3D Unstructured PIC-DSMC Simulation: Challenges and Examples

Matthew M. Hopkins, Sandia National Laboratories

International Online Plasma Science Seminar
June 3, 2021

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.
SAND2021-6707 PE

Introduction

Guidance for this presentation was one of a tutorial or review nature.

What do we do that is “special”:

> Large scale PIC-DSMC on 3D unstructured meshes
> (and lots of other detailed models: surface models, photonic processes, …)

The goal of this talk is to give an introduction to large-scale, 3D, unstructured mesh, PIC-DSMC simulations, and an overview of some of the challenges.

The target audience is a plasma physics non-expert or graduate student interested in computational modeling of low temperature plasmas.

Motivation

Sandia National Laboratories is one of the National Nuclear Security Administration laboratories in the US Department of Energy (as are Los Alamos and Lawrence Livermore).

Our national security mission requires understanding (modeling) many different kinds of low temperature plasma systems:

- Vacuum arcs for failure and operation (high voltage electronics, insulator flashover, switches, space)
- Low pressure discharges (plasma processing, high altitude)
- Atmospheric pressure discharges (high energy arcing faults, lightning, switches)

Not all work is public, which includes a lot of great complex, technical work.

Annual budget: $3.9B
Approximately 50% NNSA weapons
14,600 employees in 2021

Motivation

Sometimes fundamental physics investigations in simplified systems are sufficient, but other times we need to simulate behavior in full 3D.

Considering vacuum arc expansion in 3D, *Debye lengths can vary over many orders of magnitude*.

Regular Cartesian meshes (or “outer product” compositions) are not sufficient.

One alternative approach is unstructured meshes. Other alternatives include adaptive mesh refinement (AMR) of Cartesian meshes, or mesh-free methods (which often are not really mesh-free!). *All approaches have pros and cons.*

Problems can be extremely important: huge effort in verification and validation (V&V).

Outline

1. Introduction to the PIC(-DSMC) method
2. Steps to support unstructured mesh and large-scale/3D models
3. Examples

The Boltzmann Equation

The particle methods we use generate solutions to the Boltzmann equation,

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + F \cdot \nabla_v f = \left(\frac{\partial f}{\partial t}\right)_{coll}$$

where

f(x, v, t) = distribution function in phase space,
x = particle location,
v = particle velocity,
F = external applied force, and
$(\partial f/\partial t)_{coll}$ represents changes due to particle collisions.

For example, $n(x,t) = \int f(x,v,t)dv$.

In 3D the Boltzmann equation is 7-dimensional (!).

The Boltzmann Equation

We discretize the Boltzmann equation in space and time.

We discretize the spatial portion of *(x, v)* phase space by employing a mesh.

We discretize in time by using a time integration method over discrete time steps t_n, t_{n+1}, …

This effectively reduces the problem to evolving the velocity distribution function (vdf) in each cell and over each time step:

$$f(x_i, t_n, v) \rightarrow f(x_i, t_{n+1}, v), \text{ or}$$
$$f_{cell,n}(v) \rightarrow f_{cell,n+1}(v).$$

We also sometimes use f as if it were an energy distribution function. And we often drop the explicit connection to the discretization.

(There are attempts to solve a full Boltzmann equation.)

The Boltzmann Equation

We approximate/discretize $f(v)$ in each cell by a discrete set of particles with individual velocities. Because the physical number of particles in a cell can be quite large, we will further approximate the vdf by assuming each computational particle (or notional particle) represents some number of real ones.

This real-to-computational particle ratio is referred to as the “macroparticle weight” or just “particle weight”, w_p.

The basic solution methodology advances a set of computational particles in a mesh from one discrete time to another accounting for particle motion, particle forces, and particle collisions.

Unlike continuum methods where densities, velocities, energies (temperatures), etc., are the primary “solution variables”, the primary “solution variables” in the kinetic methods we use are particle positions and particle velocities. *Everything else is derived from this.*

Particle-in-Cell (PIC)

PIC is focused on part of the Boltzmann equation,

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + F \cdot \nabla_v f = \left(\frac{\partial f}{\partial t}\right)_{coll}$$

An alternative derivation goes through the Klimontovich equation.

where PIC typically assumes collisionless particles (RHS = 0). Replacing *F* with electric and magnetic forces,

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + \frac{q}{m}(E + v \times B) \cdot \nabla_v f = 0$$

gives us the Vlasov equation with *q* the particle charge, *E* the electric field, and *B* the magnetic field. We consider the electrostatic (ES) case where we assume *B* = 0,

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + \frac{qE}{m} \cdot \nabla_v f = 0$$

and will couple to Poisson's equation, although there are many electromagnetic (EM) PIC codes that couple to Maxwell's equations and solve for a consistent *B*.

Particle-in-Cell (PIC)

In addition to integrating charged particle trajectories, we need to solve Poisson's equation,

$$\nabla(\varepsilon_0 \nabla V) = -\rho = q_e(n_{i,total} - n_e)$$

where ε_0 is the permittivity of free space, $n_{i,total}$ is total ion number density (written assuming only single ionizations for simplicity) and n_e is electron number density. We generally don't care about V directly but need to compute the electric field, $E = -\nabla V$.

There are many ways to solve Poisson's equation.

If using a Cartesian mesh with fixed spacing a finite difference method (FDM) is a great choice.

Much of the numerical analysis diversity of PIC methods involve representations of ρ, and how E is computed at particle locations, giving different interpolation/approximation orders. Higher order approximations generally require larger computational stencils.

Particle-in-Cell (PIC)

Because the Poisson equation is elliptic *the overall method is globally coupled* and requires solution of a global linear system. This has considerable impact on parallel implementations and performance.

It can also cause instantaneous “action-at-a-distance”. For finite perturbation speed you need to use an EM method.

The methodology described here is explicit in time. There are methods that are semi-implicit, and even fully implicit (with significant caveats).

Particle-in-Cell (PIC)

Basic ES PIC iteration to advance from time step *n* to *n+1* uses a time-splitting method:

1. Update particle velocities over *Δt/2* and positions with *Δt*,

$$v_i^{n+1/2} = v_i^n + \frac{q_i E^n(x_i^n)}{m_i}\frac{\Delta t}{2}$$
$$x_i^{n+1} = x_i^n + v_i^{n+1/2}\Delta t$$

2. Solve Poisson's equation to get new fields,

$$\nabla(\varepsilon_0 \nabla V^{n+1}) = -\rho = q_e(n_i^{n+1} - n_e^{n+1})$$
$$E^{n+1} = -\nabla V^{n+1}$$

3. Compute final update to velocities with new forces,

$$v_i^{n+1} = v_i^{n+1/2} + \frac{q_e E^{n+1}(x_i^{n+1})}{m_i}\frac{\Delta t}{2}$$

Particle-in-Cell (PIC)

Requirements/assumptions for employing ES PIC include:

1. Cell sizes must resolve Debye length λ_D to avoid numerical heating,

$$\Delta x < \lambda_D = \sqrt{\frac{k_B T_e \varepsilon_0}{n_e q_e^2}}$$

2. Time step must resolve plasma frequency ω_p,

$$\Delta t < \frac{2}{\omega_p} = 2\sqrt{\frac{\varepsilon_0 m_e}{n_e q_e^2}}$$

3. Should satisfy Courant-Friedrichs-Lewy (CFL) condition similar to continuum CFD,

$$\Delta t < \frac{\Delta x}{v_{max}}$$

Particle-in-Cell (PIC)

Requirements/assumptions for employing ES PIC include: (cont.)

4. Electrostatic solvers usually expect some resolution of $|\mathrm{grad}(V)|$ or $|\mathrm{grad}(V)|^2$. It is often unclear how to interpret this as there are combinations of quasi-neutral plasma, non-neutral regions, and high applied fields.

These constraints would ideally apply to the most extreme constraints (minimum λ_D, maximum ω_p, maximum v on minimum Δx), but because particle properties are stochastic this cannot be guaranteed. This is a recurring theme in kinetic particle methods.

Often, the thermal speed is used for v_{max}; caveat emptor!

Direct Simulation Monte Carlo (DSMC)

DSMC is focused on computing solutions to a different part of the Boltzmann equation,

$$\frac{\partial f}{\partial t} + v \cdot \nabla_x f + F \cdot \nabla_v f = \left(\frac{\partial f}{\partial t}\right)_{coll}$$

where DSMC typically assumes *F = 0* (no external forces).

DSMC is a *completely local* method. Only information within a computational cell is required. It is “embarrassingly parallelizable”. Not true for electrostatic PIC.

Within a single cell actual particle locations are assumed irrelevant; all particles in the cell are candidates to collide with all other particles in a cell.

Assume instantaneous binary collisions separate from motion.

The Monte Carlo Collision (MCC) method can be used when one of the reactant species is assumed fixed (or perhaps solved by a fluid method).

Direct Simulation Monte Carlo (DSMC)

Requirements/assumptions for employing no-time-counter DSMC include:

1. Cell size must resolve the collision mean free path λ_{mfp} (and other vdf gradient length scales),

$$\Delta x < \lambda_{mfp} = \frac{1}{n\sigma}$$

2. Time step must resolve collision frequency ν_c,

$$\Delta t < \nu_c^{-1} = \frac{\lambda_{mfp}}{v} = \frac{1}{n\sigma v}$$

These constraints would ideally apply to the most extreme constraints (minimum λ_{mfp} and maximum ν_c), but because particle properties are stochastic this cannot be guaranteed. This is a recurring theme in kinetic particle methods.

Particle Weights

Particle weights, w_p, determine the number of computational particles in a cell. Number of computational particles in a cell determines how well the vdf is resolved. Usually, particles within a collection of cells are aggregated for vdf analysis. The collection extent can be complicated. Different species can have different particle weights. Dynamic problems (e.g., discharge) require *dynamic particle weighting*.

For PIC, cells can have 0 particles, or 1 particle, and all is well.

For DSMC, to resolve collision rates, the rule of thumb for neutral species is 30 particles per cell. We will typically use:

background neutral species: 10
excited states and fast neutrals: 20
ions: 40
electrons: 200

Actual numbers are very problem dependent and should be checked for convergence issues. (Skipping long story about proper solution verification)

3D Particle Weights

For many problems, especially at high (atmospheric) pressure, $\Delta x \leq 1$ μm.

In 2D, this results in volumes < 10^{-12} m^3.

In 3D, this results in volumes < 10^{-18} m^3.

Using a particle weight of $w_p = 1$ means the "floor" for intensive quantities (e.g., number density) is quite high, i.e., minimum representable number density is 10^{18} m^{-3}.

For well-resolved vdf's, we may want 200+ particles/cell → particle weights in 3D can be << 1. What does this mean? Concerns about textbook/model "uniform background" vs. real-world background.

For discharge simulations, one "advantage" of a tiny particle weight is the exponential multiplicative effect is essentially guaranteed to begin at $t = 0$.

Circuit noise is also impacted with a lower particle weight.

