Jpp

M. de Jong

26/06/2018

Introduction (1/2)

- Jpp [yi-pee-pee]
  - an exclamation used to express joy, exultation, or the like
- Jpp is a collection of Java inspired C++ interfaces, classes, methods and applications
- Jpp consists of various “packages”
  - organised in corresponding sub-directories and name spaces
  - most packages can be used without prior compilation
- Applications and libraries can be produced with a standard make procedure

Introduction (2/2)

- Available from GIT server
  - see KM3NeT wiki
- $JPP_DIR/software/JXXX
  - include files
  - source files
- $JPP_DIR/examples/JXXX
  - about 300 easy-to-read examples
- Documentation based on Doxygen
  - make doc
  - integrates documents, presentations, etc.
  - also available from document pages server

Packages (1/8)

- JLang/
  - language specific auxiliary classes and methods
    - type holder, type list, etc.
  - I/O iterator interfaces (see separate presentation on I/O)
- JMath/
  - base class for data structures with arithmetic capabilities
    - automatically provides implementations for standard operators
      `+ - * / += -= *= /=`
  - matrix operations, inversions, etc.
  - auxiliary object JMATH::zero for consistent zero value of all primitive data types and composite data structures

Packages (2/8)

- JGeometry2D/
  - 2D geometry classes, tools and algorithms
    - convex hull, minimum enclosing disk, smallest distance
- JGeometry3D/
  - 3D geometry classes, tools and algorithms
    - 3D rotations, intersections, smallest distance

JRotate3D(getDirection(..)) rotates coordinate system such that given direction (e.g. track) is along z-axis.

Packages (3/8)

- JTools/
  - constants
  - multi-dimensional interpolations of tabulated function values
    - polynomial or spline
    - equidistant or non-equidistant abscissa values
    - various return types
  - multi-dimensional histograms
  - numerical integration in multiple dimensions

Packages (4/8)

- JDetector/
  - detector data structures and auxiliaries
    - O(1) lookup tables for PMT and optical module data
    - simulation of PMT response and CLB firmware
    - detector dependent PMT mapping
      - readout channel {0, .., 30} to logical position {A1, .., F6}
- JTrigger/
  - KM3NeT trigger software
    - L1-L2 local coincidence logic (see slides Examples)
    - general purpose cluster methods, see Analysis e-log 34

Packages (5/8)

- JPhysics/
  - muon energy loss methods
  - shower profile methods
  - neutrino interaction cross sections
  - optical properties of water at deep-sea sites
  - probability density functions (PDFs) of the arrival time of Cherenkov light
  - PDF creation, drawing and comparison applications
- JAstronomy
  - interface to slalib
  - auxiliaries

Packages (6/8)

- JROOT/
  - handler for TTree and TObjectWrite (see slides on I/O)
  - ASCII I/O based on ROOT dictionaries
- JAAnet/
  - implementation of interfaces between km3net-dataformat and Jpp
  - JHead : Head {};
    - copies string values of specific tags to concrete data types
    - implements equals() and add() methods
- JDAQ/
  - binary I/O of DAQ data formats

Packages (7/8)

- JFit/
  - general purpose fit algorithms
    - JEstimator <JModel_t> linear fit
    - JSimplex <JModel_t> Powell's method
    - JGandalf <JModel_t> Levenberg-Marquardt method
- JReconstruction/
  - muon trajectory fit applications
  - low-energy shower fit applications

Packages (8/8)

- JSystem/
  - I/O with operating system
    - getRAM(), getMemoryUsage(), ls(..), which(..)
- JGizmo/
  - print and file formatting applications
  - drawing applications
  - histogram operations, fit and plot applications

User (1/3)

- Environment
  - source setenv.[c]sh
    - set PATH and LD_LIBRARY_PATH
- Libraries
  - predefined symbolic names
- Applications
  - common command line interface (see next slide)
- Scripts
  - option -h will print usage

User (2/3)

