

Three-Dimensional Simulations for Accelerator Physics Using ARGUS*

A. Mondelli, A. Mankofsky, J. Petillo, W. Krueger, C. Kostas, and A. Drobot

Science Applications International Corporation, McLean, VA 22102

and

R. Ryne, R.K. Cooper, G. Rodenz, and M.J. Browman

Los Alamos National Laboratory, Los Alamos, NM 87545

Abstract

ARGUS is a three-dimensional, electromagnetic, particle-in-cell (PIC) simulation code that uses a modular code architecture to allow multiple physics modules to share common utilities for grid and structure setup, memory management, disk I/O, differential vector field operators, and diagnostics. The code includes modules for electrostatic and electromagnetic field solutions, electromagnetic eigenvalue problems, and PIC simulation in either a steady-state or a time-dependent algorithm. SAIC and LANL (Accelerator Code Group) are collaborating to release ARGUS to the U.S. accelerator community. The code is available, with documentation, both at LANL and at NERSC (LLNL). Applications of ARGUS to waveguide and cavity problems and wake-field calculations are presented.

I. DESCRIPTION OF ARGUS

The ARGUS code [1], [2] has been under development at SAIC since 1983. The code architecture is designed specifically to handle the problems associated with three-dimensional electromagnetic, PIC simulations. It uses sophisticated memory management and data handling techniques [3] to deal with the large volume of data that is generated in such simulations.

A modular architecture is employed so that all physics modules share code utilities in common. Complicated geometrical structures can be represented on the computational grid, and the grid can be non-uniform in all three directions. Cartesian and cylindrical coordinate systems have been implemented throughout ARGUS. Physics modules are in place to compute electrostatic and electromagnetic fields, the eigenmodes of rf structures, and PIC simulation in either a time-dependent mode or a steady-state (gun code) mode. The PIC modules include multiple particle species, relativistic particle dynamics, and algorithms for creation of particles by emission from material surfaces and by injection onto the grid. A plasma chemistry module allows species to be created or destroyed based on specified rate processes.

Structure input in ARGUS is carried out using combinatorial geometry. The code stores a library of simple solid objects (e.g., a rectangular solid, an elliptical cylinder, an el-

lipsoid, etc.). These objects are combined by the user with logical operations (to either *add* or *delete* the library object) to produce objects of arbitrary shape. The structures so specified are represented on the computational grid by a *structure mask* array, which stores the material and electrical properties of each cell on the grid.

Material properties are associated with structures. The code allows perfectly conducting materials, as well as materials with complex values of both permittivity and permeability; hence lossy materials are allowed. Furthermore, the permittivity and permeability are specified as diagonal tensors to treat certain classes of non-isotropic materials.

ARGUS offers a wide variety of diagnostic plots that are selected by the user and are available at run time. In addition, the user can generate HDF files for exporting data to external visualization tools.

Version 25 of ARGUS is now available, with user support, through the Los Alamos Accelerator Code Group. This paper describes some examples from the ARGUS Primer, which is part of the ARGUS documentation.

II. TOMBSTONE CAVITY

A. Cold Test of the "Tombstone" Cavity

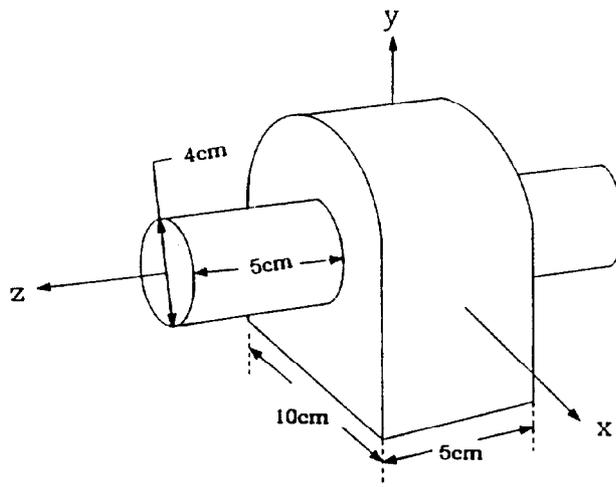


Figure 1. The "Tombstone" Cavity

* Work supported by the Department of Energy under contract number DE-AC05-91ER40625.

An example in the MAFIA User's Guide consists of analyzing the normal modes of the structure shown in Figure 1. The structure is a cylindrical drift tube attached to a cavity made of a rectangular box topped with a half cylinder. MAFIA and ARGUS represent this structure in different ways. ARGUS uses a "stairstep" representation on the grid, while MAFIA allows triangular "half cells".

A comparison of MAFIA and ARGUS cold-test results for the "Tombstone" cavity is given in Figure 2. The ten lowest modes of the structure were computed. The discrepancy is generally in the 1-2% range, and is due to the differences in the structure representation by the two codes. This result is consistent with other comparisons of these codes on comparable meshes. With finer meshing, the codes agree more closely.

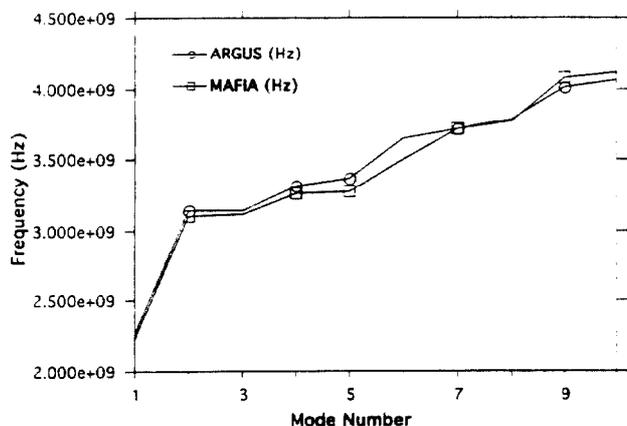


Figure 2. Comparison of ARGUS and MAFIA results for "Tombstone" cavity cold test.

