UCLA Samueli
Electrical & Computer Engineering

BERKELEY LAB

Implementation of a Mesh refinement algorithm into the quasi-static PIC code QuickPIC

Qianqian Su[1], Fei Li[1], Weiming An[2], Yujian Zhao[1], Lance Hildebrand[1],
Viktor Decyk[1], Paulo Alves[1], Ann Almgren[3], Warren Mori[1]

[1] University of California, Los Angeles, US
[2] Beijing Normal University, Beijing, China
[3] Lawrence Berkeley National Lab

1

Work at UCLA supported by DOE through a FNAL subcontract 644405, and DE-SC0010064, and by NSF through award 2108970

Work at LBNL supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under contract DE-AC02-05CH1123

3D quasi-static PIC code QuickPIC

- Plasma wakefield acceleration (PWFA) has made great progress with the help of particle-in-cell (PIC) simulation
- QuickPIC is an efficient quasi-static PIC code that can provide more than hundreds times of speed up compared to standard PIC codes for accurate PWFA simulation

Quasi-static approximation (QSA)

Cartesian coordinates $(x, y, z; t)$

↓

Co-moving coordinates

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

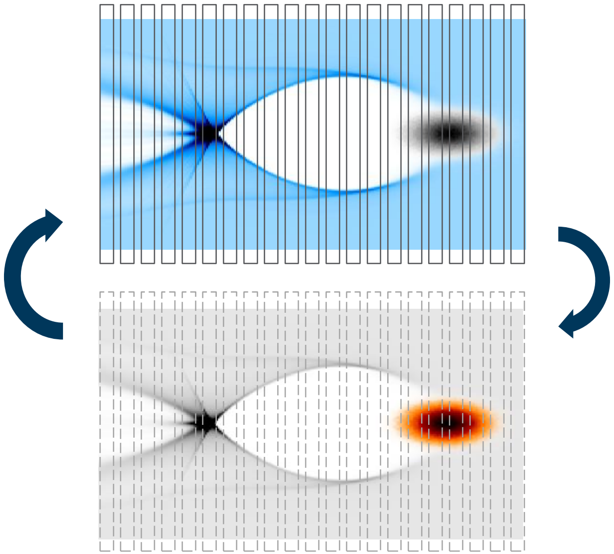
$\partial_s \ll \partial_\xi$

Beam Plasma

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

Beam: $(x, y, \xi; s)$

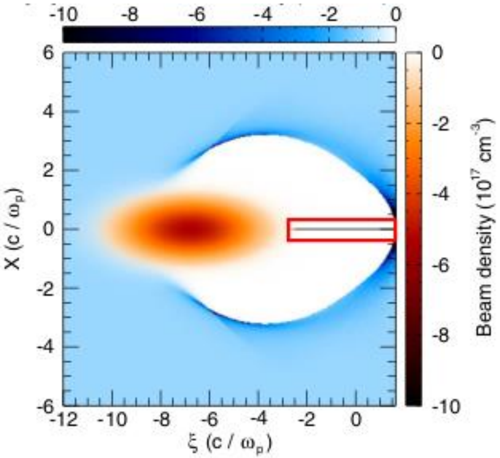
Quasi-static loop: beam is frozen while plasma and field is evolved



3D loop: field and plasma particles are frozen while beam is advanced.

C. Huang et. al. Journal of Computational Physics, 217(2):658–679, 2006,
W. An, et. al. Journal of Computational Physics, 250:165–177, 2013.

PWFA simulation for linear collider simulations is computationally expensive both in time and memory

