Space Charge Solver Based On Tensor Decomposition

Jiayin Du, CSNS

dujiayin@ihep.ac.cn

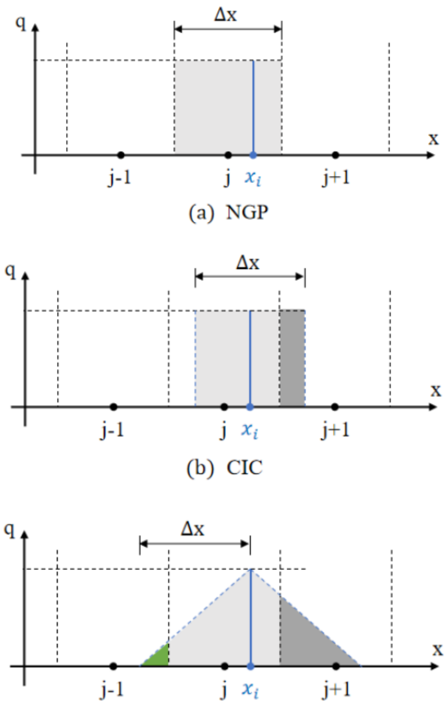
2025.10.30

Table of Contents

Overview of space charge solver

Also called **Poisson Solver**

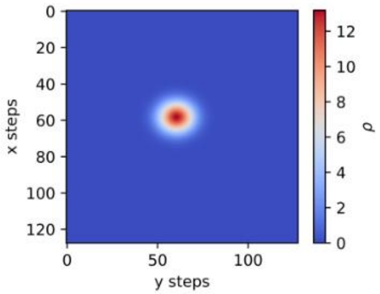
Source Depositor → **Field Solver** → **Particle Pusher**



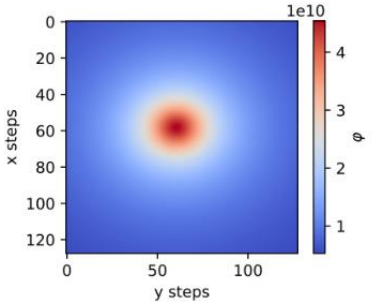
(a) NGP

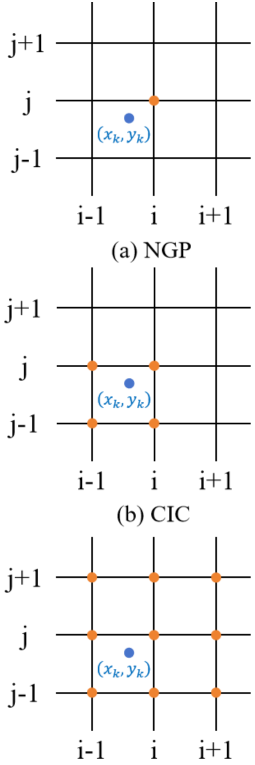
(b) CIC

(c) TSC



FFT, Spectral Method,
Finite Difference Method ...





(a) NGP

(b) CIC

(c) TSC

Tensor Decomposition

Consider a N-order tensor $\mathcal{A} \in R^{I_1 \times I_2 \times \ldots \times I_N}$

Define an operator $\times_n$ which is tensor mapped to tensor， for any matrix $M \in R^{I_j \times I_i}$， it has：

$$(\mathcal{A} \times_i M)_{I_1 I_2 \ldots I_j \ldots I_N} = \sum_{i=1}^{I_i} \mathcal{A}_{I_1 I_2 \ldots i \ldots I_N} * M_{I_j i}$$

This operation equates to swapping the involved dimension of the tensor with that of the product matrix, i.e., it implements the mapping:

$$R^{I_1 \times \ldots \times \boxed{I_i} \times \ldots \times I_N} \to R^{I_1 \times \ldots \times \boxed{I_j} \times \ldots \times I_N}$$

Specifically, if tensor $\mathcal{A}$ is a square matrix, then the operation becomes:

$$\mathcal{A} \times_1 U \times_2 V = \boxed{U \cdot \mathcal{A} \cdot V^T}$$ Singular Value Decomposition

So this is High-order Singular Value Decomposition, HOSVD.

Solver Algorithm and benchmarking

- The discretized Poisson equation with boundary conditions

$$\nabla^2\varphi(x,y,z) = -\frac{\rho(x,y,z)}{\epsilon_0}$$

$$\downarrow$$

$$\left[\frac{\partial^2\varphi}{\partial x^2}+\frac{\partial^2\varphi}{\partial y^2}+\frac{\partial^2\varphi}{\partial z^2}\right]_{ijk} = \hat{\rho}_{ijk}$$

$\downarrow$ The discretized Laplace operator

$$\boxed{\left[\frac{\partial^2\varphi}{\partial x^2}\right]_{ijk} = \frac{\varphi_{i+1,j,k} - 2\varphi_{i,j,k} + \varphi_{i-1,j,k}}{\Delta x^2}}$$

$$\left[\frac{\partial^2\varphi}{\partial y^2}\right]_{ijk} = \frac{\varphi_{i,j+1,k} - 2\varphi_{i,j,k} + \varphi_{i,j-1,k}}{\Delta y^2}$$

$$\left[\frac{\partial^2\varphi}{\partial z^2}\right]_{ijk} = \frac{\varphi_{i,j,k+1} - 2\varphi_{i,j,k} + \varphi_{i,j,k-1}}{\Delta z^2}$$

