Published by IOP Publishing for Sissa Medialab
Received: *July 21, 2021*
Accepted: *August 31, 2021*
Published: *October 11, 2021*
ICFA Beam Dynamics Newsletter#82 — Advanced Accelerator Modelling

Simulations of future particle accelerators: issues and mitigations

D. Sagan,[a,*] M. Berz,[b] N.M. Cook,[c] Y. Hao,[d] G. Hoffstaetter,[a] A. Huebl,[e] C.-K. Huang,[f] M.H. Langston,[g] C.E. Mayes,[h] C.E. Mitchell,[e] C.-K. Ng,[h] J. Qiang,[e] R.D. Ryne,[e] A. Scheinker,[f] E. Stern,[i] J.-L. Vay,[e] D. Winklehner[j] and H. Zhang[k]

[a] *Cornell University, Ithaca, NY, 14853, U.S.A.*
[b] *Michigan State University, East Lansing, MI, 48824, U.S.A.*
[c] *RadiaSoft LLC, RadiaSoft LLC, Boulder, CO, 80301, U.S.A.*
[d] *Brookhaven National Laboratory, 98 Rochester St, Upton, NY, 11973, U.S.A.*
[e] *Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, U.S.A.*
[f] *Los Alamos National Laboratory, Los Alamos, NM, 87545, U.S.A.*
[g] *Reservoir Labs Inc., New York, NY, 10012, U.S.A.*
[h] *SLAC National Accelerator Laboratory, Menlo Park, CA, 94025, U.S.A.*
[i] *Fermi National Accelerator Laboratory, Batavia, IL, 60510, U.S.A.*
[j] *Massachusetts Institute of Technology, Cambridge, MA, 02138, U.S.A.*
[k] *Thomas Jefferson National Accelerator Facility, Newport News, VA, 23606, U.S.A.*

E-mail: david.sagan@cornell.edu

Abstract: The ever increasing demands placed upon machine performance have resulted in the need for more comprehensive particle accelerator modeling. Computer simulations are key to the success of particle accelerators. Many aspects of particle accelerators rely on computer modeling at some point, sometimes requiring complex simulation tools and massively parallel supercomputing. Examples include the modeling of beams at extreme intensities and densities (toward the quantum degeneracy limit), and with ultra-fine control (down to the level of individual particles). In the future, adaptively tuned models might also be relied upon to provide beam measurements beyond the resolution of existing diagnostics. Much time and effort has been put into creating accelerator software tools, some of which are highly successful. However, there are also shortcomings such as the general inability of existing software to be easily modified to meet changing simulation needs. In this paper possible mitigating strategies are discussed for issues faced by the accelerator community as it endeavors to produce better and more comprehensive modeling tools. This includes lack of coordination between code developers, lack of standards to make codes portable and/or reusable, lack of documentation, among others.

Keywords: Accelerator modelling and simulations (multi-particle dynamics, single-particle dynamics); Beam dynamics; Beam Optics; Simulation methods and programs

ArXiv ePrint: 2108.11027

*Corresponding author.

© 2021 IOP Publishing Ltd and Sissa Medialab
https://doi.org/10.1088/1748-0221/16/10/T10002

2021 JINST 16 T10002

Contents

2021 JINST 16 T10002

1 Introduction

Particle accelerator simulation is critical to the design, commissioning, operation, and upgrading of accelerator facilities which cost many millions to billions of dollars. Accelerator simulation is a large, complex topic, and much time and effort has been spent in developing simulation software. Nevertheless, in the field of accelerator physics, simulation code development has often been a haphazard affair. Due to developers retiring or moving on to other projects, numerous simulation programs have been completely abandoned or are seldom used. Examples include: AGS, ALIGN, COMFORT, DESIGN, DIMAD, HARMON, LEGO, PETROS, RACETRACK, SYNCH, TRACY, TRANSPORT, TURTLE, UAL to name a few [1, 2].

Oftentimes 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 difficult to modify theses program to simulate something the program is not designed to simulate *a priori*. Adding a new type of lattice element that a particle can be tracked through is one such example.

Abandoned simulation programs represent a huge cost [3], not only in terms of time and money spent in developing a program, but also in terms of researchers leveraging existing technology.

Indeed, a researcher who wants to simulate something that existing programs are unable to, will, due to time and monetary constraints, generally not be able to fully develop a comprehensive simulation program from scratch as compared to what could have been done if existing software could be leveraged.

A related problem involves programmers only considering the problem at hand during software development with the possible sharing of simulation data between programs only considered, if at all, as an afterthought. The result is that the ability to benchmark software, crosscheck results, perform regression tests, and do quality assurance testing for consistency is hindered. These types of checks are crucial in validating software accuracy and persistence.

2 Modeling needs

As simulation programs become more complex due to the ever-increasing demands placed upon machine performance, the situation will become worse if not addressed. Such demands include the accelerator and beam physics Grand Challenges that have been identified recently by the community [4, 5]

- Increasing beam intensities by orders of magnitude.
- Increasing the beam phase-space density by orders of magnitude, towards the quantum degeneracy limit.
- Complete and highly accurate start-to-end "virtual particle accelerators" simulations.
- Fast and accurate multi-objective optimization methods to speed up the design process.

Accelerator and beam modeling software development should allow for extensive testing of new functionality while preserving demonstrated capabilities on previously validated scenarios [6]. Performance and interoperability must be constantly improved to increase understanding (multi-physics problems) and optimization (machine learning). One must also ensure a transfer of knowledge over generations of scientists in form of formal education and easy accessibility to the tools.

Addressing these Grand Challenges will require a community effort to coordinate and modernize the current set of modeling tools, with capabilities that extend far beyond what the current toolset can do, including interdisciplinary simulations and advanced models for virtual prototyping of complete accelerator systems.

2.1 Interdisciplinary simulations

Interdisciplinary simulations are important in a number of areas. In vacuo particle tracking coupled with particle/matter interactions is an example of a growing need. One application is in simulating the radiation induced by "dark current" electrons in accelerating cavities. This radiation may cause damage to cavities which leads to shortened lifetimes of the devices and a radiation safety hazard for the surrounding environment. Dark current induced problems have been observed at many facilities such as the CEBAF [7], LCLS-II [8], ANL, etc. Sufficient shielding is required to properly contain the radiation which in turn requires a good understanding and prediction of radiation levels through simulations. Another example is the modeling of positron production in a target from the impact of

2021 JINST 16 T10002

high-energy electron beams accelerated through a linac injector. These simulations require accurate calculations using electromagnetic RF codes for accelerator structures and beam dynamics codes for particle transport in a beamline to characterize the beam profile before it hits the accelerator enclosure or the target.

Particle/matter simulation codes exist. Examples include Geant4 [9], FLUKA [10], and MARS [11] which have traditionally been used for detector simulation in HEP experiments. However, since these codes and accelerator simulation codes have all been developed without common standards, interfacing them is a laborious task. A seamless simulation requires the proper transfer of field and particle data from accelerator to radiation codes. Communication in a standardized format such as openPMD [12], which has been adopted in some accelerator codes, would help ensure efficient and error-free field and particle data transfers [13]. Another issue with an integrated simulation is in matching of the geometry of the vacuum chamber surface. The surface geometry in accelerator simulations is generally poorly defined if at all. The most comprehensive simulations define the surface using a finite element mesh generated from a CAD model. In contrast, radiation codes generally employ a faceted representation of the CAD model boundary. A converter for mapping finite element curved surfaces to faceted divisions on an interface boundary is required to accurately determine the location of a particle crossing from one computational domain to another. Much time and effort would be saved if the surface geometry descriptions were standardized so that a single converter module could be used in multiple codes.

