Particle-in-Cell Simulations of Plasma Accelerators and Electron-Neutral Collisions

David L. Bruhwiler,[a] Rodolfo E. Giacone,[b] John R. Cary,[a,b] John P. Verboncoeur,[c] Peter Mardahl,[c] Eric Esarey,[d] W. P. Leemans[d] and B. A. Shadwick[d,e]

[a]*Tech-X Corporation, 5541 Central Ave., Suite 135, Boulder CO 80301, USA*
[b]*University of Colorado, Physics Department, Boulder CO 80309-0390, USA*
[c]*University of California Berkeley, EECS Department, Berkeley CA 94720-1770, USA*
[d]*Lawerence Berkeley National Laboratory, University of California, Berkeley CA 94720, USA*
[e]*Institute for Advanced Physics, 10875 US Hwy 285, Suite 199, Conifer CO 80433, USA*

We present 2-D simulations of both beam-driven and laser-driven plasma wakefield accelerators, using the object-oriented particle-in-cell code XOOPIC, which is time explicit, fully electromagnetic, and capable of running on massively parallel supercomputers. Simulations of laser-driven wakefields with low (~10^{16} W/cm^2) and high (~10^{18} W/cm^2) peak intensity laser pulses are conducted in slab geometry, showing agreement with theory and fluid simulations. Simulations of the E-157 beam wakefield experiment at the Stanford Linear Accelerator Center, in which a 30 GeV electron beam passes through 1 m of preionized lithium plasma, are conducted in cylindrical geometry, obtaining good agreement with previous work. We briefly describe some of the more significant modifications to XOOPIC required by this work, and summarize the issues relevant to modeling relativistic electron-neutral collisions in a particle-in-cell code.

Motivation

The quest to understand the fundamental nature of matter requires ever higher energy particle collisions, which in turn leads to ever larger and more expensive particle accelerators. Plasma-based accelerators can sustain electron plasma waves (EPW) with longitudinal electric fields on the order of the nonrelativistic wave breaking field, $E_0=cm_e\omega_p/e$, where $\omega_p=(4\pi n_e e^2/m_e)^{1/2}$ is the plasma frequency at an electron density n_e (see Ref. 1 for a review of plasma based accelerator concepts). For $n_e=10^{18}$ cm^{-3}, the electric field is $E_0\cong 100$ GV/m, with a phase velocity close to the speed of light. Laser plasma accelerators have demonstrated accelerating gradients of 100 GV/m -- several orders of magnitude higher than for conventional structures -- providing hope for reaching new energy regimes.

Equally large amplitude wakefields can also be driven by intense relativistic electron beams. Plasma lenses, which can provide focussing strengths 1000 times greater than conventional quadrupole magnets, are a closely related application involving intense relativistic beams interacting with a plasma. Plasma lenses will be a crucial component of any plasma-based accelerator system. In cases where the plasma is not fully ionized, relativistic electron-neutral collisions can play an important role.

Laser-Driven Plasma Wakefield Acceleration (LWFA)

Research in laser plasma acceleration is very active, with many innovative concepts being explored through theory[2-9] and experiment.[10-16] In the “standard” laser wakefield

accelerator (LWFA) concept,[1] a single short (<1ps), ultrahigh intensity (>10^{18} W/cm^2) laser pulse injected into an underdense plasma excites an EPW behind the pulse. The plasma wake is excited by the ponderomotive force created by rapid oscillations of the electromagnetic field. The wakefield amplitude is maximum when the laser pulse length L is approximately equal to the plasma wavelength $L=\lambda_p$, where $\lambda_p=2\pi c/\omega_p$. A correctly placed trailing electron bunch can be accelerated by the longitudinal electric field and focused by the transverse electric field of the plasma wake.

Both 2-D and 3-D LWFA simulations are extremely demanding computationally, due to multiple time and space scales. The multiple scales arise, because the laser radiation field and the transverse electron oscillations evolve on a short time scale -- governed by the laser frequency ω -- with a correspondingly short wavelength, while the longitudinal plasma dynamics and consequent particle acceleration evolve on a much longer time scale -- governed by the electron plasma frequency ω_p -- and longer wavelength. Depending on the density of the plasma, the ratio ω/ω_p can vary from order unity to as high as 100. Thus, simulation codes for these problems must be parallelizable, so they can run on massively parallel processors (MPP), and in many instances a moving window algorithm leads to significant computational savings, allowing one to follow the laser pulse over distances long compared to the pulse length.

A particle-in-cell (PIC) treatment of laser plasma acceleration[17-20] provides a very detailed simulation of the relevant physics, but is generally constrained to follow the short time scale evolution of the laser pulse, and thus is the most computationally expensive approach. Fluid treatments[21-24] are computationally more efficient, especially models that average over the faster time scales, and are less noisy than PIC, but cannot model the dynamics of accelerated electrons. A third approach[25] uses a PIC treatment of time-averaged equations, along with the use of the quasistatic approximation[26] and sometimes other assumptions. Quasistatic approximations impose the assumption that there is a primarily forward-propagating laser pulse, thus ruling out certain instabilities, as well as all accelerating concepts involving multiple laser pulses that are incident from various angles.

Beam-Driven Plasma Wakefield Acceleration (PWFA)

Beam-driven plasma wakefield accelerators (PWFA)[27,28] are also capable of providing dramatic accelerating gradients, and thus may lead to a next-generation of smaller, cheaper high-energy accelerators. The acceleration mechanism in the PWFA is analogous to that in the LWFA, only in the PWFA the EPW is excited by the space charge force of the drive bunch, as opposed to the ponderomotive force of the laser pulse. It has been proposed[29] to use the PWFA concept as a means of doubling the beam energy of the Stanford Linear Accelerator Center (SLAC) Linear Collider (SLC) in a distance of only seven meters. This so called “afterburner” would possibly enable detection of the Higgs particle.

A PWFA experiment, referred to as the E-157 experiment,[30-33] aimed at demonstrating accelerating gradients on the order of 1 GeV/m is currently underway at SLAC. In this experiment, a 30 GeV electron bunch is injected into a 1 to 1.5 m long plasma column

with density on the order of 2-3 x 10^{14} cm^{-3}. E-157 operates in the "blow-out" regime of the PWFA, meaning the number density of the electron bunch is greater than the plasma density, so that all of the plasma electrons are expelled from the axis in the vicinity of the electron bunch. The EPW generated by the electron bunch is expected to accelerate electrons in the tail of the bunch to higher energies. The plasma afterburner concept is a scaled up version of E-157, which will operate at much higher plasma density, thus requiring a much short duration electron bunch, which will generate an EPW with much stronger longitudinal fields.

In E-157, the laser-ionized lithium plasma density is roughly 10% of the neutral lithium density, $n_0 \sim 2 \times 10^{15}$ cm^{-3}. One proposal for an SLC afterburner[29] requires a plasma density two orders of magnitude larger, corresponding to a neutral lithium density of $n_0 \sim 2 \times 10^{17}$ cm^{-3}. At such high densities, the effects of relativistic electron-neutral collisions can become significant.

Plasma Lenses

When a high-energy electron bunch enters a plasma, the plasma electrons exit the beam volume so as to neutralize the space-charge-induced radial electric field. Thus, the outward force of the radial electric field can no longer balance the inward force of the azimuthal magnetic field, and the electron bunch is focussed. This force can be as much as 1000 times stronger than that of a conventional magnetic quadrupole, and it has the added benefit of focussing simultaneously in both transverse planes.

In a plasma lens experiment at LBNL, a 30 ps pulse of 50 MeV electrons was seen to induce significant ionization in a Tripropylamine target, leading again to focussing of the beam.[34] Also, in the E-150 plasma lens experiment at SLAC,[35,36] a 2.3 ps pulse of 30 GeV electrons has been seen to induce 7% ionization in a high-density Nitrogen gas, leading to significant beam focussing. Photoionization by the copropagating optical field, direct impact ionization by the 30 GeV electrons, and subsequent impact ionization by the resulting electrons are the likely mechanisms for this ionization.

Background Material

In this section, we provide background material on time explicit, electromagnetic PIC simulations, including Monte Carlo collision models. We also discuss some aspects of the XOOPIC code, which was originally developed in a pioneering effort to apply the principles of object-oriented programming to a full PIC code.

Particle-in-Cell Approach to Plasma Simulation

Plasma simulation with many particles was started by Dawson and Buneman in the late 1950's, with fields obtained from Gauss's Law in one dimension. The step to two and three dimensions followed, but this required a mesh for charge and current collection as well as solution of the field equations; this method was called particle-in-cell (PIC). The PIC method was codified in the 1960's-1980's by Birdsall and Langdon[37] and by Hockney and Eastwood.[38] More recent additions include techniques for including

boundaries and external circuits[39] as well as Monte Carlo collisions (MCC) with neutral particles.[40]

The standard PIC-MCC scheme solves the equations representing a coupled system of charged particles and fields. The particles are followed in a continuum space, while the fields are computed on a mesh. Interpolation provides the means of coupling the continuum particles and the discrete fields.

A scheme for advancing the particles and fields one timestep is shown in Figure 1. First, the forces due to the electric and magnetic fields are used to advance the velocities of the particles, and subsequently the velocity is used to advance the position. Particle boundary conditions such as emission and absorption are then applied. If collisions with a neutral background gas are included, the velocities are updated to reflect elastic and inelastic collisions. Next, the particle positions and velocities are used to compute the charge density and current density on the mesh. The charge density and current density provide the source terms for the integration of the field equations (Poisson equation in the electrostatic limit, or Maxwell's equations in the electromagnetic case) on the mesh. The fields resulting from the integration are then interpolated to particle locations to provide the force on the particles.

