



A Faster, More Intuitive RooFit

Stephan Hageboeck
EP-SFT CERN

<https://root.cern>

Introduction: RooFit

- C++ package for creating statistical models [1]
- Used by all LHC experiments for statistical inference
- Crucial for final steps of analysis
- LHC's Run 3: more data and more elaborate models
- **Goals:**
 - Speed up RooFit $\geq 5x$
 - Simpler interfaces, better interoperability

1. Better Interfaces

ROOT 6.18

- "RooLinkedList" has been replaced by `std::vector`
 - 25% faster iteration, simpler C++ loops

```
Iterator* it = pdf.getParameters(obs)->createIterator();
RooAbsArg* p;
while ((p=(RooAbsArg*)it->Next())) {
    p->Print();
}
delete it;
```

Classic RooFit
(Supported, old
code still works)

```
for (auto p : pdf.getParameters(obs))
    p->Print();
```

ROOT 6.18
25% faster

- Also enables **pythonic loops**, as PyROOT works well with STL containers

```
for p in pdf.getParameters(obs):
    p.Print()
```

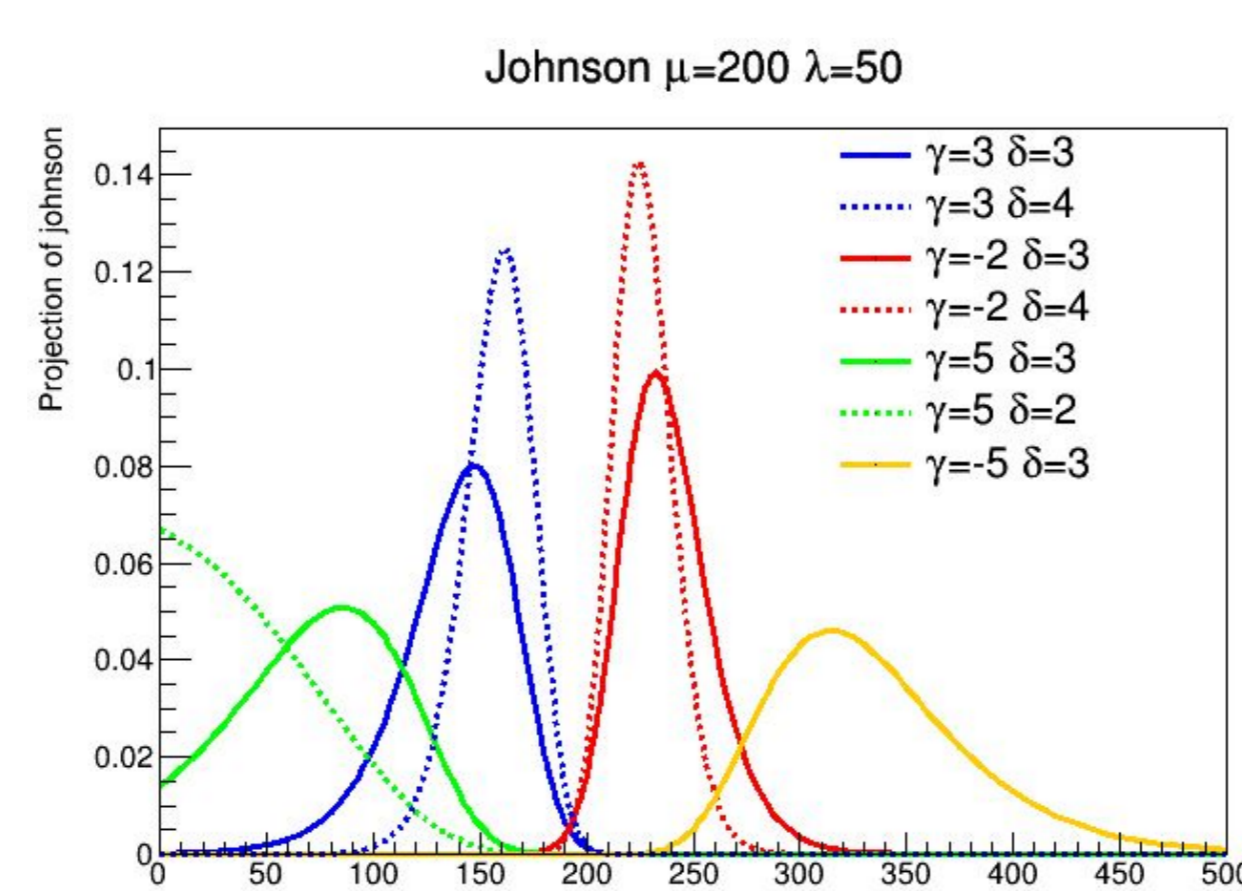
ROOT 6.18

- **Speeds up real-world fits:**
ATLAS $H \rightarrow b\bar{b}$ Run 1 measurement [2] **20% faster:**
11:30 min \rightarrow 9:20 min, identical result
- **Plans:**
 - Improve interoperability. Read data from e.g.:
 - RDataFrame, STL containers, Numpy arrays
 - Add more STL-like, simpler C++ interfaces
 - Will allow designing pythonic interfaces when new PyROOT available ([Talk in Track 6](#))
 - Ease of use of Python, speed of C++

2. New Built-in PDFs

ROOT 6.18

- Include frequently used PDFs for LHCb
 - Citeable standard
 - Allows for optimisations, e.g. **RooJohnson:**
 - > 6x faster, better stability, more accurate integrals
- **In preparation:**
 - RooHypatia2
 - Double-sided crystal ball shape



3. Just-in-time-compiled PDFs

ROOT 6.20

- ROOT's [TFormula](#) integrated in RooFit
 - Pass any C++ expression, compiled just in time
 - Benefit from compiler optimisations
 - Can call user/library functions that have been loaded:

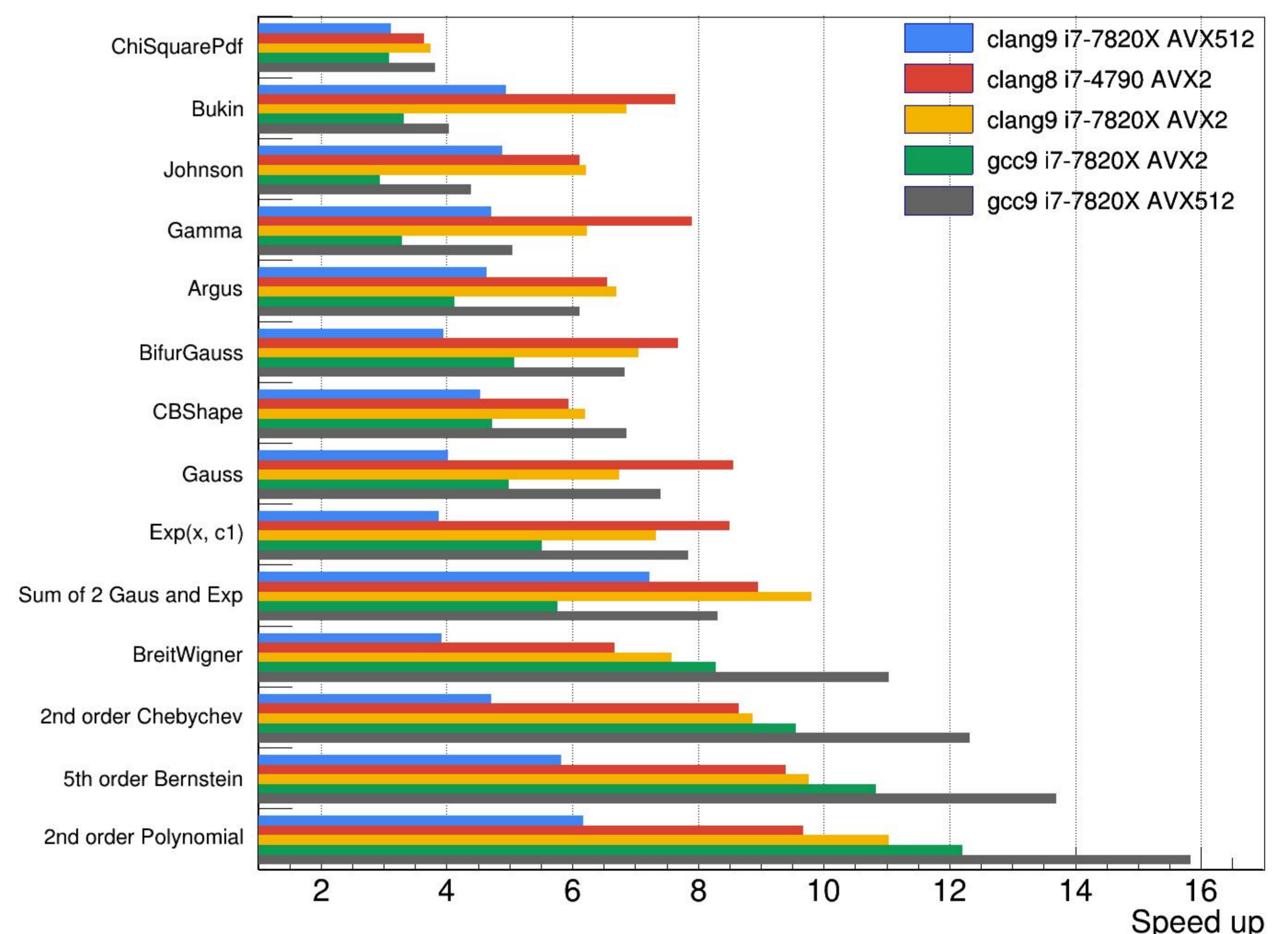
```
// In library/included code
double func(double x, double a) { return a*x*x + 1.;}
[...]
// When building fit model
RooRealVar x("x", "Observable", 2.);
RooRealVar a("a", "Parameter", 3.);
RooGenericPdf pdf("pdf", "func(x, a)", {x, a});
```

4. Faster PDF Computations

ROOT 6.20

- In ROOT 6.20, RooFit can compute likelihoods of **large batches of events in one pass**
 - Higher cache locality, fewer function calls
 - Speed up of 2x - 3.5x
- New data structures, simpler code + VDT math functions [3] allow for auto vectorisation
- **Combined speed up 4x - 16x**

Compute a log-likelihood: batched & vectorised vs. Classic RooFit



- **Plans:**
 - Better data structures & vectorisation for binned fits, without needing to change existing models
 - Vectorisation without recompilation:
 - Collect computation kernels in small library
 - Compile for SSE, AVX, AVX2, AVX512 ...
 - Load dynamically depending on hardware
 - Parallel loops, abstraction libraries for GPU offloading

References

- [1] W. Verkerke and D. Kirkby, 2003, *The RooFit toolkit for data modeling*, [arXiv:physics/0306116](https://arxiv.org/abs/physics/0306116)
- [2] ATLAS Collaboration, 2015, *Search for the $H \rightarrow b\bar{b}$ decay of the Standard Model Higgs boson in associated $(W/Z)H$ production with the ATLAS detector*, [JHEP01\(2015\)069](https://arxiv.org/abs/1506.02501)
- [3] D. Piparo, V. Innocente and T. Hauth, 2014, *Speeding up HEP experiment software with a library of fast and auto-vectorisable mathematical functions*, [J.Phys.:Conf.Ser. 513 052027](https://arxiv.org/abs/1312.5207)

stephan.hageboeck@cern.ch