



TECHNISCHE
UNIVERSITÄT
DARMSTADT



Real Parameterized and 2nd Order Complexity Theory: From Computability in Analysis to Numerical Practice



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Martin Ziegler





Examples from Computable Analysis

Folklore: For $x \in \mathbb{R}$ the following are equivalent:

- a) x has a decidable binary expansion
- b) x has a recursive *signed-digit* expansion
- c) There exists a recursive sequence $(a_n) \subseteq \mathbb{Z}$

s.t. $|x - a_n/2^{n+1}| \leq 2^{-n}$ numerics / iRRAM

- d) There exist recursive sequences $(p_n), (q_n) \subseteq \mathbb{Q}$
- s.t. $\sup_n p_n = x = \inf_n q_n$

Folklore: Every computable $f: [0;1] \rightarrow \mathbb{R}$ with $f(0) \cdot f(1) < 0$ has a computable root.

Obstacles to practice:

- i) only *non-uniformly*
- ii) no running time bound

Real Function Complexity



Function $f:[0,1] \rightarrow \mathbb{R}$ **computable** in time $t(n)$

if some TM can, on input of $n \in \mathbb{N}$ and of

$(a_m) \subseteq \mathbb{Z}$ with $|x - a_m|/2^{m+1} \leq 2^{-m}$ $=: \rho \equiv_p \rho_{sd}$

in time $t(n)$ output $b \in \mathbb{Z}$ with $|f(x) - b|/2^{n+1} \leq 2^{-n}$.



Examples: a) $+, \times, \exp$ polytime on $[0;1]!$

b) $f(x) \equiv \sum_{n \in L} 4^{-n}$ iff $L \subseteq \{0,1\}^*$ polytime-decidable

c) $\text{sign}(e^x)$ is not polytime-computable

Observation i) If f computable \Rightarrow continuous.

ii) If f computable in time $t(n)$, then

$O(t(O(n)))$ is a modulus of uniform continuity of f .



Discrete Complexity Reminder

Real Parameterized and 2nd-Order Complexity Theory:
From Computability in Analysis to Numerical Practice

Definition: \mathcal{M} decides set $L \subseteq \{0,1\}^*$ in **time** if

$L = \{ \underline{x} \in \{0,1\}^n \mid \text{on input } \underline{x}, \mathcal{M} \text{ prints } 1 \text{ and terminates} \}$

• on inputs $\underline{x} \notin L$ prints 0 and terminates.

Example: $L = \{ 10, 11, 101, 111, 1011, 1101, \dots \}$

Def: \mathcal{M} runs in **polynom. time / space** if $\exists p \in \mathbb{N}[N]$:

\mathcal{M} on input $\underline{x} \in \{0,1\}^n$ makes at most $p(n)$ steps
/ uses at most $p(n)$ bits of memory.

$\mathcal{P} = \{ L \subseteq \{0,1\}^* \text{ decidable in polynomial time} \}$

$\subseteq \mathcal{NP} = \{ L \text{ verifiable in polynomial time} \}$

$\subseteq \mathcal{PSPACE} := \{ L \text{ decidable in polyn. space} \}$

$\subseteq \mathcal{EXP} = \{ L \text{ decidable in exponential time} \}$

1000'000



Nonuniform Complexity of Operators

$f:[0;1] \rightarrow [0;1]$ polytime computable (\Rightarrow continuous)

- Max: $f \rightarrow \text{Max}(f): x \rightarrow \max\{ f(t): t \leq x \}$

$\text{Max}(f)$ computable in exponential time;
polytime-computable iff $\mathcal{P} = \mathcal{NP}$

- $\int: f \rightarrow \int f: (x \rightarrow \int_0^x f(t) dt)$

$\int f$ computable in exponential time;

