Representations for feasibly approximable functions

Michal Konečný, Eike Neumann

Aston University, Birmingham, United Kingdom

Thirteenth International Conference on Computability and Complexity in Analysis (CCA2016), 15th June 2016

Introduction

- Given: continuous real function $f\colon [-1,1] \to \mathbb{R}$
- Problem: compute integral and range
- Generally perceived to be easy in practice
- Ko, Friedman (1982, 1984): integration and range $\#$P-hard and NP-hard in general, even for smooth functions: There exists a polytime computable smooth $f\colon [-1,1] \to \mathbb{R}$ s.t.
  - $i(t) = \int_{-1}^{t} f(s)\,\mathrm{d}s$ is polytime computable iff $\mathrm{FP} = \#\mathrm{P}$.
  - $m(t) = \max_{s\in[-1,t]} f(s)$ is polytime computable iff $\mathrm{P} = \mathrm{NP}$.

The elephant in the room

- Where does this discrepancy between “theory” and “practice” come from?
- One explanation: good asymptotic running time in output accuracy does not correspond to practical feasibility.
- Other approach: operators are polytime on well-behaved classes of functions.

Positive results

- Müller (1987): if f is polytime real analytic then $t \mapsto \int_{-1}^{t} f(s)\,\mathrm{ds}$ and $t \mapsto \max_{s\in[-1,t]} f(s)$ are polytime computable.
- Labhalla, Lombardi, Moutai (2001): Same is true for polytime smooth functions with well-behaved quantitative growth of derivatives (*Gevrey*-functions):

  $$||f^{(n)}||_\infty \leq M \cdot R^n \cdot n^{\alpha n},$$

  with $\alpha > 0$. Note that $\alpha = 1$ corresponds to analytic functions.
- Kawamura, Müller, Rösnick, Ziegler (2015): these results translate to uniform results in second-order complexity theory.

Closure properties of polytime Gevrey functions

The polytime Gevrey functions functions lack some "obvious" closure properties:

1. Not closed under parametric maximisation $t \mapsto \max_{s \in [-1,t]} f(s)$.
2. Not closed under square roots.
3. More subtle: the polytime Gevrey functions *are* closed under bounded division (division f/g where $g \geq 1$)...
4. but not *uniformly* closed.
5. The family of analytic functions

$$\frac{1}{1 + ax^2}$$

is not uniformly polytime Gevrey (in $\log a$).

Can we do better?

Goal: Find a class of polytime computable functions with better closure properties on which integral and maximum are uniformly polytime computable.

Feasibly approximable functions

Theorem (Labhalla et al., 2001)

A polytime computable function is α-Gevrey for some $\alpha > 0$ if and only if it is feasibly polynomially approximable.

A function f is feasibly polynomially approximable iff there is a polytime computable sequence $(\tilde{f}_n)_n$ of polynomials with

$$||f - \tilde{f}_n|| < 2^{-n}.$$

$\Rightarrow$ study more general classes of feasibly approximable functions.

- Feasibly approximable $\simeq$ polytime computable point of a Cauchy representation.

$\Rightarrow$ study more general Cauchy-representations.

- Commonly used representations besides polynomials: rational approximations and splines or piecewise polynomials.

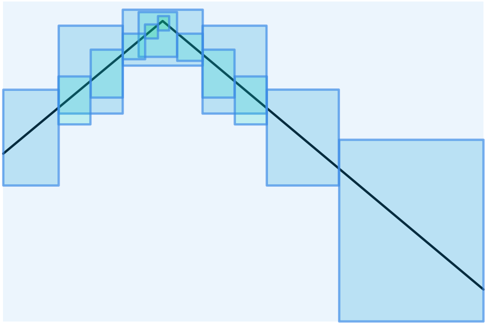
On the theoretical foundation

- Officially we work with second-order complexity as introduced by Kawamura and Cook (2010).
- We will present very concrete constructions on very concrete representations though...
- So we will be quite informal in this talk.