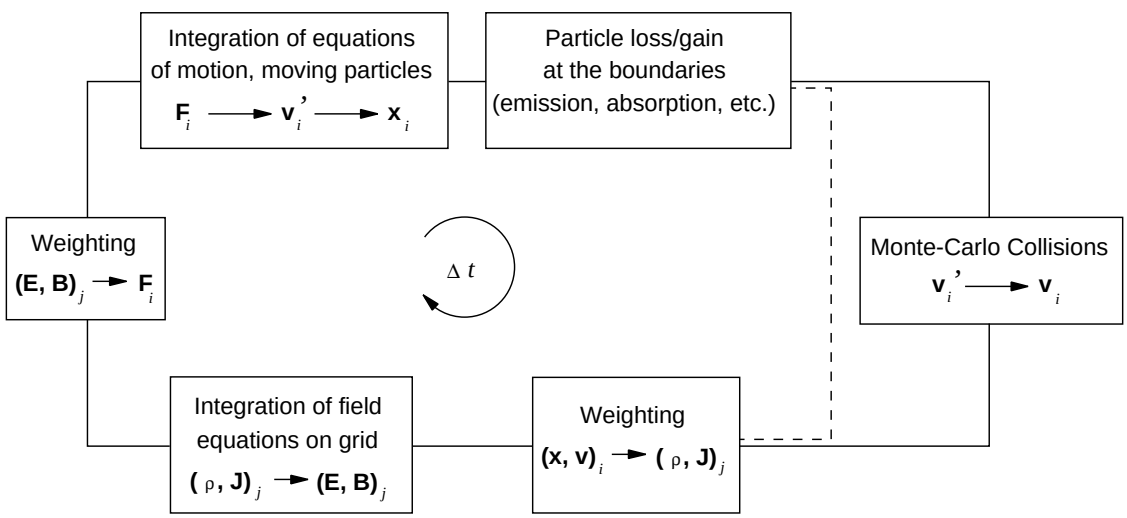

FIGURE 1. Flow chart for an explicit PIC-MCC scheme (adapted from Ref. 41).

The XOOPIC Particle-in-Cell Code

The XOOPIC (X11-based object oriented particle-in-cell)[42] code started as a pioneering effort to apply object oriented techniques to plasma simulation codes. A brief discussion of object-oriented programming techniques is provided in the appendix below. XOOPIC is written in C++, and includes the XGrafix[43] user interface. XOOPIC is widely used by the plasma and beam-microwave community worldwide, with applications ranging from high pressure discharges to relativistic microwave devices.

XOOPIC models two spatial dimensions in both Cartesian (x,y) and cylindrical (r,z) geometry, including all three velocity components, with both electrostatic and electromagnetic models available. All three components of both the electric and the magnetic fields are modeled, but there is no spatial variation along the ignored coordinate.

The code presently supports a non-uniform orthogonal mesh and arbitrary placement of most boundary conditions on that mesh. Static magnetic fields can be added analytically using the equation evaluator, or read from an external file. A number of different charge and current weighting algorithms are available, as well as Poisson and Langdon-Marder divergence corrections for non-conservative current weighting schemes. The code includes a fully relativistic model for inertial particles, as well as a Boltzmann model for inertialess electrons. Particles and fields can each run on independently subcycled time steps, improving computational efficiency. A temporal filtering scheme reduces high frequency noise, and a spatial digital filtering algorithm reduces short wavelength noise.

XOOPIC also includes volumetric and surface plasma injection, including thermionic and field emission models. Particle statistics can be collected at arbitrary surfaces, and field and particle data can be averaged over arbitrary volumes and surfaces. A Monte Carlo collision (MCC) technique[40] allows multiple background gases at arbitrary partial pressures. The features described are all adjustable from the input file, using MKS (or arbitrary) units for input parameters.

XOOPIC uses the message passing interface (MPI)[44] to take advantage of massively parallel (MPP), symmetric multiprocessor (SMP) and distributed architectures. For the PWFA simulations presented below, XOOPIC has demonstrated linear scaling for up to 16 processors on the Cray T3E. Improved parallel performance on more processors can be obtained on larger-scale problems with optimized domain decomposition.

Recent Modifications to XOOPIC

Previous to the work presented here, XOOPIC had been used to simulate the plasma lens experiments at LBNL,[45] in which 50 MeV electron beams were propagated through 1-3 cm long plasmas of density $n_e \sim 10^{14}$ cm^{-3}. Furthermore, XOOPIC has been used extensively to model microwave devices, plasma diodes, plasma display panels, and other beam and plasma devices. The authors have modified and enhanced XOOPIC so that it can be used to model high-energy plasma-based accelerators in 2-D Cartesian or cylindrical geometry, whether on a single-processor workstation, a networked cluster, or the massively parallel Cray T3E. The object-oriented architecture of XOOPIC made it possible to complete this task in a relatively short time.

Development of a Moving Window Algorithm

Plasma-based accelerators are too large to simulate the entire device, and it is only the small region in the vicinity of the particle beam or laser pulse that must be modeled. Because this beam or pulse is moving near the speed of light, it is possible to implement a

"moving window" algorithm, such that the simulation follows the small region of interest and ignores the rest of the device.

There are two fundamental approaches to implementing a moving window. One is to move the mathematical mesh along with the particles, and give the background and walls a velocity relative to the mesh. A 3-D moving-window algorithm for cylindrical geometry was implemented in this manner in the ELBA code.[46] The other approach is to keep the mesh stationary with respect to the background, create new particles and fields on the leading edge, shift existing particles and fields to neighboring mesh points, and discard any particles and fields on the trailing edge. The second approach was used previously in the PEGASUS code[47] and more recently in the OSIRIS code.[48] This approach is also used for XOOPIC, because it requires no modifications to the basic field solve and particle push, and because it eliminates numerous other complications.

For a window which is following a group of particles moving to the right, new analytical fields (typically all zero) are introduced into the rightmost row of mesh points, and the fields in the rightmost row of mesh points are copied to the row immediately to the left, and so on. When this shift in the fields takes place, all the particles must also be shifted. At this time, any particles in the leftmost row of cells (for a rightward moving window) are discarded, for they have left the window. New particles may be introduced in the rightmost row of cells, if required.

Boundary conditions present no difficulty, if the moving window travels at the speed of light. In the case of a rightward-moving window, disturbances at the leftmost boundary cannot propagate into the moving window, because all electromagnetic waves are constrained to move with a velocity less than or equal to the speed of light. Similarly, incoming fields on the right hand side are not affected by the contents of the moving window to the left, so fields here may be specified analytically in a simple way.

Combining parallel operation and the moving window leads to some additional complication. Whenever a shift takes place (typically every few time steps), the shifted fields and particles must be passed to the downstream computational region. The moving window in XOOPIC uses MPI to pass particles and fields across computational boundaries.

Adding a New Electromagnetic Pulse Launcher

We designed and implemented within XOOPIC the ability to launch a linearly polarized electromagnetic pulse in 2-D Cartesian geometry. The initial longitudinal spatial profile is Gaussian. The initial transverse spatial profile is Gaussian along one direction and has no variation along the other direction, which is aligned with the ignorable coordinate. From the input file, the user can specify the on-axis intensity of the electromagnetic pulse, as well as the frequency, the pulse length, the spot size, the location of the waist and the polarization direction.

The electromagnetic pulse is launched from a given boundary by controlling the temporal and spatial dependence of the electric field at that boundary. In 2-D slab geometry, the magnitude of the electric field at the boundary is specified to be the real part of

$$E(x,z,t) = \hat{E}_0(x,z)\exp[i(kz - \omega t)], \tag{1}$$

where z is the direction of propagation. The spatial dependence of the complex electric field amplitude $\hat{E}_0(x,z)$ is given by

$$\hat{E}_0(x,z) = E_0\left(\frac{W_0}{W(z)}\right)^{1/2} \exp\left[\frac{-x^2}{W^2(z)} + i\frac{kx^2}{2R(z)} - i\frac{\phi(z)}{2}\right], \tag{2}$$

where the constant E_0 is the peak field amplitude, and x is the transverse coordinate.

The width of the electromagnetic pulse is $W(z)$, which takes its minimum value W_0 in the plane $z = z_f$, as can be seen from Eq. (3):

$$W(z) = W_0\sqrt{1 + (z - z_f)^2 / z_0^2}\ , \tag{3}$$

where the spot size of the pulse is given by the waist diameter, $2W_0$. Both the phase and the radius of curvature of the wavefront are strong functions of the Rayleigh length z_0 and the distance from the waist: $R(z) = (z - z_f) + z_0^2/(z - z_f)$, $\phi(z) = \tan^{-1}[(z - z_f)/z_0]$.

Generalization of the Particle Beam Emitters

The beam emission boundary conditions in XOOPIC have been extended to handle more general cases. Spatial dependence has been added to both the BeamEmitter algorithm, which emits particles of a specified computational particle weight as well as the VarWeightBeamEmitter algorithm, which emits particles of variable weights. Particle weight is defined as the ratio of the charge of a computational particle to that of a physical particle, $w = q_c / q_p$. Also, the particle weight in the VarWeightBeamEmitter has been generalized to have both spatial and temporal dependence, and the weight can be set to adjust automatically to emit a fixed number of particles per timestep.

The previously existing emitter models in XOOPIC emitted a specified time-dependent current I(t), which could be specified from the input file. Only uniform current density was possible. Furthermore, the VarWeightBeamEmitter only allowed for variation of the weight of particles based on the radial origin of emission for particles, $w(r) = w_{\max} r / r_{\max}$.

Both emitter models now have full spatial and temporal dependence for the current density, $J = J(x,t)$, specified from the input file. Furthermore, the VarWeightBeamEmitter now sets the particle weight proportional to the current density for Cartesian coordinates ($w \propto J(x,t)$), and proportional to the product of the radius and the current density for the cylindrical model ($w \propto rJ(r,t)$).

