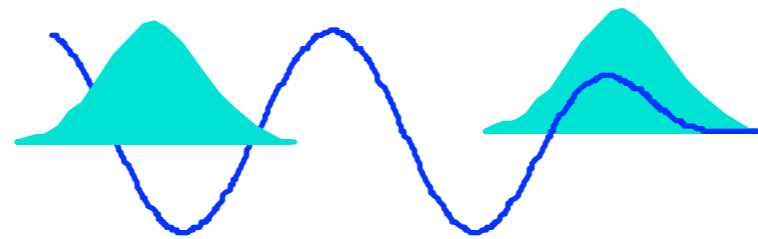


Status and Future Plans for Open Source QuickPIC

Weiming An
University of California Los Angeles
anweiming@ucla.edu

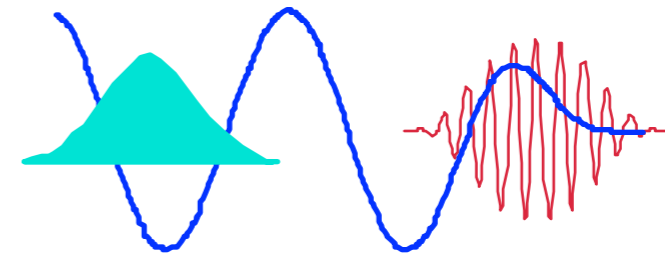


QuickPIC^[1,2] is a 3D parallel Quasi-Static PIC code, which is developed based on the framework UPIC^[3].



Full PIC(Osiris):

$$dt \sim 0.02\omega_p^{-1}$$



Courant Condition

QS PIC(QuickPIC):

$$dt \sim 20.0\omega_p^{-1}$$

$$\sim \sqrt{\gamma} \text{ of the beam}$$

Free of CC!

