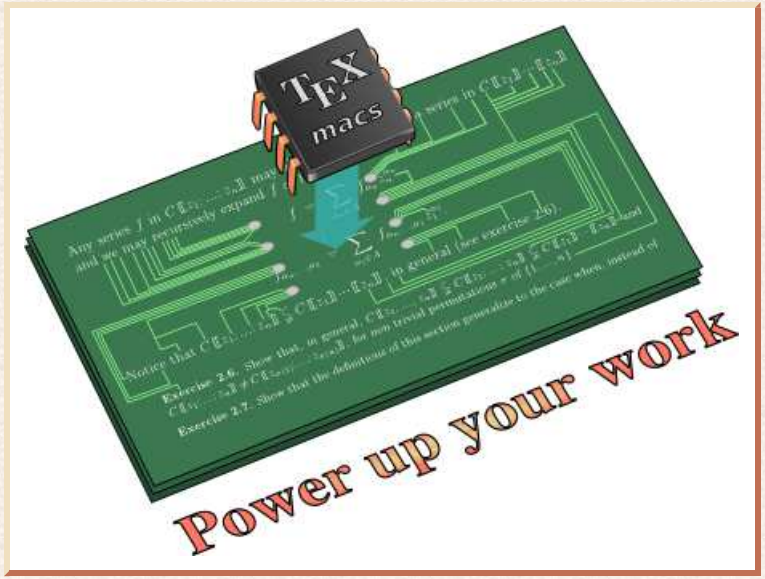
On the practical complexity of computations with real numbers

Joris van der Hoeven

CNRS, École polytechnique



CiE, Milano, 2013

http://www.TEXMACS.org

Computer algebra *and* analysis system

Computer algebra

Exact computations with algebraic, arithmetic or combinatorial objects

Computer algebra *and* analysis system

Computer algebra

Exact computations with algebraic, arithmetic or combinatorial objects

Numerical analysis

Approximate numerical computations with (generally) manual error estimates

```
maple> 1.00000000000000000000001 - 1;
                                0.
```

Computer algebra *and* analysis system

Computer algebra

Exact computations with algebraic, arithmetic or combinatorial objects

Numerical analysis

Approximate numerical computations with (generally) manual error estimates

```
maple> 1.000000000000000000000001 - 1;
                                0.
```

Computer analysis

Mathematically exact computations with analytic objects.

Example 1: certified integration

Pendulum

$$\ddot{y} = -\alpha \sin y,$$

For $\alpha = 1$, $y(0) = 0$, $\dot{y}(0) = 1$, compute $\tilde{y} \in \mathbb{Q}$ (where $\tilde{y} \in \mathbb{D} = \mathbb{Z}\,2^{\mathbb{Z}}$) with

$$|\tilde{y} - y(2013)| < 2^{-2013}.$$

Example 1: certified integration

Dynamical systems

Let $P = (P_1, ..., P_d) \in \mathbb{Q}[Y]^d = \mathbb{Q}[Y_1, ..., Y_d]^d$ and consider

$$\dot{Y} = P(Y).$$

For $Y(0) \in \mathbb{Q}^d$, $t \in \mathbb{Q}^>$ and $\varepsilon \in \mathbb{Q}^>$, compute (when possible) $\tilde{Y} \in \mathbb{Q}^d$ with

$$\|\tilde{Y} - Y(t)\| < \varepsilon.$$

Example 1: certified integration

Dynamical systems

Let $P = (P_1, \ldots, P_d) \in \mathbb{Q}[Y]^d = \mathbb{Q}[Y_1, \ldots, Y_d]^d$ and consider

$$\dot{Y} = P(Y).$$

For $Y(0) \in \mathbb{Q}^d$, $t \in \mathbb{Q}^>$ and $\varepsilon \in \mathbb{Q}^>$, compute (when possible) $\tilde{Y} \in \mathbb{Q}^d$ with

$$\|\tilde{Y} - Y(t)\| < \varepsilon.$$

Special functions

- Elementary functions $\exp, \log, \cos, \sin, \tan$, etc.
- Holonomic functions: $L(f) = L_r f^{(r)} + \cdots + L_0 f = 0$, $L \in \mathbb{Q}[z][\partial]$.
- Other : $\wp$, Γ, ζ, theta fonctions, Matthieu functions, etc.

Example 2: polynomial system solving

Given a zero dimensional system

$$\begin{cases} P_1(z_1, \ldots, z_n) = 0 \\ \vdots \\ P_n(z_1, \ldots, z_n) = 0 \end{cases}$$

with $P_1, \ldots, P_n \in \mathbb{Q}[z_1, \ldots, z_n]$, find the zeros "in $\mathbb{C}^n$".

Example 2: polynomial system solving

Given a zero dimensional system

$$\begin{cases} P_1(z_1, \ldots, z_n) = 0 \\ \vdots \\ P_n(z_1, \ldots, z_n) = 0 \end{cases}$$

with $P_1, \ldots, P_n \in \mathbb{E}[z_1, \ldots, z_n]$, find the zeros "in $\mathbb{C}^n$".

$\mathbb{E}$ = "field of exp-log constants", constructed from $\mathbb{Q}$ using $+, -, \times, \exp, \log$.

Example 2: polynomial system solving

Given a zero dimensional system

$$\begin{cases} P_1(z_1, \ldots, z_n) = 0 \\ \vdots \\ P_n(z_1, \ldots, z_n) = 0 \end{cases}$$

with $P_1, \ldots, P_n \in \mathbb{E}[z_1, \ldots, z_n]$, find the zeros "in $\mathbb{R}^n$".

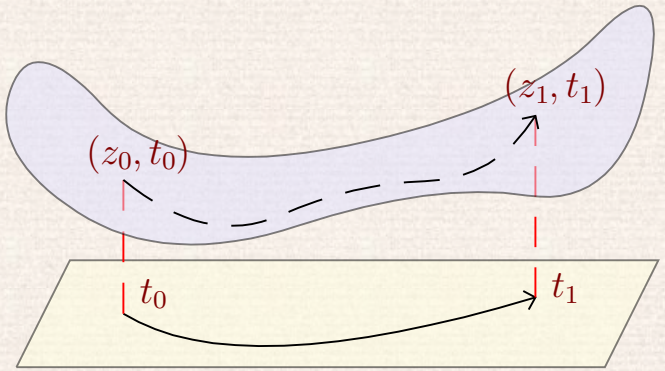
$\mathbb{E}$ = "field of exp-log constants", constructed from $\mathbb{Q}$ using $+, -, \times, \exp, \log$.

Example 3: path following

$$H(z,t)=0 \qquad \begin{cases} H_1(z_1,\ldots,z_n,t)=0 \\ \vdots \\ H_n(z_1,\ldots,z_n,t)=0 \end{cases},$$

with $H(z,t)\in\mathbb{Q}[z,t]^n$ zero-dimensional in z for $t\in\mathbb{C}\backslash\Sigma$ and finite Σ.

Given (z_0,t_0) avec $H(z_0,t_0)=0$ and a path $t_0\rightsquigarrow t_1$ which avoids Σ, compute the path $z_0\rightsquigarrow z_1$ with $H(z_\lambda,t_\lambda)=0$ for all $\lambda\in[0,1]$.



Example 3: path following, application

Polynomial system solving using numerical homotopies

$$P(x,y) \quad \begin{cases} x^2+2\,x\,y-y^2-3\,x+5\,y+8=0 \\ 3\,x^2-x\,y+y^2+8\,x-2\,y+7=0 \end{cases}$$

$$\text{Easy}(x,y) \quad \begin{cases} x^2-1=0 \\ y^2-1=0 \end{cases}$$

$$H(x,y,t) \quad \begin{cases} (x^2-1)\,t+(x^2+2\,x\,y-y^2-3\,x+5\,y+8)\,(1-t)=0 \\ (y^2-1)\,t+(3\,x^2-x\,y+y^2+8\,x-2\,y+7)\,(1-t)=0 \end{cases}$$

Example 4: Bell numbers

$$B(z) \;=\; \sum_{n=0}^{\infty} \frac{B_n}{n!}\, z^n \;=\; \mathrm{e}^{\mathrm{e}^z - 1}.$$

Problem 1: compute B_{10000} with a relative error 10^{-10}

Problem 2: automatically find/guess asymptotic expansion

$$\log\left(\frac{B_n}{n!}\right) \;=\; n\left(-\log\log n + \frac{1}{\log n} + \frac{\log\log n}{\log n} + \mathcal{O}\left(\frac{1}{\log^2 n}\right)\right)$$

Problem 3: for $n \geqslant 10^{10}$, find an explicit constant for the $\mathcal{O}\left(\frac{1}{\log^2 n}\right)$ term

Challenges

- Which problems can be solved, at least theoretically?
- Automatic computation of error bounds for numerical computations
- Redevelop numerical analysis for multiple precision computations

Challenges

- Which problems can be solved, at least theoretically?

  Computable analysis, BSS computability, ...
- Automatic computation of error bounds for numerical computations
- Redevelop numerical analysis for multiple precision computations

Challenges

- Which problems can be solved, at least theoretically?

  Computable analysis, BSS computability, ...
- Automatic computation of error bounds for numerical computations

  Interval analysis, ball arithmetic
- Redevelop numerical analysis for multiple precision computations

Challenges

- Which problems can be solved, at least theoretically?

  Computable analysis, BSS computability, ...

- Automatic computation of error bounds for numerical computations

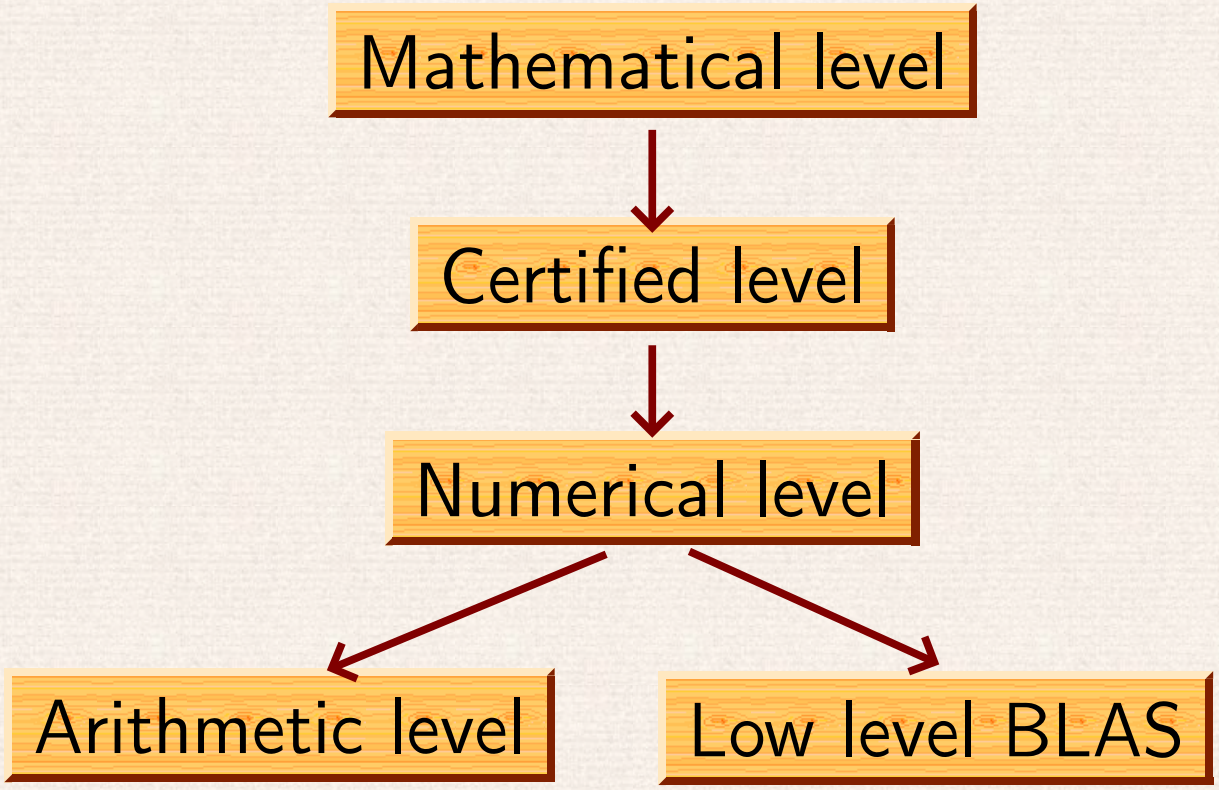
  Interval analysis, ball arithmetic

- Redevelop numerical analysis for multiple precision computations

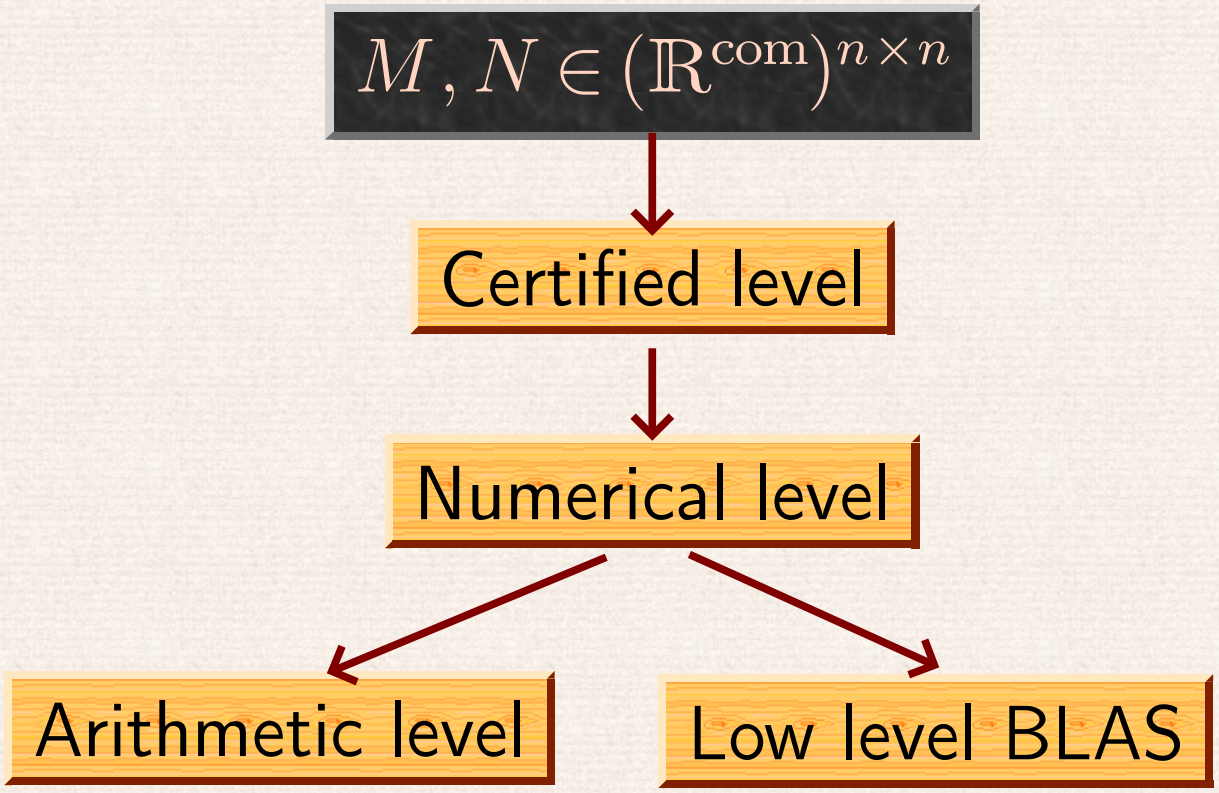
  Runge Kutta $\rightsquigarrow$ Taylor methods

  Complexity, conditioning, ...

