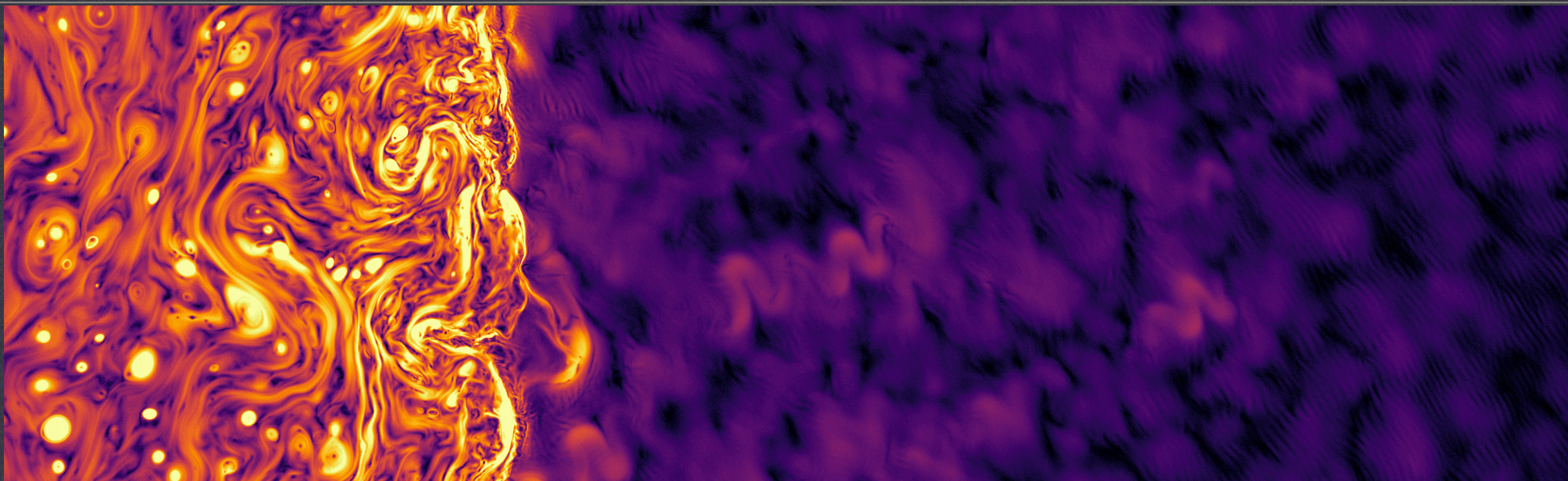


Is There a Better PIC Simulation?

Anatoly Spitkovsky
(Princeton University)

OBLIGATORY DISCLAIMERS:

1. This research is not only not finished, it has not really been started.

2. It is brought to your attention in the hopes that you will shoot it down and spare me the agony. If you can't shoot it down, I hope you will help me solve it.

3. Some dirty (numerical) laundry will be aired.

4. This presentation may cause you to (briefly) question reality.

5. I hope that at the end of it you will be able to stop worrying and learn to love PIC simulations (again).

SO, WHAT'S THE QUESTION?

COLLISIONLESS PLASMA AND PARTICLE-IN-CELL METHOD

We are solving a Maxwell-Vlasov system by discretizing plasma with particles

$$\frac{\partial f}{\partial t} + \vec{v} \cdot \frac{\partial f}{\partial \vec{x}} + \frac{q}{m}\left(\vec{E} + \frac{\vec{v} \times \vec{B}}{c}\right) \cdot \frac{\partial f}{\partial \vec{v}} = 0$$

$$\nabla \cdot \vec{E} = 4\pi \int q f d^3\vec{v},$$

$$\nabla \times \vec{B} = \frac{1}{c}\frac{\partial \vec{E}}{\partial t} + \frac{4\pi}{c}\int q f \vec{v} d^3\vec{v},$$

$$\nabla \cdot \vec{B} = 0, \quad \nabla \times \vec{E} = -\frac{1}{c}\frac{\partial \vec{B}}{\partial t}.$$

$$\frac{d\vec{v}_j}{dt} = \frac{q_j}{m_j}(\vec{E} + \frac{\vec{v}_j \times \vec{B}}{c})$$

$$\frac{d\vec{x}}{dt} = \vec{v}$$

$$\nabla \times \vec{E} = -\frac{1}{c}\frac{\partial \vec{B}}{\partial t}$$

$$\nabla \cdot \vec{B} = 0, \quad \nabla \cdot \vec{E} = 4\pi\rho$$

$$\nabla \times \vec{B} = \frac{4\pi\vec{j}}{c} + \frac{1}{c}\frac{\partial \vec{E}}{\partial t}$$

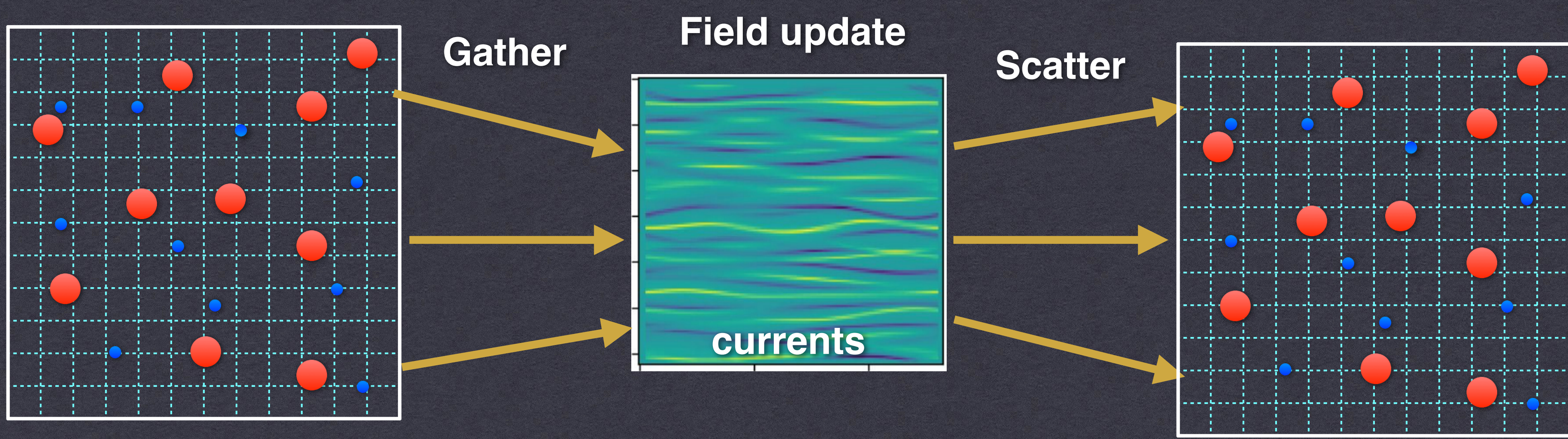
$$\rho(\vec{x}) = \sum_j q_j \delta(\vec{x} - \vec{x}_j)$$

$$\vec{j}(\vec{x}) = \sum_j q_j \vec{v}_j \delta(\vec{x} - \vec{x}_j)$$

Fields know about particles only through current (in EM PIC)

COLLISIONLESS PLASMA AND PARTICLE-IN-CELL METHOD

PIC simulation as a “transformer network”

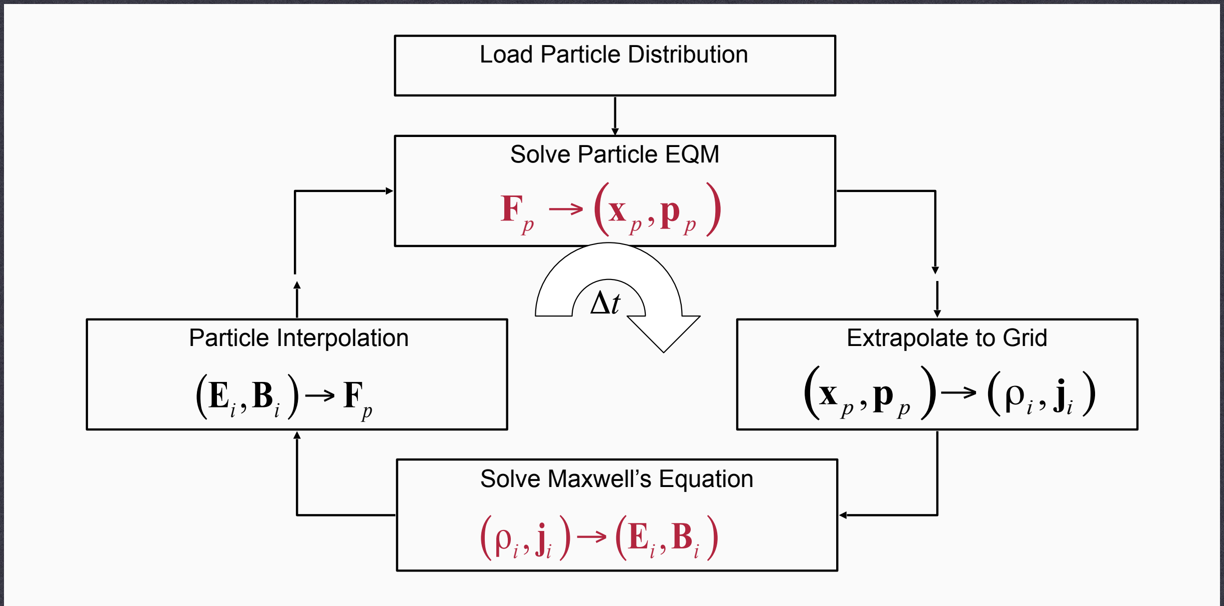


Current array is a reduction of 6D plasma dynamics

Can we filter this array in a smart way to bring out interesting features?

