Advances in Plasma Simulation

John R Cary, CEO, Tech-X

Why do we compute?

- Prediction
- Design
- Discovery

How do we want to compute?

- Faster: less time to correct problem setup
- Faster: less time to problem execution
- Faster: less execution time (faster algorithm, faster implementations)
- Faster: less time to analysis
- Faster: less time to visualization

Outline

- Getting it right before getting it fast
- Shorten time from concept to simulation definition
- Shorten time from setup to parallel computing
- Shorten computing time, start to finish
  - With GPU capabilities
  - With new algorithms, like SLPIC
- Get your analyses done more quickly
- New capabilities in visualization

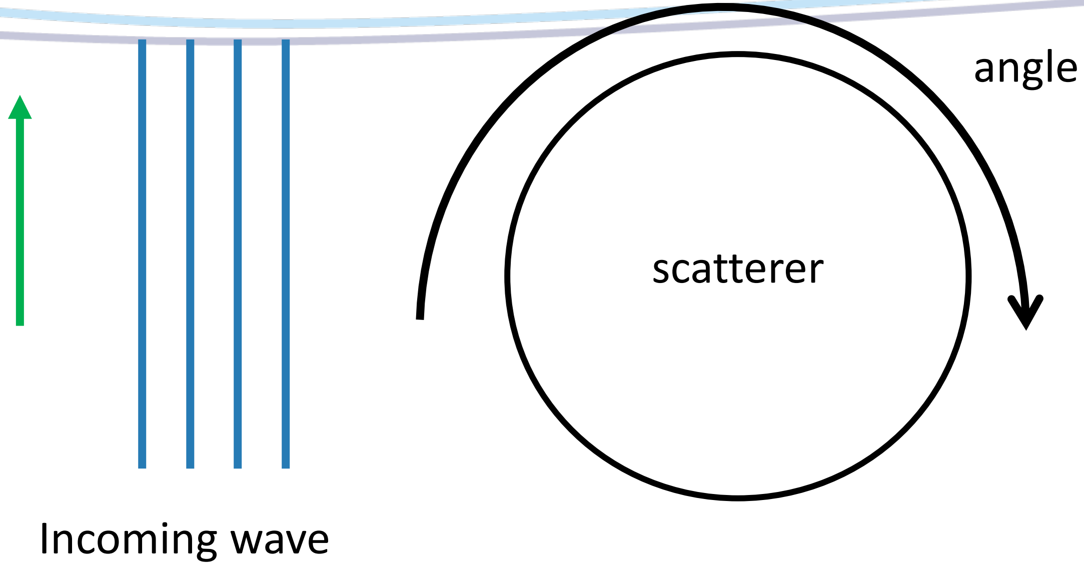
Get it right before getting it fast

- It's tough to make **predictions, especially about the future**. (Danish parliament, 1936, https://quoteinvestigator.com/2013/10/20/no-predict/)

Olde Stage Fire, Boulder, Jan 2009
The Denver Channel

- Example: Mie scattering

Just to show that we do EM right



- Decomposition of scattered wave

$$\begin{bmatrix} E_{\|_s} \\ E_{\perp_s} \end{bmatrix} = \frac{e^{ik(r-z)}}{-ikr} \begin{bmatrix} S_2 & 0 \\ 0 & S_1 \end{bmatrix} \begin{bmatrix} E_{\|_i} \\ E_{\perp_i} \end{bmatrix}$$

