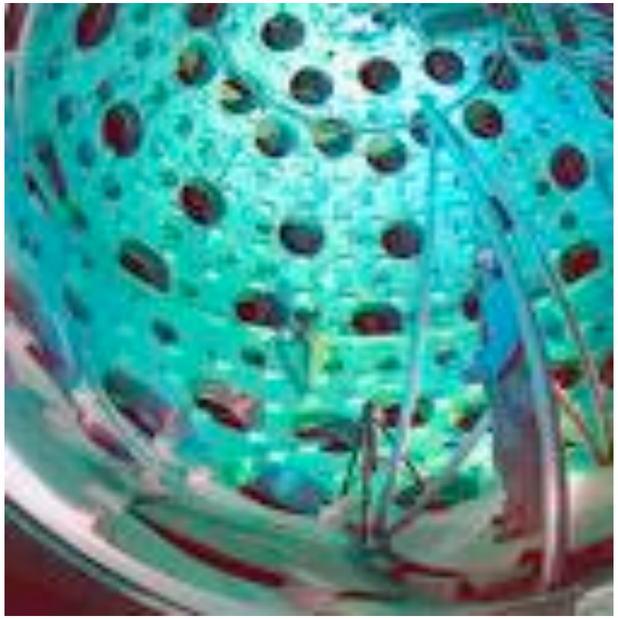
INRIA Sophia-Antipolis - July 2008

# Some Experiences and Open Questions in the Development of a DG-FEM Based Particle-in-Cell Method

Jan S Hesthaven
Brown University
Jan.Hesthaven@Brown.edu



# Thanks!

- People making this possible
  - Dr Stephane Lanteri
  - The kind people of INRIA Sophia-Antipolis - you!

- Collaborators
  - Gustaaf Jacobs (SDSU - formerly Brown)
  - Akil Narayan (Brown)
  - Andreas Kloeckner (Brown)
  - Tim Warburton (Rice University)
  - PIC group at Kirkland AFB
  - Giovanni Lapenta, LANL

- Funding agencies - AFOSR, ANL

# Kinetic Plasma Physics

In high-speed plasma problems dominated by kinetic effects, one needs to solve for f(x,p,t)

*Vlasov/Boltzmann equation*

$$\partial_t f + v \cdot \partial_x f + q(E + v \times B) \cdot \partial_p f = \langle \text{Sources} \rangle - \langle \text{Sinks} \rangle.$$

*Maxwell's equations*

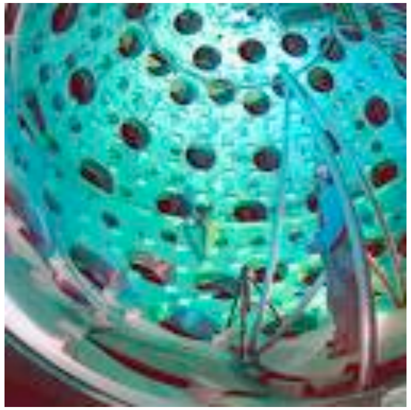
$$\begin{aligned} \partial_t E - \frac{1}{\varepsilon} \nabla \times H &= -\frac{j}{\varepsilon}, \\ \partial_t H + \frac{1}{\mu} \nabla \times E &= 0, \\ \nabla \cdot H = 0, \qquad \nabla \cdot E &= \frac{\rho}{\varepsilon}. \end{aligned}$$

Coupled through $\rho := \int f \, dv, \qquad j := \int v f \, dv.$

# Kinetic Plasma Physics

Important applications

- High-power/High-frequency microwave generation
- Particle accelerators
- Laser-matter interaction
- Fusion applications, e.g., plasma edge
- etc



# Challenges in the Problem

- Full coupling between plasma and fields
- Large scale separation in both time and space
- Electrically large problems
- Time-dependent and highly dynamic
- Often critical phenomena where particles interact with geometric features
- Emphasis is on high-speed problems where full EM is required

Challenge: To solve a 6+1 dimensional problem in complex geometries over long times

# Particle-in-Cell (PIC) Methods

This is an attempt to solve the Vlasov/Boltzmann equation by sampling with P particles

$$f(x,p,t) = \sum_{n=1}^{P} q_n S(x - x_n(t))\delta(p - p_n(t)),$$

$$\rho(x,t) = \sum_{n=1}^{P} q_n S(x - x_n(t)), \quad j(x,t) = \sum_{n=1}^{P} v_n q_n S(x - x_n(t))$$

Ideally we have

$$S(x) = \delta(x) \longleftarrow$$ a point particle

However, this is not practical, nor reasonable - so S(x) is a **shape-function**

# Particle-in-Cell Methods

## Maxwell's equations

$$\varepsilon\partial_t E - \nabla \times H = -j, \quad \mu\partial_t H + \nabla \times E = 0,$$

$$\nabla \cdot (\varepsilon E) = \rho, \quad \nabla \cdot (\mu H) = 0,$$

## Particle/Phase dynamics

$$\frac{dx_n}{dt} = v_n(t) \qquad \frac{dmv_n}{dt} = q_n(E + v_n \times H) \quad m = \frac{1}{\sqrt{1-(v_n/c)^2}}$$

## Particles-to-fields

$$\rho(x,t) = \sum_{n=1}^{P} q_n S(x - x_n(t)), \quad j(x,t) = \sum_{n=1}^{P} v_n q_n S(x - x_n(t))$$

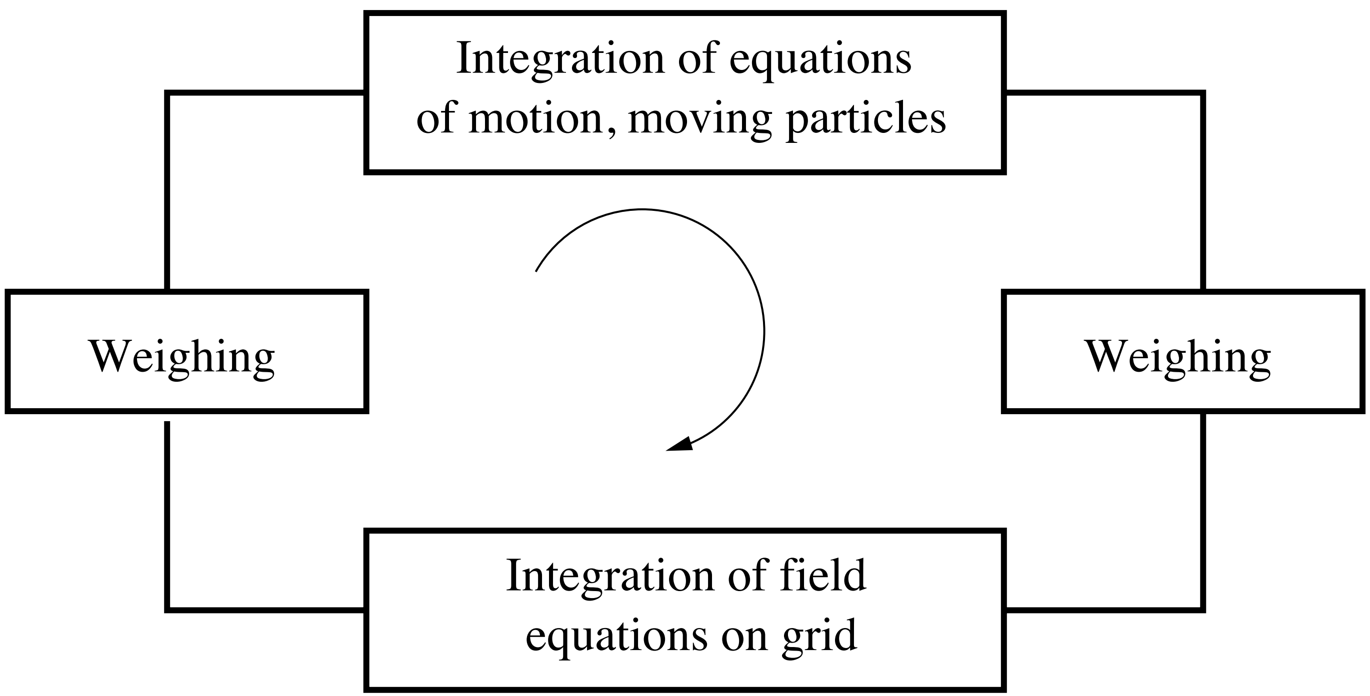
## Fields-to-particles

$$E(x_n), H(x_n)$$

# Classic Particle-in-Cell Methods

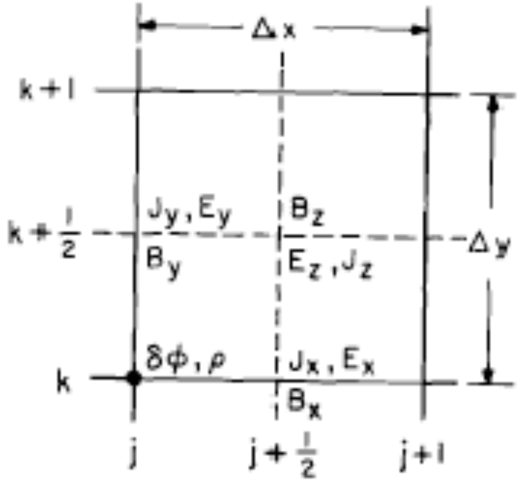
Four stages: Assume $E^n, H^n, j^n, \rho^n$ are given

- Advanced Maxwell's equations
- Interpolate fields to particles
- Advance particles
- Deposit charges and currents to fields

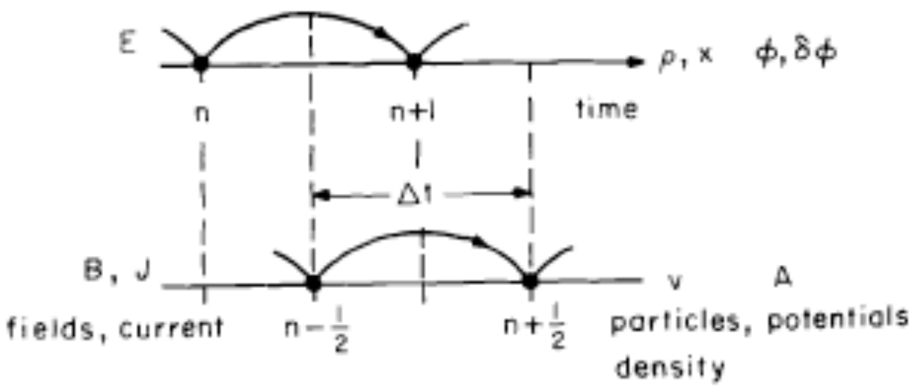


# Classic Particle-in-Cell Methods

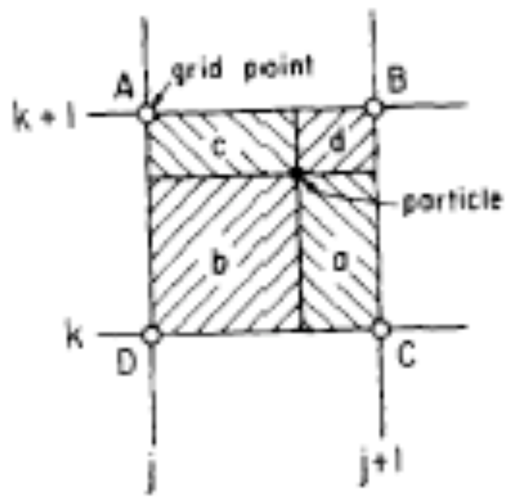
Staggered/Yee grid in space




Leap-frog in time




Piecewise constant charges/shape functions




Scheme due to Villasenor/Buneman (1992)

# Classic Particle-in-Cell Methods

The central advantages of this scheme are

- Exact energy and charge conservation
- Simple
- Very fast
- Very widely used and tested/validated on nontrivial problems with > 1 billion particles

However, there is also a number of well recognized problems and limitations

# Classic Particle-in-Cell Methods

## *Problems*

- The exactness of charge and energy is tightly coupled to the Cartesian grid - i.e., no support for local grid-refinement
- No geometric flexibility and staircasing
- (Very) poor accuracy close to boundaries
- 2nd order accuracy in fields (dispersion errors)
- 1st order accuracy in currents/charges
- Numerical Cherenkov radiation
- Poor performance on large scale computers
- Not well suited for multi-physics modeling

# Classic Particle-in-Cell Methods

This translates into problems and limitations like

- Electrically large problems
- Problems requiring long time integration
- Problems where the interaction with geometries are important, e.g., secondary emission
- High-density problems
- Problems suggesting a hybrid fluid/particle model
- Problems requiring a peta-scale platform

These are the characteristics of many problems

We need to look for an alternative

# A new Particle-in-Cell Methods ?

.. but what should we look for ?

- Geometric flexibility and non-uniform grids
- High/variable order accuracy in fields
- Improved accuracy in currents/charges
- Robustness and flexibility for hybrid problems
- High efficiency

- .. while doing the physics right !

This is harder than it looks

... and we are 20+ years behind

... so teamwork is essential!

# Brief overview of what remains

- DG-FEM for the fields
  High-order, general grids and all of that
- Particles
  Shapes, moves, identification etc
- Charge conservation
- Boundary-particle interactions
- Tests, Tests
  Sanity tests
  More complex tests
- Closer to the application
- Open problems and outlook
- Something extra (time permitting)

# Solving for the fields

Consider Maxwell's equations

$$\varepsilon\partial_t E - \nabla \times H = -j, \qquad \mu\partial_t H + \nabla \times E = 0,$$

Write it on conservation form as

$$\frac{\partial q}{\partial t} + \nabla \cdot F = -J$$

$$q = \begin{bmatrix} E \\ H \end{bmatrix} \quad F = \begin{bmatrix} -\hat{e} \times H \\ \hat{e} \times E \end{bmatrix} \quad J = \begin{bmatrix} j \\ 0 \end{bmatrix}$$

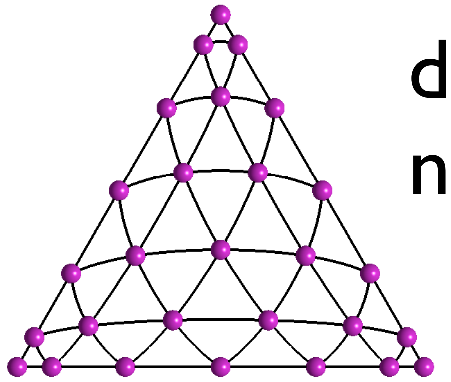
Represent the computational domain as

$$\Omega = \sum_k D^k$$

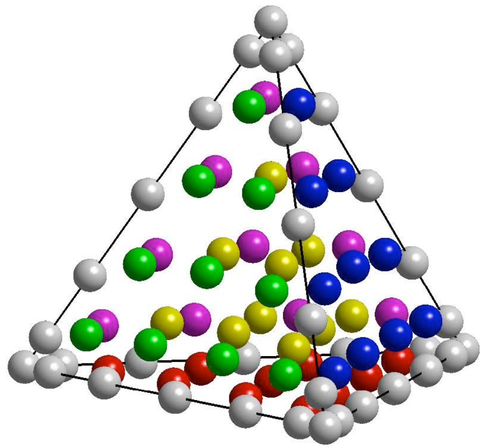
# Solving for the fields

On each element we assume

$$\boldsymbol{q}_N(\boldsymbol{x},t) = \sum_{j=1}^{N} \boldsymbol{q}(\boldsymbol{x}_j,t)L_j(\boldsymbol{x}) = \sum_{j=1}^{N} \hat{\boldsymbol{q}}_j(t)L_j(\boldsymbol{x}), \qquad N = \begin{pmatrix} n+d \\ n \end{pmatrix} \simeq \frac{n^d}{d!}$$

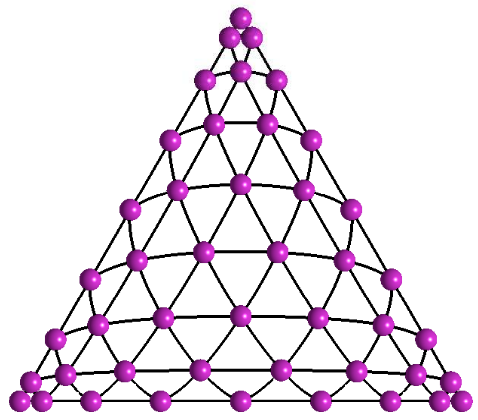


d=2
n=6

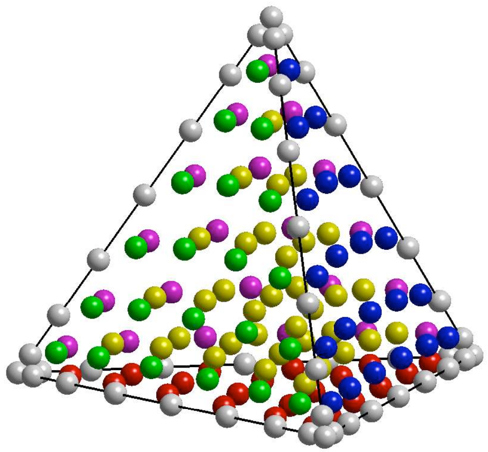


d=3
n=6

d=2
n=8



d=3
n=8



# Solving for the fields

On each element we then require

$$\int_D \left( \frac{\partial \boldsymbol{q}_N}{\partial t} + \nabla \cdot \boldsymbol{F}_N - \boldsymbol{J}_N \right) L_i(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \oint_{\partial D} L_i(\boldsymbol{x}) \hat{\boldsymbol{n}} \cdot [\boldsymbol{F}_N - \boldsymbol{F}^*] \, \mathrm{d}\boldsymbol{x}.$$

With the numerical flux given as

$$\hat{\boldsymbol{n}} \cdot [\boldsymbol{F} - \boldsymbol{F}^*] = \begin{cases} \boldsymbol{n} \times (\gamma \boldsymbol{n} \times [\boldsymbol{E}] - [\boldsymbol{B}]), \\ \boldsymbol{n} \times (\gamma \boldsymbol{n} \times [\boldsymbol{B}] + [\boldsymbol{E}]), \end{cases} \qquad [Q] = Q^- - Q^+$$

Define the local operators

$$\hat{M}_{ij} = \int_D L_i L_j \, \mathrm{d}\boldsymbol{x}, \quad \hat{S}_{ij} = \int_D \nabla L_j L_i \, \mathrm{d}\boldsymbol{x}, \quad \hat{F}_{ij} = \oint_{\partial D} L_i L_j \, \mathrm{d}\boldsymbol{x},$$

To obtain the local matrix based scheme

$$\hat{M} \frac{\mathrm{d}\hat{\boldsymbol{q}}}{\mathrm{d}t} + \hat{S} \cdot \hat{\boldsymbol{F}} - \hat{M} \hat{\boldsymbol{J}} = \hat{F} \hat{\boldsymbol{n}} \cdot [\hat{\boldsymbol{F}} - \hat{\boldsymbol{F}}^*],$$

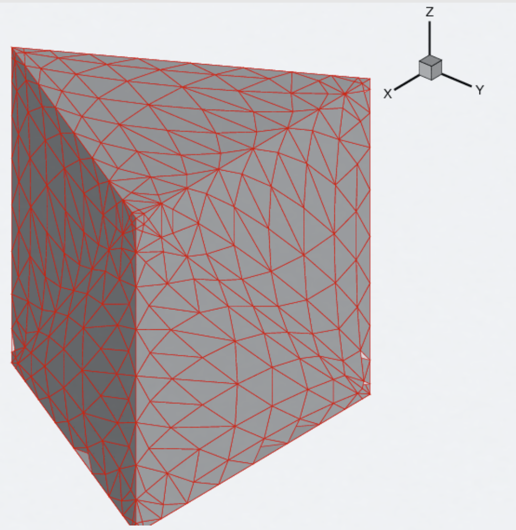
# Solving for the fields

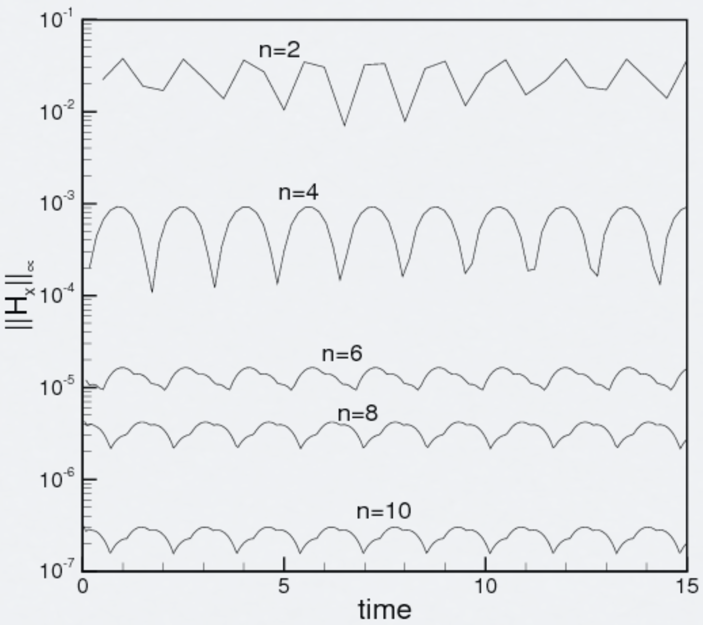
In time we use a 4th order Runge-Kutta method