PIC-DSMC Simulation Requirements

$$\Delta x < \lambda_D = \sqrt{\frac{k_B T_e \varepsilon_0}{n_e q_e^2}}$$

$$\Delta x < \lambda_{mfp} = \frac{1}{n\sigma}$$

$$\Delta t < \frac{2}{\omega_p} = 2\sqrt{\frac{\varepsilon_0 m_e}{n_e q_e^2}}$$

$$\Delta t < \frac{\Delta x}{v_{max}}$$

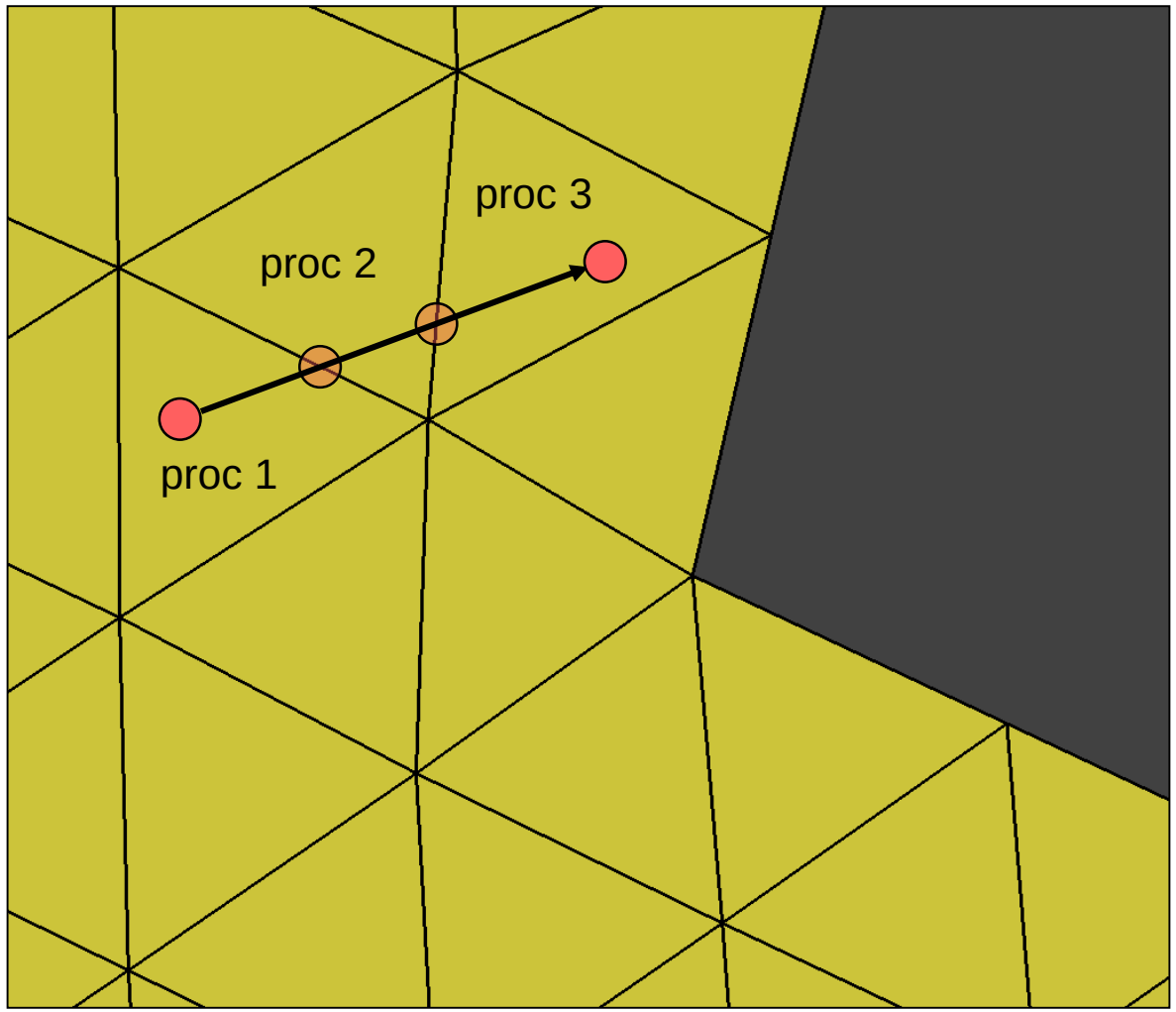
$$\Delta t < \nu_c^{-1} = \frac{\lambda_{mfp}}{v} = \frac{1}{n\sigma v}$$

Once physics determines Δx and Δt, and w_p is selected,

computational work scales as $P \cdot T \cdot L^d$,

P = pressure, T = total time, L = domain length, d = dimension (0, 1, 2, 3)
(for similar P)

Unstructured Meshes: Particle Push



- Particle push from PIC algorithm: move from x^n to x^{n+1}. In Cartesian meshes, final particle location lookup is “easy”.
- In unstructured meshes, we pass the particle from cell to cell.
- Especially critical in parallel!

Particle push algorithm:

compute x^{n+1} for all particles on this processor
while(particles still to move on any processor):
  for each particle on this processor:
    if particle intersected edge,
      update cell owner, or
      store in lists to send to other processors
  send lists to other processors
  receive lists from other processors

Solving Poisson's Equation

Solving large-scale/3D discretizations of Poisson's equation requires use of advanced linear solver technology.

Cartesian meshes typically discretize via the finite different method (FDM).

Unstructured meshes typically discretize via the finite element method (FEM).

Letting A = linear system from discretization method, x = solution vector of unknown potentials (V), and b = right-hand side (ρ), instead of solving

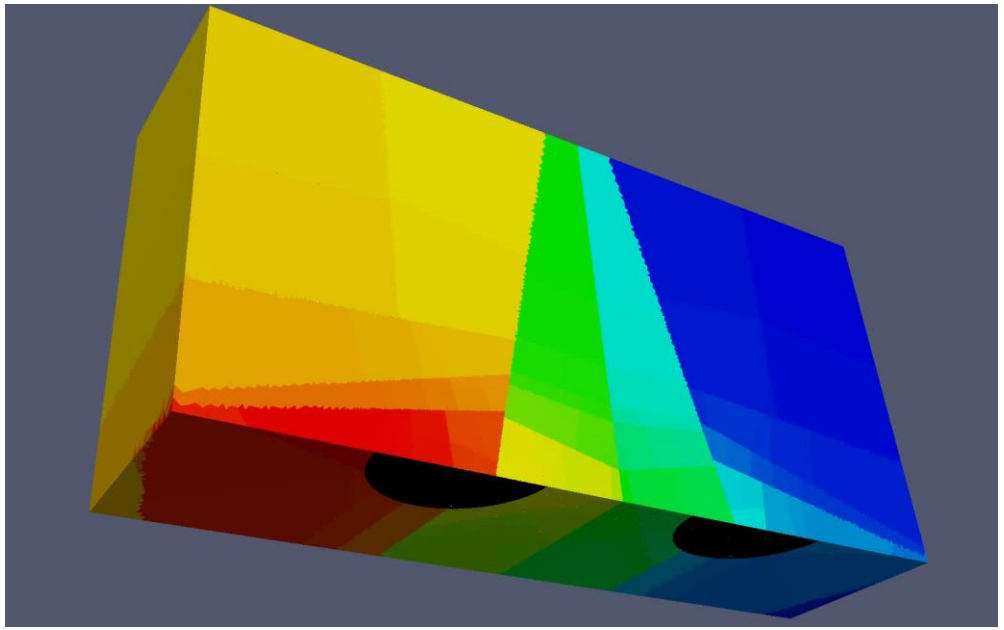
$$Ax = b,$$

directly (not possible for large/3D problems), we use iterative solvers (e.g., CG or GMRES) and compute a pre-conditioner M so that $M \sim A^{-1}$. Typically only $y = MAx$ operations are required,

$$x^{k+1} = S(x^k, b, M, A)$$

Incomplete LU factorizations are a popular class of preconditioners. At very large scale, these become ineffective and multilevel/algebraic multigrid methods are employed (with coarsening and cycling).

Aleph Simulation Tool Capabilities



- 1, 2, or 3D Cartesian
- Unstructured FEM (compatible with CAD)
- Massively parallel
- PIC + DSMC (PIC-MCC)
- Electrostatics
- Fixed B field
- Solid conduction
- Advanced surface (electrode/dielectric) models
- Advanced particle weighting methods
- Dynamic load balancing (tricky)
- e- approximations (quasi-neutral ambipolar, Boltzmann)
- Collisions, charge exchange, chemistry, excited states, ionization
- Finite-rate n-body reactions
- Photon transport, photoemission, photoionization, photoexcitation, radiative transitions
- Dual mesh (Particle and Electrostatics/Output)
- Restart (with all particles)
- Agile software infrastructure for extending BCs, post-processed quantities, etc.
- Currently utilizing up to 64K processors (>1B elements, >1B particles)

Timing Information

```
Run:  134433 (99.8035%) [500000]
   Particles:  39735.1 (29.5575%) [1500000]
     Repopulate:  23.3093 (0.0586616%) [500000]
     Apply_BCs:  0.209268 (0.000526659%) [500000]
     Verlet_Initial:  4944.07 (12.4426%) [500000]
     Find_Intersections:  22487.9 (56.5947%) [500000]
       Locate:  7271.57 (32.3354%) [2021576]
       Communicate:  1506.68 (6.69994%) [2021576]
         Send_Recv_All:  818.395 (54.3178%) [2021576]
           Pre_Send:  301.774 (36.8739%) [2021576]
           Wait_Recv_Count:  118.126 (14.4338%) [2021576]
           Recv_Data:  344.948 (42.1493%) [2021576]
           Flush_Sends:  50.3211 (6.14876%) [2021576]
           Other:  3.22621 (0.394212%) [0]
         Allocate_Mem:  0.27693 (0.0183802%) [2021576]
         Mem_Copy:  46.9851 (3.11846%) [2021576]
         Allreduce:  638.79 (42.3972%) [2021576]
         Other:  2.23195 (0.148137%) [0]
       Sort_1:  11972.4 (53.2391%) [500000]
         Sort_Memory:  11962.6 (99.9179%) [500000]
         Other:  9.82558 (0.0820688%) [0]
       Elemental_Coords_1:  0.0538756 (0.000239575%) [500000]
       Other:  1737.27 (7.72532%) [0]
     Inject_Provided_Particles:  0.184597 (0.000464569%) [500000]
     Sort_1:  0.0610631 (0.000153675%) [500000]
     Compute_F:  11031.9 (27.7636%) [500000]
```

```
       Interactions:  336.008 (0.84562%) [50000]
       Sort_2:  891.465 (2.24352%) [500000]
         Sort_Memory:  891.265 (99.9776%) [50000]
         Other:  0.199597 (0.0223897%) [0]
       Other:  19.9512 (0.0502105%) [0]
     Fields:  33280 (24.7558%) [500000]
       Compute_V:  31078.4 (93.3844%) [500000]
         Precompute:  0.416644 (0.00134062%) [500000]
           Potential_Field_Solve:  0.0879288 (21.104%) [500000]
           Other:  0.328716 (78.896%) [0]
         Compute:  31068.1 (99.9669%) [500000]
           Potential_Field_Solve:  31067.8 (99.999%) [500000]
             Assemble_RHS:  7192.57 (23.1512%) [500000]
               Idle_At_Rho_Copy:  854.922 (11.8862%) [500000]
               Rho_Intermesh_Copy:  4886.77 (67.9419%) [500000]
```

