European Strategy for Particle Physics

PARALLEL 9 / **COMPUTING**

Experiment-oriented HEP Software: past, present, future

Eduardo Rodrigues (University of Liverpool)

23-27 JUNE 2025 Lido di Venezia

European Strategy for Particle Physics

PARALLEL 9 / COMPU

Or the impossible software review talk ☺ !

- No attempt to be comprehensive, more focus on HEP than Particle Physics at large
- No discussion of (related & very important) Computing aspects, Quantum Computing

Thank you to CERN EP-SFT, HSF, ALICE, CMS for inputs and feedback!

23-27 JUNE 2025 Lido di Venezia

Software in HEP is ubiquitous … some present trigger systems are software-only!

- **Estimation of > 50M lines of C++ & Python & Fortran (mostly) code**
  - **It all started a while back …**

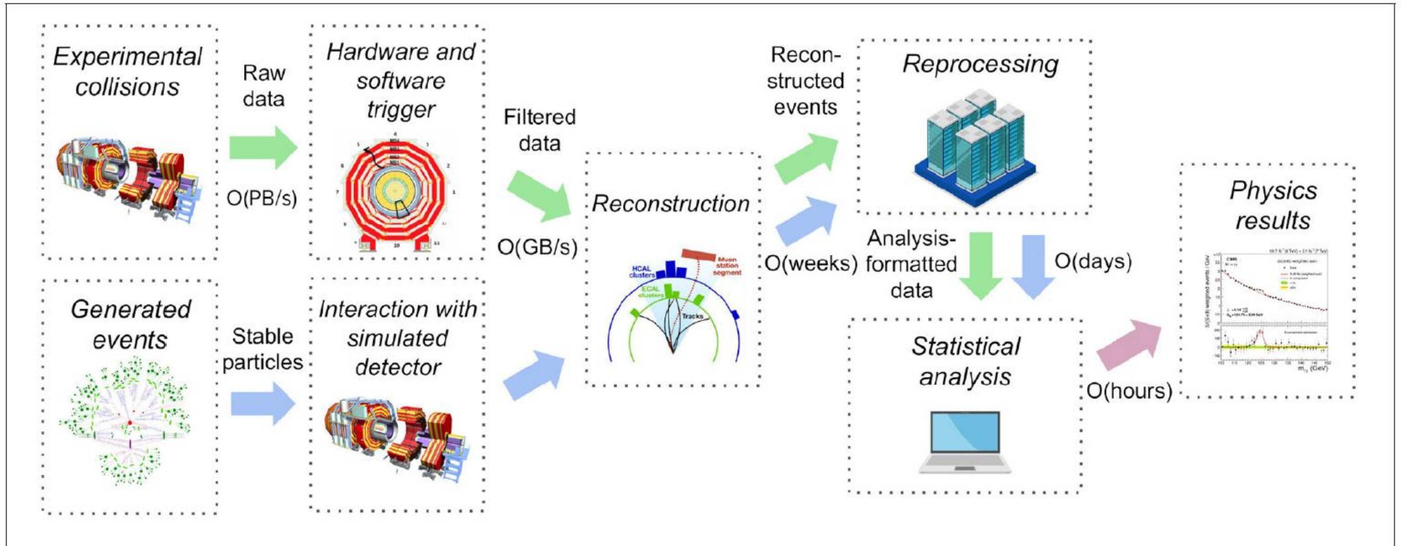


FIGURE 1 | A simplified diagram illustrating typical stages in experimental particle physics data analyses. After data acquisition that is using a multi-tiered trigger filtering step, the experimental collision data are further reduced in computing processes before they are ready for physics analyses. Events generated following theoretical models undergo a detector simulation step and are subsequently subject to the same reconstruction and processing steps as the collision data. The individual analysts then compare collision and simulated data using statistical analysis techniques. Our paper focuses mostly on the computational reproducibility challenges inherent in the last data analysis stages.

Taken from DOI:10.3389/fdata.2021.661501

Journal of Scientific Instruments (Journal of Physics E) 1969 Series 2 Volume 2

The use of computers in high energy physics experiments

D Lord and G R Macleod
Data Handling Division, CERN, Geneva, Switzerland

1.3 *Areas of computer use*

Computers are used in the planning, data acquisition and data analysis phases of high energy physics experiments, as well as in control functions for accelerators (Howard 1967a, beam switchyards (Howry 1967) and bubble chambers (Simpson 1967). In planning an experiment simulation calculations can be made by using Monte-Carlo techniques (James 1968) for estimating event rates to be expected, counting rates due to background, optimum disposition of detectors and so on. Much beam optics design (Whiteside and Gardner 1963)

J. Phys. E 2 (1969) 1-9

… and critical !

DOI: 10.5281/zenodo.2413004

The Importance of Software and Computing to Particle Physics

A contribution from the High-Energy Physics Software Foundation to the European Particle Physics Strategy Update 2018-2020

ABSTRACT: In 2017 the experimental High-Energy Physics community wrote a *Roadmap for HEP Software and Computing R&D for the 2020s*[1]. This effort was organised by the HEP Software Foundation[2] (HSF) and was supported by more than 300 physicists from more than 100 institutes worldwide. It delivered a strategy outlining the most important areas in which investment is needed to ensure the success of our experimental programme. This contribution to the ESPP is an executive summary of the most critical and relevant points raised in that white paper.

Research software is critical to the future of AI-driven research

By Michelle Barker, Kim Hartley, Daniel S. Katz, Richard Littauer, Qian Zhang, Shurui Zhou, Jyoti Bhogal

August 2024

Abstract

This position paper provides a statement on the criticality of research software in artificial intelligence (AI)-driven research and makes recommendations for stakeholders on how to

DOI:10.5281/zenodo.13350747; ReSA blog

The Critical Importance of Software for HEP

Prepared by the HEP Software Foundation, with inputs from the HEP community.

Edited by:

Christina Agapopoulou[a] Claire Antel[b] Saptaparna Bhattacharya[c] Steven Gardiner[d] Krzysztof L. Genser[d] James Andrew Gooding[e] Alexander Held[f] Michel Hernandez Villanueva[g] Michel Jouvin[a] Tommaso Lari[h] Valeriia Lukashenko[i] Sudhir Malik[j] Alexander Moreno Briceño[k] Stephen Mrenna[d] Inês Ochoa[l] Joseph D. Osborn[m] Jim Pivarski[n] Alan Price[o] Eduardo Rodrigues[p] Richa Sharma[q] Nicholas Smith[d] Graeme Andrew Stewart[b] Anna Zaborowska[b] Dirk Zerwas[r] Maarten van Veghel[s]

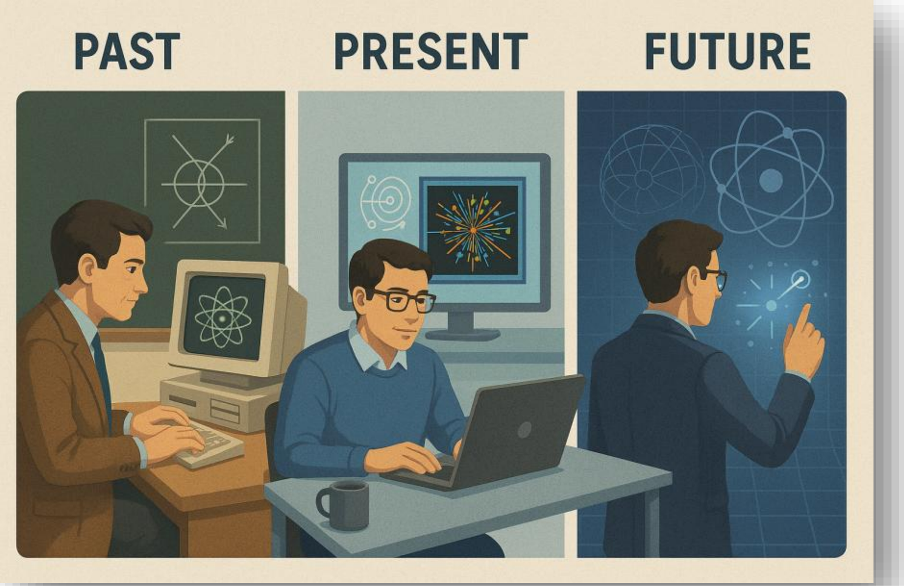
This document has been endorsed by the following experiments and communities:

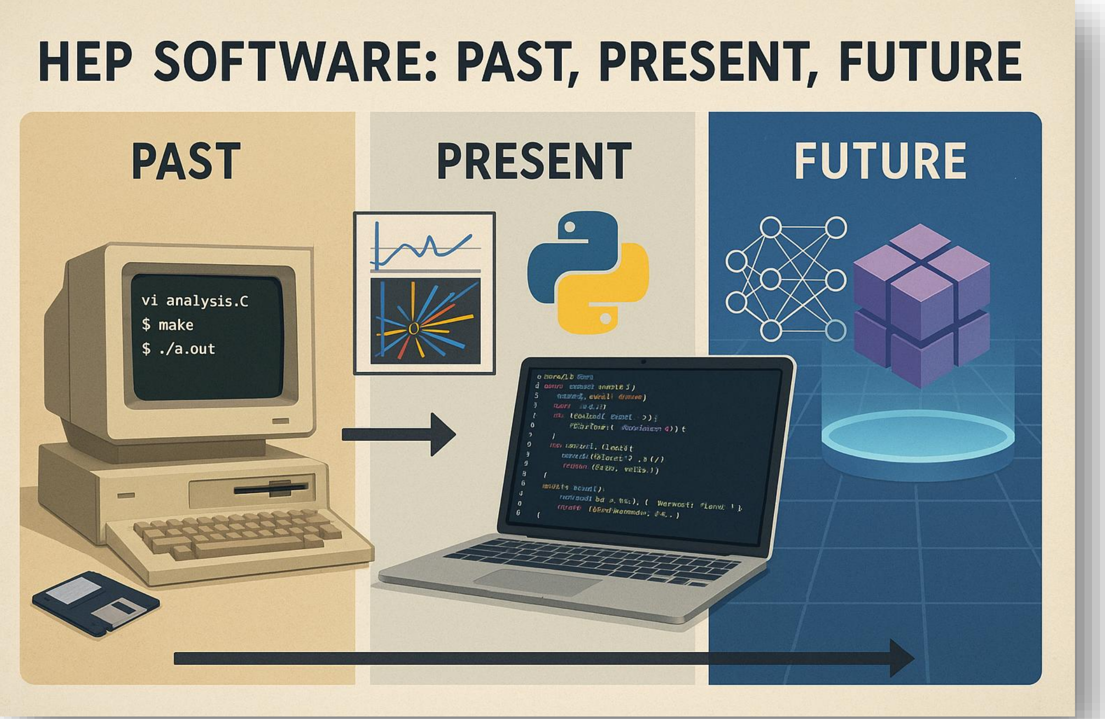
ALICE, ATLAS, Belle II, CMS, DUNE, ePIC, LHCb, MCnet, WLCG

arXiv:2504.01050 [hep-ex]

HEP Software: past, present, future – the ChatGPT executive summary/vision

- Image creation / reply on “HEP Software: past, present, future” *only*
- **Python is highlighted as far as the present goes !**
- **AI/ML stands out in the future !**
- Not bad a visual / interpretation …



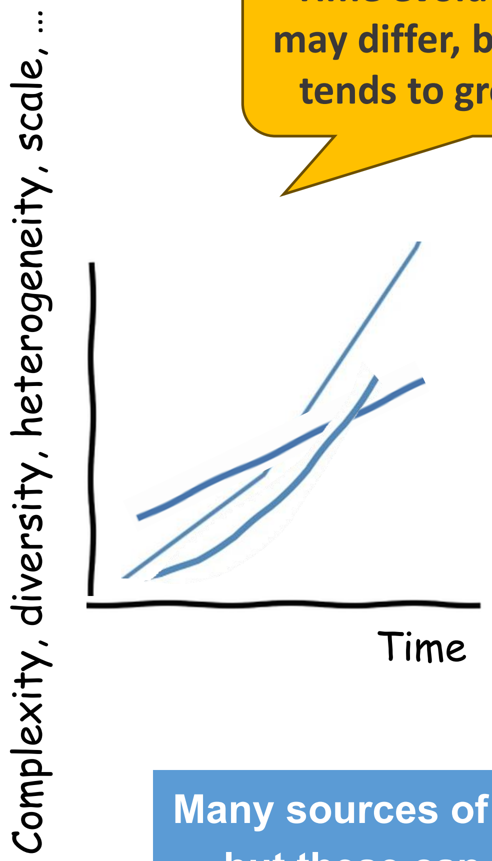


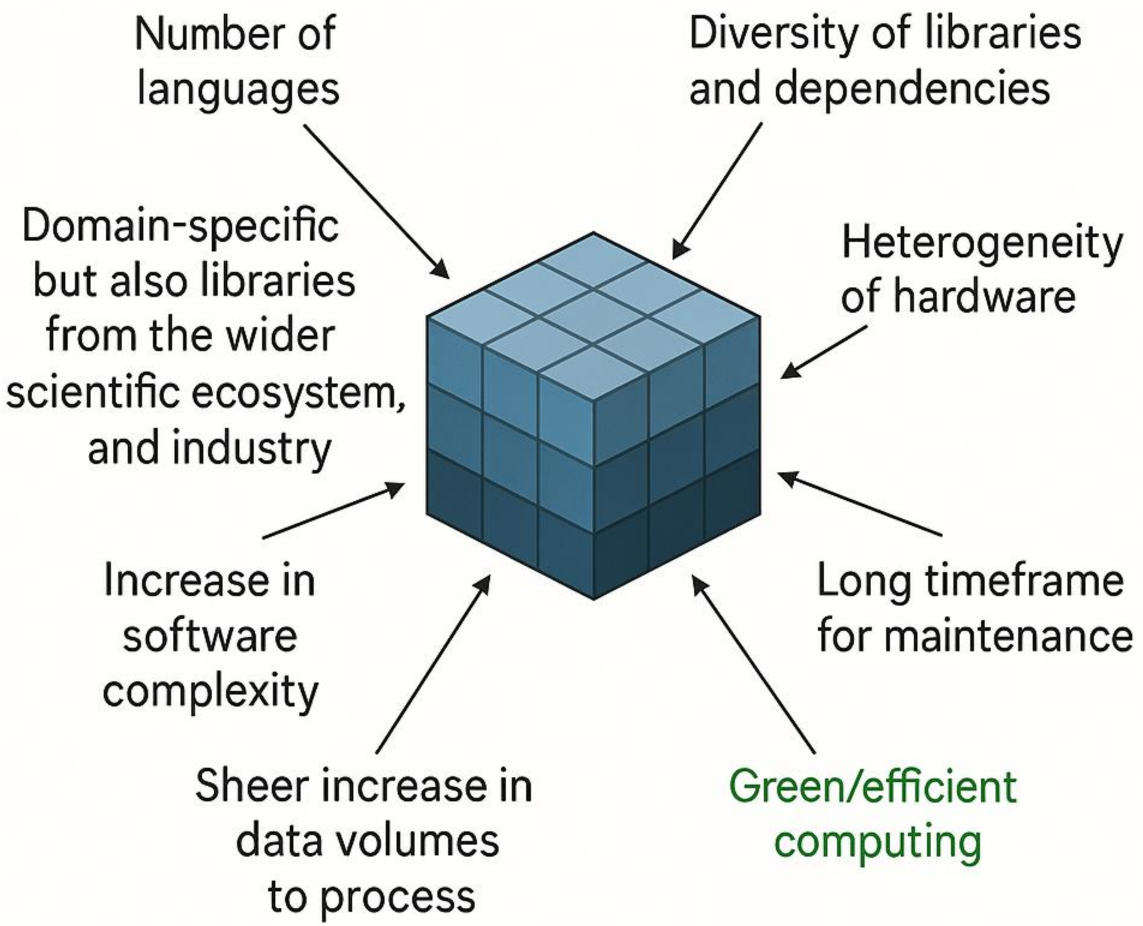
- **Past:** From manual calculations to the first computer simulations, HEP software evolved from basic tools to full-fledged systems for simulation and analysis.
- **Present:** Frameworks like ROOT, advances in distributed and grid computing, and the rise of machine learning have significantly transformed HEP software into a global, collaborative, and high-performance tool.
- **Future:** Quantum computing, AI-driven automation, and an increasingly open and collaborative scientific ecosystem will drive the next phase of HEP software evolution.

HEP software will continue to evolve in response to both scientific challenges and technological advances. The integration of AI, quantum computing, and open-source collaboration will pave the way for the next generation of breakthroughs in particle physics.

HEP Software (& Computing) – trend of a “curse of dimensionality in software”

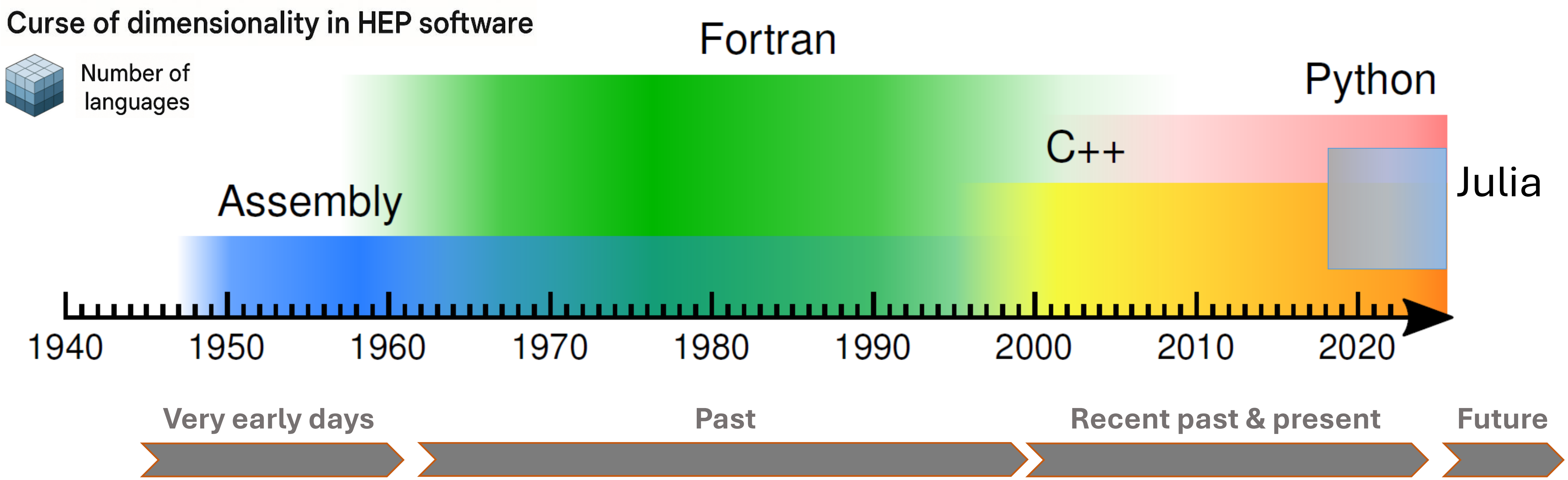
Time evolution (function) may differ, but every aspect tends to grow with time!





