Novel Methods in the Particle-In-Cell Accelerator Code-Framework Warp

by

J.-L. Vay, D.P. Grote, R.H. Cohen, A. Friedman

AFRD
Accelerator and Fusion Research Division
Ernest Orlando Lawrence Berkeley National Laboratory
Berkeley, California 94720

Sept. 7-9, 2011

This work was supported by the Director, Office of Science, Office of Fusion Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor The Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or The Regents of the University of California.

This work was supported by the Director, Office of Science, Office of Fusion Energy Sciences, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Novel Methods in the Particle-In-Cell Accelerator Code-Framework Warp

J.-L. Vay
Lawrence Berkeley National Laboratory, Berkeley, CA, USA
E-mail: `jlvay@lbl.gov`

D. P. Grote, R. H. Cohen, A. Friedman
Lawrence Livermore National Laboratory, Livermore, CA, USA

Abstract. The Particle-In-Cell (PIC) Code-Framework Warp is being developed by the Heavy Ion Fusion Science Virtual National Laboratory (HIFS-VNL) to guide the development of accelerators that can deliver beams suitable for high energy density experiments and implosion of inertial fusion capsules. It is also applied in various areas outside the Heavy Ion Fusion program to the study and design of existing and next-generation high-energy accelerators, including the study of electron cloud effects and laser wakefield acceleration for example. This paper presents an overview of Warp's capabilities, summarizing recent original numerical methods that were developed by the HIFS-VNL (including Particle-In-Cell with Adaptive Mesh Refinement, a large-timestep "drift-Lorentz" mover for arbitrarily magnetized species, a relativistic Lorentz invariant leapfrog particle pusher, simulations in Lorentz boosted frames, an electromagnetic solver with tunable numerical dispersion and efficient stride-based digital filtering), with great emphasis on the description of the mesh refinement capability. Selected examples of applications of the methods to the abovementioned fields are given.

Keywords: particle-in-cell, plasma simulation, adaptive mesh refinement

1. Introduction

The Particle-In-Cell (PIC) Framework Warp [1] was originally developed to simulate space-charge-dominated beam dynamics in induction accelerators for heavy-ion fusion (HIF). It is currently being developed primarily by the Heavy Ion Fusion Science Virtual National Laboratory (HIFS-VNL) collaboration, to guide the development of accelerators that can deliver beams suitable for high energy density experiments and implosion of inertial fusion capsules [2]. In recent years, the physics models in the code have been generalized, so that Warp can model beam injection, complicated boundary conditions, denser plasmas, a wide variety of accelerator lattice components, and the non-ideal physics of beams interacting with walls and plasmas. The code now has an international user base and is being applied to projects both within and far removed from the HIF community. Ongoing or recent examples of applications outside HIF include the modeling of plasma traps for the production of anti-Hydrogen [3], Paul traps [4, 5], non-conventional Penning-Malmberg micro-trap [6], transport of electron beams in the UMER ring [7], ECR ion sources [8], capture and control of laser-accelerated proton beams [9], and fundamental studies of multipacting [10]. It is also applied to the study and design of existing and next generation high-energy accelerators including the study of electron cloud effects [11], coherent synchrotron radiation [12] and laser wakefield acceleration [13].

Numerical simulations are essential for all aspects of the Heavy Ion Fusion Science program. Detailed simulations are needed for planning and interpreting the results of ongoing experiments, as well as for modeling physics of the NDCX-II accelerator (currently being assembled at Lawrence Berkeley National Laboratory) [14] and advanced fusion-driver concepts. In addition, simulations are the principal tools in basic beam physics studies, such as the dynamics of space-charge-dominated beams in the presence of stray electrons, neutralized drift-compression of intense beams, plasma injection, and the use of solenoids to transport high-current ion beams. Other activities include the exploration of basic questions of accelerator physics, such as emittance growth (dilution of the phase space), beam instabilities, formation of outlying particle populations called halos, the analysis of experimental data, and the optimization of accelerator components, such as the final focus beam optics system.

Simulations are crucial to all these pursuits because, in the regimes of interest, the physics cannot be thoroughly captured by analytic means, due to the complicated geometries, the non-ideal applied fields, and the intense, non-uniform space-charge field of the beam along with plasma and stray charged particles. These studies have necessitated the introduction or development of advanced numerical methods, including methods to model multiple-species effects in accelerators and chambers, efficient ensemble methods, particle advance algorithms that allow a longer time step, and adaptive mesh refinement (AMR). The algorithms that have been implemented span the range between (a) “computationally intensive” massively-parallel methods based on low levels of approximation, typically with explicit solvers on uniform grids, small time steps, and large numbers of macroparticles, and (b) “algorithmically intensive” moderately-parallel methods using higher levels of approximations, typically involving implicit methods, global solvers, AMR grids, large time steps, and moderate numbers of macroparticles.

Warp uses a flexible multi-species particle-in-cell model to describe beam dynamics and the electrostatic or electromagnetic fields in particle accelerators. While the core routines of Warp solve finite-difference representations of Maxwell’s equations and relativistic or non-relativistic motion equations, the code also uses a large collection of subordinate models to describe lattice elements (machine components) and such physical processes as beam injection, desorption, and ionization. Warp is written in a combination of Fortran for the computationally intensive tasks, Python for the high level controlling framework, and C for the interface between the two. The interface generator Forthon [15] is used to generate the necessary wrapping code which allows access at the Python level to the Fortran database and to the Fortran subroutines and functions. The use of dynamic loading of modules at the Python level gives to the user a very high degree of flexibility and expandability. Warp can be used as a code (the user gives input parameters, runs the main loop and gathers data from embedded diagnostics), as a framework (the user writes the main loop at the Pyhon level using existing modules and eventually additional modules developed by the user or by a third party), or in an intermediate mode, i.e. as a code with framework capability (the user uses the main loop completed by other modules). This design allows for great versatility while keeping the quantity and size of the core modules to a minimum, since specialized modules can be provided and maintained by the users who need them.

A general description of the code including its Python interface is given in section 2. Sections 3 and 4 present the latest developments in Warp in particle pushing and field solving algorithms, with great emphasis on mesh refinement. The various modes of operation with examples are given in section 5 and an outlook in section 6.

2. General description

Warp is a 3D time-dependent multiple-species particle-in-cell (PIC) framework, with the addition of an accelerator lattice description. Warped coordinates are used to advance particles in a curved beam pipe [16]. Self-fields are obtained via Poisson equations for the scalar and vector potentials, or via Maxwell equations for a full electromagnetic description, while simplified models are available for the self- magnetic and inductive forces for ultra-relativistic beams [17]. Warp also has 2D models, using Cartesian or cylindrical geometry, as well as a module representing the beam with a 4-D Vlasov formulation and with low-order moment equations. Models are available for background gas, wall effects (e.g. secondary emission of electrons using the subroutines from the Posinst code [18, 19]), stray electrons, space-charge-limited and source-limited particle emission, and atomic processes such as ionization.

Elaborate initialization and runtime options allow realistic modeling of charged particle accelerators. A beam may be initialized with one of a broad selection of analytic distributions [20] or with a distribution synthesized from experimental data, or can be emitted from a flat or curved surface. The lattice description includes an extensive list of elements including electric or magnetic dipoles, quadrupoles, sextupoles, solenoids, accelerating gaps, *etc.* Lattice-element fields may be represented at several levels of detail, from simple hard-edge analytic forms to first-principles 3D calculations. The fields can be time-dependent. When hard-edged fields are applied to the particles, “residence corrections” are used, where, upon entering or exiting the element, the applied field is scaled by the fraction of the time-step spent inside the element. Poisson’s equation can be solved using several methods, including FFT and Multigrid. With multigrid, the Shortley-Weller method [21] for the subgrid-resolution description of conductors allows the use of complicated boundary conditions. With the FFT solvers, capacity matrices can be used to incorporate internal boundary conditions. The electromagnetic (EM) solver is based on the Yee staggered discretization of the field components, using either standard Leapfrog finite-difference (FDTD) integration [22] or non-standard finite-difference (NSFD) [23]. Adaptive mesh refinement (AMR) is available with the multigrid Poisson solver [24], and mesh refinement (MR) is available with the electromagnetic solver [25].

Warp's parallelization is based on domain decomposition using the MPI library for message passing, and allows decomposition in 1, 2 or 3 dimensions. With the electrostatic solver, Warp allows independent spatial decompositions for particles and field quantities, enabling the particle and field advances to be load-balanced independently. In transverse-slice 2D runs, the field solution is repeated on each node, but solved in parallel by processors within a node. Parallel communications are typically performed at the Fortran level for core calculations and diagnostics, and sometimes (mostly for high level diagnostics and prototyping) at the Python level.

WARP provides a very high level of flexibility to the user for data analysis and visualization. WARP has an extensive set of 2D and 3D plotting routines based on the graphical packages pygist [26] and Opyndx (python interface to OpenDX) [27]. With Pygist, plots can be displayed on a screen allowing interactive manipulation, and they can be saved to disk in the vector CGM format, which is compact and portable. Opyndx is used to create stills and movies of 3D rendering for better understanding and intuition development of the increasingly complex physical phenomena being simulated. The rendering can be manipulated interactively and written to disk. Runtime diagnostic plots are made by a single processor, which assembles data from the other processors. Data can be saved into various portable binary file formats, including Python pickle, HDF5 (via pytables), PDB (Portable Data Binary from LLNL pact library) or any other I/O library that the user can access via Python. For highest efficiency, data dumps are made by having each processor write its data to a separate file. The user has direct access to all the data (even Fortran derived type objects), allowing data selection or reduction to be performed in-situ. Saving only the reduced datasets to disk for offline analysis significantly reduces the amount of data that is to be written.

2.1. Python interface

The user interface of Warp is Python, a high level, object oriented, interactive and scripting language designed for ease of use and flexibility. Warp is built as a module that is importable into a standard Python session. From Python, all of the data in Warp is accessible and its routines callable. An input file for Warp is a Python program, allowing use of the full richness of the Python programming language to set up, control, steer, diagnose and post-process a Warp simulation. This allows great flexibility in how Warp is used. Input files range from short files with minimal setup that use a standard advance scheme, to extensive collections of files that pick and choose the pieces of Warp that are used and carefully orchestrate how a simulation proceeds. A user can extend

Warp with his or her own collection of Python scripts, as well as Fortran, C or C++ routines with Python wrappers.