- S_1 and S_2 are functions of θ from MieSolver*

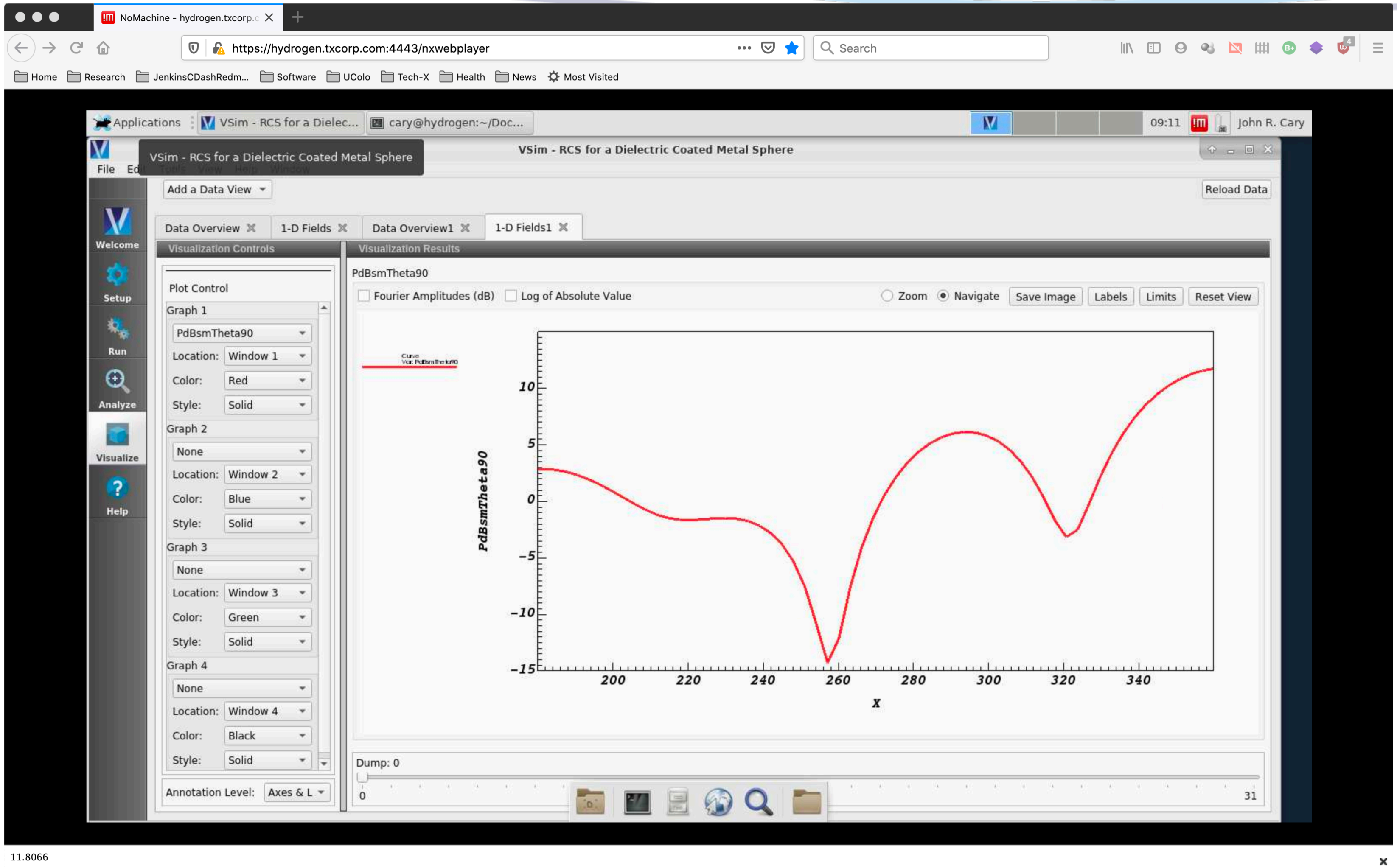
$$RCS = \frac{4\pi^2 P_s}{P_i} = \frac{|S_2|^2 \sin^2\phi + |S_1|^2 \cos^2\phi}{\pi} \lambda^2$$

*http://philiplaven.com/mieplot.htm

Time domain simulation has a few basic steps

- Problem setup
- Excite a source
- Absorb outgoing waves
- Analyze the data for far fields, power, …

Demo: Dielectric coated sphere (Remoting, cluster)

