Parallelization of a 3D Plasma Fluid Turbulence Model on NERSC's CRAY T3E

Jean-Noel Leboeuf and Vickie Lynch
Fusion Energy Division
Oak Ridge National Laboratory
Oak Ridge, TN 37831-8071

Users Helping Users (UHU) Workshop
NERSC
April 8-9, 1998

Outline

- Motivation
- Model and Numerics
- Parallel Implementation on T3E
- T3E Issues

Motivation

- Part of the Numerical Tokamak Turbulence Project (NTTP), a DoE Phase II Grand Challenge

- Task is to develop fluid models of plasma transport across toroidal (i.e. doughnut-shaped) magnetic fusion confinement devices (e.g. tokamaks) which cover the whole plasma cross section or the full torus

- Full torus models require more memory and compute power than is available on the C90 (80Mw) and J90s (512Mw)

  => Parallel Implementation on the T3E

Model and Numerics: Equations

- The code solves 3 partial differential equations for many radial grid points and Fourier harmonics:

electrostatic potential

$$\frac{\partial}{\partial t}\nabla_{\perp}^{2}\tilde{\Phi}-\frac{|e|B_0^2}{n_{eq}Mc^2}\frac{\partial\tilde{n}}{\partial t}=\frac{|e|B_0^2}{n_{eq}Mc^2}\nabla_{\|}\tilde{V}_{\| i}-\frac{|e|B_0}{n_{eq}Mc}\frac{dn_{eq}}{dr}\frac{1}{r}\frac{\partial\tilde{\Phi}}{\partial\theta}$$

$$-\frac{cT_i^{eq}}{|e|B_0 n_{eq}}\frac{dn_{eq}}{dr}\left(1+\eta_i\right)\frac{1}{r}\frac{\partial}{\partial\theta}\left(\nabla_{\perp}^{2}\tilde{\Phi}\right)-\frac{c}{B_0}[(\hat{z}\times\nabla\tilde{\Phi})\cdot\nabla]\nabla_{\perp}^{2}\tilde{\Phi}$$

parallel ion velocity

$$\frac{\partial}{\partial t}\tilde{V}_{\| i}=-\frac{1}{M}\nabla_{\|}\tilde{T}_i-\frac{T_i^{eq}}{n^{eq}M}\nabla_{\|}\tilde{n}-\frac{|e|}{M}\nabla_{\|}\tilde{\Phi}-\frac{c}{B_0}[(\hat{z}\times\nabla\tilde{\Phi})\cdot\nabla]\tilde{V}_{\| i}$$

