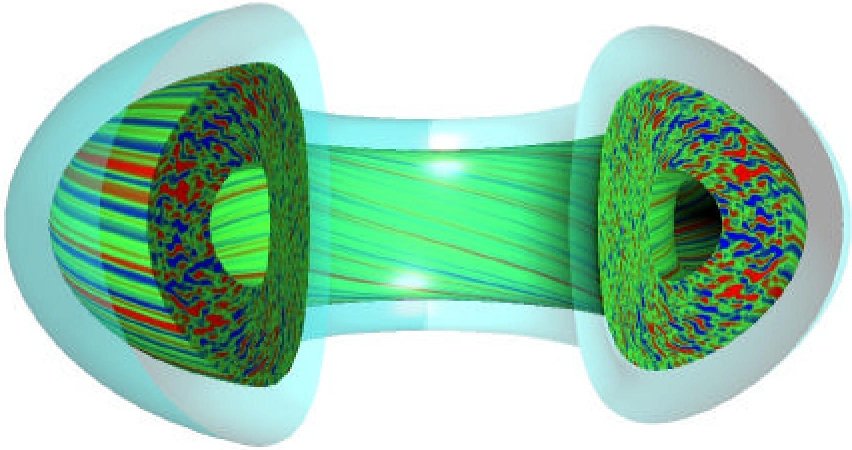
Microturbulence in Fusion Plasmas



W.M. Nevins ( )
For the
Plasma Microturbulence Project

Plasma Microturbulence Project Goal:

The Plasma Microturbulence Project is dedicated to the development, application, and dissemination of computer applications for the direct numerical simulation of plasma microturbulence (further information at **http://fusion.gat.com/theory/pmp**)

- *An important problem* — The transport of energy associated with plasma microturbulence is the key issue determining the size (and cost) of a burning plasma experiment (a key goal of the US magnetic fusion program).

- *Computer simulation* as a 'proxy' for plasma experiments:
  - Better diagnostics
  - Direct tests of theoretical models
  - Modeling experimental facilities before construction (or formal proposal)

- *Key Issue* — The 'fidelity' of the computational model
  - Continual improvements to the numerical model
  - Detailed comparisons between simulation and experiment

Three Ways to Study Plasma Turbulence

Experiments

Analytic Theory

Direct Numerical Simulation

QuickTime™ and a
decompressor
are needed to see this picture.

Our Game-Plan for the Direct Numerical Simulation of Plasma Turbulence

- Develop “high-fidelity” numerical models
  - Very good now… but there is always room for improvement

- Benchmark numerical models against
  - Each other
  - Experiments

⇒ Use simulations as “Proxies” for experiment
  - Easier to build
  - Easier to diagnose (with the proper tools)
  - Easier to run
  - More scope for parameter variations
    - Turn “physics” on/off
    - Vary machine size

We Support a 2x2 Matrix of Kinetic Codes for Simulating Plasma Core Turbulence

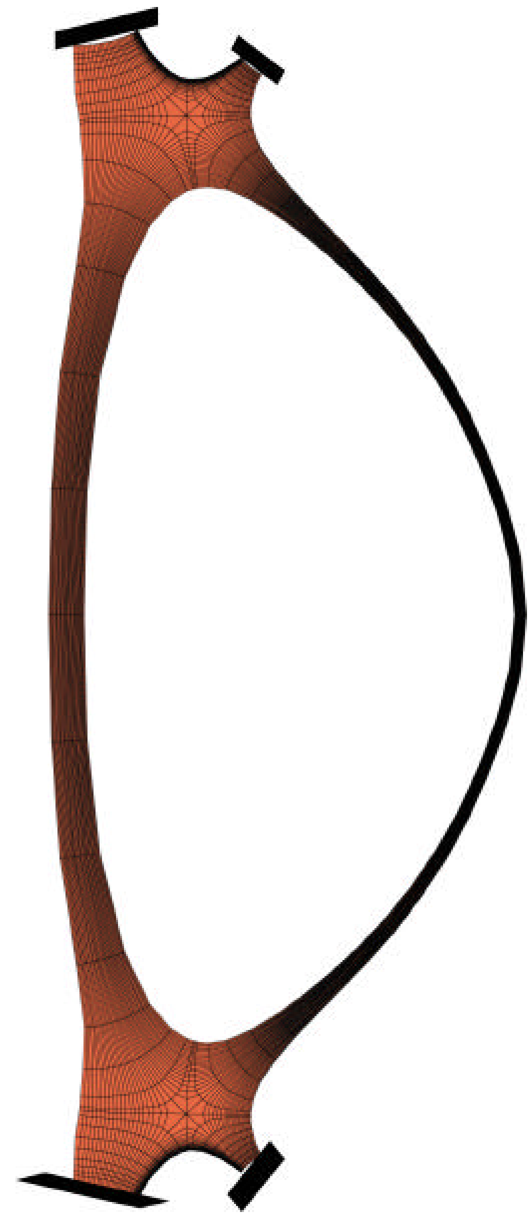
	Continuum	PIC
Flux Tube	GS2	SUMMIT
Global	GYRO	GTC