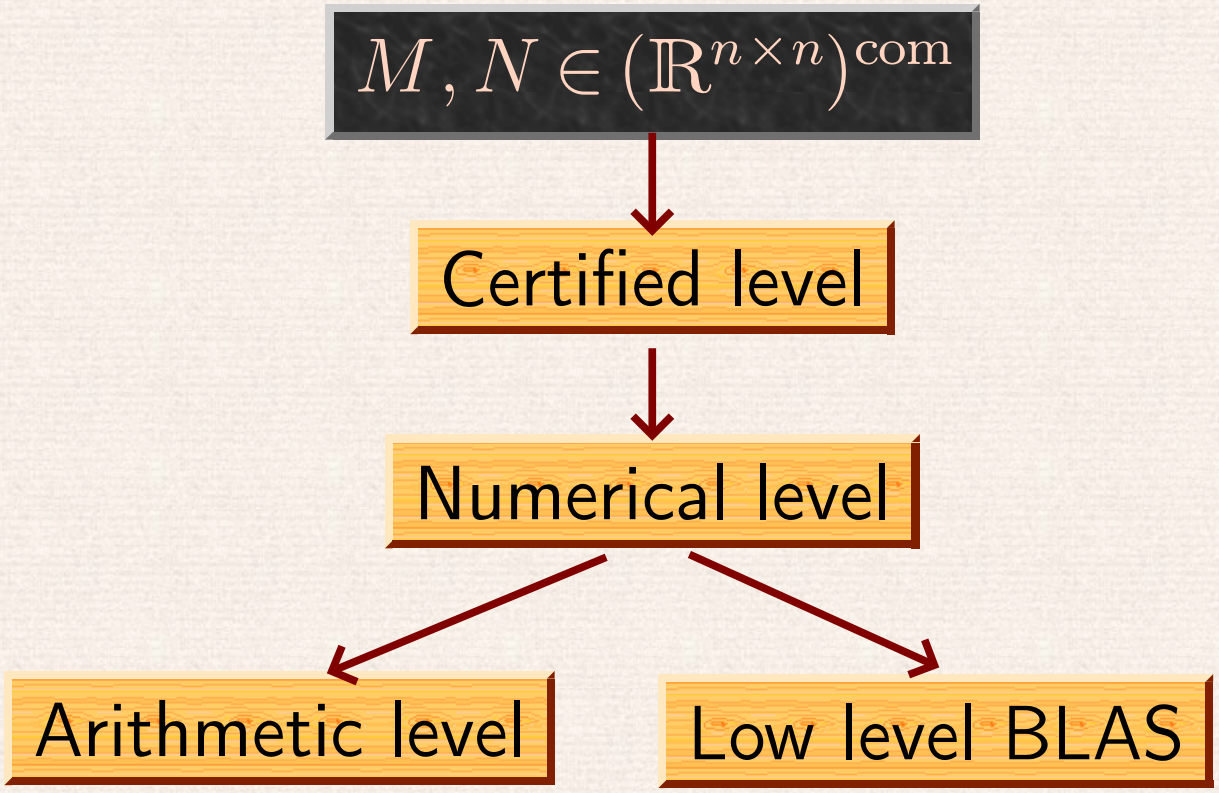
Numerical hierarchy



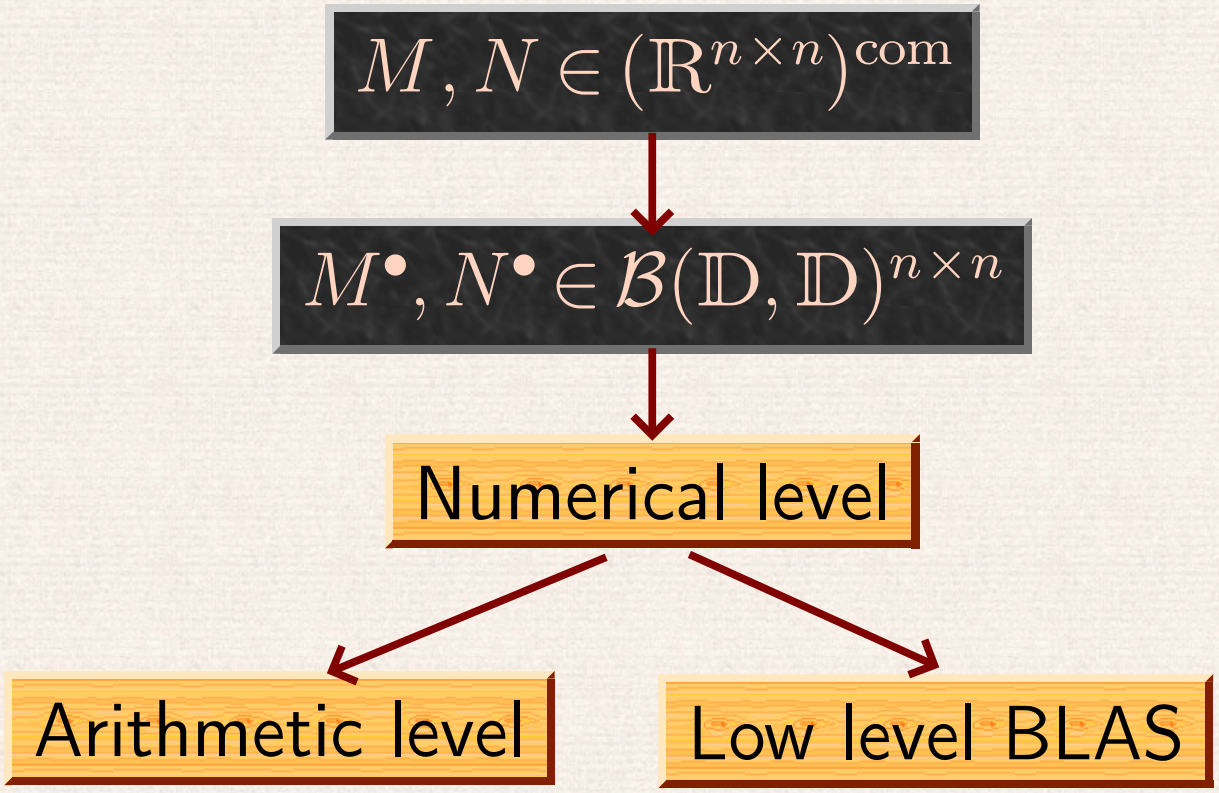
Numerical hierarchy



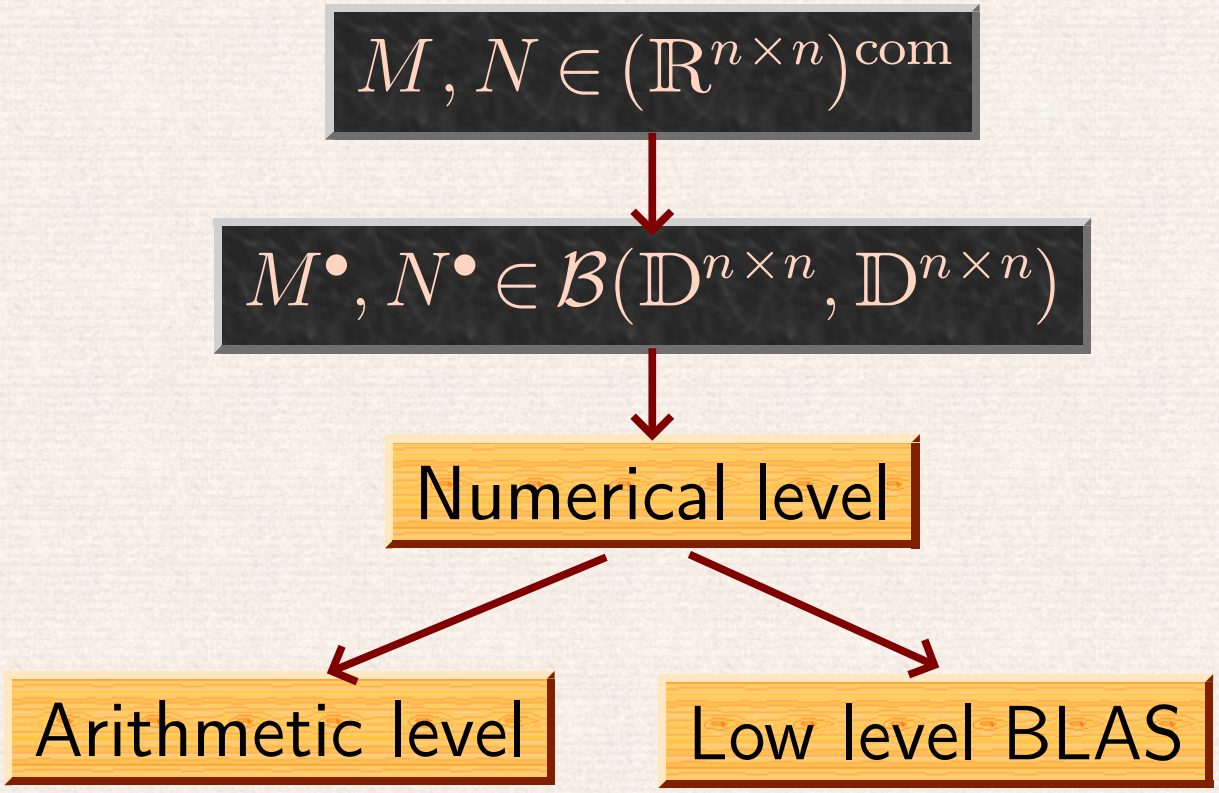
Numerical hierarchy



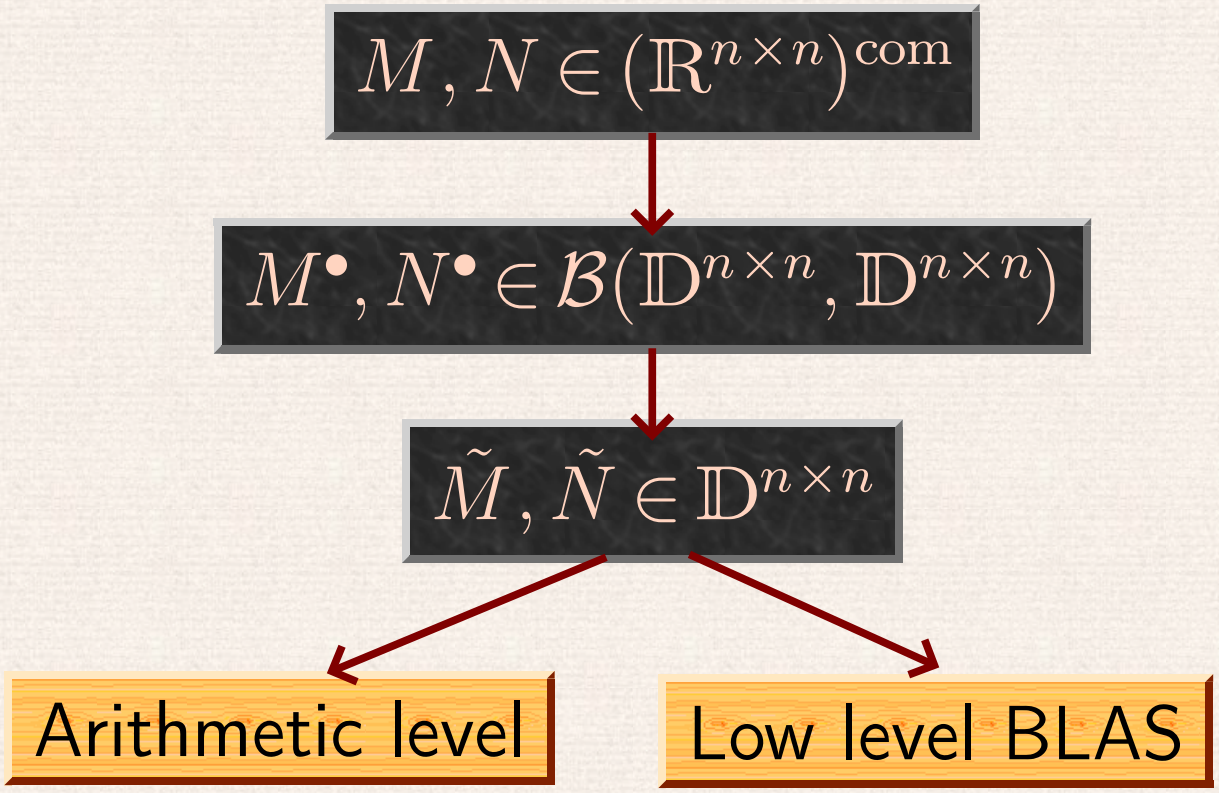
Numerical hierarchy



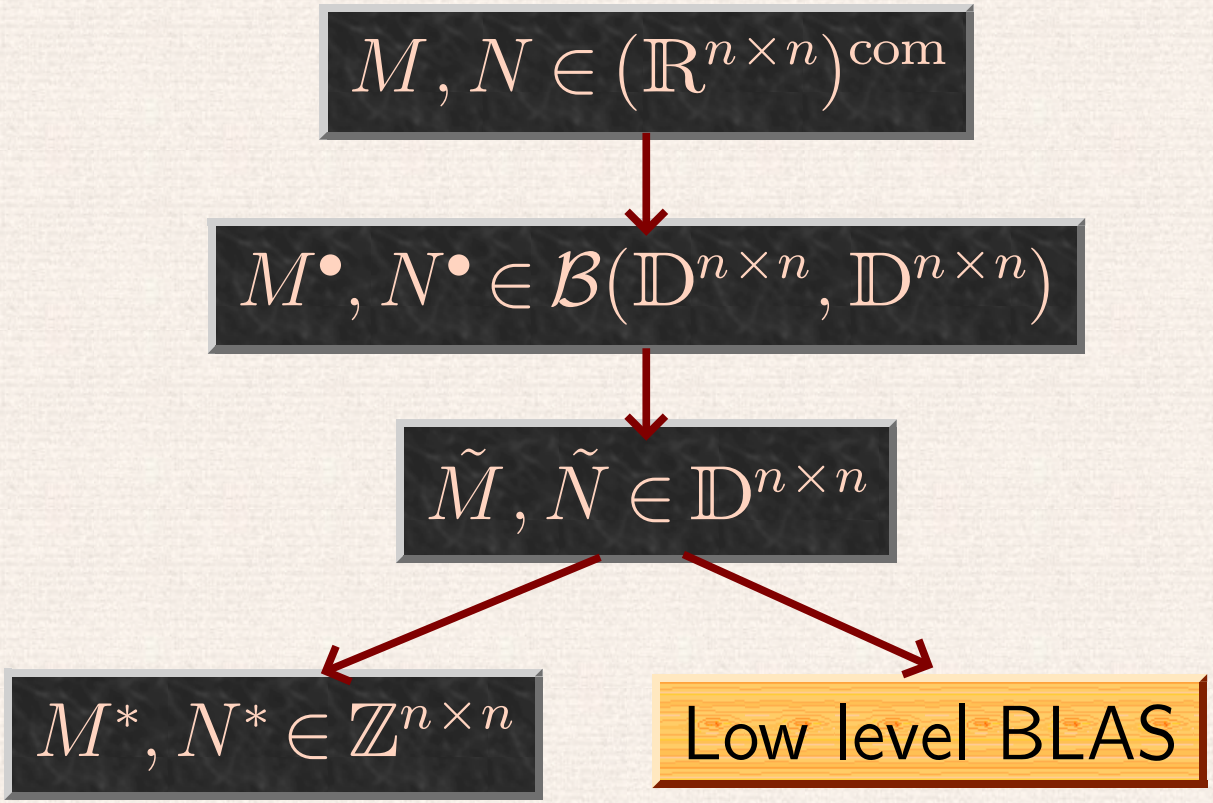
Numerical hierarchy



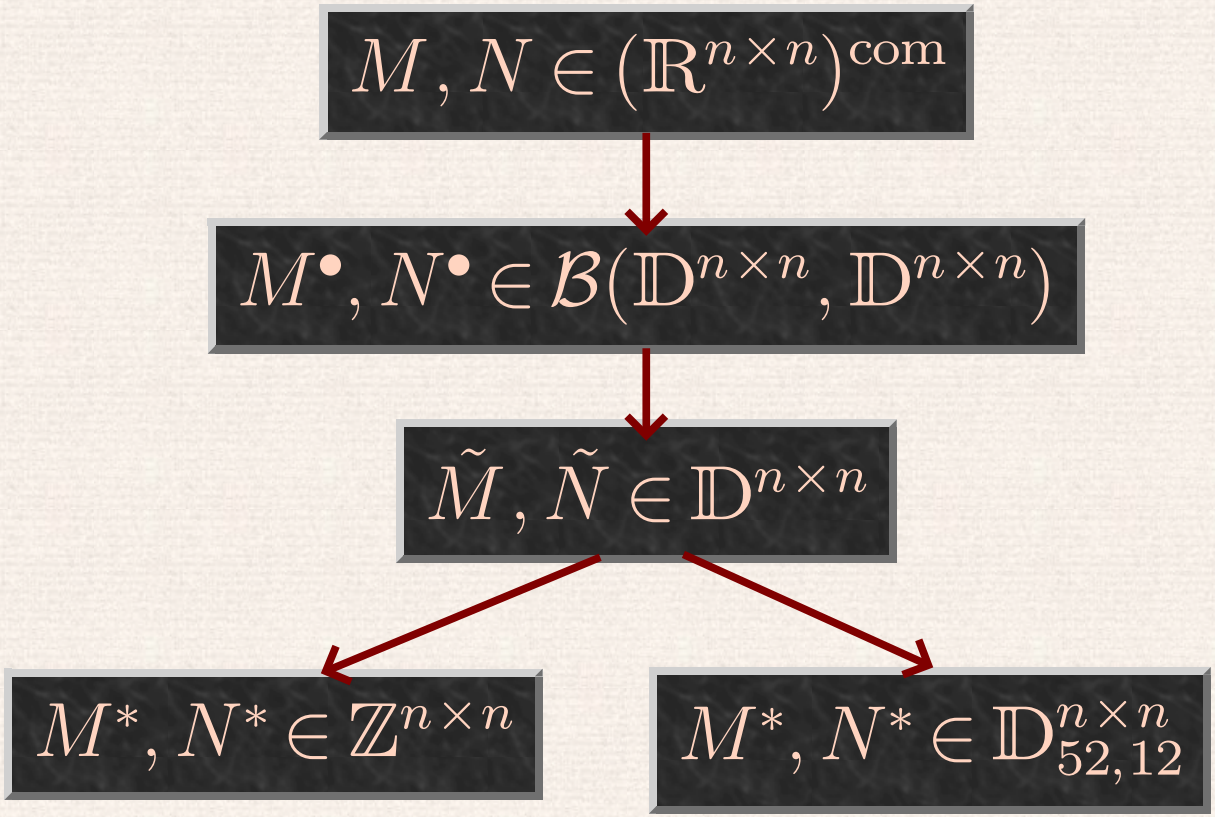
Numerical hierarchy



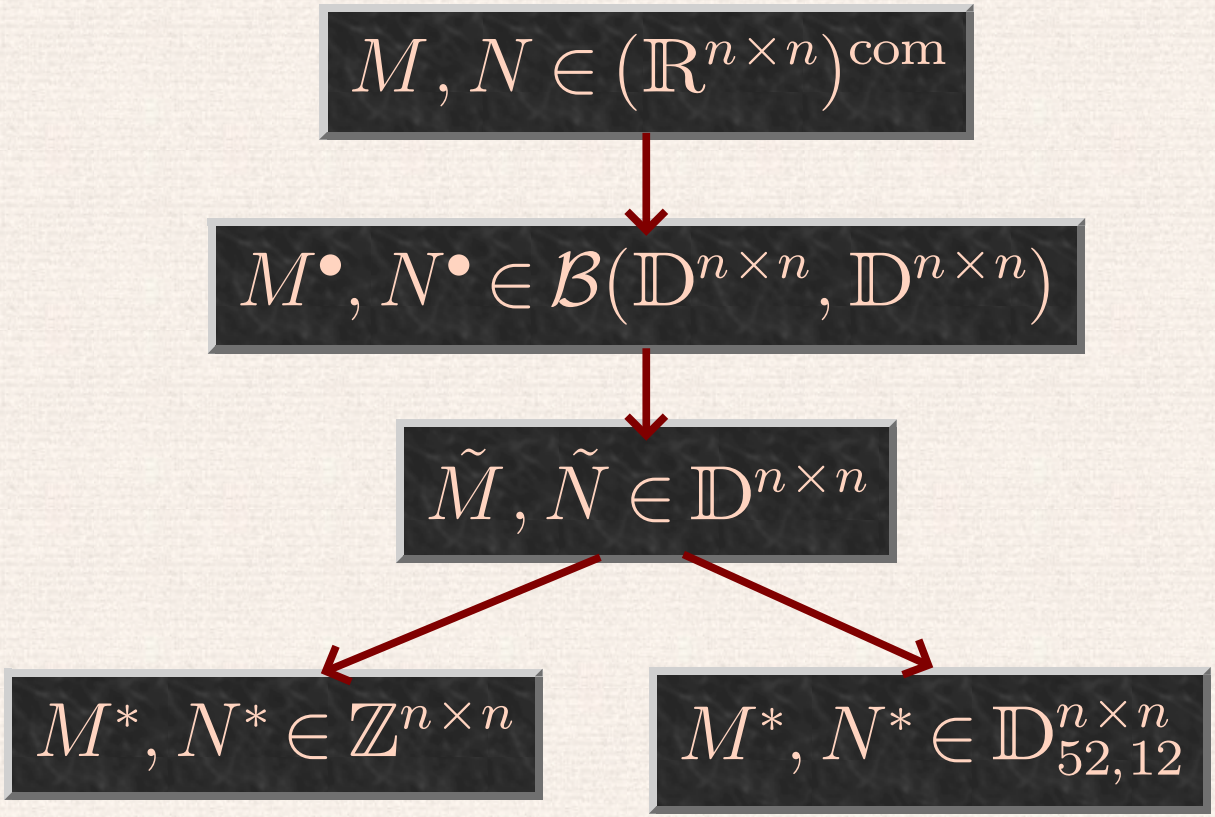
Numerical hierarchy



Numerical hierarchy

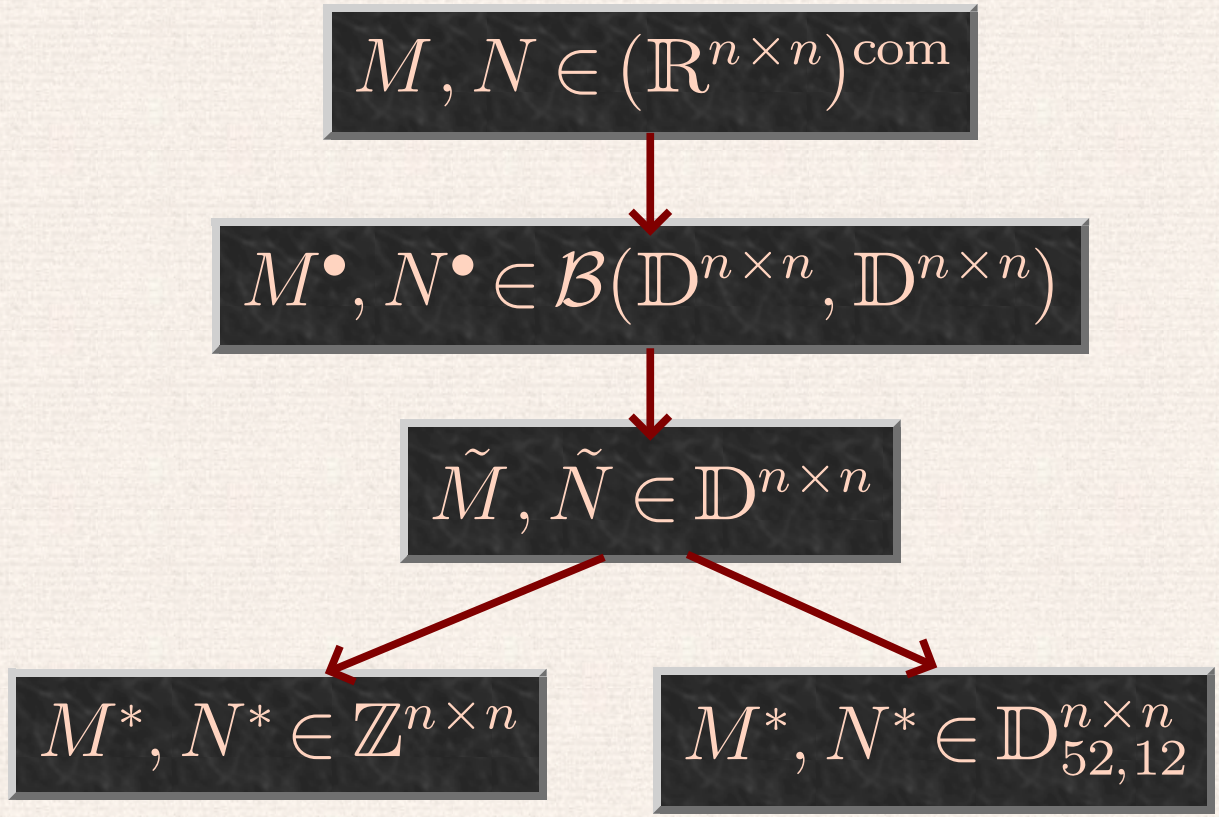


Numerical hierarchy



TTE computable analysis

Numerical hierarchy



TTE computable analysis

BSS computability

Computable real number

Definition. $x \in \mathbb{R}$ *is* ***computable*** *if there exists an algorithm which takes* $\varepsilon \in \mathbb{D}^{>}$ *on input and produces an* ε*-approximation* $\tilde{x} \in \mathbb{D}$ *of* x *with* $|\tilde{x} - x| \leqslant \varepsilon$*. We denote* $\mathbb{R}^{\mathrm{com}}$ *to be the set of computable real numbers.*

```
Mmx] use "asymptotix"; type_mode? := true;
Mmx] one: Approximator (Floating, Floating) == 1;
Mmx] e == exp one
Mmx] approximate (e, 1.0e-1000)
Mmx]
```

Computable real number

Definition. *$x \in \mathbb{R}$ is **computable** if there exists an algorithm which takes $\varepsilon \in \mathbb{D}^{>}$ on input and produces an ε-approximation $\tilde{x} \in \mathbb{D}$ of x with $|\tilde{x} - x| \leqslant \varepsilon$. We denote $\mathbb{R}^{\mathrm{com}}$ to be the set of computable real numbers.*

```
Mmx] use "asymptotix"; type_mode? := true;
Mmx] one: Approximator (Floating, Floating) == 1;
Mmx] e == exp one
```

2.7182818284590452: Approximator(Floating, Floating)

```
Mmx] approximate (e, 1.0e-1000)
Mmx]
```

Computable real number

