Jas4pp - a Data-Analysis Framework for Physics and Detector Studies

S.V. Chekanov[a], N. A. Graf[b], G. Gavalian[c]

[a] *HEP Division, Argonne National Laboratory, 9700 S. Cass Avenue, Argonne, IL 60439, USA.*
[b] *SLAC National Accelerator Laboratory, Menlo Park, CA 94025 USA.*
[c] *Thomas Jefferson National Accelerator Facility, Newport News, Virginia 23606, USA.*

This Shownmass21 contribution will discuss the Jas4pp program (Java Analysis Studio for Particle Physics) for data analysis and event visualization [1]. It is currently supported by the HEP division at ANL. The core part of Jas4pp is the package called Jas3 (Java Analysis Studio) [2] created at SLAC. It is an AIDA compliant environment which became a flexible platform for different types of experiments. Jas4pp, being backward compatible with the original Jas3, focuses on data analysis in collider particle physics. It includes 2D/3D visualisation libraries from the DataMelt platform [3], GROOT libraries developed at JLab, mathematical libraries from the Apache foundation and a software library [4] to validate Monte Carlo events from the HepSim repository [5].

Jas4pp fully supports the LCIO file format [6], ProMC [7] and ProIO [8] event formats based on Google Protocol Buffers, which allows variable byte encoding. A limited support for reading ROOT files is available too. The native to Java and Python I/O methods are also available.

Jas4pp includes the popular HEP libraries by default, instead of using Jas3 plugins. For example, the LCSIM library is a part of Jas4pp. This makes Jas4pp a self-contained program for analysis and visualization of reconstructed events. The LCSIM library was significantly improved to increase the processing speed of tracking hits when complex events are visualized inside Jas4pp.

The main features of Jas4pp are: (a) a full stack of data analysis libraries (histogramming, ntuples, non-linear regression using interactive GUI); (b) 2D / 3D visualization of data and functions; (c) Physics libraries supporting Lorentz vectors, various data containers, event shape and jet reconstruction. In particular, the antiKT4 algorithm for analysis of pp events is included, in addition to the standard set of jet algorithms used in e^+e^-. Jas4pp can be used to analyze truth-level events from the HepSim repository (which contains more than 100 scripts executed directly in Jas4pp), as well as to process and visualize events after the Geant4 simulation [9].

Jas4pp supports three programming languages for analysis: Java, Python and Groovy. The latter two are scripting (dynamically typed) languages. With the rise of the Python language, Jas4pp adopts Jython (version 2.7.2) as the main language for

Email addresses: chekanov@anl.gov (S.V. Chekanov), ngraf@slac.stanford.edu (N. A. Graf), gavalian@jlab.org (G. Gavalian)

user analysis. It is supported in the GUI mode (using built-in editor with the console for output), as well as using the batch mode. In addition, Apache Groovy [10] version 2.6 is also supported. Similar to Jython, it is optionally typed, dynamic language. The main advantage of Groovy is in the fact that the execution speed is significantly faster than equivalent Jython or CPython code (implemented in C) Benchmarks indicate that execution of Groovy code that implements long loops is a factor 10 faster than the equivalent Jython / CPython programs.

In the past, the Jas3 analysis platform was extensively used for the SiD detector concept [11] of the ILC. Jas4pp was used in a number of recent studies, such as CEPC silicon tracking studies [12], FCC-hh [13, 14, 15, 16] detector studies, TOPSiDE detector and physics studies for the EIC future collider [17]. It has also been used for a number of more general studies for the HL-LHC and HE-LHC where HepSim Monte Carlo event samples were used.

In conclusion, Jas4pp provides a modern platform-independent program supported on the most popular end-user operating system (Linux, Mac OS, Windows). It allows programming using fast dynamically-typed language (Groovy), supports the Python syntax, includes variable-byte encoding for I/O, libraries for data visualization in 2D/3D and HEP physics libraries for reconstructed events. A built-in universal event display can be used for visualization of detectors. This environment will be further expanded and documented during Snowmass 2021. Together with the description of this program, a number of tutorials and advanced examples will be described.

