

The MPC library – a quick introduction

Andreas Enge

Project-team LFANT
INRIA Bordeaux–Sud-Ouest
andreas.enge@inria.fr
<http://www.math.u-bordeaux1.fr/~enge>

GNU Hackers Meeting, Paris, 28 August 2011



The MPC library – a quick introduction

1 Complex numbers

2 MPC

Real floating point numbers

- (Binary) precision p
- $x = m2^e$
 - ▶ $\frac{1}{2} \leq |m| < 1$ with p bits: $m = 0.\underbrace{1*\dots*}_p$ mantissa/significand
 - ▶ e exponent

Cartesian coordinates

- $z = x + iy$ with $x, y \in \mathbb{R}$
- **Addition:** trivial

$$1A_{\mathbb{C}} = 2A_{\mathbb{R}}$$

- **Multiplication:** so, so

$$(x_1 + iy_1)(x_2 + iy_2) = (x_1x_2 - y_1y_2) + i(x_1y_2 + x_2y_1)$$

$$1M_{\mathbb{C}} = 4M_{\mathbb{R}} + 2A_{\mathbb{R}}$$

- **Standard C99 Annex G (informative)**

Polar coordinates

- $z = \rho e^{it}$ or $z = \rho e^{2\pi it}$
- **Multiplication:** trivial

$$1M_{\mathbb{C}} = 1M_{\mathbb{R}} + 1A_{\mathbb{R}}$$

- **Addition:** disastrous

$1A_{\mathbb{C}} =$ several real (inverse) trigonometric functions

$$e^{it} = \cos t + i \sin t$$

The MPC library – a quick introduction

1 Complex numbers

2 MPC

MPC in one slide

- **URL:** <http://mpc.multiprecision.org/>
- **Authors:** AE, Paul Zimmermann, Philippe Théveny, Mickaël Gastineau
- **Institutions:** INRIA LFANT and CAMEL, CNRS
- **License:** LGPL 2.1+ (LGPL 3+, GFDL for version 1.0)
- **Current version:** 0.9 "Epilobium montanum"
- **Language:** C99
- **Dependencies:** gmp \geq 4.3.2, mpfr \geq 2.4.2
- **Installation**
 - ▶ `./configure; make; make install`
 - ▶ `./configure --prefix=/usr/local/mpc-0.9 \\
--with-gmp=/usr/local/gmp-5.0.2 \\
--with-mpfr=/usr/local/mpfr-3.0.1
make; make install`
- **Usage**
 - ▶ `#include "mpc.h"`
 - ▶ `gcc ... -lmpc -lmpfr -lgmp ...`



Milestones

- 2002 Version 0.1, AE and PZ
- 2005 Inclusion into Magma
- 2007 Inclusion into Trips, Windows support (MG)
- 2008 Debian packages for squeeze (Laurent Fousse)
`libmpc2`, `libmpc-dev`, `mpclib` (src)
- 2007–2009 PT
Trigonometric functions
Testing framework
- 2009 Summer school on `mpfr` and `mpc` in Nancy
- 2010 Mandatory for GCC 4.5
- 2011 First joint `mpfr` and `mpc` developers' meeting
- 2011 GNU project, license change
- 2011 (?) Copyright assignment to FSF

Language bindings

- Perl: `Math::MPC`
- R6RS Scheme: `Nausicaa`
- Eiffel
- Sage (written in Python)

```
typedef struct {  
    mpfr_t re;  
    mpfr_t im;  
} __mpc_struct;  
typedef __mpc_struct mpc_t[1];
```

- Each coordinate carries its own precision.
- Coordinate access
 - ▶ `mpc_realref (z)`
 - ▶ `mpc_imagref (z)`

- Initialisation

- ▶ `mpc_init2 (z, prec)`
- ▶ `mpc_init3 (z, prec_re, prec_im)`

- Assignment

- ▶ `mpc_set`
- ▶ `mpc_set_fr, mpc_set_ui, ...`
- ▶ `mpc_set_fr_fr, mpc_set_ui_ui, ...`