$$\sim \omega_0/\omega_p$$

1000 Times Faster

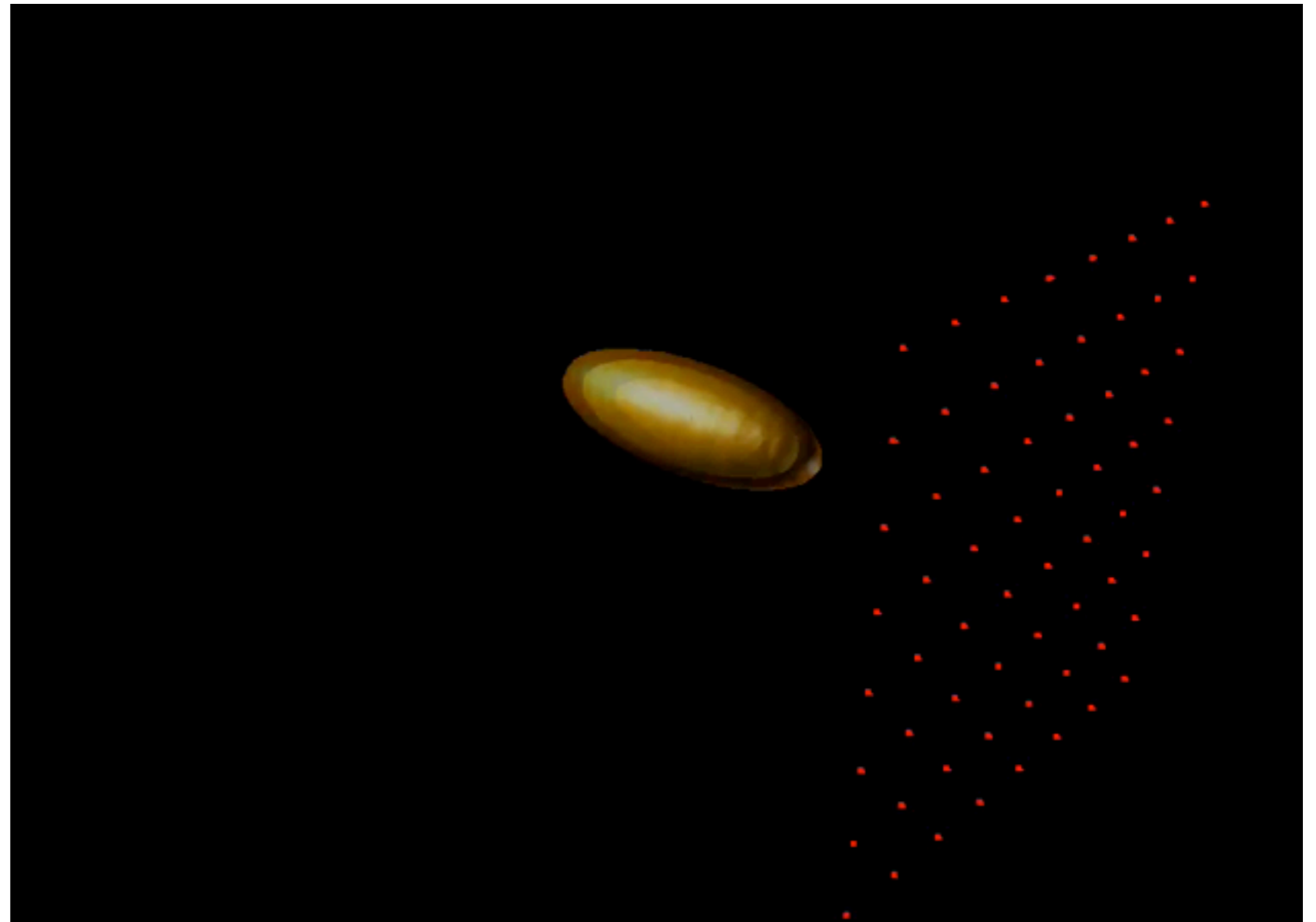
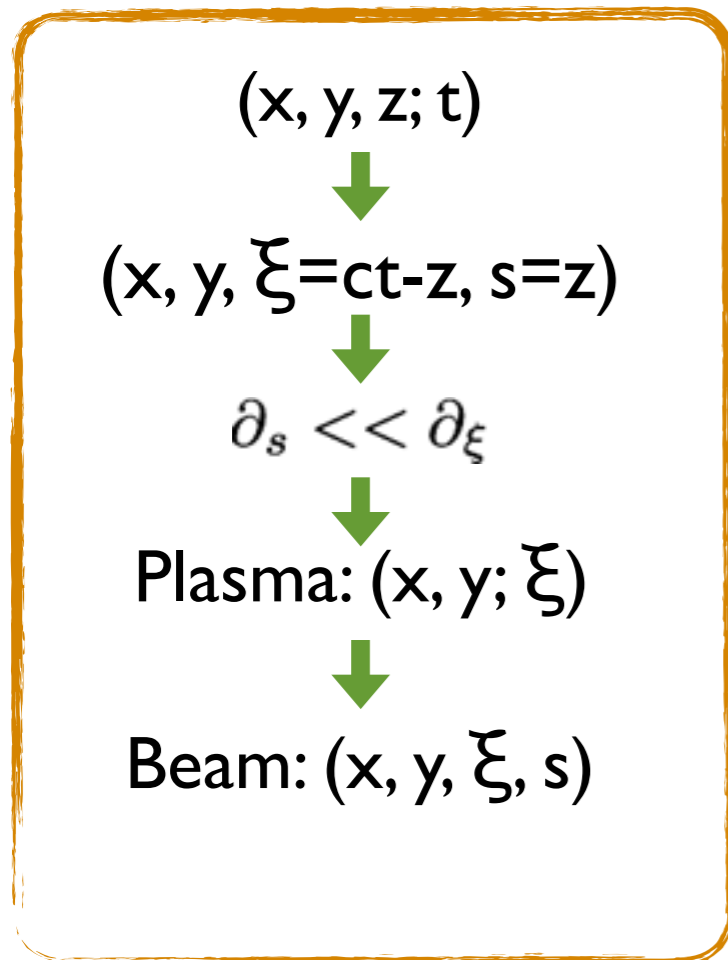
[1] C. Huang et al., J. Comp. Phys. 217, 658 (2006).

[2] W. An et al., J. Comp. Phys. 250, 165 (2013).

[3] V. K. Decyk, Computer Phys. Comm. 177, 95 (2007).



