar{y}\right)\right], \\
X - Y &= \left[\triangledown\left(\underline{x} - \bar{y}\right), \vartriangle\left(\bar{x} - \underline{y}\right)\right], \\
X \times Y &= \left[\min\left(\triangledown\left(\underline{x}\cdot\underline{y}\right), \triangledown\left(\underline{x}\cdot\bar{y}\right), \triangledown\left(\bar{x}\cdot\underline{y}\right), \triangledown\left(\bar{x}\cdot\bar{y}\right)\right),\right. \\
&\qquad \left.\max\left(\vartriangle\left(\underline{x}\cdot\underline{y}\right), \vartriangle\left(\underline{x}\cdot\bar{y}\right), \vartriangle\left(\bar{x}\cdot\underline{y}\right), \vartriangle\left(\bar{x}\cdot\bar{y}\right)\right)\right], \\
X / Y &= \left[\min\left(\triangledown\left(\underline{x}/\underline{y}\right), \triangledown\left(\underline{x}/\bar{y}\right), \triangledown\left(\bar{x}/\underline{y}\right), \triangledown\left(\bar{x}/\bar{y}\right)\right),\right. \\
&\qquad \left.\max\left(\vartriangle\left(\underline{x}/\underline{y}\right), \vartriangle\left(\underline{x}/\bar{y}\right), \vartriangle\left(\bar{x}/\underline{y}\right), \vartriangle\left(\bar{x}/\bar{y}\right)\right)\right] \quad if\, 0 \notin Y,
\end{aligned}
$$

where $\triangledown$ and $\vartriangle$ denote rounding to $-\infty$ and $+\infty$ respectively.

4.1.2. Floating-point interval arithmetic

Remark

Standard machine floating-point numbers are not always sufficient, e.g., to work with very small intervals. We may also use multiple-precision floating-point numbers as bounds for our intervals. An example of a library which offers support for multiple precision interval arithmetic is MPFR[1].

[1] http://www.mpfr.org

4.2. Interval functions

Definition

Let $D \subset \mathbb{R}$, and let $f : D \to \mathbb{R}$. We denote

$$R(f, D) = \{f(x) : x \in D\}$$

the range of f over D.

4.2. Interval functions

Definition

Let $D \subset \mathbb{R}$*, and let* $f : D \to \mathbb{R}$*. We denote*

$$R(f, D) = \{f(x) : x \in D\}$$

the range of f *over* D*.*

Remark

Finding the exact image of a (usually multivariate) function, and, in particular, a value where f *attains its minimum is a whole subdomain of Math and CS called Global Optimization.*

4.2. Interval functions

Let $X = [\underline{x}, \bar{x}] \in \mathbb{IR}$. By monotonicity, interval functions defined as follows give the exact range of the corresponding real functions:

$$\begin{aligned} e^X &= [\exp \underline{x}, \exp \bar{x}]\,, \\ \sqrt{X} &= \left[\sqrt{\underline{x}}, \sqrt{\bar{x}}\right], & \underline{x} \geqslant 0, \\ \log X &= [\log \underline{x}, \log \bar{x}]\,, & \underline{x} > 0, \\ \arctan X &= [\arctan \underline{x}, \arctan \bar{x}]\,, \end{aligned}$$

4.2. Interval functions

For some other functions like x^n, trigonometric functions..., writing down $R(f, D)$ is also possible, as long as we know their extrema. For instance, let $n \in \mathbb{Z}$, $X \in \mathbb{IR}$,

$$X^n = \text{pow}(X, n) = \begin{cases} \text{if } n \in 2\mathbb{N}+1, [\underline{x}^n, \bar{x}^n] \\ \text{if } n \in \mathbb{N} \setminus \{0\}, n \text{ even}, \\ \qquad\qquad [\min(\underline{x}^n, \bar{x}^n), \max(\underline{x}^n, \bar{x}^n)] \text{ if } 0 \notin X, \\ \qquad\qquad [0, \max(\underline{x}^n, \bar{x}^n)] \text{ otherwise}, \\ [1, 1] \text{ if } n = 0, \\ [1/\bar{x}, 1/\underline{x}]^{-n} \text{ if } -n \in \mathbb{N} \text{ and } 0 \notin X. \end{cases}$$

4.2. Interval functions

Exercise

Write the analogous formulas for sin, cos, tan. For sin and tan, consider

$$S_1^+ = \left\{2k\pi + \frac{\pi}{2}, k \in \mathbb{Z}\right\}, \quad S_1^- = \left\{2k\pi - \frac{\pi}{2}, k \in \mathbb{Z}\right\}.$$

For cos, consider

$$S_2^+ = \{2k\pi, k \in \mathbb{Z}\}, \quad S_2^- = \{2k\pi + \pi, k \in \mathbb{Z}\}.$$

4.2. Interval functions

The example of $f(x) = x^2 - x + 1$ over $[0, 2]$ illustrates two important issues:

- overestimation;
- dependency on the way the function is written.

4.2. Interval functions

The example of $f(x) = x^2 - x + 1$ over $[0, 2]$ illustrates two important issues:

- overestimation;
- dependency on the way the function is written.

We have $f(x) \in [0, 2]^2 - [0, 2] + [1] = [0, 4] + [-2, 0] + [1] = [-1, 5]$.

4.2. Interval functions

The example of $f(x) = x^2 - x + 1$ over $[0,2]$ illustrates two important issues:

- overestimation;
- dependency on the way the function is written.

We have $f(x) \in [0,2]^2 - [0,2] + [1] = [0,4] + [-2,0] + [1] = [-1,5]$.

Now write $f(x) = x(x-1) + 1$. We have
$f(x) \in [0,2]\,[-1,1] + [1] = [-2,2] + [1,1] = [-1,3]$.

4.2. Interval functions

The example of $f(x) = x^2 - x + 1$ over $[0, 2]$ illustrates two important issues:

- overestimation;
- dependency on the way the function is written.

We have $f(x) \in [0, 2]^2 - [0, 2] + [1] = [0, 4] + [-2, 0] + [1] = [-1, 5]$.

Now write $f(x) = x(x - 1) + 1$. We have
$f(x) \in [0, 2] [-1, 1] + [1] = [-2, 2] + [1, 1] = [-1, 3]$.

Actually, $R(f, [0, 2]) = [3/4, 3]$.

4.2. Interval functions

Definition

(Interval extension.) Let $X \in \mathbb{IR}$, and let $f : X \to \mathbb{R}$. A function $\tilde{f} : \mathbb{IR} \to \mathbb{IR}$ is called an interval extension of f over X if:

- *for all $x \in X$, $R(f, \{x\}) = \tilde{f}([x, x])$,*
- *for all $Y \subset \mathbb{IR}$ with $Y \subset X$, we have*

$$R(f, Y) \subset \tilde{f}(Y).$$

Several interval extensions are possible for the same function over the same X. Interval extensions of $\exp$ over $[-1, 1]$ include

- the function $[\underline{x}, \bar{x}] \mapsto [e^{\underline{x}}, e^{\bar{x}}]$.
- but also?