Timing Information

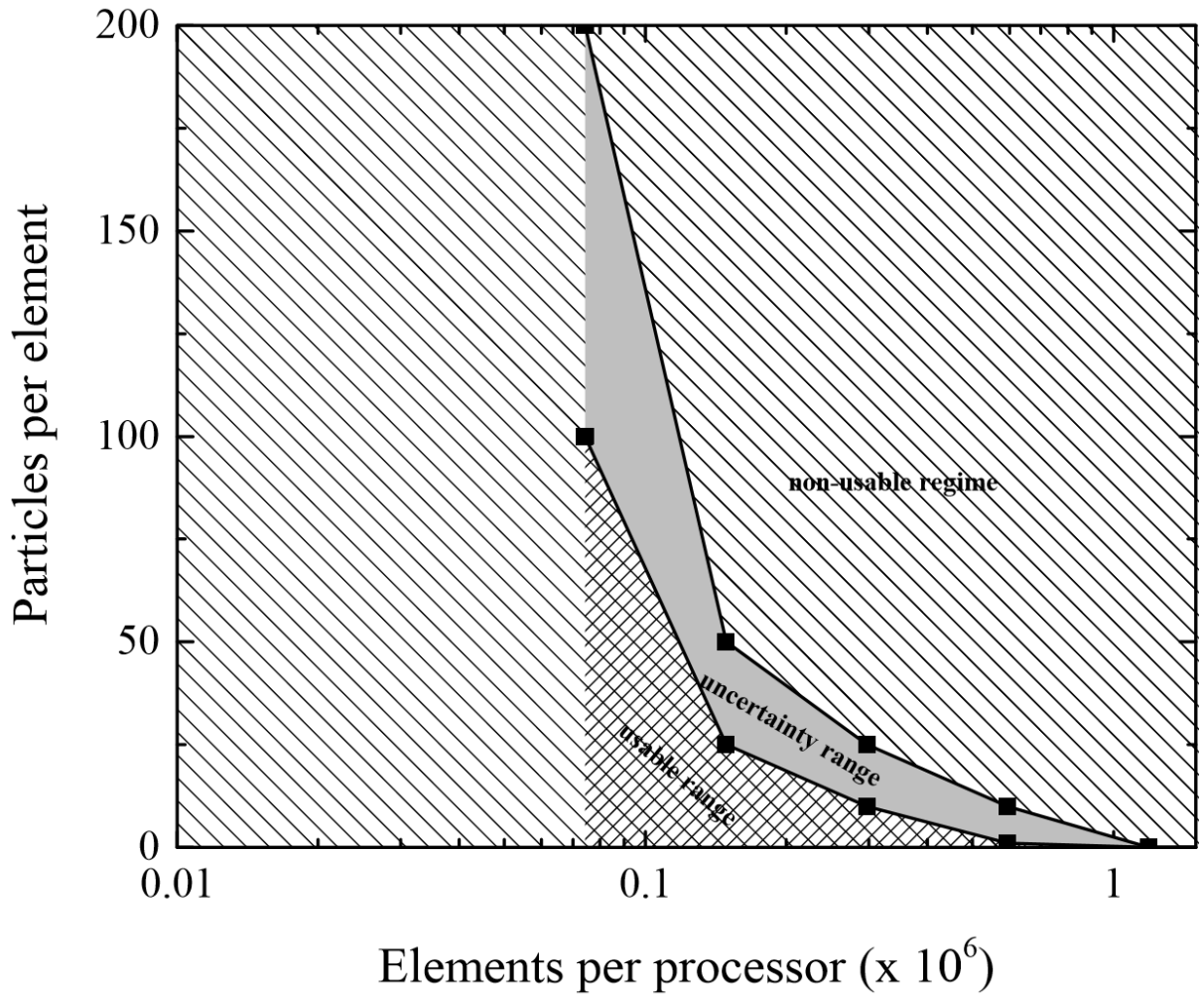
```
Run  (avg=134433, min=134422, max=134441)   Standard Deviation = 4.28216   [  1 4 2 8 6 10 12 10 5 6 ]  Out of balance ratio = 0.00013804
     Particles  (avg=39735.1, min=38166.4, max=40556.2)   Standard Deviation = 542.131   [  1 4 1 2 7 5 13 11 11 9 ]  Out of balance ratio = 0.0177423
       Repopulate  (avg=23.3093, min=0.747947, max=85.6636)   Standard Deviation = 29.4383   [  36 6 1 0 2 1 4 5 6 3 ]  Out of balance ratio = 0.000630416
       Apply_BCs  (avg=0.209268, min=0.161728, max=0.300854)   Standard Deviation = 0.029581   [  3 17 12 11 8 5 2 4 1 1 ]  Out of balance ratio = 1.03288e-06

     ...

     Idle_At_Evals  (avg=865.679, min=514.704, max=1393.53)   Standard Deviation = 170.202   [  3 5 8 21 12 7 3 2 1 2 ]  Out of balance ratio = 0.00652443
     Evaluations  (avg=59981.2, min=59977.3, max=59986.3)   Standard Deviation = 2.37894   [  8 8 8 6 5 7 8 9 4 1 ]  Out of balance ratio = 6.68452e-05
       Precompute  (avg=65.7567, min=34.3397, max=182.412)   Standard Deviation = 31.796   [  24 17 9 4 3 2 0 4 0 1 ]  Out of balance ratio = 0.0010993
         Species_Particle_Count_Evaluation  (avg=4.56685, min=2.02651, max=14.4043)   Standard Deviation = 2.61873   [  24 17 9 5 3 1 2 2 0 1 ]  Out of balance ratio = 9
         Average_Particle_Velocity  (avg=21.7398, min=8.11548, max=71.3385)   Standard Deviation = 13.6429   [  24 16 10 3 4 2 0 4 0 1 ]  Out of balance ratio = 0.0004693
         Evaluation_Data_Transfer  (avg=0.765688, min=0.710515, max=0.86018)   Standard Deviation = 0.0344946   [  9 7 12 8 12 6 6 0 2 2 ]  Out of balance ratio = 1.11112
         Maximum_Particle_Velocity  (avg=10.284, min=3.82024, max=34.0602)   Standard Deviation = 6.47312   [  24 16 10 3 4 2 0 4 0 1 ]  Out of balance ratio = 0.0002245
         particle_CFL_Evaluation  (avg=3.66107, min=1.54569, max=11.6476)   Standard Deviation = 2.14972   [  25 16 10 2 4 2 2 1 1 1 ]  Out of balance ratio = 7.49966e-05
         Temperature_Evaluation  (avg=7.12431, min=2.88412, max=22.3872)   Standard Deviation = 4.21581   [  24 16 10 3 4 2 0 3 1 1 ]  Out of balance ratio = 0.000144792
         Sideset_Current_Global_Evaluation  (avg=0.290127, min=0.260513, max=0.341291)   Standard Deviation = 0.0183598   [  8 11 3 13 20 2 0 2 2 3 ]  Out of balance rati
         Species_Particle_Density_Evaluation  (avg=5.34804, min=2.62853, max=15.1271)   Standard Deviation = 2.65789   [  22 18 10 4 3 2 1 3 0 1 ]  Out of balance ratio =
         Surface_Charge_Accumulation_Evaluation  (avg=0.0842019, min=0.0722215, max=0.115557)   Standard Deviation = 0.00951744   [  15 11 14 8 4 6 4 0 1 1 ]  Out of bala
         Surface_Charge_Volume_Density_Evaluation  (avg=0.0553855, min=0.03707, max=0.0660501)   Standard Deviation = 0.00854639   [  3 7 7 1 1 0 4 16 19 6 ]  Out of bala
         Particle_Operator_Global_Evaluation:return_one  (avg=0.0685139, min=0.0613213, max=0.0784082)   Standard Deviation = 0.00385156   [  6 6 6 14 9 10 7 2 3 1 ]  Out
         Charge_Density  (avg=0.00147, min=0.000722408, max=0.0040884)   Standard Deviation = 0.00073889   [  24 16 9 3 4 3 0 4 0 1 ]  Out of balance ratio = 2.49892e-08
         Species_Computational_Particle_Count_Evaluation  (avg=0.00549925, min=0.00252962, max=0.0168593)   Standard Deviation = 0.00308679   [  25 17 8 5 1 3 0 4 0 1 ]
         DeltaX_LambdaD_Evaluation  (avg=0.00156671, min=0.000782967, max=0.00444937)   Standard Deviation = 0.000777335   [  23 17 10 3 4 2 1 3 0 1 ]  Out of balance rat
         OmegaPe_DeltaT_Evaluation  (avg=0.00144324, min=0.000682354, max=0.00425363)   Standard Deviation = 0.000770809   [  25 15 10 2 5 2 0 4 0 1 ]  Out of balance rat
       Compute  (avg=57602.4, min=55903.2, max=58074.2)   Standard Deviation = 469.08   [  1 1 3 0 3 3 3 11 18 21 ]  Out of balance ratio = 0.0161172
         Species_Particle_Count_Evaluation  (avg=5349.46, min=5012.71, max=5531.61)   Standard Deviation = 99.1611   [  2 0 1 5 4 6 16 20 8 2 ]  Out of balance ratio = 0.
         Average_Particle_Velocity  (avg=6036.94, min=5764.9, max=6218.75)   Standard Deviation = 84.0199   [  2 0 1 2 12 14 14 12 4 3 ]  Out of balance ratio = 0.0033693
         Evaluation_Data_Transfer  (avg=0.660764, min=0.505657, max=0.852213)   Standard Deviation = 0.0654181   [  4 4 7 5 15 23 1 4 0 1 ]  Out of balance ratio = 2.5728
```

Problem Sizing

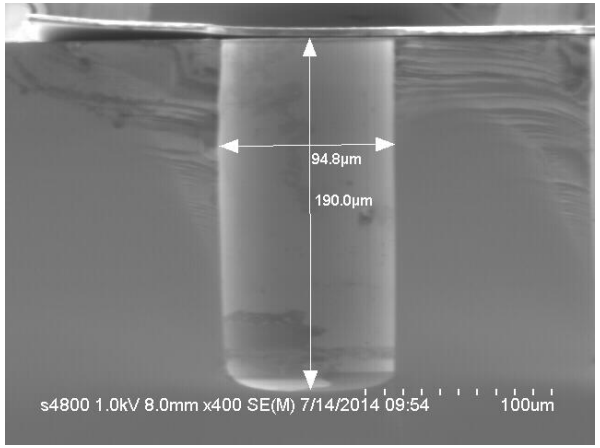
It can be complicated to know how many processors are required.

Typically, 1,000 – 100,000 cells/processor, 100,000 particles/processor.

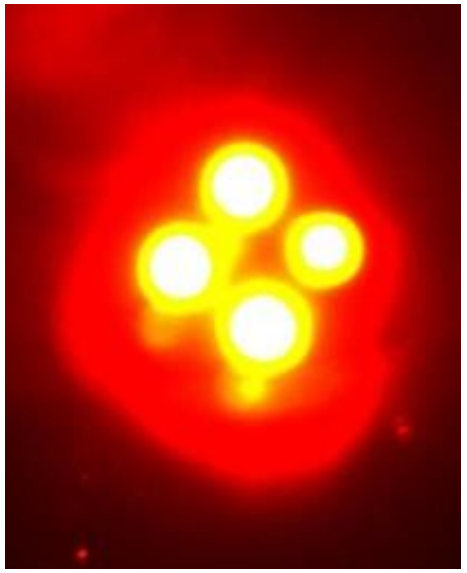
Dynamic problems can be very, very, challenging to size. Requires good load balancing.

