PDFFlow

Parton distribution functions on GPU

Stefano Carrazza, Juan Cruz-Martinez, Marco Rossi

14^{th} November 2023

Event generators' and N(n)LO codes' acceleration, CERN
arXiv:2009.06635 [hep-ph]
GitHub repository

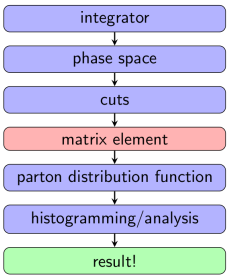
PDFs

Parton distribution functions in Monte Carlo generators

$$\sum_{a,b} \int_{x_{\min}}^{1} dx_1\, dx_2\, |\mathcal{M}_{ab}(\{p_n\})|^2\, \mathcal{J}_m^n(\{p_n\})\, f_a(x_1, Q^2) f_b(x_2, Q^2),$$

a multi-dimensional integral where:

- $|\mathcal{M}|$ is the matrix element,
- $f_i(x, Q^2)$ are Parton Distribution Functions (PDFs),
- $\{p_n\}$ phase space for n particles,
- $\mathcal{J}_m^n$ jet function for n particles to m.

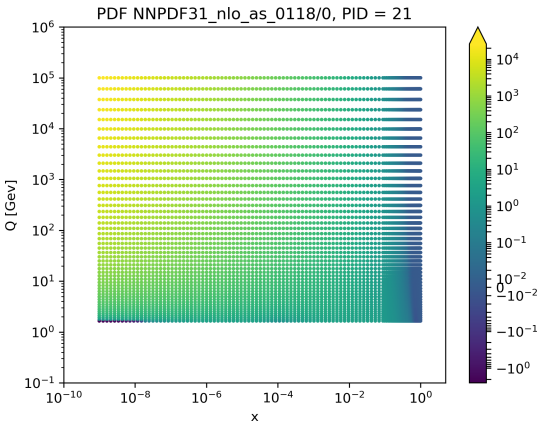


PDF evaluation

DGLAP evolution is replaced with interpolation grids in $(x, Q^2) \Rightarrow$ PDFs are interpolated.

PDF grid format

- LHAPDF6 provides official pdf sets
- A PDF set contains versions with grids for all flavors
- Only some points are measured: interpolation needed
- Interpolation in grid range: $(x, Q) \in [10^{-9}, 1] \times [1.65, 10^5]$
- Need to extrapolate outside grid range



Example: PDF points from gluon NNPDF31_nlo_as_0118/0

PDF interpolation zones

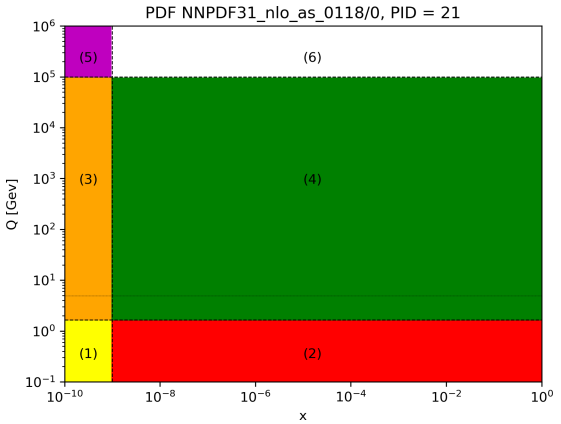
(4) Inside grid: log-bicubic interpolation

(3)-(6) Low x and high Q^2 regions: (log-)linear extrapolation from the two nearest grid knots

(2) Low Q^2 region: anomalous dimension interpolation between $\gamma(Q_{min})$ and 1 $\rightarrow$ $xf(x,Q^2) = xf(x,Q_{min})\,(Q^2/Q^2_{min})^{\gamma(Q^2_{min})}$

(1) Low x, Low Q^2 region: extrapolation

(5) Low x, High Q^2 region: extrapolation

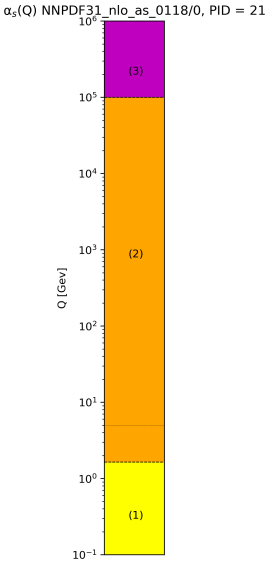


α_s interpolation zones

(1) Low Q region: $\alpha_s(Q) = \alpha_s(Q_{min})\,(Q^2/Q^2_{min})^{\frac{\partial \log \alpha_s}{\partial Q^2}\Big|_{Q^2=Q^2_{min}}}$ for $Q < Q_{min}$

