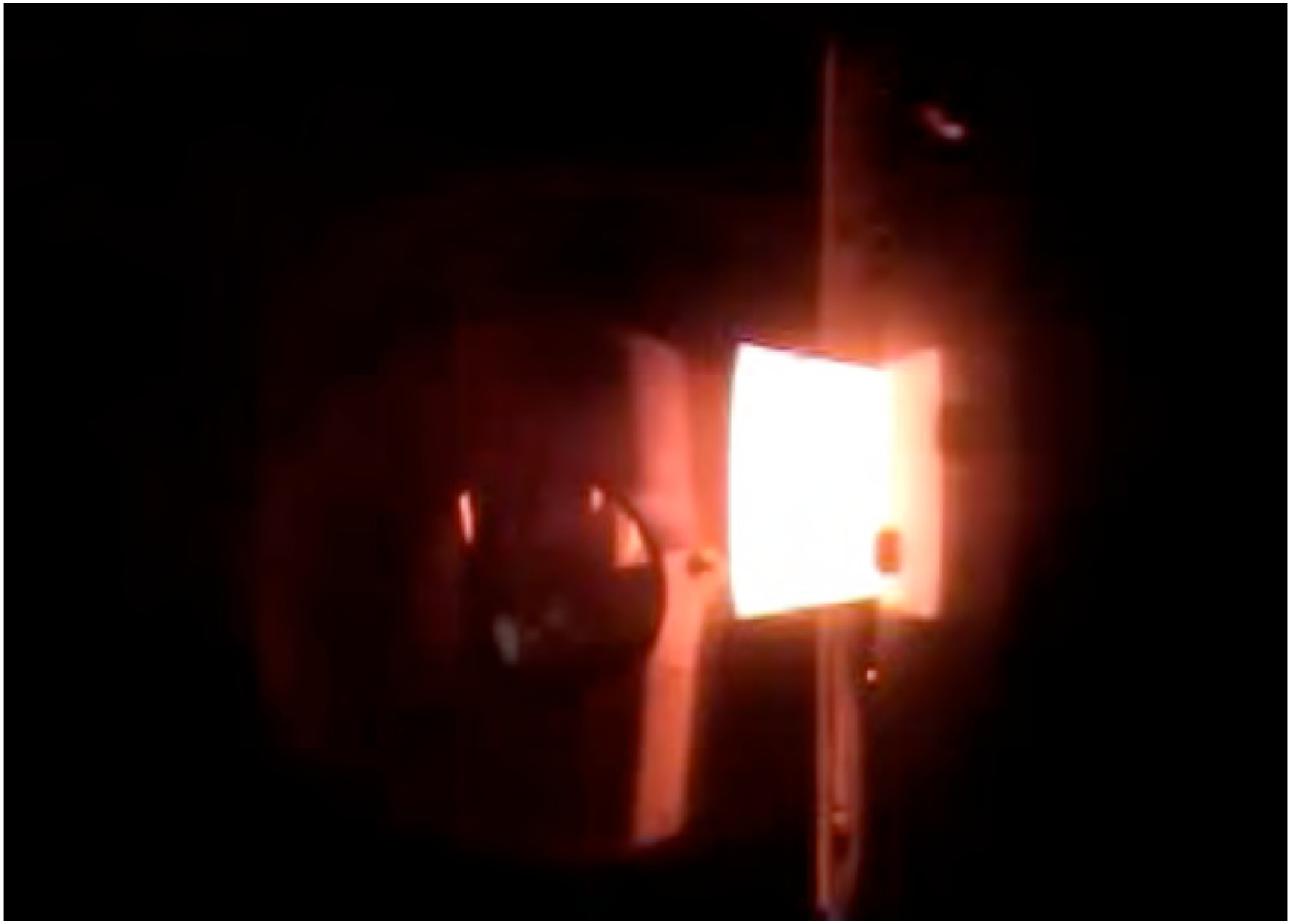
Direct Numerical Simulation of Magnetohydrodynamic Thruster Plasmas