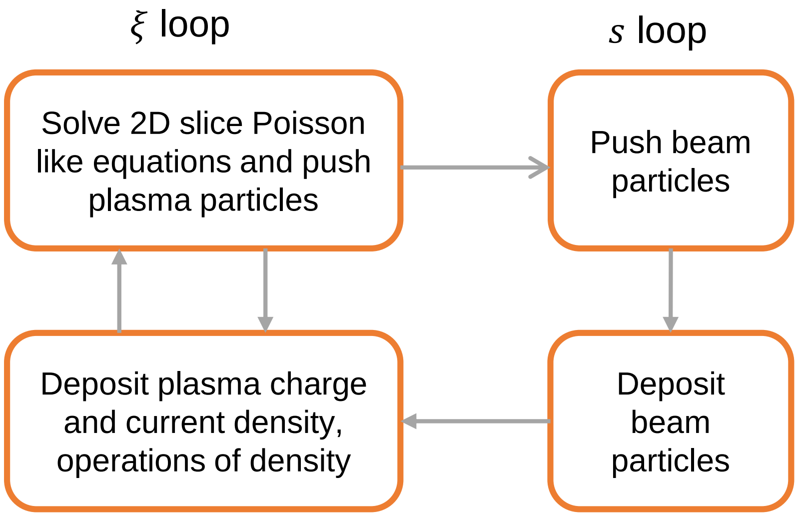


- **PWFA-LC parameter for trailing bunch**
  - Normalized emittance in x, y direction ~ 2um, 0.005um
  - Matched spot size ~ 463.9 nm, 23.2 nm (Bubble radius ~ tens of μm)
- **Challenges**
  - Scales difference of bubble radius and trailing beam
  - Tightly focused trailing beam ($k_p\sigma_r \sim 10^{-3}$) causes ion motion
  - To fully resolve the trailing beam and ion motion, full QuickPIC takes more than 10^9 cpu hours (from estimation)
- **Speed-up method**
  - Mesh refinement (HiPACE ++, QuickPIC)
  - Quasi-3d method (QPAD)
  - GPU (HiPACE++, QuickPIC)

Q. S, et. al., arXiv:2405.00886
T. J. Mehrling, et. al. *IEEE AAC*, 2018
F. Li, et. al. Computer Physics Communications, 261:107784, 2021
S.Diederichs, et. al. Computer Physics Communications, 278: 108421, 2022

W. An, et. al., Physical Review Letters, 118:244801, 2017,

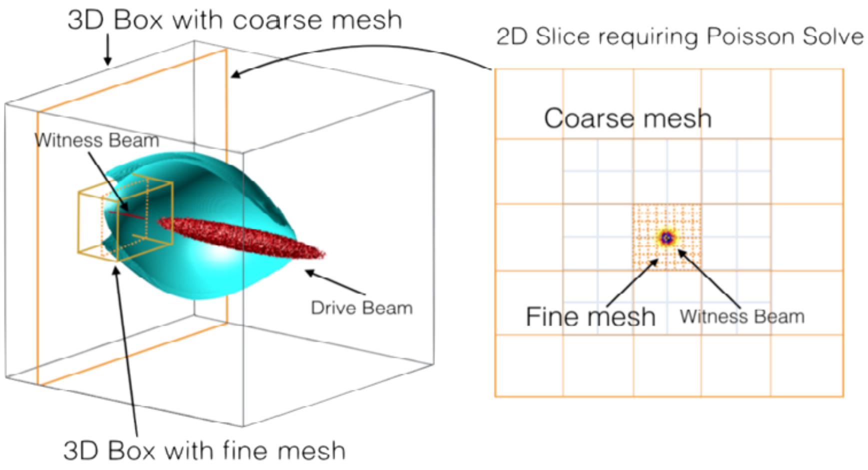
Mesh refinement algorithmic flow in QuickPIC



All the operations done for both fine mesh and coarse mesh

$$\nabla_\perp^2 \phi_l = q_l, \nabla_\perp^2 \phi_l - \phi_l = q_l$$

where q_l is the source at grid l



Interpolate field on coarse grid ϕ_l onto the boundary of ϕ_{l+1}
Use FFT solver to solve ϕ_1, multigrid algorithm to solve $\phi_{l>1}$

