Predicting the Performance of Nuclear Fusion Experiments

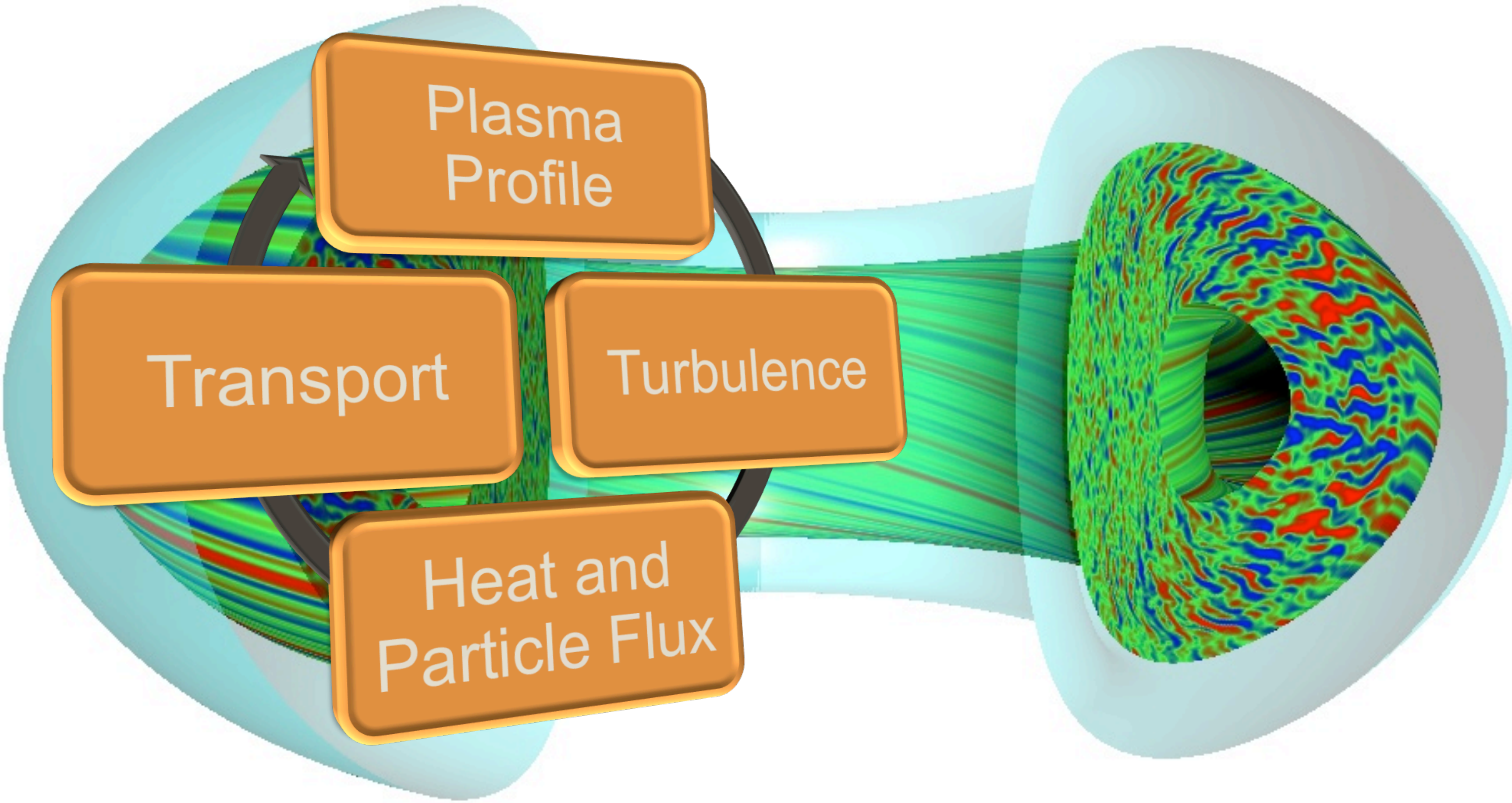
J. Luc Peterson

Princeton Plasma Physics Laboratory

SciDAC Center for the Study of Plasma Microturbulence

This work is supported by the Princeton Plasma Physics Laboratory, which is managed for the Department of Energy by Princeton University under DOE contract DE-AC02-09CH11466