Quasi-Static Approximation*



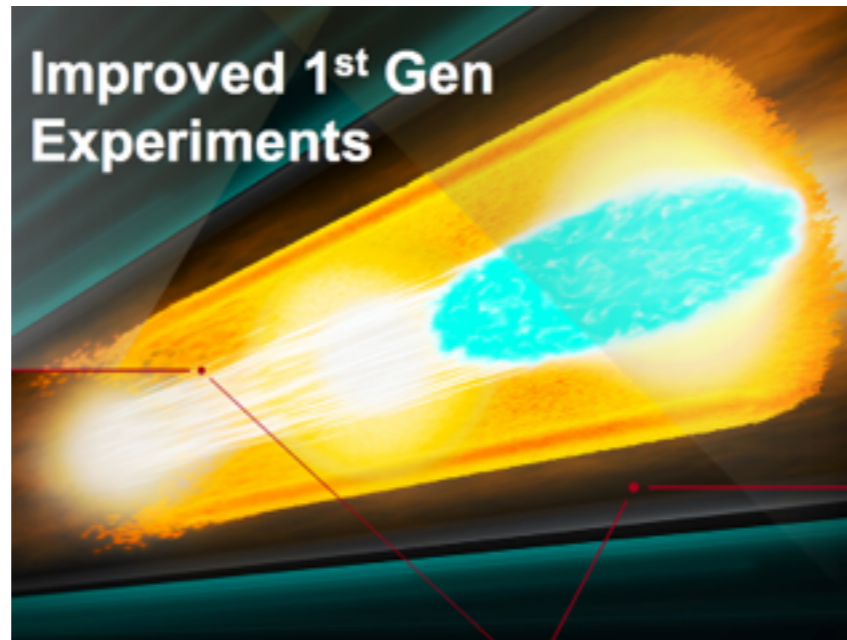
*P. Sprangle, et al., PRA 41, 4463 (1990)

Embeds a 2D PIC code inside a 3D PIC code based on UPIC Framework.

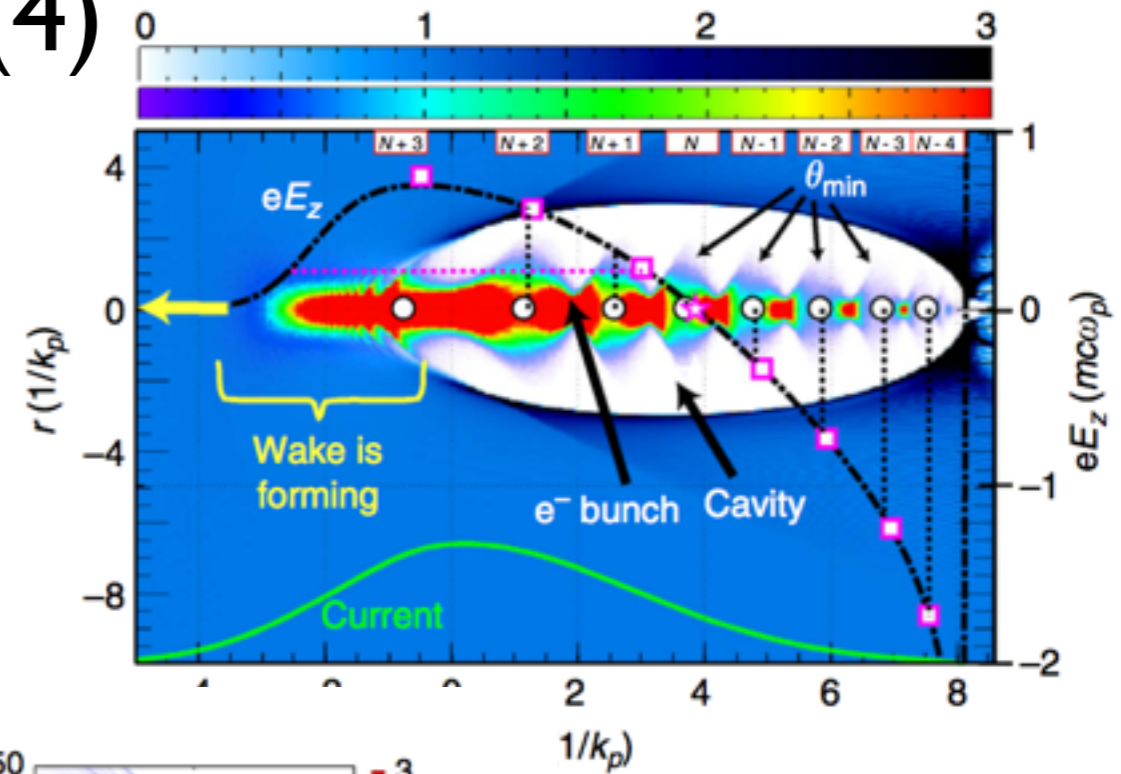
(1)



