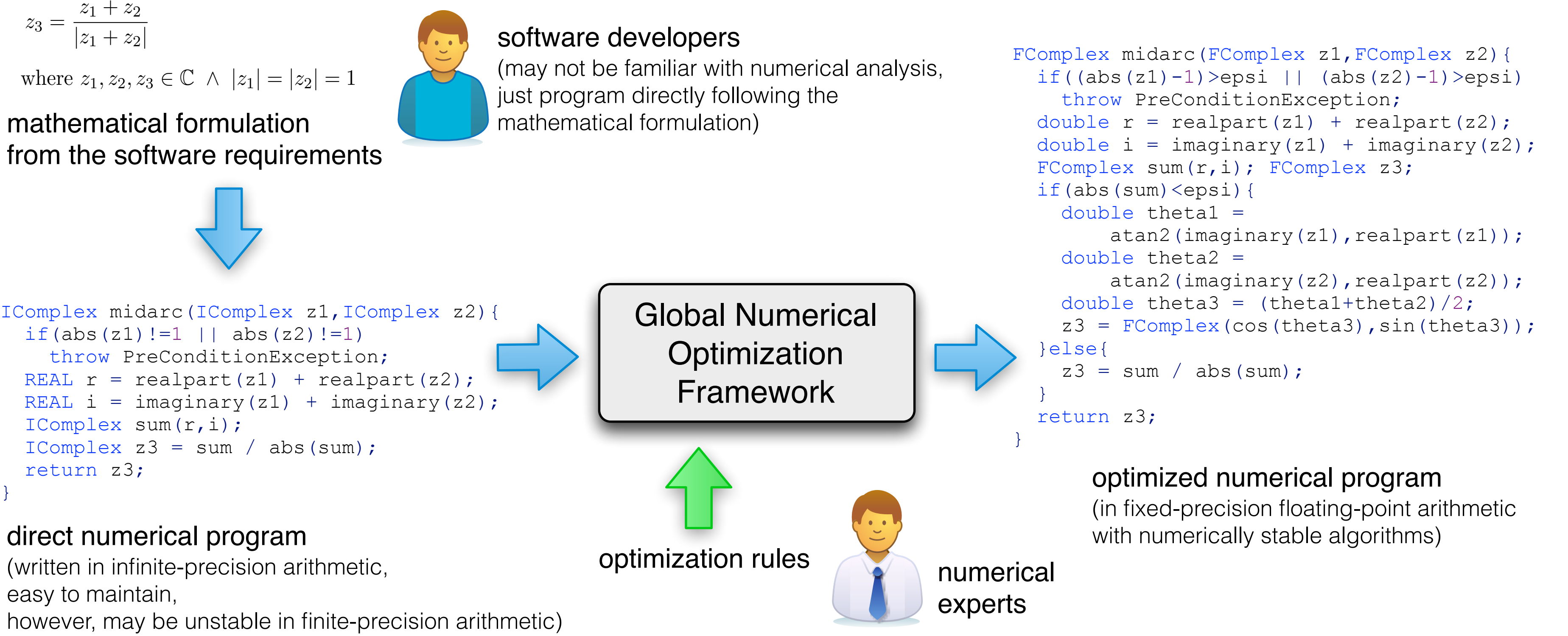
$z_3 = \frac{z_1 + z_2}{|z_1 + z_2|}$
where $z_1, z_2, z_3 \in \mathbb{C} \ \wedge \ |z_1| = |z_2| = 1$
mathematical formulation
from the software requirements
software developers
(may not be familiar with numerical analysis,
just program directly following the
mathematical formulation)
IComplex midarc(IComplex z1,IComplex z2){
if(abs(z1)!=1 || abs(z2)!=1)
throw PreConditionException;
REAL r = realpart(z1) + realpart(z2);
REAL i = imaginary(z1) + imaginary(z2);
IComplex sum(r,i);
IComplex z3 = sum / abs(sum);
return z3;
}
direct numerical program
(written in infinite-precision arithmetic,
easy to maintain,
however, may be unstable in finite-precision arithmetic)
Global Numerical
Optimization
Framework
optimization rules
numerical
experts
FComplex midarc(FComplex z1,FComplex z2){
if((abs(z1)-1)>epsi || (abs(z2)-1)>epsi)
throw PreConditionException;
double r = realpart(z1) + realpart(z2);
double i = imaginary(z1) + imaginary(z2);
FComplex sum(r,i); FComplex z3;
if(abs(sum)<epsi){
double theta1 =
atan2(imaginary(z1),realpart(z1));
double theta2 =
atan2(imaginary(z2),realpart(z2));
double theta3 = (theta1+theta2)/2;
z3 = FComplex(cos(theta3),sin(theta3));
}else{
z3 = sum / abs(sum);
}
return z3;
}
optimized numerical program
(in fixed-precision floating-point arithmetic
with numerically stable algorithms)