Kinetic Plasma Physics in the Ionosphere □

Meers Oppenheim

BU Summer School on Plasma Processes in Space Physics 2012

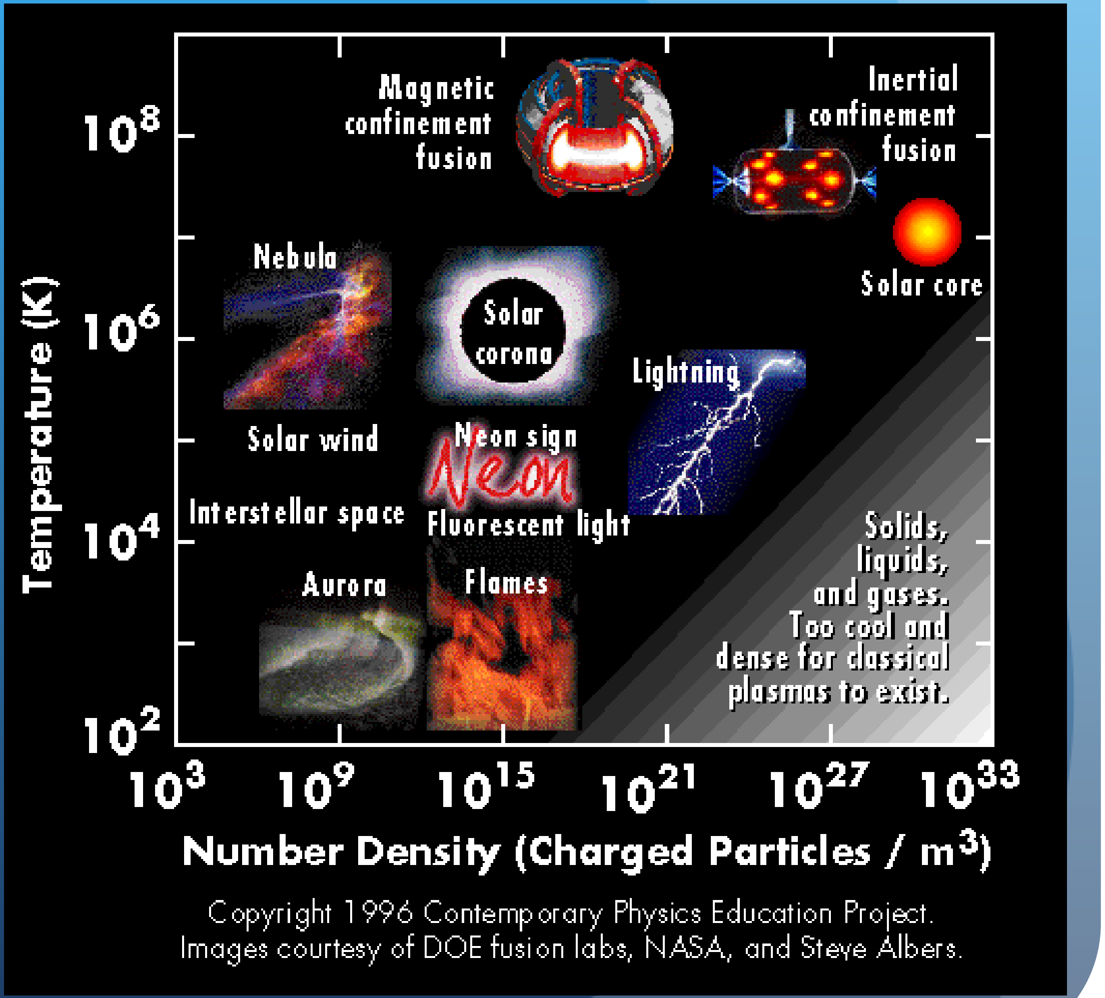
BOSTON UNIVERSITY

C S P

Center for Space Physics

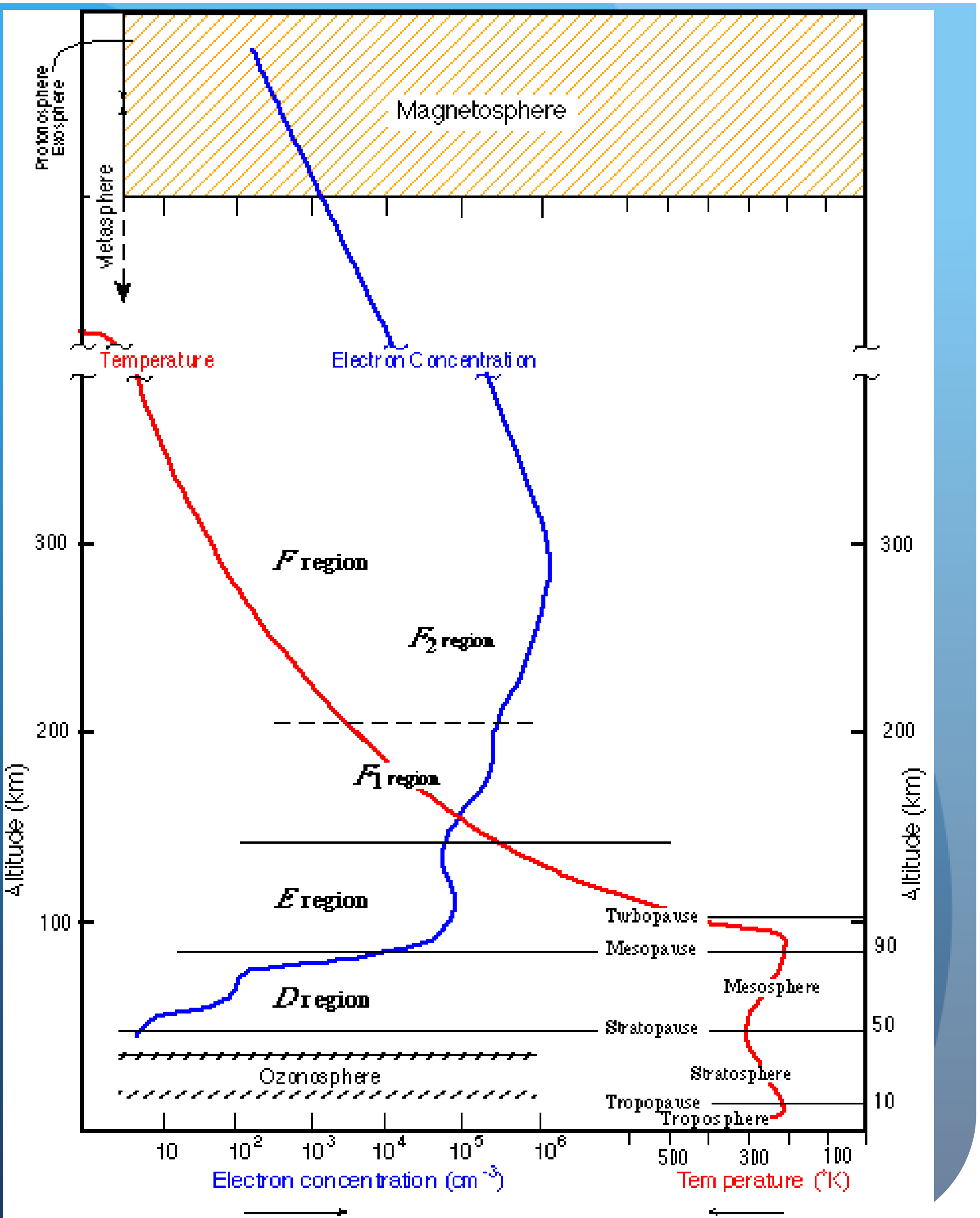
Talk Outline

- Plasmas in the Ionosphere
- Kinetic vs. Fluid Physics
- Plasma Instabilities
- Kinetic Simulations
- Example problems in Space Physics
- Limitations of these methods

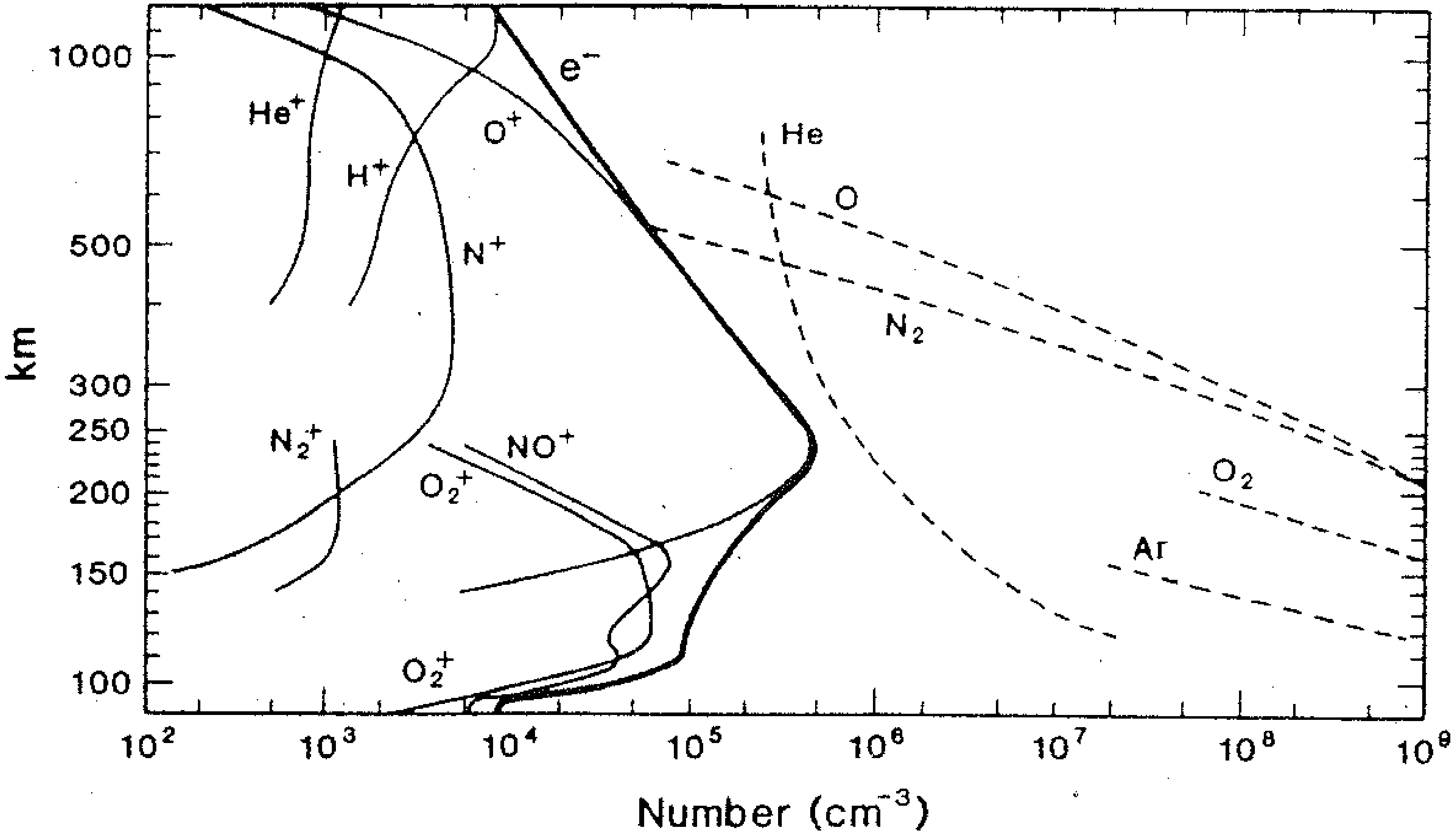


Copyright 1996 Contemporary Physics Education Project.
Images courtesy of DOE fusion labs, NASA, and Steve Albers.