Definition. $x \in \mathbb{R}$ *is* ***computable*** *if there exists an algorithm which takes* $\varepsilon \in \mathbb{D}^{>}$ *on input and produces an* ε*-approximation* $\tilde{x} \in \mathbb{D}$ *of* x *with* $|\tilde{x} - x| \leqslant \varepsilon$*. We denote* $\mathbb{R}^{\mathrm{com}}$ *to be the set of computable real numbers.*

```
Mmx] use "asymptotix"; type_mode? := true;
```

```
Mmx] one: Approximator (Floating, Floating) == 1;
```

```
Mmx] e == exp one
```

2.7182818284590452: Approximator(Floating, Floating)

```
Mmx] approximate (e, 1.0e-1000)
```

2.71828182845904523536028747135266249775724709369995957496696762772407663035354759457138217852516642742746639193200305992181741359662904357290033429526059563073813232862794349076323382988075319525101901157383418793070215408914993488416750924476146066808226480016847741185374234544243710753907774499206955170276183860626133138458300075204493382656029760673711320070932870912744374704723069697720931014169283681902551510865746377211125238978442505695369677078544996996794686445490598793163688923009879312773617821542499922957635148220826989519366803318252886939849646510582093923982948879332036250944311730123819706841614039701983767932068328237646480429531180232878250981945581530175671736133206981125099618188159304169035159888851934580727386673858942287922849989208680582574927961048419844436346324496848756023362482704197862320900216099023530436994184914631409343173814364054625315209618369088707016768396424378140592714563549061303107208510383750510115747704171898610687396965521267154688957035035402123407848: Floating

```
Mmx]
```

Computable power series

```
Mmx] use "algebramix"; type_mode? := true;
Mmx] z == series (0, 1)
Mmx] B == exp (exp z - 1)
Mmx] B[200] * 200!
Mmx]
```

