SIMULATION OF COHERENT SYNCHROTRON RADIATION ON PARALLEL HYBRID GPU/CPU PLATFORM [*]

B. Terzić[†], K. Arumugam, D. Duffin, A. Godunov, T. Islam, D. Ranjan, S. Sangam, M. Zubair
Old Dominion University, Norfolk, VA 23529, USA

Abstract

Coherent synchrotron radiation (CSR) is an effect of self-interaction of an electron bunch as it traverses a curved path. It can cause a significant emittance degradation, as well as fragmentation and microbunching. Numerical simulations of the 2D/3D CSR effects have been extremely challenging due to computational bottlenecks associated with calculating retarded potentials via integrating over the history of the bunch. Our new high-performance 2D, particle-in-cell code uses massively parallel multicore GPU/GPU platforms to alleviate computational bottlenecks. The code formulates the CSR problem from first principles by using the retarded scalar and vector potentials to compute the self-interaction fields. The speedup due to the parallel implementation on GPU/CPU platforms exceeds three orders of magnitude, thereby bringing a previously intractable problem within reach. The accuracy of the code is verified against analytic 1D solutions (rigid bunch) and semi-analytic 2D solutions for the chirped bunch. Here we use the new code in conjunction with a genetic algorithm and Bayesian optimization to optimize the design of a fiducial chicane.

COHERENT SYNCHROTRON RADIATION

Electromagnetic radiation is the result of the acceleration of charged particles. When a large group of these charged particles emit this radiation, the power emitted can be classified into two groups: coherent and incoherent [1]. The coherent portion encompasses the lower frequency spectrum with little variation in wavelength and phase difference. The constructive interference that results exaggerates the power allocated to the coherent portion. This coherent portion is called Coherent Synchrotron Radiation (CSR). CSR is proportional to the square of the number of charged particles (N^2) and inversely to the longitudinal size of the beam ($1/\Delta s$). CSR is a mechanism whereby the beam self-interacts. Namely, the radiation that the beam emitted at a previous time can interact with the beam at a later time. This simultaneous interaction between beam and radiation leads non-linear, systemic effects that cannot be ignored. Among these effects that are of interest to study are emittance degradation and microbunching.

CODE

All classical electromagnetic effects are governed by Maxwell's equations and the Lorentz force equation, but analytical solutions to beam behavior consisting of many thousands of charged particles are unfeasible without drastic simplifying assumptions. Numerical solutions must be implemented to quantify beam behavior. However a numerical approach presents its own limitations: (i) the position and strength of self-interacting radiation depends on the beam at previous times. Therefore an algorithm attempting to calculate self interaction needs to allocate the entire beam history to memory (high memory usage); (ii) the Lorentz force equation leads to the subtraction of very large numbers by virtue of the electric and magnetic contributions to the force (loss of significance); (iii) the large number of particles involved result in an exponential rise in function evaluations, especially for naive implementations (poor scaling). Our innovative 2D particle tracking code has been implemented to mitigate all of these issues [2]. The code assumes a planar beam and tracks the phase space distribution (position and momenta of each particle). The self-interaction of the beam is quantified in a time-dependent function called the retarded potential. The simulation uses a particle-in-cell (PiC) method.

A rough outline of the code is as follows [2]: (1) Sample the initial distribution function; (2) Bin particles using a deposition scheme; (3) Compute retarded potentials in the grid frame using both the previous beam history and external fields; (4) Compute spatial and temporal derivatives of the potentials; (5) Compute the self-forces on each particle using the derivative values; (6) Advance particles in time by a push from calculated self-forces; (7) Loop back to Step 2 until the end of the simulation is reached. The bulk of the execution time is taken up by Step 3–computation of retarded potentials by integrating over the history of the beam. The high precision and speed of the new CSR code required a design and implementation of a new adaptive multi-dimensional quadratures on a hybrid GPU/CPU platform [3, 4].

ACCELERATOR DESIGN

Understanding how the geometry and setup of the accelerator affect the beam a priori is of great interest. We combine two optimization schemes–genetic algorithm (GA) and Bayesian optimization–with our new CSR code in order to compute geometries that produce an optimal beam quality.

Here we demonstrate a proof of concept for this idea. A GA and a Bayesian optimization were run to determine which geometry of a simple three-bend bunch compressor (Fig. 1) results in an optimal transverse emittance and longitudinal spread (multi-objective problem). In transitioning form single- to multi-objective problems, a notion of opti-

[*] Work supported by the National Science Foundation grant 1535641.
[†] bterzic@odu.edu

mality changes from the single optimal solution (a minimum or a maximum) to a set of *non-dominated* solutions–so called *Pareto-optimal front*. A solution A is said to dominate solution B if one or more of A's objectives is better than the corresponding values for B, and any remaining objectives are equal to B's. The dimensionality of the problem is constrained by the fact that the entrance and exit direction are the same, thereby reducing the problem to two independent inputs (2D problem). The beam and simulation parameters are given in Tables 1 and 2.