Plasma in the Ionosphere



Ionosphere Plasma Composition



Ionosphere Plasma Density Variability

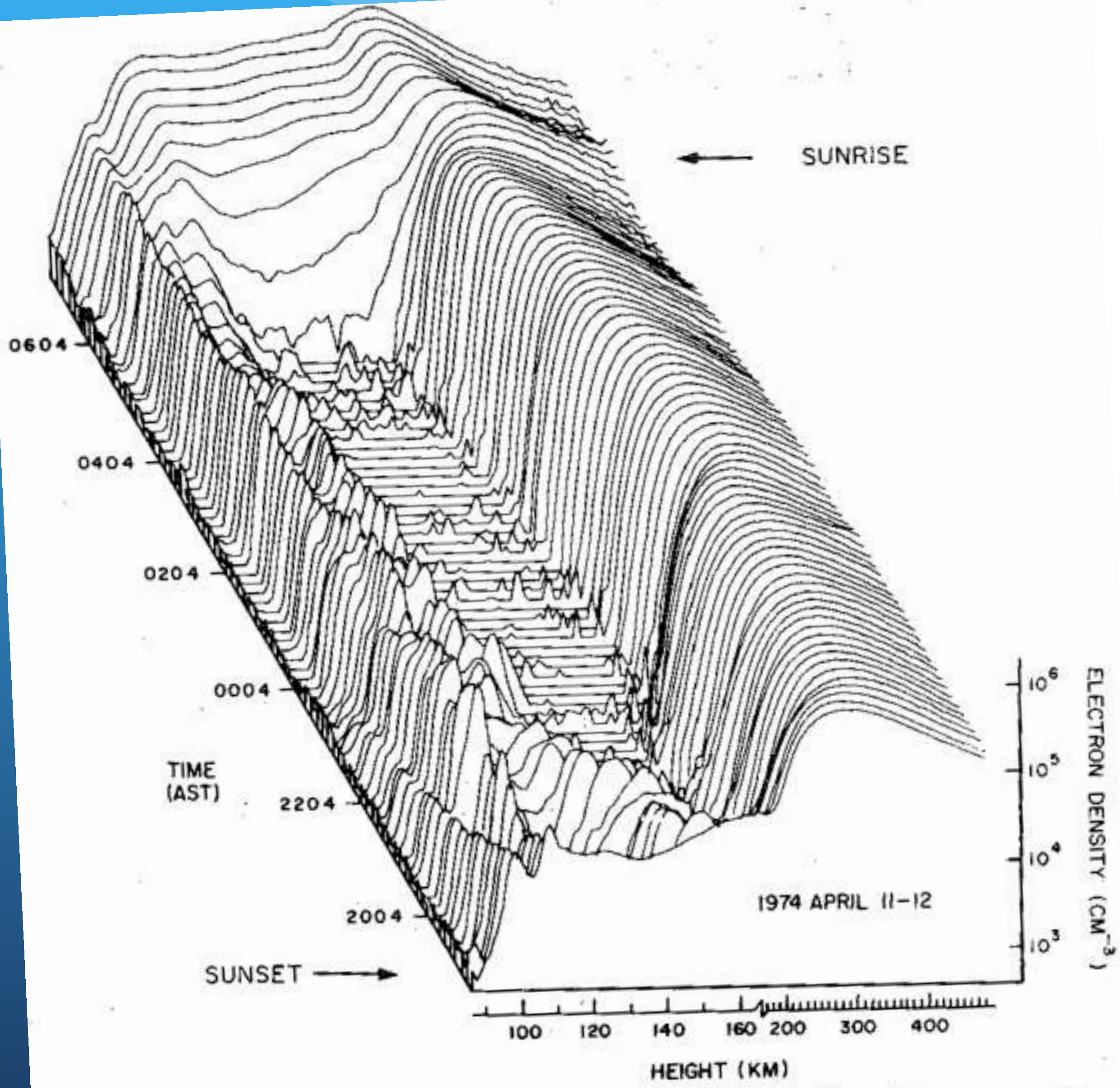
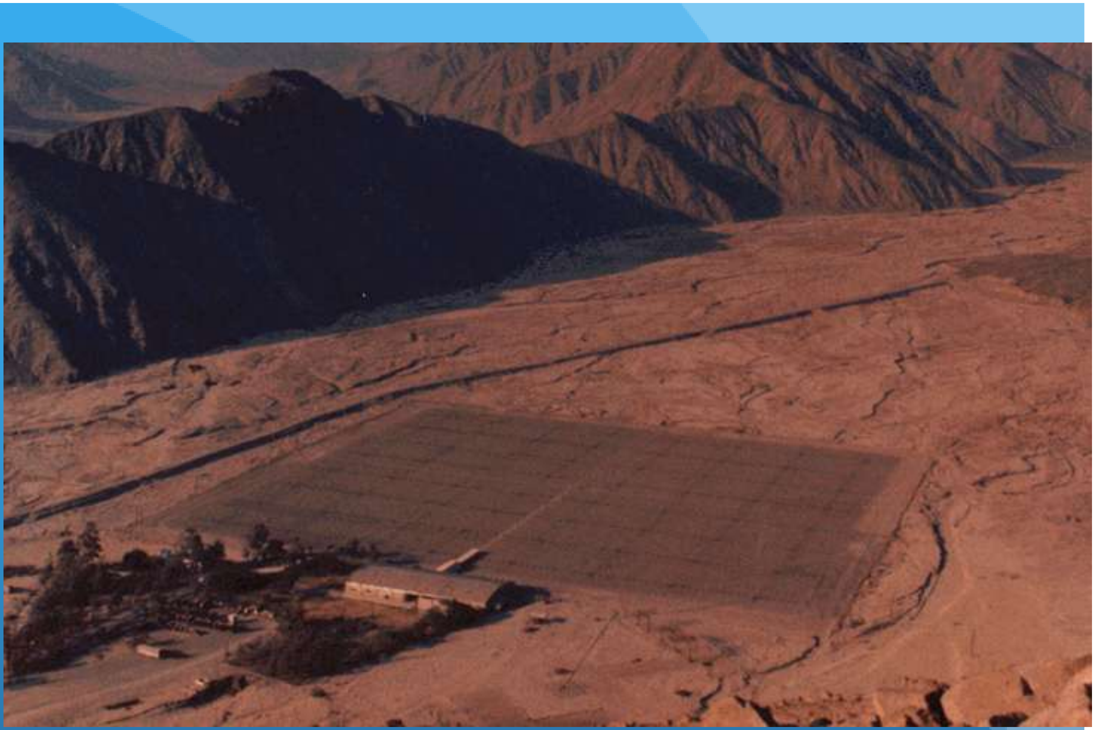


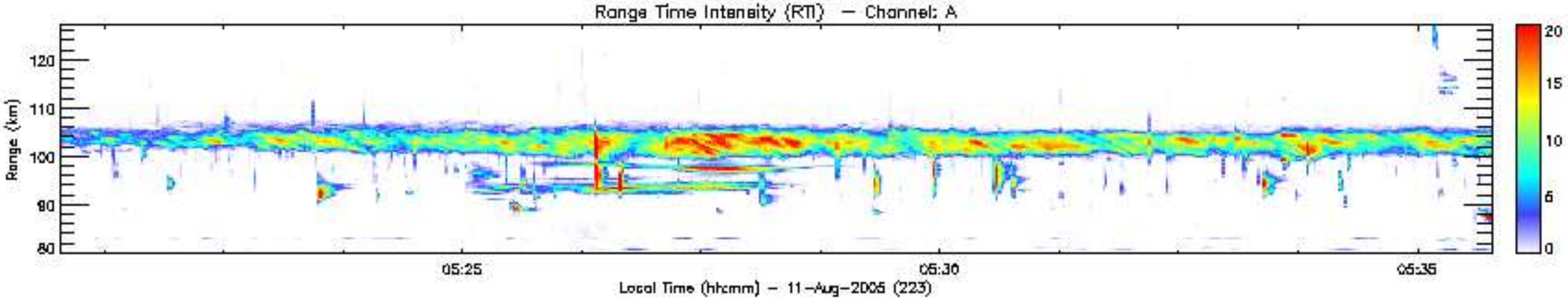
Fig. 1.3. Plasma density contours during a typical night over Arecibo, Puerto Rico. [After Shen *et al.* (1976). Reproduced with permission of the American Geophysical Union.]

Arecibo Incoherent scatter radar

Coherent Radar reflections from the Ionosphere

- Bragg Scatter: $\omega_{radar} >> \omega_p$
- Example:
    - Scatter off E-region ionosphere
        - ~90-130 km altitude
        - Electrojet irregularities
        - Meteor plasmas



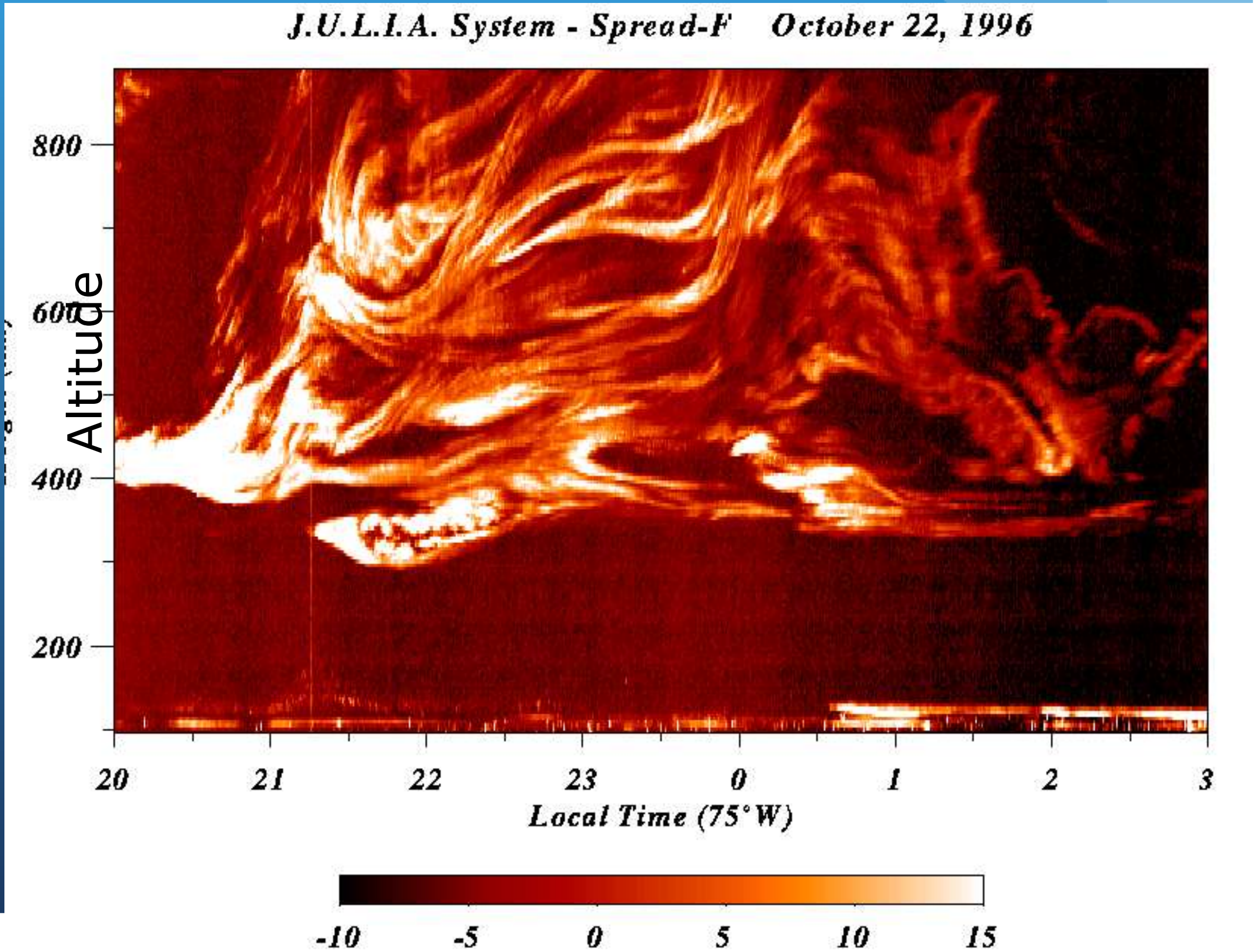

Range Time Intensity (RTI) — Channel: A
Range (km)
120
110
100
90
80
05:25
05:30
05:35
Local Time (hh:mm) – 11-Aug-2005 (223)
20
15
10
5
0

Radar measurements in the F region

Spread-F Turbulence:

Plasma Depletions which bubble up at night (sometimes)

Radar measurement of plasma density fluctuations



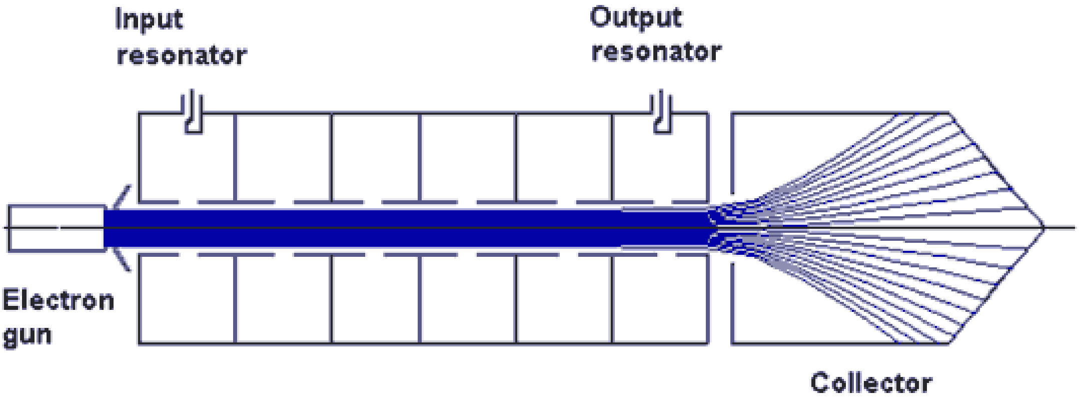
Plasma Physics Approaches

- Fluid Approaches
  - Cold
  - Warm
- MHD Approaches
  - Ideal
  - Resistive
  - Hall
- Kinetic Approaches
- What are the differences?

