Update on development for single-particle tracking with e-clouds

Konstantinos Paraschou[1,2], Giovanni Iadarola[1]

[1]CERN, Switzerland
[2]Aristotle University of Thessaloniki, Greece

Acknowledgements: R. De Maria, Y. Papaphillipou, L. Sabato, M. Schwinzerl

Electron Cloud WG
CERN, Friday, 29th November 2019

Previously on Electron Cloud WG

- To do long-term tracking we require symplecticity in 6D.
- Linear interpolation of the fields $E_x(x, y)$, $E_y(x, y)$ violates the symplecticity condition even in 4D ($\partial_x E_y \neq \partial_y E_x$).
- Symplecticity can be recovered by interpolating a scalar potential and taking analytical derivatives
- as long as interpolation scheme guarantees C^1 continuity.

[1] Electron Cloud Meeting #67, 10th May 2019
https://indico.cern.ch/event/811014/contributions/3379525/

Electron cloud kick

Strategy

Implement a 6D symplectic electron cloud kick[7] in tracking produced by the Hamiltonian $H(x, y, \zeta; s) = \frac{qL}{\beta^2\gamma mc^2}\phi(x, y, \zeta)\,\delta(s)$, where the scalar potential ϕ describes the e-cloud[8].

$$
\begin{aligned}
x &\mapsto x \\
p_x &\mapsto p_x - \frac{qL}{\beta^2\gamma mc^2}\frac{\partial\phi}{\partial x}(x, y, \zeta) \\
y &\mapsto y \\
p_y &\mapsto p_y - \frac{qL}{\beta^2\gamma mc^2}\frac{\partial\phi}{\partial y}(x, y, \zeta) \\
\zeta &\mapsto \zeta \\
\delta &\mapsto \delta - \frac{qL}{\beta^2\gamma mc^2}\frac{\partial\phi}{\partial \zeta}(x, y, \zeta)
\end{aligned}
$$

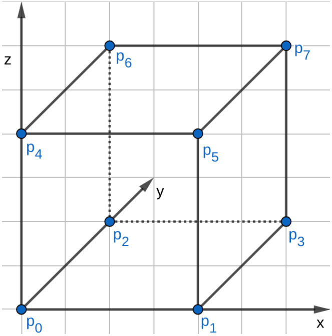
[7] Thin-lens, rigid beam, "recorded" pinch approximation, $\zeta = \frac{\beta}{\beta_0}s - \beta ct$

[8] see G. Iadarola, CERN-ACC-NOTE-2019-0033.

Tricubic Interpolation

Objective

If the e-cloud scalar potential, $\phi^{(i,j,k)}$ is known on a 3D grid, Tricubic Interpolation[9] can produce symplectic 6D kicks.



$$f(x, y, z) = \sum_{i=0}^{3} \sum_{j=0}^{3} \sum_{k=0}^{3} a_{ijk} x^i y^j z^k$$

a_{ijk} is found by imposing

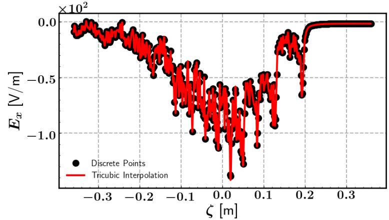
$$\left\{ f, \frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}, \frac{\partial^2 f}{\partial x \partial y}, \frac{\partial^2 f}{\partial x \partial z}, \frac{\partial^2 f}{\partial y \partial z}, \frac{\partial^3 f}{\partial x \partial y \partial z} \right\}$$

on the 8 corners.

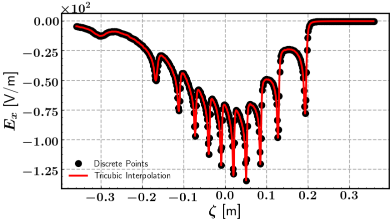
- Derivatives are evaluated with Finite Differences.

[9] Lekien, F & J. E., Marsden. (2005). Tricubic Interpolation in Three Dimensions. International Journal for Numerical Methods in Engineering. 63. 10.1002/nme.1296.

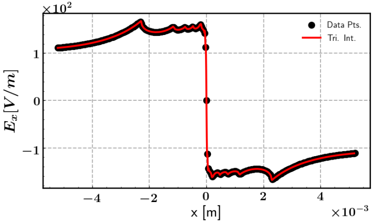
Average Pinch



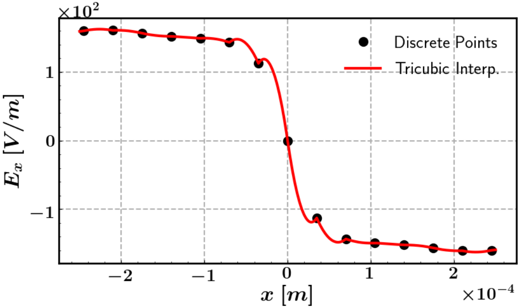
⇓



- Macroparticle noise can be **significantly** reduced by averaging many (2000) electron cloud simulations.