- JParser: command line parser

  applies to all JXXX applications

  -h print help // exit

  -h! print default & possible values // exit

  -v print SVN revision // exit

  -- end of options // continue

  --! print actual values // continue

User (3/3)

- Meta data

  applies to JXXX applications with ROOT I/O

  - output
    - JMeta(argc, argv); // command line options

  - JPrintMeta –f <file name> [-A <application>]
    - prints name of application
    - SVN release
    - ROOT release
    - command line options
    - system information

User (4/4)

- typical common command line options
  - -f <input file>
  - -f "a.root b.root c.root"
  - -n <number of events>
  - -n <first event>:<number of events>
  - -o <output file>
  - -a <detector file>
  - -@ "<trigger parameter>=<value>; ..."
  - -@ <trigger parameter file>

File formats

- Monte Carlo
  - ASCII .evt header and event data
  - gzipped .gz gzipped header and event data
- data file
  - ROOT .root DAQ data types
  - binary .dat DAQ data types
- detector calibration
  - ASCII .detx standard
  - gzipped .gz gzipped standard format
  - gendet .det Monte Carlo, only input
  - binary .dat internal format

Applications (1/5)

- JPrint — print any data to terminal
- JPrintTree — print TTree statistics
- JPrintChain — print TChain statistics
- JPrintHeader — print sum of Monte Carlo headers
- JPrintMeta — print meta data to terminal
- JConvert — convert file format

examples:

- JConvert -f <input file> -o <output file> -n 12345:1
  - copies event 12345 to output
- JPrintHeader -f <> -k “livetime.numberOfSeconds”
  - prints just live time

Applications (2/5)

• JDetector	create detector
• JPrintDetector	print detector to terminal
• JConvertDetectorFormat	convert file format
• JDrawDetector2D	draw footprints (2D)
• JDrawDetector3D	draw detector in 3D
• JCompareDetector	print differences between two detectors

Applications (3/5)

- JDrawPDX¶ — draw PDF
- JMakePDX — create PDF tables
- JMergePDX — merge PDF tables
- JPlotPDX — plot PDF tables
- JDiffPDX — print differences between two PDFs
- JHistHDX — histogram Monte Carlo data
- JMakeHDX — convert histogram to PDF

¶ X = F corresponds to light from muon (4D); X = G to single EM-shower (5D).

Applications (4/5)

- JPlot1D/JPlot2D
  - -f <input file>:<histogram name>
  - .. // \[<label text>\] may appear in file or histogram name
- JFit/JFit2D
  - -f <input file>:<histogram name>
  - -F "[0] * exp(-0.5 * (x-[1])*(x-[1]) / ([2]*[2]))"
  - -@ "p0 = GetMaximum"
  - -@ "p1 = GetMean"
  - -@ "p2 = 0.5*GetRMS"

Applications (5/5)

- JSirene
  - fast simulation of detector response to muons and showers
- JTriggerEfficiency
  - apply standard trigger to Monte Carlo data
- JTriggerProcessor
  - apply standard trigger to (random) data
- JMuonPrefit – JMuonSimplex – JMuonStart – JMuonGandalf – JMuonEnergy
  - fit muon trajectory, energy and start position

Examples (1/3)

- JTools/JPolynomeXX
  - interpolation methods in various numbers of dimensions
- JTools/JSphereXX
  - integration methods in various numbers of dimensions
- JTools/JMultiPDF
  - create and fill 4D histogram
  - create PDF: $f\,(x_0, x_1 \mid x_2, x_3)$
  - performance of PDF

Examples (2/3)

- JTrigger/JSignalL1 & JRandomL1
  - test efficiency and purity of L1-L2 coincidences
- JTrigger/JFilter
  - test efficiency and purity of cluster methods
- JTrigger/JHitL1
  - create time slewing correction method for L1 hits
- JDetector/JTTS
  - create PDF and CDF of transition times of PMT

Examples (3/3)

- JDAQ/JDAQXXX
  - plot basic histograms of real data
- JSirene/JLight
  - plot optical properties of water
- JSirene/JPMT
  - plot PMT characteristics