Much of Warp is written directly in Python. There is extensive Python code for handling the interface to various pieces of Warp, for example the multiple field solvers, the particle scrapers, the lattice description, diagnostics and plotting. These provide a high level interface for ease of use. The underlying computation intensive part of Warp is written in Fortran 95. The interface between the Fortran and Python is created by the Forthon package [15].

Forthon generates the C code and runtime variable database which allows access to the Fortran data and wraps the Fortran subroutines and functions so that they are callable from Python. It first generates the interface from description files which lay out the modules, variables and subroutines that are to be accessible from Python, then creates a database linking the memory addresses of variables in the modules to Python level variables, allowing direct read and write access.

Forthon uses Numpy for handling arrays which can be dynamically allocatable. In that case, Forthon keeps track of the location of the array, updating it as needed when the array is allocated or reallocated. Arrays can be allocated in either Fortran or in Python. For subroutines and functions, Forthon creates a Python callable wrapper in C that does the appropriate conversion of Python variables into Fortran accessible memory. When multi-dimensional arrays are passed in, Forthon checks the ordering and, if needed, does the appropriate transpose to put the array in Fortran ordering. Thus the ordering of indices is, to the user, the same in Python as it is in Fortran. Input arguments are checked for validity, ensuring that the input has the correct type and that arrays are dimensioned appropriately.

Forthon provides wrapping for Fortran derived type variables. These are described in the variable description files, similarly to a Fortran module. For each instance of a derived type that is created, Forthon sets up a database of the memory, giving access to it from Python. Instances can be created in either Fortran or Python, and can be passed into Fortran from Python. A derived type can include instances of derived type, either statically or through a pointer, including instances of the same type.

Warp makes extensive use of the capabilities of Forthon. The vast majority of data is handled using dynamically allocated multi-dimensional arrays. There are various uses of derived types, encapsulating data and allowing complex, hierarchical data structures. This code architecture allows rapid development and testing (in both Fortran and Python) of both the underlying code and the user input files.

3. Particle pushers

The default particle pusher is the so-called “Boris” pusher which is based on centered finite difference (leapfrog) and is second order [28], with optional “$\tan\alpha/\alpha$” correction for exact gyration angle [29].

3.1. Hybrid Drift-Lorentz

It was observed in [30] that the Boris pusher causes particles to gyrate with spuriously large radius for time steps that are large compared to the gyroperiod, albeit with the correct drift velocities (provided the gradients are still sampled adequately). A new solver that interpolates between the Boris velocity push and a drift kinetic advance was developed and implemented in Warp [31, 32]; it reproduces both the correct drift velocity and gyroradius for an arbitrarily large ratio of time step δt relative to cyclotron period τ_c, as well as correct detailed orbit dynamics in the small-timestep limit. Schematically, the mover updates the particle position using an interpolated velocity perpendicular to the magnetic field, $\mathbf{v}_{\perp,eff} = \alpha\mathbf{v}_{\perp} + (1-\alpha)\mathbf{v}_d$ where $\mathbf{v}_{\perp}$ and $\mathbf{v}_d$ are, respectively, the perpendicular component of the full particle velocity, including gyro motion but with a correction for magnetic-mirror forces and $\mathbf{v}_d$ is the drift velocity, and α is an interpolation coefficient, $\alpha = 1/[1+(\omega_c\delta t/2)^2]^{1/2}$, where ω_c and δt are the cyclotron frequency and timestep, repsectively. The pusher has provided an order of magnitude or more saving in computing resources in the simulations of electron cloud effects in the HCX experiment [31, 33]. An implicit time-advance scheme incorporating drift-Lorentz interpolation has also been developed [32]. An alternative approach, which does not require explicit drift-velocity calculations and considerably relaxes but does not eliminate the $\delta t/\tau_c$ constraint, has been developed by Genoni *et al.* [34].

3.2. Lorentz invariant advance

The relativistic version of the Boris (or Hybrid Lorentz-Drift) particle pusher does not maintain strict Lorentz invariance, resulting eventually in unacceptably large inacuracies when modeling the transport of ultra-relativistic beams in accelerators. To this effect, an alternative to the Boris pusher that conserves strict Lorentz invariance (to machine precision) was developed and implemented in Warp, and its effectiveness demonstrated on the modeling from first principles of the interaction of a 500 GeV proton beam with a background of electrons [17]. The pusher has subsequently been implemented by others and has also proven useful for correctly capturing the drift speed of electrons of a highly magnetized relativistic electron-ion flow in astrophysical simulations using the code TRISTAN [35].

3.3. linear maps

For the modeling of high energy beams for which space charge effects are relatively weak, and thus time steps can be large compared to the residence time within lattice elements, particles are pushed using linear maps as $(\mathbf{x},\mathbf{v})_{s+ds} = \mathcal{M}(\mathbf{x},\mathbf{v})_s$ where $\mathbf{x}$ and $\mathbf{v}$ are respectively the position and velocity of the particles and $\mathcal{M}$ is a 6×6 transfer matrix. While Warp does not normally require definition of a “reference orbit”, the maps do assume a paraxial limit, but there are no difficulties in principle to augment the order of the map to include some nonlinear effects if these are deemed important. Mapping is currently available for continuous "smooth" focusing, drifts, bends and quadrupoles.

4. Field solvers and Mesh Refinement

4.1. Field solvers

4.1.1. Electrostatic/Magnetostatic Warp includes electrostatic and magnetostatic solvers. For electrostatic, either an FFT based solver or a multigrid based solver can be used to solve Poisson's equation for the electrostatic potential. Only multigrid can be used with the magnetostatic solver, which solves the Poisson equation for each component of the vector potential. Various boundary conditions are supported, and both the FFT and multigrid methods allow internal conductors.

The FFT solvers use standard methods. In Fourier space, solving Poisson's equation is a simple division of the transformed charge density by k^2. The solver assumes pipe-like boundary conditions. Transversely, a zero-Dirichlet boundary is applied using a sine transform, or a zero-Dirichlet with symmetry at the axis using a cosine transform. Along the z-axis, the pipe axis, a periodic transform is done. An alternate formulation does a tridiagonal matrix solve along the z-axis (instead of the FFT) - this is somewhat faster (since the tridiagonal solver scales as n which is faster than the $n \ln n$ scaling of the FFT) and allows Dirichlet boundaries in z. The solvers allow arbitrary Dirichlet boundary conditions, by placing the appropriate image charges near the boundary. Internal boundaries can be enforced using the standard capacity matrix method. The solvers include optional filtering in Fourier space.

The multigrid solvers were implemented to allow arbitrary internal boundary conditions. Warp uses the standard second-order finite-difference stencil, 5 points in 2D and 7 points in 3D - multigrid is used to solve the resulting matrix system. Various versions of multigrid are implemented, including V-cycles, full multigrid, and full approximation multigrid. The V-cycles version is mostly used. For an isolated Poisson solve, full multigrid is faster than only V-cycles, but any advantage is lost when the V-cycles can use the solution from the previous time step as a first guess. The full approximation method is used when the solver directly includes a Boltzmann electron distribution, resulting in a non-linear system of equations. The exterior boundary conditions implemented include Dirichlet, zero-Neumann and periodic, with any of the conditions on each boundary plane. Internal boundary conditions are handled using the methods of Shortley and Weller [21], or of Hewett [36], applying Dirichlet conditions on the surface of conductors. The methods allow subgrid resolution of the location of the conductor, increasing accuracy with minimal computational cost. For grid points that are near the surface of a conductor, the finite difference form of Poisson's equation is modified to include the location of the surface and its potential. The multigrid method easily incorporates this non-uniform, non-symmetric modification of the matrix system. Warp allows a variety of shapes for the conductors, such as cylinders, aperture plates and arbitrary surfaces of revolution (revolving about a line parallel to one of the grid axis). The conductors can be combined, taking the union, intersection or difference of multiple conductors. This is handled at the Python level in an object oriented manner.

Several additional field solvers are available, which solve modified versions of Poisson's equation. The first, as mentioned above, solves Poisson's equation including as a source a Boltzmann distribution of electrons. The electron distribution function is written as $n_e = n_i \exp\left((\phi - \phi_p)/T_e\right)$, where ϕ_p and T_e are the user supplied plasma potential and electron temperature, and ϕ is the potential to be solved. This non-linear Poisson equation is solved using the full approximation multigrid method. This solver is used primarily in the simulation of plasma ion sources, avoiding the need for the costly detailed simulation of the electrons in the plasma. The second solver allows a variable dielectric constant. The user supplies a grid giving the spatial variation of the dielectric. Only 2D planar and axisymmetric versions are implemented. The third solver is used with the implicit PIC method. Warp uses the direct implicit method, with implementations in both 2D and 3D. The full implicit susceptibility is included without approximation beyond the usual linearization.

4.1.2. Electromagnetic Warp's electromagnetic solver is based on the Non-Standard Finite-Difference (NSFD) technique [37, 38], which is an extension of the Finite-Difference Time-Domain technique to larger stencils in the plane perpendicular to the direction of the finite difference. In effect, a finite average (or digital filtering) is performed orthogonally to the direction of the finite difference. The coefficients of the finite average can be set (by the user) to arbitrary values within some bounds and the rule that the sum equals unity for energy conservation. This gives the user some control on the numerical dispersion and Courant time step limits which do depend on those parameters. For a given set of parameters, the stencil reduces to the Yee stencil [22], for which, for cubic cells, the Courant time step multiplied by the speed of light is given by the cell size divided by $\sqrt{3}$, and the numerical dispersion vanishes along the cell 3D diagonals. As shown in [39], for a different given set of parameters, and for cubic cells, the Courant time step multiplied by the speed of light equals the cell size, and the numerical dispersion vanishes along the main axes. More details on the solver implementation and characteristics for several sets of coefficients are available in [23]. Also described in [23] are the implementation of Perfectly Matched Layers for the absorption of waves at grid boundaries and of Friedman's damping algorithm for noise control [40]. In the same paper, it is shown that introducing a stride in the usage of standard linear filtering allows for construction of efficient iterative sideband digital filters that are nonetheless compact, thus well suited for implementation on parallel computers. In the current implementation, internal conducting surfaces are implemented by enforcing a null field within the conductors, and complicated conductor shapes are thus approximated by following grid lines.