Option to correct coarse grid solution by iteration between meshes

Option to use square cells and rectangular box for refined mesh

Q. S, et. al., arXiv:2405.00886

Plasma particles are deposited on an extended mesh

2D Poisson-like equations solved by FFT+Multigrid algorithm

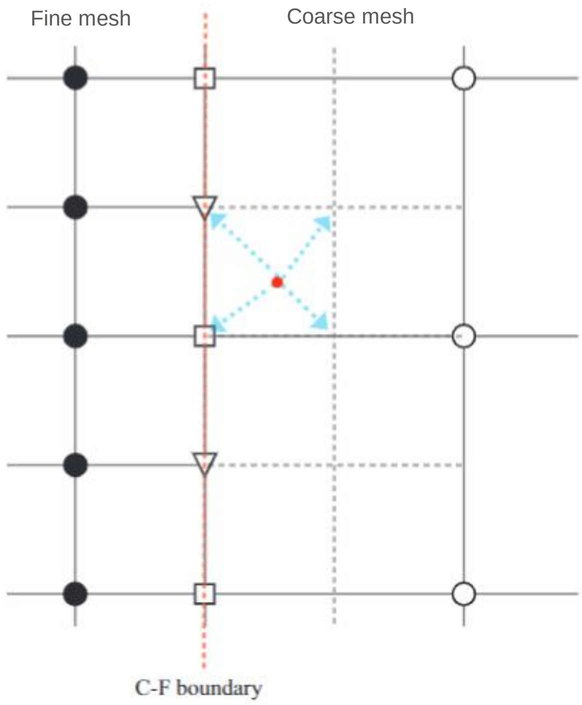
$$
\begin{aligned}
\nabla_\perp^2 \psi &= -(\rho - J_z), \\
\nabla_\perp^2 \vec{B}_\perp &= \hat{z} \times \left(\frac{\partial}{\partial \xi} \vec{J}_\perp + \nabla_\perp J_z \right), \\
\nabla_\perp^2 B_z &= -\nabla_\perp \times \vec{J}_\perp, \\
\nabla_\perp^2 \vec{E}_\perp &= \nabla_\perp \rho + \frac{\partial}{\partial \xi} \vec{J}_\perp, \\
\nabla_\perp^2 E_z &= \nabla_\perp \cdot \vec{J}_\perp,
\end{aligned}
$$

ρ, J are deposited on the finest extended mesh

$$
\nabla_\perp \cdot \vec{J}_\perp = \frac{1}{2h_l}(J_x(i+1,j) - J_x(i-1,j) + J_y(i,j+1) - J_y(i,j-1))
$$

The densities on the coarser level are obtained by coarsening

High order sources near the boundary requires more complex deposit approach



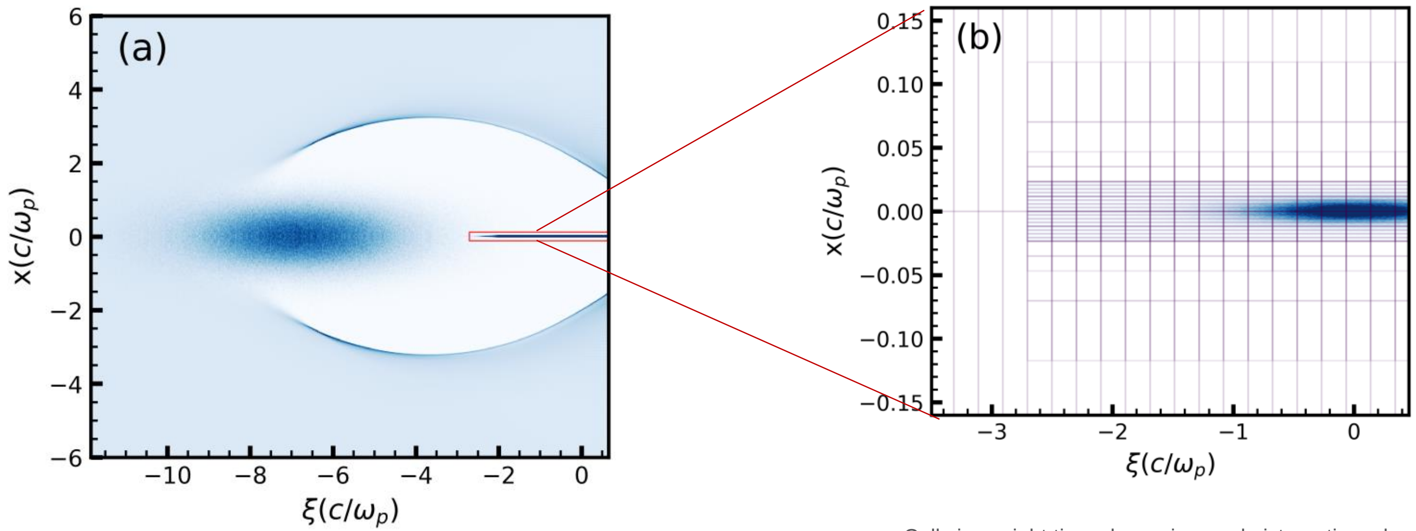
Features of the mesh refinement code

- **Fast fine mesh solver**
  - Parallelized with MPI + OpenMP
  - Scatter fine grid to other mpi ranks
- **Load balanced pipelining parallelization**
- **Multiple levels of refinement**
- **Adaptive mesh refinement with evolving refinement level, box size and resolution**
- **Evolving simulation box size and resolution**
- **Evolving time step in s**

Several benchmark examples

- Symmetric electron PWFA
- Asymmetric electron PWFA
- Positron acceleration
- Adaptive mesh refinement with evolving refinement level, box size and resolution

Fully self-consistent mesh refinement simulation



Trailing beam parameter $k_p\sigma_r = 0.00595$, $k_p\sigma_z = 0.595$

Cell sizes eight times larger in x and sixteen times larger in ξ for each level, compared to the cell sizes employed in the actual simulation.