Increasingly, accelerator simulation tools are also incorporating more micro-physics models to better describe the complex interplay of the various physics phenomena. One particular example being the emission modeling of a high brightness electron photocathode gun. A photocathode gun provides a high brightness electron source for the downstream accelerator beamline where the beam brightness can only be degraded, not improved. Thus, it is essential to understand the cathode emission characteristics and the method to control the beam quality in the gun environment through validated simulations. While Monte-Carlo photo electron emission simulations have been widely employed in studying photocathode performance for dedicated experiments, its potential in integrated simulations has only been explored recently [14]. For such purposes, a tight integration of the micro-physics models into existing gun simulations can be achieved via the best practices and standardization as discussed below.

2.2 Realistic models for virtual prototyping of complete accelerator systems

The increasing demand that accelerator models faithfully predict the performance of future facilities requires the development of more realistic simulation models. Here, two examples are given.

The first example is from single-particle nonlinear dynamics: many accelerator codes use idealized models of beamline elements such as quadrupoles, sextupoles, bends, etc. The simplest models omit fringe fields. Better models are based on a fringe field that is a step function longitudinally. Unfortunately these models contain some terms that are infinite in the hard-edge limit, and codes with hard-edge models typically set these terms to zero [15]. A better approximation is to assume some smooth analytical form for the fringe field. Though this approach is an improvement over the simpler models, it is still an idealization and there is no reason to expect that all of its nonlinear properties will precisely match those of the physical beamline element.

2021 JINST 16 T10002

The precise prediction of single-particle nonlinear dynamics in accelerators can be accomplished using surface methods [16]. These methods have been known for many years but are not yet in widespread use in the accelerator community. The main idea of a surface method is to measure or numerically model the fields of a beamline element on a surface near but within the beam pipe. From there, the fields can be extrapolated inward and are represented by so-called generalized gradients that satisfy Maxwell's equations. In the process, measured or computed errors in the fields at the surface are damped, leading to an accurate representation of the generalized gradients in the beam region. The generalized gradients can then be used to compute realistic transfer maps. These methods are now starting to gain popularity [17–20].

The second example is from the modeling of collective effects: the accurate simulation of 3D radiative phenomena, including coherent synchrotron radiation (CSR). Poisson solvers are essential to modeling space-charge effects in high intensity beams. In fact, there has been tremendous progress in modeling space-charge effects, and 3D parallel Poisson solvers are now ubiquitous. But despite advances in space-charge modeling, the understanding and modeling of 3D radiative phenomena remains an open problem. Most simulations involving CSR use a 1D model [21]. More complicated 2D, 2.5D, and even some 3D models exist. However, these models, particularly the 3D models, sometimes involve questionable approximations and are extremely slow. There is presently no code that can accurately and reliably model 3D CSR effects including transient effects with a performance that makes it useful as a beamline design tool. As accelerators push boundaries with higher brightness beams, the ability to accurately model 3D CSR will become a key issue. This is true for accelerators with very high peak current beams such as those in beam-driven and laser-driven plasma accelerators. It is also true for future X-Ray free electron lasers (FELs) where complicated beam gymnastics involving bright electron beams is used to prepare the beam prior to entering the FEL.

The preceding describes just two examples among many of the need for more realistic models of accelerator components and phenomena. The development and application of new, more realistic models would enable virtual prototyping of entire accelerator systems including their nonlinear properties. It would allow the precise prediction of important properties — nonlinear dynamics, the thresholds for collective instabilities, etc. — before a beamline is constructed and reduce the need for magnet shimming, the use of nonlinear correctors, etc. Ultimately such advanced capabilities would reduce cost, reduce risk, and improve the performance of future accelerator facilities.

3 Software sustainability

There are several aspects that must be addressed to enable the development of the quality software that will be needed for the machines of tomorrow. One facet can be put under the rubric of "software sustainability" which can be defined as [22]:

> "the capacity of the software to endure. In other words, sustainability means that the software will continue to be available in the future, on new platforms, meeting new needs."

There are many aspects to software sustainability, as illustrated in figure 1. Broadly, these aspects can be grouped into the "intrinsic" characteristics of the software itself and the "extrinsic"

2021 JINST 16 T10002

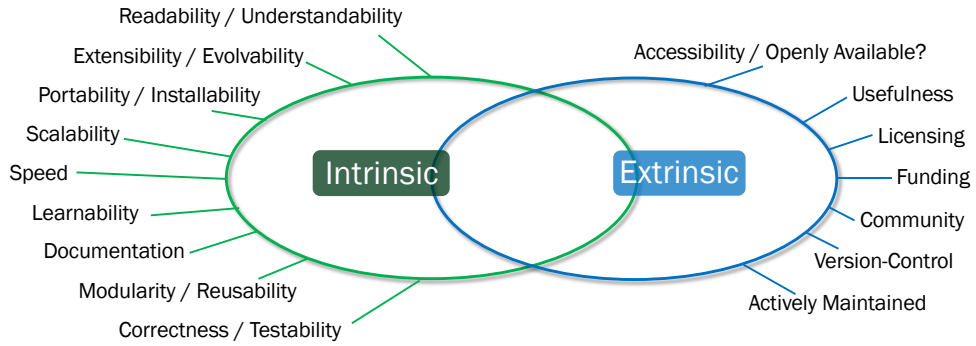

Figure 1. There are a number of aspects that make software sustainable. Broadly, they can be grouped into the "intrinsic" characteristics of the software itself and the "extrinsic" environment in which the software is developed and used.

environment in which the software is developed and used. Software sustainability has been studied academically and there is even a Software Sustainability Institute [23], which promotes "the advancement of software in research by cultivating better, more sustainable, research software to enable world-class research".

As mentioned above, many software packages developed for simulating accelerators, while showing excellent results for their specific application and their era, are not "sustainable" in the long run. However, software sustainability is extremely important given the limited resources that the accelerator community has for code development in conjunction with the even more limited resources for maintaining codes. To meet future needs, it is imperative that there is a community wide effort to promote sustainable practices.

3.1 Software toolkits

One aspect of developing sustainable software is the creation of software "toolkits". A software toolkit is an integrated set of modular routines that are used to develop and maintain applications. As well as strengthening interoperability and sustainability of simulation software, a toolkit makes it possible to develop new programs in less time, with less effort, and with fewer bugs. There are many advantages to organizing software via a toolkit. This includes:

- Increased safety, since modular code provides a firewall. For example, a buggy module introduced into the toolkit will not affect programs that do not use it.
- A greater chance that bugs will be spotted since code modules get reused in different programs and therefore get greater scrutiny.
- By having modules that can read and write lattice information and data, the sharing of information between programs is made easier.

A well-known example is Geant4 [9], which is a toolkit for the simulation of the passage of particles through matter. Geant4 has helped many researchers solve problems that would not be possible if a given researcher had tried to write the simulation code from scratch.

2021 JINST 16 T10002

It is envisaged that toolkits would be developed for different accelerator physics purposes [24]. As described in more detail in other sections, an important facet to maximizing sustainability would be to have accelerator community-wide policies and standards [13] that would then enable the meshing of the toolkits into an ecosystem [25], making it easier to develop the start-to-end simulations that are needed for the next generation of machines [5].

Along with toolkit development, there would be a need to develop general purpose, extensible simulation programs 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 [26], MAD [27], or Elegant [28] 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.