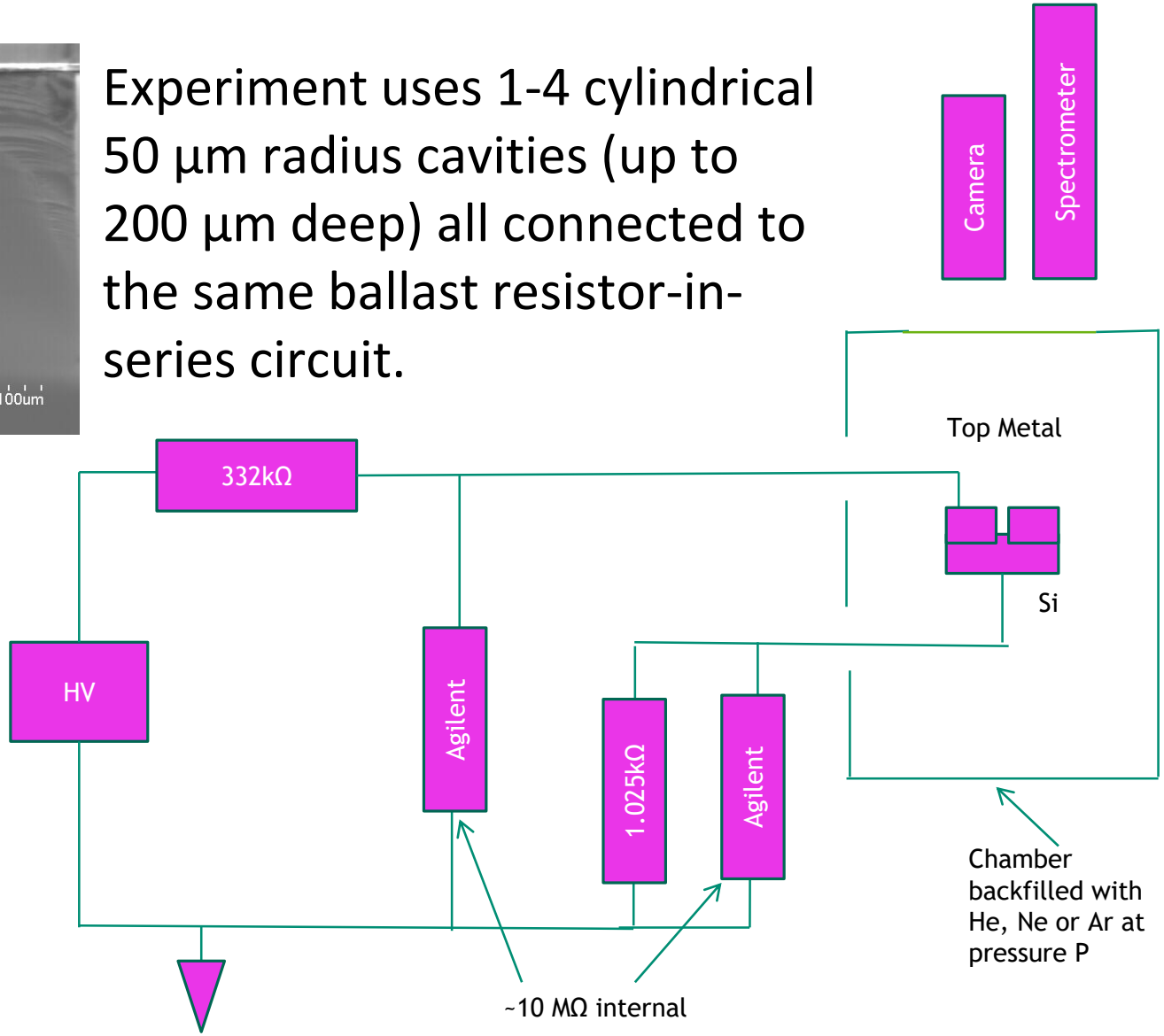


Example: 3D Microscale Discharge in 655 Torr Neon



Experiment uses 1-4 cylindrical 50 µm radius cavities (up to 200 µm deep) all connected to the same ballast resistor-in-series circuit.





Example: 3D Microscale Discharge in 655 Torr Neon

Experiment

655 Torr 300 K Ne
332 kΩ resistor-in-series w/circuit elements
50 µm radius, 200 µm depth, 10 µm spacer
1-4 full microcavities
Full chemistry

ε = 3 10 µm polyimide dielectric

Model

655 Torr 300 K Ne (n_{Ne} = 2.1 x 10^{25}/m^3)
$V_A = V_{PS} - IR$, R = 332 kΩ, I averaged ~ 10 ps
50 µm radius, 200 µm depth, 10 µm spacer
Single 3D 20 degree sector
Ionization, excitation, elastic (6 tracked species), from LXCat, www.lxcat.net
ε = 3 10 µm polyimide dielectric w/ surface charging
SEE γ = 0.15 for Ne+

Computational Parameters

Targeting $n_{e-} < 10^{20}/m^3$, T_e = 4 eV,

λ_D > 1.1 µm → Δx < 1.1 um, [Debye length]

λ_{mfp} > 1.6 µm → Δx < 1.6 um, [Collision mean free path]

Use Δx = 1.0 µm.

Targeting ΔV < 200 V, v_{max} = maximum e- speed (~ 9.4 x 10^6 m/s including thermal),

$\omega_p < 5.6 \times 10^{11}$/s → Δt < 3.5 ps, [Plasma e- frequency]

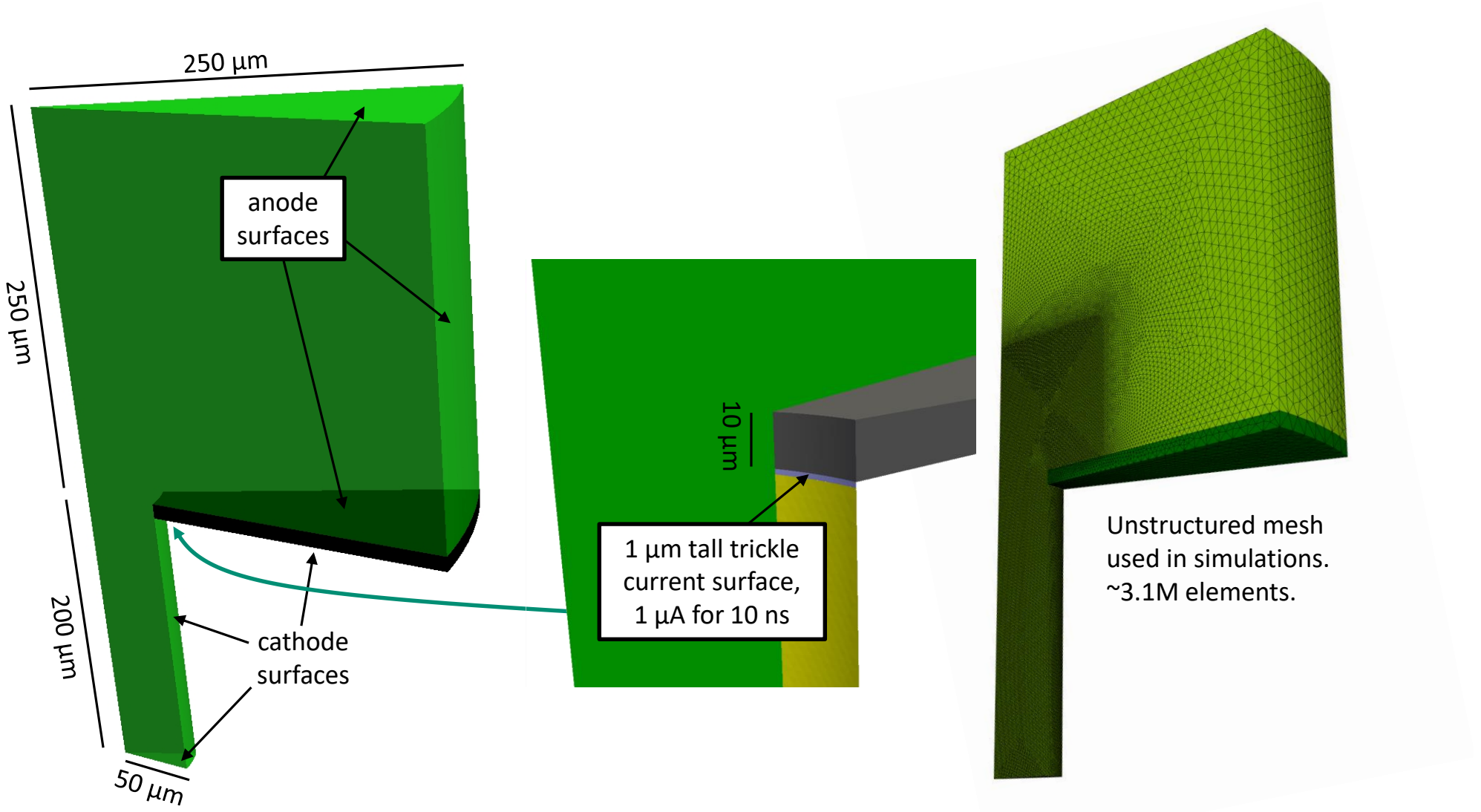
$\Delta t < \Delta x / v_{max}$ → Δt < 100 fs, [CFL]

$\Delta t_{collide} < (n_{Ne} \sigma_{max} v_{max})^{-1}$ → Δt < 170 fs, [Collision frequency]