$$\frac{1}{\Delta x^2}\begin{bmatrix} \ddots & \cdots & \cdots & \cdots & \ddots \\ \vdots & -2 & 1 & 0 & \vdots \\ \vdots & 1 & -2 & 1 & \vdots \\ \vdots & 0 & 1 & -2 & \vdots \\ \ddots & \cdots & \cdots & \cdots & \ddots \end{bmatrix}\begin{bmatrix} \vdots \\ \varphi_{i-1,j,k} \\ \varphi_{i,j,k} \\ \varphi_{i+1,j,k} \\ \vdots \end{bmatrix} = \sum_{i=1}^{N_x} \varphi_{i,j,k}(D_x)_{ii'}$$

$$= \varphi \times_1 D_x$$

$$\downarrow$$

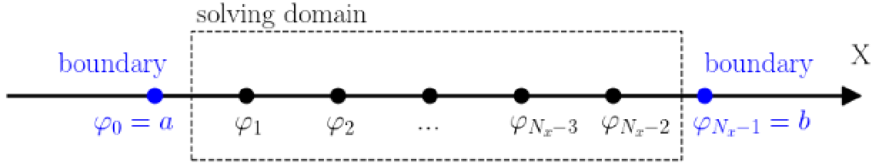
$$\varphi \times_1 D_x + \varphi \times_2 D_y + \varphi \times_3 D_z = \hat{\rho}$$

The form of matrix D_X varies depending on the boundary conditions.

Du J Y, et al. 3d space charge solver based on tensor decomposition forhigh-intensity beams [J/OL]. Progress of Theoretical and Experimental Physics, 2025, 2025 (4): 043G01.

Solver Algorithm and benchmarking

- The discretized Poisson equation with boundary conditions



Dirichlet boundary

$$D_x = \frac{1}{\Delta x^2}\begin{bmatrix} \ddots & \cdots & \cdots & \cdots & \ddots \\ \vdots & -2 & 1 & 0 & \vdots \\ \vdots & 1 & -2 & 1 & \vdots \\ \vdots & 0 & 1 & -2 & \vdots \\ \ddots & \cdots & \cdots & \cdots & \ddots \end{bmatrix}$$

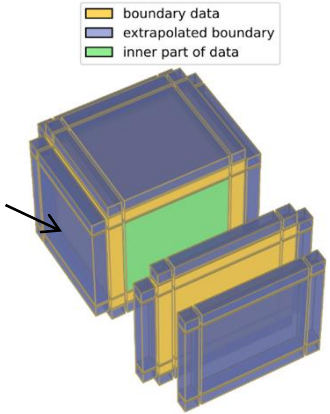
The handling of **Dirichlet boundary condition** is rather special: the form of the differential operator matrix remains unchanged, and it is only necessary to superimpose the electric potential at the boundary onto the charge distribution.

$$\left[\frac{\partial^2 \varphi}{\partial x^2}\right]_{1jk} = \frac{\varphi_{0jk} - 2\varphi_{1jk} + \varphi_{2jk}}{\Delta x^2} = \hat{\rho}_{1jk}$$

$$\downarrow$$

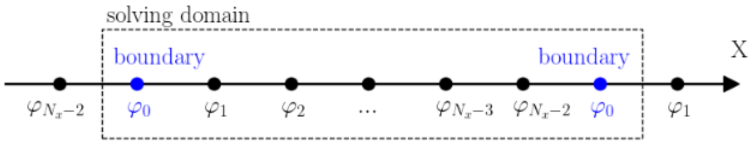
$$\frac{-2\varphi_{1jk} + \varphi_{2jk}}{\Delta x^2} = \hat{\rho}_{1jk} - \frac{\varphi_{0jk}}{\Delta x^2}$$

‘shell’



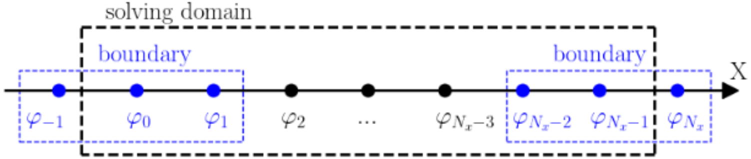
Solver Algorithm and benchmarking

- The discretized Poisson equation with boundary conditions



Period boundary

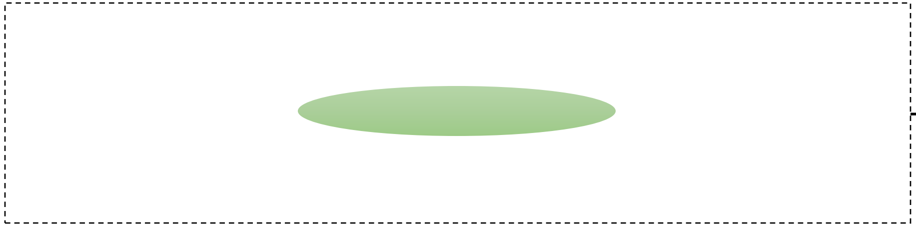
$$D_x = \frac{1}{\Delta x^2}\begin{bmatrix} \ddots & \cdots & \cdots & \cdots & 1 \\ \vdots & -2 & 1 & 0 & \vdots \\ \vdots & 1 & -2 & 1 & \vdots \\ \vdots & 0 & 1 & -2 & \vdots \\ 1 & \cdots & \cdots & \cdots & \ddots \end{bmatrix}$$



$$\left(\frac{\partial\varphi}{\partial X}\right)_0 = \frac{\varphi_1-\varphi_{-1}}{2h_X} = a \qquad \left(\frac{\partial\varphi}{\partial X}\right)_{N_x-1} = \frac{\varphi_{N_x}-\varphi_{N_x-2}}{2h_X} = b$$