4.2. Interval functions

Let's try to propose a systematic process for computing interval extensions.

4.2. Interval functions

Let's try to propose a systematic process for computing interval extensions.

If $f(x)$ is a rational *expression*, one means to get an interval extension of the function it denotes is to replace each occurrence of the variable x by the interval X, and "overload" all arithmetic operations with interval operations. The resulting extension is called *the natural interval extension*.

4.2. Interval functions

Let's try to propose a systematic process for computing interval extensions.

If $f(x)$ is a rational *expression*, one means to get an interval extension of the function it denotes is to replace each occurrence of the variable x by the interval X, and "overload" all arithmetic operations with interval operations. The resulting extension is called *the natural interval extension*.

Theorem

Given a rational expression denoting a real-valued function f, and its natural interval extension F, which we assume to be well-defined over some interval $X \in \mathbb{IR}$, then

1. $Z \subset Z' \subset X$ *implies* $F(Z) \subset F(Z')$ *(inclusion isotonicity);*
2. $R(f, X) \subset F(X)$ *(range enclosure).*

4.2. Interval functions

We now would like to extend this notion of natural interval extension to a larger class of functions.

Definition

We call basic (or standard) functions the elements of

$$\mathfrak{S} = \left\{ \sin, \cos, \exp, \tan, \log, x^{p/q}, \ldots \right\}$$

for which we can determine the exact range over a given interval based on a simple rule.

These functions are said to have a sharp interval enclosure.

4.2. Interval functions

We now would like to extend this notion of natural interval extension to a larger class of functions.

Definition

We call basic (or standard) functions the elements of

$$\mathfrak{S} = \left\{ \sin, \cos, \exp, \tan, \log, x^{p/q}, \ldots \right\}$$

for which we can determine the exact range over a given interval based on a simple rule.

These functions are said to have a sharp interval enclosure.

Definition

We call elementary function a symbolic expression built from constants and basic functions using arithmetic operations and composition. The class of elementary functions will be denoted $\mathcal{E}$. A function $f \in \mathcal{E}$ is given by an expression tree (or dag, for directed acyclic graph).

4.2. Interval functions

Definition

An interval valued function $F : X \cap \mathbb{IR} \to \mathbb{IR}$ *is inclusion isotonic over* $X \in \mathbb{IR}$ *if* $Z \subset Z' \subset X$ *implies* $F(Z) \subset F(Z')$.

4.2. Interval functions

Definition

An interval valued function $F : X \cap \mathbb{IR} \to \mathbb{IR}$ *is inclusion isotonic over* $X \in \mathbb{IR}$ *if* $Z \subset Z' \subset X$ *implies* $F(Z) \subset F(Z')$.

Theorem

Given an elementary function f *and an interval* X *over which the natural interval extension* F *of* f *is well-defined:*

1. F *is inclusion isotonic over* X;
2. $R(f, X) \subset F(X)$.

4.2. Interval functions

Example

Consider

$$f(x) = \left(\cos x - x^3 + x\right)(\tan x + 1/2)$$

over $[0, \pi/4]$*. To show that* f *has no zero in this range, we compute the natural interval extension*

$$f([0, \pi/4]) = \left[\frac{\sqrt{2}}{2} - \frac{\pi^3}{64}, 1 + \frac{\pi}{4}\right]\left[\frac{1}{2}, \frac{3}{2}\right] \subset [0.11, 2.68].$$

4.2. Interval functions

Example

Consider

$$f(x) = \left(\cos x - x^3 + x\right)(\tan x + 1/2)$$

over $[0, \pi/4]$*. To show that* f *has no zero in this range, we compute the natural interval extension*

$$f([0, \pi/4]) = \left[\frac{\sqrt{2}}{2} - \frac{\pi^3}{64}, 1 + \frac{\pi}{4}\right]\left[\frac{1}{2}, \frac{3}{2}\right] \subset [0.11, 2.68].$$

Exercise

Show that $f(x) = x - \sin x + 2/5$ *has no zero over* $[0, \pi/4]$*.*

4.2. Interval functions

Theorem

Let $X \in \mathbb{IR}$. Let f be an elementary function such that any subexpression of f is Lipschitz continuous. Let F be an inclusion isotonic interval extension such that $F(X)$ is well-defined. Then, there exists $\kappa > 0$, depending on F and X, such that, if $X = \bigcup_{i=1}^{k} X_i$, with $X_i \in \mathbb{IR}$ for all i,then

$$R(f, X) \subset \bigcup_{i=1}^{k} F(X_i) \subset F(X)$$

and

$$\operatorname{rad}\left(\bigcup_{i=1}^{k} F(X_i)\right) \leqslant \operatorname{rad}(R(f, X)) + \kappa \max_{i=1,\ldots,k} \operatorname{rad} X_i.$$

4.2. Interval functions

However, the number of subdivisions needed may be very large.

Example

Let $f(x) = e^{1/\cos x}$, *and let* p *be a degree-10 minimax approximation of* f *over* $[0, 1]$. *Let*

$$\varepsilon(x) = f(x) - p(x).$$

Using the natural interval extension of ε, *we get* $\|\varepsilon\| \leqslant 298$. *But one can show that obtaining the actual value* $\|\varepsilon\| \approx 3.8325 \cdot 10^{-5}$ *by subdivision would require about* 10^7 *subintervals.*

Approximation Theory and Proof Assistants: Certified Computations

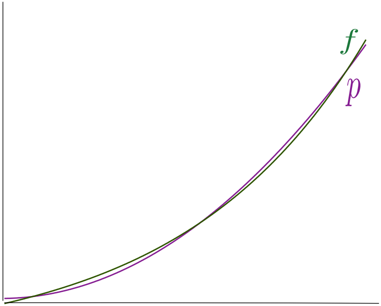
Nicolas Brisebarre and Damien Pous

Master 2 Informatique Fondamentale
École Normale Supérieure de Lyon, 2020-2021

Chapter 5. Rigorous Polynomial Approximations

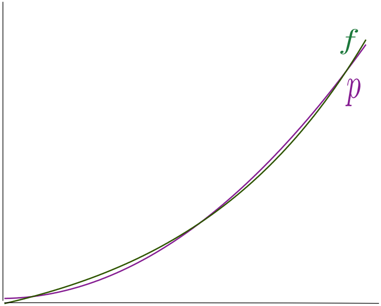
When Interval Arithmetic does not suffice: Computing supremum norms of approximation errors

