Accurate and efficient expression evaluation and linear algebra, or Why it can be easier to compute accurate eigenvalues of a Vandermonde matrix than the accurate sum of 3 numbers

James Demmel
UC Berkeley
Math and EECS Depts.

Joint work with
Ioana Dumitriu, Olga Holtz, Plamen Koev

Topics.

1. Getting the right answer
   - At all? In polynomial time?
   - Depends on the model of arithmetic

Topics.

1. Getting the right answer
   - At all? In polynomial time?
   - Depends on the model of arithmetic
2. Getting the same answer
   - When running same problem on two different machines?
   - When running same problem twice on same machine?

Topics.

1. Getting the right answer
   - At all? In polynomial time?
   - Depends on the model of arithmetic
2. Getting the same answer
   - When running same problem on two different machines?
   - When running same problem twice on same machine?
3. Getting a fast answer
   - Arithmetic is cheap, moving data is expensive
   - How does this change algorithms?

Topics.

1. Getting the right answer
   - At all? In polynomial time?
   - Depends on the model of arithmetic
2. Getting the same answer
   - When running same problem on two different machines?
   - When running same problem twice on same machine?
3. Getting a fast answer
   - Arithmetic is cheap, moving data is expensive
   - How does this change algorithms?

Motivating Example (1/2)

Def: *Accurate* means relative error less than 1

How do the following 3 kinds of accurate evaluation problems differ in difficulty?

1. Motzkin polynomial $z^6 + x^2y^2(x^2 + y^2 - 3z^2)$, or
   eig(V) with $V_{ij} = x_i^j$, where $0 < x_1 < x_2 < \ldots$
2. Eigenvalues of $\nabla \cdot (\theta \nabla u) + \lambda \rho u = 0$ discretized with the FEM on a triangular mesh, or
   $x + y + z$
3. Determinant of a Toeplitz matrix

Motivating Example (2/2)

Accurate alg. for Motzkin polynomial $p = z^6 + x^2y^2(x^2 + y^2 - 3z^2)$

$$
\begin{array}{rl}
\text{if} & |x-z| \le |x+z| \wedge |y-z| \le |y+z| \\
p = & z^4 \cdot [4((x-z)^2 + (y-z)^2 + (x-z)(y-z))] + \\
& +z^3 \cdot [2(2(x-z)^3 + 5(y-z)(x-z)^2 + 5(y-z)^2(x-z) + \\
& \qquad 2(y-z)^3)] + \\
& +z^2 \cdot [(x-z)^4 + 8(y-z)(x-z)^3 + 9(y-z)^2(x-z)^2 + \\
& \qquad 8(y-z)^3(x-z) + (y-z)^4] + \\
& +z \cdot [2(y-z)(x-z)((x-z)^3 + 2(y-z)(x-z)^2 + \\
& \qquad 2(y-z)^2(x-z) + (y-z)^3] + \\
& +(y-z)^2(x-z)^2((x-z)^2 + (y-z)^2) \\
\text{else} & \text{... 7 more analogous cases}
\end{array}
$$

Can we automate the discovery of such algorithms? Or prove they do not exist, i.e. that extra precision is necessary? How much extra precision?

Getting the right answer: Outline

1. Problem statement and (more) motivating examples
2. Classical Model (CM) and Black-Box Model (BBM) of arithmetic
3. Necessary and sufficient conditions for accurate evaluation in CM
4. Necessary and sufficient conditions for accurate evaluation in BBM
5. Consequences for finite precision arithmetic
6. Is it worth getting the right answer? Conditioning
7. Open problems

Getting the right answer: Outline

1. Problem statement and (more) motivating examples
2. Classical Model (CM) and Black-Box Model (BBM) of arithmetic
3. Necessary and sufficient conditions for accurate evaluation in CM
4. Necessary and sufficient conditions for accurate evaluation in BBM
5. Consequences for finite precision arithmetic
6. Is it worth getting the right answer? Conditioning
7. Open problems

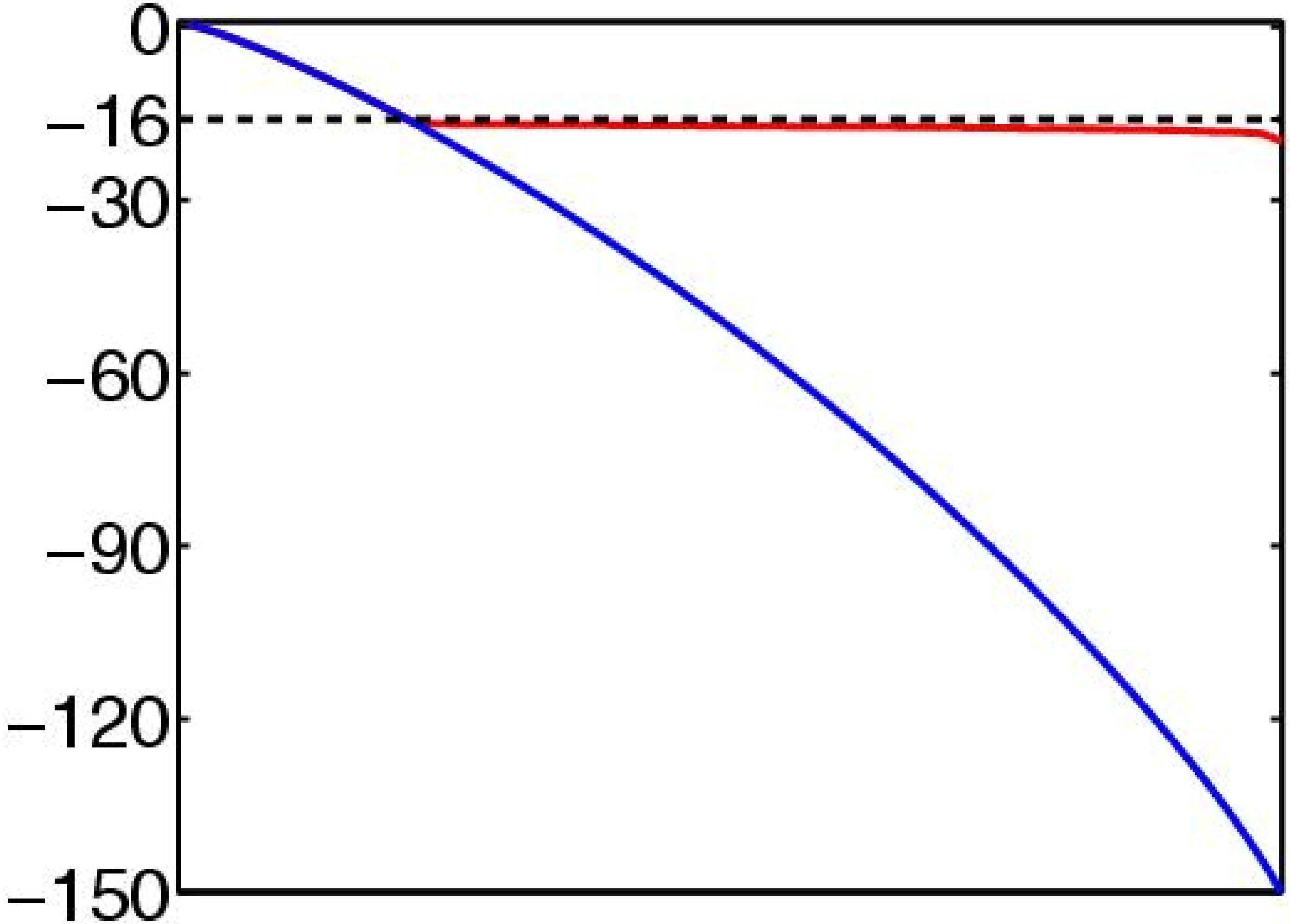
Problem statement

Given a polynomial (or a family of polynomials) p, either produce an **accurate** algorithm to compute $y = p(x)$, or prove that none exists.

