On the development of a reliable integrated computational environment

Brigitte Verdonk

Overview

Reliable

Integrated

Computational

Environment

- compiler
- operating system
- hardware

Floating-point standardization

IEEE 754-854

$$\mathbb{F}_0(\beta, t, L, U) = \{\pm d_0.d_1 \ldots d_{t-1} \times \beta^e \mid$$
$$0 \le d_i \le \beta - 1, d_0 \ne 0, L \le e \le U\}$$

(base β, precision t and normalized exponent range $[L, U]$)

Principle 1: exact rounding

Rounding $\bigcirc : \mathbb{R}_0 \to X$ satisfies

$$\bigcirc(x) = x \qquad \forall x \in X$$
$$x \le y \Rightarrow \bigcirc(x) \le \bigcirc(y) \qquad \forall x, y \in \mathbb{R}_0$$

Roundings: round to nearest, round up, round down, truncate

Principle 2: exactly rounded operations

$$x \circledast y = \bigcirc(x * y) \qquad \forall x, y \in X$$

Operations: $+, -, \times, /, \sqrt{\ }$, remainder, I/O and conversions between number sets X_i.

IEEE 754-854

Principle 3: closed number system

$$\begin{aligned} \overline{\mathbb{R}} &= \mathbb{R}_0 \cup S\mathbb{R}_0 \\ &= \mathbb{R}_0 \cup \{0, (\pm)\infty, \mathrm{INV}\} \end{aligned}$$

$$\overline{X} \stackrel{?}{=} X \cup SX$$

- denormal numbers
  $\pm d_0.d_1 \ldots d_{t-1} \times \beta^L, d_0 = 0, \exists d_i \neq 0$

  $\rightarrow$ no underflow exception in addition

- signed infinities: $\pm\infty$

  $\rightarrow$ affine arithmetic

- signed zeroes: ± 0

  $+0 \approx +\epsilon$, $-0 \approx -\delta$, $(+0) \oplus (-0) = ?$

  – IEEE-based arithmetic:
    $(+0) \oplus (-0) = +0$

  – TI-based arithmetic: $(+0) \oplus (-0) = 0$

IEEE 754-854

- NaN: result of invalid operation

$$+\infty + (-\infty) = \mathrm{NaN}$$

$$+\infty - (+\infty) = \mathrm{NaN}$$

$$-\infty - (-\infty) = \mathrm{NaN}$$

$$\frac{\pm\infty}{\pm\infty} = \mathrm{NaN}$$

$$\pm 0 \times \pm\infty = \mathrm{NaN}$$

$$\frac{\pm 0}{\pm 0} = \mathrm{NaN}$$

$$x \,\mathrm{REM}\, \pm 0 = \mathrm{NaN}$$

$$\pm\infty \,\mathrm{REM}\, y = \mathrm{NaN}$$

$$\sqrt{x} = \mathrm{NaN} \text{ when } x < 0$$

Conclusion:

$$\overline{\mathbb{F}} = \mathbb{F}_0 \cup \{\pm 0, (\pm)\infty, \mathrm{denormals}, \mathrm{NaN}\} = \mathbb{F}_0 \cup S\mathbb{F}_0$$

Strange output?

[Rump]:

$a = 77617, b = 33096$

$$
\begin{aligned}
z &= 333.75b^6 + a^2(11a^2b^2 - b^6 - 121b^4 - 2) \\
x &= 5.5b^8 \\
y &= z + x + \frac{a}{2b} \\
&\stackrel{?}{=} 5.76461\ldots \times 10^{17} \\
&\stackrel{?}{=} 6.33825\ldots \times 10^{29} \\
&\stackrel{?}{=} 1.1726\ldots \\
&\stackrel{?}{=} -0.827396\ldots
\end{aligned}
$$

Strange output?

INTEL Pentium (Borland C++ compiler):

(1) literals and variables: t=24

intermediate results: t=64

$$\tilde{y} = 5.76461\ldots \times 10^{17}$$

(2) literals and variables: t=24

intermediate results: t=24

$$\tilde{y} = 6.33825\ldots \times 10^{29}$$

(3) literals and variables: t=53

intermediate results: t=64

$$\tilde{y} = 5.76461\ldots \times 10^{17}$$

(4) literals and variables: t=53

intermediate results: t=53

$$\tilde{y} = 1.1726\ldots$$

(5) literals and variables: t=64

intermediate results: t=64

$$\tilde{y} = 5.76461\ldots \times 10^{17}$$

Strange output?

SUN Sparc (Sun f77, Sun cc):

(1) literals and variables: t=24

intermediate results: t=64

$\tilde{y}$ = unavailable

(2) literals and variables: t=24

intermediate results: t=24

$\tilde{y} = 6.33825\ldots \times 10^{29}$

(3) literals and variables: t=53

intermediate results: t=64

$\tilde{y}$ = unavailable

(4) literals and variables: t=53

intermediate results: t=53

$\tilde{y} = 1.1726\ldots$

(5) literals and variables: t=113

intermediate results: t=113

$\tilde{y} = 1.1726\ldots$

Machine epsilon

```
#include <iostream.h>
int main()
{ double epsilonZero, epsilonOne, epsTmp;
  int i;

  epsilonZero = 1.0; i = 0;
  while (epsilonZero  > 0.0)
     {
     i = i+1;
     epsTmp = epsilonZero;
     epsilonZero = epsilonZero/2.0;
     }
  cout << i << "  " << epsTmp << "\n";

  epsilonOne = 1.0; i = 0;
  while (1.0+epsilonOne > 1.0)
     {
     i = i+1;
     epsTmp = epsilonOne;
     epsilonOne = epsilonOne/2.0;
     }
  cout << i << "  " << epsTmp << "\n"; }
```

```
=1= CC -o macheps.x -fast macheps.cpp
=2= macheps.x
   1023    2.22507e-308
   53      2.22045e-16
=3= CC -o macheps.x macheps.cpp
=4= macheps.x
   1075    4.94066e-324
   53      2.22045e-16
```

Counting to six

[Higham]:

$$2 - 1$$

$$\left(\frac{1}{\cos(100\pi + \pi/4)}\right)^2$$

$$3 \times \frac{\tan(\arctan(10000))}{10000}$$

$$\left(\left(\dots\left(\sqrt{\sqrt{\dots\sqrt{4}}}\right)^2\dots\right)^2\right)^2$$

$$5 \times \frac{(1 + e^{-100}) - 1}{(1 + e^{-100}) - 1}$$

$$\frac{\ln(e^{6000})}{1000}$$

Counting to six

$$
\begin{aligned}
\text{one} &= \texttt{1.0000000000000000} \\
\text{two} &= \texttt{2.0000000000001110} \\
\text{three} &= \texttt{2.9999999999971618} \\
\text{four} &= \texttt{2.7182818081824731} \\
\text{five} &= \texttt{NaN} \\
\text{six} &= \texttt{Infinity}
\end{aligned}
$$

Exception handling

[Kahan]:

$$\forall x \in \mathbb{R}_0 \cup \{0, \pm\infty\} : x^0 = 1$$

$$\Downarrow$$

$$\mathrm{NaN}^0 = 1$$

[IEEE 754-854]:

$$\begin{aligned}
x_1 &= +\infty \\
x_2 &= -\infty \\
x &= x_1 + x_2 \Rightarrow x = \mathrm{NaN} \\
y &= \sqrt{-1} \quad \Rightarrow y = \mathrm{NaN}
\end{aligned}$$

$$\begin{aligned}
x^0 &= \mathrm{NaN} \\
y^0 &= \mathrm{NaN}
\end{aligned}$$

Reliable

Integrated

Computational

Environment

- libraries: numerics, graphics, statistics, …
- Matlab, Octave, ...
- Maple, Mathematica, ...

