The UCLA Particle-in-Cell and Kinetic Simulation Software Center (PICKSC)

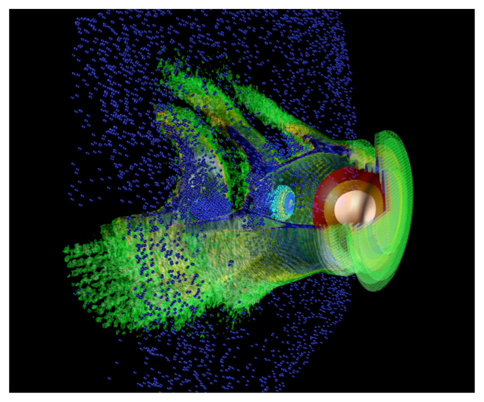
UCLA

PI: W.B. Mori; Co-PIs: V.K. Decyk, F.S.Tsung, R.E. Caflisch

International Partner: R.A. Fonseca IST and ISCTE Portugal

Some activities are also supported by DOE and the UCLA Institute for Digital Research and Education.

PICKSC Center Mission and Activities



The mission of the Particle-in-Cell and Kinetic Simulation Software Center (PICKSC) is to support an international community of PIC and plasma kinetic software developers, users, and educators, and to increase the use of PIC software for accelerating the rate of scientific discovery.

It aims to make available and document illustrative software programs for different computing hardware, a flexible framework for rapid construction of parallelized PIC programs, and several distinct production programs.

It will also include activities on developing and comparing different PIC algorithms and documenting best practices for developing and using PIC programs.

It will also develop educational software for undergraduate and graduate courses in plasma physics and computer science.

It will also sponsor an annual workshop to help build a community of developers and users. Visitors with specific needs are welcome.

Particle-in-Cell Codes

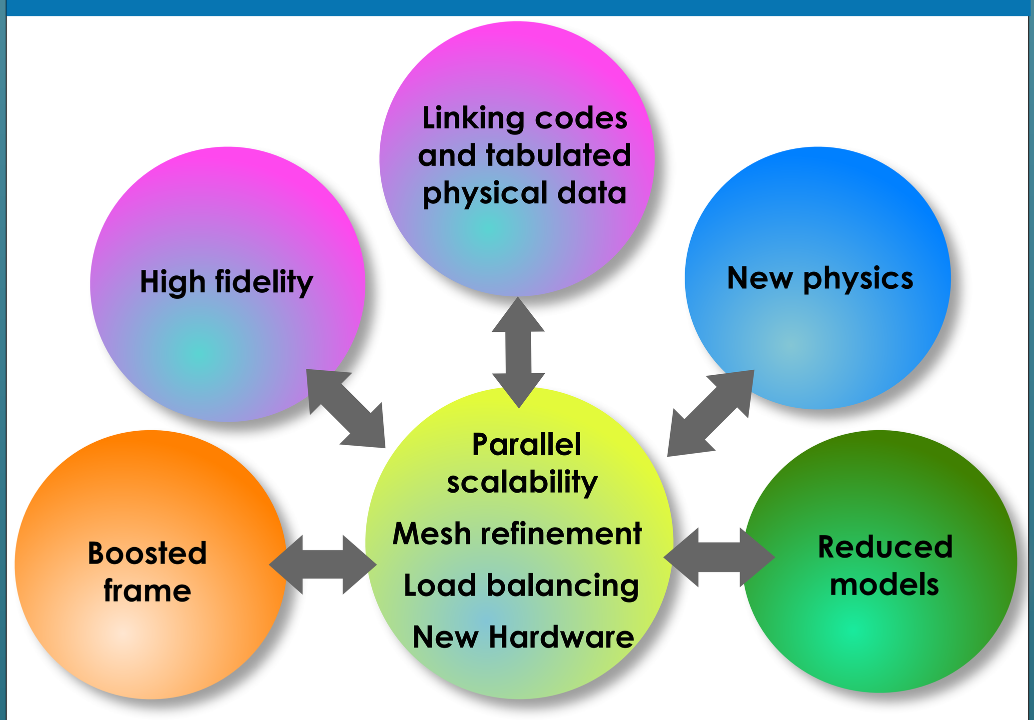
PIC codes integrate the trajectories of many particles interacting self-consistently via electromagnetic fields. They model plasmas at the most fundamental, microscopic level of classical physics.