" $\#\mathcal{P}$ -complete"

another class between
 \mathcal{NP} and \mathcal{PSPACE}

even when
restricting
to $f \in C^\infty$

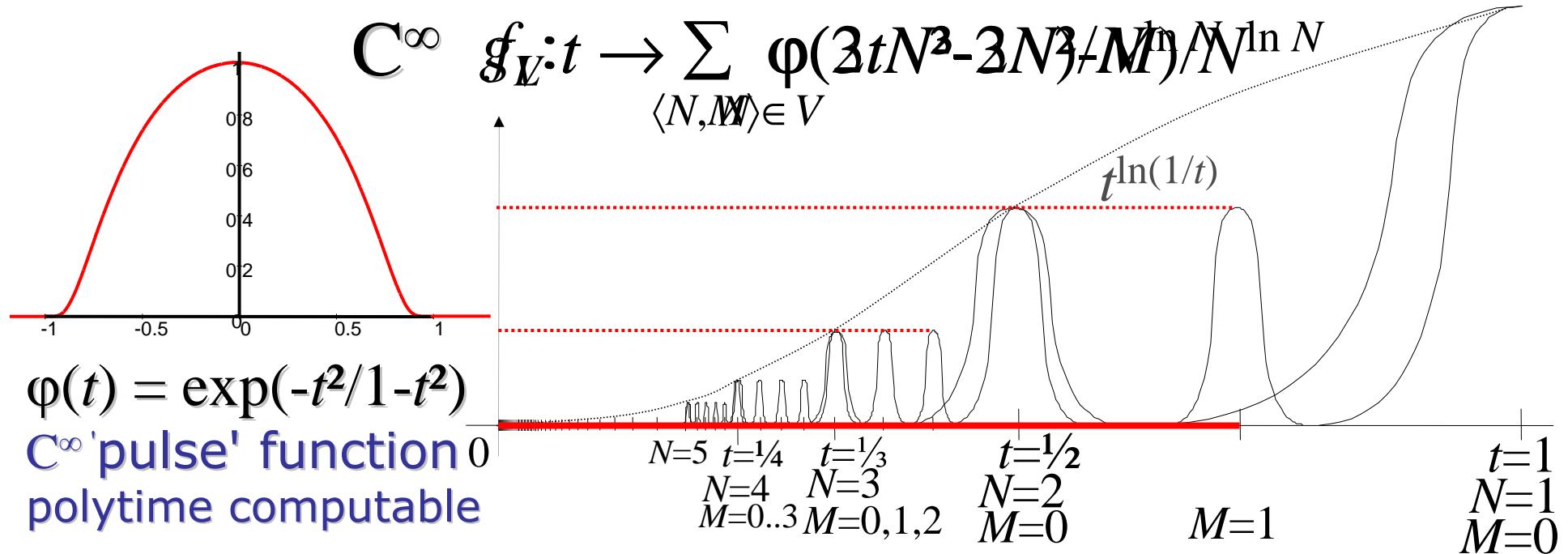
- dsolve: $C[0;1] \times [-1;1] \ni f \rightarrow z: \dot{z}(t) = f(t, z), z(0) = 0.$
 - in general no computable solution $z(t)$
 - for $f \in C^1$ \mathcal{PSPACE} -"complete" [Kawamura'10,
 - for $f \in C^k$ \mathcal{CH} -"hard" Kawamura et al]

[Friedman&Ko'82]

'Max is \mathcal{NP} -hard'

Real Parameterized and 2nd-Order Complexity Theory:
From Computability in Analysis to Numerical Practice

$$\mathcal{NP} \ni L = \{ N \in \mathbb{N} \mid \exists M < N : \langle N, M \rangle \in V \text{ polytime} \}, V \in \mathcal{P}$$



To every $L \in \mathcal{NP}$ there exists a polytime computable C^∞ function $g_L: [0,1] \rightarrow \mathbb{R}$ s.t.: $[0,1] \ni t \mapsto \max_{\langle N, M \rangle \in V} g_L|_{[0,t]}$ again polytime iff $L \in \mathcal{P}$



Uniform Complexity of Operators ?

$f:[0;1] \rightarrow [0;1]$ polytime computable (\Rightarrow continuous)

- Max: $f \rightarrow \text{Max}(f): x \rightarrow \max\{ f(t): t \leq x \}$

$\text{Max}(f)$ computable in exponential time;
polytime-computable iff $\mathcal{P} = \mathcal{NP}$

- $\int: f \rightarrow \int f: (x \rightarrow \int_0^x f(t) dt)$ non-uniform
 $\int f$ computable in exponential time;
 $\#P$ -"complete"

even when
restricting
to $f \in C^\infty$
but for
analytic f
polytime

- dsolve: $C[0;1] \times [-1;1] \ni f \rightarrow z: \dot{z}(t) = f(t, z), z(0) = 0.$
 - in general no computable solution $z(t)$
 - for $f \in C^1$ $PSPACE$ -"complete" [Kawamura'10,
 - for $f \in C^k$ CH -"hard" Kawamura et al]