Computable power series

```
Mmx] use "algebramix"; type_mode? := true;
Mmx] z == series (0, 1)
```

$z + O(z^{10})$: Series(Integer)

```
Mmx] B == exp (exp z - 1)
Mmx] B[200] * 200!
Mmx]
```

Computable power series

```
Mmx] use "algebramix"; type_mode? := true;
```

```
Mmx] z == series (0, 1)
```

$z + O(z^{10})$: Series(Integer)

```
Mmx] B == exp (exp z - 1)
```

$1 + z + z^2 + \frac{5}{6}\,z^3 + \frac{5}{8}\,z^4 + \frac{13}{30}\,z^5 + \frac{203}{720}\,z^6 + \frac{877}{5040}\,z^7 + \frac{23}{224}\,z^8 + \frac{1007}{17280}\,z^9 + O(z^{10})$: Series(Rational)

```
Mmx] B[200] * 200!
```

```
Mmx]
```

Computable power series

```
Mmx] use "algebramix"; type_mode? := true;
Mmx] z == series (0, 1)
```

$z + O(z^{10})$: Series(Integer)

```
Mmx] B == exp (exp z - 1)
```

$1 + z + z^2 + \frac{5}{6}\,z^3 + \frac{5}{8}\,z^4 + \frac{13}{30}\,z^5 + \frac{203}{720}\,z^6 + \frac{877}{5040}\,z^7 + \frac{23}{224}\,z^8 + \frac{1007}{17280}\,z^9 + O(z^{10})$: Series(Rational)

```
Mmx] B[200] * 200!
```

$$624748477619370179475169176526172040868676631541156088557319107233592458655068693596197197036539240046478541880130367251305801992731793116433134463929399319082714196961652168013602803041772182448739785880587721883539643696750170168607678730423675712080858884633763266376601388$$: Rational

```
Mmx]
```

Principle

Systematic approximation of real numbers $x \in \mathbb{R}$ by closed **balls**

$$x^\bullet = \mathcal{B}(c,r) = c + \mathcal{B}(r) \;=\; \{x \in \mathbb{R} : |x-c| \leqslant r\},$$

with $c \in \mathbb{R}$ and $r \in \mathbb{R}^{\geqslant}$ (or $c \in \mathbb{D} \cup \{\pm\infty\}$ and $r \in \mathbb{D}^{\geqslant} \cup \{\infty\}$)

$\longrightarrow$ while guaranteeing that the « genuine result » x lies in $x^\bullet$

Principle

Systematic approximation of real numbers $x \in \mathbb{R}$ by closed **balls**

$$x^\bullet = \mathcal{B}(c,r) = c + \mathcal{B}(r) \;=\; \{x \in \mathbb{R} : |x-c| \leqslant r\},$$

with $c \in \mathbb{R}$ and $r \in \mathbb{R}^{\geqslant}$ (or $c \in \mathbb{D} \cup \{\pm\infty\}$ and $r \in \mathbb{D}^{\geqslant} \cup \{\infty\}$)

$\longrightarrow$ while guaranteeing that the « genuine result » x lies in $x^\bullet$

$\longrightarrow$ analogy with Landau's notation $f(z) = 1 + z + \frac{1}{2} z^2 + \mathcal{O}(z^3)$

Principle

Systematic approximation of real numbers $x \in \mathbb{R}$ by closed **balls**

$$x^\bullet = \mathcal{B}(c,r) = c + \mathcal{B}(r) \;=\; \{x \in \mathbb{R} : |x-c| \leqslant r\},$$

with $c \in \mathbb{R}$ and $r \in \mathbb{R}^{\geqslant}$ (or $c \in \mathbb{D} \cup \{\pm\infty\}$ and $r \in \mathbb{D}^{\geqslant} \cup \{\infty\}$)

$\longrightarrow$ while guaranteeing that the « genuine result » x lies in $x^\bullet$

$\longrightarrow$ analogy with Landau's notation $f(z) = 1 + z + \frac{1}{2}z^2 + o(z^2)$

Principle

Systematic approximation of real numbers $x \in \mathbb{R}$ by closed **balls**

$$x^\bullet = \mathcal{B}(c,r) = c + \mathcal{B}(r) \;=\; \{x \in \mathbb{R} : |x-c| \leqslant r\},$$

with $c \in \mathbb{R}$ and $r \in \mathbb{R}^{\geqslant}$ (or $c \in \mathbb{D} \cup \{\pm\infty\}$ and $r \in \mathbb{D}^{\geqslant} \cup \{\infty\}$)

$\longrightarrow$ while guaranteeing that the « genuine result » x lies in $x^\bullet$

$\longrightarrow$ analogy with Landau's notation $f(z) = 1 + z + \frac{1}{2} z^2 + \mathcal{O}(z^2)$

Operations

$f^\bullet : \mathcal{B}(\mathbb{R}, \mathbb{R}^>)^d \to \mathcal{B}(\mathbb{R}, \mathbb{R}^>)$ **extension** of $f : \mathbb{R}^d \to \mathbb{R}$ if

$$f(x_1^\bullet, \ldots, x_d^\bullet) = \{f(x_1, \ldots, x_d) : x_1 \in x_1^\bullet, \ldots, x_n \in x_n^\bullet\} \;\subseteq\; f^\bullet(x_1^\bullet, \ldots, x_n^\bullet),$$

for all $x_1^\bullet, \ldots, x_d^\bullet \in \mathcal{B}(\mathbb{R}, \mathbb{R}^>)$

Computability revisited

Computability = Ultimate approximability using ball arithmetic

Definition. *$x \in \mathbb{R}$ is **computable** if there exists an algorithm which takes $p \in \mathbb{N}$ on input and which produces a ball $x_p^\bullet \in \mathcal{B}(\mathbb{D}, \mathbb{D})$ with $\{x\} = \bigcap_p x_p^\bullet$.*

Definition. *a function $f\colon \mathbb{R} \to \mathbb{R}$ is **computable** if there exists an algorithm which takes $p \in \mathbb{N}$ on input and which produces a computable function $f_p^\bullet\colon \mathcal{B}(\mathbb{D}, \mathbb{D}) \to \mathcal{B}(\mathbb{D}, \mathbb{D})$ with $f(x^\bullet) \subseteq f_p^\bullet(x^\bullet)$ for all $x^\bullet \in \mathcal{B}(\mathbb{D}, \mathbb{D})$ and such that for each sequence $(x_p^\bullet)_{p\in\mathbb{N}} \in \mathcal{B}(\mathbb{D}, \mathbb{D})^{\mathbb{N}}$ with $\{x\} = \bigcap_p x_p^\bullet$, we have $\{f(x)\} = \bigcap_p f_p^\bullet(x_p^\bullet)$.*

Remark. Equivalent to TTE computability,
but relying on ordinary Turing machines

Exact formulas

$$\begin{aligned}
\mathcal{B}(x,r) +^\bullet \mathcal{B}(y,s) &= \mathcal{B}(x+y, r+s) \\
\mathcal{B}(x,r) -^\bullet \mathcal{B}(y,s) &= \mathcal{B}(x-y, r+s) \\
\mathcal{B}(x,r) \times^\bullet \mathcal{B}(y,s) &= \mathcal{B}(x \times y, r \times (|y|+s) + |x| \times s)
\end{aligned}$$

Elementary operations

Exact formulas

$$\begin{aligned}
\mathcal{B}(x,r) +^\bullet \mathcal{B}(y,s) &= \mathcal{B}(x+y, r+s) \\
\mathcal{B}(x,r) -^\bullet \mathcal{B}(y,s) &= \mathcal{B}(x-y, r+s) \\
\mathcal{B}(x,r) \times^\bullet \mathcal{B}(y,s) &= \mathcal{B}(x \times y, r \times (|y|+s) + |x| \times s)
\end{aligned}$$

Approximate formulas

$x, y, r, s \in \mathbb{D}_{p,e} = \pm\{0, \ldots, 2^p - 1\}\, 2^{\pm\{0,\ldots,2^e-1\}} \cup \{\pm\infty\}$

$$\begin{aligned}
\mathcal{B}(x,r) +^\bullet \mathcal{B}(y,s) &= \mathcal{B}(x +^{\text{near}} y, r +^\uparrow s +^\uparrow \text{error}_+(x,y)) \\
\mathcal{B}(x,r) -^\bullet \mathcal{B}(y,s) &= \mathcal{B}(x -^{\text{near}} y, r +^\uparrow s +^\uparrow \text{error}_-(x,y)) \\
\mathcal{B}(x,r) \times^\bullet \mathcal{B}(y,s) &= \mathcal{B}(x \times^{\text{near}} y, r \times^\uparrow (|y| +^\uparrow s) +^\uparrow |x| \times^\uparrow s +^\uparrow \text{error}_\times(x,y))
\end{aligned}$$

Elementary operations

Exact formulas

$$\begin{aligned}
\mathcal{B}(x,r) +^\bullet \mathcal{B}(y,s) &= \mathcal{B}(x+y, r+s) \\
\mathcal{B}(x,r) -^\bullet \mathcal{B}(y,s) &= \mathcal{B}(x-y, r+s) \\
\mathcal{B}(x,r) \times^\bullet \mathcal{B}(y,s) &= \mathcal{B}(x \times y, r \times (|y|+s) + |x| \times s)
\end{aligned}$$

Approximate formulas

$x, y, r, s \in \mathbb{D}_{p,e} = \pm\{0, \ldots, 2^p - 1\}\, 2^{\pm\{0,\ldots,2^e-1\}} \cup \{\pm\infty\}$

$$\begin{aligned}
\mathcal{B}(x,r) +^\bullet \mathcal{B}(y,s) &= \mathcal{B}(x +^{\text{ieee}} y, r +^\uparrow s +^\uparrow \text{error}(x +^{\text{ieee}} y)) \\
\mathcal{B}(x,r) -^\bullet \mathcal{B}(y,s) &= \mathcal{B}(x -^{\text{ieee}} y, r +^\uparrow s +^\uparrow \text{error}(x -^{\text{ieee}} y)) \\
\mathcal{B}(x,r) \times^\bullet \mathcal{B}(y,s) &= \mathcal{B}(x \times^{\text{ieee}} y, r \times^\uparrow (|y| +^\uparrow s) +^\uparrow |x| \times^\uparrow s +^\uparrow \text{error}(x \times^{\text{ieee}} y))
\end{aligned}$$

Interval arithmetic

Interval arithmetic	Ball arithmetic
Good hardware support	Poor hardware support
Natural standardization	Non canonical standardization
Natural for subdivision algorithms	Not natural for subdivision algorithms
Loses factor 2 for multiple precision	Efficient for multiple precision
Less natural for vectorization	Easy to vectorize
Requires total ordering	Natural extension to $\mathbb{C}$ and beyond
Not natural under perturbations	Natural for perturbation algorithms
ad hoc theoretical foundations (IEEE)	Well understood ε-δ calculus

Interval arithmetic

Interval arithmetic	Ball arithmetic
Good hardware support	Poor hardware support
Natural standardization	Non canonical standardization
Natural for subdivision algorithms	Not natural for subdivision algorithms
Loses factor 2 for multiple precision	Efficient for multiple precision
Less natural for vectorization	Easy to vectorize
Requires total ordering	Natural extension to $\mathbb{C}$ and beyond
Not natural under perturbations	Natural for perturbation algorithms
ad hoc theoretical foundations (IEEE)	Well understood ε-δ calculus

References

Moore (1965): Interval analysis
Alefeld & Herzberger (1983): Introduction to interval analysis
Rump (2010): Verification methods: rigourous results using FP arithmetic

Digression: *a priori* error estimates

Example: computing an ε-approximation for $z = x + y$

Compute $\frac{\varepsilon}{2}$-approximations $\tilde{x}$ and $\tilde{y}$ for x and y
Return $\tilde{x} + \tilde{y}$

Digression: *a priori* error estimates

Example: computing an ε-approximation for $z = x + y$

Compute $\frac{\varepsilon}{2}$-approximations $\tilde{x}$ and $\tilde{y}$ for x and y
Return $\tilde{x} + \tilde{y}$

Advantage: adaptive precision in $1_{\text{cheap}} + 10^{-100}_{\text{expensive}}$

Digression: *a priori* error estimates

Example: computing an ε-approximation for $z = x + y$

Compute $\frac{\varepsilon}{2}$-approximations $\tilde{x}$ and $\tilde{y}$ for x and y
Return $\tilde{x} + \tilde{y}$

Advantage: adaptive precision in $1_{\text{cheap}} + 10^{-100}_{\text{expensive}}$

Problem 1: Horner evaluation of $P_d\, x^d + \cdots + P_0$ at $x = 1$

$\longrightarrow$ computes $\frac{\varepsilon}{2^{d-k}}$-approximation of P_k for $k = 0, \ldots, d-1$

$\longrightarrow$ small tolerances and bad complexity

Digression: *a priori* error estimates

Example: computing an ε-approximation for $z = x + y$

Compute $\frac{\varepsilon}{2}$-approximations $\tilde{x}$ and $\tilde{y}$ for x and y
Return $\tilde{x} + \tilde{y}$

Advantage: adaptive precision in $1_{\text{cheap}} + 10^{-100}_{\text{expensive}}$

Problem 1: Horner evaluation of $P_d\, x^d + \cdots + P_0$ at $x = 1$

$\longrightarrow$ computes $\frac{\varepsilon}{2^{d-k}}$-approximation of P_k for $k = 0, \ldots, d-1$

Tolerance balancing: compute $\frac{\varepsilon}{d+1}$-approximation of P_k for each k

Digression: *a priori* error estimates

Example: computing an ε-approximation for $z = x + y$

Compute $\frac{\varepsilon}{2}$-approximations $\tilde{x}$ and $\tilde{y}$ for x and y
Return $\tilde{x} + \tilde{y}$

Advantage: adaptive precision in $1_{\text{cheap}} + 10^{-100}_{\text{expensive}}$

Problem 1: Horner evaluation of $P_d\, x^d + \cdots + P_0$ at $x = 1$

$\longrightarrow$ computes $\frac{\varepsilon}{2^{d-k}}$-approximation of P_k for $k = 0, \ldots, d-1$

Tolerance balancing: compute $\frac{\varepsilon}{d+1}$-approximation of P_k for each k