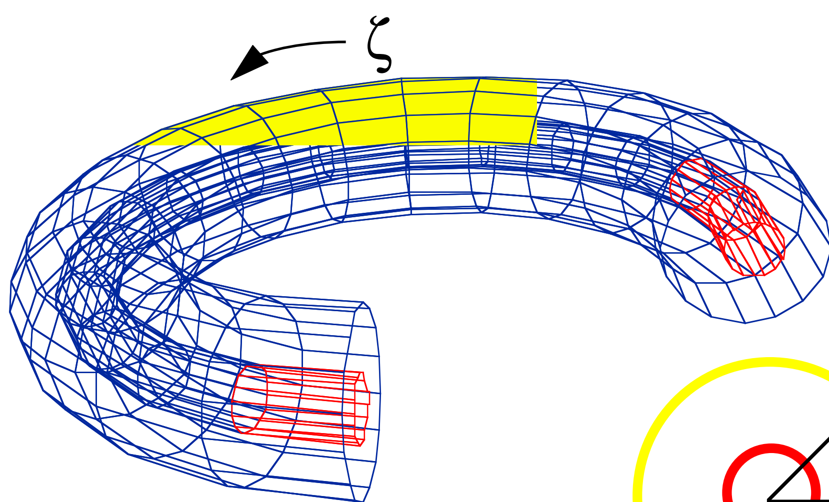
ion temperature

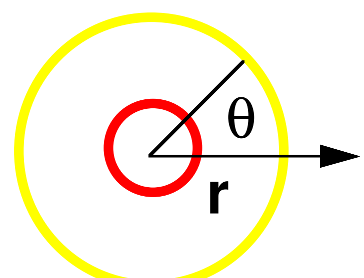
$$\frac{\partial\tilde{T}_i}{\partial t}=-T_i^{eq}\nabla_{\|}\tilde{V}_{\| i}+\frac{c}{B_0}\frac{dT_i^{eq}}{dr}\frac{1}{r}\frac{\partial\tilde{\Phi}}{\partial\theta}-\left[\frac{2^{2/3}}{\sqrt{\pi}}\left|k_{\|}\right|V_{T_i}\right]\tilde{T}_i-\frac{c}{B_0}[(\hat{z}\times\nabla\tilde{\Phi})\cdot\nabla]\tilde{T}_i$$

Model and Numerics

- Finite differences are used for the radial coordinate, r.

- Fourier series expansions are used for the angular variables, θ (short way around torus) and ζ (long way around torus).





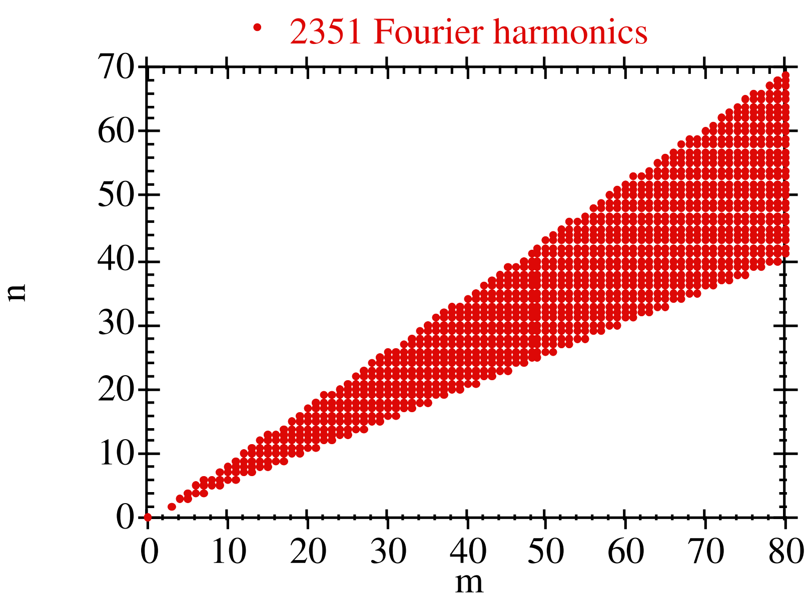
Poloidal cross section

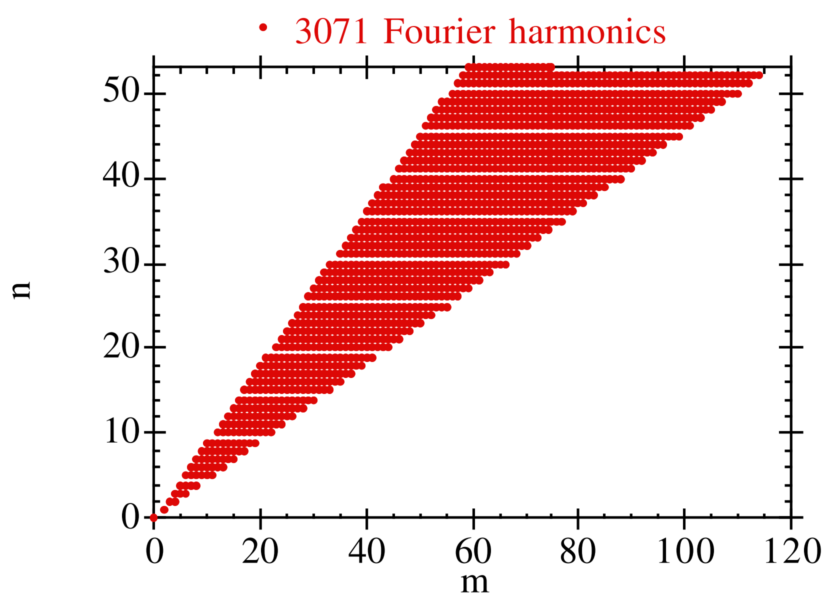
Model and Numerics

- The representation used for the variables in the equations is:

$$\tilde{f}(r,\theta,\zeta)=\sum_{n=0}^{\infty}\sum_{m=0}^{\infty}[f^{c}_{m,n}(r)(\cos m\theta+n\zeta)+f^{s}_{m,n}(r)(\sin m\theta+n\zeta)]$$

