HPSIM - ADVANCED ONLINE MODELING FOR PROTON LINACS*

L. J. Rybarcyk[†], Los Alamos National Laboratory, Los Alamos, NM, 87545, USA

Abstract

High-power proton linacs seek to operate with low and stable losses. This aspect is carefully evaluated with multi-particle beam dynamics codes during the design stage. However, it is just as important to evaluate the performance of the actual operating linac, which is typically more tedious and complicated when using these same design codes. To improve this situation, we have developed a high-performance, multi-particle online modeling tool, HPSim, with the goal of providing near real-time simulation results for our 800-MeV proton linac at Los Alamos. This presentation will cover the motivation, code features, benefits and applications.

INTRODUCTION

High-power linacs, like the 800-MeV proton linac at the Los Alamos Neutron Science Center are designed and operated to achieve low and stable beam loss. During the accelerator design phase, envelope and multi-particle codes, such as TRACE 2-D & 3-D [1], PARMILA [2], TRACK [3], TRACEWIN [4], and others are used to create the linac layouts and evaluate them for losses. While the envelope codes are being used as online models for the operating machines, the multi-particle codes, until very recently [5], have not. This has been due in part to a combination of performance and computer resource issues associated with using these codes in a control room setting.

The typical multi-particle code, like those mentioned above and others, e.g. IMPACT [6], etc. used to successfully simulate proton beams in linacs, offer benefits and advantages not available from an envelope code. However, using these to simulate beam in an operating linac can be tedious and complicated. This is because all of the relevant beam and machine parameters and settings for a *single snapshot* of the operation must be transferred to the input format used by the code. Only then can a simulation be performed from which results can be compared to measurements under those same conditions.

Routine use of accurate online multi-particle simulations to setup, track and study performance of an operating linac would be beneficial. This is especially true for new and existing high-power linacs where detailed knowledge of the beam distribution along the linac can provide new insight into issues that produce beam spill. HPSim was developed for this purpose.

MOTIVATION

The LANSCE accelerator utilizes both proton and H-beams and delivers them to several user facilities for basic

and applied research. The pulsed accelerator operates with a typical maximum beam duty factor of 120 Hz x 625 µs for each species. Initially, the beams are accelerated to 750 keV using Cockcroft-Walton (C-W) style injectors. The high-current DC beams are subsequently bunched prior to injection into our 201.25-MHz 100-MeV drift tube linac (DTL) using two 201.25 MHz single-gap cavities. This scheme enables us to routinely achieve ~80% longitudinal capture of the standard proton and H-beams. However, this approach also results in significant longitudinal tails on each beam. These tails are clearly seen in the simulated longitudinal phase space of the proton beam at the entrance and exit of the DTL, as shown in Figure 1. To operate the DTL in a manner that produces the lowest beam spill along the entire side-coupled cavity linac (CCL) and further downstream, operations staff have found a new set of phase and amplitude set points for the DTL that are significantly different from design. The phase and amplitude set points of the DTL were empirically determined over years of high-power (800 kW) beam operation and are similar even under today's ~100 kW operation.
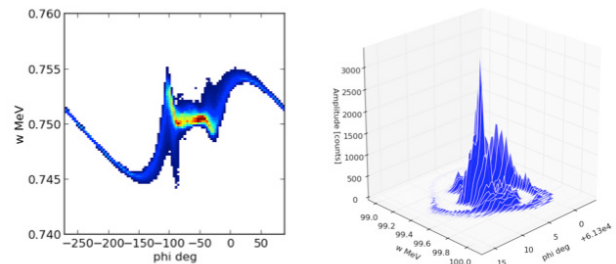

Figure 1: Simulated longitudinal phase space distributions for the LANSCE proton beam at the entrance (left) and exit (right) of the DTL.

The annual tune-up process for the LANSCE linac involves the typical physics-based approach of using envelope and previously established single- and multi-particle model results to determine good starting points for focusing magnets and accelerator modules along the linac. Once this phase is complete, the process becomes much more empirical and the operators resort to tweaking many of the newly established RF set points in order to further reduce beam loss along the accelerator. This is the same approach used at many other accelerator facilities as mentioned in a previous HB workshop [7].