Besides the aforementioned general accelerator physics components, it is also important to have components for specific phenomena. Consider, for example, a Poisson solver library for modeling space-charge effects in high intensity/high brightness beams [29]. Subject to different boundary conditions, those Poisson solvers involve different numerical methods that solve Poisson's Equation on a grid. For an open boundary condition, a FFT-based Green's function method could be used. For a closed boundary condition with regular shape, a finite difference spectral method could be used. For a closed boundary condition with irregular shape, a multigrid finite difference method is often used. Rather than re-implementing the same functionalities (FFT-based or multigrid Poisson solver) multiple times in separate, incompatible and uncoordinated codes, the community would benefit greatly from consolidation in the development of a few selected toolkits for reuse across codes.

Accelerator simulation toolkits already exist: Accelerator Toolbox [30], Bmad [31], Cosy Infinity [32], Merlin++ [33], Warp [34], IMPACT [35] and FPP/PTC [36]. Moving forward, one option would be to further develop existing toolkits. Another approach that would strengthen long-term sustainability would be to develop a new toolbox based on modern software practices. To not "reinvent the wheel", this new toolbox should reuse existing algorithms and code wherever appropriate [25].

3.2 Software ecosystem — interoperability and policies

Simulations of accelerator facilities can extend far beyond the accelerator itself. For example, in an XFEL, a comprehensive simulation will go from the creation of the X-ray pulses produced by electrons in an undulator, all the way to the simulation of the X-ray experiment. This includes X-ray transport, photon-sample interaction, signal transport, detector response, and data analysis. Realistically, no single software package or toolkit would cover all of these domains.

A software "ecosystem" is the set of libraries (toolkits) and applications that are developed somewhat independently but with a common set of rules and standards [13, 25]. An ecosystem facilitates the implementing of frameworks for start-to-end workflows that span over multiple components. Recently developed examples of such frameworks include LUME [37], PaNOSC [38], and SIMEX [39]. Such integration platforms aid in the rapid adoption for new design and analysis needs, using complex tools in automated workflows such as needed for AI/ML and optimization.

As of today, data exchange between individual applications is often based on widely supported file formats that implement a common meta-data schema, see section 3.3. One of the opportunities with a compatible software ecosystem would be the further adoption of such standards as well as

2021 JINST 16 T10002

abstraction of low-level data layouts, which would foster innovative combinations of toolkits with bespoke software into new applications for tightly coupled models. A large community of developers can focus on domain-specific needs and help improve software capability and sustainability across projects, while reducing maintenance and development time due to common practices and fundamentals developed by specialists. Fundamental building blocks would also address performance and portability needs in low-level libraries. This could include support, for example, of, CPUs, GPUs, multi-node communication patterns, and multi-platform compiling. Inefficiencies and bugs are spotted earlier due to re-use. A progressive path for both adoption and continued modernization of such components is possible, starting from existing libraries.

Beyond the data exchange through common file formats and meta-data schemata, it is sometimes desirable to exchange data among the simulation tools in a more integrated, finer-grained manner that may be at the level of each simulation time step [4, 14, 40]. This may happen when multiple physics effects are to be simultaneously simulated or when coupling multiple solvers, where data exchange can be either in single direction or bi-directional. Such an in-memory or cross-node data transfer can be made to respect physics constraints (conservation, divergence free conditions, etc.) and is also much more efficient than file I/O based transfers [41–43]. This need can be best served with the common low-level data layouts and interfaces as mentioned above and/or by promoting/enabling interoperable mesh/particle capabilities among existing tools in the ecosystem.

In order to advance accelerator modeling further in the direction of a compatible, extensible ecosystem, some general standardization needs, including best practices in software development, have been identified [25]. These are based on community policies that have been established over years by teams of specialists in scientific software development and computing. For instance, the accelerator community could be leveraging on the Interoperable Design of Extreme-scale Application Software (IDEAS) project [44] and its Extreme-scale Scientific Software Development Kit (xSDK) [45].

An ecosystem would be best developed on the basis of permissive open source licenses to allow reuse, cross-institutional and international collaboration and adoption. For vertical software integrations, wide-ranging open source libraries from numerical solvers to optimizers (e.g., Hypre [46], libEnsemble [47]), various meshing and mesh-refinement (AMReX [48]), and particle (for example, CoPA Cabana [49]) and mesh/particle remapping (e.g., Portage [50]) libraries could be combined as a foundation for the accelerator and beam physics components.

3.3 I/O standardization

Traditionally, existing accelerator modeling applications were driven by individual groups with little coordination between development activities. This is detrimental for a number of reasons including hindering the ability to benchmark the software, and crosscheck results. Indeed, it is the consensus within the computer science and majority of computational science sub-domains that computational results need to be reproducible and independently replicable [51].

As accelerator software becomes more complex, and there is a strong desire to integrate capabilities (such as new methods, start-to-end modeling, common analysis needs and machine-guided optimization), standards for data and workflows can help increase productivity and sustainability [13]. Recently, activities such as those supported by the Consortium for Advanced Modeling of Particle Accelerators (CAMPA) [52], helped coordinate the standardization of data exchange and

2021 JINST 16 T10002

simulation control, with the aim of connecting large existing frameworks and enabling innovative workflows. Such standardization efforts can also provide the basis for an integration into scientific data portals to curate and re-use modeling data [53–55]. Besides improving reproducibility of simulations and preserving value of collected data, such efforts can also aid meta-studies and tests for new theories.

With respect to accelerator simulation data, historically, there exists a variety of low-level data file formats, which individual modeling tools use with custom-made meta-data conventions to express the domain-specific data consumed and generated by individual tools. Yet new, efficient data formats and highly-tuned I/O libraries are continuously being developed by computer scientists and existing paradigms, such as POSIX file-I/O, are overtaken by modern approaches such as data streams, object storage and relaxed I/O-constrains in highly-parallel computing environments. Adoption of these low-level file formats for accelerator modeling is needed to utilize the progress in modern storage and data transport technology and overcome bottlenecks arising from file-based storage of high-fidelity data as well as manual data analysis and curation efforts.

The Open Standard for Particle-Mesh Data Files (openPMD) [12] successfully demonstrates that defining compatible meta-data in a file-format agnostic organization for accelerator and beam data is possible, while using scalable, modern file formats from computer science [56–58]. OpenPMD is organized around a written, versioned text document that is supported by tooling for validation, examples, a project catalogue, libraries and programs, which all have their respective documentation and tutorials. Individual compatible projects, software, and data are published by a variety of authors [59].

Complementary to openPMD, the Standard Input Format for Particle-In-Cell Codes (PICMI) addresses the challenge of unified simulation design by defining a common input layer that focuses on the physical description of a problem set [60]. Currently, PICMI is implemented as a high-level Application Programming Interface (API), which is an approach similar to successful community math and HPC APIs, for example, BLAS for linear algebra and MPI for multi-node message passing on supercomputers. The programming language used for the PICMI API is Python, which is a well-known scripting language suitable for rapid simulation design and backed by a vast, extensible software ecosystem.

4 Advanced concepts and future computing

4.1 Advanced concepts

Advanced Accelerator Concepts (AAC) offer accelerating gradients that go beyond the limitations of standard RF technologies, sometimes by an order of magnitude or more, leading to the prospect of much more compact — and in some cases proportionally cheaper — technologies. AAC includes laser-driven plasma acceleration (aka LWFA or LPA), charged-particle-beam-driven plasma acceleration (aka PWFA), structure-based wakefield accelerators (SWFA), dielectric laser acceleration (DLA) or laser-ion acceleration. While the modeling of DLA involves mostly simulation tools that are already used for conventional accelerators, the modeling of the other schemes involve different models (and thus simulation tools) and can be significantly more challenging computationally, in particular for those involving plasmas. The description of the specific needs and challenges of

2021 JINST 16 T10002

these tools is given in another paper of this issue [61]. Aside from the different physics, numerical methods and computational needs, other challenges and solutions are essentially the same as for conventional accelerators with regard to, e.g., software sustainability, standardization, validation, verification, usability, integrated workflows and frameworks, toolkits, ecosystems, centers, evolving and future architectures (including quantum computing) that are described in this paper.

