ADVANCED MODELING OF ACCELERATORS FOR NEXT GENERATION LIGHT SOURCES*

J. Qiang[#], C. Mitchell, R. Ryne, M. Venturini, LBNL, Berkeley, CA94720, U.S.A.

Abstract

Next generation light sources are critical to scientific discovery in numerous fields such as materials science, chemistry, and the biosciences. In this paper, we will review status of accelerator modeling of next generation light sources and report on the progress that we have made in developing advanced computational tools for high fidelity start-to-end simulation and optimization of these sources. We will present numerical methods and application examples for modeling of a next generation FEL-based light source.

INTRODUCTION

Next generation high brightness FEL X-ray light sources provide great opportunity for scientific discovery in many fields. Some grand challenges in basic energy science research involve controlling material processes at the electron level, designing energy efficient synthesis of new matter with desired properties, understanding the effects of complex correlation between the atomic and the electronic constituents on matter properties, mastering energy and information on the nanoscale, and characterizing and controlling matter away or far away from equilibrium [1]. To meet those challenges, one needs ultrafast, ultrabright, tunable X-ray light sources. A next generation FEL-based X-ray light source with its high spatial and temporal coherence provides many orders of magnitude higher peak brightness than a conventional third generation light source. A number of next generation FELs were built and are under construction or design. The performance of those next generation light sources depends critically on the quality of the electron beam entering the radiation undulator. However, the presence of collective effects (space-charge, coherent synchrotron radiation (CSR), structure and resistive wakefields) inside the beam can significantly degrade the beam quality. Figure 1 shows the longitudinal phase space distribution of an electron beam at the entrance to an undulator with and without space-charge and CSR effects. It is seen that the presence of those collective effects causes large phase space modulation, increases energy spread and significantly degrades the beam quality. Advanced computational methods are needed in order to accurately model those effects.

ADVANCED COMPUTATIONAL METHODS

The simulation of a next generation light source starts with an initial particle distribution behind the cathode. In this study, we developed a second-order computational
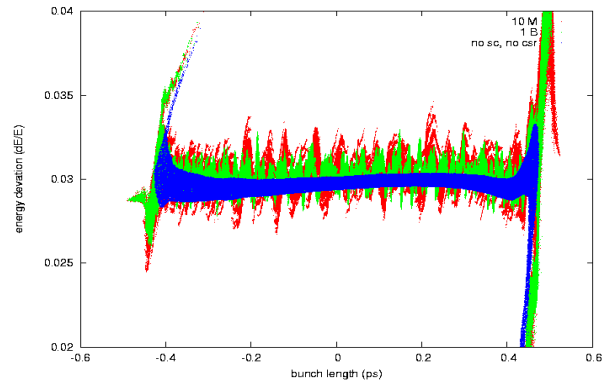
Figure 1: Final longitudinal phase space without (blue) and with longitudinal space-charge and CSR effects (green and red).

model to simulate the generation of photo-electrons from a photo-cathode driven by an external laser. For a given laser temporal pulse distribution and spatial distribution, a number of electrons carrying the total emitted charge are generated behind the cathode with the same transverse distribution as the laser's and the same longitudinal distribution as the laser's temporal profile times a reference longitudinal velocity v_0. Those electrons are moved outside the photocathode using N time steps. Here, the time step size is $\Delta t = t_{laser}/N$, where t_{laser} is the total laser pulse length. Figure 2 shows a schematic plot of the emission process in the simulation. In the second-order photo-electron emission model, the positions and the velocities of an electron after emission are given by

$$
\begin{aligned}
x &= x_0 + v_{x0}\delta t + \frac{1}{2}a_x(\delta t)^2 \\
v_x &= v_{x0} + a_x\delta t \\
z &= v_{z0}\delta t + \frac{1}{2}a_z(\delta t)^2 \\
v_z &= v_{z0} + a_z\delta t
\end{aligned}
\tag{1}
$$