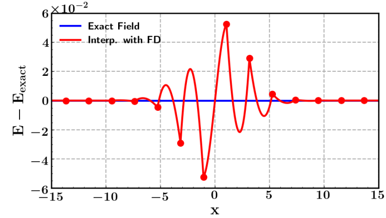
Interpolator Issues



Closer look reveals **irregularities**.

1D Analytical Investigation

$$\phi = \frac{x}{2} - \log\left(1 + e^{x}\right), \qquad E = \frac{1}{1 + e^{x}} - \frac{1}{2}$$

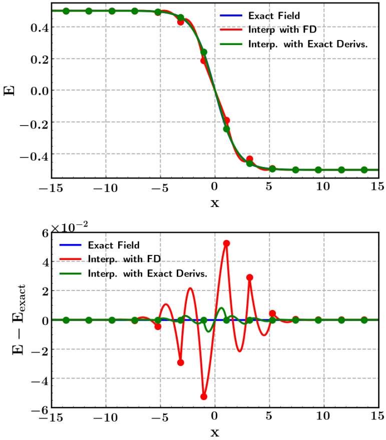


- Calculating $E = -\partial_x \phi$ with Finite Differences we observe the same irregularities when interpolating.

We need to improve accuracy of derivatives.

1D Analytical Investigation

$$\phi = \frac{x}{2} - \log\left(1 + e^{x}\right), \qquad E = \frac{1}{1+e^{x}} - \frac{1}{2}$$

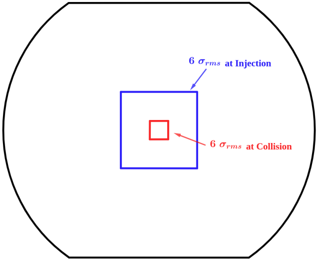


- Calculating $E = -\partial_x \phi$ with Finite Differences we observe the same irregularities when interpolating.
- When using the exact values of the derivative to interpolate, error is significantly reduced.
- Small error still there related to step size.

To improve the electron cloud fields we define a refinement procedure.

Refinement Procedure

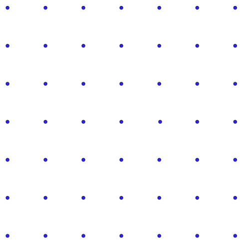
- Beam size is very small compared to chamber.
- We need to focus on a region around the beam.
- Even with 500 × 500 points, resolution is still not enough.



Refinement Procedure

We must improve ϕ and its derivatives.

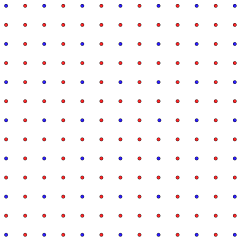
1. ρ, ϕ known on a coarse grid.
2.
3.
4.
5.
6.
7.



Refinement Procedure

We must improve ϕ and its derivatives.

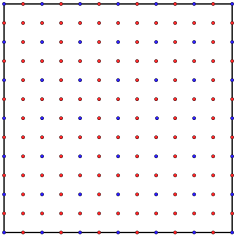
1. ρ, ϕ known on a coarse grid.
2. Create a finer grid.
3.
4.
5.
6.
7.



Refinement Procedure

We must improve ϕ and its derivatives.

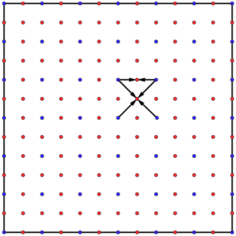
1. ρ, ϕ known on a coarse grid.
2. Create a finer grid.
3. Linearly interpolate ϕ-boundary.
4.
5.
6.
7.



Refinement Procedure

We must improve ϕ and its derivatives.

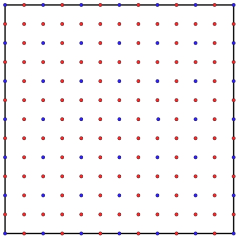
1. ρ, ϕ known on a coarse grid.
2. Create a finer grid.
3. Linearly interpolate ϕ-boundary.
4. Linearly interpolate ρ inside.
5.
6.
7.



Refinement Procedure

We must improve ϕ and its derivatives.