- Since the distribution of Fourier harmonics and the size of the radial region are strongly coupled, the calculation uses a narrow wedge of *m* and *n* modes





Model and Numerics

- A two-step second-order-accurate, time-centered advancement, implicit linear, explicit nonlinear, scheme is used.

- Equations: $$L_{m,n}\frac{\partial X_{m,n}}{\partial t} = R_{m,n}X_{m,n} + N_{m,n}(X)$$

- Numerical Scheme:

$$L_{m,n}X_{m,n}^{i+1/2} = (L_{m,n} + \frac{\Delta t}{2}R_{m,n})X_{m,n}^{i} + \frac{\Delta t}{2}N_{m,n}(X^{i})$$

$$(L_{m,n} - \frac{\Delta t}{2}R_{m,n})X_{m,n}^{i+1} = (L_{m,n} + \frac{\Delta t}{2}R_{m,n})X_{m,n}^{i} + \Delta t\, N_{m,n}(X^{i+1/2})$$

Model and Numerics: Numerical Scheme

Linear Terms (L and L+$\Delta t/2$ R)

$\Downarrow$

Implicit

$\Downarrow$

3 Point Finite Differences

$\Downarrow$

Block Tridiagonal Matrices

$\Downarrow$

BTMS by Hindmarsh

(gaussian elimination)

Nonlinear Terms (N(X))

$\Downarrow$

Explicit

$\Downarrow$

Convolutions over Poloidal and Toroidal Fourier Harmonics

$\Downarrow$

Analytic Convolutions

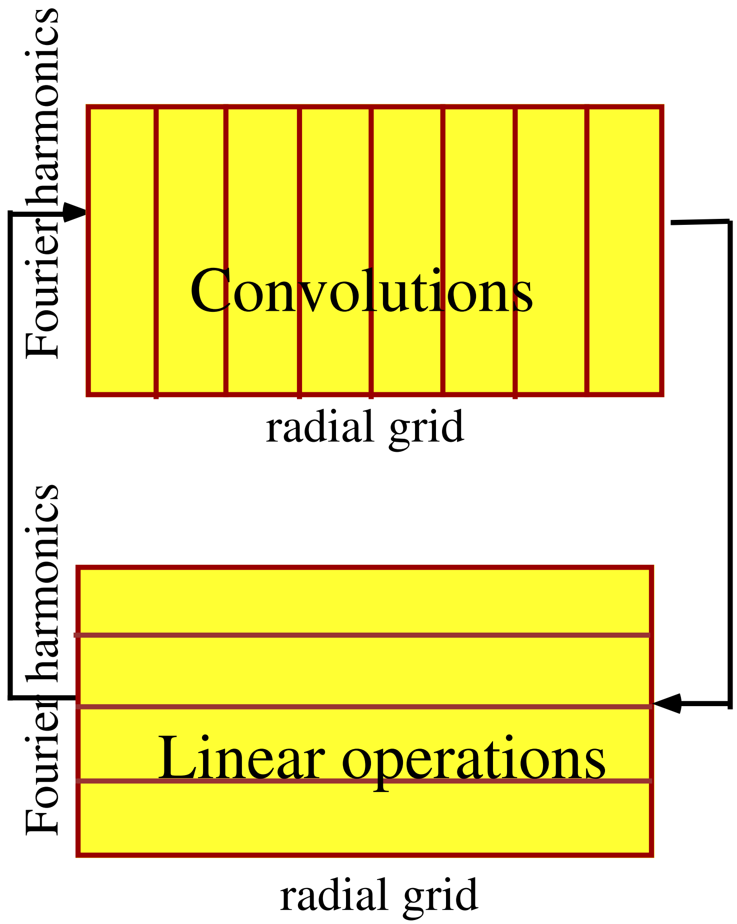
(narrow wedge of harmonics)

Multi-CPU implementation

- For the linear operations, each processor does all of the radial grid for a subset of the Fourier harmonics and the matrix storage is allocated at runtime for the number of processors requested.