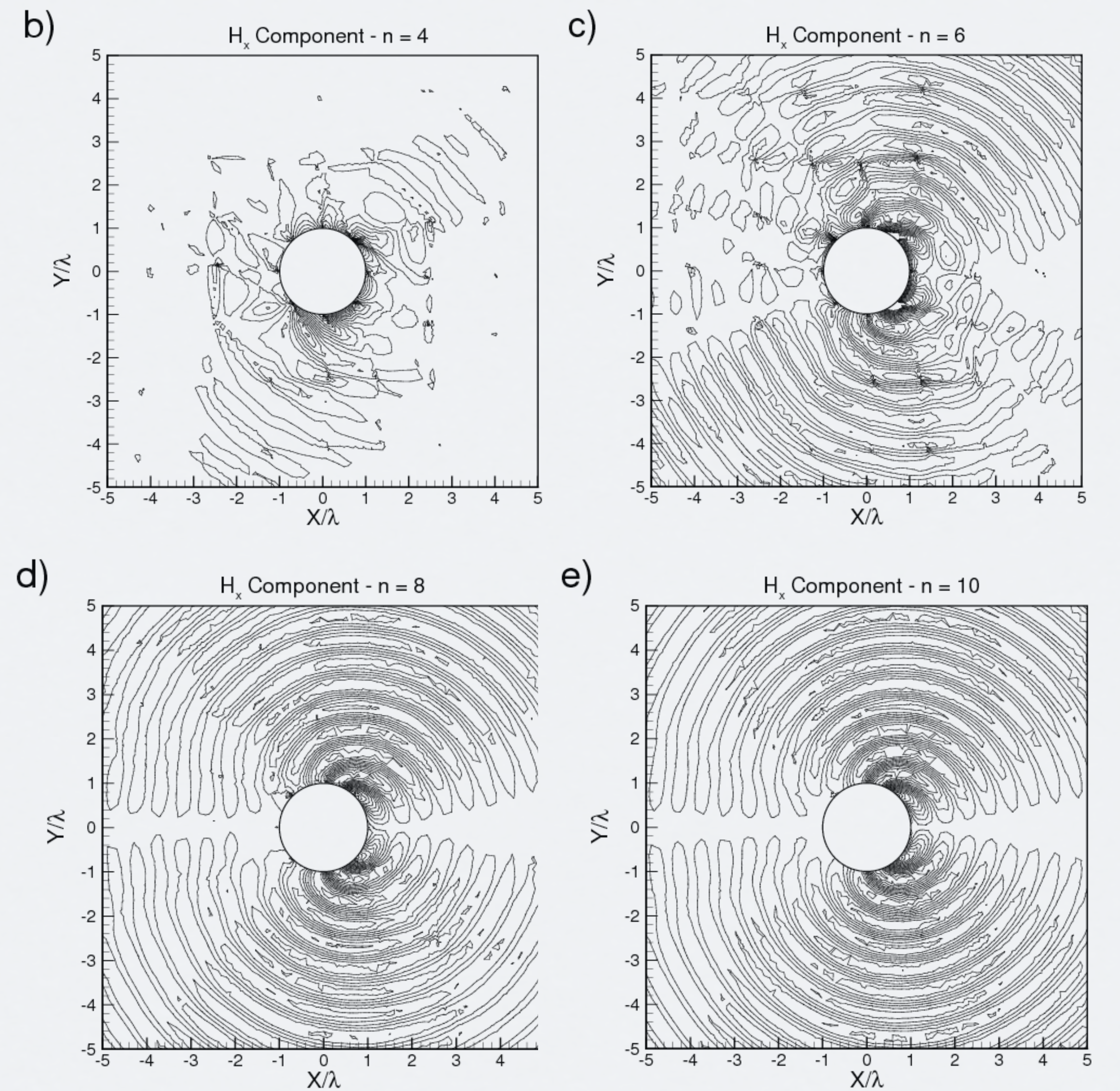
$$\left.\begin{aligned} \boldsymbol{w}_i &= \alpha_i \boldsymbol{w}_{i-1} + \Delta t \boldsymbol{F}(t_{i-1}, \boldsymbol{q}^{(i-1)}) \\ \boldsymbol{q}^{(i)} &= \boldsymbol{q}^{(i-1)} + \beta_i \boldsymbol{w}_i \end{aligned}\right\}, \quad i = 1, 2, \ldots, s,$$

- Scheme is fully explicit
- Well understood for both electrostatics and electromagnetics
- Supports general grids, variable order, complex geometries
- High parallel efficiency
- Used by several groups for EM across the world

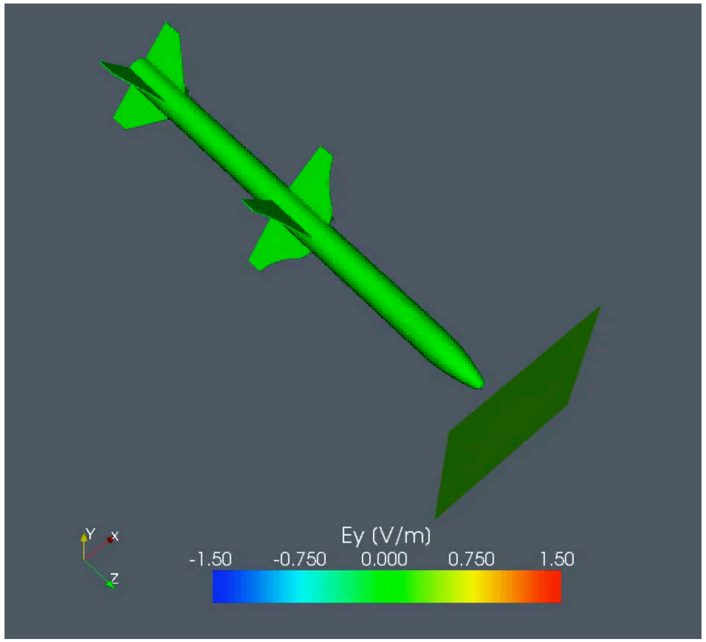
# Solving the field equations

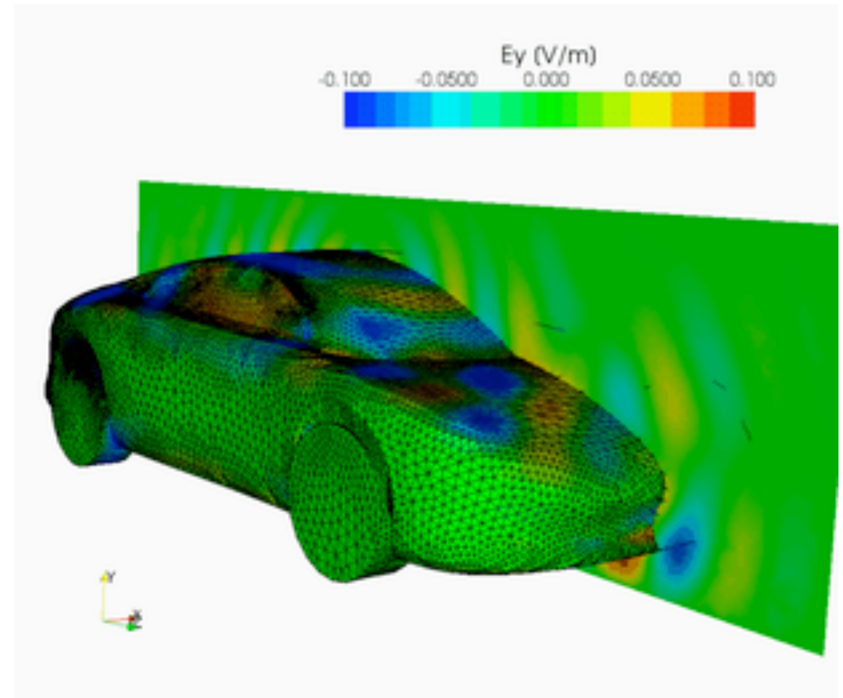


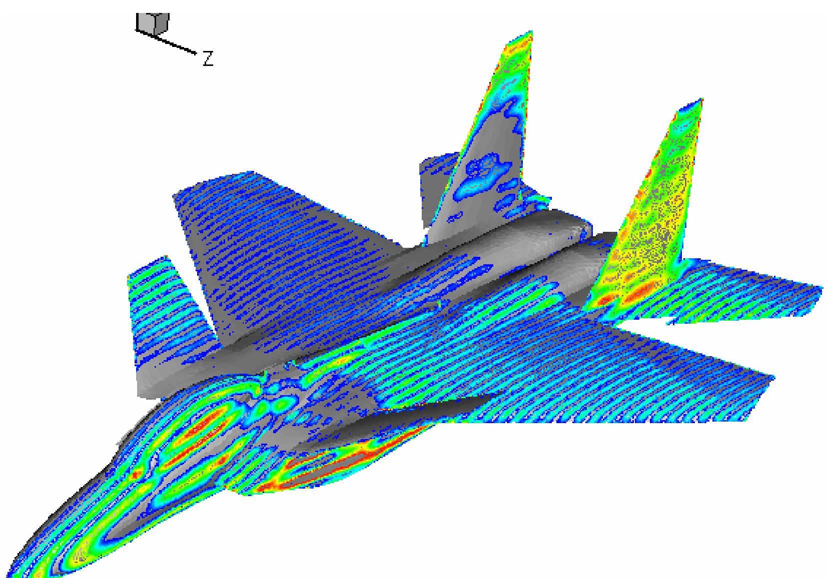



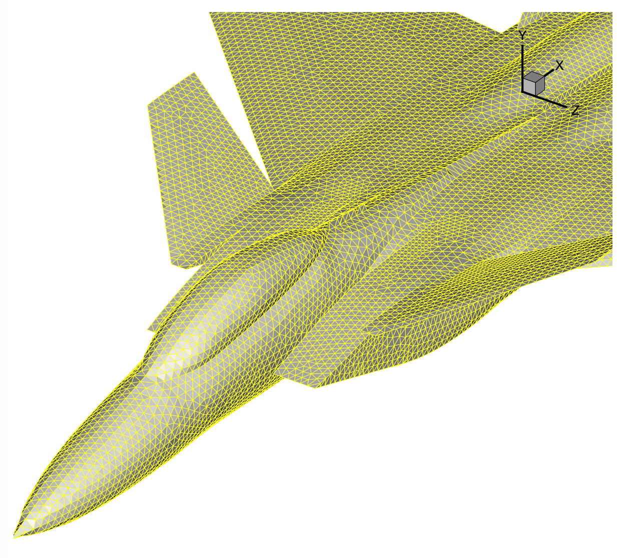



# Maxwell's equations



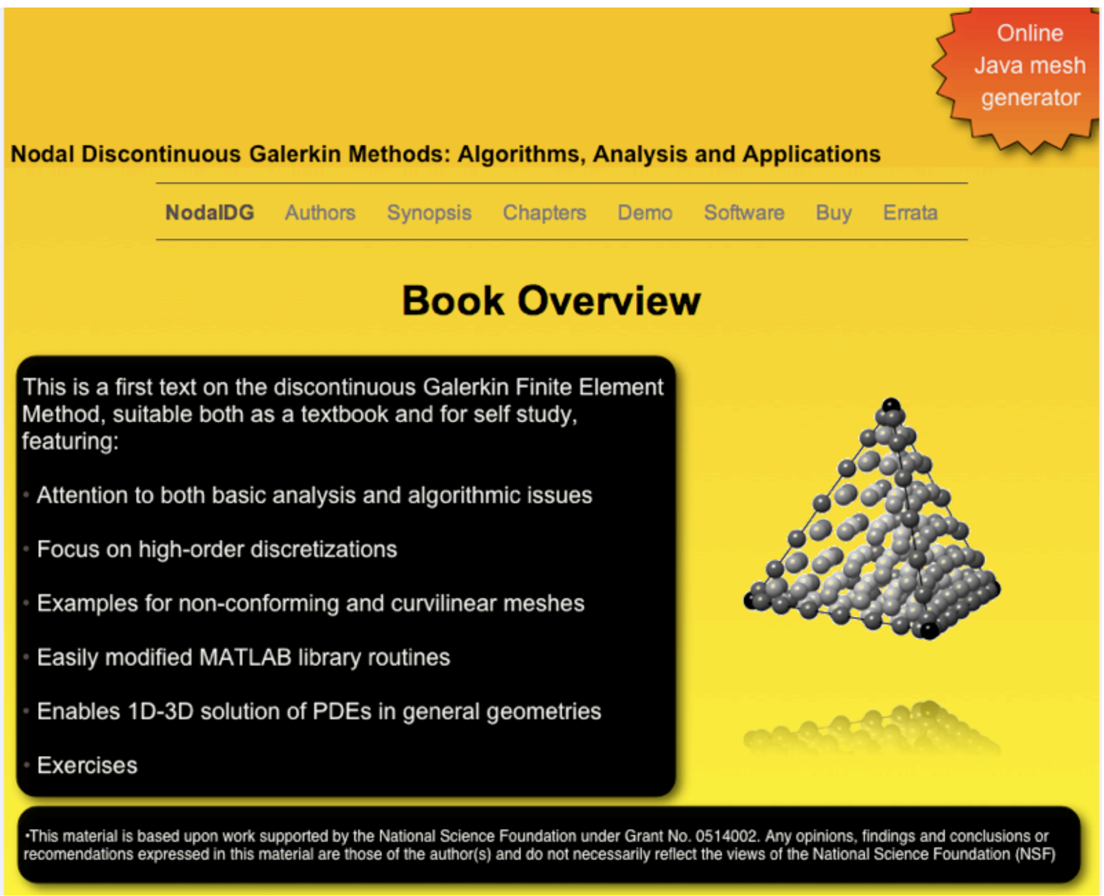


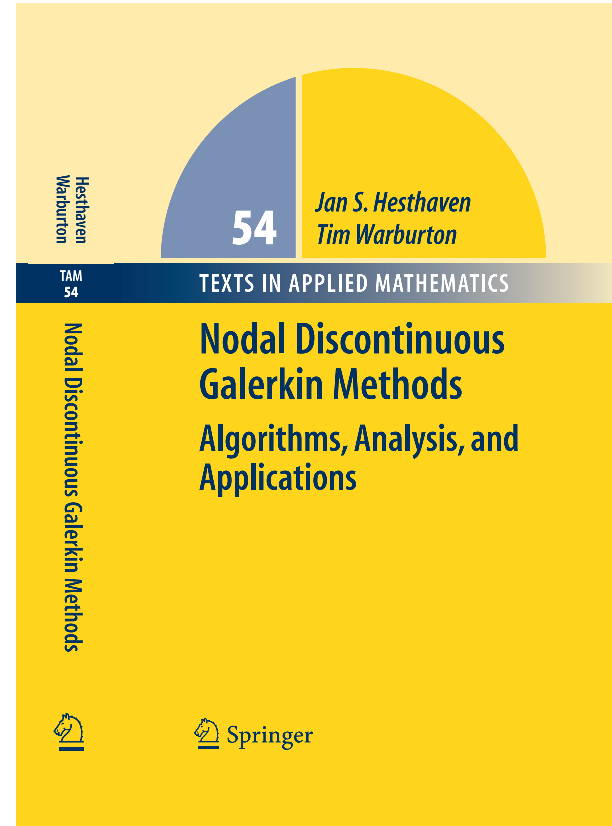
Animations by Nico Godel Hamburg using NuDG

# A bit of promotion ..

Naturally, the devil is in the details
.... and (some of) the details you can find in

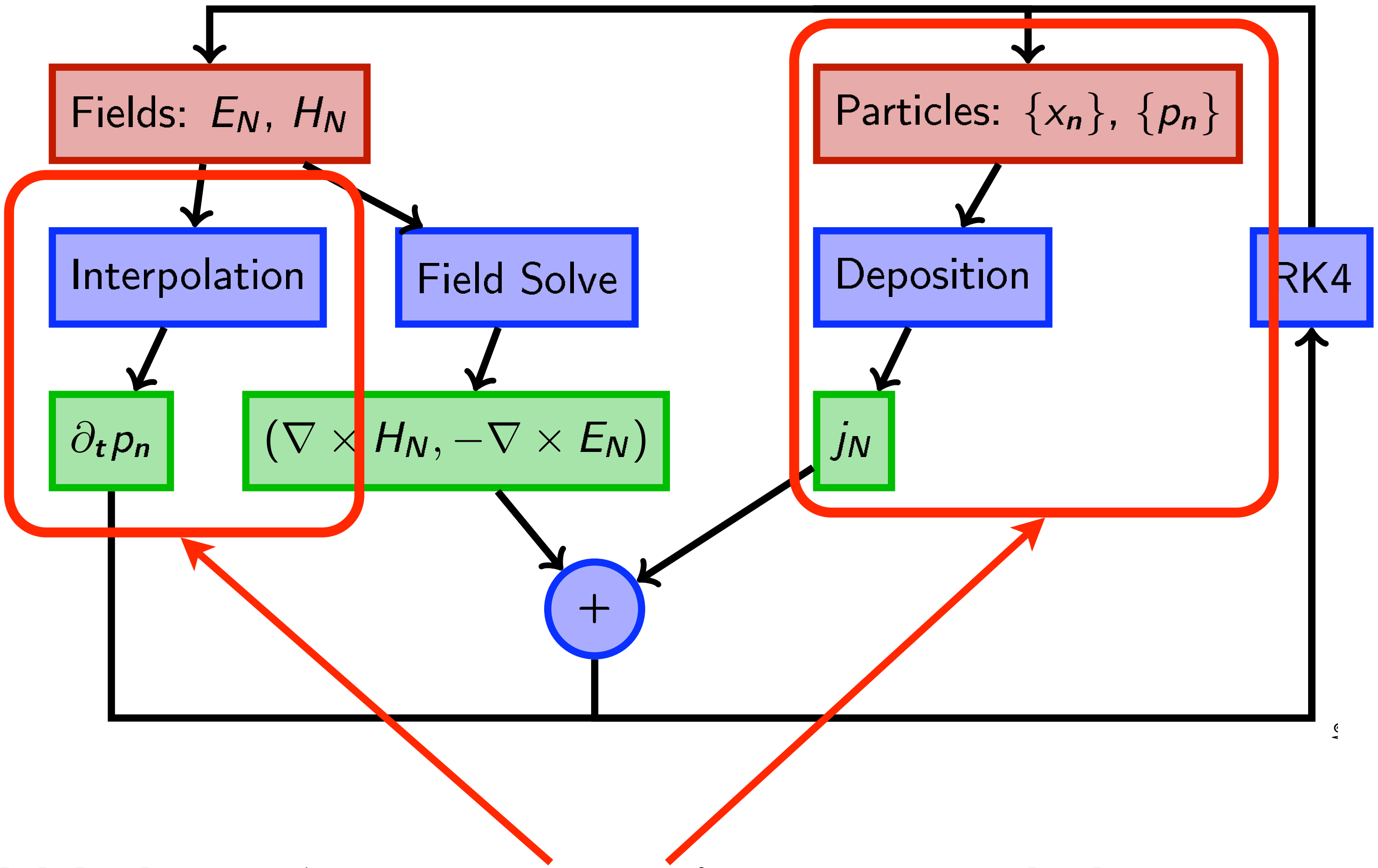
*http://www.nudg.org*




Matlab/C++ software available

# Scheme so far




Note: No staggering in time.

Everything is collocated.

More on that later

We have 'conviniently' neglected these two components - both key to the PIC model

# The particles

This is a much harder problem!