4 refinement levels, each increase 4x resolution
Coarse grid has 1024 × 1024 × 1024 grids

Load 1 ion particle per refined cell

The finest grid has resolution of 6.1 × 6.1 × 115.23 nm

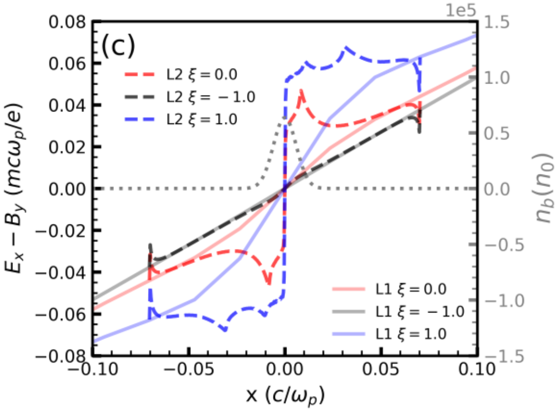
Need multiple refinement levels for accurate fields at the boundaries

$$\nabla_\perp^2 \psi = -(\rho - J_z)$$

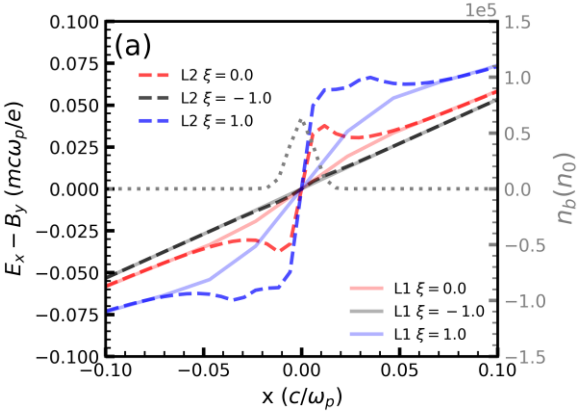
The error induced when calculating focusing force by $\boldsymbol{E}_\perp + e_z \times \boldsymbol{B}_\perp = -\nabla_\perp \psi$

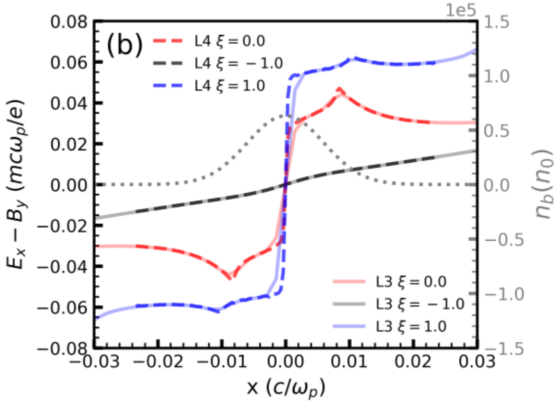
Match the Dirichlet boundary from linear interpolation, but derivative not continuous between different levels

1 refinement level
64x higher resolution



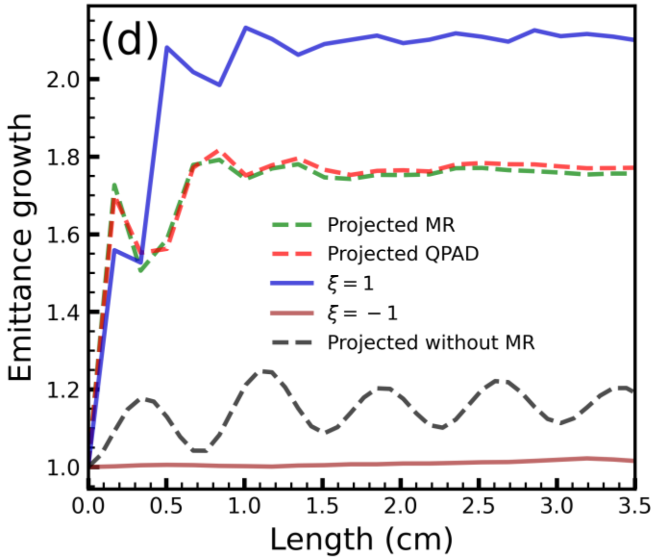
3 refinement levels each 4x higher resolution