What are simulations?

- Views of nature:
  - Physicists think that the real world approximates equations.
  - Engineers think that equations approximate the real world.
  - Mathematicians don't care…
- Simulations are a mathematical description, or model, of a real system typically in the form of a computer program
- Simulations explore the behavior of systems too complex for analytical theory
  - Inhomogeneous systems
  - Nonlinear systems
  - Turbulence

First Plasma Particle Simulations: Klystrons



1939: Klystron inventors William Hansen and brothers Russell and Sigurd Varian examine early model

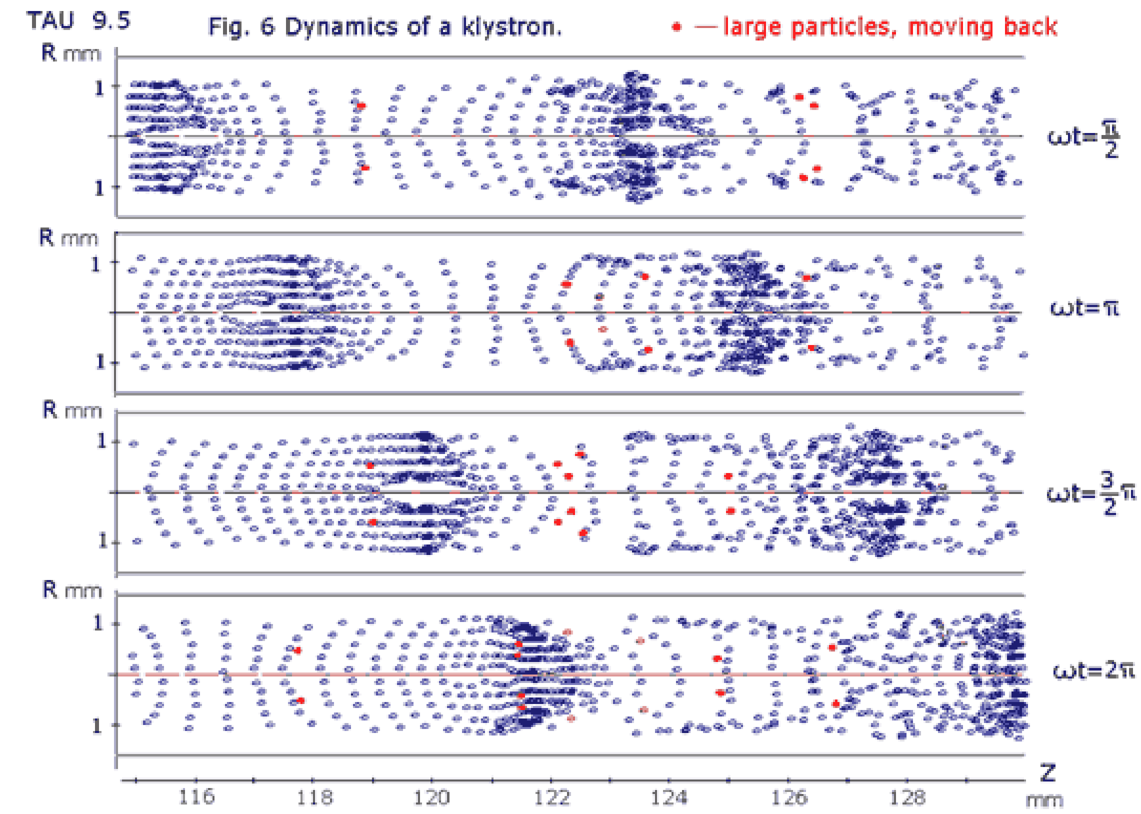
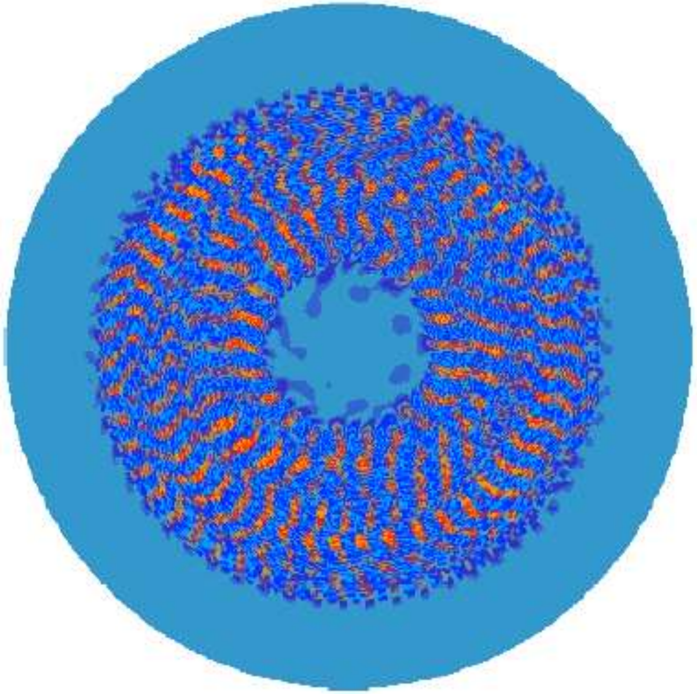


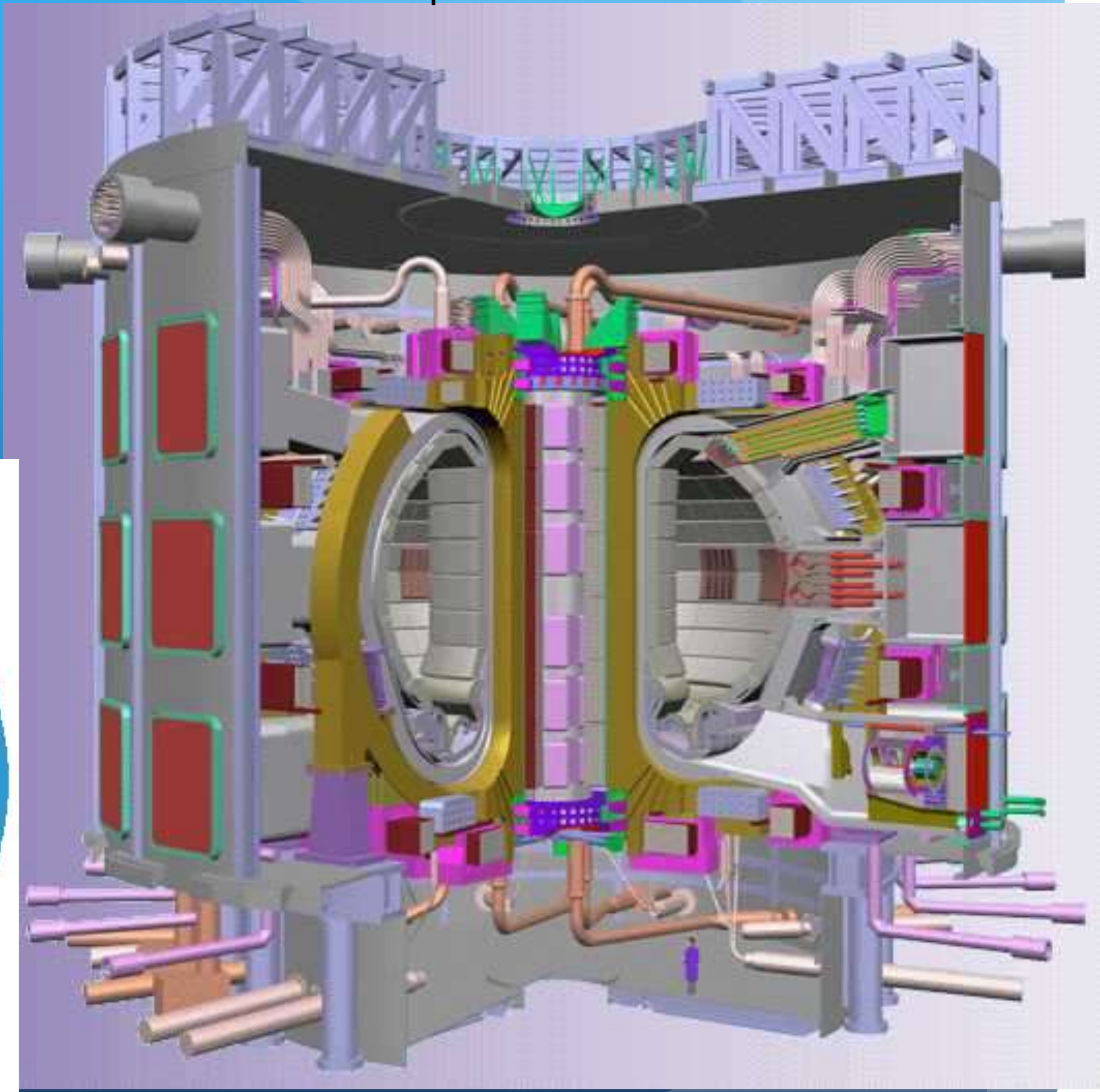
Fig. 6 Dynamics of a klystron.

© 2001 Solver Company Ltd, St.-Petersburg, Russia.

Fusion Energy Simulations



Proposed ITER Tokomak

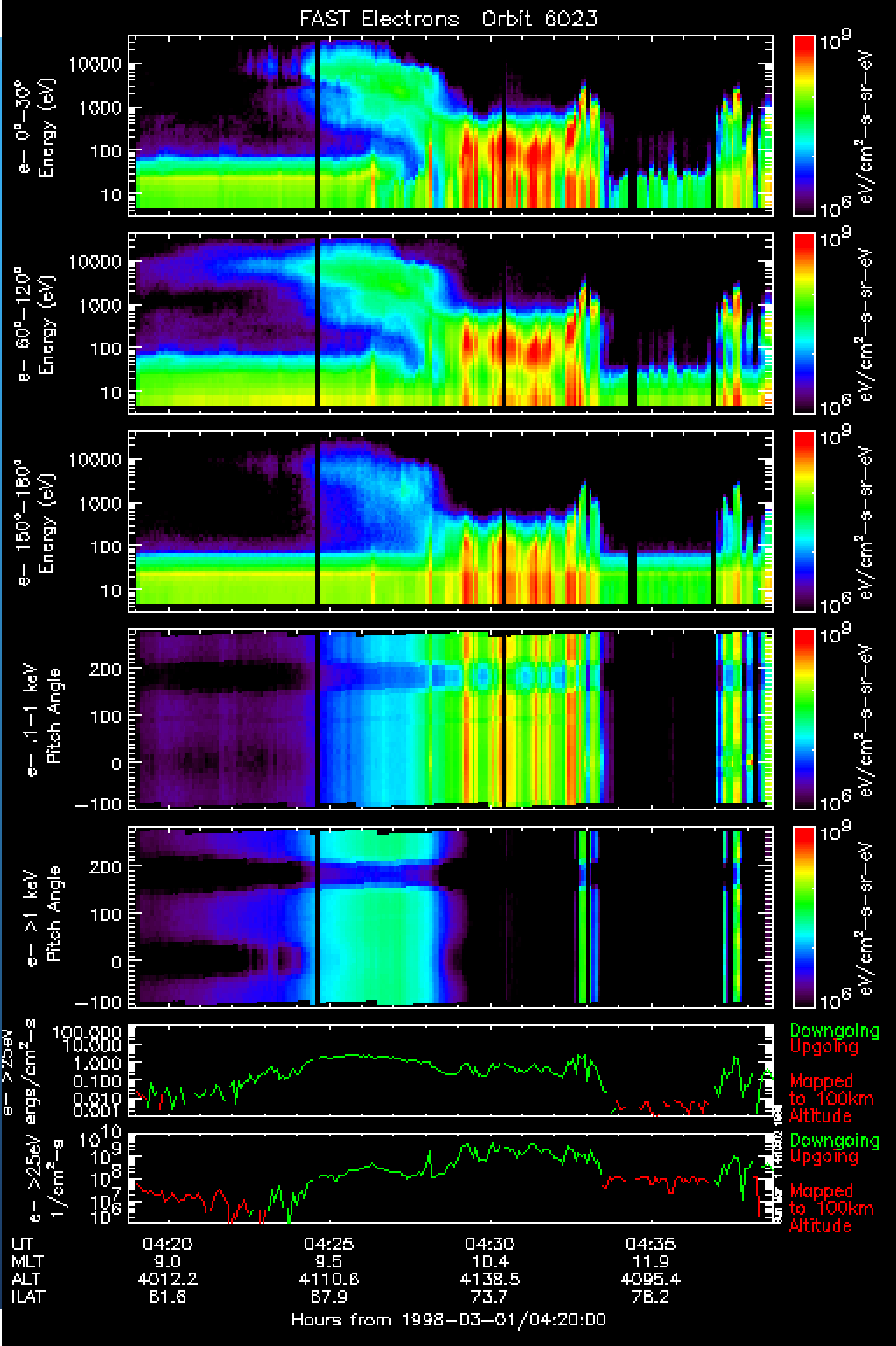


Temperature fluctuations from *Numerical Tokamak Turbulence Calculations on the CRAY T3E1* by Lynch, et al., Proceedings of the ACM/IEEE SC97 Conference (SC' 97)