Accurately means relative error $\eta < 1$, i.e.

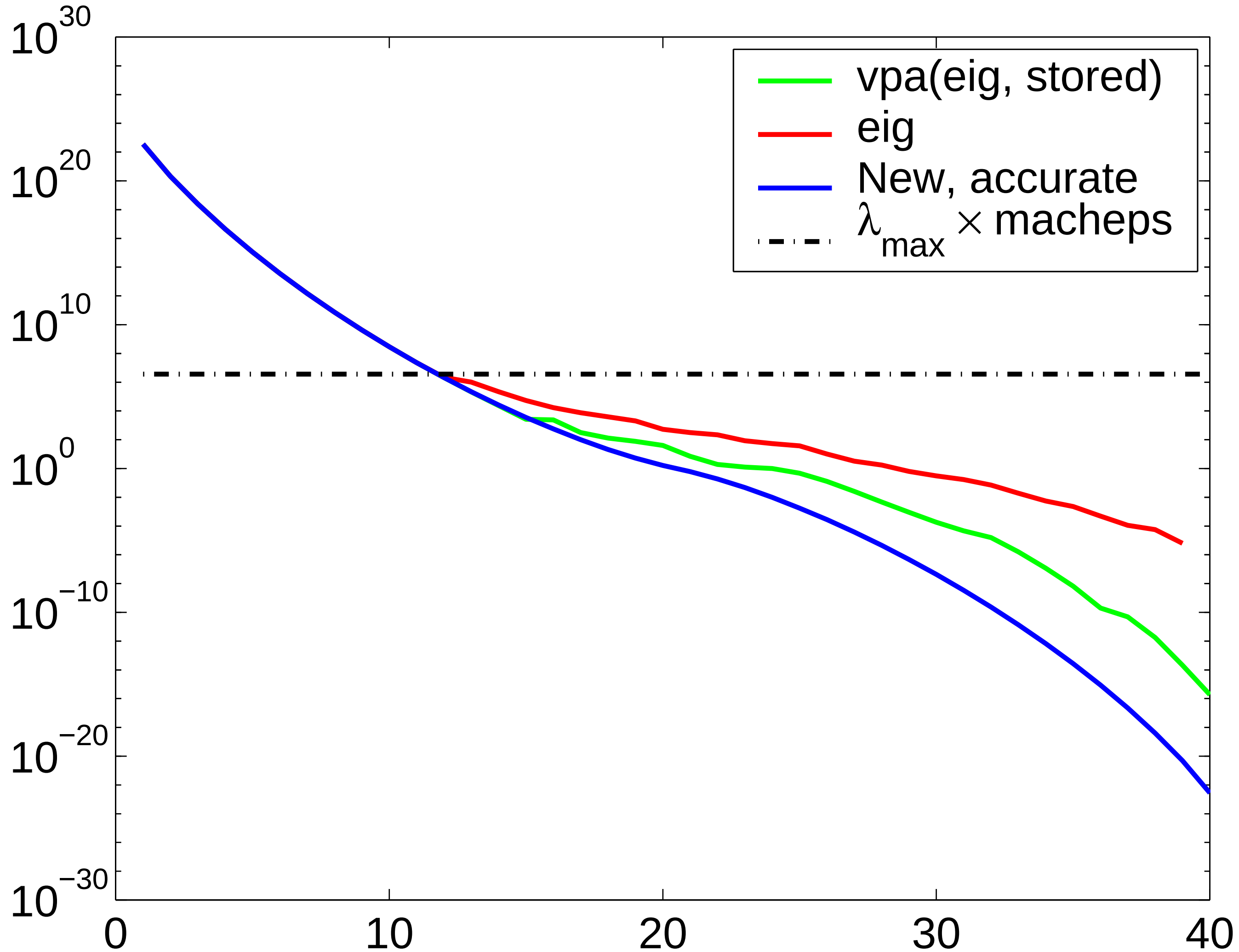
- $|y_{\text{computed}} - y| \leq \eta\, |y|$,
- $\eta = 10^{-2}$ yields two digits of accuracy,
- $y_{\text{computed}} = 0 \iff y = 0$.

50x50 Hilbert Matrix - $\log_{10}$(eigenvalues)



