2008 PSACI report of the FACETS SciDAC Project

Presenter: J.R. Cary†*

†Tech-X Corporation; *University of Colorado

presented at PSACI

June 5, 2008

Project profile

Why core-edge coupling?

Edge componentization and parallelization

Core solver, its improvement through performance analysis, and V&V

Core sources

Framework

Wall

Summary: previous comments and answer to charges

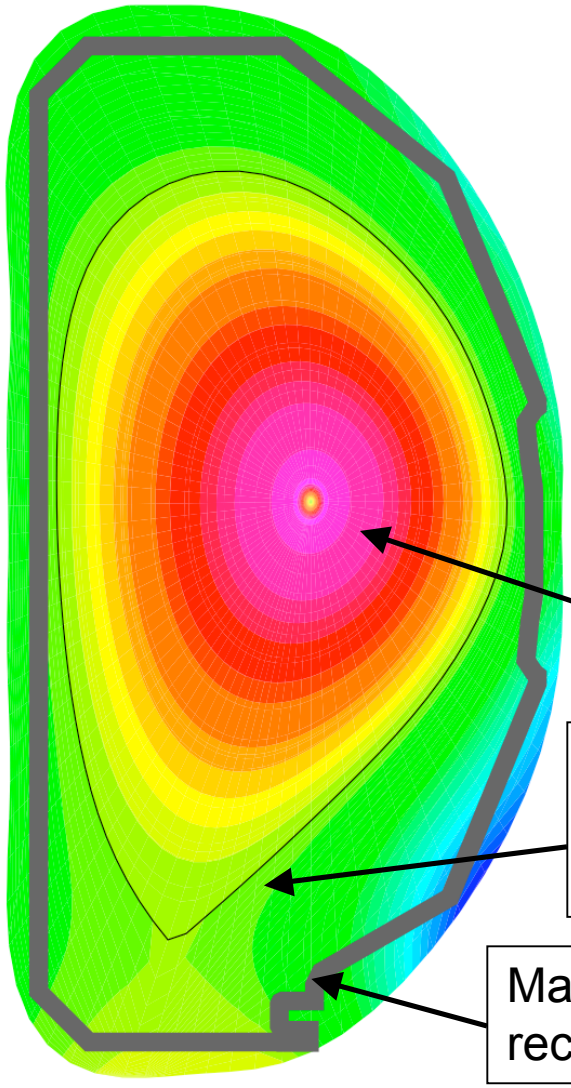
FACETS is a highly collaborative project with many contributors

- ANL (solvers): McInnes, Zhang, Balay
- ANL (coupling): Larson
- CSU (sensitivity research): Estep
- General Atomics (GYRO, turbulence understanding, long-range and/or refinement): Candy
- Lehigh (core modeling, SBIR subcontract): Pankin
- LLNL (edge physics): Rognlien, Cohen
- LLNL (interlanguage): Epperly
- ORNL (modeling, evangelism): Cobb
- ORNL (SSGKT): Fahey
- ParaTools (performance analysis): Maloney, Morris, Shende
- PPPL (core sources): McCune, Indireshkumar
- UCSD (wall): Pigarov
- Tech-X (framework, core): Cary, Hakim, Kruger, Pletzer, Vadlamani, Miah

FACETS is one of the newer projects in the SciDAC portfolio

- **CORRECTION ON TIME SCALE: FACETS was started in Jan. 2007. Now at 1.4 yrs.**
- **Building new**
  - Core solver
  - Framework
  - Engineering infrastructure (build, test)
- **Reusing**
  - Core sources
  - Core flux models
  - Edge
  - Wall
- **In collaboration with**
  - TOPS/PETSc
  - TASCS/Babel
  - PERI/TAU
- **Researching for future: coupling, sensitivity**

FACETS goal: tight coupling framework for core-edge-wall



- Coupling on short time scales
- Inter-processor and in-memory communication
- Implicit coupling

Hot central plasma: nearly completely ionized, magnetic lines lie on flux surfaces, 3D turbulence embedded in **1D** transport

Cooler edge plasma: atomic physics important, magnetic lines terminate on material surfaces, 3D turbulence embedded in **2D** transport

Material walls, embedded hydrogenic species, recycling

Why do core-edge coupling?