COLLISIONLESS PLASMA AND PARTICLE-IN-CELL METHOD

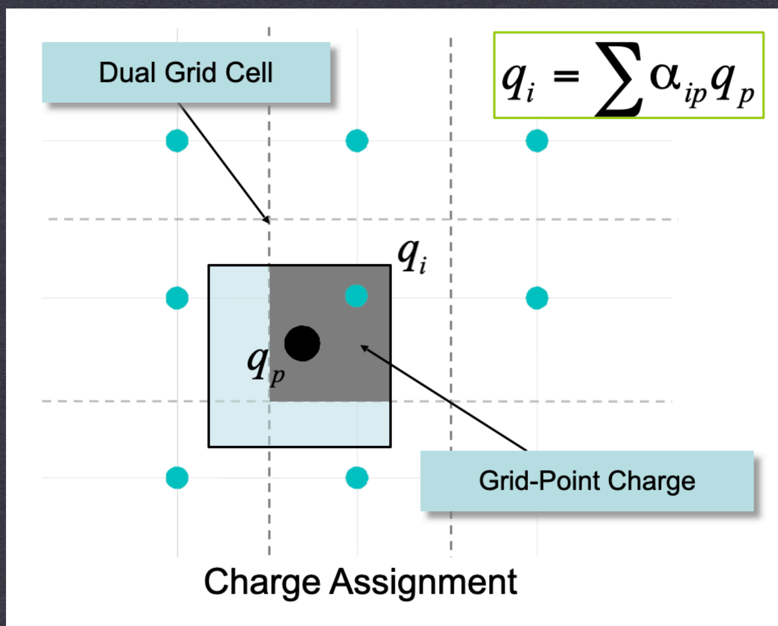
PIC time loop

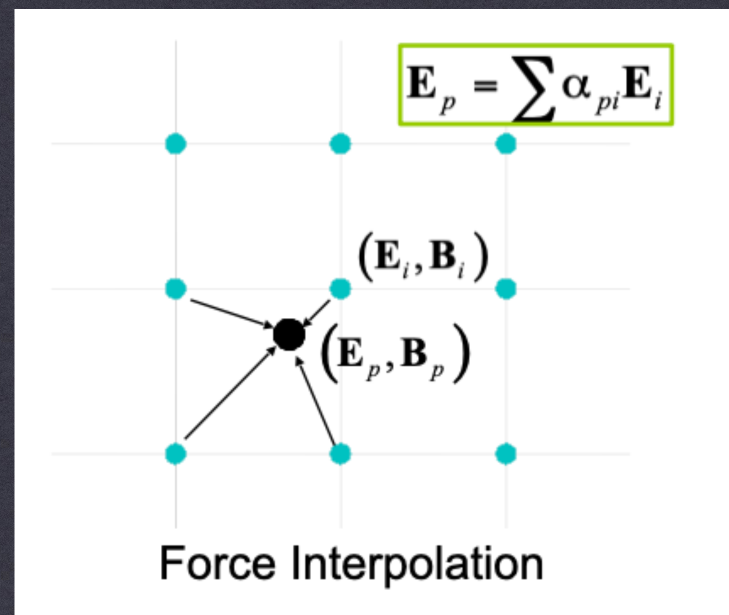


Extrapolation (scatter) step is done via particle shape function

COLLISIONLESS PLASMA AND PARTICLE-IN-CELL METHOD

PIC scatter and gather





$\alpha_{pi} := \alpha_{ip}$ (zero self-force)

Particle shape function is used for scatter and gather

COLLISIONLESS PLASMA AND PARTICLE-IN-CELL METHOD

The force law between finite-size particles

The finite-size particle considerably reduces the coulomb collision.

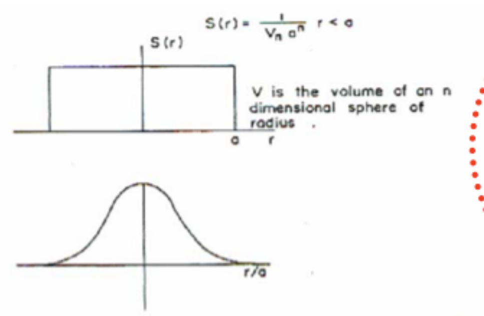


FIG. 4. Square and Gaussian charge shapes—two shapes often used for finite-sized particles.

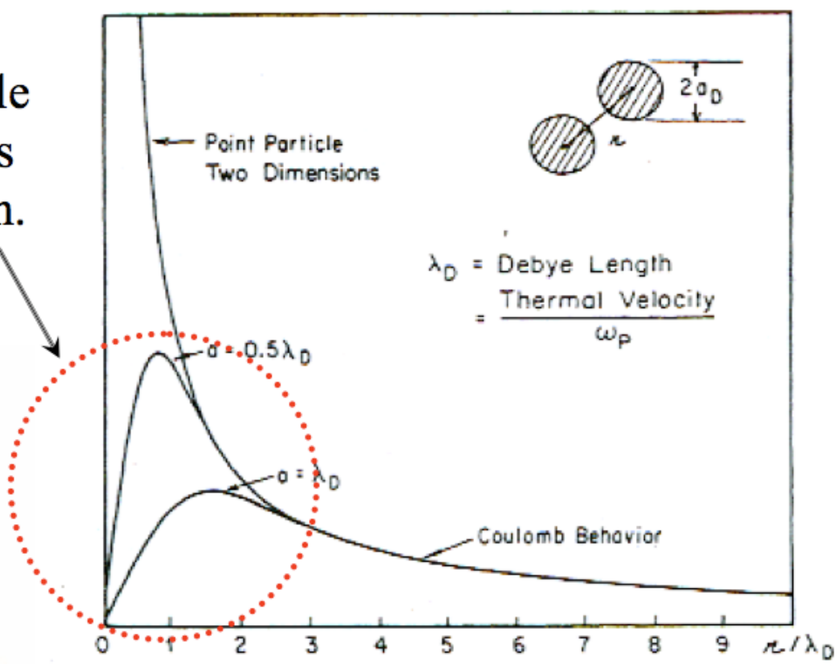


FIG. 2. Force law between finite-size particles in two dimensions for various sized particles. A Gaussian-shaped charge-density profile was used.

COLLISIONLESS PLASMA AND PARTICLE-IN-CELL METHOD

Particle shape also used to reduce noise

Charge Assignment and Force Evaluation by Cloud-in-Cell in 1D

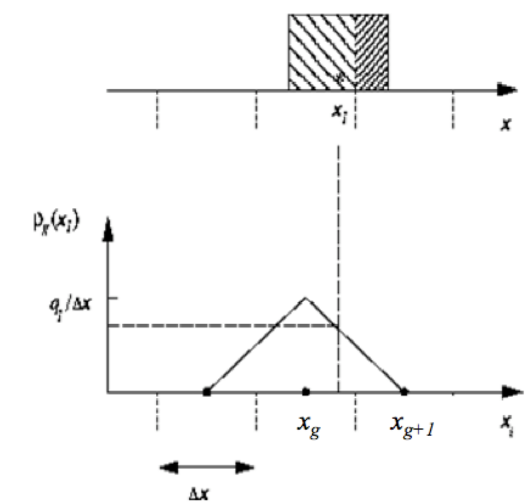
To ensure momentum conservation, the same interpolation scheme is used to compute the force on a particle as was used to perform the assignment of the particles charge to the mesh.