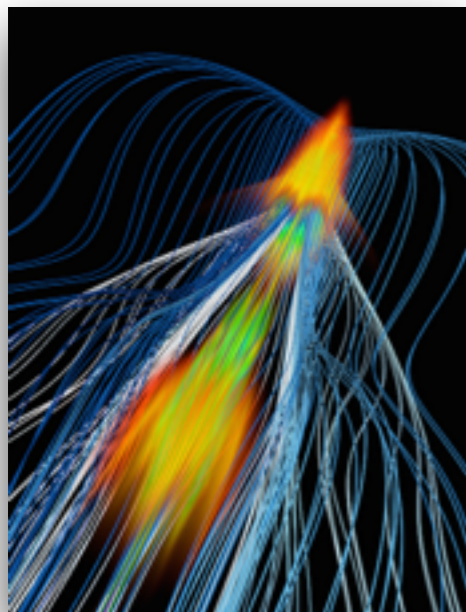
(3)



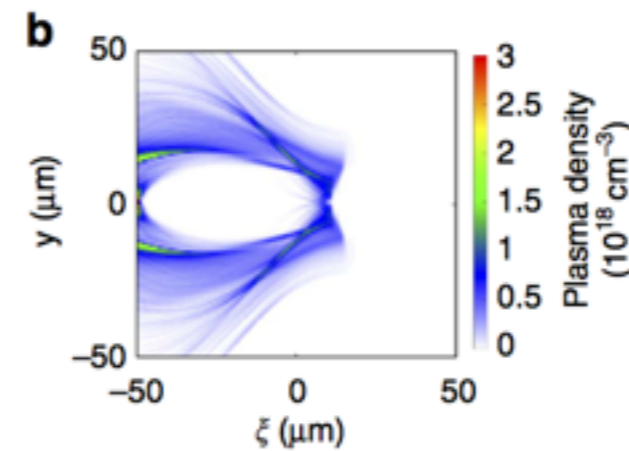
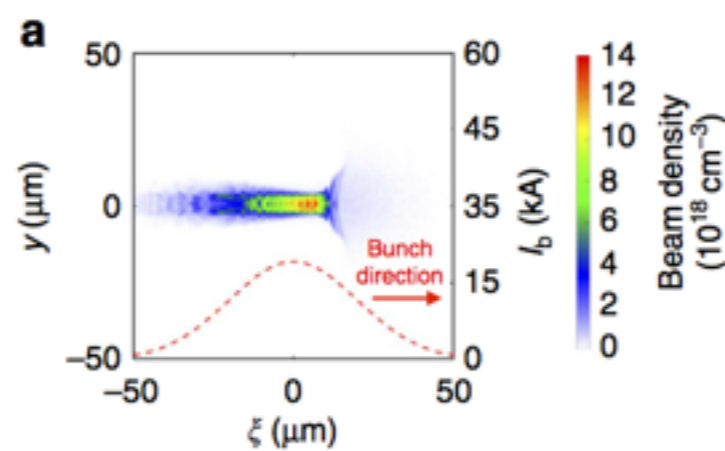
(4)



(2)

