Apache Commons Math

A Java library of mathematical tools

Thomas Neidhart
tn@apache.org

Gilles Sadowski
erans@apache.org

FOSDEM 2013

February 02, 2013

Overview of the talk

- What is it?
- What does it provide?
- Some simple examples
- Who is using it?
- What to do after the presentation?

What is it?

- 100% pure, self-contained Java library
- mathematical methods & algorithms
- covers a wide range of topics
- aims for clean, documented and object-oriented code
- well tested and maintained
- following the Apache way
- Apache 2.0 license

Commons Math in Numbers

As of January 1, 2013

- 57 packages
  829 classes
  77,761 lines of code
- 50,512 lines of comments
  99.4% documented API
- 4,537 unit tests
  87.6% of code covered
- 98 open issues
  831 resolved issues
- 0 dependencies

What topics does it cover?

- Automatic Differentiation
- Numerical Integration
- Interpolation
- Root finders
- Arbitrary precision arithmetic
- Probability distributions
- Linear Algebra
- Fitting (non-linear least-squares, curve fitting)
- Ordinary differential equations (ODE)
- Linear/Non-linear optimization
- Random number generators
- Computational Geometry
- Special functions (Beta, Gamma)
- Kalman filter
- Fast Fourier transform
- Genetic algorithms
- Statistics (correlation, regression, descriptive, inference)

FastMath

- pure Java replacement for java.lang.Math
- better performance for most of the functions
- better accuracy than Math and StrictMath
- currently not always faster :-(

FastMath benchmark I

Number of calls: 10^9

Java 1.7.0_03 (1.7.0_03-b21) OpenJDK 64-Bit Server VM (22.0-b10)

Function	Class	Time/call (ms)	Ratio (%)
log	StrictMath	5.22349648e-05	100
	Math	2.04540923e-05	39
	FastMath	2.42863614e-05	47
log10	StrictMath	7.06720929e-05	100
	Math	2.05568765e-05	29
	FastMath	7.56041686e-05	107
log1p	StrictMath	3.72191413e-05	100
	Math	3.78759816e-05	101
	FastMath	7.50277567e-05	201
pow	StrictMath	1.73481768e-04	100
	Math	1.64277768e-04	94
	FastMath	1.05758357e-04	60
exp	StrictMath	3.47634732e-05	100
	Math	2.28322028e-05	65
	FastMath	1.72182428e-05	50
sin	StrictMath	2.86189157e-05	100
	Math	2.72834491e-05	95
	FastMath	2.13875014e-05	75
asin	StrictMath	3.61743365e-05	100
	Math	3.63858979e-05	100
	FastMath	6.65170752e-05	184
cos	StrictMath	3.23569717e-05	100
	Math	3.09660350e-05	96
	FastMath	2.37797205e-05	73
acos	StrictMath	3.74091467e-05	100
	Math	3.72805271e-05	99
	FastMath	8.99709198e-05	240
tan	StrictMath	4.36050840e-05	100
	Math	3.94326807e-05	90
	FastMath	3.96236785e-05	91

Function	Class	Time/call (ms)	Ratio (%)
atan	StrictMath	2.38076340e-05	100
	Math	2.38346576e-05	100
	FastMath	2.78287849e-05	117
atan2	StrictMath	6.14545612e-05	100
	Math	6.25871086e-05	102
	FastMath	6.24287698e-05	102
hypot	StrictMath	2.51933255e-04	100
	Math	2.51892886e-04	100
	FastMath	1.78561127e-05	71
cbrt	StrictMath	7.08035203e-05	100
	Math	7.09881088e-05	100
	FastMath	4.12985279e-05	58
sqrt	StrictMath	6.69162232e-06	100
	Math	6.63196797e-06	99
	FastMath	6.62463595e-06	99
cosh	StrictMath	7.51256426e-05	100
	Math	7.52944441e-05	100
	FastMath	3.91062794e-05	52
sinh	StrictMath	7.52920027e-05	100
	Math	7.58819054e-05	100
	FastMath	5.80563672e-05	77
tanh	StrictMath	6.94914940e-05	100
	Math	7.01421819e-05	100
	FastMath	6.10102135e-05	88
expm1	StrictMath	4.19029705e-05	100
	Math	4.18109329e-05	100
	FastMath	3.19643446e-05	76
abs	StrictMath	5.08597800e-06	100
	Math	5.14014393e-06	101
	FastMath	5.37928625e-06	105

Example - Gauss Integration

```
1  import o.a.c.m.analysis.UnivariateFunction;
2  import o.a.c.m.analysis.function.Cos;
3  import o.a.c.m.analysis.integration.gauss.GaussIntegratorFactory;
4  import o.a.c.m.analysis.integration.gauss.GaussIntegrator;
5
6  // ...
7
8  GaussIntegratorFactory factory = new GaussIntegratorFactory();
9  UnivariateFunction cos = new Cos();
10 // create an Gauss integrator for the interval [0, PI/2]
11 GaussIntegrator integrator = factory.legendre(7, 0, 0.5 * Math.PI);
12 double s = integrator.integrate(cos);
```