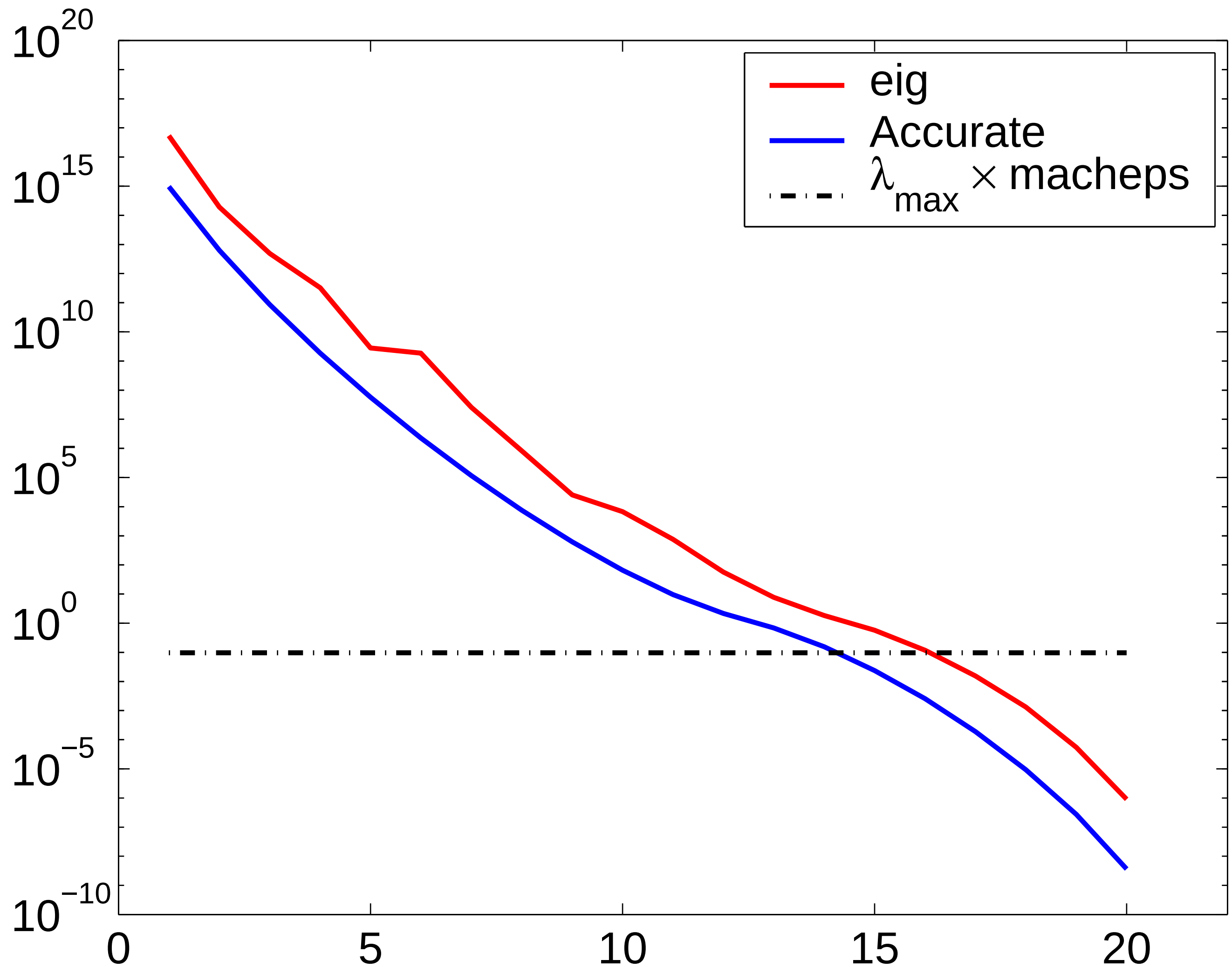
$H_{ij} = 1/(i + j - 1)$

40x40 Pascal Matrix - eigenvalues



$$P_{ij} = \binom{i+j-2}{i-1}$$

20x20 Schur Complement of 40x40 Vandermonde Matrix - eigenvalues



$$V_{ij} = i^{j-1}$$

Complexity of Accurate Algorithms for General Structured Matrices

Type of matrix		$\det A$	A^{-1}	Any minor	LDU	SVD	Sym EVD
Acyclic (bidiagonal and other)		n	n^2	n	$\le n^2$	n^3	N/A
Total Sign Compound (TSC)		n	n^3	n	n^4	n^4	n^4
Diagonally Scaled Totally Unimodular (DSTU)		n^3	n^5?	n^3	n^3	n^3	n^3
Weakly diagonally dominant M-matrix		n^3	n^3	?	n^3	n^3	n^3
Displacement Rank One	Cauchy	n^2	n^2	n^2	$\le n^3$	n^3	n^3
Displacement Rank One	Vandermonde	n^2	?	?	?	n^3	n^3
Displacement Rank One	Polynomial Vandermonde	n^2	?	?	?	?	?
Toeplitz		?	?	?	?	?	?

Complexity of Accurate Algorithms
for Totally Nonnegative (TN) Matrices

Type of Matrix	det A	A^{-1}	Any minor	Gauss. elim. NP	PP	CP	NE NP	Ax=b	SVD	Eig. Val.
Cauchy	n^2	n^2	n^2	n^2	n^3	n^3	n^2	n^2	n^3	n^3
Vandermonde	n^2	n^3	n^3	n^2	n^2	poly	n^2	n^2	n^3	n^3
Generalized Vandermonde	n^2	n^3	poly	n^2	n^2	poly	n^2	n^2	n^3	n^3
Any TN in Neville form	n	n^3	n^3	n^3	n^3	n^3	0	n^2	n^3	n^3

Def: A is Totally Positive (TP) if all minors are positive.
A is Totally Nonnegative (TN) if all minors are nonnegative.

Theorem: The class of TN matrices for which we can do accurate linear algebra in polynomial time is closed under multiplication, taking submatrices, Schur complement, J-inverse and converse.

Getting the right answer: Outline

1. Problem statement and (more) motivating examples
2. Classical Model (CM) and Black-Box Model (BBM) of arithmetic
3. Necessary and sufficient conditions for accurate evaluation in CM
4. Necessary and sufficient conditions for accurate evaluation in BBM
5. Consequences for finite precision arithmetic
6. Is it worth getting the right answer? Conditioning
7. Open problems

Model(s) of Arithmetic.

- All quantities are arbitrary real numbers, or complex numbers (bits come later)
- $fl(a \otimes b) = (a \otimes b)(1+\delta)$, with arbitrary roundoff error $|\delta| < \epsilon \ll 1$
- Operations?

Model(s) of Arithmetic.

- All quantities are arbitrary real numbers, or complex numbers (bits come later)
- $fl(a \otimes b) = (a \otimes b)(1+\delta)$, with arbitrary roundoff error $|\delta| < \epsilon \ll 1$
- Operations?
  - in Classical Model (CM), $+$, $-$, $\times$; also exact negation;

Model(s) of Arithmetic.

- All quantities are arbitrary real numbers, or complex numbers (bits come later)
- $fl(a \otimes b) = (a \otimes b)(1+\delta)$, with arbitrary roundoff error $|\delta| < \epsilon \ll 1$
- Operations?
  - in Classical Model (CM), $+$, $-$, $\times$; also exact negation;
  - in Black-Box Model (BBM), in addition to the above, polynomial expressions (e.g. $x - y \cdot z$ (FMA), $x + y + z$, dot products, small determinants, ...)
- Constants?

Availability of constants?

- Classical Model:
  - without $\sqrt{2}$, we cannot compute
  $$x^2 - 2 = (x - \sqrt{2})(x + \sqrt{2})$$
  accurately.
  – no loss of generality for homogeneous, integer-coefficient polynomials.
- Black-Box Model:
  - any constants we choose can be accommodated.

Model(s) of Arithmetic.

- All quantities are arbitrary real numbers, or complex numbers (bits come later)
- $fl(a \otimes b) = (a \otimes b)(1+\delta)$, with arbitrary roundoff error $|\delta| < \epsilon \ll 1$
- Operations?
  - in Classical Model (CM), $+$, $-$, $\times$; also exact negation;
  - in Black-Box Model (BBM), in addition to the above, polynomial expressions (e.g. $x - y \cdot z$ (FMA), $x + y + z$, dot products, small determinants, ...)
- Constants? none in Classical Model, anything in Black-Box Model.

Model(s) of Arithmetic.

- All quantities are arbitrary real numbers, or complex numbers (bits come later)
- $fl(a \otimes b) = (a \otimes b)(1 + \delta)$, with arbitrary roundoff error $|\delta| < \epsilon \ll 1$
- Operations?
  - in Classical Model (CM), $+$, $-$, $\times$; also exact negation;
  - in Black-Box Model (BBM), in addition to the above, polynomial expressions (e.g. $x - y \cdot z$ (FMA), $x + y + z$, dot products, small determinants, ...)
- Algorithms?
  - exact answer in finite # of steps in absence of roundoff error

Model(s) of Arithmetic.

- All quantities are arbitrary real numbers, or complex numbers (bits come later)
- $fl(a \otimes b) = (a \otimes b)(1+\delta)$, with arbitrary roundoff error $|\delta| < \epsilon \ll 1$
- Operations?
  - in Classical Model (CM), $+$, $-$, $\times$; also exact negation;
  - in Black-Box Model (BBM), in addition to the above, polynomial expressions (e.g. $x - y \cdot z$ (FMA), $x + y + z$, dot products, small determinants, ...)
- Algorithms?
  - exact answer in finite # of steps in absence of roundoff error
  - branching based on comparisons

Model(s) of Arithmetic.

- All quantities are arbitrary real numbers, or complex numbers (bits come later)
- $fl(a \otimes b) = (a \otimes b)(1+\delta)$, with arbitrary roundoff error $|\delta| < \epsilon \ll 1$
- Operations?
  - in Classical Model (CM), $+$, $-$, $\times$; also exact negation;
  - in Black-Box Model (BBM), in addition to the above, polynomial expressions (e.g. $x - y \cdot z$ (FMA), $x + y + z$, dot products, small determinants, ...)
- Algorithms?
  - exact answer in finite # of steps in absence of roundoff error
  - branching based on comparisons
  - non-determinism (because determinism is simulable)

Model(s) of Arithmetic.