- Determine pedestal height for density and temperature profiles at core-edge interface
- Pedestal height strongly impacts fusion reactor performance
- L-H transition, ELMs, and transport barrier are dynamical processes requiring core-edge coupling for quantitative predictions
- Neutrals released by wall can penetrate core
- Coupled simulations vs monolithic codes to exploit space and time scale disparities and to make use of proven techniques for incorporating important physics in each region

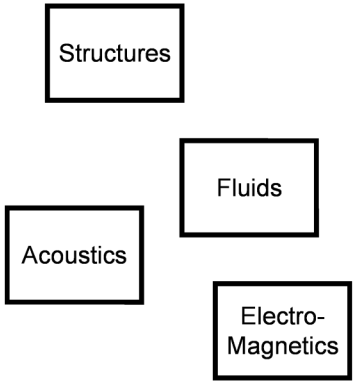
Why do reduced-model transport simulation?

- Would like to get to 1000 simulation second in one wall clock hour
- GYRO simulation: 1 ms on D3D takes 1 hour on 128 procs for well resolved.
- Off by 1e6 at the present time. Makes this a long-range research area (more parallelism, …)
- Can be used for refinement (see next talk on SSGKT)

FACETS is moving towards the "tight-coupling" framework

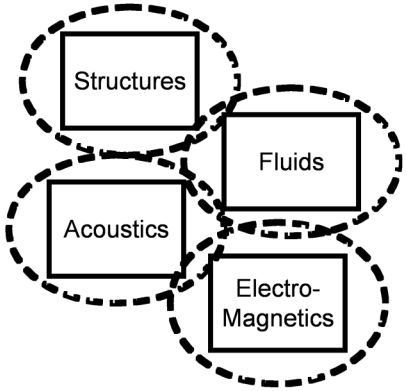
EVOLUTION OF APPLICATION PHYSICS SOFTWARE

Minimal Component Interoperability:



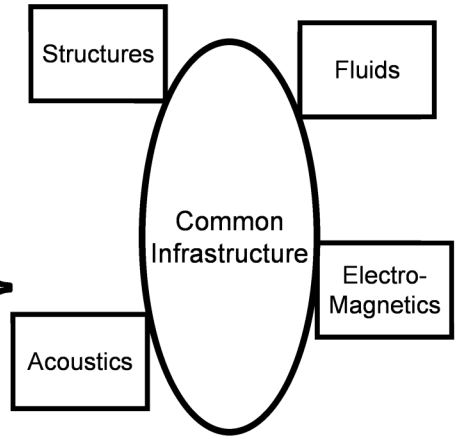
- Physics models are completely uncoupled.
- May exchange static datasets through flat files.

Shallow Component Interoperability:



- Physics models are loosely coupled.
- Data management and parallelism is independent in each module.
- Exchange common data events via wrappers (web services, etc.).

Deep Component Interoperability:



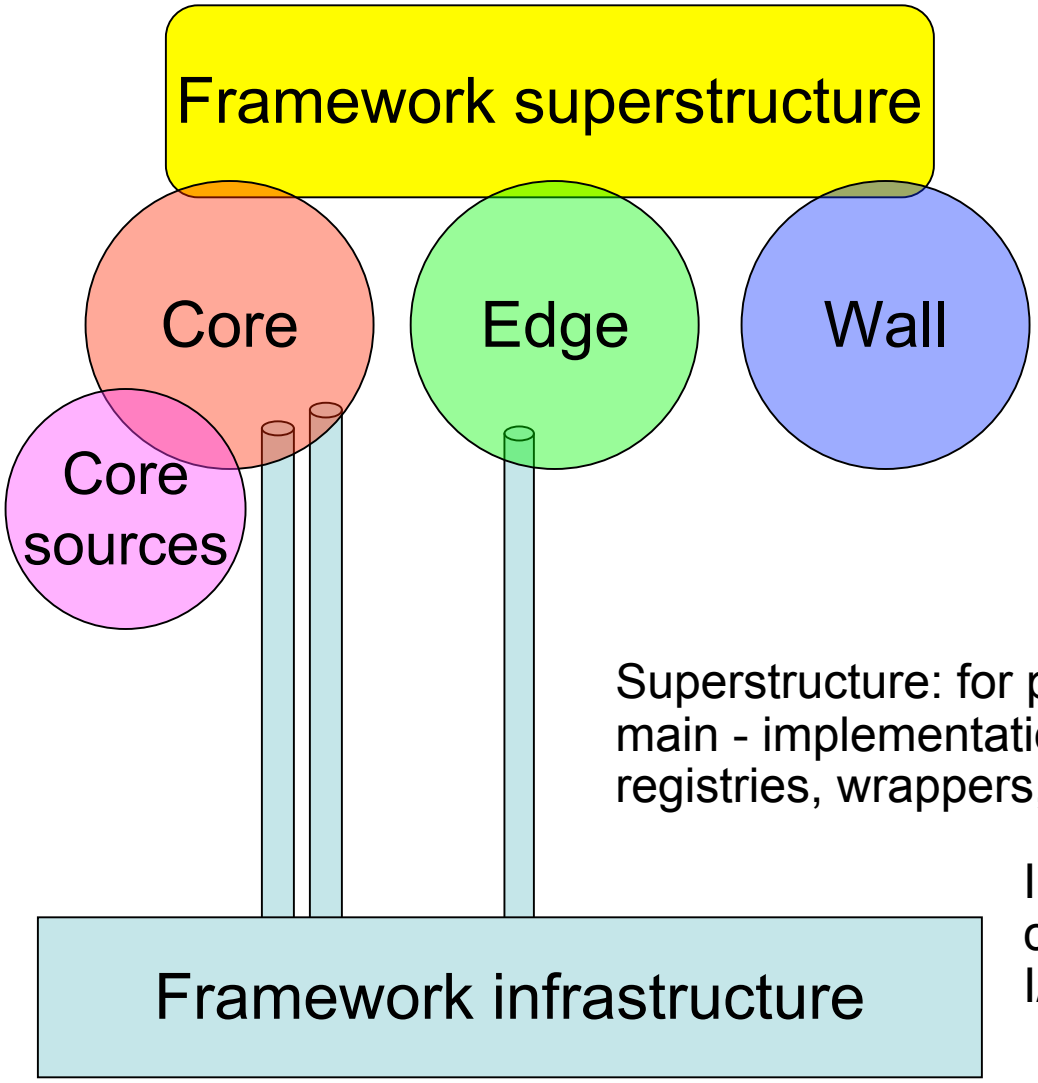
- Physics models are tightly coupled.
- Data exchange across shared service infrastructure.

Time

from Shalf, ASCAC Boulder, Apr. 30, 2008

FACETS

Reuse of legacy code requires a combination of loose and tight



- **Simple, common interfaces**
- **Native components: extended interface, built on FACETS infrastructure**
- **External components: basic interfaces, importation of pre-written codes**

Superstructure: for pulling stuff into main - implementation and object registries, wrappers, …

Infrastructure: for building new components - grids, messengers, I/O, containers, …

Edge team has updated the UEDGE "external component"

- **Team members from multiple institutions contributing:** **Tom Rognlien, Ron Cohen (LLNL); Lois Curfman McInnes, Hong Zhang (ANL)**

Table of authors: Last month

No	Author	Number	Percentage
1	trognlien	24	37.50%
2	hzhang	15	23.44%
3	cary	14	21.88%
4	mmiah	9	14.06%
5	rcohen	2	3.12%

- **In collaboration with TOPS CET**
  - Satish Balay (ANL), David Keyes (Columbia), Barry Smith (ANL)
- **Initial focus on UEDGE**
  - Incorporated PETSc interfaces for nonlinear solvers
  - Reintroduced parallel simulations