Peter Norgaard

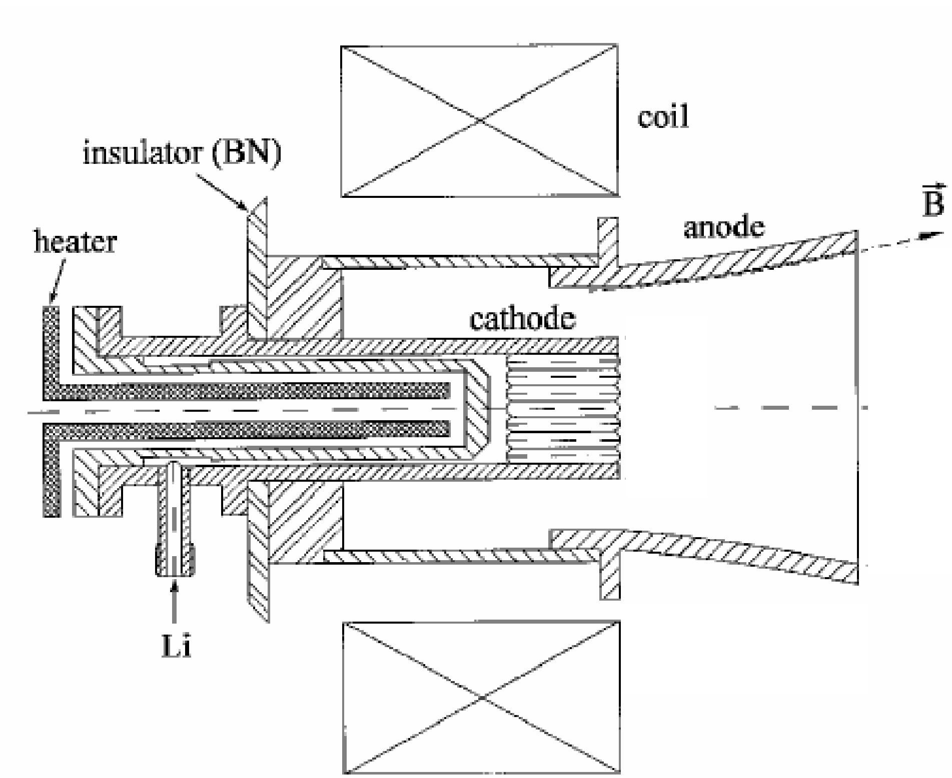
Princeton University

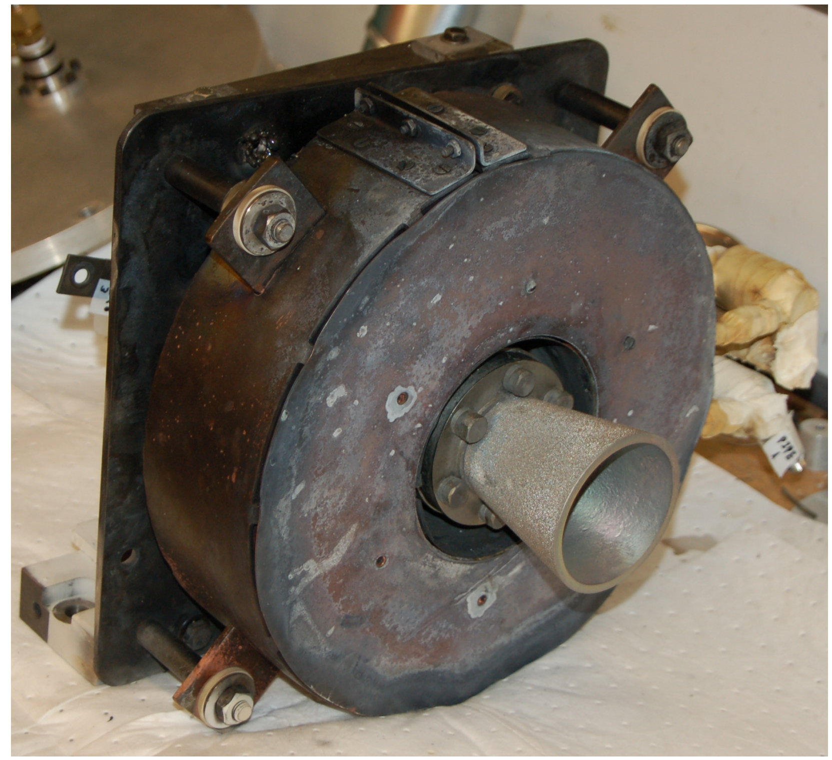
This is not a simulation…