Neumann boundary

$$D_x = \frac{1}{\Delta x^2}\begin{bmatrix} \ddots & 2 & \cdots & \cdots & \ddots \\ 1 & -2 & 1 & 0 & \vdots \\ \vdots & 1 & -2 & 1 & \vdots \\ \vdots & 0 & 1 & -2 & 1 \\ \ddots & \cdots & \cdots & 2 & \ddots \end{bmatrix}$$



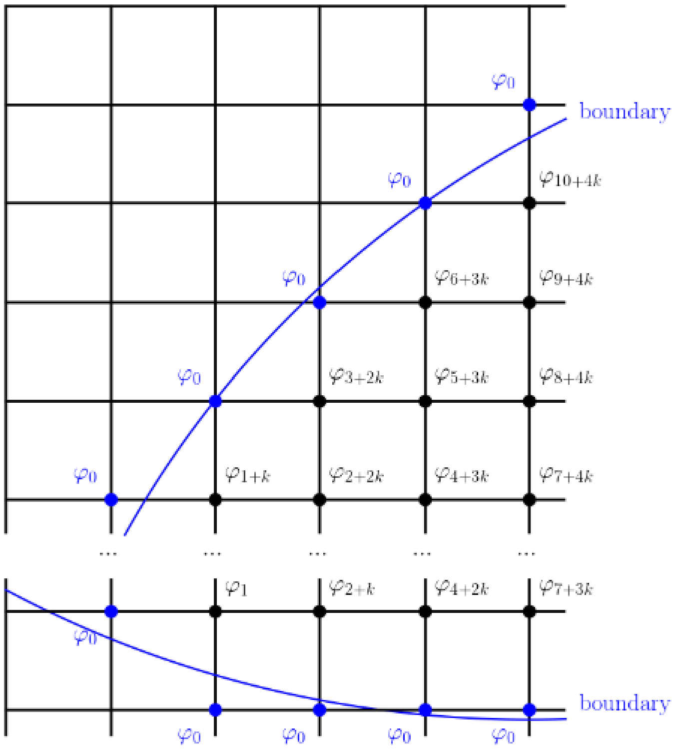
Free boundary

$$D_x = \frac{1}{\Delta x^2}\begin{bmatrix} \ddots & \cdots & \cdots & \cdots & \ddots \\ \vdots & -2 & 1 & 0 & \vdots \\ \vdots & 1 & -2 & 1 & \vdots \\ \vdots & 0 & 1 & -2 & \vdots \\ \ddots & \cdots & \cdots & \cdots & \ddots \end{bmatrix}$$

same as Dirichlet boundary which 'shell' is zero

Solver Algorithm and benchmarking

- The discretized Poisson equation with boundary conditions



Irregular boundary

$$\varphi \times_1 D_x + \varphi \times_1 D_y + \varphi \times_1 D_z = \hat{\rho}$$

We need to rebuild the operator matrix

$$\varphi^* \times_1 (I_y \otimes D_x + D_y \otimes I_x) + \varphi \times_2 D_z = \hat{\rho}^*$$

where, $\otimes$ is Kronecker product

$I_{x,y} \in R^{N_{x,y} \times N_{x,y}}$ is Identity matrix

$\varphi^*, \hat{\rho}^* \in R^{N_x N_y \times N_z}$

Solver Algorithm and benchmarking

- Algorithm construction

$$\varphi \times_1 D_x + \varphi \times_2 D_y + \varphi \times_3 D_z = \hat{\rho}$$

$$\varphi = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}$$

$$\varphi \times_1 D_x = (\mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}) \times_1 D_x$$
$$= \mathcal{S} \times_1 (D_x \cdot U^{(1)}) \times_2 U^{(2)} \times_3 U^{(3)}$$

$$D_x = U^{(1)} \cdot \Lambda_x \cdot U^{(1),-1}$$

$$\varphi \times_1 D_x = \mathcal{S} \times_1 (U^{(1)} \cdot \Lambda_x) \times_2 U^{(2)} \times_3 U^{(3)}$$

$$\varphi \times_2 D_y = \mathcal{S} \times_1 U^{(1)} \times_2 (U^{(2)} \cdot \Lambda_y) \times_3 U^{(3)}$$

$$\varphi \times_3 D_z = \mathcal{S} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 (U^{(3)} \cdot \Lambda_z)$$

$$\hat{\rho} = \mathcal{K} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}$$

$$\varphi \times_1 D_x + \varphi \times_2 D_y + \varphi \times_3 D_z = \hat{\rho}$$

$$\mathcal{S} \times_1 \Lambda_x + \mathcal{S} \times_2 \Lambda_y + \mathcal{S} \times_3 \Lambda_z = \mathcal{K}$$