- Why both Continuum and Particle-in-Cell (PIC)?
  - Cross-check on algorithms
  - Continuum was most developed (already had kinetic *e*'s , $dB_{\wedge}$)
  - PIC is catching up (and may ultimately be more efficient?)
- If we can do Global simulations, why bother with Flux Tubes?
  - Efficient parameter scans
  - Electron-scale physics, (r_e, $d_e = c/w_{pe}$) << Macroscopic scale
  - Turbulence on multiple space scales (r_e, r_i, *& meso* scales all at once)

… and One Fluid Code for Plasma Edge Turbulence BOUT (X.Q. Xu,)

- Braginskii — collisional, two fluid electromagnetic equations
- Realistic ×-point geometry (open and closed flux surfaces)
- Collisional equations not always valid

⇒ Need to develop a kinetic edge code for realistic simulations of plasma edge turbulence

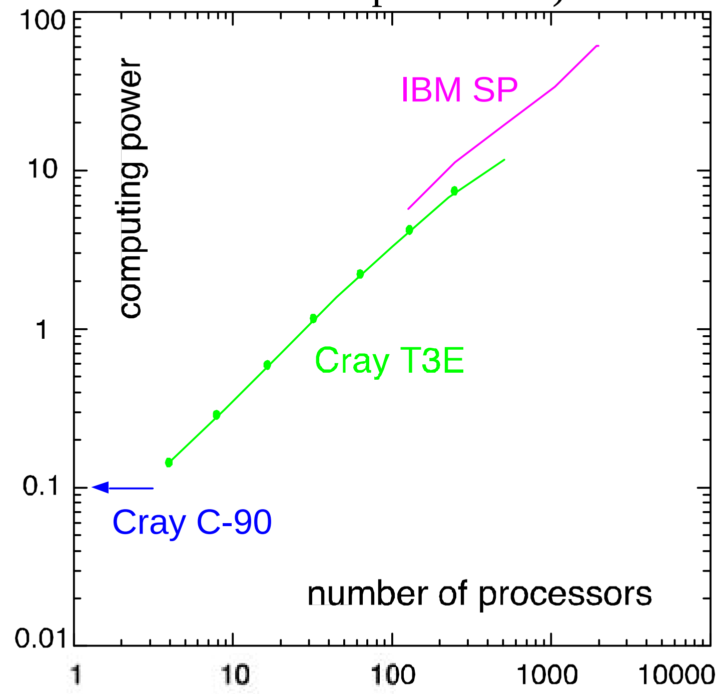


PIC Code Performance Scales Linearly to ~ 10^3 Processors

- Integrates GKE along characteristics
  - ⇒ Many particles in 5-D phase space
  - ⇒ Interactions through self consistent electric & magnetic (in progress) fields

⇒ Parallel particle advance scales favorably on massively parallel computers

GTC Performance Scaling
(problem size increasing with # of processors)

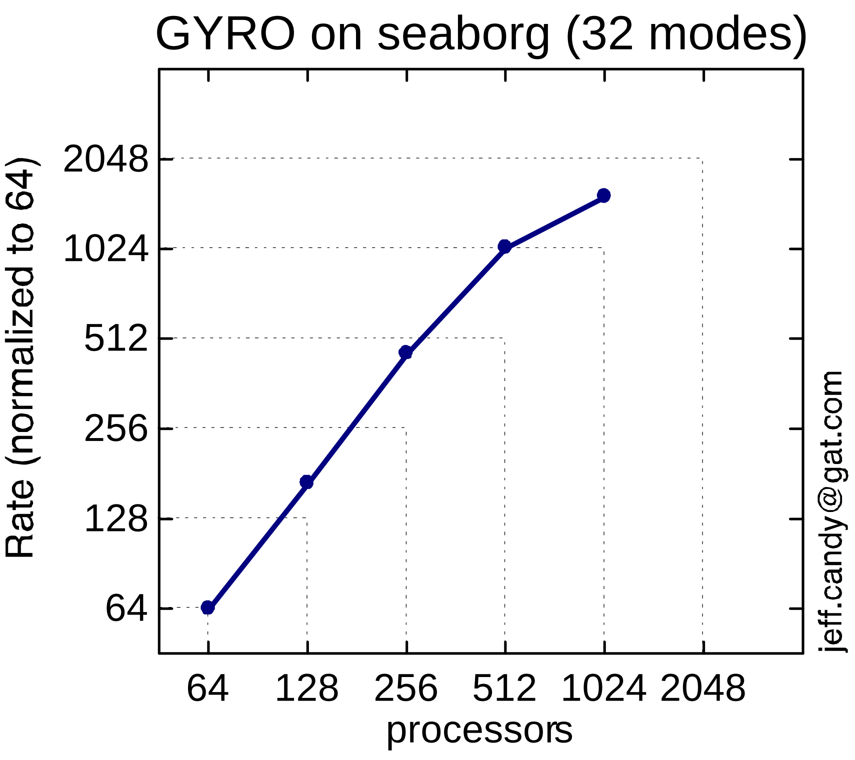


Y-axis: the number of particles (in milions) which move 1 step in 1 second

Continuum Code Performance Scales Linearly to ~ 10^3 Processors

- Solves GKE on a grid in 5-D phase space
  - Eliminates particle noise
  - Codes implements
    - Kinetic electrons
    - Magnetic perturbations
- Achieves ~linear scaling using domain decomposition
  - Linear scaling persists to more processors if problem size is increased with # of processors