Magnetoplasmadynamic Thruster

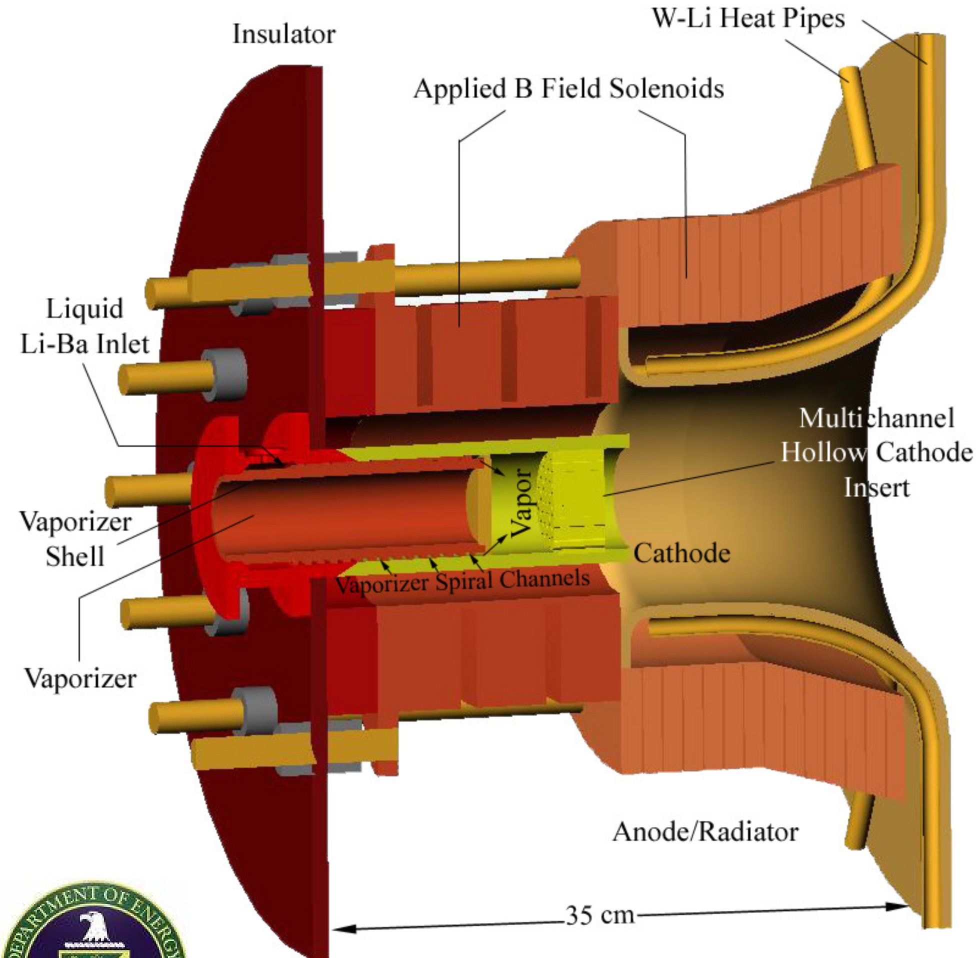
- Princeton EPPDyL applied-field lithium MPDT