Representations: Fun

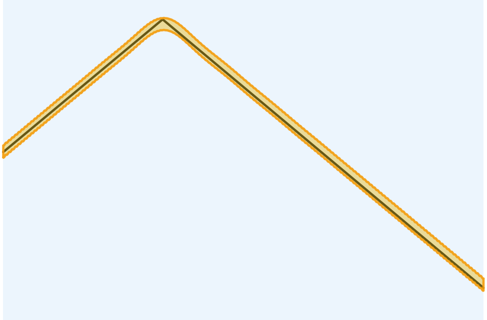
Representation Fun: Real function $f\colon[-1,1]\to\mathbb{R}$ is represented by $\varphi\colon\mathbb{ID}\to\mathbb{ID}$, taking dyadic rational intervals to dyadic rational intervals with

- $\varphi(I)\supseteq f(I)$.
- $(I_n)_n\to\{x\}\Rightarrow\varphi(I_n)\to\{f(x)\}$.



Representations: Poly

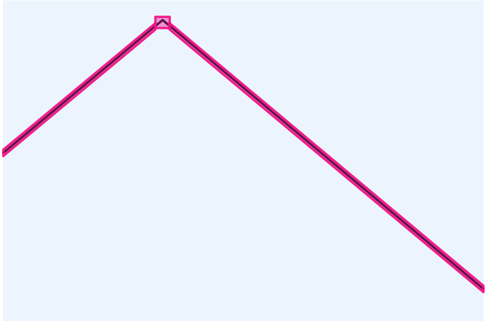
Representation Poly: Real function $f\colon [-1,1] \to \mathbb{R}$ is represented by fast converging Cauchy sequence of polynomials with dyadic rational coefficients.



Representations: PPoly

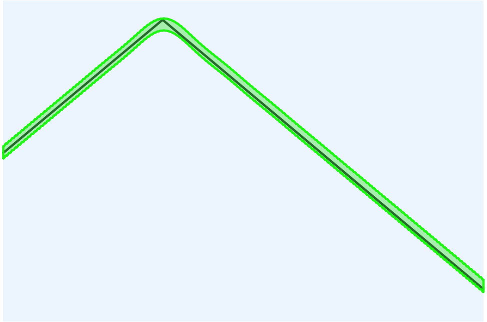
Representation PPoly: Real function $f: [-1, 1] \to \mathbb{R}$ is represented by fast converging Cauchy sequence of *piecewise polynomials*:

- $[-1, 1]$ is divided onto rational segments $[a, b]$, $a, b \in \mathbb{Q}$.
- On each segment, f is approximated by a polynomial with dyadic rational coefficients.



Representations: Frac

Representation Frac: Real function $f : [-1, 1] \to \mathbb{R}$ is represented by fast converging Cauchy sequence of rational functions P_n/Q_n with $P_n, Q_n \in \mathbb{D}[x]$ and $Q_n(x) \geq 1$ on $[-1, 1]$.



Relationship between the representations

Theorem

We have polytime computable translations

$$\mathsf{Poly} \to \mathsf{PPoly} \leftrightarrows \mathsf{Frac} \to \mathsf{Fun}$$

none of which reverses unless indicated.

Most of these proved in LABHALLA et al. (2001) The only "new" result is the translation Frac $\to$ PPoly.

Computing Frac $\rightarrow$ PPoly

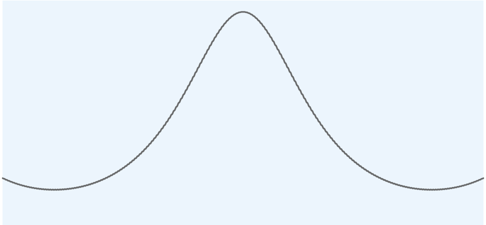
Computing $1/f$ ($f \geq 1$) w.r.t PPoly:

- Get a piecewise-polynomial approximation $\tilde{f}$ of f.

Computing Frac $\to$ PPoly

Computing $1/f$ ($f \geq 1$) w.r.t PPoly:

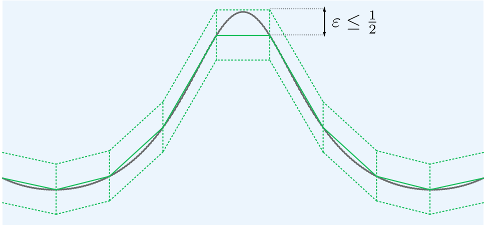
- Get a piecewise-polynomial approximation $\tilde{f}$ of f.
- Interpolate $1/\tilde{f}$ linearly up to error $\varepsilon < 1/2$.