Scaling with
Fixed Problem Size



Improving Code Fidelity: Kinetic Electrons and *dB*

SUMMIT: An Electromagnetic
Flux-Tube PIC Code

QuickTime™ and a
decompressor
are needed to see this picture.

Why is this Important?

- Kinetic electrons
  (have kinetic ions already)
  - → Electron heat transport
  - → Particle transport
  - – r_e-scale turbulence
- Electromagnetic ($dB_\wedge$)
  - – Finite-β corrections to ITG, etc.
  - – Kinetic ballooning modes
- Natural to implement together
  (*e*'s carry much of the current)
- Successfully implemented in
  three of four core turbulence codes

Current 'state-of-the-art'

Spatial Resolution

- Plasma turbulence is quasi-2-D
  - Resolution requirement along B–field determined by equilibrium structure
  - Resolution across B–field determined by microstructure of the turbulence.
  - $\Rightarrow$ ~ $64\times(a/\rho_i)^2$ ~ 2×10^8 grid points to simulate ion-scale turbulence at burning-plasma scale in a global code
  - Require ~ 8 particles / spatial grid point
  - $\Rightarrow$ ~ 1.6×10^9 particles for global ion-turbulence simulation at ignition scale
  - ~ 600 bytes/particle
  - $\Rightarrow$ 1 terabyte of RAM

$\Rightarrow$ This resolution is achievable

Temporal Resolution

- Studies of turbulent fluctuations
  - Characteristic turbulence time-scale
    $\Rightarrow$ c_s/a ~ 1 µs (10 time steps)
  - Correlation time >> oscillation period
    $\Rightarrow$ τ_c ~ 100× c_s/a ~ 100 µs (10^3 time steps)
  - Many τ_c's required
    $\Rightarrow$ $T_{simulation}$ ~ few ms (5×10^4 time steps)
  - 4×10^{-9} sec/particle-timestep (this has been achieved)
  - $\Rightarrow$ ~90 hours of IBM-SP time/run

⊠ Heroic (but within our time allocation)

(Such simulations have been performed, see T.S. Hahm, Z. Lin, APS/DPP 2001)

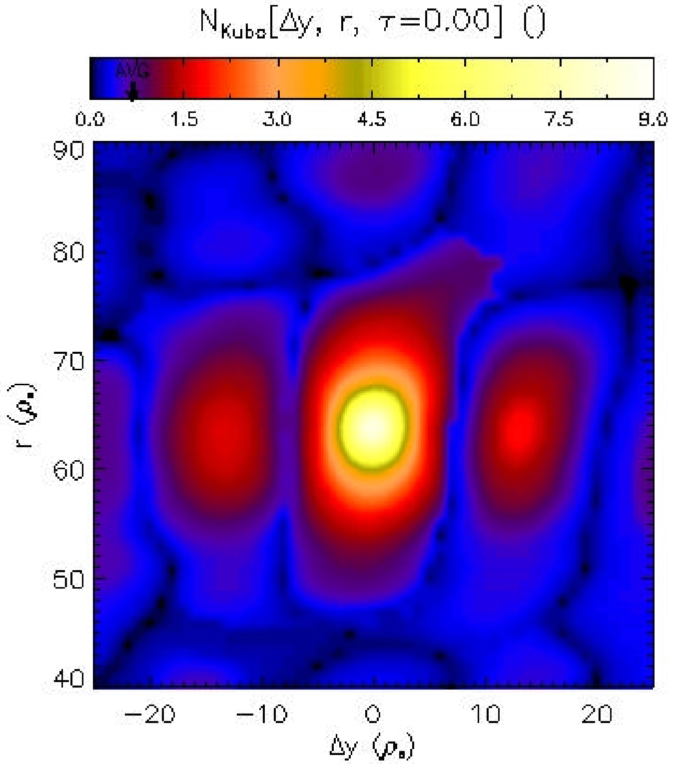
- Simulations including kinetic electrons and *dB* (short space & time scales) are not yet practical at the burning-plasma scale with a global code

Data Analysis & Visualization: The Bridge to Our User Communities

Interactive Data Analysis with GKV

- Productive data exploration
  - ⇒ "Granularity"
  - – Significant results from a few commands
  - – Flexible data exploration
- Standard analysis routines
  - – Spectral density
  - – Correlation functions
  - – …
- Custom Analysis
  - – Particle Trapping
  - – Heat Pulse Analysis
  - – …

