The Path to Full Geometry 3D Vacuum Arc Simulation

**Jeremiah Boerner[1], Matthew Hopkins[1],
Edward Barnat[2], Paul Crozier[3],
Matthew Bettencourt[3], Christopher Moore[1],
Russell Hooper[4], Lawrence Musson[5]**

[1]Nanoscale and Reactive Processes,
[2]Laser, Optics, and Remote Sensing,
[3]Scalable Algorithms,
[4]Electrical Systems Modeling,
[5]Advanced Device Technologies,
Sandia National Laboratories
Albuquerque, NM, USA

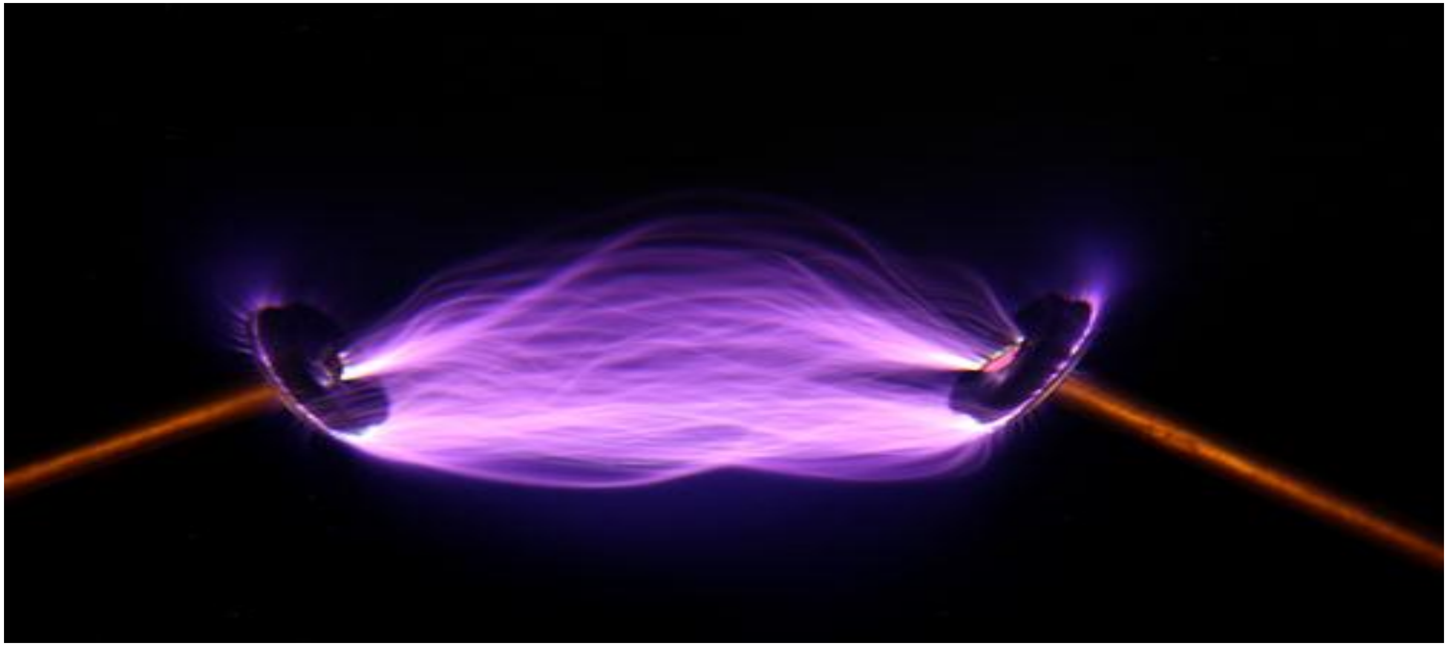
3rd International Workshop on Mechanisms of Vacuum Arcs

October 1 – 4, 2012

Sandia National Laboratories is a multi-program laboratory operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin company, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000

Overview

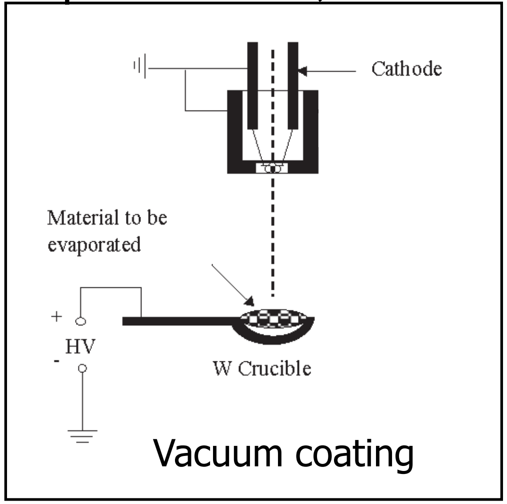
- Motivation: Low temperature plasmas and vacuum arc initiation
- Challenges for modeling and simulation of arcs
- Aleph simulation tool and examples of unique capabilities
- “State of the Arc” copper coplanar simulations



Bicone Tesla Arc, http://www.flickr.com/groups/tesla_coil/pool/34488155@N03/?view=md

Applications and Model Requirements

We're interested in low temperature collisional plasma phenomena, and transient start-up of arc-based devices.



Vacuum coating

Examples:

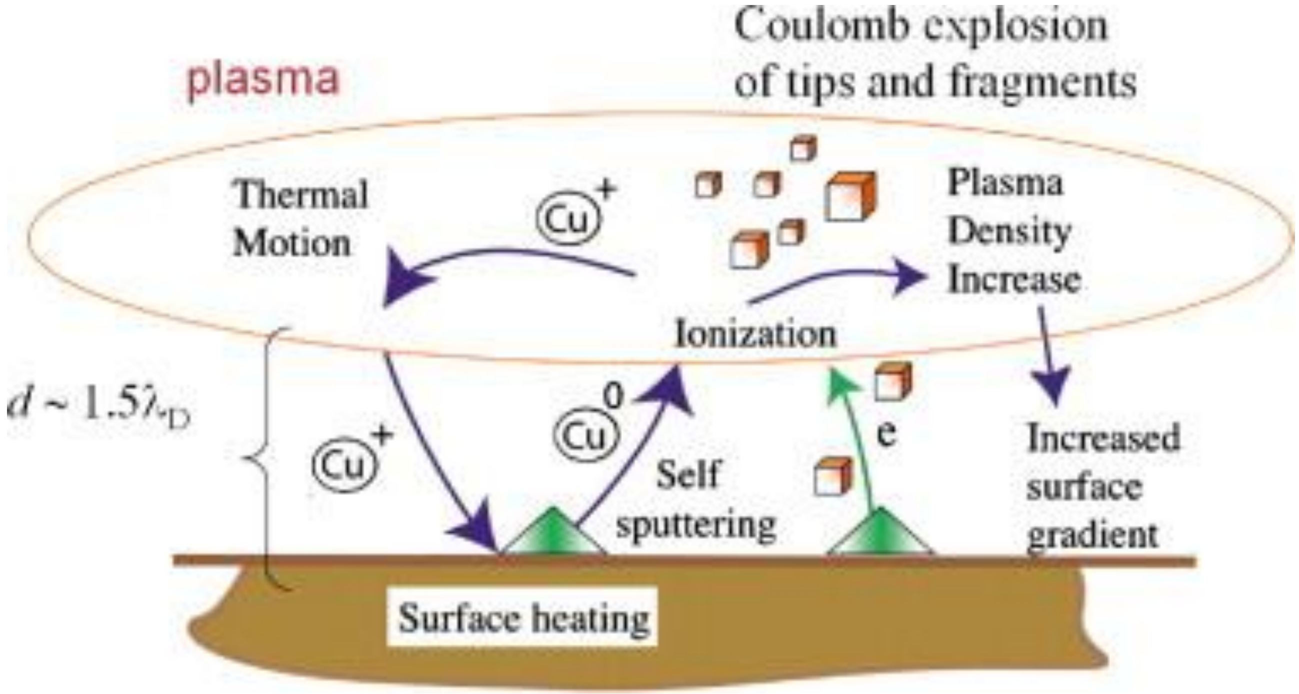
- Vacuum arc discharge
- Plasma processing
- Spark gap devices
- Gas switches
- Ion and neutral beams

Our applications generally share the following requirements:

- Kinetic description to capture non-equilibrium or non-neutral features, including sheaths, particle beams, and transients.
- Collisions/chemistry, including ionization for arcs. Neutrals are important.
- Very large variations in number densities over time and space.
- Real applications with complex geometry.

Vacuum Arc Initiation

- The physical process of vacuum arc initiation can be quite complex!
- Different features may be more or less important in a particular application.
- Our models can’t resolve everything, but need equivalent mechanisms.