Computing Frac $\to$ PPoly

Computing $1/f$ $(f \geq 1)$ w.r.t PPoly:

- Get a piecewise-polynomial approximation $\tilde{f}$ of f.
- Interpolate $1/\tilde{f}$ linearly up to error $\varepsilon < 1/2$.

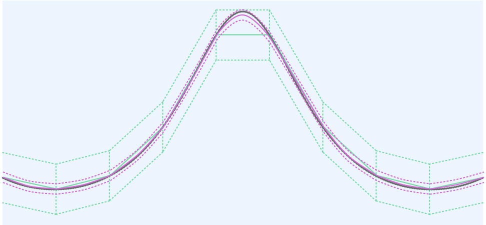


Computing Frac → PPoly

Computing $1/f$ ($f \geq 1$) w.r.t PPoly:

- Get a piecewise-polynomial approximation $\tilde{f}$ of f.
- Interpolate $1/\tilde{f}$ linearly up to error $\varepsilon < 1/2$.
- Apply Newton-Raphson division to improve approximation:

$$f_{n+1} = 2f_n - f_n^2 \tilde{f}.$$



- Degree after n iterations: $(2^n - 1)d_{\tilde{f}} + 2^n$.
- Error after n iterations: ε^{2^n}.

Computing range and integral

Theorem (Labhalla et al., 2001)

Maximisation and Integration (viewed as functionals) are uniformly polytime computable w.r.t. PPoly.

Corollary

Maximisation and Integration (viewed as functionals) are uniformly polytime computable w.r.t. Frac.

Computing Range

Computing $\max_{[-1,1]} f$ with respect to Poly:

- Get a polynomial approximation $\tilde{f}$ with accuracy ε.
- Compute the critical points of $\tilde{f}$:
  - Compute the derivative $\tilde{f}'$.
  - Compute the *separable part* $\tilde{f}_s' = \tilde{f}' / \gcd(\tilde{f}', \tilde{f}'')$.
  - Isolate the real roots of $\tilde{f}_s$.
  - Approximate the roots up to sufficient accuracy.
- Take the maximum over the critical points and the boundary points.

For PPoly, do this piecewise.

Closure properties of PPoly

Theorem

The class of feasibly piecewise-polynomially approximable functions is uniformly closed under

- *Addition, subtraction and multiplication*
- *Bounded division*
- *Composition*
- *Pairwise and parametric maximisation*
- *Taking antiderivatives*
- *Taking square roots*
- *Taking absolute values*

Ko-Friedman revisited

Corollary

If $f : [-1, 1] \to \mathbb{R}$ *is expressible as a term whose leaves can be written as polytime computable analytic functions and whose nodes use only the operations* $+, -, \times, \div, \circ, \max, \int, \sqrt{\cdot}, |\cdot|$*, then*

- f *is polytime computable.*
- $m(t) = \max_{s \in [-1,t]} f(s)$ *is polytime computable.*
- $i(t) = \int_{-1}^{t} f(s)\, \mathrm{ds}$ *is polytime computable.*

Validation

- How well do these theoretical results translate into practice?
- To validate our results we have implemented the representations and algorithms in Haskell (`http://tinyurl.com/aern2-fnreps`):
- Fun: Haskell type

$$\text{Interval MPFloat} \to \text{Interval MPFloat}.$$

- Poly: Haskell type

$$\text{Int} \to (\text{Map Int MPFloat}, \text{Double}).$$

- PPoly: Haskell type

$$\text{Int} \to ([(\text{Interval Rational}, \text{Map Int MPFloat})], \text{Double}).$$

Validation

- Polynomial approximations are constructed from Taylor series or using Chebyshev-interpolation via Discrete Cosine Transform.
- Division for PPoly uses Newton iteration, division for Poly uses Chebyshev-interpolation.
- Integration for Fun is just Riemann integration.
- To have a slightly less naive integration algorithm available for Fun we introduced a new representation DFun which encodes a Fun name of a function and its derivative.