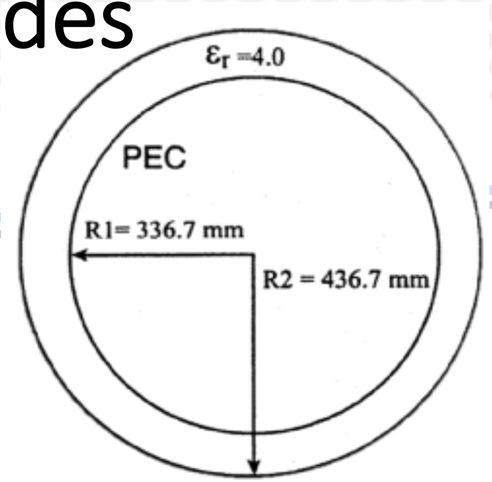


Tech-X

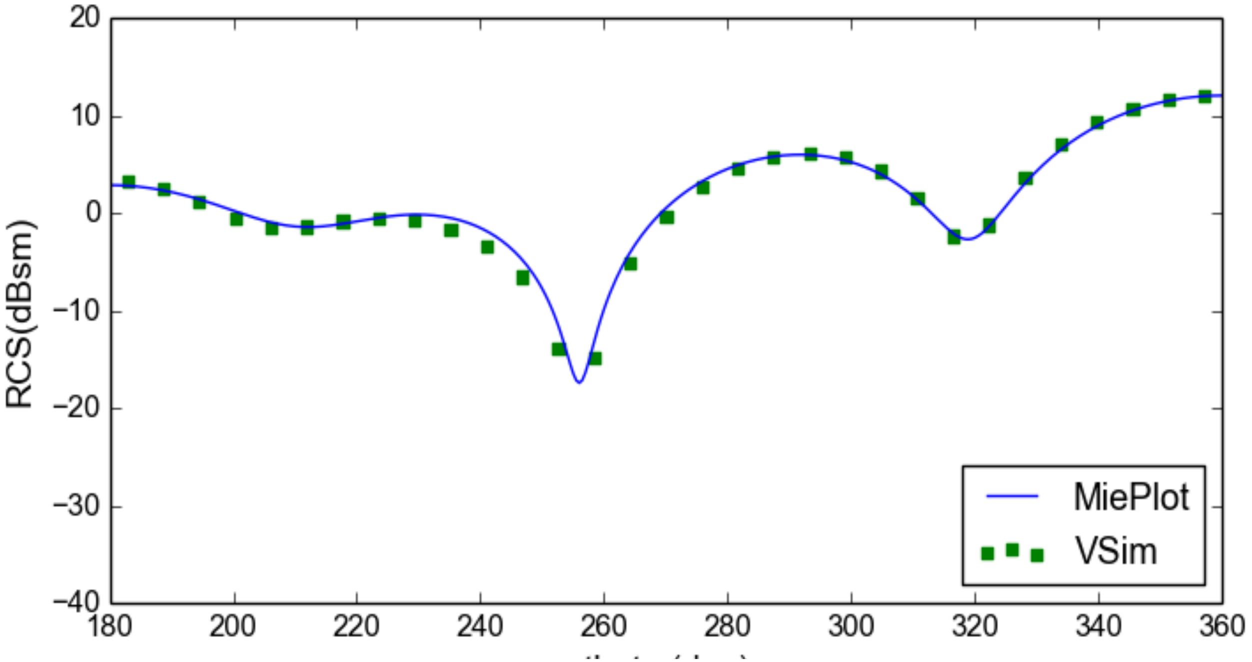
SIMULATIONS EMPOWERING INNOVATION

TWSS 2020

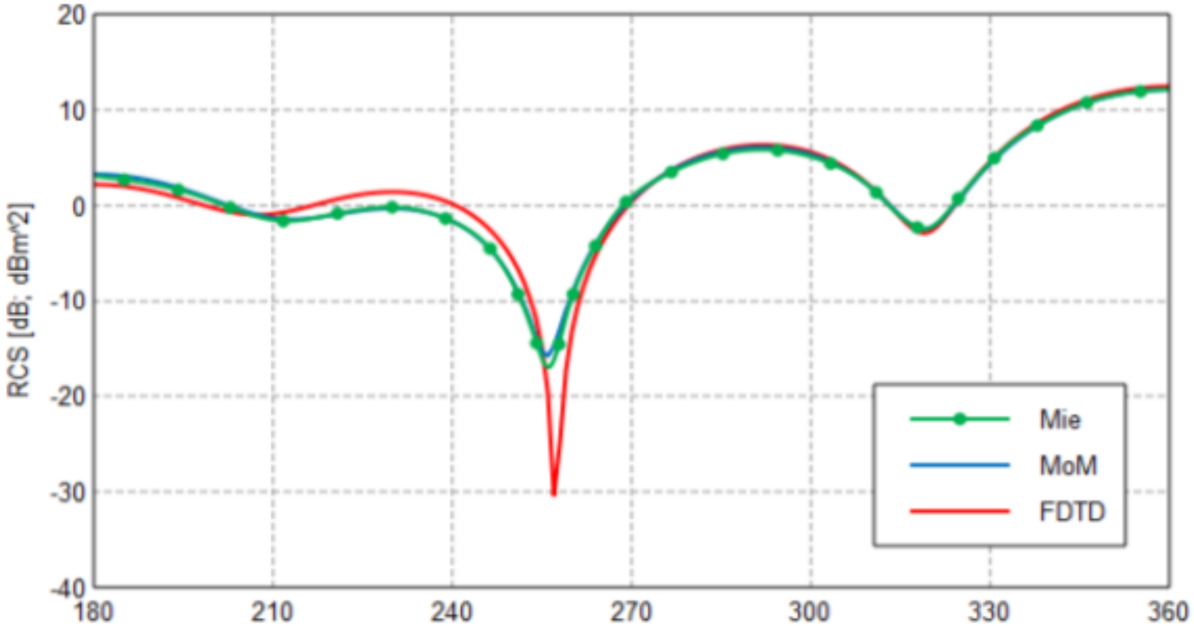
VSim 2nd-order FDTD solver for dielectrics provides accurate RCS at lower CPW



PEC Sphere of radius = 0.3367 m
Dielectric sphere of radius = 0.4367
Frequency = 300MHz
Wave electric field polarized parallel scattering plane



VSim
24 cells/wavelength



a competitor
32 cells/wavelength

The new VSimComposer will allow faster setup with more geometry options, more visual setup, faster meshing, better meshing

- DEMO: Healing
- DEMO: Shape arrays
- DEMO: Meshing plasma antennas: nstxAntVV.