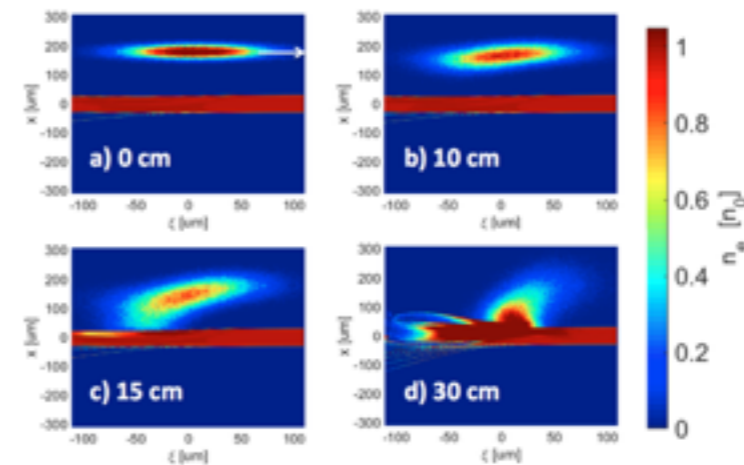
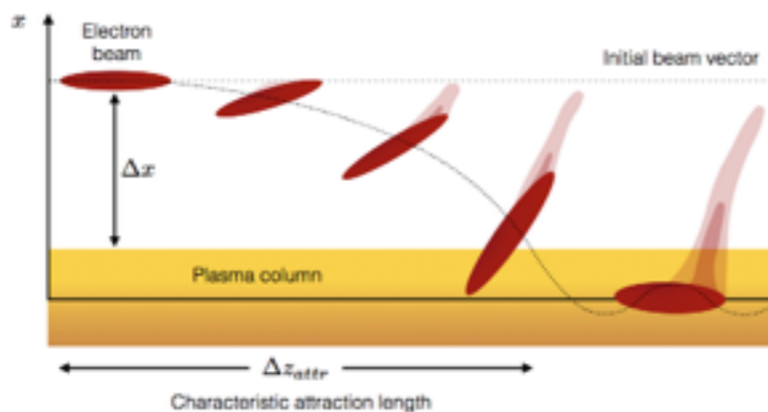


(5)



Highly Nonlinear
Nonuniform
Asymmetric
Field Ionization

(6)





Viktor K. Decyk

UPIC 2.0: MPI+OpenMP

Open Source

Github



QuickPIC with MPI+OpenMP

Boundary Condition: Conducting

Interpolation Order: 1st



Weiming An

Classes

Object Oriented
Fortran 2003

input

simulation

field2d

species2d

beam3d

field3d

fft2d

fpois2d

ufield2d

part2d

hdf5 I/O

part3d

ufield3d

fdist2d

fdist3d

parallel

parallel_pipe

perror

spect2d

spect3d

Field2d Class

```

type field2d
  private
  class(ufield2d) :: rs,ks
  class(fft2d), pointer :: ft
  class(fpois2d), pointer :: pt
  integer :: state, gcells
contains
  generic :: new => init_field2d
  generic :: del => end_field2d
  generic :: fftrk => fftrk_field2d
  generic :: fftkr => fftkr_field2d
  generic :: div => divf_field2d
  generic :: grad => gradf_field2d
  generic :: curl => curlf_field2d
  generic :: pot => potential_field2d
  generic :: smooth => smoothf_field2d
  generic :: elf => elfield_field2d
  generic :: bf => bfield_field2d
  generic :: bfqp => bfield_qp_field2d
  generic :: cg => copyguard_field2d
  generic :: ag => acopyguard_field2d

```

```

generic :: psend => pipesend_field2d
generic :: precv => piperecv_field2d
generic :: cp => copyfrom
generic :: cb => copyto
generic :: as => asc, asa
generic :: add => sum1, sum2
generic :: sub => minus1, minus2
generic :: mult => multiply1, multiply2
generic :: wr => writehdf5_field2d

```

```

procedure, private :: init_field2d, end_field2d
procedure, private :: fftrk_field2d, fftkr_field2d, divf_field2d
procedure, private :: gradf_field2d, curlf_field2d, potential_field2d
procedure, private :: smoothf_field2d, elfield_field2d, bfield_field2d
procedure, private :: bfield_qp_field2d, copyguard_field2d
procedure, private :: acopyguard_field2d, pipesend_field2d
procedure, private :: piperecv_field2d, asc, asa, sum1, multiply1
procedure, private :: writehdf5_field2d, multiply2, sum2, copyfrom, copyto
procedure :: getstate, getgcells, getsrs, getks