where $\delta t = z/v_0$, z is the electron longitudinal coordinate out of the photocathode right after emission during the time step Δt, and $\boldsymbol{a}$ is the acceleration that can be calculated using the field at the photo-cathode surface. As a comparison, we generate *300* pC photo electrons from a photo-injector gun using the second-order emission model and the first-order emission model (without including acceleration). The current profile of the beam shortly after emission is shown in Fig. 3 using different numbers of emission steps in the above emission models. It is seen that the crude first order emission model can introduce artificial modulation of the beam. A much larger number of emission steps (a factor of 4) i.e. smaller emission step size, are needed in order to achieve the same level smoothness of the current profile.
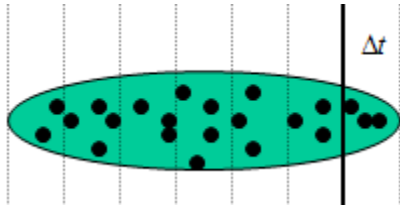
Figure 2: A schematic plot of the electron emission process at the photocathode.
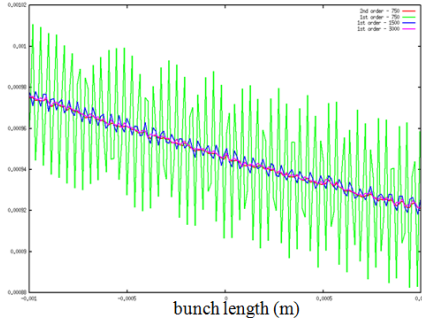

Figure 3: A section of current profile of the beam after emission using the second-order emission model (red) and the first-order emission model with different numbers of emission steps.

Efficient methods to calculate 3D space-charge effects, 1D longitudinal CSR effects, structure and resistive wall wakefields were developed in past studies [2,3,4]. However, a brute force approach to resolve the fine structure microbunching below optical wavelength throughout the entire electron beam can be computationally expensive. A multi-level Poisson solver with adaptive mesh refinement in selected regions of the beam will be more efficient. An illustration plot of this method with three levels of refinement in the longitudinal direction is shown in Fig. 4. In the multi-level solver, the potential on a grid i at level n is given as

$$\phi^n(x^i) = \sum_{j \in mesh(n-1)} G^{n-1}(x^i, x^j)\rho^j - \sum_{j \in mesh(n)(n-1)} G^{n-1}(x^i, x^j)\rho^j + \sum_{j \in mesh(n)} G^n(x^i, x^j)\rho^j$$

where *mesh(n-1)* denotes the mesh grid on level *n-1*, *mesh(n)(n-1)* denotes the mesh grid at level *n* also belonging to level *n-1*, and *mesh(n)* denotes the mesh grid on level *n*. For *n* less than or equal to *1*, the Green function G is *0*. In the above equation, the first two terms correspond to the long-range forces contributed by electrons on the parent grid level *n-1*. The third term denotes the short-range force on the child grid level *n*. For the open boundary condition assumed here, each convolution in the above equation can be computed by using an FFT-based method after zero padding. This results in a computational complexity of *O(Nlog(N)),* where *N* is the number of mesh grid points at each level. In the multi-level particle-mesh method, the convolution at child mesh grid level *n* can be further decomposed into the summation of a long-range force at level *n* and a short-range force at level *n+1* in a selected domain. Such a decomposition process can be repeated for many levels until the desired resolution is

achieved in the desired region. The field for particles located inside the computational domain at level *n* is equal to the summation of field values interpolated from this level and the levels below the level *n*.
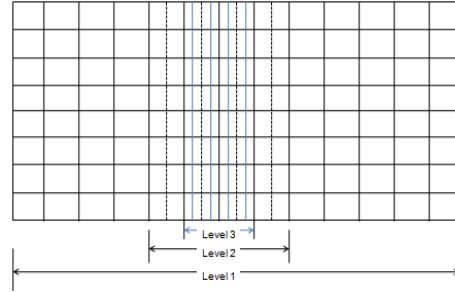

Figure 4: An illustration plot of multi-level particle mesh refinement.
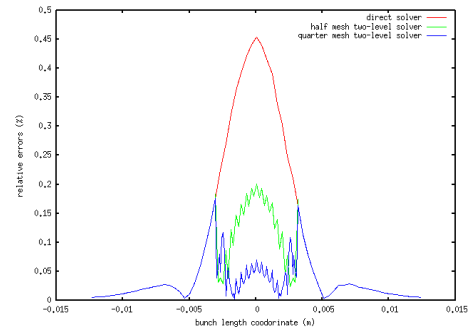

Figure 5: Relative electric potential errors across the beam using direct Poisson solver (red) and the multi-level Poisson solver (green and blue).