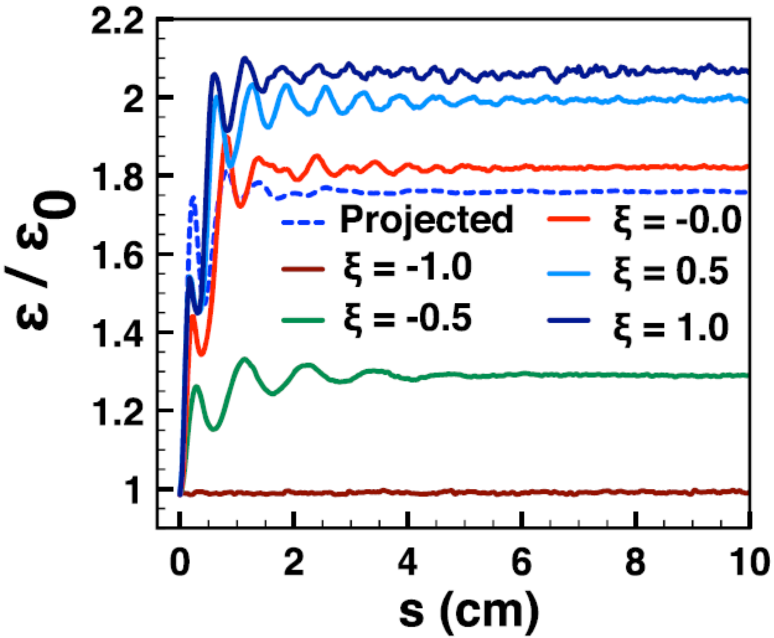




Long range acceleration with mesh refinement shows agreement with QPAD and previous results