The total current for both models is specified in the input file. A spatial-temporal profile factor is also specified, $f(x,t)$, where x refers to the coordinate along the emitter. The final relevant parameter is the particle weight, w. The current density is given by

$$J(x,t) = f(x,t)\frac{I_0}{A}, \tag{4}$$

where I_0 is the specified current, and A is the cross-sectional area of the emitter. Using these definitions, the total current as a function of time can be written:

$$I(t) = \int_{x_1}^{x_2} g(x)J(x,t)dx = \frac{I_0}{A}\int_{x_1}^{x_2} g(x)f(x,t)dx, \tag{5}$$

where $g(x)$ is a geometric factor given by $g(x)=1$ for Cartesian coordinates and $g(x) = 2\pi r$ for axisymmetric cylindrical coordinates. Then the mean current density is $J_{avg}(x,t) = I(t)/A$, and the number of particles emitted per timestep is $N = \Delta t I(t)/q_c$, where $q_c=q_p w$ is the charge on a computer particle of weight w.

In the XOOPIC implementation for BeamEmitter, the function $f(x,t)$ is integrated each timestep (or less when the spatial and temporal dependencies are decoupled). The integration yields the cumulative distribution function,

$$F(x,t) = \frac{\int_{x_1}^{x} g(x')f(x',t)dx'}{\int_{x_1}^{x_2} g(x)f(x,t)dx}, \tag{6}$$

which maps the probability of a particle position x to a uniformly distributed random number $0 \le R < 1$ by setting $F=R$.

For the XOOPIC implementation of the VarWeightBeamEmitter, the particle positions are distributed uniformly in x, and the weight is set according to $w(x,t) = w_0 g(x) f(x,t)$, where w_0 is the user-specified reference weight. The VarWeightBeamEmitter now also accepts specification of the number of particles to emit per timestep, adjusting the value of w_0 appropriately to achieve the desired result.

Laser-Driven Plasma Accelerator Simulations

We present two simulations of the standard LWFA, one driven by a low (5.5×10^{16} W/cm^2) and the other a high (3×10^{18} W/cm^2) peak intensity laser pulse, both in slab geometry. These simulations have relevance to ongoing LWFA experiments at the l'OASIS laboratory of Lawrence Berkeley National Laboratory.[49,50] PIC simulations are necessary to understand the detailed particle trapping mechanisms in these experiments.

These results demonstrate the capabilities of XOOPIC. Preliminary XOOPIC studies of the effects of colliding laser pulses[4,5] have been presented in Ref. 51.

Modeling the Wakefield Generated by a Low Intensity Laser Pulse

We first consider the plasma wakefield generated by a low intensity laser pulse. The electron plasma density is n_e=3x10^{19} cm^{-3}, which corresponds to an EPW wavelength of $\lambda_p = 6\ \mu m = 6.2\ c/\omega_p$ and a plasma frequency of ω_p=3.1 x10^{14} rad/s.

The laser pulse is linearly polarized, with a transverse Gaussian profile. The minimum laser spot size is 5 μm = 5.2 c/ω_p, and the Rayleigh length is Z_R = 97 μm = 100 c/ω_p = 16 λ_p. In order to maximize the EPW amplitude, the laser pulse length is chosen to be of order λ_p, with a full width at half maximum τ_{fwhm} = 6.7 fs, which is equivalent to 2 μm or $\lambda_p/3$. The peak laser intensity is I_L=5.5x10^{16} W/cm^2, corresponding to a dimensionless amplitude a_0=0.2, and the laser wavelength is $\lambda = c/\omega_p$=1 μm.

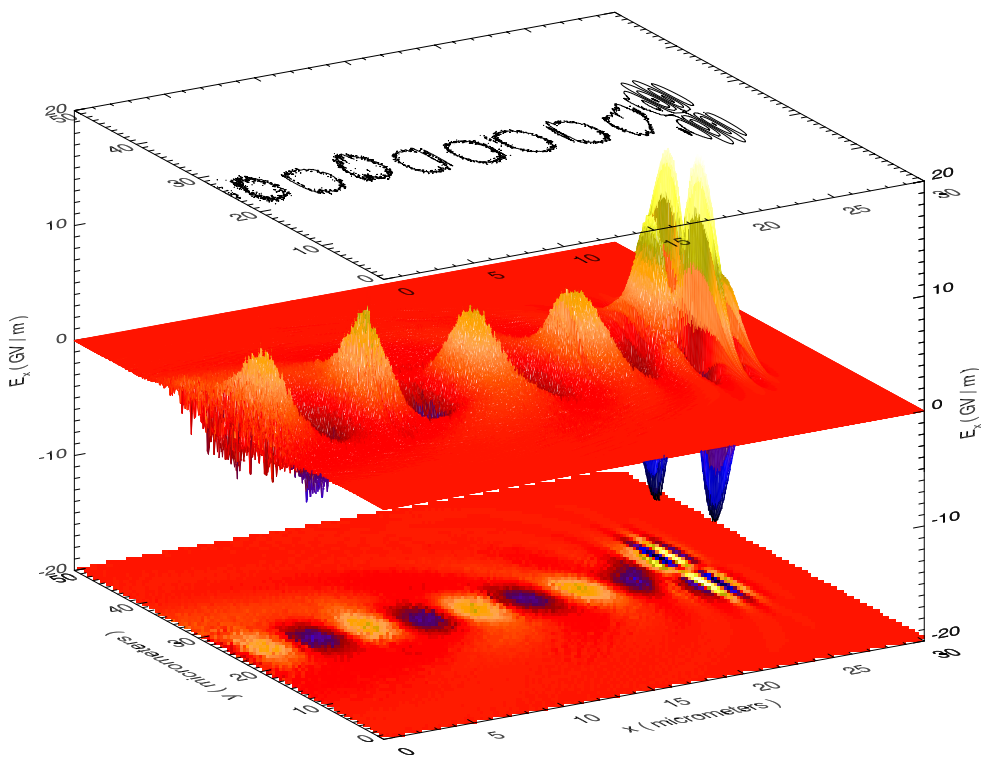

FIGURE 2. Surface plot of the longitudinal electric field E_x generated by the 5.5x10^{16} W/cm^2 (a_0=0.2) laser pulse (large peaks to the right) and the resulting plasma wake (smaller peaks, left and center). The structure of the laser pulse is seen clearly in the contour plot (above) and the surface plot projection (below). E_x is shown in GV/m, while the coordinates x, y are shown in μm.

Figure 2 shows a surface plot of the longitudinal electric field E_x over the mesh. The length of the simulation region is L_x = 30 μm = 31 c/ω_p in the x (longitudinal) direction and L_y = 50 μm = 52 c/ω_p in the y (transverse) direction. The simulation uses seven

particles per cell to represent the plasma, and the initial plasma is cold. The plasma wake can be clearly seen behind the laser pulse.

Figure 3 shows a line-out of E_x along the axis of the laser pulse and the EPW. The EPW is linear, with a peak gradient of $E_x \sim 5.5$ GV/m $\sim 0.01\ E_0$. In this regime, the wake amplitude is in good agreement with that predicted by linear theory.[1] The noise seen in Fig.'s 2 and 3 is caused by the finite number of particles, and can be reduced by increasing the number of particles per grid cell.

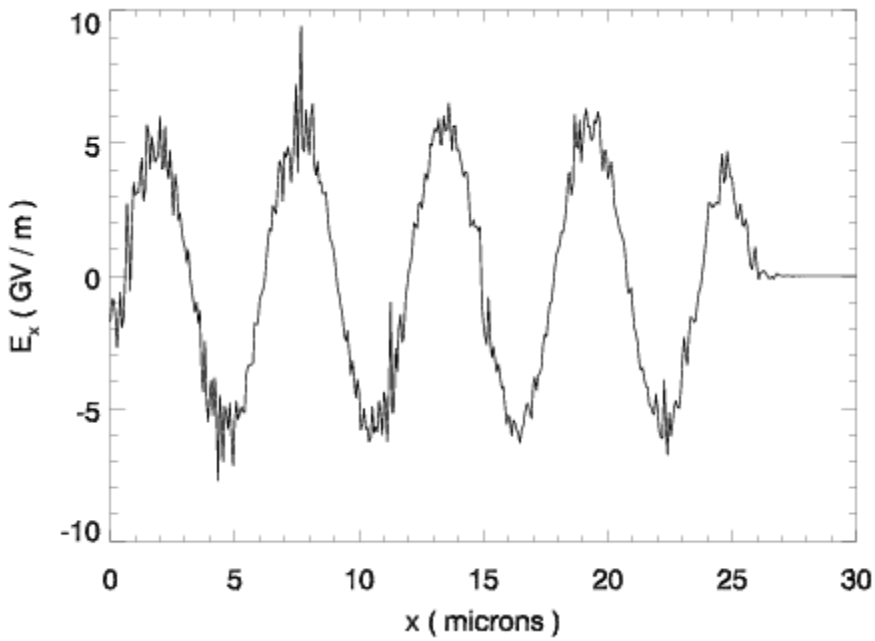

FIGURE 3. Line-out of the longitudinal electric field E_x generated by the 5.5×10^{16} W/cm^2 (a0=0.2) laser pulse, along the line y=25 μm, which is down the middle of the pulse.

The Wakefield of a High-Intensity Laser Pulse

We now consider the plasma wakefield generated by a high intensity laser pulse. All the physical and simulation parameters are the same as for the low intensity pulse of the previous subsection, but the peak intensity is now $I_L = 3 \times 10^{18}$ W/cm^2, corresponding to a dimensionless amplitude $a_0 = 1.5$.