In terms of numerical approaches, advanced concepts include the need to investigate novel numerical methods and approaches beyond Particle-In-Cell (PIC) methods [35, 62–64] that have shown promise for future incorporation into simulation software as modules. These approaches could potentially tackle different types of advanced simulation problems. Examples include symplectic multiparticle space-charge simulations [65, 66], Fast Multipole Method (FMM)-based approaches [67–71], boundary integral solvers [72–74] and hybrid solvers such as *Vico-Greengard-Ferrando* [75, 76].

4.2 Evolving architectures in standard computing

Developments in existing and planned DOE High Performance Computing (HPC) facilities indicate an increasing reliance on architectures with attached co-processors, styled as GPU computing. These processors offer an attractive boost in computing performance with respect to any of the common denominators such as hardware cost, electricity cost, or wall clock time to accomplish a particular workload. This boost in performance is achieved by implementing highly parallel computational engines acting on localized data at the expense of general purpose computing capabilities. Computational problems whose algorithms can be cast in this paradigm benefit greatly from this style of computing architecture.

Fortunately, many accelerator simulation problems are within this class. Propagating many independent particles through beamline components is seemingly custom made for this kind of computing. The issue facing the field then is that programming these devices requires specialized techniques. Typically, data for computations should be transferred to co-processor memory and arranged carefully for efficient parallel processing. This transfer is usually slow and should be minimized or eliminated over the course of a long computation. It may not be possible to implement some current algorithms on co-processor hardware; new algorithms will need to be developed.

Computing on current general purpose processors will continue to be an important part of the landscape. Software should be built to run on either standard hardware or new co-processor based architectures, of which there are several. Although the most well-established solutions for GPU computing are provided by NVidia, AMD and Intel are also building HPC systems. Some accelerator simulation codes have been implemented in CUDA, the NVidia specific GPU programming language which is specific to NVidia GPUs. Ideally, the accelerator simulation community would avoid dependence on a single co-processor supplier. Fortunately, there are several major software efforts to build platform-independent parallel computing frameworks that can support execution on either CPU or GPU processors with a single high-level code base. No particular framework is clearly superior to the others, and each is targeted for use in a particular language (C++ or Fortran) and level of abstraction. The important lesson is that the community and maintainers of import simulation tools should be supported in either upgrading or re-implementing their algorithms for use in the near and medium future on what will be the dominant scientific computing architecture.

2021 JINST 16 T10002

4.3 Adaptively tuned simulations as online virtual diagnostics

Some of the most detailed accelerator diagnostics are X-band transverse deflecting cavities. However, deflecting cavities are limited to a resolution of 1 µm or 3.3 fs/pixel, and destroy the beam during the measurement process [77]. This measurement floor is of concern since beams at advanced plasma wakefield accelerators and free electron laser (FEL) facilities are starting to exceed those limits. The Facility for Advanced Accelerator Experimental Tests (FACET-II) will provide bunch lengths as low as (1 µm or $\sim$ 3 fs) at 12 GeV [78], attosecond two-color X-ray pulses have been achieved at the SwissFEL [79], and designs for the international linear collider call for 3.2 nC bunches with 30 µm bunch lengths at 250 GeV [80].

As bunch lengths decrease beyond the resolution of existing diagnostics, simulations will be relied on as virtual diagnostics. Simulation-driven approaches are common in other scientific fields, such as coherent diffraction imaging where physics simulations translate 2D X-ray diffraction intensity measurements to 3D electron densities of crystals. Using simulations as online diagnostics requires a close match with accelerator performance. This is a non-trivial problem even for particle-tracking codes with millions of macro-particles because once accelerators are built they do not perfectly match the designs that simulation models are based on. During installation misalignments are introduced, the electromagnetic fields of components do not perfectly match simulated fields, and once operational the components and the initial input beam distributions drift with time and are perturbed by disturbances not accounted for.

Closely matching simulations to accelerators requires adaptive feedback. Diagnostics can be compared to simulation-based predictions and simulation parameters can be tuned in real-time to achieve a match between measurements and simulation outputs. Once a match is achieved, because of physics constraints within the simulation, it is likely that other beam properties are uniquely matched. Recently a LiTrack model of FACET was adaptively tuned online to match the simulated beam's energy spread spectrum $\hat{\rho}_E(x,t)$ to its measured $\rho_E(x,t)$, minimizing the error $e(t) = \int |\hat{\rho}_E(x,t) - \rho_E(x,t)|\, dx$, to track the time-varying longitudinal phase space (z, E) of the electron beam [81, 82]. Efforts are also underway to utilize ML tools to map diagnostics back to input beam distributions to be used as the initial conditions of accelerator models [83]. Such approaches are possible with any simulation tool for tracking time-varying beams. By taking advantage of GPUs and field programmable gate arrays it may be possible to use such adaptive models as virtual diagnostics in real-time shot-to-shot for high repetition rate machines.

4.4 Getting ready for the arrival of quantum computing age

Studies of collective effects are essential for modern accelerators with high intensity beams. The start-to-end simulation of an accelerator using real beams with billions or more particles remains challenging and expensive even with state-of-the-art exascale machines. The development of quantum computers provides new opportunities to enhance the particle accelerator community's simulation abilities.

A quantum computer is a device that utilizes the special properties of quantum mechanics to perform computation, which can potentially provide an exponential improvement in efficiency for some classes of simulations. The three properties of quantum superposition, interference, and entanglement of a quantum state make quantum computing different from classical computing.

2021 JINST 16 T10002

Due to quantum superposition, the information stored in a quantum system scales exponentially with the number of qubits as opposed to the linear scaling with the number of bits in a classical system. Both commercial companies and research institutes are developing techniques for building quantum computers. Current state-of-the-art quantum computers have above 50 qubits and quantum supremacy has been demonstrated on some specific problems [84, 85]. Quantum computing is currently available to the public through cloud services provided by some commercial companies such as IBM [86], D-Wave [87], Amazon [88] and Microsoft [89]. Meanwhile, in academia, studies on quantum algorithms have also been booming in the past few years. The work that is most relevant for accelerator modeling relates to solving linear systems [90–94] as well as ordinary differential equations (ODEs) and partial differential equations (PDEs) [95–99]. Solving Poisson's equation and the Vlasov equations — which are often used in the simulation of collective effects such as space charge, the beam-beam interaction, and coherent synchrotron radiation — with quantum computing is being explored [100–102]. Quantum computer simulators, as code developing-, debugging-, and testing platforms, are available for almost all mainstream programming languages, e.g. C/C++, Python, Java, Matlab, etc. There also exist some languages specifically designed for quantum computing [103]. All these provide the community with the fundamental blocks to build simulation tools for beam and accelerator physics.

Most previous work on quantum algorithms focused on the realization of an algorithm with quantum circuits, but applying the algorithm to solve a practical scientific problem is seldom discussed [104]. Clearly there is a gap between the development of the algorithms and their implementation. To make a problem suitable for quantum simulations, it has to be described by unitary operators so that the system, which may be classical, is mathematically equivalent to a quantum system. Additionally, in quantum computing, all variables are stored in quantum states. Preparing the initial states and reading out the results accurately can be time intensive. The traditional way of using a large number of particles in simulation and producing all of their coordinates as output would be unpractical.