$f(x) = e^{1/\cos(x)},\ x \in [0,1],\ p(x) = \sum_{i=0}^{10} c_i x^i,$



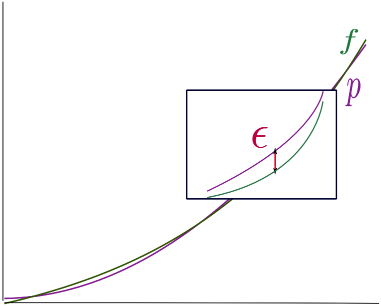
When Interval Arithmetic does not suffice: Computing supremum norms of approximation errors

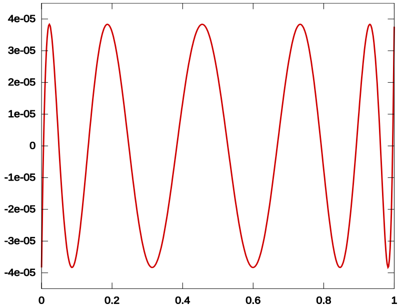
$f(x) = e^{1/\cos(x)},\ x \in [0, 1],\ p(x) = \sum_{i=0}^{10} c_i x^i,\ \varepsilon(x) = f(x) - p(x)$



When Interval Arithmetic does not suffice: Computing supremum norms of approximation errors

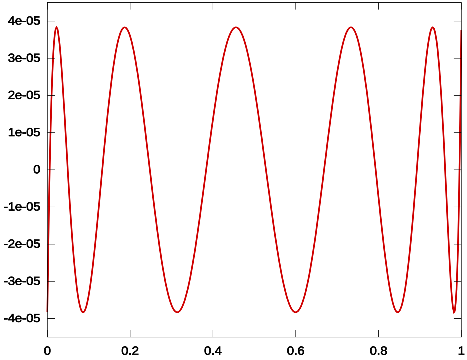
$f(x) = e^{1/\cos(x)}$, $x \in [0,1]$, $p(x) = \sum_{i=0}^{10} c_i x^i$, $\varepsilon(x) = f(x) - p(x)$ s.t. $\|\varepsilon\|_\infty = \sup_{x \in [a,\, b]} \{|\varepsilon(x)|\}$ is as small as possible (Remez algorithm)





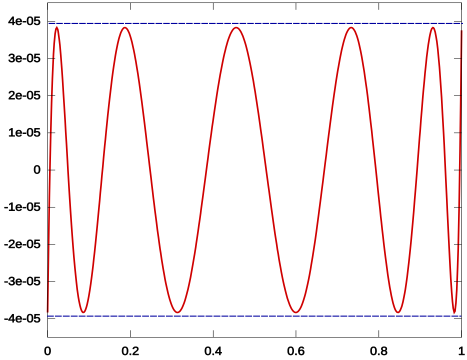
When Interval Arithmetic does not suffice: Computing supremum norms of approximation errors

$f(x) = e^{1/\cos(x)},\ x \in [0,1],\ p(x) = \sum_{i=0}^{10} c_i x^i$, $\varepsilon(x) = f(x) - p(x)$ s.t. $\|\varepsilon\|_\infty = \sup_{x \in [a,\, b]} \{|\varepsilon(x)|\}$ is as small as possible (Remez algorithm)



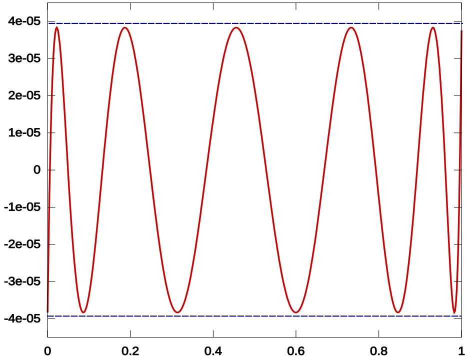
When Interval Arithmetic does not suffice: Computing supremum norms of approximation errors

$f(x) = e^{1/\cos(x)},\ x \in [0,1],\ \ p(x) = \sum_{i=0}^{10} c_i x^i$, $\varepsilon(x) = f(x) - p(x)$ s.t. $\|\varepsilon\|_\infty = \sup_{x\in[a,\,b]}\{|\varepsilon(x)|\}$ is as small as possible (Remez algorithm)



When Interval Arithmetic does not suffice: Computing supremum norms of approximation errors

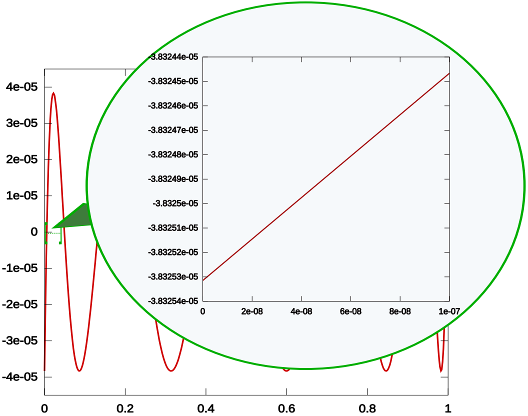
$f(x) = e^{1/\cos(x)}$, $x \in [0, 1]$, $p(x) = \sum_{i=0}^{10} c_i x^i$, $\varepsilon(x) = f(x) - p(x)$ s.t. $\|\varepsilon\|_\infty = \sup_{x \in [a,\, b]}\{|\varepsilon(x)|\}$ is as small as possible (Remez algorithm)



Using IA, $\varepsilon(x) \in [-233, 298]$, but $\|\varepsilon(x)\|_\infty \simeq 3.8325 \cdot 10^{-5}$

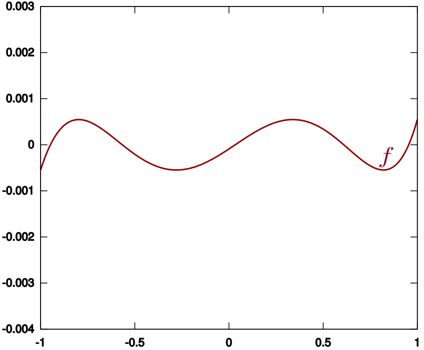
Why IA does not suffice: Overestimation

Overestimation can be reduced by using intervals of smaller width.



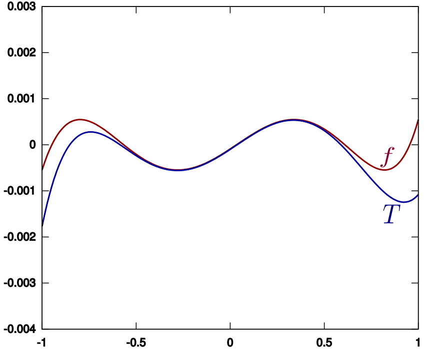
In this case, over $[0, 1]$ we need 10^7 intervals!