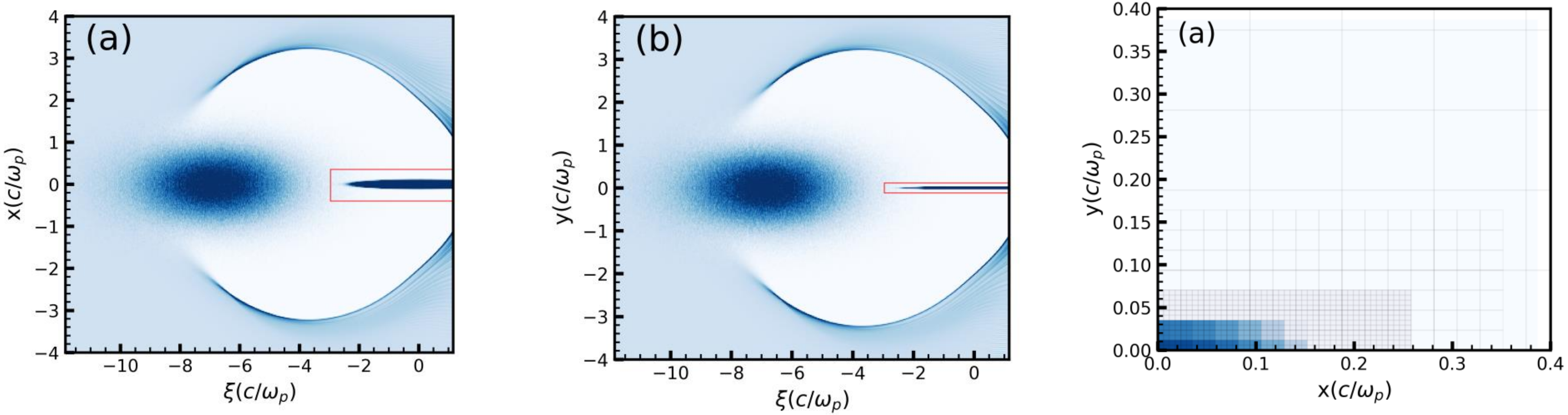
Mesh refinement





W. An, et. al., Physical Review Letters, 118:244801, 2017

Mesh refinement also works for asymmetric beam with rectangular fine region

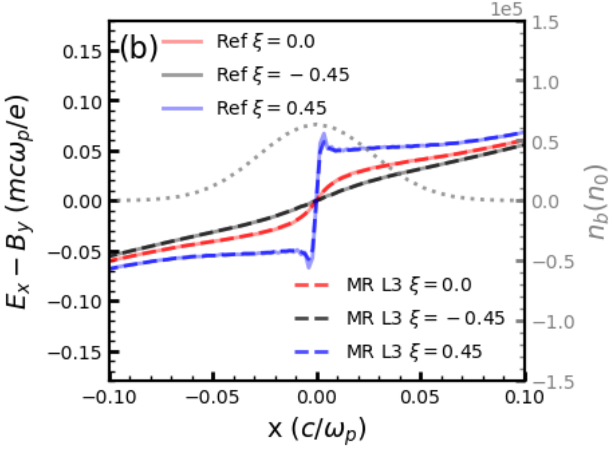


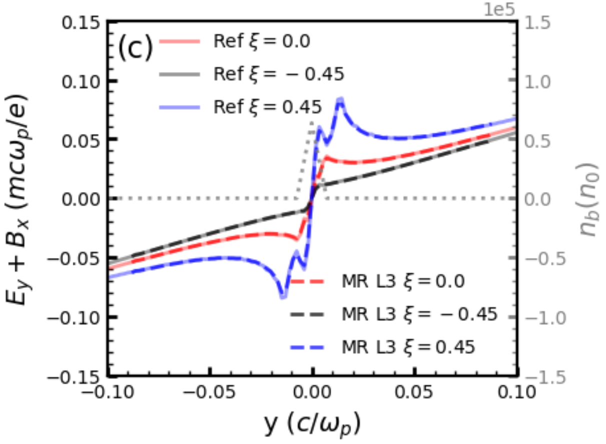
Cell sizes four times larger for each level compared to the sizes employed in the simulation.

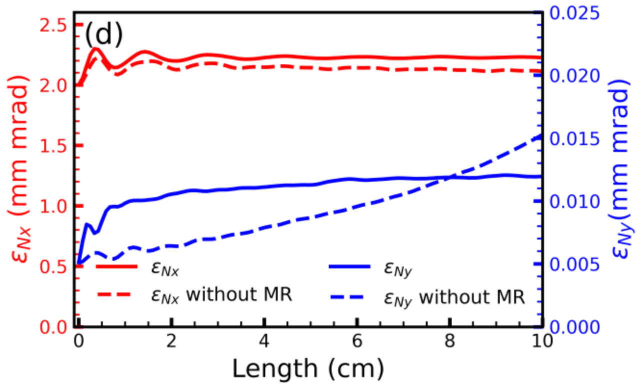
Refinement ratio 32

Matched trailing beam parameter $k_p\sigma_x = 0.0273, k_p\sigma_y = 0.0014,\ k_p\sigma_z = 0.595$