```

end type

Main 2D Loop

```

if (j == nstep2d) then
  call MPI_WAIT(id(2),istat,ierr)
endif
call qb%cp(qeb,j+1,(/1/),(/1/))
call qb%fftrk(1)
call qb%elf(epwb)
call qe%mult(qi,-1.0)
do l = 1, sim%sim%nspecies
  call qe0(l)%as(0.0)
  call spe(l)%qdp(qe0(l))
  call qe%add(qe,qe0(l))
end do
call qe%fftrk(1)
call qe%pot(psi)
call psi%grad(fxy)
call fxy%fftkr(1)
call psi%fftkr(1)
do l = 1, sim%sim%nspecies
  call spe(l)%extpsi(psi,dex)
end do
call psi%mult(psi,dex*dex)
if (allocated(psi3d)) call psi%cb(psi3d,j+1,(/1/),(/1/))
call cu%div(div_vpot)
call div_vpot%pot(psit)
call psit%mult(psit,-dex)
call psit%fftkr(1)
call cu%bf(bxyz)
do l = 1, iter
  call bxyz%mult(epw,(/1,2/),(/2,1/),(/-dex,dex/))
  call bxyz%mult(bxyz,(/3/),(/3/),(/dex/))
  call bxyz%sub(bxyz,epwb,(/1/),(/1/),(/2/))
  call bxyz%add(bxyz,epwb,(/2/),(/2/),(/1/))
  call bxyz%fftkr(2)
  call cu%as(0.0)
  call dcu%as(0.0)
  call amu%as(0.0)
  do m = 1, sim%sim%nspecies
    call cu0(m)%as(0.0)
    call dcu0(m)%as(0.0)
    call amu0(m)%as(0.0)
    call spe(m)%amjdp(fxy,bxyz,psit,cu0(m),amu0(m),dcu0(m),dex)
    call cu0(m)%mult(cu0(m),dex)

```

```

    call amu0(m)%mult(amu0(m),dex)
    call dcu0(m)%mult(dcu0(m),dex)
    call cu%add(cu,cu0(m))
    call amu%add(amu,amu0(m))
    call dcu%add(dcu,dcu0(m))
  end do
end do
if (l == iter) then
  do m = 1, sim%sim%nspecies
    call reg%add(qe0(m),cu0(m),(/1/),(/1/),(/3/))
    call reg%fftrk(1)
    call reg%smooth(reg)
    call reg%fftkr(1)
    call qe0(m)%as(reg)
    call qe0(m)%cb(qep(m),j+1,(/1/),(/1/))
  end do
  if (allocated(cu3d)) then
    call cu%cb(cu3d,j+1,(/1,2,3/),(/1,2,3/))
  end if
endif
call cu%fftrk(1)
call dcu%fftrk(1)
call amu%fftkr(3)
call cu%bfqp(dcu,amu,epw,dex2,dex)
call cu%div(div_vpot)
call div_vpot%pot(psit)
call psit%mult(psit,-dex)
call psit%fftkr(1)
call cu%bf(bxyz)
enddo
call bxyz%mult(epw,(/1,2/),(/2,1/),(/-dex,dex/))
call bxyz%mult(bxyz,(/3/),(/3/),(/dex/))
call bxyz%sub(bxyz,epwb,(/1/),(/1/),(/2/))
call bxyz%add(bxyz,epwb,(/2/),(/2/),(/1/))
call bxyz%fftkr(2)
call fxy%mult(fxy,-1.0)
call fxy%add(fxy,bxyz,(/1/),(/1/),(/2/))
call fxy%sub(fxy,bxyz,(/2/),(/2/),(/1/))
call dcu%mult(dcu,dxi)
call cu%sub(cu,dcu,(/1,2/),(/1,2/),(/1,2/))
do m = 1, sim%sim%nspecies
  call spe(m)%push(fxy,bxyz,psit,dex)
end do
call fxy%mult(fxy,dex)
call fxy%cb(bxyz,j+1,(/1,2/),(/1,2/))
call psit%cb(bxyz,j+1,(/1/),(/3/))
call bxyz%mult(bxyz,(/1,2/),(/1,2/),(/dex,dex/))
call bxyz%cb(bbxyz,j+1,(/1,2,3/),(/1,2,3/))