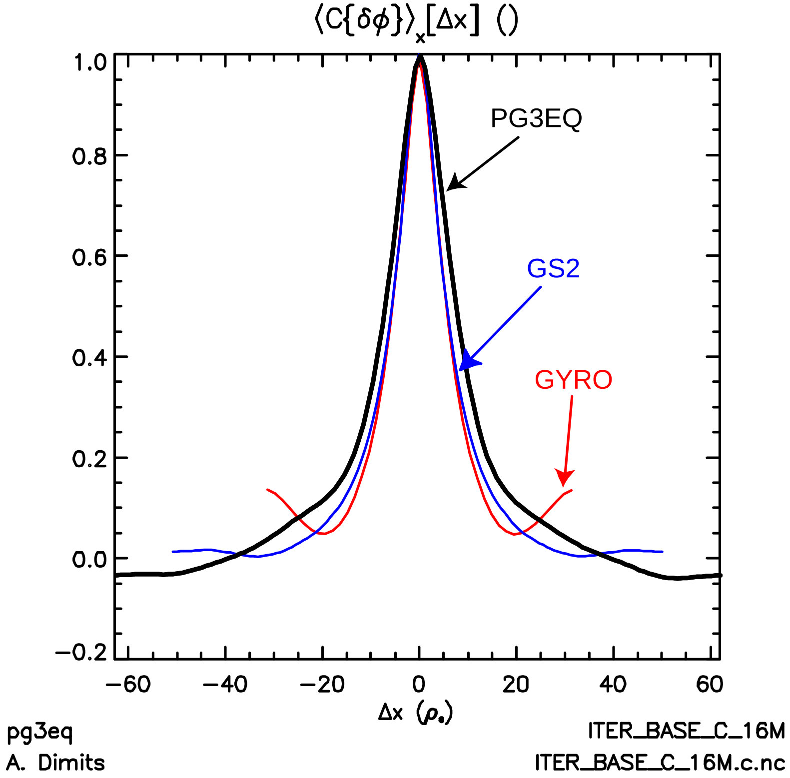
Quantifying the Importance
Of particle trapping