1. ρ, ϕ known on a coarse grid.
2. Create a finer grid.
3. Linearly interpolate ϕ-boundary.
4. Linearly interpolate ρ inside.
5. Re-solve Poisson equation on finer grid.
6.
7.

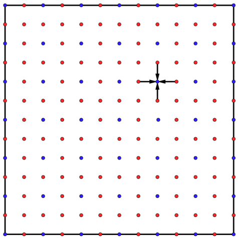


At this point, ϕ solution is much smoother.

Refinement Procedure

We must improve ϕ and its derivatives.

1. ρ, ϕ known on a coarse grid.
2. Create a finer grid.
3. Linearly interpolate ϕ-boundary.
4. Linearly interpolate ρ inside.
5. Re-solve Poisson equation on finer grid.
6. Use Finite differences on finer grid to calculate derivatives:
$$\left\{\phi, \frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial\zeta}, \frac{\partial^2\phi}{\partial x\partial y}, \frac{\partial^2\phi}{\partial x\partial\zeta}, \frac{\partial^2\phi}{\partial y\partial\zeta}, \frac{\partial^3\phi}{\partial x\partial y\partial\zeta}\right\}.$$
7.

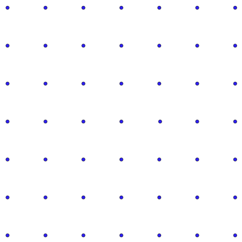


We now have a better interpolation (better derivatives, smaller step size).

Refinement Procedure

We must improve ϕ and its derivatives.

1. ρ, ϕ known on a coarse grid.
2. Create a finer grid.
3. Linearly interpolate ϕ-boundary.
4. Linearly interpolate ρ inside.
5. Re-solve Poisson equation on finer grid.
6. Use Finite differences on finer grid to calculate derivatives:
$$\left\{\phi, \frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial\zeta}, \frac{\partial^2\phi}{\partial x\partial y}, \frac{\partial^2\phi}{\partial x\partial\zeta}, \frac{\partial^2\phi}{\partial y\partial\zeta}, \frac{\partial^3\phi}{\partial x\partial y\partial\zeta}\right\}.$$
7. Keep **refined** ϕ and derivatives on the original grid.



Better interpolation without sacrificing too much memory.

Computational requirements

- Grid in PyECLOUD already pushes the limits of a typical RAM.
- Significant development to optimize memory consumption during the refinement procedure.
- Even then, solution of Poisson equation on such a fine grid can easily exceed 100 GBs of memory.

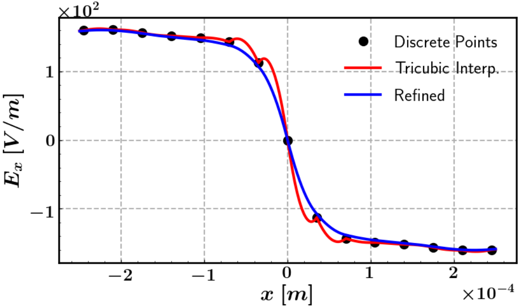
Computational requirements

- Grid in PyECLOUD already pushes the limits of a typical RAM.
- Significant development to optimize memory consumption during the refinement procedure.
- Even then, solution of Poisson equation on such a fine grid can easily exceed 100 GBs of memory.

Used special resources:

- **LIUPSGPU** machine of ABP with 24 cores and 256 GB memory.
- HTCondor nodes (BigMem) with 24 cores and 1 TB memory.
- HTCondor nodes (BigMem) with 48 cores and 512 GB memory.

Interpolator Issues

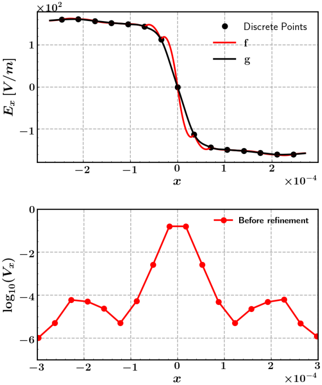


Irregularities are significantly suppressed through this procedure.

Quantifying artifacts

In order to systematically study these artifacts, they must be somehow quantified.

- A Tricubic Interpolation on the field would provide a **much more accurate interpolation** but not **not be symplectic**.
- To have an indicator of the accuracy we compare the symplectic interpolation from the potential (we call it f) against the accurate interpolation on the field (we call it g)



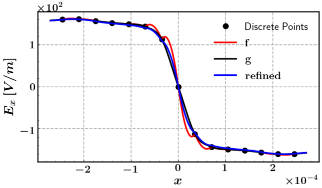
Quantifier:

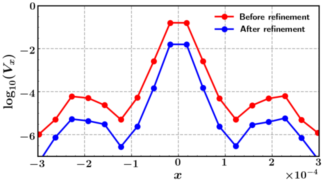
$$V_x = \frac{\int_{cell} (f-g)^2 dV}{\int_{cell} g^2 dV}$$

Quantifying artifacts

In order to systematically study these artifacts, they must be somehow quantified.