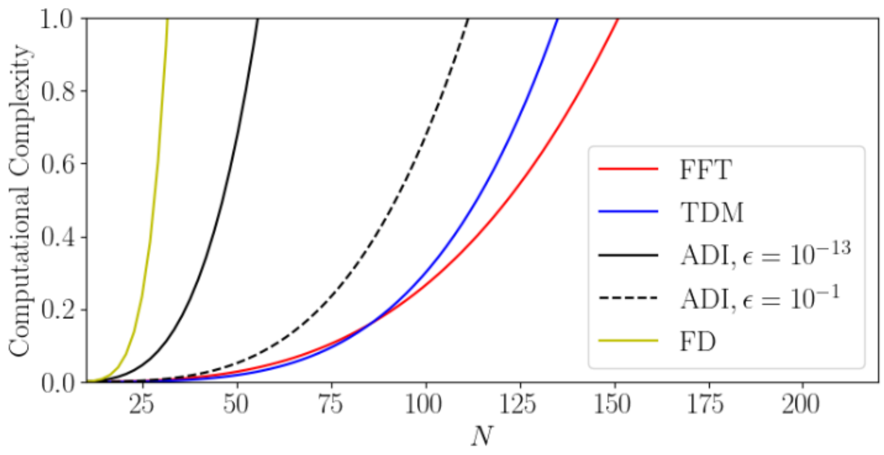
$$\mathcal{S}_{ijk} = \frac{\mathcal{K}_{ijk}}{\Lambda_{x,ii} + \Lambda_{y,jj} + \Lambda_{z,kk}}$$

Final form

Solver Algorithm and benchmarking

- Time complexity of algorithm

Algorithm	Time complexity
FD	$O(N^6)$
Green & FFT	$O(40N^3 log_2 N)$
ADI	$O(N^3(log_2 N)^3 log_2(1/\epsilon))$
TDM	$O(3N^4)$

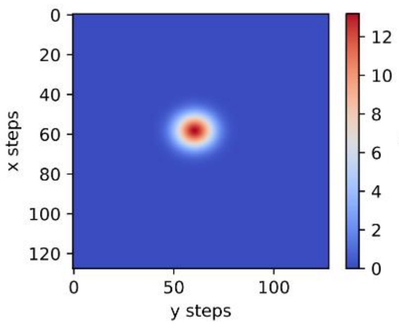


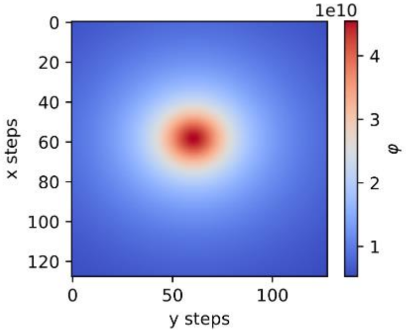
Solver Algorithm and benchmarking

- Benchmarking: open boundary

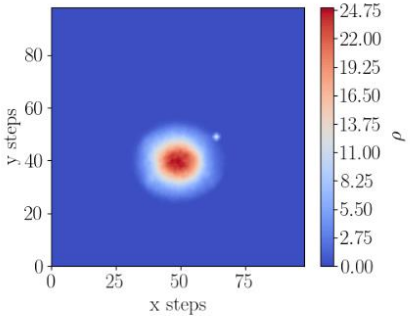
$$\rho(r) = \frac{Q}{(2\pi)^{3/2}\sigma^3} e^{-r^2/(2\sigma^2)}$$

$$\varphi(r) = \frac{1}{4\pi\epsilon_0}\frac{Q}{r} erf\left(\frac{r}{\sqrt{2}\sigma}\right)$$