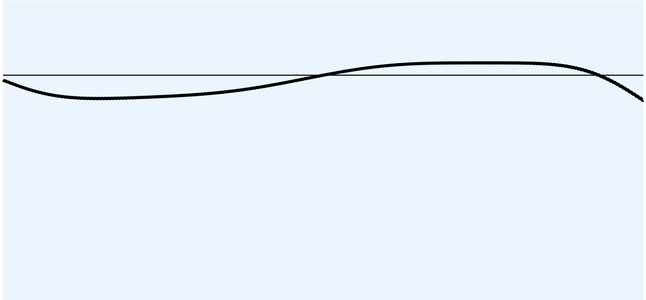
Maximisation revisited

- How to compute the maximum in practice?
- Initial attempt: use root isolation. Compute separable part using signed subresultant sequences.
- Quadratic blowup in coefficient size in the signed subresultant sequence turns out to be a bit much.

$\Rightarrow$ avoid computing separable part.

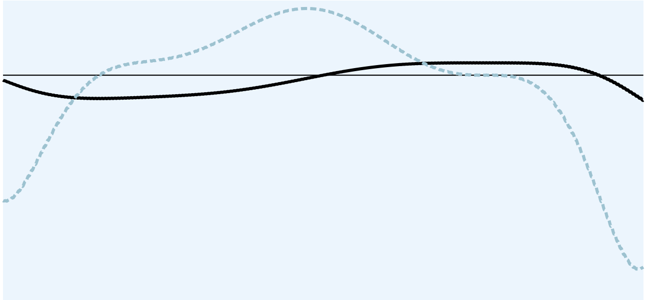
Maximisation algorithm

- Get a polynomial approximation.



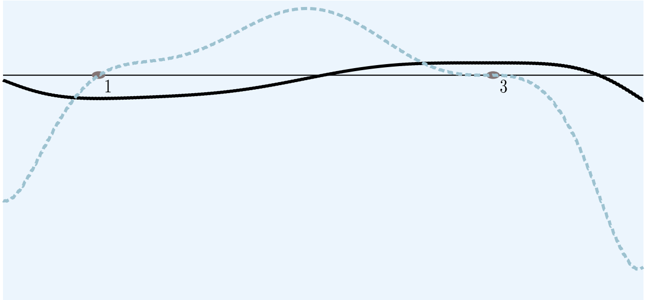
Maximisation algorithm

- Get a polynomial approximation.
- Compute the derivative.



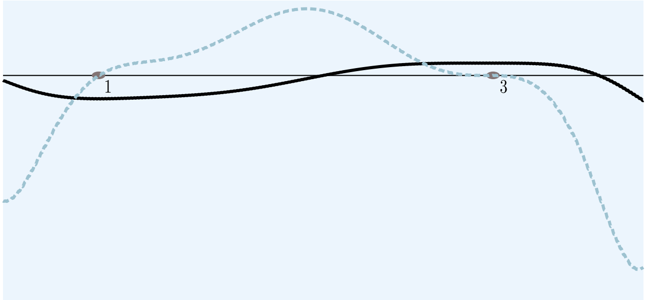
Maximisation algorithm

- Get a polynomial approximation.
- Compute the derivative.



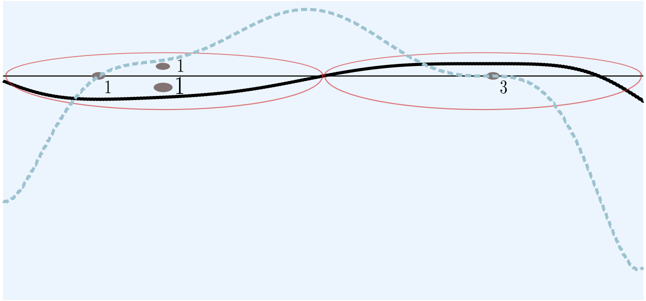
Maximisation algorithm

- Get a polynomial approximation.
- Compute the derivative.
- Estimate the maximum on the interval using a Lipschitz constant.