Optimization of other properties of the beam (single- or multi-objective) with a different geometry and restrictions can be easily obtained by a simple extension of the paradigm presented here.

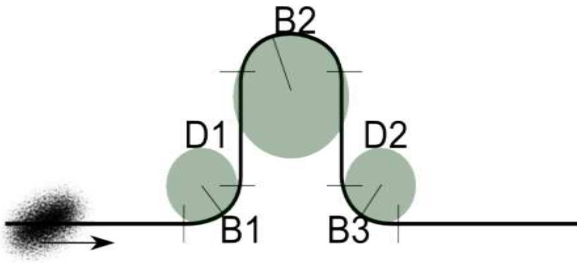

Figure 1: Layout of the fiducial three-bend chicane. The symmetry imposes that the B1 = B3 = −B2 and D1 = D2, thereby making it a 2D problem. The parameters are varied in the range 0 ≤ B1 ≤ 1 and −5 ≤ D1 ≤ −0.1.

Table 1: Beam Properties

Quantity	**Value**
Normalized horizontal emittance	10^{-9} m rad
Initial energy spread	$-10.0\,\mathrm{m}^{-1}$
Uncorrelated energy spread	10^{-7}

Table 2: Numerical Simulations Properties

Quantity	**Value**
Grid resolution	64×64
Number of simulation particles	200, 000
Number of steps along the lattice	106

Each individual simulation takes about 240 seconds to execute on a single GPU (Tesla K40). The execution time on multiple GPUs scales roughly as the inverse of the number of GPUs [5, 6].

Genetic Algorithm

The genetic algorithm (GA) is a tool for multi-dimensional, non-linear optimization that takes inspiration from the principles of biological evolution [7, and references therein]. GAs are well-suited for functions with many interrelated variables and specifications. In addition, GAs are highly parallelizable which suits them for computation clusters.

The general method for a GA works by evaluating the function for a given set of inputs (individuals). These evaluations are then analyzed for *fitness*. The individuals with the most fit outputs are replicated, combined (recombination), and perturbed (mutation) for the next set of individuals (generation). As the algorithm iterates over several generations, the individuals converge to the input that optimizes the function.

GAs also excel in solving multi-objective problems with more than one output (optimizing multiple quantities simultaneously). In order to achieve this, the GA determines if a individual is dominated. A dominated individual is one where another individual's objectives are all more fit. A multi-objective GA does not solve for a particular input but rather a collection of inputs called the Pareto-optimal front: a non-dominated set of individuals.

We use PISA platform developed at ETH Zürich [8].

Bayesian Optimization

Bayesian optimization is a sequential design strategy for global optimization of black-box functions, which does not require derivatives [9]. It is an example of machine learning process in which information about previous observations of the function to be optimized is used to determine the next optimal point. We use GPyOpt python routine [10] to implement the Bayesian optimization.

Bayesian optimization is by design single-objective. In order to use it to compute multi-objective Pareto-optimal fronts, the objective functions, say f_1 and f_2, as we have here, are combined into a single objective function f as $f = c_1 f_1 + c_2 f_2$, with $c_1 + c_2 = 1$, as explained in [11, Appendix A]. In order to produce the front shown in Fig. 2 below, constants are varied in the range $c_1 \in [0, 1]$ in increments of 0.1, yielding a total of 11 directions tangent to the Pareto-optimal front [11]. All non-dominated points compose the Pareto-optimal front from the Bayesian optimization.

RESULTS

Our intent here is to combine our new CSR code with the two powerful tools of non-linear, multi-dimensional optimization–GA Bayesian optimization–and compare their performance on equal footing. The results reported in this section are based on the two optimization tools executing a comparable number of function evaluations. In this case, each function evaluation is a single simulation executed on a GPU, for parameters reported in Tables 1 and 2.

Figure 3 shows Pareto-optimal fronts obtained by roughly the same number of function evaluations with the two optimizations techniques–GA and Bayesian. The two fronts are close to each other. Their quality is estimated by their location in relation to the two objective functions shown–the better front is one that is farther left and down. The two fronts agree extremely well in the low transverse emittance (high longitudinal spread) region; in the intermediate region the GA is slightly better, while in the high transverse emittance (low longitudinal spread) region, the Bayesian optimization performs better.

When plotted in search space to show where the solutions of the two Pareto-optimal fronts lie, it is evident that the two fronts represent vastly different sets of solutions. We will investigate why this happens, and under which conditions the results from the two optimization techniques converge.