Parallel UEDGE Solvers: TOPS Partnership

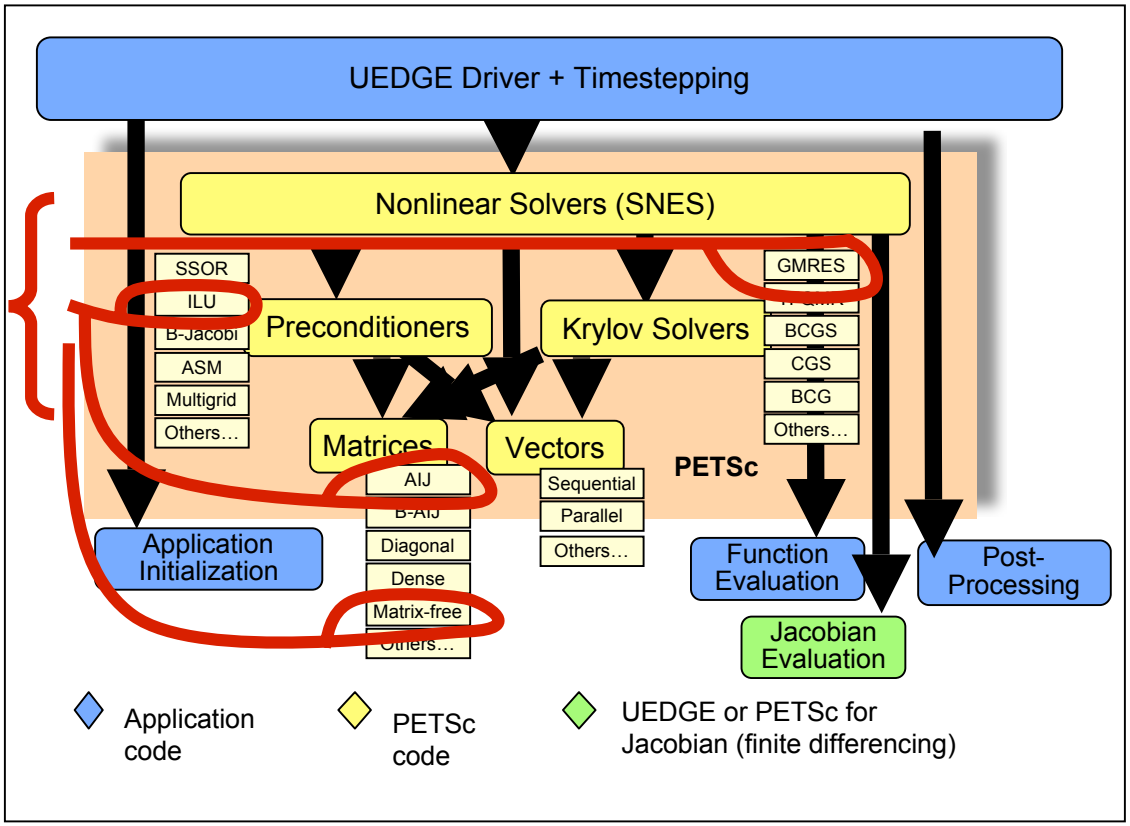
Solve F(u) = 0: Fully implicit matrix-free Newton-Krylov methods

$$F'(u^{k-1})\,\delta u^{k} = -F(u^{k-1})$$
$$u^{k} = u^{k-1} + \delta u^{k}$$

TOPS
Towards Optimal Petascale Simulations

- SciDAC TOPS Center for Enabling Technology (PI: David Keyes) www.scalablesolvers.org
- Nonlinear solvers in PETSc: www.mcs.anl.gov/petsc
- Explore scalable parallel preconditioners
- Can we increase robustness for cases with difficult nonlinearities?
- Get set for coupled core-edge models

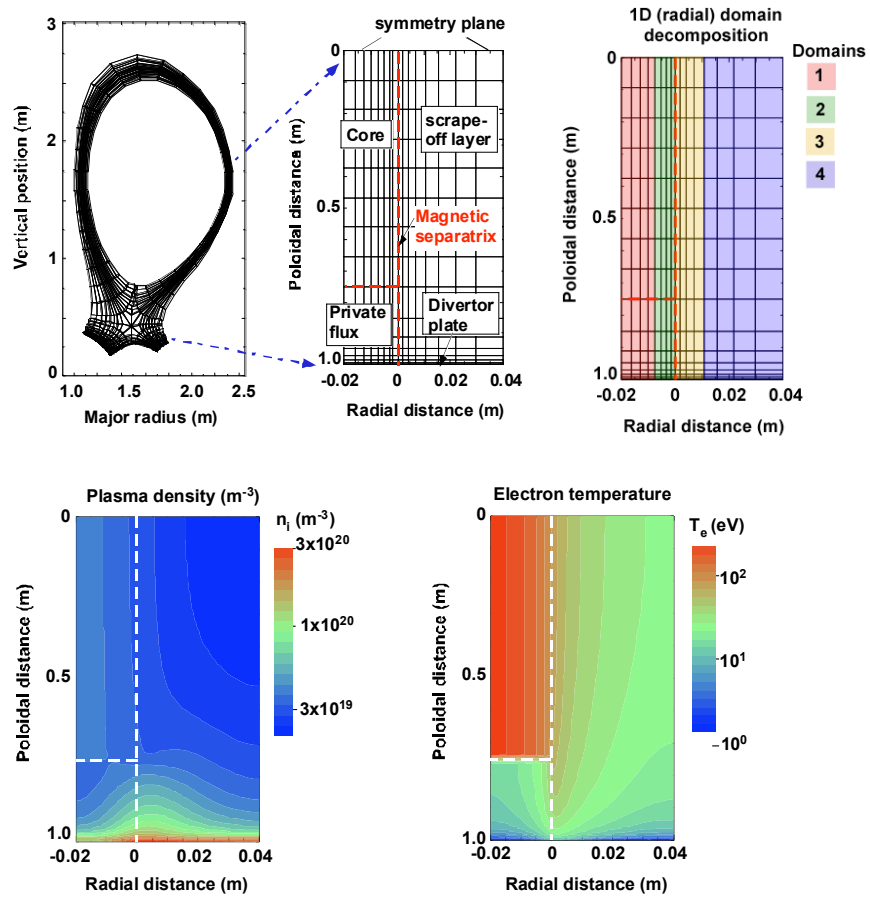
Algorithms and data structures originally employed by UEDGE