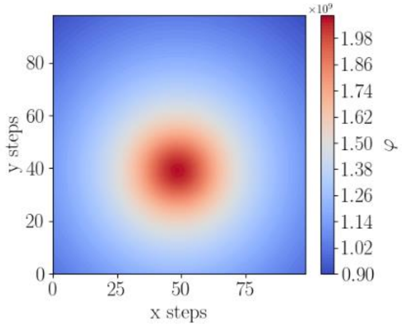




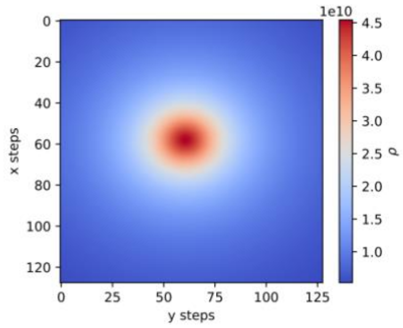
Random Distribution



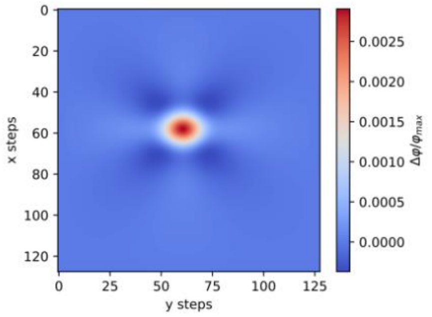
Green & FFT Solver



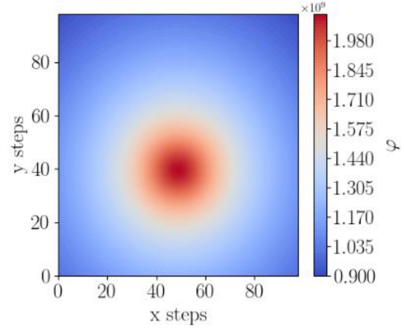
TDM Solver



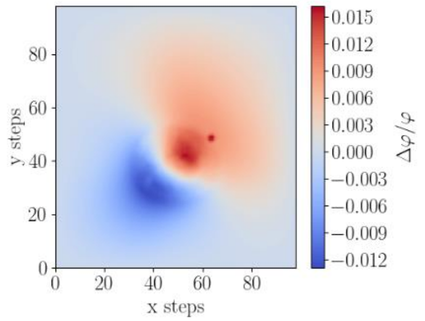
Compare to Analytical Solution



TDM Solver



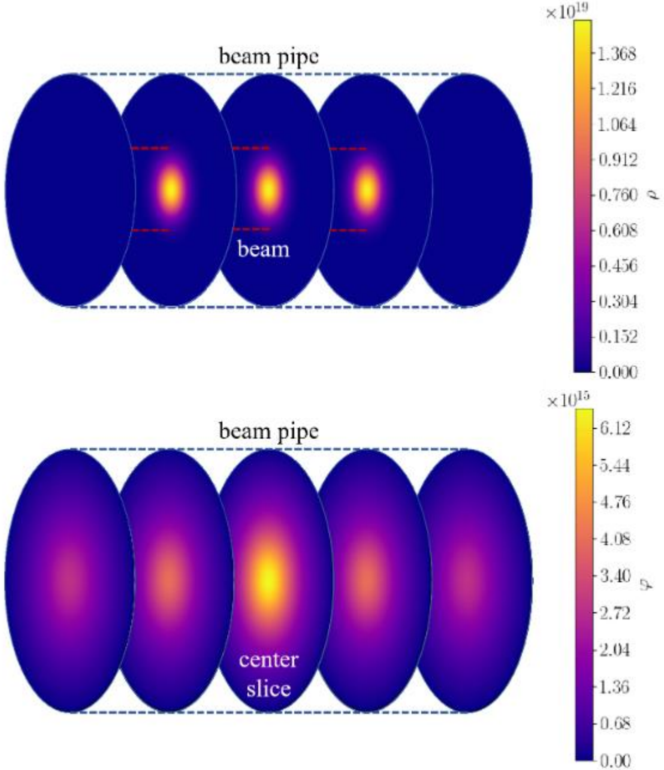
Compare to FFT Solution



Solver Algorithm and benchmarking

- Benchmarking: transverse round dirichlet boundary， longitude open boundary

$$\rho(r,z) = \frac{Q}{2\pi l \sigma^2} e^{-r^2/(2\sigma^2)} \left[H(z+\frac{l}{2}) - H(z-\frac{l}{2}) \right]$$

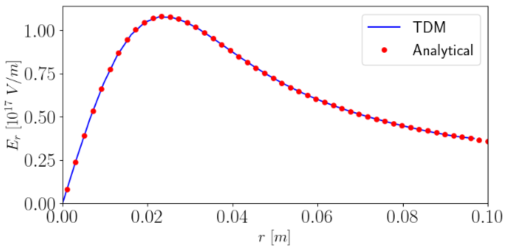


For a continuous beam with a transverse Gaussian distribution, the electric field it generates within a circular ideal conducting vacuum chamber is:

$$E(r) = \frac{\lambda}{2\pi\varepsilon_0} \frac{1 - e^{-r^2/(2\sigma^2)}}{r}$$

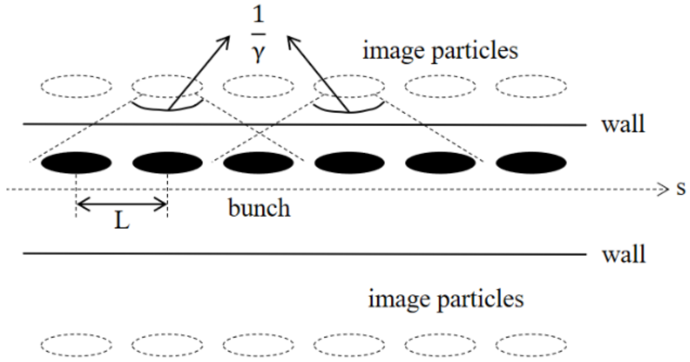
In reality, infinitely long beam streams do not exist. Therefore, by increasing the beam length, the electric field generated at the central slice of the beam can be used to approximate that of an infinitely long beam.

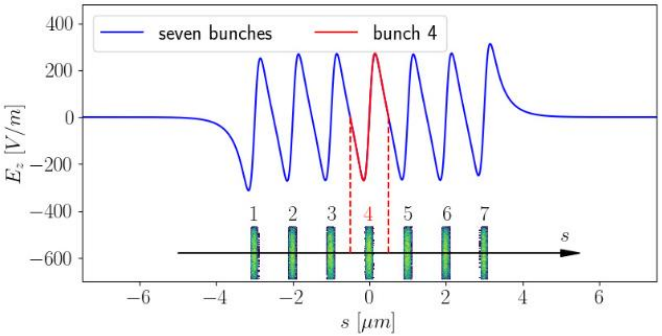
The central slice



Solver Algorithm and benchmarking

- Benchmarking: transverse round dirichlet boundary， longitude period boundary





For a beam comprising multiple micro-pulses, with transverse Dirichlet boundary conditions applied, the space charge force can generally be calculated using longitudinal periodic boundary conditions, owing to the periodic micro-pulse structure and the approximate uniformity of parameters across individual micro-pulses.

Assume there are seven micro-pulses in total.