Rigorous polynomial approximations



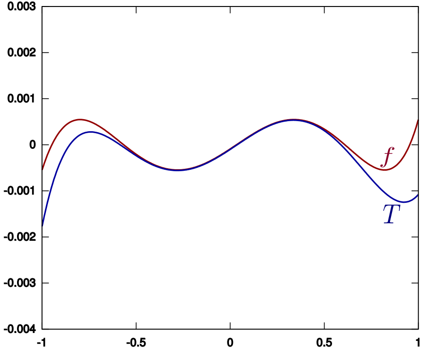
Rigorous polynomial approximations

f replaced with
- polynomial approximation T of degree n



Rigorous polynomial approximations

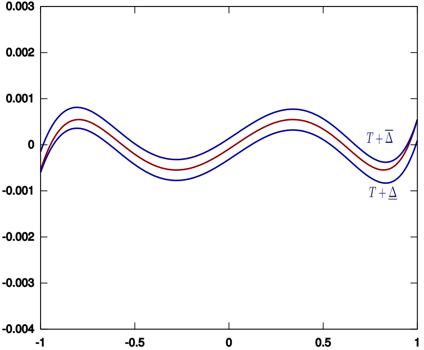
f replaced with
- polynomial approximation T of degree n
- interval $\boldsymbol{\Delta}$ s. t. $f(x) - T(x) \in \boldsymbol{\Delta}, \forall x \in [a, b]$



Rigorous polynomial approximations

f replaced with a rigorous polynomial approximation : $(T, \mathbf{\Delta})$

- polynomial approximation T of degree n
- interval $\mathbf{\Delta}$ s. t. $f(x) - T(x) \in \mathbf{\Delta}, \forall x \in [a, b]$



How to compute $(T, \mathbf{\Delta})$?

Chebyshev Models

Over $[-1,1]$, Chebyshev polynomials: $T_n(x) = \cos\left(n \arccos x\right), n \geqslant 0$.

Chebyshev Models

Over $[-1,1]$, Chebyshev polynomials: $T_n(x) = \cos\left(n \arccos x\right), n \geqslant 0$.

Let $I = [a,b]$, we define Chebyshev polynomials over I as

$$T_n^{[a,b]}(x) = T_n\left(\frac{2x-b-a}{b-a}\right).$$

Chebyshev Models

Over $[-1,1]$, Chebyshev polynomials: $T_n(x) = \cos\left(n \arccos x\right), n \geqslant 0$.

Let $I = [a,b]$, we define Chebyshev polynomials over I as

$$T_n^{[a,b]}(x) = T_n\left(\frac{2x-b-a}{b-a}\right).$$

$T_{n+1}^{[a,b]}$ has $n+1$ distinct real roots in $[a,b]$ (Chebyshev nodes of the first kind):

$$\mu_k = \frac{a+b}{2} + \frac{b-a}{2}\cos\left(\frac{(k+1/2)\,\pi}{n+1}\right),\ k = 0,\dots,n.$$

Chebyshev Models

We recall

Lemma 1

The polynomial $W_{\overline{\mu}}(x) = \prod_{k=0}^{n} (x - \mu_k)$*, is the monic degree-*$(n+1)$ *polynomial that minimizes the supremum norm over* $[a, b]$ *of all monic polynomials in* $\mathbb{C}[x]$ *of degree at most* $n+1$*. We have*

$$W_{\overline{\mu}}(x) = \frac{(b-a)^{n+1}}{2^{2n+1}} T_{n+1}^{[a,b]}(x)$$

and

$$\max_{x \in [a,b]} |W_{\overline{\mu}}(x)| = \frac{(b-a)^{n+1}}{2^{2n+1}}.$$

Chebyshev Models

Lemma 2

(Taylor-Lagrange-like formula.) Let $n \in \mathbb{N}$, *and let* $f \in \mathcal{C}^{n+1}([a,b])$. *Let* $P \in \mathbb{R}_n[X]$ *be the interpolation polynomial of* f *at the Chebyshev nodes* $(\mu_k)_{0 \leqslant k \leqslant n}$. *For all* $x \in [a,b]$, *there exists* $\xi_x \in (a,b)$ *such that*

$$f(x) = P(x) + \frac{(b-a)^{n+1} f^{(n+1)}(\xi_x)}{2^{2n+1}} T_{n+1}^{[a,b]}(x).$$

Chebyshev Models - How do we obtain them?

Let $n \in \mathbb{N}$, $n+1$ times differentiable function f over $[a, b]$.

Chebyshev Models - How do we obtain them?

Let $n \in \mathbb{N}$, $n+1$ times differentiable function f over $[a,b]$.

- $f(x) = \underbrace{\sum_{k=0}^{n} p_k T_k^{[a,b]}(x)}_{T(x)} + \underbrace{\Delta_n(x,\xi)}_{\text{remainder}}$
- $\Delta_n(x,\xi) = \frac{(b-a)^{n+1} f^{(n+1)}(\xi_x)}{2^{2n+1}} T_{n+1}^{[a,b]}(x)$, $x \in [a,b]$, ξ lies strictly between a and b

Chebyshev Models - How do we obtain them?

Let $n \in \mathbb{N}$, $n+1$ times differentiable function f over $[a,b]$.

- $$f(x) = \underbrace{\sum_{k=0}^{n} p_k T_k^{[a,b]}(x)}_{T(x)} + \underbrace{\Delta_n(x,\xi)}_{\text{remainder}}$$
- $\Delta_n(x,\xi) = \frac{(b-a)^{n+1} f^{(n+1)}(\xi_x)}{2^{2n+1}} T_{n+1}^{[a,b]}(x)$, $x \in [a,b]$, ξ lies strictly between a and b

- How to compute the coefficients p_i of $T(x)$?
- How to compute an interval enclosure $\boldsymbol{\Delta}$ for $\Delta_n(x,\xi)$?

Chebyshev Models: computations of the coefficients

$$P(x) = \sum_{i=0}^{n} p_i T_i^{[a,b]}(x), \text{ with } p_i = \sum_{k=0}^{n} \tfrac{2}{n+1} f(\mu_k) T_i^{[a,b]}(\mu_k).$$

Reminder: Clenshaw's method for evaluating Chebyshev sums

Algorithm

Input Chebyshev coefficients $c_0, \ldots, c_N$, a point t

Output $\sum_{k=0}^{N} c_k T_k(t)$

1. $b_{N+1} \leftarrow 0$, $b_N \leftarrow c_N$
2. for $k = N-1, N-2, \ldots, 1$
    1. $b_k \leftarrow 2tb_{k+1} - b_{k+2} + c_k$
3. return $c_0 + tb_1 - b_2$

This algorithm runs in $O(N)$ arithmetic operations.