Figure 4 shows a surface plot of the longitudinal electric field E_x over the mesh for the high-intensity case. The much larger magnitude EPW dwarfs the longitudinal field of the laser pulse, which is now partially hidden at the far right of the plot. (Of course, the transverse components of the laser field are very much larger than the longitudinal component.) The structure of the laser pulse can be seen better in the contour plot located above the surface plot of Fig. 4. The phase fronts of the plasma wake are now curved behind the laser pulse due to nonlinear shifts in the plasma wavelength. These results are in good agreement with recent nonlinear fluid simulations.[24]

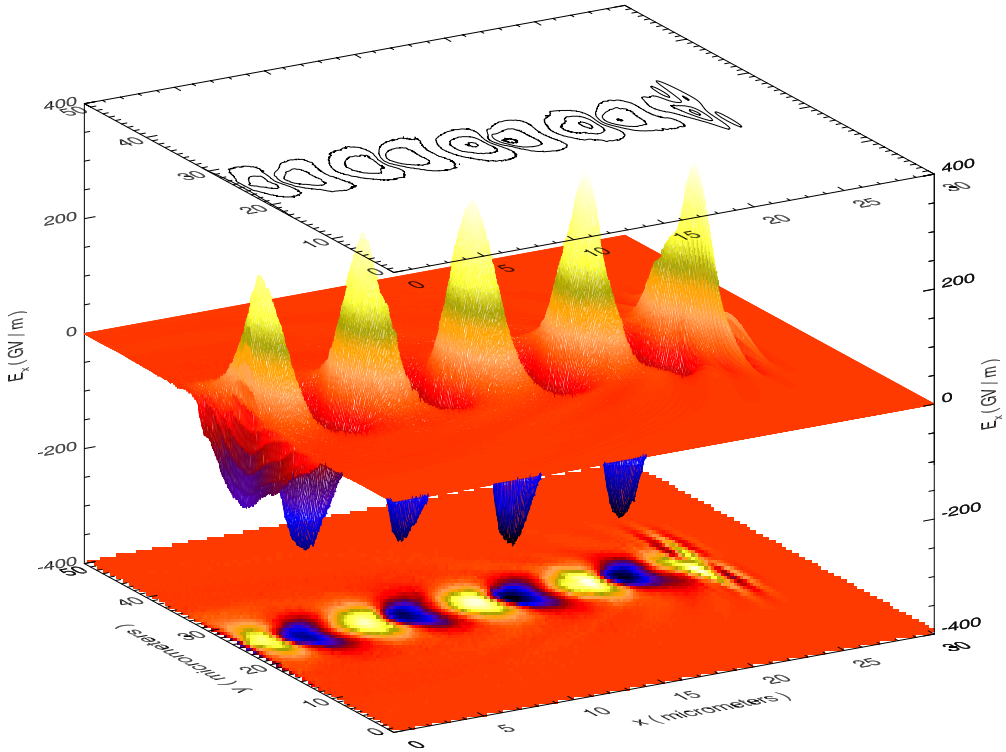

FIGURE 4. Surface plot of the longitudinal electric field generated by the $3x10^{18}$ W/cm^2 (a_0=1.5) laser pulse (smaller, partially hidden peaks to the far right) and the resulting plasma wake (larger peaks). The structure of the laser pulse is seen in the contour plot (above) and the surface plot projection (below). E_x is shown in GV/m, while the coordinates x, y are shown in μm.

Figure 5 shows a line-out of E_x along the axis of the laser pulse and the EPW. The EPW has a peak gradient of $E_x \sim 300$ GV/m $\sim 0.56\, E_0$. The wake amplitude is found to increase linearly with the peak laser intensity (as the square of the dimensionless amplitude) when I_L<$3x10^{18}$ W/cm^2, in agreement with theory.

The relative simulation noise seen in Fig.'s 4 and 5 is much smaller than for the low-intensity results shown in Fig.'s 2 and 3. One might expect the relative noise to be the same in these two simulations, because the fields in an electromagnetic PIC code are driven by the particle currents, and the finite number of particles per cell leads to a relative error in the current. This reduction in the relative error associated with particle statistics due to the use of nearest-grid-point (NGP) current deposition and the maximum excursion of the oscillating electrons that drive the EPW.

These simulations used the charge-conserving current deposition algorithm of Villasenor and Buneman,[52] which uses NGP in the direction of particle motion and linear interpolation in the transverse direction. The field component E_x is driven by the particle current j_x, which is calculated from the charge and velocity of the electrons, and these particles are primarily oscillating in the x-direction. Thus, the NGP current deposition leads to a lot of statistical noise in both j_x and E_x.

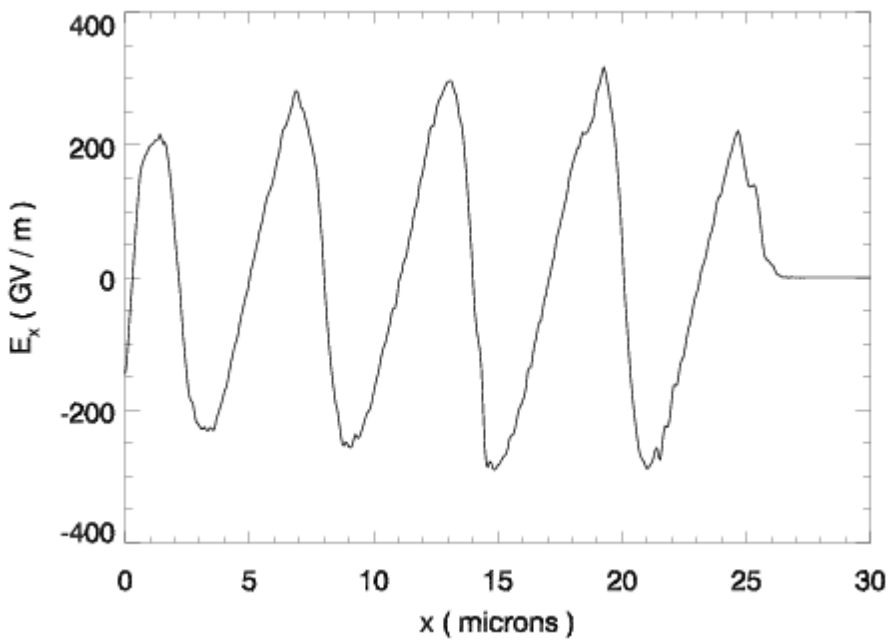

FIGURE 5. Lineout of the longitudinal electric field generated by the $3x10^{18}$ W/cm^2 (a_0=1.5) laser pulse, along the line y=25 μm, which is down the middle of the pulse.

For the simulation shown in Fig.'s 2 and 3, the maximum excursion of the oscillating particles is only 13% of the cell size. In this case, only a small fraction of the oscillating particles ever cross a cell boundary, so that the NGP current deposition leads to local currents that reflect the full statistical variation of particle locations. For the higher-intensity simulations shown in Fig.'s 4 and 5, the excursion amplitude is 6.5 times the cell size, a factor of 50 larger, corresponding to the ratio of the electric field amplitudes for the two cases. In this case, all of the particles are frequently crossing cell boundaries, which tends to average out the statistical variations.

Thus, the relative error associated with finite particle statistics in this case does not scale simply as $N_p^{-1/2}$, where N_p is the number of particles per cell. Instead of considering N_p, one must consider the number of particles crossing cell boundaries in the x-direction per unit time, which implies that the relative error scales as $(N_pE_{max})^{-1/2}$. For this reason, the relative noise in the higher-intensity simulation is smaller by roughly a factor of seven. It would be worthwhile to verify this scaling rigorously and to compare the results with simulations using bilinear current deposition (with corrections to preserve charge conservation), but such a study is beyond the scope of the present work.

Beam-Driven Plasma Accelerator Simulations

In this section, we present XOOPIC simulations of the E-157 PWFA experiment[32,33] at SLAC, showing good agreement with results obtained previously using the OSIRIS[48] code. We also discuss the important issue of electron-neutral collisions.

Modeling the SLAC E-157 Experiment

The simulation region, in 2-D cylindrical geometry, is 0.9 mm in r by 5.4 mm in z, with the corresponding number of grid points n_r=32 and n_z=192, for a total of 6144 cells. With 20 particles per cell representing the plasma electrons, there are 120,000 plasma particles. The 30 GeV electron beam is represented by 1000 particles per cell (on average), and the beam covers 8 by 64 grids (initially) for 510,000 beam particles. The grid size is dr = dz = 28 μm. The time step, chosen to satisfy the Courant condition, is dt = 0.4 dz/c = 3.8×10^{-14} s. Thus, it requires 88,000 time steps to propagate the beam through the 1 m lithium plasma.

The plasma density is taken to be 2.1×10^{14} cm^{-3}, which implies an electron plasma frequency of $\omega_p = 8.2 \times 10^{11}$ rad/s. Thus, ω_p dt=0.03 and the electron plasma frequency is being resolved, which is required for stability in a time-explicit PIC code. The lithium plasma is assumed to be cold, and very little numerical heating is observed, because the moving window algorithm "sweeps" the electrons through at the speed of light. Giving the plasma electrons an initial temperature increases the statistical noise without any significant change to the physical results.

Figure 6 shows the initial 30 GeV beam in cylindrical coordinates. The particles are variably-weighted with radius, allowing for a particle density that is independent of radius, while the initial charge density is Gaussian in both the longitudinal and transverse directions. The beam enters the simulation region from the left with zero energy spread and zero emittance, propagating to the right with a velocity very close to the speed of light. Once the beam is close to the right edge of the simulation region, the moving window algorithm is invoked, keeping the beam in the same relative location.