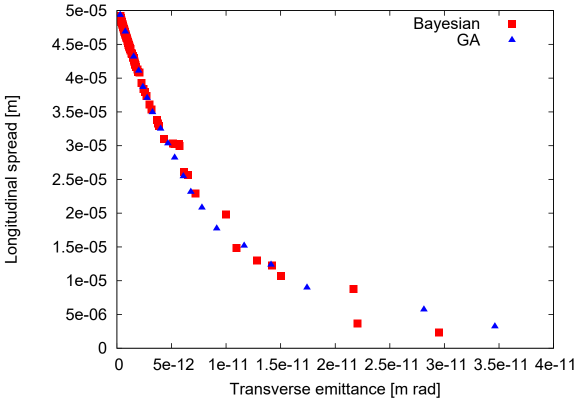

Figure 2: Pareto-optimal front from a GA with 37 generations and 20 individuals (blue triangles), and Bayesian optimization with 770 evaluations (red squares). Each simulation uses nearly the same number of function evaluations.

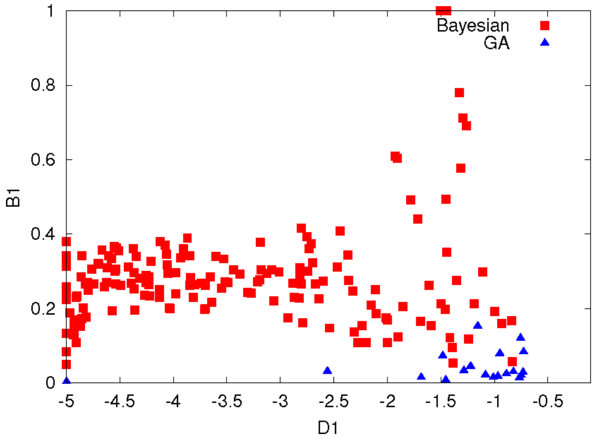

Figure 3: Pareto-optimal front from Fig. 2 shown in search space. Even though the two methods produce similar-looking Pareto fronts in objective function space, in search space these solutions look quite different.

SUMMARY

We presented the proof of concept in which we demonstrated that combining powerful optimization tools–GA and Bayesian optimization–with our new CSR code [2] produces a chicane geometry with optimal properties of the beam at the exit. In this work, the relevant properties are beam's longitudinal spread and its transverse emittance, but the same powerful mechanism can easily be extended to other beam properties.

REFERENCES

[1] J. Schwinger, "On radiation by electrons in a betatron," LBNL, US, LBNL-39088, Jan 1998. Edited and transcribed by M. Furman.

[2] B. Terzić, A. Godunov, K. Arumugam, D. Ranjan, and M. Zubair, "High-performance simulations of coherent synchrotron radiation on multicore GPU and CPU platforms," in *Proc. of International Particle Accelerator Conf. (IPAC'15)*, Richmond, USA, May 2015.

[3] K. Arumugam, A. Godunov, D. Ranjan, B. Terzić, and M. Zubair, "An efficient deterministic algorithm for adaptive multidimensional integration on GPUs," in *42nd International Conference on Parallel Processing (ICPP)*, Lyon, France, Oct. 2013.

[4] K. Arumugam, A. Godunov, D. Ranjan, B. Terzić, and M. Zubair, "A memory efficient algorithm for adaptive multidimensional integration with multiple GPUs," in *Proc. of High Performance Computing (HiPC)*, Bangalore, Dec. 2013.

[5] K. Arumugam, A. Godunov, D. Ranjan, B. Terzić, and M. Zubair, "High fidelity simulation of collective effects in electron beams using an innovative parallel method," in *Summer Computer Simulation Conference*, Chicago, July 2015.

[6] K. Arumugam, A. Godunov, D. Ranjan, B. Terzić, and M. Zubair, "A memory efficient parallel simulation of electron beam dynamics using GPUs," in *23rd Annual IEEE International Conference on High Performance Computing*, Hyderabad, India, Dec. 2016.

[7] A. Hofler, B. Terzić, M. Kramer, A. Zvezdin, V. Morozov, Y. Roblin, F. Lin, and C. Jarvis, "Innovative applications of genetic algorithms to problems in accelerator physics," *Phys. Rev. ST Accel. Beams*, vol. 16, p. 010101, Jan. 2014.

[8] S. Bleuler, M. Laumanns, L. Thiele, and E. Zitzler, "PISA–A platform and programming language independent interface for search algorithms," ETH Zürich, Institut fur Technische Informatik und Kommunikationsnetze ETH Zurich, Tech Report 154, 2002.

[9] J. Mockus, "Bayesian approach to global optimization: theory and applications," Doordrecht, Netherlands: Kluwer Academic, 2013.

[10] `http://sheffieldml.github.io/GPyOpt/index.html`

[11] B. Terzić, A. Hofler, C. Reeves, S. Khan, G. Krafft, J. Benesch, A. Freyberger, and D. Ranjan, "Simultaneous optimization of the cavity heat load and trip rates in linacs using a genetic algorithm," *Phys. Rev. ST Accel. Beams*, vol. 17, p. 101003, Oct. 2014.