Beam Dynamics Toolkit

David Sagan*[1], Robert Ryne[2], Jean-Luc Vay[2], D.L. Bruhwiler[3], Dave Grote[4], Georg Hoffstaetter[1], Chengkun Huang[5], Ji Qiang[2], Alex Friedman[4], and Andreas Adelmann[6]

[1]Cornell University, Ithaca, NY, USA
[2]Lawrence Berkeley National Laboratory, Berkeley, CA, USA
[3]RadiaSoft LLC, Boulder, Colorado USA
[4]Lawrence Livermore National Laboratory, Livermore, CA USA
[5]Los Alamos National Laboratory, Los Alamos, NM, USA
[6]Paul Scherrer Institut, Villigen, Switzerland

August 2020

Abstract

The need for an accelerator-community wide approach for the computer simulation of particle accelerators has been pointed out in the past and efforts have been made to write codes with wide appeal and community support. However, none have been wide enough to find general support. As a consequence, many codes have fallen out of use or can no longer be supported. Others are used in sub-communities but can not easily be transported to other accelerators or compared to other codes. We therefore intend to use the Snowmass process to promote a toolkit effort for accelerator simulation, along with standardized formats. This toolkit approach has been successful in other fields.

*david.sagan@cornell.edu

1 The Problem

- Accelerator simulation is critical to the design, commissioning, operation, and upgrade of accelerator facilities costing many millions to billions of dollars.
- Accelerator simulation is a large, complex topic.
- Much time and effort has been spent in developing accelerator simulation software.
- Many simulation programs have been abandoned due to developers retiring or moving on to other projects. For example: AGS, ALIGN, COMFORT, DESIGN, DIMAD, HARMON, LEGO, SYNCH, TRACY, TRANSPORT, TURTLE, UAL, etc., etc.[1]
- There is a huge impediment to maintaining these programs due to poorly written code and lack of documentation. Additionally, many of the programs that are available tend to be "rigid". That is, it is generally hard to modify a program to simulate something the program is not already setup to simulate. For instance, adding a new type of lattice element that a particle can be tracked through.
- Abandoned simulation programs represent a huge cost [2]. Not only of the time and money spent in developing a program, but also in the fact that a researcher, who wants to develop a simulation program to simulate something that existing programs do not, will not likely be able (due to time and monetary constraints) to develop from scratch as good a program compared to if the researcher could leverage code that has already been developed.
- A related problem is that with different authors only thinking about the problem at hand, the possible sharing of simulation data between programs is often an afterthought if it is considered at all. The result is that the ability to crosscheck results, which is crucial in validating a programs accuracy, is hindered. A related LoI addresses the importance of common standards for the compatibility of accelerator simulation codes [3].

2 Proposal

- To strengthen interoperability and sustainability of accelerator simulation software, and to make it possible to develop new programs in less time and with less effort and with fewer bugs, it is proposed that a modular "software toolkit" be developed.
- There are many advantages with a toolkit:
  - Cuts down on the time needed to develop programs.
  - Since code modules get reused in different programs, there is a greater chance that bugs will be spotted and therefore the code will tend to have fewer bugs.
  - By having modules that can read and write lattice information and data, the sharing of information between programs is made easier.
- An example of a successful toolkit is Geant4[4] which has helped many researchers solve problems that would not be possible if the researcher had tried to write the simulation code from scratch.

- It is envisaged that this toolkit would be part of a proposed wider community of toolkits. [5]. The policies and standards of this toolkit then would mesh with the ecosystem making it easier to develop the start-to-end simulations that are needed for the next generation of machines. [6].
- Besides strengthening the maintainability and reusability of existing software capabilities, the toolkit should contain new capabilities that are needed by the accelerator community. For example, there is presently no coherent capability for using surface methods to compute transfer maps for realistic beamline elements, although individual (uncoordinated) capabilities exist in various places.
- Along with the toolkit, it is proposed to develop a general purpose, extensible simulation program that can do the common tasks that accelerator physicists routinely do such as Twiss and orbit calculations, nonlinear optimization, lattice design, etc. That is, a program roughly equivalent to what programs like Tao, MAD, or Elegant can do. This alleviates the need for a researcher to have to do programming when the researcher only wants to do a common task like calculating the closed orbit.
- Accelerator simulation toolkits already exist: Accelerator Toolbox [7], Bmad [8], Cosy Infinity [9], Merlin++ [10], Warp [11] and FPP/PTC [12]. So one option would be to further develop one of these existing toolkits. However, all of these toolkits have drawbacks in terms of long term sustainability so it is better to develop a new toolbox. To not "reinvent the wheel", this new toolbox should reuse existing algorithms and code where ever appropriate.

