technische universität dortmund

Funded by DFG Deutsche Forschungsgemeinschaft German Research Foundation

E5a/ Physik

chromo

A novel interface to hadronic models

Hans Dembinski[1], Anatoli Fedynitch[2], Anton Prosekin[2]
[1]TU Dortmund, [2]Academia Sinica, Taipei, Taiwan

Astrophysics Workshop on Numerical Multimessenger Modeling, Bochum, Feb 2023

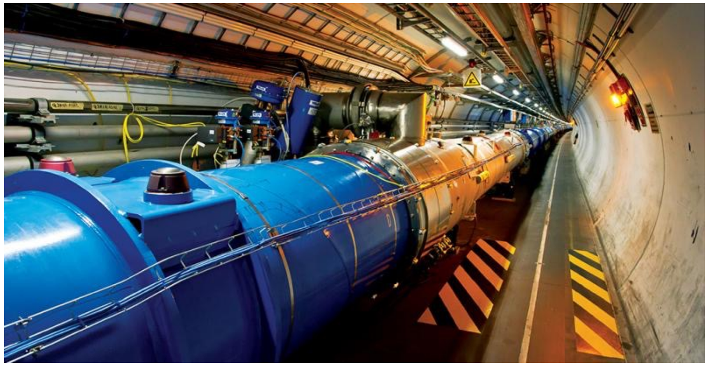
Quick facts

- Python frontend to generators written in Fortran & C++
  - DPMJet-III*, PhoJet*, EPOS-LHC, Pythia-6.4, Pythia-8.3, QGSJet*, QGSJet-II*, SIBYLL*, SOPHIA, UrQMD 3.4 (* = several versions)
  - Use as Python library or command-line interface
- Open source development on Github
  - https://github.com/impy-project/chromo
  - BSD 3-clause license, contributions welcome
- Main authors
  - Anatoli Fedynitch (project lead), Hans Dembinski, Anton Prosekin
- Beta status
  - Authors already use it for science projects
  - pip install chromo not yet ready, but coming very soon!
  - For installation from source, see README.md

Introduction

- Applications in (astro)particle physics require simulations of particle production in interactions of photons, hadrons, and nuclei
  - Cosmic ray propagation through galaxy
  - Air showers
  - Min-bias physics and underlying event at colliders
- No standard event generator (yet)
  - Common: compute result with input from several generators to estimate systematic uncertainty
- Event generators have no standard interface
  - Varying event representations, particle IDs, and data structures
- Most generators implemented in Fortran 77; modern generators in C++
- Majority of scientific computing, education, and data science have moved to Python ecosystem
- Chromo (formerly named impy) provides
  - Standard Python interface over generators
  - Taps into rich Python ecosystem for extra features
  - CLI to generate HepMC & ROOT output or SVG images