$$\rho_g = q_i \frac{x_{g+1} - x_i}{\Delta x}$$

$$\rho_{g+1} = q_i \frac{x_i - x_g}{\Delta x}$$

$$F_i = q_i\left(\frac{x_{g+1} - x_i}{\Delta x} E_g + \frac{x_i - x_g}{\Delta x} E_{g+1}\right)$$

$$where\ x_g \le x_i \le x_{g+1}$$

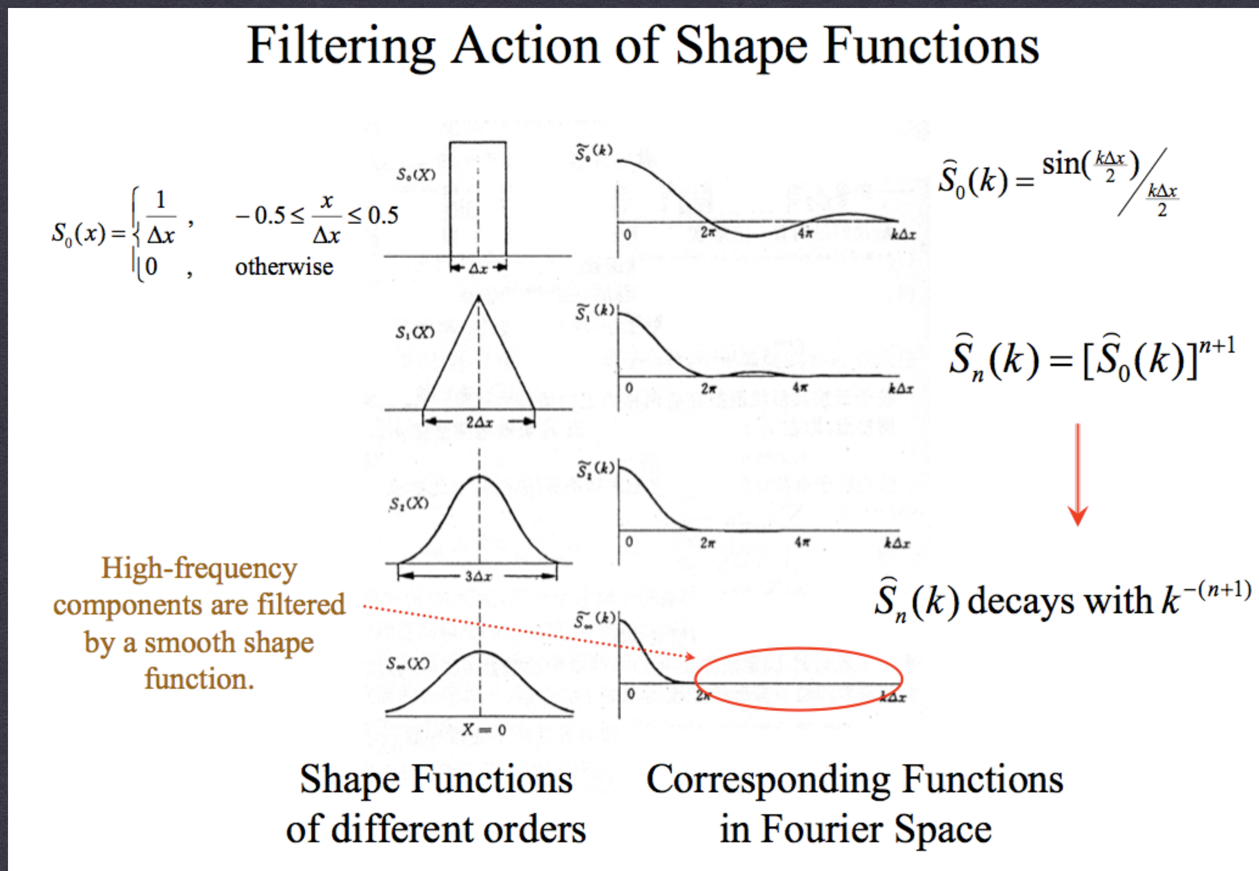


COLLISIONLESS PLASMA AND PARTICLE-IN-CELL METHOD

Particle shape also used to reduce noise

$$\mathbf{F}(r_j) = q \int S(r - r_j)\mathbf{E}(r)\, d^n r$$

$$\nabla \cdot \mathbf{E} = 4\pi q \int f(r', \mathbf{v}')\, s(r - r')\, d^n r'\, d^n v'$$

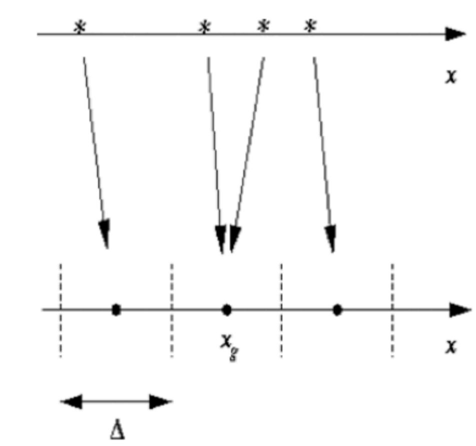


COLLISIONLESS PLASMA AND PARTICLE-IN-CELL METHOD

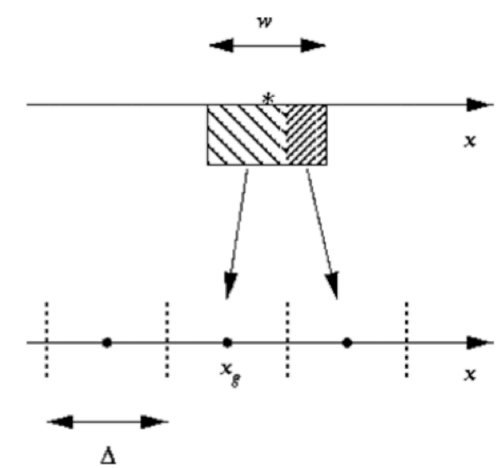
Typical PIC codes use compact shape functions for efficiency reasons.

Charge conservative codes have the lowest order (NGP) deposition in direction of motion of particle and higher order in transverse dimension (current deposit)

Non-charge conservative codes typically use cloud-in-cell in all directions.



Charge Assignment by
(Nearest Grid Point) NGP in 1D.



Charge Assignment by
(Cloud-in-Cell) CIC in 1D.

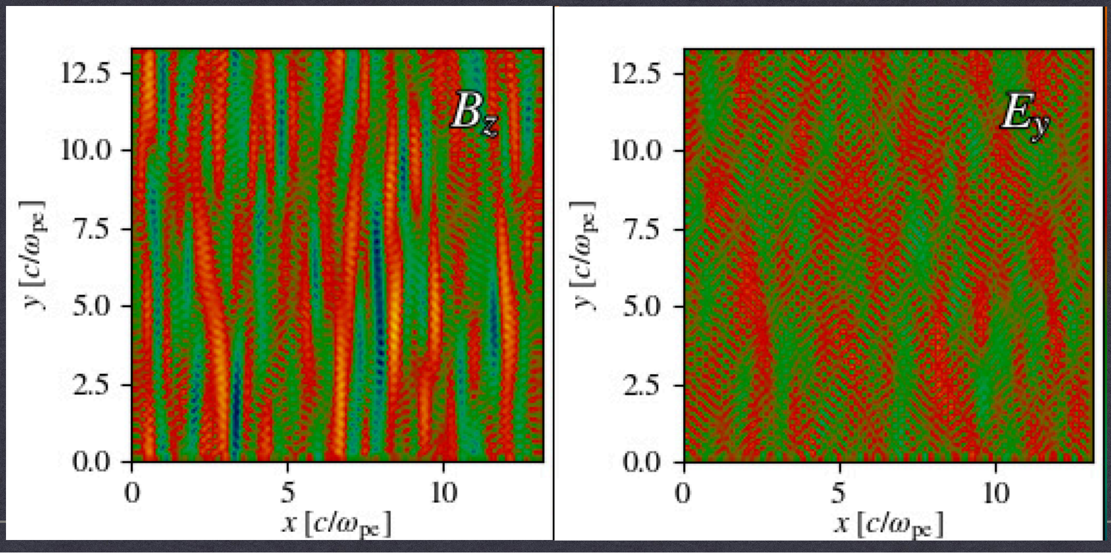
COLLISIONLESS PLASMA AND PARTICLE-IN-CELL METHOD

The end result of this is that the current deposition is very noisy.

