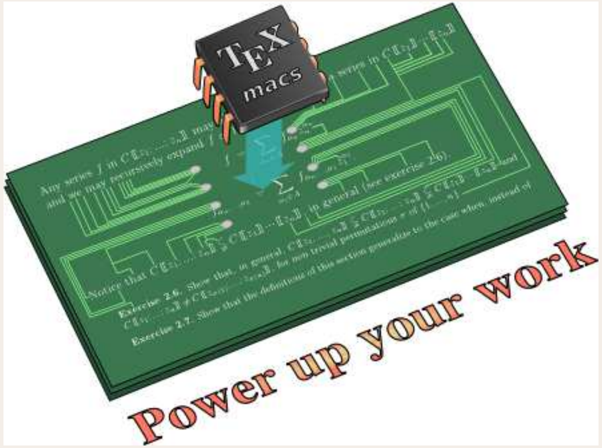
Effective real numbers in Mmxlib

by Joris van der Hoeven



Presentation with GNU TeXmacs (www.texmacs.org)

Effective real numbers in Mmxlib

http://www.mathemagix.org/mmxweb/web/welcome-mml.en.html

Mmx ≫ x: Real == sin sin real 2;

Mmx ≫ x

$7.891\,10^{-1}$ Real

Mmx ≫ approximate(x, 1.0e−35);

$7.890723435728883614314030424868841 2\,10^{-1}$ Interval

Mmx ≫ M: Matrix Real == $\begin{pmatrix} x & x+2 \\ 2-x^2 & \cos x \end{pmatrix}$;

Mmx ≫ M;

$$\begin{bmatrix} 7.891\,10^{-1} & 2.789 \\ 1.377 & 7.045\,10^{-1} \end{bmatrix}$$ Matrix(Real)

Mmx ≫ M^{20};

$$\begin{bmatrix} 2.284\,10^{8} & 3.181\,10^{8} \\ 1.571\,10^{8} & 2.188\,10^{8} \end{bmatrix}$$ Matrix(Real)

Mmx ≫ $\exp(x + \exp(-\mathrm{real}\,100)) - \exp(x)$;

$8.189\,10^{-44}$ Real

Mmx ≫ $\exp(x + \exp(-\mathrm{real}\,1000)) - \exp(x)$;

Mmx ≫

Effective analytic functions in Mmxlib

Mmx ≫ z: Analytic == analytic $(0, 1)$;

Mmx ≫ $\exp(z)$;

$1.000 + 1.000\ z + 5.000\ 10^{-1}\ z^2 + 1.667\ 10^{-1}\ z^3 + 4.167\ 10^{-2}\ z^4 + 8.333\ 10^{-3}\ z^5 + 1.389\ 10^{-3}\ z^6 + 1.984\ 10^{-4}\ z^7 + 2.480\ 10^{-5}\ z^8 + 2.756\ 10^{-6}\ z^9 + O(z^{10})$ Analytic

Mmx ≫ $\exp(z)$[int 20];

$4.110\ 10^{-19}$ Complex

Mmx ≫ ℓ: Analytic == $\log(1 - z)$;

Mmx ≫ radius(ℓ);

$9.9993772618472576135\ 9\ 10^{-1}$ Floating

Mmx ≫ evaluate$(\ell$, complex$(1/2))$;

$-6.931\ 10^{-1}$ Complex

Mmx ≫ continuate$(\ell$, complex$(1/2))$;

$-6.931\ 10^{-1} - 2.000\ z - 2.000\ z^2 - 2.667\ z^3 - 4.000\ z^4 - 6.400\ z^5 - 1.067\ 10^1\ z^6 - 1.829\ 10^1\ z^7 - 3.200\ 10^1\ z^8 - 5.689\ 10^1\ z^9 + O(z^{10})$ Analytic

Mmx ≫ continuate$(\ell$, turn(complex$(1)))$;

$6.283\ i - 1.000\ z - 5.000\ 10^{-1}\ z^2 - 3.333\ 10^{-1}\ z^3 - 2.500\ 10^{-1}\ z^4 - 2.000\ 10^{-1}\ z^5 - 1.667\ 10^{-1}\ z^6 - 1.429\ 10^{-1}\ z^7 - 1.250\ 10^{-1}\ z^8 - 1.111\ 10^{-1}\ z^9 + O(z^{10})$ Analytic

Solving differential equations

$$f'' = (z^2 + 1)\, f' + \mathrm{e}^z\, f; \qquad f(0) = 1,\ f'(0) = 1 + 2\,\mathrm{i}.$$

Mmx ≫ $f\colon \mathrm{Analytic} == \mathrm{solve\_lde}((z^2 + 1, \exp(z)), (\mathrm{complex}(1), \mathrm{complex}(1, 2)));$