B. Wake Fields in the "Tombstone" Cavity

The time-dependent PIC module in ARGUS can be used to compute wake fields by initializing a Gaussian bunch of heavy particles and allowing them to drift through the structure. This method has been employed to display the wake fields in the "Tombstone" cavity. Figure 3 shows three snapshots of the particles and axial electric field as the bunch traverses the structure. The Gaussian bunch is located on the axis of the cylindrical drift tube and launched into the cavity. The particles are visible in the figure as a line charge on axis. In the first frame, at time step 25, the particles are just entering the cavity from the lower drift tube. The middle frame, at time step 75, shows the particles as they leave the cavity and enter the upper drift tube. In the third frame, at time step 125, they are leaving the upper drift tube (by being absorbed on the upper boundary of the simulation).

III. RADIATING OPEN WAVEGUIDE IN THE TIME DOMAIN

The ability of the ARGUS code to handle open, radiating boundary conditions in the time domain sets it apart from several other codes. Two separate utilities exist for implementing this capability. The first is a "port" boundary condition, which treats a specified opening in the simulation boundary as though it were connected to a waveguide extending to infinity. The second is an implementation of the Lindman [4] algorithm for radiating boundaries. While the port condition only matches the boundary for radiation at a specified frequency, the Lindman condition is a general outgoing-wave boundary condition.

This simulation provides a demonstration of the Lindman algorithm for a rectangular waveguide driven at its lower end, and open at its upper end. Figure 4 shows the radiation fields at two time steps selected to show the phase slippage at the top (open) boundary.

IV. REFERENCES

- [1] A. Mankofsky, *Three-Dimensional Electromagnetic Particle Codes and Applications to Accelerators, Linear Accelerators and Beam Optics Codes*, C.R. Eminheizer, ed., A.I.P. Conf. Proc. No. 177 (American Institute of Physics, New York, 1988), p. 137ff.
- [2] A. Mondelli, C. Chang, A. Drobot, K. Ko, A. Mankofsky, and J. Petillo, *Application of the ARGUS Code to Accelerator Design Calculations*, Proc. 1989 IEEE Part. Accel Conf. (Chicago, IL, IEEE Cat. No. 89CH2669-0, 1989), p. 877.
- [3] A. Mankofsky, J.L. Seftor, C.L. Chang, K. Ko, A. Mondelli, A. Drobot, J. Moura, W. Aimonetti, S. Brandon, D. Nielsen, Jr., and K. Dyer, *Domain Decomposition and Particle Pushing for Multiprocessing Computers*, Comp. Phys. Commun. **48**, 155 (1988).
- [4] E.L. Lindman, "Free-Space" Boundary Conditions for the Time Dependent Wave Equation, J. Comp. Phys. **18**, 66 (1975).

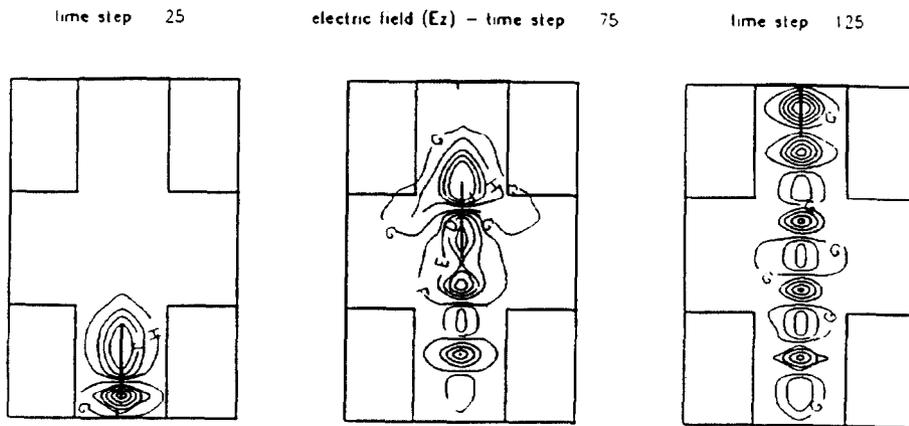


Figure 3. Wake fields due to a Gaussian bunch passing through the "Tombstone" Cavity.

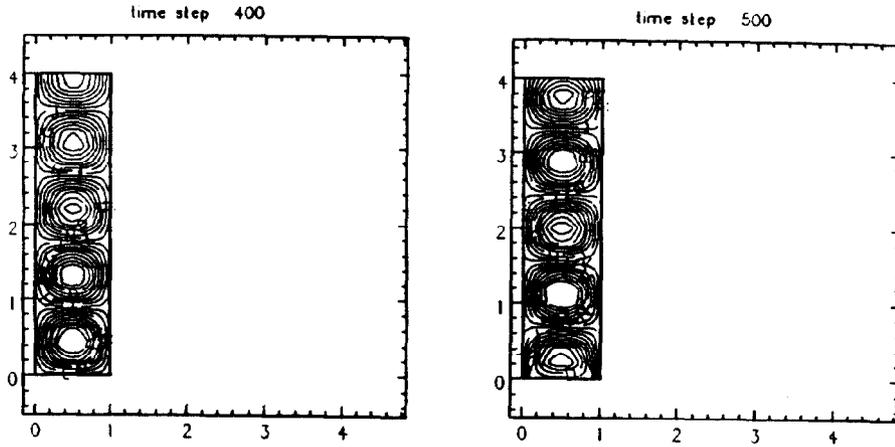


Figure 4. A radiating waveguide in the time domain.