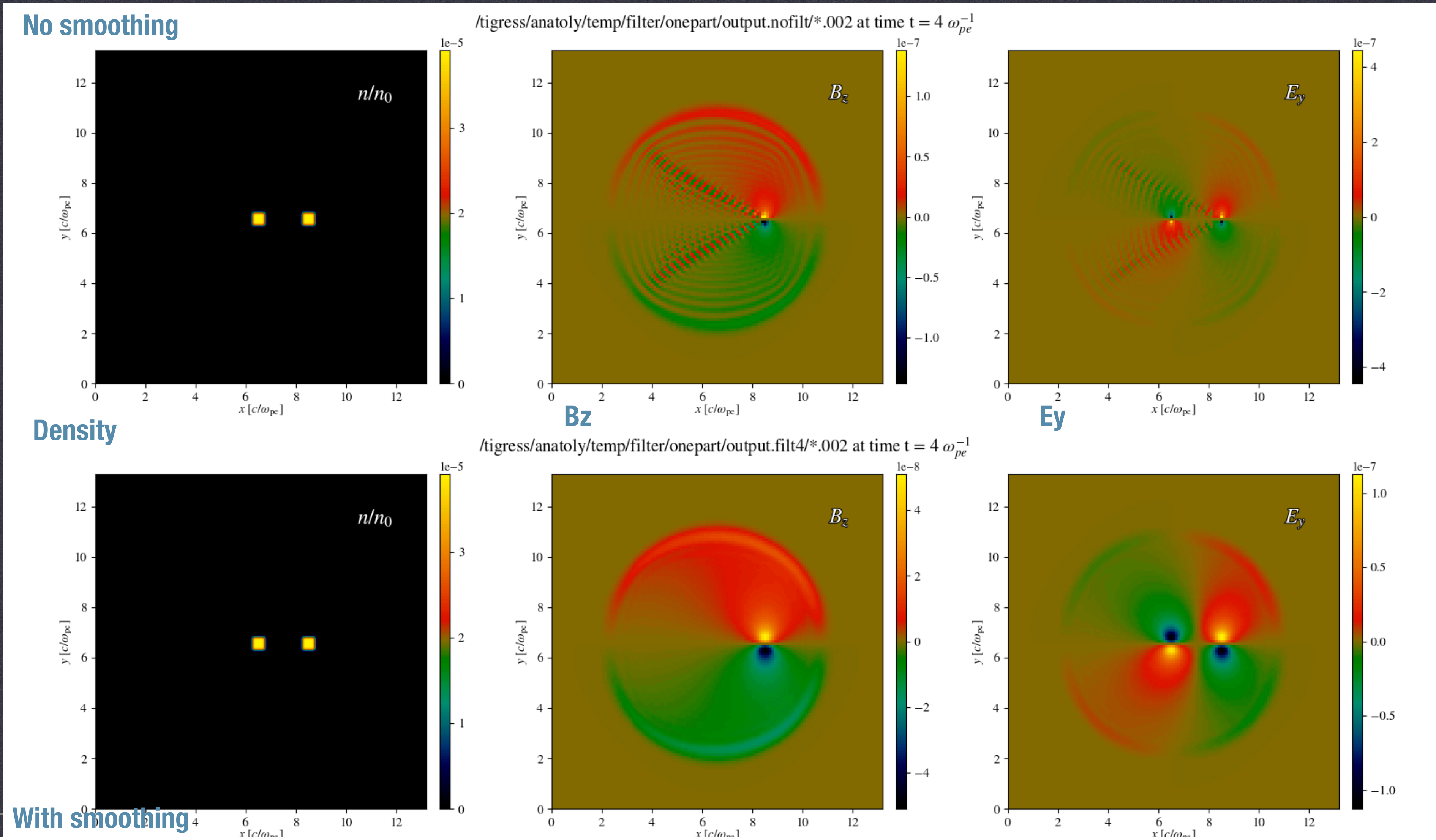
There are two kinds of noise: particle statistics noise (sensitive to number of particles per cell — same as in ES codes)
and
electromagnetic noise, specific to EM PIC: sharp jumps in current cell-to-cell cause high frequency EM waves (not very sensitive to ppc).

Dirty laundry:

Weibel instability with no current processing

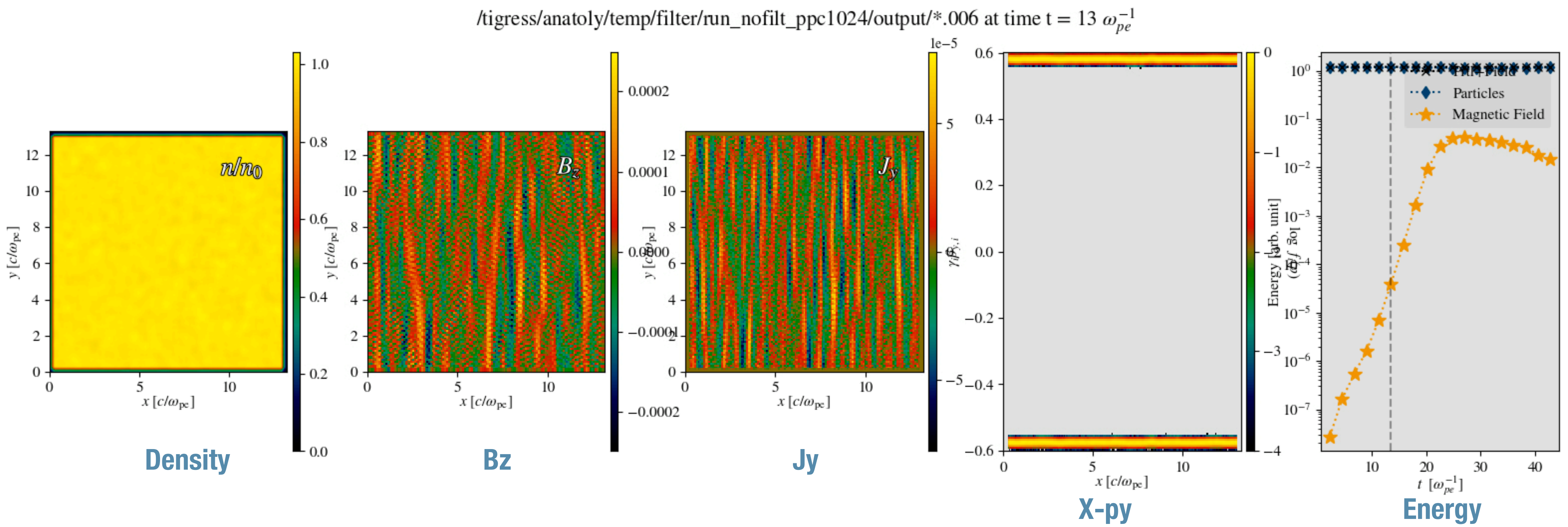


SINGLE PARTICLE FIELDS IN EM PIC



COLLISIONLESS PLASMA AND PARTICLE-IN-CELL METHOD

Running Weibel instability with no smoothing



PIC CURRENT FILTERING

This is typically solved with a healthy dose of “digital filter:”

Successive passes of 1-2-1 weighting in all directions.

Replace

$$\phi_j \quad \text{with} \quad \frac{W\phi_{j-1} + \phi_j + W\phi_{j+1}}{1 + 2W}$$

“Binomial” filter for W=1/2

Improves overall accuracy and reduces noise at short wavelengths (smoothing or attenuating)
Improves agreement with theory at long wavelengths k delta x -> 0. Sometimes combined with “compensator step”. Filtering works because it’s a linear problem.

In Fourier codes can be done in k-space. In finite difference codes — in grid space. For memory efficiency, can be done in multiple passes.

Filtering of current

Fourier transform:

$$\phi_{\text{filtered}}(k) = \sum_{j=1}^{N} \frac{W\phi_{j-1} + \phi_j + W\phi_{j+1}}{1+2W} e^{ikX_j}$$

$$\phi_f(k) = \frac{1 + 2W\cos k\Delta x}{1+2W}\phi_0(k) = \mathrm{SM}_W(\theta)\phi_0(k),$$

$$\theta \equiv k\Delta x$$

W<0.5, or SM reverses sign
W=0.5, SM always positive, approaches 0

Application of filter N times: $\cos^{2N}(\theta/2)$

$$\text{three point:} \quad \frac{1}{4}(1, 2, 1) \quad \rightarrow \cos^2\frac{\theta}{2}$$

$$\text{five point:} \quad \frac{1}{16}(1, 4, 6, 4, 1) \quad \rightarrow \cos^4\frac{\theta}{2}$$

$$\text{seven point:} \quad \frac{1}{64}(1, 6, 15, 20, 15, 6, 1) \rightarrow \cos^6\frac{\theta}{2}$$

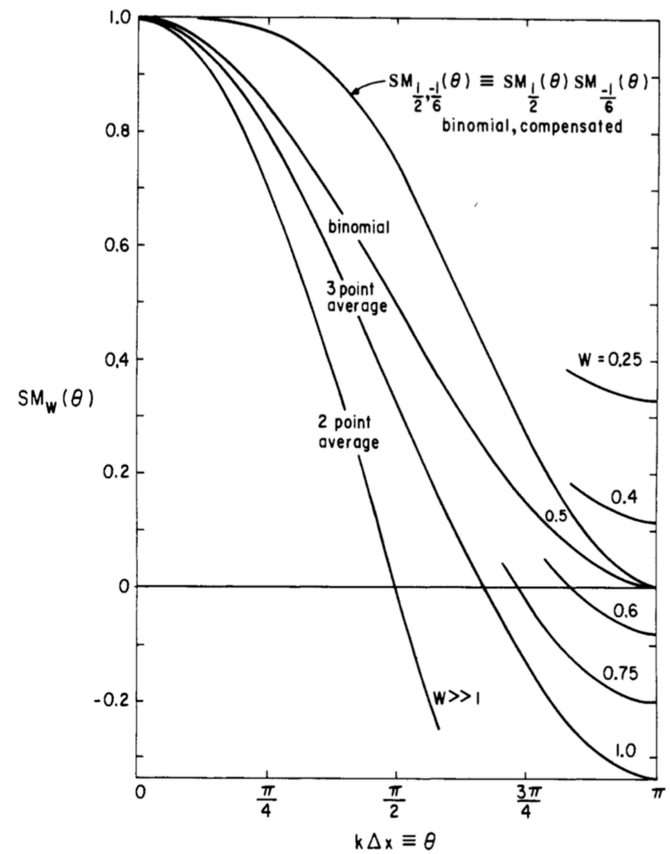


Figure Ca Smoothing function $\mathrm{SM}_W(\theta)$ of (5) for various W. The two and three point averages (as well as any $W > 0.5$) produce $\mathrm{SM}_W(\theta) < 0$ which alters the physics undesirably. Using first $W = 0.5$, then $W = -1/6$ produces the compensated curve shown.

