

DEVELOPMENT OF A PARALLEL CODE FOR MODELING PLASMA BASED ACCELERATORS

R.G.Hemker, F.S.Tsung, V.K.Decyk, W.B.Mori, UCLA, Los Angeles, CA
 S.Lee and T.Katsouleas, USC, Los Angeles, CA

Abstract

Advances in computational speed and memory make it now possible to do full scale 2D and 3D PIC simulations of laser and beam plasma interactions. However, the increased complexity of these codes and interactions make it necessary to apply modern programming approaches like object oriented frameworks to the development of these codes. We report here on our progress in developing the object oriented parallel PIC code OSIRIS that is implemented in Fortran 90. In its current state, the code can be used for 1D, 2D, and 3D simulations in Cartesian coordinates and for 2D simulations in cylindrically symmetric coordinates. We will present benchmarks of simulations done with this code for a proposed plasma wake field accelerator experiment [1,2].

1 INTRODUCTION

The introduction of massively parallel computers with hundreds of processors provides now the computational speed and the memory needed to do full scale 2D and 3D simulation of plasma based accelerators and other problems in laser-plasma and beam-plasma physics. In order to take full advantage of the possibilities of this development it has become necessary to use more complex simulation codes. The increased complexity of codes arises for two reasons. One reason is that the realistic simulation of a problem requires a larger number of more complex algorithms interacting with each other than the simulation of a rather simple model system. An example would be that initializing an arbitrary laser or plasma beam in 3D is a much more difficult problem than doing the same in 1D or 2D. The other reason that simulation codes are becoming more complex is that the computer systems are more complex and the performance obtained from them can dramatically differ depending on the code strategy. Parallelized codes that have to handle the problems of parallel communication and parallel IO are an example of this. A way to deal with this increased complexity is to use an object oriented programming style, which divides the code and data structures into independent classes of objects. This programming style maximizes code reusability and reliability.

2 DESCRIPTION OF THE CODE AND CODE DEVELOPMENT

The goal of the code development program was to create a code that breaks up the large problem of a simulation into a set of essentially independent smaller problems that can

be solved separately from each other. Object oriented programming achieves this by handling different aspects of the problem in different modules (classes) that communicate through well-defined interfaces. The programming language we chose for this purpose was Fortran 90, mainly because it allow us to more easily integrate already available Fortran algorithms into the new OSIRIS-framework. In its current state, the code contains algorithms for 1D, 2D, and 3D simulations in Cartesian coordinates and for 2D simulations in cylindrically symmetric coordinates. For all of these algorithms the code is fully relativistic and presently uses a charge-conserving current deposition algorithm. It allows for a moving simulation window and arbitrary domain decomposition for any number of dimensions. Benchmarking of the code has so far shown a slowdown of less than 15% compared to other codes using similar algorithms due to the additional overhead that is a side effect of the object oriented implementation.

Figure 1 shows the class hierarchy of OSIRIS. The main physical objects used are particle objects, electromagnetic field objects, and source field objects (for currents and densities). The particle object is an aggregate of an arbitrary number of particle species objects. The most important support classes are the variable-dimensionality-field class, which is used by the electromagnetic and source field class and encapsulates many aspects of the dimensionality of a simulation, and the domain-decomposition class, which handles all communication between nodes. For easy portability of the code to different architectures all code that is machine dependent is encapsulated in the system-, the file-system-, and the utility-module.

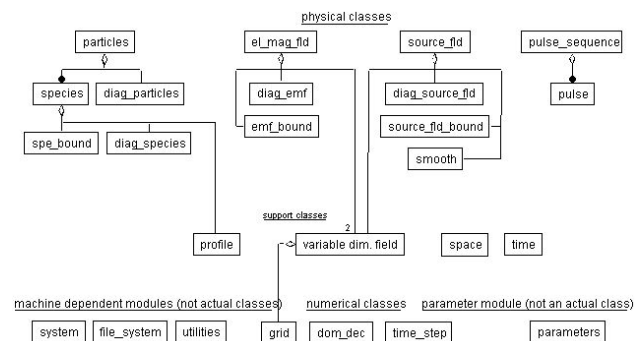


Figure 1: The class hierarchy of OSIRIS

The code was first developed for 1D and 2D simulations in Cartesian coordinates, which took about a year of time.

After this fundamental work was completed, extending the code to 2D cylindrically symmetric coordinates took about four weeks and implementation of 3D simulations took about a week. This short time required for the extension to cylindrically symmetric 2D and 3D simulations was possible because the object-oriented style of the code allowed for two specific features. First, most of the code does not make any explicit use of the dimensionality or the coordinate system. Secondly, all communication between the physical domains assigned to the different nodes of a parallel computer is handled as a part of the boundary conditions of the physical domains assigned to each node. This handling of node-to-node communication allows it to consider almost all algorithms as single-node algorithms. Once a single node-algorithm is implemented in OSIRIS, it is automatically already parallelized since it can take advantage of the already existing boundary conditions.

