NIMROD scaling with parallel closures

2007 Sherwood CEMM Meeting

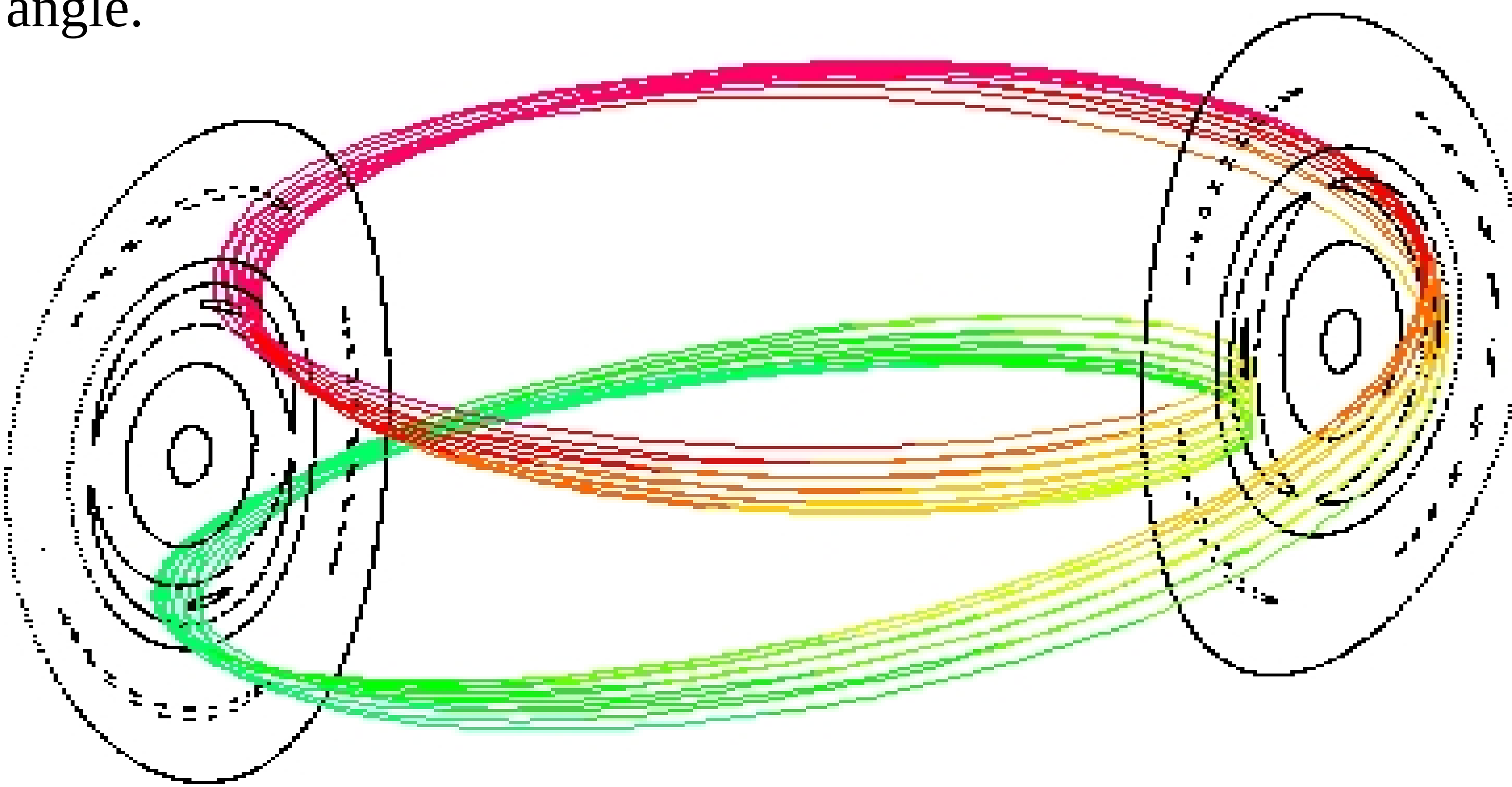
Annapolis, MD

E. D. Held and NIMROD Team

Calculation of parallel closures requires many field line integrations.

