MPFR - A library for arbitrary precision correct rounding of mathematical functions

(www.mpfr.org)

David Daney, Guillaume Hanrot, Vincent Lefèvre, Fabrice Rouillier, Paul Zimmermann

Spaces Project, INRIA Lorraine/LORIA, Nancy, France

(with contributions from Sylvie Boldo, Mathieu Dutour, Emmanuel Jeandel, Nathalie Revol)

Introduction

Need of a tool to perform **exact rounding** in **arbitrary** precision...

Q1. what gives $\sin(10^{22})$ with a mantissa of 100 bits?

Q2. what is the smallest precision for which we can decide the sign of Rump's "polynomial" for $x = 77617, y = 33096$?

$$R(x,y) = \frac{1335y^6}{4} + x^2(11x^2y^2 - y^6 - 121y^4 - 2) + \frac{11y^8}{2} + \frac{x}{2y}.$$

Q3. what is the worst case for atan in $[0.5, 1]$ in single precision for directed rounding?

... that is **correct, portable and efficient**.

```
#include "gmp.h"
#include "mpfr.h"

main(int argc, char *argv[])
{
  unsigned int prec = atoi(argv[1]);
  mpfr_t x, y;

  mpfr_init2 (x, prec);
  mpfr_init2 (y, prec);
  mpfr_set_d (x, 1e22, GMP_RNDN);
  mpfr_sin (y, x, GMP_RNDN);
  mpfr_out_str (stdout, 2, 0, y, GMP_RNDN); printf(" (");
  mpfr_out_str (stdout, 10, 0, y, GMP_RNDN); printf(")\n");
}
```

MPFR inside

- $+$ and $-$ based on `mpn_add`, `mpn_sub` and `mpn_[lr]shift`: $O(n)$
- $\times$ based on `mpn_mul`: $M(n)$
- $/$ based on `mpn_divrem`: $O(M(n))$
- $\sqrt{\ }$ based on `mpn_sqrtrem`: $O(M(n))$
- agm and log in $O(M(n) \log n)$
- exp has complexity $O(M(n) \log^2 n)$
- $\sin, \cos$ in $O(M(n) \log^2 n)$

Current state and future plans

mpfr-2001 implements input/output ($2 \leq \beta \leq 36$), conversions from/to integer or double, basic operations ($+, -, \times, /, \sqrt{}$), comparison, $\exp, \log, \sin, \cos$, agm, the constants π and $\log 2$.

Next version (planned for end 2001) will implement:

- all additional functions from the ISO C 9X standard (acos, asin, atan, atan2, tan, acosh, asinh, atanh, cosh, sinh, tanh, log10, expm1, log1p, exp2, log2, pow, hypot, cbrt, erf, erfc, gamma, lgamma),
- (global) exponent range control,
- better overflow/underflow handling.

Related packages

MPFR++ (N. Revol): C++ interface

MPFI (N. Revol, F. Rouillier): interval arithmetic

Arithmos (Univ. of Cant): integrated environment with user-chosen internal base ($2 \leq \beta \leq 2^{24}$)

iRRAM (N. Müller): RealRAM implementation in C++

Quad-double library of Hida, Li and Bailey

Answer to Q1

```
[zimmerma@ecrouves tests]$ ./ng 43
-1.111111001010111001000001110100011111101111e-1 (-9.96758488121e-1)

[zimmerma@ecrouves tests]$ ./ng 53
-1.1011010001010011101010110111011010111111001110010111e-1
(-8.52200849767189e-1)

[zimmerma@ecrouves tests]$ ./ng 100
-1.10110100010100111010101101110110101111110011100101110000111110100001\
0100110111100100000111011100101 0e-1 (-8.5220084976718880177270589375 3e-1)
```

Answer to Q2

```
[zimmerma@ecrouves tests]$ ./rump 1
-2.e39 <= R <= 6.e39

[zimmerma@ecrouves tests]$ ./rump 121
-2.827396059946821368141165095479816 30 <= R <=
1.172603940053178631858834904520183 71

[zimmerma@ecrouves tests]$ ./rump 122
-8.27396059946821368141165095479816293e-1 <= R <=
-8.27396059946821368141165095479816291e-1
```

Answer to Q3

```
% ./worst atan 0.5 1.0 24
x=1.10010011101101010000110e-1
y=1.01010101110110101001110111111111111111111111111111010e-1 [27]

% ./worst exp 1.0 2.0 24
x=1.11111100000001011101110
y=1.11010001100111000000110111111111111111111111111101001e2 [25]

% ./worst log 2.0 3.0 24
x=1.01001001010001000101101e1
y=1.11100011110000011110011000000000000000000000011111000e-1 [22]

% ./worst sin 1.0 2.0 24
x=1.01001111000001100101010
y=1.11101110100000110110101111111111111111111111111111110e-1 [29]
```