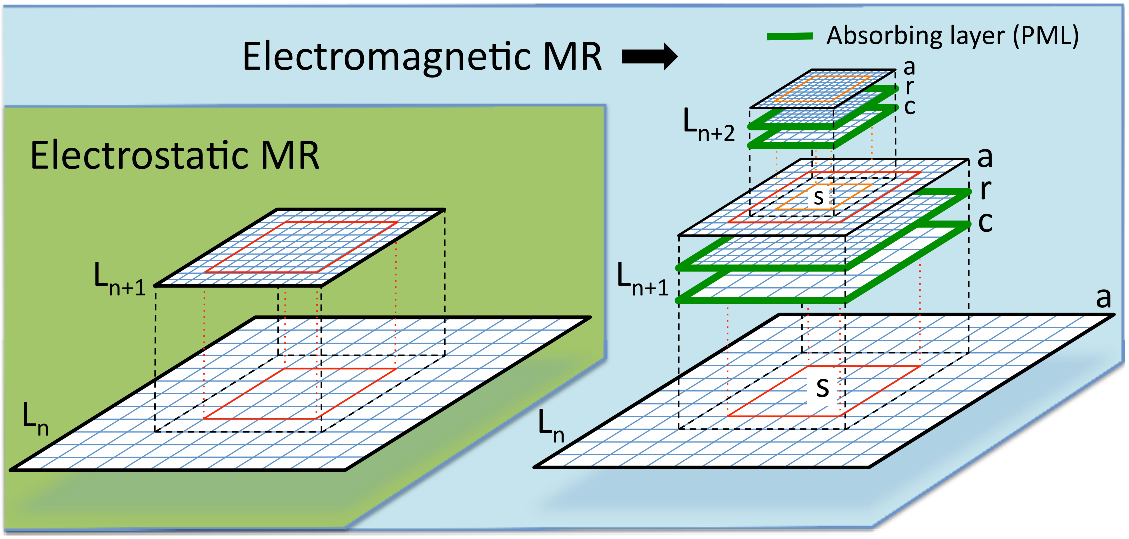


Figure 1. Sketches of the implementation of mesh refinement in Warp with the electrostatic (left) and electromagnetic (right) solvers. In both cases, the charge/current from particles are deposited at the finest levels first, then interpolated recursively to coarser levels. In the electrostatic case, the potential is calculated first at the coarsest level L_0, the solution interpolated to the boundaries of the refined patch r at the next level L_1 and the potential calculated at L_1. The procedure is repeated iteratively up to the highest level. In the electromagnetic case, the fields are computed independently on each grid and patch without interpolation at boundaries. Patches are terminated by absorbing layers (PML) to prevent the reflection of electromagnetic waves. Additional coarse patch c and fine grid a are needed so that the full solution is obtained by substitution on a as $F_{n+1}(a) = F_{n+1}(r) + I[F_n(s) - F_{n+1}(c)]$ where F is the field, and I is a coarse-to-fine interpolation operator. In both cases, the field solution at a given level L_n is unaffected by the solution at higher levels L_{n+1} and up, allowing for mitigation of some spurious effects (see text) by providing a transition zone via extension of the patches by a few cells beyond the desired refined area (red & orange rectangles) in which the field is interpolated onto particles from the coarser parent level only.

4.2. Mesh Refinement

The mesh refinement methods that have been implemented in Warp were developed following the following principles: i) avoidance of spurious effects from mesh refinement, or minimization of such effects; ii) user controllability of the spurious effects' relative magnitude; iii) simplicity of implementation. The two main generic issues that were identified are: a) spurious self-force on macroparticles close to the mesh refinement interface [41, 42]; b) reflection (and possible amplification) of short wavelength electromagnetic waves at the mesh refinement interface [43]. The two effects are due to the loss of translation invariance introduced by the asymmetry of the grid on each side of the mesh refinement interface.

In addition, for some implementations where the field that is computed at a given level is affected by the solution at finer levels, there are cases where the procedure violates the integral of Gauss' Law around the refined patch, leading to long range errors [41, 42]. As will be shown below, in the procedure that has been developed in Warp, the field at a given refinement level is not affected by the solution at finer levels, and is thus not affected by this type of error.

4.2.1. Electrostatic A cornerstone of the Particle-In-Cell method is that assuming a particle lying in a hypothetical infinite grid, then if the grid is regular and symmetrical, and if the order of field gathering matches the order of charge (or current) deposition, then there is no self-force of the particle acting on itself: a) anywhere if using the so-called "momentum conserving" gathering scheme; b) on average within one cell if using the "energy conserving" gathering scheme [29]. A breaking of the regularity and/or symmetry in the grid, whether it is from the use of irregular meshes or mesh refinement, and whether one uses finite difference, finite volume or finite elements, results in a net spurious self-force (which does not average to zero over one cell) for a macroparticle close to the point of irregularity (mesh refinement interface for the current purpose) [41, 42].

A sketch of the implementation of mesh refinement in Warp is given in Figure 1 (left). Given the solution of the electric potential at a refinement level L_n, it is interpolated onto the boundaries of the grid patch(es) at the next refined level L_{n+1}. The electric potential is then computed at level L_{n+1} by solving the Poisson equation. This procedure necessitates the knowledge of the charge density at every level of refinement. For efficiency, the macroparticle charge is deposited on the highest level patch that contains them, and the charge density of each patch is added recursively to lower levels, down to the lowest.

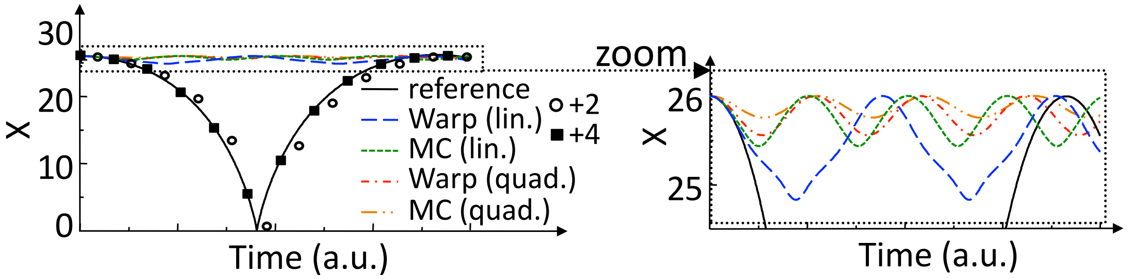


Figure 2. Position history of one charged particle attracted by its image induced by a nearby metallic (dirichlet) boundary. The particle is initialized at rest. Without refinement patch (reference case), the particle is accelerated by its image, is reflected specularly at the wall, then decelerates until it reaches its initial position at rest. If the particle is initialized inside a refinement patch, the particle is initially accelerated toward the wall but is spuriously reflected before it reaches the boundary of the patch whether using the method implemented in Warp or the MC method. Providing a surrounding transition region 2 or 4 cells wide in which the potential is interpolated from the parent coarse solution reduces significantly the effect of the spurious self-force.

The presence of the self-force is illustrated on a simple test case that was introduced in [41] and also used in [42]: a single macroparticle is initialized at rest within a single refinement patch four cells away from the patch refinement boundary. The patch at level L_1 has 32×32 cells and is centered relative to the lowest 64×64 grid at level L_0 ("main grid"), while the macroparticle is centered in one direction but not in the other. The boundaries of the main grid are perfectly conducting, so that the macroparticle is attracted to the closest wall by its image. Specular reflection is applied when the particle reaches the boundary so that the motion is cyclic. The test was performed with Warp using either linear or quadratic interpolation when gathering the main grid solution onto the refined patch boundary. It was also performed using another method based on the algorithm given in [44], which employs a more elaborate procedure involving two-ways interpolations between the main grid and the refined patch. A reference case was also run using a single 128×128 grid with no refined patch, in which it is observed that the particle propagates toward the closest boundary at an accelerated pace, is reflected specularly at the boundary, then slows down until it reaches its initial position at zero velocity. The particle position histories are shown for the various cases in Fig. 2. In all the cases using the refinement patch, the particle was spuriously reflected near the patch boundary and was effectively trapped in the patch. We notice that linear interpolation performs better than quadratic, and that the simple method implemented in Warp performs better than the other proposed method for this test (see discussion below).

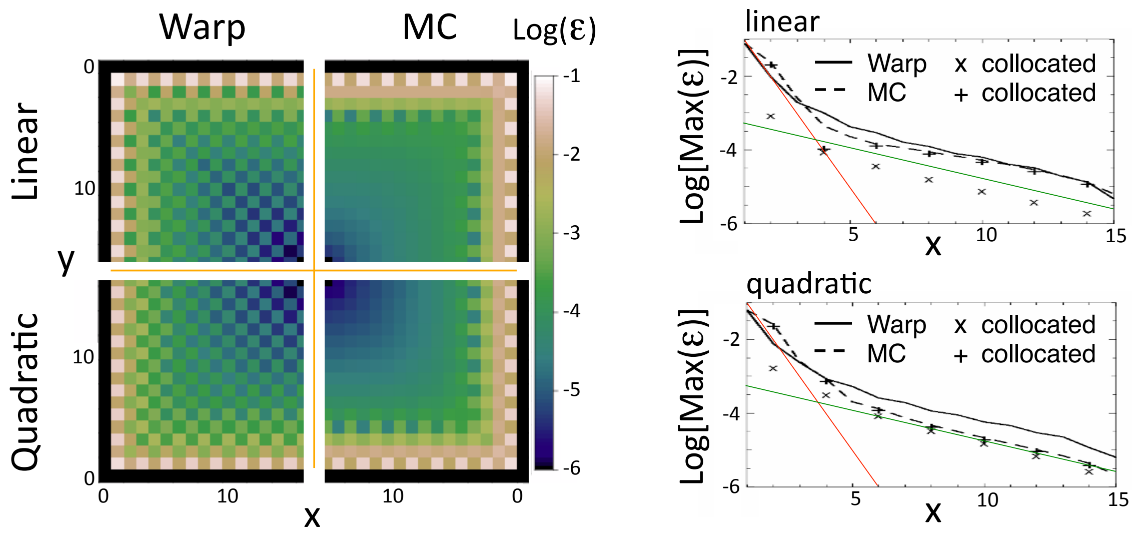


Figure 3. (left) Maps of the magnitude of the spurious self-force ϵ in arbitrary units within one quarter of the refined patch, defined as $\epsilon = \sqrt{(E_x - E_x^{ref})^2 + (E_y - E_y^{ref})^2}$, where E_x and E_y are the electric field components within the patch experienced by one particle at a given location and E_x^{ref} and E_y^{ref} are the electric field from a reference solution. The map is given for the Warp and the MC mesh refinement algorithms and for linear and quadratic interpolation at the patch refinement boundary.
(right) Lineouts of the maximum (taken over neighboring cells) of the spurious self-force. Close to the interface boundary (x=0), the spurious self-force decreases at a rate close to one order of magnitude per cell (red line), then at about one order of magnitude per six cells (green line).