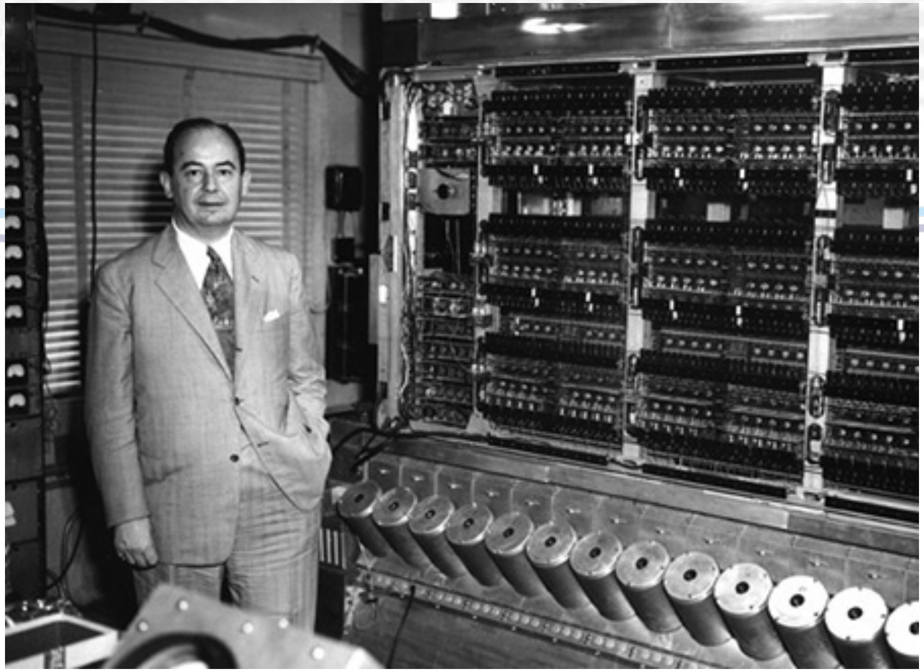
VSimComposer allows job control

- DEMO: dipole radiation

GPU computing: next step in faster computing devices

The Baby had a 32-bit word length and a memory of 32 words (1 kilobit) … The program consisted of 17 instructions and ran for 52 minutes before reaching the correct answer of 131,072, after the Baby had performed 3.5 million operations (for an effective CPU speed of 1.1 kIPS)

Replica of Manchester Baby (Wikipedia), was the world's first stored-program computer. 1948



John von Neumann,1945, with the IAS stored program computer, which did not run a stored program unit 1951. See *Turing's Cathedral*

Kiloscale (1945?)
Megascale
Gigascale
Terascale
Petascale
Exascale (2023?)

Running on GPUs in VSimComposer

- Euler Fluid

Speed Limited Particle In Cell: speeding up simulations by slowing down particles

- PIC simulations have limitations
- For slow phenomena, can solver alternate equation without those limitations
- Verifies (agrees with experiment)

Particle-in-Cell methods imply limits on time stepping, grid resolution

- $v_p \Delta t < \Delta x$: Particles must not move over many grid cells in a time step to get an accurate force and to provide an accurate current
- $\omega_p \Delta t \leq 1$: otherwise get strong instability, i.e., plasma CFL,
- $\Delta x \leq \lambda_D$: Debye length resolution needed to prevent grid instability
- All very related
- For electromagnetics, also EM CFL, again related for relativistic particles

These numerical limits are not related to resolution requirements

- Cold plasma oscillations: wavelength determines the physics, not Debye length (yet have to resolve for stability)
- MHD: electrons mostly just cancel electric field
- Ion-acoustic modes (electrons basically Boltzmann response)
- Plasma sheaths (one-sided, chopped electron Maxwellians)
- <u>Plasma discharges</u> (resolve ion crossing time, mfp)

Basic PIC methods – solve for distribution function by method of characteristics

- Conservation form

$$\partial_t f(\mathbf{x},\mathbf{v},t) + \nabla_x[\mathbf{v} f(\mathbf{x},\mathbf{v},t)] + \nabla_v[\mathbf{a}(\mathbf{x},\mathbf{v},t) f(\mathbf{x},\mathbf{v},t)] = 0$$

- Advection form

$$\partial_t f(\mathbf{x},\mathbf{v},t) + \mathbf{v}\cdot\nabla_x[f(\mathbf{x},\mathbf{v},t)] + \mathbf{a}(\mathbf{x},\mathbf{v},t)\cdot\nabla_v[f(\mathbf{x},\mathbf{v},t)] = 0$$

- Solution:

$$f(\mathbf{x},\mathbf{v},t) = \sum_p w_p \delta\left(\mathbf{x} - \mathbf{x}_p(t)\right)\delta\left(\mathbf{v} - \mathbf{v}_p(t)\right)$$

- w_p= particle weight
- x_p, v_p = particle trajectory, satisfying

$$\dot{\mathbf{x}}_p = \mathbf{v}_p \qquad \dot{\mathbf{v}}_p = \mathbf{a}(\mathbf{x}_p,\mathbf{v}_p,t)$$

- Discretize, put on grid, add fields…

SLPIC is based on a simple ansatz

$$f(\mathbf{x},\mathbf{v},t) = \beta(\mathbf{x},\mathbf{v},t)g(\mathbf{x},\mathbf{v},t)$$

$$\partial_t\left[\beta g(\mathbf{x},\mathbf{v},t)\right] + \nabla_x[\beta\mathbf{v}g(\mathbf{x},\mathbf{v},t)] + \nabla_v[\beta\mathbf{a}(\mathbf{x},\mathbf{v},t)g(\mathbf{x},\mathbf{v},t)] = 0$$

$$\partial_t\left[g(\mathbf{x},\mathbf{v},t)\right] + \nabla_x[\beta\mathbf{v}g(\mathbf{x},\mathbf{v},t)] + \nabla_v[\beta\mathbf{a}(\mathbf{x},\mathbf{v},t)g(\mathbf{x},\mathbf{v},t)] = \partial_t\left[(1-\beta)g(\mathbf{x},\mathbf{v},t)\right]$$

- Choose β such that
  - For slow particles, β = 1, so RHS vanishes
  - For fast particles, β ➔0, but RHS unimportant compared with phase space derivatives
  - In both cases, RHS can be neglected
- Distribution evolves as if velocity *and* acceleration reduced for fast particles

$$\partial_t\left[g(\mathbf{x},\mathbf{v},t)\right] + \nabla_x[\beta\mathbf{v}g(\mathbf{x},\mathbf{v},t)] + \nabla_v[\beta\mathbf{a}(\mathbf{x},\mathbf{v},t)g(\mathbf{x},\mathbf{v},t)] = 0$$

SLPIC fits into the DSMC-PIC cycle (almost – more later)