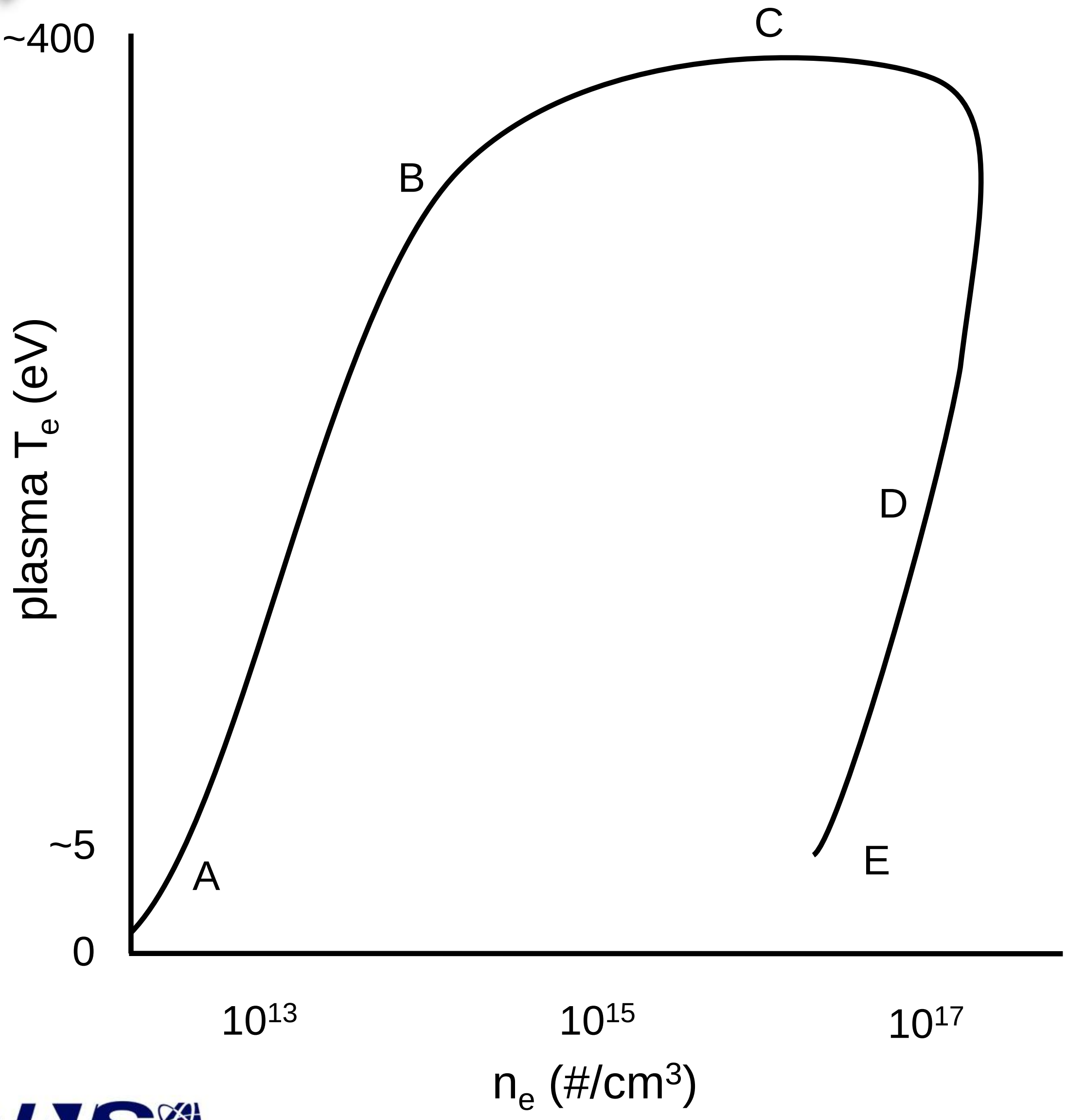


Z. Insepov, J. Norem, S. Veitzer, Atomistic self-sputtering mechanisms of rf breakdown in high-gradient linacs, Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms, Volume 268, Issue 6, 15 March 2010, Pages 642-650.

Challenges for Numerical Simulation

- Over the evolution of a vacuum arc, the plasma will grow in spatial extent from atom-scale collisions to device scales (cm), and will increase from vacuum to “high” density ($>10^{16}$ cm^{-3}).
- Particles initially exhibit essentially ballistic motion in the applied fields, but eventually the plasma density becomes high enough that collisions (including ionization) become important and sheaths begin to isolate quasi-neutral regions from the external fields.
- The model must capture the full dynamic range in plasma properties, which is accomplished by choosing appropriate spatial grid size and time step size. At the same time, the grid size and time steps must be chosen so that the simulation remains computationally tractable.

Plasma Properties Through Breakdown



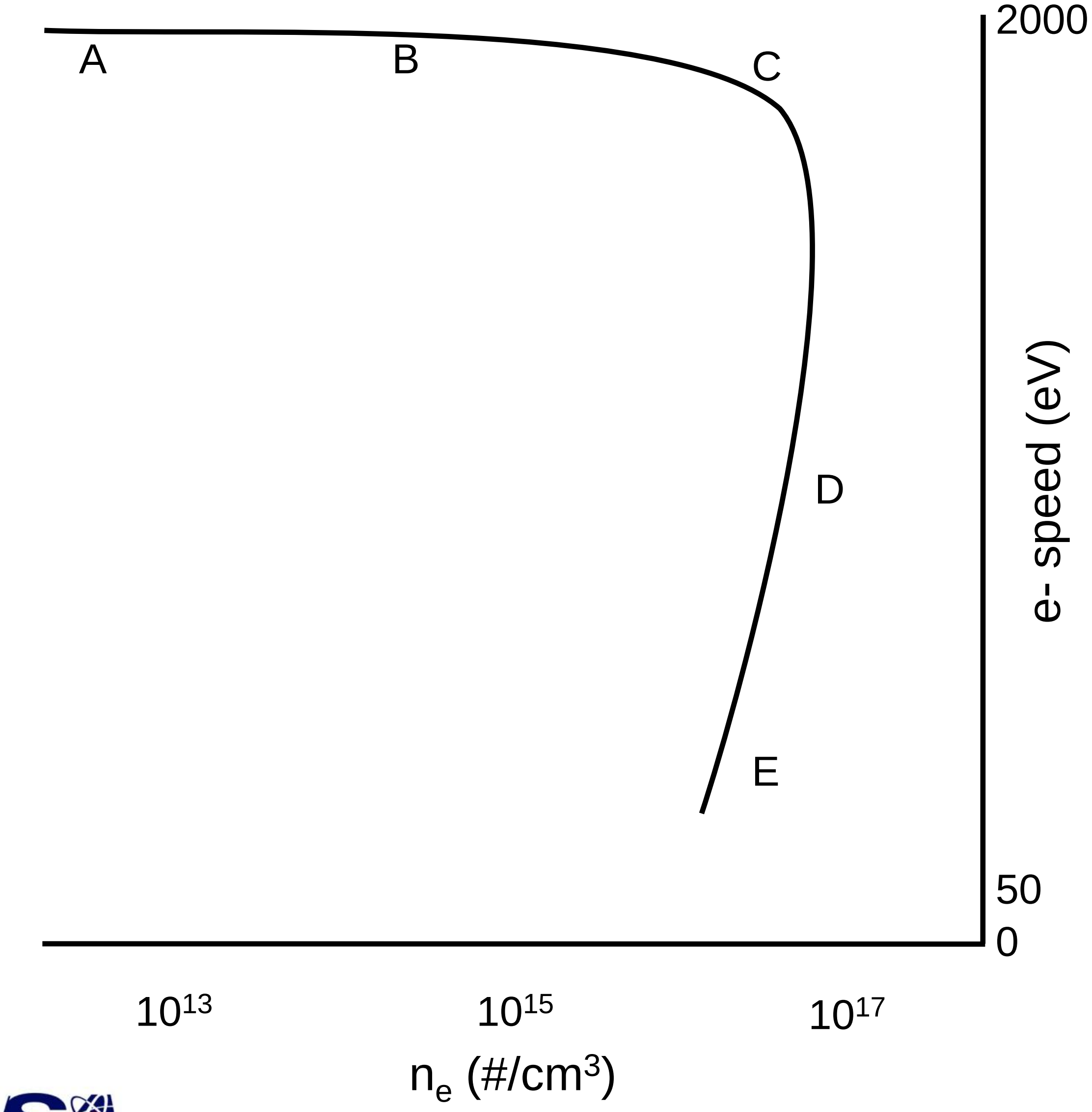
A: Initial injection of e-
(no plasma yet)
B: Cathode plasma grows
C: Breakdown
D: Relax to steady operation
(ΔV drops to ~50V)
E: Steady operation
(ΔV ~50V, I ~100A)

Model parameters:

$\Delta x \sim \lambda_D \sim (T_e/n_e)^{1/2}$

$\Delta t \sim \omega_p^{-1} \sim n_e^{-1/2}$

Electron Speeds Through Breakdown



A: Initial injection of e-
(no plasma yet)

B: Cathode plasma grows

C: Breakdown

D: Relax to steady operation
(ΔV drops to ~50V)

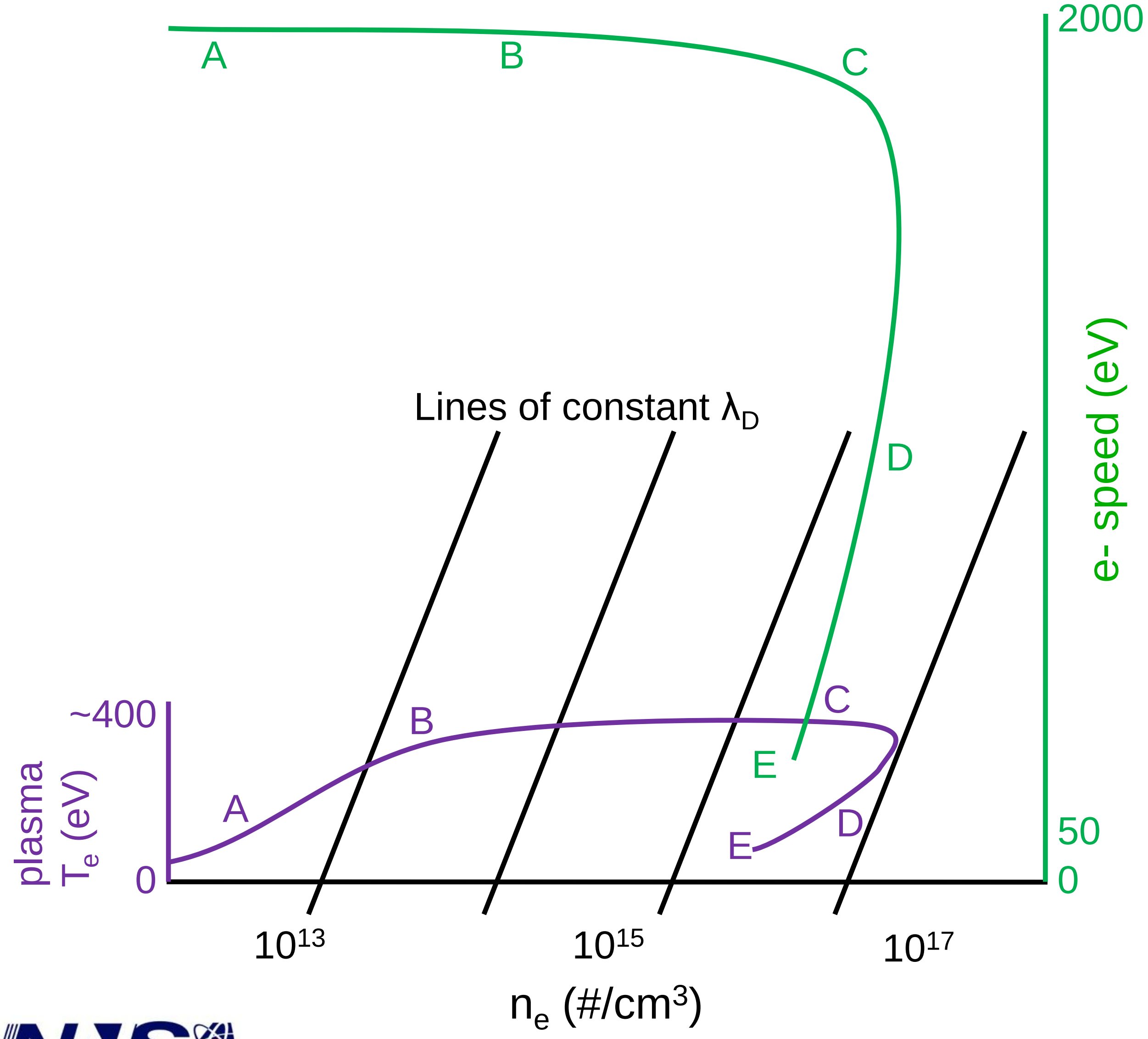
E: Steady operation
(ΔV ~50V, I ~100A)

