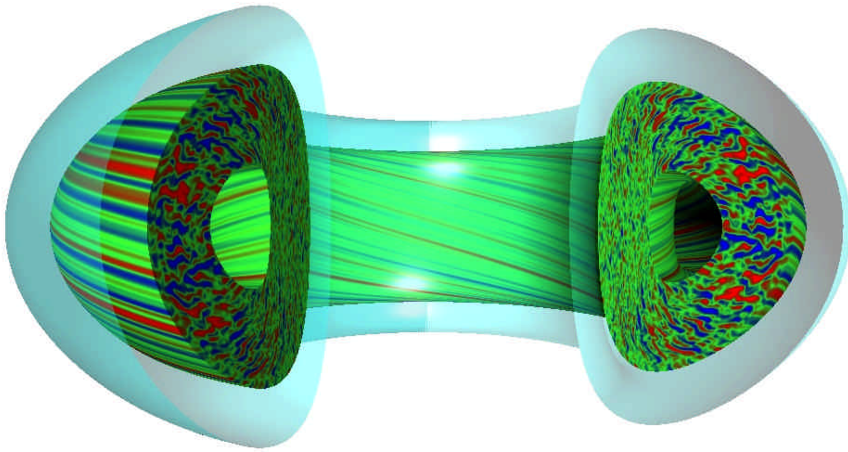
Turbulence and Transport in the Core and Edge Plasmas



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

How I Look at Simulations

- Simulations are *Proxies* for experiment —
  - Easier to build
  - Easier to diagnose
  - Easier to run
  - Have more scope for parameter variations
- The *Fidelity* of the simulation code
  - Simulation codes are imperfect proxies
    - The underlying model (fluid/kinetic, electrostatic/electromagnetic)
    - The numerical implementation (grid resolution, integration errors)
  - But they are still useful in their present (imperfect) state

A code is useful if it is able to

- Reproduce phenomena observed in experiments
- Make accurate predictions about future experiments

My "Reductionist" 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
- Validate numerical models against experiment
  - Build confidence in the numerical models
  - Generates interest expanded user base

Use simulations as "Proxies" to study plasma turbulence

- Easier to build
- Easier to diagnose (if you have the tools)
- Easier to run
- More scope for parameter variations
  - Turn "physics" on/off
  - Vary machine size

A Vehicle for Executing this Game Plan: The Plasma Microturbulence Project

- 2x2 Matrix of codes:

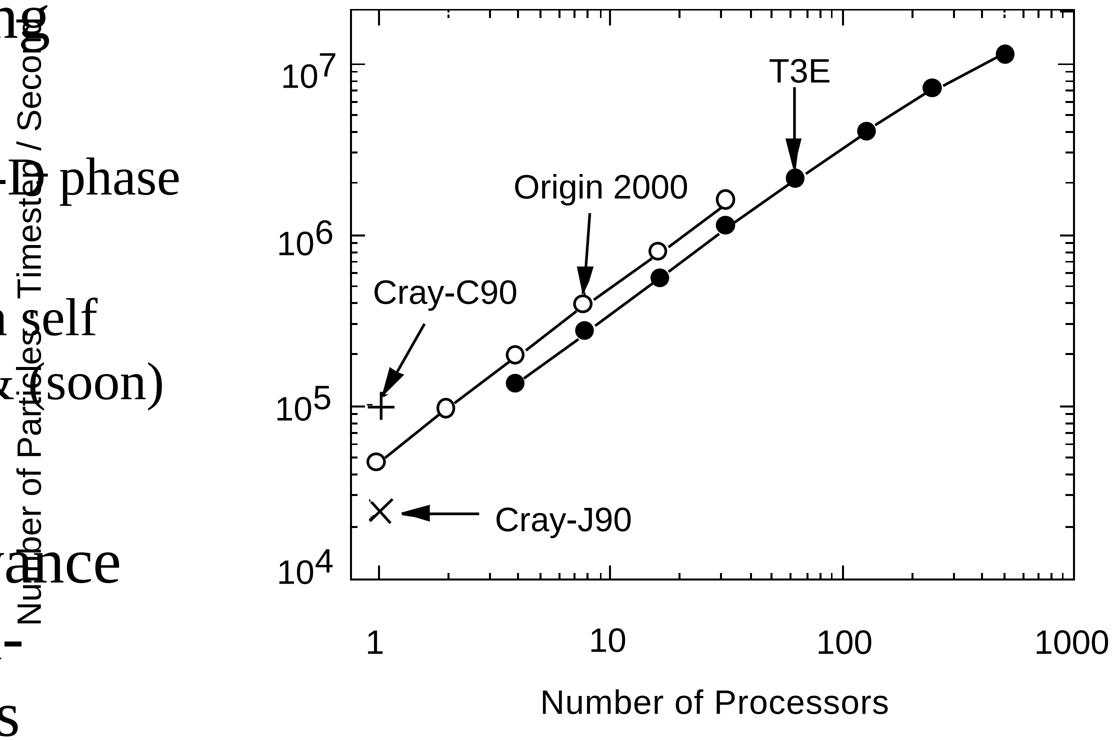
	Continuum	PIC
Flux Tube	GS2	SUMMIT
Global	GYRO	GTC