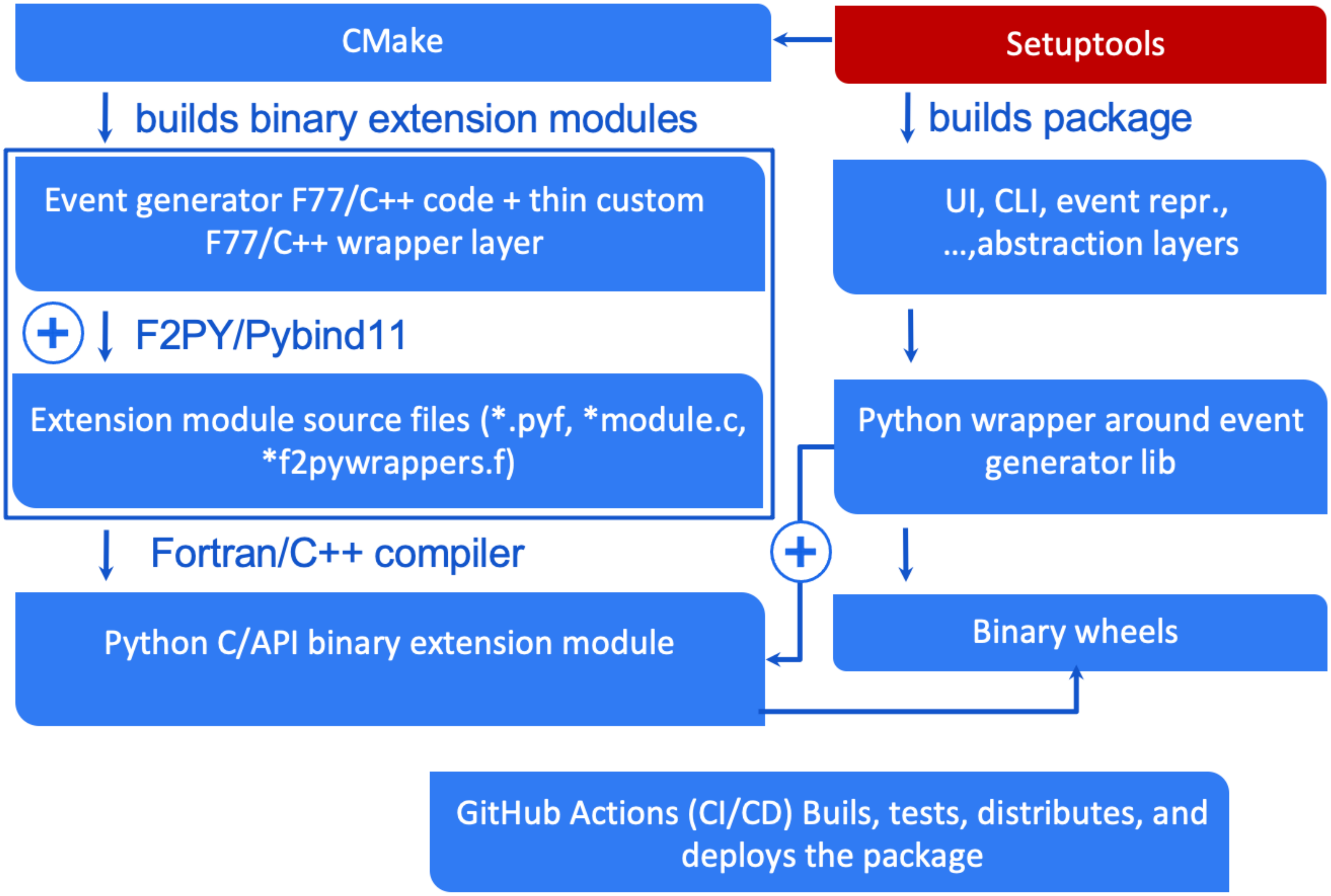
Artist impression of air shower



Supported event generators

- DPMJET-III 3.0.6 & PHOJET 1.12-35
- DPMJET-III 19.1 & PHOJET 19.1
- DPMJET-III 19.3 & PHOJET 19.3
- EPOS-LHC
- PYTHIA 6.4
- PYTHIA 8.3
- QGSJet-01
- QGSJet-II-03
- QGSJet-II-04
- SIBYLL-2.1
- SIBYLL-2.3
- SIBYLL-2.3c
- SIBYLL-2.3d
- SOPHIA 2.0
- UrQMD 3.4

Technical concept



Code example

Import libraries

```
import chromo
from chromo.constants import TeV
import numpy as np
import boost_histogram as bh
import matplotlib.pyplot as plt
```

Prepare histograms (our choice boost.histogrgam)

```
pid_categories = bh.axis.IntCategory([2212, 111, 211, -211])
hist_xf  = bh.Histogram(pid_categories, bh.axis.Regular(50, -1, 1))
hist_eta = bh.Histogram(pid_categories, bh.axis.Regular(50, -7, 7))
```