Alpha Square MPDT

- Design of a 250 kW class thruster



Princeton University

JPL

Marshall Space Flight Center

Worcester Polytechnic Institute

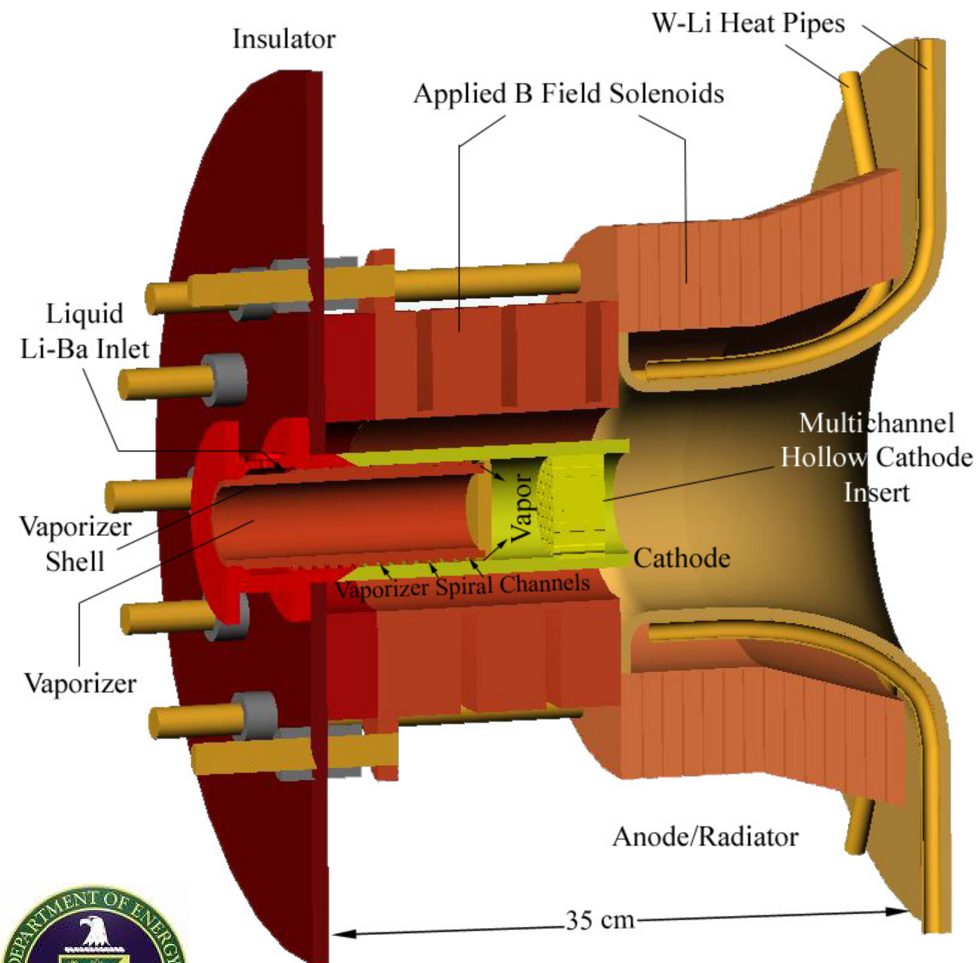
Ohio Aerospace Institute

University of Michigan

Glenn Research Center

Alpha Square MPDT

- Design of a 250 kW class thruster



Thrust = 5 N

Specific Impulse = 6200 s

Efficiency > 50%

Appropriate for Mars mission

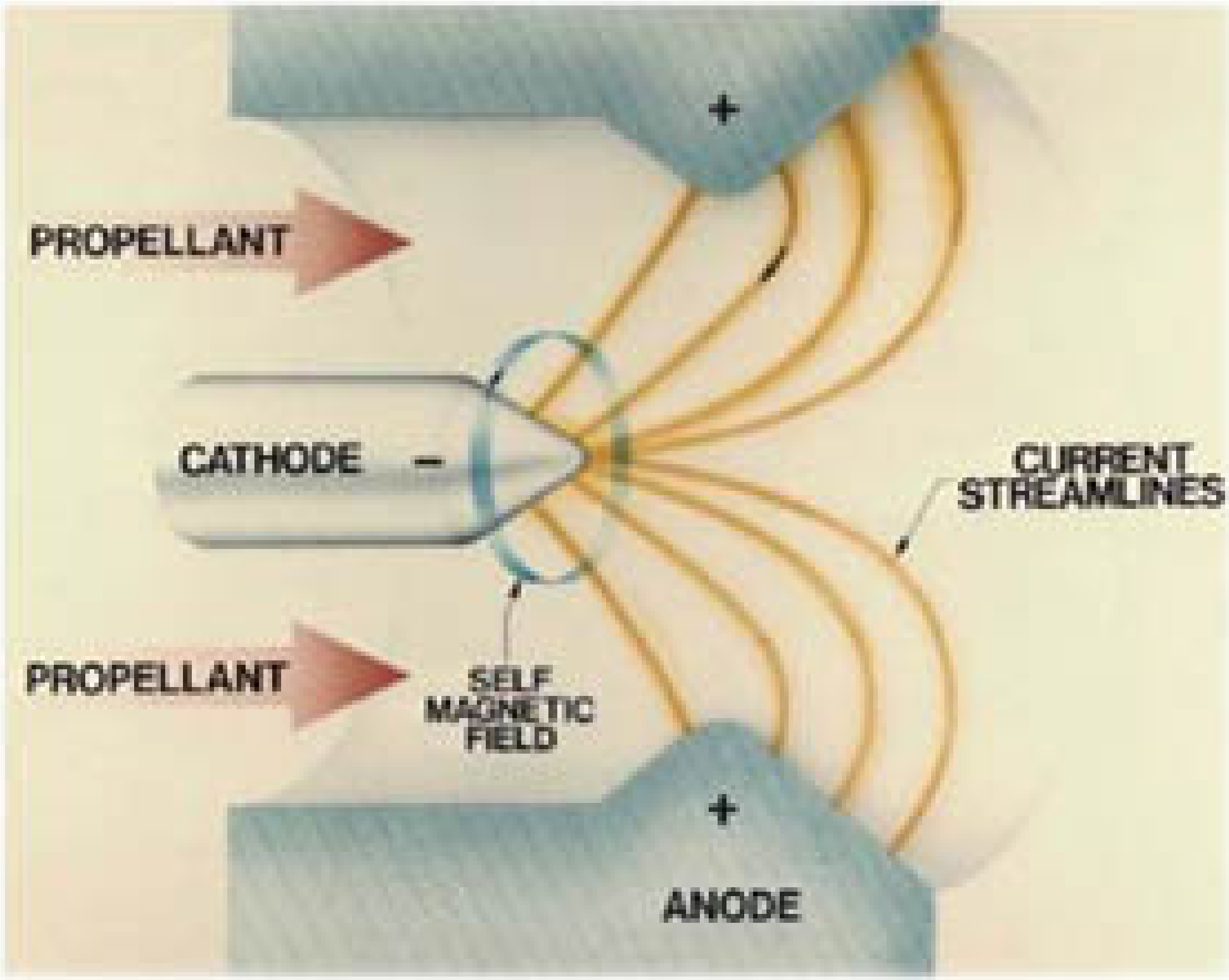
- two 250 kW thrusters

- 18,000 kg of propellant

vs. 175,000 kg!