The toolkit era

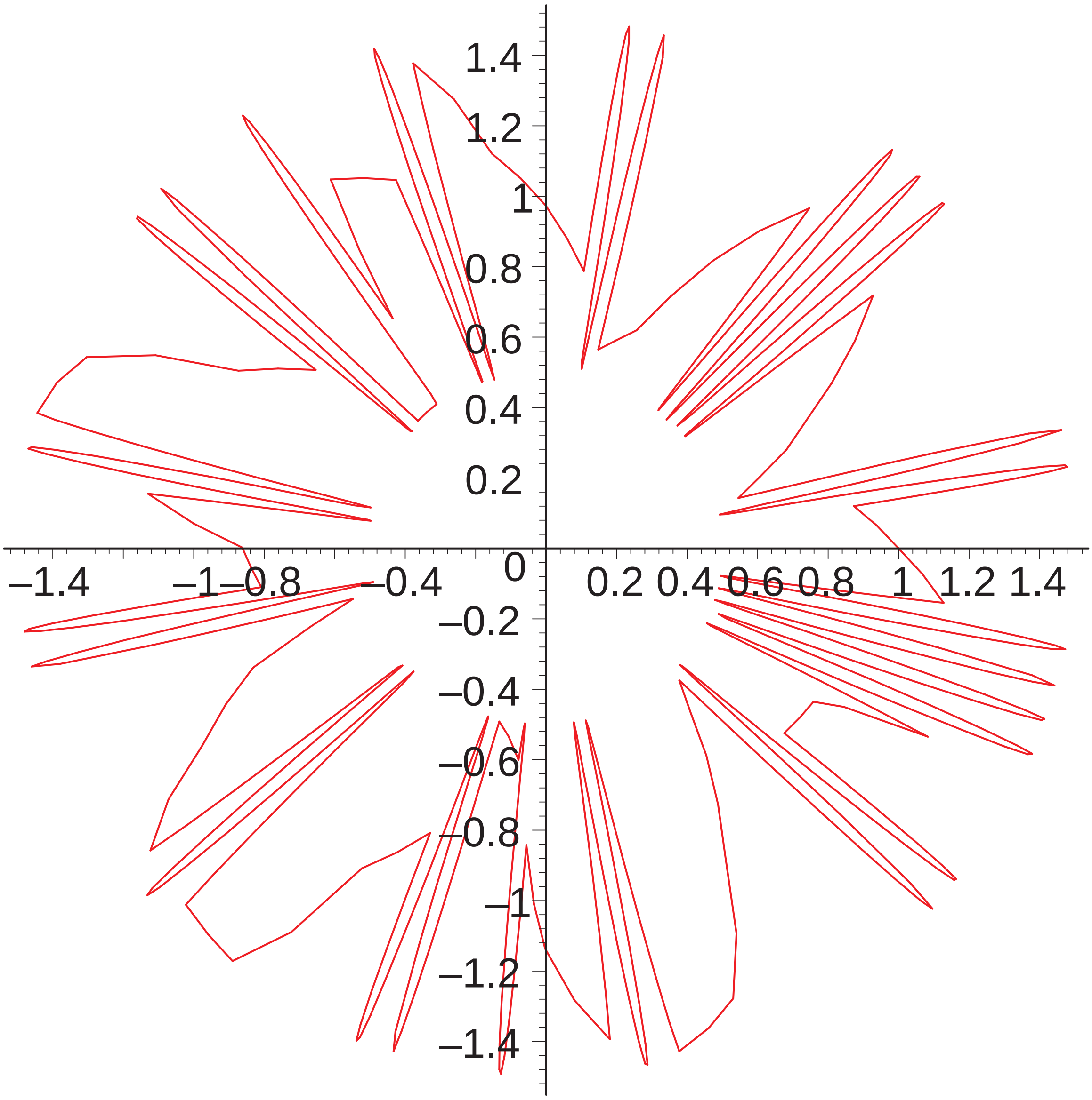
Graphics

“daisy with 90 leaves” [Geulig & Krämer]:

$$
\begin{aligned}
r &= \frac{1}{2}\sin(90t) \\
x &= (1+r)\cos(t) \\
y &= (1+r)\sin(t) \qquad t = 0 \ldots 2\pi
\end{aligned}
$$