Three Effects in Real Complexity

that numerical scientists might be interested in / should be aware of

- a) natural emergence of multivaluedness
(aka non-extensionality) → ε -semantics of " $<$ "
- b) Uniform computation may require discrete advice or otherwise 'enriched' representations (TTE)
 - which yield canonical C++ declarations
- c) Parameterized uniform upper complexity bounds

Example (Brattka&Z, *Computable Spectral Thm*)

Finding an eigenvector (basis) to a given real

Example: +, exp computable in time polynomial

in n on $[0;1]$; on $[0;2^k]$: + in time polynomial in

$n+k$, exp in time polynomial in $n+2^k$.

independent of x
on *compact* dom



Parameterized Complexity in TTE

- Definition:** a) A partial $F: \subseteq \{0,1\}^\omega \rightarrow \{0,1\}^\omega$ is **computable in time** $t: \mathbb{N} \rightarrow \mathbb{N}$ if a Type-2 machine can convert $\underline{\sigma} \in \text{dom}(F)$ to $\underline{\tau} = F(\underline{\sigma})$ s.t. the n -th symbol of $\underline{\tau}$ appears within $t(n)$ steps.
- b) For spaces X, Y equipped with representations α, β , call (multivalued partial) $f: \subseteq X \Rightarrow Y$ is **(α, β)-computable in time t** if it admits an (α, β) -realizer F computable in time t .
- c) A **parameter** to a space X with representation α is a mapping $k: \text{dom}(\alpha) \rightarrow \mathbb{N}$.
- d) For X, Y spaces with representations α, β and parameters

Remark: As above fully polytime (α, k, β, t) -computable (e.g., impossible) representations induce trivial notions of complexity.

[Weihrauch'03] and [Schröder'04] have devised (meta-) conditions on representations and F such that avoid such degeneracies.



Complexity Theory of Operators

Evaluation $\text{Eval}:(f,x) \rightarrow f(x)$

a) requires $\geq \mu(n)$ steps, $\mu:\mathbb{N} \rightarrow \mathbb{N}$ mod. of continuity to f .

"Parameter" $\mu(f)$ is not \mathbb{N} -valued but $\mathbb{N}^{\mathbb{N}}$ -valued!

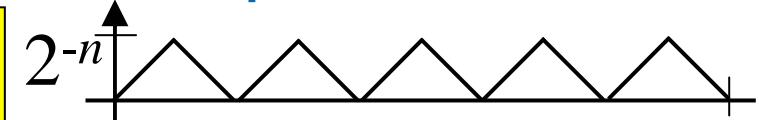
b) Even restricted to the compact domain sequent.
access

$\mathcal{L}_1 := \{ f:[0;1] \rightarrow [0;1] \text{ 1-Lipschitz} \}$ (Arzela-Ascoli)

there exists no representation $\delta: \subseteq \{0,1\}^\omega \rightarrow \mathcal{L}_1$

rendering Eval computable in subexponential time.

$\geq 2^{2^{n-1}}$ functions pairwise differing
when evaluating up to error 2^{-n}
but only $2^{t(n)}$ different initial
segments of δ -names that can
be read within $t(n)$ steps. q.e.d.



$\approx 2^{n-1}$ 'hats'

mapping $k:\text{dom}(\alpha) \rightarrow \mathbb{N}^{\mathbb{N}}$?
vers $k, l: (\alpha, \beta)$ -realizer F

required computable on inputs $\underline{\sigma}$ within $\text{poly}(n+k(\underline{\sigma}))$ steps.



Complexity Theory of Operators

Evaluation $\text{Eval}:(f,x) \rightarrow f(x)$

a) requires $\geq \mu(n)$ steps, $\mu:\mathbb{N} \rightarrow \mathbb{N}$ mod. of continuity to f .

Parameter $\mu(f)$ is not \mathbb{N} -valued but $\mathbb{N}^{\mathbb{N}}$ -valued!

b) Even restricted to the compact domain *sequent.
access*

$\mathcal{L}_1 := \{ f:[0;1] \rightarrow [0;1] \text{ 1-Lipschitz} \}$ (Arzela-Ascoli)

there exists no representation $\delta: \subseteq \{0,1\}^\omega \rightarrow \mathcal{L}_1$

rendering Eval computable in *subexponential* time.