Mesh refinement also works for asymmetric beam with rectangular fine region





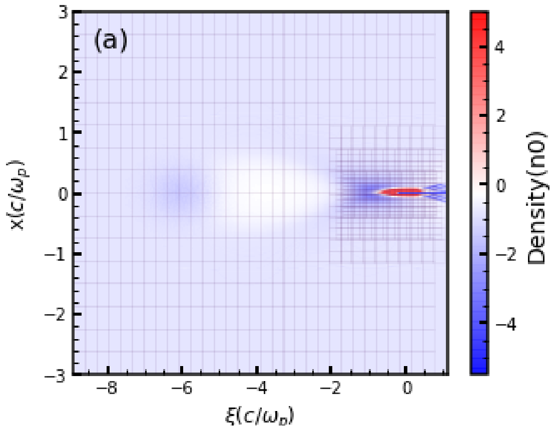


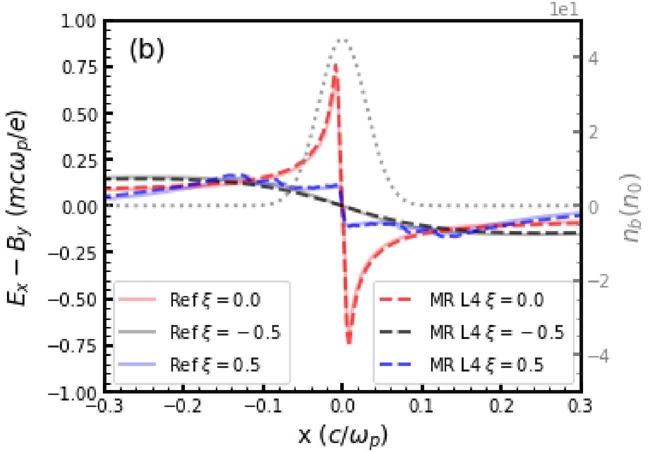
Using Hydrogen plasma

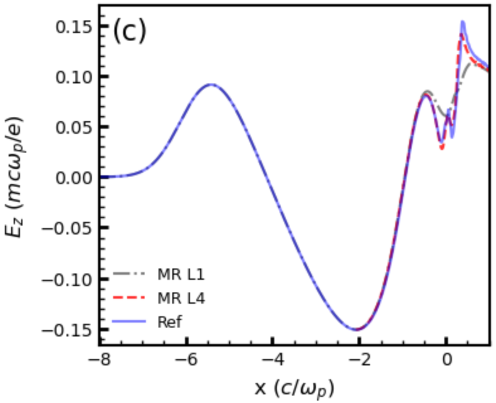
Matched trailing beam parameter $k_p\sigma_x = 0.0273, k_p\sigma_y = 0.0014,\ k_p\sigma_z = 0.595$

350 times speed up compared with one time step 3D QuickPIC simulation with fine resolution everywhere

Positron simulation shows agreement of fields







Refinement ratio 8, each level refinement ratio of 2

Drive beam $k_p\sigma_r = 0.3, k_p\sigma_z = 0.6$

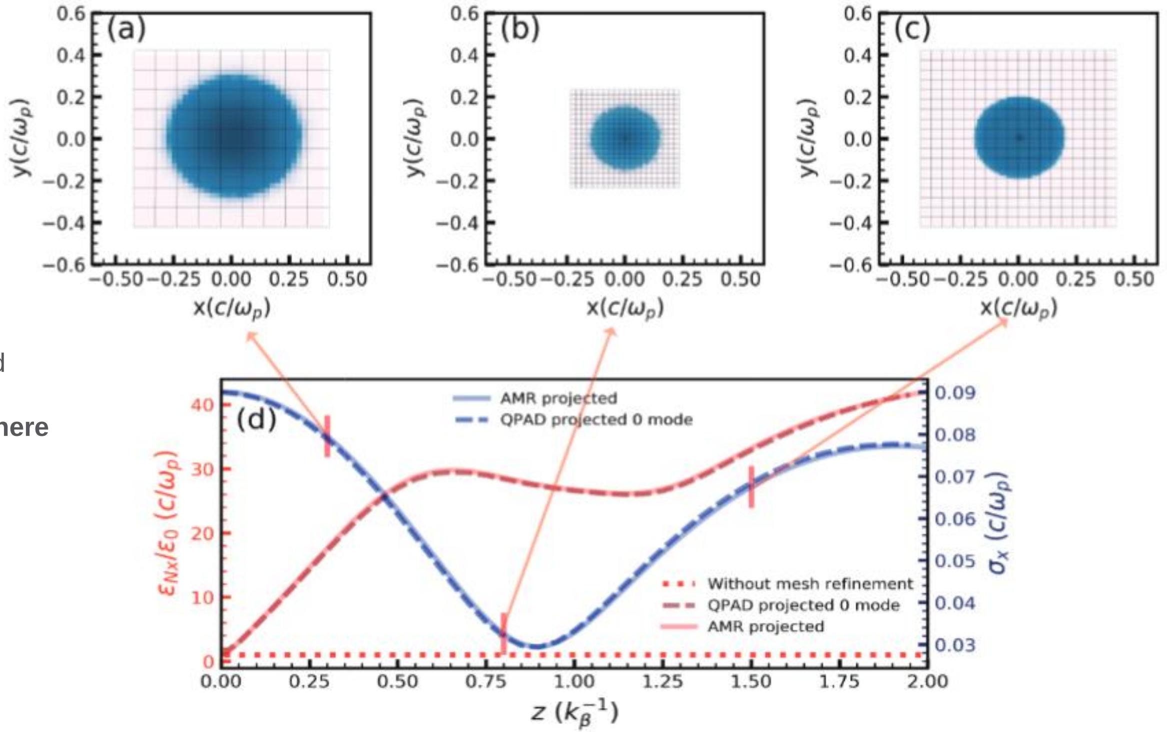
Trailing beam $k_p\sigma_r = 0.03, k_p\sigma_z = 0.4$

