GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

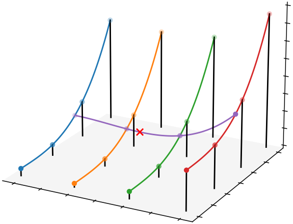
University of Glasgow

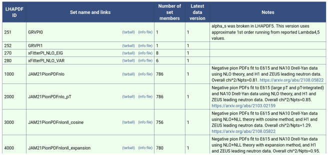
Performance Improvements in LHAPDF

Max Knobbe
23. MCnet Meeting

(Very) brief Introduction

- LHAPDF-6 is a general purpose C++ interpolator, initially released in 2013
  → Mostly used for PDF+α_s interpolations
- Interfaced to core MCnet software projects (Herwig, Pythia, Sherpa, ...)
- Today (December 6, 2021) 1153 different PDF sets available via
  https://lhapdf.hepforge.org/pdfsets
  → Interpolations grids provided by fitting groups (CTEQ, NNPDF, MMHT, ...)
  → LHAPDF interpolates between these grids/knots

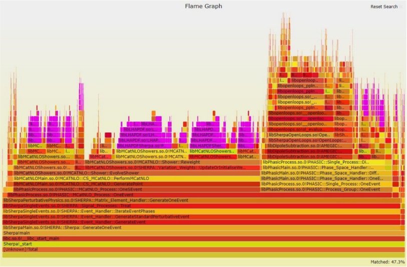




LHAPDF ID	Set name and links			Number of set members	Latest data version	Notes
251	GRVPI0	(tarball)	(info file)	1	1	alpha_s was broken in LHAPDF5. This version uses approximate 1st order running from reported Lambda4,5 values.
252	GRVPI1	(tarball)	(info file)	1	1	
270	xFitterPI_NLO_EIG	(tarball)	(info file)	8	1	
280	xFitterPI_NLO_VAR	(tarball)	(info file)	6	1	
1000	JAM21PionPDFnlo	(tarball)	(info file)	786	1	Negative pion PDFs fit to E615 and NA10 Drell-Yan data using NLO theory, and H1 and ZEUS leading neutron data. Overall chi^2/Npts=0.81. https://arxiv.org/abs/2108.05822
2000	JAM21PionPDFnlo_pT	(tarball)	(info file)	786	1	Negative pion PDFs fit to E615 (large pT and pT-integrated) and NA10 Drell-Yan data using NLO theory, and H1 and ZEUS leading neutron data. Overall chi^2/Npts=0.85. https://arxiv.org/abs/2103.02159
3000	JAM21PionPDFnlonll_cosine	(tarball)	(info file)	756	1	Negative pion PDFs fit to E615 and NA10 Drell-Yan data using NLO+NLL theory with cosine method, and H1 and ZEUS leading neutron data. Overall chi^2/Npts=1.29. https://arxiv.org/abs/2108.05822
4000	JAM21PionPDFnlonll_expansion	(tarball)	(info file)	780	1	Negative pion PDFs fit to E615 and NA10 Drell-Yan data using NLO+NLL theory with expansion method, and H1 and ZEUS leading neutron data. Overall chi^2/Npts=0.95.

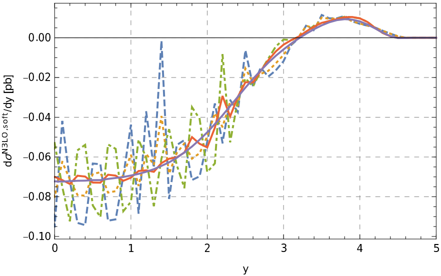
Room for improvement / Issues with the current LHAPDF version

[Tim Martin at SwiftHep/ExcaliburHep workshop]


Flame Graph
Reset Search
libSherpaPerturbativePhysics.so.0!SHERPA::Matrix_Element_Handler::GenerateOneEvent
libSherpaSingleEvents.so.0!SHERPA::Event_Handler::GenerateEvent
libSherpaMain.so.0!SHERPA::Sherpa::GenerateOneEvent
Sherpa!main
Sherpa!_start
[Unknown]!Total
Matched: 47.3%

Identified shortcomings / Outline

1. **Interpolation Speed**
   → Known problem, HL-LHC requires too much CPU hours with current software
2. **Non-physical side effects in high-order computations**
   → New interpolation strategy for special computations
3. **Modern/Heterogeneous Systems**
   → Currently no support for e.g. GPU computations



[1710.03016]

1. Reduce execution speed / changes in LHAPDF-6.4.0

LHAPDF workflow – what is using up the computing time?