Our first-hand experience, as well as those at other similar facilities, motivated us to develop HPSim. The existing envelope and single-particle models cannot provide the needed insight into the actual beam distribution expected from a particular set of accelerator operating set points. However, to be effective in an accelerator control room setting, the code needs to be fast and accurate with the ability to access the control system to minimize the

overhead of setting up the simulation. HPSim meets these requirements.

HPSIM

The new **H**igh-**P**erformance **Sim**ulator, HPSim, was developed as a new stage in online modeling and simulation of proton linacs. It is designed to provide more accurate beam dynamics simulations than the envelope model, faster performance than typical multi-particle simulations that run on desktop workstations, and utilize a connection to the accelerator's EPICS [8] control system.

Physics Model

The physics model used in the code is derived from the PARMILA (Phase And Radial Motion in Ion Linac Accelerators) linac design and simulation code. PARMILA is a well-established tool that has been used in the community for decades and was used to design and/or simulate the LANSCE, SNS and other linacs. It is multiparticle in nature and presently uses a 2D r-z particle-in-cell (PIC) algorithm to produce the nonlinear space-charge force on the beam macroparticles in the bunch. It is a "z code" that employs transfer maps in the beam transport regions. The particle transformation across the rf gap is represented as a "drift + kick-at-the-midplane + drift" and uses the transit-time-factor formulation [9]. It has been compared and benchmarked against other codes and shown to be in good agreement [10].

HPSim, however, has some notable differences. First, it was written to only simulate beam in ion linacs and does not produce a design. The layout must be generated by another code, e.g. PARMILA for the linac geometry and design field information and SUPERFISH [11] for the cell-data and transit-time-factors for each rf gap. Second, to better and more easily simulate real world, non-ideal linac operation, it keeps track of the particles' absolute phase, which is essential when tracking particles in scenarios where an intermediate accelerating section of the linac is operationally turned-off. An example would be when a superconducting section of the linac is taken offline and the downstream module set points adjusted to restore acceleration for maximum available energy, which is done at SNS. Another example, from LANSCE, when delivering beam at an intermediate CCL output energy, one or more downstream modules can be repurposed to bunch the beam and reduce the phase spread of the final beam for neutron time-of-flight experiments. Third, transit-time-factors can be functions of β to be more accurate for particles that are off-energy in an rf-gap. Fourth, the space-charge routine focuses on particles that are in the bucket, to improve the simulation where sizable off-energy beam is present. Fifth, the space-charge algorithm employs a scaling feature that allows the previously calculated field table to be reused when the beam is not too different in energy or size. This improves the performance without a noticeable degradation in accuracy in situations where these quantities are changing slowly between successive space-charge kicks, e.g. in our CCL.

Presently, HPSim supports several standard ion linac elements. Quadrupole focusing magnets use the hard-edge model with a single space-charge kick in the middle. Steering magnets are impulse correctors. Empty spaces are simple drifts with an overall user defined maximum interval between space-charge kicks. A buncher cavity is a single RF gap. A DTL or CCL RF cell is a single gap with a space-charge kick in the middle. The DTL structure is an Alvarez style with quad magnets in the drift tubes. The CCL structure comprises multicell tanks with quad magnets located in between them. It also supports circular and rectangular transverse apertures and a space-charge compensation factor that is used to reduce the effective current in the beam where neutralization is present.

GPU Accelerated

Our goal was to have HPSim be fast enough to perform a simulation in seconds to a small fraction of a minute and be available 24/7 in the LANSCE control room. The cost was also a consideration. This led us to choose state-of-the-art graphics processing unit (GPU) hardware to achieve the desired performance and availability at minimal cost. Although GPU technology was developed for the gaming industry, its power is now being harnessed for scientific computations, e.g. the ORNL Titan, one of the world's fastest supercomputers. In addition, to take full advantage of the GPU power, HPSim was developed from scratch, using a modern programming approach.

The GPU technology is also very cost effective. A single NVIDIA Tesla K20c has 2496 CUDA cores, 5 GB RAM with peak double/single precision performance of 1.17/3.52 Tflops and a street price of ~$3K US.