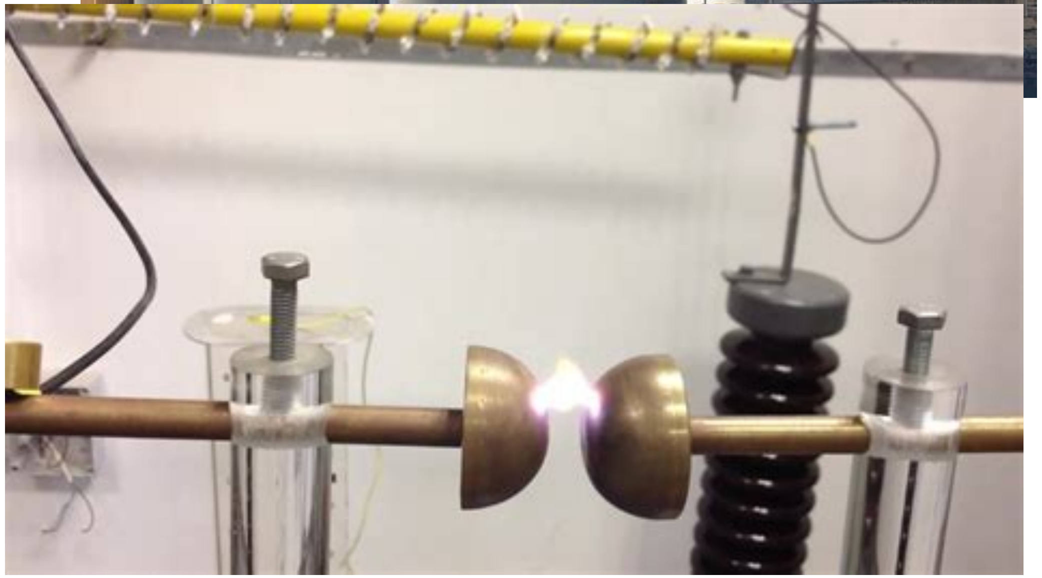
- Field solve (unchanged)
- Particles
  - Interpolate: same
  - Accelerate: modified acceleration, point-wise implicit algorithms solved by quartic for unmagnetized
  - Move: Just move less by β (could be implicit when β depends on x)
  - Deposit: change from standard pic is the variation of β from one end to other. More on this.
- Collisions are put in at end of particle push (makes this Direct Simulation Monte Carlo – Particle In Cell)

Expect big gains in computational speeds when

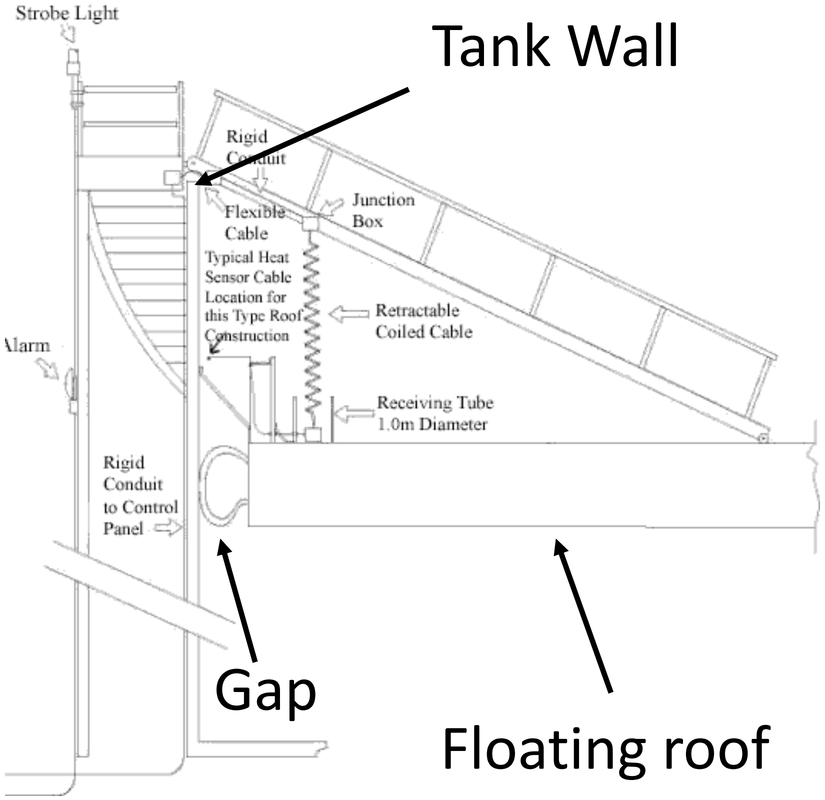
- $v_0 << v_e$
- Need not resolve electron plasma oscillations
- Especially good for
  - $T_e > T_i$
  - Large mass ions
- Examples
  - plasma sheath
  - free expansion
  - plasma thrusters

Computing electrical breakdown (in complex shapes) has a wide array of applications

- To avoid:
  - Tank farm fires
  - Electrical power distribution
- To enable
  - Nanoparticle generation
  - Plasma medicine
- To learn:
  - Undergraduate physics labs