Kawamura&Cook'10 (based on Cook&Kapron'96):
Remedy to a): *second*-order complexity theory
Remedy to b): *second*-order representations



Second-Order Complexity Theory

Evaluation $\text{Eval}:(f,x) \rightarrow f(x)$

a) requires $\geq \mu(n)$ steps, $\mu:\mathbb{N} \rightarrow \mathbb{N}$ mod. of continuity to f .

Parameter $\mu(f)$ is not \mathbb{N} -valued but $\mathbb{N}^{\mathbb{N}}$ -valued!

Def: A **second-order polynomial** $P(n,\lambda)$ is a term built from $+$, \times , integer constants and (first-order) variable n (ranging over \mathbb{N}) and second-order variable λ (ranging over $\mathbb{N}^{\mathbb{N}}$).

Example: $\lambda^3(\lambda(n^2) \cdot n + \lambda^2(n)) + n^{17}$

Observation: a) Second-order polynomials are closed under *both* kinds of composition

$(Q \circ P)(n,\lambda) := Q(P(n,\lambda),\lambda)$ and $(Q \square P)(n,\lambda) := Q(n,P(\cdot,\lambda))$

b) For $\lambda \in \mathbb{N}[n]$, $P(n,\lambda)$ is an ordinary polynomial.



Second-Order Computation

Example: $\{0,1\}^{\{0,1\}^*} \ni Q \rightarrow (\{0,1\}^* \ni \underline{v} \rightarrow \exists \underline{u} \in \{0,1\}^{|\underline{v}|}: Q(\underline{v}, \underline{u}))$
can be computed in 2nd-order polytime by a non-deterministic oracle machine but *not* by a deterministic one.
provably

$$\begin{aligned} |\psi(n)| \\ := |\psi(1^n)| \end{aligned}$$

Def: A **second-order polynomial** $P(n, \lambda)$ is a term built from $+, \times$, integer constants and (first-order) variable n (ranging over \mathbb{N}) and second-order variable λ (ranging over $\mathbb{N}^\mathbb{N}$).

Recall that $(\{0,1\}^*)^{\{0,1\}^*}$ denotes the set $\{ \psi: \{0,1\}^* \rightarrow \{0,1\}^* \}$.
For $F: \subseteq (\{0,1\}^*)^{\{0,1\}^*} \rightarrow (\{0,1\}^*)^{\{0,1\}^*}$, oracle Turing machine $\mathcal{M}^?$ **computes** F if \mathcal{M}^ψ on input $\underline{v} \in \{0,1\}^*$ outputs $\underline{w} = F(\psi)(\underline{v})$.

Call $\psi \in (\{0,1\}^*)^{\{0,1\}^*}$ **length-monotone** if $|\psi(\underline{v})| \leq |\psi(\underline{w})| \quad \forall |\underline{v}| \leq |\underline{w}|$.
 $\mathcal{M}^?$ runs in **2nd-order polytime** if, for some 2nd-order polynomial P , \mathcal{M}^ψ on input $\underline{v} \in \{0,1\}^*$ makes $\leq P(|\underline{v}|, |\psi|)$ steps.



Second-Order Representations

Real Parameterized and 2nd-Order Complexity Theory:
From Computability in Analysis to Numerical Practice

Even on compact $\mathcal{L}_1 = \{ f:[0;1] \rightarrow [0;1] \text{ 1-Lipschitz} \}$
there is no representation $\delta: \subseteq \{0,1\}^\omega \rightarrow \mathcal{L}_1$ **seq. access**
rendering Eval computable in subexponential time.

{ length-monotone ψ }

Def: A second-order representation of X
is a surjective partial mapping $\Delta: \subseteq \text{LM} \rightarrow X$

A (Δ, Γ) -realizer of $f: \subseteq X \Rightarrow Y$ is a mapping $F: \text{LM} \rightarrow \text{LM}$ s.t....

For $F: \subseteq (\{0,1\}^*)^{\{0,1\}^*} \rightarrow (\{0,1\}^*)^{\{0,1\}^*}$, oracle Turing machine $\mathcal{M}?$ computes F if \mathcal{M}^ψ on input $\underline{v} \in \{0,1\}^*$ outputs $\underline{w} = F(\psi)(\underline{v})$.