HPSim Code Design and Performance

Code Design The code design was developed with the philosophy that it should be fast enough for a control room setting and yet easy to use with the same control interface as the actual machine. This led us to create a code with several distinctive aspects. First, the number crunching simulation kernels are written in NVIDIA CUDA C and C++, which take full advantage of the GPU's performance [12]. Details of the implementation are given elsewhere [13]. Second, the user interface is available as Python/C APIs for ease-of-use, which hides all the complex code from the user's experience and enables one to communicate with the code through powerful, yet easy to use, Python scripts. This also allows the user to take advantage of the extensive collection of numerical and visualization libraries available for Python. The hierarchical structure of the code is shown in Figure 2. Third, since the linac design is fixed and the model is meant to be controlled through the same variables as that of the real machine, the representation of the linac and control parameters all reside in a database.

The code is divided into several major components, which reside on either the CPU(host) or GPU(device), as shown in Figure 3. A data server is used to acquire real machine parameter data via EPICS and update a flat-file

like serverless SQLite database. The database contains the linac design, as well conversion rules and calibration quantities required to transform each control parameter engineering value into a calibrated physics model quantity. The database provides the beamline model that is written to special CPU-side memory, aka "pinned", that can be efficiently accessed by the GPU. To maximize performance, the beam distribution is created and resides on the device, as do all the simulation kernels that represent the element transformations and space-charge algorithms. In the online mode, output from the simulation is then produced in graphical form using OpenGL as either 2D quantitative plots or a 3D particle distribution along the linac. In the offline mode, Python modules are available to analyze and plot the beam results.
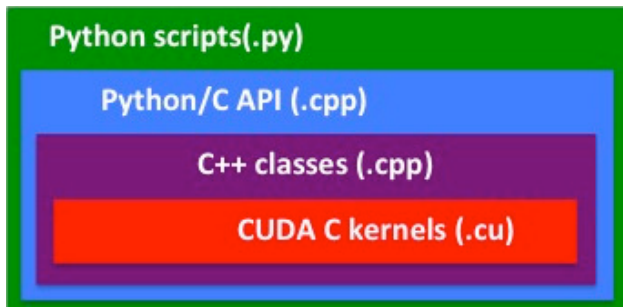

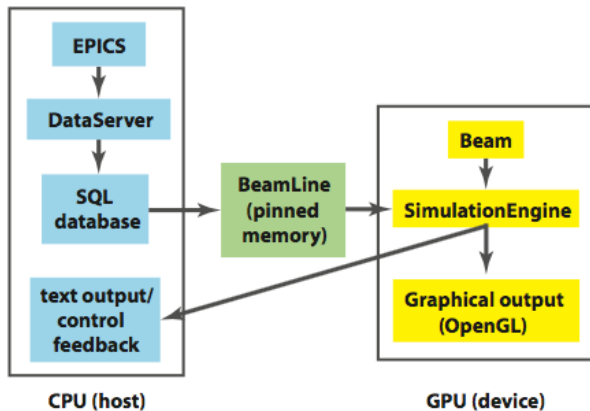Figure 2: Hierarchical structure of HPSim code.

Figure 3: High-level code structure and data flow indicated by arrows.

Code Performance The performance of HPSim exemplifies the speedup possible when employing GPU technology. A comparison of code performance was made on same-era hardware, i.e. an Intel Xeon E5520 2.27 GHz CPU and an NVIDIA GTX 580 GPU (Fermi architecture). We observed a maximum speedup of ~160 for beam transported without space-charge and ~45 for the space-charge routine alone.

The performance in simulating the LANSCE linac is very impressive and is based upon a single NVIDIA K20c GPU in a Linux workstation. Starting with a DC waterbag beam distribution of 64K macro-particles located in the 750-keV low-energy-beam-transport (LEBT) and transporting them up to the end of the 800-MeV CCL takes ~5.5 seconds. This problem space is ~800 m in length and

contains over 5100 RF gaps, 400 quads and 6000 space-charge kicks.

Model Calibration

The model must be calibrated for a meaningful and accurate prediction of beam in the linac. All necessary calibration information is held in the database. This enables control set points to be transformed on the fly into physics quantities in the model. Typical devices that require calibration are focusing and steering magnets and rf bunchers and accelerating modules (combination of accelerator cavities/tanks driven by a single amplifier). At LANSCE, EPICS provides engineering quantities associated with the devices that power/control the accelerator and beamline components. These EPICS values, e.g. power supply current (amps) that drives a quadrupole magnet, LLRF cavity phase (degrees) and amplitude (a.u.) set points are then converted to physics model quantities, e.g. quadrupole gradient (T/m), and accelerating cavity phase (deg) and amplitude E_0 (MV/m), respectively.