Initialize an event generator instance

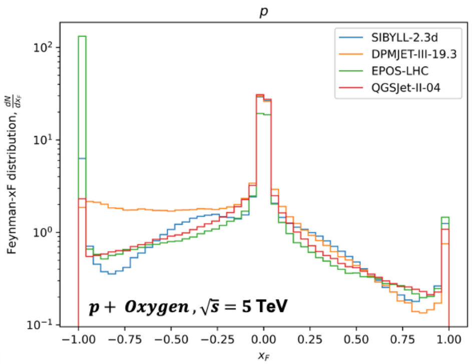
```
kinematics = chromo.kinematics.CenterOfMass(5 * TeV, "proton", "O16")
event_generator = chromo.models.Sibyll23d(kinematics)
```

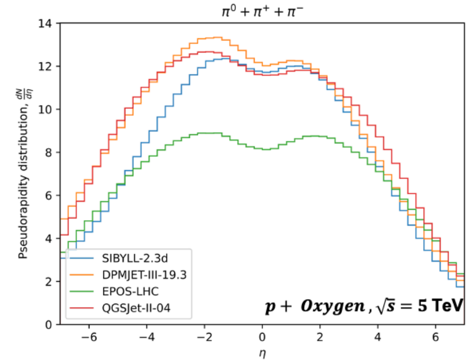
Generate 10000 events

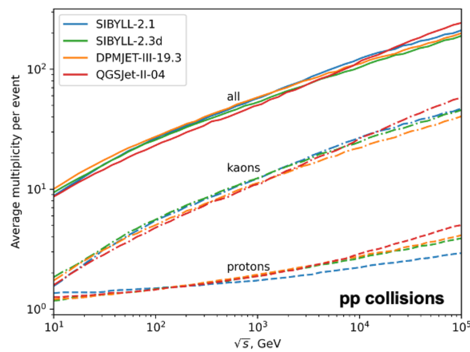
```
for event in event_generator(10000):
    event = event.final_state()
    hist_xf.fill(event.pid, event.xf)   # Feynman-x distributions
    hist_eta.fill(event.pid, event.eta) # Pseudorapidity distributions
```

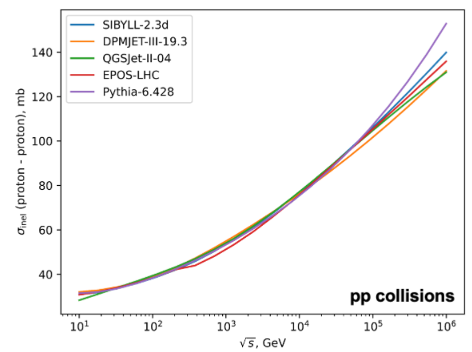
Plot Feynman-x distribution for protons

```
xf_grid = hist_xf.axes[1]
prot_hist = hist_xf.values(True)[0, 1:-1]
prot_xf_dist = prot_hist / 10000 / xf_grid.widths
plt.stairs(prot_xf_dist, xf_grid.edges)
```