- Why both Continuum and PIC?
  - Cross-check on algorithms
  - Continuum currently most developed (already has kinetic e's , δB)
  - PIC may ultimately be more efficient
- If we can do Global simulations, why bother with Flux Tubes?
  - Electron-scale (ρ_e, $\delta_e = c/\omega_{pe}$) physics (ETG modes, etc.)
  - More efficient parameter scans

Existing Codes (I) Gyrokinetic Particle Codes

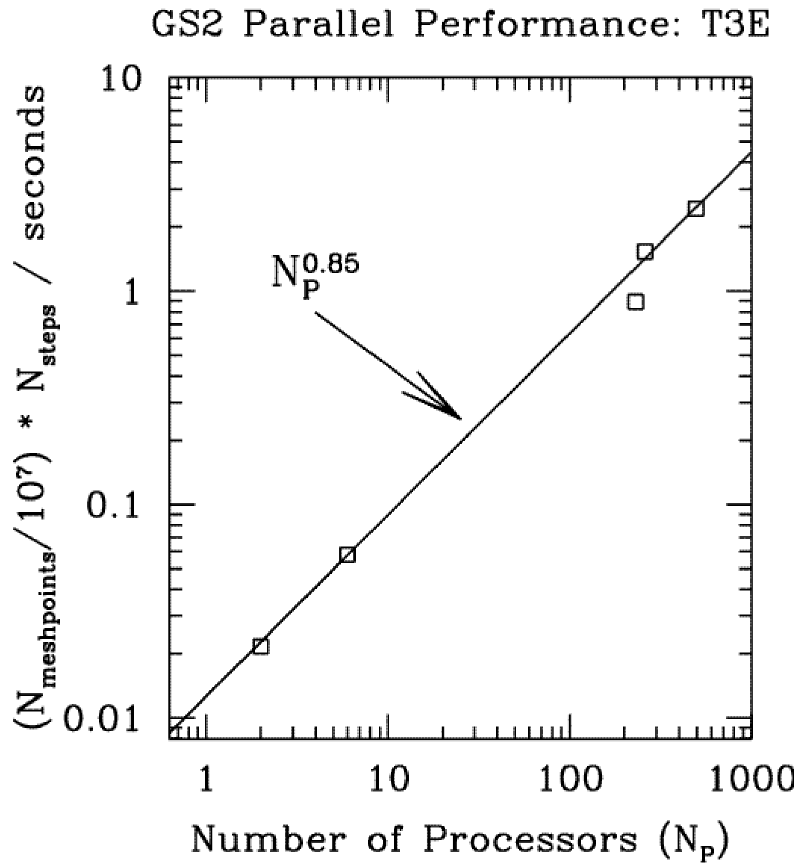
- Integrates GKE along characteristics
  - Many particles in 5-D phase space
  - Interactions through self consistent electric & (soon) magnetic fields

Parallel particle advance scales well on multi-processor computers



Existing Codes (II): 5-D Continuum Codes

- Solves GKE on a grid in 5-D phase space
- Eliminates discrete particle noise
- Linear physics is handled implicitly
  - Kinetic electrons & electromagnetism have less impact on size of time step



Current ‘state-of-the-art’

(similar performance achieved in Continuum codes)

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.

    $\sim 64\times(a/\rho_i)^2 \sim 2\times10^8$ grid points to simulate ion-scale turbulence at burning-plasma scale in a global code
  - Require ~ 8 particles / spatial grid point

    $\sim 1.6\times10^9$ particles for global ion-turbulence simulation at ignition scale
  - ~ 600 bytes/particle

    1 terabyte of RAM

  This resolution is achievable

Temporal Resolution

- Studies of turbulent fluctuations
  - Characteristic turbulence time-scale

    $c_s/a \sim 1$ μs (10 time steps)
  - Correlation time >> oscillation period

    $\tau_c \sim 100\times\, c_s/a \sim 100$ μs

    (10^3 time steps)
  - Many τ_c’s required

    $T_{simulation} \sim$ few ms

    (5×10^4 time steps)
  - 4×10^{-9} sec/particle-timestep

    (this has been achieved)

    ~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 electrons and δB (short space & time scales) are not yet practical at the burning-plasma scale with a global code

A Problem with Timescales!