- For the nonlinear part of the right-hand-side including the convolutions, each processor does all the Fourier harmonics for a radial slice.

- Global communication follows both types of parallel calculation.



PVM Implementation

- Serial code is replicated on all processors

- Only global communications are for the matrices and the convolutions

- Only the memory of the matrices is divided between processors: The memory is “allocated” at run time according to the number of processors requested

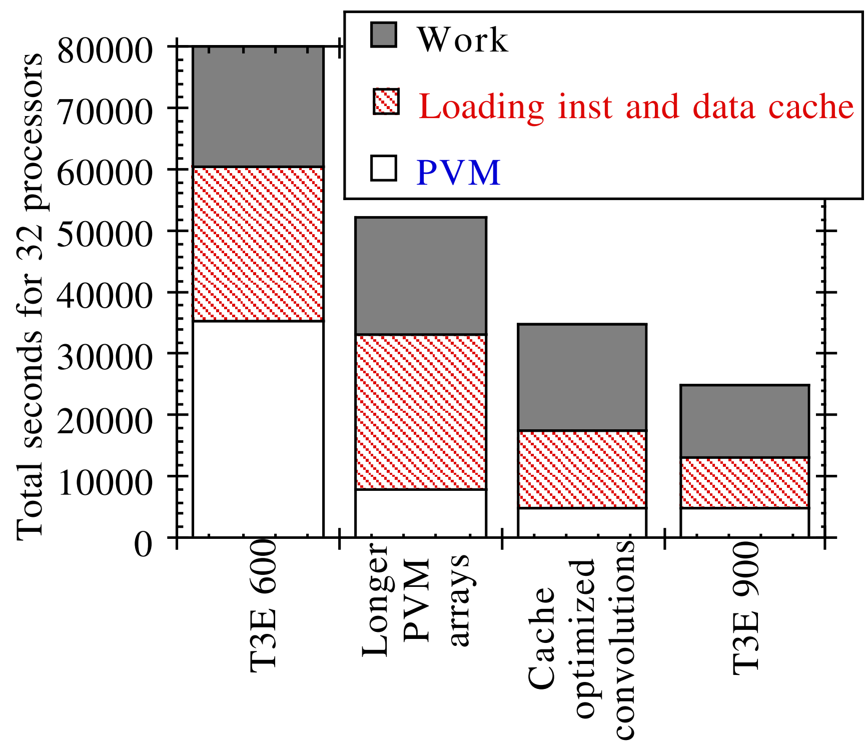
- This means that once the code is compiled any number of processors can be tried without having to recompile

Multi-CPU Optimization

- PVM time was reduced by packing longer arrays for global communications.