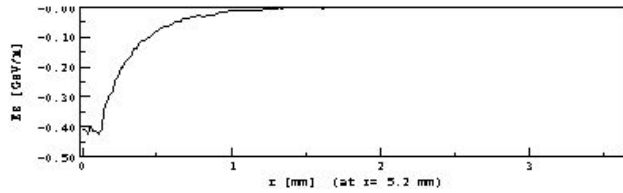


Figure 2: E_z lineout as a function of r at a position of 5.2mm from the back of the simulation box (see figure 4) after a beam propagation of 1m

3 SIMULATION RESULTS

We have benchmarked the 2D cylindrical algorithms of OSIRIS against previously performed simulations [3] of a plasma wakefield accelerator. In this model problem a 30 GeV electron bunch of 4×10^{10} electrons with Gaussian profiles ($\sigma_z=0.6\text{mm}, \sigma_r=75\mu\text{m}$) and an emittance of $\epsilon_N = 15 \pi$ mm mrad will propagate through 1m of an underdense plasma with a density of $n=2.1 \times 10^{14} \text{cm}^{-3}$. The simulation of the problem was done on a 500×200 grid with 9 particles per cell for the background plasma and 25 particles per cells for the beam. The simulation area, which moved with the beam because of the moving window, had an extension along the axis of $25 c/\omega_p$ and a radial extension of $10 c/\omega_p$. At the density of $2.1 \times 10^{14} \text{cm}^{-3}$ this is $9.2\text{mm} \times 3.7\text{mm}$. This grid lead to a grid cell size of $0.05c/\omega_p \times 0.05c/\omega_p$ and therefore a time step of $0.02\omega_p^{-1}$ was chosen. The simulation was done on 10 nodes of a Cray T3E and took about 2 1/2 days of turnaround time for the 136000 required time steps. By comparison, another code used before for similar simulations took about 19 days to finish the task[4].

Since the plasma density is less than the beam density a total blowout of the plasma electrons in the wake of the electron bunch, creating a positively charged ion column, is expected. The simulation results clearly support this. Figure 2 shows a lineout of E_z as a function of r . The plot

shows the flattening of the accelerating field E_z close to the axis, characteristic of the blowout regime.

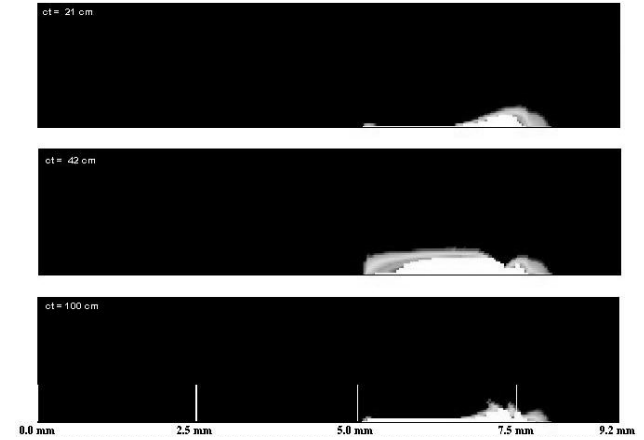


Figure 3: Charge density plot of the beam at three different propagation distances. The betatron oscillations are clearly visible.

The blowout should also lead to betatron oscillations of the middle and tail part of the beam with a wavelength given by

$$\lambda_\beta/2 = \pi \sqrt{\frac{\gamma mc^2}{2\pi n_0 e^2}} \quad (1)$$

Figure 3 shows a charge density plot of the beam at different times. The upper and middle plot are the minimum and maximum of the oscillation at the propagation distances where they would be expected according to equation (1). The lowest plot shows the beam at the end of the simulation after propagating for 1m.

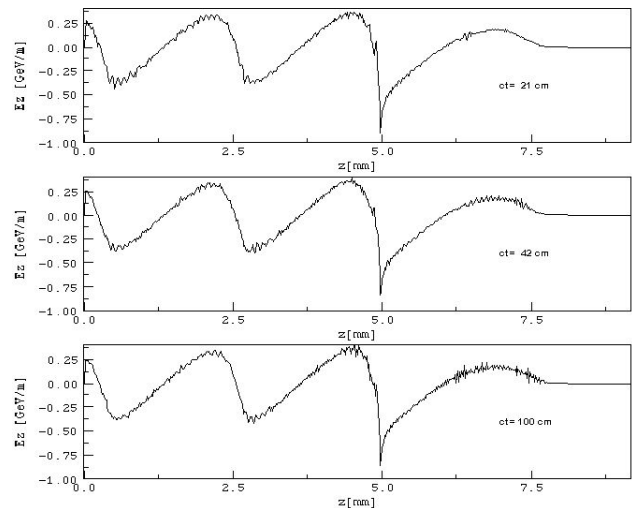


Figure 4: Lineout of the accelerating field along the axis at the first minimum and maximum of the betatron oscillations and at the final propagation distance of 1m

Together the density plots confirm the wavelength of the betatron oscillation given by equation (1) and therefore the underlying ideas about the beam-plasma interactions. It should also be noted that as expected the head of the beam, where the electrons are not fully blown out, oscillates with a longer period that depends on the axial position within the beam.