```


Classes

input

simulation

Code Development

field2d

species2d

beam3d

field3d

fft2d

fpois2d

ufield2d

part2d

hdf5 I/O

part3d

ufield3d

fdist2d

fdist3d

parallel

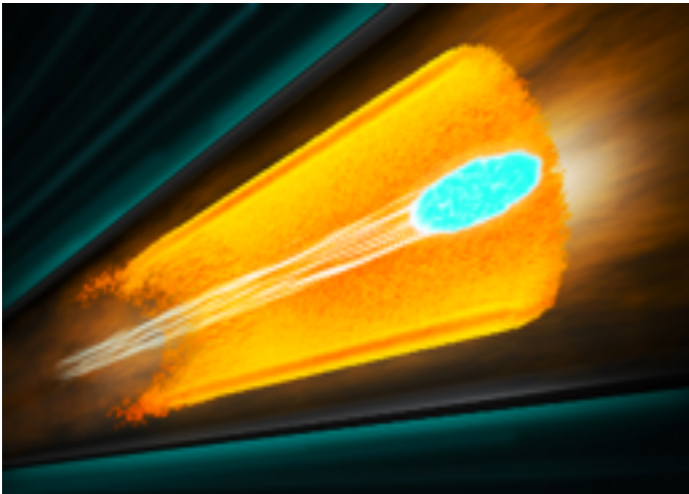
parallel_pipe

perror

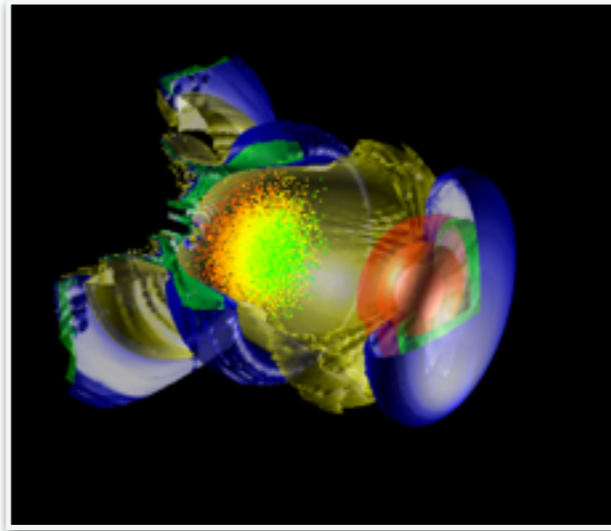
spect2d

spect3d

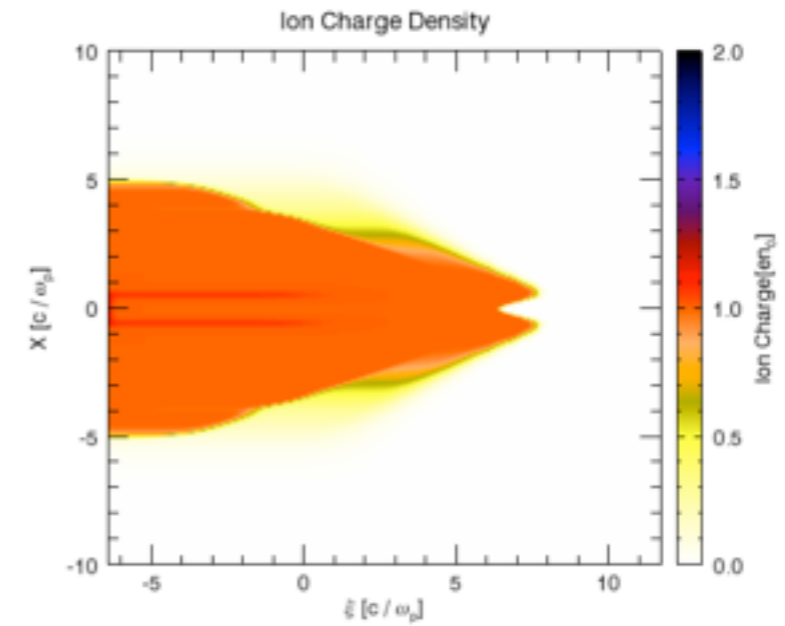
Particles with Varying Charge



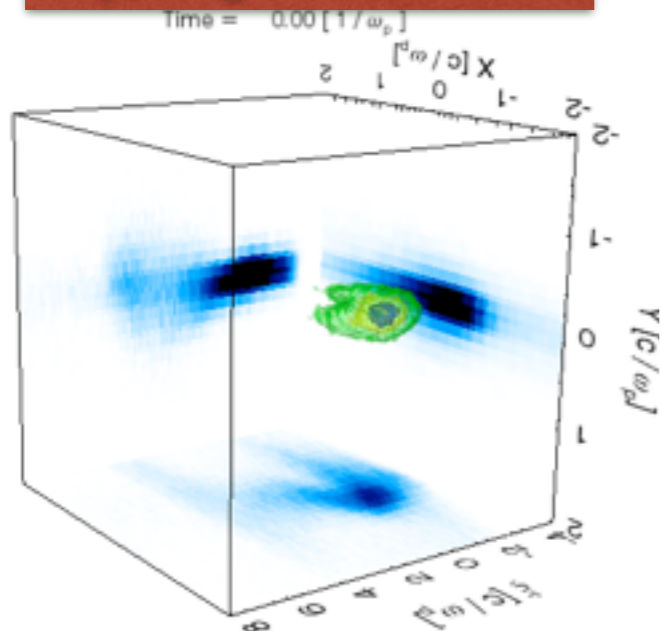
Laser Module



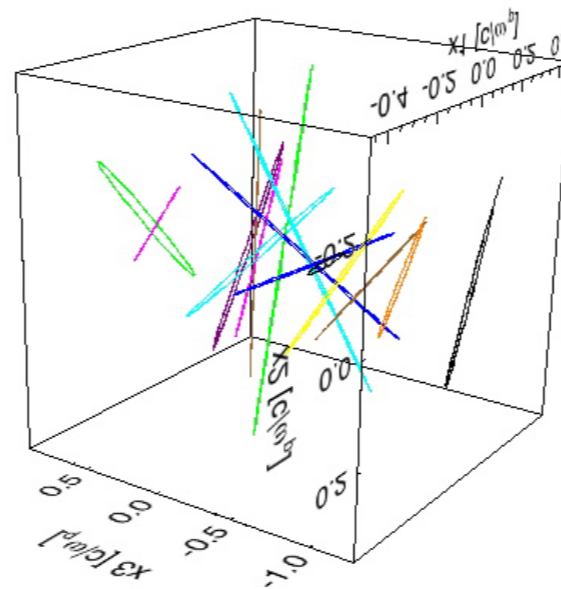
