

*Inria*

PariTwine

Atelier PARI-GP 2020, Grenoble

- Use C code for evaluating two-dimensional  $\vartheta$ -functions from inside GP.

- Use C code for evaluating two-dimensional  $\vartheta$ -functions from inside GP.
- Use low-level C libraries from inside GP:
  - > GNU multiprecision: GMP, MPFR, MPC
  - > ARB
  - > CM (not yet done), CMH
  - > FPLLL (not yet done)
  - > ...

- new name
- autotools based installation
- automatic detection of wrapped libraries
- texinfo (html, pdf, . . . ) documentation
- automatic creation of the .gp file

<http://paritwine.multiprecision.org/>

Version 0.1 of 2019.

- `./configure`  
`make`  
`make check`  
`make install`  
`make install-html`  
`make install-pdf`  
`...`

<http://paritwine.multiprecision.org/>

Version 0.1 of 2019.

- `./configure`  
`make`  
`make check`  
`make install`  
`make install-html`  
`make install-pdf`  
...
- `export LOC=$HOME/local`  
`./configure --prefix=$LOC/pt --with-cmh=$LOC/cmh`

<http://paritwine.multiprecision.org/>

Version 0.1 of 2019.

- `./configure`  
`make`  
`make check`  
`make install`  
`make install-html`  
`make install-pdf`  
...
- `export LOC=$HOME/local`  
`./configure --prefix=$LOC/pt --with-cmh=$LOC/cmh`
- `gp /usr/local/share/paritwine/paritwine.gp`

- Built-in

```
z = 1+I
```

```
sin (z)^2 + cos (z)^2
```

- GNU MPC

```
s = mpc_sin (z)
```

```
c = mpc_cos (z)
```

```
mpc_add (mpc_mul (s, s), mpc_mul (c, c))
```

- ARB

```
s = acb_sin (z)
```

```
c = acb_cos (z)
```

```
acb_add (acb_mul (s, s), acb_mul (c, c))
```



- Real values

```
\pb 1000
```

```
zeta (2)
```

```
mpfr_zeta (2)
```

```
acb_zeta (2)
```

```
\pb 128
```

- Complex values on the critical line

```
#
```

```
zeta (1/2 + 10^5*I)
```

```
zeta (1/2 + 10^5*I)
```

```
acb_zeta (1/2 + 10^5*I)
```

```
acb_zeta (1/2 + 10^15*I)
```

- Dedekind  $\eta$

$z = 1/4 + 2*I$

`eta (z, 1)`

`acb_modular_eta (z)`

`acb_modular_eta (z)[1]`

- Weierstraß  $\wp$ -function

$\tau = 1+I$

$z = (1+3*\tau)/5$

`ellwp ([1,tau], z)`

`acb_elliptic_p (z, tau)`

- Jacobi  $\vartheta$ -functions

```
acb_modular_theta (z, tau)
```

```
apply (x -> x[1], acb_modular_theta (z, tau))
```

```
apply (x -> x[1], acb_modular_theta (0, tau))
```

- Two-dimensional  $\vartheta$ -constants

```
tau = [I+1/2, 1/4; 1/4, I+1]
```

```
cmh_4theta (tau)
```

```
cmh_10theta2 (tau)
```

```
apply (x -> x^2, cmh_4theta (tau))
```

- Igusa–Streng invariants

```
cmh_I2I4I6I10 (tau)
```

All this is also usable from within your C code:

- conversion functions between the basic types
  - > GEN t\_INT and mpz\_t
  - > GEN t\_REAL and mpfr\_t
  - > GEN t\_COMPLEX and mpc\_t, acb\_t
  - > ...
- function wrappers
  - > pari\_mpc\_cos calls mpc\_cos with GEN types
  - > ...

```
firefox  
/usr/local/share/doc/paritwine/paritwine.html/  
index.html
```

- Ask us to wrap your favourite missing function.
- Wrap your favourite library.