Discussion notes: languages

- We don't want to be locked into Python/C++ forever - new languages bring new capabilities! How do we support new languages while maintaining maintainability?
    - Lindsey: FFI layer. We have a strong need for interoperability and supportability. Supporting these might also provide support for language evolution.
    - Jerry: Julia for HEP analysis have made significant progress between 2020-2022, should we mention it?
        - Ankur: Julia has been really nice for migrating people off Matlab and providing easy exposure to optimizing code and scaling up computational scale
    - Amy: does LLVM help us here?
    - Ian: The driving force behind language transitions isn't us! We may not have a ton of influence/control!
        - Axel: We can participate/give feedback to, e.g., hardware vendor language extensions and the ISO C++ committee (as ROOT does), and thus can influence industry trends. We should continue to integrate with these ecosystems for our needs & extend them.
- Giordon: May not be productive to focus on a specific language recommendation. Instead, what is useful about languages in use? For example, python bindings are one of the reason the language is so useful. A columnar approach (e.g. through numpy) is not as natively tied into python, versus Julia where a lot of these things are considered more integral to the

language (and not as an “afterthought”). I prefer to pick the right tool for the job and let things be a little bit more modular.

- Peter: need to be aware of sunk costs. Jeremy: sunk costs does not apply to new projects and experiments.

Discussion: Training

- Is easier the right approach?
  - Lindsey: easy to read/debug is paramount
- Alex: People’s interest in computing is varied.
  - Amy: I think we’re discussing end-user analysis expectations/requirements, support for software development needs in the community, and training. End-user analysis tools don’t have to address all these
- Doa:
- “What is the set of computing skills analyzer need?”
  - Rob: No clear consensus! Involved in a small experiment and recently lost an entire computing cadre. One of the issues is they’ve relied on magic postdocs instead of building a community. Amy: THIS IS ALL OF US. Rob: we really do need general skills.
  - Ian: When we create end-user analysis tools, how black-box is this? Is it a technical activity? Do we require people to have more general training?
    - Gabriel Perdue: it would be ideal to incorporate this into the curriculum
      - Doa: I took some of this as an undergraduate but really needed to learn more for my research. I spent a lot of time re-inventing the wheel. The training

        available for under graduate students was really insufficient for PhD work

    - [Alexander Held] There is a big difference between being a black box wrt. performance (what kind of magic built-in optimization does some library you’re using have) and a black box wrt. physics (that’s the important part that users need to understand!)

- Jerry: Computational methods in Python is not useful because they’re not production-ready, ideally a language that reads like pseudo code and runs fast

Discussion: Analysis preservation

- Amy: what do we do with non-structured metadata like wiki pages, email chains?
    - Giordon: If you are properly structuring your analysis within GitHub, etc. then nonstructed metadata is less of an issue. Part of the issue is a gap between interest in learning these tools and using these tools (like Docker containers). We need education and training and awareness. There is also a validation hurdle - having your analysis code/workflow in a github repo is not enough, you need someone to try to test it. A recommendation needs to include support for both these aspects
    - Rob: I’ve had trouble finding a funding path for this. Labs provide services for experiments and expect experiments to do this. Experiments expect labs to do this!
    - Peter: this is the kind of thing we can make recommendations on
    - Jeremy: How do we make the argument that this, out of all the other emergencies, needs funding

  - Ian the Blasphemer: We've struggled/failed with this for the last 70 years. Outside experiments it's virtually impossible to reproduce experiments. Why do we always fail at this? Is it really important? If it is then we really need a different approach.
  - Jeremy: Haaaaas anyone reached out to other frontiers to ask what their needs are? Amy: probably a good idea!
- Giordan: On lab-funded vs. university-funded code. Major features are now being dropped because the university can't afford to maintain the code anymore. From a preservation perspective, how do we preserve code that's not in our scope?
- Ankur: A representative from the accelerator frontier! Information transfer has been a huge issue with many retiring and new people coming in. A lot of code is LabView! They've had to break things down into pseudo code that can be re=interpred in other languages and also just understood! Peter: this ties back into validation.
- Alex: What you want is to have the data so that you can analyze it in a whole new way as understanding changes. Peter: there is a layer below which data is simply not accessible. The moment you need oracle you're done.
- Stephen Bailey: Analysis preservation as a black box vs. analysis preservation to understand what was done for education/learning/reminding what exactly we did to be useful for the future

    - Gabriel Perdue: “The key thing you need to be able to do is rerun the old analysis on top of a new Monte Carlo”
- Amit: Data preservation challenges can be very useful (interface with CompF7)
- Peter: many different levels of analysis preservation, skimmed data, raw data,
- There’s a distinction between preserving data and preserving analysis. And then we need to think about preserving an analysis within a collaboration and outside a collaboration. Preserving an analysis even within a collaboration is very difficult!!! Amy: +1 million.
    - CompF5 is most naturally focused on preserving an analysis within an collaboration

General comments about the report

- Heidi: important to talk about exactly what the HSF is doing, they’ve done a lot of work and it’s important to know what that is
- Daniel: when will a draft be available?
- Alex: should we think about the transition from analysis code as a small set of libraries/scripts to something that’s maintained by someone (sometimes a lab)?
- Heidi: the first analysis framework is often not the best.

We went with “let a thousand flowers bloom,” sometimes there’s real value in letting people try things and have people vote with their feet.

  - Alex: yes, this can also become a maintenance problem

- Heidi: end analysis and simulation data are two entirely different sets of need, you can’t use the same frameworks/code for both of these. Alex: cannot agree enough!
- Giordan: On lab-funded vs. university-funded code. Major features are now being dropped because the university can’t afford to maintain the code anymore. From a preservation perspective, how do we preserve code that’s not in our scope?