- All quantities are arbitrary real numbers, or complex numbers (bits come later)
- $fl(a \otimes b) = (a \otimes b)(1+\delta)$, with arbitrary roundoff error $|\delta| < \epsilon \ll 1$
- Operations?
  - in Classical Model (CM), $+$, $-$, $\times$; also exact negation;
  - in Black-Box Model (BBM), in addition to the above, polynomial expressions (e.g. $x - y \cdot z$ (FMA), $x + y + z$, dot products, small determinants, ...)
- Algorithms?
  - exact answer in finite # of steps in absence of roundoff error
  - branching based on comparisons
  - non-determinism (because determinism is simulable)
  - domains to be $\mathbb{C}^n$ or $\mathbb{R}^n$ (but some domain-specific results).

Problem Statement, formally:

◇ Notation:

– $p(x)$ multivariate polynomial to be evaluated, $x = (x_1, \ldots, x_k)$.

– $\delta = (\delta_1, \ldots, \delta_m)$ is the vector of error (rounding) variables.

– $p_{comp}(x, \delta)$ is the result of algorithm to compute p at x with errors δ.

◇ Goal: Decide if $\exists$ algorithm $p_{comp}(x, \delta)$ to accurately evaluate $p(x)$ on $\mathcal{D}$:

$\forall\, 0 < \eta < 1$ ... for any η = desired relative error

$\exists\, 0 < \epsilon < 1$ ... there is an ϵ = maximum rounding error

$\forall\, x \in \mathcal{D}$ ... so that for all x in the domain

$\forall\, |\delta_i| \le \epsilon$ ... and for all rounding errors bounded by ϵ

$|p_{comp}(x, \delta) - p(x)| \le \eta \cdot |p(x)|$ ... relative error is at most η

◇ Given $p(x)$ and $\mathcal{D}$, seek effective procedure (”compiler”) to exhibit algorithm, or show one does not exist

Examples in classical arithmetic over $\mathbb{R}^n$ (none work over $\mathbb{C}^n$).

- $M_2(x,y,z) = z^6 + x^2 \cdot y^2 \cdot (x^2 + y^2 - 2 \cdot z^2)$
  - Positive definite and homogeneous, easy to evaluate accurately
- $M_3(x,y,z) = z^6 + x^2 \cdot y^2 \cdot (x^2 + y^2 - 3 \cdot z^2)$
  - Motzkin polynomial, nonnegative, zero at $|x| = |y| = |z|$

$$
\begin{array}{ll}
\text{if} & |x-z| \le |x+z| \wedge |y-z| \le |y+z| \\
p = & z^4 \cdot [4((x-z)^2 + (y-z)^2 + (x-z)(y-z))] + \\
& +z^3 \cdot [2(2(x-z)^3 + 5(y-z)(x-z)^2 + 5(y-z)^2(x-z) + \\
& \quad 2(y-z)^3)] + \\
& +z^2 \cdot [(x-z)^4 + 8(y-z)(x-z)^3 + 9(y-z)^2(x-z)^2 + \\
& \quad 8(y-z)^3(x-z) + (y-z)^4] + \\
& +z \cdot [2(y-z)(x-z)((x-z)^3 + 2(y-z)(x-z)^2 + \\
& \quad 2(y-z)^2(x-z) + (y-z)^3] + \\
& +(y-z)^2(x-z)^2((x-z)^2 + (y-z)^2) \\
\text{else} & \ldots\ 2^{\#\text{vars}-1} \text{ more analogous cases}
\end{array}
$$

- $M_4(x,y,z) = z^6 + x^2 \cdot y^2 \cdot (x^2 + y^2 - 4 \cdot z^2)$
  - Impossible to evaluate accurately

Sneak Peak.

The variety,

$$V(p) = \{x : p(x) = 0\} \ ,$$

plays a necessary role.

Getting the right answer: Outline

1. Problem statement and (more) motivating examples
2. Classical Model (CM) and Black-Box Model (BBM) of arithmetic
3. Necessary and sufficient conditions for accurate evaluation in CM
4. Necessary and sufficient conditions for accurate evaluation in BBM
5. Consequences for finite precision arithmetic
6. Is it worth getting the right answer? Conditioning
7. Open problems

Allowable varieties in Classical Model of arithmetic.

Define *basic allowable sets*:

- $Z_i = \{x : x_i = 0\}$,
- $S_{ij} = \{x : x_i + x_j = 0\}$,
- $D_{ij} = \{x : x_i - x_j = 0\}$.

A variety $V(p)$ is *allowable* if it can be written as a finite union of intersections of basic allowable sets.

Denote by

$$\mathbf{G(p) = V(p)} - \cup_{\mathbf{allowable\ A \subset V(p)}} \mathbf{A}$$

the set of points *in general position*.

$V(p)$ unallowable $\quad \Rightarrow \quad G(p) \neq \emptyset$.

Necessary condition.

Theorem 1: $V(p)$ unallowable $\quad \Rightarrow p$ cannot be evaluated accurately on $\mathbb{R}^n$ or on $\mathbb{C}^n$.

Theorem 2: On a domain $\mathcal{D}$, if $\mathrm{Int}(\mathcal{D}) \cap G(p) \neq \emptyset$, p cannot be evaluated accurately.

Examples on $\mathbb{R}^n$, revisited.

- $p(x,y,z) = x + y + z$

  UNALLOWABLE

- $M_2(x,y,z) = z^6 + x^2 \cdot y^2 \cdot (x^2 + y^2 - 2 \cdot z^2)$

  ALLOWABLE, $V(p) = \{0\}$.

- $M_3(x,y,z) = z^6 + x^2 \cdot y^2 \cdot (x^2 + y^2 - 3 \cdot z^2)$

  ALLOWABLE, $V(p) = \{|x| = |y| = |z|\}$

- $M_4(x,y,z) = z^6 + x^2 \cdot y^2 \cdot (x^2 + y^2 - 4 \cdot z^2)$

  UNALLOWABLE

- $V(\det(\text{Toeplitz}))$, UNALLOWABLE $\Rightarrow$ no accurate linear algebra for Toeplitz (need arbitrary precision arithmetic, as we will see later).

Necessary condition, real and complex.

Theorem 1: $V(p)$ unallowable $\Rightarrow p$ cannot be evaluated accurately on $\mathbb{R}^n$ or on $\mathbb{C}^n$.

Theorem 2: On a domain $\mathcal{D}$, if $\mathrm{Int}(\mathcal{D}) \cap G(p) \neq \emptyset$, p cannot be evaluated accurately.

Sketch of proof.

Simplest case: non-branching, no data reuse (except for inputs), non-determinism.

Algorithm can be represented as a tree with extra edges from the sources, each node corresponds to an operation $(+, -, \times)$, each node has a specific δ, each node has two inputs, one output.

Let $x \in G(p)$ and define $Allow(x)$ as the smallest allowable set containing x.

Necessary condition, real and complex.

Theorem 1: $V(p)$ unallowable $\Rightarrow p$ cannot be evaluated accurately on $\mathbb{R}^n$ or on $\mathbb{C}^n$.

Theorem 2: On a domain $\mathcal{D}$, if $\mathrm{Int}(\mathcal{D}) \cap G(p) \neq \emptyset$, p cannot be evaluated accurately.

Sketch of proof, cont'd.

Key fact: for a positive measure set of δs in δ-space, a zero output can be "traced back" down the tree to "allowable" condition ($x_i = 0$ or $x_i + x_j = 0$), or trivial one ($x_i - x_i = 0$).

So for a positive measure set of δs, either

- $p_{comp}(x, \delta)$ is not 0 (though $p(x) = 0$), or
- for all $y \in Allow(x) \setminus V(p)$, $p_{comp}(y, \delta) = 0$ (though $p(y) \neq 0$).