Mmx ≫ $f;$

$1.000 + (1.000 + 2.000\,\mathrm{i})\, z + (1.000 + 1.000\,\mathrm{i})\, z^2 + (6.667\ 10^{-1} + 1.000\,\mathrm{i})\, z^3 + (5.417\ 10^{-1} + 5.833\ 10^{-1}\,\mathrm{i})\, z^4 + (3.500\ 10^{-1} + 4.833\ 10^{-1}\,\mathrm{i})\, z^5 + (2.278\ 10^{-1} + 2.750\ 10^{-1}\,\mathrm{i})\, z^6 + (1.343\ 10^{-1} + 1.742\ 10^{-1}\,\mathrm{i})\, z^7 + (7.882\ 10^{-2} + 9.767\ 10^{-2}\,\mathrm{i})\, z^8 + (4.361\ 10^{-2} + 5.572\ 10^{-2}\,\mathrm{i})\, z^9 + O(z^{10})$ Analytic

Mmx ≫ $u\colon \mathrm{Complex} == \mathrm{evaluate}(f, \mathrm{complex}(1/10));$

Mmx ≫ $u;$

$1.111 + 2.111\ 10^{-1}\,\mathrm{i}$ Complex

Mmx ≫ $\mathrm{approximate}(u, 1.0e{-}81);$

$1.1107245753779445710229272557483056635705254130884819662668704756751790282839283 4 + 2.1106346012282867466007605052618438398248510727864880851400427655460836641117663\ 10^{-1}\,\mathrm{i}$

Complexify(Interval)

Mmx ≫

Definition of effective real numbers

- $\tilde{x} \in \mathbb{D} = \mathbb{Z}\, 2^{\mathbb{Z}}$ is an ε-approximation of $x \in \mathbb{R}$ if $|\tilde{x} - x| < \varepsilon$.
- *Approximation algorithm* for x: computes $\varepsilon \longmapsto \varepsilon$-approximation of x.
- *Effective real number*: $x \in \mathbb{R}$ which admits an approximation algorithm.
- *Complexity* of x: time needed to compute a 2^{-l}-approximation.
- No zero-test for effective real numbers.
- References: Bishop and Bridges, Blanck, Müller, vdH, etc.

Implementation by layers

Fast arithmetic.

Fast computations on mantissas (Karatsuba, FFT, Brent, Chudnovsky^2, VdH, etc.).

Implementations: Gmp, Cln, ...

Low-level validated arithmetic.

Correct rounding or well-specified bounds on the error.

Implementations: Mpfr, ...

High-level validated arithmetic.

Template types for intervals, balls, Lipschitz balls, etc.

Implementations: Mpfi, Irram, Mmxlib, ...

High-level interface.

Given a required precision for the result, automatically find precisions for all intermediate computations.

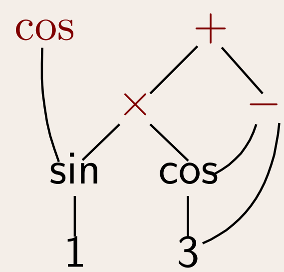
Implementations: Irram, Mmxlib, ...

Modeling computations with effective real numbers

- Example: addition

```
class add_real_rep: public real_rep {
  real x, y;
  add_real_rep (const real& x2, const real& y2):
    x (x2), y (y2) {}
  dyadic approximate (const dyadic& eps) {
    return x->approximate (eps/2) + y->approximate (eps/2); }
};
```

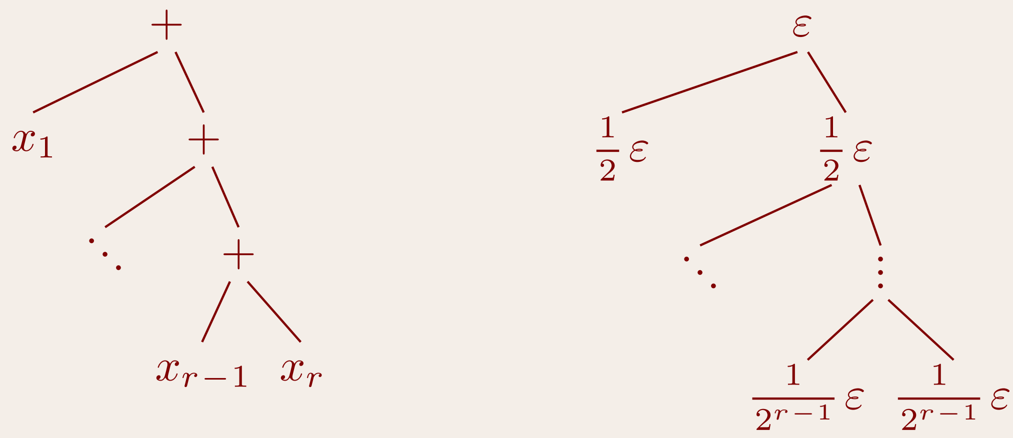
- Model sets of effective real numbers by acyclic graphs:



- Computations stored in memory → don't use classical numerical algorithms.