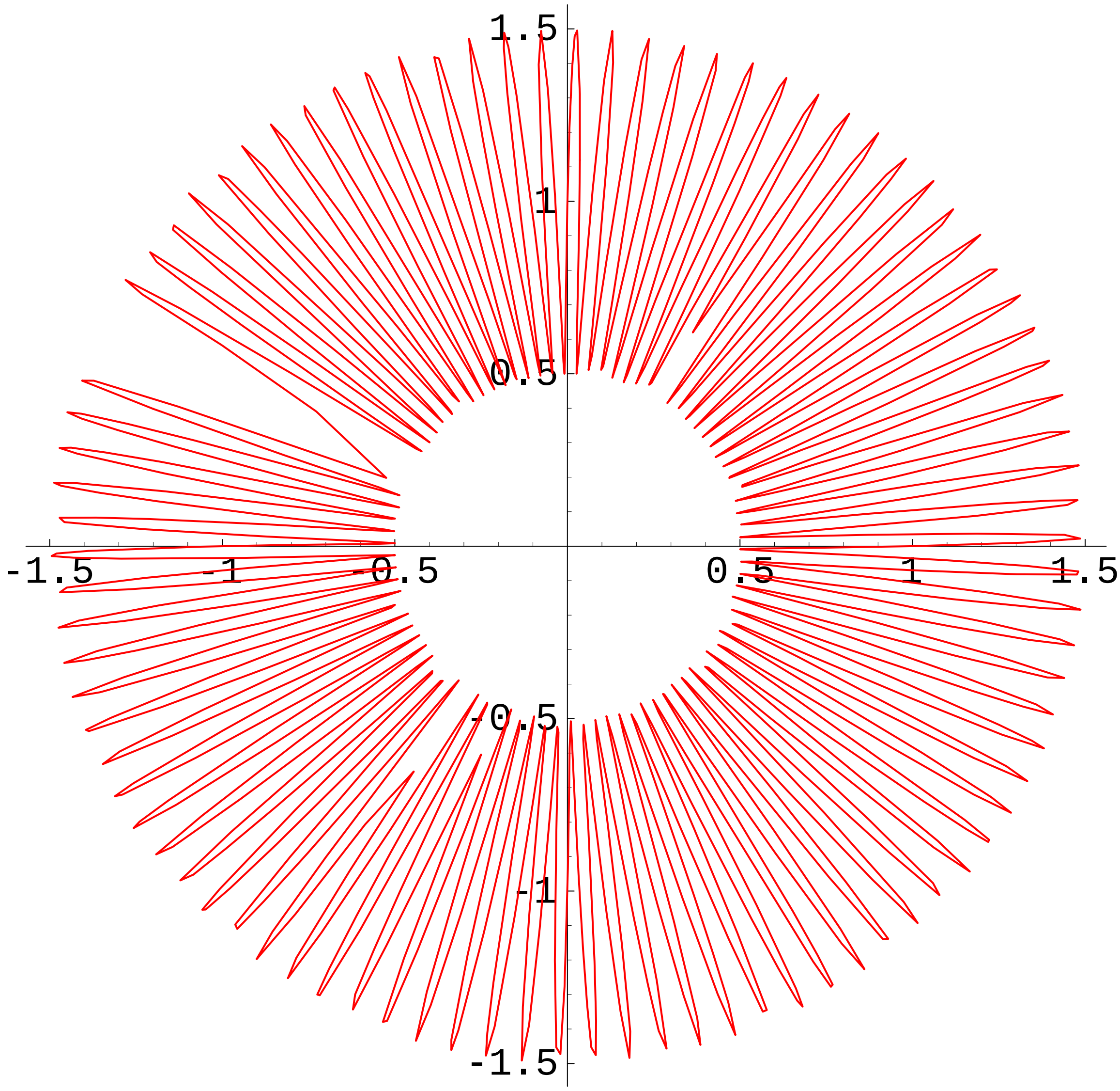
Graphics

Maple 5.0:



Graphics

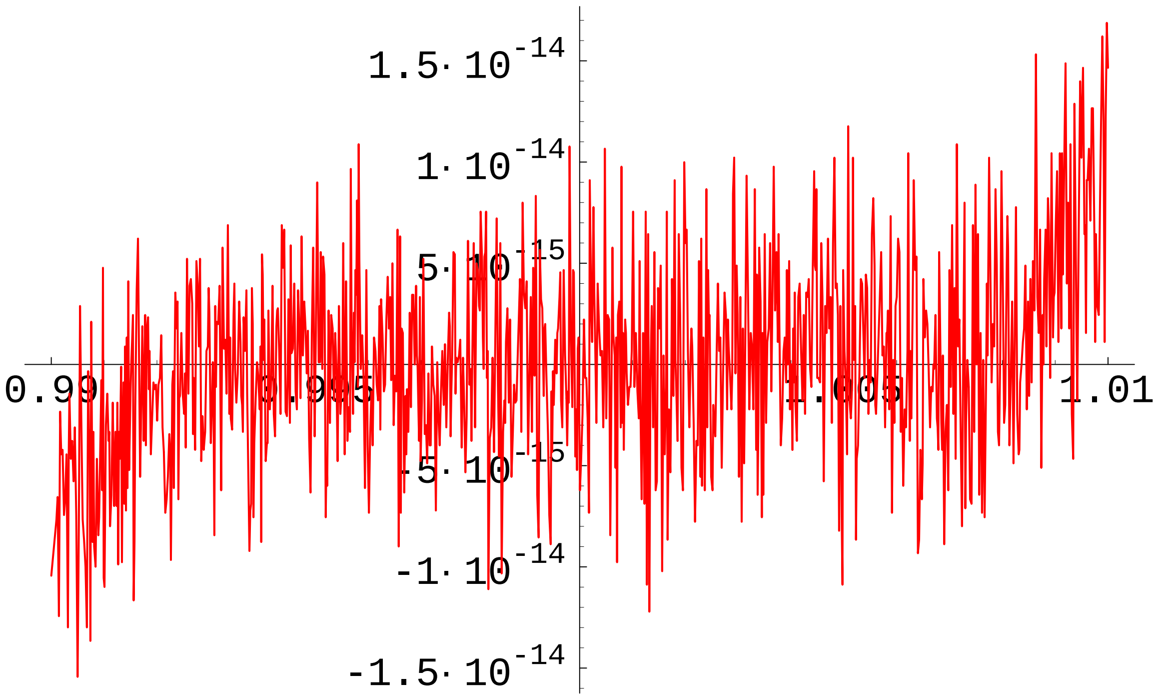
Mathematica 4.02:



Graphics

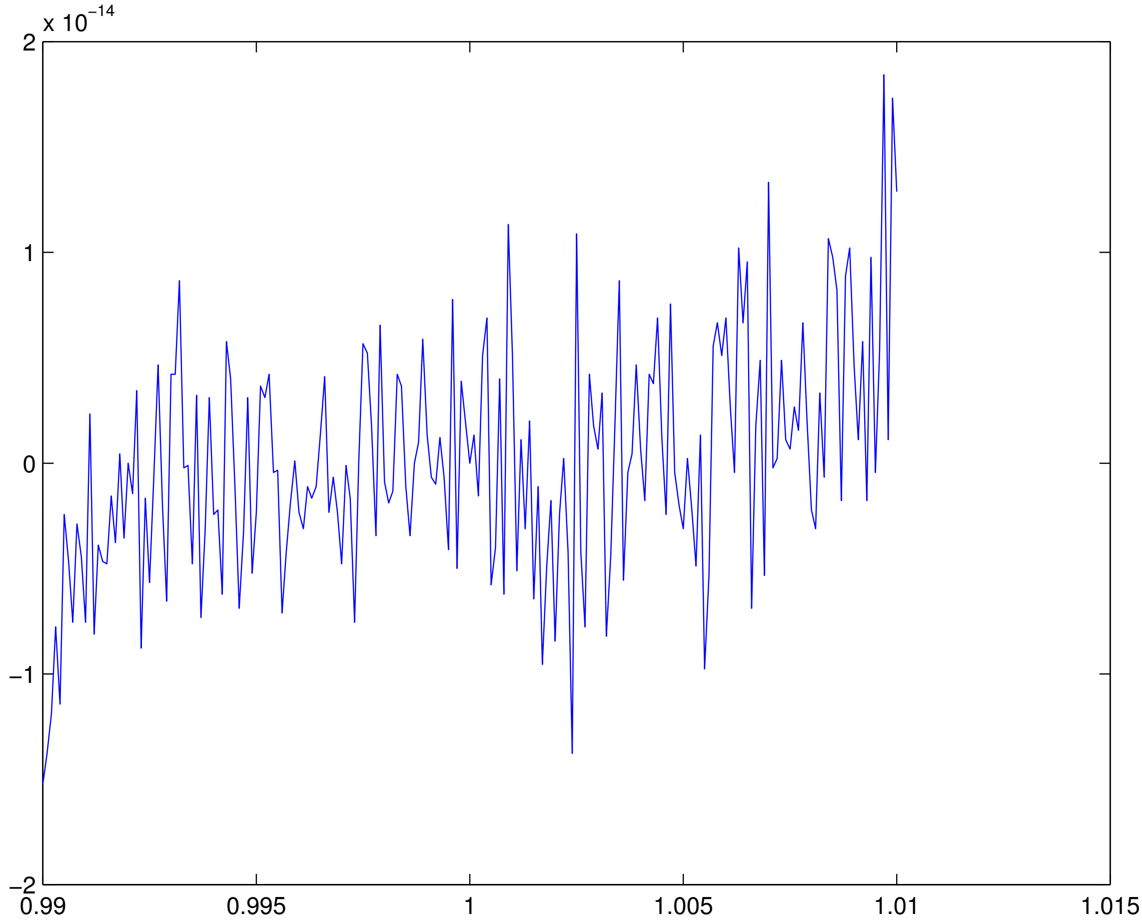
$$f(x) = \sum_{i=0}^{7} \binom{7}{i} (-1)^{7-i} x^i = (x-1)^7$$

