

UCLA Parallel PIC Framework

Viktor K. Decyk

Department of Physics and Astronomy
University of California, Los Angeles
Los Angeles, California 90095-1547

Abstract

The UCLA Parallel PIC Framework (UPIC) is being developed to provide trusted components for the rapid construction of new, parallel Particle-in-Cell (PIC) codes. The Frameworks uses object-based ideas in Fortran95, and is designed to provide support for various kinds of PIC codes on various kinds of hardware. The framework under development is designed for student programmers. The framework will support multiple numerical methods, different physics approximations, different numerical optimizations and implementations for different hardware. It is designed with “defensive” programming in mind, meaning that it contains many error checks and debugging helps. Above all, it is designed to hide the complexity of parallel processing. The framework is designed with layers, with student programmers in mind. The lowest layer consists of highly optimized Fortran77 routines from UCLA’s 25 year legacy of PIC codes.

The middle layer primarily provides a much safer and simpler interface to the complex Fortran77 legacy subroutines by encapsulating many details, such as data layouts on parallel machines. It also provides polymorphism, and as a run-time option, verifies that required pre- and post-conditions are satisfied when calling the legacy code. It does not currently perform any demanding calculations, so high performance is not required. This middle layer could be rewritten in another object-based or object-oriented language that can call Fortran77, such as C++, or a scientific programming environment such as IDL, MATLAB, or Mathematica, without impacting the lower layers.

The upper layer consist of powerful high level classes that can easily be reused for those parts of the code which the students do not intend to modify. These high level classes provide simple interfaces that encapsulate the implementation details of a large block of code. These high level classes have various properties to enable it to respond to “messages” to perform some action. For example, in order for an object to be FFTable, it must encapsulate within itself pointers to appropriate fft tables. Typical properties include being writable for output or displayable on the screen.

The framework is currently being used by a number of new PIC codes. One such code is QuickPIC , a quasi-static code for studying plasma-based accelerators. It consists of a coupled 3D and 2D PIC code. Another code is QPIC, a 2D quantum PIC code using a semi-classical scheme based on integrating classical Feynman paths. A third code is a new version of BEPS , a code used in teaching plasma physics.