Given the above considerations, at least in the near future, a quantum computer will not replace a classical computer but will probably work together with one. It is thus probably better for now to focus on how to make new quantum simulation tools that collaborate with the existing pool of accelerator modeling programs. Ideally a protocol will be invented through which the quantum packages could be called by the classical programs. Also needed are innovative analyzing tools to process the simulation results, which may happen before the reading-out. To achieve these tasks, the accelerator physics problems that will benefit from quantum computing need to be identified and for each of them the proper mathematical model will need to be established, which may be different from the conventional classical one. For this, contributions from experts in both accelerator physics and quantum computing are required.

5 Centers for accelerator and beam physics modeling

It quickly becomes clear that, in order to achieve what is described and proposed in the other sections, a coherent and consolidated effort is needed. This is best achieved in the form of dedicated Centers for Accelerator and Beam Physics Modeling [105]. Other areas of computer science have already embraced this fact. New colleges for computing are established at universities to consolidate

2021 JINST 16 T10002

the dispersed computing efforts of the various departments (e.g. MIT's Schwarzman College of Computing [106]), and new centers for Quantum Computing [107, 108] have been built. Exascale computing has been embraced through the Exascale Computing Project [109]. The US Department of Energy (DOE) has founded SciDAC [110] to accelerate progress in scientific computing across the different programs supported by DOE: Advanced Scientific Computing Research, Basic Energy Sciences, Biological and Environmental Research, Fusion Energy Sciences, High-Energy Physics, and Nuclear Physics.

The respective communities have benefited strongly from these new centers and the partnerships across disciplines.

Accelerator and Beam Physics Modeling would no doubt benefit similarly. The centers can be at a given location or distributed geographically and among institutions across laboratories, academia and industrial partners. They would bring together domain scientists (computational accelerator and beam physicists), applied mathematicians, computer scientists and software engineers with collaborations across the full landscape of accelerator modeling. In addition, some of the computer science centers mentioned above are already supporting accelerator modeling efforts on which the Centers for Accelerator and Beam Physics Modeling could build.

Depending on the overall size, the centers could enable part or all of the following:

- Community development and maintenance of codes using industry-standard quality processes by dedicated, specialized teams [6].
- Collect libraries for field solvers, particle trackers, and other modules.
- Provide a modular community ecosystem for multiphysics particle accelerator modeling and design [25].
- Standardize input scripts, output data, lattice description and start-to-end workflows [13].
- Provide compatibility layers to use the same libraries and modules in a number of programming languages.
- Development and maintenance of End-to-end Virtual Accelerators (EVA) [5].
- User support, high-quality and detailed documentations, online tutorials, and training.
- Easy-to-use, standardized, user interfaces for preparation and analysis of simulations.
- Automated tools for ensemble simulations for optimization with builtin AI/ML support.
- Suite of test problems with well-characterized solutions for benchmarking, quality assurance and regression testing.
- Development, analysis and efficient implementation of novel algorithms and numerical methods (e.g., high-order solvers, symplectic multiparticle tracking, Fast Multipole Methods, adaptive mesh refinement).
- Providing a space to meet (physically or virtually) for the integration of developments from contributors into larger codes, such as PhD projects from external groups, organizing development hackathons, knowledge-transfer, and onboarding.

2021 JINST 16 T10002

- Developing and organizing workshops for developers and users of codes alike. Inviting national and international speakers/developers (travel/hosting funds).
- Interacting with existing schools, by developing and maintaining state-of-the-art educational resources (e.g. tutorials, lectures) on codes.
- Exploration of novel use of machine learning for accelerator modeling, and, further in the future, of quantum computing [111].

Multiple Centers can be organized through a Consortium (e.g., CAMPA [52]). Except for special restrictions such as export control, it would be desirable for the software developed by the Center to be open source, enabling crosschecking, testing and contribution by the community at large, beyond the participants to the Center(s) [112].

6 Conclusion

The historically disorganized nature of accelerator software development has been a major impediment to creating the quality applications that are needed to both run existing machines as well as to design future ones:

> *Computer simulations play an indispensable role in all accelerator areas. Currently, there are many simulation programs used for accelerator physics. There is, however, very little coordination and cooperation among the developers of these codes. Moreover there is very little effort currently being made to make these codes generally available to the accelerator community . . .*
>
> — HEPAP report, 2015

Consider the case of the Superconducting Super Collider (SSC) [113]. The SSC was terminated, after millions of dollars were spent, in part due to a flawed simulation. The decision to increase the vacuum chamber bore diameter from 4 cm to 5 cm was based in part upon a faulty simulation using the code SSCTRK [114]. This code was newly developed for the SSC [115] and had not been thoroughly vetted at the time the decision was made. After refinements to the Monte Carlo error model, and after upgrading the SSCTRK code, it was realized that the original 4 cm bore had been adequate. The unnecessary enlargement to 5 cm lead to cost increases along with some turmoil caused by having to redesign vacuum chambers, magnets and other machine components. Ultimately, this was a major factor in the demise of the SSC.

Communities in the computational sciences face similar challenges, and physics modeling groups would benefit from incorporating research results, best practices, participate actively in aforementioned bodies, and anticipate trends in the broader computer science and computational science community. In sustainable workflows, one wants to avoid heroic efforts relying on few individuals, and instead provide an environment that is inclusive and thrives with contributions from various educational backgrounds. Physics groups that integrate early with external computer science, applied mathematics and computational physics efforts can share modular solutions [25], drive cross-domain visions and avoid missing out on solutions developed in related scientific domains.

2021 JINST 16 T10002

There have been notable improvements in recent years. Some are centered around novel, open source particle-in-cell codes for laser-plasma modeling and around collaborations on standards [13]. Yet other projects are essentially walled gardens with varying access levels to simulation programs, their source code, documentation, support, and usage rights. Another challenge lies in the publication of computational work. Analysis routines, source codes, inputs, and simulation data are often not openly archived in sufficient detail, which hinders reproducibility and adoption of published methods by other groups. This situation poses a significant risk and calls for an advancement of modeling practices that can adequately address the needs of decade-long basic science projects.

Part of the problem is that, traditionally, performance metrics of scientific success aim solely on publication numbers. The extra time and effort to make code sustainable is, by this criterion, unproductive. The end result, however, is extensive waste. It is imperative that funding for software sustainability be made available to the accelerator community. Ultimately, such funding will pay for itself many times over.

Acknowledgments

We acknowledge the support of the Office of Science, Office of High Energy Physics, of the U.S. Department of Energy (DOE) under Contract numbers DEAC02-05CH11231, DE-SC0018719 and DE-AC02-07CH11359 along with the Exascale Computing Project (No. 17-SC-20-SC), a collaborative effort of the DOE's Office of Science and National Nuclear Security Administration. Work was also supported by DOE contracts DE-SC0018370, DE-FG02-08ER41546 and DE-SC0018636 as well as the DOE Office of Science, Office of Nuclear Physics under contract DE-AC05-06OR23177. Part of this work was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project numbers 20180670ER and 20190131ER.

We acknowledge the support by the US National Science Foundation under award number PHY-1505858, as well as the Bose Foundation.

References

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

[2] A.W. Chao, K.H. Mess, M. Tigner and F. Zimmermann, eds., *Handbook of accelerator physics and engineering*, World scientific (2013).

[3] C. Goble, *Better software, better research*, *IEEE Internet Comput.* **18** (2014) 4.