[1] ANL/HEP, JAS4pp. Java Analysis Studio for Particle Physics.
URL https://atlaswww.hep.anl.gov/asc/jas4pp/

[2] FreeHep, JAS3. Java Analysis Studio.
URL http://jas.freehep.org/jas3/

[3] S. Chekanov, Numeric Computation and Statistical Data Analysis on the Java Platform, (Book), Springer, London, 2016.
URL https://datamelt.org

[4] S. Chekanov, I. Pogrebnyak, D. Wilbern, Cross-platform validation and analysis environment for particle physics, Comput. Phys. Commun. 220 (2017) 91–96. arXiv:1510.06638, doi:10.1016/j.cpc.2017.06.017.

[5] S. V. Chekanov, HepSim: a repository with predictions for high-energy physics experiments, Advances in High Energy Physics 2015 (2015) 136093, http://atlaswww.hep.anl.gov/hepsim/ (accessed on Feb. 1st, 2019). arXiv:1403.1886, doi:10.1155/2015/136093.

[6] LCIO (Linear Collider I/O) is a persistency framework and event data model for linear collider detector studies, Web page. [link].
URL https://github.com/iLCSoft/LCIO

[7] S. Chekanov, K. Strand, P. Van Gemmeren, E. May, ProMC: Input-output data format for HEP applications using varint encoding, Comput. Phys. Commun. 185 (2014) 2629–2635. arXiv:1311.1229, doi:10.1016/j.cpc.2014.06.016.

[8] D. Blyth, J. d. J. Alcaraz, S. Binet, S. V. Chekanov, ProIO: An Event-Based I/O Stream Format for Protobuf Messages, Computer Physics Communications 241. doi:10.1016/j.cpc.2019.03.018.

[9] J. Allison, et al., Recent developments in Geant4, Nuclear Instruments and Methods in Physics Research A 835 (2016) 186.

[10] Apache Groovy, Web page. [link].
URL https://groovy-lang.org/

[11] H. Abramowicz, et al., The International Linear Collider Technical Design Report - Volume 4: DetectorsarXiv:1306.6329.

[12] CEPC Study Group Collaboration, M. Dong, et al., CEPC Conceptual Design Report: Volume 2 - Physics & DetectorarXiv:1811.10545.

[13] S. Chekanov, M. Beydler, A. Kotwal, L. Gray, S. Sen, N. Tran, S. S. Yu, J. Zuzelski, Initial performance studies of a general-purpose detector for multi-TeV physics at a 100 TeV pp collider, JINST 12 (06) (2017) P06009. arXiv:1612.07291, doi:10.1088/1748-0221/12/06/P06009.

[14] C. Yeh, S. Chekanov, A. Kotwal, J. Proudfoot, S. Sen, N. Tran, S. Yu, Jet Substructure Variables with the SiFCC Detector at 100 TeV, PoS ICHEP2018 (2019) 905. arXiv:1811.12805, doi:10.22323/1.340.0905.

[15] C.-H. Yeh, S. Chekanov, A. Kotwal, J. Proudfoot, S. Sen, N. Tran, S.-S. Yu, Studies of granularity of a hadronic calorimeter for tens-of-TeV jets at a 100 TeV *pp* collider, JINST 14 (05) (2019) P05008. arXiv:1901.11146, doi:10.1088/1748-0221/14/05/P05008.

[16] S. Chekanov, A. Kotwal, C.-H. Yeh, S.-S. Yu, Physics potential of timing layers in future collider detectors, ANL-HEP-159872, contribution to Snowmass 2021. arXiv:2005.05221.

[17] J. Repond, TOPSiDE: Concept of an EIC Detector, XXVI International Workshop on Deep-Inelastic Scattering and Related Subjects (DIS2018) 16-20 April 2018 Kobe University, Kobe, Japan, 2018.