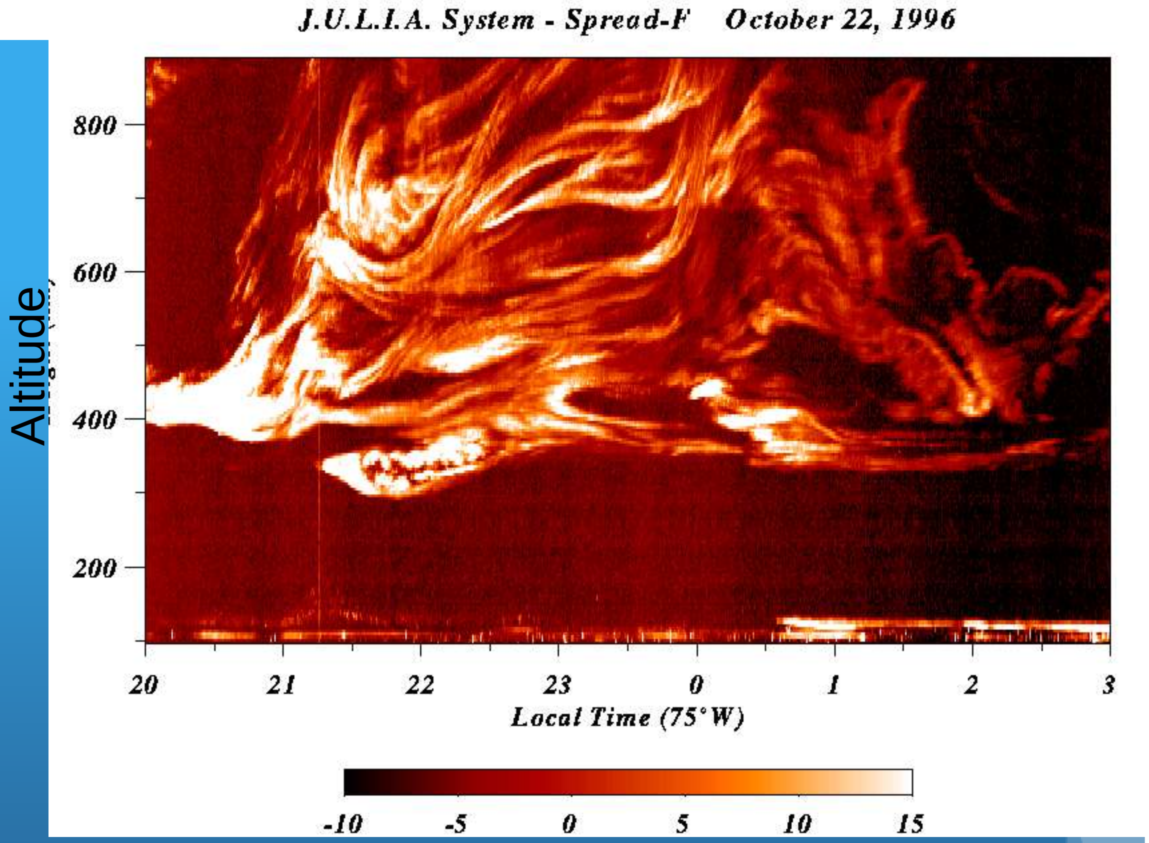
Where does one need simulations in Ionospheric Physics?

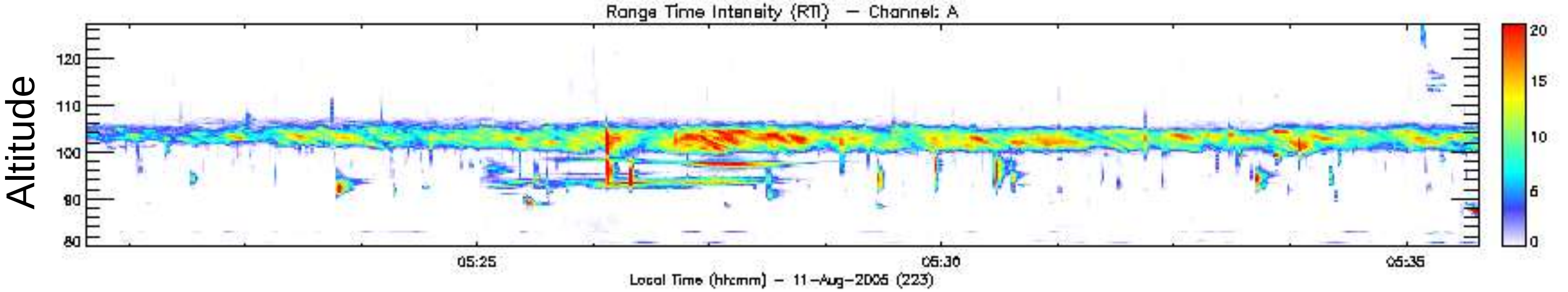
- The Auroral Ionosphere: Electrons accelerate from 3000-1500 km altitude by unknown mechanisms

FAST Spacecraft measures turbulent auroral plasmas



Radars Measure Electron Density Irregularities in Ionosphere





Plasma Theory in 5 Minutes

1. Charged particles create fields: *Maxwell's Equations*

$$\vec{\nabla} \bullet \vec{E} = \frac{e}{\varepsilon_0}(n_i - n_e) \qquad \vec{\nabla} \bullet \vec{B} = 0$$

$$\vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \qquad \vec{\nabla} \times \vec{B} = \mu_0 \frac{\partial \vec{E}}{\partial t} + \mu_0 \varepsilon_0 \vec{J}$$

2. Lorentz Force Accelerates Particles:

$$\frac{d\vec{v}_i}{dt} = \frac{q_i}{m_i}\left[E(\vec{x},t) + \vec{v}_i \times B(\vec{x},t)\right]$$

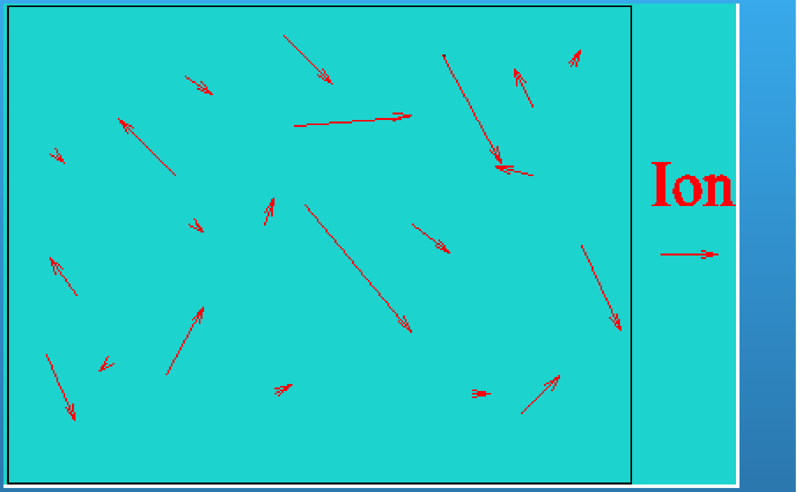
3. Equation of Motion

4. Collisions deflect particles (important in the lower ionosphere and other regimes)

Too many particles – Need simplifications!

Particle Simulations

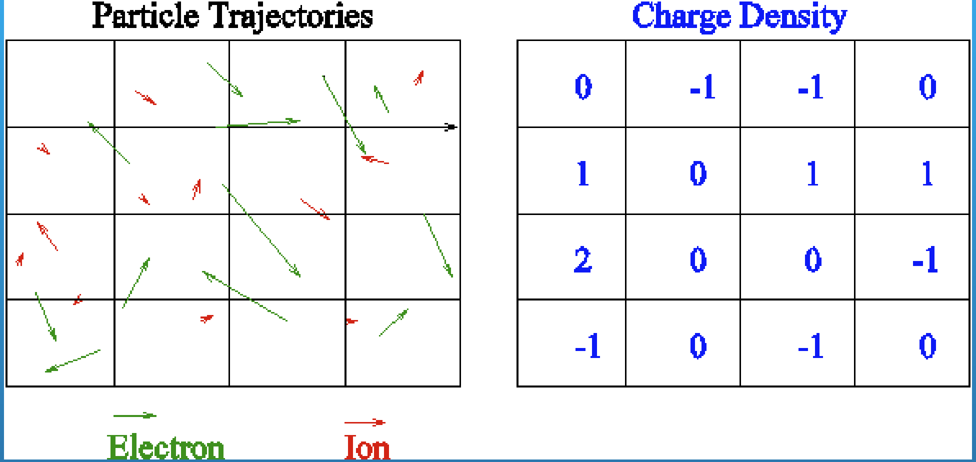
- Particles move within a box:
  - Position: $\boldsymbol{x}_i$
  - Velocity: $\boldsymbol{v}_i$
- Particles generate fields which accelerate other particles
- Too Slow! Speed proportional to the number of particles squared.



$$\vec{F}_{ij} = \frac{q_i q_j}{4\pi\varepsilon_0(\vec{x}_i - \vec{x}_j)^2}$$

Electrostatic Kinetic Simulation Method: Particle-In-Cell

1. Gather to determine charge density, ρ



0	-1	-1	0
1	0	1	1
2	0	0	-1
-1	0	-1	0

$$r(x) = \underset{particles}{\text{å}} q_i d(x - x_i)$$

2. Calculate Electric field:

$$\vec{\nabla} \bullet \vec{E} = \rho / \varepsilon_0$$

3. Update velocities:

$$\frac{d\vec{v}_i}{dt} = \frac{q_i}{m_i}\left[E(\vec{x}_i,t) + \vec{v}_i \,´\, B(\vec{x}_i,t)\right]$$

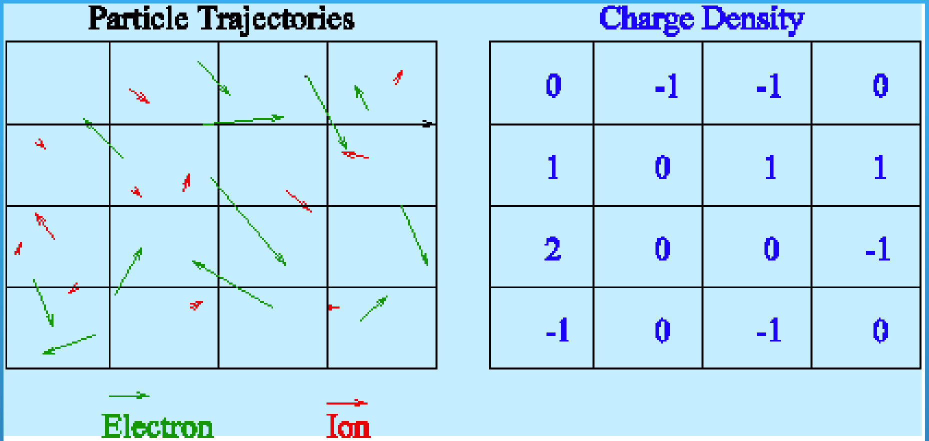
4. Update Positions:

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

5. Collide particles with neutrals

6. Go to Step 1

Assumptions made by PIC



Charge Density

0	-1	-1	0
1	0	1	1
2	0	0	-1
-1	0	-1	0

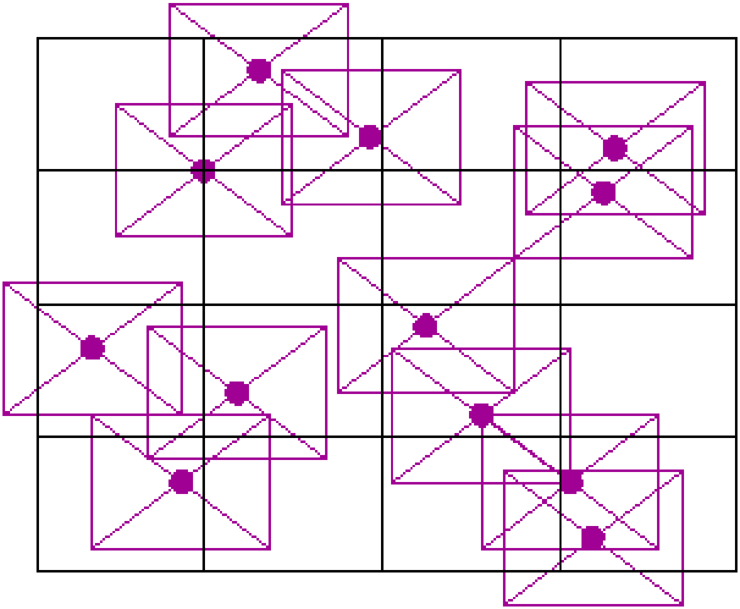
- Short range interactions eliminated
  - Simulators with a meshes cannot model behavior smaller than the mesh
  - Features must be bigger than the mesh
- Each PIC particle models the behavior of more than 10^6 real particles
- Fluid Simulators also use a mesh
  - Only one velocity in one location (unlike kinetic simulators)
  - Misses some physics but is less costly (per cell)
- Full kinetic physics represented
  - Particle trapping – resonant acceleration
  - Landau damping – resonant wave damping

One Problem with PIC

- Particle noise from limited numbers of particles
  - Random walk statistics:
  - Example n=144 particles/cell -> σ_n=8.3%
  - Fixes:
    - Nature reduces this through electrostatic shielding
    - Use non-point particles
    - Use millions and millions of particles
    - Use super computers!