Model parameters:

$\Delta x \sim \lambda_D \sim (T_e/n_e)^{1/2}$

$\Delta t \sim \Delta x/v_e$ (CFL)

Competing Criteria on Model Parameters



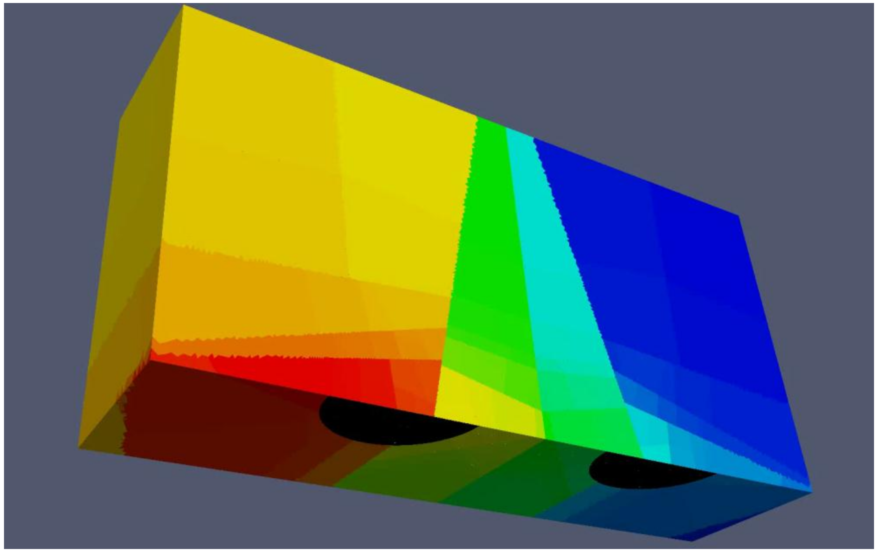
Peak density gives Δx:
$\Delta x \sim \lambda_D \sim (T_e/n_e)^{1/2}$
$\Delta x(n_e = 10^{16}\ \mathrm{cm}^{-3}) \sim 0.25\ \mu\mathrm{m}$

ω_p-based Δt from peak density:
$\Delta t(n_e = 10^{16}\ \mathrm{cm}^{-3}) \sim 10$ ps

CFL-based Δt from peak voltage:
$\Delta t(V = 2000\mathrm{V}) \sim 3$ps

→ CFL-based Δt dominates unless accel voltage <250V

Description of Aleph

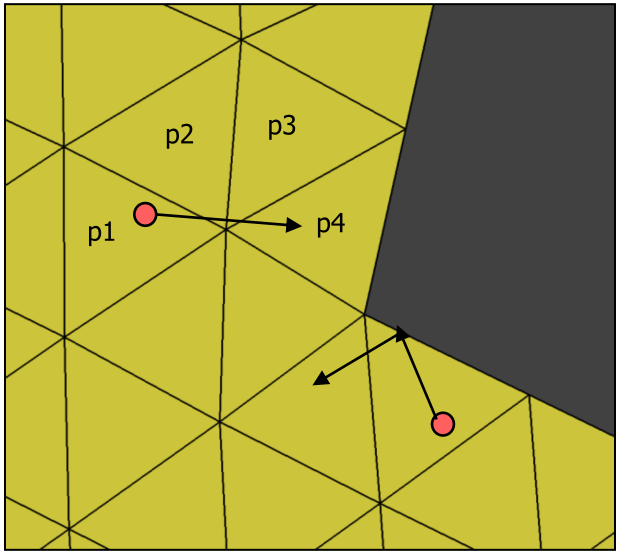

256 core particle-driven load balancing example

- Hybrid PIC + DSMC (PIC-MCC)
- Electrostatics or fixed magnetic field
- 1, 2, or 3D Cartesian unstructured FEM
- Dual mesh (Particle and Electrostatics)
- Conductive and dielectric boundaries
- Fully kinetic or fluid e- approximations (quasi-neutral, ambipolar, Boltzmann)
- Advanced surface physics models
- Surface charging and conduction
- Collisions, chemistry, charge exchange, excited states, ionization
- Advanced particle reweighting methods
- Restart with all particles
- Massively parallel, up to 64K processors (>1B elements), dynamic load balancing
- Agile software infrastructure for easily extending BCs, post-processed quantities, etc.

Simplified Aleph Iteration Cycle

Basic algorithm for one time step:

1. Move each particle ½ Δt with initial electrostatic force.
2. Detect and resolve intersections (non-trivial in parallel).
3. Transfer charge from particles to electrostatic mesh.
4. Solve for electrostatic potential.
5. Transfer electric fields from mesh to particles.
6. Move each particle for ½ Δt with updated force.
7. Perform DSMC collisions, chemistry, and ionizations:
    - determine expected number of reactions
    - sample a pair of particles in an element
    - determine cross section and probability of collision
    - Roll a digital die to determine if they collide
8. Reweight particles to maintain desired number per cell.
9. Compute output, post-processing, and other quantities.
10. Rebalance particle mesh (variety of determination methods).



Techniques to Address Challenges

The most pressing challenges for vacuum arc initiation are dealing with the orders-of-magnitude increase in density and the long heating and evaporation phase. We address these issues with several methods.

- Dynamic particle reweighting to handle increase of density. Simulation particle weight is automatically adjusted to keep target number of particles per cell within desired range.
- Hierarchical or implicit time-stepping methods speed up the heating phase.
  - Hierarchical method separates particle motion according to physical time scales. Electrons move on small dt, ions move on larger dt.
  - Implicit method relaxes stability constraints on time step size.
- Particle-Particle Particle-Mesh (P3M) method helps with both space and time requirements. A more accurate force calculation allows larger cells and longer time steps to be used throughout the simulation.

We will discuss the dynamic particle weighting and hierarchical time-stepping method today, and leave the others until they are fully implemented and exercised in our simulations.

Dynamic Particle Reweighting

- Maintain particle velocity distribution function (vdf) to the extent possible. Don’t use grid-based methods or assume the functional form of the vdf to “resample” particle velocities.
- Minimize energy discrepancy when it cannot be avoided.
- All other PIC/DSMC methods adapted to handle variably weighted particles – every particle can have a different weight.

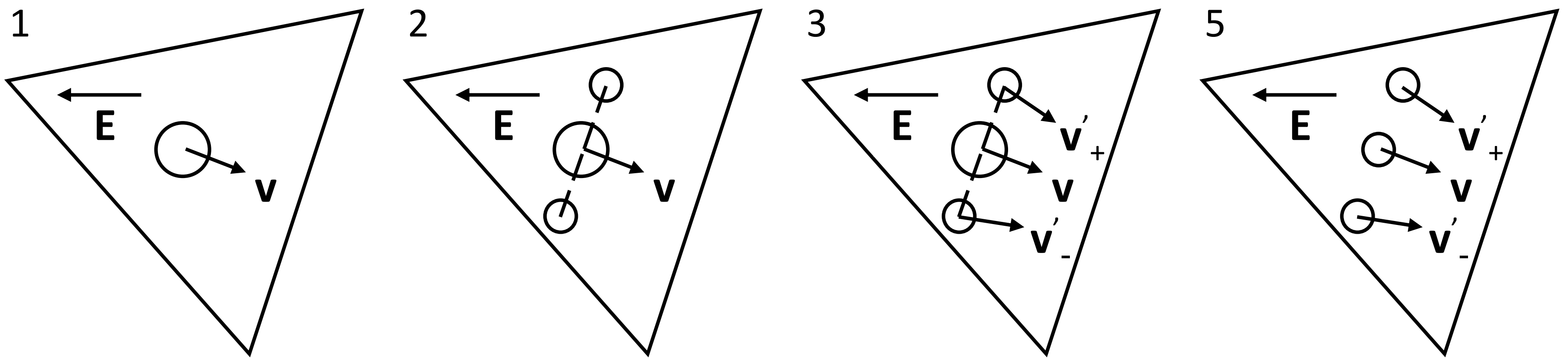
Basic idea:

```
for each cell,
    for each particle type $S$,
        let $N_s$ = # of particles of type $S$
        if $N_s < N_{S,low}$, clone more $S$ particles
        if $N_s > N_{S,high}$, merge some $S$ particles
```

Cloning Technique

1. Choose a high weight parent particle.
2. Generate a pair of random positions in the element, displaced symmetrically about the parent position.
3. Compute modified velocities at the new positions by accounting for displacement in the potential field.
4. If nonphysical velocities result, repeat 2-3.
5. Adjust weights for parent and new particles.