Problem 2: necessitates « dags » for the intermediate results

The enemy

Overestimation of the error

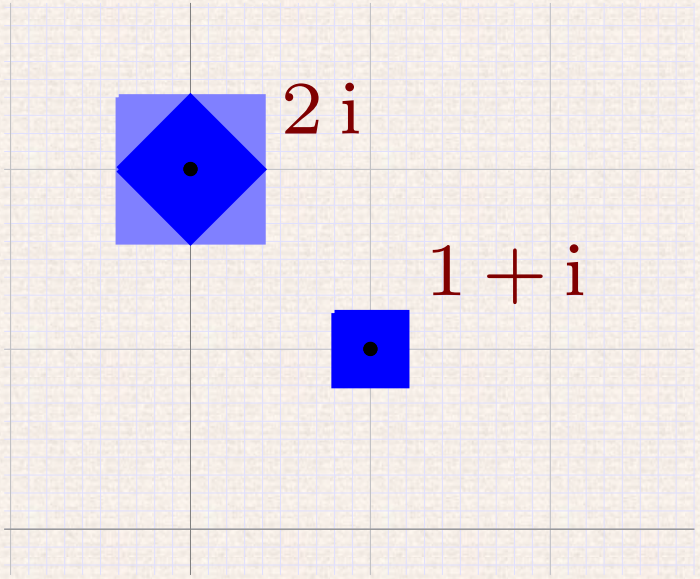
$$\mathcal{B}(0,1) - \mathcal{B}(0,1) \;=\; \mathcal{B}(0,2)$$

Overestimation of the error

$$\mathcal{B}(0,1) - \mathcal{B}(0,1) \;=\; \mathcal{B}(0,2)$$

Wrapping effect

$$([1-\varepsilon, 1+\varepsilon] + [1-\varepsilon, 1+\varepsilon]\,\mathrm{i})^2 \;=\; [-2\,\varepsilon, 2\,\varepsilon] + [2-2\,\varepsilon, 2+2\,\varepsilon]\,\mathrm{i}$$



Example

```
Mmx] use "algebramix";
Mmx] power (z, n) == if n = 1 then z else z * power (z, n-1);
Mmx] z == complex (interval (1.0, 1.0000000001),
                   interval (1.0, 1.0000000001))
Mmx] [ power (z, 4*n) || n in 1 to 10 ]
Mmx]
```

Example

```
Mmx] use "algebramix";
```

```
Mmx] power (z, n) == if n = 1 then z else z * power (z, n-1);
```

```
Mmx] z == complex (interval (1.0, 1.0000000001),
                   interval (1.0, 1.0000000001))
```

$1.000000000 + 1.000000000\,\mathrm{i}$

```
Mmx] [ power (z, 4*n) || n in 1 to 10 ]
```

```
Mmx]
```

Example

```
Mmx] use "algebramix";
```

```
Mmx] power (z, n) == if n = 1 then z else z * power (z, n-1);
```

```
Mmx] z == complex (interval (1.0, 1.0000000001),
                   interval (1.0, 1.0000000001))
```

$1.000000000 + 1.000000000\,\mathrm{i}$

```
Mmx] [ power (z, 4*n) || n in 1 to 10 ]
```

$$\begin{bmatrix} -4.00000000 \\ 16.0000000 \\ -64.000000 \\ 256.0000 \\ -1024.000 \\ 4096.00 \\ -16384.0 \\ 6.5536e4 \\ -2.62e5 \\ 1.05e6 \end{bmatrix}$$

```
Mmx]
```

Changing the kind of enclosures

```
Mmx] use "algebramix";
Mmx] power (z, n) == if n = 1 then z else z * power (z, n-1);
Mmx] z == ball (complex (1.0, 1.0), 0.0000000001)
Mmx] [ power (z, 4*n) || n in 1 to 10 ]
Mmx]
```

Changing the kind of enclosures

```
Mmx] use "algebramix";
Mmx] power (z, n) == if n = 1 then z else z * power (z, n-1);
Mmx] z == ball (complex (1.0, 1.0), 0.0000000001)
```

$1.000000000 + 1.000000000\,\mathrm{i}$

```
Mmx] [ power (z, 4*n) || n in 1 to 10 ]
Mmx]
```

Changing the kind of enclosures

```
Mmx] use "algebramix";
Mmx] power (z, n) == if n = 1 then z else z * power (z, n-1);
Mmx] z == ball (complex (1.0, 1.0), 0.0000000001)
```

$$1.000000000 + 1.000000000\,\mathrm{i}$$

```
Mmx] [ power (z, 4*n) || n in 1 to 10 ]
```

$$\begin{bmatrix} -4.00000000 \\ 16.0000000 \\ -64.000000 \\ 256.000000 \\ -1024.00000 \\ 4096.0000 \\ -16384.0000 \\ 65536.000 \\ -262144.00 \\ 1.04857600e6 \end{bmatrix}$$

```
Mmx]
```

Minimizing the depth of the computation

```
Mmx] use "algebramix";
```

```
Mmx] power (z, n) ==
       if n = 1 then z
       else power (z, n quo 2) * power (z, n - (n quo 2));
```

```
Mmx] z == complex (interval (1.0, 1.0000000001),
                   interval (1.0, 1.0000000001))
```

```
Mmx] [ power (z, 4*n) || n in 1 to 10 ]
```

```
Mmx]
```

Minimizing the depth of the computation

```
Mmx] use "algebramix";
```

```
Mmx] power (z, n) ==
       if n = 1 then z
       else power (z, n quo 2) * power (z, n - (n quo 2));
```

```
Mmx] z == complex (interval (1.0, 1.0000000001),
                   interval (1.0, 1.0000000001))
```

$1.000000000 + 1.000000000\,\mathrm{i}$

```
Mmx] [ power (z, 4*n) || n in 1 to 10 ]
```

```
Mmx]
```

Minimizing the depth of the computation

```
Mmx] use "algebramix";
```

```
Mmx] power (z, n) ==
       if n = 1 then z
       else power (z, n quo 2) * power (z, n - (n quo 2));
```

```
Mmx] z == complex (interval (1.0, 1.0000000001),
                   interval (1.0, 1.0000000001))
```

$1.000000000 + 1.000000000\,\mathrm{i}$

```
Mmx] [ power (z, 4*n) || n in 1 to 10 ]
```

$$\begin{bmatrix} -4.00000000 \\ 16.0000000 \\ -64.000000 \\ 256.000000 \\ -1024.00000 \\ 4096.0000 \\ -16384.000 \\ 65536.000 \\ -262144.00 \\ 1.0485760e6 \end{bmatrix}$$

```
Mmx]
```

Perturbation method (Hansen)

Algorithm: invert the matrix $M \in \mathcal{B}(\mathbb{D}, \mathbb{D})^{n \times n}$

- Write $M = \mathcal{B}(C, R)$, with $C \in \mathbb{D}^{n \times n}$ et $R \in (\mathbb{D}^{>})^{n \times n}$
- Numerically invert $U :\approx C^{-1}$
- Compute $E := C\,\mathcal{B}(U, 0) - \mathcal{B}(1, 0)$, so that

$$M^{-1} \;=\; U\,(MU)^{-1} = U\,(1+E)^{-1}$$

- With $\varepsilon := \|E\|_\infty = \max_{1 \leqslant i \leqslant n} (|E_{i1}| + \cdots + |E_{in}|)$, we have

$$\|(1+E)^{-1} - 1\|_\infty = \|-E + E^2 - E^3 + \cdots\|_\infty \leqslant \frac{\varepsilon}{1-\varepsilon}$$

- Let $\mathbb{1}$ be the matrix with entries $\mathbb{1}_{ij} = 1$
- Return $V := U\,\mathcal{B}(1, \frac{\varepsilon}{1-\varepsilon}\,\mathbb{1})$

Perturbation method (Hansen)

Algorithm: invert the matrix $M \in \mathcal{B}(\mathbb{D},\mathbb{D})^{n\times n}$

- Write $M = \mathcal{B}(C,R)$, with $C \in \mathbb{D}^{n\times n}$ et $R \in (\mathbb{D}^{>})^{n\times n}$
- Numerically invert $U :\approx C^{-1}$
- Compute $E := C\,\mathcal{B}(U,0) - \mathcal{B}(1,0)$, so that

$$M^{-1} \;=\; U\,(MU)^{-1} = U\,(1+E)^{-1}$$

- With $\varepsilon := \|E\|_\infty = \max_{1\leqslant i\leqslant n}(|E_{i1}| + \cdots + |E_{in}|)$, we have

$$\|(1+E)^{-1} - 1\|_\infty = \|-E + E^2 - E^3 + \cdots\|_\infty \leqslant \frac{\varepsilon}{1-\varepsilon}$$

- Let $\mathbb{1}$ be the matrix with entries $\mathbb{1}_{ij} = 1$
- Return $V := U\,\mathcal{B}(1, \frac{\varepsilon}{1-\varepsilon}\,\mathbb{1})$

More generally $\rightsquigarrow$

Phase of **guessing** or **prospection**, next phase of **validation**

Example

```
Mmx] use "analyziz"
```

```
Mmx] rnd () == {
       x == uniform_deviate (0.0, 1.0);
       return ball (x, 0.00000001);
     };
```

```
Mmx] M == [ rnd () | j in 1 to 7 || i in 1 to 7 ]
```

```
Mmx] row_echelon M
```

```
Mmx]
```

Example

```
Mmx] use "analyziz"
```

```
Mmx] rnd () == {
       x == uniform_deviate (0.0, 1.0);
       return ball (x, 0.00000001);
     };
```

```
Mmx] M == [ rnd () | j in 1 to 7 || i in 1 to 7 ]
```

$$\begin{bmatrix} 0.6003643 & 0.3710189 & 0.4791752 & 0.1559513 & 0.4159951 & 0.1407053 & 0.0044911 \\ 0.5120348 & 0.7247114 & 0.1947940 & 0.655922 & 0.9355430 & 0.251486 & 0.2468534 \\ 0.462639 & 0.5648943 & 0.744350 & 0.4275757 & 0.7804102 & 0.9920186 & 0.7571225 \\ 0.2508314 & 0.0131642 & 0.8669620 & 0.6802075 & 0.7552889 & 0.3166660 & 0.3160914 \\ 0.2876469 & 0.988199 & 0.6639689 & 0.303261 & 0.498002 & 0.0277941 & 0.9222224 \\ 0.1304958 & 0.7945297 & 0.7988684 & 0.5890020 & 0.3421886 & 0.0285099 & 0.9893513 \\ 0.3882938 & 0.7500550 & 0.9237840 & 0.594590 & 0.8730709 & 0.5362681 & 0.134547 \end{bmatrix}$$

```
Mmx] row_echelon M
```

```
Mmx]
```

Example

```
Mmx] use "analyziz"
```

```
Mmx] rnd () == {
       x == uniform_deviate (0.0, 1.0);
       return ball (x, 0.00000001);
     };
```

```
Mmx] M == [ rnd () | j in 1 to 7 || i in 1 to 7 ]
```

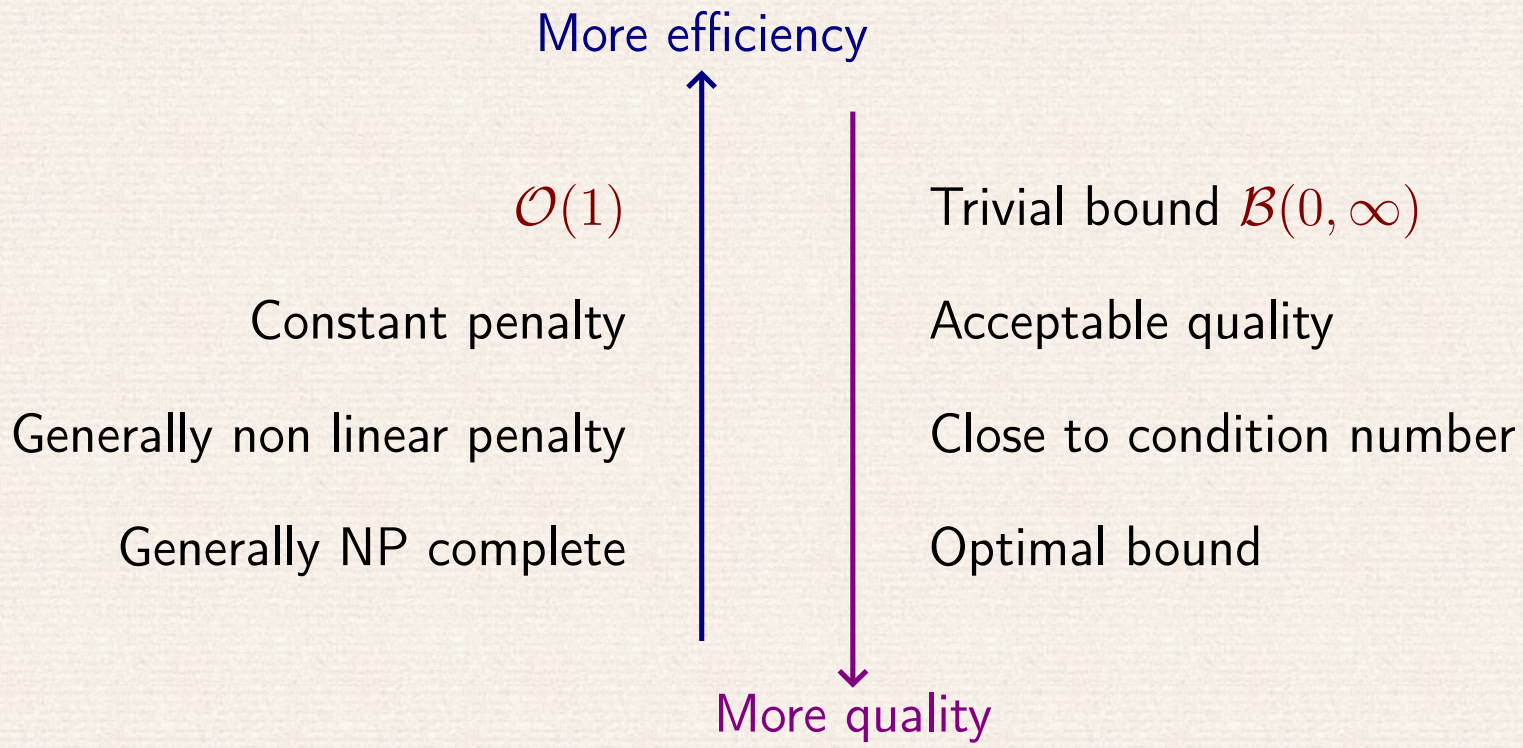
$$\begin{bmatrix} 0.6003643 & 0.3710189 & 0.4791752 & 0.1559513 & 0.4159951 & 0.1407053 & 0.0044911 \\ 0.5120348 & 0.7247114 & 0.1947940 & 0.655922 & 0.9355430 & 0.251486 & 0.2468534 \\ 0.462639 & 0.5648943 & 0.744350 & 0.4275757 & 0.7804102 & 0.9920186 & 0.7571225 \\ 0.2508314 & 0.0131642 & 0.8669620 & 0.6802075 & 0.7552889 & 0.3166660 & 0.3160914 \\ 0.2876469 & 0.988199 & 0.6639689 & 0.303261 & 0.498002 & 0.0277941 & 0.9222224 \\ 0.1304958 & 0.7945297 & 0.7988684 & 0.5890020 & 0.3421886 & 0.0285099 & 0.9893513 \\ 0.3882938 & 0.7500550 & 0.9237840 & 0.594590 & 0.8730709 & 0.5362681 & 0.134547 \end{bmatrix}$$

```
Mmx] row_echelon M
```