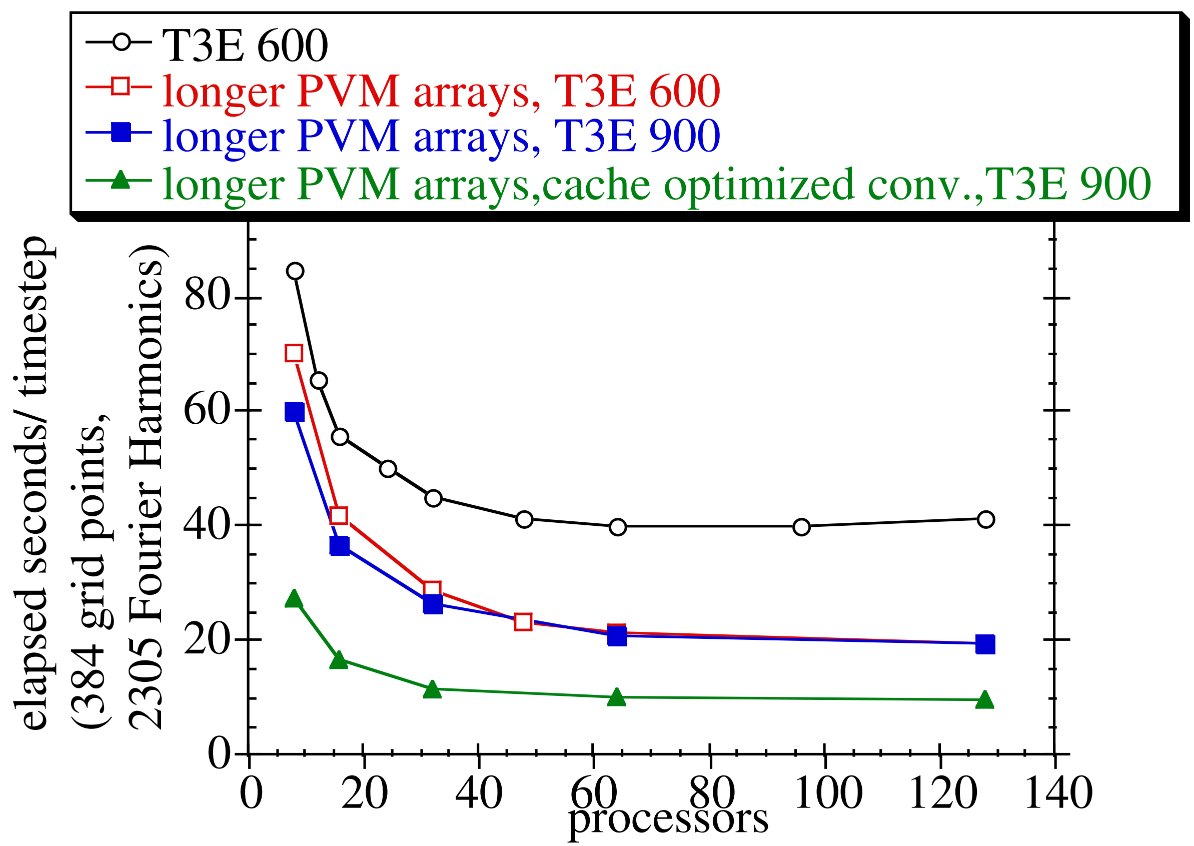
- Loading instructions and data cache time was reduced by making the outer loop for convolutions over the radial dimension for maximum re-use of cache residency.

- The 512 processor T3E-900 model at NERSC has both faster processors and communication.