- Particle shapes
- Particle pushers and current/charge deposition
- Particle interactions with geometries
- Numerical Cherenkov radiation
- Charge conservation
- ... and many other issues

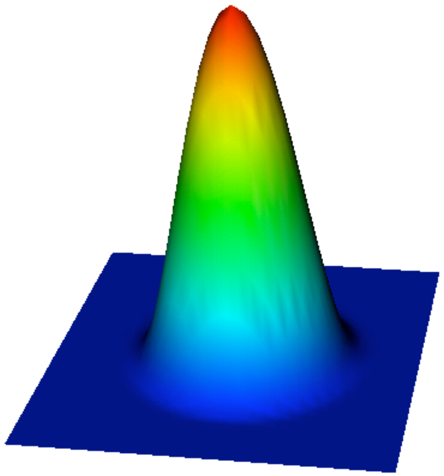
I will share many results with you -- not clear we have a ‘steady state’ strategy yet

# Particles et al

We are using a grid independent shape function:

$$S_{\text{poll}} = \frac{\alpha + 1}{\pi R^2}\left[1 - \left(\frac{r}{R}\right)^2\right]^\alpha, \quad r = 0, \ldots, R,$$

- Truncated polynomial
- $S(r) \to \delta$ as $\alpha \to \infty$
- $S \in C^{\alpha-1}$
- S is compactly supported



High values of $\alpha$ is physically appealing but that requires many particles

**Note**: Problems with highly non-uniform grids

# Particle pushing

Requires two steps

- Computation of forces on particles
- Advance

$$\frac{dx_n}{dt} = v_n(t) \qquad \frac{dmv_n}{dt} = q_n(E + v_n \times H) \qquad m = \frac{1}{\sqrt{1-(v_n/c)^2}}$$

For the latter we use RK as for the fields

For the former, *a paradox* arises

- For deposition, particles are clouds
- For pushing they are points

# Particle pushing

The force computation at a point is straightforward

$$V_{ij} = \psi_j(\eta_i) \qquad m_i = \psi_i(x_n)$$

then

$$E(x_n) = (V^{-T} m)^T E \qquad L(x) = V^{-T}\psi(x)$$

This vectorizes very well

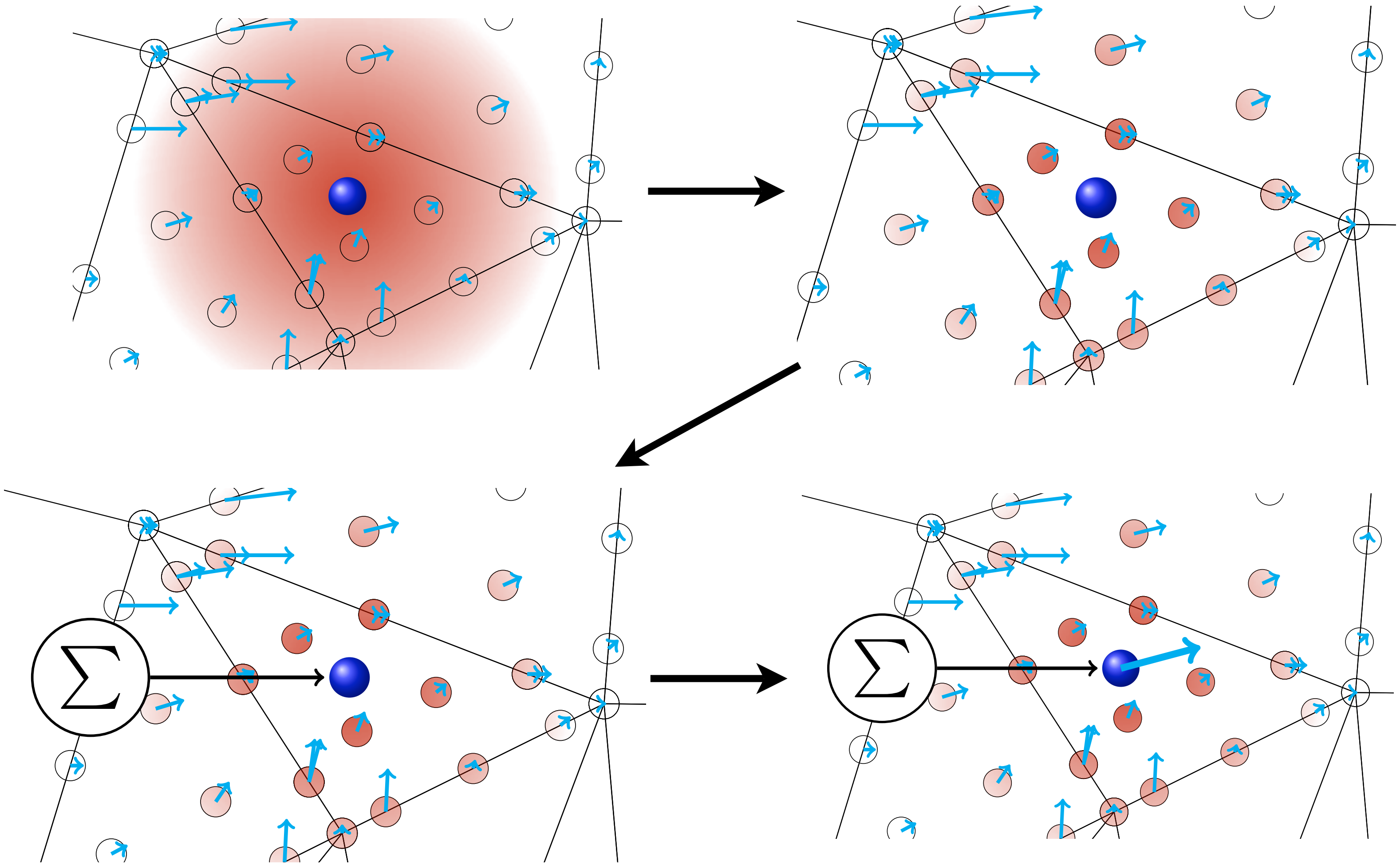
Here we have the orthonormal basis $\psi(x)$

However, this is too expensive to evaluate so for force evaluation we use a simple monomial basis

$$\psi(x,y,z) = x^i y^j z^k, \;\; 0 \le i+j+k \le n$$

# Particle pushing

One way to address the apparent paradox is to push by the cloud averaged forces



# Particle pushing

## Averaged pushing

- Physically more intuitive
- Computationally more expensive
- One can use the spread of point forces over a particle as a resolution measure.

.. but the test is -- does it work better

We will see shortly

# Current/Charge deposition

We shall discuss two different approaches

- Deposition by shape function
- Deposition by Cartesian overlaid grid

This stage is critical both for accuracy but also for speed.

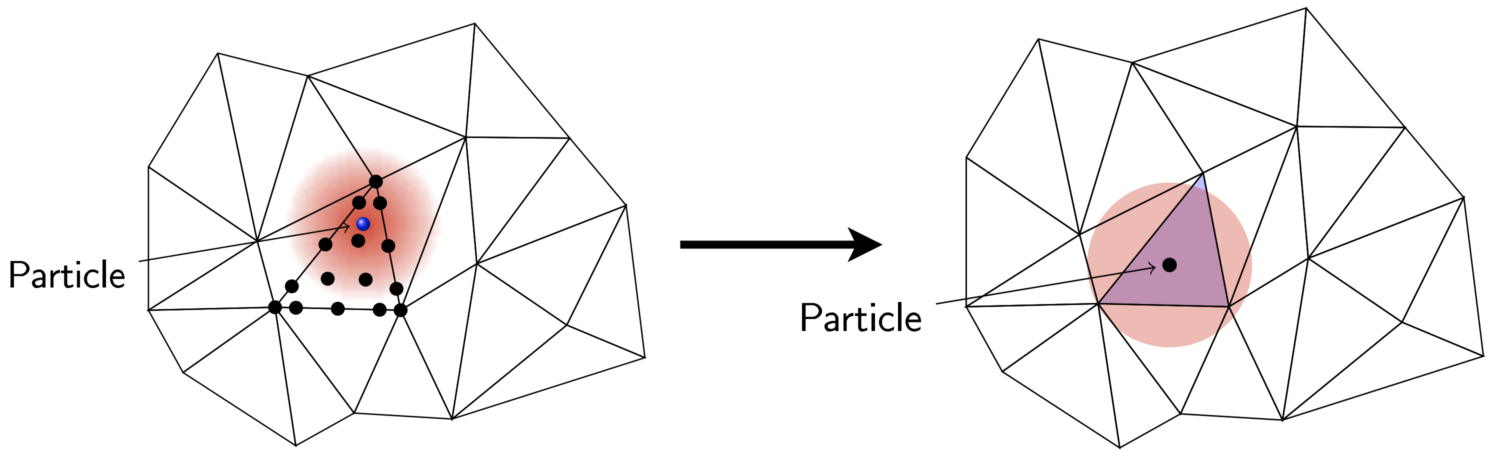
There is a clear tradeoff between simple/fast particles and complex/slow particles

.. and the particles play a dual role !

# Deposition by shape function

The idea is quite simple

- Identify which elements are reached by the cloud
- Deposit according to the cloud

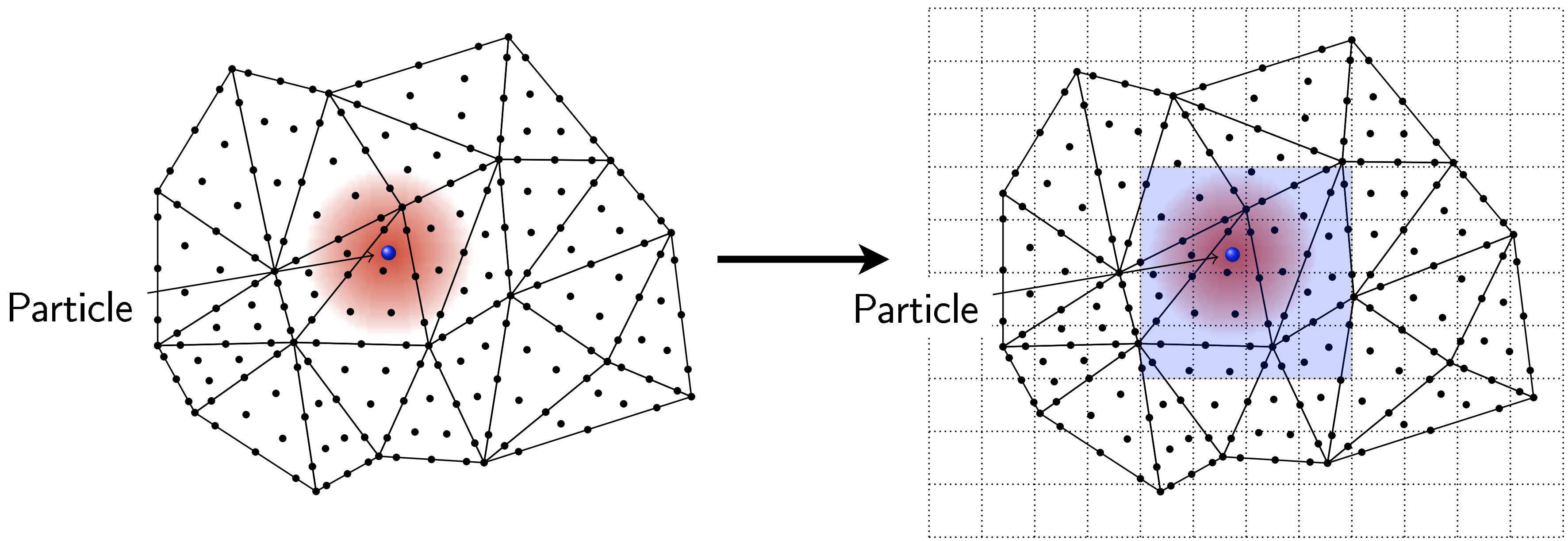



- Face based search (or vertex based)
- Velocity and normals used to identify new target element

# Deposition by shape function

*Computational bottleneck:* Deposition is expensive due to uneven nodal distribution.

*Idea:* Use a Cartesian grid for search



# Deposition by shape function

Features

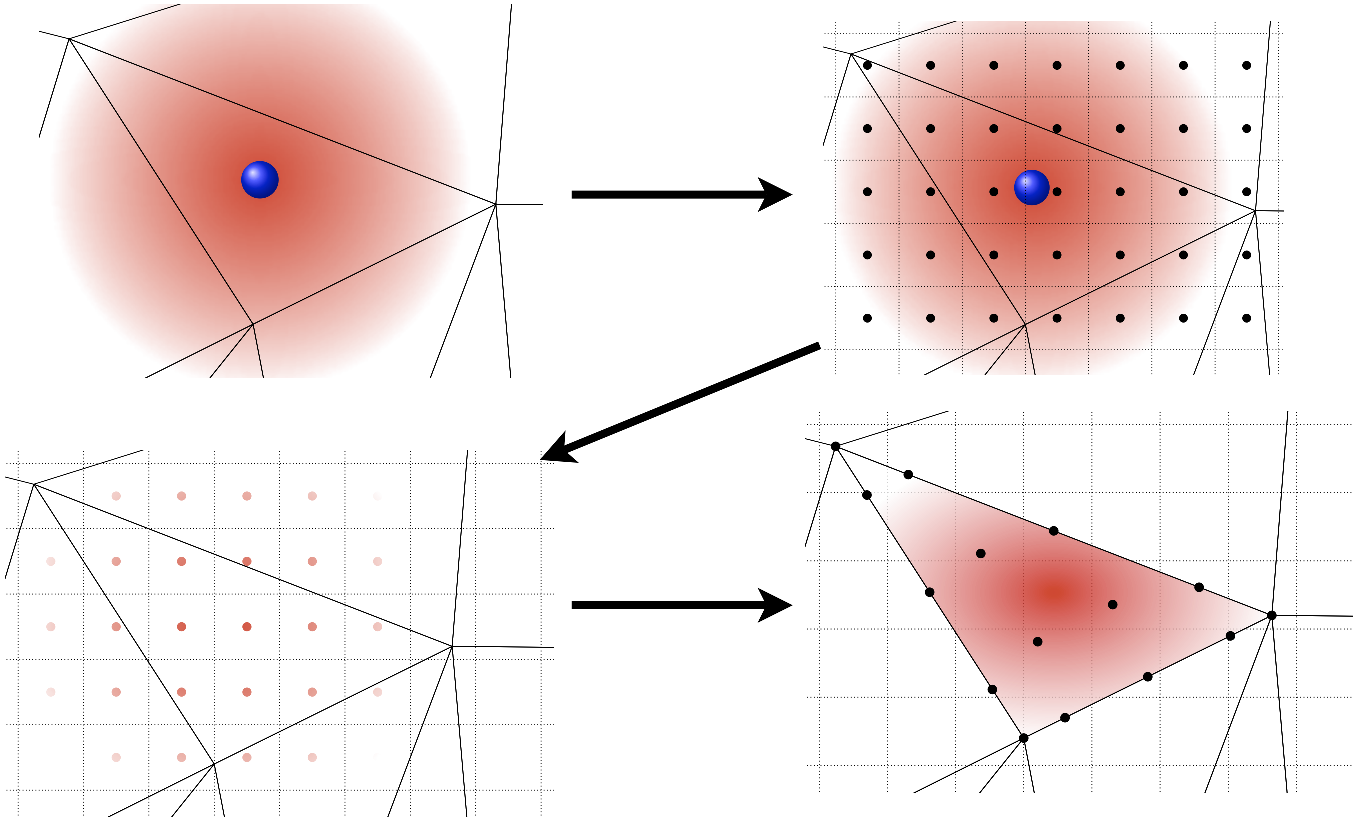
- About twice as fast as without the grid
- Results identical to simple shape deposition
- Cartesian grid must conform to local resolution in a locally adaptive way - or be very fine.

In most of the applications and tests so far, we have used deposition by shape.

# Deposition by Cartesian grid

The idea is

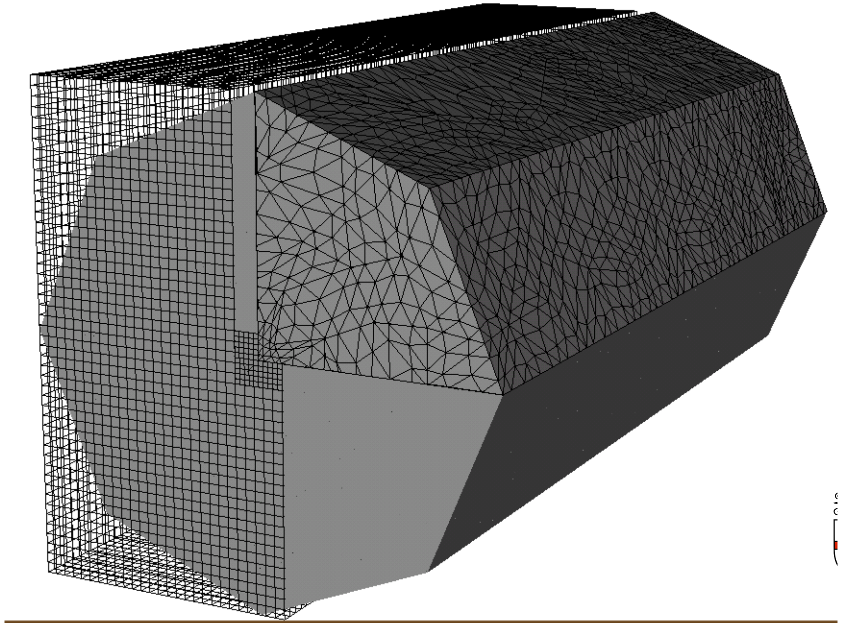
- Overlay grid with Cartesian grid
- Deposit all particles by shape onto Cartesian grid
- Map total local charge onto local nodes



# Deposition by Cartesian grid

The Cartesian->nodal mapping is often illconditioned

- Compute mapping in preprocessing
- Evaluate conditioning through SVD
- If poorly conditioned, reduce order of mapping
- Recompute mapping in LSQ sense



# Deposition by Cartesian grid

Features

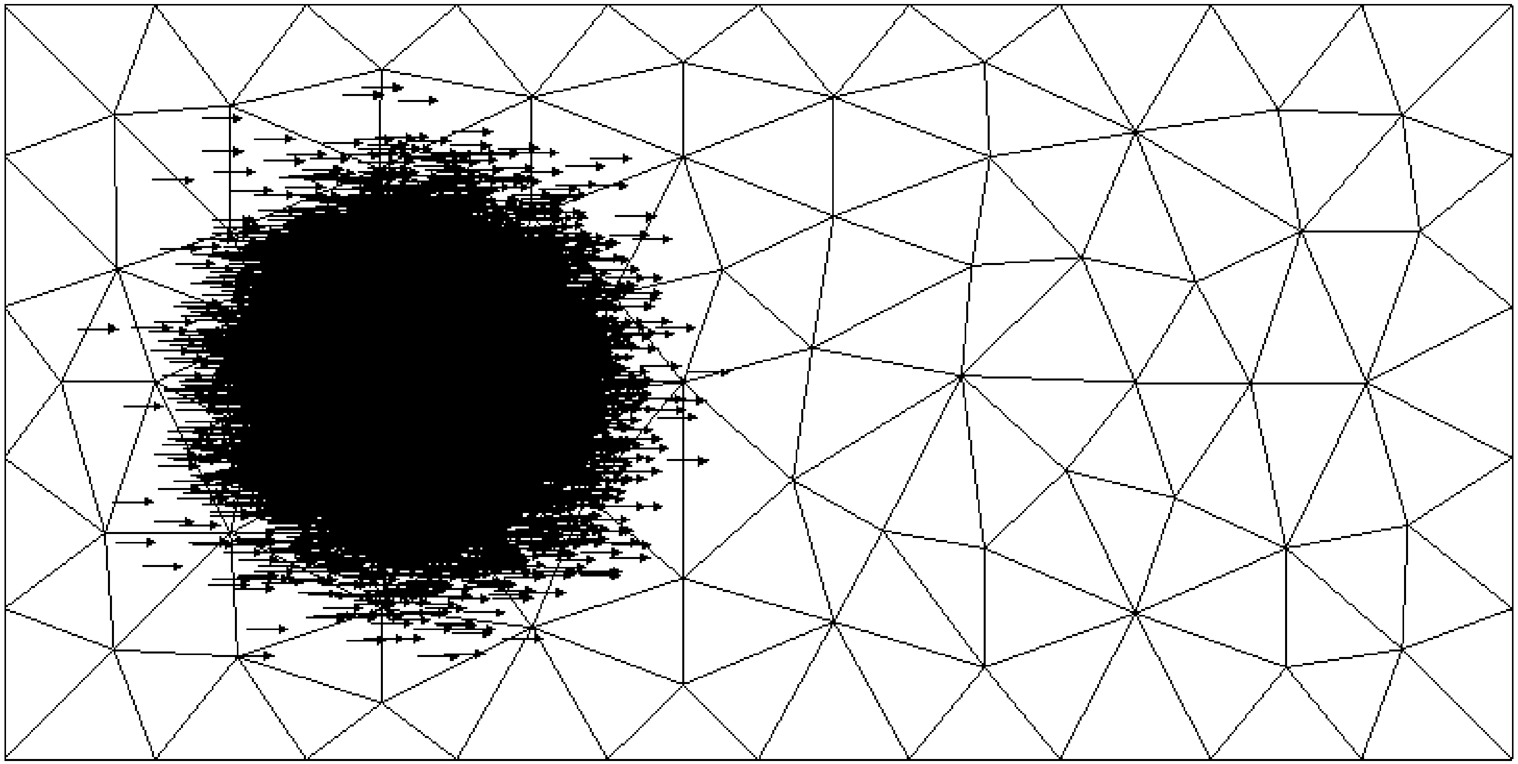
- Faster than other methods
- Requires locally adapted grid

.. but what about accuracy ?