Reminder: Clenshaw's method for evaluating Chebyshev sums

Algorithm

Input Chebyshev coefficients $c_0, \ldots, c_N$, a point t

Output $\sum_{k=0}^{N} c_k T_k(t)$

1. $b_{N+1} \leftarrow 0$, $b_N \leftarrow c_N$
2. for $k = N-1, N-2, \ldots, 1$
    1. $b_k \leftarrow 2tb_{k+1} - b_{k+2} + c_k$
3. return $c_0 + tb_1 - b_2$

This algorithm runs in $O(N)$ arithmetic operations.

It works also if t and the c_k's are intervals!

Chebyshev Models: computations of the coefficients

$$P(x) = \sum_{i=0}^{n} p_i T_i^{[a,b]}(x), \text{ with } p_i = \sum_{k=0}^{n} \frac{2}{n+1} f(\mu_k) T_i^{[a,b]}(\mu_k).$$

Chebyshev Models: computations of the coefficients

$$P(x) = \sum_{i=0}^{n} p_i T_i^{[a,b]}(x), \text{ with } p_i = \sum_{k=0}^{n} \tfrac{2}{n+1} f(\mu_k) T_i^{[a,b]}(\mu_k).$$

We replace the μ_k's and the $f(\mu_k)$'s with interval enclosures, and then perform an interval evaluation with Clenshaw's method

Chebyshev Models: computations of the coefficients

$$P(x) = \sum_{i=0}^{n} p_i T_i^{[a,b]}(x), \text{ with } p_i = \sum_{k=0}^{n} \tfrac{2}{n+1} f(\mu_k) T_i^{[a,b]}(\mu_k).$$

We replace the μ_k's and the $f(\mu_k)$'s with interval enclosures, and then perform an interval evaluation with Clenshaw's method: the coefficients p_i are intervals.

Chebyshev Models: bounding the remainder

$\Delta_n(x,\xi) = \frac{(b-a)^{n+1} f^{(n+1)}(\xi_x)}{2^{2n+1}} T_{n+1}^{[a,b]}(x)$, $x \in [a,b]$, ξ lies strictly between a and b.

Chebyshev Models: bounding the remainder

$\Delta_n(x,\xi) = \frac{(b-a)^{n+1} f^{(n+1)}(\xi_x)}{2^{2n+1}} T_{n+1}^{[a,b]}(x)$, $x \in [a,b]$, ξ lies strictly between a and b.

$|\Delta_n(x,\xi)|$ is bounded by $\frac{(b-a)^{n+1}\left|f^{(n+1)}([a,b])\right|}{2^{2n+1}}$.

Chebyshev Models: bounding the remainder

$\Delta_n(x,\xi) = \frac{(b-a)^{n+1} f^{(n+1)}(\xi_x)}{2^{2n+1}} T_{n+1}^{[a,b]}(x)$, $x \in [a,b]$, ξ lies strictly between a and b.

$|\Delta_n(x,\xi)|$ is bounded by $\frac{(b-a)^{n+1} \left| f^{(n+1)}([a,b]) \right|}{2^{2n+1}}$.

If f satisfies a differential equation with polynomial coefficients: fairly easy to retrieve an upper bound for $\left| f^{(n+1)}([a,b]) \right|$.

Chebyshev Models: bounding the remainder

$\Delta_n(x,\xi) = \frac{(b-a)^{n+1} f^{(n+1)}(\xi_x)}{2^{2n+1}} T_{n+1}^{[a,b]}(x)$, $x \in [a,b]$, ξ lies strictly between a and b.

$|\Delta_n(x,\xi)|$ is bounded by $\frac{(b-a)^{n+1} \left| f^{(n+1)}([a,b]) \right|}{2^{2n+1}}$.

If f satisfies a differential equation with polynomial coefficients: fairly easy to retrieve an upper bound for $\left| f^{(n+1)}([a,b]) \right|$.

Otherwise?

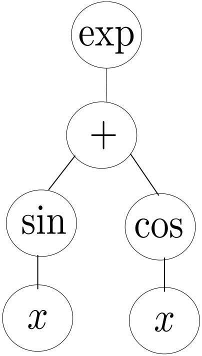
Chebyshev Models "Philosophy"

For bounding the remainders:

- For "basic functions" use Taylor-Lagrange-like statement.
- For "composite functions" use a two-step procedure:
  - compute models $(T, \mathbf{\Delta})$ for all basic functions;
  - apply algebraic rules with these models, instead of operations with the corresponding functions.

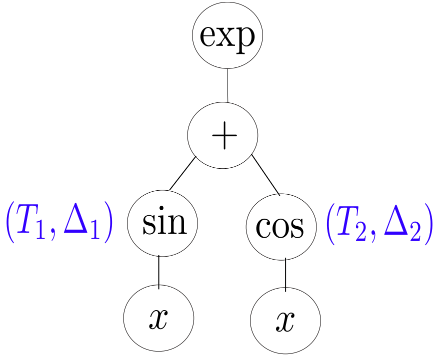
Chebyshev Models - Two-step procedure

Example: $f_{\text{comp}}(x) = \exp(\sin(x) + \cos(x))$



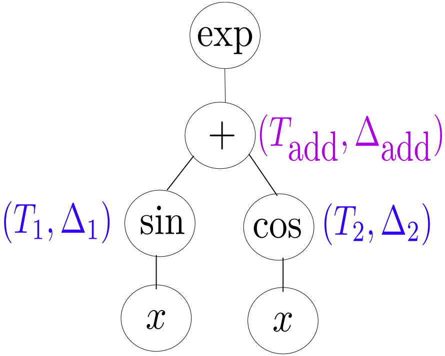
Chebyshev Models - Two-step procedure

Example: $f_{\text{comp}}(x) = \exp(\sin(x) + \cos(x))$



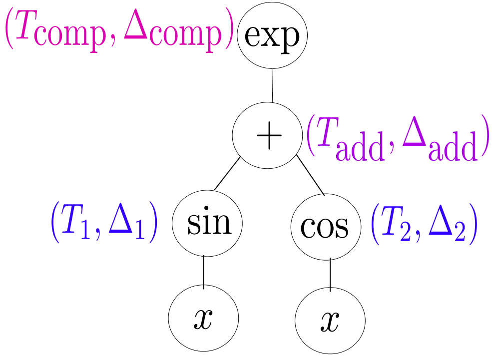
Chebyshev Models - Two-step procedure

Example: $f_{\text{comp}}(x) = \exp(\sin(x) + \cos(x))$



Chebyshev Models - Two-step procedure

Example: $f_{\text{comp}}(x) = \exp(\sin(x) + \cos(x))$



Chebyshev Models - Operations: Addition

Given two Chebyshev Models for f_1 and f_2, over $[a, b]$, degree n:
$f_1(x) - P_1(x) \in \mathbf{\Delta}_1$ and $f_2(x) - P_2(x) \in \mathbf{\Delta}_2$, $\forall x \in [a, b]$.