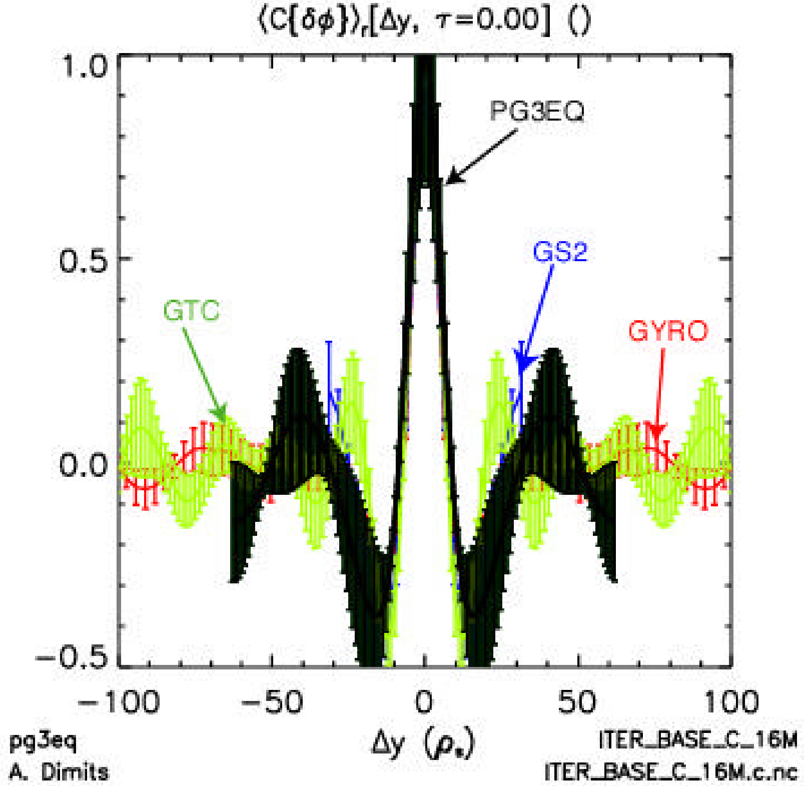
Benchmarking Codes Against Each Other

$$C_{df}(r, Dz, t=0 \mid r')$$

Radial Separation

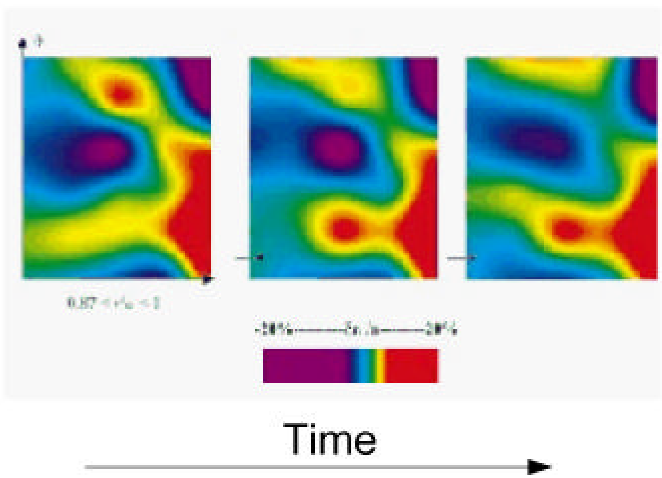


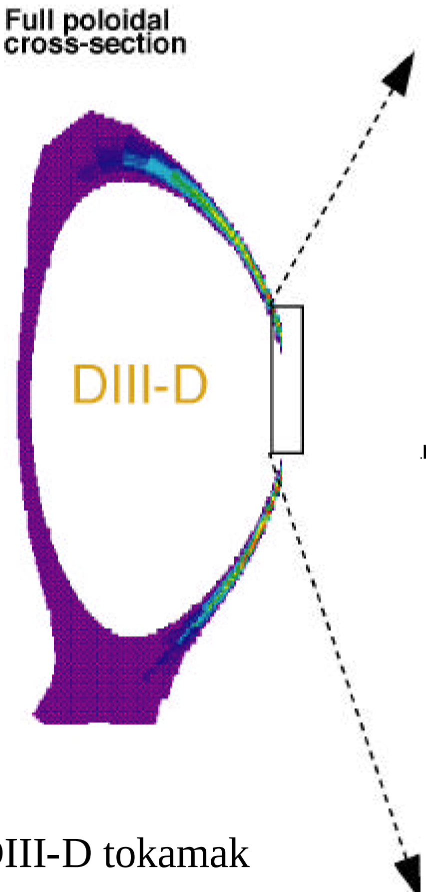
Poloidal Separation



Benchmarking Codes Against Experiment

BES Expt.





QuickTime™ and a
Video decompressor
re needed to see this picture.

L-Mode Edge Turbulence in the DIII-D tokamak

Major Computational and Applied Mathematical Challenges

- **Continuum codes** solve an advection/diffusion equation on a 5-D grid
  - Linear algebra and sparse matrix solves (LAPAC, UMFPAC, BLAS)
  - Distributed array redistribution algorithms (we have developed or own)
- **Particle-in-Cell codes** advance particles in a 5-D phase space
  - Efficient “gather/scatter” algorithms which avoid cache conflicts and provide random access to field quantities on 3-D grid
- **Continuum** and **Particle-in-Cell codes** perform elliptic solves on 3-D grids (often mixing Fourier techniques with direct numerical solves)
- **Other Issues:**
  - Portability between computational platforms
  - Characterizing and improving computational efficiency
  - Distributed code development
  - Expanding our user base
  - Data management

Q-1: What Has the Plasma Microturbulence Project Accomplished?

- Our expanding user-base enables MFE program to use terascale computing to study plasma turbulence
  - GS2 available as a web-based application (GS2 has more than 20 users beyond the GS2 development group)
  - GYRO user group (currently ~10 users) is expanding
- Kinetic electrons and δB enables new science
  - Electron heat flux – Particle flux – ρ_e-scale turbulence
  - Allows turbulence to tap the free-energy from electron gradients
  - Allows turbulence which is fundamentally electromagnetic (for example, kinetic ballooning modes)
  - Allows accurate modeling of actual tokamak discharges (and detailed comparisons between codes and experiment)

Q-2: How has the SciDAC team approach changed the way you conduct research?

- Closer contact with other SciDAC centers
  - The Fusion Collaboratory (connection to fusion community)
  - PERC to characterize and improve code performance
  - CCSE for efficient parallel solvers on unstructured grids
  - "Advanced Computing for 21st Century Accelerator Science and Technology" SciDAC center on PIC methods
- Improved interaction within Fusion community
  - Multiple-institution code development groups
  - Users who are not part of the code development group
- Common data analysis tools
  - Improved characterization of simulation results
  - Facilitates comparisons
    - Among codes
    - Between simulations and experiment

Q-3: What Software Tools does the Plasma Microturbulence Project Provide?

- Plasma microturbulence simulations codes
  - GS2 (available as a web application on Linux cluster at U. of MD)
  - GYRO (distributed to users at PPPL, MIT, U of Texas, …)
  - SUMMIT (users at U of CO, LLNL, UCLA)
  - GTC (users at PPPL, UC Irvine)
- GKV — a package of Data analysis and visualization tools
  - Open source w/Users manual — written in IDL (product of RSI)
  - Interfaces with all PMP codes
  - Users at LLNL, PPPL, U of MD, U of CO, U of TX, UCLA, …
- Tools from Other ISIC's ⇒ see previous viewgraph

Q-4: What are our Plans for Next Year?