The magnet calibrations are typically done offline in a mapping facility where integral field measurements versus current are made. These calibration curves are stored in the database.

The buncher cavity field amplitude calibration can also be done offline. Using a combination of EM cavity field modeling results to determine the theoretical E_0TL, Q_u and P_{cav} and measurements of the cavity Q along with amplifier output power vs. amplitude set point will produce the effective gap voltage vs. amplitude set point. A beam-based measurement can then be used to determine the phase offset that calibrates the cavity phase with respect to the other rf devices in the linac.

The accelerator modules must be calibrated with beam-based measurements, from which cavity field phase offsets and amplitude scale factors are determined. The measurements that are used at LANSCE are phase scans. The location and width of the phase scan distributions are related to the cavity phase and amplitude, respectively. For the DTL and LEBT bunchers the phase scan procedure measures the beam current of particles with energies above a specified threshold as a function of the phase setting of the tank or buncher being scanned. The measured and simulated distribution produced by scanning LANSCE DTL module 1 with only the last (main) LEBT buncher on is shown in Figure 4. It should be noted that reproducing these kinds of measurements is not possible with a single particle or envelope code. For the CCL, the phase scan is based upon beam phase measurements versus module phase. Fitting the data also provides a field phase offset and amplitude scale factor for each module. The measured and simulated phase scan results for the first module of the CCL are shown in Figure 5.

Online & Offline Modes

HPSim functions in two different modes. In the online mode, a stand-alone executable that uses a predefined input beam runs in a continuous loop. As the actual machine parameters are adjusted, the data server copies the

updated values into the SQLite db. At the start of each pass, the updated model values are placed into pinned memory. This allows the simulation to automatically track the changes in the machine parameters and reflect them in the simulation results. The graphical 2D display is updated after each pass and includes quantities like output beam phase space distributions and profiles and rms widths, emittances and loss profiles along the linac. The EPICS data can also be obtained from a soft IOC, which is a stand-alone copy of the machine parameters and not actually tied to the operating machine hardware.
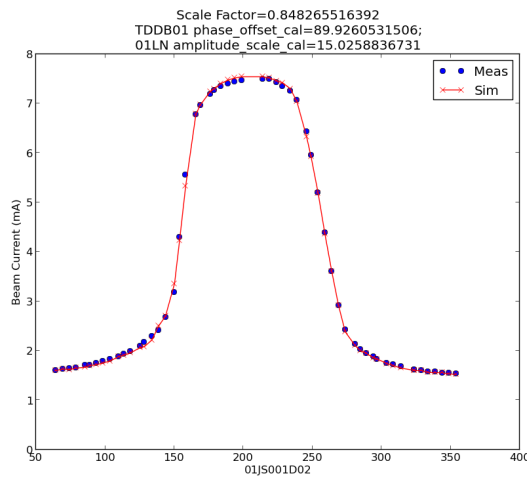

Figure 4: Measured and simulated phase scan of LANSCE DTL tank 1.
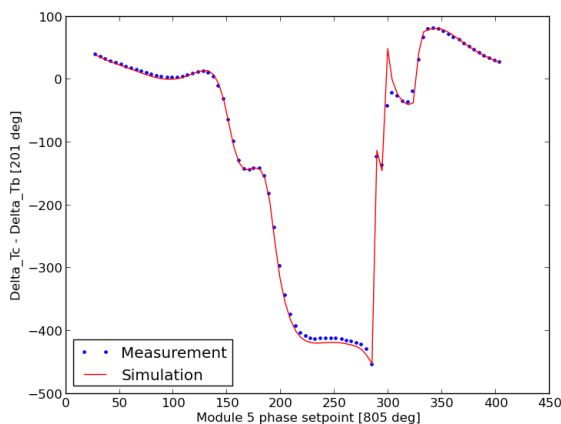

Figure 5: Measured and simulated phase scan of the first LANSCE CCL module.