As a test, we calculated the electric potential in the middle of a beam with the direct Poisson solver and the above new multi-level Poisson solver. The relative errors of the solution from both methods are given in Fig. 5. It is seen that the new multi-level Poisson solver results in about a factor of four less errors in comparison to the direct solver. Such a multi-level Poisson solver with high accuracy in the selected region can significantly save the computational time in the simulation.

Numerical integration is used to advance the particle spatial and momentum coordinates with given space-charge and external fields. A widely used numerical integrator in the literature is the so-called Boris integrator [5]. The Boris integrator has 2nd order numerical accuracy and works well at lower energy when the relativistic factor γ is small. As the electron beam energy increases, there is a strong cancellation between the electric space-charge force and the magnetic space-charge force in the laboratory frame. In the case of a large γ factor for a relativistic electron beam, the numerical errors in computing the cancellation between the electric space-charge force and the magnetic space-charge force using the Boris integrator become worse due to the fact that the magnetic force and the electric force are not calculated in the same time step. In the limit as γ goes to infinity, the Boris integrator requires that both the magnetic field and the electric field disappear with any given time step size, which is unphysical. Here, we developed a new numerical integra-

tor that has 2nd-order numerical accuracy but works correctly for the relativistic electron beam. The one step update in particle coordinates for the new numerical integrator is given as:

$$x^{i+1/2} = x^{i-1/2} + \Delta t \, v^i$$
$$(\gamma v)_1^{i+1} = (\gamma v)^i + \Delta t \, \frac{q}{m}(E^{i+1/2} + v^i \times B^{i+1/2})$$
$$u^i = (v_1^{i+1} + v^i)/2$$
$$(\gamma v)^{i+1} = (\gamma v)^i + \Delta t \, \frac{q}{m}(E^{i+1/2} + u^i \times B^{i+1/2})$$

The new numerical integrator computes the electric force and the magnetic force in the same time step and correctly handles the cancellation of the two forces in the laboratory frame. As a test example, we simulated a *1* nC, *250* MeV bunch that is spherical in the beam frame and that freely expands in vacuum. Due to the symmetry of the space-charge forces in the beam frame, the beam should stay spherical. Figure 6 shows the evolution of the horizontal, vertical and longitudinal rms sizes in the beam frame from the simulation using the Boris integrator and the new proposed integrator with the same time step size. It is seen that the Boris integrator gives a wrong prediction of the transverse X and Y sizes while the new algorithm correctly predicts the rms size evolution.
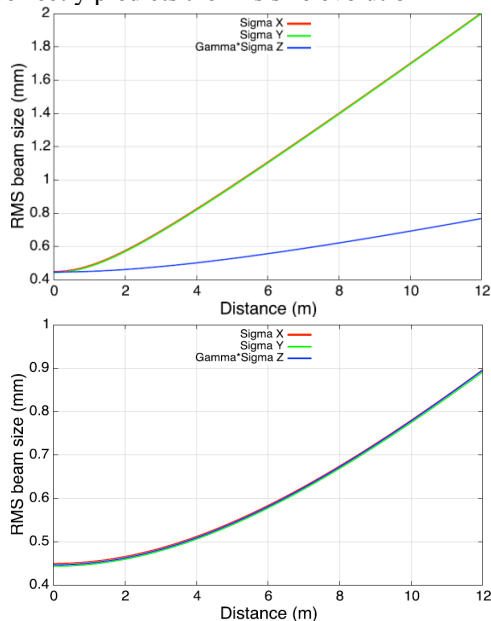

Figure 6: Transverse and longitudinal rms sizes evolution from the Boris integrator (top) and from the new integrator (bottom) in a freely expanding sphere beam.