Call $\psi \in (\{0,1\}^*)^{\{0,1\}^*}$ **length-monotone** if $|\psi(\underline{v})| \leq |\psi(\underline{w})| \quad \forall |\underline{v}| \leq |\underline{w}|$.
 $\mathcal{M}?$ runs in **2nd-order polytime** if, for some 2nd-order polynomial P , \mathcal{M}^ψ on input $\underline{v} \in \{0,1\}^*$ makes $\leq P(|\underline{v}|, |\psi|)$ steps.



Examples of 2nd-Order Complexity

Real Parameterized and 2nd-Order Complexity Theory:
From Computability in Analysis to Numerical Practice

- a) An ordinary representation $\delta: \subseteq \{0,1\}^\omega \rightarrow X$ induces a 2nd-order representation Δ where $\psi: \{0,1\}^* \rightarrow \{0,1\}$ is a Δ -name of $x \in X$ iff $(\psi(1^n))_n$ is a δ -name of x .
- b) Define a $\text{p}^D\text{-name}$ of $f \in C[0;1]$ as a $\psi \in LM$ s.t.
- $$|f(\text{bin}(\underline{v})/2^{|\underline{v}|+1}) - \text{bin}(\psi(\underline{v}))/2^{|\underline{v}|+1}| \leq 2^{-|\underline{v}|}$$

Lemma a) Polytime δ -computability is uniformly equivalent to 2nd-order polytime Δ -computability
b) Evaluation $(f,x) \rightarrow f(x)$ is *not* $(\text{p}^D \times P, P)$ -computable.

A 2nd-order representation of X is a surjective $\Delta: \subseteq LM \rightarrow X$

Call $\psi \in (\{0,1\}^*)^{\{0,1\}^*}$ length-monotone if $|\psi(\underline{v})| \leq |\psi(\underline{w})| \quad \forall |\underline{v}| \leq |\underline{w}|$.

\mathcal{M}^ψ runs in 2nd-order polytime if, for some 2nd-order polynomial P , \mathcal{M}^ψ on input $\underline{v} \in \{0,1\}^*$ makes $\leq P(|\underline{v}|, |\psi|)$ steps.



Examples of 2nd-Order Complexity

b) Define a ρ^D -name of $f \in C[0;1]$ as a $\psi \in LM$ s.t.

$$|f(\text{bin}(\underline{v})/2^{|\underline{v}|+1}) - \text{bin}(\psi(\underline{v}))/2^{|\underline{v}|+1}| \leq 2^{-|\underline{v}|}$$

c) Define a $\rho^D \Pi Lip$ -name of $f \in Lip_{2^\ell}[0;1]$ as

$\{0,1\}^* \ni \underline{v} \mapsto 1^\ell \overline{0} \psi(\underline{v})$ for a ρ^D -name ψ of f .

d) A $[\rho \rightarrow \rho]$ -name of $f \in C[0;1]$ is and μ a modulus of continuity of f
 $\{0,1\}^* \ni \underline{v} \mapsto 1^{\mu(|\underline{v}|)} \overline{0} \psi(\underline{v})$ for ρ^D -name ψ of f .

Lemma c) Evaluation on $Lip[0;1]$ is 2nd-order

polytime ($\rho^D \Pi Lip \times P, P$)-computable

d) and 2nd-order polytime ($[\rho \rightarrow \rho] \times P, P$)-computable on $C[0;1]$.

\mathcal{M}^ψ runs in 2nd-order polytime if, for some 2nd-order polynomial P , \mathcal{M}^ψ on input $\underline{v} \in \{0,1\}^*$ makes $\leq P(|\underline{v}|, |\psi|)$ steps.



Some (recent) literature

Real Parameterized and 2nd-Order Complexity Theory:
From Computability in Analysis to Numerical Practice