Many sources of *challenges*
… but these can be turned into *opportunities, e.g. better physics and faster development* !
- Requires R&D, investment in people, etc.

HEP Software: it all revolves around a language, or several



Adapted from J. Pivarski

- **40-ish years before the 2000s: the reign of Fortran**
- **Present: C++ & Python (Python increasingly used in user analysis)**
- **Julia sneaking into the arena**, especially in smaller experiments
  - “High performance meets high productivity”, ever-growing community & (HEP) tooling
- **Present & future: increasing mix of languages**

Metric	C++	Python	Julia
Performance	✅		✅
Ease-of-use		✅	✅
Learning Curve		✅	✅
Safety (memory)		✅	✅

HEP Software: software runs on hardware – heterogeneity of hardware

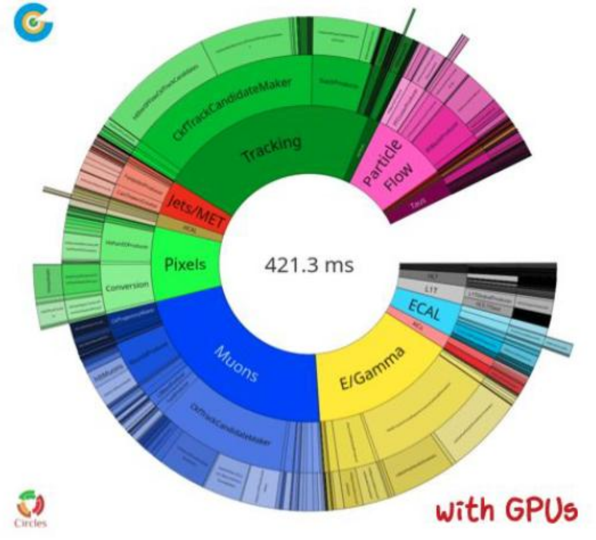
Curse of dimensionality in HEP software

Heterogeneity of hardware

- **CPUs in the past; now also GPUs, FPGAs, TPUs, IPUs, APUs**
- **Accelerators / heterogeneous architectures keep growing in importance**
- **This may be an (increasing) issue for HEP, at least it brings challenges**:
  - **Outside world of AI/ML dictating the evolution** – what's coming next ?
  - Different technologies often come with their software libraries / languages !
    - CUDA, Alpaka, HIP, Kokkos, HPX, OpenACC, OpenMP, OpenCL, oneAPI, TBB, SYCL, …
  - Lots of possible routes – challenge is how to converge and **ensure long-term support**
    - **Portability** seen as an important way for HEP to be "adaptive"

- **We must embrace technology developments & evolution to fully exploit the HL-LHC programme**
- **Promising route – making opportunities out of challenges:**
  - **Continue to engage in R&D on new topics and areas of special importance to HEP**
  - **Adopt ML solutions that the hardware "naturally" supports**
  - **Contribute to software developments to try and ensure HEP use cases aren't ignored**

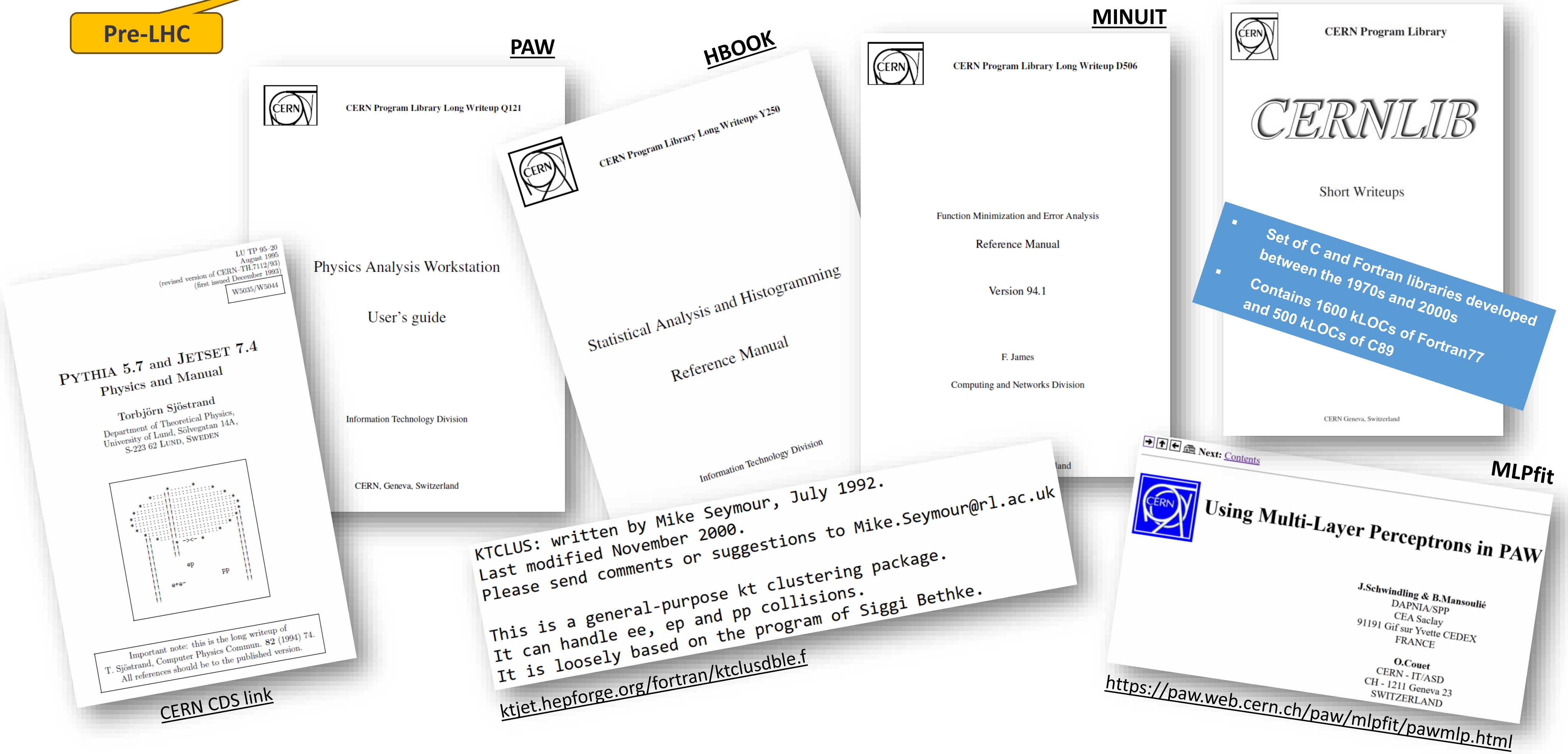
CMS example



Thanks to the use of GPUs
- 50% better event processing throughput
- 35% less processing time per event
- 15% - 20% better performance at initial cost
- 15% - 25% better performance per kW

See CHEP 2024 A. Bocci's talk.
Slide taken from summary talk.

HEP Software: **past**, present, future – key Fortran *domain-specific* libraries



Pre-LHC

PYTHIA 5.7 and JETSET 7.4
Physics and Manual
Torbjörn Sjöstrand
Department of Theoretical Physics, University of Lund, Sölvegatan 14A, S-223 62 LUND, SWEDEN
LU TP 95–20, August 1995 (revised version of CERN-TH.7112/93) (first issued December 1993) W5035/W5044
Important note: this is the long writeup of T. Sjöstrand, Computer Physics Commun. 82 (1994) 74. All references should be to the published version.
CERN CDS link

PAW
CERN Program Library Long Writeup Q121
Physics Analysis Workstation
User's guide
Information Technology Division
CERN, Geneva, Switzerland

HBOOK
CERN Program Library Long Writeups Y250
Statistical Analysis and Histogramming
Reference Manual
Information Technology Division

MINUIT
CERN Program Library Long Writeup D506
Function Minimization and Error Analysis
Reference Manual
Version 94.1
F. James
Computing and Networks Division

CERN Program Library
CERNLIB
Short Writeups
CERN Geneva, Switzerland

- Set of C and Fortran libraries developed between the 1970s and 2000s
- Contains 1600 kLOCs of Fortran77 and 500 kLOCs of C89

```
KTCLUS: written by Mike Seymour, July 1992.
Last modified November 2000.
Please send comments or suggestions to Mike.Seymour@rl.ac.uk

This is a general-purpose kt clustering package.
It can handle ee, ep and pp collisions.
It is loosely based on the program of Siggi Bethke.
```
ktjet.hepforge.org/fortran/ktclusdble.f

MLPfit
Next: Contents
Using Multi-Layer Perceptrons in PAW
J.Schwindling & B.Mansoulié, DAPNIA/SPP, CEA Saclay, 91191 Gif sur Yvette CEDEX, FRANCE
O.Couet, CERN - IT/ASD, CH - 1211 Geneva 23, SWITZERLAND
https://paw.web.cern.ch/paw/mlpfit/pawmlp.html

HEP Software: past, **present**, **future** – a plethora of *domain-specific* libraries