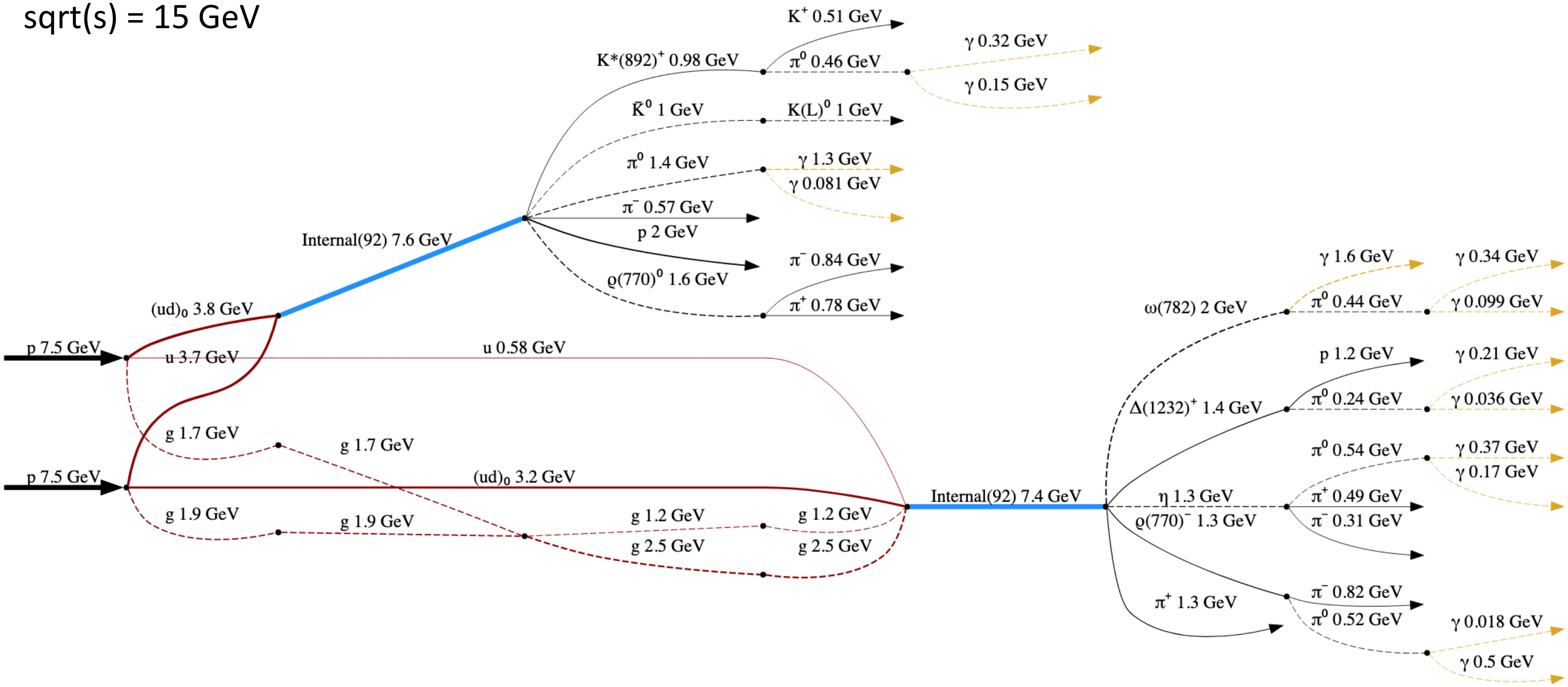






Event visualization (via pyhepmc)

Pythia-6.4 event
sqrt(s) = 15 GeV



- pyhepmc is Scikit-HEP library and frontend to HepMC3 C++ library
- Optional event visualization via Graphviz library

Output in HepMC3 format (via pyhepmc)