The magnitude of the spurious self-force as a function of the macroparticle position was mapped and is shown in Fig. 3 for the Warp and MC algorithms using linear or quadratic interpolations between grid levels. It is observed that the magnitude of the spurious self-force decreases rapidly with the distance between the particle and the refined patch boundary, at a rate approaching one order of magnitude per cell for the four cells closest to the boundary and about one order of magnitude per six cells beyond. The method implemented in Warp offers a weaker spurious force on average and especially at the cells that are the closest to the coarse-fine interface where it is the largest and thus matters most.

A method was devised for reducing the magnitude of self-force near the coarse-fine boundaries by orders of magnitude for the MC method by using a special deposition procedure near the interface [42]. The Warp method offers a simpler alternative as the method leaves the coarse grid solution free of self-force, within and around the patch, offering the possibility of reducing the effect of the self-force by simply extending the refinement patch by a few "transition" cells beyond the desired "effective" refined area. Within the effective area, the particles gather the potential in the fine grid. In the extra transition cells surrounding the refinement patch, the force is gathered directly from the coarse grid (an option, which has not yet been implemented, would be to interpolate between the coarse and the grid field solution within the transition zone so as to provide continuity of the force experienced by the particles at the interface). The number of cells allocated in the transition zones of patches is controllable by the user in Warp, giving the opportunity to check whether the spurious self-force is affecting the calculation by repeating it using different thicknesses of the transition zones. The control of the spurious force using the transition zone is illustrated in Fig. 2, where the calculation with Warp using linear interpolation at the patch interface was repeated using either two or four cells transition regions (measured in refined patch cell units). Using two extra cells allowed for the particle to be free of spurious trapping within the refined area and follow a trajectory that is close to the reference one, and using four extra cells improved further to the point where the resulting trajectory becomes undistinguishable from the reference one.

Automatic remeshing has been implemented in Warp following the procedure described in [45], refining on criteria based on measures of local charge density magnitude and gradients. AMR Warp simulations were applied to the modeling of the front end injector of the High Current Experiment (HCX) [46], and provided the first numerically converged estimates of phase space beam distorsions, which directly affects beam quality [24]. Fig. 4 shows snapshots from 2D axisymmetric simulation of the souce area illustrating the automatic placement of refined patches, and 3D simulation of the full injector showing the beam generation, acceleration and transport.

4.2.2. Electromagnetic The method that is used for electrostatic mesh refinement is not directly applicable to electromagnetic calculations. As was shown in section 3.4 of [47], refinement schemes relying solely on interpolation between coarse and fine patches

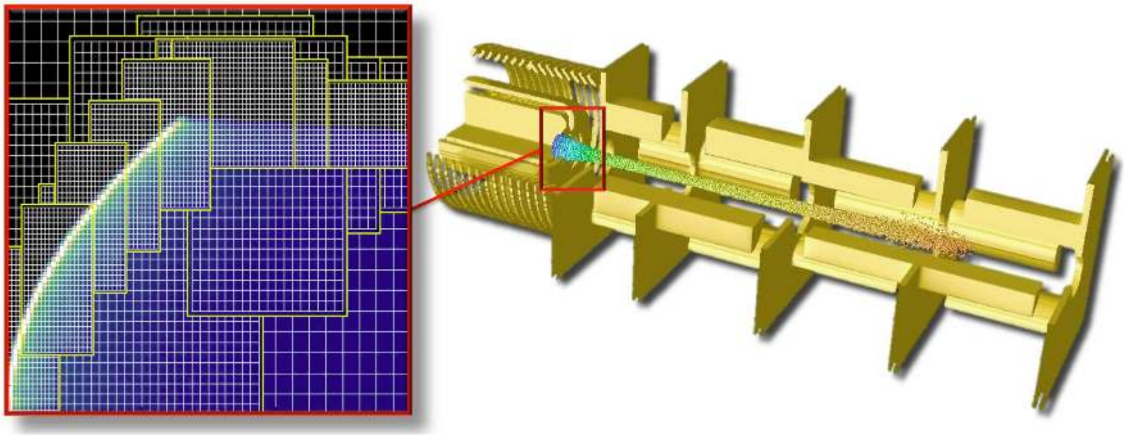


Figure 4. Snapshot from a 3D self-consistent simulation of the injector in the High Current Experiment shows the beam emerging from the source at low energy (blue) and being accelerated (green-yellow-orange) and transported in a four quadrupole front end. The automatic layout of the mesh refinement patches from a 2D axisymmetric simulation of the source area shows 2 levels of refinement, concentrating the finer meshes around the emitter (white curve surface) and the beam edge (dark blue).

lead to the reflection with amplification of the short wavelength modes that fall below the cutoff of the Nyquist frequency of the coarse grid. Unless these modes are damped heavily or prevented from occurring at their source, they may affect particle motion and their effect can escalate if trapped within a patch, via multiple successive reflections with amplification.

To circumvent this issue, an additional coarse patch (with the same resolution as the parent grid) is added, as shown in Fig. 1-right and described in [25]. Both the fine and the coarse grid patches are terminated by Perfectly Matched Layers, reducing wave reflection by orders of magnitude, controllable by the user [48, 49]. The source current resulting from the motion of charged macroparticles within the refined region is accumulated on the fine patch and is then interpolated onto the coarse patch and added onto the parent grid. The process is repeated recursively from the finest level down to the coarsest. The Maxwell equations are then solved for one time interval on the entire set of grids, by default for one time step using the time step of the finest grid. The field on the coarse and fine patches only contain the contributions from the particles that have evolved within the refined area but not from the current sources outside the area. The total contribution of the field from sources within and outside the refined area is obtained by adding the field from the fine grid $F(f)$, and adding an interpolation I of the difference between the relevant subset s of the field in the parent grid $F(s)$ and the field of the coarse grid $F(c)$, on an auxiliary grid a, i.e. $F(a) = F(f) + I[F(s) - F(c)]$. In effect, there is substitution of the coarse field calculated in the patch area by its fine resolution counterpart. The operation is carried recursively starting at the coarsest level up to the finest. An option has been implemented in which various grid levels are pushed with different time steps, given as a fixed fraction of the individual grid

Courant conditions (assuming same cell aspect ratio for all grids and refinement by integer factors). In this case, the fields from the coarse levels, which are advanced less often, are interpolated in time.

The substitution method has two potential drawbacks due to the inexact cancellation between the coarse and fine patches of : (i) the remnants of ghost fixed charges created by the particles entering and leaving the patches (this effect is due to the use of the electromagnetic solver and is different from the spurious self-force that was described for the electrostatic case); (ii) the electromagnetic waves traveling on each patch at slightly different velocity due to numerical dispersion. The first issue results in an effective spurious multipole field whose magnitude decreases very rapidly with the distance to the patch boundary, similarly to the spurious self-force in the electrostatic case. Hence, adding a few extra transition cells surrounding the patches mitigates this effect very effectively. The tunability of Warp's electromagnetic solver provides the means to optimize the numerical dispersion so as to minimize the second effect for a given application, which has been demonstrated on the laser-plasma interaction test case presented in [25]. Both effects and their mitigation are described in more detail in [25].

As a test to the electromagnetic PIC implementation, Warp simulations of wave excitations by a beam propagating through plasma, as described in [50], were conducted. In these simulations, a hard-edged, elliptical, rigid beam propagates at constant velocity $v_z = 0.5c$ where c is the speed of light through an initially cold neutral plasma of initial density n_0. The beam has a flat-top density profile of $n_b = n_0/2$, and an elliptical shape of length $l = 15c/\omega_p$ and diameter $d = l/10$, where ω_p is the electron plasma frequency. It is shown in [50] that waves with a wavenumber of approximately $2\omega_p/v_z$ are generated in the plasma by the beam's electrostatic field, and have larger amplitude inside the beam, due to their interaction with the beam's sharp edges.

Resolving the beam edge and the small structures developing in the wake inside the beam forces small cell sizes. The resolution that is needed for macroscopic convergence was explored in 2-1/2D in a series of four runs where the number of grid cells was varied from 64×160 to 512×1280 by incremental factors of 2. Third order spline interpolation was used for the beam and plasma macroparticle current deposition and force gathering. The details of the plasma wake were very similar between the two highest resolution cases, indicating that macroscopic convergence was reached. The results from the runs using 128×320 and 512×1280 grids are shown in Fig. 5. The result from the highest resolution run serves as the reference for subsequent calculations with mesh refinement.

A run was conducted where the main grid had 128×320 cells and was complemented by two refinement patches (with successive refinement factors of 2 in each direction), such that the resolution in the central patch matched the resolution of the case of reference. The number and weight of the injected plasma macroparticles was varied, such that the number of macroparticles per cell in each grid at injection was constant. Results are plotted in Fig. 5 (bottom-left) showing a good reproduction of the fine scale structures within the central fine patch in good agreement with the reference case.