Turbulence Simulation

- $T_{simulation} \approx 10^{-3}$ s
- $T_{IBM\text{-}SP} \approx 10^{2}$ hrs

Discharge Simulation

- $T_{simulation} \approx 10^{2}$ s
- $T_{computer} << 10^{2}$ hrs

- Need to speed-up by factor of $\sim 10^{6}$
  - Moore's Law?
  - Time Scale separation?
  - Reduced models?

Moore's Law

- Computer speed doubles every 18 months!

  (Moore was referring to computer memory, but …)
  - 10 yrs/(18 months) = 6.666 doublings
  - $2^{6.666\ldots} \approx 100$

We're 1/3 of the way there!

(there's still a factor of ~10^4 left to find …)

Time Scale Separation

(see Cohen & Lodestro, “Coupling of disparate-timescale simulations”)

- Run turbulence simulation for time $T_{simulation}$ advance discharge simulation for Δt
  - Discharge simulation $\Delta t \ll \tau_E \sim 1$ s
  - Turbulence simulation $T_{simulation} \sim 10^{-3}$ s
- Potential speed-up:

  $\sim \Delta t / T_{simulation} \sim 10^2$

We’re 2/3 of the way there!

(there’s still a factor of $\sim 10^2$ left to find …)

Credibility of Reduced Models Depends on Understanding Microturbulence

- Gyro-fluid models
  - Reduce phase space from 5-D to 3-D
  - Experience with Griffin, etc. (need to fix zonal flows …)
  - How to close fluid equations?
- Reduced kinetic models
  - Cleverly select a few modes …
  - Some "promising results", but hasn't worked yet …
- Improved 1-1/2D Transport models
  - See Glen Bateman on "Comprehensive Integrated Predictive Modeling"
  - Construct table of fluxes vs. local discharge parameters?
    (must reduce # of parameters, or dimensionality is too large)

Summary of the Timescale Problem

- Need to speed-up by factor of ~10^6
  - Moore's Law 10^2 ?
  - Time Scale separation 10^2 ??
  - Reduced models 10^2 ???
- "time scale separation" and "Reduced models" benefit from improved understanding/modeling of plasma microturbulence

  Need to increase our plasma microturbulence simulation effort!

Recommendations (I)

Strengthening PMP Support to Integrated Modeling

(1) Improve the fidelity and performance of Plasma Microturbulence Project codes

(2) Validate these codes against experiment

(3) Expand the user base of the PMP codes

(4) Initiate the development of a kinetic edge turbulence simulation code.

Why a Kinetic Edge Turbulence Code?

- The H-mode pedestal is the greatest source of uncertainty in projecting confinement
- BOUT experience geometry matters
  - Important qualitative physics from flux tube codes
  - Quantitative results from BOUT (for C-Mod)
- In typical (not C-Mod) plasma edge