APPLICATIONS*

The advanced computational tools developed here were implemented into a parallel 3D particle-in-cell tracking code suite, IMPACT [2, 6, 7]. IMPACT has become an essential tool for the design of the LCLS-II [8], and in particular for the investigation of the microbunching instability [9]. As a way to illustrate the code capabilities, we briefly discuss the application of the IMPACT to a specific problem of LCLS-II lattice-design optimization.

Figure 7 shows a schematic layout of the LCLS-II beam delivery system [10]. The combination of the long

(>2 km) transport beam line following the L3-Linac and leading to the FEL undulators, the presence of several doglegs, and a relatively low electron-beam energy *(4 GeV)*, aggravates the microbunching effects. To mitigate the problem it was proposed to insert weak chicanes adjacent to the doglegs' dipoles to provide local R_{56}-compensation [9]. Specifically, two local compensation chicanes have been designed to compensate R_{56} in the first dogleg (DL1), with three more compensation chicanes inserted after the spreader in the hard x-ray transport line, and two more inserted in the soft x-ray transport line. Because of an interesting 3D space-charge effect [11], it turns out that exact local R_{56}-compensation does not provide maximum suppression of the microbunching instability. To find the optimum setting of the compensating chicanes we scanned the bending angle of all dipole magnets in all compensation chicanes by a same scaling factor; with *0%* corresponding to the nominal design setting for exact R_{56}-compensation. As a measure of the instability we considered the rms current fluctuations along the bunch core relative to a smoothened current profile fitted to a cubic polynomial, as observed at the entrance of the FEL undulators.

Figure 8 shows the rms current fluctuation as a function of the percentage increase from nominal of the compensating chicanes' bend angle (for transport of *100 pC* bunches to the hard x-ray FEL). It is seen that *20%* increase, corresponding to about 1.5 times larger $|R_{56}|$, yields the least current fluctuation.

Figure 9 shows the final longitudinal phase space and current profile with the nominal compensation setting and 20% increase of bend angle. Notice how the *20%* setting results in much reduced microbunching as seen in both longitudinal phase space and current profile.
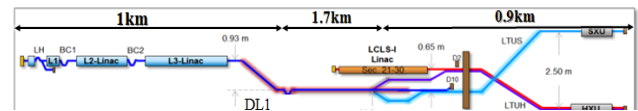

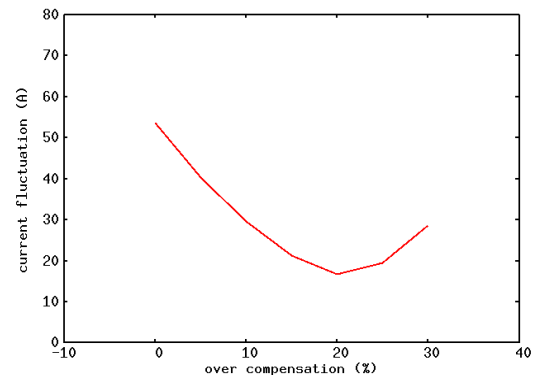Figure 7: Schematic of the LCLS-II beam delivery system.