(2) Inside α_s grid: cubic interpolation

(3) High Q region: $\alpha_s(Q) = \alpha_s(Q_{max})$ for $Q > Q_{max}$

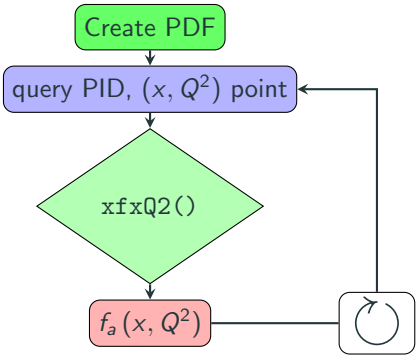


Technical Implementation

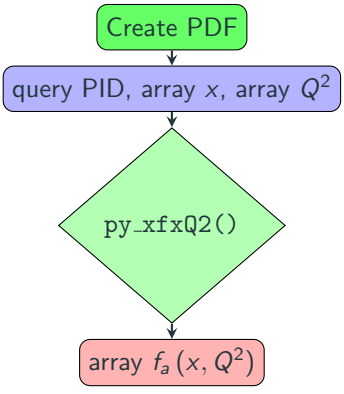
Parallel vs Sequential Queries

Mimic the `LHAPDF` interpolation methods, parallelize them

`LHAPDF6` algorithm:



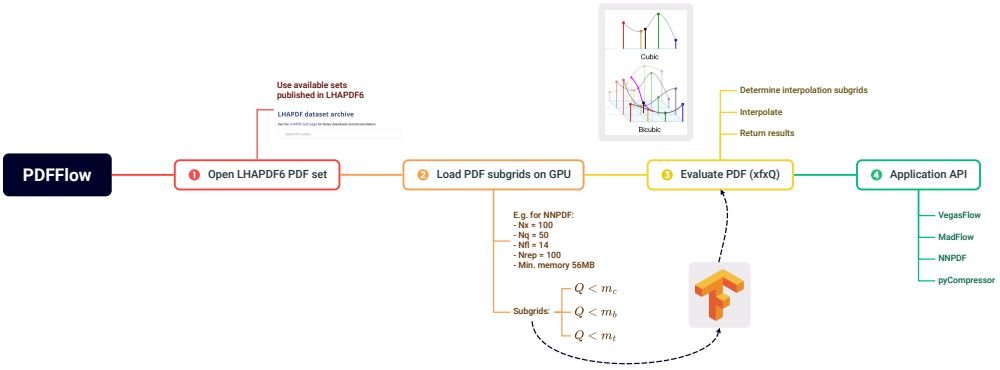
`PDFFlow` algorithm:



! Query points are independent

✓ `PDFFlow` benefits from hardware acceleration (multithreading CPU, GPU)

PDFFlow pipeline



TensorFlow library

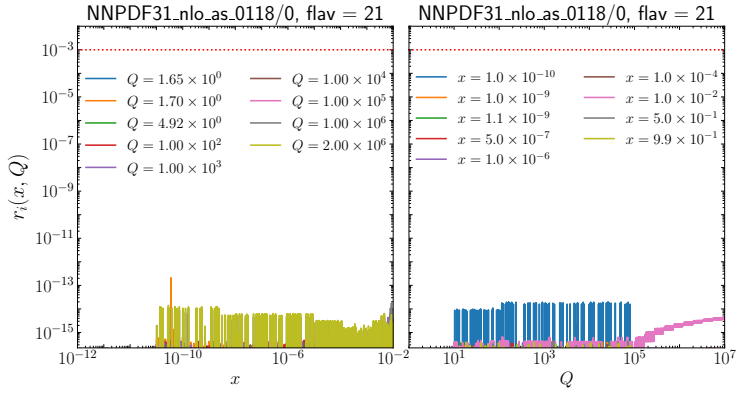
- End-to-end open source platform for machine learning (by Google)
- Rich python API library: `tf v2.x` compatibility
- Automatic multi-threading CPU and GPU management
- No need to go through specific GPU code (CUDA, OpenCL)
- *Graph* mode

Accuracy benchmark

Absolute relative difference of `PDFFlow` against `lhapdf`:

$r_i(x, Q) = \frac{|f_{lhapdf} - f_{pdfflow}|}{|f_{lhapdf}| + \epsilon}$, where regulator ϵ avoids division by 0.

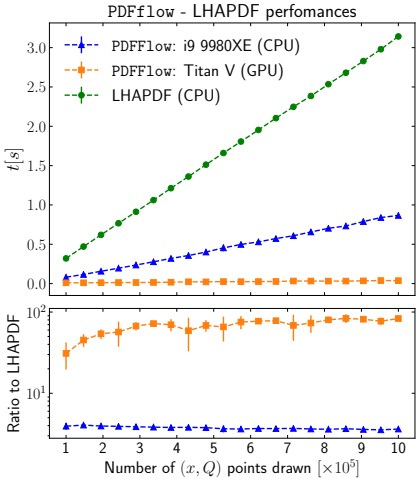
Acceptable error threshold is 10^{-3} according to LHAPDF6 paper.