GEANT4
A SIMULATION TOOLKIT

ROOT
Data Analysis Framework

ROOT
Data Analysis Framework

TMVA

DD4hep

H7

Scikit HEP

Clad

<Add your favourite / guess future ones …>

SHERPA

MADGRAPH5
aMC@NLO

FastJet

KEY4HEP

acts

HEP.TrkX

- **Not to mention experiment-specific & "infrastructure" libraries and frameworks**
  - CVMFS, DIRAC, Rucio, SWAN, etc.
- **… and R&D software**
  - AdePT, Celeritas, traccc, etc.
- **Number of libraries is bound to increase …**
- **… especially when adding AI/ML to the mix …**

HEP Software: diversity of libraries and dependencies – example on analysis

Curse of dimensionality in HEP software

Diversity of libraries and dependencies | Increase in software complexity | Domain-specific but also libraries from the wider scientific ecosystem and industry

PAST:

- 1 language – Fortran
- Mostly domain-specific libraries, hence few(er) dependencies with the outside
- Rather centralised developments

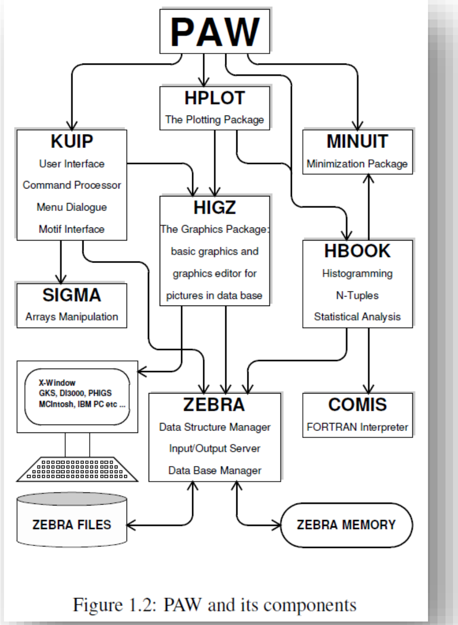
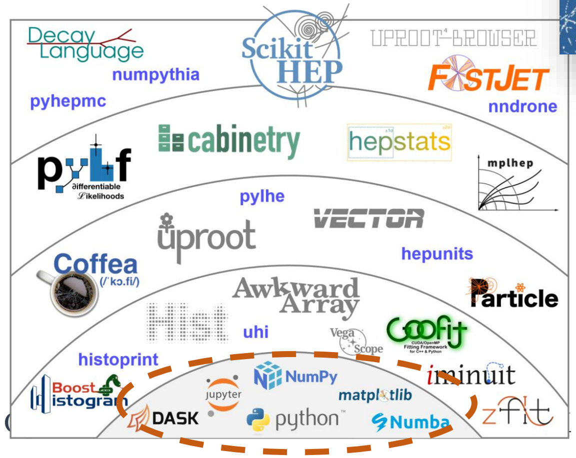


Figure 1.2: PAW and its components

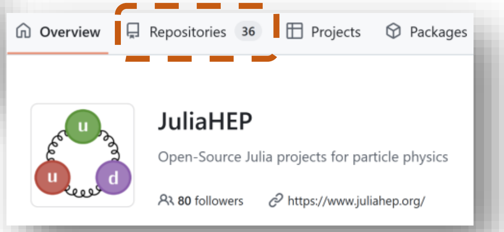
From CERN CDS record. PAW website.

PRESENT & FUTURE:

- Mostly C++ & Python (& Julia?)
- Increasingly higher # of dependencies with non-HEP (data science) and Industry tools
  - Especially true for AI/ML …
- More de-centralised developments, increase in community projects
- Software stacks, analysis models & workflows keep increasing in complexity
  - Necessity to deal with complexity of data to process & analyse
- Trend to move from rather rigid frameworks to more flexible toolsets
  - Makes evolution of individual pieces less painful than previously and lowers the risk of being stuck in local minima. Then **interoperability** is a key feature



Non-HEP



- **And many other libraries !**
- **E.g., not listing experiment-specific ones such as CMS Combine, workflow languages, …**

HEP Software: past, **present**, **future** – AI/ML to the mix

Curse of dimensionality in HEP software

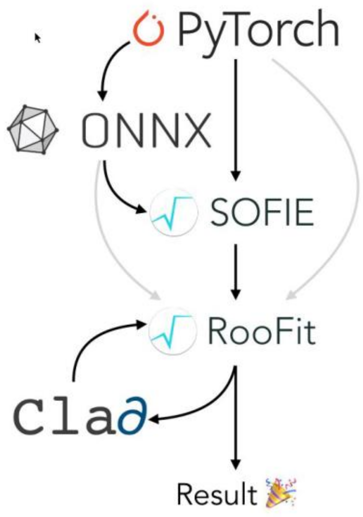
Diversity of libraries and dependencies

Increase in software complexity

Domain-specific but also libraries from the wider scientific ecosystem and industry

- **AI/ML has been, and will be ever more, entangled with everything we do**
  - Generators, simulation, reconstruction, object identification, analysis, etc.
- It also starts aiding with code writing and documentation, and operations …
- **We never depended on / needed so much software from the wider scientific world and Industry !**
  - Software landscape mostly driven by Industry
- **Opportunity:**
  - **Engagement with the outside is happening but can be enhanced, bi-directionally !**
  - **Encourage direct (code) contributions from HEP**

Differentiable programming came a few years to the arena. Watch this space.



ML model translated to C++ code with TMVA-SOFIE, differentiation in ROOT provided by Clad

Non-HEP

scikit learn

DMLC XGBoost

…

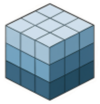
CERN EP-SFT's ML4EP project

(Machine Learning Tools for the Experiments)

hls4ml

HEP Software: sheer increase in data volumes to process

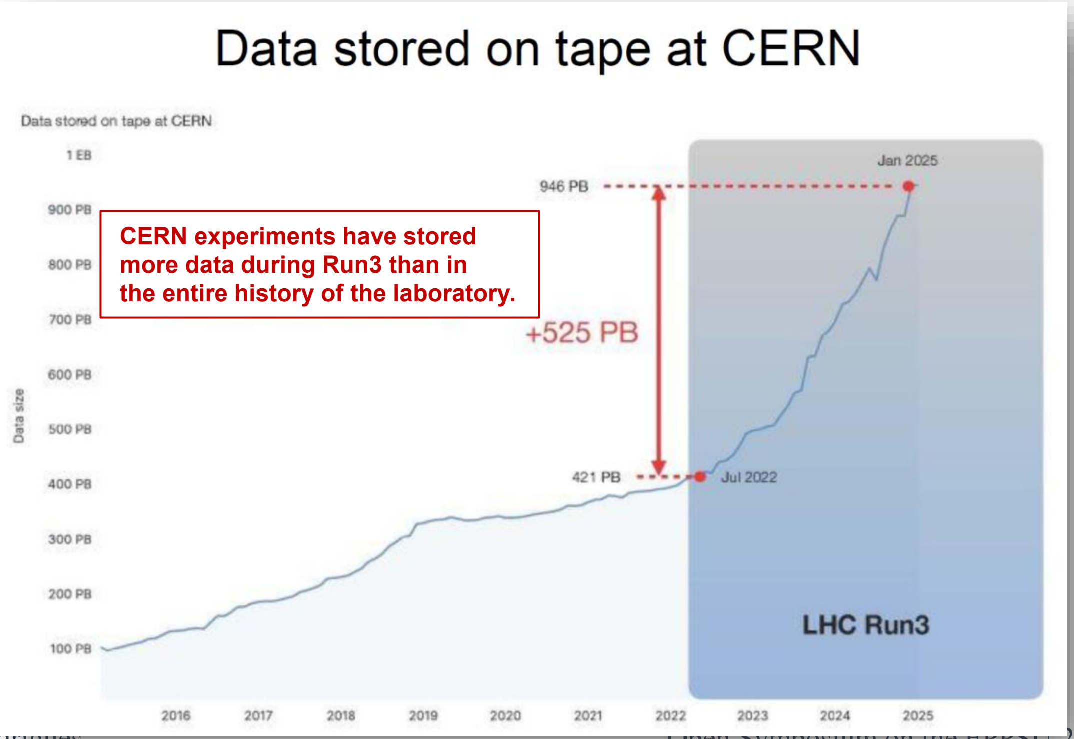
Curse of dimensionality in HEP software



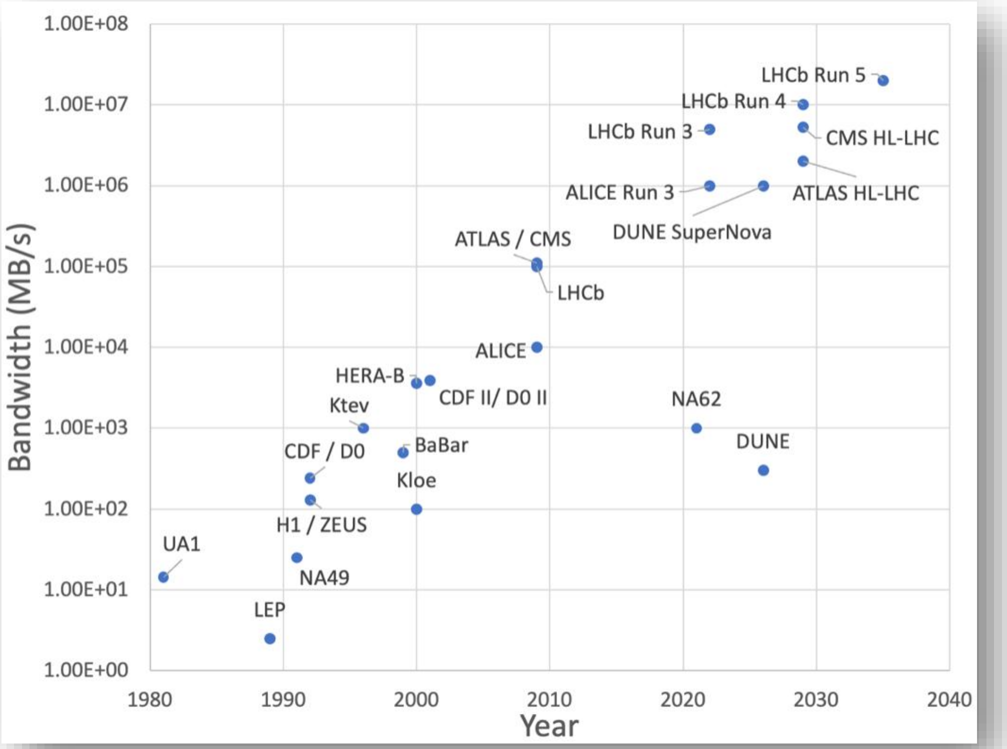
Sheer increase in data volumes to process