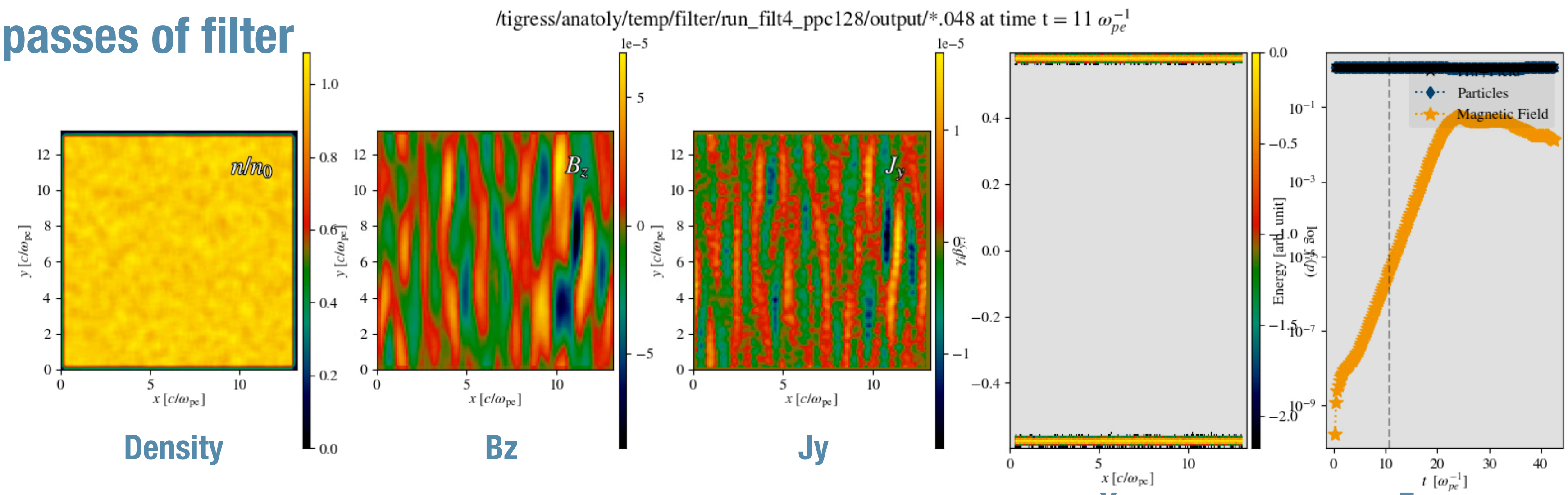
W=0.5 is "binomial" filter. W=-1/6 is "compensator" $(1/16)(-1, 4, 10, 4, -1)$

Filter can be called many times — optimization is essential

see Birdsall & Langdon 1991, Appendix C.

EXAMPLE: WEIBEL INSTABILITY

4 passes of filter



Density

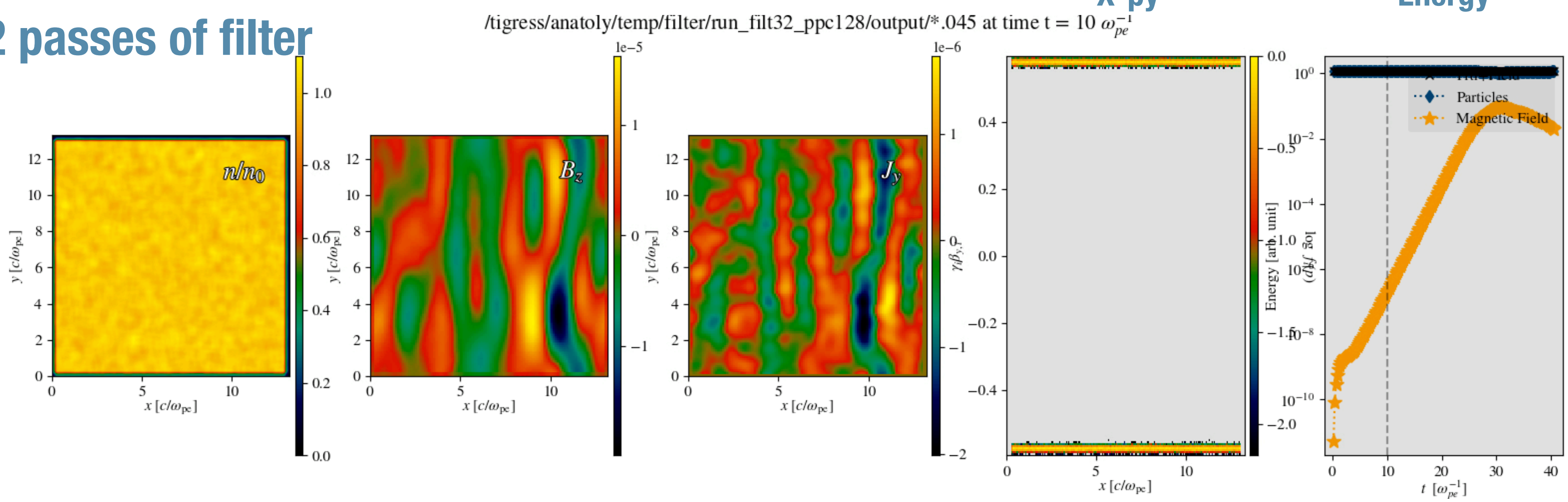
Bz

Jy

X-py

Energy

32 passes of filter



"Numerical stabilizer works fine, but it is physically disgusting."

Bruno Despres (Vienna, 2023)

Digital filtering is spreading the particles over a wide area, equivalently to a Gaussian shape. This does help with the noise.

If not used excessively, it kills noise under skin depth scale, although sometimes one can get carried away with small skin depths and large number of filters.

It is another free variable to play with for convergence. BTW, what does convergence mean in PIC?

Are there other filters we could use?

FILTER EXPERIMENTS

We can think of filtering as spreading individual particle shapes, or as processing the current array (an image).

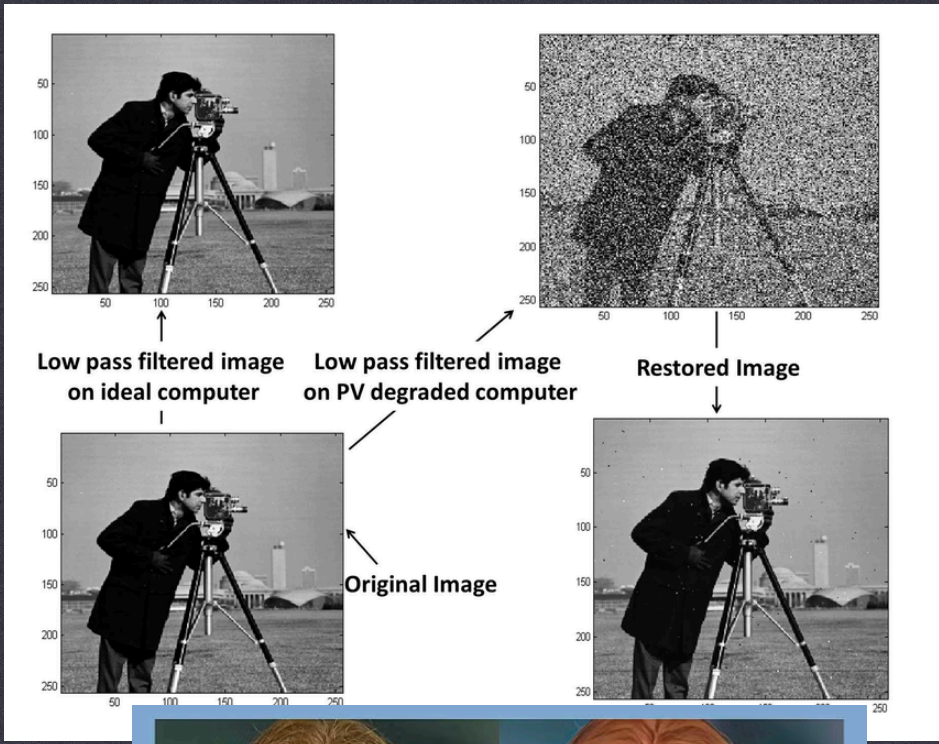
There are a number of techniques for image filtering, including noise removal, edge detection, feature segmentation, etc., etc. And even some fun ones…

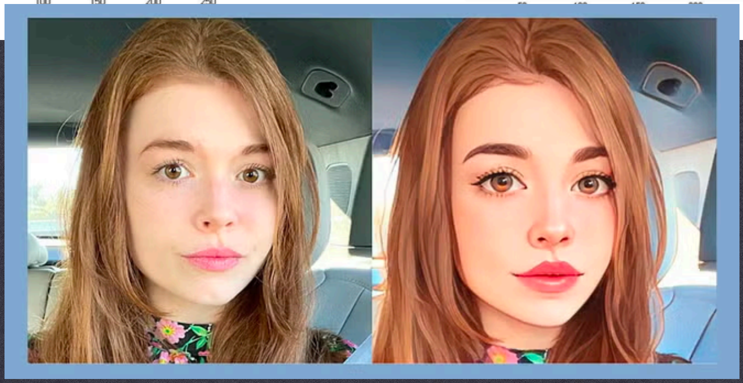
However, what features in the current are we trying to enhance, detect, or remove?

And what are the constraints we want to satisfy when filtering the current?

Are the ideal filters going to be problem-dependent?