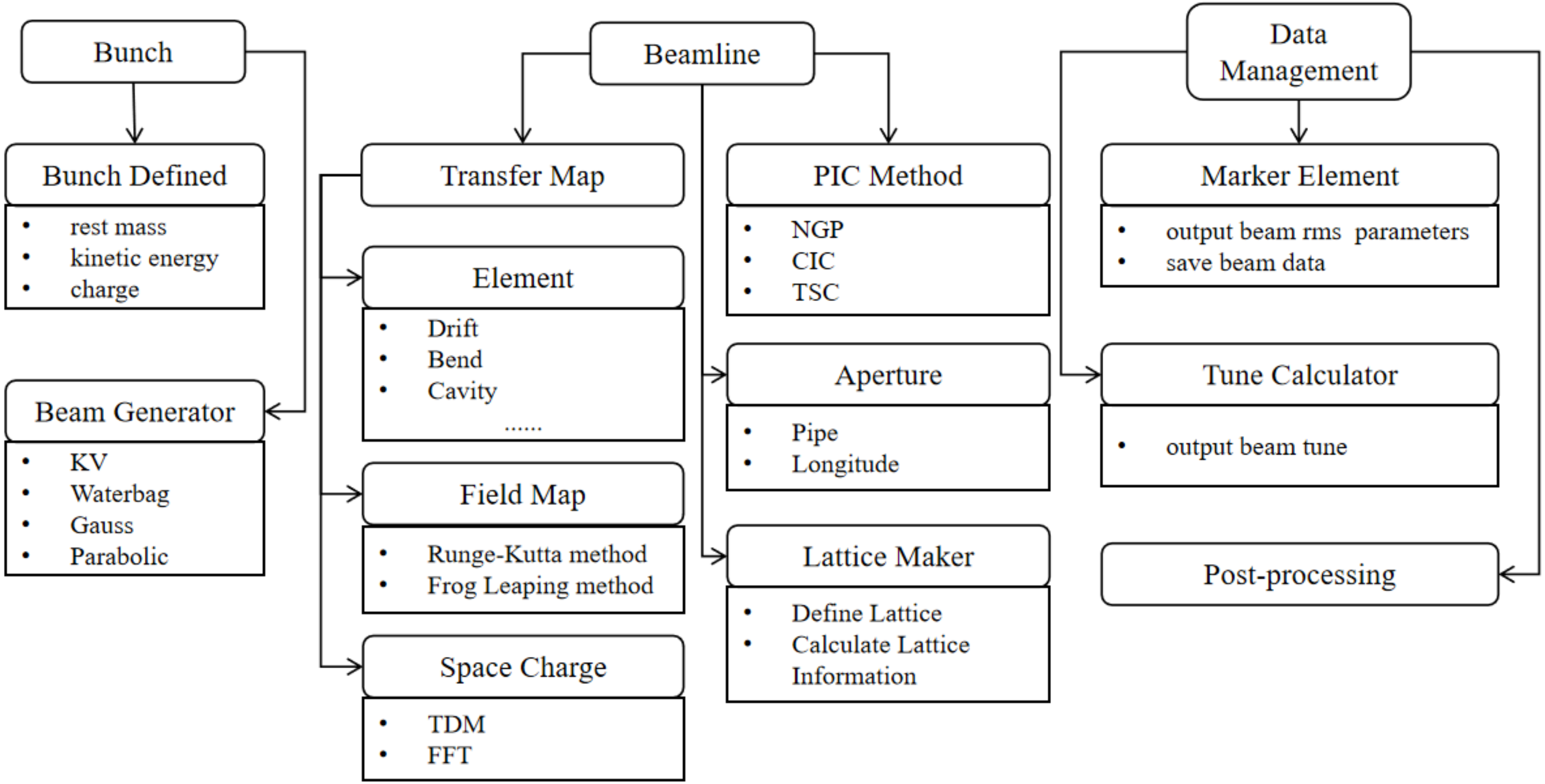
1. The seven micro-pulses are treated as a single entity and solved using transverse Dirichlet boundary conditions and longitudinal open boundary conditions.

2. Only the central micro-pulse is solved, using transverse Dirichlet boundary conditions and longitudinal periodic boundary conditions.

The results from both methods, as shown in the figure, are consistent. Therefore, for this type of multi-micro-pulse structure, it is sufficient to employ longitudinal periodic boundary conditions for the solution.