Figure 8: RMS bunch current-profile fluctuations as a function of relative deviation from nominal of the compensation chicanes' bend angle (entrance of the hard x-ray FEL; *100* pC charge) .
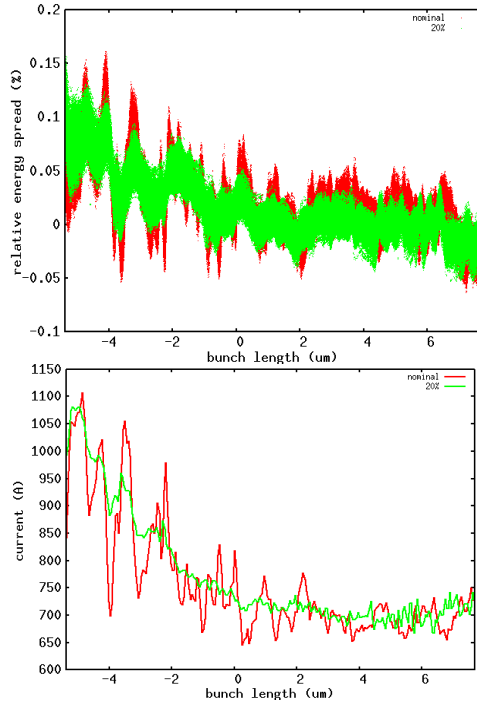
Figure 9: Longitudinal phase space (top) and current profile (bottom) with nominal (red) and *20%* (green) larger bend angle in the compensating chicanes (*100* pC, entrance of hard x-ray FEL)

PARALLEL DESIGN PARAMETER OPTIMIZATION

Differential evolution is a simple yet efficient population-based, stochastic, evolutionary algorithm for global parameter optimization [12]. In a number of studies, the differential evolution algorithm performed effectively in comparison to several stochastic optimization methods such as simulated annealing, controlled random search, evolutionary programming, the particle swarm method, and genetic algorithm.

The differential evolution algorithm starts with a population initialization. A group of NP solutions in the control parameter space is randomly generated to form the initial population. This initial population can be generated by sampling from a uniform distribution within the parameter space if no prior information about the optimal solution is available, or by sampling from a known distribution (e.g., Gaussian) if some prior information is available. After initialization, the differential evolution algorithm updates the population from one generation to the next generation until reaching a convergence condition or until the maximum number of function evaluations is reached. At each generation, the update step consists of three operations: mutation, crossover, and selection. The mutation and the crossover operations produce new candidates for the next generation population and the selection operation is used to select from among these candidates the appropriate solutions to be included in the next generation. During the mutation operation stage, for each population member (target vector) x_i, $i = 1, 2, 3, \bullet\bullet\bullet, NP$

at generation G, a new mutant vector v_i is generated by following a mutation strategy. Some commonly used mutation strategies in past studies are [12]:

$$\text{DE/rand/1}: \ \vec{v}_i = \vec{x}_{r_1} + F_{xc}(\vec{x}_{r_2} - \vec{x}_{r_3})$$

$$\text{DE/rand/2}: \ \vec{v}_i = \vec{x}_{r_1} + F_{xc}(\vec{x}_{r_2} - \vec{x}_{r_3}) + F_{xc}(\vec{x}_{r_4} - \vec{x}_{r_5})$$

$$\text{DE/best/1}: \ \vec{v}_i = \vec{x}_b + F_{xc}(\vec{x}_{r_1} - \vec{x}_{r_2})$$

$$\text{DE/best/2}: \ \vec{v}_i = \vec{x}_b + F_{xc}(\vec{x}_{r_1} - \vec{x}_{r_2}) + F_{xc}(\vec{x}_{r_3} - \vec{x}_{r_4})$$

$$\text{DE/current-to-best/1}: \ \vec{v}_i = \vec{x}_i + F_{cr}(\vec{x}_b - \vec{x}_i) + F_{xc}(\vec{x}_{r_1} - \vec{x}_{r_2})$$

$$\text{DE/current-to-best/2}: \ \vec{v}_i = \vec{x}_i + F_{cr}(\vec{x}_b - \vec{x}_i) + F_{xc}(\vec{x}_{r_1} - \vec{x}_{r_2})$$
$$+ F_{xc}(\vec{x}_{r_3} - \vec{x}_{r_4})$$

$$\text{DE/current-to-rand/1}: \ \vec{v}_i = \vec{x}_i + F_{cr}(\vec{x}_{r_1} - \vec{x}_i) + F_{xc}(\vec{x}_{r_2} - \vec{x}_{r_3})$$