3 Conclusion

- As evidenced by the numerous accelerator simulation programs that are now no longer maintained and cannot be used, much time and effort has been spent "reinventing the wheel" when accelerator simulation software has been developed. Additionally, much time and effort has been spent trying to use outdated software that was not designed for the task at hand or trying to port lattices or data from one program to another.
- The development of an accelerator toolkit plus a general purpose simulation program would save many hours of researchers' time and thus save many millions of dollars. Essentially, this project would pay for itself and then some.
- To be effective, this development should be a community effort. A proposal that would help this effort is the Center(s) for Accelerator and Beam Physics Modeling [13].

References

[1] *Accelerator physics codes*. URL: https://en.wikipedia.org/wiki/Accelerator_physics_codes.

[2] Carole Goble. "Better Software, Better Research". In: *IEEE Internet Computing* 18 (5 2014).

[3] Axel Huebl et al. "Develop/integrate data standards and start-to-end workflows". In: *Snowmass21 LOI* (2020).

[4] J. Allison et al. "Recent developments in Geant4". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 835 (2016), pp. 186–225.

[5] Jean-Luc Vay et al. "A modular community ecosystem for multiphysics particle accelerator modeling and design". In: *Snowmass21 LOI* (2020).

[6] Jean-Luc Vay et al. "End-to-End Virtual Accelerators (EVA)". In: *Snowmass21 LOI* (2020).

[7] A. Terebilo. "Accelerator modeling with MATLAB accelerator toolbox". In: *PACS2001. Proceedings of the 2001 Particle Accelerator Conference (Cat. No.01CH37268)*. Vol. 4. 2001, pp. 3203–3205.

[8] D. Sagan. "Bmad: A relativistic charged particle simulation library". In: *Nuclear Instruments and Methods in Physics Research Section A* 558.1 (2006), pp. 356–359.

[9] Martin Berz. "New features in COSY INFINITY". In: *AIP Conference Proceedings* 297.1 (1993), pp. 267–278.

[10] J. Molson et al. "Advances with merlin - A beam tracking code". In: *IPAC 2010 - 1st International Particle Accelerator Conference* (2010), pp. 1853–1855.

[11] A. Fiedman et al. "3D particle simulation of beams using the warp code". In: *Part. Accel.* 37-38 (1992), pp. 131–139.

[12] E. Forest, Y. Nogiwa, and F. Schmidt. "The FPP and PTC libraries". In: *Proceedings of ICAP 2006* (2006), pp. 17–21.

[13] Jean-Luc Vay et al. "Center(s) for Accelerator and Beam Physics Modeling". In: *Snowmass21 LOI* (2020).

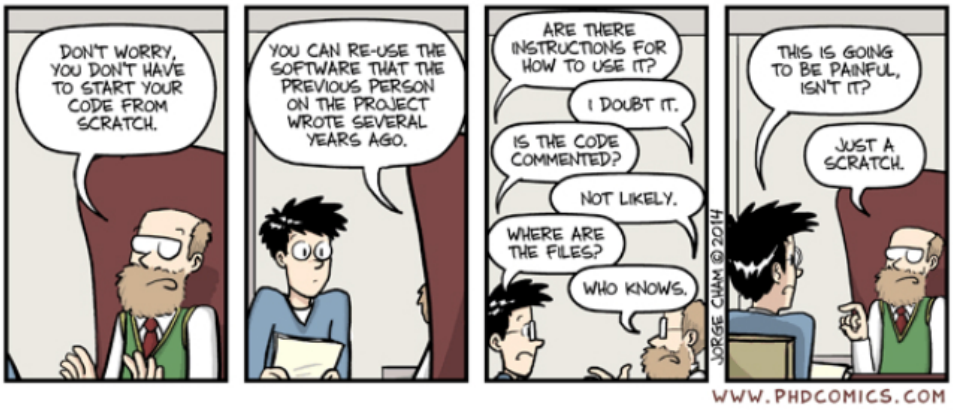

Figure 1: "Piled Higher and Deeper" by Jorge Cham (Cham@www.phdcomics.com). Used with permission.