In either case, the polynomial is not accurately evaluable arbitrarily close to x, q.e.d.

Sufficient Condition, complex case.

Theorem. Let p be a polynomial over $\mathbb{C}^n$ with integer coefficients. If $V(p)$ is allowable, then p is accurately evaluable.

Sketch of proof.
Can write

$$p(x) = c \prod_i p_i(x) \ ,$$

where $p_i(x)$ is a power of some x_j or $x_j \pm x_k$, and c is an integer; all operations are accurate.

Sufficient Condition, complex case.

Theorem. Let p be a polynomial over $\mathbb{C}^n$ with integer coefficients. If $V(p)$ is allowable, then p is accurately evaluable.

Sketch of proof.
Can write

$$p(x) = c \prod_i p_i(x) \ ,$$

where $p_i(x)$ is a power of some x_j or $x_j \pm x_k$, and c is an integer; all operations are accurate.

Corollary. If p is a complex multivariate polynomial, p is accurately evaluable iff p has integer coefficients and $V(p)$ is allowable.

Sufficient Condition, real case.

Trickier... Allowability *not* sufficient:

- $q = (u^4 + v^4) + (u^2 + v^2)(x^2 + y^2 + z^2)$, $V(p) = \{u = v = 0\}$: allowable and accurately evaluable
- $p = (u^4 + v^4) + (u^2 + v^2)(x + y + z)^2$, $V(p) = \{u = v = 0\}$: allowable but NOT accurately evaluable!
- Has to do with locally dominant behavior (in this case, near the set $\{u = v = 0\}$).

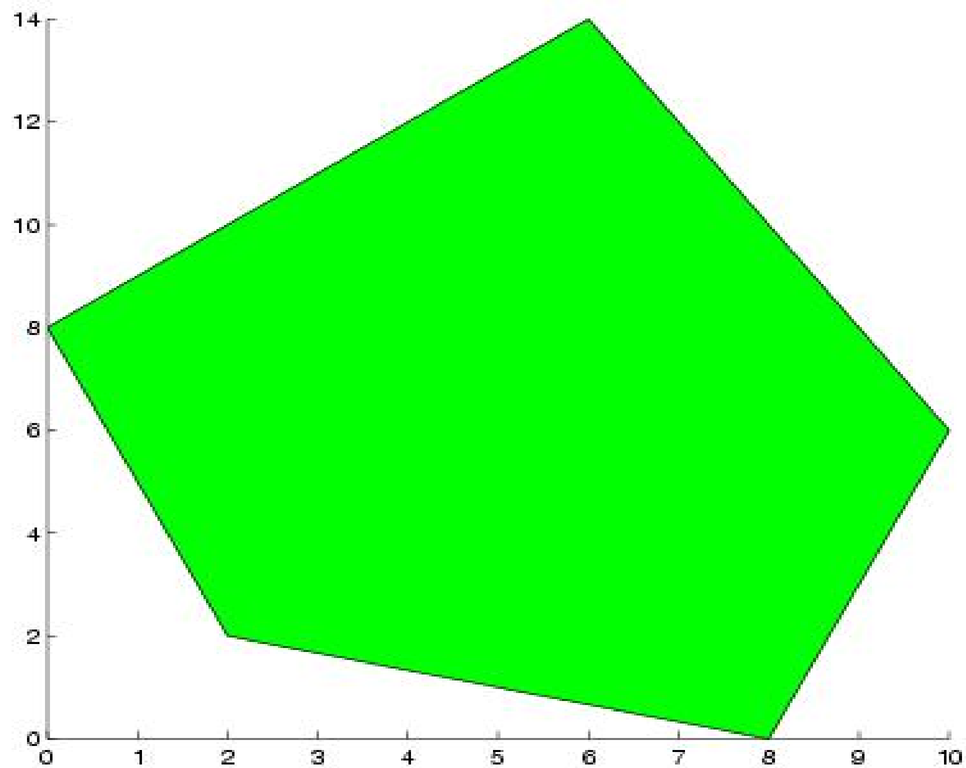
Sufficient Condition, real case.

Trickier... Allowability *not* sufficient:

- $q = (u^4 + v^4) + (u^2 + v^2)(x^2 + y^2 + z^2)$, $V(p) = \{u = v = 0\}$: allowable and accurately evaluable
- $p = (u^4 + v^4) + (u^2 + v^2)(x + y + z)^2$, $V(p) = \{u = v = 0\}$: allowable but NOT accurately evaluable!
- Has to do with locally dominant behavior (in this case, near the set $\{u = v = 0\}$).

Theorem. If all "dominant terms" are accurately evaluable on $\mathbb{R}^n$ then p is accurately evaluable. In non-branching case, if p is accurately evaluable on $\mathbb{R}^n$, then so are all "dominant terms".

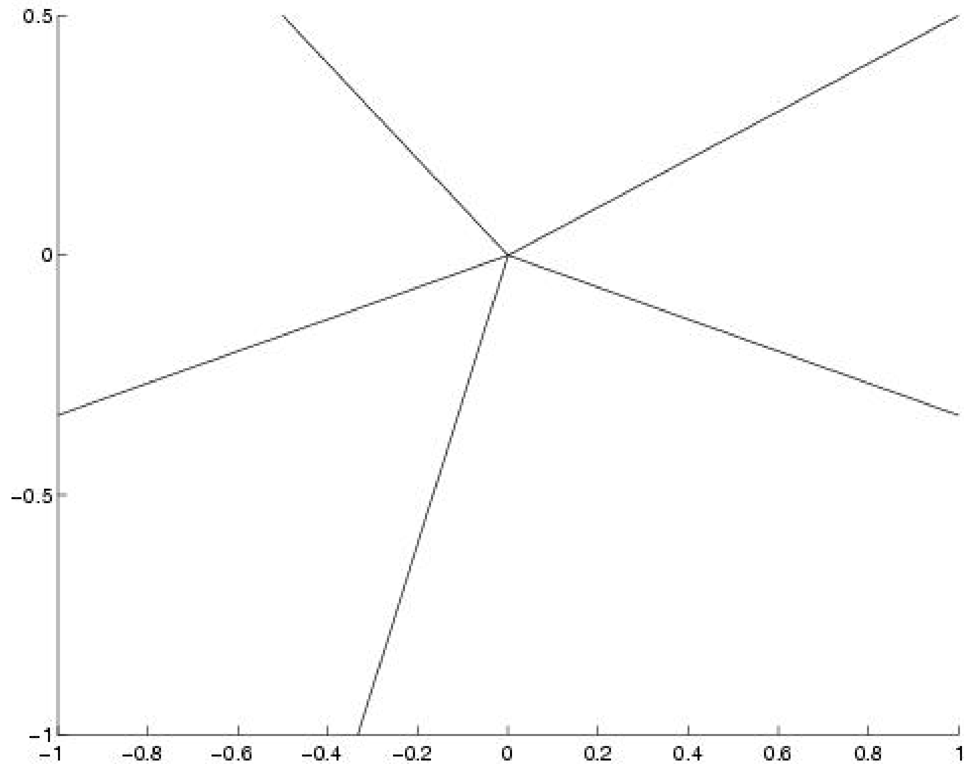
What is dominance? Newton Polytope



$$p(x, y, z) = y^8 z^{12} + x^2 y^2 z^{16} + x^8 z^{12} + x^6 y^{14} + x^{10} y^6 z^4$$