$$\text{DE/current-to-rand/2}: \ \vec{v}_i = \vec{x}_i + F_{cr}(\vec{x}_{r_1} - \vec{x}_i) + F_{xc}(\vec{x}_{r_2} - \vec{x}_{r_3})$$
$$+ F_{xc}(\vec{x}_{r_4} - \vec{x}_{r_5})$$

$$\text{DE/rand-to-best/1}: \ \vec{v}_i = \vec{x}_{r_1} + F_{cr}(\vec{x}_b - \vec{x}_i) + F_{xc}(\vec{x}_{r_2} - \vec{x}_{r_3})$$

$$\text{DE/rand-to-best/2}: \ \vec{v}_i = \vec{x}_{r_1} + F_{cr}(\vec{x}_b - \vec{x}_i) + F_{xc}(\vec{x}_{r_2} - \vec{x}_{r_3})$$
$$+ F_{xc}(\vec{x}_{r_4} - \vec{x}_{r_5})$$

where the integers r_1, r_2, r_3, r_4 and r_5 are chosen randomly from the interval *[1,NP]* and are different from the current index i, F_{xc} is a real scaling factor that controls the amplification of the differential variation, x_b is the best solution among the *NP* population members at the generation G, and F_{cr} is a weight for the combination between the original target vector and the best parent vector or the random parent vector.

The use of multiple mutation strategies makes the differential evolution algorithm complicated to implement and use appropriately. In this study, we developed a new adaptive unified differential evolution (AuDE) algorithm for global optimization [13]. This algorithm uses only a single mutation expression, but encompasses almost all commonly-used mutation strategies as special cases. It is mathematically simpler than the conventional algorithm with its multiple mutation strategies, and also provides users the flexibility to explore new combinations of conventional mutation strategies during optimization. By making the control parameters in the mutation and cross-over stages self-adaptive, it also sets the user free from choosing an appropriate set of control parameters for each optimization problem.

This single unified mutation expression can be written as:

$$\vec{v}_i = \vec{x}_i + F_1(\vec{x}_b - \vec{x}_i) + F_2(\vec{x}_{r_1} - \vec{x}_i) + F_3(\vec{x}_{r_2} - \vec{x}_{r_3}) + F_4(\vec{x}_{r_4} - \vec{x}_{r_5})$$

This unified mutation expression represents a combination of exploitation (from the best found solution) and exploration (from the random solutions) to generate a new mutant solution. For example, one can see that for $F_1 = 0$, $F_2 = 1$, and $F_4 = 0$, this equation reduces to DE/rand/1; for $F_1 = 1$, $F_2 = 0$, and $F_4 = 0$, it reduces to DE/best/1. Using the single unified expression the ten mutation strategies of the standard differential evolution algorithm can be written as a single mutation expression. This new expression provides an opportunity to explore more broadly the space of mutation operators. By using a different set of parameters F_1, F_2, F_3, F_4, a new mutation strategy can be achieved.

The goal of multi-objective optimization is to find the Pareto front in the objective function solution space. The Pareto optimal front is a collection of all non-dominated solutions in the whole feasible solution space. Any other solution in the feasible solution space will be dominated by those solutions on the Pareto optimal front. In the multi-objective optimization, a solution A is said to dominate a solution B if all components of A are at least as good as those of B (with at least one component strictly better). The solution A is non-dominated if it is not dominated by any solution within the group. In this study, we have developed a new parallel multi-objective differential evolution algorithm with variable population size and external storage. The algorithm in each generation and external storage can be summarized in the following steps:

• Step 0: Define the minimum parent size, NPmin and the maximum size, NPmax of the parent population. Define the maximum size of the external storage, NPext.
• Step 1: An initial population of NPini parameter vectors is chosen randomly to cover the entire solution space.
• Step 2: Generate the offspring population using the adaptive unified differential evolutionary algorithm.
• Step 3: Check the new population against boundary conditions and constraints.
• Step 4: Combine the new population with the existing parent population from external storage. Nondominated solutions (Ndom) are found from this group of solutions and min(Ndom, NPext) of solutions are put back into external storage. Pruning is used if Ndom > NPext. NP parent solutions are selected from this group of solutions for next generation production. If NPmin ≤ Ndom ≤ NPmax, NP = Ndom. Otherwise, NP = NPmin if Ndom < NPmin and NP = NPmax if Ndom > NPmax. The elitism is emphasized through keeping the non-dominated solutions while the diversity is maintained by penalizing the over-crowded solutions through pruning.
• Step 5: If the stopping condition is met, stop. Otherwise, return to Step 2.