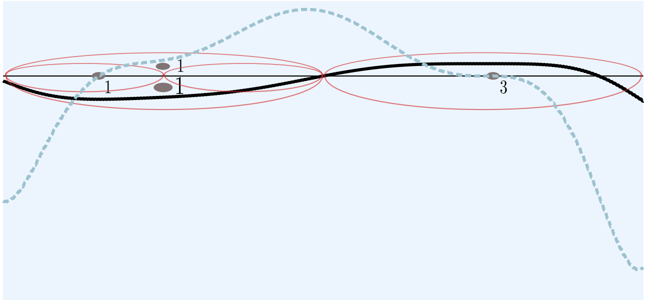
Maximisation algorithm

- Get a polynomial approximation.
- Compute the derivative.
- Estimate the maximum on the interval using a Lipschitz constant.
- If the estimate is not accurate enough, split the interval in two, and count the complex roots around the sub-intervals.



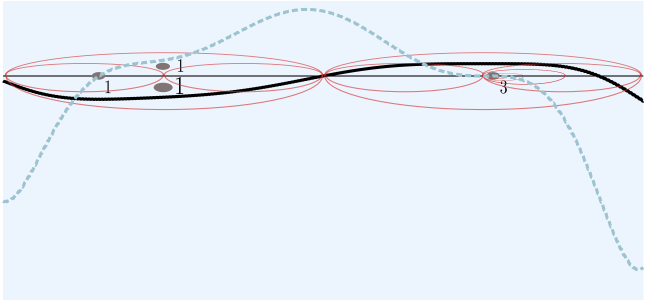
Maximisation algorithm

- Get a polynomial approximation.
- Compute the derivative.
- Estimate the maximum on the interval using a Lipschitz constant.
- If the estimate is not accurate enough, split the interval in two, and count the complex roots around the sub-intervals.

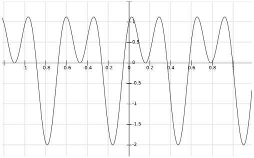


Maximisation algorithm

- Get a polynomial approximation.
- Compute the derivative.
- Estimate the maximum on the interval using a Lipschitz constant.
- If the estimate is not accurate enough, split the interval in two, and count the complex roots around the sub-intervals.

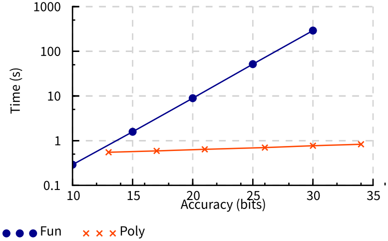


Benchmarks: A simple analytic function

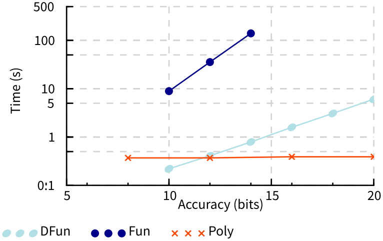


$$f(x) = \sin(10x) + \cos(20x)$$

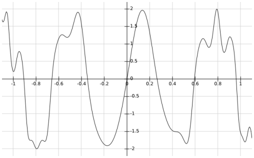
Maximum:



Integral:

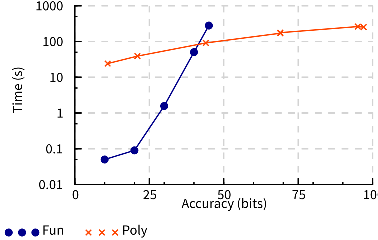


Benchmarks: A more complicated analytic function involving composition

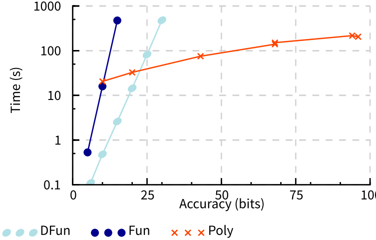


$$f(x) = \sin(10x + \sin(20x^2)) + \sin(10x)$$

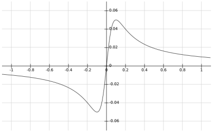
Maximum:



Integral:

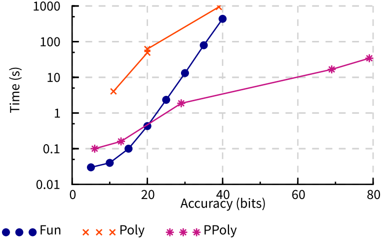


Benchmarks: An simple analytic function with singularities near the origin

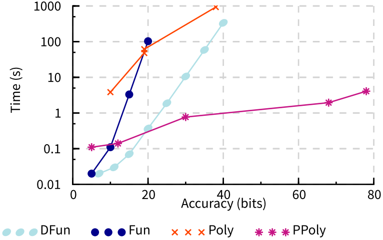


$$f(x) = \frac{x}{1 + 100x^2}$$

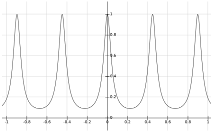
Maximum:



Integral:

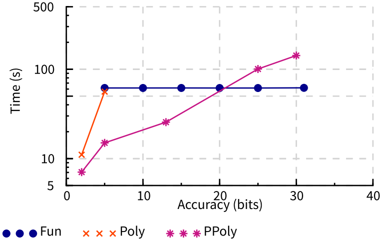


Benchmarks: A more complicated analytic function with singularities near the origin

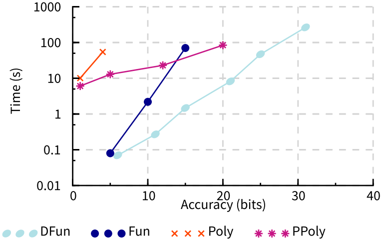


$$f(x) = \frac{1}{1 + 10\sin^2(7x)}$$

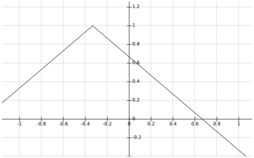
Maximum:



Integral:

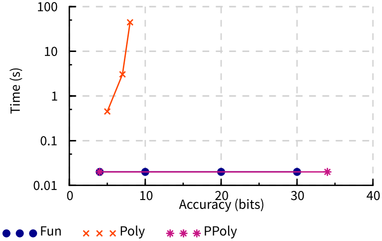


Benchmarks: A very simple non-smooth function

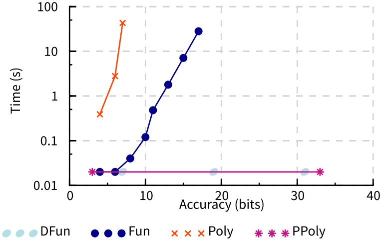


$$f(x) = 1 - |x + 1/3|$$

Maximum:



Integral:



Conclusion

- In terms of polytime reducibility, *piecewise polynomials* are better than polynomials and equivalent to rational functions.
- In contrast to the standard function space representation, they render integration and maximisation polytime computable.
- Thus, we've found a class of polytime computable functions which
  - is uniformly closed under integration and maximisation.
  - enjoys nice further closure properties.
- For the functions we tested, polytime computability and practical feasibility seem to be match quite well.
  - with some exceptions...

Future work

- Improve division algorithm.
- Improve PPoly-integration.
- Introduce representations for multivariate functions and study differential equations.
- Is there a nice characterisation of class of feasibly piecewise-polynomially approximable functions?

References

Akitoshi Kawamura and Stephen A. Cook.
Complexity theory for operators in analysis.
ACM Transactions on Computation Theory, 4(2):5, 2012.

Ker-I Ko and Harvey Friedman.
Computational complexity of real functions.
Theoretical Computer Science, 20:323–352, 1982.

Akitoshi Kawamura, Norbert Müller, Carsten Rösnick, and Martin Ziegler.
Computational benefit of smoothness: Parameterized bit-complexity of numerical operators on analytic functions and Gevrey's hierarchy.
Journal of Complexity, 31(5):689 – 714, 2015.

Ker-I Ko.
Complexity Theory of Real Functions.
Birkhäuser, 1991.

S. Labhalla, H.Lombardi, and E.Moutai.
Espaces métriques rationnellement présentés et complexité, le cas de l'espace des fonctions réelles uniformément continues sur un intervalle compact.
Theoretical Computer Science, 250:265–332, 2001.

Norbert Th. Müller.
Uniform computational complexity of Taylor series.
In *Automata, Languages and Programming*, volume 267 of *Lecture Notes in Computer Science*, pages 435–444. Springer, 1987.