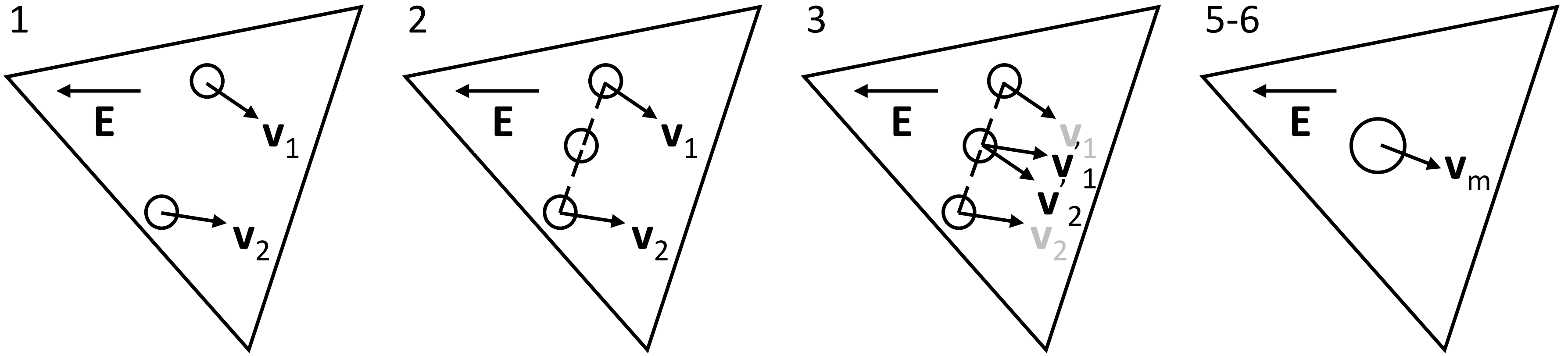
Repeat 1-5 until target number or limiter is met.



Merging Technique

1. Choose a random pair of particles.
2. Compute center of mass position.
3. Compute modified velocities at the center of mass by accounting for displacement in the potential field.
4. If velocities are "too different," reject pair and repeat 1-3.
5. Calculate average velocity, conserving momentum.
6. Adjust weight and record difference in kinetic energy (lost thermal part).

Repeat 1-6 until target number or limiter is met.



Merging Criteria

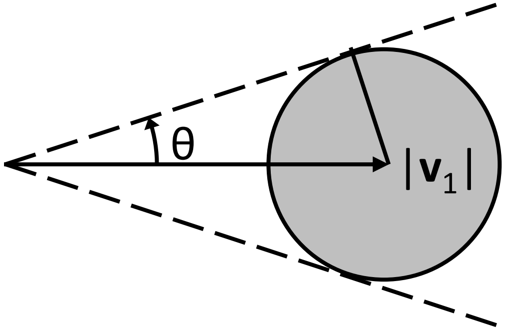
Only approve merge pairs that are close in both position and velocity.

- The spatial bin is the element, approves any pair.
- The velocity bin has many options. We use velocity interval, since it is easy to compute and adjusts based on local temperature.

Much faster to sort particles in element by speed, then choose one at random and check neighbors for valid merge partner.

Velocity Sphere

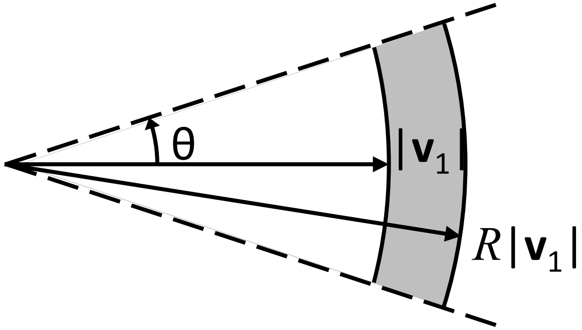
$$|\mathbf{v}_2 - \mathbf{v}_1| < |\mathbf{v}_1| \sin(\theta)$$



Velocity Proportion

$$\mathbf{v}_1 \cdot \mathbf{v}_2 > |\mathbf{v}_1||\mathbf{v}_2| \cos(\theta)$$

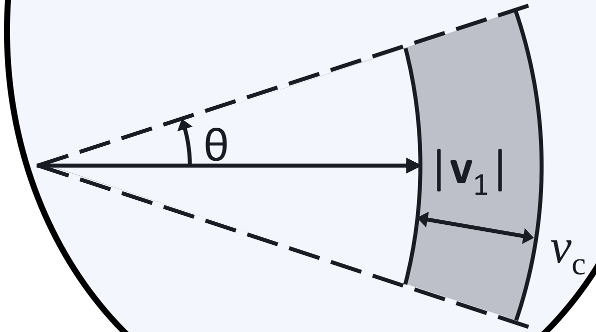
$$|\mathbf{v}_2| < R|\mathbf{v}_1|$$



Velocity Interval

$$\mathbf{v}_1 \cdot \mathbf{v}_2 > |\mathbf{v}_1||\mathbf{v}_2| \cos(\theta)$$

$$|\mathbf{v}_2| - |\mathbf{v}_1| < v_c = \alpha \sqrt{k_B T / m}$$



Example: Slowly Growing Xenon Sheath

Injection

$n_{Xe+} = n_e = 10^{10}/\text{cm}^3$ to $10^{12}/\text{cm}^3$ over 20 transit times

$v_D = 3$ cm/μs

$T_e = 1$eV

$T_{Xe+} = 300$K

$V = 0$V

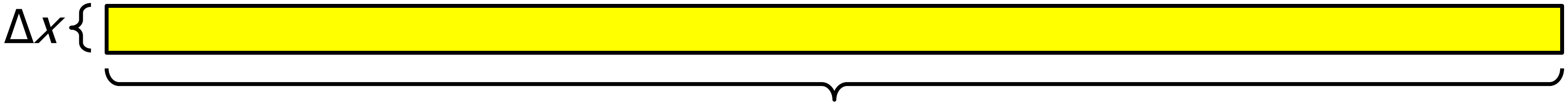
Side walls

$dV/dn = 0$

specular

Wall

$V = -5$V



$300\Delta x = (10 \text{ to } 100)\lambda_D$

Simulation and plasma parameters

$\Delta x = 2.5 \times 10^{-4}$ cm

$\Delta t = 20$ ps

$v_{\text{Bohm}} = 0.086$ cm/μs

$\lambda_D = 7.4 \times 10^{-3}$ cm to 7.4×10^{-4} cm

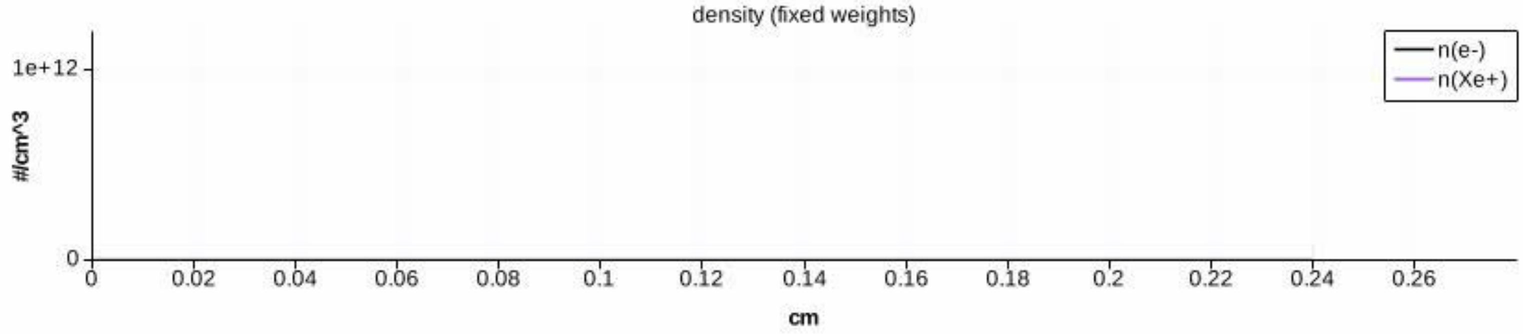
$\lambda_D / \Delta x = 30$ to 3

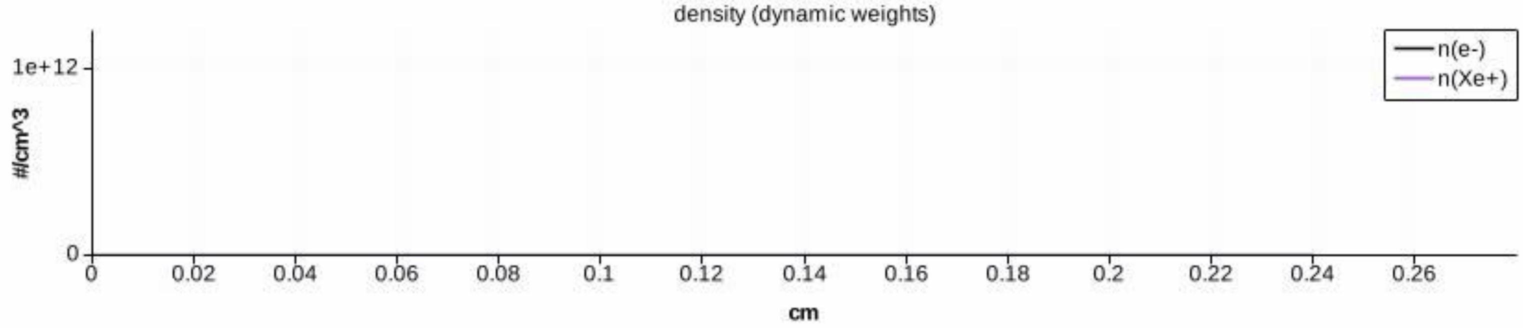
$\omega_p \Delta t = 0.11$ to 1.1

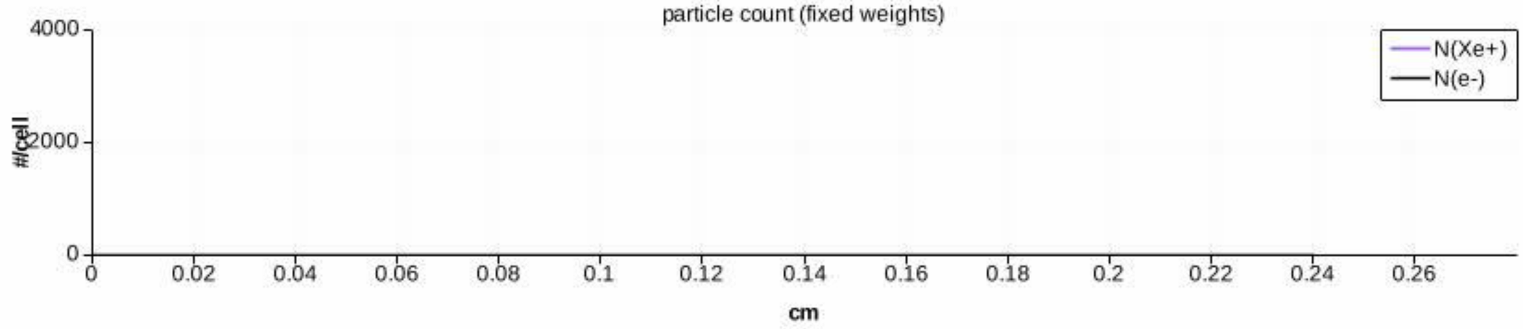
Two solutions:

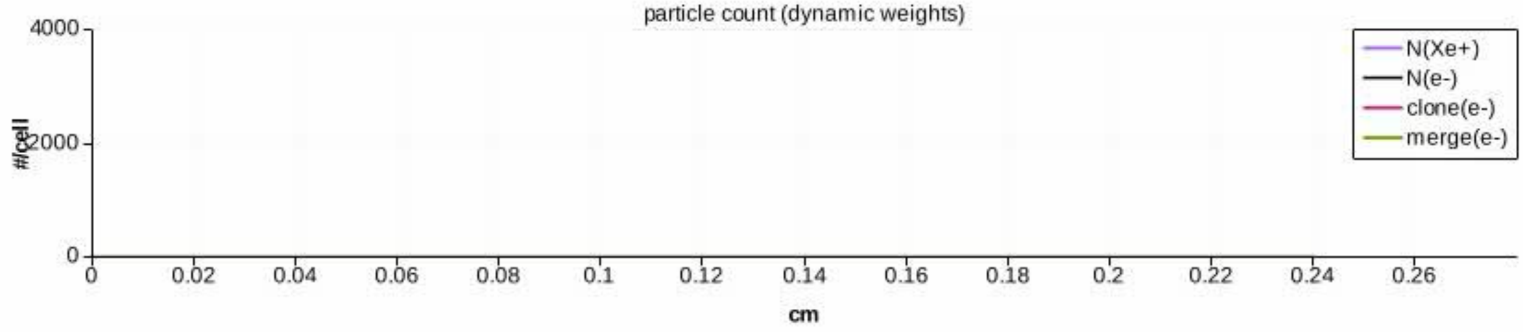
- Fixed particle weight
- Dynamic particle weight (Merge + Clone)

Small weight vs. large weight vs. requirements...

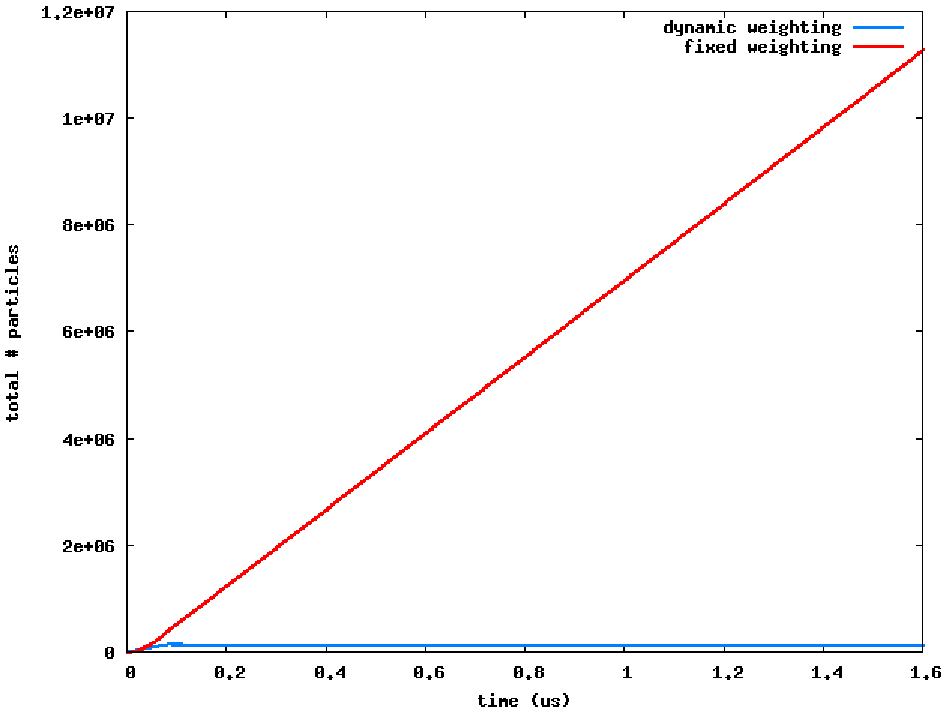

density (fixed weights)

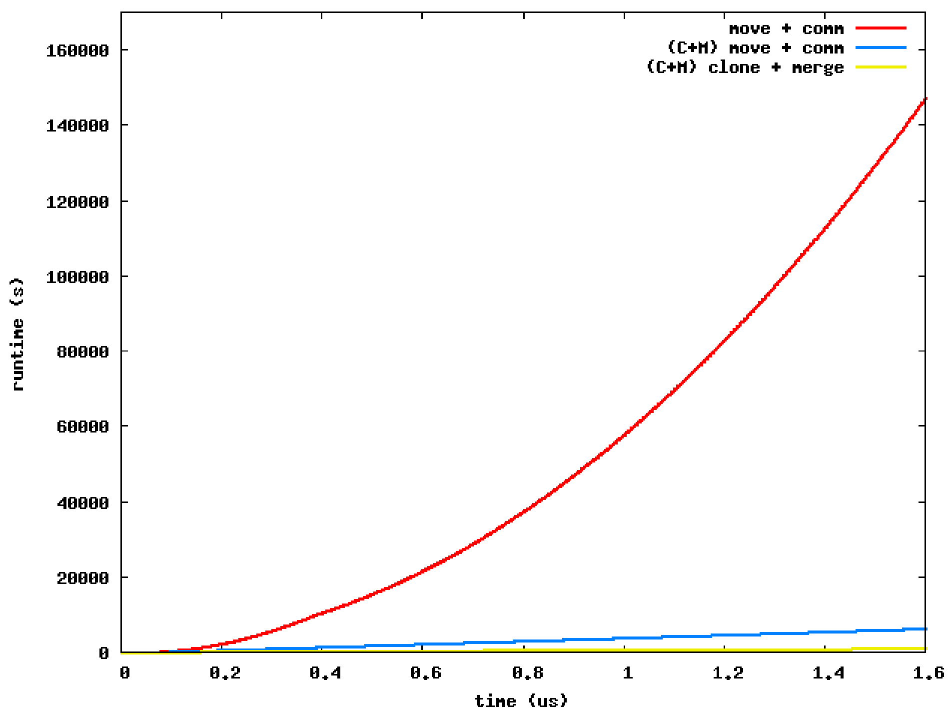

density (dynamic weights)


particle count (fixed weights)


particle count (dynamic weights)

Performance Impact





- Without reweighting, grows to ~110M particles and takes 41 hours.
- With reweighting, stabilizes at ~150k particles and takes just 2 hours.
- Reweighting achieves a 20x speed up, or 95% runtime savings.
- Despite logarithmic increase in density, reweighting maintains close to linear relationship of elapsed wall time to simulation time: constant load.

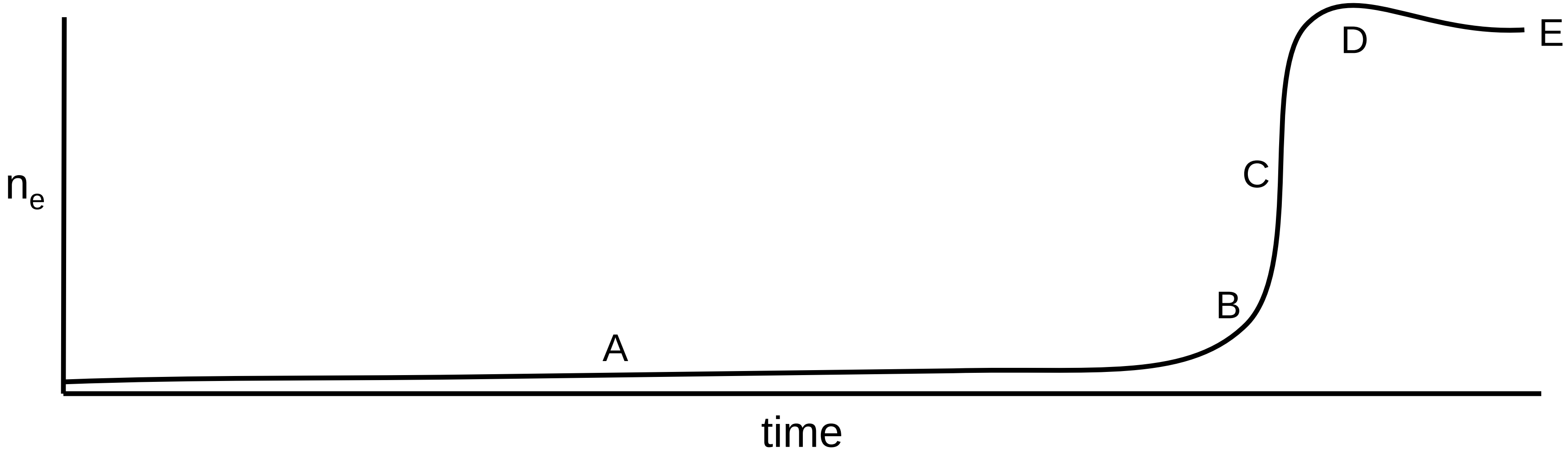
Dynamic Particle Reweighting Summary

- Dynamic particle reweighting can provide significant runtime savings.
- As with every other model/algorithm, one size does *not* fit all!
  - Good target: transient, growing simulations where accuracy is required at all timescales.
  - Bad target: simulations with essentially fixed densities.
- Current headache: varying element sizes

Future Work

- Allow [N_{low}, N_{high}] to vary by location, time, element size, collisionality, or other state parameters.
- Identify a good problem where cloning does more than just provide smooth output. E.g., reaction system based on trace species. Cloning will provide more particles of the trace species for less noise in the reaction rate.

Hierarchical Time Stepping



Slow heating and neutral expansion during phase A, few ionizations occur.

- Why solve everything during stage A, especially at the tiny e- time step?
- Solve a series of quasi-static stages until we approach B:
  - Evolve ions at $\Delta t_{ion} = N \times \Delta t_{e-}$.
  - Find quasi-static e- solution (10 steps at Δt_{e-} from last e- solution).
  - Ionize for Δt_{ion} timestep.
  - Neutrals can move at N’ x Δt_{ion}, but referenced to Δt_{e-} for now.