Performance benchmark

PDF set: NNPDF31_nlo_as_0118, PDF ID: 0. Parton: gluon

- Execution time per number of query points
- CPU ratio is flat, value of 3x – 4x
- GPU ratio improves with the number of queries, peaks at 100x



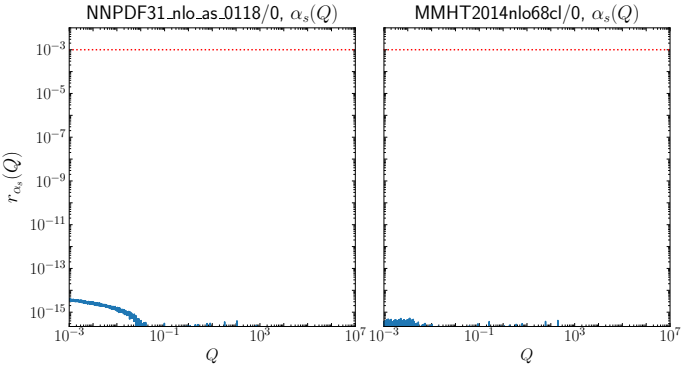
Strong running coupling

The strong coupling $\alpha_s(Q)$ evolution is driven by renormalization group equation

Modern PDF sets come with a grid for α_s points in Q space

Mimic the `PDFFlow` algorithm in 1 dimension

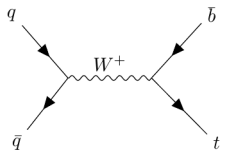
✓ Exact matching between `PDFFlow` and `LHAPDF` interpolations



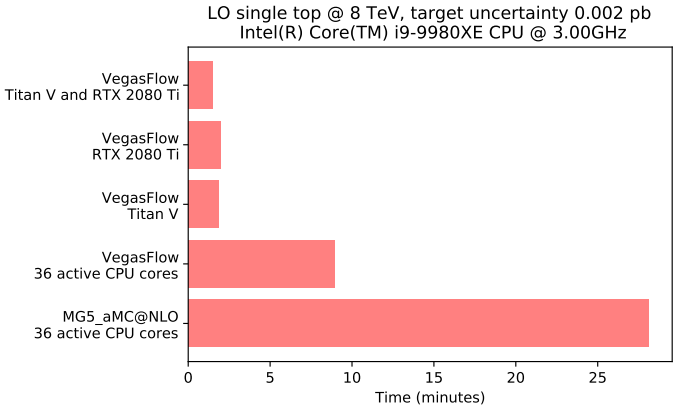
Application example - Single top production at LO

Combine `PDFFlow` and `VegasFlow` (MC integrator, 10.1016/j.cpc.2020.107376)

Speed comparison for `PDFFlow` + `VegasFlow` against MG5_aMC@NLO

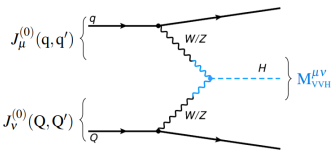


Born diagram for $q\bar{q} \to \bar{b}t$



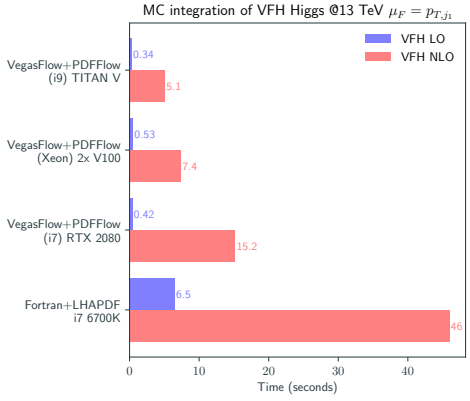
Single top run timings

Application example - VBF Higgs production at NLO



Born diagram for $qQ \to qQH$

hep-ph/arXiv:1807.07908



- ✓ Best speed-up at LO: 19x
- ✓ Best speed-up at NLO: 9x

Time per iteration
(C) consumer-grade
(P) professional-grade hardware
CPU implementation: `LHAPDF` + Fortran code
GPU implementation: `PDFFlow` + `VegasFlow`

Basic usage

PDFFlow installation

How to install PDFFlow?

pip install pdfflow

How to install LHAPDF tools?

pip install lhapdf-management

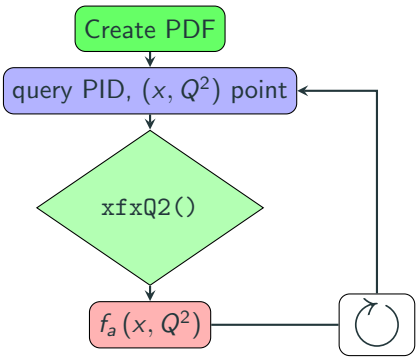
The `pdfflow` is a pure Python package and therefore you can pip install without having to worry about complicated LHAPDF installations.