PIC codes are used in almost all areas of plasma physics, such as fusion energy research, plasma accelerators, space physics, ion propulsion, plasma processing, and many other areas.

PIC codes are the most complete, but most computationally expensive models. Used when more simple models fail, or to verify the realm of validity of more simple models.

What distinguishes PIC codes from molecular dynamics codes is that a grid is used as a scaffolding to calculate fields rather than direct binary interactions => reduces calculation to order N rather than N^2.

Software Engineering of Complex Code Development is a Grand Challenge: Aim to develop skeleton codes, eduational codes, a Framework for rapid code development, and production codes



2014 PICKSC Workshop: Software Interoperability within the PIC Community

Invited major developers within the explicit PIC community
- This community shares ideas well, but not software
- There is no large community code
- For verification, reproduce results with independently developed codes

There was agreement on:
- Desire to easily reproduce results of a paper in hours, not months
- Desire for standard benchmark problems to verify or validate new codes
- Desire for standard inputs and outputs, to enable easy comparison
- Desire for code of ethics for sharing software
- Desire for standard unit tests to illustrate new algorithms

Barriers to code interoperability:
- Different units, languages
- Internals of codes not known or documented within community

2015 Workshop is being planned

PICKSC Codes

Open Source Skeleton PIC Codes for HPC (mini-apps)

Bare-bones but fully functional multi-D PIC codes illustrate different parallelization strategies
- Electrostatic and electromagnetic, with Darwin under construction
- Examples using MPI, OpenMP, SIMD Vectorization, CUDA, and hierarchies of these
- Examples with 0-3 levels of parallelism, from beginner level to HPC expert
- Each code in Fortran and C, with interoperability between languages
- 2,000-4,000 lines of code

Intended uses:
- Illustration of parallel PIC codes for beginning students
- Example of parallelization techniques for an irregular problem
- Non-trivial benchmarks for computer science research
- Provide a simple test environment for prototyping new ideas and algorithms
- Ideas can be mined to incorporate to other codes
- Can be expanded to a production code by adding diagnostics, initial conditions, and other boundary conditions

Excellent Performance
- Electromagnetic code, 0 levels of parallelism: ~ 110 nsec/particle/step on 1 CPU
- Electromagnetic code, 3 levels of parallelism: ~ 13 ps/particle/step on 96 GPUs

https://idre.ucla.edu/parallel-plasma-pic-codes

Open Source Interactive Educational PIC Codes for University Instruction

Laboratory experiments to illustrate important concepts in plasma physics:
- Plasma waves and Landau damping, Two stream instabilities, Wakes created by test charges, Plasma echoes, Parametric processes in plasma, Wave-breaking, Plasma fluctuations and particle discreteness effects, Relaxation processes with non-Maxwellian plasmas, Diffusion and drag in velocity space, etc.

Developing a graphical user interface to replace a command line driven code
- Usable from web as well as from individual laptops
- Goal is to make this code freely available to anyone studying plasma physics
- Solicit contributions from community to develop a library of "concept demonstrations"

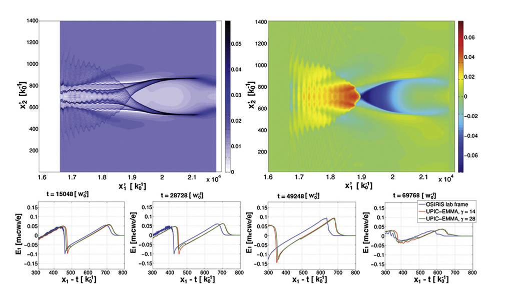
UPIC 1.0 Framework

Large library for constructing new parallel spectral PIC codes from trusted components
- MPI and pthreads parallelization
- Provides multiple generic main codes illustrating use of components (ES, EM, Darwin)
- Both procedural and Fortran90 object-based layers are available

Idea is to customize generic code by adding specialized features
- QuickPIC, UPIC-EMMA and other codes constructed this way

UPIC 2.0 Framework
- Will use OpenMP, MPI and SIMD vectorization
- Will be generalized to finite-difference as well as spectral components
- Will use Object-Oriented features of Fortran2003
- Will provide interoperability with C