- P.Oliva: "Polynomial-time Algorithms from Ineffective Proofs", pp.128-137 in *Proc. 18th IEEE Symp. on Logic in Computer Science* (LiCS'03)
- K.Weihrauch: "Computational Complexity on Computable Metric Spaces", pp.3-21 in *Mathematical Logic Quarterly* vol.**49:1** (2003).
- M.Schröder: "Spaces Allowing Type-2 Complexity Theory Revisited", pp.443-459 in *Mathematical Logic Quarterly* vol.**50** (2004).
- M.Braverman: "On the Complexity of Real Functions", pp.155-164 in *Proc. 46th Annual IEEE Symposium on Foundations of Computer Science* (FOCS'05).
- Z.Du, C.K.Yap: "Uniform Complexity of Approximating Hypergeometric Functions with Absolute Error", pp.246-249 in *Proc. 7th Asian Symp. on Computer Math.* (ASCM 2005)
- N.Müller, X.Zhao: "Complexity of Operators on Compact Sets", pp.101-119 in *Proc. Computability and Complexity in Analysis* (CCA'07), vol.**202** (2008)
- K.-I.Ko, F.Yu: "On the Complexity of Convex Hulls of Subsets of the Two-Dimensional Plane", pp.121-135 in *ENTCS* vol.**202** (2008)
- R.Rettinger: "Lower Bounds on the Continuation of Holomorphic Functions", pp.207-217 in *Proc. 5th Int. Conf. on Computability and Complexity in Analysis* (CCA 2008), ENTCS vol.**221**
- R.Rettinger: "Towards the Complexity of Riemann Mappings", in *Proc. 6th Int. Conf. on Computability and Complexity in Analysis* (CCA 2009)



Some (recent) literature

Real Parameterized and 2nd-Order Complexity Theory:
From Computability in Analysis to Numerical Practice

- G.Hotz: "On in Polynomial Time Approximable Real Numbers and Analytic Functions", pp.155-164 in *Informatik als Dialog zwischen Theorie und Anwendung*, Vieweg+Teubner (2009).
- A.Kawamura: "Lipschitz Continuous Ordinary Differential Equations are Polynomial-Space Complete", pp.305-332 in *Computational Complexity* vol.**19:2** (2010)
- O.Bournez, D.S.Graça, A.Pouly: "Solving Analytic Differential Equations in Polynomial Time over Unbounded Domains", pp.170-181 in *Proc. 36th Int. Symp. Mathem. Found. of Comp.Sc.* (MFCS'2011), Springer LNCS vol.**6907**
- A.Kawamura, S.A.Cook: "Complexity Theory for Operators in Analysis", *ACM Transactions in Computation Theory* vol.**4:2** (2012), article 5.
- A.Kawamura, H.Ota, C.Rösnick, M.Z.: "Computational Complexity of Smooth Differential Equations", pp.578-589 in *Proc. 37th Int. Symp. on Mathem. Found. of Computer Science* (MFCS'2012), Springer LNCS vol.**7464**
- C.Spandl: "Computational Complexity of Iterated Maps on the Interval", pp.1459-1477 in *Mathematics and Computers in Simulation* vol.**82:8** (2012).
- H.Férée, W.Gomaa, H.Hoyrup: "Query Complexity of Real Functionals", Proc. 28th ACM/IEEE Symp. on Logic in Computer Science (LiCS'2013).
- U.Brandt, K.Ambos-Spies, M.Z.: "Real Benefit of Promises and Advice", pp.1-11 in *Proc. 9th Conf. on Computability in Europe* (CiE'2013).



Numerical Engineering

`nag_opt_one_var_deriv (e04bbc)` normally computes
a sequence of x values which tend in the limit
to a minimum of $F(x)$ subject to the given bounds

- + hardware support / large data / high-dim matrices
- heuristics, ad-hoc approaches, unspecified class of permitted inputs, non-guaranteed behavior, vague/inconsistent semantics, various notions of error, not closed under composition, empirical "proofs" of correctness & performance, const.-factor acceleration

"The iterative methods used to solve problems of nonlinear programming differ according to whether they evaluate Hessians, gradients, or only function values. While evaluating Hessians and gradients improves the rate of convergence, such evaluations increase the computational complexity (or computational cost) of each iteration. In some cases, the computational complexity may be excessively high."

Real Complexity Theory: Foundation for the Future of Mathematical Numerics



TECHNISCHE
UNIVERSITÄT
DARMSTADT,
Martin Ziegler

Real Parameterized and 2nd-Order Complexity Theory:

- guaranteed error bounds Bloch, Feigenbaum, Bremble-Hilbert
- high accuracy test of conjectures in classical analysis
- fully specified algorithms with runtime bounds
- consistent semantics closed under composition
- modular software development of certified libraries
- concepts such as multivaluedness and enrichment/information theory (TTE)
 - canonical interface declaration of implementation

Kreinovich, Yap
Revol, Plum,
v.Gudenberg?

Practical proofs-of-concept in iRRAM

Alan Turing was also a Numerical Scientist!
Let's collaborate with, and approach, e.g. the
Interval and Computer-Assisted Proof community