Iguana

News

https://github.com/JeffersonLab/iguana

Jefferson Lab

clas

Christopher Dilks

CLAS Collaboration Meeting, November 2024

What is Iguana?

Encapsulate, centralize, and preserve common needs in **Iguana Algorithms**

- Methodology preservation (*cf*. data preservation efforts)
- Reproducibility
- Allow for focus on the important parts of an analysis

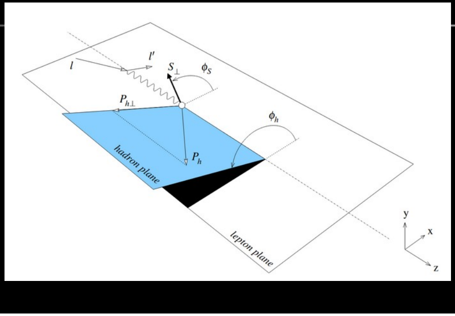
Details in last collaboration meeting: https://indico.jlab.org/event/863/

Updates of Existing Algorithms

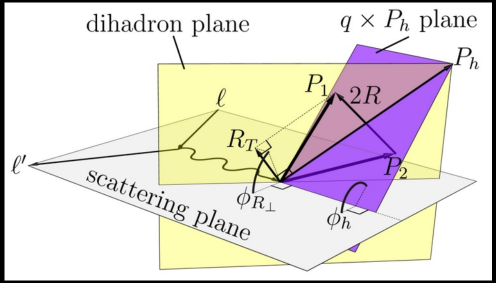
- **ZVertexFilter**
  - More flexibility in the configuration
  - Updates for thread safety
- **SectorFinder**
  - Support charged and neutral particles
  - Added validator
- **LorentzTransformer**
  - Removed, since it was just an example, and not general enough

New Algorithms

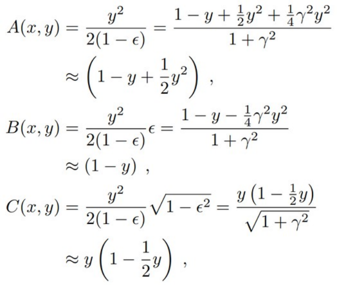
- **SingleHadronKinematics**
  - SIDIS kinematics of ep → ehX
  - z_h, $P_{h\perp}$, M_X, x_F, ϕ_h, ξ, Y_h



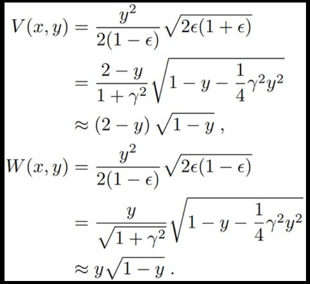
- **DihadronKinematics**
  - SIDIS kinematics of ep → ehhX
  - z_{hh}, $P_{hh\perp}$, M_X, M_{hh}, x_F, ϕ_h, ϕ_R, θ, Y_{hh}



- **Depolarization** (coming soon)
  - Calculate depolarization factors
  - ε, A, B, C, W, V



$$
\begin{aligned}
A(x,y) &= \frac{y^2}{2(1-\epsilon)} = \frac{1-y+\frac{1}{2}y^2+\frac{1}{4}\gamma^2y^2}{1+\gamma^2} \\
&\approx \left(1-y+\frac{1}{2}y^2\right) , \\
B(x,y) &= \frac{y^2}{2(1-\epsilon)}\epsilon = \frac{1-y-\frac{1}{4}\gamma^2y^2}{1+\gamma^2} \\
&\approx (1-y) , \\
C(x,y) &= \frac{y^2}{2(1-\epsilon)}\sqrt{1-\epsilon^2} = \frac{y\left(1-\frac{1}{2}y\right)}{\sqrt{1+\gamma^2}} \\
&\approx y\left(1-\frac{1}{2}y\right) ,
\end{aligned}
$$