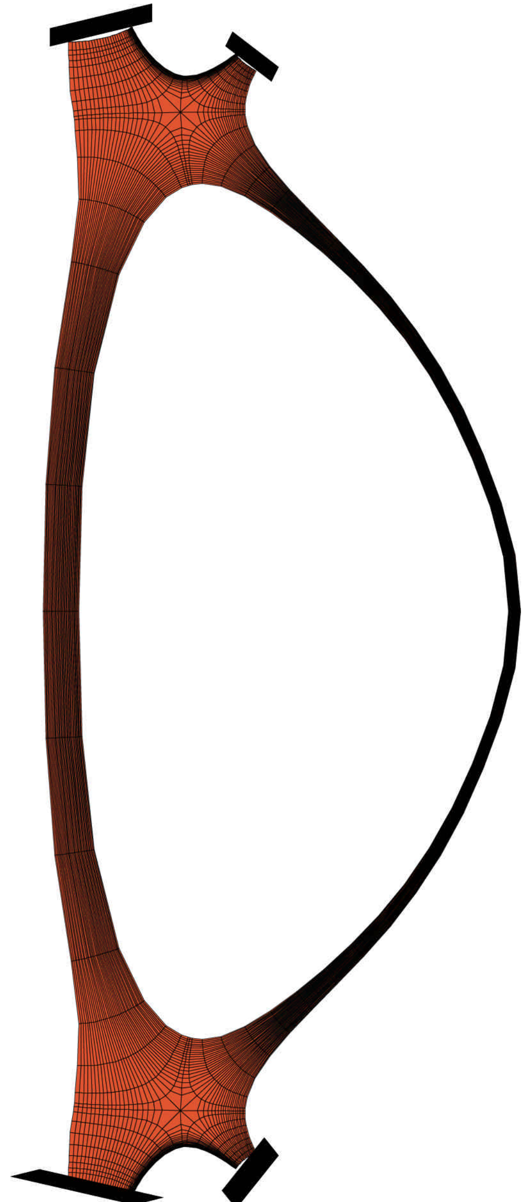
  Mean-free-path > connection length

  Over most of the edge pedestal
- A major undertaking! (requiring additional funding)

3-D Fluid Simulations of Plasma Edge Turbulence BOUT (X.Q. Xu,)

- Braginskii — collisional, two fluid electromagnetic equations
- Realistic ×-point geometry (open and closed flux surfaces)
- BOUT is being applied to DIII-D, C-Mod, NSTX, …
- There is LOTS of edge fluctuation data!

An Excellent opportunity for code validation



The Geometry Really Does Matter: The Resistive X-Point Mode

Resistive ballooning mode

- Driven by
  - p and "bad" curvature
  - Dissipation (disconnects "good" and "bad" curvature)

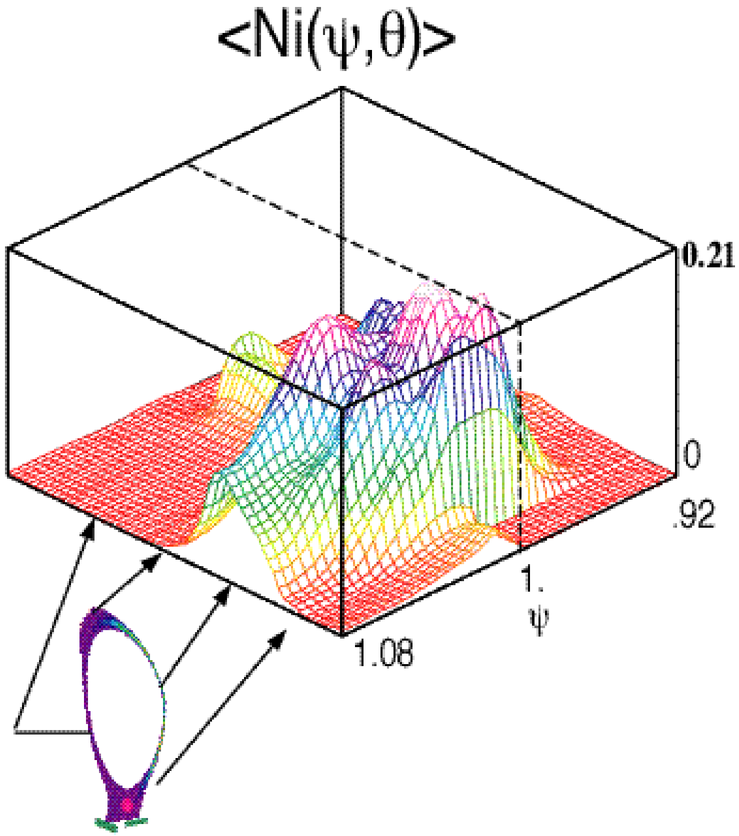
\+ ×-point geometry

- Geometric increase in k

→ Enhanced dissipation

Resistive ×-point mode

- Identified in simulations of DIII-D, C-Mod, NSTX



My Vision for ISOFS

- An interacting suite of codes
  (rather than one giant code)
  - Common user interface
  - Communications between codes
  - Access to experimental & simulation data bases
- Expand existing OFES computational efforts
  (we are already working on the right problems!)
- and start some new ones
  (assuming we actually get significant new funding …)

Recommendations (II)
(how PMP Can Benefit from ISOFS)

(5) Provide user interface from PMP codes to
- Archived experimental data
- Discharge simulations
- …

(6) Provide interface from PMP data analysis and visualization tools to the experimental data base

(7) Provide system for archiving data from PMP simulations.

Major Computational and Applied Mathematical Challenges

- **Continuum kernels** 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 kernels** 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 kernels** 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