

**NAME**

complex – basics of complex mathematics

**SYNOPSIS**

```
#include <complex.h>
```

**DESCRIPTION**

Complex numbers are numbers of the form  $z = a + b*i$ , where  $a$  and  $b$  are real numbers and  $i = \sqrt{-1}$ , so that  $i*i = -1$ .

There are other ways to represent that number. The pair  $(a,b)$  of real numbers may be viewed as a point in the plane, given by X- and Y-coordinates. This same point may also be described by giving the pair of real numbers  $(r,\phi)$ , where  $r$  is the distance to the origin  $O$ , and  $\phi$  the angle between the X-axis and the line  $Oz$ . Now  $z = r*\exp(i*\phi) = r*(\cos(\phi) + i*\sin(\phi))$ .

The basic operations are defined on  $z = a + b*i$  and  $w = c + d*i$  as:

**addition:**  $z + w = (a + c) + (b + d)*i$

**multiplication:**  $z * w = (a*c - b*d) + (a*d + b*c)*i$

**division:**  $z / w = ((a*c + b*d)/(c*c + d*d)) + ((b*c - a*d)/(c*c + d*d))*i$

Nearly all math function have a complex counterpart but there are some complex-only functions.

**EXAMPLE**

Your C-compiler can work with complex numbers if it supports the C99 standard. Link with `-lm`. The imaginary unit is represented by `I`.

```
/* check that exp(i * pi) == -1 */
#include <math.h>          /* for atan */
#include <stdio.h>
#include <complex.h>

int
main(void)
{
    double pi = 4 * atan(1.0);
    double complex z = cexp(I * pi);
    printf("%f + %f * i\n", creal(z), cimag(z));
}
```

**SEE ALSO**

**cabs(3), acos(3), acosh(3), carg(3), casin(3), casinh(3), catan(3), catanh(3), ccos(3), ccosh(3), cerf(3), cexp(3), cexp2(3), cimag(3), clog(3), clog10(3), clog2(3), conj(3), cpow(3), cproj(3), creal(3), csin(3), csinh(3), csqrt(3), ctan(3), ctanh(3)**

**COLOPHON**

This page is part of release 5.02 of the Linux *man-pages* project. A description of the project, information about reporting bugs, and the latest version of this page, can be found at <https://www.kernel.org/doc/man-pages/>.