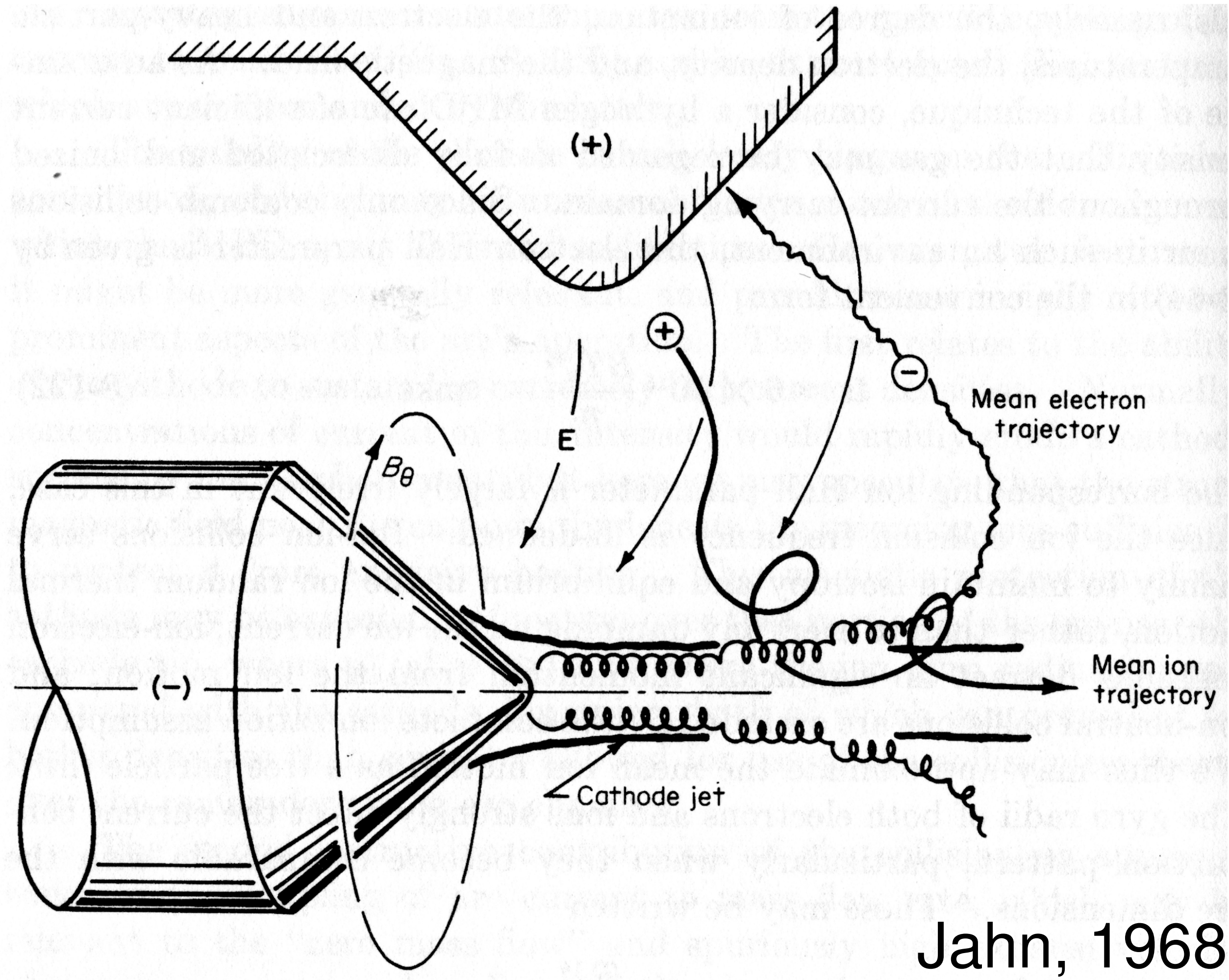
Physics of Operation

- Fluid Perspective: F = J x B



Physics of Operation

- Particle Perspective: F = q/m (E + v x B)



Jahn, 1968

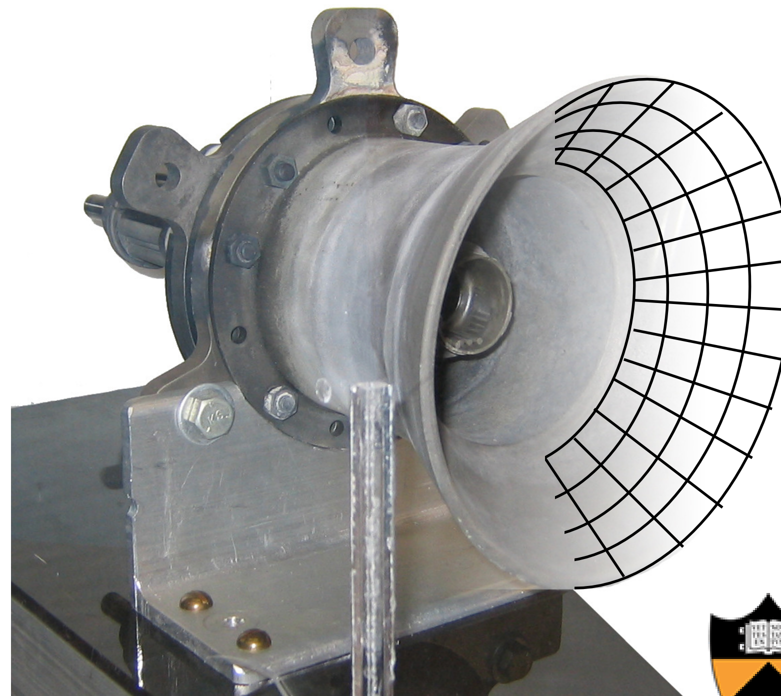
Why Numerical Simulation?

- Very expensive to build and operate

  Tungsten thruster, cryogenic system, massive vacuum system, …

- Difficult to make measurements

  - Probes disturb the flow-field, only provide point measurements
  - Passive systems have limitations & are also expensive (magnetic flux probes, laser interferometers, laser induced fluorescence, spectrometry, Thomson scattering, Zeeman splitting…