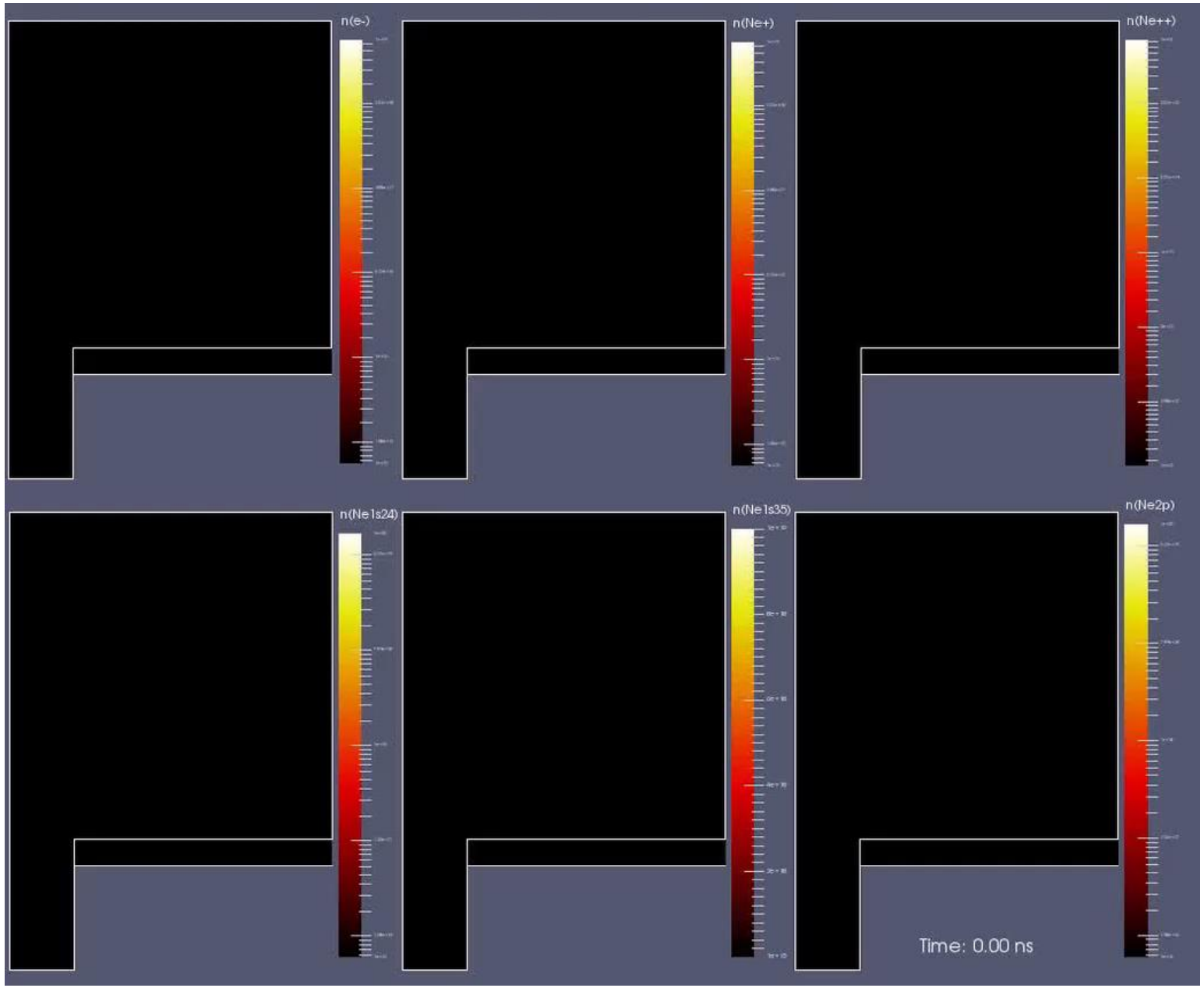
Use Δt = 50 fs.

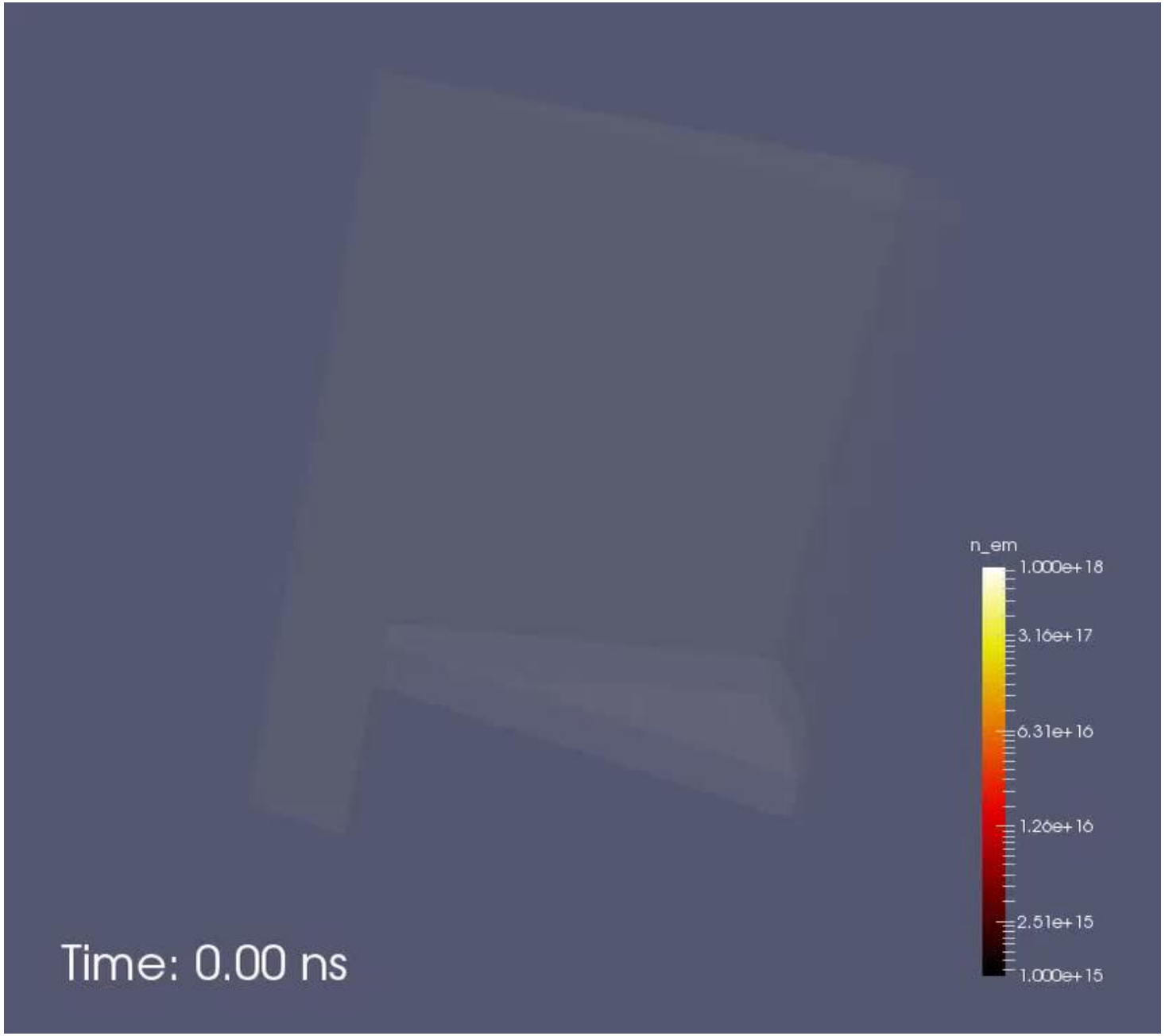
Use w_p = 0.01 (initially)

Example: 3D Microscale Discharge in 655 Torr Neon



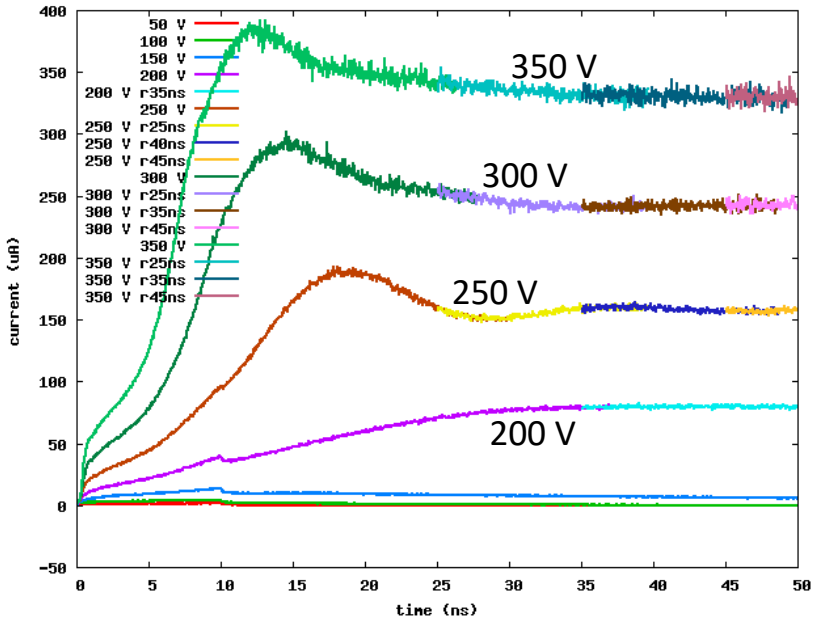
e-	Ne+	Ne++
$Ne(1s_{2,4})$	$Ne(1s_{3,5})$	Ne(2p)

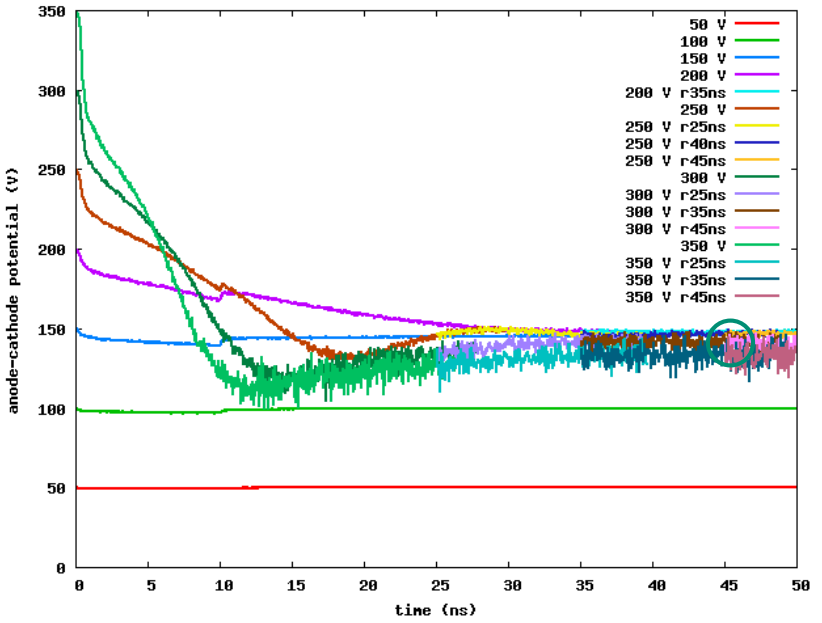


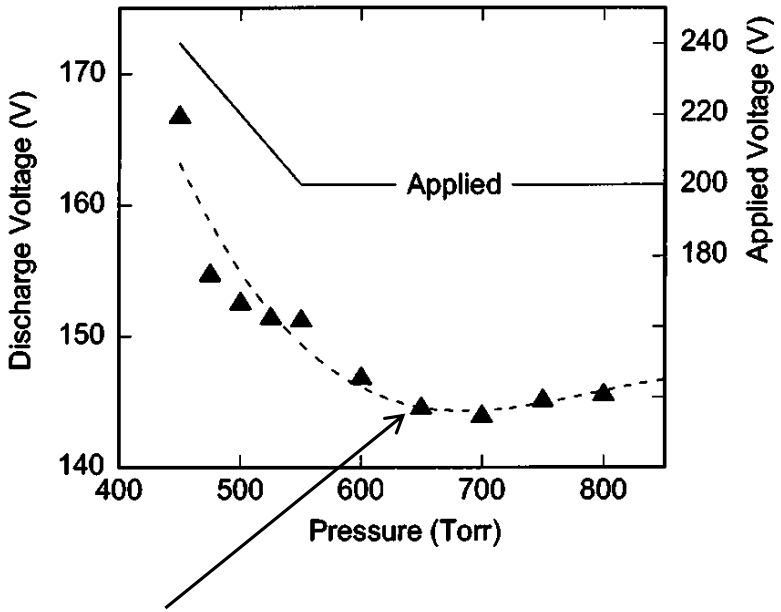

n_em
1.000e+18
3.16e+17
6.31e+16
1.26e+16
2.51e+15
1.000e+15
Time: 0.00 ns

Example: 3D Microscale Discharge in 655 Torr Neon

Time-resolved results varying drive voltage over 50-350 V. Breakdown at 200 +/- 50 V. Calibrated Paschen model (A = 4.4/Torr/cm, B = 111 V/Torr/cm) estimates 210 V.