- **Data collection rates increased by ~6 orders of magnitude over 40 years !**
- Software throughput did not, to the same level
- **We went distributed (the Grid) and parallel and multi-threaded**
  - Help here from technology evolution, notably GPUs
- **Need to continue improving the software (& computing) to be able to digest data in reasonable timeframes …**

WLCG Intro talk, WLCG-HSF Workshop 2025



Instantaneous data rates (bandwidth) of HEP experiments over the past four decades



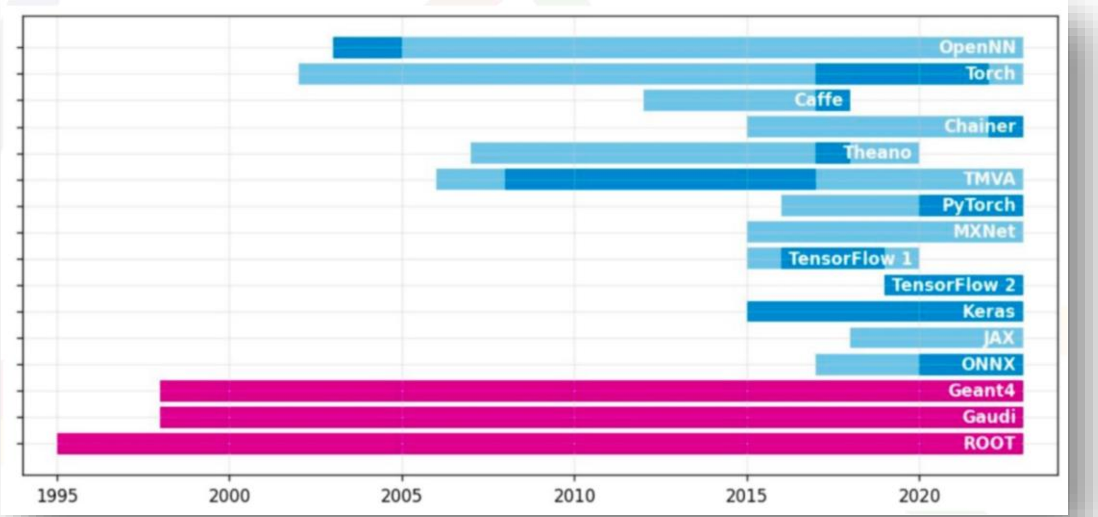
Eur. Phys. J. Plus 138 (2023) 11, 1005

HEP Software: past, present, future – maintenance, sustainability, preservation

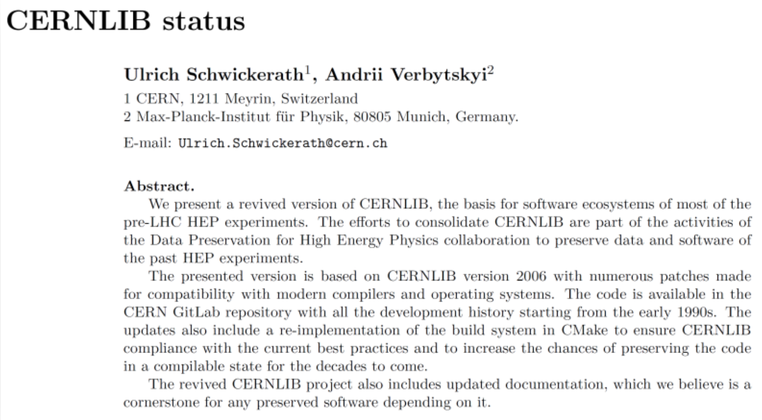
Curse of dimensionality in HEP software

Long timeframe for maintenancce

- **HEP software is historically developed and maintained for long, often decades**
- In **stark contrast** with much of what we see happening outside, especially for **AI/ML related software from Industry**
- **Considerable challenge for HEP – mitigate HEP written & maintained with from-industry ever-more-leveraged software!**



From M. Mazurek



CERNLIB status

Ulrich Schwickerath[1], Andrii Verbytskyi[2]

1 CERN, 1211 Meyrin, Switzerland
2 Max-Planck-Institut für Physik, 80805 Munich, Germany.

E-mail: Ulrich.Schwickerath@cern.ch

Abstract.
We present a revived version of CERNLIB, the basis for software ecosystems of most of the pre-LHC HEP experiments. The efforts to consolidate CERNLIB are part of the activities of the Data Preservation for High Energy Physics collaboration to preserve data and software of the past HEP experiments.
The presented version is based on CERNLIB version 2006 with numerous patches made for compatibility with modern compilers and operating systems. The code is available in the CERN GitLab repository with all the development history starting from the early 1990s. The updates also include a re-implementation of the build system in CMake to ensure CERNLIB compliance with the current best practices and to increase the chances of preserving the code in a compilable state for the decades to come.
The revived CERNLIB project also includes updated documentation, which we believe is a cornerstone for any preserved software depending on it.

arXiv:2303.07506 [physics.comp-ph]

- Notable example: for **software preservation purposes** (DPHEP) and usage with LEP stacks, the Fortran CERNLIB from the 1970s to 2000s was consolidated

HEP Software: past, present, future – the community is more and more organised

arXiv:1311.2567 [physics.comp-ph]

Computing Frontier: Software Development, Staffing and Training

Conveners: David Brown[1], Peter Elmer[2]
Observer: Ruth Pordes[3]
Contributing Authors: David Asner[4], Gregory Dubois-Felsmann[5], V. Daniel Elvira[3], Robert Hatcher[3], Chris Jones[3], Robert Kutschke[3], David Lange[6], Elizabeth Sexton-Kennedy[3], Craig Tull[1]

Thoughts on Software Engineering Ideas Applied to High Energy Physics Software

Spencer Smith, smiths@mcmaster.ca
Computing and Software Department, Faculty of Engineering, McMaster University, Hamilton, Ontario, Canada
June 16, 2017

arXiv:2004.07675 [physics.data-an]

Software Challenges For HL-LHC Data Analysis

Computing and Software for Big Science (2021) 5:22

Software Training in HEP

Received: 1 August 2021 / Accepted: 20 September 2021 / Published online: 8 October 2021
© The Author(s) 2021

arXiv:2302.01317 [hep-ex]

IRIS-HEP Strategic Plan for the Next Phase of Software Upgrades for HL-LHC Physics

February 3, 2023 - Version 1.0

Computing and Software for Big Science (2019) 3:7

A Roadmap for HEP Software and Computing R&D for the 2020s

The HEP Software Foundation[5] · Johannes Albrecht[69] · Antonio Augusto Alves Jr[81] · Guilherme …

1 Introduction
2 Software and Computing Challenges
3 Programme of Work
3.1 Physics Generators
3.2 Detector Simulation
3.3 Software Trigger and Event Reconstruction
3.4 Data Analysis and Interpretation
3.5 Machine Learning
3.6 Data Organisation, Management and Access
3.7 Facilities and Distributed Computing
3.8 Data-Flow Processing Framework
3.9 Conditions Data
3.10 Visualisation
3.11 Software Development, Deployment, Validation and Verification
3.12 Data and Software Preservation
3.13 Security
4 Training and Careers
5 Conclusions
Appendix A - List of Workshops
Appendix B : Glossary
References

arXiv:2210.05822 [hep-ex]

The Future of High Energy Physics Software and Computing

Report of the 2021 US Community Study on the Future of Particle Physics
organized by the APS Division of Particles and Fields

CompF5: End User Analysis Topical Group Report

Gavin S. Davies[1], Peter Onyisi[2], and Amy Roberts[3]
(and contributors from the community)

Submitted to the Proceedings of the US Community Study on the Future of Particle Physics (Snowmass 2021)

arXiv:2209.14984 [physics.comp-ph]

HEP Software: past, present, future – the community is more and more organised

Growing number of national R&D projects:

- EP R&D, NextGen (CERN)
- ExCALIBUR-HEP, SWIFT-HEP (UK)
- HEP-CCE, IRIS-HEP (US)
- Etc.

Community organisations:

- HSF (with PyHEP and JuliaHEP, training)
- IML
- Etc.

And international projets:

- AIDAInnova (European Commission)
- Geant4
- ROOT
- Scikit-HEP
- Etc.

Significant impact from any of these !