24 x 48 grid with quartic finite-elements requires ~ 20,000 integrations at a single toroidal angle.

$$q=\int_L dL'[T(\ell - L') - T(L')]K(L')$$



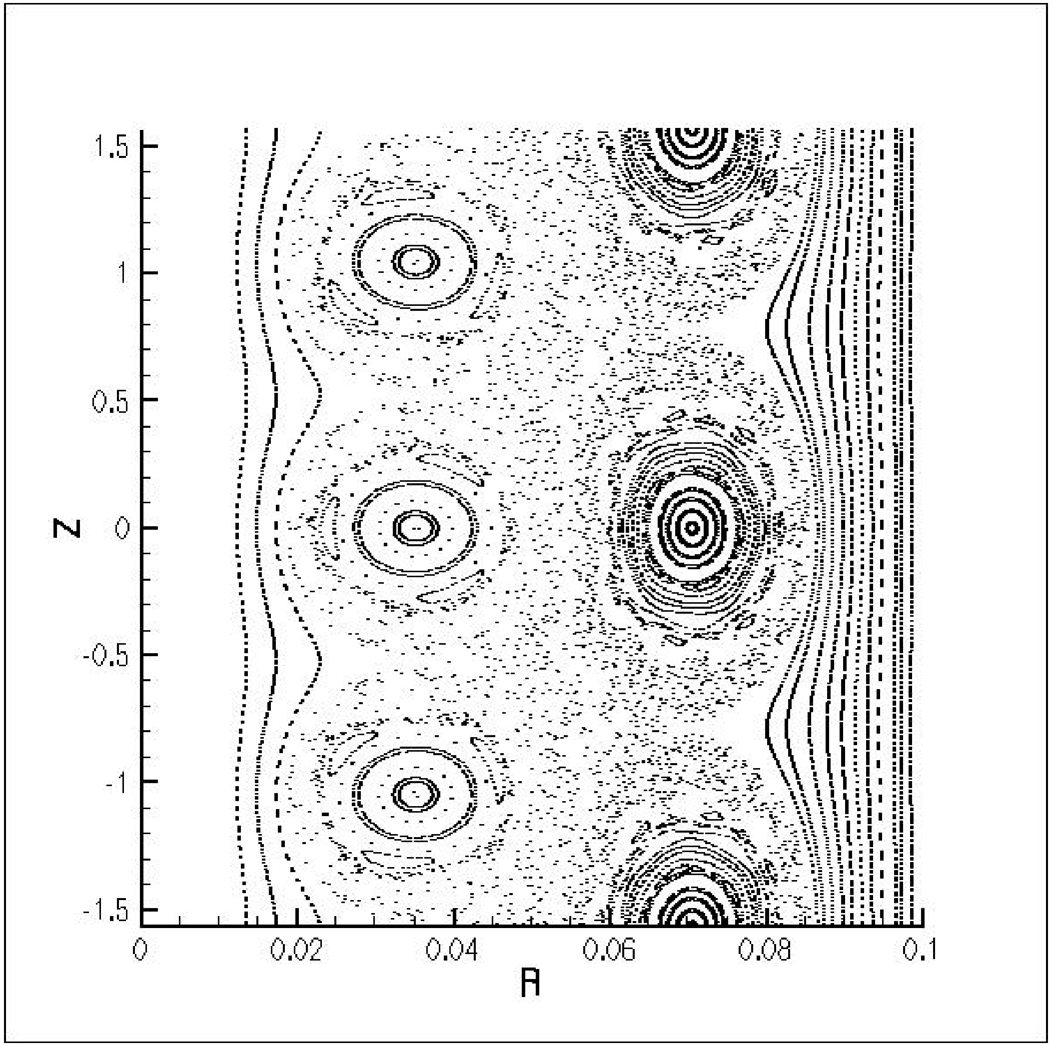
Implementation uses 3 groups of processors.