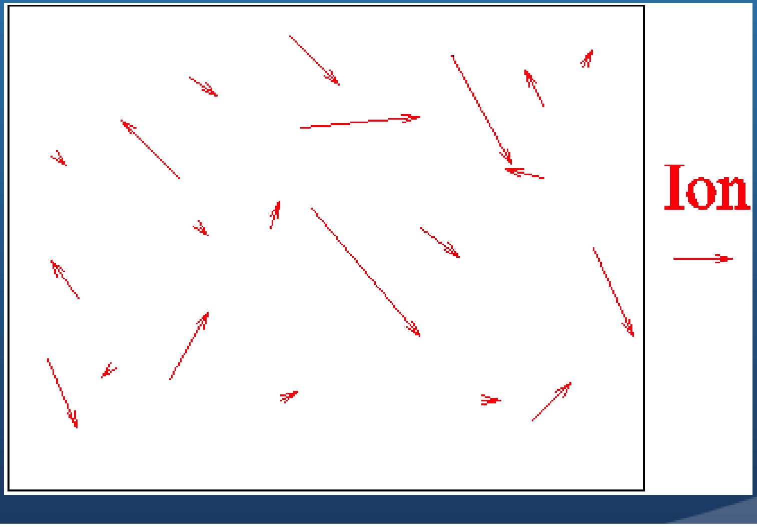
$$\sigma \propto \sqrt{n_{particles/cell}}$$

Macro-particles fill a volume



Boundary Conditions (BC)

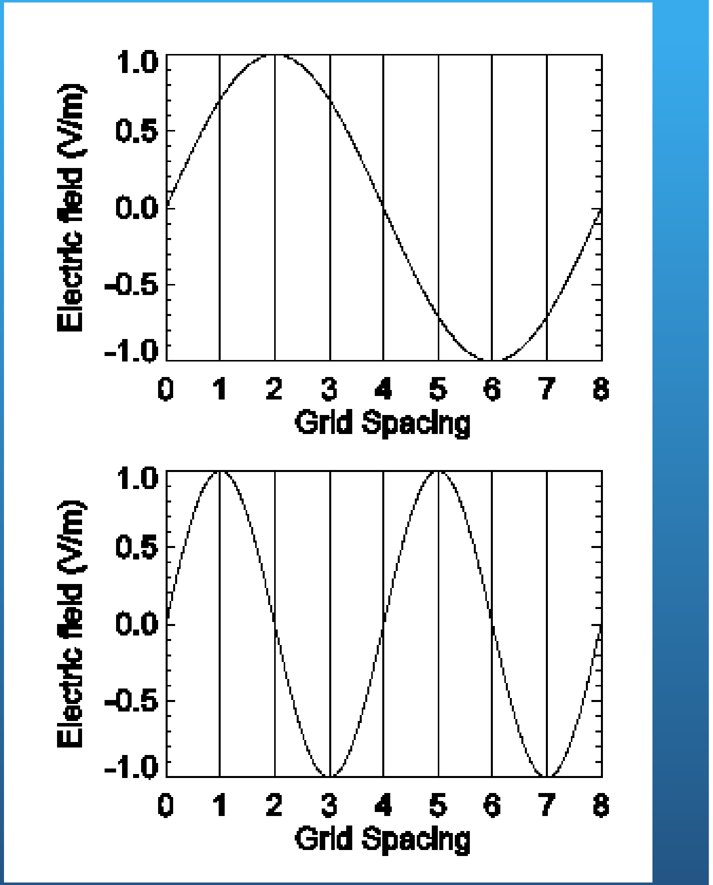
- Simulations of all types require BC
- BC introduce limitations and, sometimes, error
- Example: *Periodic* is the simplest BC
  - The right side connects to the left
  - The top to the bottom
  - Particles leaving the Left reenter on the Right and visa versa
  - Particles leaving the top -> bottom …



Boundary Conditions Cause Limitations

- Periodic boundaries quantize the simulation:
  - Only a full wave or
  - Integer multiples allowed
  - Simulations must not focus on waves spanning the system
- Other BC have other issues
- True in fluid simulators as well

Example in 1D



PIC Code…

```
 //  Read parameters from the input file:

  infile(argv[1]);

  //  Initialize the dynamic variables :

  init_misc();

  init_particles(pic, w, misc);

  init_fluid(fspecie, pic);

  init_field(Efield, rho);

//  Calculate the charges and currents on the grid.

  charges(rho, pic, fspecie, 0);

  //  Find the electric field on the grid at t=n:

  efield(Efield, rho);

 // Output any initial diagnostics:

 output (argv[1], pic, fspecie, Efield, rho, misc, w, it);
```

```
// Main timestep loop:

 for (it = it0; it <= nt; it++) {

// Apply the standard leapfrog method

    leapadv_subcycle(pic, fspecie, rho, Efield, w, misc);

   //Deal with any Boundary condition issues

   boundary(pic, Efield, w, misc, it);

   // Output data, diagnostics and restart:

   output (argv[1], pic, fspecie, Efield, rho, misc, w, it);

   }

 }//  End of main timestep loop
```

Charges.cc & density.cc

```
void charges(FArrayND &rho, particle *pic, fluid
      *fspecie, int it) {

  rho = 0.;

  for (int id=0; id<ndist; ++id) {

// Density returns the charge density of each species.

      density(den, id, pic, fspecie, qd[id]);

      rho += den;

} /* charges */
```

```
void density(FArrayND &den, int id, particle *pic, fluid
      *fspecie, FTYPE scaler) {

gather(den, INDICIES(pic[id].x, pic[id].y, pic[id].z),
      scaler*pic[id].n0);

}
```

Gather.cc

```
// 1-D Gather
void gather(FArrayND &den, PTYPEAVec &x, FTYPE n0)
{
  den=0;
  //  For each particle ...
  for (i = 0; i < np; ++i) {
    // Define the nearest grid points:
    ixl = (int) x(i);
    ixh = ixl + 1;
    if (ixh == nx) ixh = 0;
    // and the corresponding linear weighting factors:
    wxh = x(i) - ixl;
    wxl = 1. – wxh;
    // Add this particle's contribution to den:
    den(ixh) += wxh;
    den(ixl) += wxl;
  } // end for (i = 0; i < np; ++i)
  // Express in physical units:
  den *= nscale;
} // End 1-D gather
```

Field Solvers

- Electrostatic: Gauss Law

$$\nabla \bullet \vec{E} = \rho / \varepsilon_0 \Rightarrow \nabla^2 \phi = -\rho / \varepsilon_0$$

- How to solve on a mesh?
  - Spectrally:
    - Fourier Transform density, ρ

$$F(r) = \tilde{r}$$

    - Solve for Fourier Transformed potential

$$-k^2 \tilde{f} = -\tilde{r} / e_0 \text{ Þ } \tilde{f} = \tilde{r} / e_0 / k^2$$

    - Inverse transform potential

$$F^{-1}(\tilde{f}) = f$$

- Finite Difference
  - In 1D, qt the mesh point i, solve for ϕ_i,

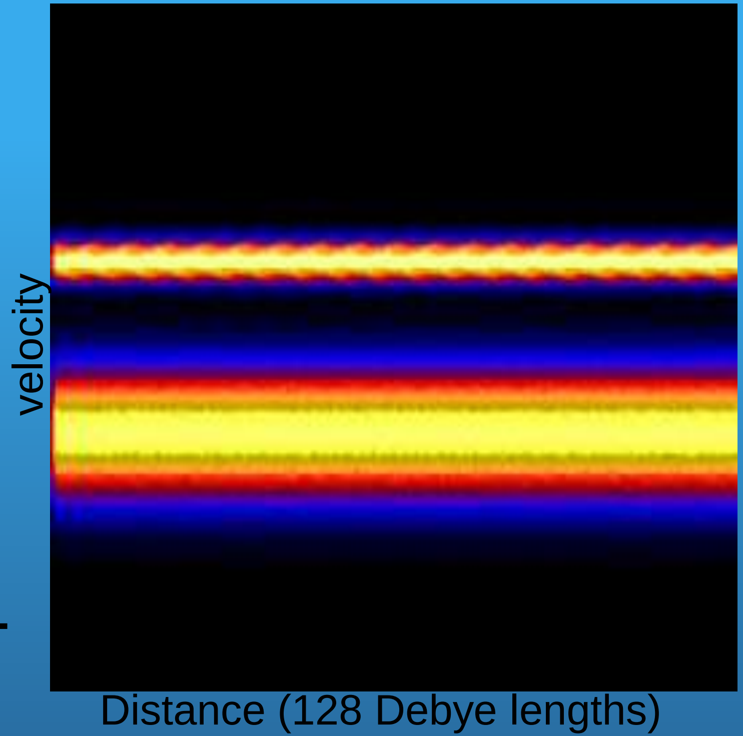
$$f_{i-1} - 2f_i + f_{i+1} = r_i / e_0$$

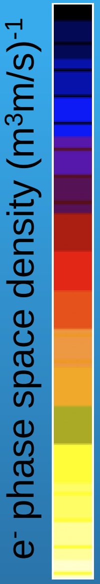
  - Requires Matrix Solve

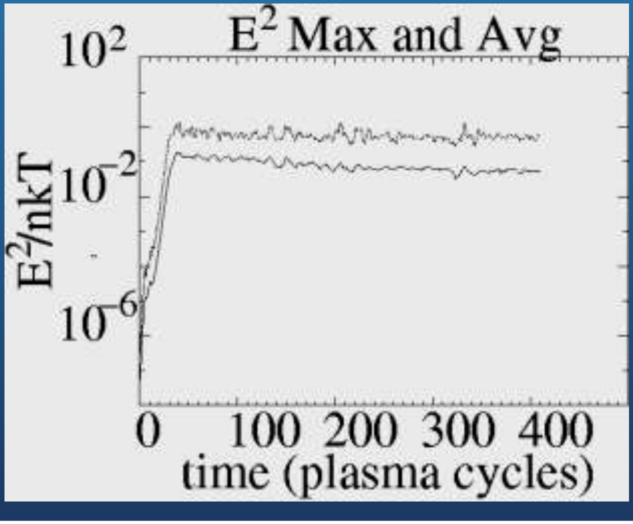
- Electromagnetic:
  - Leapfrog E and B on the mesh
  - Other Methods?