Addition
$(P_1, \mathbf{\Delta}_1) + (P_2, \mathbf{\Delta}_2) = (P_1 + P_2, \mathbf{\Delta}_1 + \mathbf{\Delta}_2)$.

Chebyshev Models - Operations: Multiplication

For multiplication, we have: $T_m^{[a,b]}(x) \cdot T_n^{[a,b]}(x) = \dfrac{T_{m+n}^{[a,b]} + T_{|m-n|}^{[a,b]}}{2}$.

Chebyshev Models - Operations: Multiplication

For multiplication, we have: $T_m^{[a,b]}(x) \cdot T_n^{[a,b]}(x) = \dfrac{T_{m+n}^{[a,b]} + T_{|m-n|}^{[a,b]}}{2}$.

Consider $P(x) = \sum_{i=0}^{n} p_i T_i^{[a,b]}(x)$ and $Q(x) = \sum_{i=0}^{n} q_i T_i^{[a,b]}(x)$.

We have $P(x) \cdot Q(x) = \sum_{k=0}^{2n} c_k T_k^{[a,b]}(x)$, where

$$c_k = \left(\sum_{|i-j|=k} p_i q_j + \sum_{i+j=k} p_i q_j \right) /2.$$

The cost is $O(n^2)$ operations.

Chebyshev Models - Operations: Multiplication

Given two Chebyshev Models for f_1 and f_2, over $[a, b]$, degree n:
$f_1(x) - P_1(x) \in \mathbf{\Delta}_1$ and $f_2(x) - P_2(x) \in \mathbf{\Delta}_2$, $\forall x \in [a, b]$.

Multiplication
We need algebraic rule for: $(P_1, \mathbf{\Delta}_1) \cdot (P_2, \mathbf{\Delta}_2) = (P, \mathbf{\Delta})$ s.t.
$f_1(x) \cdot f_2(x) - P(x) \in \mathbf{\Delta}$, $\forall x \in [a, b]$

Chebyshev Models - Operations: Multiplication

Given two Chebyshev Models for f_1 and f_2, over $[a, b]$, degree n:
$f_1(x) - P_1(x) \in \mathbf{\Delta}_1$ and $f_2(x) - P_2(x) \in \mathbf{\Delta}_2$, $\forall x \in [a, b]$.

Multiplication
We need algebraic rule for: $(P_1, \mathbf{\Delta}_1) \cdot (P_2, \mathbf{\Delta}_2) = (P, \mathbf{\Delta})$ s.t.
$f_1(x) \cdot f_2(x) - P(x) \in \mathbf{\Delta}$, $\forall x \in [a, b]$

$$f_1(x) \cdot f_2(x) \in \underbrace{P_1(x) \cdot P_2(x)} + \underbrace{\boldsymbol{P_2} \cdot \mathbf{\Delta}_1 + \boldsymbol{P_1} \cdot \mathbf{\Delta}_2 + \mathbf{\Delta}_1 \cdot \mathbf{\Delta}_2}_{\boldsymbol{I_2}}.$$

$$\underbrace{(P_1(x) \cdot P_2(x))_{0 \dots n}}_{P(x)} + \underbrace{(P_1(x) \cdot P_2(x))_{n+1 \dots 2n}}_{\boldsymbol{I_1}}$$

$$\mathbf{\Delta} = \boldsymbol{I_1} + \boldsymbol{I_2}$$

Chebyshev Models - Operations: Multiplication

Given two Chebyshev Models for f_1 and f_2, over $[a,b]$, degree n:
$f_1(x) - P_1(x) \in \mathbf{\Delta}_1$ and $f_2(x) - P_2(x) \in \mathbf{\Delta}_2$, $\forall x \in [a,b]$.

Multiplication
We need algebraic rule for: $(P_1, \mathbf{\Delta}_1) \cdot (P_2, \mathbf{\Delta}_2) = (P, \mathbf{\Delta})$ s.t.
$f_1(x) \cdot f_2(x) - P(x) \in \mathbf{\Delta}$, $\forall x \in [a,b]$

$$f_1(x) \cdot f_2(x) \in \underbrace{P_1(x) \cdot P_2(x)} + \underbrace{\boldsymbol{P_2} \cdot \mathbf{\Delta}_1 + \boldsymbol{P_1} \cdot \mathbf{\Delta}_2 + \mathbf{\Delta}_1 \cdot \mathbf{\Delta}_2}_{\boldsymbol{I_2}}.$$

$$\underbrace{(P_1(x) \cdot P_2(x))_{0\dots n}}_{P(x)} + \underbrace{(P_1(x) \cdot P_2(x))_{n+1\dots 2n}}_{\boldsymbol{I_1}}$$

$$\mathbf{\Delta} = \boldsymbol{I_1} + \boldsymbol{I_2}$$

In our case, for bounding "$\boldsymbol{P}$s": IA evaluation.

Chebyshev Models - Operations: Composition

Given CMs for f_1 over $[c, d]$, for f_2 over $[a, b]$, degree n:
$f_1(y) - P_1(y) \in \mathbf{\Delta}_1$, $\forall y \in [c, d]$ and $f_2(x) - P_2(x) \in \mathbf{\Delta}_2$, $\forall x \in [a, b]$.

Chebyshev Models - Operations: Composition

Given CMs for f_1 over $[c, d]$, for f_2 over $[a, b]$, degree n:
$f_1(y) - P_1(y) \in \mathbf{\Delta}_1$, $\forall y \in [c, d]$ and $f_2(x) - P_2(x) \in \mathbf{\Delta}_2$, $\forall x \in [a, b]$.

Remark: $(f_1 \circ f_2)(x)$ is f_1 evaluated at $y = f_2(x)$.
We need: $f_2([a, b]) \subseteq [c, d]$, checked by $\boldsymbol{P_2} + \mathbf{\Delta}_2 \subseteq [c, d]$

Chebyshev Models - Operations: Composition

Given CMs for f_1 over $[c, d]$, for f_2 over $[a, b]$, degree n:
$f_1(y) - P_1(y) \in \mathbf{\Delta}_1$, $\forall y \in [c, d]$ and $f_2(x) - P_2(x) \in \mathbf{\Delta}_2$, $\forall x \in [a, b]$.