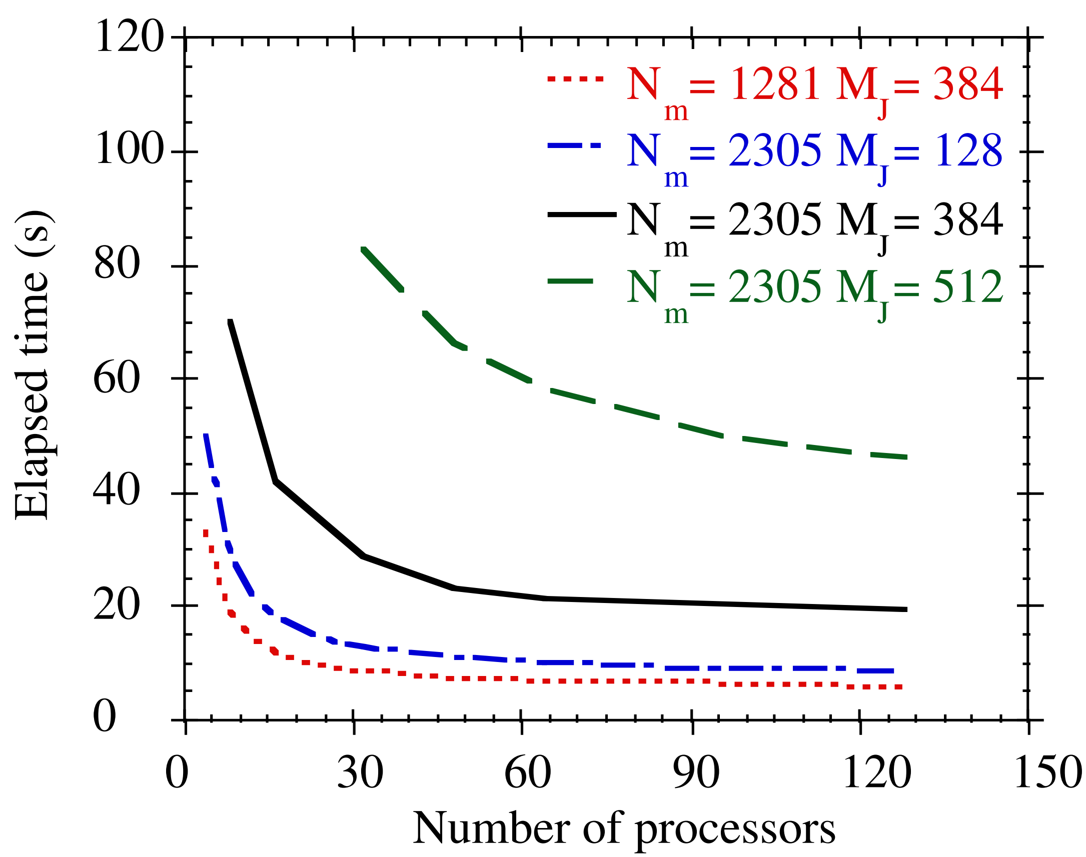
Multi-CPU optimization

- The elapsed time per step decreased from about 40 to 10 with optimizations.
- The optimal number of processors for this problem size increased from about 32 to 64.