Figure 4 shows lineouts of the accelerating electric fields along the axis at the same propagation distances as the density plots in figure 2. The simulation results indicate that the magnitude of the peak accelerating field does not vary by more than about 5 -10% during the beam propagation despite of betatron oscillations.

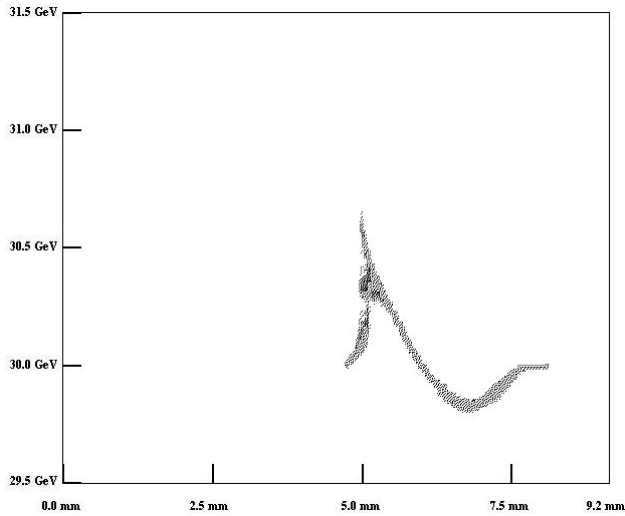


Figure 5: Plot of the energy of the beam particles as a function of position along the axis after 1m of propagation

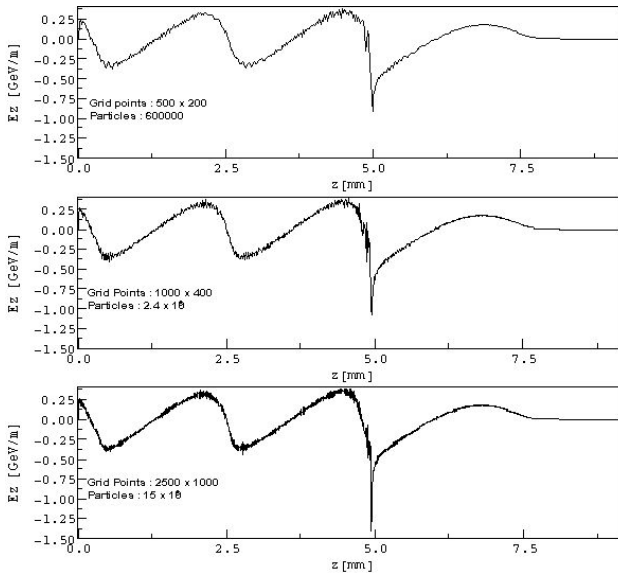


Figure 6: Outline of the accelerating field along the axis after 3 cm of propagation for three simulations with different numbers of grid cells and particles.

Figure 5 shows the energy of the beam particles at the end of the simulation. There are three important features that should be noted in this plot. The front part of the beam did not change energy, the middle part of the beam lost about 200 MeV, and particles in the tail of the beam gained energies up to 700 MeV. These results are mostly consistent with previously presented calculations and simulations of this system [3].

We also did simulations for the same physical setup but with a larger number of grid cells and particles and therefore with a larger numerical accuracy. Figure 6 shows the lineouts of the accelerating electric fields along the axis for the original as well as two higher resolution simulations after 3 cm of propagation. The results differ significantly only in that the peak accelerating field increases with the resolution of the simulation. This leads to the conclusions that the 2D cylindrically symmetric simulations describe most aspects of the PWFA well but that it is hard to get a reliable estimate of the peak field and therefore the maximum obtainable particle energies with this algorithm. Full 3D simulations are probably required to clarify this question [2].

4 CONCLUSIONS

The object oriented framework OSIRIS for laser-plasma and beam-plasma physics has been developed and successfully used for simulations of plasma wake field accelerators. These simulations support previous research results. The code will be used for future research in this area as well as in other areas. New modules adding new physics to the framework will be implemented as our research motivates it. Due to the object oriented programming style these new modules will be able to immediately take full advantage of all the other features already implemented.

Work supported in part by DOE, NSF, and LLNL

5 REFERENCES

- [1] R. Assmann et al., paper WEBL6 of these proceedings;
- [2] Modeling of E-157 is done in collaboration with E. Esarey at LBNL and C. Schroeder at UC Berkeley;
- [3] R. Assmann et al., "Proposal for a GeV Plasma Wakefield Accelerator", SLAC-Pub, 1997;
- [4] S. Lee in Proc. AACW, Baltimore (AIP Conf. PROC. 1998);