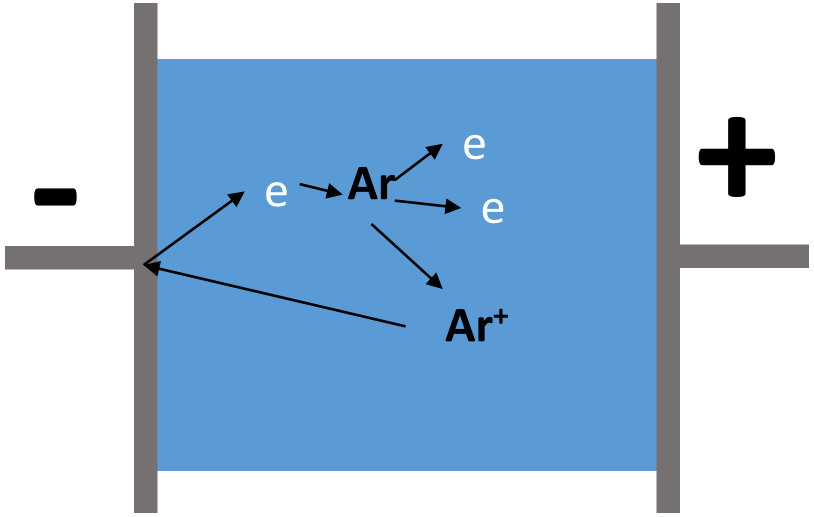
Oil tank: electric fields induced by lightning



- Lightning hits wall
- Induces electric field between wall and floating roof
- Floating roof sitting on a dielectric
- Put shorting straps between wall and roof
- Gap is 10's of cm
- Shorting straps are a few cm in width
- Tank is 30m diameter
- How many are needed?
- Ultimately unsuccessful proposal

Simplest case: Paschen breakdown

- Electrons accelerated from cathode to anode, producing on average α ions.
- Ions accelerate traveling back towards cathode and cause secondary emission on average of γ electrons
- If $\alpha\gamma > 1$, the current grows exponentially
- Process time is some multiple of the time for an ion (slower species) to travel to the anode
- Well understood for plate geometry, but what about tank farms, high-voltage circuit breakers, electrostatic accelerators?
- Let's just code this up and get the answer!



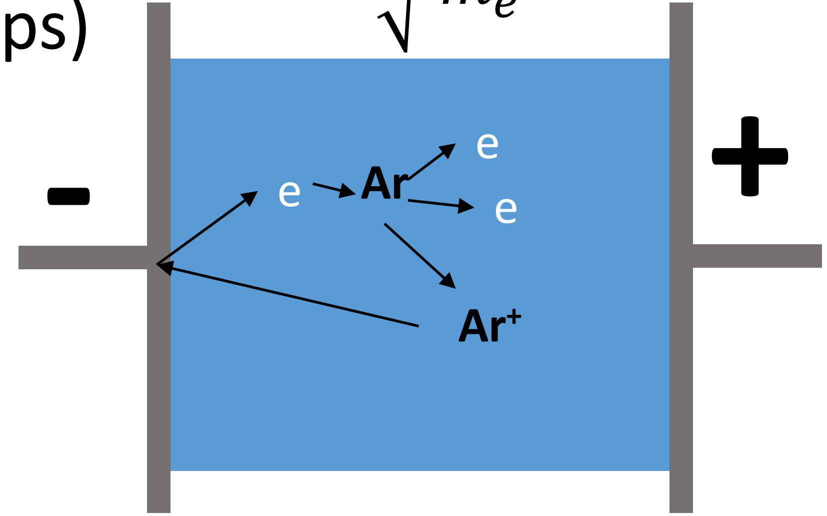
Verification: compare with PIC
Validation: compare with experiment

- Initiate with 0.625A/m^2
- Ionization
- Elastic collisions
- Excitation
- Secondary emission
- Energy gain = voltage
- SEY = 0.07 (Phelps)