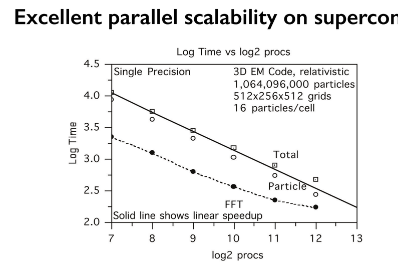
UPIC-EMMA

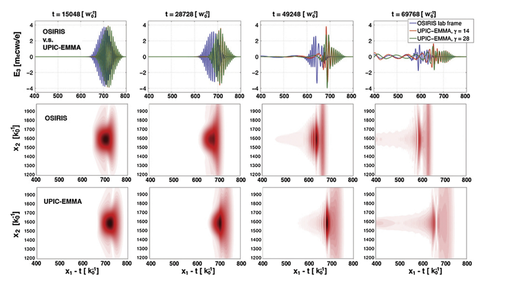


- Laser wakefield simulation in the Lorentz boosted frame with UPIC-EMMA, a spectral EM-PIC code.

- High fidelity simulation results compared with lab frame simulation, with 1000 time speed-up obtained using boosted frame technique.

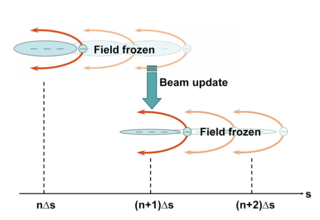
Excellent parallel scalability on supercomputer





QuickPIC

QuickPIC[1] is a 3D parallel Quasi-Static PIC code, developed based on the framework UPIC[2].



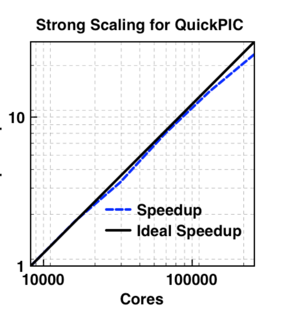
Quasi-Static Approximation

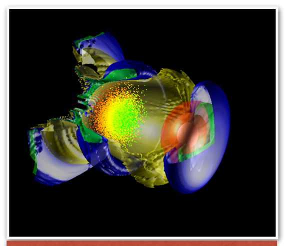
$(x, y, \xi=ct-z, s=z)$

$\partial_s << \partial_\xi$

Plasma: $(x, y; \xi)$

Beam: (x, y, ξ, s)





QuickPIC simulation of LWFA with a beam load

QuickPIC simulation of two-bunch electron-driven PWFA experiment at FACET[3] shown on the Nature cover.

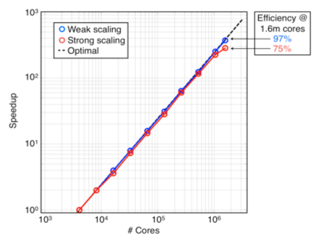
[1] W. An et al., J. Comp. Phys. 250, 165 (2013).
[2] V. K. Decyk, Comp. Phys. Comm. 177, 95 (2007).
[3] M. Litos et al., Nature 515, 92 (2014)

OSIRIS Framework

- Parallel, Relativistic & Explicit EM Particle-in-Cell (PIC) Code
- Visualization and Data Analysis Infrastructure
- Developed by the osiris.consortium

⇒ UCLA + IST

Weak and strong scaling results on the Sequoia supercomputer at LLNL. For weak scaling, OSIRIS is shown to be >97% efficient on > 1.5 million cores. For strong scaling, the code is 79% efficient on the full Sequoia machine.



OSIRIS simulation from the discovery of a laser-plasma acceleration mechanism that generates 20 MeV proton beams with a 1% spread, shown on the Nature cover.

Includes:
- Higher Order (smoother) Particle Shapes
- Binary Collision Module
- PML absorbing BC
- Dynamic Load Balancing
- Parallel I/O w/ new diagnostics such as Particle Tracking
- Cylindrical coordinates with azimuthal mode decomposition.
- Hybrid FFT-Finite difference solver

OSHUN

VLASOV

$$\frac{\partial f}{\partial t} + \mathbf{v}\cdot\frac{\partial f}{\partial \mathbf{r}} + q\left(\mathbf{E} + \frac{\mathbf{v}}{c}\times \mathbf{B}\right)\cdot\frac{\partial f}{\partial \mathbf{p}} = 0$$