Note: It is tempting to also perform push at the Cartesian grid in the spirit of VB-strategy although boundary problems persists in this case.

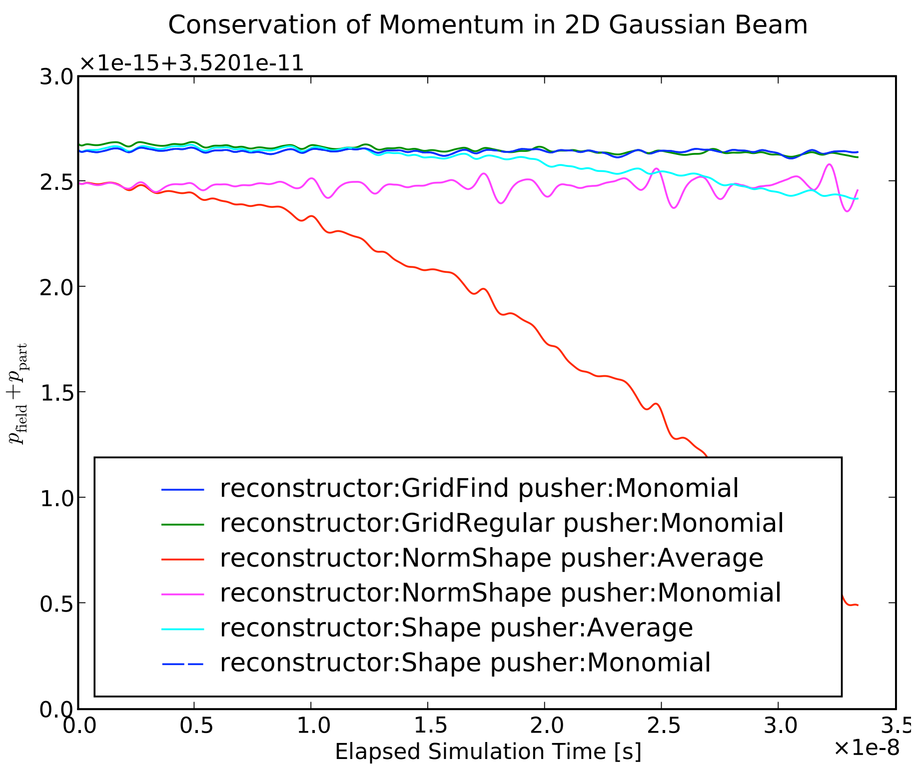
This has not been tried yet!

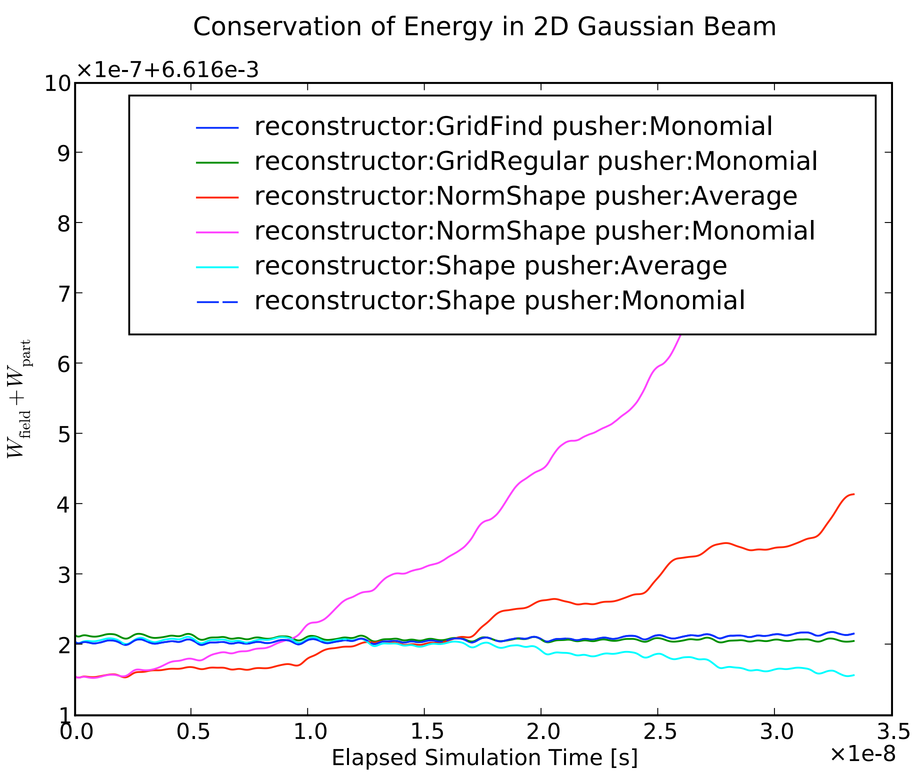
# Brief comparison



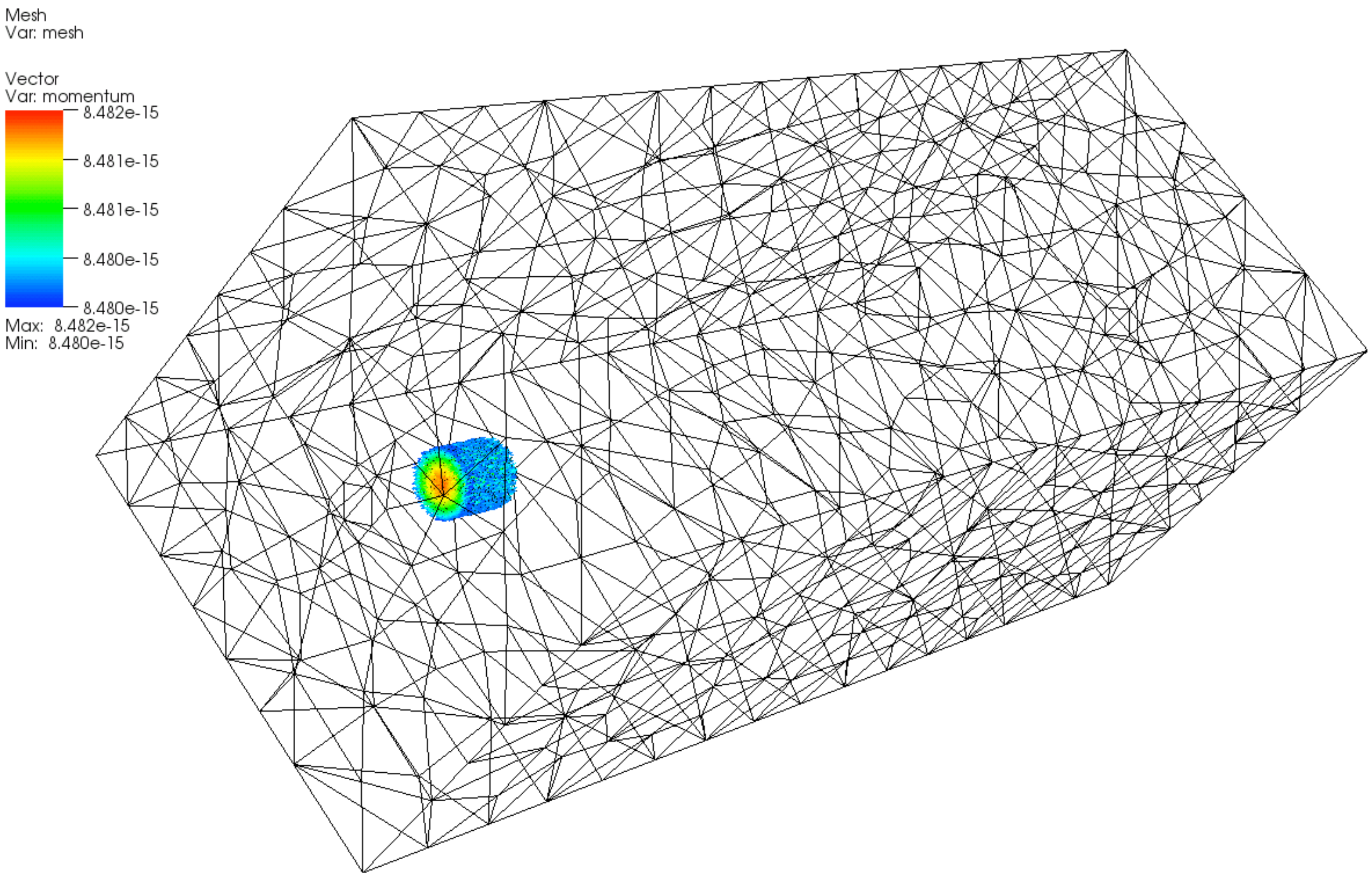
**Test case**: Gaussian beam in smooth beam tube.
2D, 20k particles, 3rd order elements

# Brief comparison



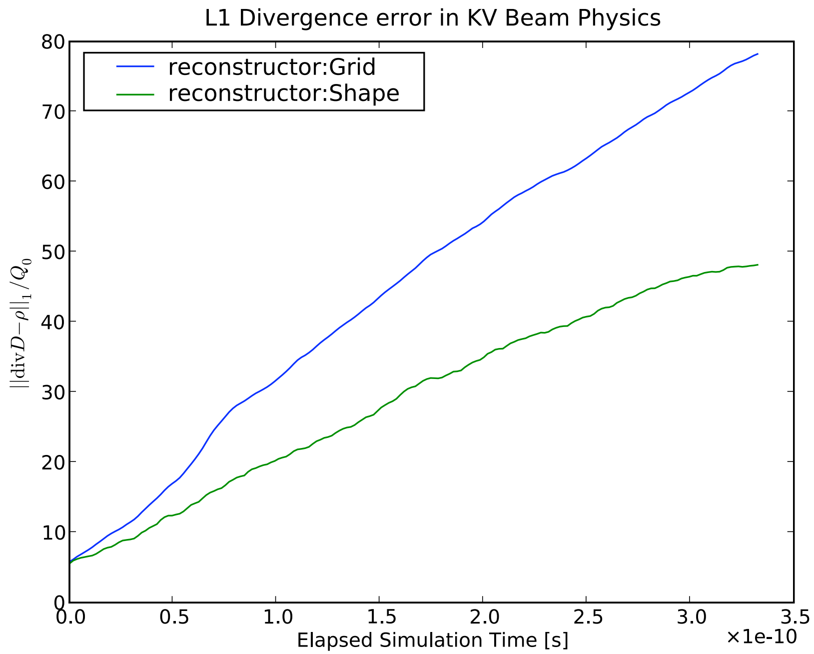



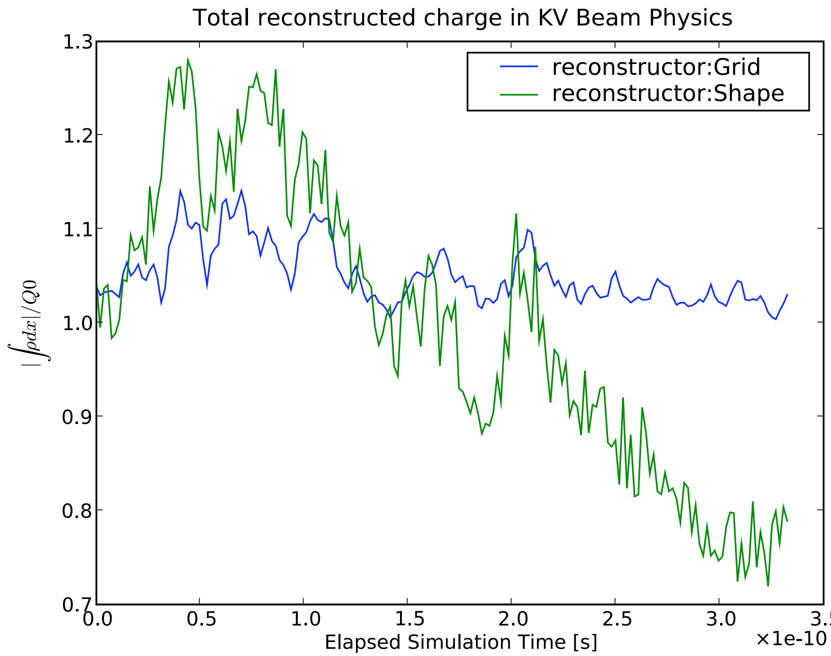
# Brief comparison

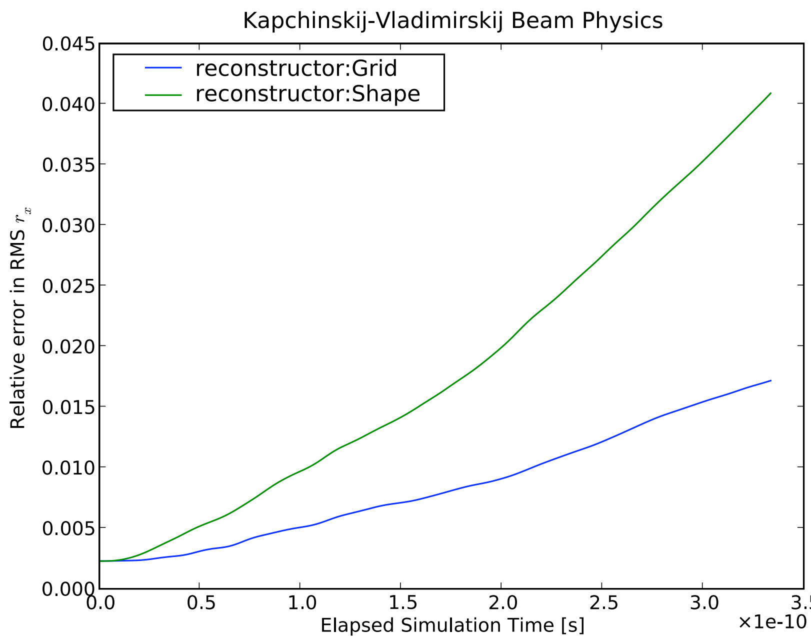



**Test case**: Kapchinsky-Vladimirsky beam in smooth beam tube.
3D, 20k particles, 3rd order elements

# Brief comparison




**Conclusion:**

.. Average force computation is not a good idea

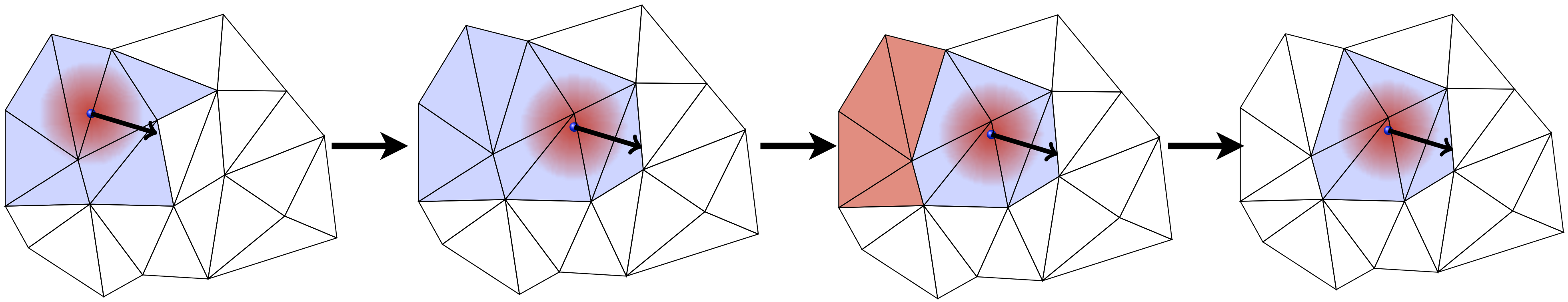
.. Grid based deposition is comparable in accuracy

.. but faster

# Charge conservation

The schemes do not guarantee this essential quality

**Goal**: satisfied to the order of the scheme.



Advective scheme solves local $(\rho_n)_t + v \cdot \nabla \rho_n = 0$

Preserves charge/energy -- but is VERY expensive (yet)

# Charge conservation

We currently consider two different techniques

**1. Boris correction**

$$E^* = E + \nabla\phi$$

$$\nabla^2\phi = \nabla \cdot E^* - \rho, \;\; \phi = 0, \; x \in \partial\Omega$$

$$E = E^* - \nabla\phi$$

- Enforced charge conservation exactly
- Requires global solve (or relaxation)
- Questionable at relativistic speeds

# Charge conservation

## II. Hyperbolic cleaning

Modify Maxwell's equations as

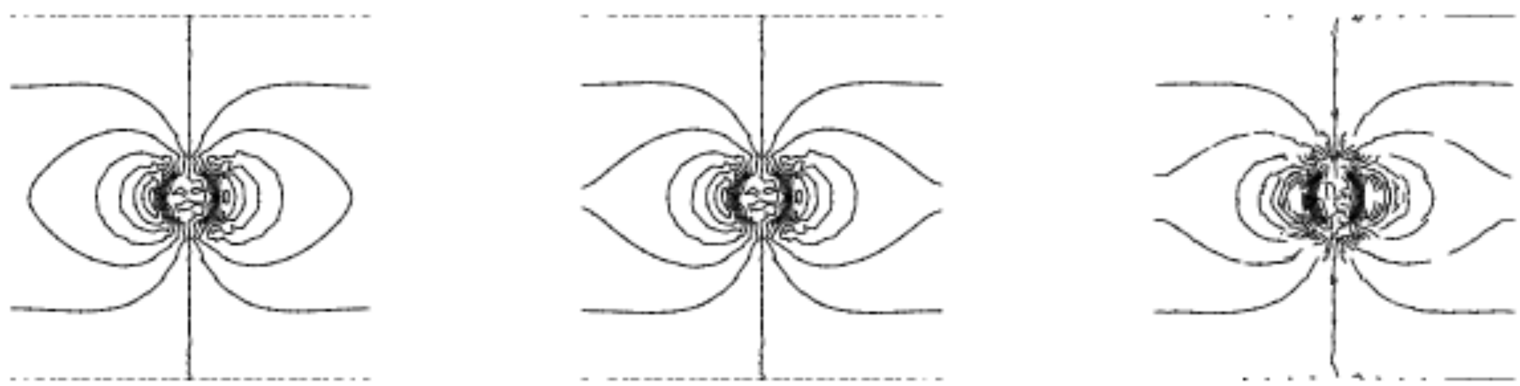
Munz et al, 2000

$$
\begin{aligned}
\varepsilon\partial_t E - \nabla \times H + \chi\nabla\phi &= -j \\
\mu\partial_t H + \nabla \times E + \chi\nabla\psi &= 0 \\
\partial_t\phi + \chi(\nabla\cdot E - \rho) &= -\nu\phi \\
\partial_t\psi + \chi\nabla\cdot H &= -\nu\psi
\end{aligned}
$$