Mathematica 4.02:



Graphics

Matlab 5.03:



Tough sign problem

Wilkinson polynomial

$$
\begin{aligned}
p_0(x) &= (x-1)\ldots(x-20)\\
&= 2432902008176640000 - 8752948036761600000x + \\
&\quad 13803759753640704000x^2 - 12870931245150988800x^3 + \\
&\quad 8037811822645051776x^4 - 3599979517947607200x^5 + \\
&\quad 1206647803780373360x^6 - 311333643161390640x^7 + \\
&\quad 63030812099294896x^8 - 10142299865511450x^9 + \\
&\quad 1307535010540395x^{10} - 135585182899530x^{11} + \\
&\quad 11310276995381x^{12} - 756111184500x^{13} + \\
&\quad 40171771630x^{14} - 1672280820x^{15} + 53327946x^{16} \\
&\quad -1256850x^{17} + 20615x^{18} - 210x^{19} + x^{20}
\end{aligned}
$$

Real roots:

$$z_i = i \qquad\qquad i = 1, \ldots 20$$

Modified Wilkinson polynomial

$$p_1(x) = p_0(x) - 2^{-23}x^{19}$$

Real roots:

$$
\begin{array}{ll}
z_1 = 1.0000000\ldots & z_2 = 2.0000000\ldots\\
z_3 = 3.0000000\ldots & z_4 = 4.0000000\ldots\\
z_5 = 4.9999993\ldots & z_6 = 6.0000069\ldots\\
z_7 = 6.9996972\ldots & z_8 = 8.0072676\ldots\\
z_9 = 8.9172502\ldots & z_{10} = 20.846908\ldots
\end{array}
$$

Tough sign problem

Aim: Obtain reliable bounds for real roots of
$p_1(x) = 0$

False Position Method

Given:

$$(x_\ell, f_\ell), (x_h, f_h) \qquad f_\ell < 0, f_h > 0$$

Compute:

$$x_n = x_h + f_h \tfrac{x_h - x_l}{f_l - f_h}$$
$$f_n = f(x_n)$$
$$\text{if } f_n < 0 \text{ then } x_l = x_n$$
$$\text{else } x_h = x_n$$

Mathematically:

$$z_i \in [x_l, x_h]$$

Tough sign problem

Implementation: IEEE arithmetic

- data error in representation of coefficients of $p_1(x)$
- rounding error in evaluation of $p_1(x)$

Implementation yields floating-point bounds $[\tilde{x}_l, \tilde{x}_h]_{t=53}$

$$z_i \in [\tilde{x}_l, \tilde{x}_h]?$$

No guarantee!

$$z_9 = 8.917250249\ldots$$

$$z_9 \notin [\tilde{x}_i, \tilde{x}_j]_{53} = [8.9172245567595, 8.9172249342899]$$

$$\begin{aligned} \tilde{p}_1(\tilde{x}_i)_{t=53} &= +5.369856e7 \\ \tilde{p}_1(\tilde{x}_j)_{t=53} &= -2.626468e8 \end{aligned}$$

$$\begin{aligned} p_1(\tilde{x}_j) &= +4.87759e7 \\ \tilde{p}_1(\tilde{x}_j)_{t=53} &\in [-1.177340e9, +1.324868e9] \end{aligned}$$

Reliable

Integrated

Computational

Environment

- IEEE compliant multiprecision floats
- rational arithmetic
- complex arithmetic
- sharp interval arithmetic
- reliable graphics
- calculator, parser, precompiler, GUI

Reliability in view

Strange output unraveled

[Cuyt&Verdonk]:

- multiprecision float ($t = 122$) :

$$
\begin{aligned}
z &= -7.917111340668961361101134701524942850e36 \\
x &= +7.917111340668961361101134701524942848e36 \\
z + x &= -2.00000000000000000000000000000000000 \\
y &= -8.27396059946821368141165095479816292e{-1}
\end{aligned}
$$

- single precision interval ($t = 24$) :

$$
\begin{aligned}
z &= [-7.917116e36; -7.917105e36] \\
x &= [7.917108e36; 7.917112e36] \\
z + x &= [-7.605904e30; 6.338254e30] \\
y &= [-7.605904e30; 6.338254e30]
\end{aligned}
$$

- double precision interval ($t = 53$) :

$$
\begin{aligned}
z &= [-7.91711134066897e36; -7.91711134066895e36] \\
x &= [7.91711134066895e36; 7.91711134066896e36] \\
z + x &= [-8.26414134502188e21; 7.08354972430447e21] \\
y &= [-8.26414134502188e21; 7.08354972430447e21]
\end{aligned}
$$

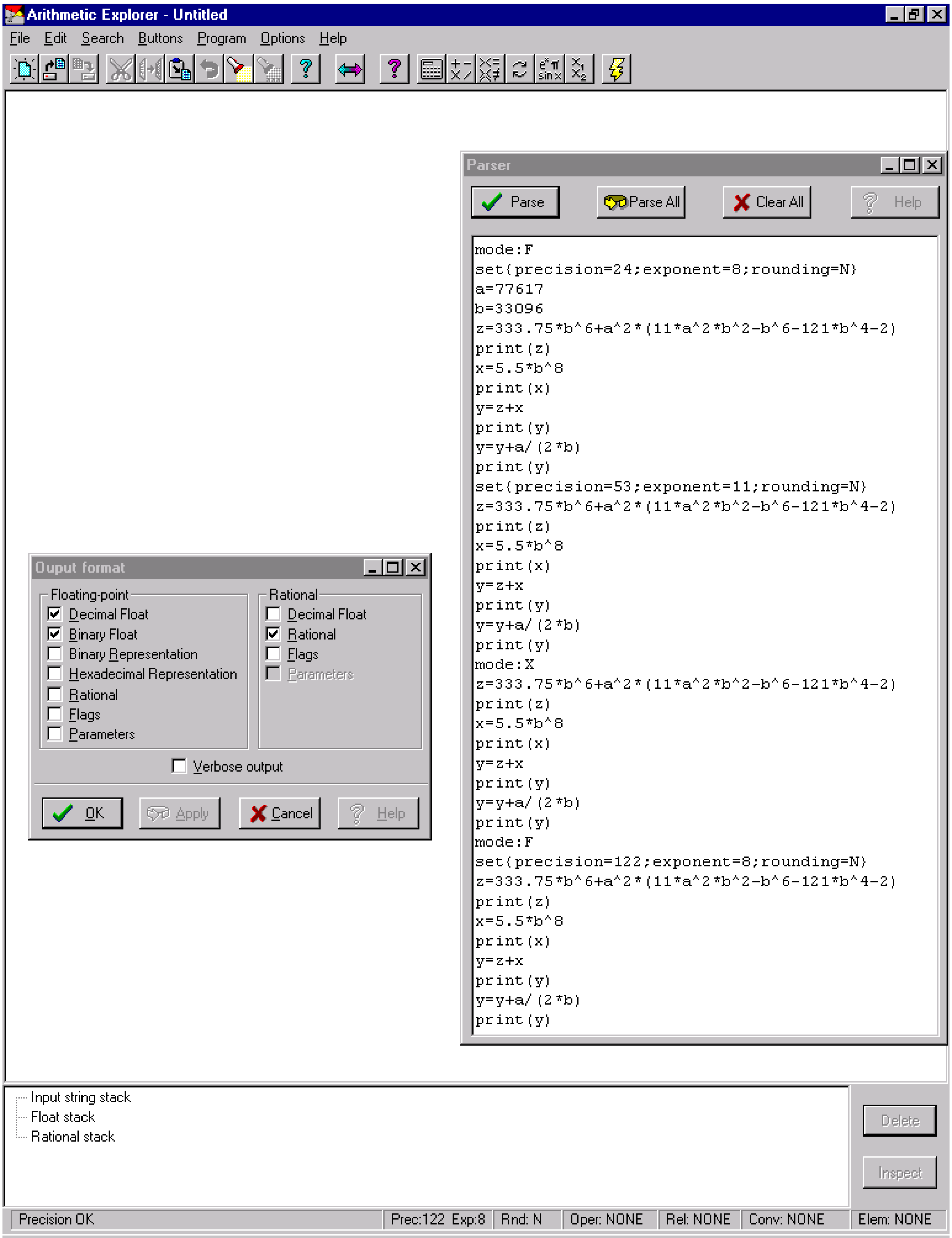
- multiprecision interval ($t = 122$) :