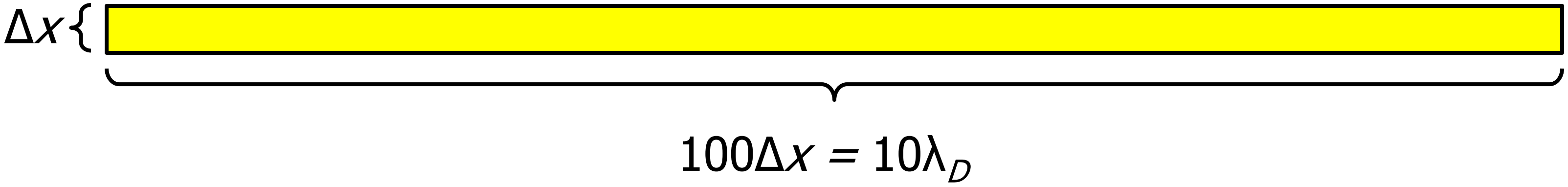
  ... continue until there are “significant” fields, i.e., plasma (stage B).

Example: Steady Xenon Sheath

Injection
$n_{Xe+} = n_e = 1.1 \times 10^8$ /cm^3
$v_D = 0.24$ cm/μs
$T_e = 1$eV
$T_{Xe+} = 300$K
$V = 0$V

Side walls
$dV/dn = 0$
specular

Wall
$V = -5$V



Δx

$100\Delta x = 10\lambda_D$

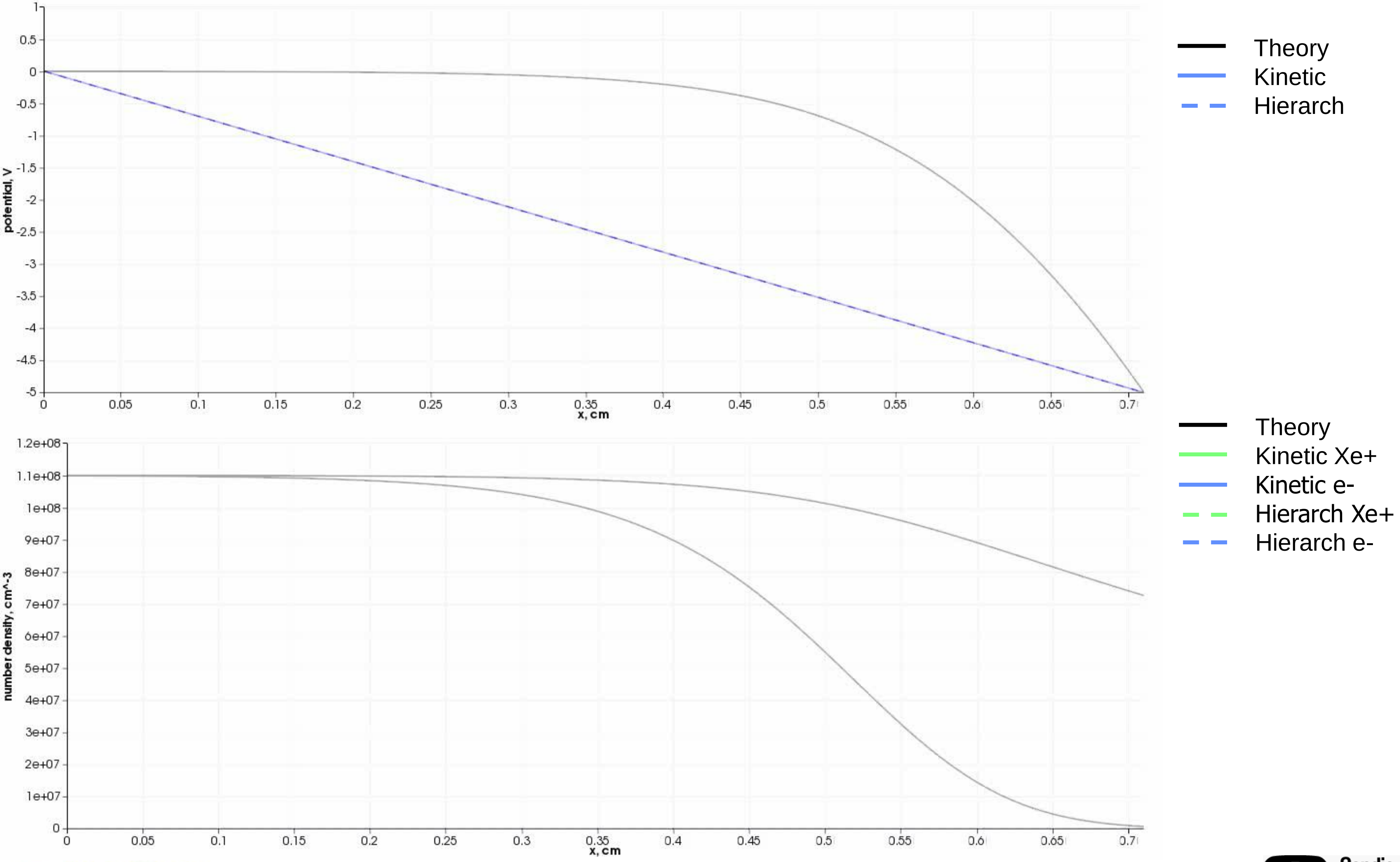
Simulation and plasma parameters
$\Delta x = 7.1 \times 10^{-3}$ cm
$\Delta t = 25$ ps
$v_{\mathrm{Bohm}} = 0.086$ cm/μs
$\lambda_D = 7.1 \times 10^{-2}$ cm
$\lambda_D / \Delta x = 10$
$\omega_p \Delta t = 3.0 \times 10^{-5}$

Two solutions:

- Kinetic time, $\Delta t_{Xe+} = \Delta t_{e-}$
- Hierarchical time, $\Delta t_{Xe+} = 5\ \Delta t_{e-}$

Solutions displayed at matching ion times.

Hierarchical Timestepping on Steady Sheath

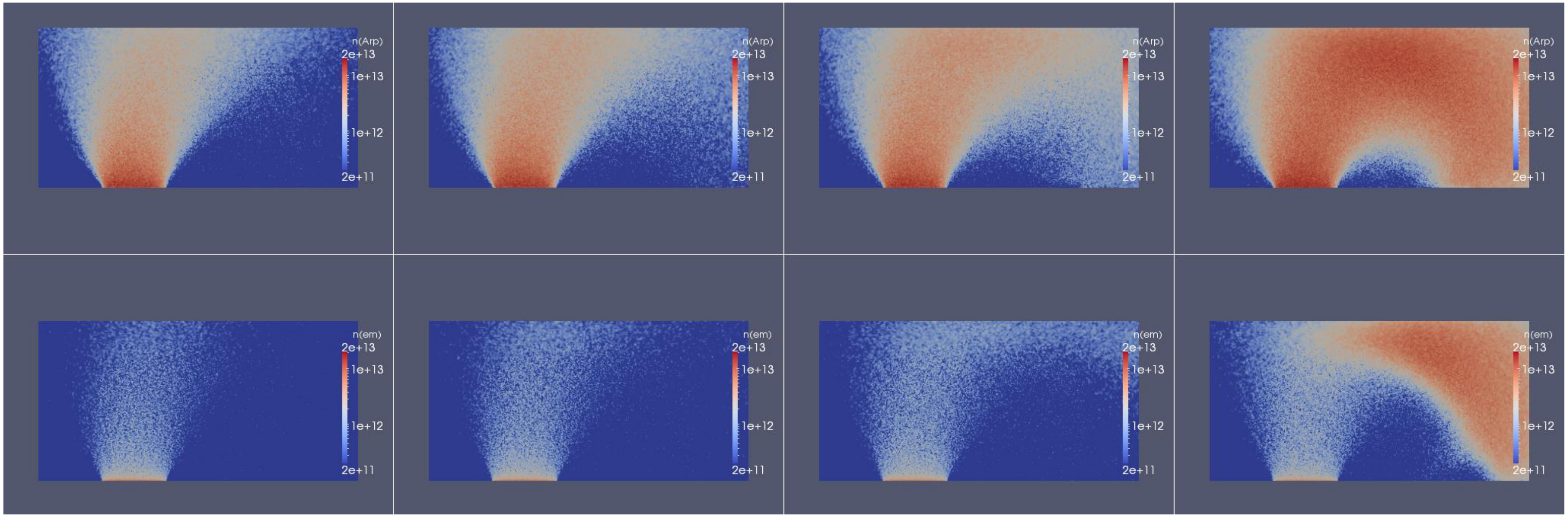


Comments on Hierarchical Time Stepping

Performance Impact

- Using kinetic time, converged to 53,800 Xe+ and 30,800 e-, after 1:32.
- Using hierarchy time, converged to 53,600 Xe+ and 30,900, after 0:17.
- Hierarchical time stepping achieves 5.5x speed up, or 82% time savings.

Limitation: Need to keep time factor small (N<10) for “physical” solution.



N=1 N=3 N=5 N=10

Electron fountain ionizing argon at 1 torr, 300 K, using different time factors. N = 10 is clearly too large.

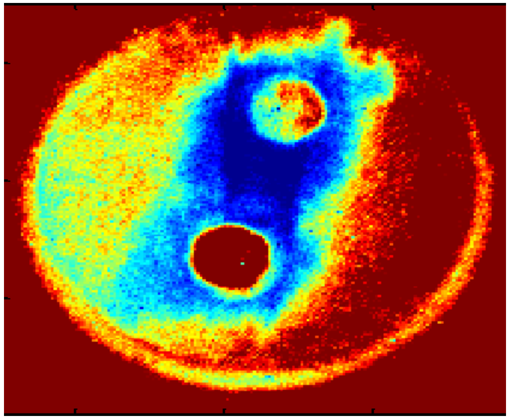
Hierarchical Time Stepping Summary

- Hierarchical time stepping can provide significant runtime savings.
- Useful for generating physically plausible, complicated initial conditions.
- One size does *not* fit all:
  - Good target: quasi-steady evolution with weak coupling between fast- and slow- time scale phenomenon.
  - Bad target: rapidly evolving or periodic conditions.

Future Work

- Automate time factor selection based on a transport or collision property.
- Develop a smarter (automatic) way to identify when plasma fields have changed enough to switch between one time scale to the next.
- Go to full implicit time stepping?