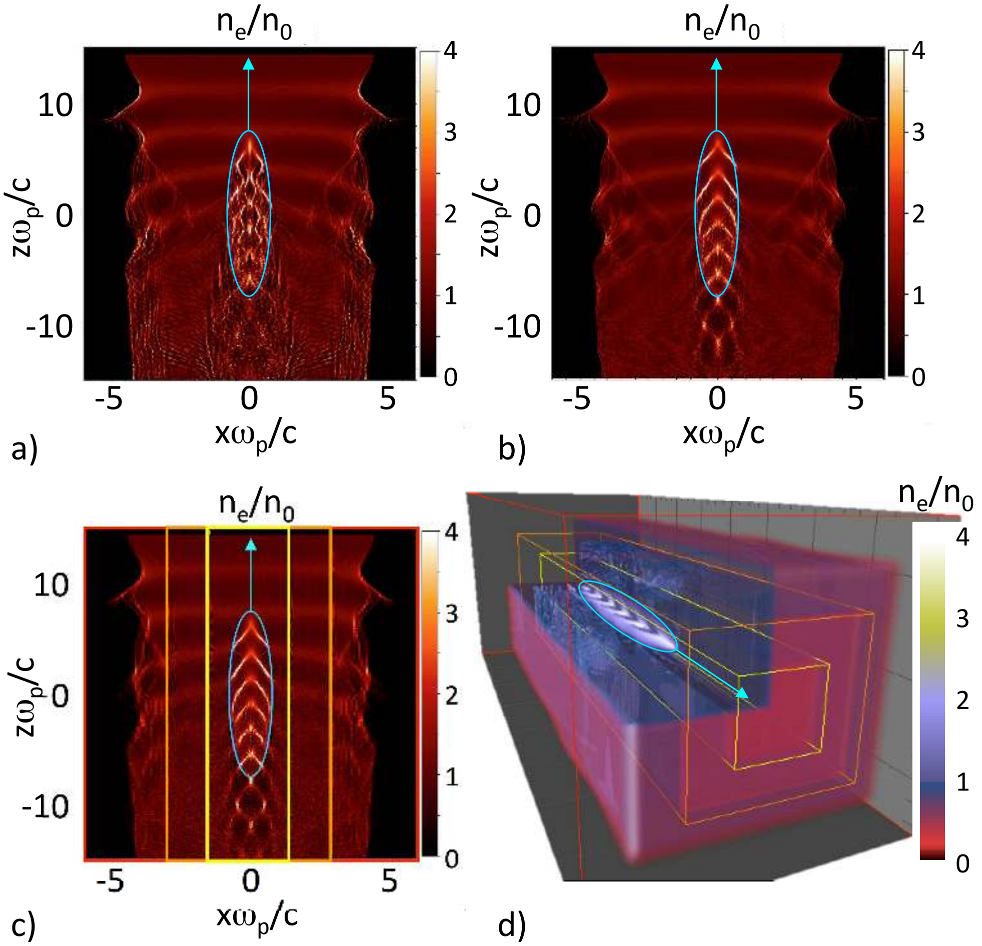
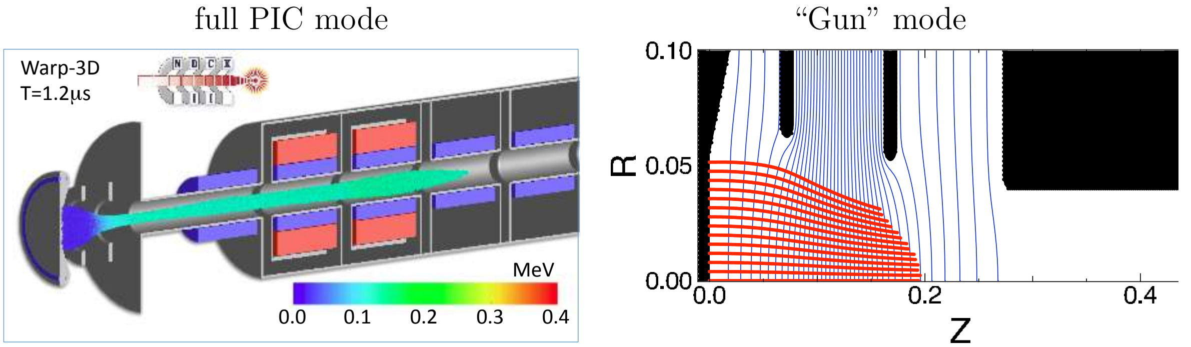


Figure 5. Electron density n_e (normalized to the density of the injected plasma) from Warp simulations in 2-1/2D for a), b), c) and 3D for d) of a rigid beam (thin light-blue outline) propagating through a neutral plasma, for grid sizes of a) 128×320, b) 512×1280, c) 128×320 (main grid, red box) + 128×640 (patch 1, orange box) + 128×1280 (patch 2, yellow box), such that the resolution of patch 2 matched the resolution of the grid used for b), d) grid size of $64 \times 64 \times 160$ (main grid, red box) + $64 \times 64 \times 320$ (patch 1, orange box) + $64 \times 64 \times 640$ (patch 2, yellow box). For c) and d), the number and weight of injected plasma macroparticles was adjusted to keep the number of macroparticles per cell constant in each grid at injection in front of the beam.

Lastly, a three-dimensional simulation with mesh refinement of the same physical setup was conducted. The grid setup and 3D isosurfaces of the plasma electron density as the beam enters the plasma are shown in Fig. 5 (bottom-right). As expected, structures similar to the ones observed in 2D are present within the beam envelope. The speedup achieved by the use of mesh refinement was estimated to be approximately one order of magnitude in 3D.

5. Modes of operation

Thanks to a high degree of modularity and the usage of the Python language which allows for dynamic loading of functionalities, Warp can be used in various configurations: “standard Particle-In-Cell”, “transverse-slice”, “Gun mode”, “build-up” and “quasistatic”.


full PIC mode

“Gun” mode

Figure 6. (left) Rendering from a full 3D self-consistent time-dependent simulation of NDCX-II, using the injector voltage configuration computed in gun mode, (right) snapshot from calculation of steady flow of charged particles emitted from a hot plate in the NDCX-II injector, using the gun mode. The equipotentials (blue lines) were computed using the charge distribution of the previous iteration. Particle trajectories are computed and accumulated (red) for the calculation of the next electrostatic potential. Voltage is interactively adjusted on the electrodes (black) until an acceptable solution is found.

5.1. Standard Particle-In-Cell

In the default “standard Particle-In-Cell” mode, Warp follows a collection of charged macroparticles evolving under the influence of their self-field and externally applied fields in 3D (xyz), 2D axisymmetric (rz) or 2D planar (xy or xz). In the case of the modeling of charged particle beams (the default), z is the direction of (initial) beam propagation and a moving window is used to follow the beam as it propagates. A special warped-coordinate particle advance is used to treat particles in bends [16] and in that case, z maps to s, the path length coordinate on the nominal machine center line, as commonly defined in the accelerator community (but s is not a reference orbit). Independently of the dimensionality of the self-fields, the particles are advanced in 3D and have 3D positions and velocities. For reduced dimensional models, the appropriate integrations and projections are done for the charge deposition and field gather. With the axisymmetric model, advancing the particles in 3D avoids problems on axis. With particles in 3D, the diagnostics are consistent among the different models and can be directly compared.

5.2. Transverse slice

The transverse slice model is used to simulate beams approximated as having infinite extent along the z-axis, i.e. effectively this is a steady-flow model. The particles are all initialized at a common z location, and each time step are advanced to the next z location, $z + dz$. Each particle will have its own time step, dependent on its z-velocity. In the warped coordinates, the particles are advanced to the next angle around the bend. The particle time step sizes will thus also be dependent on the radial position. Since the particle's z-velocity can change during a time step, iterations are done correcting the time step sizes so that all particles are advanced to the same value of s. In the charge deposition, the weight of the particles is optionally scaled by v_b/v_p, the ratio of the beam velocity and the particle's velocity, so that each particle represents a unit of current (rather than line-charge density). Also optionally, the fields from the previous and current time step can be used to calculate a self E_z field using a backward finite difference. This self E_z can become important if there are significant variations in transverse distribution along the beam.

5.3. "Gun" mode

The gun mode is used to model continuous steady flow of charged macroparticles that are at equilibrium with the externally applied fields, in non-paraxial situations where an accurate self longitudinal electric field is important and the slice model does not suffice. This mode is used to design charged particle injectors. In this mode, particles are injected on one time step only and the injection is turned off. Those particles are then tracked through the system until there is no particle left. On each time step, the charge density from the particles is accumulated. After all of the particles leave the system, the new field is calculated using the accumulated charge density. A selection of particles is saved each time step for diagnostic and plotting. This procedure is iterated until convergence to a steady state solution. Several non-exclusive optional procedures are available to speedup the convergence and reduce jitter: (i) a running averaging of the charge density is performed from one iteration to the next, (ii) the particles are tracked initially through a small fraction of the system which increases progressively after each iteration until it covers the entire system, (iii) solutions are computed using increasingly high resolution and particle statistics. Figure 6 shows snapshots from a full 3D time-dependent simulation of the NDCX-II front end and from a "gun" steady flow calculation of the NDCX-II injector.

5.4. Lorentz boosted frame

A method was recently proposed to speed up full PIC simulations of a certain class of relativistic interactions by performing the calculation in a Lorentz boosted frame [51], taking advantage of the properties of space/time contraction and dilation of special relativity to render space and time scales (that are separated by orders of magnitude in

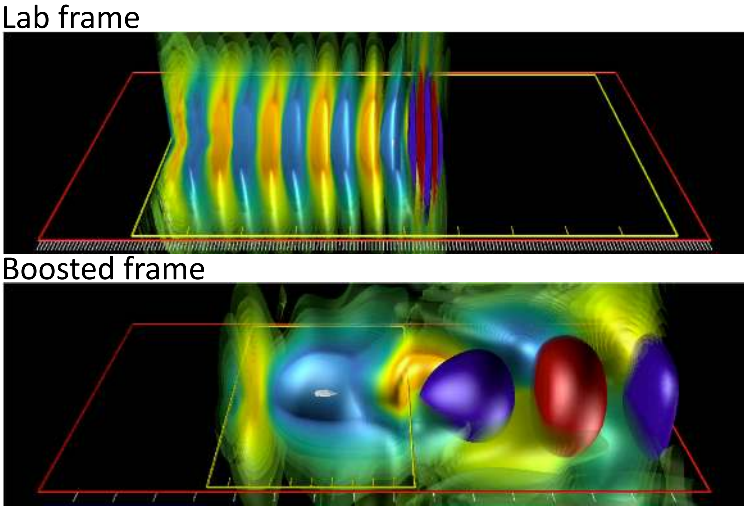


Figure 7. Warp simulations of scaled laser plasma acceleration stages: (top) in the lab; (bottom) in a Lorentz boosted frame (laser pulse in blue/red; plasma wakefield in pale blue/yellow). For a single 10 GeV acceleration stage, the simulation in the boosted frame is more than 10,000 times faster than the simulation in the laboratory frame.

the laboratory frame) commensurate in a Lorentz boosted frame, resulting in far fewer computer operations. The method has been applied successfully to the modeling of laser plasma acceleration [52, 53, 54, 13], electron cloud effects [17], free electron lasers [55], coherent synchrotron radiation [12], and production of ultrabright attosecond x-ray pulses [56].