- 2D3P relativistic + multispecies
- **Parallel:** decomposition in 2D configuration space
- **Explicit: Maxwell's equations** plasma and EM waves, instabilities, basic plasma physics
- *or implicit*: $J = \nabla \times B$ non-local transport, full target simulations

FOKKER-PLANCK

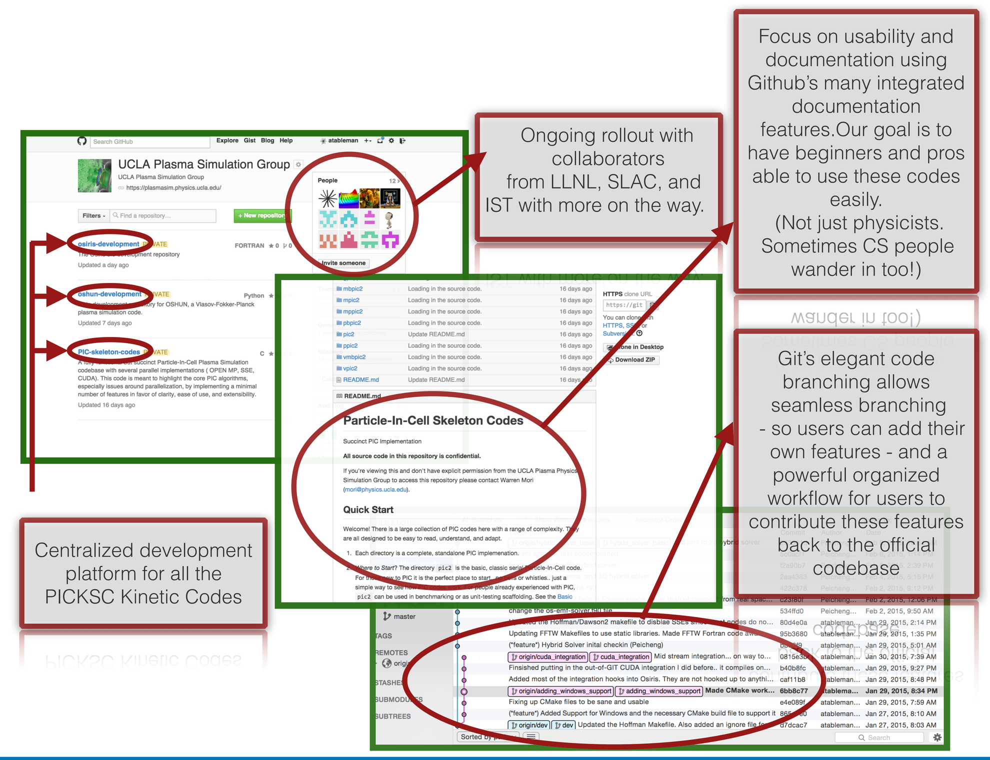
$$\left(\frac{\delta f_0^m(v)}{\delta t}\right)_{ee} = \frac{4\pi\Gamma_{ee}}{3}\frac{1}{v^2}\frac{\partial}{\partial v}\left[\frac{1}{v}\frac{\partial W(f_0^m(v),v)}{\partial v}\right]$$

- **Explicit**, nonlinear sub-cycling for the isotropic part of f(p)
- **Conserves** energy and number density
- **GPU** version

$$\frac{1}{\Gamma_{ee}}\left(\frac{\delta f}{\delta t}\right) = \frac{4\pi}{\mu} F f + \left(\frac{\mu-1}{\mu+1}\right)\nabla\mathbb{H}(F)\cdot\nabla f + \frac{\nabla\nabla\mathbb{G}(F):\nabla\nabla f}{2}$$

- **Implicit**, linear
- **GPU** version

PICKSC on GitHub



Web Interface for PIC codes

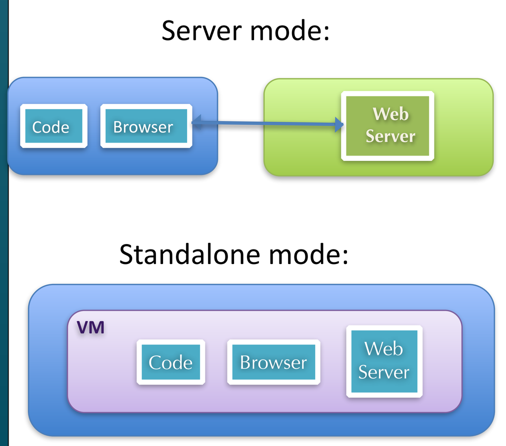
Production PIC codes can have many hundreds or even thousands of input parameters
- How can this be presented in a comprehensible way to users?
- Can we make the same interface usable by many PIC codes?