Model Cu-Cu Arc System



1 Torr argon background

Copper electrodes (this picture is Cu-Ti)

- 1.5 mm center to center distance
- 0.75 mm diameter electrodes

20Ω resistor in series

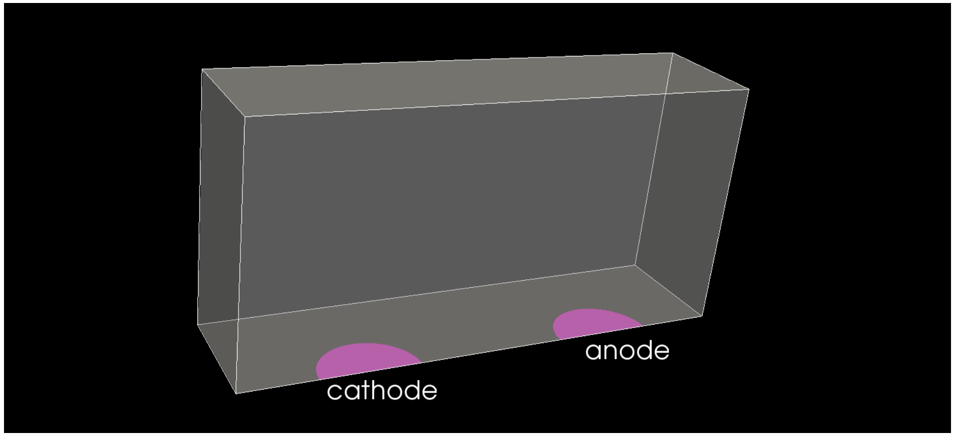
2 kV drop across electrodes

Steady conditions around 50V, 100A

Breakdown time << 100ns

Ionization mfp = 1.5 mm at maximum σ

→ $n_i \sim 10^{16} – 10^{17}$ #/cm^3



Work Estimate for Coplanar Arc

Assume a rate of 1x10^{-6} s wall time per particle per timestep (s/part/dt).

vol_{Domain} = 4 mm x 2 mm x 0.75 mm = 6 mm^3
\+ Δx = 1/5 $\lambda_D(n_e = 10^{16}\ \text{cm}^{-3})$ = 5 x 10^{-3} mm
\+ $vol_{Element}$ = $\Delta x^3/6$ = 2 x 10^{-8} mm^3

~ 300M elements or 30B particles (5 types with 20 particles per cell)

10 ns breakdown time at 10 fs Δt, 1M timesteps
\+ 100 ns evolution time at 100 fs Δt, 1M timesteps

2M timesteps

time = 1x10^{-6} s/part/dt x 30B parts x 2M dt = 60B s, or
16,000 cores for 1,000 hours!

Instead using 1/10th scale domain (1/1000th of the work) and even going to 2D for exploratory simulations. Then 256 cores for ~60 hours can get useful results.

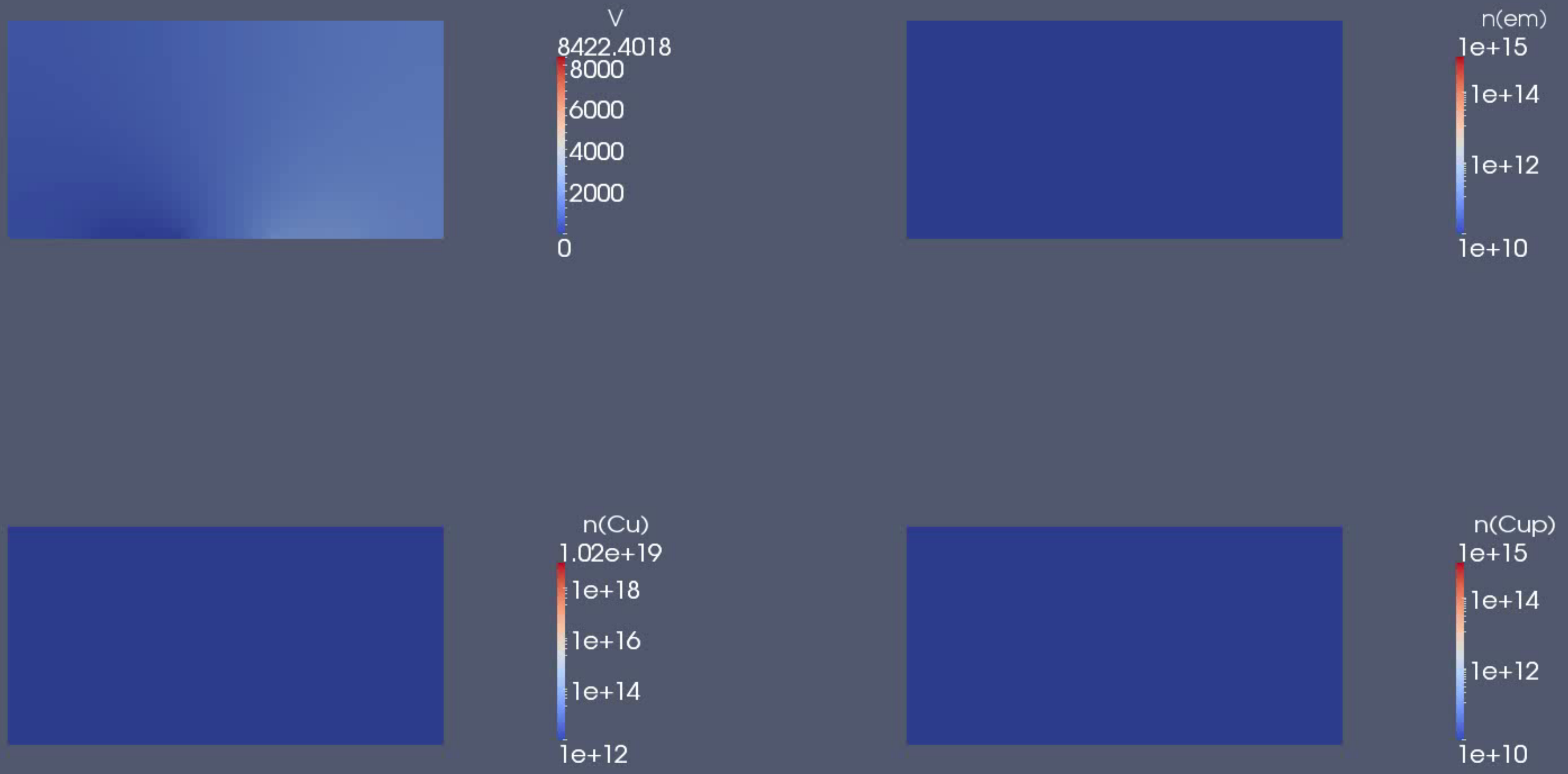
Representative Arc Simulations

- "Initial" results show early evaporation and neutral expansion.
  - 2D on 1/10th scale domain (110k elements)
  - fully resolved in Δx and dt
  - 80+ hours on 16 cores

- "Intermediate" results show quasi-steady evaporation and expansion.
  - 3D on 1/10th scale domain (11M elements)
  - fully resolved in Δx and dt
  - 180+ hours on 256 cores.

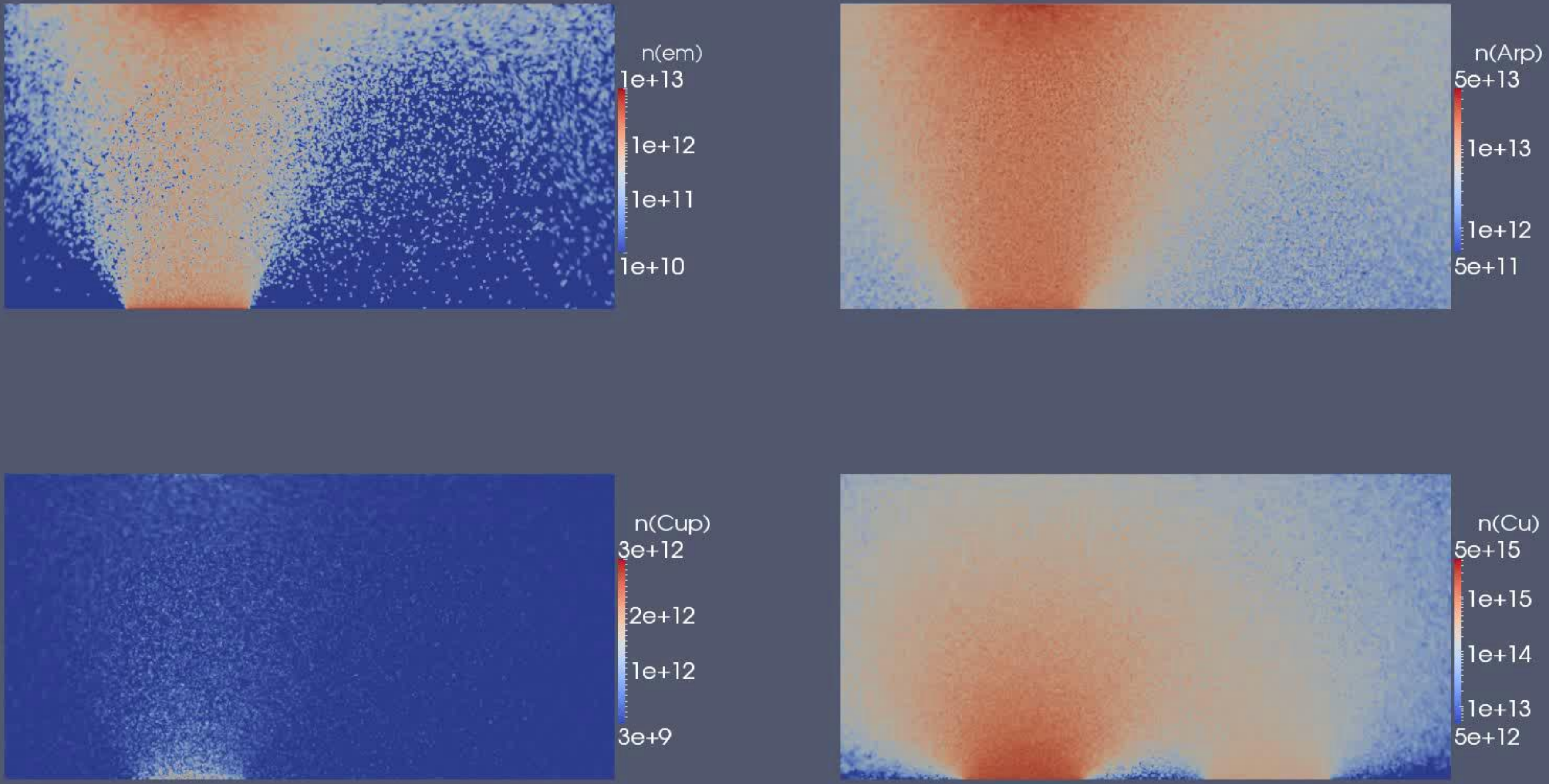
- "Late" results show representative "moment of breakdown."
  - 3D on 1/10th domain size
  - Under-resolved: 2x Δx and 10x dt, only unstable near the end
  - starting with background of 10^{18} cm^{-3} neutral copper
  - <96 hours on 256 cores