Multiple Field Ionization Module



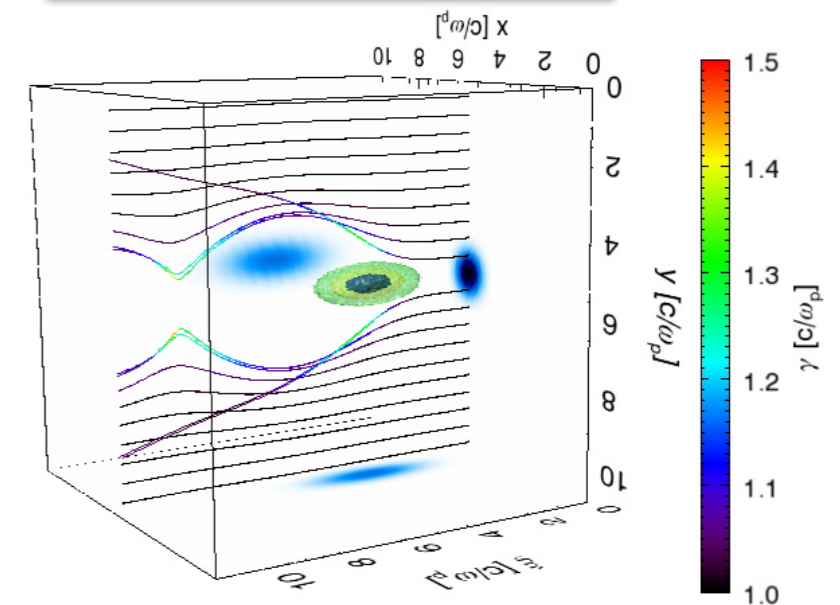
Importing Beam Raw Data



Beam Particle Tracking



Plasma Particle Tracking



- Intel Phi Algorithm (Vectorization)
- Dynamic load balancing
- Adaptive 2d and 3d time steps
- Adaptive particle loading
- Adaptive mesh refinement

<https://github.com/UCLA-Plasma-Simulation-Group/QuickPIC-OpenSource.git>



<http://picksc.idre.ucla.edu>

Weiming An
anweiming@ucla.edu