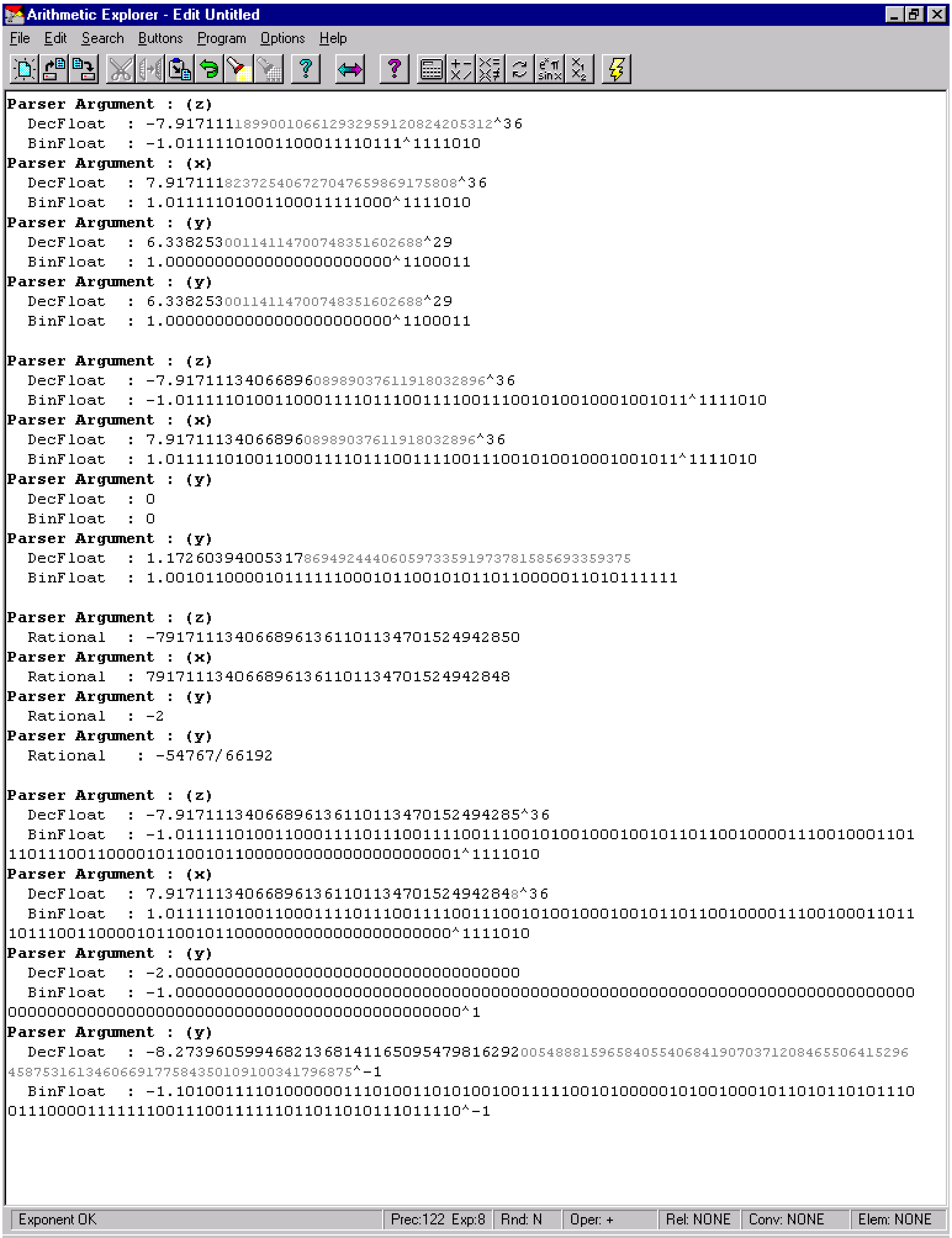
$$z = [-7.9171113406689613611011347015249428\genfrac{}{}{0pt}{}{4}{5}e36]$$