Example: 1D electron two-stream Instability

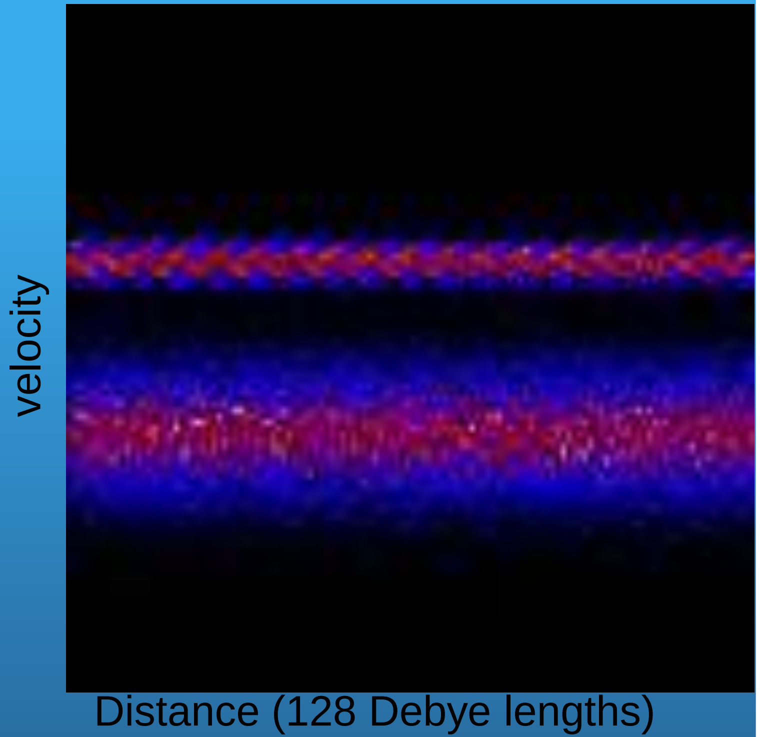
~1 Million particles

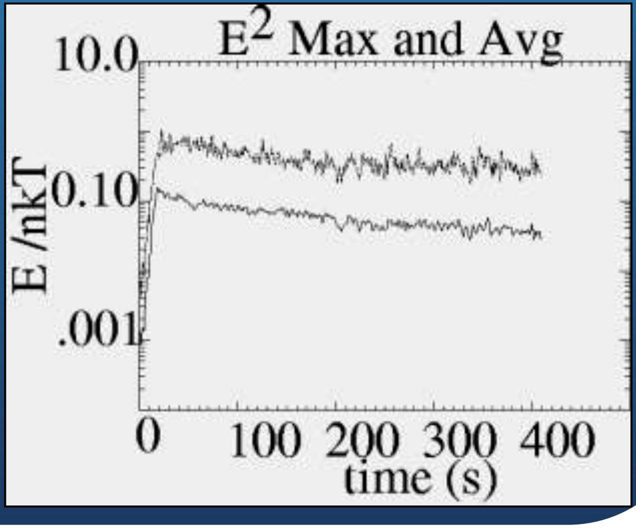




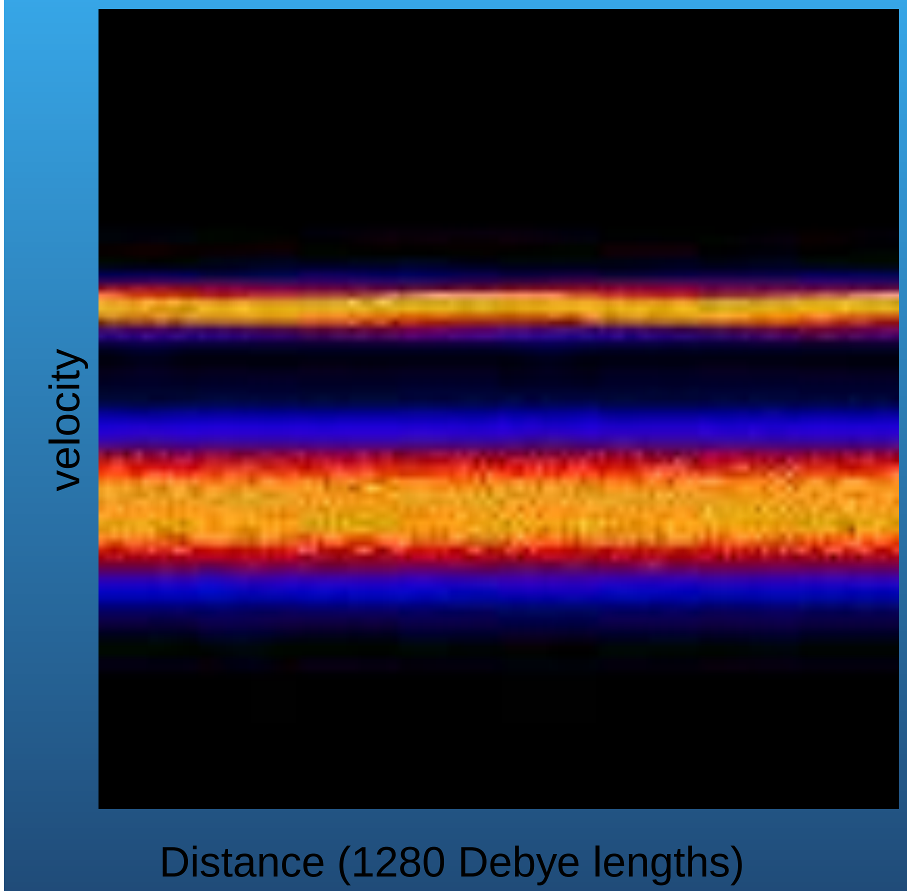


~1/64 Million particles

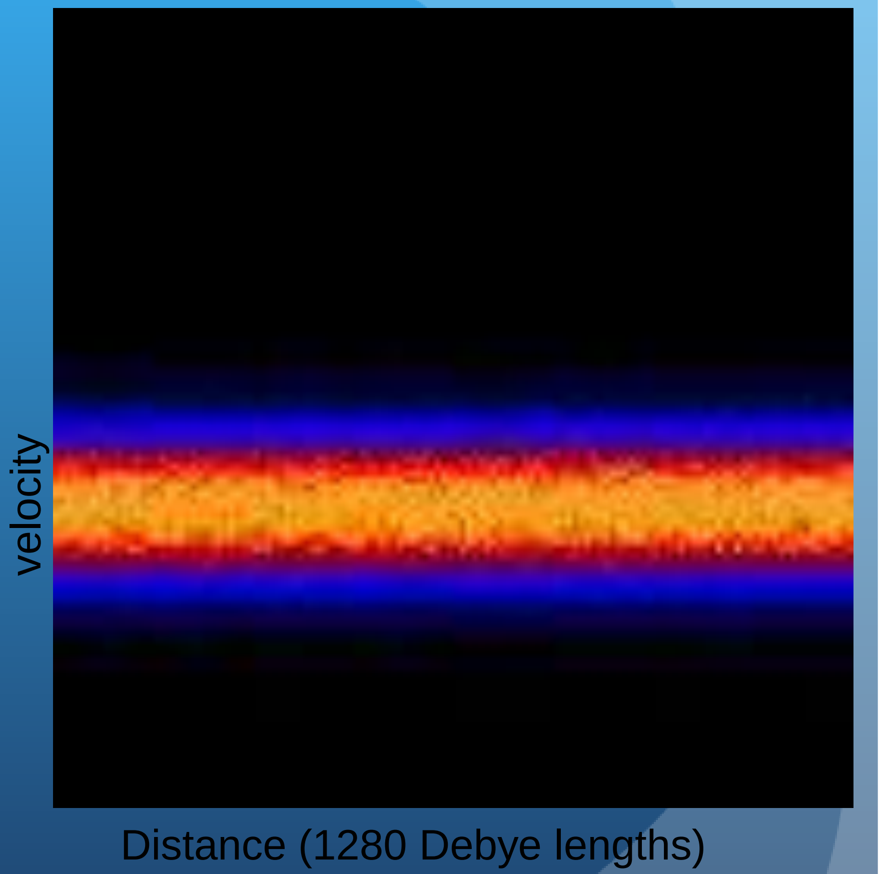




Expand grid spacing 10X



Eliminate Beam



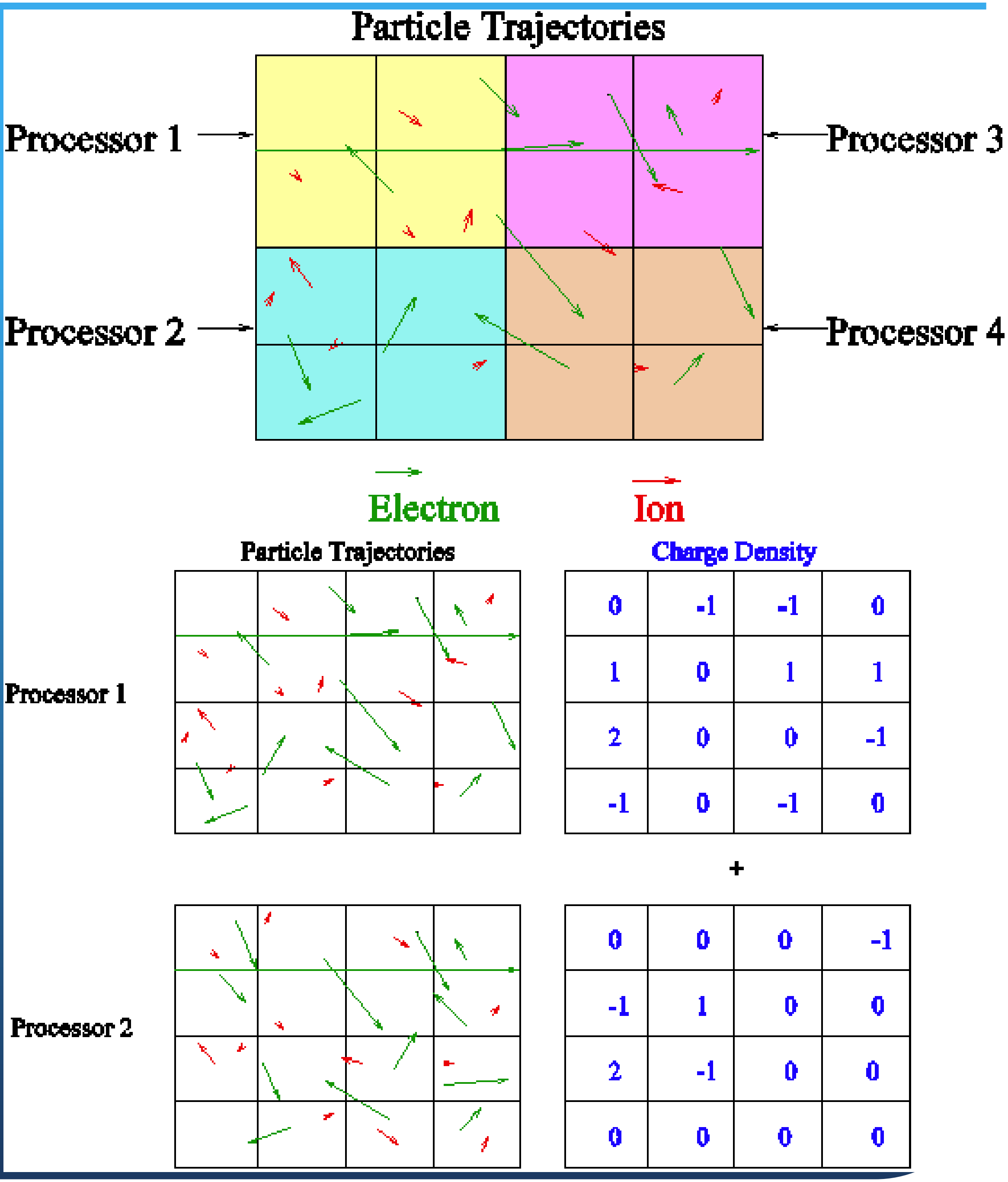
(8x longer simulation in time,
shown 16x as fast)

Simulation Limitations

- Systematic:
  - Do the equations represent the physics?
  - Do they resolve the important scales?
- Numerical:
  - Stability
  - Accuracy