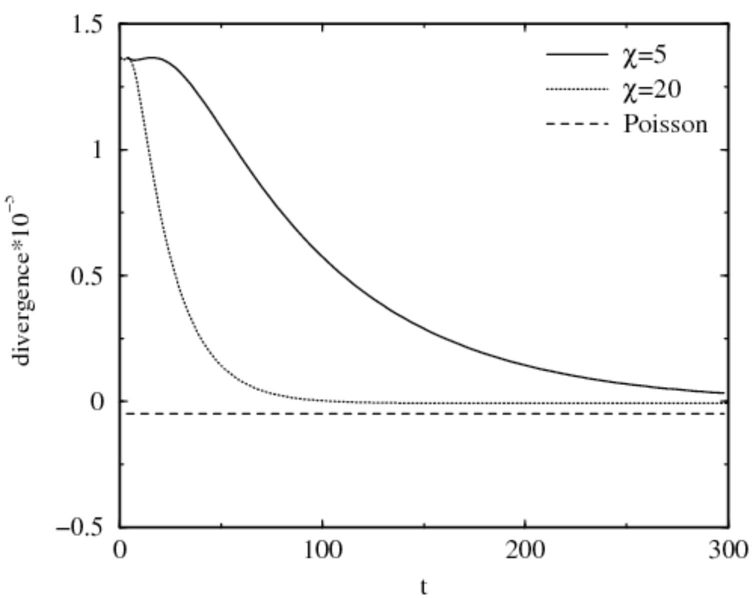
- Removes DC modes of Maxwell's equations
- Sweeps errors out with speed $\chi \gg c$
- Physical in relativistic regime
- Problematic for resonant problems and large problems with intermittant activity

# Brief comparison

## 2D ring of charges



$\chi=5$ $\chi=20$ Poisson




**Observations:**

.. hyperbolic cleaning seems superior and fast

.. but a high artificial velocity is needed

# Particles-boundary interactions

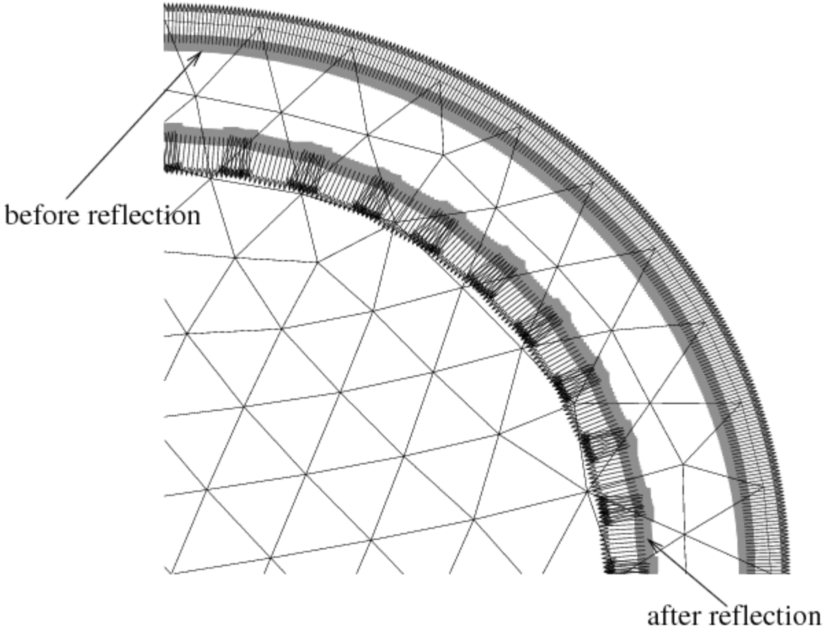
To correctly model particle behavior close to boundaries, we represent the geometry by a levelset

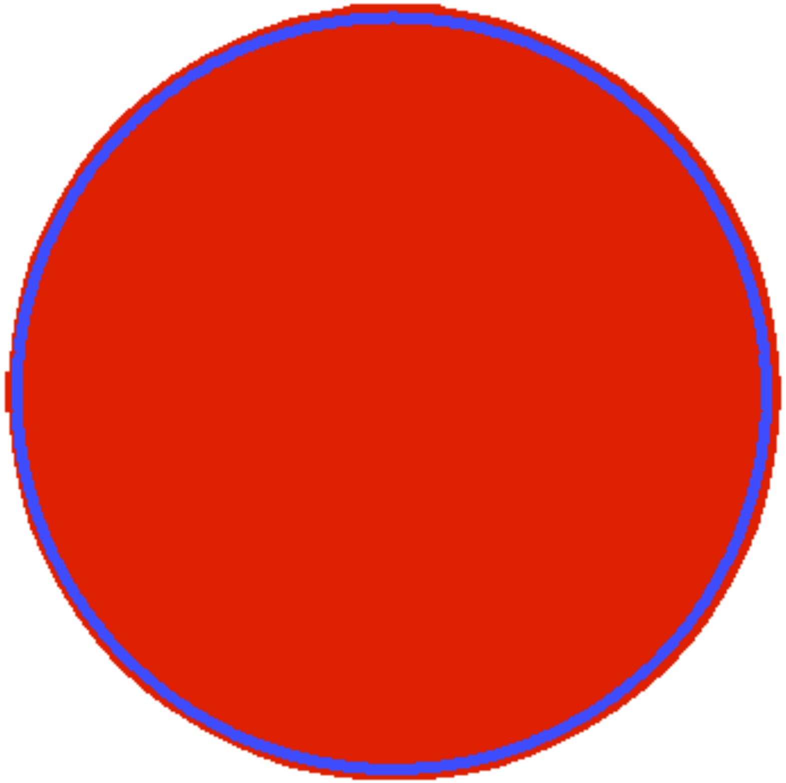
$$\partial_\tau \gamma + w \cdot \nabla \gamma = \mathrm{sgn}(\gamma_0) + \nu \nabla^2 \gamma, \quad w = \mathrm{sgn}(\gamma_0) \frac{\nabla \gamma}{|\nabla \gamma|}$$

- $(\gamma, w)$ represents the distance and normal to the geometry given by $\gamma_0$
- Boundary interactions can now be done using physical guidelines
- This works for any geometry

# Particles-boundary interactions

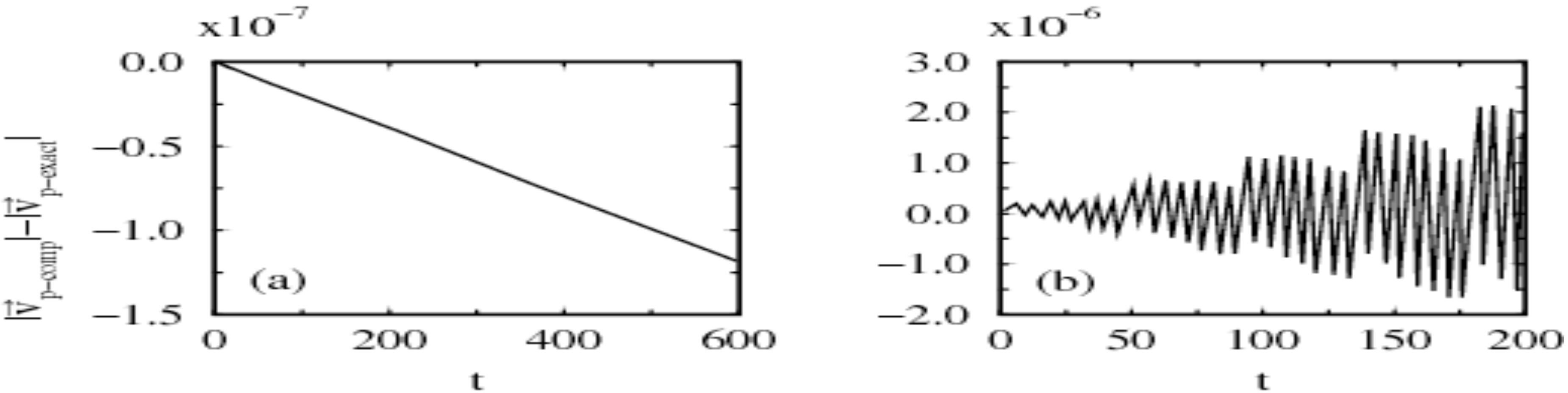
A simple reflection test



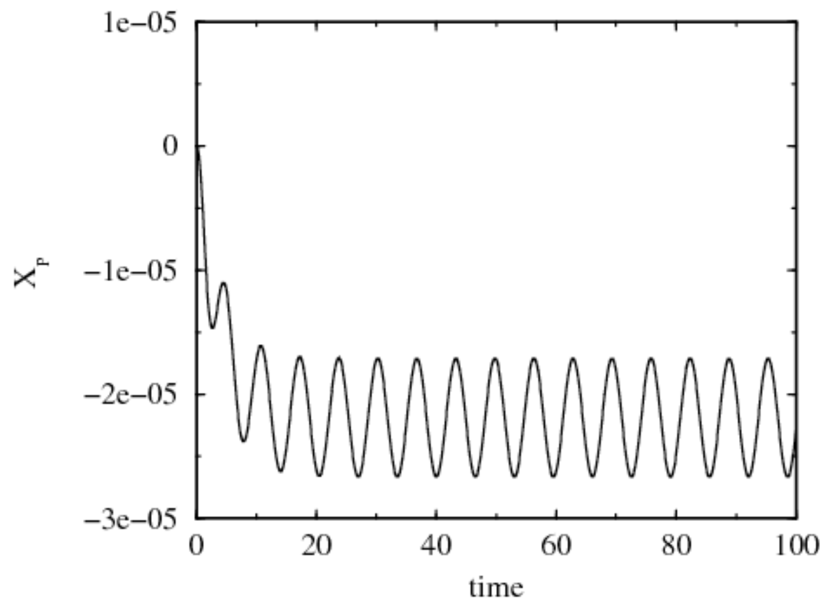



# A few sanity tests of everything

Larmor radius test in steady field




Test of self-force on single particle



# A few sanity tests of everything

*Grid heating*

- This is related to a physical requirement of resolving the Debye shielding length.
- It is a major problem is dense plasma simulations

- Typical solution - increased resolution or filtering

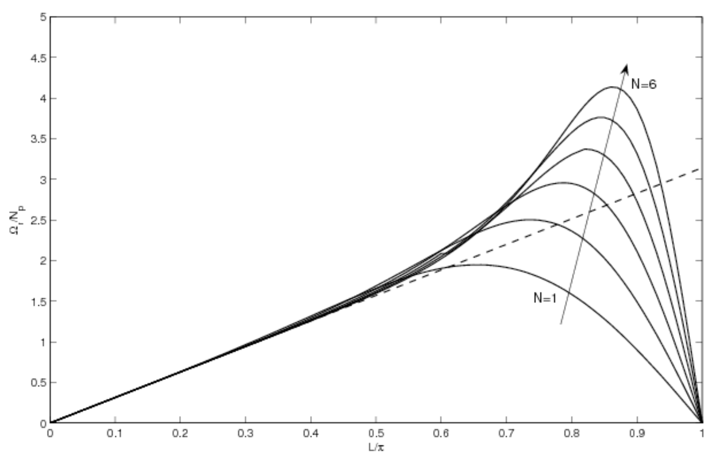
- Further options
  - Larger particles
  - Smoother particles

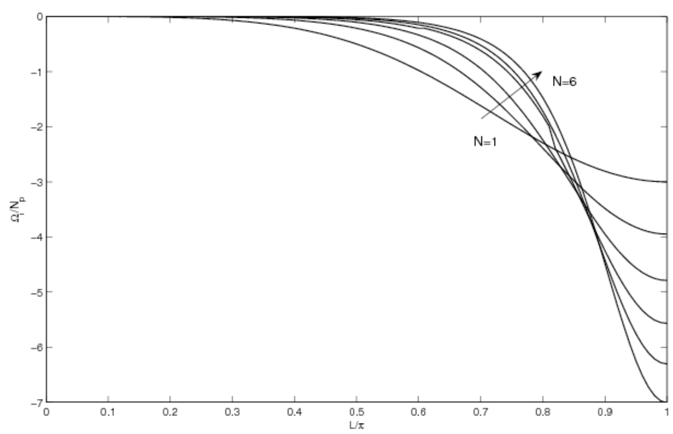
Grid heating persists - but is much better controlled

# A few sanity tests of everything

Numerical Cherenkov radiation

- High frequency waves propagates too slow
- Fast particles are able to pass waves
- This creates a numerical Cherenkov radiation
- Usually cured by filtering/damping

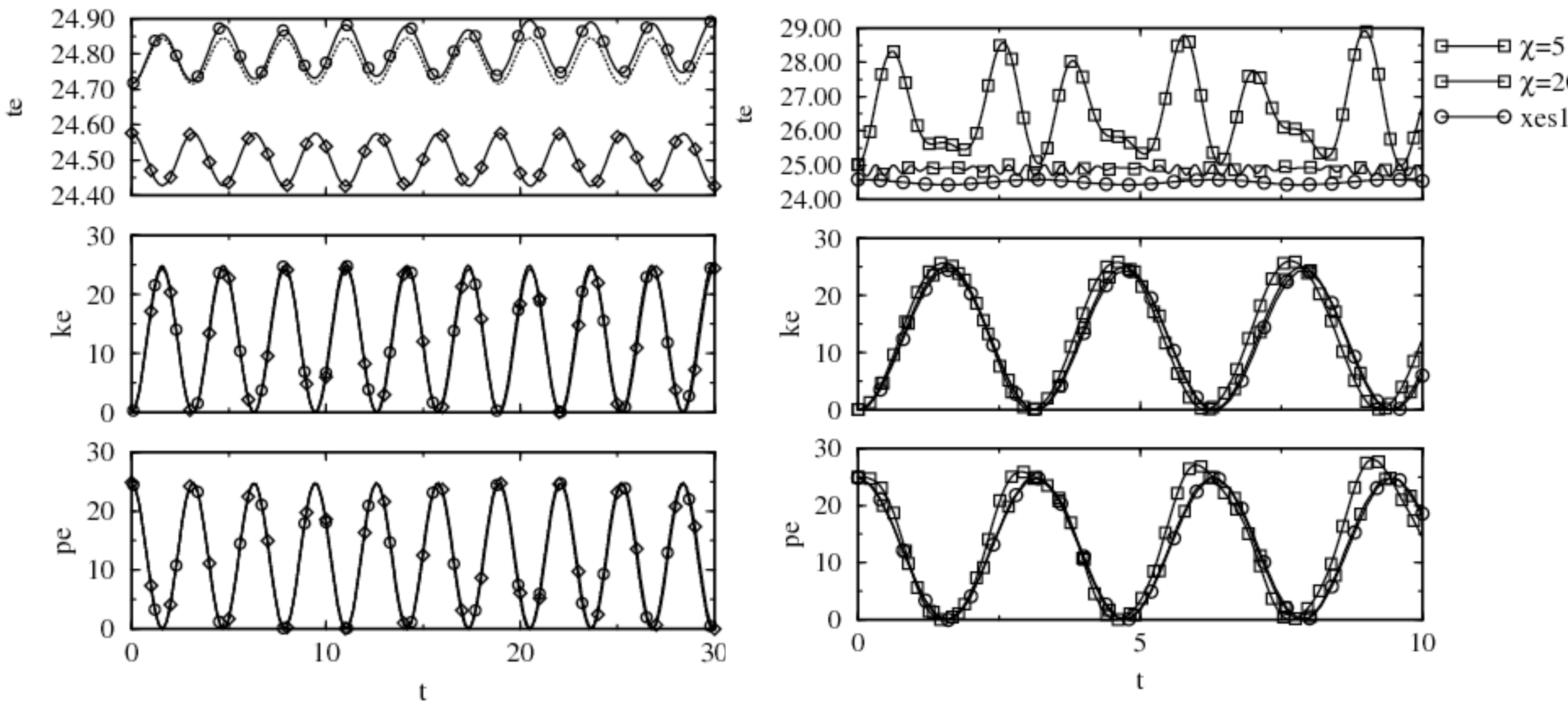



An upwind biased DG does not cure it
...but it helps a lot

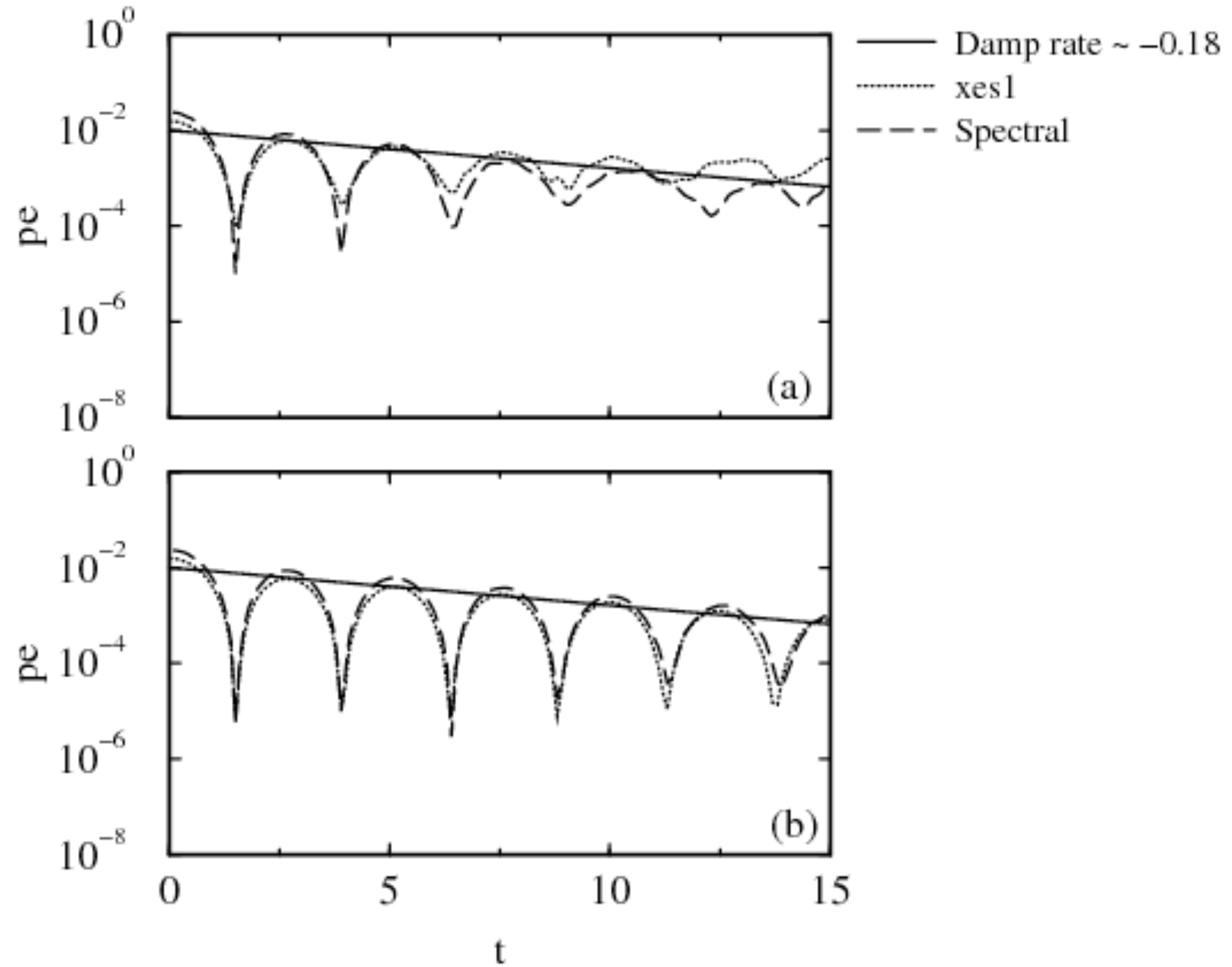
# A few sanity tests of everything

## Plasma waves and divergence cleaning



# A few sanity tests of everything

## Landau damping - strongly kinetic problem



# Sanity tests of the full scheme

A few observations

- *Larmor radius* shows RK to be sligthly dissipative
- Negligible *self-force* on single particle
- *Grid heating* remains but is much better controlled
- *Cherenkov radiation* remains but is much less of a concern
- Plasma wave problem confirms the importance of having *a high artificial velocity* (>10c) in the hyperbolic cleaning method
- Tests for other standard tests such as two-stream instabilities confirm good results