Turbulent Transport is a Grand Challenge in Fusion Modeling



CSPM Studies the Relationship Between Turbulence and Transport with 3 Major Code Suites

- GYRO/NEO/TGYRO
  - General Atomics
- GS2/Trinity
  - U. Maryland, Oxford
- GEM
  - U. Colorado

Turbulence, Transport "Continuum"

Turbulence "PIC"

Emphasis on Physics Fidelity

Turbulence Codes Solve the Gyrokinetic Equation

- Evolution of a 5-dimensional distribution function
  - Coupled PDEs
  - 3 space dimensions, 2 velocity dimensions
- Continuum Codes
  - Evolve function using finite difference/volume and spectral techniques
- PIC Codes
  - Sample distribution function with Monte Carlo particles

Wish-list upgrades for Continuum Turbulence Codes

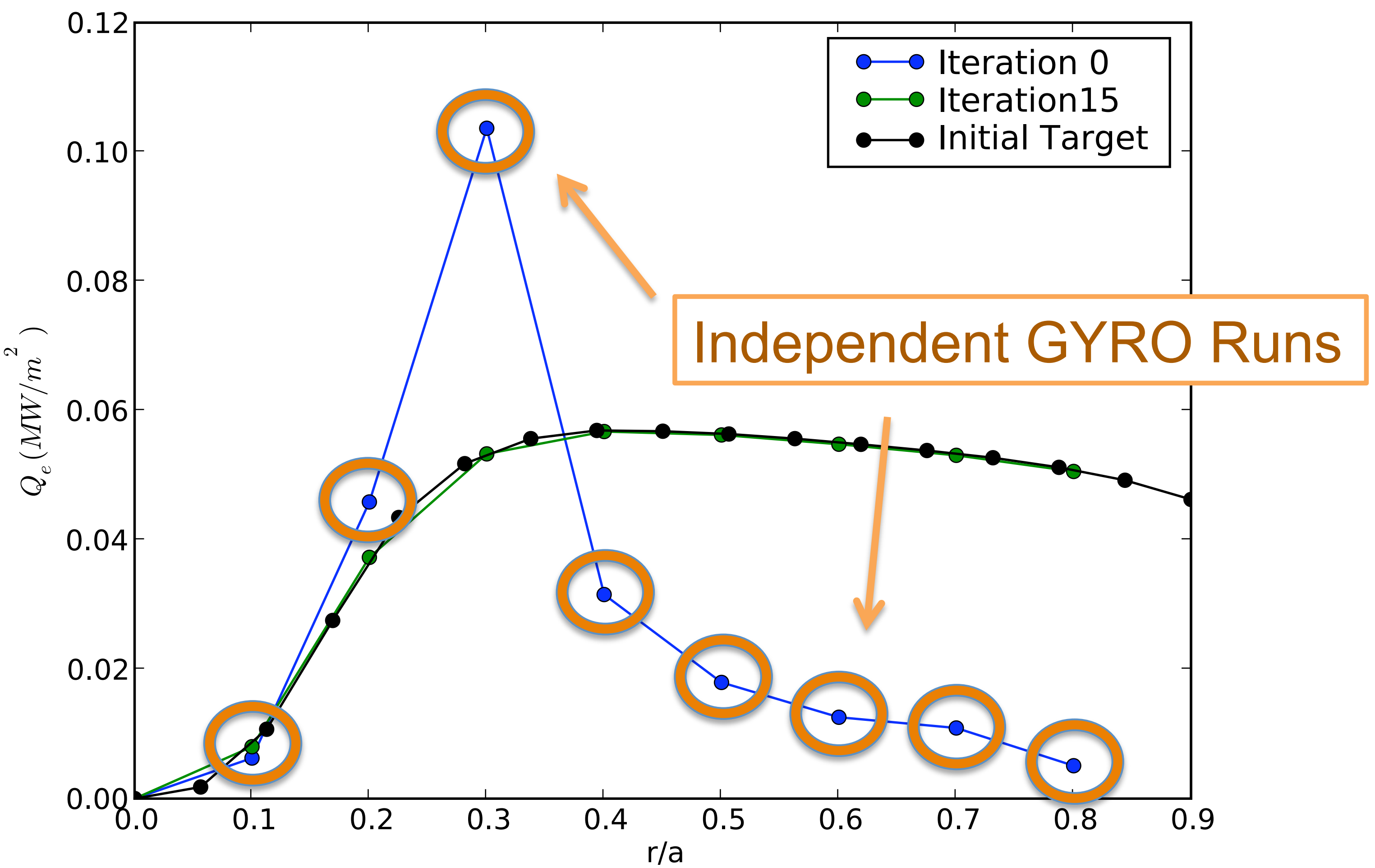
- Better multi-core performance
- Scalability above 5,000-10,000 processors
- Efficient dense matrix operators
  - Originally designed for problems with banded matrices
  - Next generation physics problems have dense structure
- E. Bass - GYRO