HepMC: Lingua franca for simulation software used at CERN

SIBYLL-2.1, pp, sqrt(s) = 20 GeV

```
HepMC::Version 3.02.05
HepMC::Asciiv3-START_EVENT_LISTING
T SIBYLL\|2.1\|
E 0 7 23
U GEV MM
P 1 0 2224 -8.9205041527748108e-02 1.3491769134998322e-01 2.0344371795654297e+00 2.3833706378936768e+00 1.2309999465942383e+00 2
P 2 0 111 8.9463070034980774e-02 -4.4863110780715942e-01 3.4179518222808838e+00 3.4519975185394287e+00 1.3496999442577362e-01 2
P 3 0 331 1.4990800619125366e-01 4.6003237366676331e-01 2.0358951091766357e+00 2.3069748878479004e+00 9.5749998092651367e-01 2
P 4 0 221 -1.6207817196846008e-01 -2.4423867464065552e-01 1.9545348882675171e+00 2.0511729717254639e+00 5.4879999160766602e-01 2
P 5 0 -211 -5.2947159856557846e-02 2.5346320867538452e-01 -5.1904711872339249e-02 2.9869863390922546e-01 1.3956999778747559e-01 1
P 6 0 111 -2.4904966354370117e-02 3.6575528979301453e-01 -1.8918764591217041e+00 1.9317893981933594e+00 1.3496999442577362e-01 2
P 7 0 2212 8.9764386415481567e-02 -5.2129906415939331e-01 -7.4990553855895996e+00 7.5760145187377930e+00 9.3826997280120850e-01 1
P 8 1 2212 -1.1027524620294571e-01 9.2852763831615448e-02 1.1708897352218628e+00 1.5073537826538086e+00 9.3826997280120850e-01 1
P 9 1 211 2.1069953218102455e-02 4.2065303772687912e-02 8.6355310678482056e-01 8.7602353096008301e-01 1.3956999778747559e-01 1
P 10 2 22 -1.5905208885669708e-02 -2.5956141948699951e-01 1.9417071342468262e+00 1.9595654010772705e+00 0.0000000000000000e+00 1
P 11 2 22 1.0539282113313675e-01 -1.8919278681278229e-01 1.4771823883056641e+00 1.4933792352676392e+00 0.0000000000000000e+00 1
P 12 3 211 1.3005101121962070e-02 9.5943860709667206e-02 2.2564361989498138e-01 2.8325849771499634e-01 1.3956999778747559e-01 1
P 13 3 -211 9.5603697001934052e-02 6.0167539864778519e-02 1.2169665843248367e-01 2.1799579262733459e-01 1.3956999778747559e-01 1
P 14 3 221 4.1300963610410690e-02 3.0392596125602722e-01 1.6885783672332764e+00 1.8057471513748169e+00 5.4879999160766602e-01 2
P 15 4 22 -1.3510279357433319e-01 4.4372059404850006e-02 1.5330873727798462e+00 1.5396685600280762e+00 0.0000000000000000e+00 1
P 16 4 22 -2.6978071779012680e-02 -2.8861477971076965e-01 4.2147985100746155e-01 5.1153844594955444e-01 0.0000000000000000e+00 1
P 17 6 22 -4.5049645006656647e-02 2.6162242889404297e-01 -1.5857362747192383e+00 1.6078042984008789e+00 0.0000000000000000e+00 1
P 18 6 22 2.0137846469879150e-02 1.0423322767019272e-01 -3.0665934085845947e-01 3.2451510429382324e-01 0.0000000000000000e+00 1
P 19 14 211 1.6239669173955917e-02 1.1180111020803452e-01 3.0710572004318237e-01 3.5670593380928040e-01 1.3956999778747559e-01 1
P 20 14 -211 -5.0198074430227280e-02 7.7754214406013489e-02 2.6761403679847717e-01 3.1666630506515503e-01 1.3956999778747559e-01 1
P 21 14 111 7.5260899960994720e-02 1.1438217759132385e-01 1.1139335632324219e+00 1.1324540376663208e+00 1.3496999442577362e-01 2
P 22 21 22 6.8888634443283081e-02 -3.4993162844330072e-03 5.4559254646301270e-01 5.5093902349472046e-01 0.0000000000000000e+00 1
P 23 21 22 6.3929148018360138e-03 1.1791287362575531e-01 5.6864660978317261e-01 5.8182567358016968e-01 0.0000000000000000e+00 1
HepMC::Asciiv3-END_EVENT_LISTING
```