- Computation

- Freeing

- ▶ `mpc_clear (z)`

- Rounding modes
 - ▶ Separate for each coordinate
 - ▶ $\text{MPC_RNDNU} = \text{MPFR_RNDN} + i \text{MPFR_RNDU}$
- Correct rounding of each coordinate for atomic operation
- Nonary return value
 - ▶ Result exact/rounded down/rounded up by coordinate
 - ▶ $\text{ret}==0$ iff both coordinates exact
 - ▶ $\text{MPC_INEX_RE}(\text{ret}) \in \{0, -1, 1\}$
 - ▶ $\text{MPC_INEX_IM}(\text{ret}) \in \{0, -1, 1\}$
- Comparison
 - ▶ $\text{mpc_cmp}(z1, z2)$
 - ▶ Returns nonary value

All functions in the C99 standard since version 0.9

- Field arithmetic

- ▶ `int mpc_add (rop, op1, op2, rnd)`
- ▶ `mpc_sub`, `mpc_neg`, `mpc_mul`, `mpc_div`, `mpc_sqr`, `mpc_fma`
- ▶ Some mixed operations: `mpc_ui_ui_sub`, `mpc_mul_fr`,
`mpc_mul_2exp`, `mpc_div_2exp`, ...
- ▶ `mpc_mul_i`

- \mathbb{C} as algebraic extension of \mathbb{R}

- ▶ `mpc_conj`: $x - iy$
- ▶ `mpc_norm`: $x^2 + y^2$
- ▶ `mpc_abs`: $\sqrt{x^2 + y^2}$

- Powers and logarithms
 - ▶ `mpc_sqrt`, `mpc_exp`, `mpc_log`
 - ▶ `mpc_pow`, `mpc_pow_d`, ...
- Trigonometric and inverse trigonometric functions
 - ▶ `mpc_sin`, `mpc_cos`, `mpc_tan`, `mpc_sin_cos`,
`mpc_sinh`, `mpc_cosh`, `mpc_tanh`
 - ▶ `mpc_asin`, `mpc_acos`, `mpc_atan`,
`mpc_asinh`, `mpc_acosh`, `mpc_atanh`

- **Format:**

3.1415926 (1.25e+7 +.17) (@nan@ 2) (-0 -7) -@inf@

- **Strings**

- ▶ **Input**

- ★ `mpc_strtoc`: reads a complex number from a string, advances the pointer

- ★ `mpc_set_str`: reads a complete string

- ▶ **Sortie:**

- ★ `mpc_get_str`: returns a string

- ★ optionally: allocates the string, free with `mpc_free_str`

- **Streams**

- ▶ **Input:** `mpc_inp_str`

- ▶ **Output:** `mpc_out_str`

Example: $\operatorname{asin}(2i)$

```
#include <stdio.h>
#include <mpc.h>
int main (void) {
    mpc_t z;
    int inex;
    mpc_init2 (z, 123);
    mpc_set_ui_ui (z, 0, 2, MPC_RNDNN);

    inex = mpc_asin (z, z, MPC_RNDNN);
    mpc_out_str (stdout, 10, 0, z, MPC_RNDNN);
    printf ("\n%i %i\n", MPC_INEX_RE (inex), MPC_INEX_IM (inex));

    mpc_clear (z);
}
```

(0 1.44363547517881034249327674027310526938)

0 -1

Development difficulties

- Most bug reports on **exotic platforms** (\neq GNU-Linux):
 - ▶ Solaris
 - ▶ Cygwin, Mingw
 - ▶ Apple Darwin
 - ▶ Windows
- Self-inflicted pain
 - ▶ `-std=c99 -pedantic -Wno-long-long -Wall -Wextra -Werror -Wdeclaration-after-statement ...`
only in the svn, not for releases
- Partial solutions
 - ▶ Friendly and helpful mailing list participants (Windows)
 - ▶ Debian packages (exotic hardware)
 - ▶ INRIA Pipol platform (Solaris, Apple)
nightly builds
essentially dead (outdated systems)
 - ▶ **Continuous integration with hydra** (GNU-Linux x86, Cygwin, Apple):
<http://hydra.nixos.org/jobset/gnu/mpc-trunk/>
 - ▶ GCC compile farm? — **no Solaris, no Apple**

Contribute

- [URL: http://mpc.multiprecision.org/](http://mpc.multiprecision.org/)
Menu item “Development”
- Hosted on [INRIA Gforge](#)
<http://gforge.inria.fr/projects/mpc/>
 - ▶ [Discussion list \(bug reports\)](#):
mpc-discuss@lists.gforge.inria.fr
 - ▶ [Bug tracker](#)
 - ▶ (Anonymous) [subversion access](#)