In a laser plasma accelerator, a laser pulse is injected through a plasma, creating a wake of regions with very strong electric fields of alternating polarity [57]. An electron beam that is injected with the appropriate phase can thus be accelerated to high energy in a distance that is much shorter than with conventional acceleration techniques [58]. The simulation of a laser plasma acceleration stage from first principles using the Particle-In-Cell technique in the laboratory frame is very demanding computationally, as the evolution of micron-scale long laser oscillations needs to be followed over millions of time steps as the laser pulse propagates through a meter long plasma for a 10 GeV stage. As illustrated in Fig. 7 showing snapshots from simulations of a downscaled LPA stage, in the laboratory frame the laser pulse is much shorter than the wake, whose wavelength is also much shorter than the acceleration distance ($\lambda_{laser} \ll \lambda_{wake} \ll \lambda_{acceleration}$). In a Lorentz boosted frame moving at a speed near the speed of light with the laser in the plasma, the laser will be Lorentz expanded (by a factor $(1 + v_f/c)\gamma_f$ where $\gamma_f = (1 - v_f^2/c^2)^{-1/2}$ and v_f is the velocity of the frame and c is the speed of light). The plasma (now moving opposite to the incoming laser at velocity $-v_f$) is Lorentz contracted (by a factor γ_f). In a boosted frame moving with the wake ($\gamma_f \approx \gamma_{wake}$), the laser wavelength, the wake and the acceleration length are now commensurate ($\lambda_{laser} < \lambda_{wake} \approx \lambda_{acceleration}$), leading to far fewer time steps by a factor $(1 + v_f/c)^2\gamma_f^2$, hence computer operations [51, 13].

A speedup of up to a million times was reported for Warp modeling of a hypothetical 1 TeV stage [59]. Control of a violent numerical instability that had been plaguing early attempts was obtained via the combination of: (i) the use of Warp's tunable electromagnetic solver and efficient wideband filtering [23], (ii) observation of the benefits of hyperbolic rotation of space-time on the laser spectrum in boosted frame simulations [59], and (iii) identification of a special time step at which the growth rate of the instability is greatly reduced [23]. In addition, a novel numerical method for injecting the laser pulse through a moving planar antenna was introduced in Warp [13].

5.5. Build-up

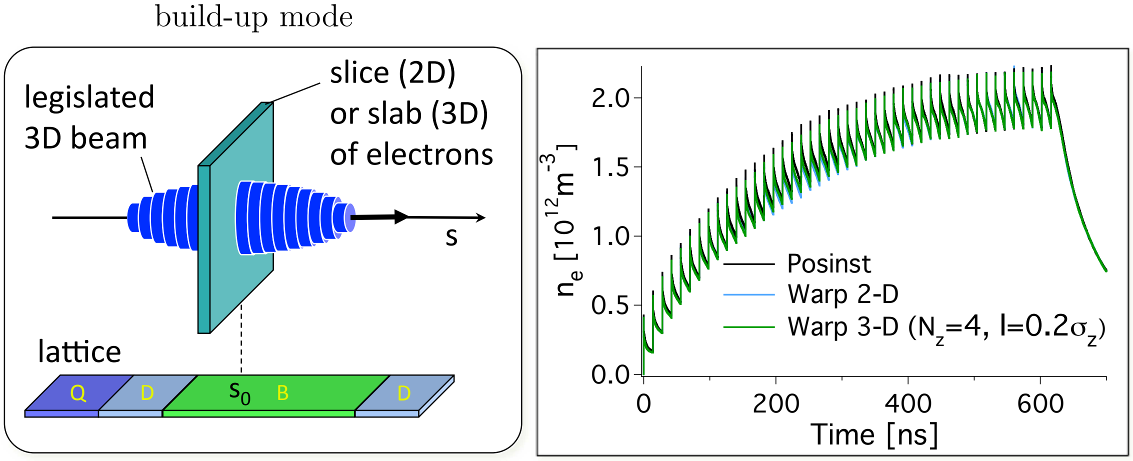


Figure 8. (left) Sketch of the build-up mode. The dynamics of electrons is followed for a thin (2D) or thick (3D) slice located at a given location in the lattice, under the influence of a legislated particle beam passing through the slice.; (right) Electron density versus time from a benchmark of Warp versus Posinst in the simulation of the build-up of electron cloud in a magnetic dipole section.

The build-up mode is used to study the accumulation of electrons at a given location in a particle accelerator. In this mode, the dynamics of electrons is followed for a thin (2D) or thick (3D) slice located at a given location in the lattice, under the influence of a legislated particle beam passing through the slice (Fig. 8-left). The electrons are described by a collection of macro-particles evolving under the influence of their own space charge, plus the field of an external beam, following the standard Particle-In-Cell (PIC) technique. The electron electric field is obtained in the static approximation from solving the Poisson equation. The field from the external (positively charged) beam is either prescribed analytically (using the Bassetti-Erskine formula [60]) or given from solving the Poisson equation over a prescribed charge distribution. The build-up mode has been successfully benchmarked against the 2D build-up code Posinst [18, 19]. An example of electron density history from Warp build-up simulations is contrasted with Posinst results in Fig. 8-(right).

5.6. *Quasistatic*

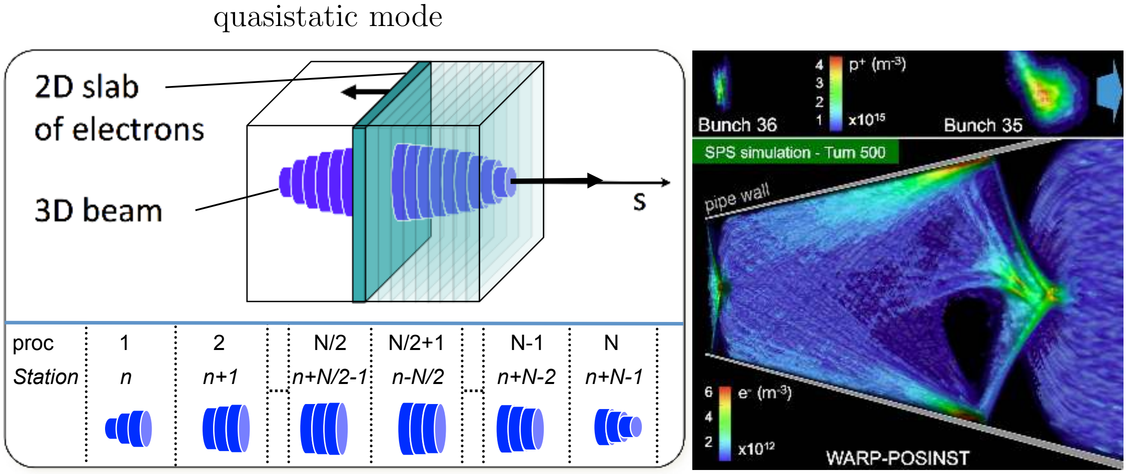


Figure 9. (left) Sketch of the quasistatic mode. A 2D slab of electron macroparticles is stepped backward (with small time steps) through the beam field. The 2D electron fields (solved at each step) are stacked in a 3D array, that is used to give a kick to the beam. Finally, the beam particles are pushed forward (with larger time steps) to the next station of electrons. In the quasistatic mode, the beam is distributed among n slices, that are uniformly spread among N processors. Using a pipelining algorithm, slices on a given processor are pushed from one station to the next, while the slices of the previous processor will be pushed to the same station one time step later; (right) Warp-Posinst simulation of two consecutive bunches (top) interacting with an electron cloud (bottom) – the bunches and electron clouds have been separated for clarity.

The quasistatic mode is used to model the interaction of relativistic beams with electron clouds in particle accelerators, taking advantage of the separation of space and time scales between the beam particles and the electron cloud dynamics [61]. In this mode, a 2D slab of electron macroparticles is stepped backward (with small time steps) through the beam field (see Fig. 9-left). The 2D electron fields (solved at each step) are stacked in a 3D array, that is used to give a kick to the beam. Finally, the beam particles are pushed forward (with larger time steps) to the next station of electrons, using either maps or a Leap-Frog pusher.

The parallelization in the transverse direction (perpendicular to s) uses domain decomposition of the particles and fields. The parallelization in the longitudinal direction (along s) uses pipelining, similarly (but not identical) to the strategy developed in QuickPIC [62]. Assuming that the beam is distributed among n slices of equal thickness along the longitudinal dimension, and that N processors are used for a run, n/N consecutive slices are assigned to each processor, as sketched in Fig. 9-left. During the first iteration, the electron distribution from the first station in the ring is evolved through the slices of processor N while processors 1 through $N - 1$ stay idle. The electron distribution is then passed to processor $N - 1$ and evolves through the slices that it contains, while processor N pushes the beam to station 2 and starts evolving the corresponding distribution of electrons. After N steps, all processors are active and the procedure is repeated until the beam slices on processor 1 reach the desired propagation distance.

Posinst routines for electron cloud generation and build-up are accessible through the Warp package in all modes including quasistatic, allowing for three-dimensional fully self-consistent simulations of the electron cloud build-up and its effect on the beam dynamics simultaneously, therefore including the memory of electron clouds between bunches, hence multi-bunch effects. Such simulations have been performed for 1000 turns of up to three trains of 72 bunches each circling in the CERN SPS ring [11]. For moderately high resolution and statistics, 8 CPUs were used per RF bucket, for a total of 5-10 hours runs using 11,520 CPUs on Franklin at NERSC. A colored three-dimensional rendering of two consecutive bunches and the interacting electron cloud is shown in Fig. 9-(right). More details are available in [11].