Approach: Properties of inputs defined in a hierarchical description XML file
- Allows inputs to be grouped in a 3 level hierarchy
- Can limit display to only those inputs that differ from default values
- Help available for each input

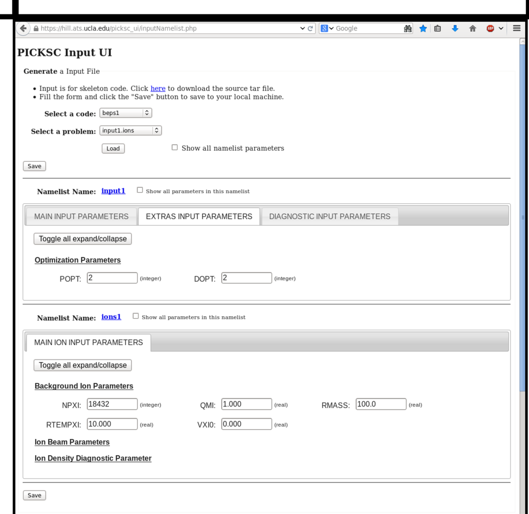
Initial implementation uses standard Fortran90 namelists with Apache and PHP
- Design is generic
- Multiple namelists describe highest grouping, one XML file for each namelist
- 2 level groupings within each namelist are possible
- Namelists not needed for a problem can be omitted and not displayed

Future work: different codes use different units
- Translate units for individual codes from standard units presented to user
- Public names may differ from actual names used in code
- Prevent appropriate or invalid inputs

Web GUI for local (laptop) run



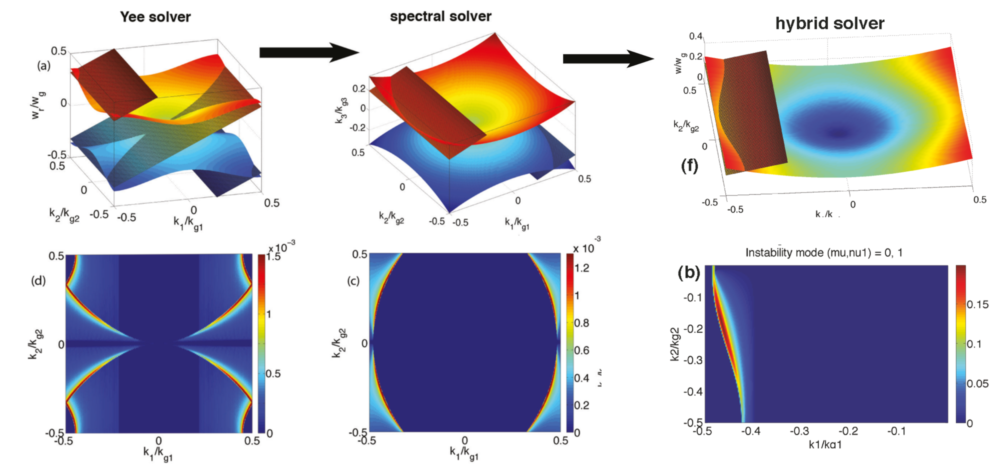
Demo Site Screen Shot



Development of New PIC Algorithms

Hybrid Spectral / Finite-Difference Algorithm

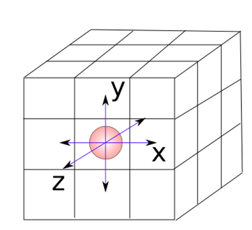
Numerical Cerenkov instability (NCI) is a numerical artifact due to the unphysical coupling of Langmuir modes and EM modes



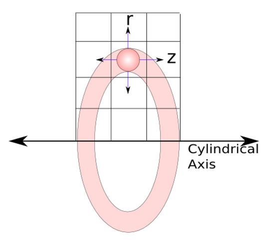
Spectral solver (hybrid solver) can efficiently eliminate NCI, thus perform high fidelity PIC simulation of relativistic plasma drift.