$$x = [7.9171113406689613611011347015249428\genfrac{}{}{0pt}{}{5}{4}e36]$$

$$z + x = [-2.0000000000000000000000000000000000\genfrac{}{}{0pt}{}{0}{1}]$$

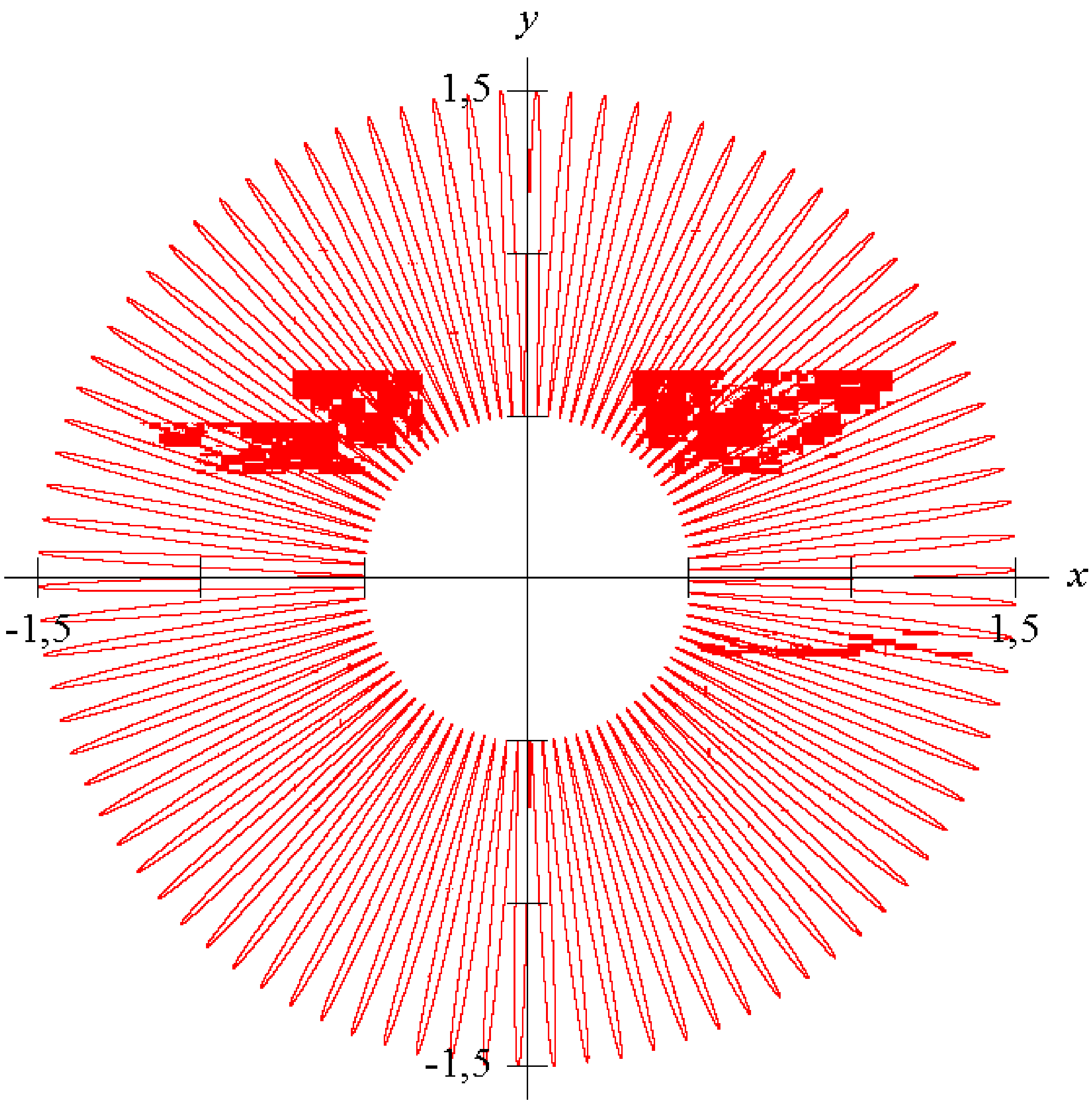
$$y = [-8.2739605994682136814116509547981629\genfrac{}{}{0pt}{}{0}{1}e{-1}]$$


Arithmetic Explorer - Untitled
File Edit Search Buttons Program Options Help
Parser
Parse
Parse All
Clear All
Help
mode:F
set{precision=24;exponent=8;rounding=N}
a=77617
b=33096
z=333.75*b^6+a^2*(11*a^2*b^2-b^6-121*b^4-2)
print(z)
x=5.5*b^8
print(x)
y=z+x
print(y)
y=y+a/(2*b)
print(y)
set{precision=53;exponent=11;rounding=N}
z=333.75*b^6+a^2*(11*a^2*b^2-b^6-121*b^4-2)
print(z)
x=5.5*b^8
print(x)
y=z+x
print(y)
y=y+a/(2*b)
print(y)
mode:X
z=333.75*b^6+a^2*(11*a^2*b^2-b^6-121*b^4-2)
print(z)
x=5.5*b^8
print(x)
y=z+x
print(y)
y=y+a/(2*b)
print(y)
mode:F
set{precision=122;exponent=8;rounding=N}
z=333.75*b^6+a^2*(11*a^2*b^2-b^6-121*b^4-2)
print(z)
x=5.5*b^8
print(x)
y=z+x
print(y)
y=y+a/(2*b)
print(y)
Ouput format
Floating-point
Decimal Float
Binary Float
Binary Representation
Hexadecimal Representation
Rational
Flags
Parameters
Rational
Decimal Float
Rational
Flags
Parameters
Verbose output
OK
Apply
Cancel
Help
Input string stack
Float stack
Rational stack
Delete
Inspect
Precision OK
Prec:122 Exp:8
Rnd: N
Oper: NONE
Rel: NONE
Conv: NONE
Elem: NONE


Arithmetic Explorer - Edit Untitled
File Edit Search Buttons Program Options Help
Parser Argument : (z)
DecFloat : -7.917111189900106612932959120824205312^36
BinFloat : -1.01111101001100011110111^1111010
Parser Argument : (x)
DecFloat : 7.917111823725406727047659869175808^36
BinFloat : 1.01111101001100011111000^1111010
Parser Argument : (y)
DecFloat : 6.3382530011411470074835160268 8^29
BinFloat : 1.00000000000000000000000^1100011
Parser Argument : (y)
DecFloat : 6.3382530011411470074835160268 8^29
BinFloat : 1.00000000000000000000000^1100011
Parser Argument : (z)
DecFloat : -7.9171113406689608989037611918032896^36
BinFloat : -1.0111110100110001111011100111100111001010010001001011^1111010
Parser Argument : (x)
DecFloat : 7.9171113406689608989037611918032896^36
BinFloat : 1.0111110100110001111011100111100111001010010001001011^1111010
Parser Argument : (y)
DecFloat : 0
BinFloat : 0
Parser Argument : (y)
DecFloat : 1.1726039400531786949244406059733591973781585693359375
BinFloat : 1.0010110000101111110001011001010110110000011010111111
Parser Argument : (z)
Rational : -7917111340668961361101134701524942850
Parser Argument : (x)
Rational : 7917111340668961361101134701524942848
Parser Argument : (y)
Rational : -2
Parser Argument : (y)
Rational : -54767/66192
Parser Argument : (z)
DecFloat : -7.91711134066896136110113470152494285^36
BinFloat : -1.011111010011000111101110011110011100101001000100101101100100001110010001101
110111001100001011001011000000000000000000000001^1111010
Parser Argument : (x)
DecFloat : 7.91711134066896136110113470152494284 8^36
BinFloat : 1.0111110100110001111011100111100111001010010001001011011001000011100100011011
10111001100001011001011000000000000000000000000^1111010
Parser Argument : (y)
DecFloat : -2.00000000000000000000000000000000000
BinFloat : -1.000000000000000000000000000000000000000000000000000000000000000000000000000
000000000000000000000000000000000000000000000000^1
Parser Argument : (y)
DecFloat : -8.27396059946821368141165095479816292005488815965840554068419070371208465506415296
45875316134606691775843501091003417968 75^-1
BinFloat : -1.101001111010000001110100110101001001111100101000001010010001011010110101110
01110000111111100111001111110110110101110111 10^-1
Exponent OK
Prec:122 Exp:8
Rnd: N
Oper: +
Rel: NONE
Conv: NONE
Elem: NONE

Reliable graphics

[Tupper]: generalized interval arithmetic



Advanced exception handling

[ARITHMOS]:

$$\begin{aligned} x_1 &= +\infty \\ x_2 &= -\infty \\ x &= x_1 + x_2 \Rightarrow x = \mathrm{NaN} \\ y &= \sqrt{-1} \quad \Rightarrow y = \mathrm{Ia}\mathbb{C}_0 \end{aligned}$$