Component of $V(p)$ where $\{x = y = 0\}$

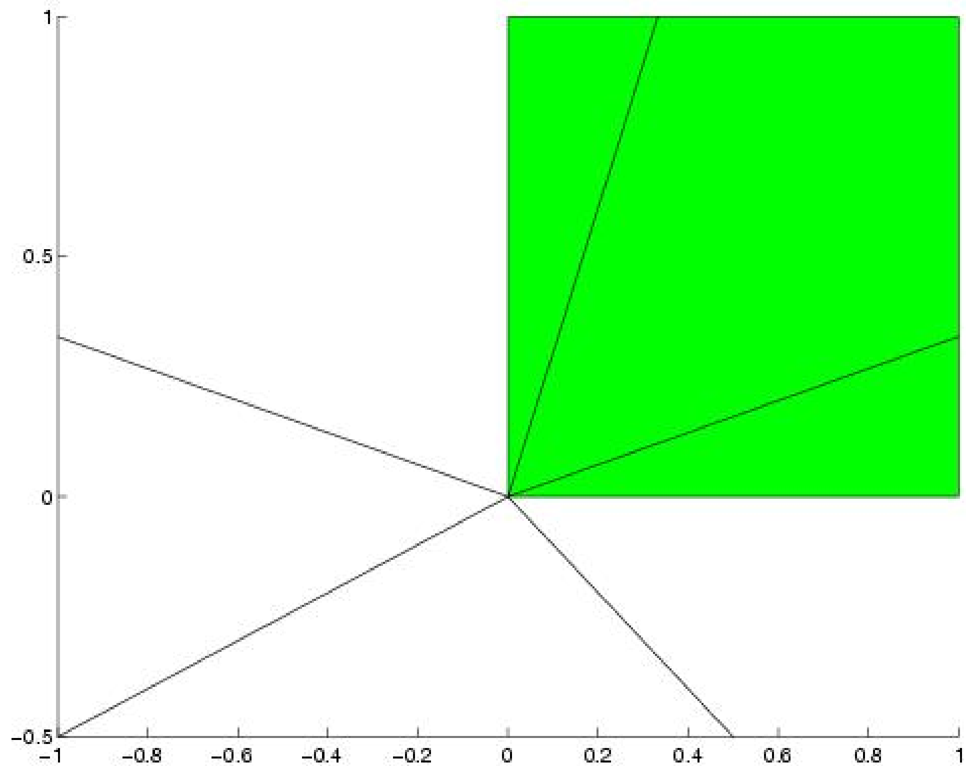
What is dominance? Normal Fan



$$p(x, y, z) = y^8 z^{12} + x^2 y^2 z^{16} + x^8 z^{12} + x^6 y^{14} + x^{10} y^6 z^4$$

Component of $V(p)$ where $\{x = y = 0\}$

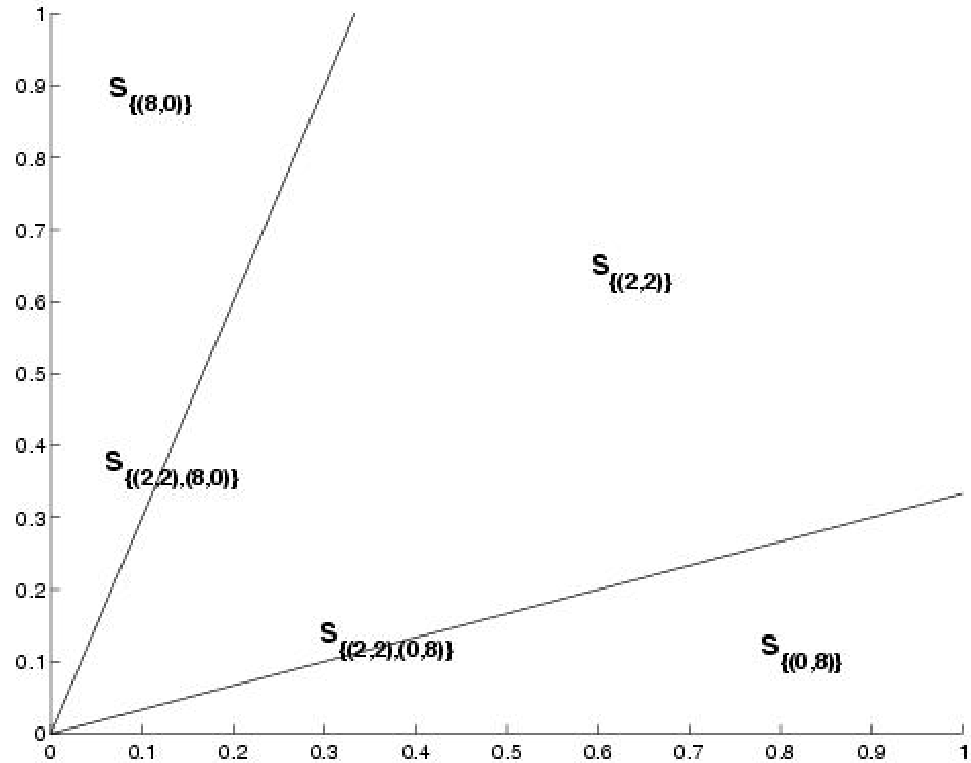
What is dominance? First orthant of -(Normal Fan)



$$p(x, y, z) = y^8 z^{12} + x^2 y^2 z^{16} + x^8 z^{12} + x^6 y^{14} + x^{10} y^6 z^4$$

Component of $V(p)$ where $\{x = y = 0\}$

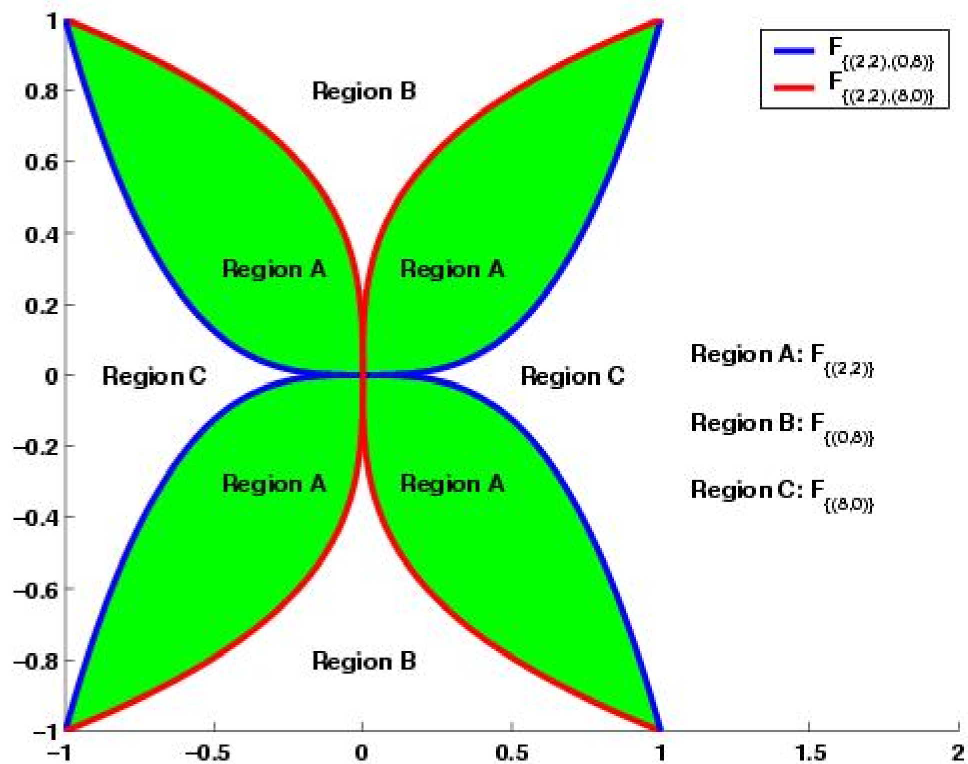
What is dominance? Labeling dominant terms



$$p(x, y, z) = y^8 z^{12} + x^2 y^2 z^{16} + x^8 z^{12} + x^6 y^{14} + x^{10} y^6 z^4$$

Component of $V(p)$ where $\{x = y = 0\}$

What is dominance? (x, y) regions where different terms dominant



$$p(x, y, z) = y^8 z^{12} + x^2 y^2 z^{16} + x^8 z^{12} + x^6 y^{14} + x^{10} y^6 z^4$$

Component of $V(p)$ where $\{x = y = 0\}$

Sufficient Condition, real case.