For installing the PDF sets you can use the lhapdf-management script.

- Documentation: `https://pdfflow.readthedocs.io`
- PDFFlow code: `https://github.com/N3PDF/pdfflow`
- LHAPDF management tools in PyPI: `https://github.com/scarlehoff/lhapdf_management`

Usage example in Python

LHAPDF6 algorithm:



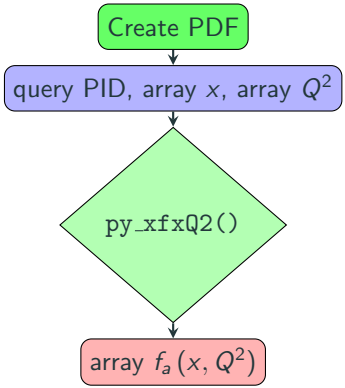

```python
import numpy as np
from lhapdf import mkPDF

# allocate pdf, x and q2 grid
p = mkPDF(pdfname, DIRNAME)
xs = np.logspace(-10,0,100000)
q2s = np.logspace(0,6,100000)

# sequencial query
for x in xs:
   for q2 in q2s:
      p.xfxQ2(21, x, q2)
```

Usage example in Python

PDFFlow algorithm:




```python
import os
import numpy as np
from pdfflow.pflow import mkPDF
from lhapdf_management import pdf_update, pdf_install

PDFNAME = "NNPDF40_nnlo_as_01180"
DIRNAME = os.environ["LHAPDF_DATA_PATH"]

pdf_update() # download the pdfsets.index
pdf_install(PDFNAME) # and install the PDF

p = mkPDF(f"{PDFNAME}/0", DIRNAME)
p.trace()

nevents = int(1e6)
xs = np.logspace(-10, 0, nevents)
q2s = np.logspace(0, 6, nevents)**2

result = p.py_xfxQ2(21, xs, q2s)
print(result.shape) # should be (nevents,)
```

Usage example using C-API

- Possibility to call PDFFlow's API using a C wrapper interface.
- GPU-CPU computations are automatically performed by PDFFlow.

```c
// This file is part of PDFFlow
#include <stdio.h>
#include "pdfflow/pdfflow.h"

int main() {
    // load pdf
    mkpdf("NNPDF31_nlo_as_0118/0", "/usr/share/lhapdf/LHAPDF/");

    // test xfxq2 and alphasq2
    const double x = 0.1, q2 = 1.65;
    for (int fl=-5; fl <=5; fl++)
        printf("flv=%d - xfx = %f\n", fl, xfxq2(fl, x, q2));
    printf("alphas(q2=%f) = %f\n", q2, alphasq2(q2));

    return 0;
}
```

Usage example using C-API with Fortran90

- Possibility to call PDFFlow's API using the Fortran wrapper interface.
- GPU-CPU computations are automatically performed by PDFFlow.

```
! This file is part of pdfflow
program example

   use, intrinsic :: ISO_C_BINDING, only: C_ptr, c_char
   implicit none

   integer, parameter :: dp = kind(1.d0)

   integer :: pid
   real(dp) :: alpha_s, q2, x, xfx

   character(kind=c_char, len=21) :: ss = "NNPDF31_nlo_as_0118/0"
   character(kind=c_char, len=24) :: pp = "/usr/share/lhapdf/LHAPDF/"

   call mkPDF(ss, pp)

   q2 = 1.65
   x = 0.1
   call alphasq2(q2, alpha_s)

   write(*, '(A, F7.5)')"The value of alpha_s is: ", alpha_s

   write(*, fmt=100)"Value of x*f(x) at q2=", q2, " x=",x
   do pid = -5, 5
      call xfxq2(pid, x, q2, xfx)
      write(*, fmt=200)"Flavour: ", pid, " value: ", xfx
   enddo

100 format (A, F6.2, A, F4.2)
200 format ("    ", A, I0, A, F10.7)

end program
```

Outlook

Summary

We presented `PDFFlow`, the first GPU port for PDF interpolation:

- benchmark against LHADPF6 for performance and accuracy, achieving a notable speedup via parallelized and `TensorFlow` optimized algorithm
- implemented α_s strong running coupling interpolation
- provided application examples: single-top at LO, VFH at NLO

Further ideas:

1. Interoperability, for example using DLPack, in order to access and share GPU objects across external libraries (MC software already using GPU).
2. Interpolation algorithm optimization.
3. Caching algorithms.