[4] S. Nagaitsev et al., *Accelerator and Beam Physics Research Goals and Opportunities*, submitted to *Snowmass21* [https://snowmass21.org/docs/files/summaries/AF/SNOWMASS21-AF1_AF7_S_Nagaitsev-056.pdf], [arXiv:2101.04107].

[5] J.-L. Vay, D. Sagan, A. Huebl, M. Thévenet, R. Lehe, C.-K. Ng et al., *End-to-End Virtual Accelerators (EVA)*, submitted to *Snowmass21* [https://snowmass21.org/docs/files/summaries/CompF/SNOWMASS21-CompF2_CompF0-AF1_AF0_Vay-067.pdf].

[6] R. Lehe, A. Huebl, J.-L. Vay, A. Friedman, M. Thevenet, C. Mitchell et al., *Embracing modern software tools and user-friendly practices, when distributing scientific codes*, submitted to *Snowmass21* [https://snowmass21.org/docs/files/summaries/CompF/SNOWMASS21-CompF2_CompF0_Lehe-076.pdf].

2021 JINST 16 T10002

[7] C. Hovater, T. Allison, G. Biallas, R. Bachimanchi, E. Daly, M. Drury et al., *Operation of the cebaf 100 mv cryomodules*, in *Proceedings of Linear Accelerator Conference*, East Lansing, MI, U.S.A., 25–30 September 2016, pp. 65–67.

[8] J.R. Lewandowski, R.C. Field, A.S. Fisher, H.-D. Nuhn and J.J. Welch, *RF gun dark current suppression with a transverse deflecting cavity at LCLS*, in *Proceedings of the 37th International Free Electron Laser Conference*, Daejeon, Korea, 23–28 August 2015, pp. 583–586.

[9] J. Allison et al., *Recent developments in Geant4*, *Nucl. Instrum. Meth. A* **835** (2016) 186.

[10] A. Ferrari, P.R. Sala, A. Fasso and J. Ranft, *FLUKA: A multi-particle transport code (Program version 2005)*, https://doi.org/10.2172/877507.

[11] N.V. Mokhov and C.C. James, *The MARS Code System User's Guide Version 15 (2016)*, https://doi.org/10.2172/1462233.

[12] A. Huebl, R. Lehe, J.-L. Vay, D.P. Grote, I. Sbalzarini, S. Kuschel et al., *openpmd: A meta data standard for particle and mesh based data*, https://doi.org/10.5281/zenodo.591699.

[13] A. Huebl, J.-L. Vay, R. Lehe, M. Thévenet, C. Mayes, D. Sagan et al., *Develop/integrate data standards and start-to-end workflows*, submitted to *Snowmass21* [https://snowmass21.org/docs/files/summaries/CompF/SNOWMASS21-CompF2_CompF3-AF1_AF6_Lehe-075.pdf].

[14] C.-K. Huang, T. Kwan, V. Pavlenko, C.-K. Ng and E. Wang, *Physics-based high-fidelity modeling of high brightness beam injectors*, submitted to *Snowmass21* [https://snowmass21.org/docs/files/summaries/AF/SNOWMASS21-AF7_AF1-CompF2_CompF0_Huang-183.pdf].

[15] R. Ryne and A. Dragt, *Numerical computation of transfer maps using lie algebraic methods*, in *Proceedings of the 12th IEEE Particle Accelerator Conference*, Washington, DC, U.S.A., 16–19 March 1987, pp. 1081–1083.

[16] R. Ryne, D. Abell, M. Berz, D. Bruhwiler, A. Dragt, K. Makino et al., *Surface methods for precision accelerator design and virtual prototyping of accelerator systems*, submitted to *Snowmass21* [https://snowmass21.org/docs/files/summaries/CompF/SNOWMASS21-CompF2_CompF0_Robert_Ryne-071.pdf].

[17] M. Borland, R. Lindberg and R. Soliday, *Tools for use of generalized gradient expansions in accelerator simulations*, in *Proceedings of IPAC*, Campinas, SP, Brazil, 24–28 May 2021, MOPAB059.

[18] L. Bojtár, *Frequency analysis and dynamic aperture studies in a low energy antiproton ring with realistic 3d magnetic fields*, *Phys. Rev. Accel. Beams* **23** (2020) 104002.

[19] S. Manikonda and M. Berz, *Multipole expansion solution of the laplace equation using surface data*, *Nucl. Instrum. Meth. A* **558** (2006) 175.

[20] S. Manikonda and M. Berz, *An accurate high-order method to solve the helmholtz boundary value problem for the 3d laplace equation*, *Int. J. Pure Appl. Math.* **23** (2005) 365.

[21] E. Saldin, E. Schneidmiller and M. Yurkov, *Analytical treatment of the radiative interaction of electrons in a bunch passing a bending magnet*, *Nucl. Instrum. Meth. A* **407** (1998) 112.

[22] D.S. Katz, *Scientific software challenges and community responses*, https://www.slideshare.net/danielskatz/scientific-software-challenges-and-community-responses, 2015.

[23] https://www.software.ac.uk/.

2021 JINST 16 T10002

[24] D. Sagan, A. Huebl, J.-L. Vay, D. Bruhwiler, R. Ryne, C.-K. Ng et al., *Beam Dynamics Toolkit*, submitted to *Snowmass21* [https://snowmass21.org/docs/files/summaries/CompF/SNOWMASS21-CompF2_CompF0_Sagan-077.pdf].

[25] J.-L. Vay, A. Huebl, D. Sagan, D. Bruhwiler, R. Lehe, C.-K. Ng et al., *A modular community ecosystem for multiphysics particle accelerator modeling and design*, submitted to *Snowmass21* [https://snowmass21.org/docs/files/summaries/CompF/SNOWMASS21-CompF2_CompF0-AF1_AF0_Vay-070.pdf].

[26] D. Sagan and J. Smith, *The TAO accelerator simulation program*, in *Proceedings of the 2005 Particle Accelerator Conference*, Knoxville, TN, U.S.A., 16–20 May 2005, pp. 4159–4161.

[27] C. Iselin, *The mad program*, in *Computing in Accelerator Design and Operation*, W. Busse and R. Zelazny, eds., Springer Berlin Heidelberg, 1984, pp. 146–151.

[28] M. Borland, *elegant: A Flexible SDDS-Compliant Code for Accelerator Simulation*, in *Proccedings of the 6th International Computational Accelerator Physics Conference*, Darmstadt, Germany, 11–14 September 2000, LS-287.

[29] J. Qiang, H. Zhang, J.-L. Vay, A. Huebl, D. Grote, K. Sonnad et al., *A Parallel Poisson Solver Library for Accelerator Modeling Applications*, submitted to *Snowmass21* [https://snowmass21.org/docs/files/summaries/CompF/SNOWMASS21-CompF2_CompF0-050.pdf],

[30] A. Terebilo, *Accelerator modeling with matlab accelerator toolbox*, in *Proceedings of the 2001 Particle Accelerator Conference*, Chicago, IL, U.S.A., 18–22 June 2001, vol. 4, pp. 3203–3205.

[31] D. Sagan, *Bmad: A relativistic charged particle simulation library*, *Nucl. Instrum. Meth. A* **558** (2006) 356.

[32] M. Berz, *New features in COSY INFINITY*, *AIP Conf. Proc.* **297** (1993) 267.

[33] J. Molson, H. Owen, A. Toader and R. Barlow, *Advances with Merlin — a beam tracking code*, in *Proceedings of the 1st International Particle Accelerator Conference*, Kyoto, Japan, 23–28 May 2010, pp. 1853–1855.