UEDGE Test Case 1

Full toroidal geometry is typically used, but initial parallel UEDGE tests with PETSc employ a simplified half-space slab

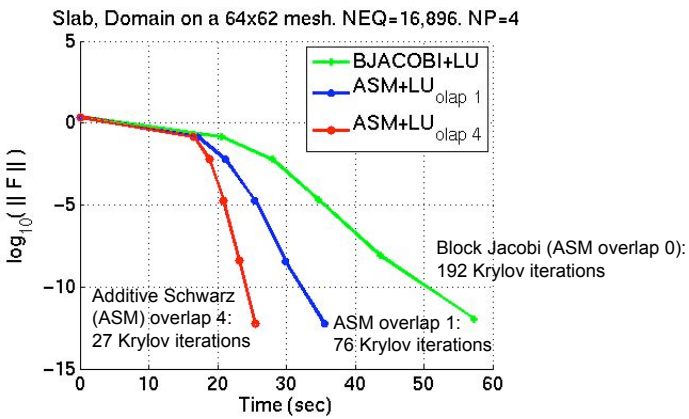
- **Outer midplane/ divertor regions are mapped to a slab with B_p/B_t = constant**
- **Same features such as closed and open B-field lines, private flux region, and divertor recycling are retained**
- **Decomposing radially is simplest because there are no internal special surfaces in that direction; 2D decomposition coding revived and being debugged**
- **Typical solutions show structure of plasma variables in edge region**
  - Ions recycle as neutral gas at the divertor (lower surface) to increase plasma density there
  - Electron heat loss at the divertor and ionization losses decreases temperature there



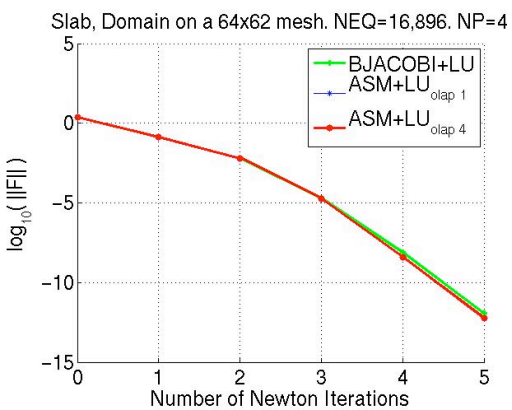
Preliminary Parallel Performance Results

Algorithmic Comparisons: Matrix-free Newton using restarted GMRES with various preconditioners

Function Norm vs. Time



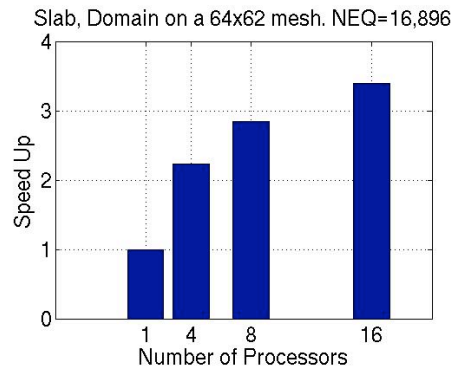
Function Norm vs. Nonlinear Iteration



Computational Environment: Jazz @ ANL: Myrinet 2000 network, 2.4 GHz Pentium Xeon processors with 1-2 GB of RAM.