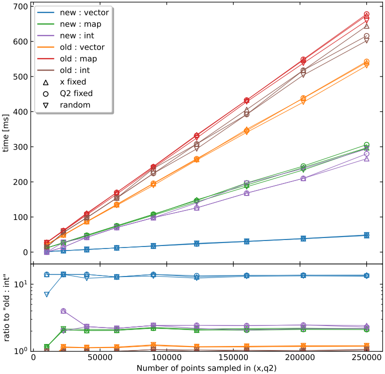
1. Find the right grid / knots
   Grids allow for maximal flexibility (discontinuous. grids, different grids for each particle)
   → 3 (binary) searches
2. Compute new anchor points
3. Perform final interpolation

Significant Changes:

- New memory layout
  → Remove unused freedom in grids
  → *S*mart memory layout
  → Single binary search
- Cache common computation if multiple pid's requested
- Reduce number of transcendental calls

→ Significant speedup of factor 3-10

Released in LHAPDF-6.4.0



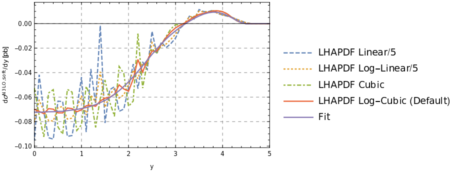
2. LHAPDF for higher-order calculations

Observed in Dulat, Mistlberger, Pelloni, soft-virtual term of rapidity contribution in the soft limit

$$\frac{d\sigma_{gg}}{dY} = \frac{\sigma_0}{9v^2}\left(\frac{\alpha_s}{\pi}\right)^5 \int_\tau^1 dz\, f_g\left(\sqrt{\frac{\tau}{zY}}\right) f_g\left(\sqrt{\frac{\tau Y}{z}}\right)\Bigg\{1124.31\,\delta(1-z) + 1466.48\left[\frac{1}{1-z}\right]_+ \\ - 6062.09\left[\frac{\log(1-z)}{1-z}\right]_+ + 7116.02\left[\frac{\log^2(1-z)}{1-z}\right]_+ \\ - 1824.36\left[\frac{\log^3(1-z)}{1-z}\right]_+ - 230\left[\frac{\log^4(1-z)}{1-z}\right]_+ + 216\left[\frac{\log^5(1-z)}{1-z}\right]_+\Bigg\}. \tag{5.1}$$

Problem:

- Higher order terms seem to be sensitive to interpolation polynomial
- Introduces oscillations in differential cross-section



[1710.03016]

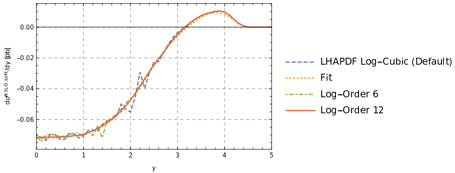
2. LHAPDF for higher-order calculations

Observed in Dulat, Mistlberger, Pelloni, soft-virtual term of rapidity contribution in the soft limit

$$\frac{d\sigma_{gg}}{dY} = \frac{\sigma_0}{9v^2}\left(\frac{\alpha_s}{\pi}\right)^5 \int_\tau^1 dz f_g\left(\sqrt{\frac{\tau}{zY}}\right) f_g\left(\sqrt{\frac{\tau Y}{z}}\right) \Bigg\{1124.31\,\delta(1-z) + 1466.48\left[\frac{1}{1-z}\right]_+ - 6062.09\left[\frac{\log(1-z)}{1-z}\right]_+ + 7116.02\left[\frac{\log^2(1-z)}{1-z}\right]_+ - 1824.36\left[\frac{\log^3(1-z)}{1-z}\right]_+ - 230\left[\frac{\log^4(1-z)}{1-z}\right]_+ + 216\left[\frac{\log^5(1-z)}{1-z}\right]_+ \Bigg\}. \qquad (5.1)$$

Solution: higher order Polynomials (Lagrange)
Applied in

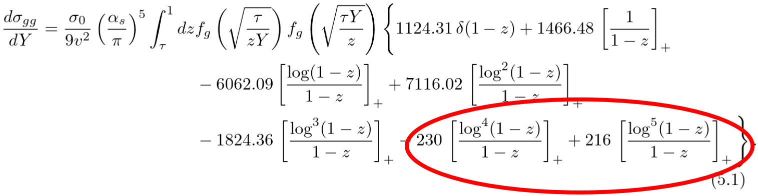
1. Global fit:
   → definitely continuous, smooth
   → potential numerical problems
   (Runge's phenomenon, interpolation errors)
2. Local fit:
   → continuation of LHAPDF-approach
   → relatively simple implementation
   → continuous, not smooth