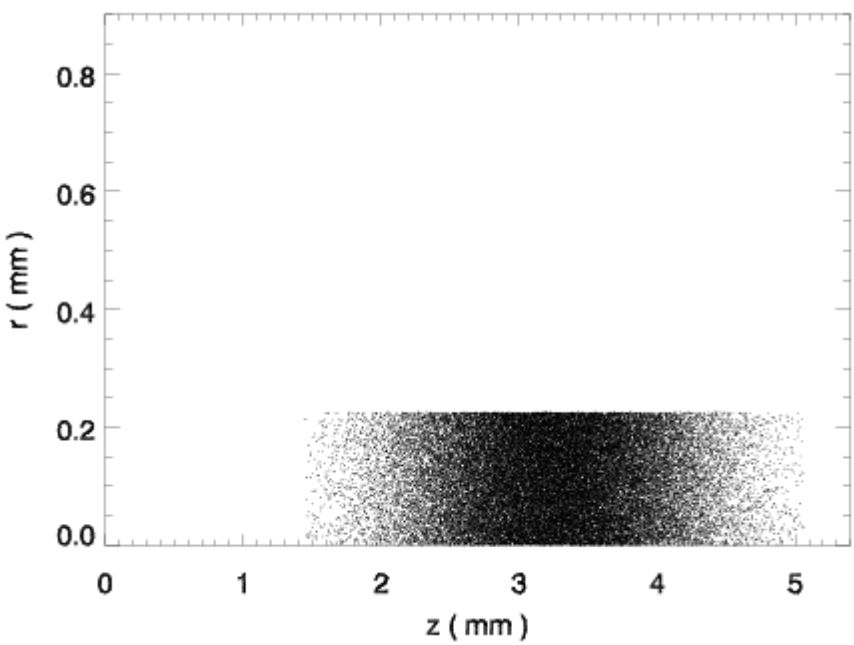

FIGURE 6. The initial distribution of the particles representing the 30 GeV electron beam shows the Gaussian profile in the longitudinal direction. The particles are uniformly distributed in radius, but are variably-weighted so that the charge density is also Gaussian in the transverse plane.

As noted above, the electron beam enters a cold lithium plasma. The plasma ions are modeled as a stationary uniform background, while the plasma electrons are modeled with uniformly-distributed particles with initially zero velocity. Because the peak beam density n_b=1.4x10^{15} cm^{-3} exceeds the plasma density n_e=2.1x10^{14} cm^{-3}, the plasma is said to be underdense and the self-fields of the beam cause "blow-out" of the plasma electrons, which is the origin of the strong plasma wake.

Figure 7 shows the plasma electrons after the 30 GeV electron beam has propagated for 80 cm. This "wake" in the particle distribution is what drives the EPW. Plasma electrons near the head of the electron beam are driven away from the axis. Most of these blown-out plasma electrons return to the axis near the tail of the electron beam.

In the vicinity of the small region centered around (z=1.6, r=0), where the blown-out electrons return to the axis, the electron plasma density is increased by almost two orders of magnitude. This localized increase in the charge density creates a strong spike in the electric field component E_z within the tail of the electron beam. Also, the space charge of the unneutralized plasma ions in the wake generates a strong radial electric field component, which focusses the bulk of the electron beam. The structure of the plasma wake is independent of the beam radius and remains approximately static throughout the simulation.

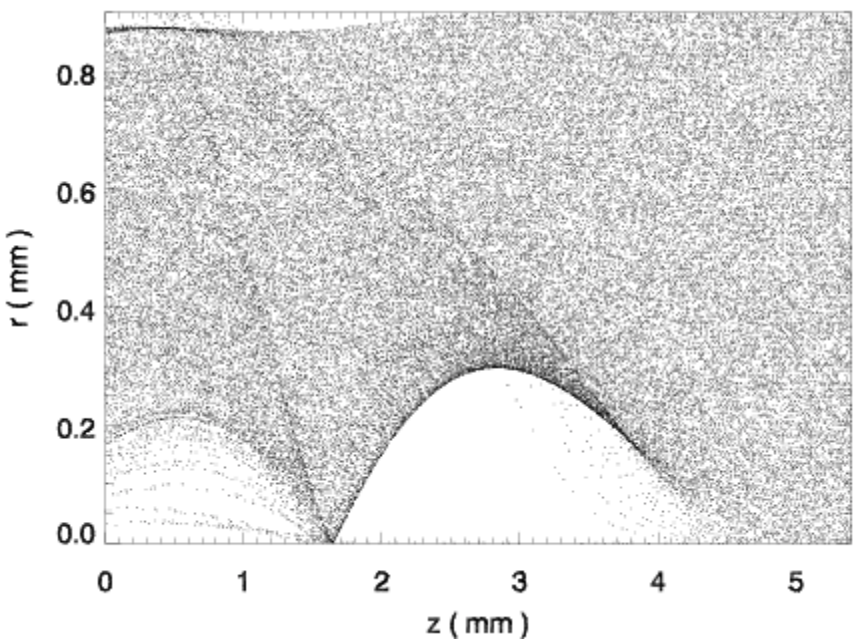

FIGURE 7. The distribution of plasma electrons shows blowout of electrons from the axis and formation of a large plasma wake. The plasma wake is approximately static throughout the simulation.

The boundaries of the simulation region are ideal conductors. While unphysical, these boundary conditions are sufficiently far away that they do not impact the relevant physics. The crossing of particle trajectories in the wake, seen in Fig. 7, indicates highly nonlaminar flow, which cannot be modeled accurately with a fluid code.

Because the strong radial focussing field of the wake exceeds the combined effects of the beam emittance and the beam self-fields, the beam undergoes betatron oscillations with a wavelength of 40 cm. The 30 GeV electron beam is seen in Fig. 8 after 20 cm of propagation, corresponding to one half of a betatron wavelength. The head of the beam is largely unaffected, because it does not see the wake. There is a transition region where the radial focussing varies along the beam, but the bulk of the beam electron distribution shows strong radial compression.

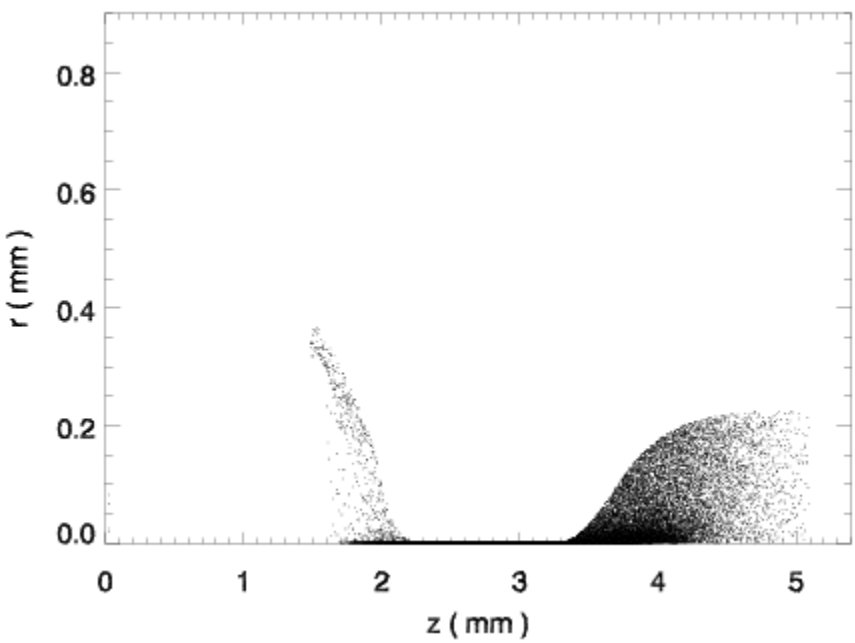

FIGURE 8. The beam electron distribution after 20 cm (one half betatron wavelength) of propagation through the lithium plasma shows that the bulk of the beam has been focussed to a waist.

Figure 9 shows the 30 GeV electron beam after 80 cm of propagation, corresponding to two betatron wavelengths. Just as was seen in Fig. 8, the head of the beam remains largely unaffected, because it does not see the wake. The transition region near the head of the beam, seen also in Fig. 8, shows here the continuing effects of the variation in focussing force. The bulk of the beam electron distribution has recovered the initial maximum radius.

Figure 10 shows the longitudinal electric field E_z generated by the wake, in GV/m, after 80 cm of propagation. The large peak value of approximately -0.8 GV/m coincides with the peak in the plasma electron density at the point where plasma electrons in the wake return to the axis of symmetry. With higher resolution, the peak field value on axis is enhanced (greater than 1 GV/m), but very little difference is observed in the structure of the field away from the peak. Thus, the present resolution is adequate to model acceleration of electrons within the tail of the 30 GeV beam. The features seen in Fig. 10 are approximately static during the full 100 cm of beam propagation.

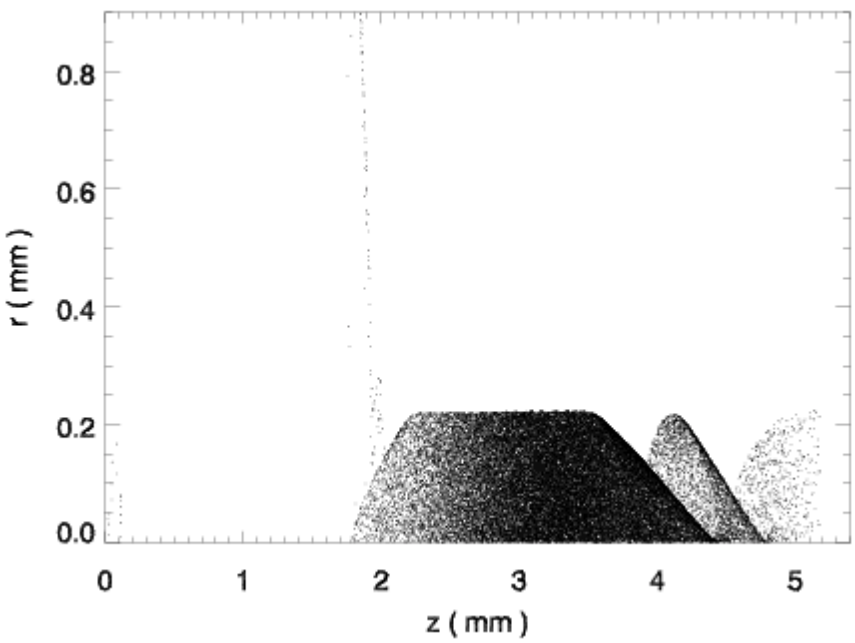

FIGURE 9. The electron beam distribution after 80 cm (two betatron wavelengths) of propagation through the lithium plasma shows that the bulk of the beam has returned to its original radial size.