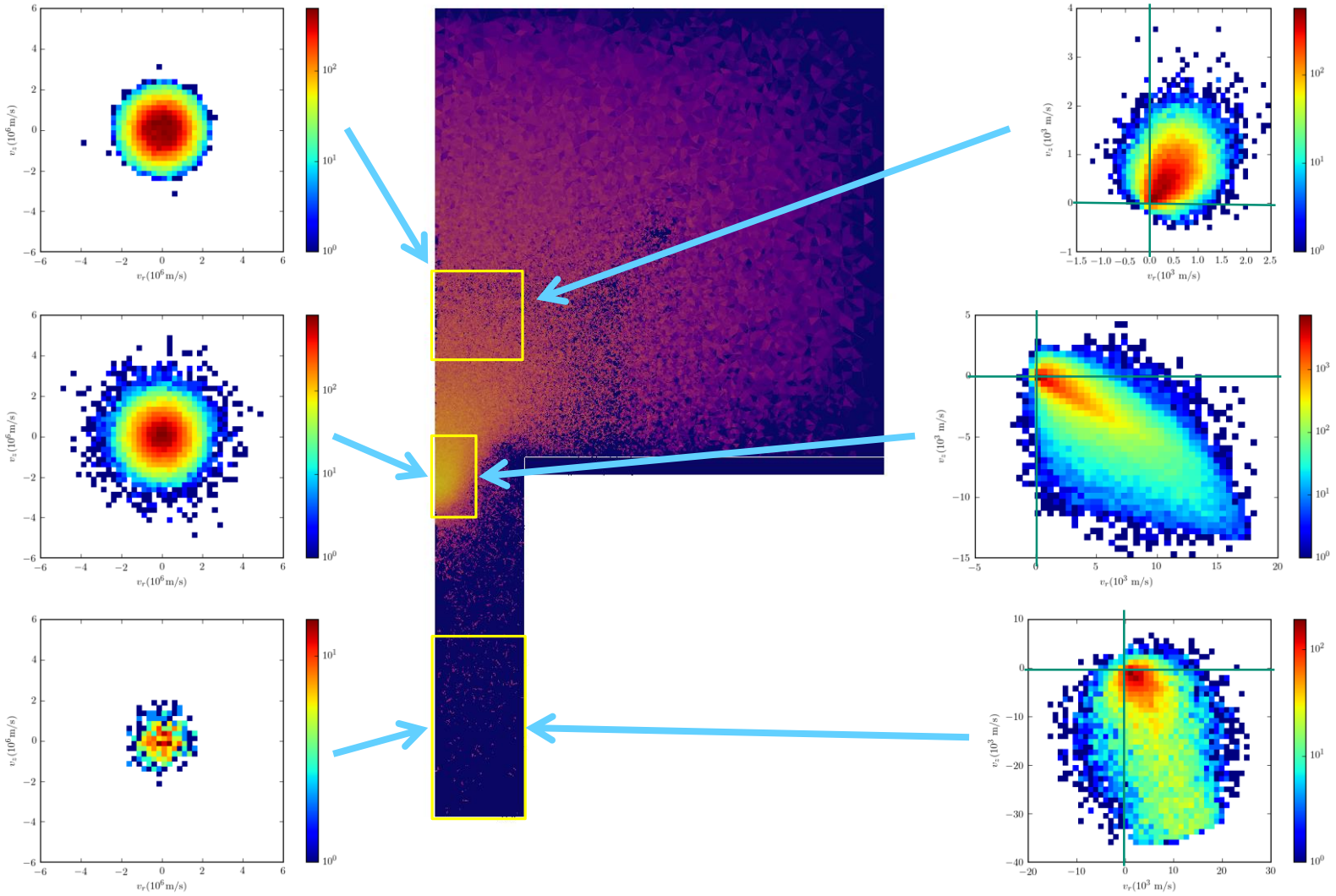


Steady state voltages approach ~145 V, a feature shared by normal glow discharges. This compares very well to a prior steady state study for a similar (not identical) system by Kushner.

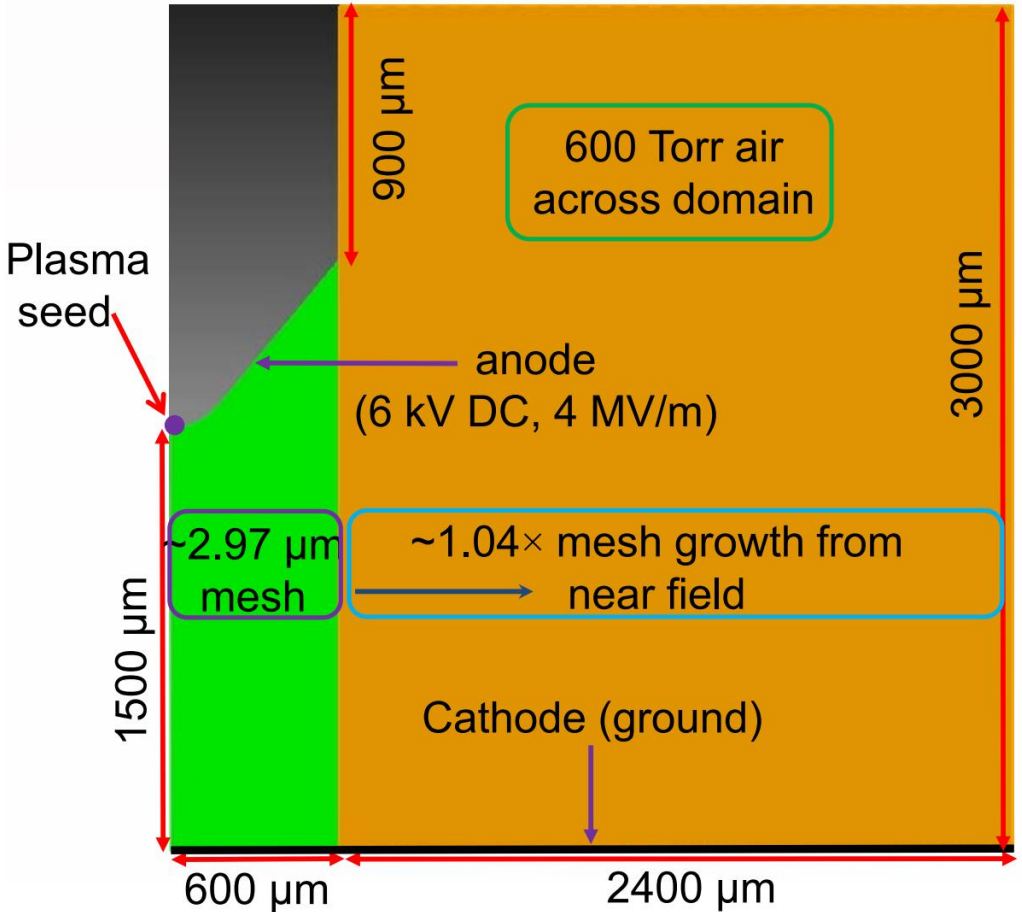
Each simulation is 48 hours on 512 cores. Results required multiple restarts (each different color above is a separate simulation).

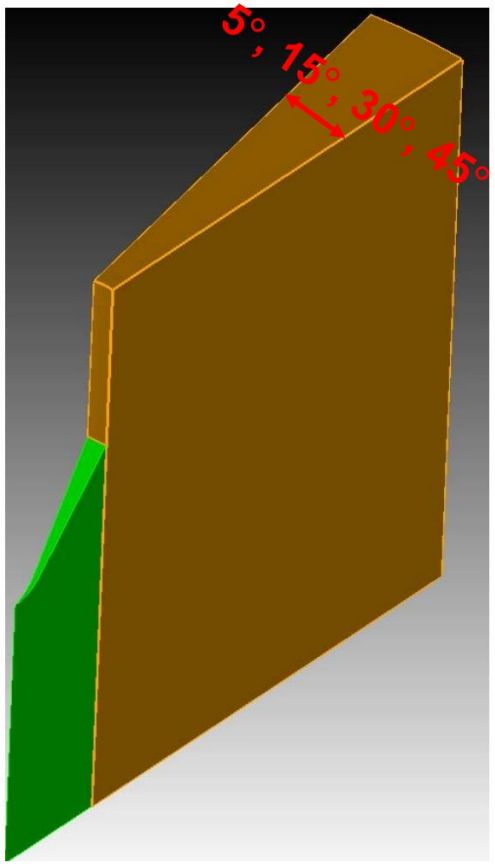
e- vdf

Ne+ vdf

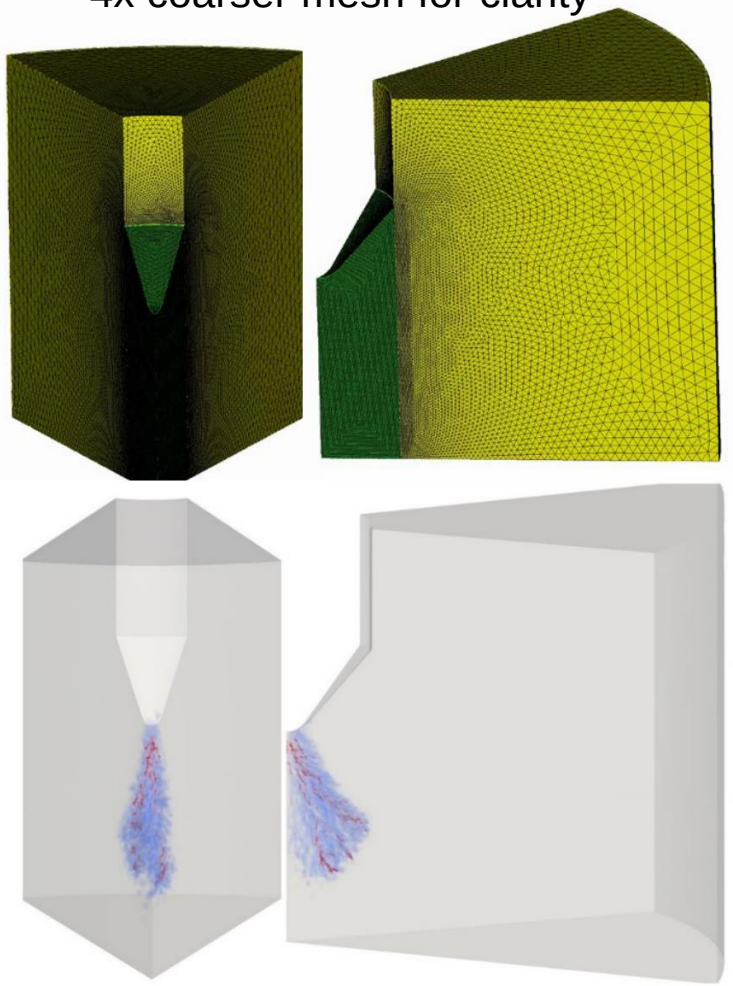

$v_z(10^6 \mathrm{m/s})$
$v_r(10^6 \mathrm{m/s})$
$v_z(10^3 \mathrm{m/s})$
$v_r(10^3 \mathrm{m/s})$

Example: 3D Pin-to-Plane Streamer in 600 Torr Air





4x coarser mesh for clarity



Total # elements (45°) ~100,000,000
Total # particles ~250,000,000
Maximum # processors = 8,192

Smallest Δx ~3 um
$\Delta t = 10^{-12}$ s

} Does not resolve; still working it!