# Weibel instability study

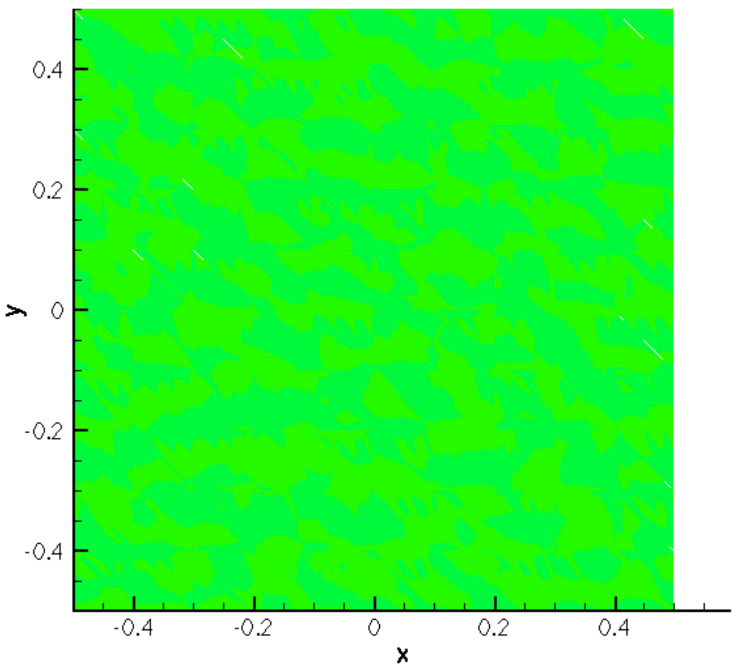
- Initial conditions
  - Homogeneous plasma with zero net charge. Constant background ion charge density.
  - Initial electron velocity (u,v)=(0.25,0.05)
  - Zero initial fields.

$$\omega_{pe} = 15\pi$$
$$u_{th} = 0.25$$
$$v_{th} = 0.05$$
$$\lambda_{De} = 0.0011$$

L=1

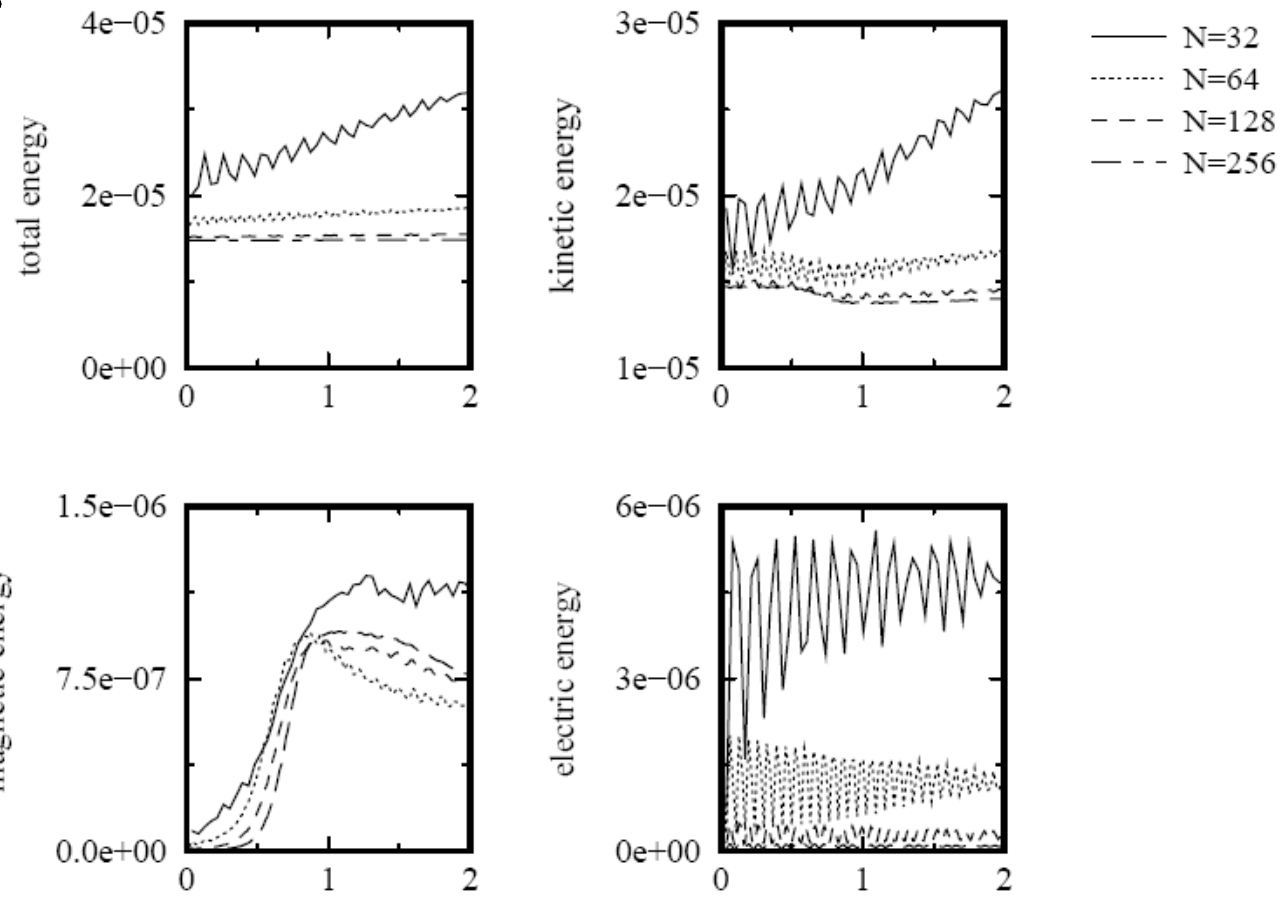
- The two velocities will evolve toward one thermal velocity

- The instability will show up as unstable growth of transverse electromagnetic waves



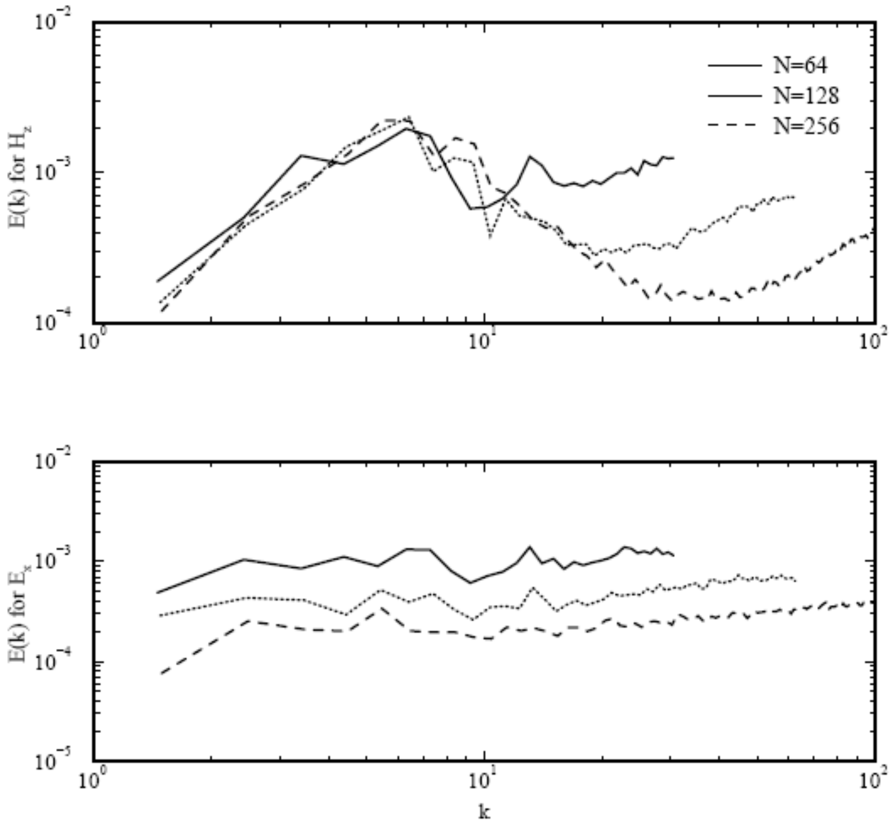
# Results with FDTD PIC

- Clear signs of *grid heating*
- At N=128 the solution has *reference value*
- Initial growth in *magnetic energy* is predicted by linear theory
- *Electric energy* is mostly noise
- 36 particles/cell



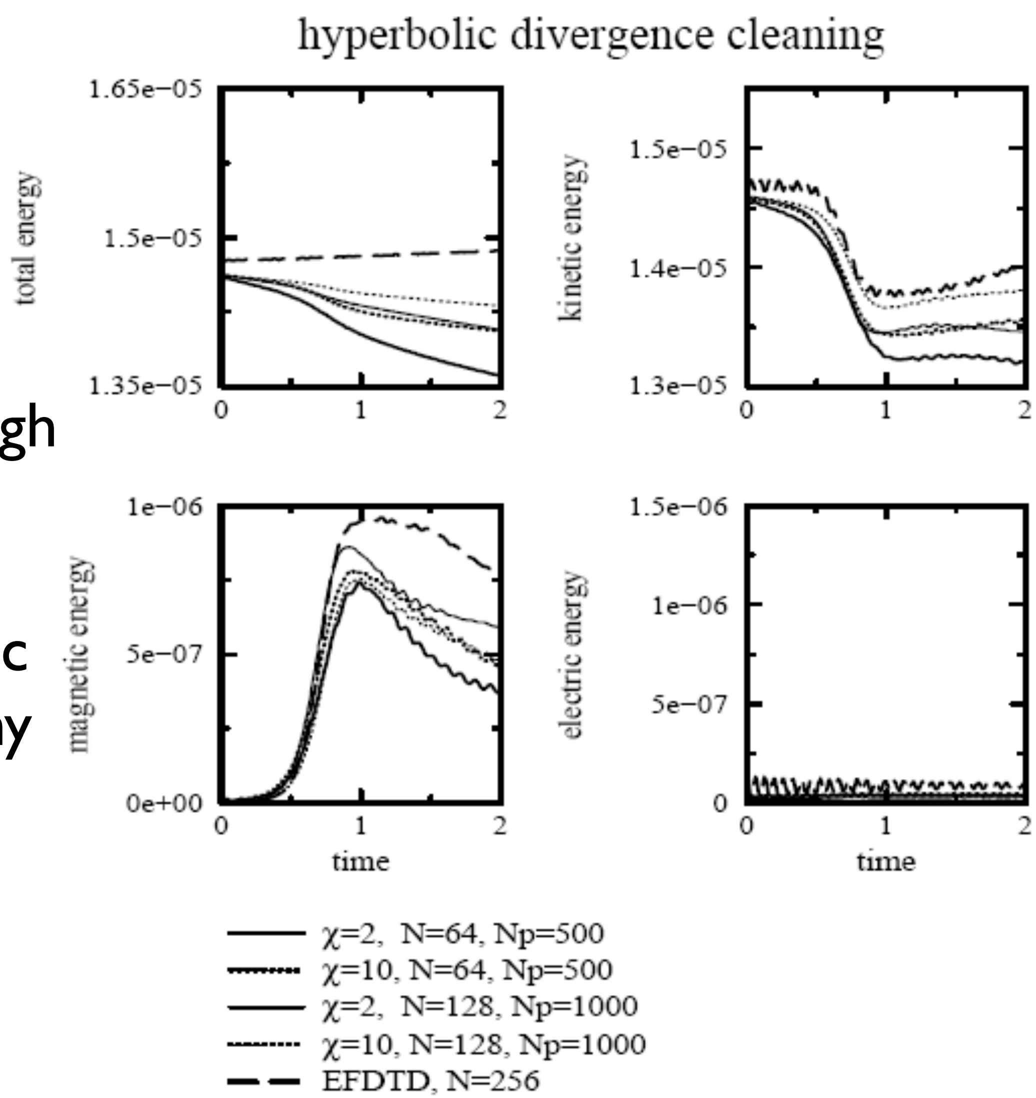
# Results with FDTD PIC

- Highest resolved wavenumber is about N/3 - higher than that the energy increases
- No dissipation yields unphysical growth
- Electric energy dominated by noise



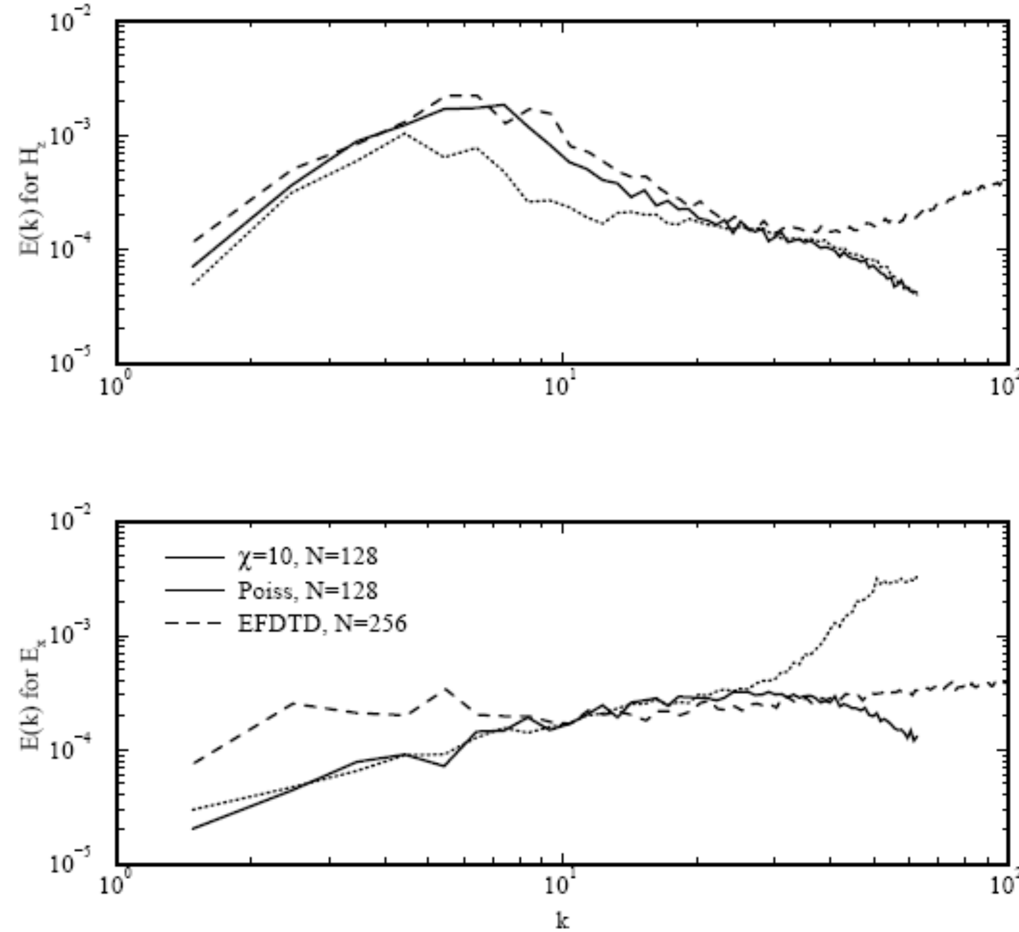
# Results with DG

- Reasonable agreement between the two cases
- Confirms the need for high artificial velocity
- Sligthly less peak magnetic energy -- not known why



# Results with DG

- Magnetic energy compares well
- Dissipative nature of DG scheme is clear in decay of spectrum

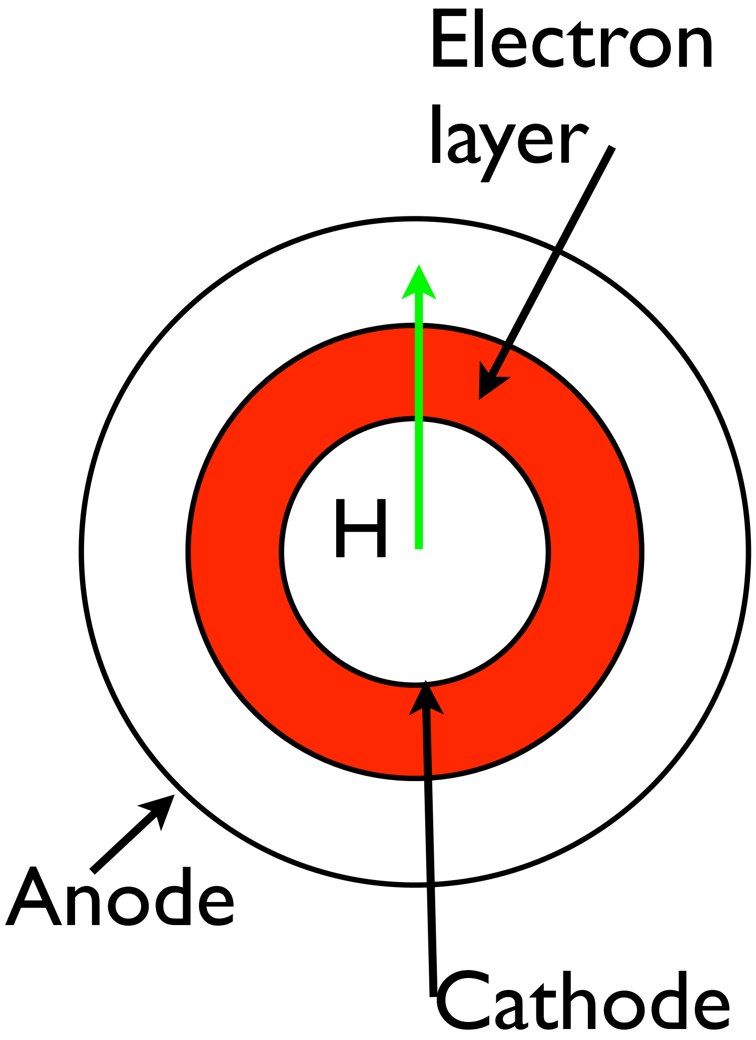


# Smooth bore magnetron

- Initial conditions
  - Constant potential and magnetic field
  - Exact 1D stationary solution (Davidson'89)

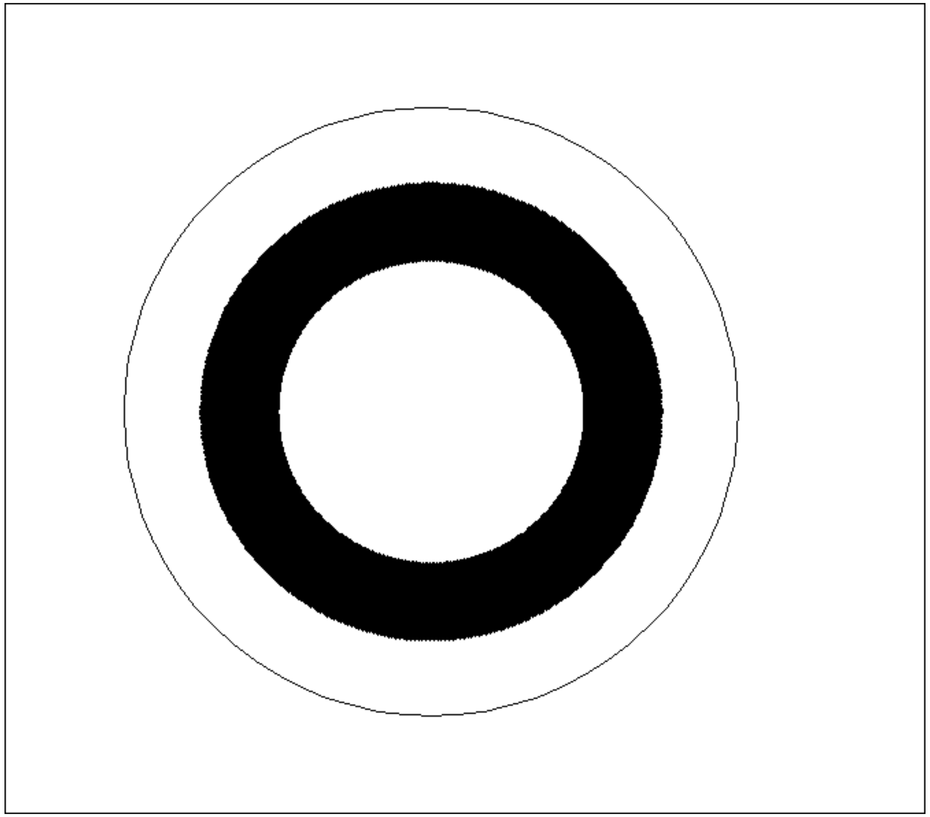
- The constant electric field rotates the electrons while the potential keeps the layer from reaching the anode