A priori error estimates

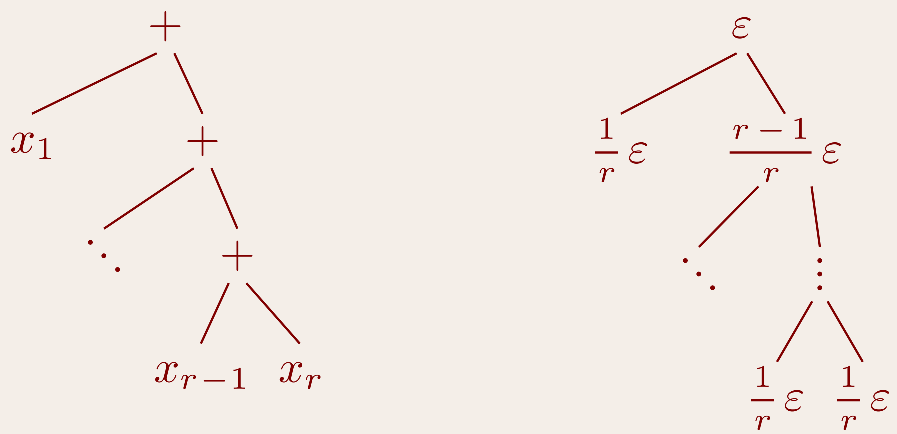
- Distribute tolerance ε *a priori* over nodes of n-ary operations ($n > 1$).
- Can be bad in case of badly nested expressions:



- Possible loss of $\log w$ bits of precision; still unacceptable.

A priori error estimates

- Distribute tolerance ε *a priori* over nodes of n-ary operations ($n > 1$).
- Balanced error estimates: redistribute as a function of *weight*:



- Possible loss of $\log w$ bits of precision; still unacceptable.

A posteriori error estimates

- Perform whole computation using interval arithmetic.

  While result not precise enough:

  Double precision and redo entire computation.

- First improvement:

  For each instance of `real_rep`, keep best current approximation in memory.

- Second improvement:

  Don't double precision, but estimated computation time.

Model for complexity analysis

Global approximation problem

Input: an acyclic graph G with for each node $\alpha \in G$:

- A real function f_α from the library.

  Induces by induction a real number $x_\alpha = f_\alpha(x_{\alpha[1]}, \ldots, x_{\alpha[|\alpha|]})$.

- A tolerance $\varepsilon_\alpha \in \mathbb{D}^>$.

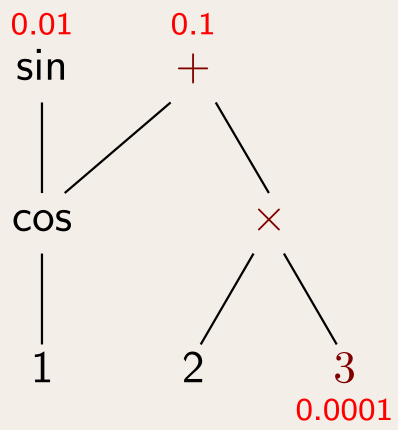
Output: for each node an interval $\boldsymbol{x}_\alpha \ni x_\alpha$ with

- $r_{\boldsymbol{x}_\alpha} < \varepsilon_\alpha$.
- $\boldsymbol{x}_\alpha \supseteq f_\alpha(\boldsymbol{x}_{\alpha[1]}, \ldots, \boldsymbol{x}_{\alpha[|\alpha|]})$.

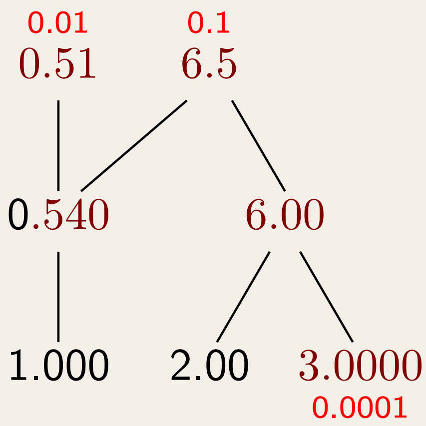
Drawbacks

- Does not model incremental computations.
- No dependency of computations on intermediate results.

Example



Example



Results

Doubling computation time approach

- Final complexity vs. total complexity $t^{\text{fin}} \leqslant t \leqslant (\log_2 t^{\text{fin}})\, t^{\text{fin}}$.
- Final complexity vs. optimal complexity $t^{\text{opt}} \leqslant t^{\text{fin}} \leqslant 2\, s\, t^{\text{opt}}$.

Faster approaches in "the rigid case"

- Rigid dag: for each $x_\alpha = f_\alpha(x_{\alpha[1]}, \ldots, x_{\alpha[|\alpha|]})$, the $\partial x_\alpha / \partial x_{\alpha[i]}$ are known with a fixed relative precision (say $1/2$).
- A sufficient precision can be determined quite sharply beforehand (IRRAM).
- Using backward error bounds and balancing, the precisions can be adapted locally in a quasi-optimal way.