Trickier... Allowability *not* sufficient:

- $q = (u^4 + v^4) + (u^2 + v^2)(x^2 + y^2 + z^2)$, $V(p) = \{u = v = 0\}$: allowable and accurately evaluable
- $p = (u^4 + v^4) + (u^2 + v^2)(x + y + z)^2$, $V(p) = \{u = v = 0\}$: allowable but NOT accurately evaluable!
- Has to do with locally dominant behavior (in this case, near the set $\{u = v = 0\}$).

Theorem. If all "dominant terms" are accurately evaluable on $\mathbb{R}^n$ then p is accurately evaluable. In non-branching case, if p is accurately evaluable on $\mathbb{R}^n$, then so are all "dominant terms".

Need inductive procedure of testing accurate evaluability, but so far no clear induction parameter.

Getting the right answer: Outline

1. Problem statement and (more) motivating examples
2. Classical Model (CM) and Black-Box Model (BBM) of arithmetic
3. Necessary and sufficient conditions for accurate evaluation in CM
4. Necessary and sufficient conditions for accurate evaluation in BBM
5. Consequences for finite precision arithmetic
6. Is it worth getting the right answer? Conditioning
7. Open problems

Allowable varieties in black-box arithmetic.

Define **black-boxes**: $q_1, q_2, \dots, q_k$ are polynomials

$\mathcal{V}_j = \{V \neq \mathbb{R}^n : V$ can be obtained from q_j through **Process A**, below$\}$

Process A:

Step 1. repeat and/or negate, or 0 out some of the inputs,

Step 2. of the remaining variables, keep some symbolic, and find the variety in terms of the others.

Example: $q_1(x, y) = x - y$ has (up to symmetry)

$$\mathcal{V}_1 = \{\{x = 0\}, \{x - y = 0\}, \{x + y = 0\}\} ,$$

$$q_2(x, y, z) = x - y \cdot z \text{ has (up to symmetry)}$$

$$\begin{aligned}\mathcal{V}_2 = \{&\{x = 0\},\ \{y = 0\} \cup \{z = 0\},\ \{x = 0\} \cup \{x = 1\},\ \{x = 0\} \cup \{x = -1\},\\ &\{x = 0\} \cup \{y = 1\},\ \{x = 0\} \cup \{y = -1\},\ \{x - y^2 = 0\},\ \{x + y^2 = 0\},\\ &\{x - yz = 0\},\ \{x + yz = 0\}\} .\end{aligned}$$

Allowable varieties in black-box arithmetic.

Define **black-boxes**: $q_1, q_2, \ldots, q_k$ are polynomials

$\mathcal{V}_j = \{V \neq \mathbb{R}^n : V$ can be obtained from q_j through **Process A**$\}$

Define *basic allowable sets*:

- $Z_i = \{x : x_i = 0\}$,
- $S_{ij} = \{x : x_i + x_j = 0\}$,
- $D_{ij} = \{x : x_i - x_j = 0\}$,
- any V for which there is a j such that $V \in \mathcal{V}_j$.

Allowable varieties in black-box arithmetic.

Define **black-boxes**: $q_1, q_2, \dots, q_k$ are polynomials

$\mathcal{V}_j = \{V \neq \mathbb{R}^n : V$ can be obtained from q_j through **Process A**$\}$

A variety $V(p)$ is *allowable* if it is a union of irreducible parts of finite intersections of basic allowable sets.

Denote by

$$\mathbf{G}(\mathbf{p}) = \mathbf{V}(\mathbf{p}) - \cup_{\text{allowable } \mathbf{A} \subset \mathbf{V}(\mathbf{p})} \mathbf{A}$$

the set of points *in general position.*

$V(p)$ unallowable $\quad \Rightarrow \quad G(p) \neq \emptyset.$

Necessary condition, real and complex.

Theorem 1: $V(p)$ unallowable $\Rightarrow p$ cannot be evaluated accurately on $\mathbb{R}^n$ or on $\mathbb{C}^n$.

Theorem 2: On a domain $\mathcal{D}$, if $\mathrm{Int}(\mathcal{D}) \cap G(p) \neq \emptyset$, p cannot be evaluated accurately.

Sufficient condition, complex, for all q_j irreducible.

Theorem: If $V(p)$ is a union of intersections of sets Z_i, S_{ij}, D_{ij}, and $V(q_j)$, then p **is** accurately evaluable.

Corollary: If all q_j are affine, then p is accurately evaluable iff $V(p)$ is allowable.

Consequences for Numerical Linear Algebra.

- $V(\det(\text{Toeplitz}))$ contains irreducible factors of arbitrarily large degree $\Rightarrow$ no set of black-boxes of bounded degree will be sufficient for accurate evaluation $\Rightarrow$ need arbitrary precision arithmetic to do NLA accurately on Toeplitz matrices.
- Same argument shows that we cannot accurately evaluate many generalized Vandermonde matrices (Schur functions as determinants).
- **Conjecture:** if the class of structured matrices has displacement rank ≥ 2, then accurate evaluation will not always be possible.

Complexity of Accurate Algorithms for General Structured Matrices

Type of matrix		$\det A$	A^{-1}	Any minor	LDU	SVD	Sym EVD
Acyclic (bidiagonal and other)		n	n^2	n	$\leq n^2$	n^3	N/A
Total Sign Compound (TSC)		n	n^3	n	n^4	n^4	n^4
Diagonally Scaled Totally Unimodular (DSTU)		n^3	n^5?	n^3	n^3	n^3	n^3
Weakly diagonally dominant M-matrix		n^3	n^3	No	n^3	n^3	n^3
Displacement Rank One	Cauchy	n^2	n^2	n^2	$\leq n^3$	n^3	n^3
	Vandermonde	n^2	No	No	No	n^3	n^3
	Polynomial Vandermonde	n^2	No	No	No	*	*
Toeplitz		No	No	No	No	No	No

$*$ = "it depends"

Getting the right answer: Outline

1. Problem statement and (more) motivating examples
2. Classical Model (CM) and Black-Box Model (BBM) of arithmetic
3. Necessary and sufficient conditions for accurate evaluation in CM
4. Necessary and sufficient conditions for accurate evaluation in BBM
5. Consequences for finite precision arithmetic
6. Is it worth getting the right answer? Conditioning
7. Open problems

Choosing a finite precision arithmetic model

- In finite precision, accuracy always possible, only question is cost
- Measure bit complexity in floating point: $(e, m) \equiv 2^e \cdot m$
- Contrasts between complexity in Floating Point and Fixed Point
  - Repeated squaring can have exponential cost in Fixed, polynomial in Float
  - Det(A) polynomial in Fixed [Clarkson], unknown in Float
  - Witness for matrix singularity (null vector) in Float can have exponentially many bits
  - Computing "middle bits" of $\prod_{i=1}^{n}(1+x_i)$ polynomial in Fixed, as hard as computing the permanent in Float
- Float seems more natural for attaining relative accuracy