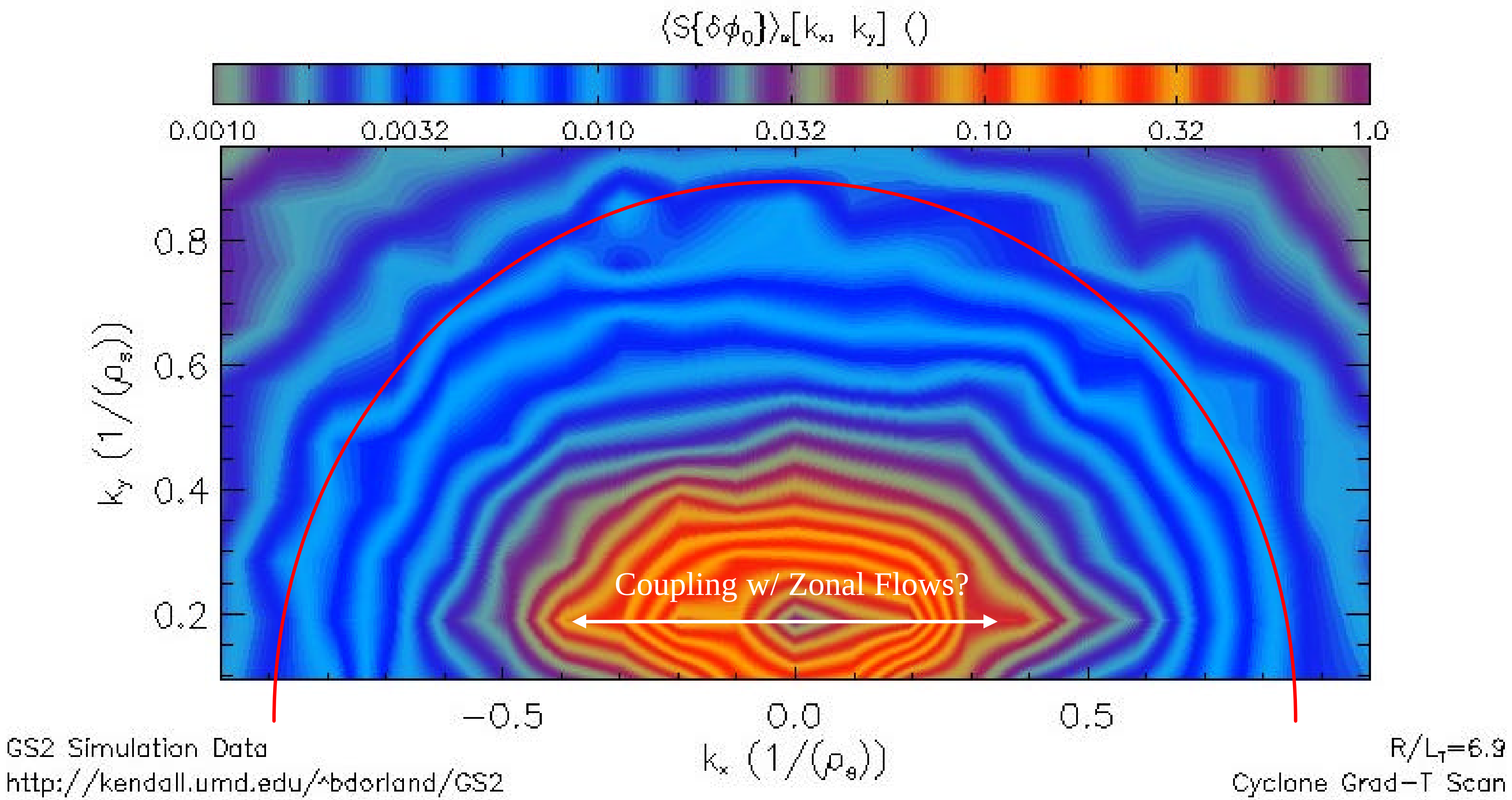
- Continue to expand our user base within the MFE community
  - GS2 – GYRO – Summit – GKV
- Complete development of
  - SUMMIT (global geometry, complete code merge, …)
  - GTC (kinetic electrons and δB)
  - GKV (additional diagnostic routines, interface to HDF5 files)
- Apply these tools to the study of plasma microturbulence
  - Continued code benchmarking (among codes and with experiment)
  - Continue to use codes to study plasma microturbulence
    - Emphasis on electron-driven turbulence and effects of δB
    - Understand mechanism for the termination of the inverse cascade

Q-5: Anticipated Resource Needs?

- Computer cycles!
  - Kinetic electrons $\Rightarrow$ More time steps/simulation
  - More users $\Rightarrow$ More simulations
  - Presently have ~ 5 Mnode-hrs between NERSC & ORNL
- Network infrastructure to support data transfer
  - Between computer centers – To mass storage
  - To user's home site for data analysis and visualization
- Data storage (and management)
  - Potentially a large problem
    (We just don’t save most of the available simulation data at present)
  - Need to do more work in this area
    (Develop a data management system linked to the Exp’t database?)