Example - Root finder

```
1  import o.a.c.m.analysis.UnivariateFunction;
2  import o.a.c.m.analysis.solvers.BrentSolver;
3  import o.a.c.m.analysis.solvers.UnivariateSolver;
4
5  // ...
6
7  UnivariateFunction f = new UnivariateFunction() {
8      public double value(double x) {
9          return FastMath.sin(x);
10     }
11 }
12 UnivariateSolver solver = new BrentSolver();
13 // we know that the root is somewhere between 3 and 4 ;-)
14 double result = solver.solve(100, f, 3, 4);
```

Example - Eigenvalue Decomposition

```
1   import o.a.c.m.linear.RealMatrix;
2   import o.a.c.m.linear.MatrixUtils;
3   import o.a.c.m.linear.EigenDecomposition;
4
5   // ...
6
7   RealMatrix matrix =
8       MatrixUtils.createRealMatrix(new double[][] {
9               {  5,   10,   15 },
10              { 10,   20,   30 },
11              { 15,   30,   45 }
12  });
13
14  EigenDecomposition ed = new EigenDecomposition(matrix);
15  double[] eigValues = ed.getRealEigenvalues();
```

Example - Linear optimization: Simplex method

Maximize $2x_2 + 6x_1 + 7x_0$
with constraints

$$x_2 + 2x_1 + x_0 <= 2$$
$$-x_2 + x_1 + x_0 <= -1$$
$$2x_2 - 3x_1 + x_0 <= -1$$

```
1   import o.a.c.m.optim.linear.LinearObjectiveFunction;
2   import o.a.c.m.optim.linear.LinearConstraint;
3   import o.a.c.m.optim.linear.SimplexSolver;
4
5   /// ...
6
7   LinearObjectiveFunction f = new LinearObjectiveFunction(new double[] { 2, 6, 7 }, 0);
8
9   ArrayList<LinearConstraint> constraints = new ArrayList<LinearConstraint>();
10  constraints.add(new LinearConstraint(new double[] { 1, 2, 1 }, Relationship.LEQ, 2));
11  constraints.add(new LinearConstraint(new double[] { -1, 1, 1 }, Relationship.LEQ, -1));
12  constraints.add(new LinearConstraint(new double[] { 2, -3, 1 }, Relationship.LEQ, -1));
13
14  SimplexSolver solver = new SimplexSolver();
15  PointValuePair solution = solver.optimize(new MaxIter(100), f, new LinearConstraintSet(constraints),
16                                            GoalType.MAXIMIZE, new NonNegativeConstraint(false));
```

Who is using Commons Math?

Responses from our community:

Field	Topic	Link
Space	Orekit - open-source space dynamics library	http://www.orekit.org
Astronomy	Gaia Project: data processing of astrometric, photometric, and spectroscopic data from 1 billion stars	http://www.rssd.esa.int/index.php?project=GAIA&page=DPAC_Introduction
Material Science	Simulate the mechanical behaviour of composite materials	http://navier.enpc.fr/BRISARD-Sebastien
Biotech	Nuclear Magnetic Resonance (NMR) data analysis package for molecular structure refinement	http://www.onemoonscientific.com
Medical Technology	Image processing of 3D human MRI and CT scans	http://www.stjude.org
Finance	Risk managment and analysis	http://www.osloclearing.no/

A maven dependency search:

- Apache Hama - parallel computing framework
- Myrrix - A real-time recommender system
- Apache Mahout - scalable machine learning
- Redberry - Symbolic Tensor Algebra System
- Facebook jcommon-stats - well you know it
- ...

How to use it yourself?

Maven:

```
<dependency>
<groupId>org.apache.commons</groupId>
<artifactId>commons-math3</artifactId>
<version>3.1.1</version>
</dependency>
```

Download:
http://commons.apache.org/math/download_math.cgi

Links

- Project homepage: http://commons.apache.org/math/
- Issue tracker: https://issues.apache.org/jira/browse/MATH
- Mailinglists: dev@commons.apache.org & user@commons.apache.org
  e-mail subject: [math]
- Wiki: http://wiki.apache.org/commons/MathWishList

How to contribute?

- Check out the user guide
  http://commons.apache.org/math/userguide/index.html
- Ask questions on the user mailinglist
- Participate in discussions on the dev mailinglist
- Create bug reports / feature requests / improvements in JIRA
- Send patches (code, documentation, examples)
- Provide feedback - most welcome!

Hot Topics

- refactoring Linear Algebra package
- refactoring Optimization package
- Exception handling - checked vs. unchecked
- Add more computational geometry algorithms
- Upgrade to more recent Java versions (fork/join framework)
- (real) FastMath vs. AccurateMath (integration of jodk)
- improve user guide / examples
- lots of new feature requests ...

The End!

- Questions?