Cost implications for Accuracy in Floating Point

- If a problem is accurately evaluable in Classical Model, the same algorithm works in Float
  - Each operation runs in in polynomial time in size of inputs
  - Ex: Motzkin polynomial, eig(Vandermonde)
- If a problem is accurately evaluable in Black-Box Model, then if you build an accurate library to evaluate each "black-box" operation, the same algorithm works in Float
  - If each "black-box" operation is of bounded degree and #terms, then each operation runs in polynomial time in size of inputs
  - Ex: eig(discretized scalar elliption PDE), $x + y + z$
  - If set of black-box operations of unbounded degree and #terms, then cost may be exponential
  - Ex: det(Toeplitz)

Getting the right answer: Outline

1. Problem statement and (more) motivating examples
2. Classical Model (CM) and Black-Box Model (BBM) of arithmetic
3. Necessary and sufficient conditions for accurate evaluation in CM
4. Necessary and sufficient conditions for accurate evaluation in BBM
5. Consequences for finite precision arithmetic
6. Is it worth getting the right answer? Conditioning
7. Open problems

Is it worth getting the right answer?

- If the answer is ill-conditioned, why bother?
- Many problems have enormous condition numbers, but moderate *structured* conditioned numbers
  - Ex: Hilbert matrix: exponentially large condition number as n grows
  - But $H_{ij} = 1/(x_i + x_j)$ with $x_i = i - .5$ well-conditioned wrt $x()$.
- True for all structured matrix examples
- So high accuracy deserved!

Getting the right answer: Outline

1. Problem statement and (more) motivating examples
2. Classical Model (CM) and Black-Box Model (BBM) of arithmetic
3. Necessary and sufficient conditions for accurate evaluation in CM
4. Necessary and sufficient conditions for accurate evaluation in BBM
5. Consequences for finite precision arithmetic
6. Is it worth getting the right answer? Conditioning
7. Open problems

Open problems

- **Complete** the decision procedure (analyze the dominant terms) when the domain is $\mathbb{R}^n$ and $V(p)$ allowable.
- **Narrow** the necessity and sufficiency conditions for the black-box case
- **Extend** to semi-algebraic domains $\mathcal{D}$.
- **Conjecture:** Same sufficient conditions for existence of accurate interval algorithms
- **Incorporate** division, rational functions, perturbation theory.
  - **Conjecture** (Demmel, '04): Accurate evaluation is possible only if condition number has only certain simple singularities (depend on reciprocal distance to set of ill-posed problems).
- **Implement** decision procedure to "compile" an accurate evaluation program given $p(x)$, $\mathcal{D}$, and minimal set of "black boxes"

Reference

For a survey with many other references, see:

"Accurate and efficient expression evaluation and linear algebra,"
J. Demmel, I. Dumitriu, O. Holtz, P. Koev,
Acta Numerica (2008), v. 17, pp 87-145

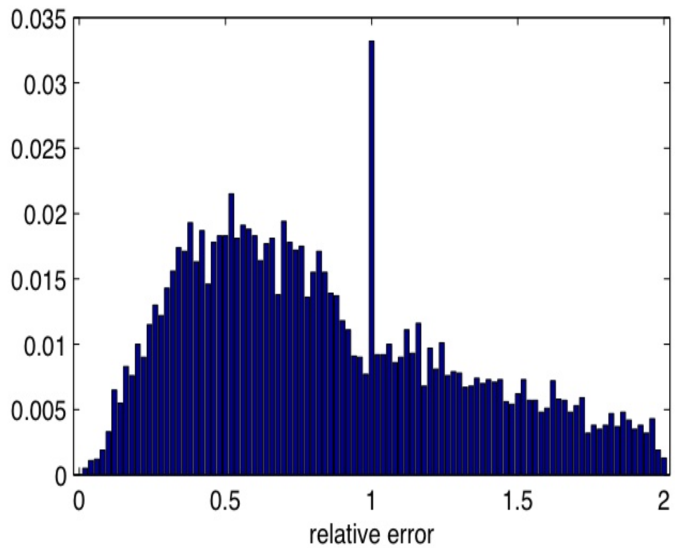
Topics.

1. Getting the right answer
   - At all? In polynomial time?
   - Depends on the model of arithmetic
2. Getting the same answer
   - When running same problem on two different machines?
   - When running same problem twice on same machine?
3. Getting a fast answer
   - Arithmetic is cheap, moving data is expensive
   - How does this change algorithms?

Why wouldn't you get the same answer?

- Run same program twice on different machines (reproducibility)
  - Different floating point semantics, compilers, ...
- Run same program twice on same machine (repeatability)
  - Floating point nonassociativity and dynamic scheduling of parallel tasks
- Who cares?
  - NA-Digest request for reproducible parallel sparse linear solver for use in a FEM package used by construction engineers with contractual obligations for repeatability
  - Subsequent informal survey of many users gave wide range of reasons for wanting repeatability
  - Debugging

Intel MKL is not repeatable



- Experiment:
  - Compute dot products of nearly orthogonal vectors, $n = 1000$.
  - Vary #thread (1-4), alignments
  - Histogram $[\max_i v_i - \min_i v_i] / \max_i |v_i|$
- Repeatability possible, question is cost [H-D. Nguyen]
  - Cost so far: 2x for $n = 1000$, 1.2x for $n = 10^5$

Topics.

1. Getting the right answer
   - At all? In polynomial time?
   - Depends on the model of arithmetic
2. Getting the same answer
   - When running same problem on two different machines?
   - When running same problem twice on same machine?
3. Getting a fast answer
   - Arithmetic is cheap, moving data is expensive
   - How does this change algorithms?

Arithmetic is cheap, moving data is expensive.

- Time to do one floating point operation already hundreds of times faster than getting data from main memory, or from another processor
- Technology trends
  - Arithmetic getting faster at $\approx 60\%$/year
  - Communication bandwidth (moving data, either between levels of memory hierarchy or processors over a network) only improving at most 25%/year;
  - Latency worse
  - Similar trends for energy

Impact on Linear Algebra

- Impact on Direct Linear Algebra (LU, QR, eig, SVD, ...)
  - Thm (Ballard, D., Holtz, Schwartz): Lower bound on communication for *any* of these problems
    - Generalizes existing lower bounds for dense matmul
    - Dense or sparse matrices, sequential or parallel
  - LAPACK/ScaLAPACK communicate asymptotically more than lower bounds
  - New algorithms do attain lower bounds - large speedups
    - Up to 13x measured (or ∞x), 29x predicted

Impact on Linear Algebra

- Impact on Direct Linear Algebra (LU, QR, eig, SVD, ...)
  - Thm (Ballard, D., Holtz, Schwartz): Lower bound on communication for *any* of these problems
    - Generalizes existing lower bounds for dense matmul
    - Dense or sparse matrices, sequential or parallel
  - LAPACK/ScaLAPACK communicate asymptotically more than lower bounds
  - New algorithms do attain lower bounds - large speedups
    - Up to 13x measured (or ∞x), 29x predicted
- Impact on Iterative Linear Algebra (Krylov Methods) - Ditto
- See bebop.cs.berkeley.edu for papers, www.cs.berkeley.edu/∼demmel for short course

Conclusion

Time to reengineer all linear algebra software!

Conclusion

Time to reengineer all linear algebra software!

Don't communic....