6. Summary and outlook

The Warp code-framework has recently been augmented with various novel methods including PIC with adaptive mesh refinement, a large-timestep mover for particles of arbitrary magnetized species, a new relativistic Lorentz invariant leapfrog particle pusher, simulations in Lorentz boosted frames, an electromagnetic solver with tunable numerical dispersion and efficient stride-based digital filtering. With its new capabilities and thanks to a design that allows for a high degree of versatility, the range of application of Warp has considerably widened far beyond the initial application to the Heavy Ion Fusion Science program.

Further developments are underway. Notably, a full implementation of AMR-PIC requires adaptive time-stepping for the particles as they cross different patches, similarly to the algorithm for multi-scale Particle-in-Cell plasma simulations proposed in [63]. An embryo of such a capability has been implemented in Warp based on sub-cycling methods derived from [64], where particles are sorted in groups that are advanced with different time steps. Progress will be reported as the implementation gets finalized.

7. Acknowledgments

The authors thank M. Furman, C. Geddes, and Marco Venturini for valuable comments. Work supported by US-DOE Contracts DE-AC02-05CH11231 and DE-AC52-07NA27344, the US-DOE SciDAC program ComPASS and the US-LHC Accelerator Research Program (LARP). Used resources of NERSC, supported by US-DOE Contract DE-AC02-05CH11231.

References

[1] D. Grote, A. Friedman, J.-L. Vay, I. Haber, The warp code: modeling high intensity ion beams, in: AIP Conference Proceedings, no. 749, 2005, pp. 55–8.

[2] B. Logan, F. Bieniosek, C. Celata, J. Coleman, W. Greenway, E. Henestroza, J. Kwan, E. Lee, M. Leitner, P. Roy, P. Seidl, J.-L. Vay, W. Waldron, S. Yu, J. Barnard, R. Cohen, A. Friedman, D. Grote, M. K. Covo, A. Molvik, S. Lund, W. Meier, W. Sharp, R. Davidson, P. Efthimion, E. Gilson, L. Grisham, I. Kaganovich, H. Qin, A. Sefkow, E. Startsev, D. Welch, C. Olson, Recent us advances in ion-beam-driven high energy density physics and heavy ion fusion, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 577 (12) (2007) 1 – 7, ¡ce:title¿Proceedings of the 16th International Symposium on Heavy Ion Inertial Fusion¡/ce:title¿ ¡ce:subtitle¿HIF 06¡/ce:subtitle¿. doi:10.1016/j.nima.2007.02.070.
URL http://www.sciencedirect.com/science/article/pii/S0168900207002847

[3] K. Gomberoff, J. Fajans, A. Friedman, D. Grote, J.-L. Vay, J. S. Wurtele, Simulations of plasma confinement in an antihydrogen trap, PHYSICS OF PLASMAS 14 (10). doi:10.1063/1.2778420.

[4] E. P. Gilson, R. C. Davidson, M. Dorf, P. C. Efthimion, R. Majeski, M. Chung, M. S. Gutierrez, A. N. Kabcenell, Studies of emittance growth and halo particle production in intense charged particle beams using the Paul Trap Simulator Experiment, PHYSICS OF PLASMAS 17 (5), 51st Annual Meeting of the Division-of-Plasma-Physics of the American-Physics-Society, Atlanta, GA, NOV 02-06, 2009. doi:10.1063/1.3354109.

[5] S. Ohtsubo, M. Fujioka, H. Higaki, K. Ito, H. Okamoto, H. Sugimoto, S. M. Lund, Experimental study of coherent betatron resonances with a Paul trap, PHYSICAL REVIEW SPECIAL TOPICS-ACCELERATORS AND BEAMS 13 (4). doi:10.1103/PhysRevSTAB.13.044201.

[6] P. Folegati, J. Xu, M. H. Weber, K. G. Lynn, Positron storage in micro-traps with long aspect ratio: results of computer simulations, Journal of Physics: Conference Series 262 (1) (2011) 012021.
URL http://stacks.iop.org/1742-6596/262/i=1/a=012021

[7] I. Haber, S. Bernal, B. Beaudoin, A. Cornacchia, D. Feldman, R. B. Feldman, R. Fiorito, K. Fiuza, T. F. Godlove, R. A. Kishek, P. G. O'Shea, B. Quinn, C. Papadopoulos, M. Reiser, D. Stratakis, D. Sutter, J. C. T. Thangaraj, K. Tian, M. Walter, C. Wu, Scaled electron studies at the University of Maryland, NUCLEAR INSTRUMENTS & METHODS IN PHYSICS RESEARCH SECTION A-ACCELERATORS SPECTROMETERS DETECTORS AND ASSOCIATED EQUIPMENT 606 (1-2) (2009) 64–68, 17th International Symposium on Heavy Ion Inertial Fusion, Tokyo, JAPAN, AUG 04-08, 2008. doi:10.1016/j.nima.2009.03.220.

[8] D. Winklehner, D. Todd, J. Benitez, M. Strohmeier, D. Grote, D. Leitner, Comparison of extraction and beam transport simulations with emittance measurements from the ECR ion source venus, JOURNAL OF INSTRUMENTATION 5. doi:10.1088/1748-0221/5/12/P12001.

[9] F. Nürnberg, A. Friedman, D. P. Grote, K. Harres, B. G. Logan, M. Schollmeier, M. Roth, Warp simulations for capture and control of laser-accelerated proton beams, Journal of Physics: Conference Series 244 (2) (2010) 022052.
URL http://stacks.iop.org/1742-6596/244/i=2/a=022052

[10] R. A. Kishek, Ping-pong modes: A new form of multipactor, Phys. Rev. Lett. 108 (2012) 035003. doi:10.1103/PhysRevLett.108.035003.
URL http://link.aps.org/doi/10.1103/PhysRevLett.108.035003

[11] J.-L. Vay, M. A. Furman, M. Venturini, Direct numerical modeling of e-cloud driven instability of a bunch train in the cern sps, in: Proc. Particle Accelerator Conference, New-York, NY, USA, 2011, wEP154.
URL http://accelconf.web.cern.ch/AccelConf/PAC2011/papers/wep154.pdf

[12] W. Fawley, J.-L. Vay, Full electromagnetic simulation of coherent synchrotron radiation via the lorentz-boosted frame approach, in: Proc. IPAC 2010, paper TUPEC064, Kyoto, Japan, 2010.

[13] J. L. Vay, C. G. R. Geddes, E. Esarey, C. B. Schroeder, W. P. Leemans, E. Cormier-Michel, D. P. Grote, Modeling of 10 GeV-1 TeV laser-plasma accelerators using Lorentz boosted simulations, PHYSICS OF PLASMAS 18 (12). doi:10.1063/1.3663841.

[14] A. Friedman, J. Barnard, R. Cohen, D. Grote, S. Lund, W. Sharp, A. Faltens, E. Henestroza, J.-Y.

Jung, J. Kwan, E. Lee, M. Leitner, B. Logan, J.-L. Vay, W. Waldron, R. Davidson, M. Dorf, E. Gilson, I. Kaganovich, Beam dynamics of the neutralized drift compression experiment-ii, a novel pulse-compressing ion accelerator, Physics of Plasmas 17 (5) (2010) 056704 (9 pp.).

[15] Http://hifweb.lbl.gov/Forthon.

[16] A. FRIEDMAN, D. GROTE, I. HABER, 3-DIMENSIONAL PARTICLE SIMULATION OF HEAVY-ION FUSION BEAMS, PHYSICS OF FLUIDS B-PLASMA PHYSICS 4 (7, Part 2) (1992) 2203–2210, 33RD ANNUAL MEETING OF THE DIVISION OF PLASMA PHYSICS OF THE AMERICAN PHYSICAL SOC, TAMPA, FL, NOV 04-08, 1991. doi:10.1063/1.860024.

[17] J. L. Vay, Simulation of beams or plasmas crossing at relativistic velocity, Physics of Plasmas 15 (5) (2008) 056701. doi:10.1063/1.2837054.

[18] M. Furman, M. Pivi, in: LBNL-49771/CBP Note-4151, Berkeley, CA, USA, 2002.

[19] M. Furman, M. Pivi, Probabilistic model for the simulation of secondary electron emission, PHYSICAL REVIEW SPECIAL TOPICS-ACCELERATORS AND BEAMS 5 (12). doi:10.1103/PhysRevSTAB.5.124404.

[20] S. M. Lund, T. Kikuchi, R. C. Davidson, Generation of initial kinetic distributions for simulation of long-pulse charged particle beams with high space-charge intensity, PHYSICAL REVIEW SPECIAL TOPICS-ACCELERATORS AND BEAMS 12 (11). doi:10.1103/PhysRevSTAB.12.114801.

[21] G. Shortley, R. Weller, The numerical solution of Laplace's equation, JOURNAL OF APPLIED PHYSICS 9 (5) (1938) 334–348. doi:10.1063/1.1710426.

[22] K. YEE, Numerical solution of initial boundary value problems involving maxwells equations in isotropic media, IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION AP14 (3) (1966) 302–&.

[23] J. L. Vay, C. G. R. Geddes, E. Cormier-Michel, D. P. Grote, Numerical methods for instability mitigation in the modeling of laser wakefield accelerators in a lorentz-boosted frame, Journal of Computational Physics 230 (15) (2011) 5908–5929. doi:10.1016/j.jcp.2011.04.003.

[24] J.-L. Vay, P. Colella, J. Kwan, P. McCorquodale, D. Serafini, A. Friedman, D. Grote, G. Westenskow, J.-C. Adam, A. Heron, I. Haber, Application of adaptive mesh refinement to particle-in-cell simulations of plasmas and beams, PHYSICS OF PLASMAS 11 (5) (2004) 2928–2934. doi:10.1063/1.1689669.

[25] J.-L. Vay, J.-C. Adam, A. Heron, Asymmetric pml for the absorption of waves. application to mesh refinement in electromagnetic particle-in-cell plasma simulations, COMPUTER PHYSICS COMMUNICATIONS 164 (1-3) (2004) 171–177. doi:10.1016/j.cpc.2004.06.026.

[26] Http://hifweb.lbl.gov/public/software/gist/pygist_html/pygist.html.

[27] Http://hifweb.lbl.gov/Opyndx.

[28] J. BORIS, R. LEE, Nonphysical self forces in some electromagnetic plasma-simulation algorithms, JOURNAL OF COMPUTATIONAL PHYSICS 12 (1) (1973) 131–136.