Initial Heating and Neutral Emission Phase



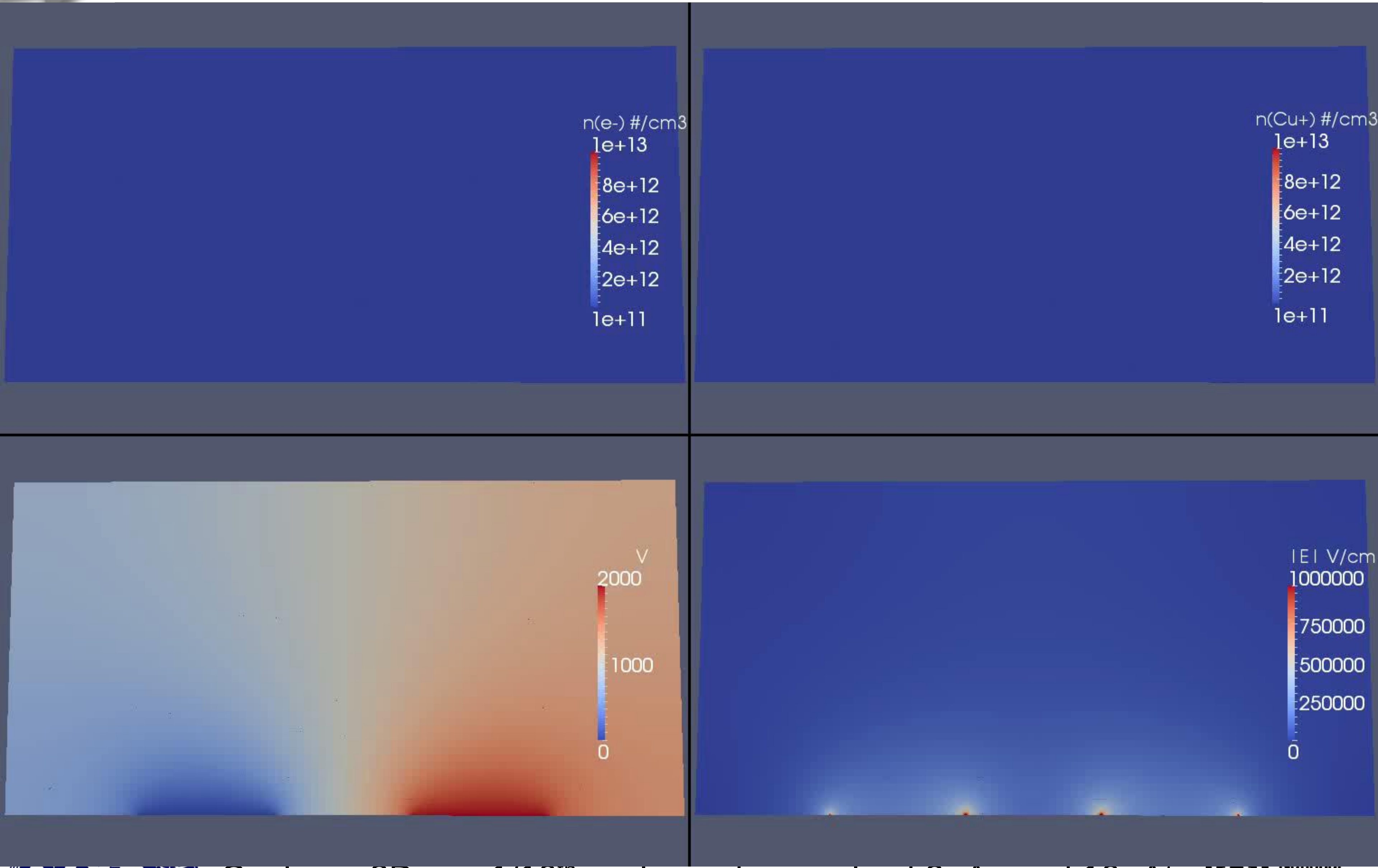
Coplanar 2D arc, 1/10th scale, properly resolved Δx and Δt.

Intermediate Heating and Neutral Emission



Coplanar 3D arc, 1/10th scale, properly resolved Δx and Δt.

Late Ionization and Breakdown Phase



Coplanar 3D arc, 1/10th scale, under-resolved 2x Δx and 10x Δt.

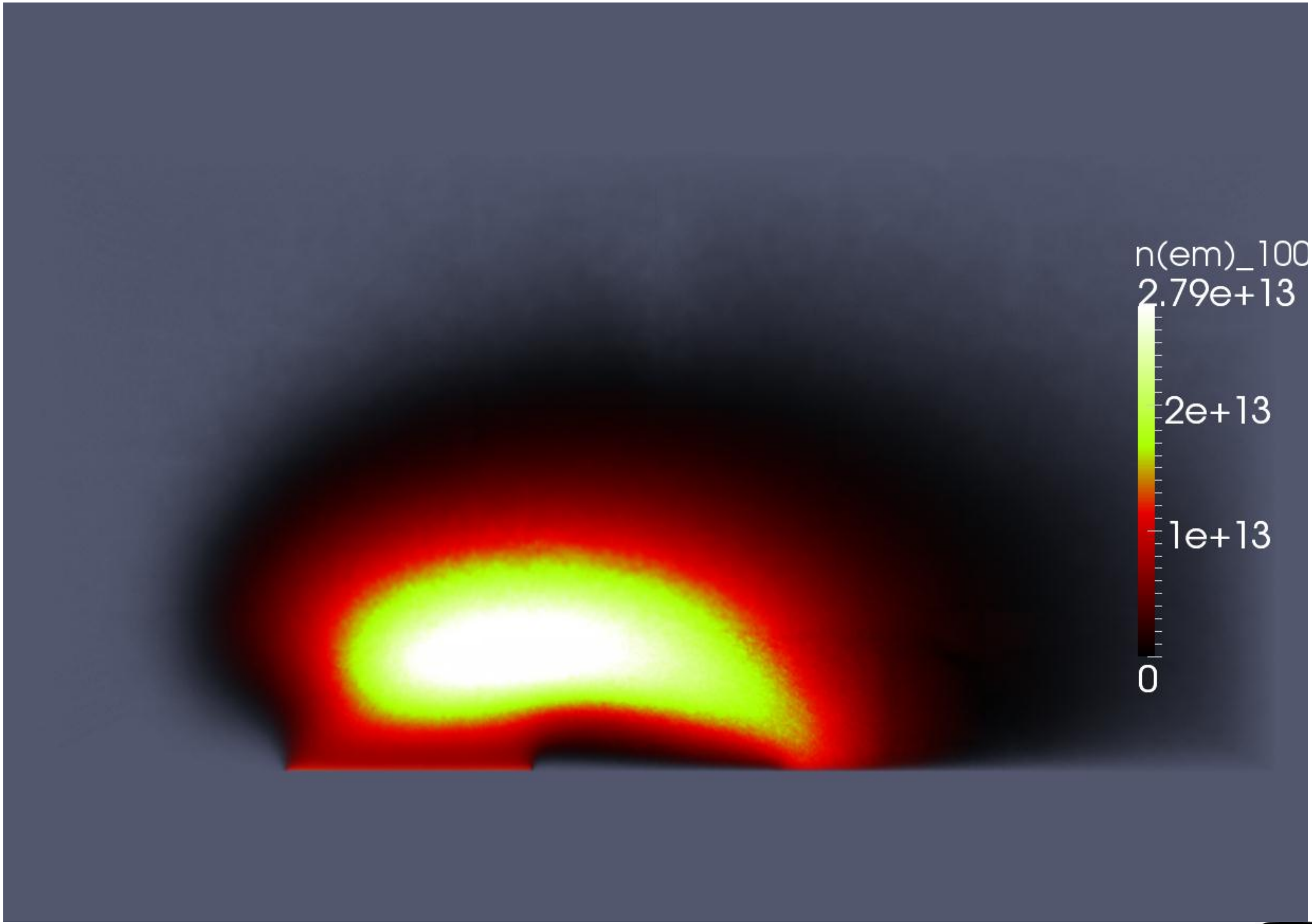
Vacuum Arc Modeling Summary

- EXPENSIVE in 3D (but we expected that).
- Gaining more experience with breakdown in 2D and 3D.
- Mesh sizes are now a big bottleneck.
- Currently scaling OK on 10K's of cores.
- Surface models (e.g., secondary yields) are necessary but not sufficient for real physical systems.

Future Work

- Separate "collision mesh" according to different particle interactions. Is this always a patch of PIC cells?
- Automate hierarchical time stepping to move between different time scales and detect when full time resolution is required.
- Hybrid approaches compatible with physically accurate arc simulation.
- Scale to 100K+ cores.

Thank You



CV & SV & V & SA & UQ

All Interesting Arc/Plasma Behavior Is Nonlinear And Coupled – How Can We Be Confident In Our Predictions?

CV: Code Verification. Necessary, woefully insufficient. Can test single simple capabilit

SV: Solution the right solution

V: Validatio ality. Ideally, tion verificat

SA: Sensitiv ermine which numerical/physical parameters impact the prediction, experimental result, and/or validation comparison. Identifies problem areas and is a source of planning decisions/efficiency.

UQ: Uncertainty Quantification. Estimate uncertainty in a code prediction, usually without experimental comparator. Incorporates error estimation and quantified code prediction uncertainties.

ALL OF THIS IS MORE COMPLICATED BECAUSE OUR BASIC MODELING METHODS ARE STOCHASTIC (PIC, MCC, MD, ...) AND DO NOT HAVE TYPICAL "GRID CONVERGENCE" BEHAVIOR