As a test of above algorithm, we used the following two objective functions:

$$f_1(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^{3}\left(x_i - \tfrac{1}{\sqrt{3}}\right)^2\right)$$
$$f_2(\mathbf{x}) = 1 - \exp\left(-\sum_{i=1}^{3}\left(x_i + \tfrac{1}{\sqrt{3}}\right)^2\right)$$

The distance to the Pareto front as a function of function evaluation number is shown in Fig. 10. It is seen that new proposed algorithm ("variation population with external storage" or VPES) can converge significantly faster than a widely used genetic algorithm NSGA-II [14].

The above population based differential evolutionary optimization algorithm naturally leads to a multi-level parallel implementation. Our method contains two levels of parallelization. First, the whole population is distributed among a number of groups of parallel processors. Each group of processors contains a subset of the whole population. Different sets of the sub-population can be tracked
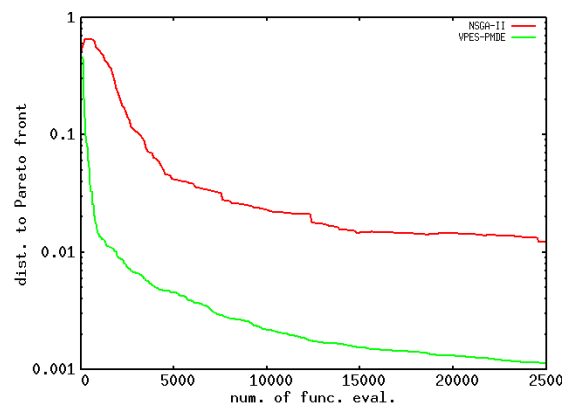

Figure 10: Distance to the Pareto front as a function of number of function evaluation using the genetic algorithm (NSGA-II) and the new proposed algorithm (VPES-PMDE).

simultaneously. Second, each objective function evaluation corresponds to an accelerator simulation, for which parallel simulation codes are available. A good scalability of the parallel differential evolution algorithm has been demonstrated on a Cray XT5 computer using 100, 000 processors using this method.

ACKNOWLEDGMENTS

REFERENCES

[1] J. Hemminger, et al., "Directing matter and energy: five challenges for science and the imagination," Report from Basic Energy Sciences Advisory Committee, 2007.
[2] J. Qiang, et al., PRST-AB 9, 044204 (2006).
[3] J. Qiang, et al, NIMA 682, 49 (2012).
[4] C. Mitchell, et al, NIMA 715, 119 (2013).
[5] J. P. Boris, "Relativistic plasma simulation-optimization of a hybrid code," in Proceedings of the Fourth Conference on Numerical Simulation Plasmas Naval Research Laboratory, Washington, D.C., 1970, pp. 3–67.
[6] J. Qiang, et al., J. of Comp. Phys., 163, 434 (2000).
[7] J. Qiang, et al., PRST-AB 12, 100702 (2009).
[8] T. O. Raubenheimer, "Technical challenges of the LCLS-II CW X-ray FEL," in proceedings of IPAC2015, Richmod, VA, USA.
[9] M. Venturini, et al., IPAC 2015, p. 1843 (2015).
[10] P. Emma et al., in proceedings of FEL'14, THP025.
[11] M. Venturini and J. Qiang, PRST-AB, 18, 054401, (2015).
[12] R. Storn and K. Price, J. of Global Optimization, 11, p. 341 (1997).
[13] J. Qiang and C. Mitchell, in Optimization Online digest, Feb. 2015.
[14] K. Deb, et al., IEEE Trans. Evolutionary Computation 6, p. 183 (2002).