[1710.03016]

2. LHAPDF for higher-order calculations

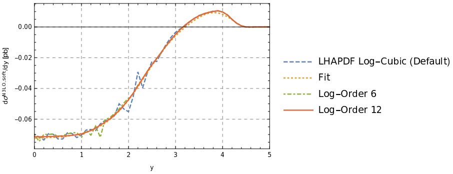
Observed in Dulat, Mistlberger, Pelloni, soft-virtual term of rapidity contribution in the soft limit



$$\frac{d\sigma_{gg}}{dY} = \frac{\sigma_0}{9v^2}\left(\frac{\alpha_s}{\pi}\right)^5 \int_\tau^1 dz\, f_g\left(\sqrt{\frac{\tau}{zY}}\right) f_g\left(\sqrt{\frac{\tau Y}{z}}\right)\Bigg\{1124.31\,\delta(1-z) + 1466.48\left[\frac{1}{1-z}\right]_+ - 6062.09\left[\frac{\log(1-z)}{1-z}\right]_+ + 7116.02\left[\frac{\log^2(1-z)}{1-z}\right]_+ - 1824.36\left[\frac{\log^3(1-z)}{1-z}\right]_+ - 230\left[\frac{\log^4(1-z)}{1-z}\right]_+ + 216\left[\frac{\log^5(1-z)}{1-z}\right]_+\Bigg\}. \tag{5.1}$$

Solution: higher order Polynomials (Lagrange)
Applied in

1. Global fit:
   → definitely continuous, smooth
   → potential numerical problems
   (Runge's phenomenon, interpolation errors)
2. **Local fit:**
   → **continuation of LHAPDF-approach**
   → **relatively simple implementation**
   → **continuous, not smooth**



[1710.03016]

3. Modern Architectures – why do we have to care?

Machine Learning Efforts:

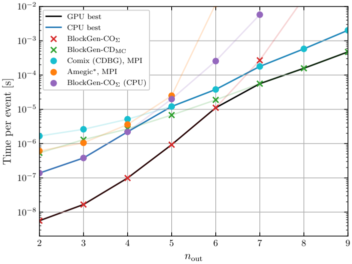
- Almost every part of MC tool-chain is investigated
  → Naturally, GPU-based computations

GPU Monte Carlo's
Some, (recent) developments in chronological order:

- Tess [Giele, Stavenga, Winter]
  C++, Cuda
- BlockGen [Bothmann, Giele, Höche, Isaacson, MK]
  C++, Cuda, (Kokkos)
- MadFlow [Carrazza, Cruz-Martinez, Rossi, Zaro]
  Python, Tensorflow
- MadGraph-GPU [Valassi, Roiser, Mattelaer, Hageboeck]
  C++, Cuda, (Alpaka, Kokkos, Sycl)

New Clusters/Supercomputer:

- Aurora (2×Intel CPU + 6×Intel GPU per note)
- More to come..?

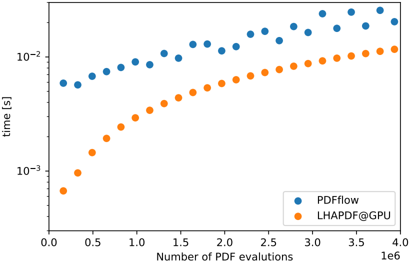


Argonne NATIONAL LABORATORY
U.S. DEPARTMENT OF ENERGY
intel
Hewlett Packard Enterprise
Aurora

[Aurora]

3. LHAPDF@Heterogenous

- Kokkos code, build on top of LHAPDF
  → receiving LHAPDF updates
  → minimal code doubling
- Can provide PDF's to aforementioned projects
- only competitor: PDFFlow
  [Carrazza, Cruz-Martinez, Rossi]
  → Probably more natural for ML



Possible application in classical frameworks:

- Many variations at once?
- All-pid version for single PS-point?
- ...
- Suggestions, ideas, requests?

→ **Not released yet, more testing needed**

Conclusion / Outlook

Conclusion:

- Significant interpolation speedup, Sherpa+LHAPDF benchmark pending
- Fixed higher order computations problems
- New interpolator, suitable for modern systems

Possible future projects:

- Higher Resolution?
- TMD's, much easier to include with generalised memory/interpolator structure

Conclusion / Outlook

Conclusion:

- Significant interpolation speedup, Sherpa+LHAPDF benchmark pending
- Fixed higher order computations problems
- New interpolator, suitable for modern systems

Possible future projects:

- Higher Resolution?
- TMD's, much easier to include with generalised memory/interpolator structure

Thank you

MCnet: for giving me the opportunity
You: for your attention!