Remark: $(f_1 \circ f_2)(x)$ is f_1 evaluated at $y = f_2(x)$.
We need: $f_2([a, b]) \subseteq [c, d]$, checked by $\boldsymbol{P_2} + \mathbf{\Delta}_2 \subseteq [c, d]$

$f_1(y) \in P_1(y) + \mathbf{\Delta}_1$

Chebyshev Models - Operations: Composition

Given CMs for f_1 over $[c, d]$, for f_2 over $[a, b]$, degree n:
$f_1(y) - P_1(y) \in \mathbf{\Delta}_1$, $\forall y \in [c, d]$ and $f_2(x) - P_2(x) \in \mathbf{\Delta}_2$, $\forall x \in [a, b]$.

Remark: $(f_1 \circ f_2)(x)$ is f_1 evaluated at $y = f_2(x)$.
We need: $f_2([a, b]) \subseteq [c, d]$, checked by $\boldsymbol{P_2} + \mathbf{\Delta}_2 \subseteq [c, d]$

$f_1(f_2(x)) \in P_1(f_2(x)) + \mathbf{\Delta}_1$

Chebyshev Models - Operations: Composition

Given CMs for f_1 over $[c,d]$, for f_2 over $[a,b]$, degree n:
$f_1(y) - P_1(y) \in \boldsymbol{\Delta}_1$, $\forall y \in [c,d]$ and $f_2(x) - P_2(x) \in \boldsymbol{\Delta}_2$, $\forall x \in [a,b]$.

Remark: $(f_1 \circ f_2)(x)$ is f_1 evaluated at $y = f_2(x)$.
We need: $f_2([a,b]) \subseteq [c,d]$, checked by $\boldsymbol{P_2} + \boldsymbol{\Delta}_2 \subseteq [c,d]$

$f_1(f_2(x)) \in P_1(P_2(x) + \boldsymbol{\Delta}_2) + \boldsymbol{\Delta}_1$

Extract polynomial and remainder: P_1 can be evaluated using only additions and multiplications: Clenshaw's algorithm

Ranges of polynomials

Observe that we heavily used enclosures of ranges of polynomials. This raises (at least) two questions:

Ranges of polynomials

Observe that we heavily used enclosures of ranges of polynomials. This raises (at least) two questions:

- How do we compute these enclosures?
- why would this process yield tight enclosures?

Ranges of polynomials - How do we compute these enclosures?

- A first option: let $p(x) = a_0 + a_1 T_1^{[a,b]}(x) + \cdots + a_n T_n^{[a,b]}(x)$, as, $p(I)$ is bounded by $p(x) = |a_0| + |a_1| + \cdots + |a_n|$.

Ranges of polynomials - How do we compute these enclosures?

- A first option: let $p(x) = a_0 + a_1 T_1^{[a,b]}(x) + \cdots + a_n T_n^{[a,b]}(x)$, as, $p(I)$ is bounded by $p(x) = |a_0| + |a_1| + \cdots + |a_n|$.
- Another possibility is to use Bernstein's basis: indeed, one can show that if

  $$p(x) = \sum_{k=0}^{n} p_k B_{n,k}(x),$$

  then for all $x \in [0, 1]$, we have

  $$\min_{[0,1]} p \geqslant \min_k p_k \qquad \text{and} \qquad \max_{[0,1]} p \leqslant \max_k p_k.$$

Ranges of polynomials - How do we compute these enclosures?

- A first option: let $p(x) = a_0 + a_1 T_1^{[a,b]}(x) + \cdots + a_n T_n^{[a,b]}(x)$, as, $p(I)$ is bounded by $p(x) = |a_0| + |a_1| + \cdots + |a_n|$.
- Another possibility is to use Bernstein's basis: indeed, one can show that if

$$p(x) = \sum_{k=0}^{n} p_k B_{n,k}(x),$$

then for all $x \in [0,1]$, we have

$$\min_{[0,1]} p \geqslant \min_k p_k \qquad \text{and} \qquad \max_{[0,1]} p \leqslant \max_k p_k.$$

Warning: need for a conversion algorithm (cost in $O(M(n))$). Problems of stability.

Ranges of polynomials - How do we compute these enclosures?

- A first option: let $p(x) = a_0 + a_1 T_1^{[a,b]}(x) + \cdots + a_n T_n^{[a,b]}(x)$, as, $p(I)$ is bounded by $p(x) = |a_0| + |a_1| + \cdots + |a_n|$.
- Another possibility is to use Bernstein's basis: indeed, one can show that if

$$p(x) = \sum_{k=0}^{n} p_k B_{n,k}(x),$$

  then for all $x \in [0,1]$, we have

$$\min_{[0,1]} p \geqslant \min_k p_k \qquad \text{and} \qquad \max_{[0,1]} p \leqslant \max_k p_k.$$

  Warning: need for a conversion algorithm (cost in $O(M(n))$). Problems of stability.
- Tighter methods based on Descartes' rule of signs, Sturm's theorem, sums of squares (Hilbert's 17th problem), companion matrices, etc.

Ranges of polynomials

Second, why would this process yield tight enclosures? Our basic functions are analytic, and hence have (fast) converging Taylor series.

Chebyshev Models: using truncated Chebyshev series

$$P(x) = {\sum_{k=0}^{n}}{}' a_k T_k(x), \text{ where } a_k = \frac{2}{\pi} \int_{-1}^{1} \frac{f(x) T_k(x)}{\sqrt{1-x^2}} \mathrm{d}x.$$

[1] solutions of Linear Differential Equations with polynomial coefficients

Chebyshev Models: using truncated Chebyshev series

$$P(x) = \sum_{k=0}^{n}{}' a_k T_k(x), \text{ where } a_k = \frac{2}{\pi} \int_{-1}^{1} \frac{f(x) T_k(x)}{\sqrt{1-x^2}} \mathrm{d}x.$$

Computation of the coefficients (for "basic" D-finite functions[1])

- recurrence formulae[2] for computing a_k

[1] solutions of Linear Differential Equations with polynomial coefficients

[2] A. Benoit and B. Salvy, Chebyshev Expansions for Solutions of Linear Differential Equations, ISSAC '09: Proceedings of the twenty-second international symposium on Symbolic and algebraic computation, 23-30, ISSAC '09. ACM, New York, NY, 23-30

Chebyshev Models: using truncated Chebyshev series

$$P(x) = {\sum_{k=0}^{n}}' a_k T_k(x), \text{ where } a_k = \frac{2}{\pi} \int_{-1}^{1} \frac{f(x) T_k(x)}{\sqrt{1-x^2}} \mathrm{d}x.$$

Computation of the coefficients (for "basic" D-finite functions[1])

- recurrence formulae[2] for computing a_k

[1] solutions of Linear Differential Equations with polynomial coefficients

[2] A. Benoit and B. Salvy, Chebyshev Expansions for Solutions of Linear Differential Equations, ISSAC '09: Proceedings of the twenty-second international symposium on Symbolic and algebraic computation, 23-30, ISSAC '09. ACM, New York, NY, 23-30