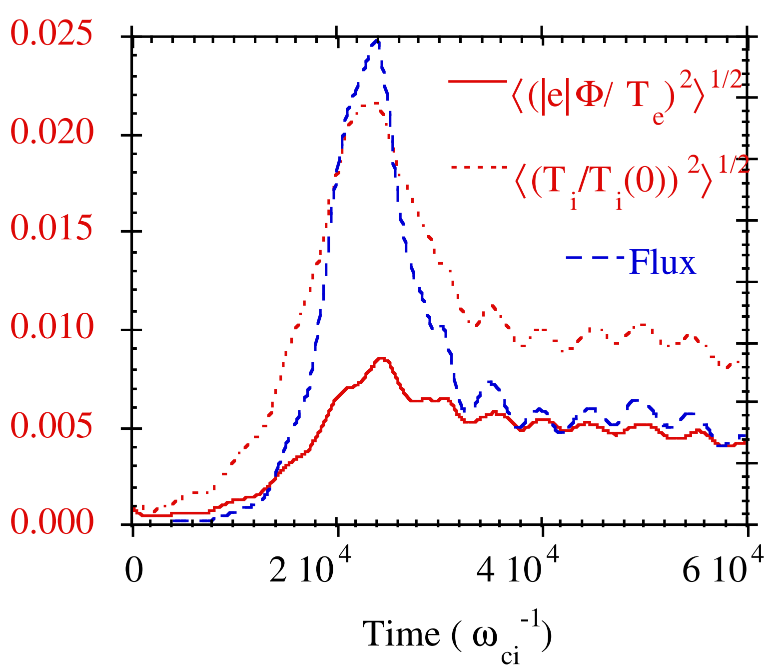
Optimal number of processors

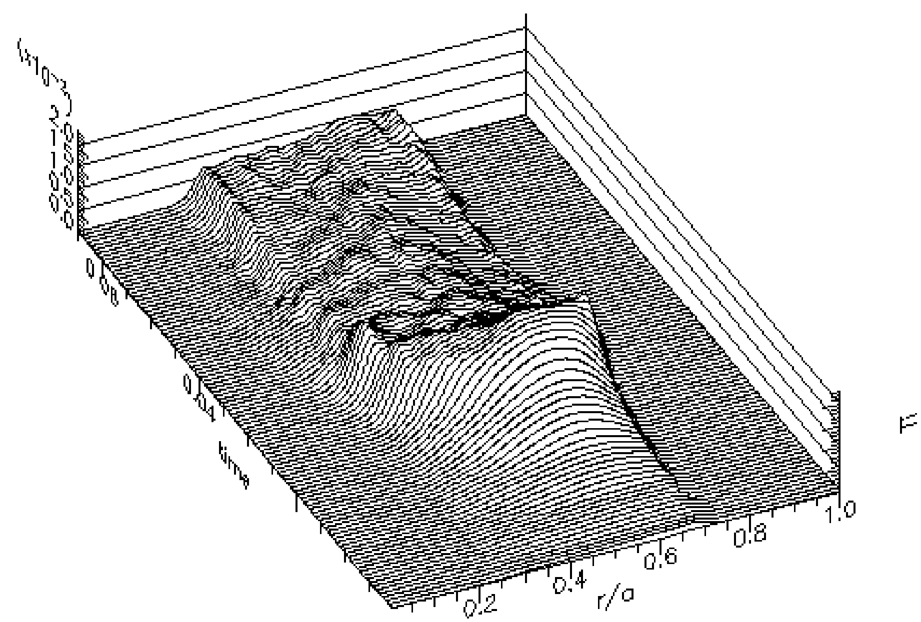
- The optimal number of processors increases with the radial grid and number of Fourier harmonics.
- The optimal number of processors for the largest problem size below is about 128.



Results of nonlinear calculations

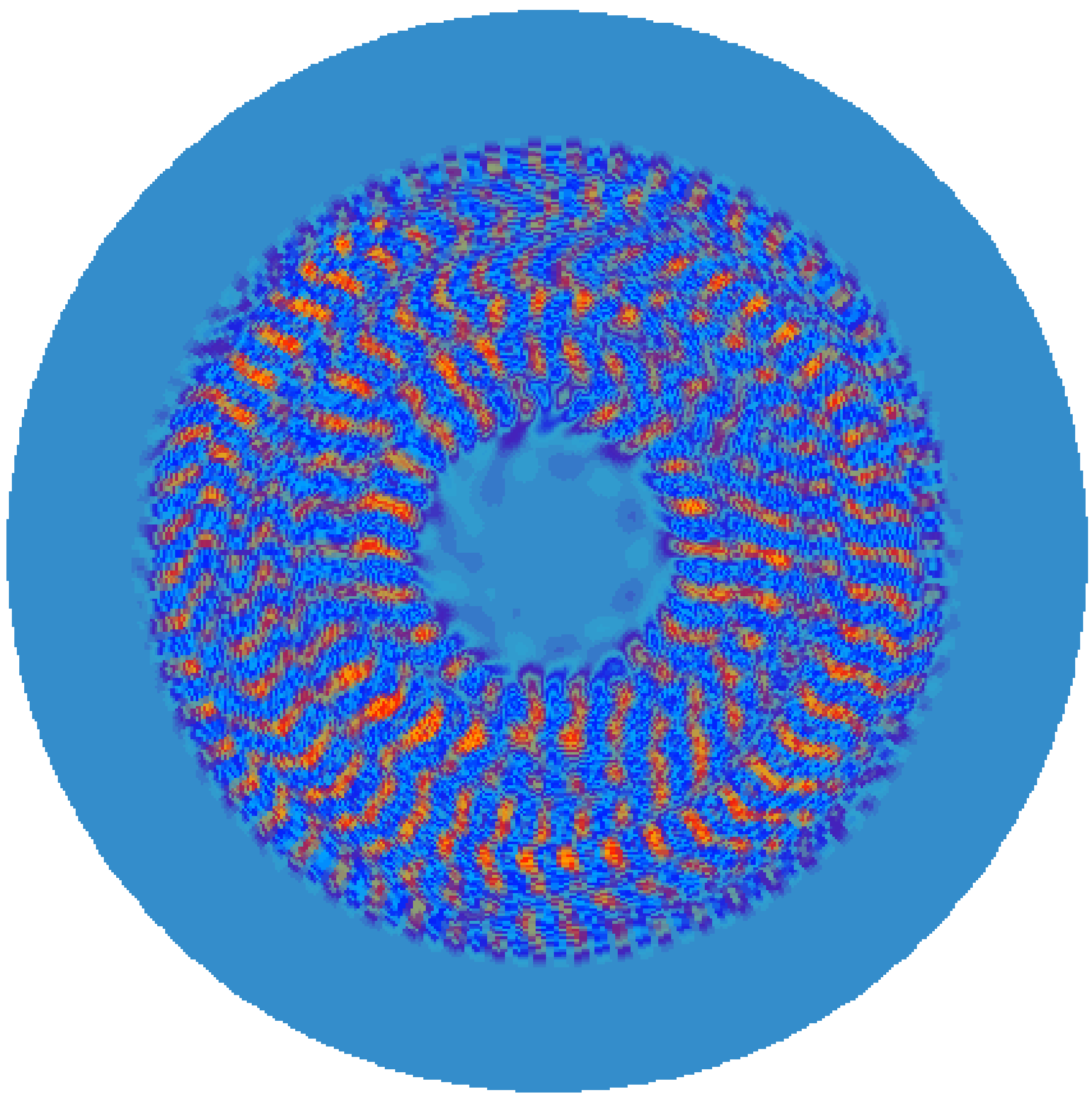
- A full 3-D calculation has been completed on the T3E.
- There is a slow decay of the fluctuations in the nearly steady state phase, but they remain radially localized during the whole nonlinear phase.