$$
\begin{aligned}
V(x,y) &= \frac{y^2}{2(1-\epsilon)}\sqrt{2\epsilon(1+\epsilon)} \\
&= \frac{2-y}{1+\gamma^2}\sqrt{1-y-\frac{1}{4}\gamma^2y^2} \\
&\approx (2-y)\sqrt{1-y} , \\
W(x,y) &= \frac{y^2}{2(1-\epsilon)}\sqrt{2\epsilon(1-\epsilon)} \\
&= \frac{y}{\sqrt{1+\gamma^2}}\sqrt{1-y-\frac{1}{4}\gamma^2y^2} \\
&\approx y\sqrt{1-y} .
\end{aligned}
$$

Continuous Integration (CI)

- **Treat warnings as errors**
  - Developers should fix their warnings
  - Off by default, to not slow down local development
- **Coverage (Gcovr) tests are broken**
  - Possible culprit is the introduction of multi-threading support
  - Not sure yet how to fix (lcov?)
- **Various small fixes and maintenance**
- **Plans**
  - Use clas12-containers images, rather than weekly rebuild of dependencies (HIPO and ROOT)
  - Migrate to code.jlab.org?

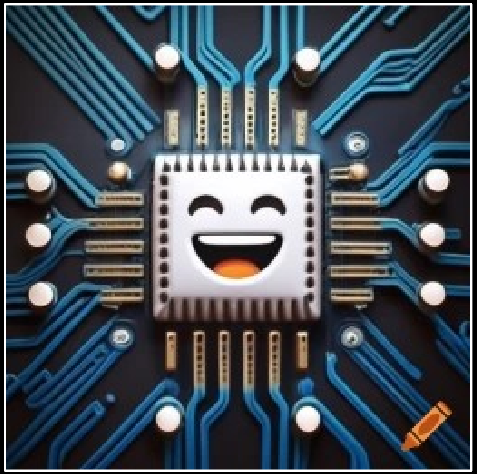
New Feature: Chameleon

- **Simple language binding and test generator**
  - Uses YAML specification of action functions to generate bindings
  - TODO: generate tests of those bindings
- **Available Languages**
  - Currently only Fortran
  - Python is still supported by cppyy (the same thing that PyROOT uses)
  - Planning on supporting Python and Java
- **Why not just use SWIG?**
  - We’ll try it; at the very least it can help inspire how to write other bindings
  - Might complicate user installation (easily solved by clas12-containers!)
  - Fortran is not supported by SWIG, but the “SWIG-Fortran” fork does
    - https://github.com/swig-fortran
    - There is a PR for adding Fortran to SWIG that has been under review for *6 years*, but its author recently closed it: https://github.com/swig/swig/pull/1195

SWIG

New Feature: Thread-Safe Configuration

- **What are you talking about?**
  - Some configuration parameters depend on the data
  - For example the electron z-vertex cuts depend on the run number
    - If the run number changes in the data Iguana is processing, the z-vertex cuts may change
    - Such change needs to be handled in a thread-safe way
    - Save time and “memoize” the configuration:
      - https://en.wikipedia.org/wiki/Memoization
      - Can enable “single-threaded” mode, which does not memoize, but is not thread safe
  - Opens the door for RCDB, CCDB, and QADB usage



- **Heads up! The QADB code will be re-implemented as an Iguana algorithm!**
  - Thread safety
  - Language bindings

QADB

Upstream Updates

- **HIPO 4.2.0**
  - Multithreaded reader
    - Used in Iguana testing of algorithm’s thread safety
    - WARNING: the API may change, since we want it to be a bit more user-friendly
  - Other changes:
    - Now using Meson build system
      - Automated testing, sanitizers, and fixed a few small bugs
      - Removed the Makefile and CMake configuration
    - Bug fix in dataframe library (thanks to Maurik and Whit)

HIPO 4

That’s it!

Stay tuned for a new Iguana Release, v0.8

Any questions?