Some Applications

- Simulation code



Some Applications

- Application: image charge effect

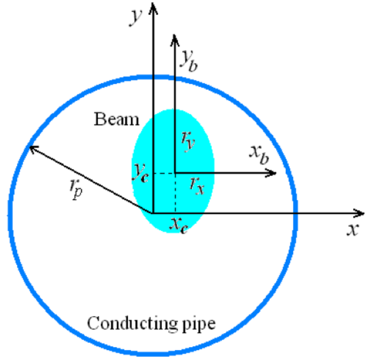


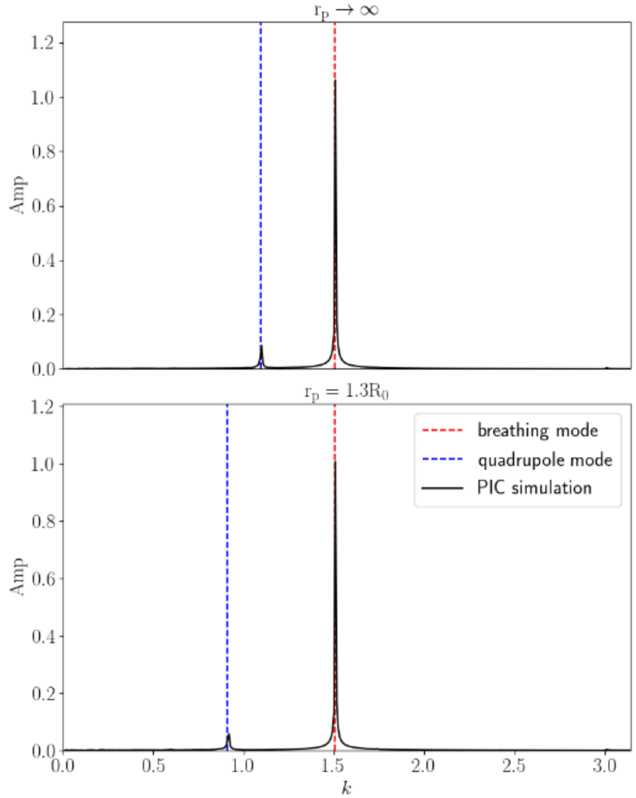
image charge effect

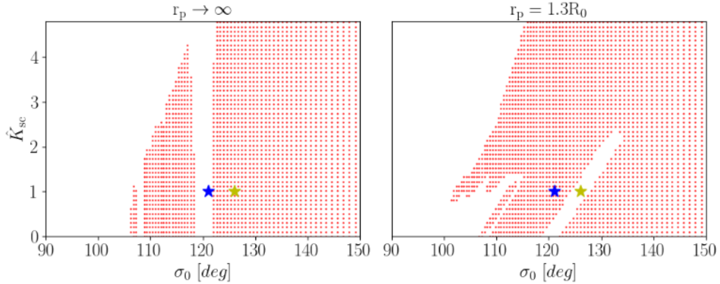
↓

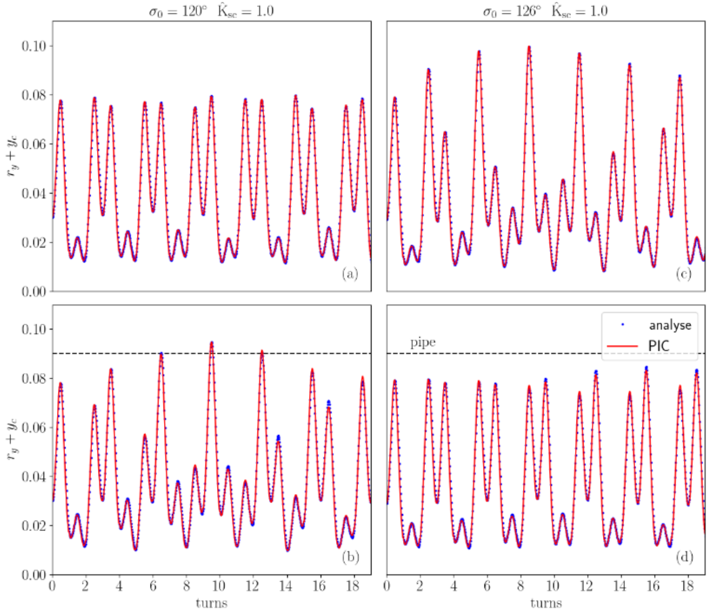
oscillation mode

↓

envelope Instability

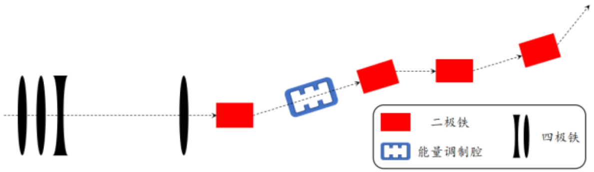






Some Applications

- Application: compare to Imapct-z



Angular Dispersion-induced Microbunching

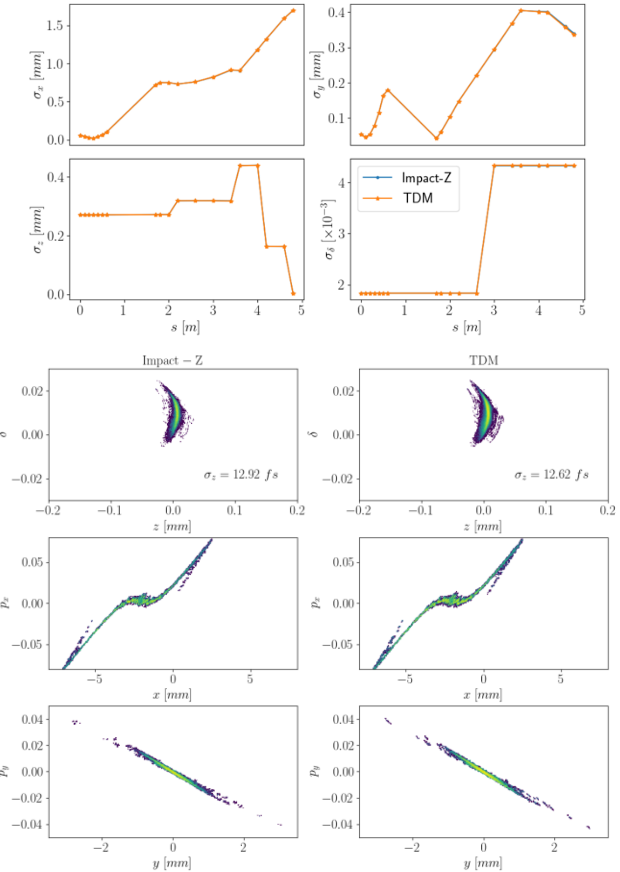
ADM

Modulation

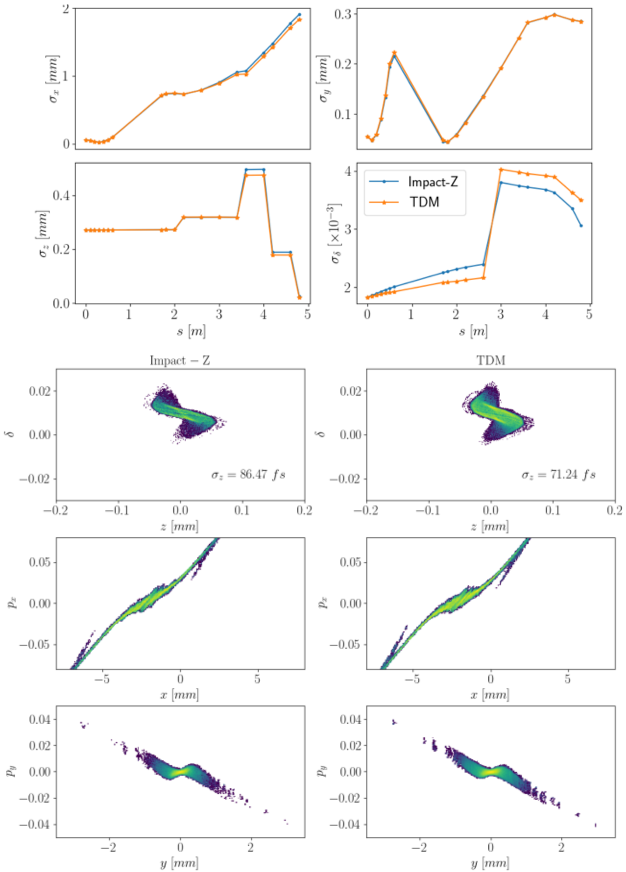