Need to formulate the right question to optimize

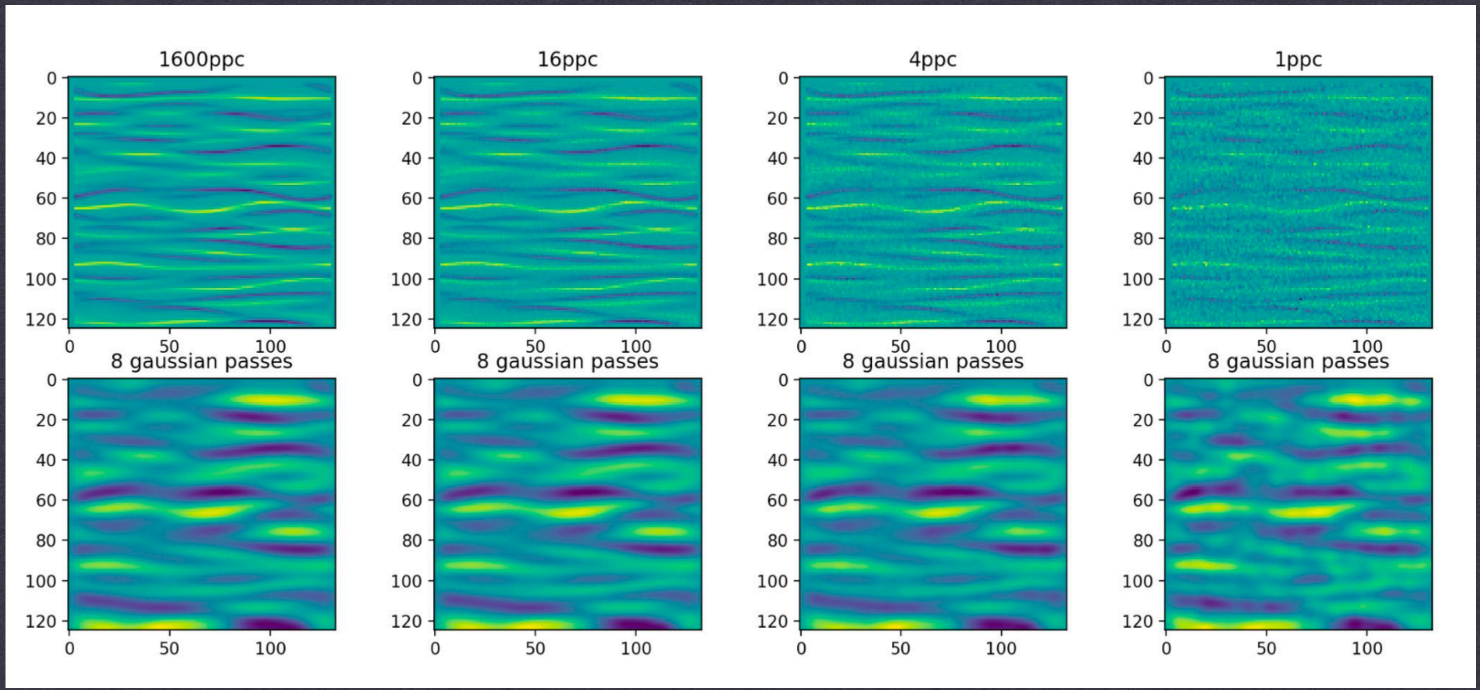




FILTER EXPERIMENTS

Some potential questions to formulate:

Can we make low particle-per-cell simulation look like high ppc simulation? Work with Jeff Shen (PU)



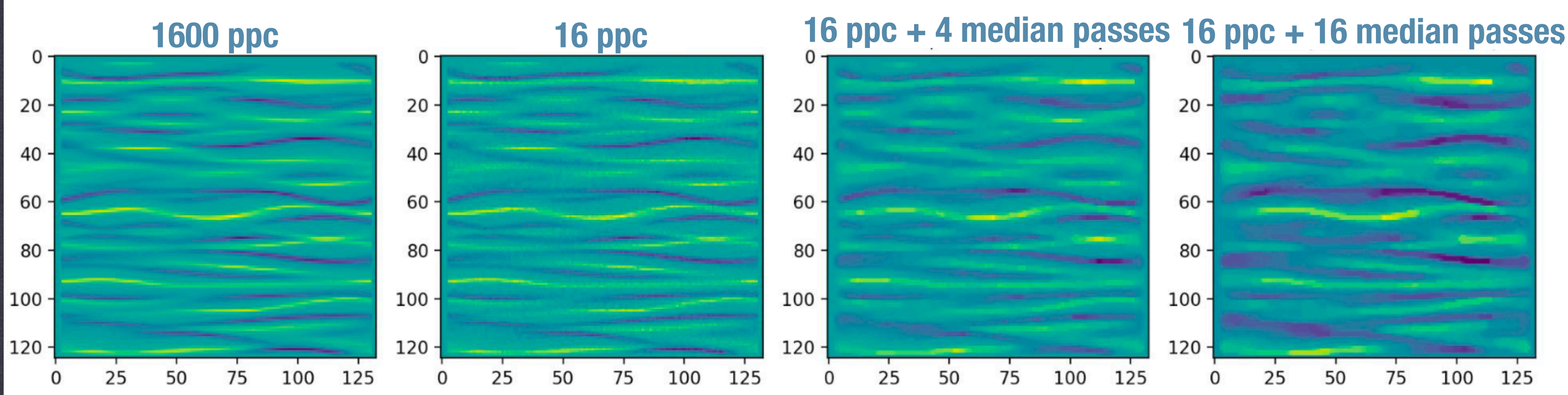
Simulations by Jeff Shen

FILTER EXPERIMENTS

Some potential questions to formulate:

Can we make low particle-per-cell simulation look like high ppc simulation?

Median filter: replace cell value with median value of surrounding cells. Nonlinear filter which keeps gradients

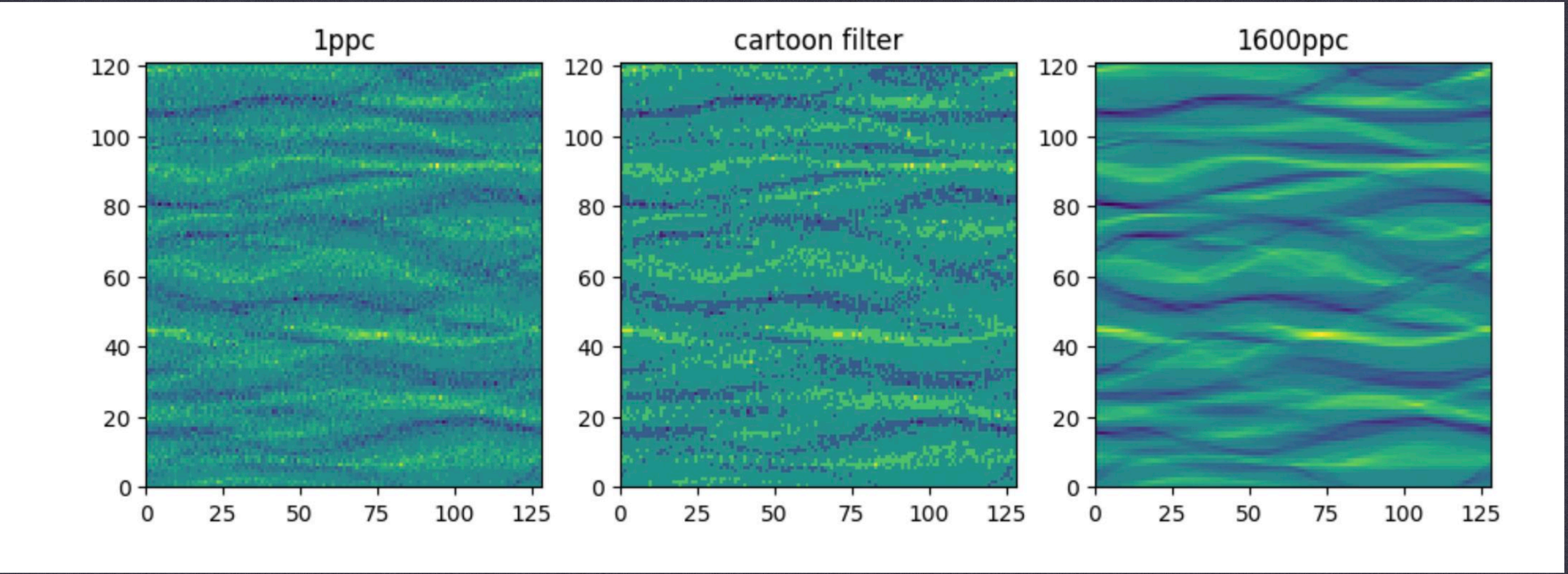


FILTER EXPERIMENTS

Some potential questions to formulate:

Can we make low particle-per-cell simulation look like high ppc simulation?

