The `ePiX` Complex Numbers Package

Andrew D. Hwang
Department of Math and CS
College of the Holy Cross

Version 1.2, August, 2008

This manual describes an add-on package for `ePiX` that provides complex arithmetic and a few complex variants of standard `C`'s math functions.

1 Complex Numbers

A *complex number* may be viewed as a formal expression $z = x + iy$ with x and y real numbers (the *real* and *imaginary parts* of z, respectively) and i a symbol satisfying $i^2 = -1$. Rigorous constructions of the complex numbers merely formalize these properties.

Complex Arithmetic

Two complex numbers are added in the obvious way, and multiplied by distributing the product:

$$\begin{aligned}(u + iv) + (x + iy) &= (u + x) + i(v + y),\\ (u + iv)(x + iy) &= (ux - vy) + i(uy + vx).\end{aligned}$$

The (*complex*) *conjugate* of $z = x + iy$ is $\bar{z} = x - iy$. The conjugation map preserves arithmetic operations; the two numbers i and $-i$ are in no way algebraically distinguishable.

For every complex number $z = x+iy$, the product $z\bar{z} = x^2+y^2 = |z|^2$ is real; thus

$$\frac{1}{z} = \frac{\bar{z}}{z\bar{z}} = \frac{\bar{z}}{|z|^2} = \frac{x - iy}{x^2 + y^2}$$

so long as $z \neq 0$. To divide a complex number by a non-zero complex number, multiply the numerator by the reciprocal of the denominator.

A complex number has a *polar form*: $z = r\cos\theta + ir\sin\theta$. The real numbers r and θ are the *norm* (or *modulus*) and *argument* of z, respectively. By de Moivre's theorem, the polar form of a complex number may also be written $z = re^{i\theta}$. The argument θ of a non-zero complex number is well-defined only up to an additive constant $2\pi ki$, k an integer.

Polar form elucidates the geometry of multiplication. To multiply $z_1 = r_1 e^{i\theta_1}$ and $z_2 = r_2 e^{i\theta_2}$, multiply the norms and add the arguments:

$$z_1 z_2 = (r_1 e^{i\theta_1})(r_2 e^{i\theta_2}) = (r_1 r_2) e^{i(\theta_1+\theta_2)}.$$

The Exponential and Logarithm

If $w = u+iv$ and $z = x+iy$, then $\exp w = e^u e^{iv} = e^u \cos v + ie^u \sin v = z$ if and only if

$$x = e^u \cos v, \qquad y = e^u \sin v.$$

Aside from the ambiguity of the argument mentioned above, these equations may be solved for u and v in terms of x and y:

$$u = \log|z| = \tfrac{1}{2}\log(x^2 + y^2), \qquad v = \operatorname{atan2}(y, x).$$

The `C` function atan2 is discontinuous along the negative real axis: Its value jumps from $-\pi$ just below the axis to π just above. The *cut plane* is the complex plane with the origin and negative real axis removed. Since atan2 is continuous on the cut plane, the function $w = \log z$ is continuous on the cut plane, as well.

Mathematically, the equation $w = \log z$ has infinitely many values, any two differing by an integer multiple of $2\pi i$. A continuous choice of w on the cut z plane is called a *branch* of log; the choice taking real values along the positive real is the *principle branch*. The imaginary part of the principle branch is the standard `C` function atan2.

The branches of log are joined like successive levels of a spiral parking garage. Walking once in a circle around the origin in the z plane advances w along a path whose endpoints differ by $2\pi i$.

Roots

Let $w = \rho e^{i\varphi}$ and $z = re^{i\theta}$ be non-zero complex numbers, and suppose $w^n = z$ for some integer n. By observations above,

$$r = \rho^n \quad \text{and} \quad \theta + 2\pi ki = n\varphi \quad \text{for some integer } k.$$

In other words, $\rho = \sqrt[n]{r}$ and $\varphi = \frac{\theta}{n} + 2\pi\frac{k}{n}i$ for some integer k; there are exactly n distinct values of w, corresponding to $k = 0, \ldots, n-1$. These numbers are the nth *roots* of z. Geometrically, the nth roots of z lie at the vertices of a regular n-gon centered at the origin.

A continuous choice of w satisfying $w^n = z$ on the cut z plane is a *branch* of the nth root function. Just as for the logarithm, there is a unique branch of the nth root taking positive real values along the positive real axis in the z plane, the *principle branch*. There are exactly n branches of the nth root, indexed by $k = 0, \ldots, n-1$ and joined *cyclically*. Walking once around the origin in the z plane advances $w = z^{1/n}$ along an arc spanning one nth of a full turn.

Trig Functions

The basic circular and hyperbolic trig functions are most easily defined in terms of the exponential function:

$$\begin{aligned} \cos w &= \frac{e^{iw} + e^{-iw}}{2}, & \cosh w &= \frac{e^{w} + e^{-w}}{2}, \\ \sin w &= \frac{e^{iw} - e^{-iw}}{2i}, & \sinh w &= \frac{e^{w} - e^{-w}}{2}. \end{aligned}$$

Clearly, $\cosh iw = \cos w$ and $\sinh iw = i \sin w$. A bit of work shows the familiar identities

$$\begin{aligned} \cos(a + b) &= \cos a \cos b - \sin a \sin b \\ \sin(a + b) &= \cos a \sin b + \sin a \cos b \end{aligned}$$

hold for all complex a and b. For example,

$$\begin{aligned} \cos(x + iy) &= \cos x \cosh y - i \sin x \sinh y, \\ \sin(x + iy) &= \sin x \cosh y + i \cos x \sinh y. \end{aligned}$$

The other trig functions are defined in terms of cos and sin just as over the reals: $\tan = \sin / \cos$, $\operatorname{sech} = 1/\cosh$, etc.

2 ePiX Functions

The functions in this section are used by including the header files `Complex.h` and `adapter.h` in an `ePiX` source file. The latter contains the "factory function" `Pt`, which accepts a `Complex` argument $x+iy$ and returns the `P` object $(x, y, 0)$. Please consult the sample files distributed with the complex number library for examples. The file must normally be compiled with appropriate compiler flags, as in

```
epix -I. -L. -lepixnumber <file>.xp
```

from the directory containing the library file `libepixnumber.a`.

Complex Arithmetic

The `ePiX` complex number library supplies data structures and functions implementing the mathematical concepts described above. The constructor `Complex(double, double);` creates a complex number object of specified real and imaginary parts; these default to zero. The class also provides arithmetic operators (including increment operators, `+=`, etc.) and functions to conjugate a number or return its components:

```
Complex z1(0,1), z2(1,2); // i and 1+2i
z2.conj();                //  z2 = 1-2i
z1 *= z2;                 //  z1 = -2+i
double x(z1.re()), y(z1.im());
Complex z3(z1/z2);
```

Non-member functions are provided for powers and roots, the exponential and logarithm, and basic trig functions. Generally, these functions are named the same as their real counterparts with a "C" appended: `expC`, `powC`, `SinC`, etc. The hyperbolic trig functions are sensitive to angle units (unlike their real counterparts), so their names are also capitalized: `CoshC`, `CschC`, etc.

The square root and the logarithm function accept an optional integer argument specifying the branch. The nth root function accepts a second mandatory argument, specifying the order:

```
logC(Complex arg, int branch = 0);
sqrtC(Complex arg, int branch = 0);
rootC(Complex arg, int order, int branch = 0);
```

For `sqrtC`, only the parity of the branch (even or odd) is significant. For `rootC`, the order must be a non-zero integer, and branch may be an arbitrary integer, though only the residue class modulo order is significant.