How to Sustain Common Software
&
Computing Operations
in HEP?
A Community Levy

Fundamental research in HEP relies on domain specific & specialized software.
The community behind scientific software is fragile.

The current state is “broken”

- No coherent plan for funding software or community behind science
- No support for maintenance or development
- There are disincentives to adopt community software
- No knowledge transfer

Future of HEP Software & Computing: Elvira, Gottlieb, Gutsche, Nachman: arXiv:2210.05822

1. The US HEP community should take a leading role in **long-term development, maintenance, and user support** of essential software packages with targeted investment.

   - A new structure is needed to fund modernization, maintenance, and user support of existing tools (grants typically only fund ground-breaking R&D or development of new software).
   - Examples include (i) event generators and simulation tools like `Geant4` [2, 3, 4] that do not belong to a particular facility, experiment, or survey, (ii) S&C tools associated with one or more experiments, and (iii) data/software preservation after an experiment has ended.

3. Support for computing professionals/researchers and physicists to conduct code re-engineering and adaptation will **enable us to use heterogeneous resources** most effectively.

   - Most HEP software runs on a single computing platform, making it difficult to use the multitude of hardware accelerators and diverse computing resources like cloud, HPC, etc.
   - To satisfy the needs of inherently serial algorithms that are still transitioning towards computing accelerators or are not cost-effective to port, an appropriate level of traditional CPU-based hardware should coexist with more powerful heterogeneous resources.

P5 Report

Area Recommendation 18: Increase targeted investments that ensure sustained support for key cyberinfrastructure components by $8M per year in 2023 dollars. This includes widely-used software packages, simulation tools, information resources such as the Particle Data Group and INSPIRE, as well as the shared infrastructure for preservation, dissemination, and analysis of the unique data collected by various experiments and surveys in order to realize their full scientific impact.

Area Recommendation 19: Research software engineers and other professionals at universities and labs are key to realizing the vision of the field and are critical for maintaining a technologically advanced workforce. We recommend that the funding agencies embrace these roles as a critical component of the workforce when investing in software, computing, and cyberinfrastructure.

No coherent plan for funding software or community behind science

- some software developed by theory/pheno as original research
- “largest” user is experiment!
- universities don’t support
- national labs don’t have graduate students
- grants focus on cutting edge R&D

No support for maintenance or development

- software ages and has bugs – consumes time
- typical physicist learn about computing in back alleys: might not be optimal or efficient
- new developments and ideas need to be coded!

There are disincentives to adopt software developed by grants/projects

- unsupported software will age
- unsupported code will fragment

Lack of knowledge transfers

- unique skills behind community not part of standard education
- experiments have solutions to bottlenecks, but no obvious mechanisms to interact with HEP software community (save joining the experiment)

Proposal:

Modest, structured **community levy** (a small, fixed percentage of experiment/project budgets) to sustain HEP software and the HEP software community.

Establish a **community consortium** to steward levy funds, set priorities, and coordinate with funding agencies and computing research programs. Governance should include experiments, laboratories, theorists, grassroots software groups, and agencies.

Enforce **accountability** via annual reports, open calls for proposals, and external review ensures transparency and impact.

Direct levy support to:

- Permanent laboratory software positions to anchor mission-critical expertise;
- Joint lab--university tenure-track appointments with a measurable deliverable of community software to cultivate academic leaders in computation for HEP;
- Inclusive training---schools, hackathons, documentation, and courses with low barriers to entry---to recruit and retain a diverse workforce.

HSF, IRIS-HEP, the lab and experiment courses, other miscellaneous schools, already provide some training.
However, training in basic simulation techniques is lacking: random number generation, Monte Carlo techniques and all of the related computing issues (speed, reproducibility, transparency, portability). The recent MC-GEN project is a step to provide this type of training.

Implementation Roadmap

- Pilot: Launch with voluntary contributions (e.g., 1--2\% of major experiment budgets) to fund high-priority maintenance, portability efforts, and user support. Identify the most used external codes, look at those projects, and decide which are suffering. Target X positions for a term of Y units (TBD).
- Formalization: Integrate the levy into DOE/NSF and partner-agency guidelines for new projects and facility operations.
- Governance: Seat a rotating board representing experiments, labs, theory, grassroots software efforts, and agencies.
- Accountability: Require public annual reports, external reviews, and open competitions for awards.
- Scaling: Extend coverage across HEP domains as the pilot demonstrates impact.