[34] A. Fiedman, D.P. Grote, D.A. Callahan, B.A. Langdon and I. Haber, *3D particle simulation of beams using the warp code*, *Part. Accel.* **37–38** (1992) 131.

[35] J. Qiang, R.D. Ryne, S. Habib and V. Decyk, *An object-oriented parallel particle-in-cell code for beam dynamics simulation in linear accelerators*, *J. Comput. Phys.* **163** (2000) 434.

[36] E. Forest, Y. Nogiwa and F. Schmidt, *The FPP and PTC libraries*, in *Proceedings of ICAP*, Chamonix, France, 2–6 October 2006, pp. 17–21.

[37] C.E. Mayes, P.H. Fuoss, J.R. Garrahan, H. Slepicka, A. Halavanau, J. Krzywinski et al., *Lightsource unified modeling environment (lume), a start-to-end simulation ecosystem*, in *Proceedings of IPAC*, Campinas, SP, Brazil, 24–28 May 2021, THPAB217.

[38] *The Photon and Neutron Open Science Cloud (PaNOSC)*, https://www.panosc.eu/.

[39] *PaNOSC-ViNYL/SimEx*, https://github.com/PaNOSC-ViNYL/SimEx, 2020.

[40] D. Dimitrov, D. Bruhwiler, J. Cary, P. Messmer, P. Stoltz, K. Jensen et al., *Development of advanced models for 3d photocathode pic simulations*, in *Proceedings of the 2005 Particle Accelerator Conference*, Knoxville, TN, U.S.A., 16–20 May 2005, pp. 2583–2585.

2021 JINST 16 T10002

[41] F. Zhang, T. Jin, Q. Sun, M. Romanus, H. Bui, S. Klasky et al., *In-memory staging and data-centric task placement for coupled scientific simulation workflows*, *Concurrency Comput. Pract. Exp.* **29** (2017) e4147.

[42] U. Ayachit, A. Bauer, E.P.N. Duque, G. Eisenhauer, N. Ferrier, J. Gu et al., *Performance analysis, design considerations, and applications of extreme-scale in situ infrastructures*, in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, Salt Lake City, UT, U.S.A., 13–18 November 2016, pp. 921–932, 2016.

[43] M. Larsen, J. Ahrens, U. Ayachit, E. Brugger, H. Childs, B. Geveci et al., *The alpine in situ infrastructure: Ascending from the ashes of strawman*, in *Proceedings of the In Situ Infrastructures on Enabling Extreme-Scale Analysis and Visualization*, Denver, CO, U.S.A., 12–17 November 2017, pp. 42–46.

[44] *IDEAS: Interoperable Design of Extreme-scale Application Software*, https://ideas-productivity.org.

[45] *xSDK: Extreme-scale Scientific Software Development Kit*, http://xsdk.info.

[46] The HYPRE Team, *HYPRE*, https://computing.llnl.gov/projects/hypre-scalable-linear-solvers-multigrid-methods.

[47] S. Hudson, J. Larson, S.M. Wild, D. Bindel and J.-L. Navarro, *libEnsemble users manual*, Tech. Rep., Revision 0.7.0, Argonne National Laboratory (2020) [https://buildmedia.readthedocs.org/media/pdf/libensemble/latest/libensemble.pdf].

[48] The AMReX Team, *AMReX*, https://amrex-codes.github.io/.

[49] The Copa-Cabana Team, *Copa-Cabana* https://github.com/ECP-copa/Cabana.

[50] The Portage Team, *Portage*, https://github.com/laristra/portage/releases [DOI: 10.5281/zenodo.4571000].

[51] *REANA Reproducible research data analysis platform* , http://reanahub.io.

[52] *CAMPA: Consortium for Advanced Modeling of Particle Accelerators*, http://campa.lbl.gov.

[53] *CERN Data Portal*, http://opendata.cern.ch.

[54] *FAIR Principles*, https://www.go-fair.org/fair-principles/.

[55] ATLAS Collaboration collaboration, *RECAST framework reinterpretation of an ATLAS Dark Matter Search constraining a model of a dark Higgs boson decaying to two b-quarks*, Tech. Rep., ATL-PHYS-PUB-2019-032, CERN, Geneva, Switzerland (2019).

[56] W.F. Godoy, N. Podhorszki, R. Wang, C. Atkins, G. Eisenhauer, J. Gu et al., *Adios 2: The adaptable input output system. a framework for high-performance data management*, *SoftwareX* **12** (2020) 100561.

[57] The HDF Group, *Hierarchical data format version 5*, http://www.hdfgroup.org/HDF5.

[58] A. Huebl, R. Widera, F. Schmitt, A. Matthes, N. Podhorszki, J.Y. Choi et al., *On the scalability of data reduction techniques in current and upcoming hpc systems from an application perspective*, in *High Performance Computing*, J.M. Kunkel, R. Yokota, M. Taufer and J. Shalf, eds., Springer International Publishing (2017), pp. 15–29.

[59] *Curated catalogue of projects supporting openPMD*, https://github.com/openPMD/openPMD-projects.

[60] *PICMI: Standard input format for Particle-In-Cell codes*, https://github.com/picmi-standard.

2021 JINST 16 T10002

[61] J.-L. Vay, A. Huebl, N. Cook, R.J. England, U. Niedermayer, P. Piot et al., *Modeling of Advanced Accelerator Concepts*, in *ICFA Beam Dynamics Newsletter#82 — Advanced Accelerator Modelling*.

[62] A. Friedman, D.P. Grote and I. Haber, *Three-dimensional particle simulation of heavy-ion fusion beams*, *Phys. Fluids B* **4** (1992) 2203.

[63] R. Hockney and J. Eastwood, *Computer simulation using particles*, 1988.

[64] J. Amundson, P. Spentzouris, J. Qiang and R. Ryne, *Synergia: An accelerator modeling tool with 3-d space charge*, *J. Comput. Phys.* **211** (2006) 229.

[65] J. Qiang, *Symplectic multiparticle tracking model for self-consistent space-charge simulation*, *Phys. Rev. Accel. Beams* **20** (2017) 014203.

[66] J. Qiang, *Symplectic particle-in-cell model for space-charge beam dynamics simulation*, *Phys. Rev. Accel. Beams* **21** (2018) 054201.

[67] L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, *J. Comput. Phys.* **73** (1987) 325.

[68] L. Ying, G. Biros, D. Zorin and H. Langston, *A new parallel kernel-independent fast multipole method*, in *Proceedings of the 2003 ACM/IEEE Conference on Supercomputing*, Phoenix, AZ, U.S.A., 15–21 November 2003, p. 14.

[69] I. Lashuk, A. Chandramowlishwaran, H. Langston, T.-A. Nguyen, R. Sampath, A. Shringarpure et al., *A massively parallel adaptive fast multipole method on heterogeneous architectures*, *Commun. ACM* **55** (2012) 101.

[70] H. Zhang and M. Berz, *The fast multipole method in the differential algebra framework*, *Nucl. Instrum. Meth. A* **645** (2011) 338.

[71] M.H. Langston, L. Greengard and D. Zorin, *A free-space adaptive FMM-based PDE solver in three dimensions*, *Commun. Appl. Math. Comput. Sci.* **6** (2011) 79.

[72] G. Biros, L. Ying and D. Zorin, *A fast solver for the stokes equations with distributed forces in complex geometries*, *J. Comput. Phys.* **193** (2004) 317.