Characterizing Plasma Turbulence

Isotropic at large ***k*** Anisotropic at small ***k***

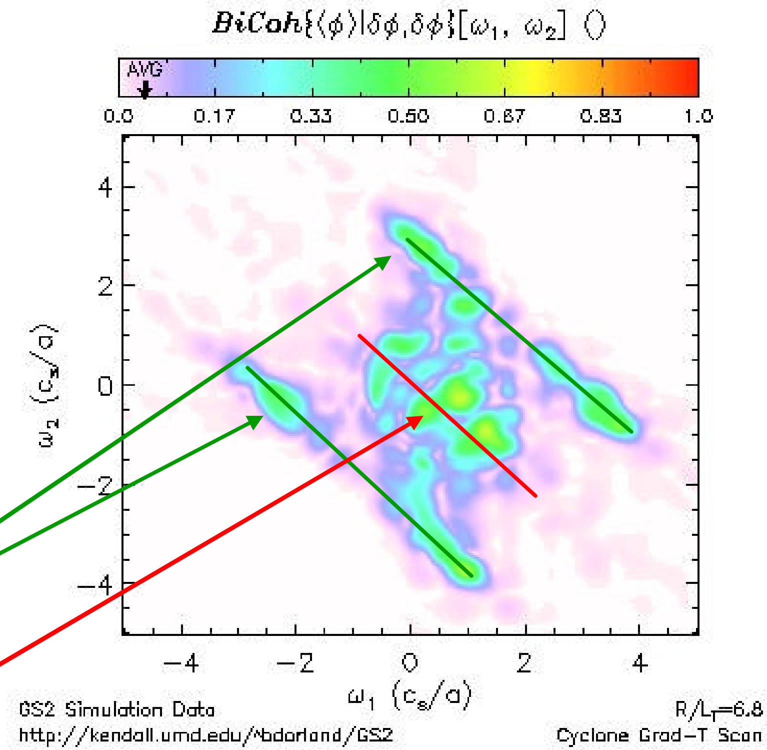


3-Wave Coupling?
$\acute{a}f\tilde{n}(k_x)$ with $d\boldsymbol{f}(\boldsymbol{k_1})$ and $d\boldsymbol{f}(\boldsymbol{k_1} + k_x\hat{\boldsymbol{e}}_{\boldsymbol{x}})$

- Compute Bicoherence
  - Measures phase coherence between selected data:

    $\acute{a}f\tilde{n}(k_x)$, $d\boldsymbol{f}(\boldsymbol{k_1})$, $d\boldsymbol{f}(\boldsymbol{k_1} + k_x\boldsymbol{e_x})$
  - Phase coherence

  $\Rightarrow$ Coupling
- Infer coupling between ITG turbulence and:
  - Geodesic-Acoustic modes
  - Zonal Flows



- And, all this only took a few minutes with GKV!

Microturbulence Sets the Minimum Size of a Burning Plasma Experiment

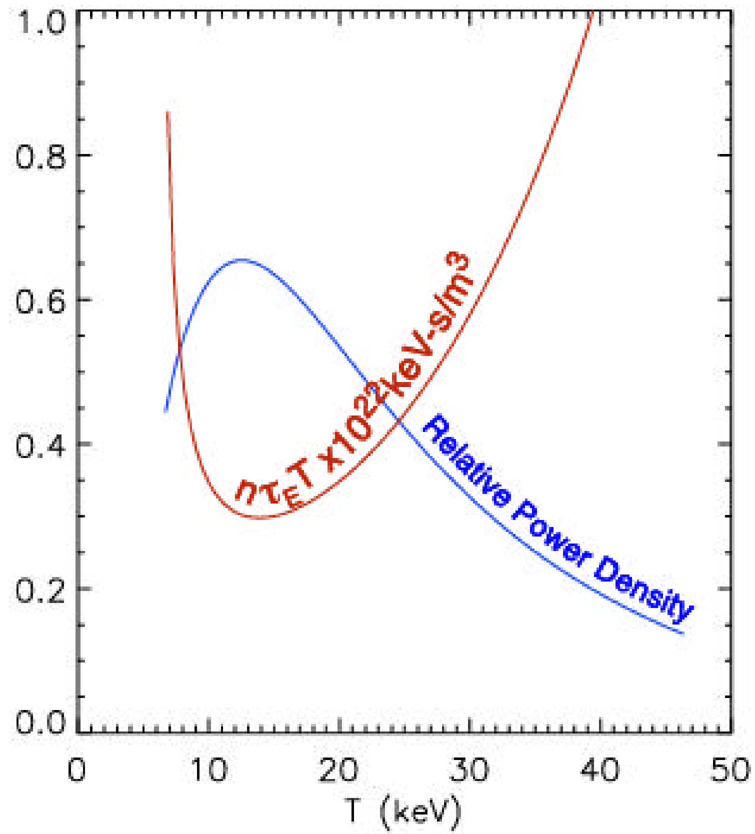
Power Balance in a Burning Plasma

Fusion heating = Radiation loss + Transport loss

$$3.5MeV \times n_d n_t \langle s v \rangle_{dt} = P_{Brem} + \frac{\frac{3}{2}\left(n_e + n_i\right)T}{t_E}$$

$$\sim n^2 f(T) \qquad \sim n^2 g(T) \qquad \sim \frac{nT}{t_E}$$

$$\Rightarrow \quad n_e t_E T = F(T)$$



Empirical Confinement Laws
(based on world tokamak data base)