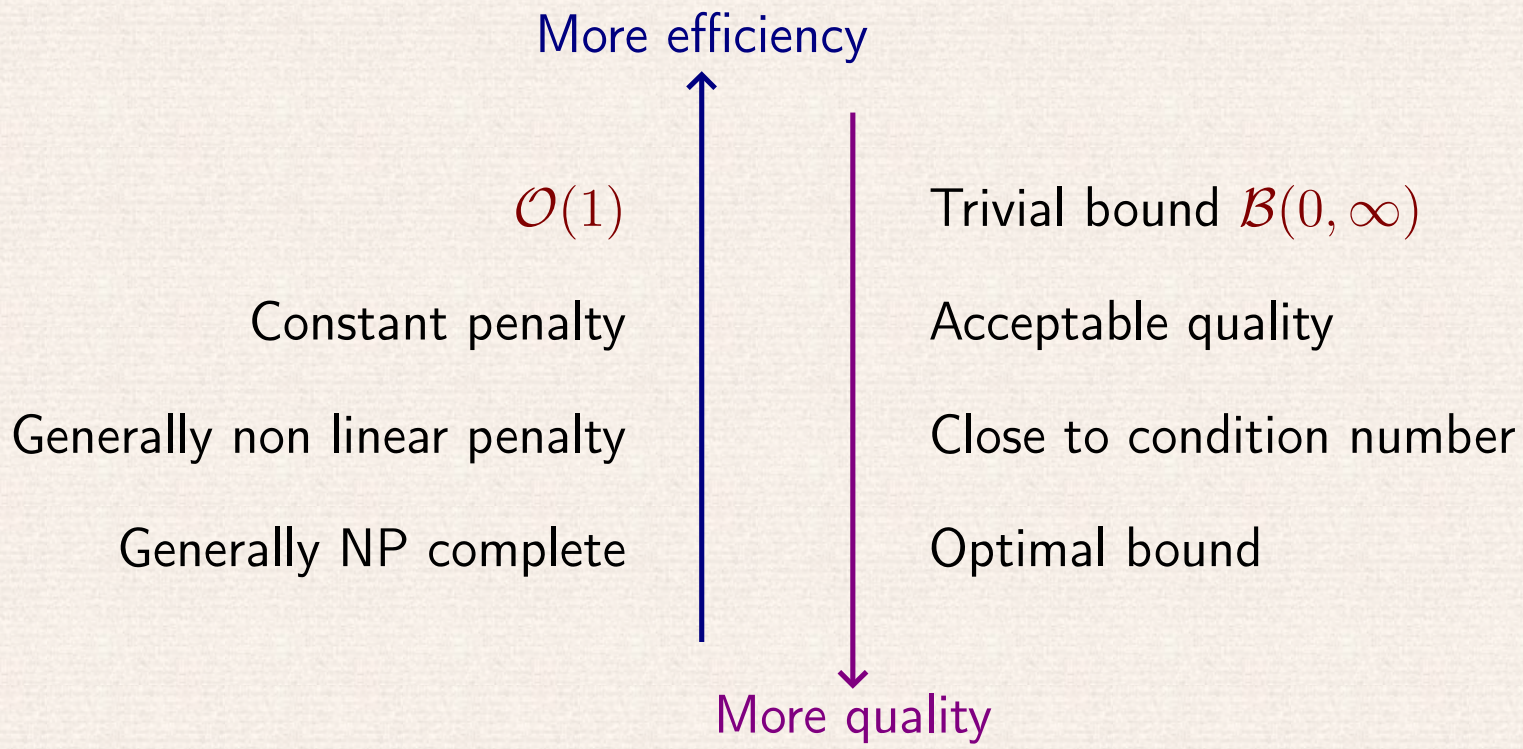
$$\begin{bmatrix} 0.6003643 & 0.3710189 & 0.4791752 & 0.1559513 & 0.4159951 & 0.1407053 & 0.0044911 \\ 0e-7 & 0.810436 & 0.4343862 & 0.2285411 & 0.298690 & -0.0396207 & 0.9200706 \\ 0e-7 & 0e-7 & 0.7427919 & 0.655052 & 0.633765 & 0.2509449 & 0.475251 \\ 0e-7 & 0e-7 & 0e-7 & 0.7893843 & 0.799480 & 0.2976308 & 0.0563714 \\ 0e-7 & 0e-7 & 0e-7 & 0e-7 & -0.3571903 & -0.102232 & -0.027367 \\ 0e-7 & 0e-7 & 0e-7 & 0e-7 & 0e-7 & 0.771327 & 0.280188 \\ 0e-7 & 0e-7 & 0e-7 & 0e-7 & 0e-7 & 0e-7 & -0.7889997 \end{bmatrix}$$

```
Mmx]
```

Graal: efficient *and* tight error bounding



Graal: efficient *and* tight error bounding



Early termination: start with fast strategy, but *a priori* of poor quality
Use slower algorithms only if this quality is not enough

Example: matrix multiplication

Majoration $|(MN)_{ij}| \leqslant [\max |M_{i\cdot}|]\,[\max |N_{\cdot j}|]$:

1 multiplication in $\mathbb{D}^{n\times n}$ and $\mathcal{O}(n^2)$ majorations, possibly bad quality

Compromise: cut M and N into $k \times k$ matrices for fixed k:

α_k multiplications in $\mathbb{D}^{n\times n}$ with $1 \leqslant \alpha_k \leqslant 3$, usually good quality

Compromise: resort to naive multiplication in case of ill conditioning

Between 1 and 3 multiplications in $\mathbb{D}^{n\times n}$, always good quality

Rewrite $M, N \in \mathcal{B}(\mathbb{D}^{n\times n}, \mathbb{D}^{n\times n})$:

3 multiplications in $\mathbb{D}^{n\times n}$ (with 2 multiplications at single precision)

Naive product M, N with $M, N \in \mathcal{B}(\mathbb{D}, \mathbb{D})^{n\times n}$:

$\mathcal{O}(n^3)$ operations in $\mathcal{B}(\mathbb{D}, \mathbb{D})$, always good quality

Example: matrix multiplication

Majoration $|(MN)_{ij}| \leqslant [\max |M_{i\cdot}|]\,[\max |N_{\cdot j}|]$:

1 multiplication in $\mathbb{D}^{n\times n}$ and $\mathcal{O}(n^2)$ majorations, possibly bad quality

Compromise: cut M and N into $k \times k$ matrices for fixed k:

α_k multiplications in $\mathbb{D}^{n\times n}$ with $1 \leqslant \alpha_k \leqslant 3$, usually good quality

Compromise: resort to naive multiplication in case of ill conditioning

Between 1 and 3 multiplications in $\mathbb{D}^{n\times n}$, always good quality

Rewrite $M, N \in \mathcal{B}(\mathbb{D}^{n\times n}, \mathbb{D}^{n\times n})$:

3 multiplications in $\mathbb{D}^{n\times n}$ (with 2 multiplications at single precision)

Naive product M, N with $M, N \in \mathcal{B}(\mathbb{D}, \mathbb{D})^{n\times n}$:

$\mathcal{O}(n^3)$ operations in $\mathcal{B}(\mathbb{D}, \mathbb{D})$, always good quality

Example: matrix multiplication

Majoration $|(MN)_{ij}| \leqslant [\max |M_{i\cdot}|]\,[\max |N_{\cdot j}|]$:

1 multiplication in $\mathbb{D}^{n\times n}$ and $\mathcal{O}(n^2)$ majorations, possibly bad quality

Compromise: cut M and N into $k\times k$ matrices for fixed k:

α_k multiplications in $\mathbb{D}^{n\times n}$ with $1 \leqslant \alpha_k \leqslant 3$, usually good quality

Compromise: resort to naive multiplication in case of ill conditioning

Between 1 and 3 multiplications in $\mathbb{D}^{n\times n}$, always good quality

Rewrite $M, N \in \mathcal{B}(\mathbb{D}^{n\times n}, \mathbb{D}^{n\times n})$:

3 multiplications in $\mathbb{D}^{n\times n}$ (with 2 multiplications at single precision)

Naive product M, N with $M, N \in \mathcal{B}(\mathbb{D}, \mathbb{D})^{n\times n}$:

$\mathcal{O}(n^3)$ operations in $\mathcal{B}(\mathbb{D}, \mathbb{D})$, always good quality

Example: matrix multiplication

Majoration $|(MN)_{ij}| \leqslant [\max|M_{i\cdot}|]\,[\max|N_{\cdot j}|]$:

1 multiplication in $\mathbb{D}^{n\times n}$ and $\mathcal{O}(n^2)$ majorations, possibly bad quality

Compromise: cut M and N into $k\times k$ matrices for fixed k:

α_k multiplications in $\mathbb{D}^{n\times n}$ with $1 \leqslant \alpha_k \leqslant 3$, usually good quality

Compromise: resort to naive multiplication in case of ill conditioning

Between 1 and 3 multiplications in $\mathbb{D}^{n\times n}$, always good quality

Rewrite $M, N \in \mathcal{B}(\mathbb{D}^{n\times n}, \mathbb{D}^{n\times n})$:

3 multiplications in $\mathbb{D}^{n\times n}$ (with 2 multiplications at single precision)

Naive product M, N with $M, N \in \mathcal{B}(\mathbb{D}, \mathbb{D})^{n\times n}$:

$\mathcal{O}(n^3)$ operations in $\mathcal{B}(\mathbb{D}, \mathbb{D})$, always good quality

Example: matrix multiplication

Majoration $|(MN)_{ij}| \leqslant [\max|M_{i\cdot}|]\,[\max|N_{\cdot j}|]$:

1 multiplication in $\mathbb{D}^{n\times n}$ and $\mathcal{O}(n^2)$ majorations, possibly bad quality

Compromise: cut M and N into $k\times k$ matrices for fixed k:

α_k multiplications in $\mathbb{D}^{n\times n}$ with $1\leqslant\alpha_k\leqslant 3$, usually good quality

Compromise: resort to naive multiplication in case of ill conditioning

Between 1 and 3 multiplications in $\mathbb{D}^{n\times n}$, always good quality

Rewrite $M, N \in \mathcal{B}(\mathbb{D}^{n\times n}, \mathbb{D}^{n\times n})$:

3 multiplications in $\mathbb{D}^{n\times n}$ (with 2 multiplications at single precision)

Naive product M, N with $M, N \in \mathcal{B}(\mathbb{D}, \mathbb{D})^{n\times n}$:

$\mathcal{O}(n^3)$ operations in $\mathcal{B}(\mathbb{D}, \mathbb{D})$, always good quality

Example: matrix multiplication

Majoration $|(MN)_{ij}| \leqslant [\max |M_{i\cdot}|]\,[\max |N_{\cdot j}|]$:

1 multiplication in $\mathbb{D}^{n\times n}$ and $\mathcal{O}(n^2)$ majorations, possibly bad quality

Compromise: cut M and N into $k \times k$ matrices for fixed k:

α_k multiplications in $\mathbb{D}^{n\times n}$ with $1 \leqslant \alpha_k \leqslant 3$, usually good quality

Compromise: resort to naive multiplication in case of ill conditioning

Between 1 and 3 multiplications in $\mathbb{D}^{n\times n}$, always good quality

Rewrite $M, N \in \mathcal{B}(\mathbb{D}^{n\times n}, \mathbb{D}^{n\times n})$:

3 multiplications in $\mathbb{D}^{n\times n}$ (with 2 multiplications at single precision)

Naive product M, N with $M, N \in \mathcal{B}(\mathbb{D}, \mathbb{D})^{n\times n}$:

$\mathcal{O}(n^3)$ operations in $\mathcal{B}(\mathbb{D}, \mathbb{D})$, always good quality

General principle (observed in practice):

Validation cost asymptotically negligible for well conditioned operations

Example: polynomial multiplication

$$P, Q \in \mathcal{B}(\mathbb{D}_p, \mathbb{D}_{52})[x]_{<d}$$

FFT multiplication
Cost $\tilde{\mathcal{O}}(d\,p)$, only good quality if all coefficients of same magnitude

FFT multiplication with scaling $[P(\alpha\,x)\,Q(\alpha\,x)] \circ (x/\alpha)$
Cost $\tilde{\mathcal{O}}(d\,p)$, good quality for truncated power series

FFT multiplication with scaling and cutting into blocks (VdH, Arb)
(vdH) Cost $\tilde{\mathcal{O}}(d\,p)$, good quality in terms of roots of polynomials
(Arb) Cost between $\tilde{\mathcal{O}}(d\,p)$ and $\tilde{\mathcal{O}}(d^2\,p)$, optimal quality

Naive multiplication of $P, Q \in \mathcal{B}(\mathbb{D}_p, \mathbb{D}_{52})[x]_{<d}$
Cost $\tilde{\mathcal{O}}(d^2\,p)$, optimal quality

Example: polynomial multiplication

$$P, Q \in \mathcal{B}(\mathbb{D}_p, \mathbb{D}_{52})[x]_{<d}$$

FFT multiplication

Cost $\tilde{\mathcal{O}}(d\,p)$, only good quality if all coefficients of same magnitude

FFT multiplication with scaling $[P(\alpha\,x)\,Q(\alpha\,x)] \circ (x/\alpha)$

Cost $\tilde{\mathcal{O}}(d\,p)$, good quality for truncated power series

FFT multiplication with scaling and cutting into blocks (VdH, Arb)

(vdH) Cost $\tilde{\mathcal{O}}(d\,p)$, good quality in terms of roots of polynomials

(Arb) Cost between $\tilde{\mathcal{O}}(d\,p)$ and $\tilde{\mathcal{O}}(d^2\,p)$, optimal quality

Naive multiplication of $P, Q \in \mathcal{B}(\mathbb{D}_p, \mathbb{D}_{52})[x]_{<d}$

Cost $\tilde{\mathcal{O}}(d^2\,p)$, optimal quality

Example: polynomial multiplication

$$P, Q \in \mathcal{B}(\mathbb{D}_p, \mathbb{D}_{52})[x]_{<d}$$

FFT multiplication

Cost $\tilde{\mathcal{O}}(d\,p)$, only good quality if all coefficients of same magnitude

FFT multiplication with scaling $[P(\alpha\,x)\,Q(\alpha\,x)] \circ (x/\alpha)$

Cost $\tilde{\mathcal{O}}(d\,p)$, good quality for truncated power series

FFT multiplication with scaling and cutting into blocks (VdH, Arb)

(vdH) Cost $\tilde{\mathcal{O}}(d\,p)$, good quality in terms of roots of polynomials

(Arb) Cost between $\tilde{\mathcal{O}}(d\,p)$ and $\tilde{\mathcal{O}}(d^2\,p)$, optimal quality

Naive multiplication of $P, Q \in \mathcal{B}(\mathbb{D}_p, \mathbb{D}_{52})[x]_{<d}$