- Fluid procs: nxbl x nybl x nlayers (100's)
  advance fluid equations;
  exchange fluid and closure data with closure procs.

- Closure procs: nxbl x nybl x nlayers (100's)
  exchange fluid and closure data with fluid procs;
  send global data to slave procs;
  participate in closure calculation with slave procs.

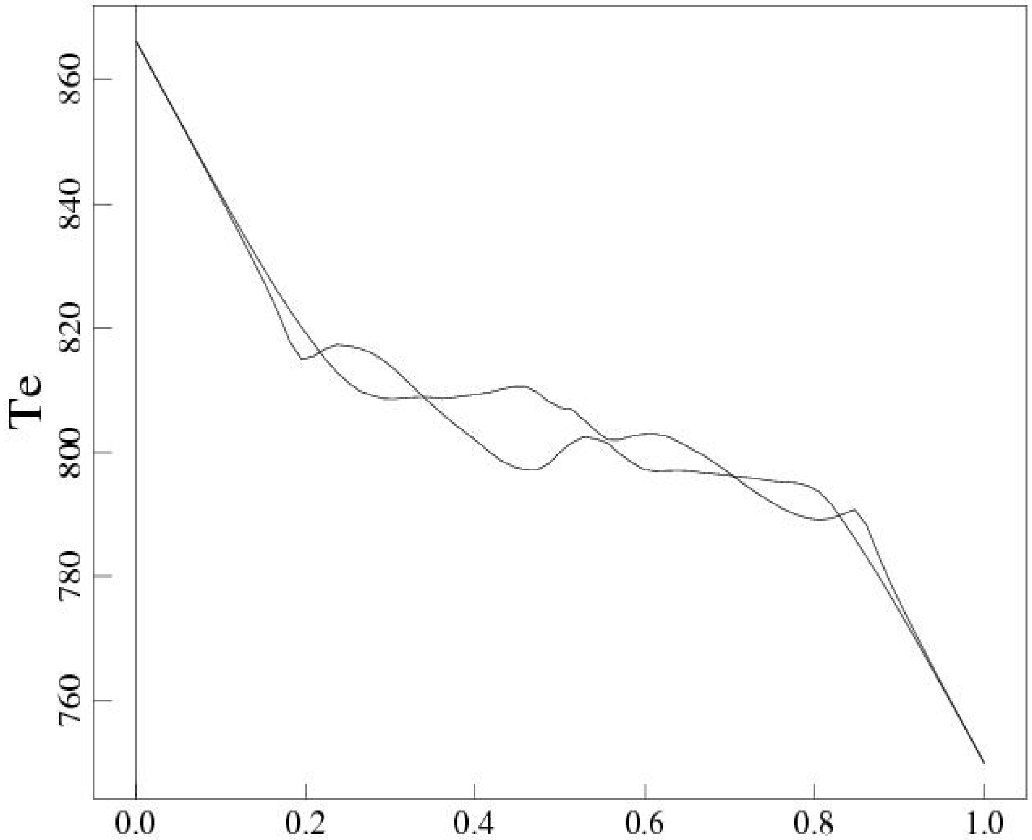
- Slave procs: large block of procs (100's to 1000's)
  receive global data from closure procs;
  perform closure calculation with closure procs.

Use heat transport calculation in slab geometry for scaling study.