- A Tricubic Interpolation on the field would provide a **much more accurate interpolation** but not **not be symplectic**.
- To have an indicator of the accuracy we compare the symplectic interpolation from the potential (we call it f) against the accurate interpolation on the field (we call it g)





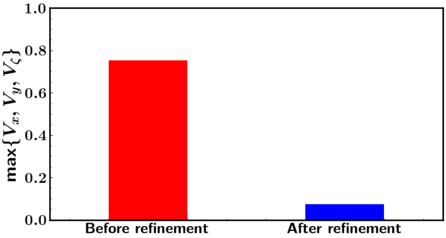
Quantifier:

$$V_x = \frac{\int_{cell} (f - g)^2 dV}{\int_{cell} g^2 dV}$$

Quantifying artifacts

$$V_{x,y,\zeta} = \frac{\int_{cell} (f-g)^2 dV}{\int_{cell} g^2 dV}$$

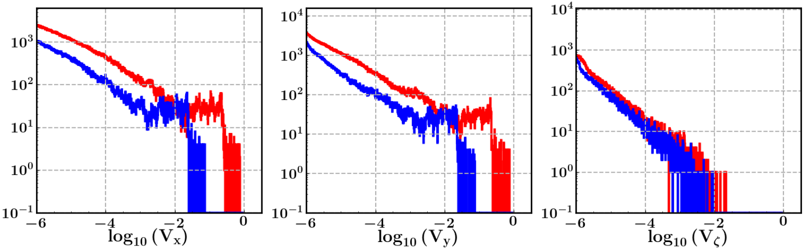
Maximum of $V_{x,y,\zeta}$ gives a good **quantitative measure**.



- Up to an order of magnitude better with just $dx \leftarrow \frac{dx}{2}$, $dy \leftarrow \frac{dy}{2}$, $d\zeta \leftarrow \frac{d\zeta}{2}$.

Global Picture

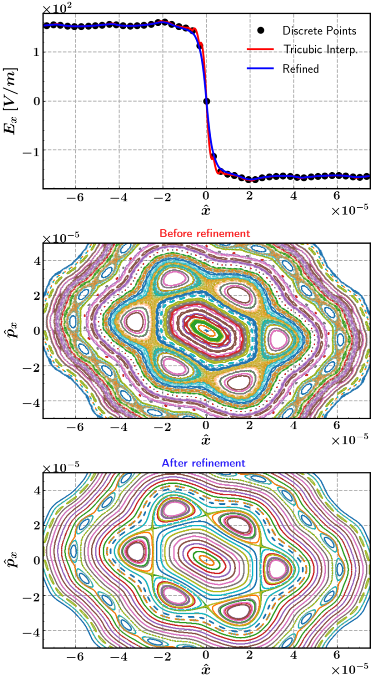
Histograms of $\log_{10}(V_{x,y,\zeta})$ before and after the refinement can give a **global picture**.



- Orders of magnitude gained with just $dx \leftarrow \frac{dx}{2}$, $dy \leftarrow \frac{dy}{2}$, $d\zeta \leftarrow \frac{d\zeta}{2}$.

Impact of irregularities

- Simple tracking of a linear 2D phase space rotation and an e-cloud symplectic kick.
- Very important to minimize irregularities.
- By reducing them, there is significant impact on the beam particle motion.



Recipe for a single electron cloud interaction

1. Average over the **many** simulations of the same electron cloud pinch to reduce particle noise.
2. Take advantage of symmetry conditions.
3. **Refine** grids to improve derivative calculation.
4. **Use the obtained map within tracking code (sixtracklib)**
   - Development almost completed, testing has started

Conclusion

Status:

- We know how to conserve the symplecticity of the e-cloud interaction (tricubic interpolation)
- We know how to mitigate macro-particle noise of PyECLOUD simulations (average over several pinches).
- We know how to minimize artifacts of interpolation scheme (derivatives evaluation on refined grid).

Future Developments:

- Benchmark interpolation scheme on analytic ζ-dependent Hamiltonians (RF-multipoles).
- Check behaviour of “long-term” observables, e.g. Dynamic Aperture.
- Install e-clouds in the arcs of the LHC (starting from MAD-X model) and study losses and emittance at injection energy.

Thank you for your attention!

Konstantinos Paraschou