Cost $\tilde{\mathcal{O}}(d^2\,p)$, optimal quality

Example: polynomial multiplication

$$P, Q \in \mathcal{B}(\mathbb{D}_p, \mathbb{D}_{52})[x]_{<d}$$

FFT multiplication

Cost $\tilde{\mathcal{O}}(d\,p)$, only good quality if all coefficients of same magnitude

FFT multiplication with scaling $[P(\alpha\,x)\,Q(\alpha\,x)] \circ (x/\alpha)$

Cost $\tilde{\mathcal{O}}(d\,p)$, good quality for truncated power series

FFT multiplication with scaling and cutting into blocks (VdH, Arb)

(vdH) Cost $\tilde{\mathcal{O}}(d\,p)$, good quality in terms of roots of polynomials

(Arb) Cost between $\tilde{\mathcal{O}}(d\,p)$ and $\tilde{\mathcal{O}}(d^2\,p)$, optimal quality

Naive multiplication of $P, Q \in \mathcal{B}(\mathbb{D}_p, \mathbb{D}_{52})[x]_{<d}$

Cost $\tilde{\mathcal{O}}(d^2\,p)$, optimal quality

Condition number (intrinsic)

$$
\begin{aligned}
f\colon \mathbb{R}^n &\to \mathbb{R}^k \\
\kappa_f(x) &= \limsup_{\varepsilon \to 0} \frac{\|f(x+\varepsilon) - f(x)\|}{\|f(x)\|} \Big/ \frac{\|\varepsilon\|}{\|x\|} \\
\kappa(M) &= \|M\| \, \|M^{-1}\|
\end{aligned}
$$

Condition numbers and precision loss

Condition number (intrinsic)

$$
\begin{aligned}
f\colon \mathbb{R}^n &\to \mathbb{R}^k \\
\kappa_f(x) &= \limsup_{\varepsilon \to 0} \frac{\|f(x+\varepsilon) - f(x)\|}{\|f(x)\|} \Big/ \frac{\|\varepsilon\|}{\|x\|} \\
\kappa(M) &= \|M\|\,\|M^{-1}\|
\end{aligned}
$$

Precision loss (of an algorithm)

$$
\begin{aligned}
f_p\colon \mathbb{D}_p^n &\to \mathbb{D}_p^k \\
\chi_{f_p}(x) &= \frac{\|f_p(x) - f(x)\|}{\|f(x)\|} / (2^p\,\kappa_f(x))
\end{aligned}
$$

Optimal extension (intrinsic)

$$
\begin{aligned}
f\colon \mathbb{R}^n &\to \mathbb{R}^k \\
f^*(\mathcal{B}(x,r)) &= \mathcal{B}\left(f(x), \sup_{x' \in \mathcal{B}(x,r)} |f(x') - f(x)| \right)
\end{aligned}
$$

Optimal extensions and overestimation

Optimal extension (intrinsic)

$$\begin{aligned} f\colon \mathbb{R}^n &\to \mathbb{R}^k \\ f^*(\mathcal{B}(x,r)) &= \mathcal{B}\Big(f(x), \sup_{x' \in \mathcal{B}(x,r)} |f(x') - f(x)| \Big) \end{aligned}$$

Overestimation and punctual overestimation (of an algorithm)

$$\begin{aligned} f^\bullet\colon \mathcal{B}(\mathbb{R},\mathbb{R})^n &\to \mathcal{B}(\mathbb{R},\mathbb{R})^k \\ \chi_{f^\bullet}(\mathcal{B}(x,r)) &= \inf_{\lambda} \{ f^\bullet(\mathcal{B}(x,r)) - f(x) \subseteq \lambda \, (f^*(\mathcal{B}(x,r)) - f(x)) \} \\ \chi_{f^\bullet}(x) &= \limsup_{r \to 0} \chi_{f^\bullet}(\mathcal{B}(x,r)) \end{aligned}$$

Optimal extensions and overestimation

Optimal extension (intrinsic)

$$\begin{aligned} f\colon \mathbb{R}^n &\to \mathbb{R}^k \\ f^*(\mathcal{B}(x,r)) &= \mathcal{B}\Big(f(x), \sup_{x' \in \mathcal{B}(x,r)} |f(x') - f(x)| \Big) \end{aligned}$$

Overestimation and punctual overestimation (of an algorithm)

$$\begin{aligned} f^\bullet\colon \mathcal{B}(\mathbb{R},\mathbb{R})^n &\to \mathcal{B}(\mathbb{R},\mathbb{R})^k \\ \chi_{f^\bullet}(\mathcal{B}(x,r)) &= \inf_{\lambda} \{ f^\bullet(\mathcal{B}(x,r)) - f(x) \subseteq \lambda \, (f^*(\mathcal{B}(x,r)) - f(x)) \} \\ \chi_{f^\bullet}(x) &= \operatorname*{limsup}_{r \to 0} \chi_{f^\bullet}(\mathcal{B}(x,r)) \end{aligned}$$

Question: for $f^\bullet$ constructed using $+^\bullet, -^\bullet, \times^\bullet$, do we have

$$\chi_{f^\bullet}(\mathcal{B}(x,r)) \leqslant \sup_{x \in \mathcal{B}(x,r)} \chi_{f^\bullet}(x) \ ?$$

Overestimation of standard arithmetic

Example

$$
\begin{aligned}
f(x) &= (x-1)^2 \\
f^\bullet(x^\bullet) &= (x^\bullet)^2 -^\bullet 2\, x^\bullet +^\bullet 1
\end{aligned}
$$

$$
\begin{aligned}
f^*(\mathcal{B}(x,\varepsilon)) &= \mathcal{B}(f(x), 2\,|x-1|\,\varepsilon + \mathcal{O}(\varepsilon^2)) \\
f^\bullet(\mathcal{B}(x,\varepsilon)) &= \mathcal{B}(f(x), 2\,|x|\,\varepsilon + 2\,\varepsilon + \mathcal{O}(\varepsilon^2)) \\
\chi_{f^\bullet}(x) &= \frac{|x|+1}{|x-1|}
\end{aligned}
$$

Overestimation of standard arithmetic

Example

$$\begin{aligned} f(x) &= (x-1)^2 \\ f^\bullet(x^\bullet) &= (x^\bullet)^2 -^\bullet 2\,x^\bullet +^\bullet 1 \end{aligned}$$

$$\begin{aligned} f^*(\mathcal{B}(x,\varepsilon)) &= \mathcal{B}(f(x), 2\,|x-1|\,\varepsilon + \mathcal{O}(\varepsilon^2)) \\ f^\bullet(\mathcal{B}(x,\varepsilon)) &= \mathcal{B}(f(x), 2\,|x|\,\varepsilon + 2\,\varepsilon + \mathcal{O}(\varepsilon^2)) \\ \chi_{f^\bullet}(x) &= \frac{|x|+1}{|x-1|} \end{aligned}$$

In general

$$\chi_{f^\bullet}(x) = \limsup_{\varepsilon\neq 0} \frac{(\bar\nabla f)(x)\cdot|\varepsilon|}{|(\nabla f)(x)|\cdot|\varepsilon|} \qquad \text{or } (d=1) \qquad \frac{(\bar\nabla f)(x)}{|f'(x)|}$$

$$\begin{aligned} \bar\nabla c &= (0,\ldots,0) && (c\in\mathbb{R}) \\ \bar\nabla X_k &= (0,\overset{k-1}{\ldots},0,1,0,\ldots,0) && (k\in\{1,\ldots,r\}) \\ \bar\nabla(f\pm g) &= \bar\nabla f + \bar\nabla g \\ \bar\nabla(f\,g) &= (\bar\nabla f)\,|g| + |f|\,(\bar\nabla g), \end{aligned}$$

Taylor model of order n on $\mathcal{B}(0,r)$

$$\begin{aligned}\mathcal{B}_r(\varphi,K) &= \{f \text{ analytic on } \mathcal{B}(0,r)\colon \forall|z|\leqslant r, |f(z)-\varphi(z)|\leqslant K\}\\ \mathcal{B}_r(\mathbb{D}[z]_{<n},\mathbb{D}) &= \{\mathcal{B}_r(f_0+\cdots+f_{n-1}\,z^{n-1},K)\colon f_0,\ldots,f_{n-1},K\in\mathbb{D}\}\end{aligned}$$

Taylor models (Berz et Makino)

Taylor model of order n on $\mathcal{B}(0,r)$

$$\begin{aligned}
\mathcal{B}_r(\varphi, K) &= \{f \text{ analytic on } \mathcal{B}(0,r) : \forall |z| \leqslant r, |f(z) - \varphi(z)| \leqslant K\} \\
\mathcal{B}_r(\mathbb{D}[z]_{<n}, \mathbb{D}) &= \{\mathcal{B}_r(f_0 + \cdots + f_{n-1} z^{n-1}, K) : f_0, \ldots, f_{n-1}, K \in \mathbb{D}\}
\end{aligned}$$

Product

$$\begin{aligned}
& \mathcal{B}_r(f_0 + \cdots + f_{n-1} z^{n-1}, K)\, \mathcal{B}_r(g_0 + \cdots + g_{n-1} z^{n-1}, L) \\
= \; & \mathcal{B}_r(f_0 g_0 + \cdots + (f_0 g_{n-1} + \cdots + f_{n-1} g_0) z^{n-1}, E^{\mathrm{tr}} + E^{\mathrm{rnd}}) \\
E^{\mathrm{tr}} = \; & |f_1|\,|g_{n-1}| + |f_2|\,(|g_{n-1}| + |g_{n-2}|) + \cdots + |f_{n-1}|\,(|g_1| + \cdots + |g_{n-1}|) \quad \text{(up)} \\
E^{\mathrm{rnd}} = \; & [|f_0|\,|g_{n-1}| + \cdots + (|f_0| + \cdots + |f_{n-1}|)\,|g_0|]\, C\, 2^{-p} \quad \text{(up)}
\end{aligned}$$

Taylor models (Berz et Makino)

Taylor model of order n on $\mathcal{B}(0,r)$

$$\begin{aligned}
\mathcal{B}_r(\varphi, K) &= \{f \text{ analytic on } \mathcal{B}(0,r) : \forall |z| \leqslant r, |f(z) - \varphi(z)| \leqslant K\} \\
\mathcal{B}_r(\mathbb{D}[z]_{<n}, \mathbb{D}) &= \{\mathcal{B}_r(f_0 + \cdots + f_{n-1} z^{n-1}, K) : f_0, \ldots, f_{n-1}, K \in \mathbb{D}\}
\end{aligned}$$

Product

$$\begin{aligned}
& \mathcal{B}_r(f_0 + \cdots + f_{n-1} z^{n-1}, K)\, \mathcal{B}_r(g_0 + \cdots + g_{n-1} z^{n-1}, L) \\
= \; & \mathcal{B}_r(f_0\, g_0 + \cdots + (f_0\, g_{n-1} + \cdots + f_{n-1}\, g_0) z^{n-1}, E^{\mathrm{tr}} + E^{\mathrm{rnd}}) \\
E^{\mathrm{tr}} = \; & |f_1|\,|g_{n-1}| + |f_2|\,(|g_{n-1}| + |g_{n-2}|) + \cdots + |f_{n-1}|\,(|g_1| + \cdots + |g_{n-1}|) \quad \text{(up)} \\
E^{\mathrm{rnd}} = \; & [|f_0|\,|g_{n-1}| + \cdots + (|f_0| + \cdots + |f_{n-1}|)\,|g_0|]\, C\, 2^{-p} \quad \text{(up)}
\end{aligned}$$

Intégration

$$\int \mathcal{B}_r(f_0 + \cdots + f_{n-1} z^{n-1}, K) = \mathcal{B}_r(f_1 + \cdots + \tfrac{f_{n-2}}{n-1} z^{n-1}, K\, r + \tfrac{|f_{n-1}|}{n} r^n + E^{\mathrm{rnd}})$$

Example

$$
\begin{aligned}
f(x) &= (x-1)^2 \\
f^\bullet(x^\bullet) &= (x^\bullet)^2 -^\bullet 2\,x^\bullet +^\bullet 1
\end{aligned}
$$

$$
\begin{aligned}
f^*(\mathcal{B}(x,\varepsilon)) &= \mathcal{B}(f(x), 2\,|x-1|\,\varepsilon + \mathcal{O}(\varepsilon^2)) \\
f^\bullet(\mathcal{B}(x,\varepsilon)) &= \mathcal{B}(f(x), 2\,|x|\,\varepsilon + 2\,\varepsilon + \mathcal{O}(\varepsilon^2)) \\
\chi_{f^\bullet}(x) &= \frac{|x|+1}{|x-1|}
\end{aligned}
$$

Example

$$\begin{aligned} f(x) &= (x-1)^2 \\ f^\bullet(x^\bullet) &= (x^\bullet)^2 -^\bullet 2\,x^\bullet +^\bullet 1 \end{aligned}$$

$$\begin{aligned} f^*(\mathcal{B}(x,\varepsilon)) &= \mathcal{B}(f(x), 2\,|x-1|\,\varepsilon + \mathcal{O}(\varepsilon^2)) \\ f^\bullet(\mathcal{B}(x,\varepsilon)) &= \mathcal{B}(f(x), 2\,|x|\,\varepsilon + 2\,\varepsilon + \mathcal{O}(\varepsilon^2)) \\ \chi_{f^\bullet}(x) &= \frac{|x|+1}{|x-1|} \end{aligned}$$

Idea: use a Taylor model of order $\geqslant 2$

$$\begin{aligned} f^{\text{Taylor}}(\mathcal{B}_r(x,0)) &= \mathcal{B}_r(y_0 + y_1\,t, \varepsilon) \\ f^\bullet(\mathcal{B}(x,r)) &= \mathcal{B}(y_0, |y_1|\,r + \varepsilon) \\ \chi_{f^\bullet}(x) &= 1 \end{aligned}$$

Example challenge

Complexity to solve

$$P(z) \;=\; 0,$$

with $P \in \mathbb{C}[z]$, $\deg P = d$, at bit precision p?

Challenge 1. Optimal complexity, uniformly in p and d

Example challenge

Complexity to solve

$$P(z) \;=\; 0,$$

with $P \in \mathbb{C}[z]$, $\deg P = d$, at bit precision p?

Challenge 1. Optimal complexity, uniformly in p and d

Challenge 2. Relation to easier question: multipoint evaluation

$p \gg d \rightsquigarrow \tilde{\mathcal{O}}(p\,d)$

Example challenge

Complexity to solve

$$P(z) \;=\; 0,$$

with $P \in \mathbb{C}[z]$, $\deg P = d$, at bit precision p?

Challenge 1. Optimal complexity, uniformly in p and d

Challenge 2. Relation to easier question: multipoint evaluation

$p \gg d \rightsquigarrow \tilde{\mathcal{O}}(p\,d)$
$p \ll d \rightsquigarrow \tilde{\mathcal{O}}(p\,d^2)$

Example challenge

Complexity to solve

$$P(z) \;=\; 0,$$

with $P \in \mathbb{C}[z]$, $\deg P = d$, at bit precision p?