Solution: Parallel Supercomputing

- Domain Decomposition
- Mesh Parallelization



Charge Density

Processor 1

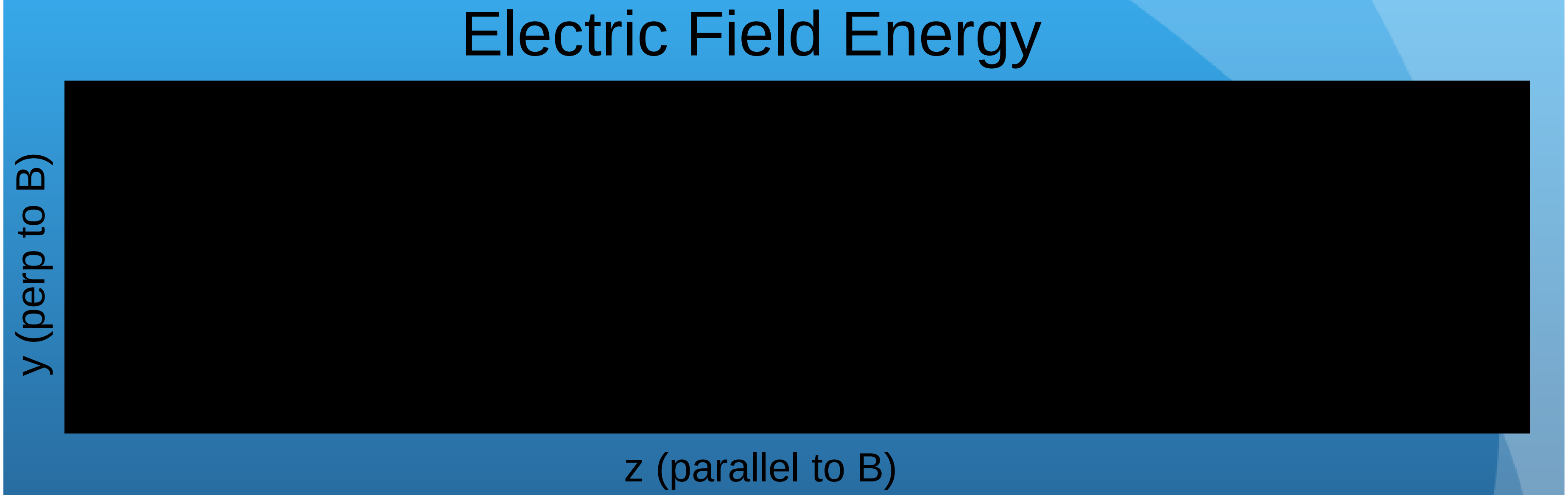
0	-1	-1	0
1	0	1	1
2	0	0	-1
-1	0	-1	0

+

Processor 2

0	0	0	-1
-1	1	0	0
2	-1	0	0
0	0	0	0

Electron Holes in 2D

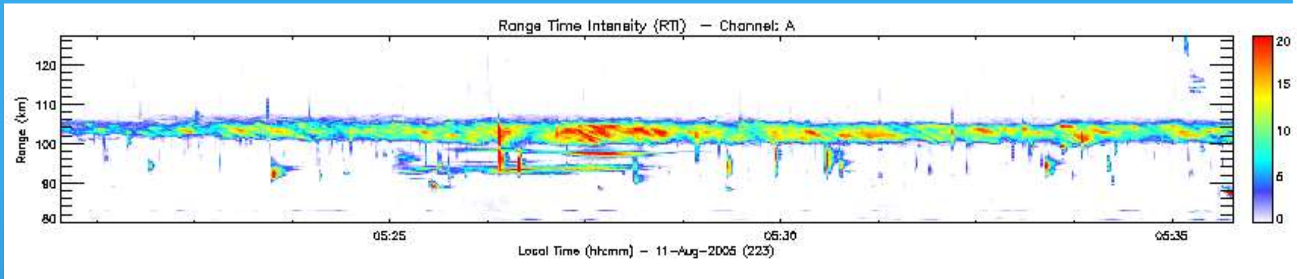


These simulations enabled us to:

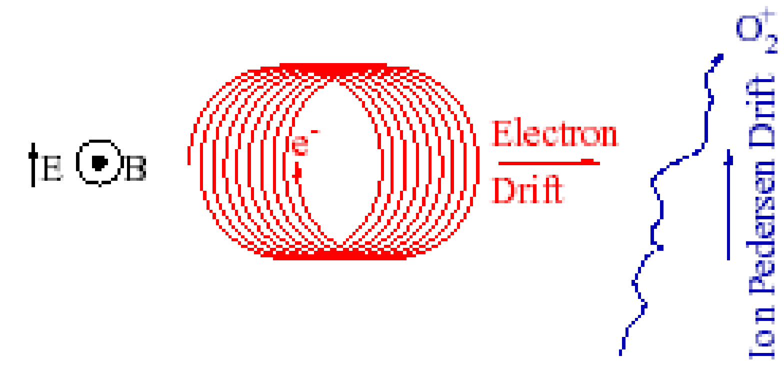
- Understand plasma evolution
- Study energy and momentum coupling
- Characterize Turbulence

Electrojet Waves

Radar Returns from Electrojet

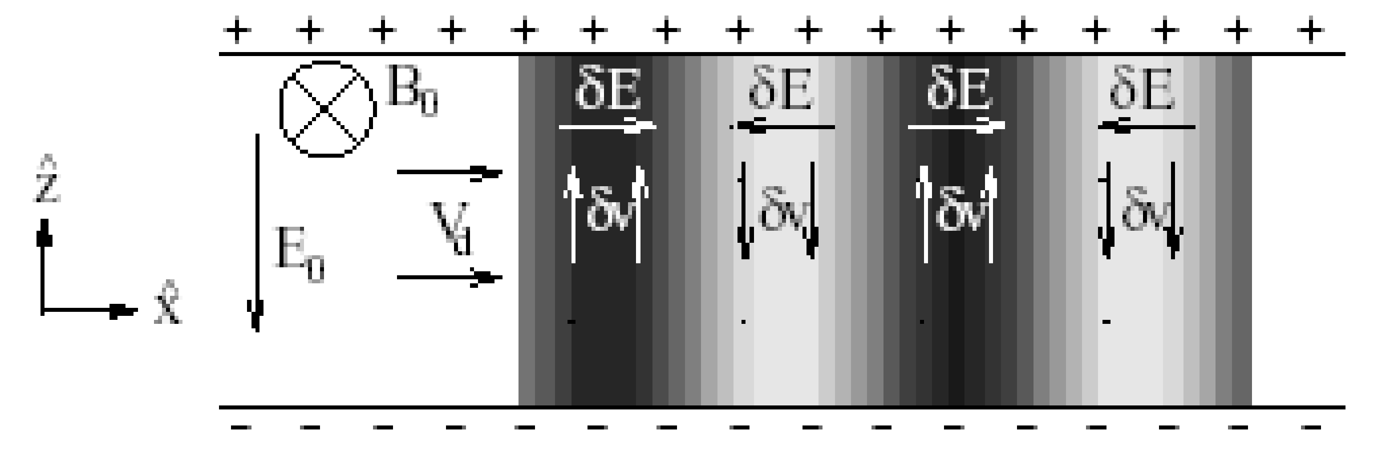


- Ions & Electrons respond differently to fields
  - Electrons remain magnetized: ***ExB*** drift
  - Ions demagnetized by collisions: flow along ***E***

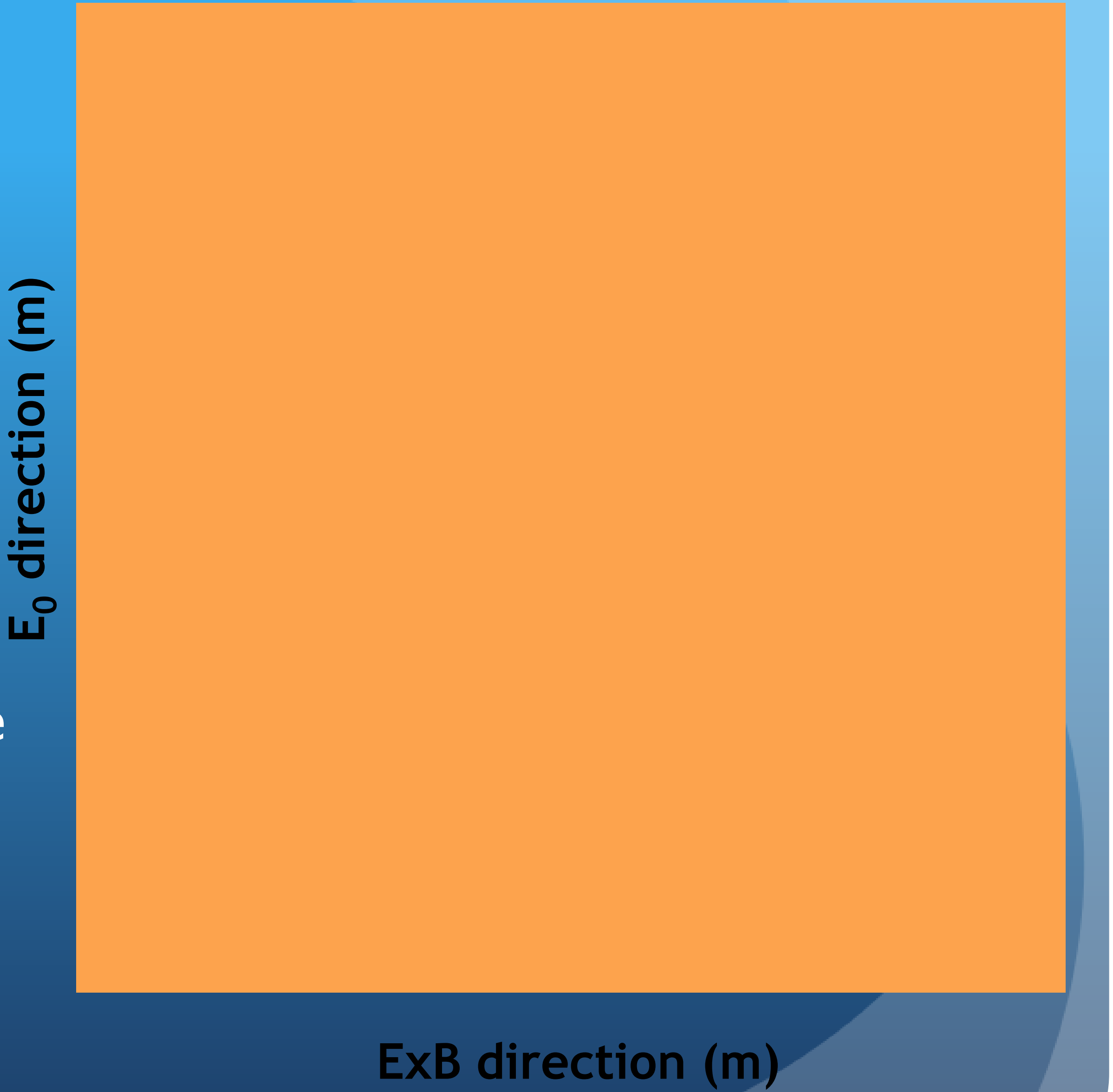


- If $V_e > C_s$, streaming instability develops

Modified two-stream or Farley-Buneman Instability



Electrojet PIC Simulation

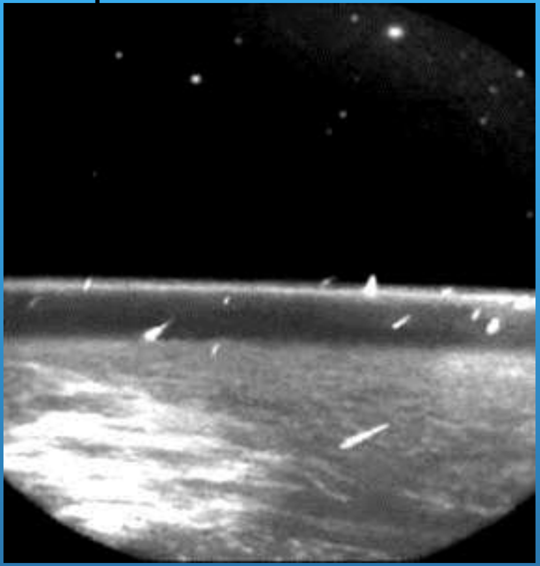


New thing learned:

- Saturation though Mode coupling
- Saturated wave speed
- Average Tilting of Wave
- Thermal Behavior