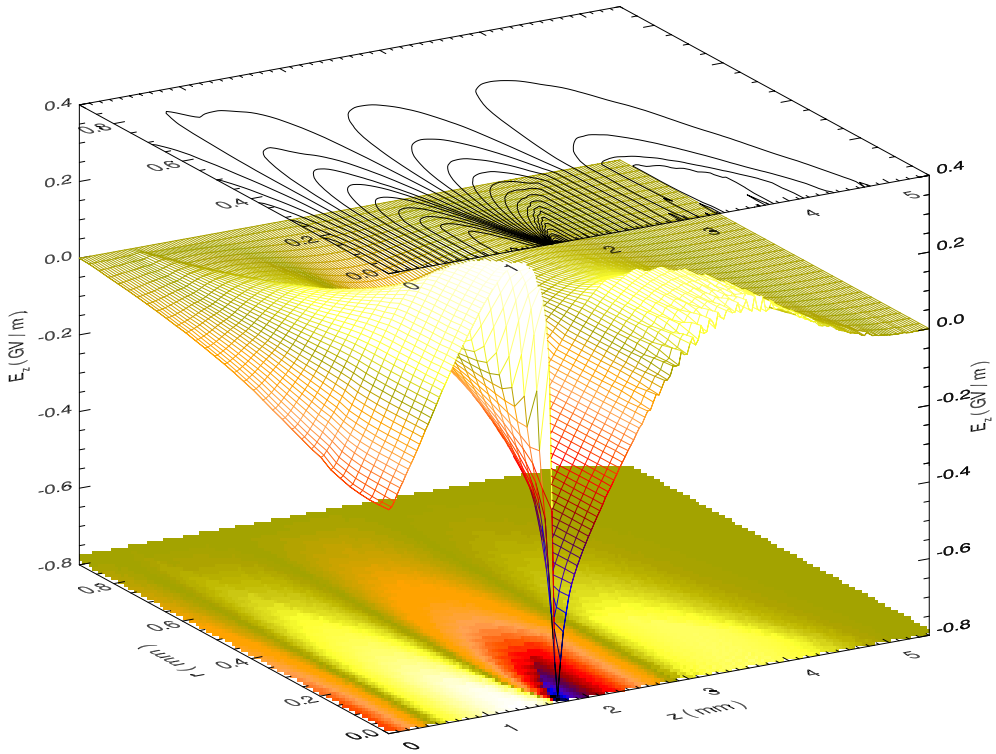

FIGURE 10. This surface plot of the longitudinal electric field E_z, due to both the 30 GeV electron beam and the resulting plasma wake, shows a strong localized peak of roughly -0.8 GV/m. The corresponding line contours (above) and the surface plot projection (below) provide further detail. E_z is shown in GV/m, while the transverse coordinates z, r are shown in mm.

Figure 11 shows a line-out of E_z along the axis of radial symmetry (r=0), after 80 cm of beam propagation through the plasma. The noise seen in Fig. 11 is caused by the finite number of particles and is exacerbated by the inherent noisiness along the symmetry axis. This noise is seen only along the axis, and it does not grow in time.

It can be seen in Fig. 11 that E_z is positive for 3 mm < z < 4 mm, which decelerates the electrons in the center of the 30 GeV beam distribution. This energy taken from the beam provides the energy needed to drive the wake. For 1.5 mm < z < 2.5 mm, E_z is large and negative, which accelerates electrons in the lower density tail of the electron beam distribution.

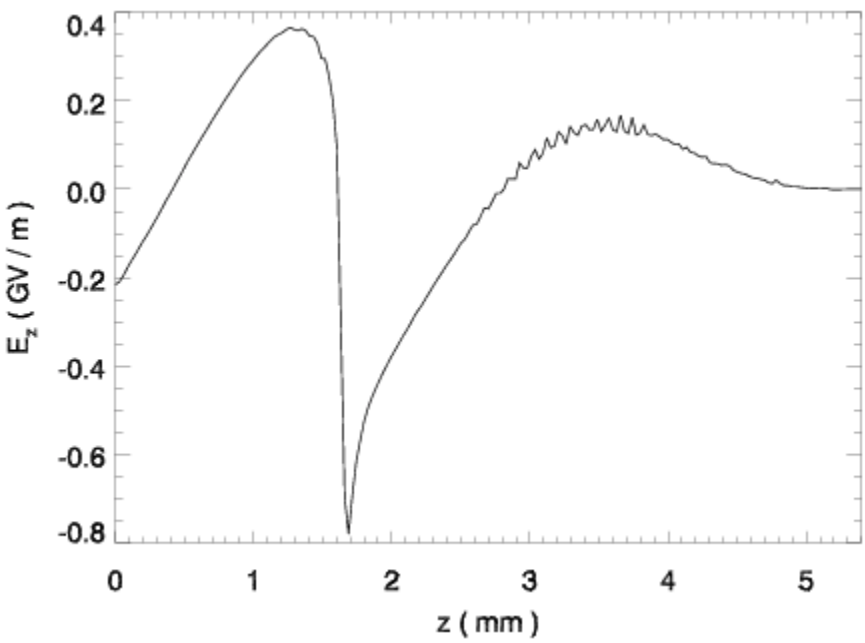

FIGURE 11. Lineout along the axis of cylindrical symmetry (r=0) of the longitudinal electric field E_z shows the large accelerating fields generated by the plasma wake behind the 30 GV/m electron beam. The noise seen to the right of the figure is found only along the symmetry axis and does not grow in time. The peak field value of –0.8 GV/m overlaps the tail of the electron beam distribution.

Figure 12 shows the energy distribution of the electrons representing the 30 GeV electron beam, after 100 cm of propagation through the lithium plasma. Particles in the head of the beam neither gain nor lose energy, because they do not see the plasma wake. Particles in the bulk of the distribution are decelerated, losing as much as 150 MeV. Particles in the tail of the distribution are accelerated, gaining as much as 550 MeV. The goal of the E-157 experiment is to measure the energy increase of the tail particles and thus confirm the existence of the strong accelerating electric fields.

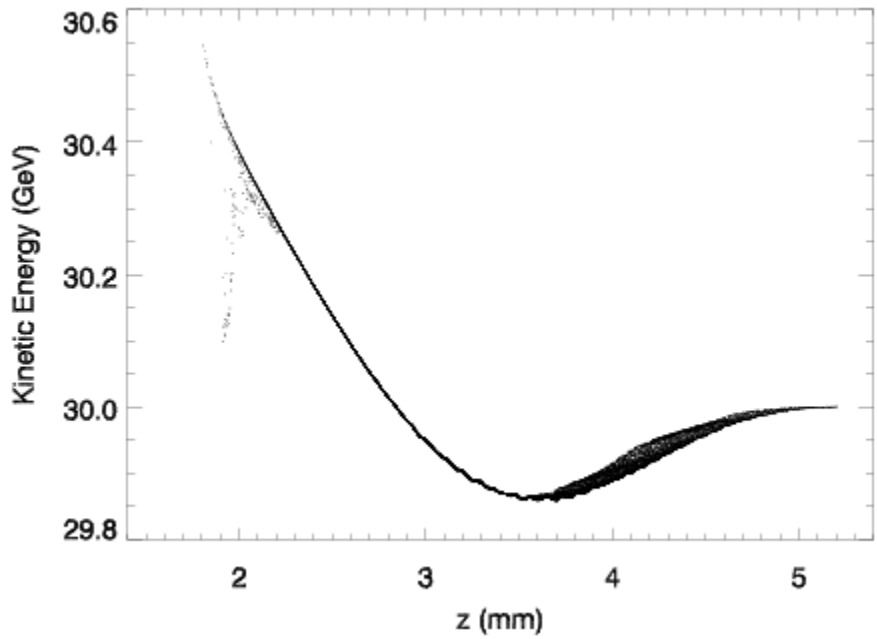

FIGURE 12. The energy distribtuion of the beam electrons, after 100 cm of propagation through the lithium plasma, showing that particles in the bulk of the beam lose up to 150 MeV, while particles in the tail of the beam gain as much as 550 MeV.

Modeling Electron-Neutral Collisions

XOOPIC uses the null collision method[40] for MCC (Monte Carlo collision) treatment of electron-impact excitation and ionization and for electron-neutral elastic scattering. MCC models for Ar, Ne, He and H have been used for some time, but the original cross section and scattering models assumed the impact energy to be nonrelativistic. Fully relativistic models are required for the study of plasma accelerator and plasma lens problems.

The total cross sections $\sigma(E)$, as originally implemented in XOOPIC for electron-neutral collisions, fall from their maximum value like $ln(E)/E$ as the impact energy E increases. This behavior is correct for E<200 KeV.[53,54] However, relativistic effects break this scaling, leading to a minimum[55,56] in $\sigma(E)$ for E~1 MeV. As E increases beyond 1 MeV, $\sigma(E)$ grows logarithmically, until it eventually saturates at a density dependent energy (the Fermi plateau).[56-59]

Reiser developed a simple fitting function for impact ionization cross sections,[60] using the ionization energy and two adjustable parameters, approximately capturing both low-energy and relativistic behavior (but not the Fermi plateau). The fitting parameters for this previous work were determined largely by data for impact energies near 1 MeV, which has been published for a number of gasses[55] (although not lithium). We found this approach to be inaccurate for energies as high as 30 GeV.