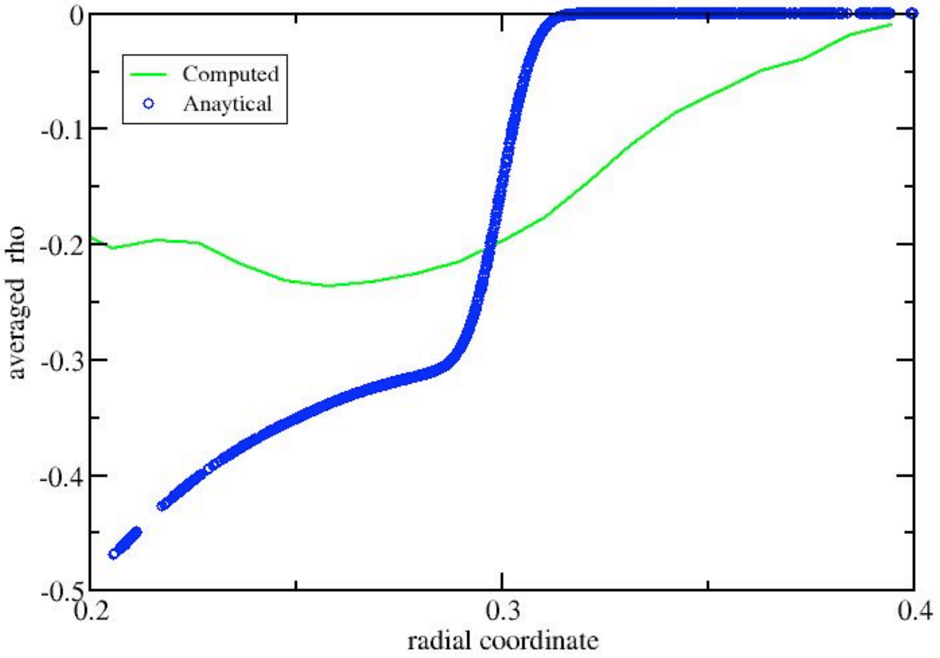
- The flow is unstable -- it is a known instability



# Smooth bore magnetron

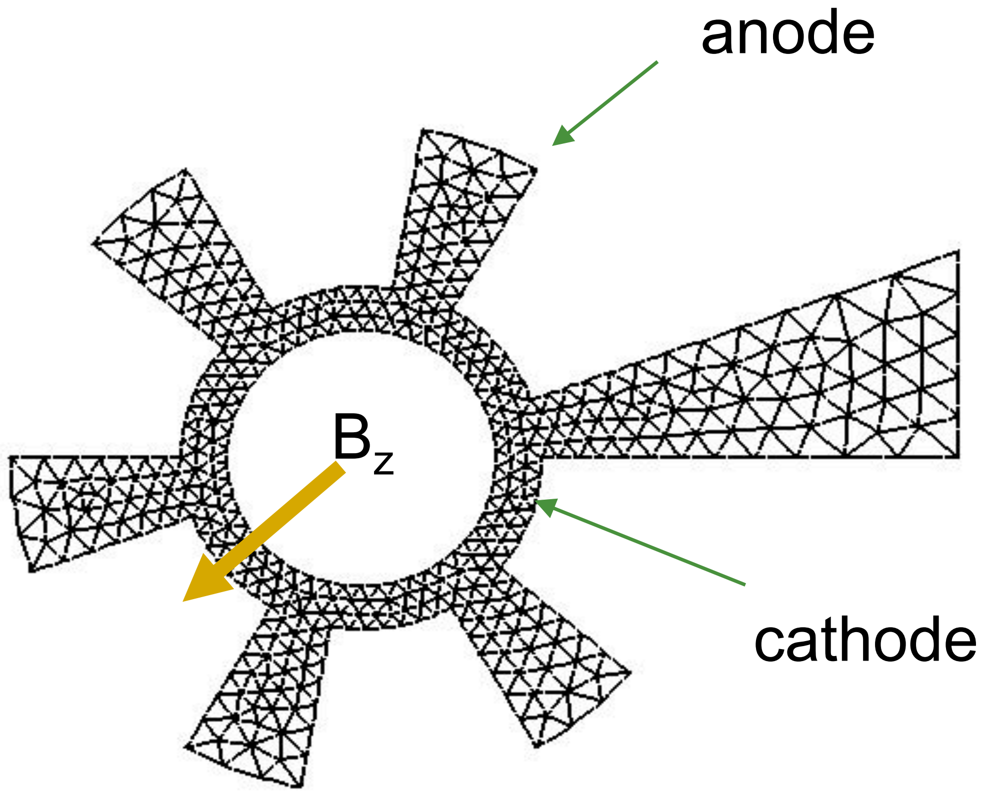
- Computations confirm the instability
- Average shows electron layer





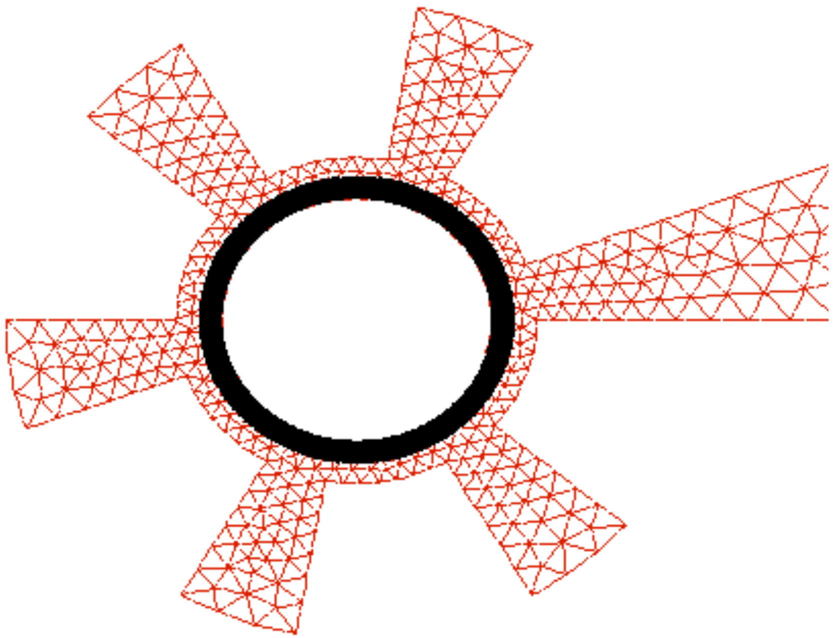
# A6 Magnetron

- Initial conditions
  - Brillioun flow
  - Fixed external magnetic field and potential
- PEC walls
- When a particle leaves the dom inject a new one

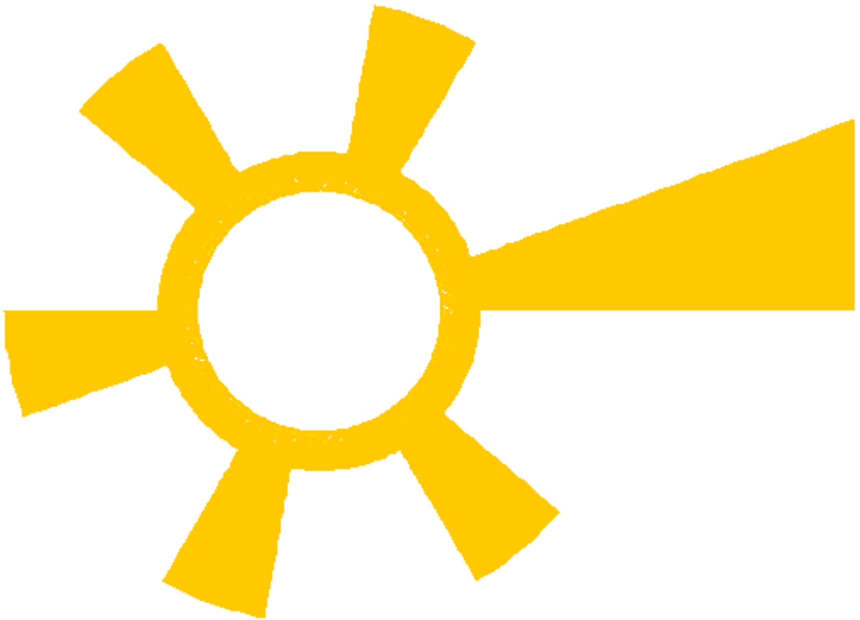


# A6 Magnetron

Particle dynamics



Radial field



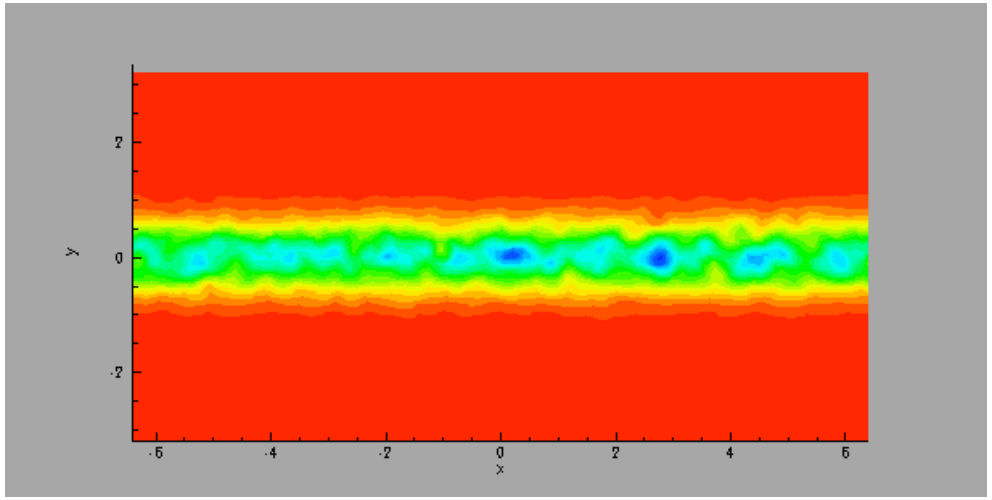
# Magnetic reconnection

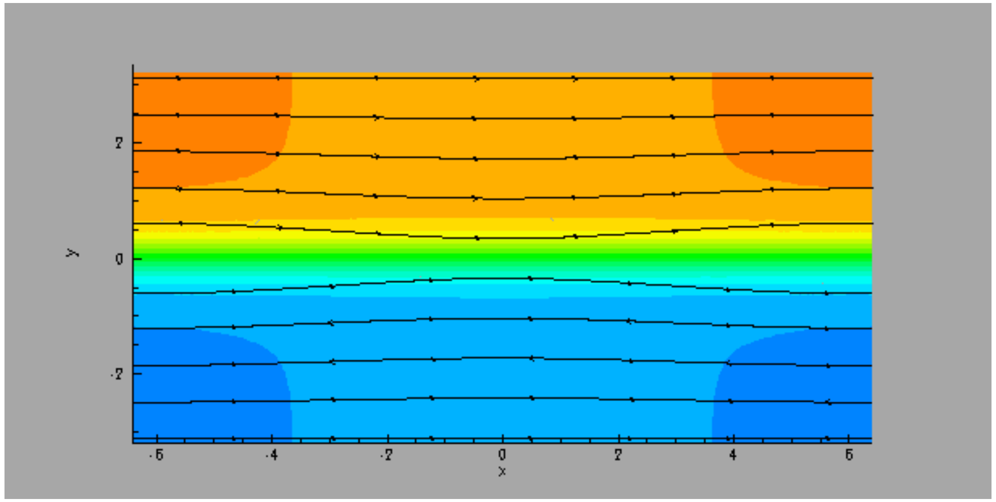
- Initial conditions
  - Harris current sheet
  - Perturbed magnetic field

- The magnetic field topology chances in time: inviscid magnetic reconnection

- The reconnection is accompanied by a sharp drop in the magnetic potential energy and an increase in the kinetic energy.

- This cannot be modeled with standard MHD



# Magnetic reconnection

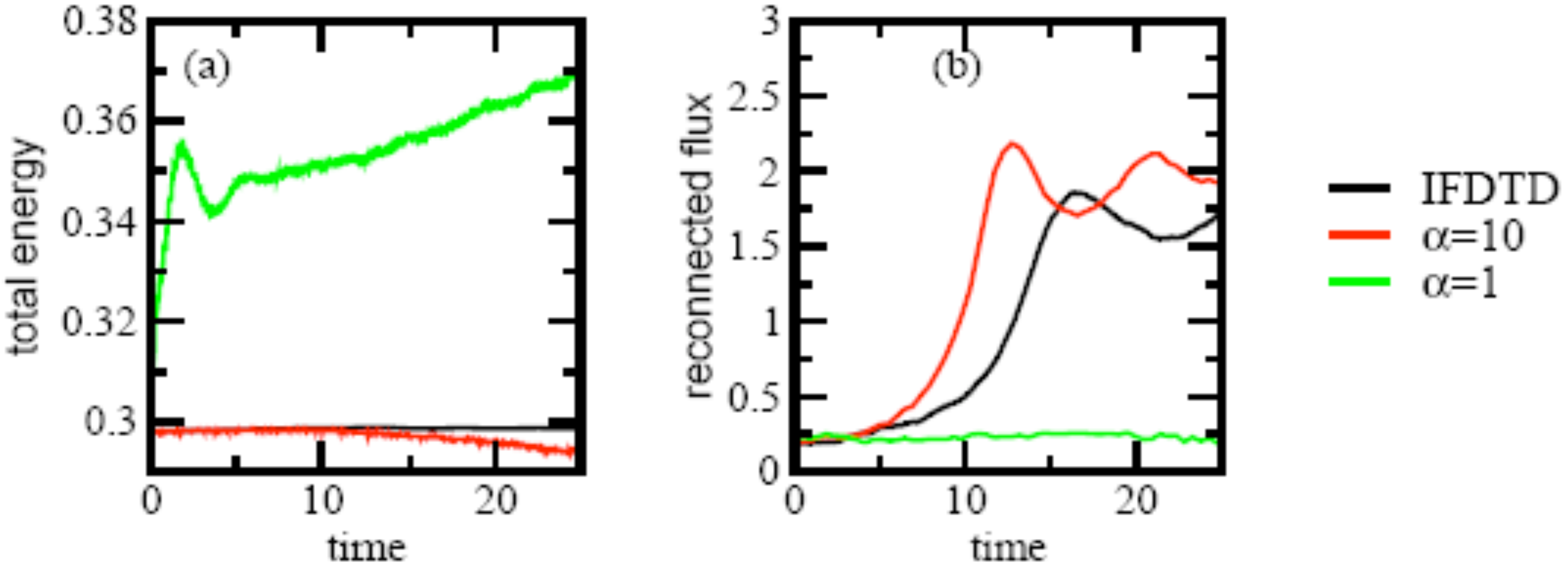
Benchmark is Implicit FDTD (w/ Lapenta, LANL)

- IFDTD
  - 32x32 grid
  - 25k particles

- DG PIC
  - 32x16x2 elements
  - 100k particles
  - Radius of particle ~ h

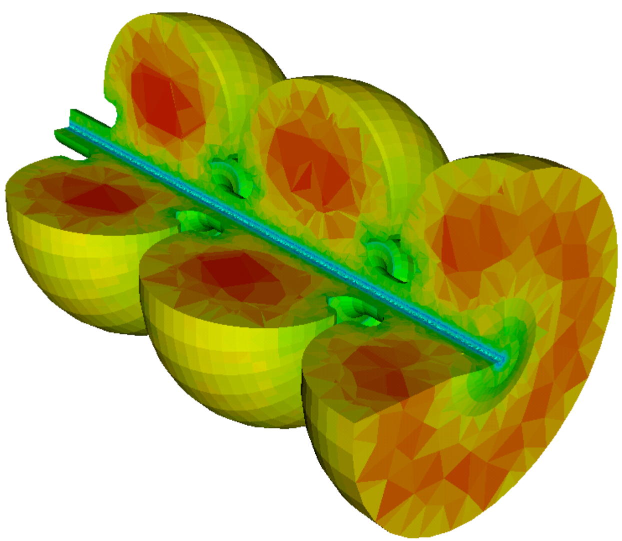
# Magnetic reconnection

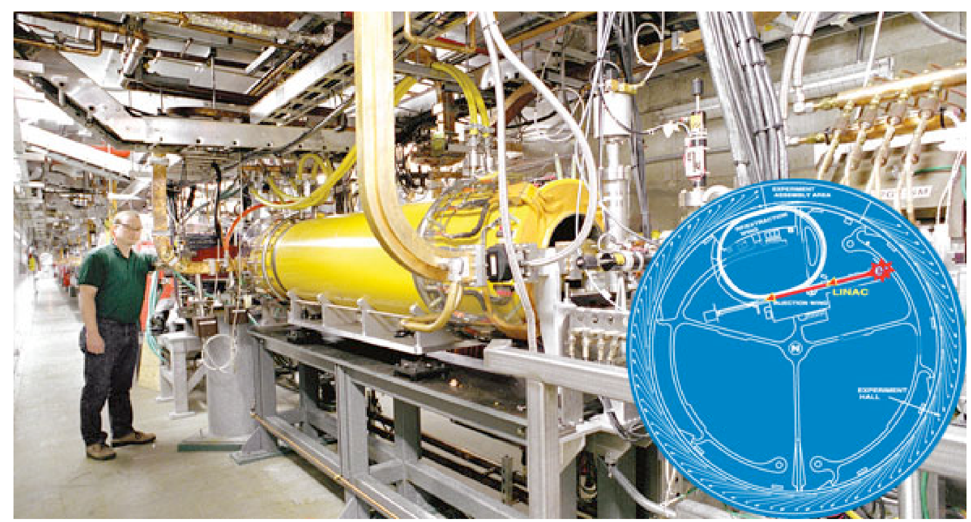
- Importance of smooth particles for grid heating is clear
- Reasonable agreement with this and results of others