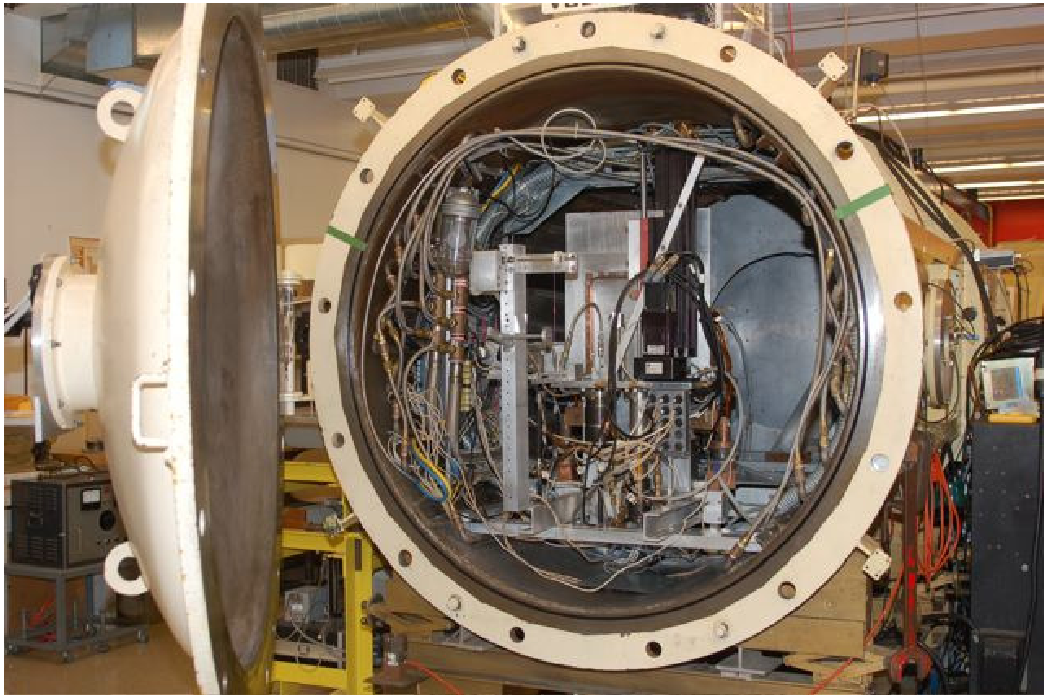
Why Numerical Simulation?

- Complexity of experimental operation

  Electrical feed system, heaters, coolers, vacuum, etc

- Dangerous to operate (environmental concerns)

  Electricity, lithium oxide contamination



Computational Approach

- Upsides:
  - Much cheaper $$$ / safer
  - Fully resolve (included) physics
  - Physics can be turned on/off

  Downsides:
  - Not real
  - Requires verification
  - No fun knobs to turn

- Choose a model
  - Geometry, equation type, what physics to include, operating parameters

- Choose a discretization and code
  - PDE solver - finite volume / finite element
  - Programming model - domain decomposition, structured grid, parallel
  - Programming language - C / C++ / Fortran90
  - Programming libraries - petsc, hdf5, superlu, lapack, netcdf, mpi

Extended MHD Equations

- Mass, Momentum, Magnetic field, and Energy
  - Often separate electron/ion energy equations
  - Alternate vector potential formulation for magnetic field
  - Additional equation to determine secondary quantities

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho \\ \rho\mathbf{v} \\ \mathbf{B} \\ \mathrm{E}\end{bmatrix} + \nabla\bullet\begin{bmatrix}\rho\mathbf{v} \\ \rho\mathbf{v}\mathbf{v} + \overleftrightarrow{p} - \overleftrightarrow{B}_M \\ \mathbf{v}\mathbf{B} - \mathbf{B}\mathbf{v} \\ (\mathrm{E} + p)\mathbf{v} - \overleftrightarrow{B}_M \bullet \mathbf{v}\end{bmatrix} = \nabla\bullet\begin{bmatrix}\mathbf{0} \\ \overleftrightarrow{\tau} \\ \overleftrightarrow{E}_{res} \\ q\end{bmatrix}$$