We have recently developed and implemented an improved parametric model for the total impact ionization cross section at energies ranging from the ionization threshold energy up to 100 GeV, which will be published elsewhere. Figure 13 shows our parametric fits

to electron-impact cross sections for lithium, with an energy range 0.1 eV < E < 10 GeV. The solid line is for ionization of neutral lithium (ejecting the outer electron from the 2s shell). The dashed line is for elastic scattering. The fitting parameters for ionization were determined from Ref. 61 (low energy) and Ref. 62 (high energy). The fitting parameters for elastic scattering were determined from Ref. 63 (low energy) and Ref. 62 (high energy).

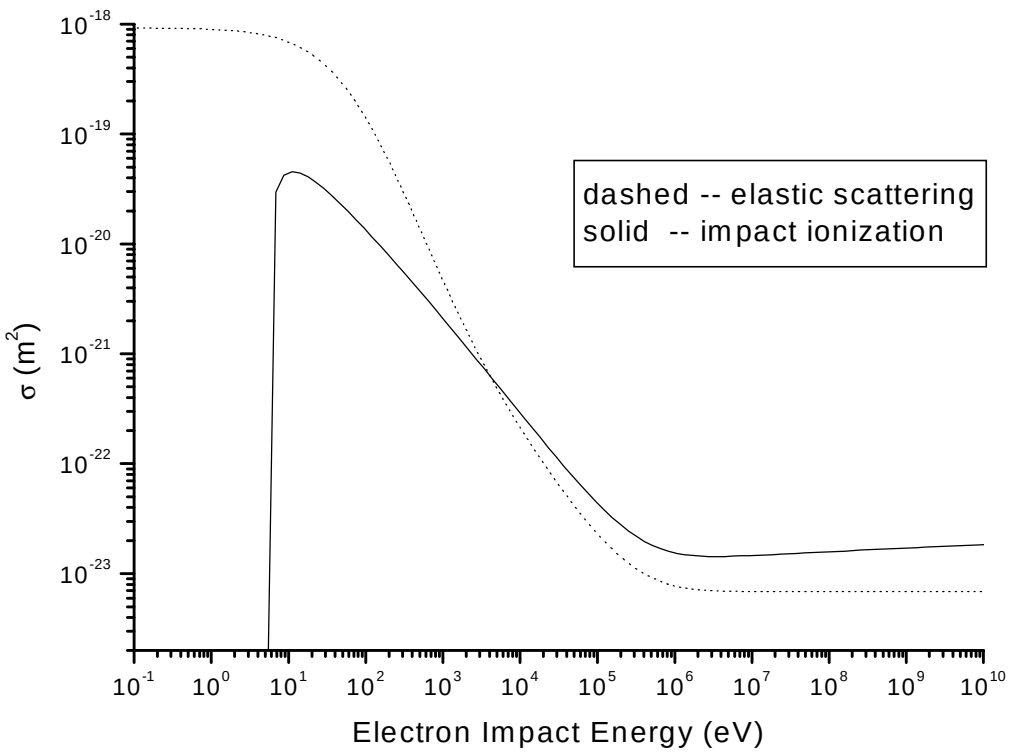

FIGURE 13. Lineout along the axis of cylindrical symmetry (r=0) of the longitudinal electric field E_z shows the large accelerating fields generated by the plasma wake behind the 30 GV/m electron beam. The noise seen to the right of the figure is found only along the symmetry axis and does not grow in time. The peak field value of –0.8 GV/m overlaps the tail of the electron beam distribution.

The original XOOPIC model for the energy distribution of the secondary electrons is also nonrelativistic. A number of parametric functions have been developed to describe the energy distribution of secondary electrons at low[64-66] and moderately high[67] energies. The theory for very high energies is well estabilished.[68-70] We have developed and implemented a new parametric model which agrees well with low energy models and also with high-energy theory, in the appropriate limits.

The original XOOPIC models for scattering of primary and secondary electrons, and for elastic scattering, were also nonrelativistic. Several works discuss the angular distribution for elastic and inelastic scattering in the nonrelativistic and relativistic regimes.[57,65,66,70-72] Unfortunately, the literature on scattering is split between the nonrelativistic and relativistic limits, with little discussion of the overlap regime. We have developed new parametric models for the scattering due to both impact ionization

and elastic collisions, which have the correct forms at both energy extremes and transition sensibly from the low energy to the high energy regime. A detailed discussion of this work is in preparation and will be published elsewhere.

Simulations of the E-157 experiment, including the effects of impact ionization, indicate very little effect on the resulting wakefield. However, the recently proposed plasma afterburner concept requires 100 times higher lithium plasma density[29] than is being used for E-157. Preliminary simulations with afterburner parameters indicate that electron-impact ionization will not significantly affect the wake, but that some secondary electrons could be accelerated to very high energies. Furthermore, the new ionization models are being used to simulate beam-induced ionization effects in the E-150 plasma lens experiments.[73] These studies will be the subject of future publications.

Summary

The code XOOPIC has been modified to enable particle-in-cell simulations of plasma based accelerators and plasma lenses, including the effects of relativistic electron-neutral collisions. These modifications include the development of a moving window algorithm, adding a new electromagnetic pulse launcher, generalization of the particle beam emitters, and further optimization to allow efficient use on parallel platforms. Also, new relativistic parametric models for impact ionization and elastic scattering, including the total cross section, the energy distribution of secondary electrons, and the scattering angles, have been developed and implemented for lithium and nitrogen.

To demonstrate the utility of XOOPIC, we presented the results of 2-D slab simulations of the standard LWFA, with both low and high intensity laser pulses, showing agreement with the theoretically predicted wake amplitudes and the results of fluid simulations. In addition, 2-D cylindrical simulations of the PWFA concept were performed, showing good agreement with previous studies for the parameters of the E-157 experiment.

Acknowledgements

The authors gratefully acknowledge many helpful and interesting conversations with C.K. Birdsall, P. Chen, R. Hemker, R. Hubbard, C. Joshi, T. Katsouleas, N. Mokhov, W. Mori, J. Ng, S. Parker, P. Stoltz and F. Tsung. This work is supported by the U.S. Department of Energy, under Contract No.'s DE-FG03-99ER82903, DE-FG03-95ER40926 and DE-AC03-76SF00098, and by Tech-X Corporation. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098.

Appendix: Object Oriented Programming

Over the last fifteen years, the commercial software industry has fully adopted the object oriented programming (OOP)[74-78] paradigm. OOP provides a superior mechanism for rapid development and testing, as well as ease of maintenance and extensibility for scientific codes. Although there is an initial overhead to an object oriented implementation, a favorable scaling over conventional procedural techniques is achieved for medium sized and large codes. For these reasons, OOP is becoming more widely accepted among computational scientists.

OOP consists of designing *classes*, where a class is a format for holding and interacting with data. An *object* is a particular instance of a class, just as 3.1 is a particular instance of a real number. Each object contains its own data, while all objects of a given class share the same functions or *methods*. A program is constructed by first defining the classes. Next, the program instantiates objects corresponding to the classes and, by calling the public methods of these objects, manipulates or displays the data. The set of public methods for a class is called its *interface*.

The three defining properties of OOP are *encapsulation*, *inheritance* and *dynamic binding*. The use of inheritance and dynamic binding together makes possible the use of *polymorphism*, which is sometimes listed in place of dynamic binding as the third defining principle of OOP. Encapsulation refers to the fact that objects are accessed only through a public interface, while their internal data and implementations remain hidden. Inheritance allows the programmer to define new classes that inherit most of their coding from existing classes, only modifying or adding data and methods as needed. This feature provides the flexibility to increase the capability of an application in ways not foreseen by the original programmers, without completely rewriting the code. Dynamic binding means that different classes can support the same interface but fulfill the requests differently at run time, through the use of function overloading. Polymorphism is the use of a derived class instance within a section of code that knows only the interface of the top-level or base class.

A number of programming languages provide varying degrees of support for OOP. Procedural languages such as Fortran 77 provide very limited support and tend to result in monolithic codes which, over time, become difficult to debug and enhance. Fortran 90 is a language designed for scientific applications and provides partial support of the object oriented model, including abstraction, encapsulation, and function overloading. While C is a procedural language, it can with effort be used to develop object oriented software, as has been demonstrated by the PETSc[79] code, developed at Argonne National Laboratory. C++ combines full support of the OOP paradigm, with the ability to achieve high-performance, as well as complete compatibility with C.

Well-designed C++ applications can achieve all the many benefits of OOP, with only a minor overhead in terms of executable size and run-time as compared to procedural C code. This requires that small-scale objects be avoided in computationally intensive parts of the code, that virtual functions are not used inside large loops, and that data structures

are designed to make effective use of cache memory. This is the approach taken in the XOOPIC code.[39]

The use of pointers in C/C++ can restrict some types of optimization that are available to Fortran compilers, like the reordering of operations inside a loop. This sometimes gives optimized Fortran 77 a slight speed advantage over optimized C code. A fairly recent development in C++ is template meta-programming, which uses compile-time techniques to generate code that can be faster by an order of magnitude than optimized Fortran. The Matrix Template Library[80,81] is an example of this approach.