Chebyshev Models: using truncated Chebyshev series

$$P(x) = \sum_{k=0}^{n}{}' a_k T_k(x), \text{ where } a_k = \frac{2}{\pi}\int_{-1}^{1} \frac{f(x)T_k(x)}{\sqrt{1-x^2}}\mathrm{d}x.$$

Computation of the coefficients (for "basic" D-finite functions[1])

Truncation Error: Bernstein-like formula (for "basic" D-finite functions)

$$\forall x \in [-1,1],\ \exists \xi \in [-1,1] \ \text{ s.t. } \ f(x) - P(x) = \frac{f^{(n+1)}(\xi)}{2^n (n+1)!}.$$

[1] solutions of Linear Differential Equations with polynomial coefficients

Chebyshev Models: using truncated Chebyshev series

$$P(x) = \sum_{k=0}^{n}{}' a_k T_k(x), \text{ where } a_k = \frac{2}{\pi} \int_{-1}^{1} \frac{f(x) T_k(x)}{\sqrt{1-x^2}} \mathrm{d}x.$$

Computation of the coefficients (for "basic" D-finite functions[1])

Truncation Error: Bernstein-like formula (for "basic" D-finite functions)

- For composite functions,use algebraic rules (addition, multiplication, composition) with models

[1] solutions of Linear Differential Equations with polynomial coefficients

Newton method

Theorem

Let $X \in \mathbb{IR}$, let $f \in \mathcal{C}^2(X)$, s.t. $f'(x) \neq 0$ for all $x \in X$ and f has a unique, simple zero x^* in X. Then if x_0 is chosen sufficiently close to x^*, the sequence $(x_k)_{k\in\mathbb{N}}$ defined by

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \text{ for } k = 0, 1, 2, \dots$$

converges quadratically fast toward x^*: there exists a constant C such that

$$\lim_{k\to+\infty} x_k = x^* \text{ and } |x_{k+1} - x^*| \leqslant C|x_k - x^*|^2.$$

Newton-like Fixed-Point Methods for A Posteriori Validation

Let $(X, \|\cdot\|_X)$ and $(Y, \|\cdot\|_Y)$ be two Banach spaces, $\mathbf{F} : X \to Y$ be a continuous and differentiable operator with respect to $\|\cdot\|_X$ and $\|\cdot\|_Y$, and $x^\circ \in X$.
Let $\mathbf{A} : X \to X$ be an injective bounded linear operator, from which we construct the Newton-like operator $\mathbf{T} : X \to X$:

$$\mathbf{T} \cdot x = x - \mathbf{A} \cdot \mathbf{F} \cdot x.$$

Newton-like Fixed-Point Methods for A Posteriori Validation

Theorem 3

Suppose $\mathbf{A}$ *"sufficiently close" to* $(\mathrm{D}\mathbf{F}_{x^\circ})^{-1}$, *so that* $\exists r > 0$ *s. t.*

(i) *There exists* $\lambda \in [0,1)$ *bounding* $\|\mathrm{D}\mathbf{T}_x\|_X$ *over* $\overline{B}(x^\circ, r) = \{x \in X \mid \|x - x^\circ\|_X \leqslant r\}$*:*

$$\|\mathrm{D}\mathbf{T}_x\|_X = \|\mathbf{1}_X - \mathbf{A} \cdot \mathrm{D}\mathbf{F}_x\|_X \leqslant \lambda, \qquad x \in \overline{B}(x^\circ, r).$$

(ii) $\overline{B}(x^\circ, r)$ *is stable under* $\mathbf{T}$, *which is ensured by:*

$$\|x^\circ - \mathbf{T} \cdot x^\circ\|_X + \lambda r < r.$$

Then $\mathbf{T}$ *admits a unique fixed point called* x^* *in* $\overline{B}(x^\circ, r)$, *and we have the following enclosure:*

$$\frac{\|x^\circ - \mathbf{T} \cdot x^\circ\|_X}{1 + \lambda} \leqslant \|x^\circ - x^*\|_X \leqslant \frac{\|x^\circ - \mathbf{T} \cdot x^\circ\|_X}{1 - \lambda}.$$

Division of two Chebyshev models

Proposition

Given:

- $f, g \in \mathcal{C}([-1,1])$ *represented by Chebyshev models* $\boldsymbol{f} = (f^\circ, \varepsilon)$ *and* $\boldsymbol{g} = (g^\circ, \eta)$,
- $h^\circ \in \mathbb{R}[x]$ *a polynomial approximation of* $h^* = f/g$,
- $k^\circ \in \mathbb{R}[x]$ *a polynomial approximation of* $1/g$,

we have the following rigorous upper bound on the approximation error:

$$\|h^\circ - f/g\|_\infty \leqslant \tau = \frac{\tau'}{1-\lambda},$$

provided that we have computed τ' *and* $\lambda < 1$ *such that:*

$$\|1 - k^\circ g^\circ\|_\infty + \eta \|k^\circ\|_\infty \leqslant \lambda,$$
$$\|k^\circ(g^\circ h^\circ - f^\circ)\|_\infty + \eta \|k^\circ h^\circ\|_\infty + \varepsilon \|k^\circ\|_\infty \leqslant \tau'.$$

Hence, $\boldsymbol{h} = (h^\circ, \tau)$ *is a Chebyshev model for* $h^* = f/g$.

Square root of a Chebyshev model

Proposition

Given:

- $f \in \mathcal{C}([-1,1])$ *represented by a Chebyshev model* $\boldsymbol{f} = (f^\circ, \varepsilon)$,
- $g^\circ \in \mathbb{R}[x]$ *a polynomial approximation of* $g^* = \sqrt{f}$,
- $k^\circ \in \mathbb{R}[x]$ *a polynomial approximation of* $1/g^\circ$,

we have the following rigorous upper bound on the approximation error:

$$\|g^\circ - \sqrt{f}\|_\infty \leqslant \eta = \frac{\eta'}{1-\lambda},$$

provided that we have computed $\lambda_0, \lambda_1, \eta', \Delta, r^\circ, \lambda$ *satisfying:*

$$\|1 - k^\circ g^\circ\|_\infty \leqslant \lambda_0 < 1,\ \|k^\circ\|_\infty \leqslant \lambda_1,\ \|k^\circ({g^\circ}^2 - f^\circ)\|_\infty + \varepsilon\|k^\circ\|_\infty \leqslant 2\eta',$$

$$\Delta := (1-\lambda_0)^2 - 4\lambda_1\eta' \geqslant 0, \qquad r^\circ := \frac{1-\lambda_0-\sqrt{\Delta}}{2\lambda_1},$$

$$\lambda := \lambda_0 + \lambda_1 r^\circ < 1.$$