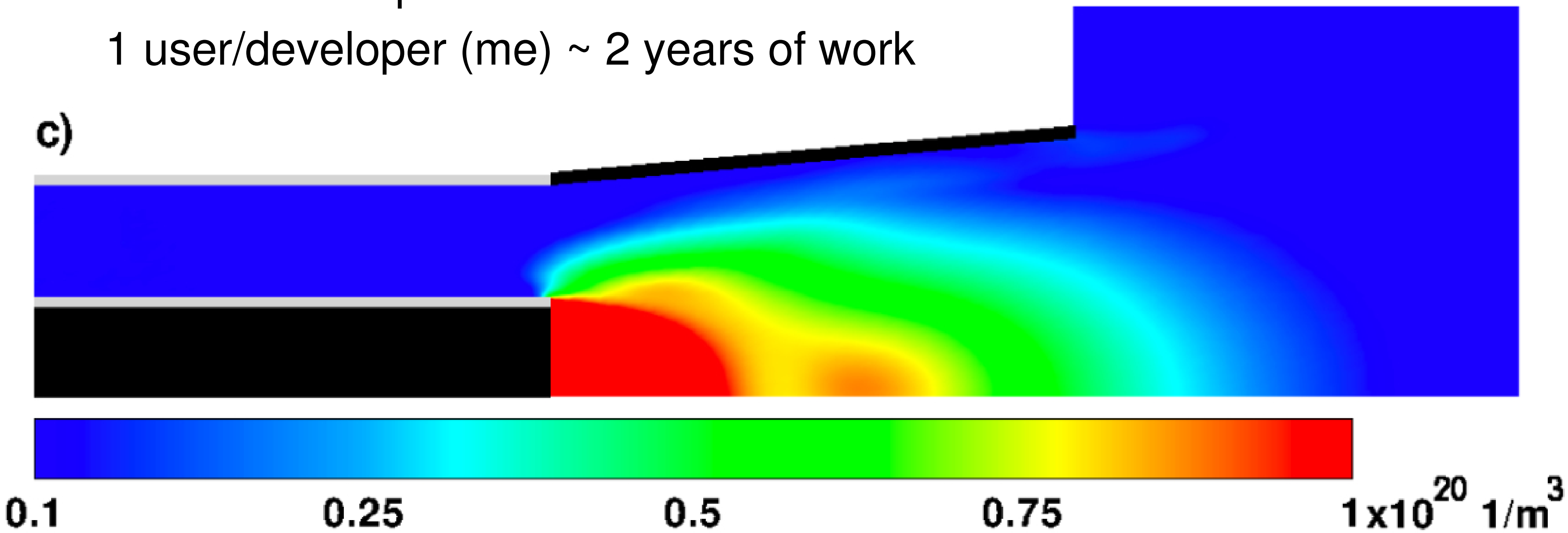
PCAPPS

- Princeton Code for Advanced Plasma Propulsion Simulation
  - Originally written by K. Sankaran, MAE grad student at Princeton
  - 2D aximmetric finite volume code (C) with 1D domain decomposition
  - Explicit time stepping with MPI share of boundary data
  - Ran on 16-64 procs of the PPPL Cluster
  - 1 user/developer (me) ~ 2 years of work

PPPL “kestrel” cluster:
224 AMD dualcore Opteron
compute nodes

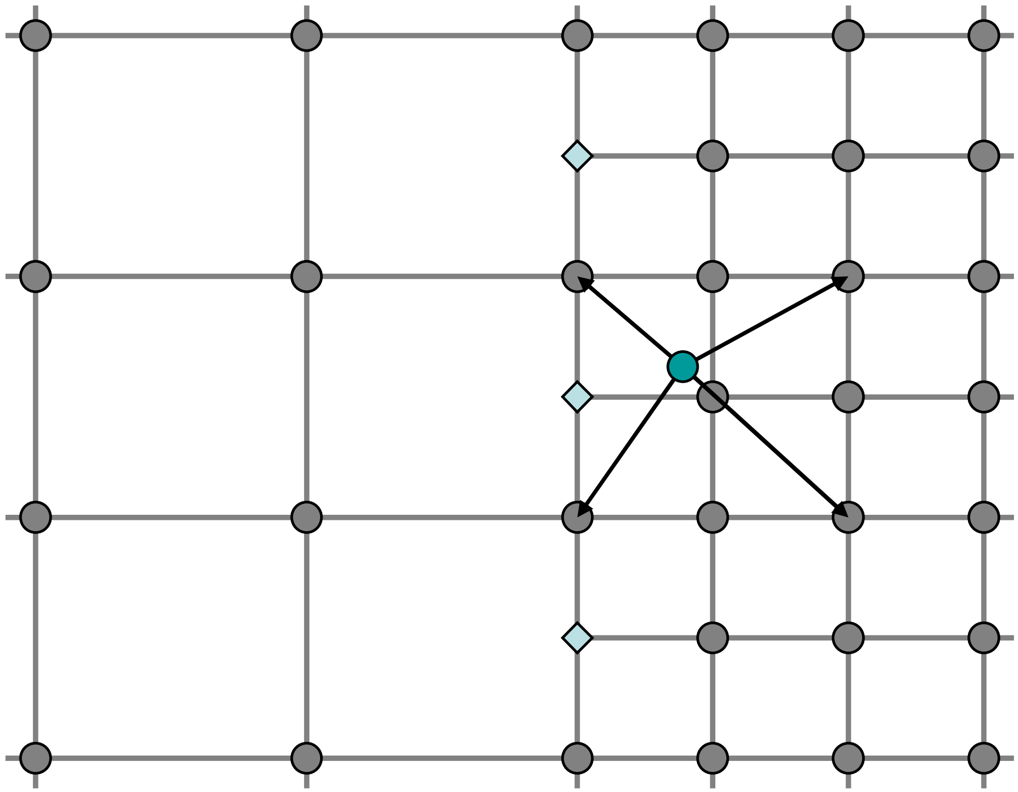
PCAPPS

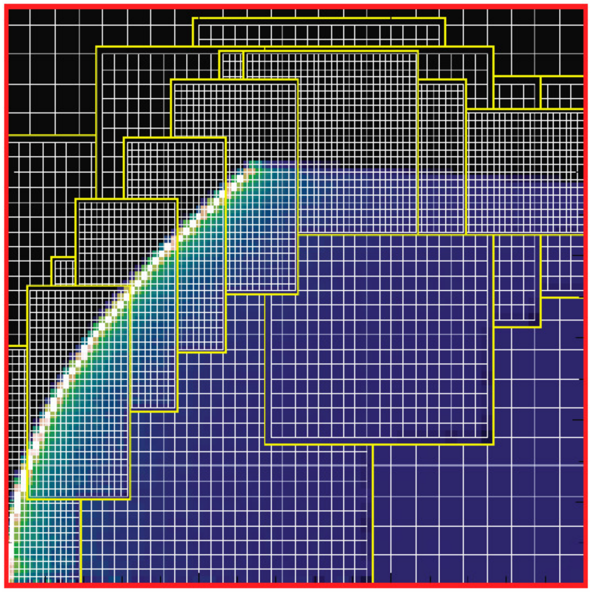
- Princeton Code for Advanced Plasma Propulsion Simulation
  - Originally written by K. Sankaran, MAE grad student at Princeton
  - 2D aximmetric finite volume code (C) with 1D domain decomposition
  - Explicit time stepping with MPI share of boundary data
  - Ran on 16-64 procs of the PPPL Cluster
  - 1 user/developer (me) ~ 2 years of work