References

1. E. Esarey *et al.*, IEEE Trans. Plasma Science **24**, p. 252 (1996).
2. D. Umstadter *et al.*, Phys. Rev. Lett. **76**, p. 2073 (1996).
3. R.G. Hemker *et al.*, Phys. Rev. Lett. **57**, p. 5920 (1998).
4. E. Esarey, R.F. Hubbard, W.P. Leemans et al., Phys. Rev. Lett. **79**, p. 2682 (1997).
5. C.B. Schroeder *et al.*, Phys. Rev. E **59**, p. 6037 (1999).
6. E. Esarey *et al.*, Phys. Plasmas **6**, p. 2262 (1999).
7. C.I. Moore *et al.*, Phys. Rev. Lett. **82**, p. 1688 (1999).
8. B.J. Duda *et al.*, Phys. Rev. Lett. **83**, p. 1978 (1999).
9. N.E. Andreev et al., Phys. Rev. Special Topics – Accel. & Beams **3**, 021301 (2000).
10. A. Modena, Z. Najmudin, A.E. Dangor *et al.*, Nature **377**, p. 606 (1995).
11. K. Nakajima, D. Fisher, T. Kawakubo *et al.*, Phys. Rev. Lett. **74**, p. 4428 (1995).
12. C. Coverdale, C.B. Darrow, C.D. Decker *et al.*, Phys. Rev. Lett. **74**, p. 4659 (1995).
13. R. Wagner *et al.*, Phys. Rev. Lett. **78**, p. 3125 (1997).
14. A. Ting, C.I. Moore, K. Krushelnick *et al.*, Phys. Plasmas **4**, p. 1889 (1997).
15. D. Gordon, K.C. Tzeng, C.E. Clayton *et al.*, Phys. Rev. Lett. **80**, p. 2133 (1998).
16. P. Volfbeyn, E. Esarey, and W.P. Leemans, Phys. Plasmas **6**, p. 2269 (1999).
17. C.D. Decker, W.B. Mori and T. Katsouleas, Phys. Rev. E **50**, p. R3338 (1994).
18. C.D. Decker *et al.*, Phys. Plasmas **3**, p. 2047 (1996).
19. S.V. Bulanov *et al.*, Phys. Fluids B **4**, p. 1935 (1992).
20. S.V. Bulanov, F. Pegoraro and A.M. Pukhov, Phys. Rev. Lett. **74**, p. 710 (1995).
21. N.E. Andreev *et al.*, JETP Letters **55**, p. 571 (1992).
22. J. Krall, A. Ting, E. Esarey and P. Sprangle, Phys. Rev. E **48**, p. 2157 (1993).
23. T.M. Antonsen and P. Mora, Phys. Fluids B **5**, p. 1440 (1993).
24. B.A. Shadwick, G.M. Tarkenton, E. Esarey, and W.P. Leemans, in *Proc. Advanced Accel. Concepts Workshop,* 2000 (to be published).
25. P. Mora and T.M. Antonsen, Phys. Plasmas **4**, p. 217 (1997).
26. P. Sprangle, E. Esarey and A. Ting, Phys. Rev. Lett. **64**, p. 2011 (1990).
27. P. Chen, J. Dawson, R. Huff and T. Katsouleas, Phys. Rev. Lett. **54**, p. 693 (1985).
28. R. Ruth, A. Chao, P. Morton, and P. Wilson, Part. Accel. **17**, p. 171 (1985).
29. T. Katsouleas *et al.*, in *Proc. Advanced Accel. Concepts Workshop,* 2000 (to be published).
30. R. Assmann *et al.*, in *Proc. 1999 Part. Accel. Conf.*, edited by A. Luccio and W. MacKay, (IEEE, Piscataway, NJ, 1999), p. 130.

31. P. Muggli *et al*., in *Proc. 1999 Part. Accel. Conf.*, edited by A. Luccio and W. MacKay, (IEEE, Piscataway, NJ, 1999), p. 3651.
32. S. Lee *et al*., Phys. Rev. E **61**, p. 7014 (2000)
33. M.J. Hogan *et al*., Phys. Plasmas **7**, p. 2241 (2000).
34. R. Govil, S.J. Wheeler and W.P. Leemans, in *Proc. 1997 Part. Accel. Conf.*, edited by M.K. Craddock and M. Reiser, (IEEE, Piscataway, NJ, 1999), p. 654.
35. J. Ng *et al*., SLAC Preprint, SLAC-PUB-8501 (2000).
36. P. Chen *et al*., in *Proc. Advanced Accel. Concepts Workshop*, 2000 (to be published).
37. C.K. Birdsall and A.B. Langdon, *Plasma Physics via Computer Simulation* (McGraw-Hill, New York, 1985).
38. R.W. Hockney and J.W. Eastwood, *Computer Simulation Using Particles*, (Institute of Physics Publishing, Bristol, 1988).
39. J.P. Verboncoeur *et al.*, J. Comp. Phys. **104**, p. 321 (1993).
40. V. Vahedi and M. Surendra, Comp. Phys. Comm. **87**, p. 179 (1995).
41. C.K. Birdsall, in *Plasma Processing of Semiconductors*, edited by P.F. Williams (Kluwer Academic Publishers, Amsterdam, 1997), p. 277.
42. J.P. Verboncoeur *et al.*, Comp. Phys. Comm. **87**, p. 199 (1995).
43. V. Vahedi *et al.*, in *Proc. 14th Conf. on Numerical Simulation of Plasmas*, 1991.
44. P.S. Pacheco, *Parallel Programming with MPI*, (Morgan Kaufmann, San Francisco, 1997).
45. R. Govil *et al.*, Phys. Rev. Lett. **83**, p. 3202 (1999).
46. G. Joyce, J. Krall and S. Slinker, Laser and Particle Beams **12**, p. 273 (1994).
47. K.C. Tzeng, W.B. Mori and C.D. Decker, Phys. Rev. Lett. **76**, p. 3332 (1996).
48. R.G. Hemker *et al*., in *Proc. 1999 Part. Accel. Conf.*, edited by A. Luccio and W. MacKay, (IEEE, Piscataway, NJ, 1999), p. 3672.
49. W.P. Leemans *et al.*, Phys. Plasmas **5**, p. 1615 (1998).
50. W.P. Leemans *et al*., in *Proc. Advanced Accel. Concepts Workshop*, 2000 (to be published).
51. R.E. Giacone *et al.*, in *Proc. Seventh European Part. Accel. Conf.*, 2000, p. 907.
52. J. Villasenor and O. Buneman, Comp. Phys. Comm. **69** (1992), p. 306.
53. S.M. Younger and T.D. Märk, in *Electron Impact Ionization*, edited by T.D. Märk and G.H. Dunn, (Springer Verlag, Vienna, 1985), pp. 24-41.
54. S.C. Brown, *Basic Data of Plasma Physics, The Fundamental Data on Electrical Discharges in Gases*, (American Institute of Physics, New York, 1994).
55. F.F. Rieke and W. Prepejchal, Phys. Rev. A **6**, p. 1507 (1972).
56. J.D. Jackson, *Classical Electrodynamics*, (Wiley, New York, 1975), pp. 618-653.
57. E. Fermi, Phys. Rev. **57**, p. 485, 1940.
58. B. Rossi, *High-Energy Particles*, (Prentice Hall, New York, 1952), pp. 27-29.
59. J.H. Cobb *et al*., Nucl. Instrum. and Meth. **133**, p. 315 (1976).
60. M. Reiser, *Theory and Design of Charged Particle Beams*, (Wiley, New York, 1994) pp. 273-278.
61. S.M. Younger, J. Quant. Spectr. Rad. Transf. **26**, p. 329 (1981).
62. S.T. Perkins *et al.*, LLNL report UCRL-50400, Vol. **31** (1991), unpublished.
63. I. Bray, D.V. Fursa and I.E. McCarthy, Phys. Rev. A **47**, p. 1101 (1993).
64. C.B. Opal, W.K. Peterson and E.C. Beatty, J. Chem. Phys. **55**, p. 4100 (1971).

65. M.E. Rudd, Phys. Rev. A **44**, p. 1644 (1991).
66. M.E. Rudd *et al.*, Phys. Rev. A **47**, p. 1866 (1993).
67. S.P. Slinker, R.D. Taylor and A.W. Ali, J. Appl. Phys. **63**, p. 1 (1988).
68. C. Moeller, Annalen der Physik **14**, p. 531 (1932).
69. H.A. Bethe and J. Ashkin, in *Experimental Nuclear Physics*, Vol. 1, edited by E. Segrè, (John Wiley & Sons, New York, 1953), pp. 166-357.
70. A.N. Kalinovskii, N.V. Mokhov, and Y.P. Nikitin, *Passage of High-Energy Particles through Matter*, (American Institute of Physics, New York, 1989), pp. 29-54.
71. N.F. Mott and H.S.W. Massey, *The Theory of Atomic Collisions*, 2nd Edition, (Oxford University Press, Oxford, 1949).
72. H. Goldstein, *Classical Mechanics*, (Addison-Wesley, Reading, 1980), p. 309.
73. D.L. Bruhwiler *et al.*, Bull. Am. Phys. Soc. **45**, No. 7, p. 27 (2000).
74. G. Booch, IEEE Trans. on Software Eng. **12**, p. 211 (1986).
75. B. Stroustrup, *The C++ Programming Language*, (Addison-Wesley, Reading, 1998).
76. S.B. Lippman and J. Lajoie, *C++ Primer*, (Addison-Wesley, Reading, 1998).
77. E. Gamma, R. Helm, R. Johnson and J. Vlissides, *Design Patterns, Elements of Reusable Object-Oriented Software*, (Addison-Wesley, Reading, 1995).
78. A. Elins, *Principles of Object Oriented Software Development*, (Addison-Wesley, Reading, 1994).
79. S. Balay, W.D. Gropp. L.C. McInnes and B.F. Smith, in *Modern Software Tools in Scientific Computing*, edited by E. Arge *et al*., (Birkhauser Press, 1997), pp. 163-202.
80. J.G. Siek and A. Lumsdaine, Computing in Science and Engineering, Vol. **1**, No. 6 (1999), p. 70.
81. Matrix Template Library home page at URL http://www.osl.iu.edu/research/mtl/