HEP Software @ European Particle Physics Strategy Update : 2020 → 2025

10.17181/CERN.JSC6.W89E

2020 UPDATE OF THE EUROPEAN STRATEGY FOR PARTICLE PHYSICS

by the European Strategy Group

4 Other essential scientific activities for particle physics

D. Large-scale data-intensive software and computing infrastructures are an essential ingredient to particle physics research programmes. The community faces major challenges in this area, notably with a view to the HL-LHC. As a result, the software and computing models used in particle physics research must evolve to meet the future needs of the field. ***The community must vigorously pursue common, coordinated R&D efforts in collaboration with other fields of science and industry, to develop software and computing infrastructures that exploit recent advances in information technology and data science. Further development of internal policies on open data and data preservation should be encouraged, and an adequate level of resources invested in their implementation.***

arXiv:2210.05822 [hep-ex]

The Future of High Energy Physics Software and Computing

Report of the 2021 US Community Study on the Future of Particle Physics

organized by the APS Division of Particles and Fields

- Long-term development, maintenance, and user support of essential software packages with targeted investment
- R&D efforts cutting across project or discipline boundaries should be supported from proof of concept to prototype to production.
- Support for computing professionals … to conduct code re-engineering
- and adaptation will enable us to use heterogeneous resources most effectively
- Strong investment in career development

We have been, and will keep, facing challenges, which we ought to / can change into opportunities:

- Software (and computing) is only scaling up and becoming more complex in various ways *"Curse of dimensionality of challenges in HEP software"*
- Heterogeneity is everywhere, software- and languages-wise, not just in terms of hardware we use
- AI/ML and heterogeneous computing increases throughput and physics reach, *if we are prepared to invest*
- Increasing usage of non-HEP software. Useful to engage bi-directionally
- Longer-term planning and support is paramount
- The future ought to be greener

Thank you for listening !

[Apologies for brevity and hence inevitable omissions.]

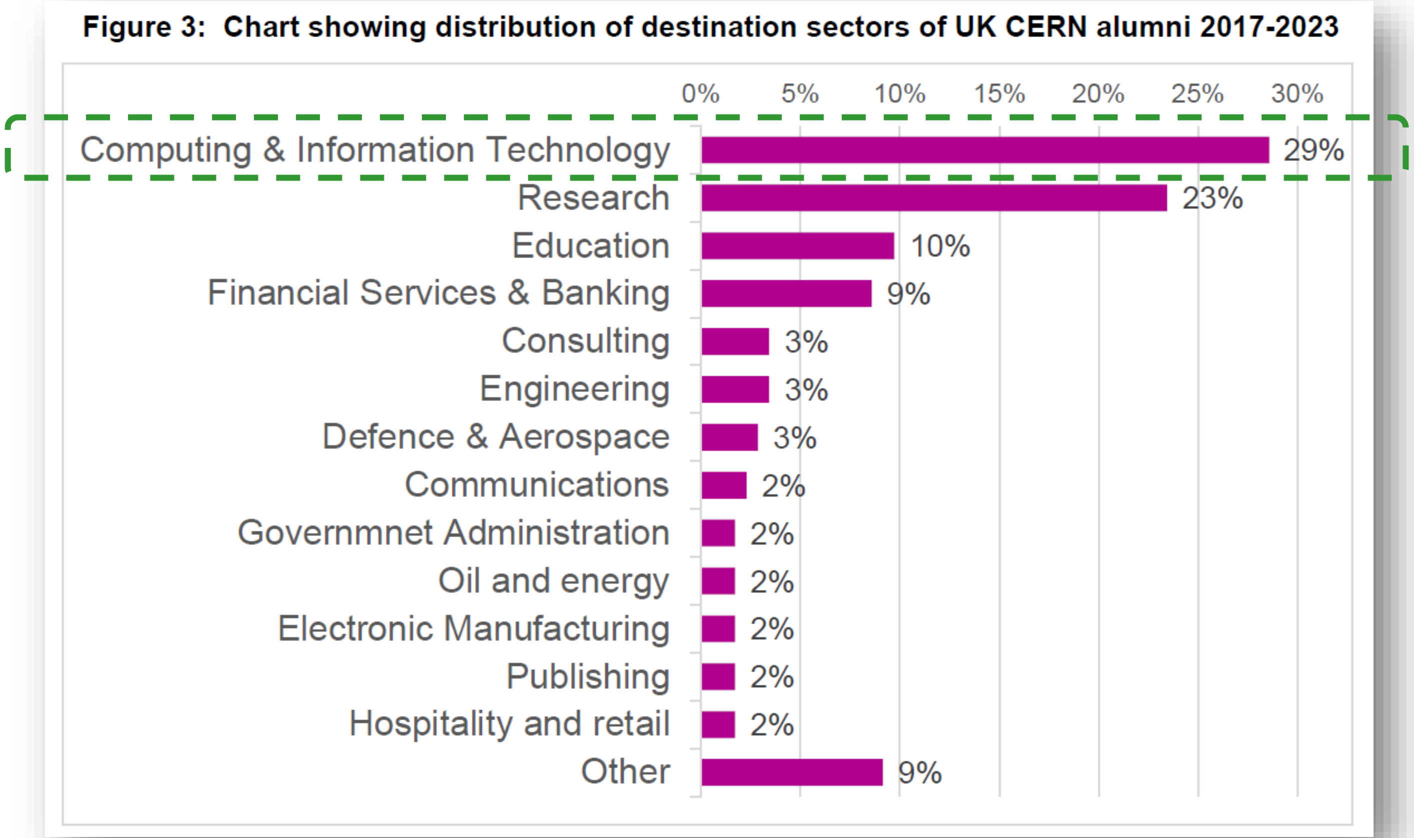
How to define / view (experimental) HEP Software ?

1. A collaboration’s software is a subdetector which is never turned off
2. This subdetector is constantly evolving
3. This subdetector is deployed in and must adjust to a constantly shifting environment
4. This subdetector must be maintained for years (decades) after it stops taking data

*One can nitpick that 2 and 3 apply to almost any system, but the point is the scale at which they apply

Taken from V. Glogorov, HSF Mini workshop: Trig & Reco Input for European Strategy for Particle Physics 2025

HEP software skills are highly valued outside HEP ! (Sad – not enough *within* HEP)



Taken from
« UK strategy for engagement with CERN: unlocking the full potential of UK membership of CERN »

HEP Software: **past**, present, future – PAW with Fortran & a DSL

Plotting histogram data

```
PAW > zon 1 2          | Divide picture into 2 vertically
PAW > set htyp -3      | Set hatch style for histogram
PAW > hi/pl 110        | Plot 1-dimensional histogram 110
PAW > hi/pl 210        | Plot 2-dimensional histogram 210
```

Plotting Ntuples

```
PAW > ZONE 1 2                   | 2 histograms one above the other
PAW > OPTION STAT                | Write statistics on plot
PAW > NT/PLOT 30.Z               | plot variable Z of Ntuple 30
PAW > 1d 300 'Z recalculated and user binning' 100 0. 10.
PAW > NT/PLOT 30.X**2+Y**2 IDH=300 | Recalculate variable Z + plot with user binning
```

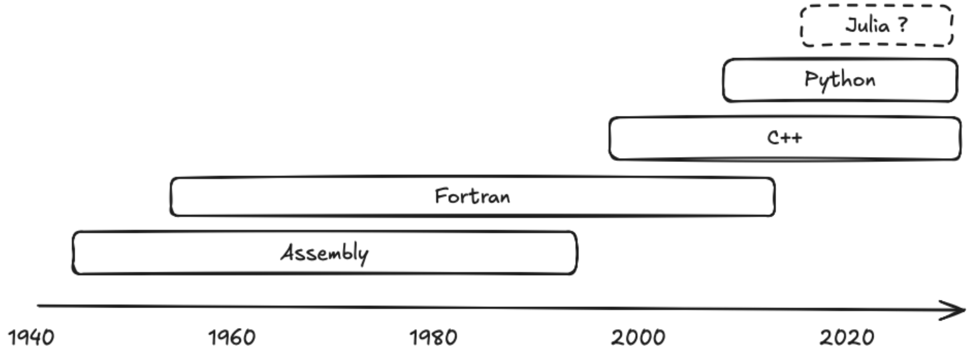
Fitting with Minuit

The User's main program

```
      PROGRAM DSDQ
      EXTERNAL FCNK0
      OPEN (UNIT=5,FILE='DSDQ.DAT',STATUS='OLD')
      OPEN (UNIT=6,FILE='DSDQ.OUT',STATUS='NEW',FORM='FORMATTED')
CC      CALL MINTIO(5,6,7)    ! Not needed, default values
      CALL MINUIT(FCNK0,0)    ! User routine is called FCNK0
      STOP
      END
```

```
*  how to
*                           - set examples from signal and background ntuples
*                           - define a network
*                           - train the network
*                           - test the network
*
* J.Schwindling 01-June-99

mlp/create 8 5      | create the network first

h/fil 1 ww.ntup
mlp/lpat/set //lun1/2000 sqrt(v1)%v2%v3%v4%v5%v6%v7%v8 1. 1. 1000 1 typ=0
mlp/tpat/set //lun1/2000 sqrt(v1)%v2%v3%v4%v5%v6%v7%v8 1. 1. 4000 5001 typ=0

h/fil 2 qq.ntup
mlp/lpat/set //lun2/2000 sqrt(v1)%v2%v3%v4%v5%v6%v7%v8 0. 1. 1000 1 ! +
mlp/tpat/set //lun2/2000 sqrt(v1)%v2%v3%v4%v5%v6%v7%v8 0. 1. 4000 5001 ! +


mlp/learn 100

read suite

1d 1000 ' ' 100 -0.5 1.5
1d 1100 ' ' 100 -0.5 1.5
nt/proj 1000 //lun1/2000.pawmlp.f(0.) typ=0
nt/proj 1100 //lun2/2000.pawmlp.f(0.)

set xmgl 2.5
set xlab 1.7
his/pl 1100
set htyp -3
his/pl 1000 s
set htyp
atit 'NN output' 'Number of events'
```

HEP Software: past, present, **future** – the julia programming language in HEP



High performance meets high productivity

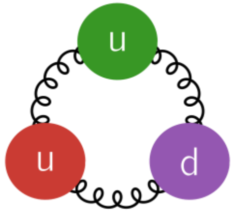
- As easy and productive as Python; as fast as C++

Modern tooling for complex HEP workflows

- Used experiment analysis, large-scale simulations and industrial applications

Ever-growing community

- HSF's JuliaHEP Activity Area
  - See also https://www.juliahep.org
- CHEP 2024 plenary talk
- JuliaHEP annual workshop (ECAP 2023, CERN 2024, Princeton 2025)



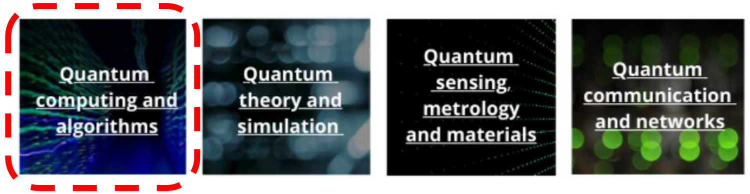
Metric	C++	Python	Julia
Performance	✅		✅
Ease-of-use		✅	✅
Learning Curve		✅	✅
Safety (memory)		✅	✅

HEP Software: past, present, **future** – the rise of Quantum …?