Assume:

$$\tau_E = C_0\, I_p^{\alpha_1} B^{\alpha_2} n^{\alpha_3} P^{\alpha_4} R^{\alpha_5} \ldots$$

Choose $\{C_0, \alpha_n\}$ by regression

The Good News:

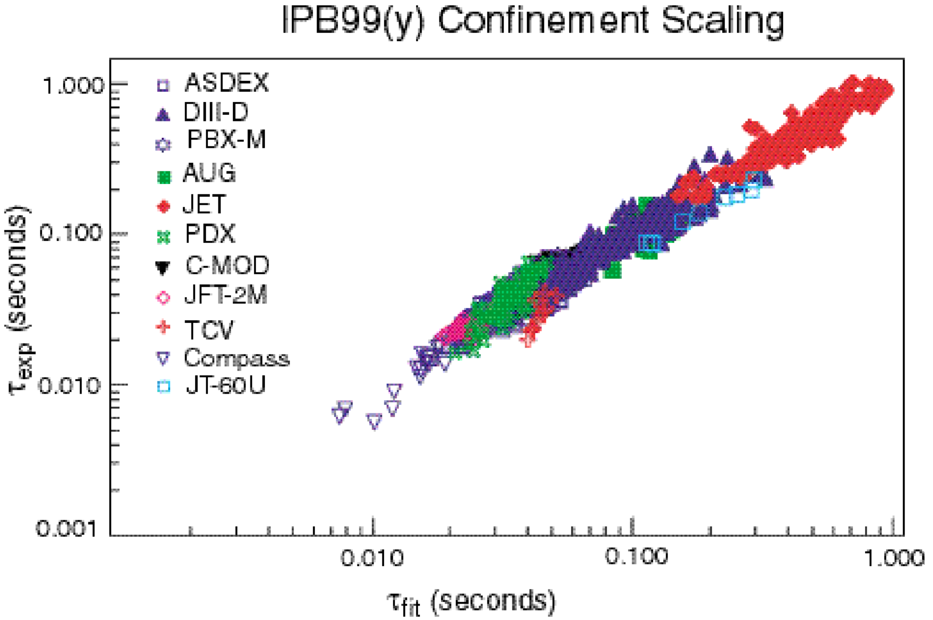
It works reasonably well
(design basis for many experiments)

The Bad News:

$$\tau_E \sim I\, R^{1.5}/\sqrt{P}$$

$\Rightarrow$ $n\, \tau_E\, T \sim I_p^{\,2}$

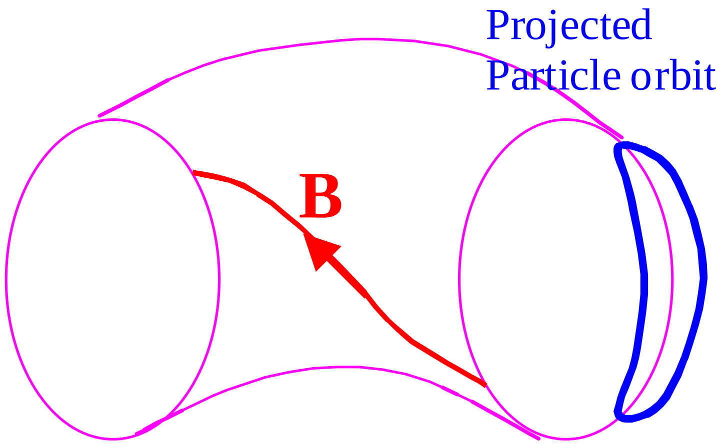
… and enough I_p for ignition costs more than Congress has been willing to pay



Plasma Microturbulence Determines Energy Confinement

- Particles (and energy) “tied” to magnetic field lines
- Field lines “cover” nested tori
- Two mechanisms transport particles (& energy) across field
  - Binary Collisions
    - $\Rightarrow$ Classical transport
  - Plasma microturbulence
    - $\Rightarrow$ Anomalous transport
- Anomalous >> Classical

$\Rightarrow$ Need to study microturbulence



GKV: A Toolkit for Data Analysis and Visualization

- Data Analysis and Visualization is the Bridge Between

  Simulation ⇔ Theory & Experiment

  – 3-D microturbulence simulations produce large datasets
    - ☞ Data analysis must be automated
  – You learn new things by looking at data in different ways
    - ☞ Data analysis should be interactive
  – Turbulence is a stochastic process
    - ☞ Realization-independent characterization of turbulence
- GKV implements this functionality in object-oriented IDL (IDL is a product of RSI widely used in the fusion community)
  – GKV imports data from our core turbulence codes
  – From our edge turbulence code (BOUT)
  – And from experiment (NSTX, C-Mod, DIII-D, JET, …)