"Cartoon" filter:
Find edges, smooth, bin/discretize pixel values, smooth again, add edges

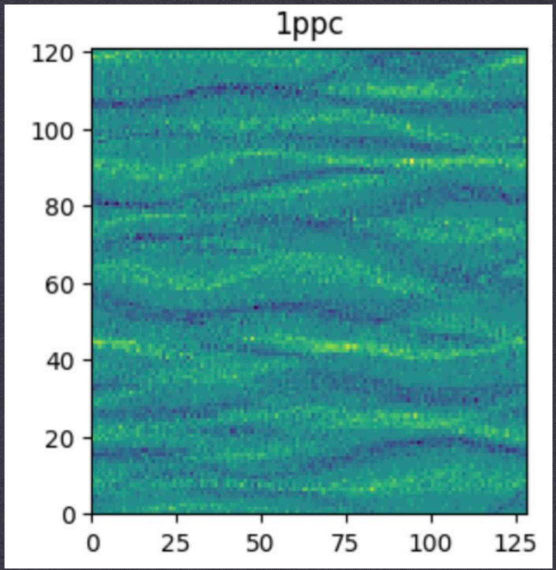


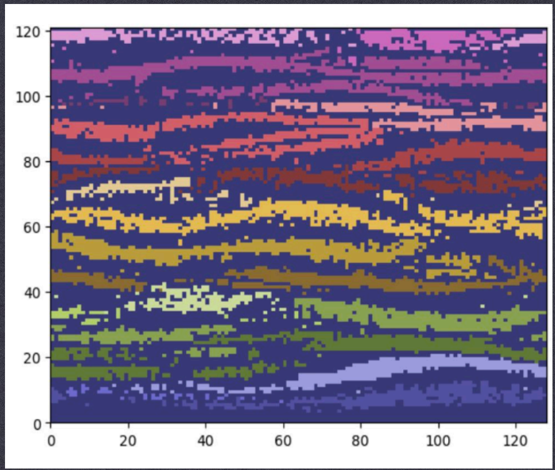
FILTER EXPERIMENTS

Some potential questions to formulate:

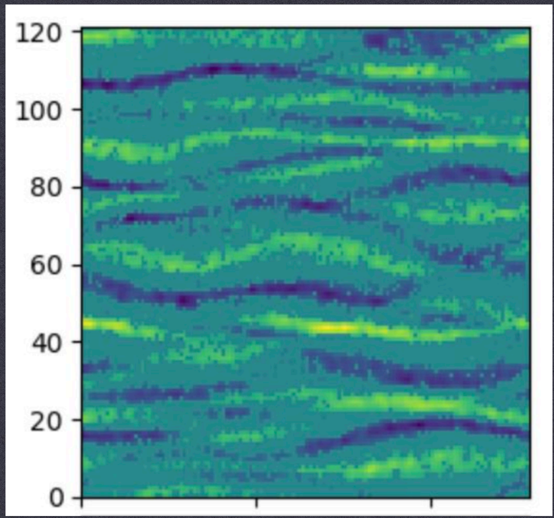
Can we make low particle-per-cell simulation look like high ppc simulation?

Segmentation filter:
Smooth, Discretize image, Find connected segments, Remove islands.
For each segment: mask out everything else, Gaussian smooth within segment, use large spread in largest segment (background), small spread in filaments
Stitch result together



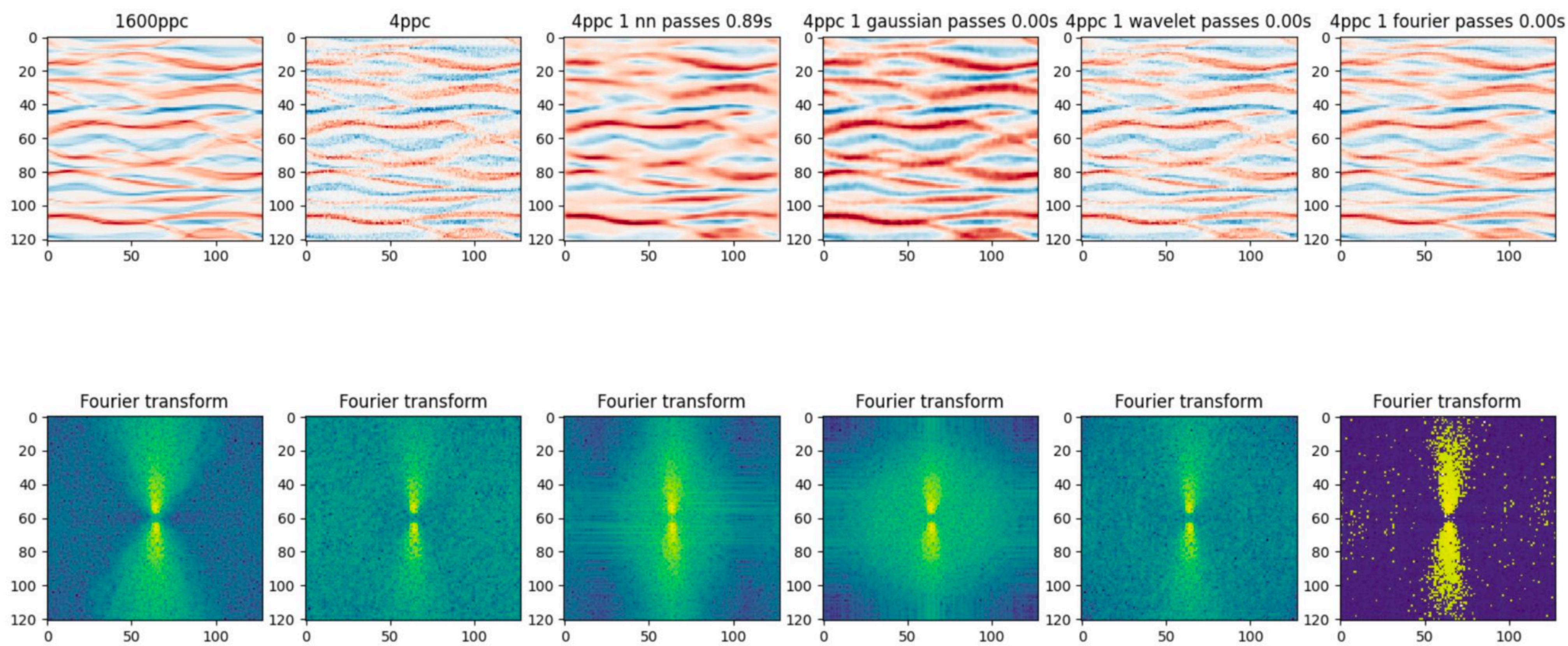


Segments



Result

FOURIER SPACE



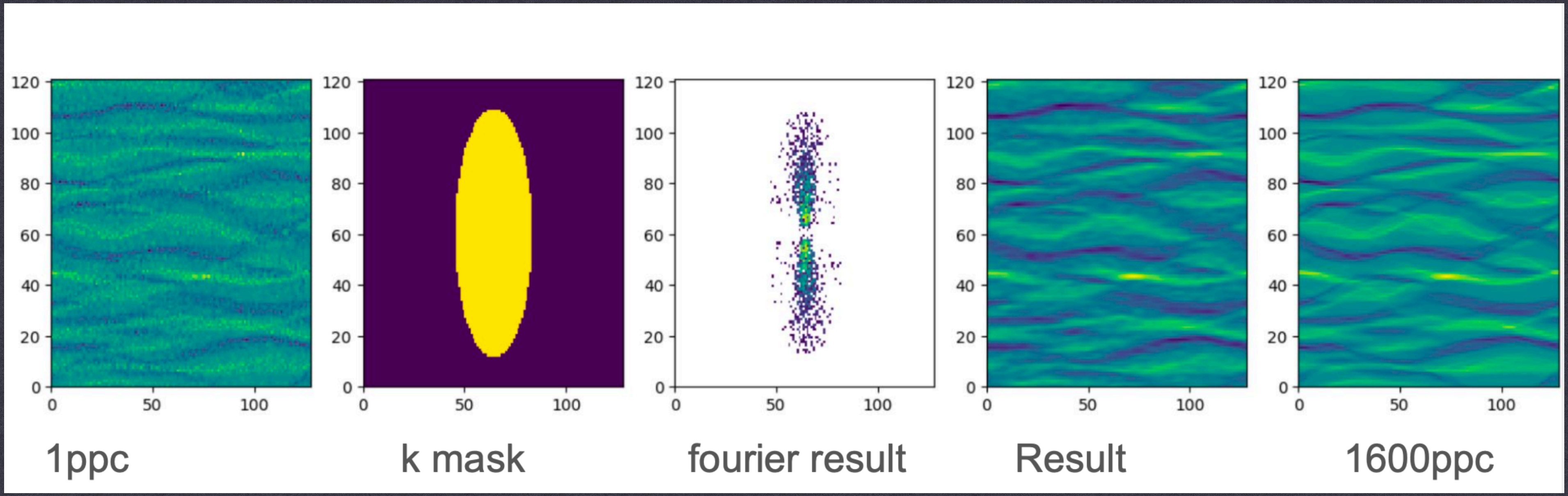
FILTER EXPERIMENTS

Some potential questions to formulate:

Can we make low particle-per-cell simulation look like high ppc simulation?