- Quantum Science & Computing is a hot topic worldwide
- **HEP has been involved in / exploring various areas since years …**
  - But still largely a niche activity
- ***Software contributions wise*, still relatively little to say**
  - Challenge – each Q computer technology has its own software
- HEP does contribute to software, e.g. QIBO
- Still a long way, but see recent reports from US Snowmass & CERN's QC4HEP
  - Fault-tolerance achieved in time for the next EPPSU …?

At CERN:

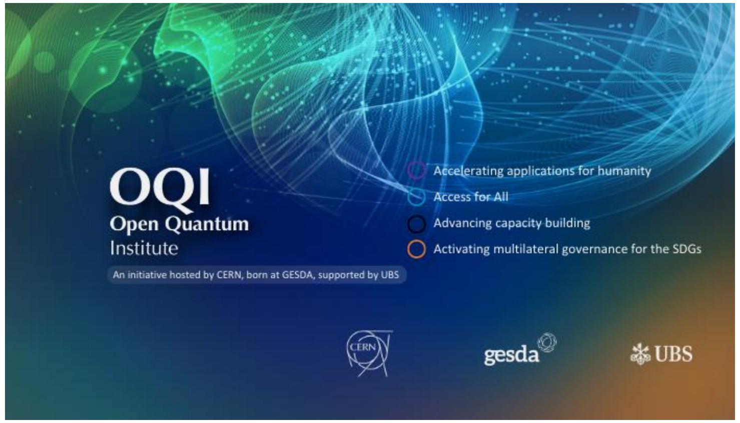
- Nov. 2018, 1st workshop on Quantum Computing in HEP
- June 2020: CERN launches the Quantum Technology Initiative (QTI)



- March 2024: launch of the Open Quantum Institute

QUANTUM TECHNOLOGY INITIATIVE

CERN Quantum Technology Initiative



March 2024: Operational launch of the Open Quantum Institute

OQI Open Quantum Institute

The Open Quantum Institute

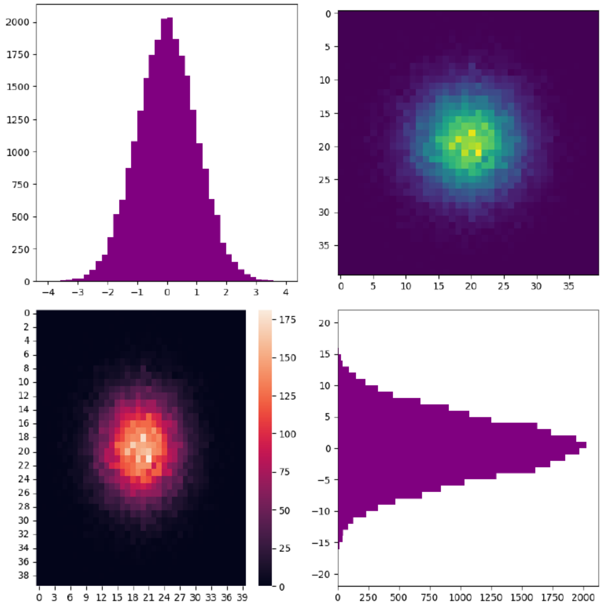
- **Code snippet with ROOT, some of Scikit-HEP, and seaborn (wider scientific Python library)**
  - BTW, you could mix/swap which libraries to use for each snippet "section"

UHI and ROOT: 2D

Yet another example of Python tools and ROOT interoperability!

```python
import ROOT
from ROOT.uhi import sum
import numpy as np
import matplotlib.pyplot as plt
import mplhep as hep
import seaborn as sns

h2 = ROOT.TH2F("h2", "", 40, -4, 4, 40, -20, 20)

for i in range(25000):
    px, py = np.random.normal(0, 1, 2)
    h2.Fill(px, 5 * py)

projh2X = h2[::sum, :]
projh2Y = h2[:, ::sum]

fig, axes = plt.subplots(2, 2, figsize=(9, 9))

axes[0, 1].imshow(h2.values(), interpolation='nearest')

sns.heatmap(h2.values(), ax=axes[1, 0])

hep.histplot(projh2X, ax=axes[0, 0], histtype='fill', color="purple")

hep.histplot(projh2Y, ax=axes[1, 1], histtype='fill', color="purple", orientation="horizontal")

plt.tight_layout()
plt.show()
```

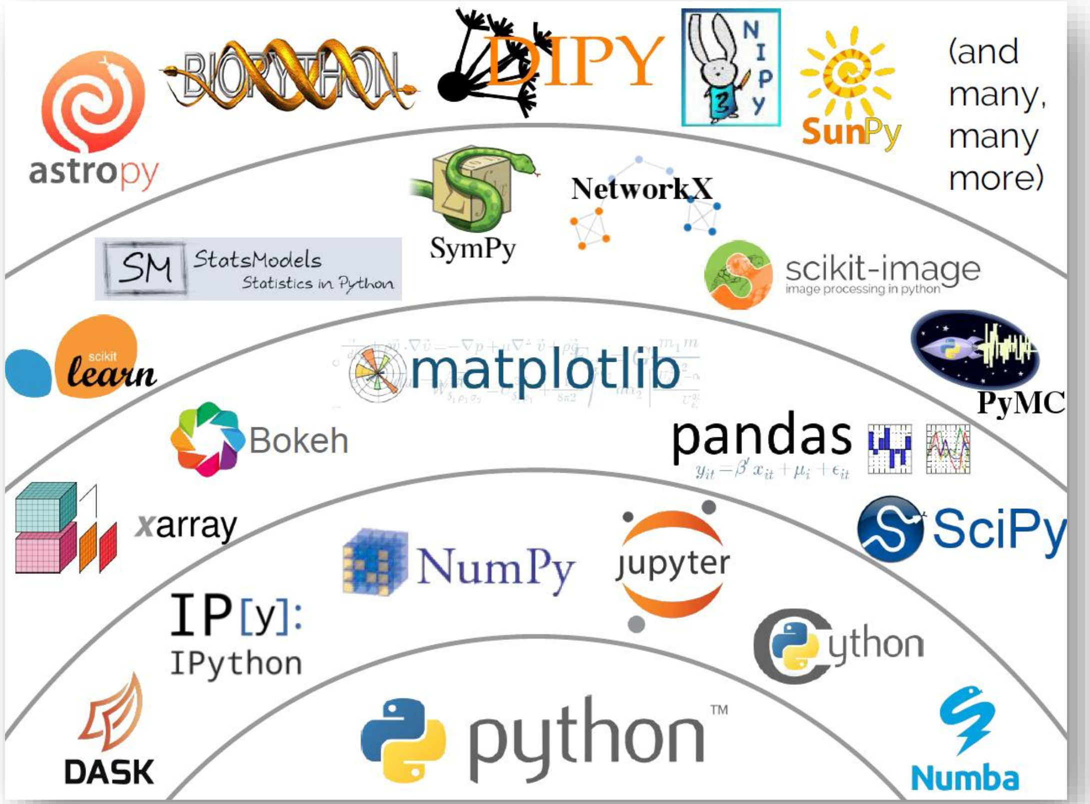
Coming next: we are experimenting with new approaches, input is welcome!