IaY: result can only be represented in $Y \supset X$

$\rightarrow$ Ia$\mathbb{C}$, Ia$\mathbb{C}_0$, Ia$\mathbb{C}$ $\sim$Inf, Ia$\mathbb{C}_0$ $\sim$Inf

$$\forall x \in Y \cup \{0, \pm\infty\} \qquad x^0 = 1$$

$$\Downarrow$$

$$(\mathrm{IaY})^0 = 1$$

$$\begin{aligned} x^0 &= \mathrm{NaN} \\ y^0 &= 1 \end{aligned}$$

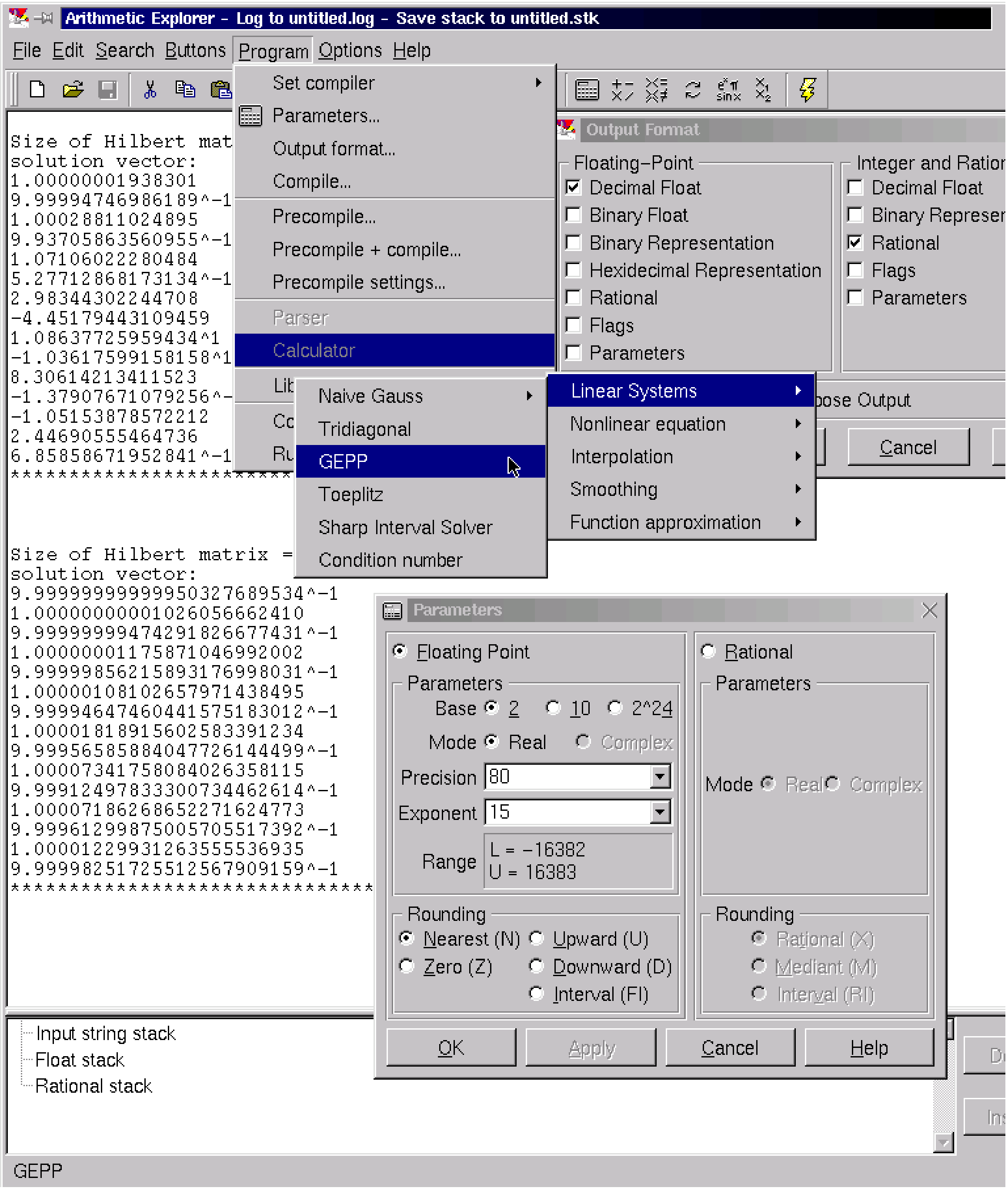
Reliable false position

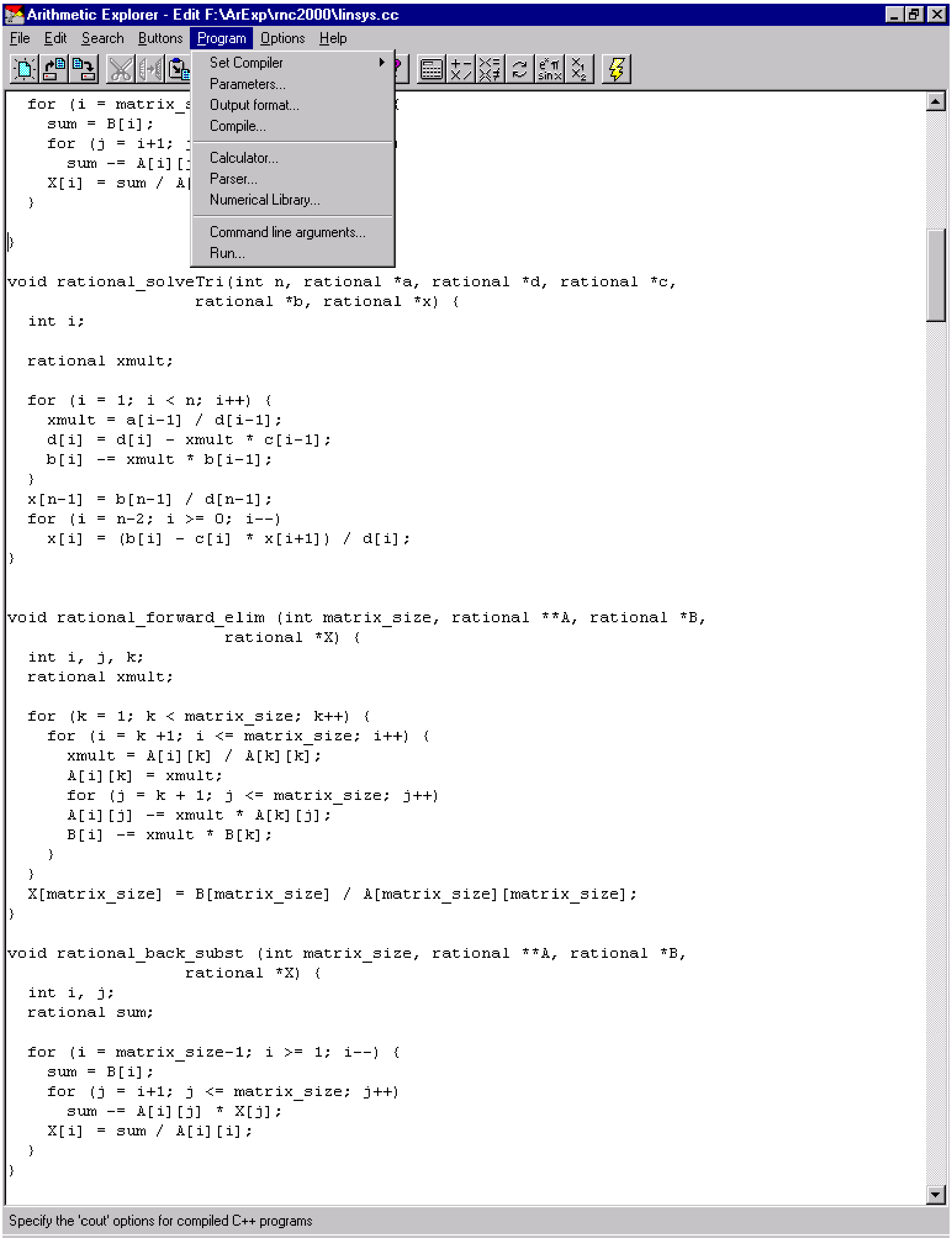
```
#include "MpIeee.h"
#include "Rational.h"
....
Rational coef[NR_OF_COEF];

MpIeee f(MpIeee xx)
{ Rational result, x(xx);

result=coef[DEGREE];
for(int i=DEGREE-1;i>=0;i--) {
   result=x*result+coef[i]; }
return result.toMpIeee(53,11);
}

void regfalsi(MpIeee x1,MpIeee x2,
              MpIeee (*f)(MpIeee x),MpIeee eps)
{ MpIeee xh, xl, fl, fh, xnew, fxnew;

...
for (int count=0;count<MAXIT;count++) {
    xnew=xh+fh*(xh-xl)/(fl-fh);
    fxnew=f(xnew);
    if (fxnew<0) {
       xl=xnew;
       fl=fxnew; }
    else {
       xh=xnew;
       fh=fxnew; }
    ... }
}
```