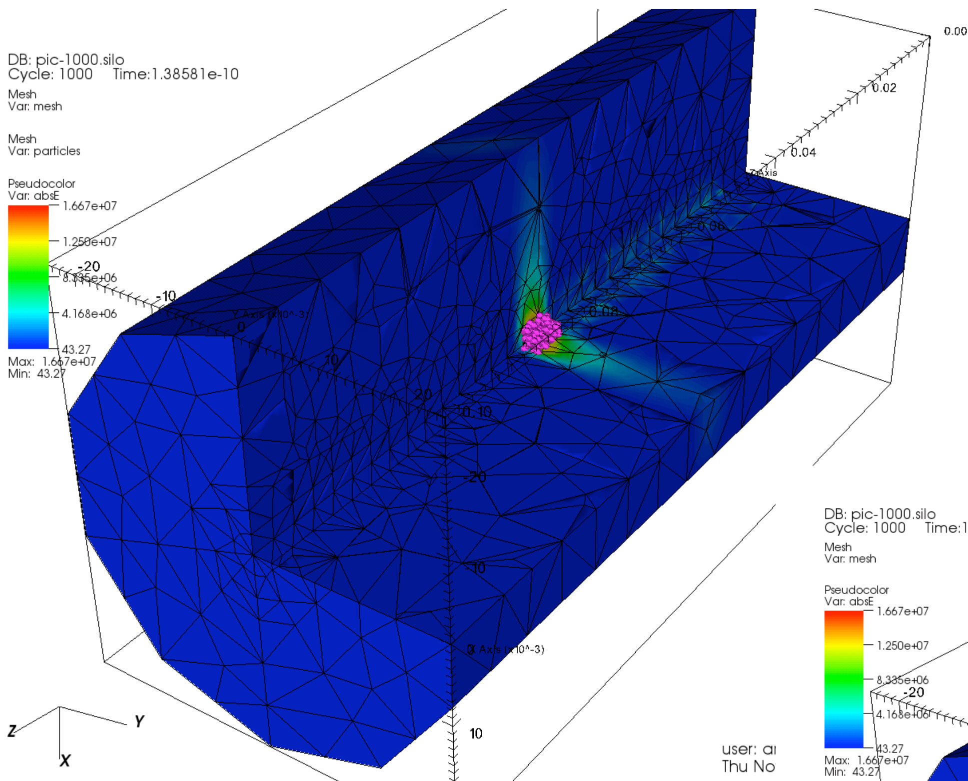
# Closer to the application

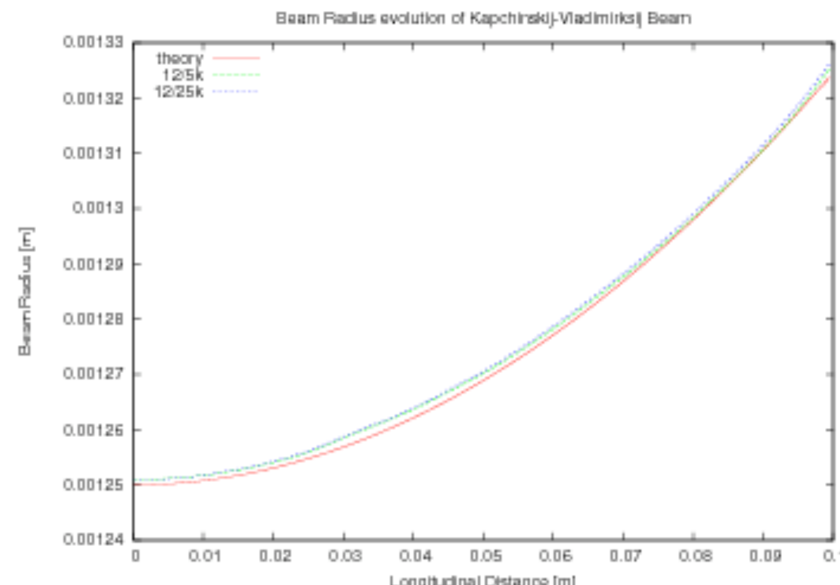
## Advanced Photon Source - ANL

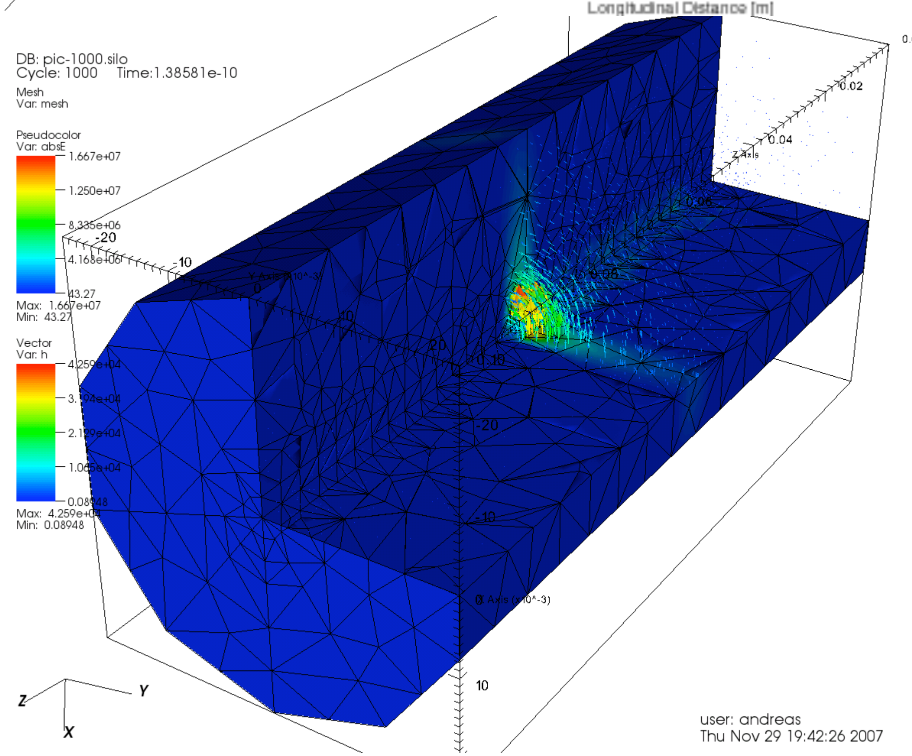





# Closer to the application

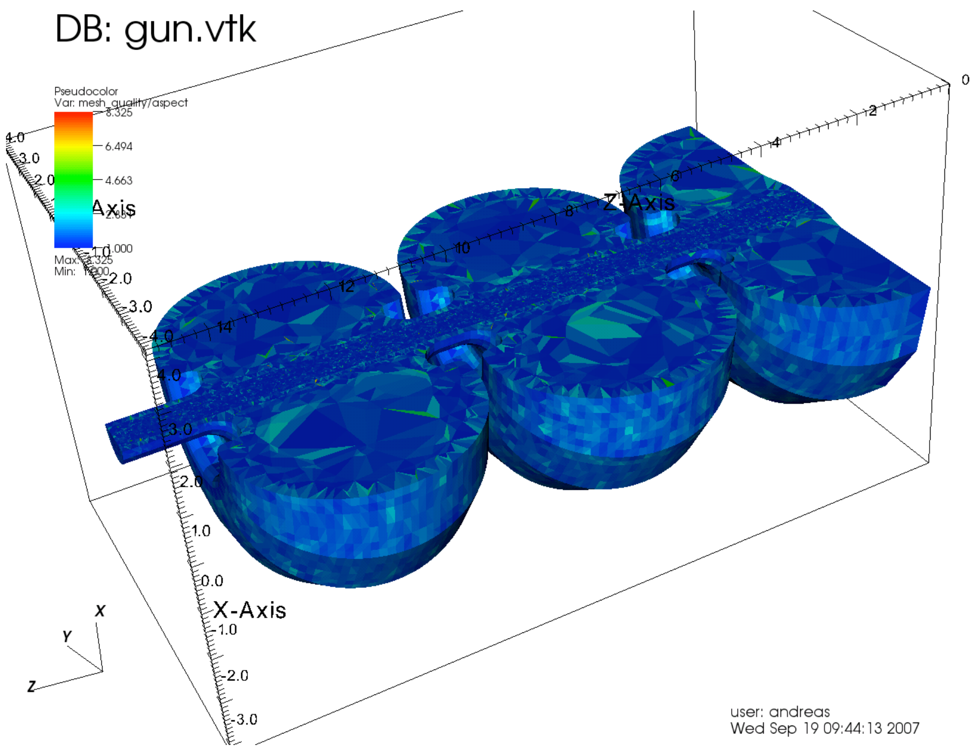


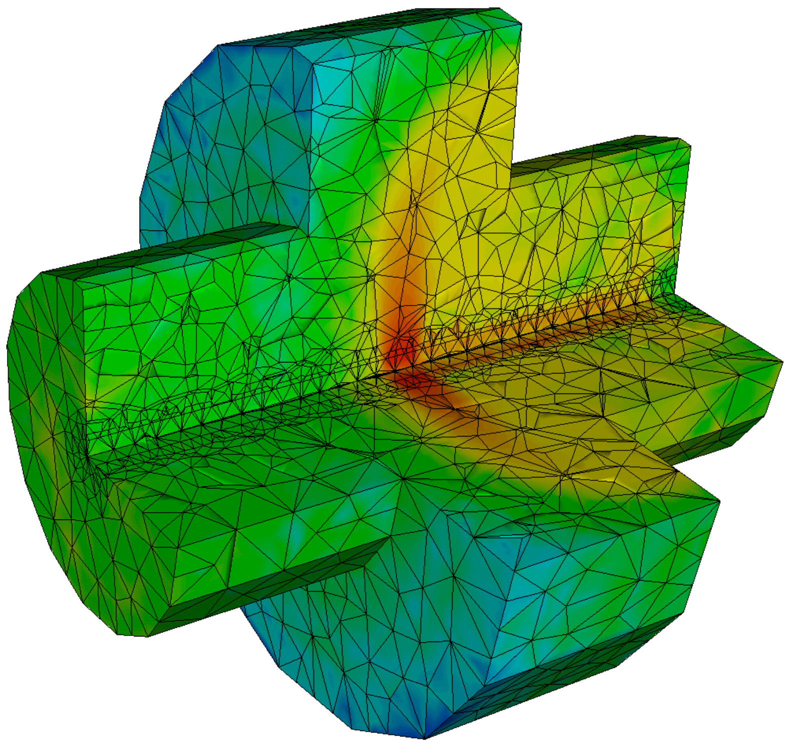



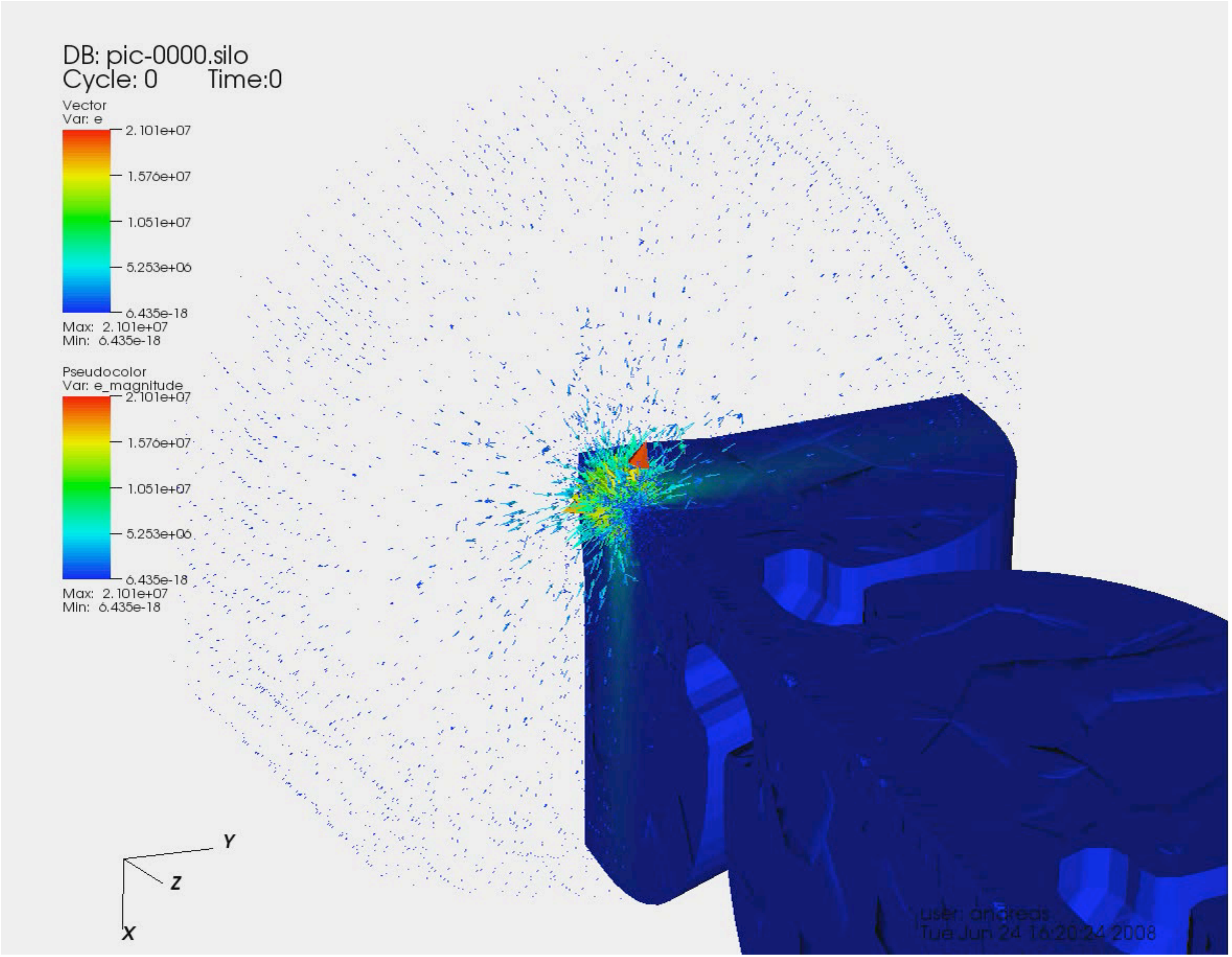
# Closer to the application

16k elements
25k particles

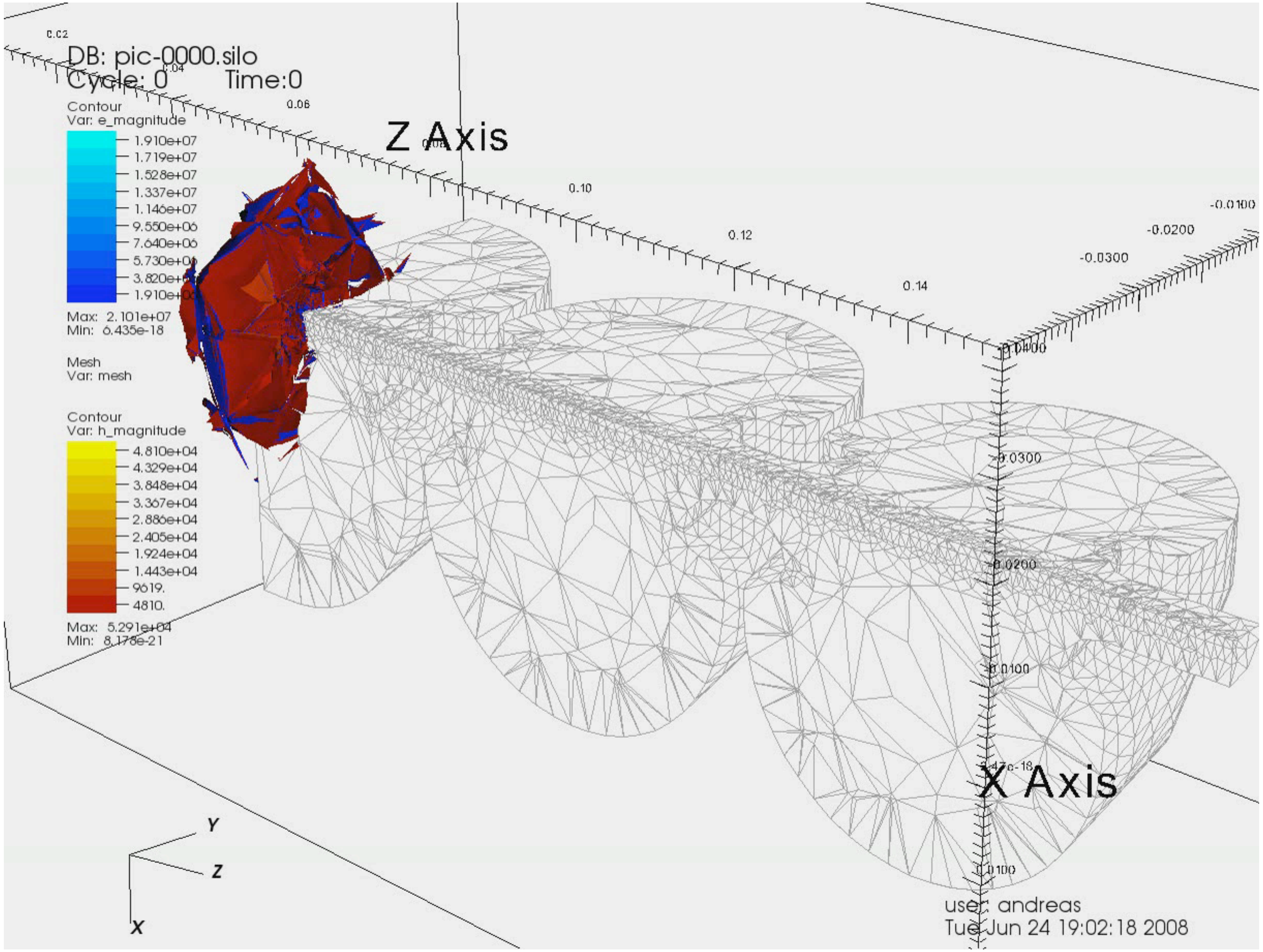


# Closer to the application



# Closer to the application



# Other related efforts

- RK-IMEX time-stepping to address stiffness in hyperbolic cleaning approach

- Steps toward particle adaptive solvers
  - Splitting/coalesce strategies
  - Kinetic error estimation

- Hybrid schemes (DG does the fluids well!)

- Efficient basis families for Vlasov and df solvers

*We can discuss these offline*

# Concluding remarks

New DG based PIC scheme shows some promise

- Geometric flexibility and variable order
- Good inherent properties
- Much progress made in particle part
- Some testing in both 2D and 3D

Still *many questions remain open*

- Better understanding of phase space resolution
- Adaptivity and parallel implementations
- Improved charge conservation
- Careful attention to boundary/emission models
- Hybrid plasma/fluid modeling
- Speed !

# Concluding remarks

The good news is that 'they' struggle with many of the same problems

... and have some additional ones we do not have

What we need more than anything is a 'killer application' - one 'they' struggle to do

- Electrically large and geometrically complex
- Lots of EM and field/particle-boundary interaction
- Hybrid physics

We have the hammer - now find the nail

# Questions/remarks

?

Thank you !

Jan.Hesthaven@Brown.edu

# Something extra

By A Kloeckner and T Warburton

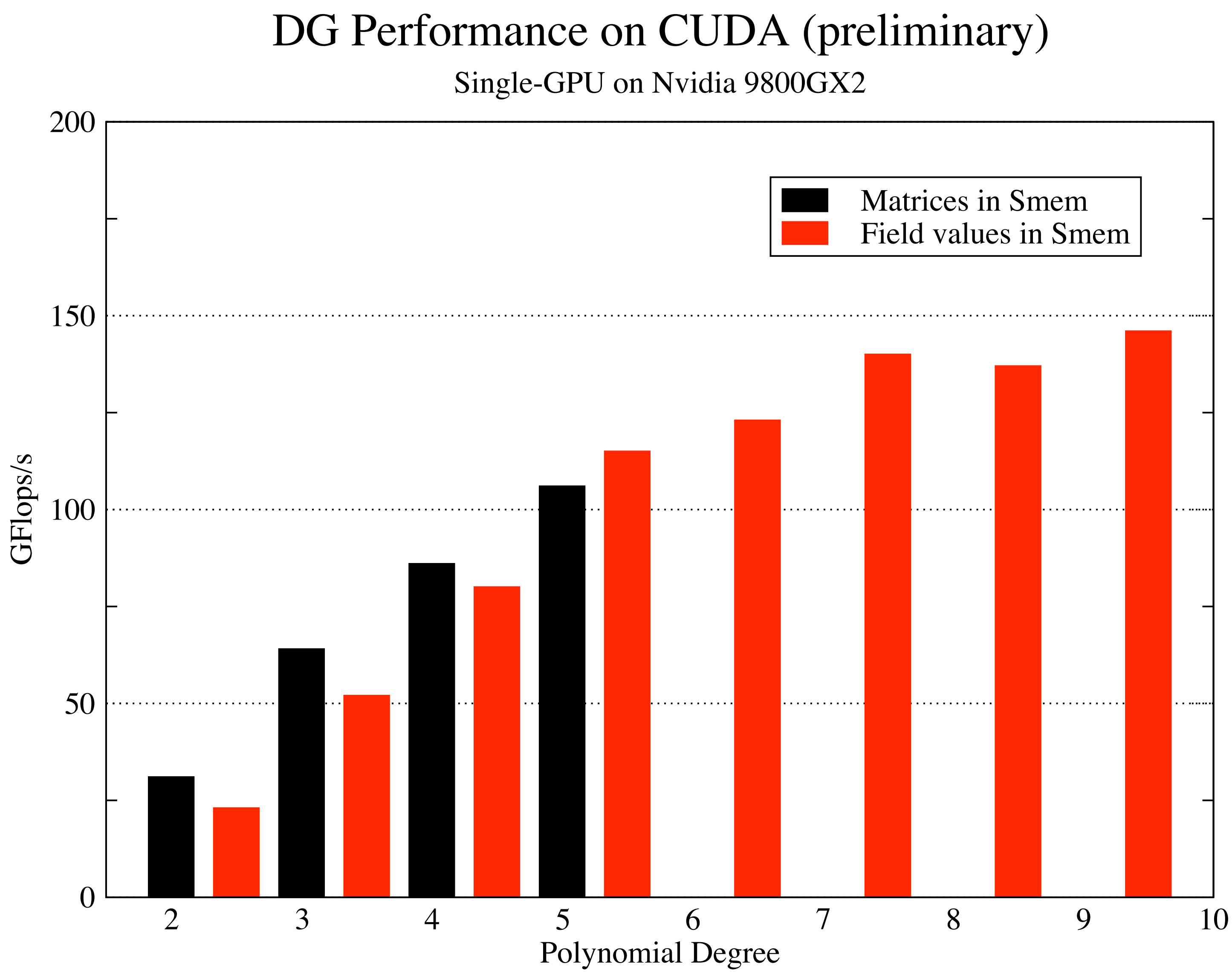
A major criticism against DG is cost

DG on Graphics Processing Units (GPU)

- GPUs have deep memory hierarchies
  - DG work is mostly local
- Compute >> memory bandwidth
  - DG is arithmetically intense (high-order)
- GPU's prefers local workloads
  - DG is local by nature

**A match made in heaven ?**

# Something extra




CPU: 2-3 Gflops

>50 speedup !

# Something extra

GPUs are truly supercomputers at commodity prices, but they are not really designed for commodity computing

Fortunately it looks like DG (with a little work) can be implemented to harvest the speed of the GPU.

This may actually be a break for DG in general.

On a 8 node/16 GPU card, T Warburton has demonstrated close to 2TFlops for Maxwell's by combining GPU and MPI!

# Questions/remarks

?

Thank you (again) !

Jan.Hesthaven@Brown.edu