Example: 3D Pin-to-Plane Streamer in 600 Torr Air

- Assume N_2 and O_2 are dominant species for heavy-heavy interactions. Model dry air and neglect N–N, N – O, and O – O interactions.
  - Include elastic (VHS), charge exchange, and quenching heavy-heavy interactions
- Include e-N_2^+ and e-O_2^+ dissociative recombination
- Include O_2^- + M detachment via cross section[5]
  - Self-consistently leads to higher detachment rate in high-field regions
- e-neutral interactions included for N_2, O_2, N, O and metastable states. Use anisotropic scattering model for all electron-neutral collisions.
  - Elastic
  - Ionization: Single (ground and metastable states), double, and dissociative
  - Attachment (3-body and Dissociative)
  - Vibrational and rotational excitation
  - Electronic excitation

Total of ~50 species, ~125 interactions, ~100 radiative transitions

- Excited states have probability to radiate a photon based on transition-specific Einstein-A coefficients, quench via collision (assumed P_{quench} = ½) with background neutrals, or, in some cases, auto-dissociate or auto-ionize with state-specific rate
- Photons are modeled as discrete particles that move and stochastically collide through a simulation timestep just like all other particles

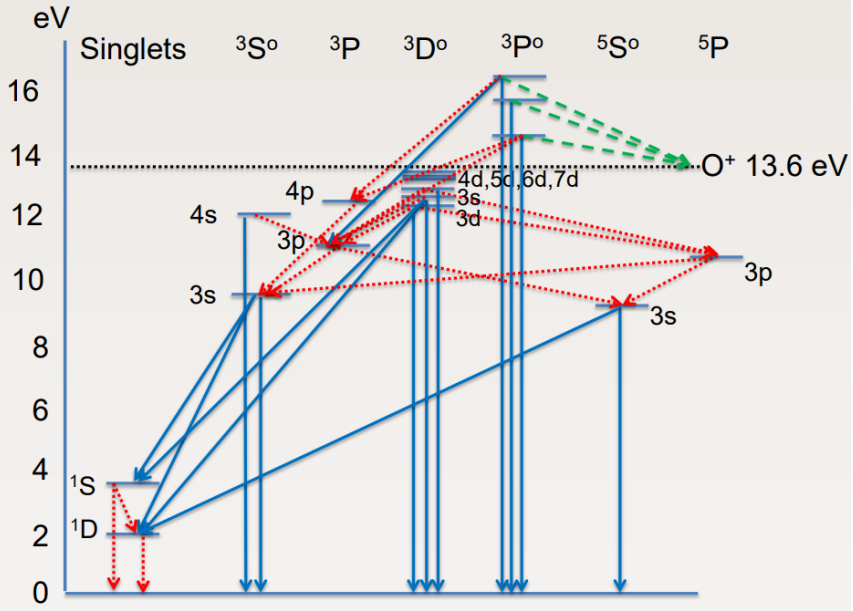
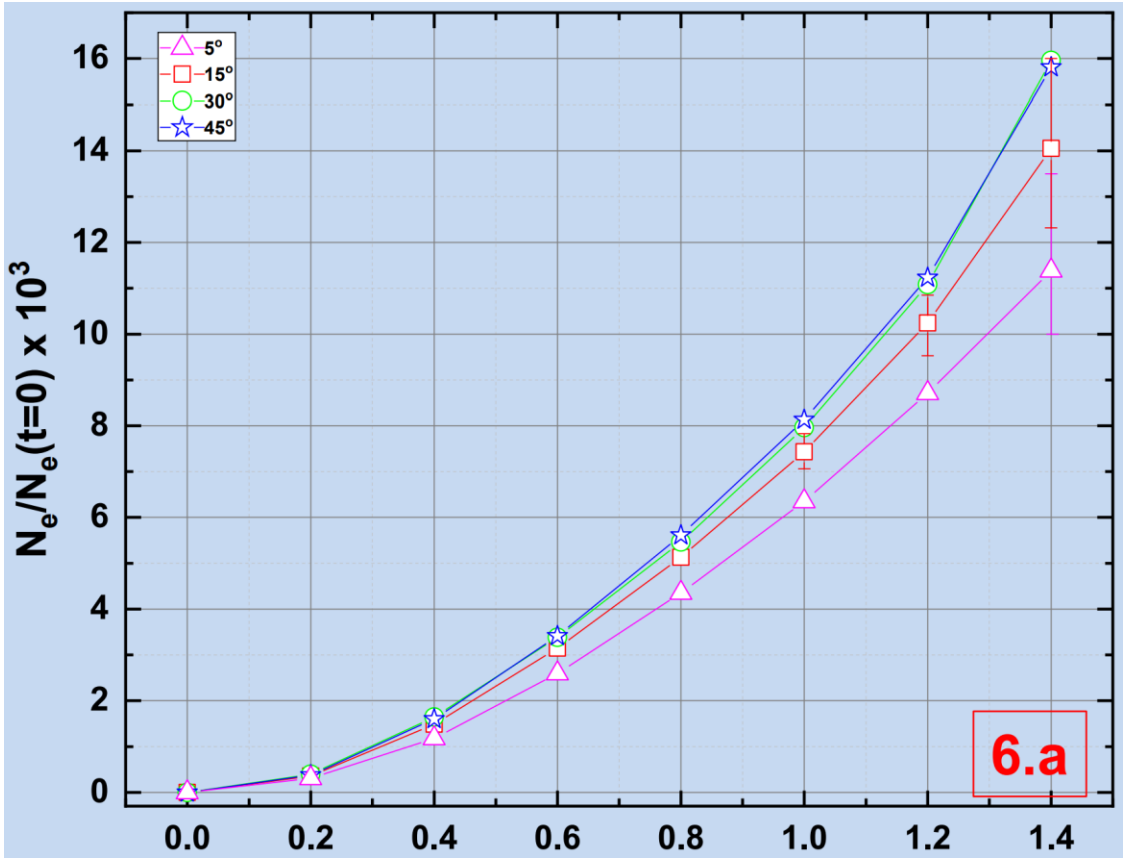
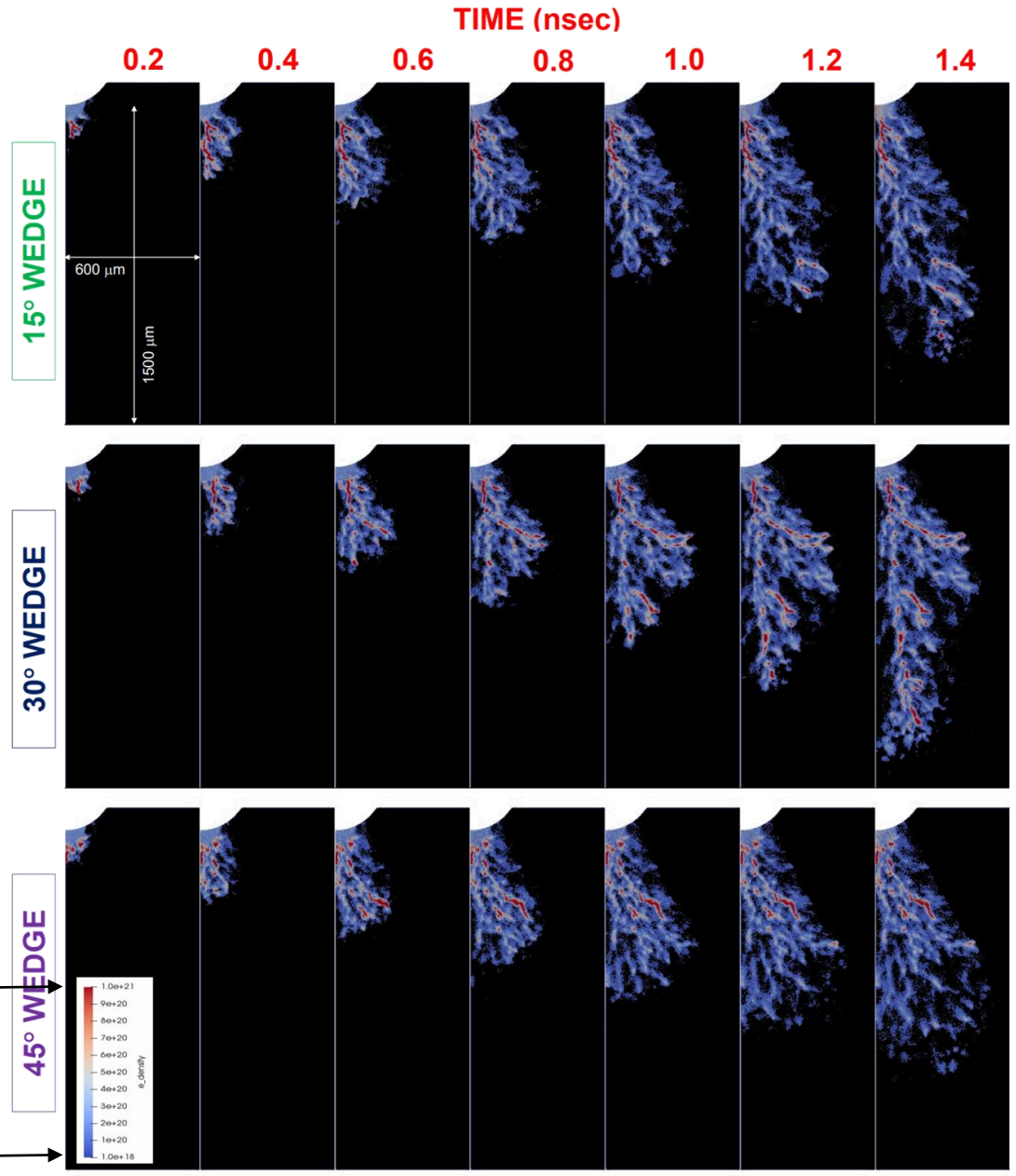


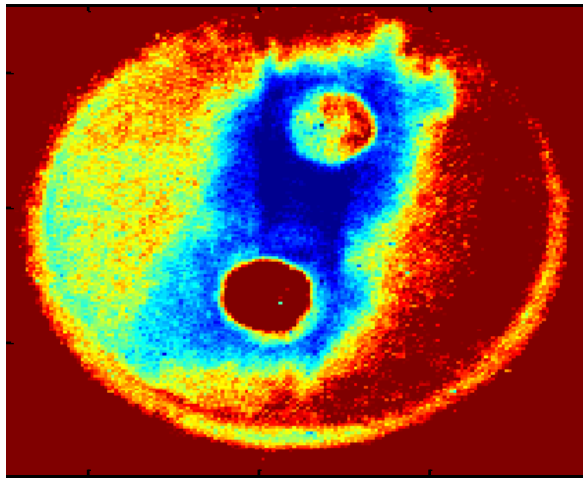
Fig. 1 Modeled energy level and transition diagram for atomic O ($^3D^o$ transitions omitted for clarity). Solid blue lines represent radiative decay in which simulation photons are generated. Red dotted lines represent decay in which a simulation photon is not generated. Green dashed lines are auto-ionizing states.

