Orekit Development Status

Orekit PMC: Paul Cefola (University Buffalo), Frank Dreger (ESOC), Nicolas Frouvelle (CS), Hank Grabowski (ADS), Sébastien Herbinière (TAS), Stéphanie Lizy-Destrez (ISAE), Luc Maisonobe (CS), Guillermo Ortega (ESTEC), Pascal Parraud (CS), Evan Ward (NRL)
Agenda

- History
- Recent releases
- Major features
- Project organisation
- Trends
History

Phase 1: closed-source
- 2002

Phase 2: open-source
- 2008

Phase 3: collaborative development
- 2011

Phase 4: open governance
- 2012
Recent releases

Java 6, Apache Commons Math

Java 8, Hipparchus

6.0 6.1
2013 2014
not maintained anymore

7.0 7.1 7.2
2015
security fix

8.0 8.0.1
2016

9.0 9.0.1 9.1
2017
Major features

- 6.X series
  - DSST propagator with mean elements
  - Jacobians
- 7.X series
  - DSST with short periodic terms
  - Second order derivatives for many models
- 8.X series
  - Switch to Hipparchus and Java 8
  - Orbit determination
- 9.X series
  - Field propagators
  - GNSS/very high precision
Orbit Determination

- Weighted Batch least square in 8.0
  - measurements: range, range-rate, az/el, PV
  - parameters: orbit (even partial elts), drag, SRP, station position, biases

- New models in 9.x
  - measurements: turn-around, $\alpha/\delta$, intersat range
  - parameters: EOP, parametric acceleration
  - multi-sat orbit determination (see later)
  - fast handling of tens of thousands of measurements
  - ground points displacements

- Upcoming
  - Kalman (probably 9.2 very soon)
Field propagators (1/3)

double f(double x, double y) {
   if (x > 0) {
      return x + g(y);
   } else {
      return x - g(y);
   }
}

double based computation

T f(T x, T y) {
   if (x.getReal() > 0) {
      return x.add(g(y));
   } else {
      return x.subtract(g(y));
   }
}

field based computation
Field propagators (2/3)

- **Field T** is an enhanced double
  - supports all double operations (+, -, *, /)
  - supports all mathematical functions (sin, ..., atanh)
  - supports all ieee functions (scalb, copysign...)
  - supports mixed operations

- **Predefined fields**
  - Dfp, Decimal64, DerivativeStructure, FieldDerivativeStructure, SparseGradient, Tuple, FieldTuple
Field propagators (3/3)

- **DerivativeStructure**
  - Taylor algebra
    - typically 6 parameters, order 3
    - uncertainties propagation
    - very fast Monte-Carlo analysis
  - Orbit determination?
    - to be discussed in roundtable

- **Tuple**
  - Parallel propagation
    - domain exploration
    - co-positioning
Taylor Algebra example
Multiple orbit propagation

- multi-threading and controller
  - used for multisat orbit determination
  - can mix propagators types
  - rendezvous on step ends
  - events only at propagator level
- field propagator and Tuple
  - co-positionning
  - multi-sat events
  - should **not** be used if satellites are too far away
Project organization

- Git-flow like branching model
  - Users with different needs have a choice
  - Better traceability of features development
- Faster process to release critical bug fixes
  - No more PMC vote delay
  - Should never be use...
  - ... was used 2 days after having been adopted
git-flow like branching model

- **Need stability?**
  - select master branch
  - it points to latest release
- **Maintain a product?**
  - stick to release-x.y
  - it includes patches for x.y
- **Want latest features?**
  - follow develop branch
  - it is bleeding edge
Security fix released
Trends (1/3)

- GNSS / very high precision
  - Multi-satellite propagation
  - Multi-satellite orbit determination
  - Parametric accelerations
  - Ground points displacements
    - Tides
    - Ocean loading
  - ANTEX files loading
  - GNSS specific eclipse/noon turns in attitude
Trends (2/3)

- Persistently adding standard data interface
  - IERS (since the beginning)
  - JPL/Inpop (4.1)
  - SP3 (6.0)
  - CCSDS ODM (6.1 for read, 9.0 for writing OEM)
  - WMM (7.1)
  - SEM – Yuma (8.0)
  - CGIM (9.0)
  - CCSDS TDM (9.0)
  - ANTEX (9.1)
  - Upcoming: RINEX, STK .e, OLES, CCSDS ADM...
    - to be discussed during roundtable
Trends (3/3)

- Closed source
- Closed development
- Open development (forge)
- Continuous Integration
- Git flow
- Nightly builds, continuous delivery?