[73] A. Klockner, A. Barnett, L. Greengard and M. O'Neil, *Quadrature by expansion: A new method for the evaluation of layer potentials*, *J. Comput. Phys.* **252** (2013) 332.

[74] M.J. Morse, A. Rahimian and D. Zorin, *A robust solver for elliptic pdes in 3D complex geometries*, *J. Comput. Phys.* **442** (2021) 110511.

[75] F. Vico, L. Greengard and M. Ferrando, *Fast convolution with free-space green's functions*, *J. Comput. Phys.* **323** (2016) 191.

[76] J. Zou, E. Kim and A.J. Cerfon, *FFT-based free space Poisson solvers: why Vico-Greengard-Ferrando should replace Hockney-Eastwood*, arXiv:2103.08531.

[77] C. Behrens, F.-J. Decker, Y. Ding, V. Dolgashev, J. Frisch, Z. Huang et al., *Few-femtosecond time-resolved measurements of X-ray free-electron lasers*, *Nature Commun.* **5** (2014) 3762.

[78] C. Joshi, E. Adli, W. An, C. Clayton, S. Corde, S. Gessner et al., *Plasma wakefield acceleration experiments at facet ii*, *Plasma Phys. Control. Fusion* **60** (2018) 034001.

[79] A. Malyzhenkov, Y.P. Arbelo, P. Craievich, P. Dijkstal, E. Ferrari, S. Reiche et al., *Single-and two-color attosecond hard x-ray free-electron laser pulses with nonlinear compression*, *Phys. Rev. Res.* **2** (2020) 042018.

[80] ILC collaboration, *The International Linear Collider. A Global Project*, arXiv:1901.09829.

2021 JINST 16 T10002

[81] K. Bane and P. Emma, *Litrack: a fast longitudinal phase space tracking code with graphical user interface*, in *Proceedings of the 2005 Particle Accelerator Conference*, Knoxville, TN, U.S.A., 16–20 May 2005, pp. 4266–4268.

[82] A. Scheinker and S. Gessner, *Adaptive method for electron bunch profile prediction*, *Phys. Rev. ST Accel. Beams* **18** (2015) 102801.

[83] A. Scheinker, F. Cropp, S. Paiagua and D. Filippetto, *Adaptive deep learning for time-varying systems with hidden parameters: Predicting changing input beam distributions of compact particle accelerators*, arXiv:2102.10510.

[84] F. Arute et al., *Quantum supremacy using a programmable superconducting proceSSOR*, *Nature* **574** (2019) 505 [arXiv:1910.11333].

[85] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo et al., *Quantum computational advantage using photons*, *Science* **370** (2020) 1460.

[86] *Ibm quantum experience*, https://quantum-computing.ibm.com/, 2020.

[87] *Leap*^{2}, https://www.dwavesys.com/, 2020.

[88] *Amazon quantum solutions lab*, https://aws.amazon.com/quantum-solutions-lab/, 2020.

[89] *Azure quantum*, https://azure.microsoft.com/en-us/services/quantum/, 2020.

[90] A.W. Harrow, A. Hassidim and S. Lloyd, *Quantum algorithm for linear systems of equations*, *Phys. Rev. Lett.* **103** (2009) 150502.

[91] B.D. Clader, B.C. Jacobs and C.R. Sprouse, *Preconditioned quantum linear system algorithm*, *Phys. Rev. Lett.* **110** (2013) 250504.

[92] A.M. Childs, R. Kothari and R.D. Somma, *Quantum algorithm for systems of linear equations with exponentially improved dependence on precision*, *SIAM J. Comput.* **46** (2017) 1920.

[93] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio and P.J. Coles, *Variational quantum linear solver: A hybrid algorithm for linear systems*, arXiv:1909.05820.

[94] Y. Lee, J. Joo and S. Lee, *Hybrid quantum linear equation algorithm and its experimental test on ibm quantum experience*, *Scientific reports* **9** (2019) 4778.

[95] S.K. Leyton and T.J. Osborne, *A quantum algorithm to solve nonlinear differential equations*, arXiv:0812.4423.

[96] D.W. Berry, *High-order quantum algorithm for solving linear differential equations*, *J. Phys. A* **47** (2014) 105301.

[97] J.M. Arrazola, T. Kalajdzievski, C. Weedbrook and S. Lloyd, *Quantum algorithm for nonhomogeneous linear partial differential equations*, *Phys. Rev. A* **100** (2019) 032306.

[98] A.M. Childs, J.-P. Liu and A. Ostrander, *High-precision quantum algorithms for partial differential equations*, arXiv:2002.07868.

[99] T. Xin, S. Wei, J. Cui, J. Xiao, I. Arrazola, L. Lamata et al., *Quantum algorithm for solving linear differential equations: Theory and experiment*, *Physi. Rev. A* **101** (2020) 032307.

[100] Y. Cao, A. Papageorgiou, I. Petras, J. Traub and S. Kais, *Quantum algorithm and circuit design solving the poisson equation*, *New J. Phys.* **15** (2013) 013021.

[101] S. Wang, Z. Wang, W. Li, L. Fan, Z. Wei and Y. Gu, *Quantum fast poisson solver: the algorithm and modular circuit design*, arXiv:1910.09756.

2021 JINST 16 T10002

[102] A. Engel, G. Smith and S.E. Parker, *Quantum algorithm for the vlasov equation*, *Phys. Rev. A* **100** (2019) 062315.

[103] *Q# development kit*, https://www.microsoft.com/en-us/quantum/development-kit, 2020.

[104] I.Y. Dodin and E.A. Startsev, *On applications of quantum computing to plasma simulations*, arXiv:2005.14369.

[105] J.-L. Vay, D. Bruhwiler, D. Sagan, A. Huebl, R. Lehe, C.-K. Ng et al., *Center(s) for Accelerator and Beam Physics Modeling*, submitted to *Snowmass21* [https://snowmass21.org/docs/files/summaries/CompF/SNOWMASS21-CompF2_CompF0-AF1_AF0_Vay-069.pdf].

[106] *MIT schwarzman college of computing*, https://computing.mit.edu/.

[107] *QuICS*, https://quics.umd.edu/.

[108] C. Fisher, *IBM | quantum computing*, https://www.ibm.com/quantum-computing/.

[109] *Exascale computing project*, https://www.exascaleproject.org/.

[110] *Scientific discovery through advanced computing*, https://www.scidac.gov/.

[111] R. Lehe, A. Hanuka, A. Edelen, X. Huang, C. Mayes, N. Cook et al., *Machine learning and surrogates models for simulation-based optimization of accelerator design*, submitted to *Snowmass21* [https://snowmass21.org/docs/files/summaries/CompF/SNOWMASS21-CompF2_CompF3-AF1_AF6_Lehe-075.pdf].

[112] A. Huebl, J.-L. Vay, R. Lehe, C. Mayes, Y.-D. Tsai, A. Friedman et al., *Aspiration for Open Science in Accelerator & Beam Physics Modeling*, submitted to *Snowmass21* [https://snowmass21.org/docs/files/summaries/CompF/SNOWMASS21-CompF2_CompF7-AF1_AF0_Huebl-081.pdf].

[113] M. Riordan, L. Hoddeson and A.W. Kolb, *Tunnel Visions: The Rise and Fall of the Superconducting Super Collider*, University of Chicago Press (2015).

[114] D. Ritson, *SLAC Memorandum: D. Ritson to R. Schwitters, March 8, 1990*, 1990.

[115] D. Ritson, *SSCTRK: A particle tracking code for the SSC*, https://inis.iaea.org/search/search.aspx?orig_q=RN:21094597, 1990.

2021 JINST 16 T10002