[29] C. Birdsall, A. Langdon, Plasma physics via computer simulation, Adam-Hilger, 1991.

[30] S. PARKER, C. BIRDSALL, NUMERICAL ERROR IN ELECTRON ORBITS WITH LARGE OMEGA-CE DELTA-T, JOURNAL OF COMPUTATIONAL PHYSICS 97 (1) (1991) 91–102. doi:10.1016/0021-9991(91)90040-R.

[31] R. Cohen, A. Friedman, M. Covo, S. Lund, A. Molvik, F. Bieniosek, P. Seidl, J. Vay, P. Stoltz, S. Veitzer, Simulating electron clouds in heavy-ion accelerators, PHYSICS OF PLASMAS 12 (5), 46th Annual Meeting of the Division of Plasma Physics of the American-Physical-Society, Savannah, GA, NOV 15-19, 2004. doi:10.1063/1.1882292.

[32] R. H. Cohen, A. Friedman, D. P. Grote, J. L. Vay, An implicit "drift-Lorentz" mover for plasma and beam simulations, NUCLEAR INSTRUMENTS & METHODS IN PHYSICS RESEARCH SECTION A-ACCELERATORS SPECTROMETERS DETECTORS AND ASSOCIATED EQUIPMENT 606 (1-2) (2009) 53–55, 17th International Symposium on Heavy Ion Inertial Fusion, Tokyo, JAPAN, AUG 04-08, 2008. doi:10.1016/j.nima.2009.03.083.

[33] A. W. Molvik, M. K. Covo, R. Cohen, A. Friedman, S. M. Lund, W. Sharp, J.-L. Vay, D. Baca,

F. Bieniosek, C. Leister, P. Seidl, Quantitative experiments with electrons in a positively charged beam, PHYSICS OF PLASMAS 14 (5), 48th Annual Meeting of the Division of Plasma Physics of the APS, Philadelphia, PA, JAN 30-NOV 03, 2006. doi:10.1063/1.2436850.

[34] T. C. Genoni, R. E. Clark, D. R. van Welch, A fast implicit algorithm for highly magnetized charged particle motion, The Open Plasma Physics Journal 3 (2010) 36.

[35] L. Sironi, A. Spitkovsky, private Communication (2011).

[36] D. W., Hewett, The embedded curved boundary method for orthogonal simulation meshes, Journal of Computational Physics 138 (2) (1997) 585 – 616. doi:10.1006/jcph.1997.5835.
URL `http://www.sciencedirect.com/science/article/pii/S0021999197958356`

[37] J. Cole, A high-accuracy realization of the yee algorithm using non-standard finite differences, IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES 45 (6) (1997) 991–996.

[38] J. Cole, High-accuracy yee algorithm based on nonstandard finite differences: New developments and verifications, IEEE TRANSACTIONS ON ANTENNAS AND PROPAGATION 50 (9) (2002) 1185–1191. doi:10.1109/TAP.2002.801268.

[39] M. Karkkainen, E. Gjonaj, T. Lau, T. Weiland, Low-dispersionwake field calculation tools, in: Proc. of International Computational Accelerator Physics Conference, Chamonix, France, 2006, pp. 35–40.

[40] A. FRIEDMAN, A 2nd-order implicit particle mover with adjustable damping, JOURNAL OF COMPUTATIONAL PHYSICS 90 (2) (1990) 292–312.

[41] J. Vay, P. Colella, P. McCorquodale, B. Van Straalen, A. Friedman, D. Grote, Mesh refinement for particle-in-cell plasma simulations: Applications to and benefits for heavy ion fusion, LASER AND PARTICLE BEAMS 20 (4) (2002) 569–575. doi:10.1017/S0263034602204139.

[42] P. Colella, P. C. Norgaard, Controlling self-force errors at refinement boundaries for AMR-PIC, JOURNAL OF COMPUTATIONAL PHYSICS 229 (4) (2010) 947–957. doi:10.1016/j.jcp.2009.07.004.

[43] J.-L. Vay, A new absorbing layer boundary condition for the wave equation, JOURNAL OF COMPUTATIONAL PHYSICS 165 (2) (2000) 511–521.

[44] P. McCorquodale, P. Colella, D. Grote, J. Vay, A node-centered local refinement algorithm for Poisson's equation in complex geometries, JOURNAL OF COMPUTATIONAL PHYSICS 201 (1) (2004) 34–60. doi:10.1016/j.jcp.2004.04.022.

[45] J. Vay, A. Friedman, D. Grote, Application of adaptive mesh refinement to PIC simulations in heavy ion fusion, NUCLEAR INSTRUMENTS & METHODS IN PHYSICS RESEARCH SECTION A-ACCELERATORS SPECTROMETERS DETECTORS AND ASSOCIATED EQUIPMENT 544 (1-2) (2005) 347–352, 15th International Symposium on Heavy Ion Inertial Fusion, Princeton, NJ, JUN 07-11, 2004. doi:10.1016/j.nima.2005.01.232.

[46] L. Prost, P. Seidl, F. Bieniosek, C. Celata, A. Faltens, D. Baca, E. Henestroza, J. Kwan, M. Leitner, W. Waldron, R. Cohen, A. Friedman, D. Grote, S. Lund, A. Molvik, E. Morse, High current transport experiment for heavy ion inertial fusion, PHYSICAL REVIEW SPECIAL TOPICS-ACCELERATORS AND BEAMS 8 (2). doi:10.1103/PhysRevSTAB.8.020101.

[47] J.-L. Vay, An extended fdtd scheme for the wave equation: Application to multiscale electromagnetic simulation, JOURNAL OF COMPUTATIONAL PHYSICS 167 (1) (2001) 72–98.

[48] J. Berenger, Three-dimensional perfectly matched layer for the absorption of electromagnetic waves, JOURNAL OF COMPUTATIONAL PHYSICS 127 (2) (1996) 363–379.

[49] J.-L. Vay, Asymmetric perfectly matched layer for the absorption of waves, JOURNAL OF COMPUTATIONAL PHYSICS 183 (2) (2002) 367–399. doi:10.1006/jcph.2002.7175.

[50] I. Kaganovich, E. Startsev, R. Davidson, Nonlinear plasma waves excitation by intense ion beams in background plasma, PHYSICS OF PLASMAS 11 (7) (2004) 3546–3552. doi:10.1063/1.1758945.

[51] J.-L. Vay, Noninvariance of space- and time-scale ranges under a lorentz transformation and the implications for the study of relativistic interactions, Physical Review Letters 98 (13) (2007)

130405/1–4.

[52] D. Bruhwiler, J. Cary, B. Cowan, K. Paul, C. Geddes, P. Mullowney, P. Messmer, E. Esarey, E. Cormier-Michel, W. Leemans, J.-L. Vay, New developments in the simulation of advanced accelerator concepts, in: AIP Conference Proceedings, Vol. 1086, 2009, pp. 29–37.

[53] J.-L. Vay, *et al.*, Application of the reduction of scale range in a lorentz boosted frame to the numerical simulation of particle acceleration devices, in: Proc. Particle Accelerator Conference, Vancouver, Canada, 2009, tU1PBI04.

[54] S. F. Martins, R. A. Fonseca, W. Lu, W. B. Mori, L. O. Silva, Exploring laser-wakefield-accelerator regimes for near-term lasers using particle-in-cell simulation in lorentz-boosted frames, NATURE PHYSICS 6 (4) (2010) 311–316. doi:10.1038/NPHYS1538.

[55] W. Fawley, J.-L. Vay, Full electromagnetic fel simulation via the lorentz-boosted frame transformation, in: Proc. FEL 2010, paper MOPB01, Sweden, 2010.

[56] H. C. Wu, J. Meyer-ter Vehn, B. M. Hegelich, J. C. Fernandez, Nonlinear coherent Thomson scattering from relativistic electron sheets as a means to produce isolated ultrabright attosecond x-ray pulses, PHYSICAL REVIEW SPECIAL TOPICS-ACCELERATORS AND BEAMS 14 (7). doi:10.1103/PhysRevSTAB.14.070702.

[57] T. TAJIMA, J. DAWSON, Laser electron-accelerator, PHYSICAL REVIEW LETTERS 43 (4) (1979) 267–270.

[58] W. P. Leemans, B. Nagler, A. J. Gonsalves, C. Toth, K. Nakamura, C. G. R. Geddes, E. Esarey, C. B. Schroeder, S. M. Hooker, Gev electron beams from a centimetre-scale accelerator, NATURE PHYSICS 2 (10) (2006) 696–699. doi:10.1038/nphys418.

[59] J. Vay, C. G. R. Geddes, E. Cormier-Michel, D. P. Grote, Effects of hyperbolic rotation in minkowski space on the modeling of plasma accelerators in a lorentz boosted frame, Physics of Plasmas 18 (3) (2011) 030701. doi:10.1063/1.3559483.

[60] M. Bassetti, G. Erskine, Closed expression for the electrical field of a two-dimensional gaussian charge, in: CERN Report No. CERN-ISRTH/80-06, 1980.

[61] C. Huang, V. K. Decyk, C. Ren, M. Zhou, W. Lu, W. B. Mori, J. H. Cooley, T. M. Antonsen, Jr., T. Katsouleas, Quickpic: A highly efficient particle-in-cell code for modeling wakefield acceleration in plasmas, JOURNAL OF COMPUTATIONAL PHYSICS 217 (2) (2006) 658–679. doi:10.1016/j.jcp.2006.01.039.

[62] B. Feng, C. Huang, V. Decyk, W. Mori, P. Muggli, T. Katsouleas, Enhancing parallel quasi-static particle-in-cell simulations with a pipelining algorithm, Journal of Computational Physics 228 (15) (2009) 5340–8.

[63] A. FRIEDMAN, S. PARKER, S. RAY, C. BIRDSALL, MULTISCALE PARTICLE-IN-CELL PLASMA SIMULATION, JOURNAL OF COMPUTATIONAL PHYSICS 96 (1) (1991) 54–70. doi:10.1016/0021-9991(91)90265-M.

[64] J. ADAM, A. SERVENIERE, A. LANGDON, ELECTRON SUB-CYCLING IN PARTICLE SIMULATION OF PLASMA, JOURNAL OF COMPUTATIONAL PHYSICS 47 (2) (1982) 229–244. doi:10.1016/0021-9991(82)90076-6.