$$v_{Ar} = \sqrt{2eV/m_{Ar}}$$

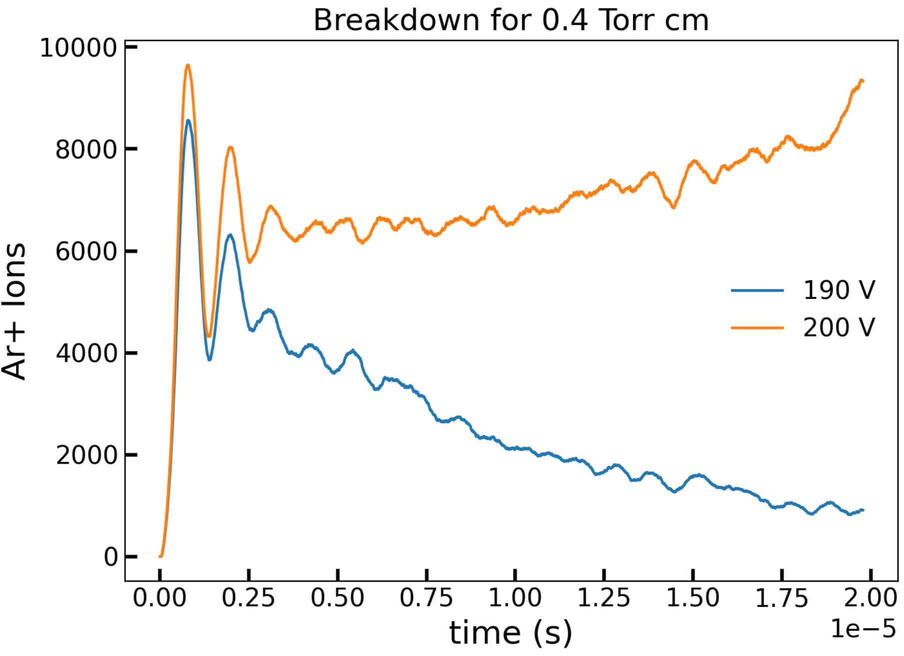
$$v_e = \sqrt{2eV/m_e}$$

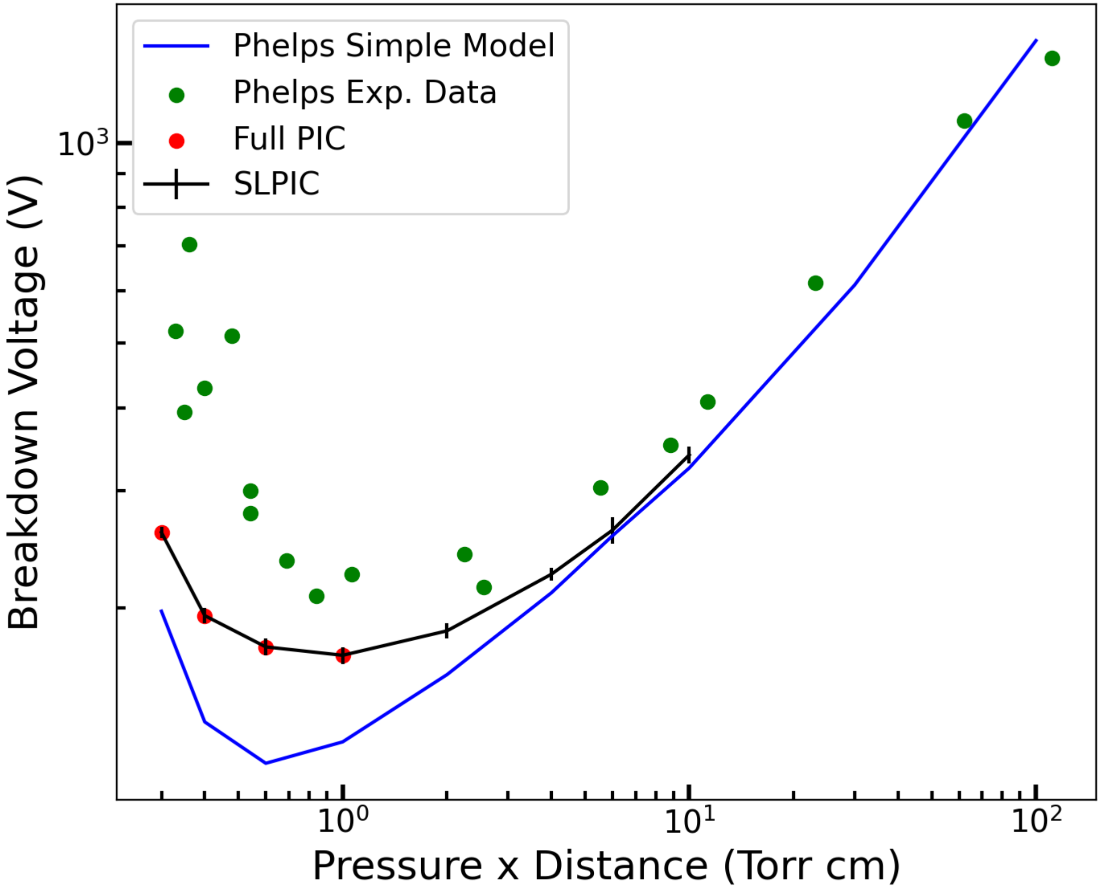
$$\sqrt{\frac{m_{Ar}}{m_e}} = 270$$



- Numerics - spatial
  - DSMC requires resolution of the mean free path
  - 1 Torr, 1 cm gap => 219 cells across gap
- Numerics – temporal
  - Simulate for 10 ion crossing times, look for exponential growth
  - Time step ~ cell size over max electron vel.
  - PIC: 4.7M time steps
  - SLPIC: *17.5k* time steps