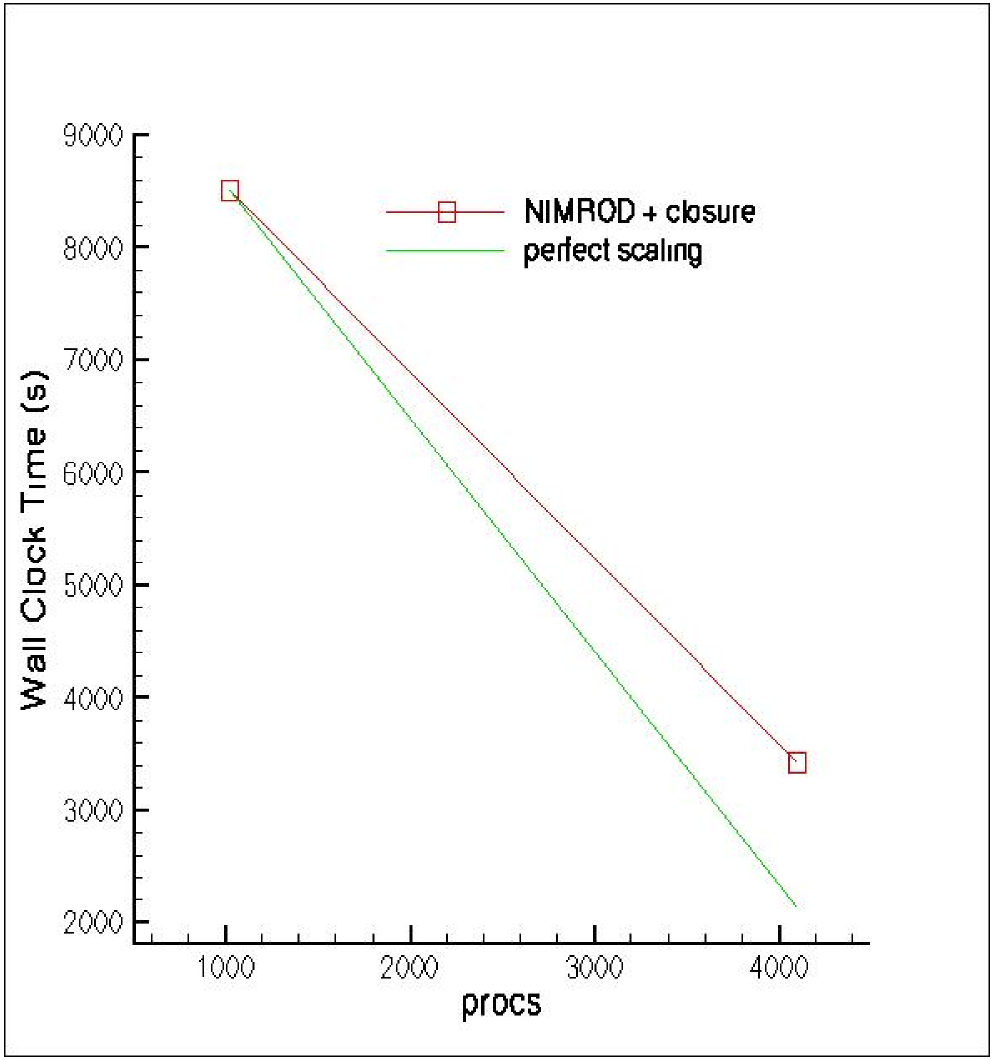
- overlapping 3/2 and 2/1 islands in slab with heat flowing in at left boundary
- 24 x 48 mesh of bicubic FE's, nxbl = nybl = 12
- 11 Fourier modes with nlayers = 1
- fluid_nprocs = 144 close_nprocs = 144

Temperature flattens across stochastic region.



- solution without islands is linear profile from 1 keV at left to 750 eV at right
- significant heat loss evident as rapid parallel transport carrys heat across island and stochastic regions.
- for numerical stability, semi-implicit 🕘$_{\parallel} \sim 10^{9}$

Strong scaling study performed on Seaborg using 1024, 2048 and 4096 processors.



- fixed problem size, 5 time steps with dt = 1.0 s
- fluid procs = closure procs = 144 for all cases
- additional processors used strictly for closure calculation
- field line integrals truncated at 5 km if not yet converged (L ~ 1 km)

Remaining work

- For near steady-state calculations, scaling may be improved by
  (i) advancing fluid equations and the checking if closure calculation is nearly finished
  (ii) if not, estimating time needed to finish and either waiting or advancing again
  (iii) closure processors use existing fluid variables rather than waiting for update.