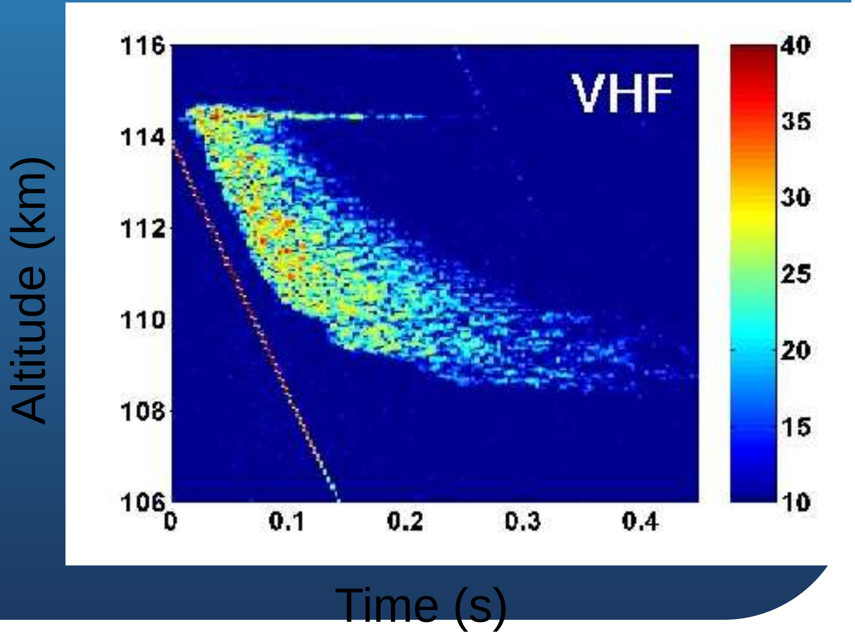
Meteor Plasma waves

Leonids picture from the shuttle

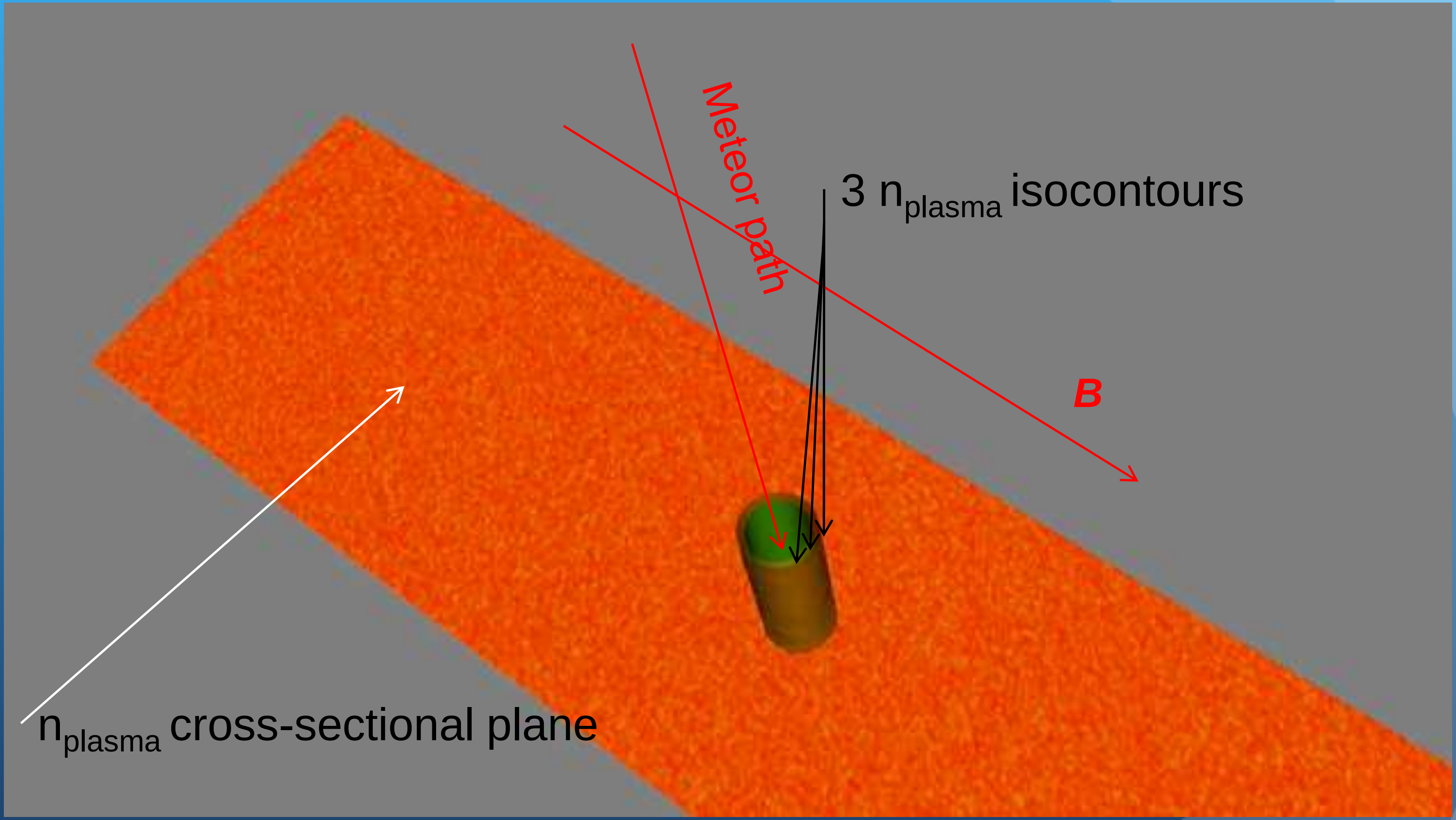


Large Aperture
Radar Detection
of a Meteor

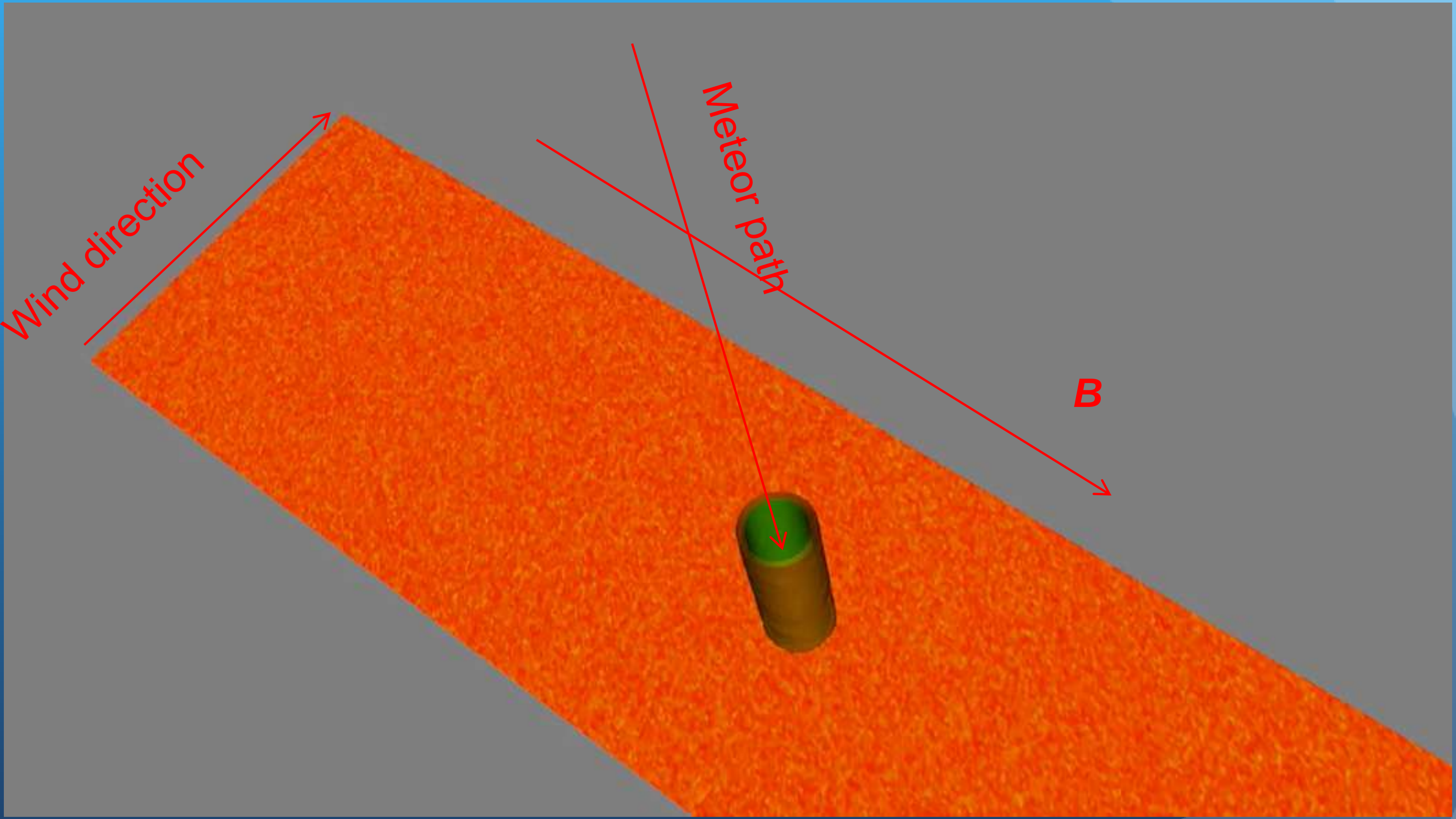
ALTAIR meteor detection



Particle in Cell Simulations of Meteor Plasma

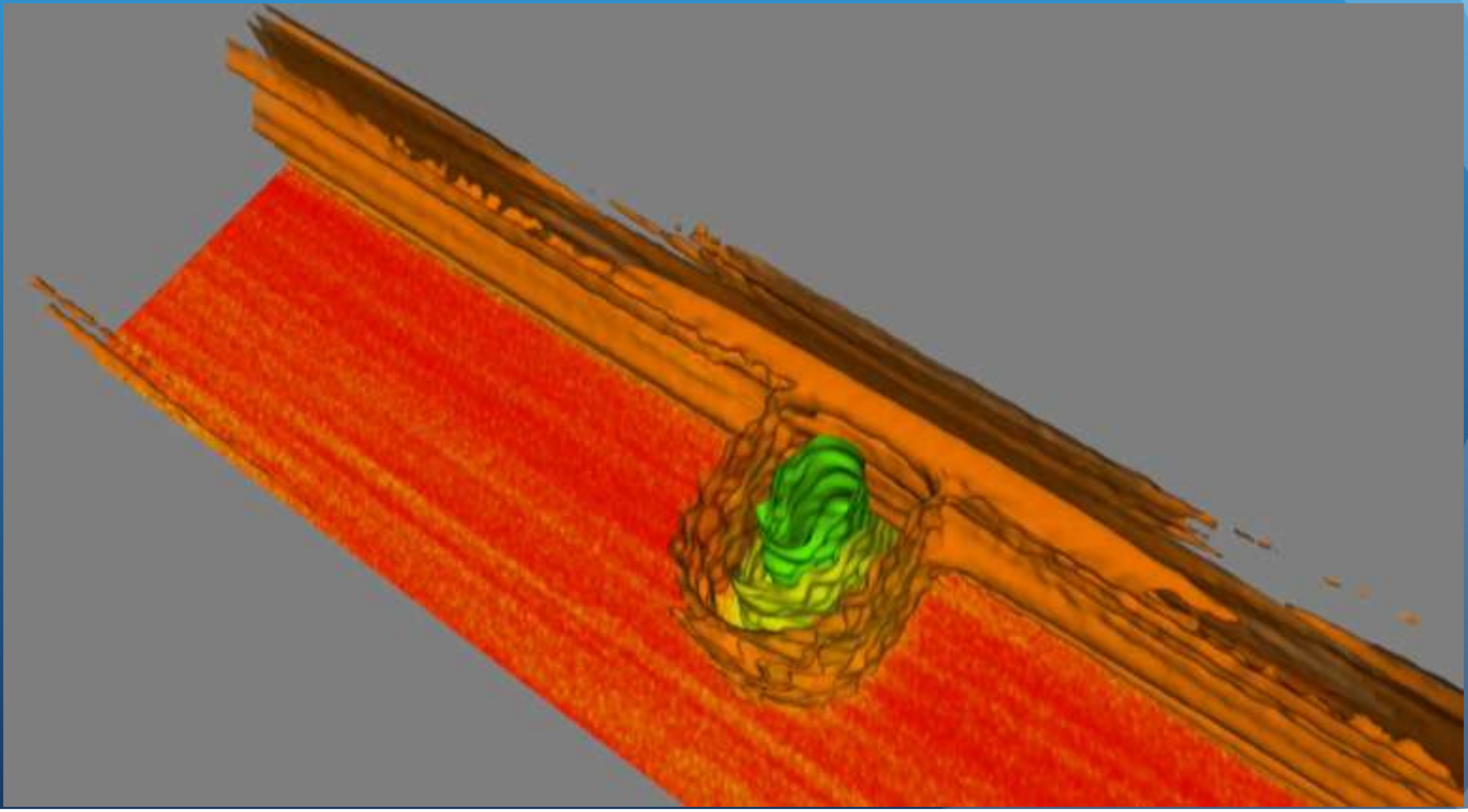

Meteor path
3 n_{plasma} isocontours
B
n_{plasma} cross-sectional plane

Particle in Cell Simulations of Meteor Plasma – with a wind



Conclusions

- New 3D Simulations
- Enables exploration of Meteor Evolution
- Future: Spectra to connect to observations



Conclusions

- Simulations enable us to explore nonlinear systems
- Simulations subject to systematic limitations and numerical errors
- Enable us to better understand our:
  - devices,
  - Models, and
  - Nature.
- Future Simulation Work:
  - Better Algorithms
  - More Parallel Efficiency
  - Vast array of applications!