

Complex Numbers in Java

John Brophy

Visual Numerics

www.vni.com

Complex Arithmetic

- ◆ Class `com.imsl.math.Complex`.
- ◆ Double precision only.
- ◆ Complex objects are immutable.

Complex objects are immutable

- ◆ Complex objects have reference semantics, not value semantics
 - Difference between value and reference semantics is minimized for immutable objects
- ◆ Normally use `x.equals(y)` instead of `x == y`
 - Same as for String
 - `==` means same instance, `equals` mean equal value

If Complex were mutable

Initial

double x
1.2

double y
3.4

After

y = x;
x = 5.6;

double x
5.6

double y
1.2

Initial

Complex x
1+2i

Complex y
3+4i

After

y = x;
x.re = 5;

Complex x
5+2i

Complex y
y.re = 5

Complex Arithmetic

◆ $x \text{ op } y$ where op is plus, minus, times, over

» `static Complex op(Complex x, Complex y)`

» `static Complex op(Complex x, double y)`

» `static Complex op(double x, Complex y)`

◆ $\text{this op } y$

» `Complex op(Complex y)`

» `Complex op(double y)`

◆ $x \text{ op this}$

» `Complex opReverse(double x)`

Complex Functions

- ◆ Implements

- sqrt, exp, log, sin, cos, tan, asin, acos, atan, sinh, cosh, tanh, asinh, acosh, atanh, pow

- ◆ Numerical properties are as in the proposed C9X standard.

- For example, sqrt has a branch cut along the negative real axis

- $\text{sqrt}(-\infty+iy) = +0+i\infty$, for all finite, positive y .

- $\text{sqrt}(-\infty-iy) = +0-i\infty$, for all finite, positive y .

Zeta Compiler

- ◆ A modified version of the IBM Jikes compiler
 - Open source Java compiler from IBM
 - Written in C++
 - Has been ported to a number of platforms

Zeta compiler enhancements

- ◆ Allows infix operations on Complex objects
 - supports + - * / += -= *= /=
- ◆ Infix expression `x += y` is compiled as if written `x = Complex.plus(x,y)`.
- ◆ Equivalent to
 - `x = new Complex(x.re+y.re, x.im+y.im)`

Example

```
import com.imsl.math.Complex;

public class Quadratic
{
    static public void main(String argv[])
    {
        Complex a = new Complex(2,3);
        Complex b = new Complex(4,5);
        Complex c = new Complex(5,6);
        Complex x1 = solve1(a, b, c);
        System.out.println("Root" + x1);
        System.out.println("Check");
        System.out.println(a*x1*x1+b*x1+c);
    }

    static Complex solve1(Complex a, Complex b, Complex c)
    {
        Complex d = Complex.sqrt(b*b-4.0*a*c);
        return (-b+d)/(2.0*a);
    }
}
```

Obtaining Zeta

- ◆ Download from www.vni.com/corner/garage/grande/index.html
- ◆ Includes source to Java classes:
 - Complex class
 - Sfun (hyperbolic trig functions, error functions)
 - JMath (pure Java version of `java.lang.Math`)
- ◆ Binary for Windows 95/98/NT version of Zeta compiler.