HPSim can also be operated in an offline mode. This is valuable for the calibration activities that require additional analysis, e.g. fitting. In this mode, the user interacts with the code via Python scripts. HPSim is simply meant to simulate the beam from point A to point B in the linac under specified operating condition. All post-processing is done on the output beam with user-created Python scripts. This approach also gives the user the freedom to explore new ideas in tuneup and operation without the need to modify the HPSim lower-level code directly.

Applications

Machine Tuning and Monitoring   The original motivation for creating HPSim was to provide the operations staff with a fast and more realistic beam simulation tool that would help with the tune-up and operation of the LANSCE linac. In this regard, it can function as a virtual beam diagnostic, providing information where none is available. This is especially helpful in high-power linacs, where measurements may not be possible due to the interceptive or deleterious nature of many beam diagnostics. Figure 6 is an example of the 2D online mode showing beam along and out of the LANSCE CCL. During typical operations, the machine parameters are adjusted to minimize beam losses. HPSim could provide additional insight into the source of the beam spill.

Machine Set Point Optimization   In an operating linac, set point optimization with regard to a desired state is usually done leading up to and following the transition to high power. The process involves physics model based setup followed by empirically searching a high-dimensional parameter space for optimal values. The offline mode of HPSim works well for this application. Using HPSim in offline mode with the multi-objective particle swarm optimization (MOPSO) and multi-objective genetic algorithm (MOGA), we were able to determine a set of buncher and DTL RF phase and amplitude set points that produce less beam loss and smaller beam emittances in simulation than operating at design set points [14]. Although this exact solution has not yet been experimentally verified, the optimized DTL RF cavity field amplitudes are in fact very close to the typical operating levels.

Virtual Accelerator and Test Bed   The simulator can also serve as a virtual accelerator providing a test bed for evaluating new approaches and techniques for the linac where realistic beam information or control of the machine set points is desired. An example that was tested with the simulator was a model-independent automatic accelerator tuning method [15]. This real-time control method enables simultaneous tuning of several parameters, even where coupling is present, to achieve good beam quality. Having HPSim available allowed the initial successful demonstration of this technique to proceed without the risk to the machine or commitment of accelerator resources, i.e. development time and expense.

STATUS AND FUTURE PLANS

To date, HPSim has been used for offline studies and model calibrations of the LANSCE linac, as well as in a few isolated troubleshooting efforts in the accelerator control room. During its development, the code has been thoroughly tested against PARMILA and also compared with various experimental measurements. It will be further incorporated into the LANSCE startup later this year. In addition, we are planning to make HPSim available as
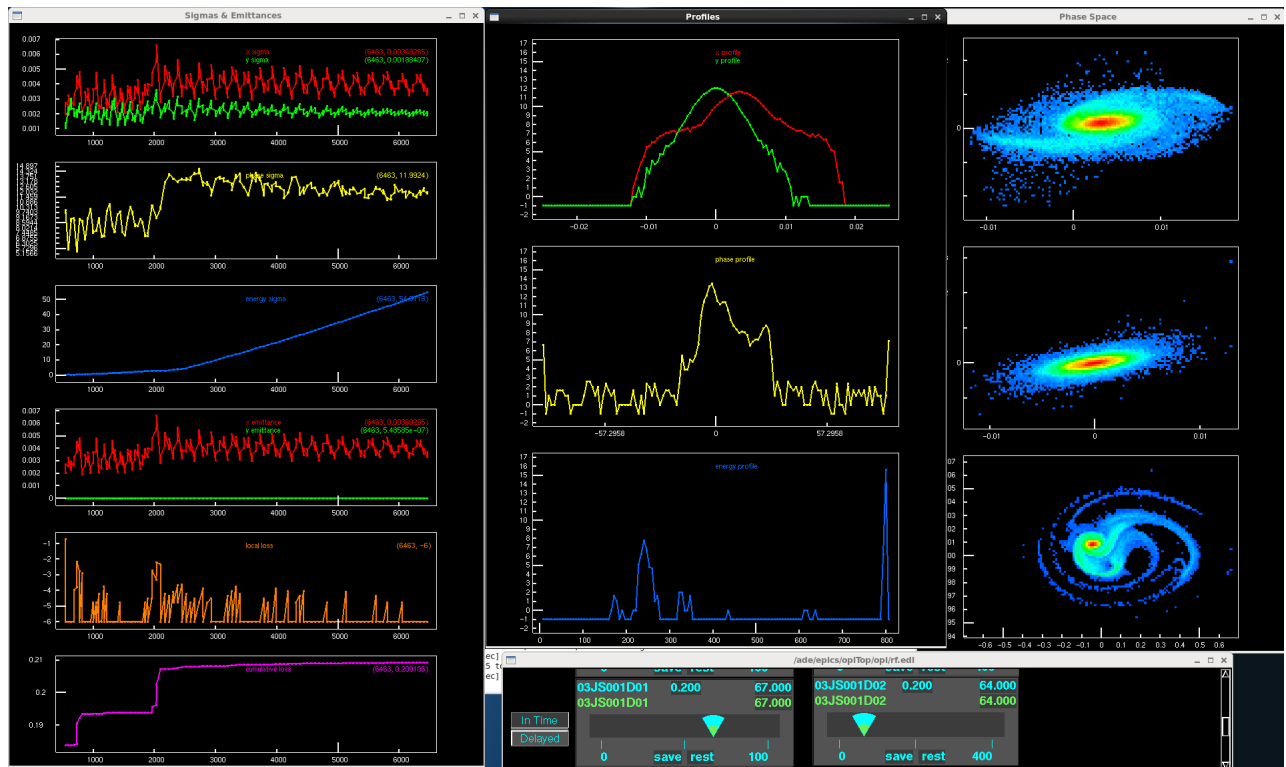
Figure 6: Screen shot of HPSim 2D online graphical output of H- beam transported along the LANSCE CCL. The display includes beam rms widths, emittances, and losses (left), output beam profiles (center) and output beam phase-space distributions (right). The EPICS sliders for DTL Tank 3 amplitude and phase controls are shown at the bottom.