Transport Drivers Use Turbulence Codes

- Output of Turbulence is Input for Transport
  - Diffusion Coefficients, Fluxes
- Turbulence Driver evolves system to equilibrium
  - Time-dependent (Trinity)
  - Time-independent (TGYRO)

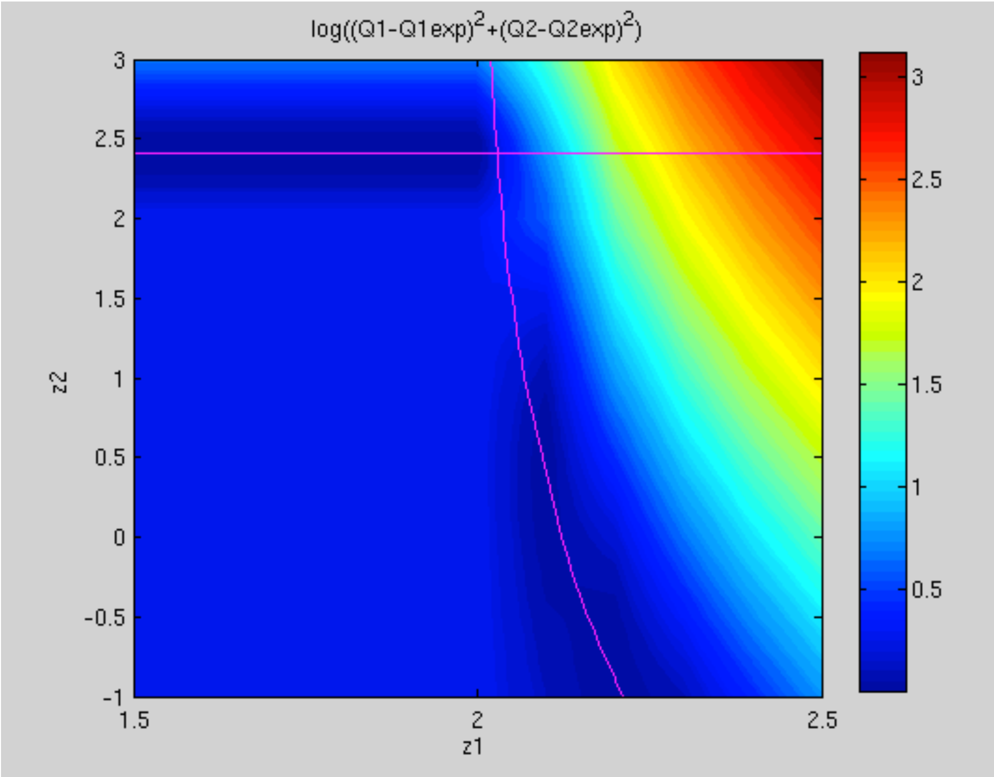
$$\vec{Q}_{code} = \vec{Q}_{exp}$$

TGYRO modifies inputs to GYRO/ NEO to find root



TGYRO is a Newton-Raphson Iterator

- Jacobian calculation is most expensive
  - GYRO ~ < 2000 processors each
  - ~ 500-100k cpu hours per GYRO call
- Phase space can be complicated



2-Point Toy Model
Shows Narrow Valleys
and Broad Plains

Computational Challenges for TGYRO

- Speed up Jacobian calculation
  - Extra layers of parallelization
  - Effective load balance
- Efficient Nonlinear Root Finding Method
  - Scales well
  - Robustly handles difficult terrain
- Wall-clock management
  - Time limits prevent many iterations