Command line interface

- Interface mimics CRMC to ease transition
- Powered by Python libraries: argparse, rich
    - Progress bar with ETA, events / sec
- Generate output in HepMC format, ROOT, or generate SVG images

```
(env_impy) -bash-4.2$ chromo -m sibyll-2.1 -n 10000 -S 1000 -o root -f hey.root
┌──────────────────────────── chromo 0.3.0+dev ────────────────────────────┐
│    Model            SIBYLL-2.1                                          │
│    Projectile       p (2212)                                            │
│    Target           p (2212)                                            │
│    sqrt(s)          1000 GeV                                            │
│    Collisions       10000                                               │
│    Seed             83207495                                            │
│    Format           root                                                │
└──────────────────────────────────────────────────────────────────────────┘
```

```
    3   6.310E+06   11.00  267.28  174.01   36.76   0.125   0.140   8.687
    3   7.943E+06   11.38  275.24  178.72   37.64   0.125   0.133   8.756
    3   1.000E+07   11.77  283.28  183.48   38.54   0.125   0.126   8.819
10000/10000 ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 100% ETA 0:00:08 1271/s
(env_impy) -bash-4.2$
```

Prior work: CRMC

- CRMC: Command-line interface written in C++
  - Used by ATLAS, CMS, LHCb, NA61, TOTEM
- Currently maintained by Tanguy Pierog
  - Former lead developer Ralf Ulrich left scientific community
- Source compilation required, no binary packages
- Output in ROOT, HepMC, LHE formats
  - Heavy dependency: ROOT framework
- No direct access to event record from Python or other language
- No built-in event visualization
- Extra models only in Chromo
  - SIBYLL-(2.1, 2.3, 2.3c), SOPHIA, Pythia-8.3, UrQMD-3.4
- Models not in Chromo
  - HIJING, GHEISHA (outdated), UrQMD 1.3 (outdated)

Performance

Table prepared by Anton

impy is now chromo

	Sibyll-2.3d	DPMJet-III 19.1	QGSJETII-04	EPOS LHC
IMPY: pyhepmc → ASCII	1116.91 ± 31.41	474.94 ± 20.55	264.78 ± 21.61	35.99 ± 2.70
IMPY: no output	6228.12 ± 281.10	1554.85 ± 272.18	382.32 ± 18.69	43.23 ± 3.13

	Sibyll-2.3d	DPMJet-III 19.1	QGSJETII-04	EPOS LHC
CRMC: HepMC3 → ASCII	30.24 ± 6.76	16.57 ± 3.37	21.09 ± 2.99	11.14 ± 1.14
CRMC: no output	2602.06 ± 166.96	914.84 ± 43.42	305.99 ± 29.15	38.00 ± 6.40

in events/sec

- Python libraries can be very fast
  - Use Python as glue between fast compiled libraries written in Fortran/C++
  - Pass large chunks of data efficiently (as "pointers") as Numpy arrays
- pp collisions at sqrt(s) = 7 TeV: Chromo factor **2 to 35 faster**
  - Event generation faster in chromo (better optimization?)
  - HEPMC output generation faster in pyhepmc
- Caveat: output not exactly identical
  - Postprocessing of raw generator output may be different
  - But small expected impact on performance

Summary

chromo

- **Easy** comparisons between a wide variety of event generators
- **Easy** visualization and manipulation of events using rich Python ecosystem
- **Easy** installation***:** automated packaging and distribution of binaries via PyPI for Linux, MacOS, and Windows (* = coming very soon)
  - Excellent choice for application and education in (astro)particle physics
- **Easy change** of simulation settings (on-the-fly)
- **Command-line interface**
  - Mimics CRMC to ease transition
- **Fast** thin wrapper, processing optimized
  - Much faster than CRMC
- **Output in standard formats**
  - HepMC (via pyhepmc), optionally gzipped
  - Root (via uproot)
  - SVG images
- Used in cosmic ray community, high-energy neutrino physics (IceCube), and HEP community (LHCb)
- **To-do**
  - Finish packaging (supporting Windows is challenging)
  - Add LHE output (via pyhepmc)
  - Add more event generators, e.g. EPOS 4.0