Challenge 1. Optimal complexity, uniformly in p and d

Challenge 2. Relation to easier question: multipoint evaluation

$p \gg d \rightsquigarrow \tilde{\mathcal{O}}(p\,d)$
$p \ll d \rightsquigarrow \tilde{\mathcal{O}}(p\,d^2)$
$p \ll \sqrt[3]{d} \rightsquigarrow \tilde{\mathcal{O}}(d^{3/2}\,p^{3/2})$

Example challenge

Complexity to solve

$$P(z) \;=\; 0,$$

with $P \in \mathbb{C}[z]$, $\deg P = d$, at bit precision p?

Challenge 1. Optimal complexity, uniformly in p and d

Challenge 2. Relation to easier question: multipoint evaluation

$p \gg d \rightsquigarrow \tilde{\mathcal{O}}(p\,d)$
$p \ll d \rightsquigarrow \tilde{\mathcal{O}}(p\,d^2)$
$p \ll \sqrt[3]{d} \rightsquigarrow \tilde{\mathcal{O}}(d^{3/2}\,p^{3/2})$

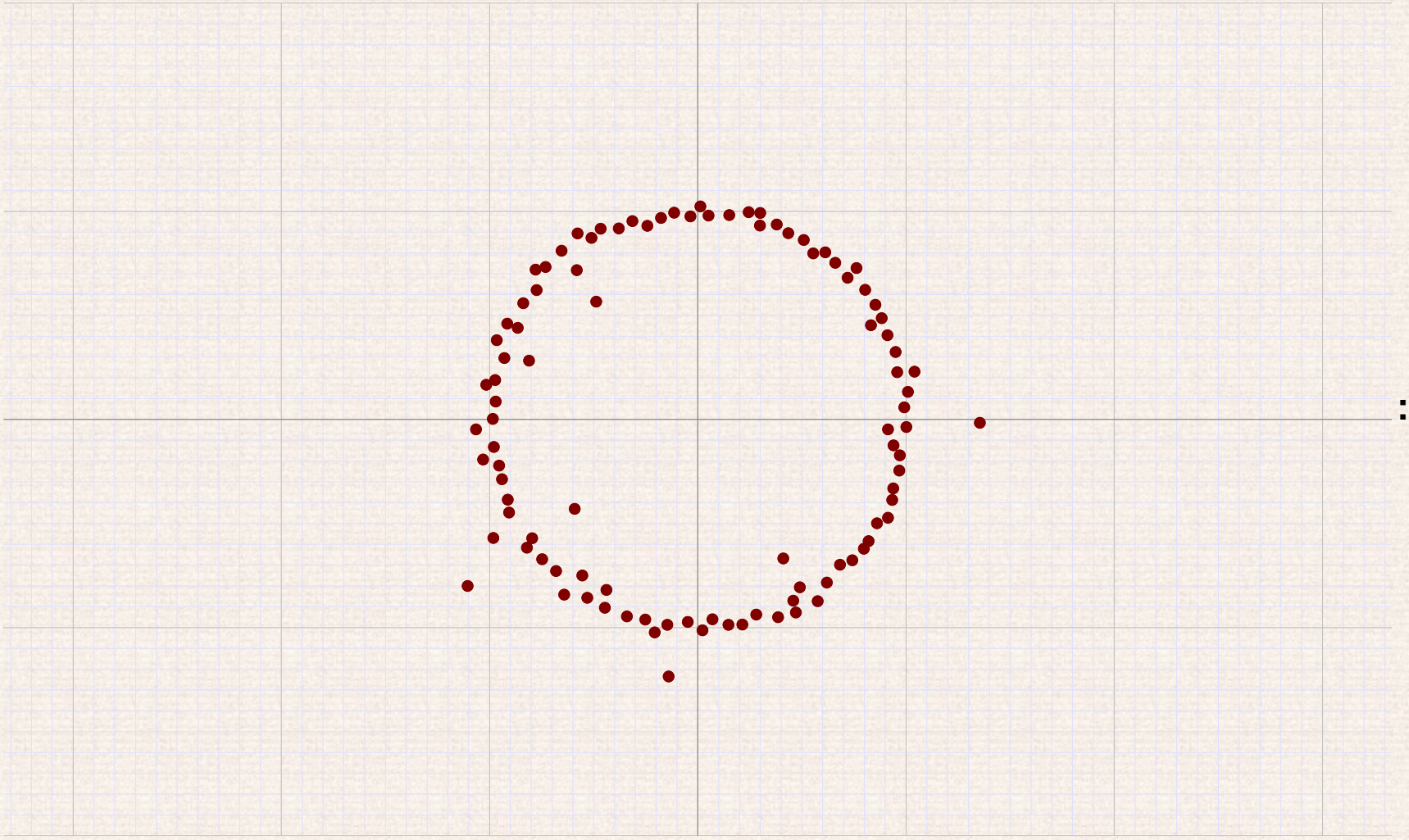
Challenge 3. Dependence on condition number

Random polynomials (well conditioned)

```
Mmx] use "analyziz"; include "graphix/points.mmx";
Mmx] pi == 4.0 * arctan 1.0;
Mmx] rnd () == exp (complex (0.0, uniform_deviate (0.0, 2*pi)));
Mmx] p == polynomial (rnd () | i in 0 to 100);
Mmx] $points roots p
Mmx]
```

Random polynomials (well conditioned)

```
Mmx] use "analyziz"; include "graphix/points.mmx";
Mmx] pi == 4.0 * arctan 1.0;
Mmx] rnd () == exp (complex (0.0, uniform_deviate (0.0, 2*pi)));
Mmx] p == polynomial (rnd () | i in 0 to 100);
Mmx] $points roots p
```



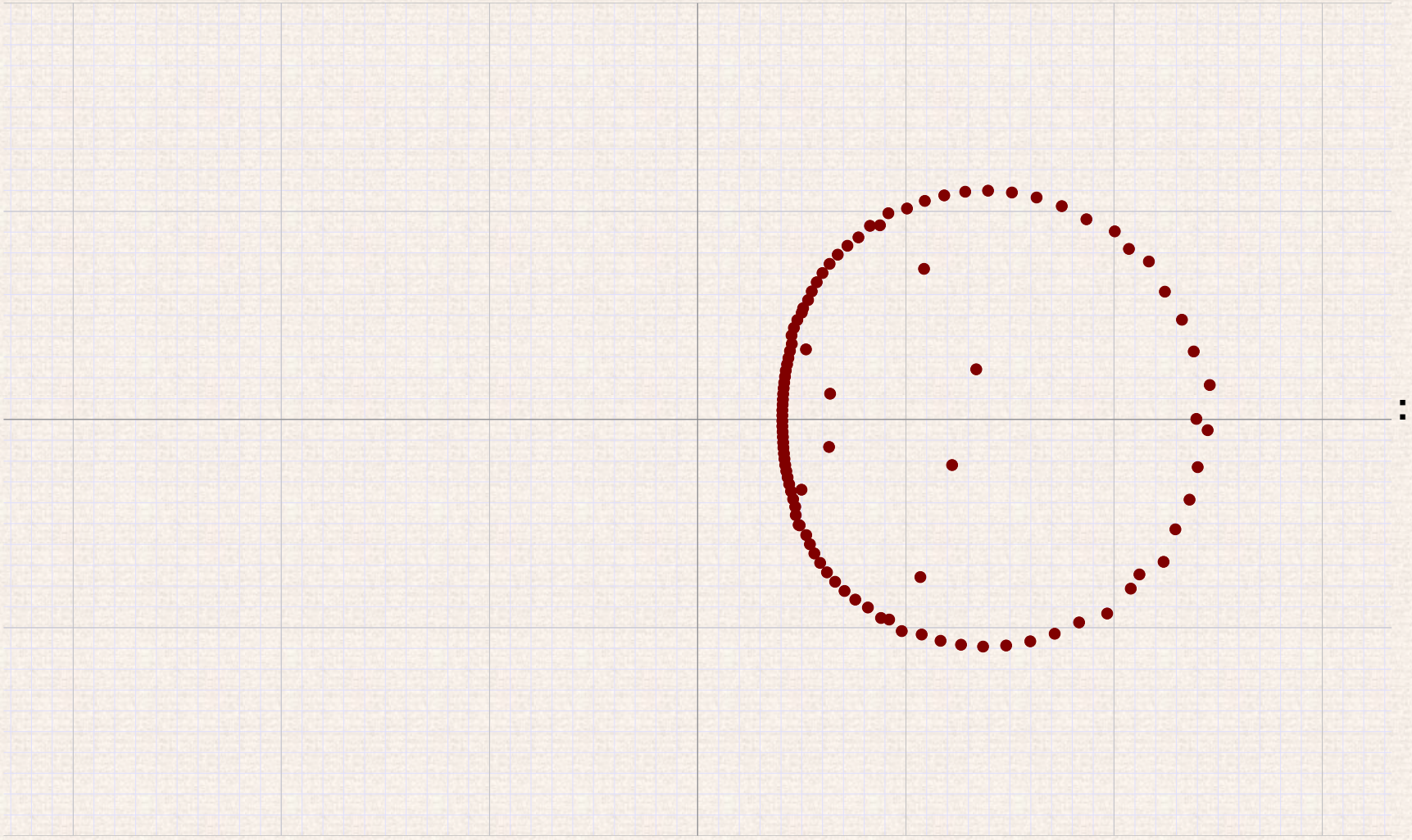
Compound

Multiple roots

```
Mmx] p == poly (1.0, -1.0)^100;  // (1 - z)^100
Mmx] $points (roots p)
Mmx]
```

Multiple roots

```
Mmx] p == poly (1.0, -1.0)^100;  // (1 - z)^100
Mmx] $points (roots p)
```



Random roots on half circle

```
Mmx] v == [ rnd () | i in 1 to 100 ];
Mmx] v == [ sqrt rnd () | i in 1 to 100 ];
Mmx] p == annulator v;
Mmx] $points (roots p)
Mmx]
```

Random roots on half circle

```
Mmx] v == [ rnd () | i in 1 to 100 ];
Mmx] v == [ sqrt rnd () | i in 1 to 100 ];
Mmx] p == annulator v;
Mmx] $points (roots p)
```

Compound

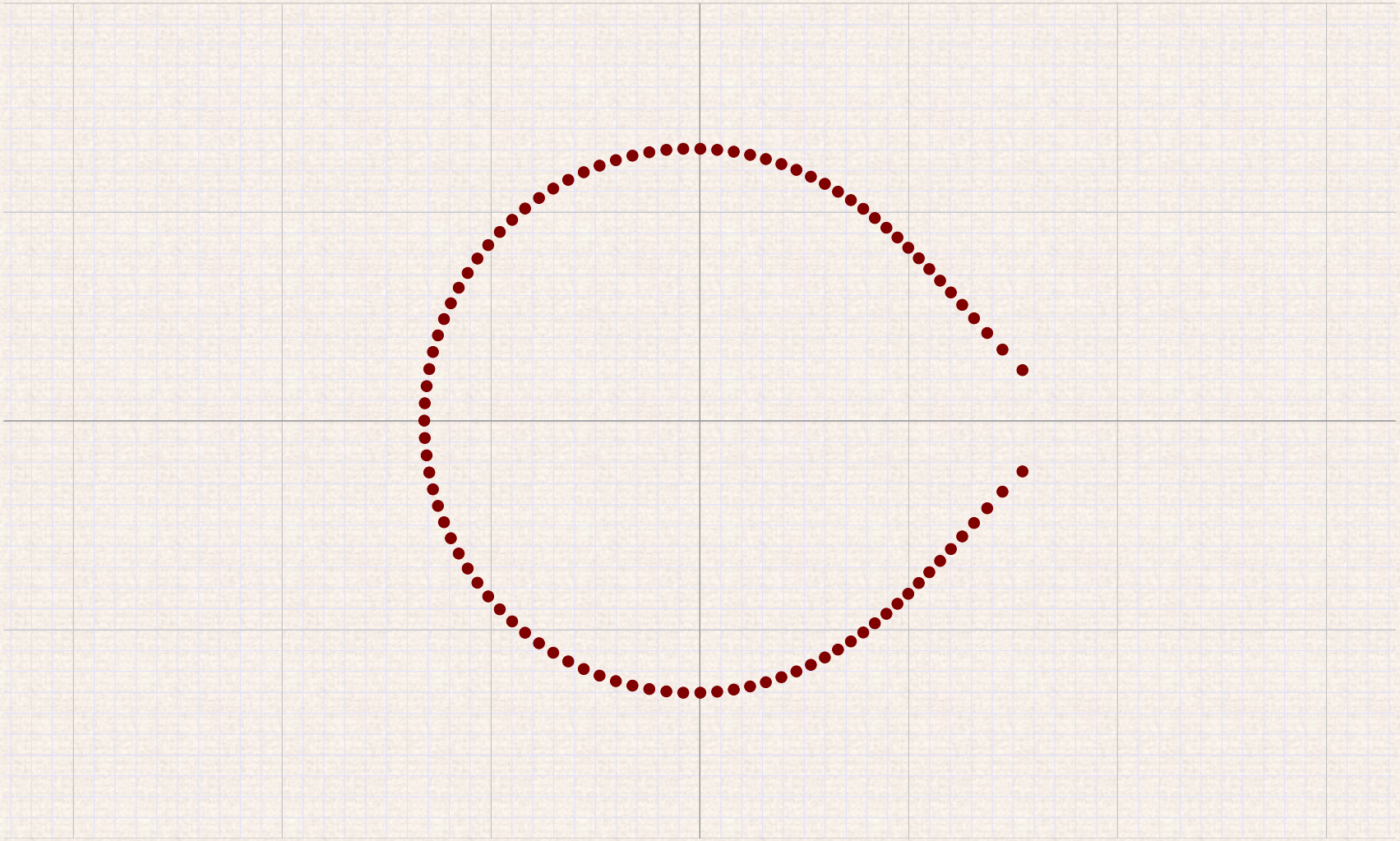
Truncated power series

```
Mmx] z == series (0.0, 1.0);
Mmx] B == exp (exp z - 1);
Mmx] p == B[0,100];
Mmx] $points (0.5 * roots p)
Mmx]
```

Truncated power series

```
Mmx] z == series (0.0, 1.0);
Mmx] B == exp (exp z - 1);
```

```
Mmx] p == B[0,100];
Mmx] $points (0.5 * roots p)
```



Truncated power series

```
Mmx] z == series (0.0, 1.0);
Mmx] f == integrate exp (-z*z);
Mmx] p == f[0,200];
Mmx] $points (0.2 * roots p)
Mmx]
```

Truncated power series

```
Mmx] z == series (0.0, 1.0);
Mmx] f == integrate exp (-z*z);
```

```
Mmx] p == f[0,200];
Mmx] $points (0.2 * roots p)
```