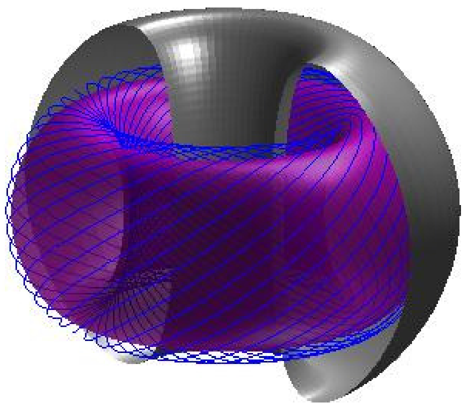
2 years of intervening time…

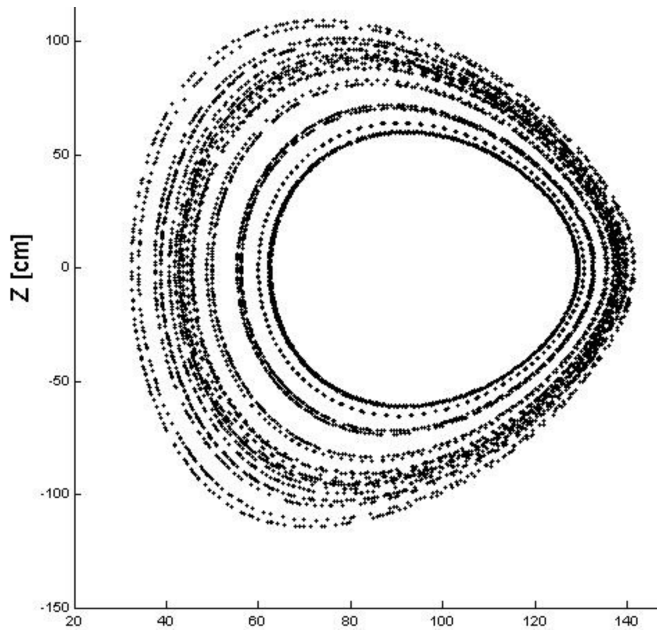
- LBL Practicum with ANAG group (P. Colella)
  - Particle in cell on Chombo adaptive mesh refinement grids
  - Large project, just working on a small part of the code using standardized C++ and F77 I/O patterns + documentation

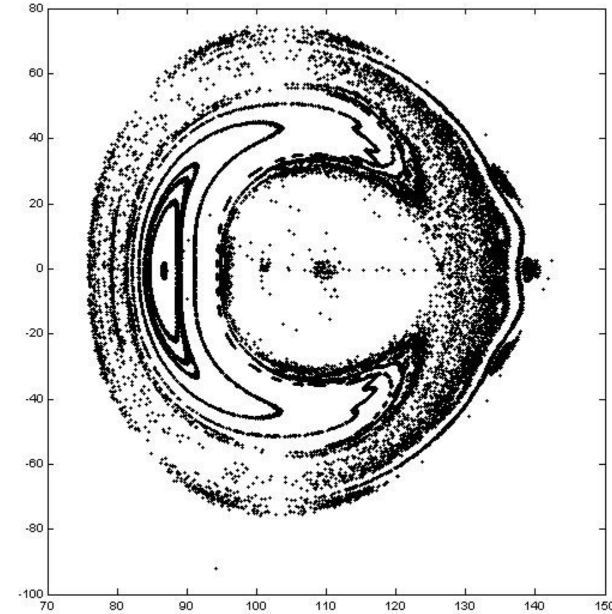




2 years of intervening time…







- Fast Ion Dynamics in a Tokamak

Studied the guiding center motion of neutral beam injected ions using R. White's "Orbit" code. Worked on methods for extracting dynamical system properties from the particle trajectories.

SEL / HiFi

- Spectral Element (2D) / High Fidelity (3D)
  - Developed by A. Glasser, S. Lukin at Princeton, written in F90
  - General Implicit non-linear PDE solver (based on PETSc)
  - Structured grid high-order finite element (Jacobi polynomials)
  - Tested on >1000 procs of the NERSC Franklin cluster
  - Many user/developers - PSI Center Group (~1 year so far for me)

NERSC Franklin cluster:
Cray XT4 MPP with 38,128 compute cores

VisIt visualization for hdf5

- https://wci.llnl.gov/codes/visit/
- The risky part of a presentation… doing anything live

Some Comments

- Learn to code in many languages
  Matlab, C, C++, F77, F90, Python…
- Learn programming models (parallelization) and use libraries
  MPI, POSIX, OpenMP…
  PETSc, SuperLU, LAPACK, HDF5,…
- Use developer tools / mesh generation / visualization
  Profile code (gprof), subversion, Cubit, TecPlot, VisIt
- Writing your own code is a good (painful) way to learn, but experience working with a larger development group is also important

Questions?

Thank you, DOE CSGF and my many advisors and collaborators!

www.princeton.edu/~norgaard