$$\boxed{z_1} = z_0 + m_{51}x + \boxed{m_{52}p_x} + m_{56}\delta$$

Initial Parameters	Value
$\boldsymbol{E_k}$	22.717 MeV
γ	45.46
z_0	900 fs

$Q_e = 0\ pC$

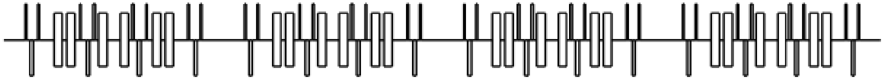


$Q_e = 100\ pC$



Some Applications

- Application: compare to Pyorbit



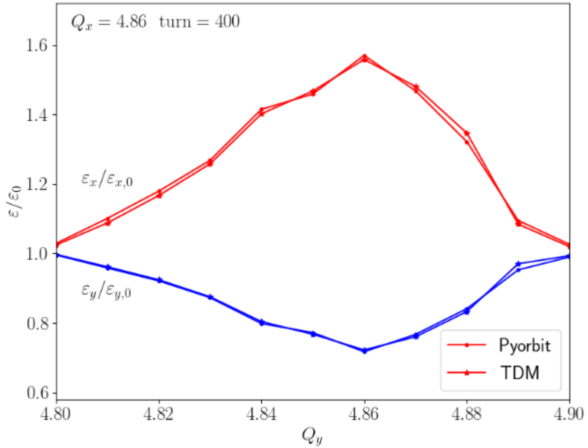
CSNS/RCS lattice

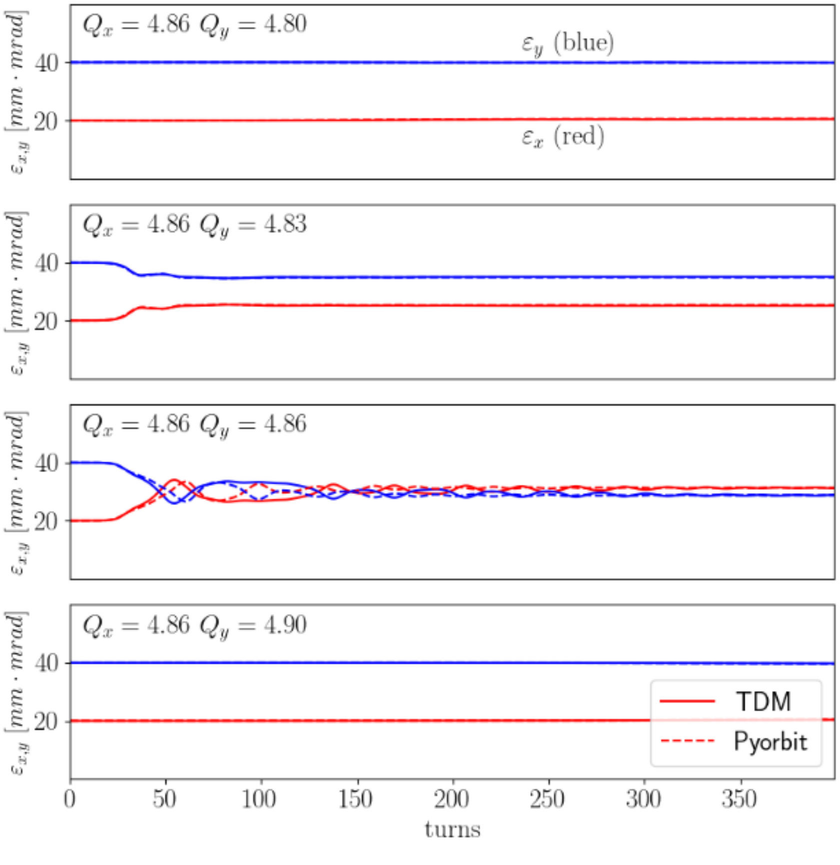
csns\rcs with fourfold symmetry produces structural resonances that satisfy the resonance condition,

$$mQ_x + nQ_y = 4 \times p, \qquad (m, n, p \text{ is integer})$$

Montague resonance,

$$2Q_x - 2Q_y = 0$$





Thank you !