Example: 3D Pin-to-Plane Streamer in 600 Torr Air

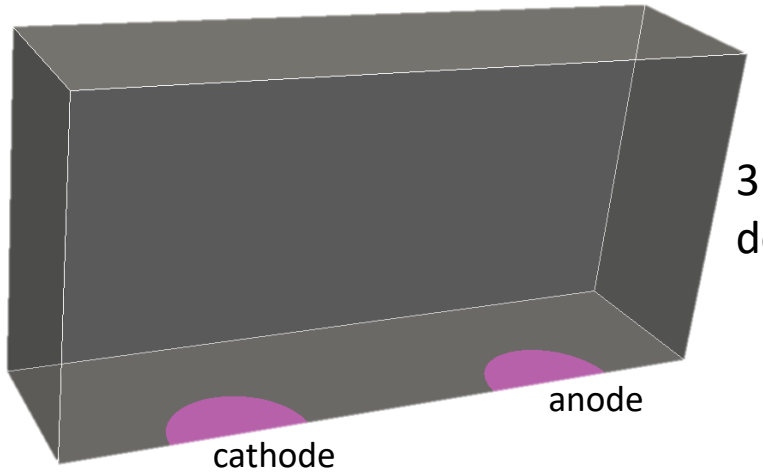




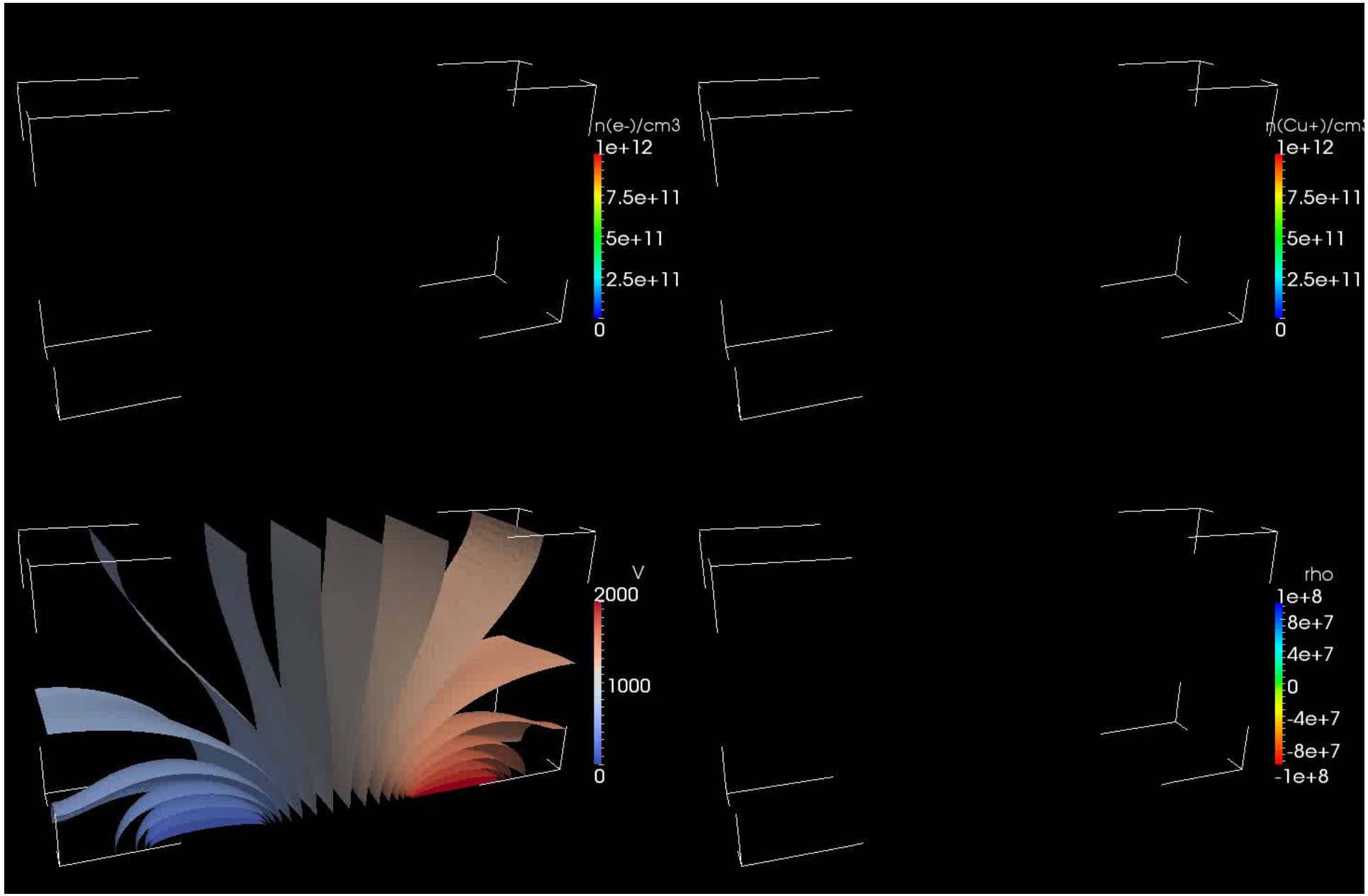
Example: 3D “Vacuum” Arc

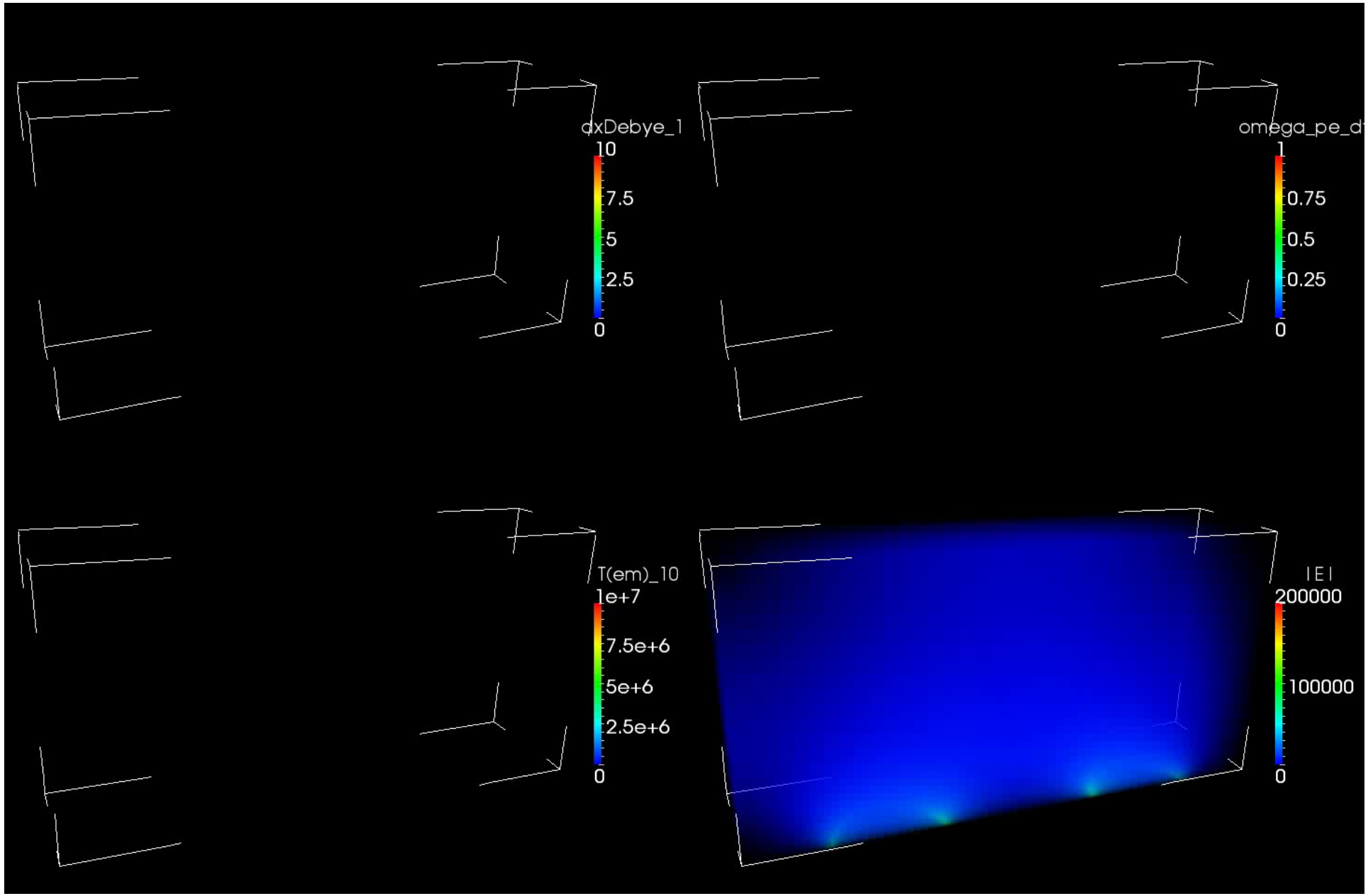


- In vacuum or 4 Torr Ar background
- 1.5 mm inner-to-inner distance
- 0.75 mm diameter electrodes
- Copper electrodes (this picture is Cu-Ti)
- 2 kV drop across electrodes
- 20 Ω resistor in series
- Steady conditions around 50V, 100A
- Breakdown time << 100ns
- To meet an ionization mean free path of 1.5 mm at maximum σ, n_{bg} ~ 10^{16} – 10^{17} #/cm^3



3D computational domain


n(e-)/cm3
1e+12
7.5e+11
5e+11
2.5e+11
0
n(Cu+)/cm3
1e+12
7.5e+11
5e+11
2.5e+11
0
V
2000
1000
0
rho
1e+8
8e+7
4e+7
0
-4e+7
-8e+7
-1e+8


dxDebye_1
10
7.5
5
2.5
0
omega_pe_d
1
0.75
0.5
0.25
0
T(em)_10
1e+7
7.5e+6
5e+6
2.5e+6
0
|E|
200000
100000
0

What Was Not Discussed?

- Adaptive mesh refinement
- Dynamic particle weighting
- Load balancing
- No formal foundation for unstructured PIC
- Hybrid modeling
- GPUs vs. CPUs (next generation SNL code, EMPIRE, is in development, EM-PIC-DSMC-hybrid)

Thank You!

Thanks to the many collaborators who contributed to the modeling discussed here, especially:

Matthew Bettencourt
Jeremiah Boerner
Paul Crozier
Andrew Fierro
Russell Hooper
Ashish Jindal
Christopher Moore

If interested in pursuing collaborations, please visit our Low Temperature Plasma Research Facility webpage, http://www.sandia.gov/prf/, funded by the US DOE Office of Science, Office of Fusion Energy Science.
Direct access to Aleph requires US citizenship.

My e-mail: mmhopki@sandia.gov