Breakdown is determined by transit time of ions





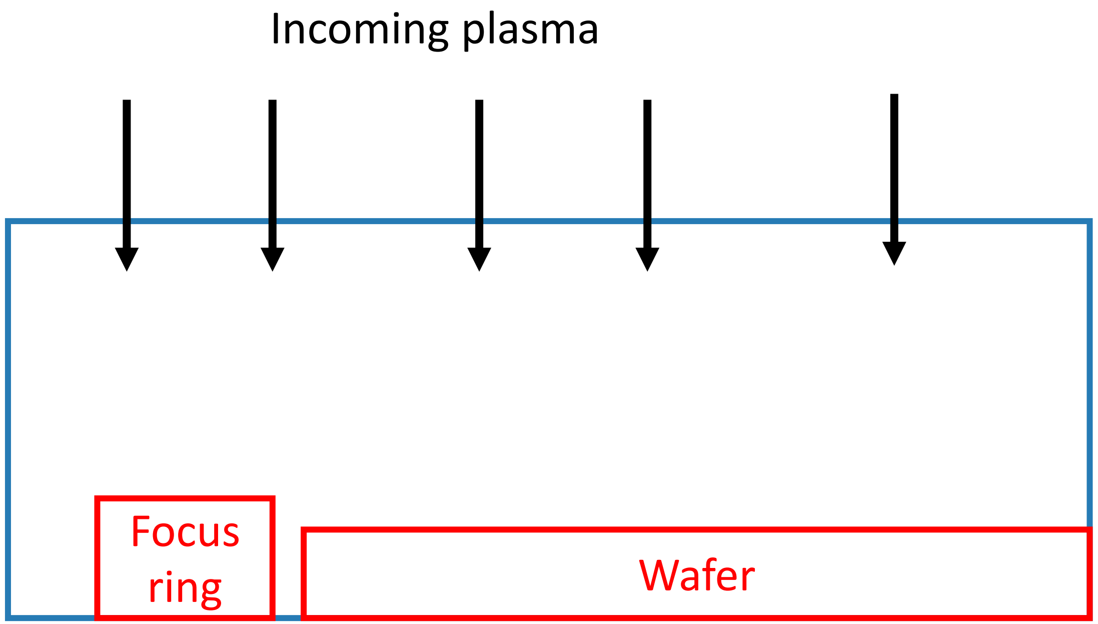
SLPIC gets the answer faster

At 1 Torr the run stats were:

- PIC: 2.3e6 steps 658 minutes
- SLPIC : 8.6e3 steps 4 minutes
- **165X Faster!**
- Not linear scaling with steps

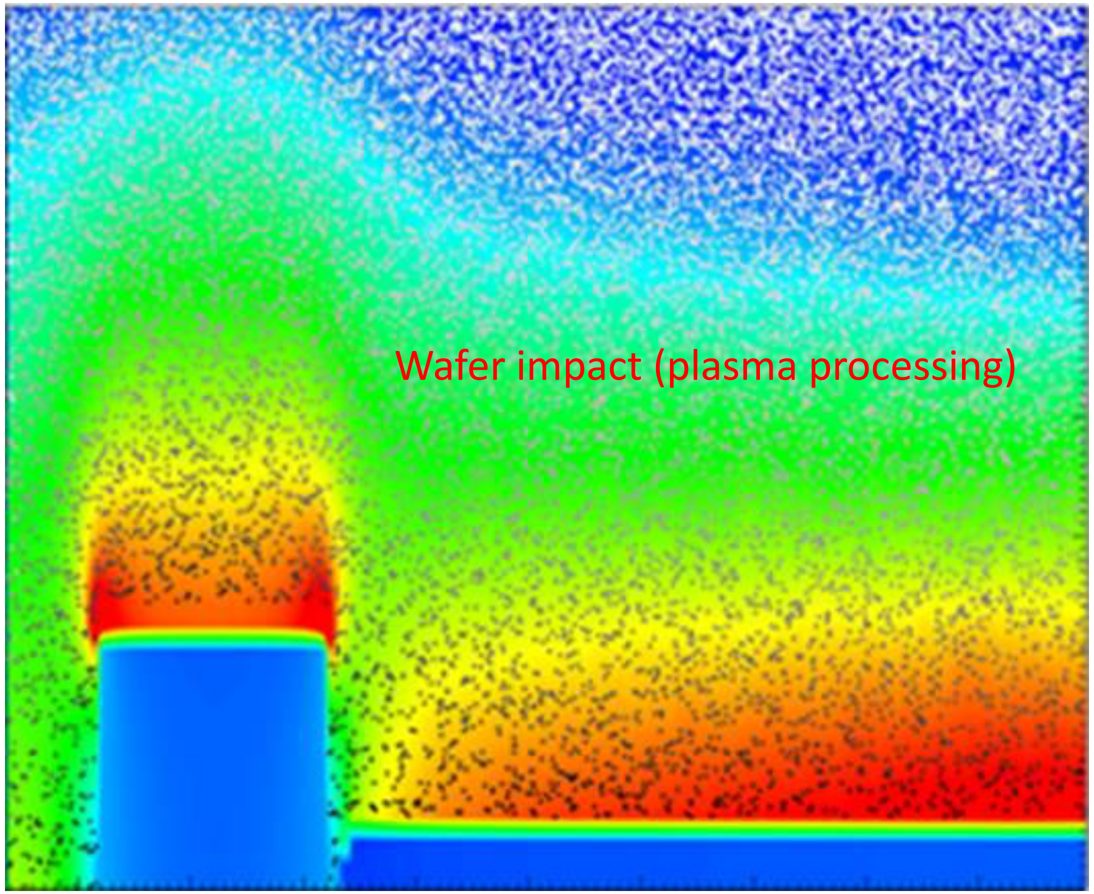
Demo: plasma (wafer)

- Want plasma uniformity to maximize wafer yield
- Edges of wafer see different fields, have different particle distributions
- Chips at edge not well formed
- Focus rings can be moved, but how does one explore an entirely new shape?
- Trick: how to treat above plasma as infinite?



VSim: Physical simulations of plasmas

- Particle capture on surfaces
- Collisional processes
- Implicit solvers
- Sputtering
- Secondary yield



Analysis and visualization will have many new features

- Multiple analysis tabs
- Persistence in input parameters
- Automatic field filling

Conclusions

- Plasma simulation is coming of age
  - Ease of use
  - Faster processing (parallelism, devices)
  - Better algorithms
- VSim is leading the way in bringing these advances to you