Fourier filters (also can play w/wavelets)

Cut by quintile and then mask in k



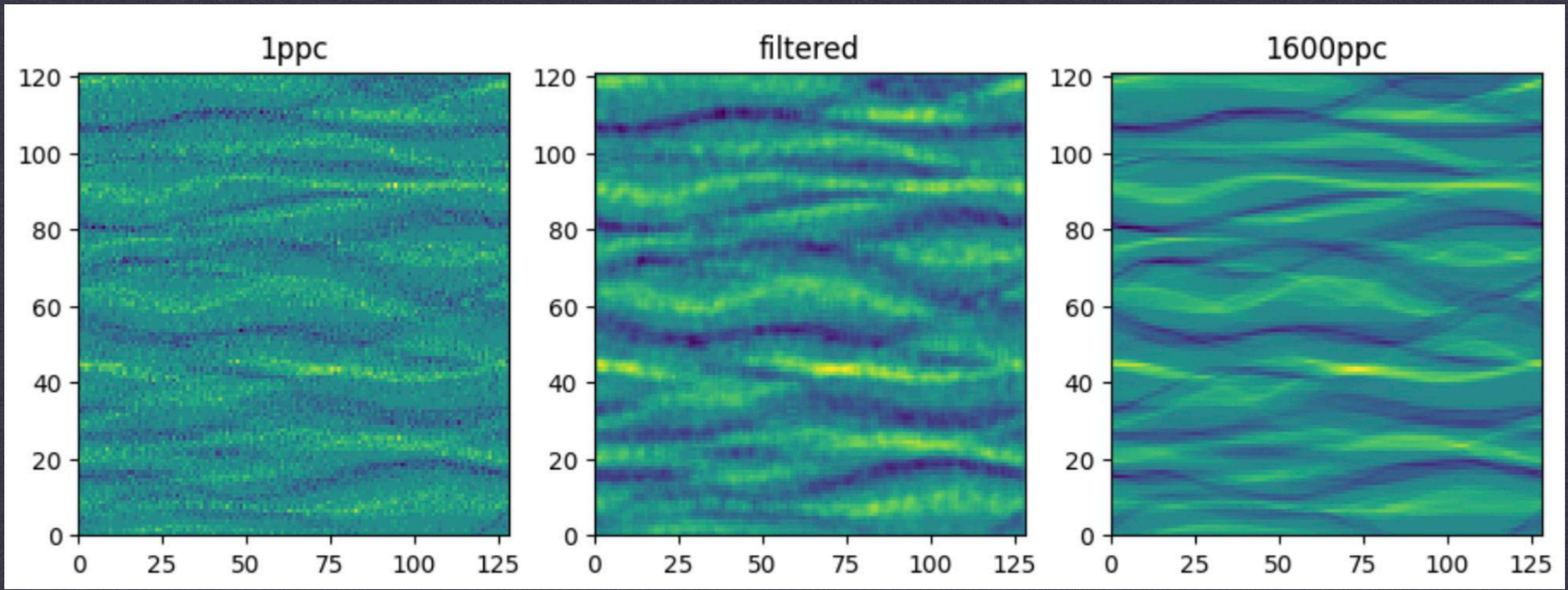
FILTER EXPERIMENTS

Some potential questions to formulate:

Can we make low particle-per-cell simulation look like high ppc simulation?

Neural network

CNN: one convolution layer.
Training on minimizing the L2 norm of difference between 1ppc and 1600ppc



FILTER EXPERIMENTS

Ok, this is lovely, but can we run with it?

Physics strikes back…

Nonlinear filters can no longer be interpreted as particle shapes. Particles no longer independent.

Nonlinear filters do not conserve charge — have to correct E_longitudinal through Poisson solve. Annoying, but maybe worth the trouble.

Nonlinear filters probably do not conserve energy (need to check) and momentum (?). Can one use momentum/energy conservation as penalty in finding new filters?

Are we limited to linear/symmetric filters? If so, can they vary in space to enhance edges?

What test problems can be used to check filtering? What is reality, really?

Momentum conservation

$$\frac{dP}{dt} = \sum_i F_i$$

$$\frac{dP}{dt} = \sum_i q_i \Delta x \sum_j E_j S(X_j - x_i)$$

$$\frac{dP}{dt} = \Delta x \sum_j E_j \sum_i q_i S(X_j - x_i)$$

$$\frac{dP}{dt} = \Delta x \sum_j \rho_j E_j$$

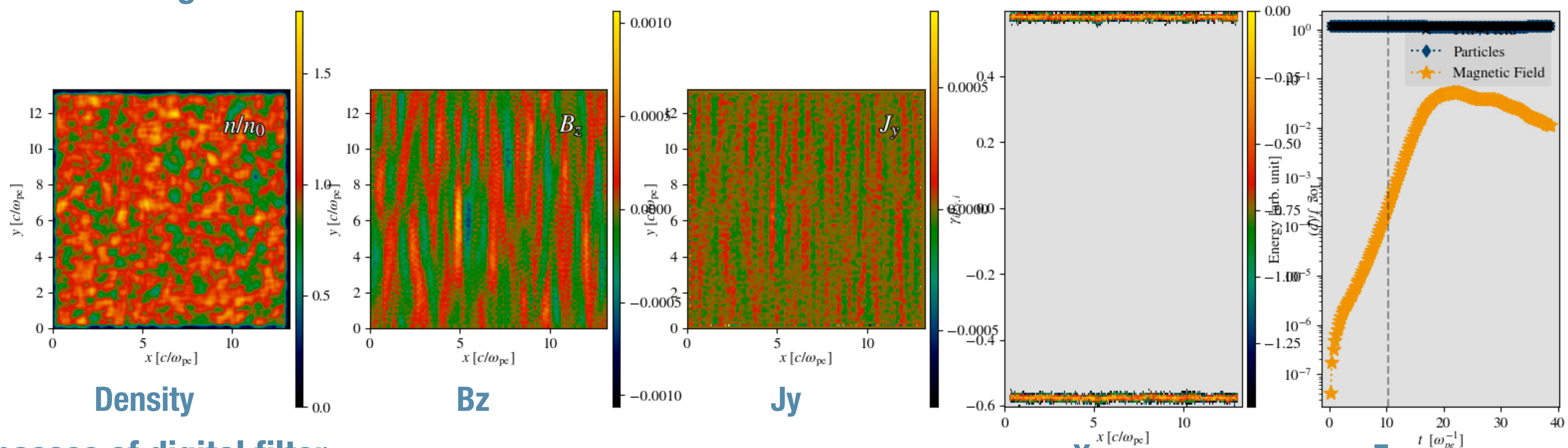
For periodic system

$$\Delta x \sum_j \rho_j E_j = 0$$

Interpolation/Deposition needs symmetric kernel

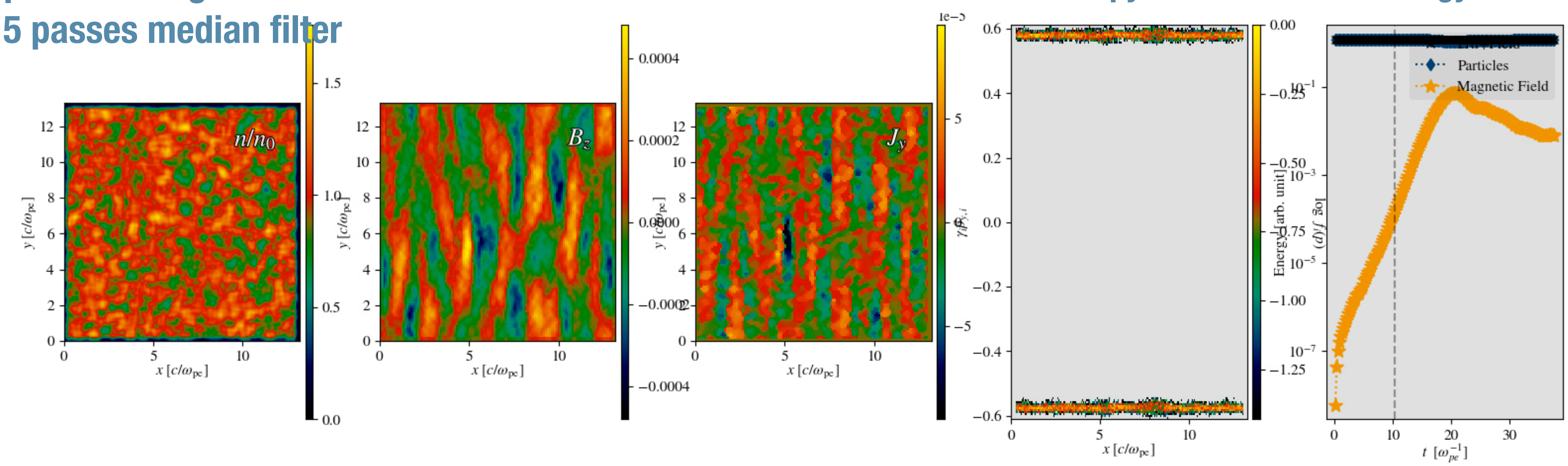
FILTER EXPERIMENTS

4 passes of digital filter



Density Bz Jy X-py Energy

4 passes of digital filter + 5 passes median filter



CONCLUSIONS

Gaussian current filtering is effective for fighting noise, but seems rather primitive and excessive.

Can one construct filters that restore or enhance information contained in the current sampled with few particles per cell?

What constraints should current filtering satisfy?

What information about the phase space is important in the current: strength, gradients, noise?

Ideas?