an open source code for the benefit of the wider accelerator community.

ACKNOWLEDGEMENT

REFERENCES

[1] K. R. Crandall, D. P. Rusthoi, "TRACE3D manual," 3rd Ed., LANL publication, LA-UR-97-886, (1997).

[2] H. Takeda and J.H. Billen, "PARMILA," LANL publication, LA-UR-98-4478, (Revised 2005).

[3] B. Mustapha et al., "TRACK: The New Beam Dynamics Code," PAC 2005, Knoxville, Tennessee, May, 2005, TPAT028, p. 2053 (2005); http://www.JACoW.org

[4] TRACEWIN web site http://irfu.cea.fr/Sacm/logiciels/index.php

[5] C. E. Peters et al., "Integration of the TRACK Beam Dynamics Model to Decrease Linac Tuning Times," ICALEPCS2015, Melbourne, Australia, Oct. 2015, MOPGF092, p. 291, (2015).

[6] J. Qiang et al., "An Object Oriented Parallel Particle-In-Cell Code for Beam Dynamics Simulation in Linear Accelerator," Journal of Computational Physics, 163, pp. 434-445, (2000).

[7] J. Galambos et al., "Summary of the Commissioning, Operations and Performance Working Group of the HB-2010 Workshop," HB2010, Morschach, Switzerland, Sept. 2010, FROP04, p. 703, (2010).

[8] EPICS, http://www.aps.anl.gov/epics

[9] Internal Report: "RF-gap Transformation in PARMILA," AOT-ABS: 12-018 (TN), (2012).

[10] S. Nath et al., "Comparison of Linac Simulation Codes", PAC'01, Chicago, 2001, TOPB003, p. 214. (2001).

[11] J. H. Billen and L. M. Young, "POISSON/SUPERFISH Reference Manual," LANL publication, LA-UR-96-1834, 1996, (Revised 2004).

[12] NVIDIA Corporation, CUDA C Programming Guide, (2014).

[13] X. Pang and L. Rybarcyk, "GPU Accelerated Online Beam Dynamics Simulator for Linear Particle Accelerators," Computer Physics Communications, 185, pp. 744-753, (2014).

[14] X. Pang and L. Rybarcyk, "Multi-Objective Particle Swarm and Genetic Algorithm for Optimization of the LANSCE Linac Operation," Nucl. Instr. Meth. A, 741, 124, (2014).

[15] A. Scheinker et al., "Model Independent Particle Accelerator Tuning," Phys. Rev. ST Accel Beams, 16, 102803, (2013).