Reliable false position

[ARITHMOS]:

```
main()
{ MpIeee (*fptr)(MpIeee);
MpIeee startl, starth, eps;

coef[0]=Rational("2432902008176640000.0");
coef[1]=Rational("-8752948036761600000.0");
coef[2]=Rational("13803759753640704000.0");
coef[3]=Rational("-12870931245150988800.0");
coef[4]=Rational("8037811822645051776.0");
...
coef[19]=Rational("-210.0")-pow(2,-23);
coef[20]=Rational("1.0");
...
regfalsi(startl,starth,fptr,eps);
}
```


Arithmetic Explorer - Log to untitled.log - Save stack to untitled.stk
File Edit Search Buttons Program Options Help
Set compiler
Parameters...
Output format...
Compile...
Precompile...
Precompile + compile...
Precompile settings...
Parser
Calculator
Naive Gauss
Tridiagonal
GEPP
Toeplitz
Sharp Interval Solver
Condition number
Linear Systems
Nonlinear equation
Interpolation
Smoothing
Function approximation
Output Format
Floating-Point
Decimal Float
Binary Float
Binary Representation
Hexidecimal Representation
Rational
Flags
Parameters
Integer and Ratior
Decimal Float
Binary Represer
Rational
Flags
Parameters
Cancel
Size of Hilbert mat
solution vector:
1.00000001938301
9.99994746986189^-1
1.00028811024895
9.93705863560955^-1
1.07106022280484
5.27712868173134^-1
2.98344302244708
-4.45179443109459
1.08637725959434^1
-1.03617599158158^1
8.30614213411523
-1.37907671079256^-
-1.05153878572212
2.44690555464736
6.85858671952841^-1
Size of Hilbert matrix =
solution vector:
9.99999999999950327689534^-1
1.00000000001026056662410
9.99999999474291826677431^-1
1.00000001175871046992002
9.99999856215893176998031^-1
1.00000108102657971438495
9.99994647460441575183012^-1
1.00001818915602583391234
9.99956585884047726144499^-1
1.00007341758084026358115
9.99912497833300734462614^-1
1.00007186268652271624773
9.99961299875005705517392^-1
1.00001229931263555536935
9.99998251725512567909159^-1
Parameters
Floating Point
Parameters
Base 2 10 2^24
Mode Real Complex
Precision 80
Exponent 15
Range L = -16382 U = 16383
Rounding
Nearest (N) Upward (U)
Zero (Z) Downward (D)
Interval (FI)
Rational
Parameters
Mode Real Complex
Rounding
Rational (X)
Mediant (M)
Interval (RI)
OK Apply Cancel Help
Input string stack
Float stack
Rational stack
GEPP


Arithmetic Explorer - Edit F:\ArExp\rnc2000\linsys.cc
File Edit Search Buttons Program Options Help
Set Compiler
Parameters...
Output format...
Compile...
Calculator...
Parser...
Numerical Library...
Command line arguments...
Run...
for (i = matrix_s
sum = B[i];
for (j = i+1;
sum -= A[i][
X[i] = sum / A[
}
}
void rational_solveTri(int n, rational *a, rational *d, rational *c,
rational *b, rational *x) {
int i;
rational xmult;
for (i = 1; i < n; i++) {
xmult = a[i-1] / d[i-1];
d[i] = d[i] - xmult * c[i-1];
b[i] -= xmult * b[i-1];
}
x[n-1] = b[n-1] / d[n-1];
for (i = n-2; i >= 0; i--)
x[i] = (b[i] - c[i] * x[i+1]) / d[i];
}
void rational_forward_elim (int matrix_size, rational **A, rational *B,
rational *X) {
int i, j, k;
rational xmult;
for (k = 1; k < matrix_size; k++) {
for (i = k +1; i <= matrix_size; i++) {
xmult = A[i][k] / A[k][k];
A[i][k] = xmult;
for (j = k + 1; j <= matrix_size; j++)
A[i][j] -= xmult * A[k][j];
B[i] -= xmult * B[k];
}
}
X[matrix_size] = B[matrix_size] / A[matrix_size][matrix_size];
}
void rational_back_subst (int matrix_size, rational **A, rational *B,
rational *X) {
int i, j;
rational sum;
for (i = matrix_size-1; i >= 1; i--) {
sum = B[i];
for (j = i+1; j <= matrix_size; j++)
sum -= A[i][j] * X[j];
X[i] = sum / A[i][i];
}
}
Specify the 'cout' options for compiled C++ programs

References

[1] A. Cuyt and B. Verdonk. A remarkable example of catastrophic cancellation unraveled. Submitted for publication, `ftp://wins.uia.ac.be/pub/CANT/Arithmos/catacanc.pdf`, 1999.

[2] I. Geulig and W. Krämer. Intervallrechnung in Maple - die erweiterung `intpakx` zum paket `intpak` der Share-library. Preprint nr. 99/2, Universität Karlsruhe, 1999.

[3] Nick Higham. Can you "count" on your computer. Public lecture for Science Week, `ftp://ftp.ma.man.ac.uk/pub/higham/talks/count98.ps.gz`, 1998.

[4] W. Kahan. Lectore Notes on the status of IEEE standard 754 for binary floating-point arithmetic. `http://www.cs.berkeley.edu/wkahan/ieee754status/ieee754.ps`, May 1996.

[5] S.M. Rump. Algorithms for verified inclusions – theory and practice. In R.E. Moore, editor, *Reliability in Computing*, pages 109–126. Academic Press, 1988.

[6] J.A. Tupper. Graphing equations with generalized interval arithmetic. thesis, Toronto, 1996.