Hybrid PIC/Gridless 3D Geometry[1-2] added to OSIRIS

- In a typical PIC algorithm, all particle positions reside in continuous space, while all Field quantities reside on a grid.

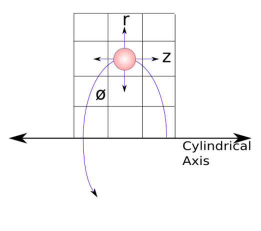


- In a traditional r-z code, the fields are solved on a 2D grid, and the macroparticle positions are in (r, z) space. There is no position in ø (The macroparticles are rings).



- Using truncated azimuthal modal decomposition, one results in field solver in which the fields **F**(r, z, ø) are gridded in (r, z) but continuous in ø. Such a model is ideal for problems with low-order azimuthal symmetry.

$$\mathbf{F}(r,z,\phi) = \Re\left\{\sum_{m=0}\mathbf{F}^m(r,z)e^{im\phi}\right\} = \mathbf{F}^0(r,z) + \Re\{\mathbf{F}^1\}\cos(\phi) - \Im\{\mathbf{F}^1\}\sin(\phi) + \Re\{\mathbf{F}^2\}\cos(2\phi) - \Im\{\mathbf{F}^2\}\sin(2\phi) + \cdots$$



[1] A. Davidson et al., J. Comp. Phys. 281, 1063 (2015).
[2] A. Lifschitz et al., J. Comp. Phys. 228, 1803 (2009).

PICKSC on CUDA

OSIRIS on GPU: Doing Production Science Runs

One, Two and Three Dimensions; All Features of Osiris Supported
Dynamic GPU Load Balancing/Tuning
Fully MPI Ready - Run On Thousands of GPU Nodes
Tailored Support for Fermi and Kepler generations.

	Plasma at 5ev	Plasma at 5KeV
2D (Linear Particle	1016 M Particles / s	862 M Particles / s
2D (Cubic Particle	276 M Particles / s	234 M Particles / s
3D (Linear Particle	311 M Particles / s	261 M Particles / s

All runs on Nvidia K40c (Kepler)

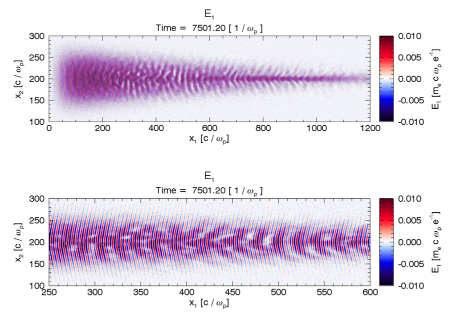
Nvidia K80

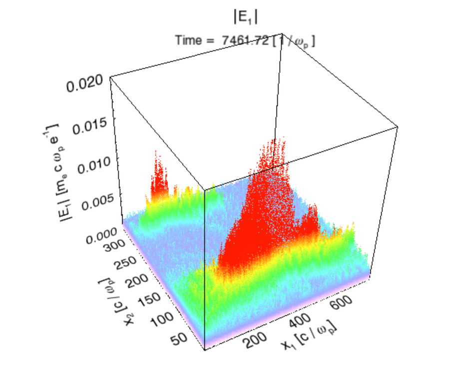
Work in Progress

Support for the newest GPU generations: Maxwell and Kepler K80
Integrated host and GPU calculations so hosts isn't idle while GPU works
Azimuthal Mode cylindrical simulations on the GPU
GPU support for PICKSC Kinetic codes: QuickPIC, OSHUN, UPIC Emma
Implement into Skeleton codes

Production Codes Using Advanced Architectures

OSIRIS-CUDA is now being used with ~100 GPUs for non-linear plasma wave and laser-plasma instability studies
- Enabled a large number of parameter scans not previously feasible
- ~2-3 nsec/particle/GPU





Nonlinear EPWs:
Multi-dimensional EPWs are affected by novel wave-particle interactions that produce sidebands (amplitude modulations), wave-bowing, and transverse localization.

SRS in speckled laser beams:
Above, large EPW activity in an above-threshold laser speckle has stimulated SRS in a below-threshold speckle (the smaller amplitude packet of EPWs).
OsirisCUDA is enabling 2D parameter scans to study inter-speckle SRS physics.