Results of nonlinear calculations

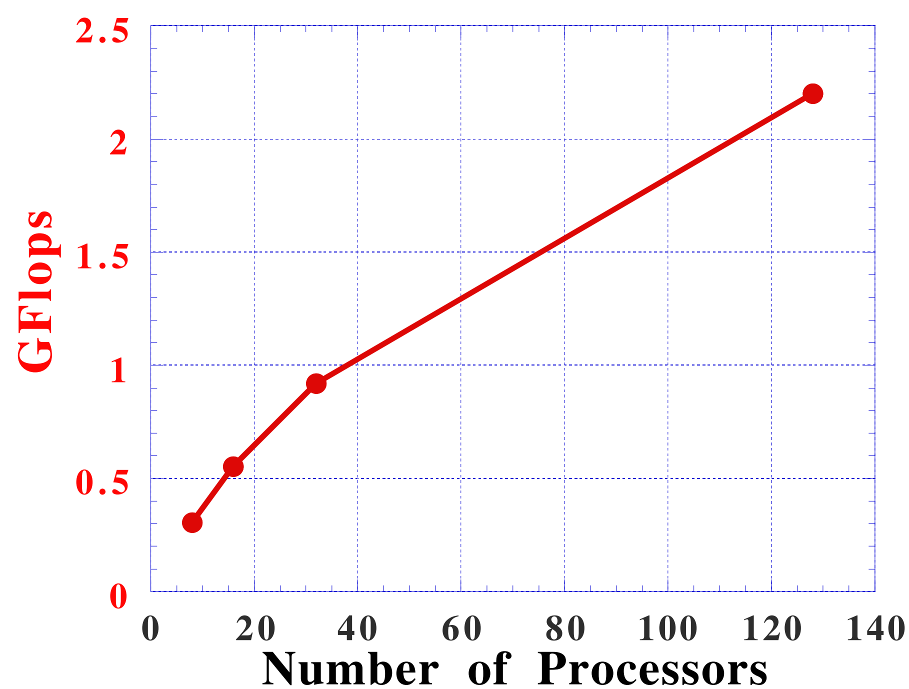
- No large-scale structures are observed, so the turbulence is localized.



T3E Issues: Performance

- Degree of parallelism achieved comparable to that on C90 (Poisson solver remains to be parallelized on the T3E)

- Performance still a factor of ~3 shy of best performance on C90 (We need access to the larger memory of the T3E to perform better resolved calculations)



T3E Issues: Administrative

- Maximum run time in queues other than gc128 and gc256 is inadequate: we get around 300 steps in 15,000 minutes and need hundreds of resubmissions to complete a calculation

- Problem size, hence ability to use more processors efficiently, seriously limited by small memory per node

- Heavy machine load downgrades performance (even though the requested number of processors is locked to our job) by adversely affecting communications

Beyond the T3E

- More memory per node

- Faster communications