More particles loaded near the center compared to ion motion case

80 times speed up compared with one time step 3D QuickPIC simulation with fine resolution everywhere

Adaptive mesh grid setting for evolving beams

Ion mass $m_i = 200\, m_e$
Trailing beam parameter
$k_p\sigma_r = 0.09\,, k_p\sigma_z = 0.595, \sigma_p = 0.1,\, \gamma = 5000$
Finest grid size can be 16 smaller than coarse grid

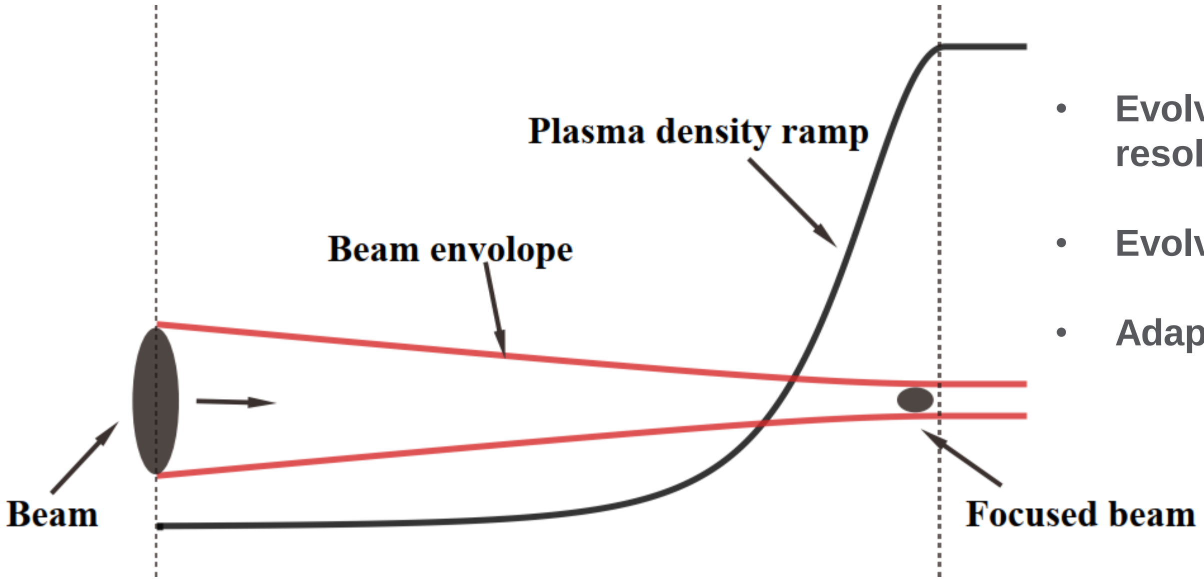


For 200 time steps with 4 MPI ranks, 1 OpenMP thread

1.5 hour with 0 Mode QPAD fine resolution everywhere
(1 d in transverse direction 8192 * 1 * 256)
4 hour with adaptive mesh refinement
Coarse resolution (512 * 512 * 256) takes 1.5 hour
Fine resolution everywhere (8192 * 8192 * 256) not possible on desktop

Both test on a local desktop with Intel i7-8700 CPU with 6 physical cores, 16GB RAM

Adaptive mesh grid setting for beam propagation in a plasma ramp



- Evolving simulation box size and resolution
- Evolving time step
- Adaptive mesh refinement

Has been use to study adiabatic plasma lens

Large simulation window
Lower refinement ratio
Large time step

→

Smaller simulation window
Higher refinement ratio
Small time step

Summary

- **We have developed a MPI + OpenMP parallelized mesh refinement code based on QuickPIC, it enables simulations using witness beams with spot sizes several orders of magnitude small than the accelerator structure.**

- **The scalability of the code has been improved by using pipelining, several strategies has been used to improve the load balancing of both field solver and pipelines.**

- **We also developed an adaptive mesh refinement option for an evolving beam.**

- **Several benchmark cases have been tested and get consistent result as previously published papers and a Quasi-3d QS PIC code QPAD.**

- **Future work on QuickPIC and QPAD**
  - Adaptive 2d step
  - Improve load balance of AMR and particle routines
  - GPU support
  - Mesh refinement in quasi-3d code

Thanks for your attention!