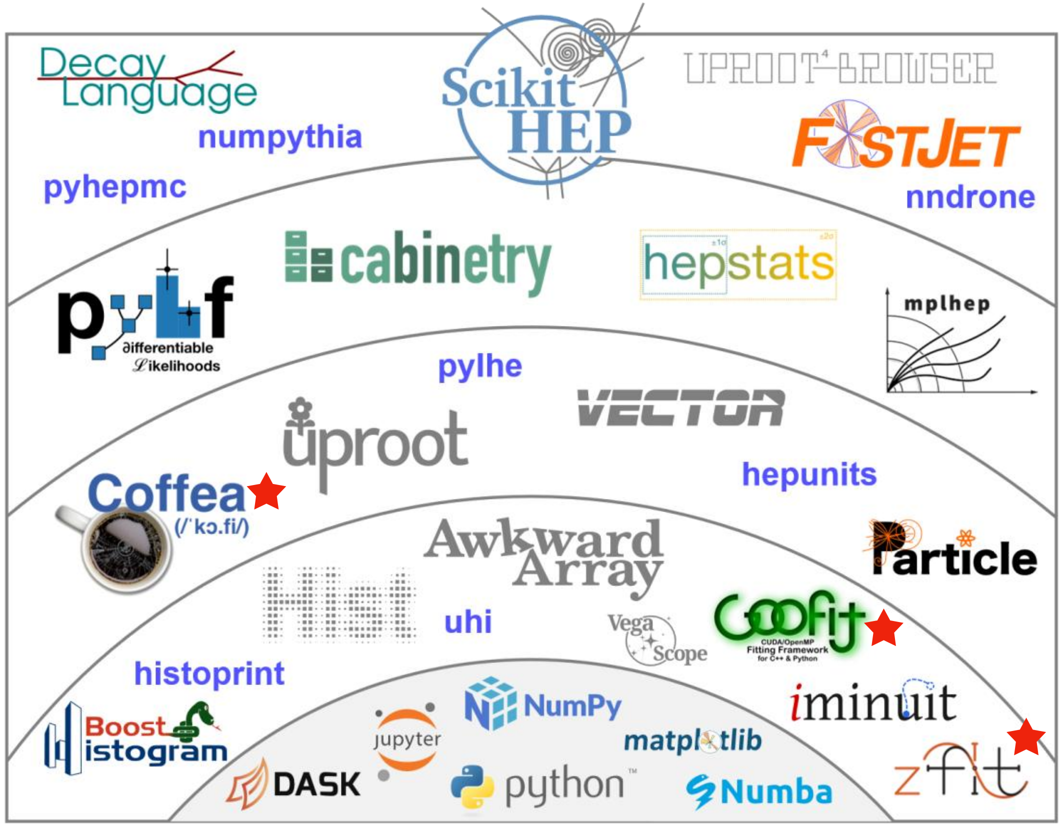


J. Rembser | CERN EP-SFT | 6-5-25 WLCG/HSF Workshop | ROOT 17

Scikit-HEP contribution to the HEP domain-specific Python analysis ecosystem

- (Back in 2024) After 7-ish years we can proudly splash **our "NumPy fueled" shell ecosystem side-by-side ☺ !**
  - **The full HEP ecosystem is of course wider, ROOT being prominent**, in particular – some call it the PyHEP ecosystem





(Not a full list.)

(In the mean time Coffea joined the org; it is no longer an affiliated package)

Affiliated package

Showcasing HEP Python packages working together atop the scientific ecosystem

- **SciPy 2024 contribution paper describing to a broad audience how a large scientific collaboration leverages the power of the Scientific Python ecosystem to tackle domain-specific challenges** and advance our understanding of the Cosmos
  - Through a simplified example of the renowned Higgs boson discovery
  - With a **Jupyter notebook** showing the various steps from data cleaning, statistical interpretation and visualization at scale, **with many Scikit-HEP packages and others as well (Dask, etc.)**

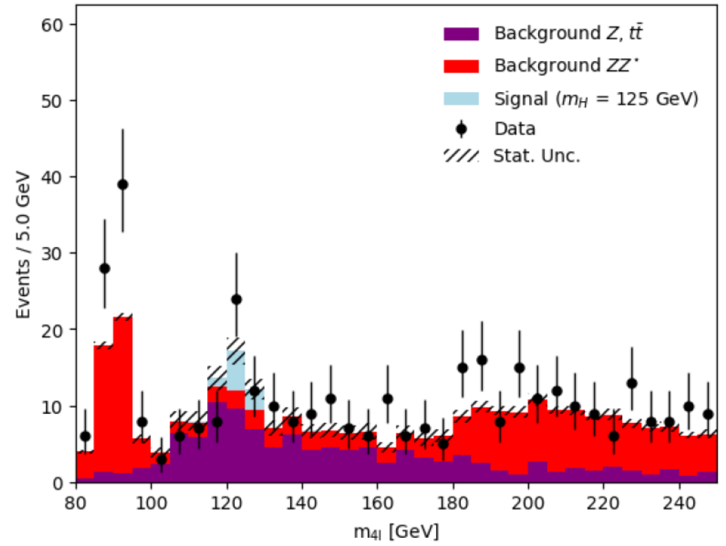


Figure 3. *Using* `mplhep`, `hist`, *and* `matplotlib` *the post-processed histograms of the simulation and the data are visualized in advance of any statistical inference of best-fit model parameters.*

SciPy 2024

July 8 - July 14, 2024

Proceedings of the 23rd Python in Science Conference
ISSN: 2575-9752

How the Scientific Python ecosystem helps answer fundamental questions of the Universe

Matthew Feickert[1], Nikolai Hartmann[2], Lukas Heinrich[3], Alexander Held[1], Vangelis Kourlitis[3], Nils Krumnack[4], Giordon Stark[5], Matthias Vigl[3], and Gordon Watts[6]

[1]University of Wisconsin--Madison, [2]Ludwig Maximilians Universitat, [3]Technical University of Munich, [4]Iowa State University, [5]Santa Cruz Institute for Particle Physics, [6]University of Washington

Abstract

The ATLAS experiment at CERN explores vast amounts of physics data to answer the most fundamental questions of the Universe. The prevalence of Python in scientific computing motivated ATLAS to adopt it for its data analysis workflows while enhancing users' experience. This paper will describe to a broad audience how a large scientific collaboration leverages the power of the Scientific Python ecosystem to tackle domain-specific challenges and advance our understanding of the Cosmos. Through a simplified example of the renowned Higgs boson discovery, attendees will gain insights into the utilization of Python libraries to discriminate a signal in immersive noise, through tasks such as data cleaning, feature engineering, statistical interpretation and visualization at scale.

https://doi.org/10.25080/KMXN4784

https://github.com/ekourlit/scipy2024-ATLAS-demo

ROOT: Foundational Open-Source Software for Science

https://root.cern

- **History & Status:** Released 1995. Current: ROOT 6. Actively developed ROOT 7 (LHC Run 4).
- **Development Model:** Open-source & open-development (GitHub). Community contributions exceed core devs 2:1. Supported by international institutes & users.
- **Massive Adoption:**
  - Core of LHC experiment software stacks (>2 Exabytes data in ROOT format).
  - Used widely, e.g.: Tevatron, Belle II, DUNE, ePIC, GSI + thousands of individual scientists.
- **Key Innovations:**
  - Rewritten multiple times for evolution.
  - Pioneered HEP IO, columnar data format & C++ reflection.
  - Created Cling (C++ interpreter) & PyROOT/Cppyy (seamless C++/Python integration).
  - Anticipated distributed computing (PROOF → Dask/Spark concepts).
- **Future Focus:**
  - ROOT 7 development (RDataFrame, RNTuple).
  - Supporting LHC Run 4 & future colliders (FCC).
  - Engaging diverse communities for future-proof solutions.

Geant4: Open-Source Particle Simulation Toolkit

- **Core Function:** Object-oriented C++ Monte Carlo toolkit for simulating particle-matter interactions
- **Scope & Adoption:**
  - 30-year international collaboration spanning HEP, medical, space, nuclear physics
  - De-facto HEP standard with > 20k citations; used from experiment design to final analysis
- **Key Innovations:**
  - Pioneered large-scale OO C++ in HEP software
  - Physics advancements:
    - EM: High-precision, G4HepEm (CPU/GPU-accelerated)
    - Hadronic: Refined models (FTF/QGS strings, BERT/BIC/INCL cascades), FLUKA interface
- **Continuous Improvement:**
  - **Precision:** User-driven model extensions (charm/bottom hadrons, hypernuclei)
  - **Speed:** Geometry optimizations (e.g., ATLAS EMEC), GPU R&D (AdePT/Celeritas), ML fast-sim (ATLAS GAN)
  - Robust validation suite (geant-val.cern.ch)
- **Future Challenges:**
  - Scaling for HL-LHC/future colliders: higher precision & throughput
  - Knowledge transfer: Replacing pioneer developers while retaining expertise

ML4EP: Machine Learning Tools for the Experiments

- **Core Concept:**
  - Developing and maintain common ML software solutions required by experiments
  - Avoiding duplication of ML activities
  - Hosting common ML activities in SFT and in the NextGen trigger project
- **Key Innovations:**
  - Diffusions models for fast simulation of calorimeter showers which can be generalised and used on different detectors
  - Translation of ML models to C++ for CPU and GPU (SOFIE)
  - Bring ML models to FPGA (hls4ml)
  - Library to compress ML models using quantization and pruning (PQuant)
- **Impact & Adoption:**
  - Develop unique ML technologies needed by experiment, not in competition with industry
  - Produced tools (e.g., hls4ml) can be used outside HEP and in industry
- **Users:**
  - All LHC experiments, future experiments and industry (ML on edge)

CVMFS: Revolutionizing Scientific Software Distribution

CernVM File system

- **Core Concept:**
  - Global read-only filesystem delivering software/data **like a streaming service**
  - "Enabled 100,000+ core workflows to run identical software without sysadmin intervention"*
- **Key Innovations:**
  - Content-addressable storage (deduplication + versioning)
  - HTTP-based global distribution with auto-caching
  - Seamless container integration (containerd/docker)
  - Cryptographic integrity + zero local maintenance
- **Impact & Adoption:**
  - Preservation: Critical for reproducible science & decade-long experiment lifecycles
  - Scale: Distributes petabytes to >1M cores (LHC runs) with minimal local storage
  - Efficiency: Eliminates local installs; updates propagate in minutes
- **Users:**
  - All LHC experiments, HENP, Astrophysics (LIGO/SKA/LSST), HPCs, industry