(64+2)x(62+2) mesh with 4 unknowns per mesh point: problem dimension 16,896

Initial Scalability (Total Simulation Time)



Work in progress: Exploring algorithmic options and scalability issues for various test cases

UEDGE brought into FACETS through Babel

- Babel: interlanguage tool (Fortran, C, C++, Python)
- Interfaces defined by
  https://facets.txcorp.com/facets/wiki/InterfacesAndNamingScheme
- SIDL written by Epperly
- Corresponding python interface written by R. Cohen used for implementation
- Problems:
  - Python generally not available on leadership-class machines (without A LOT of work)
  - Slower
- Solution: build Babel from .v files to have direct calls to Fortran

Core: fast new core solver developed, V&V proceeding well

$$\frac{\partial Q}{\partial t} + \frac{\partial F}{\partial r} = S(r,t)$$

Fluxes strongly nonlinear function of values, gradients

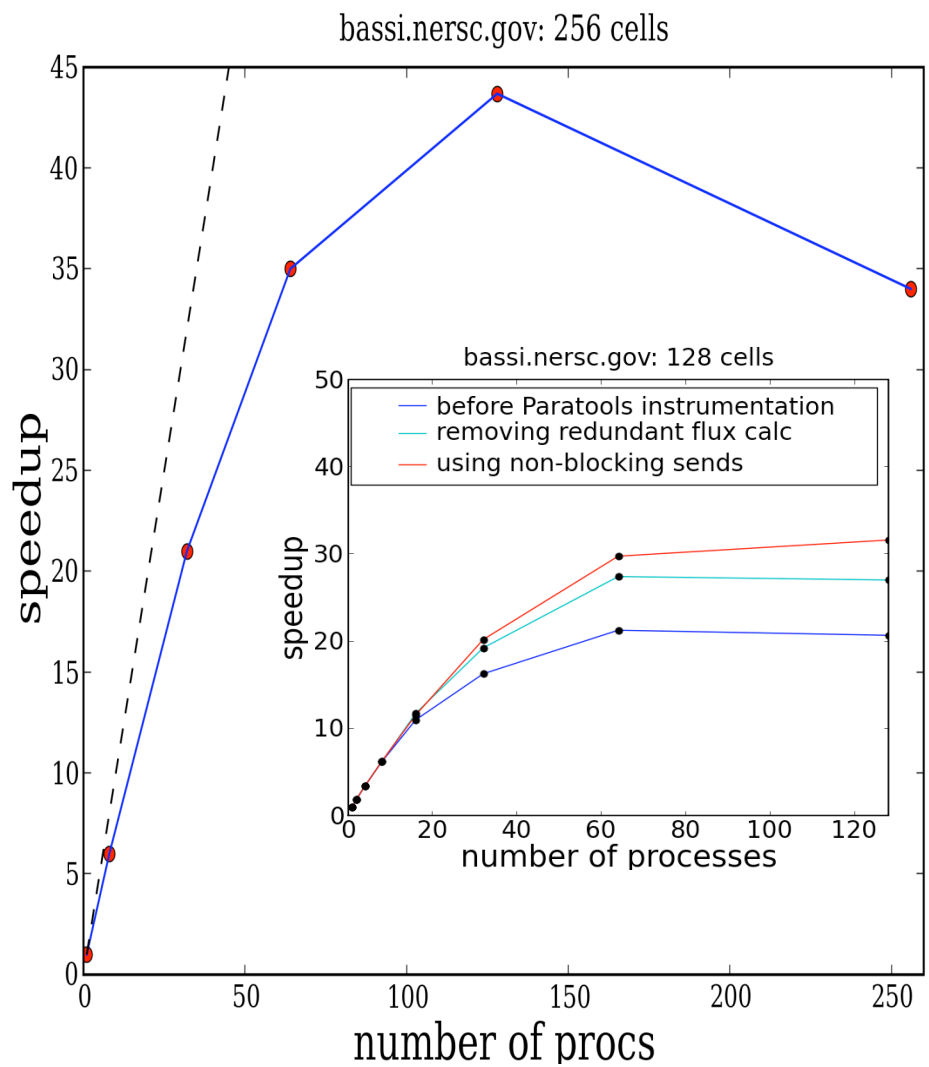
- **Well known in core modeling community that taking large time steps is difficult – convergence failure.**
- **Observation: core solves converge well at lower resolution.**
- **Implemented "nested iteration"**
  - Find solution at coarser resolution
  - Interpolate to get guess for finer resolution
- **Result: able to increase time step by 500x (ASTRA: 2.e-5s; FACETS >10ms) 7x perf.**
- **Next step: combine with block Broyden**

Core solver relies on other developments

- **Fusion Modernization and Componentization of Fusion Modules (SBIR Phase II)**
  - Common interface for MMM95, NCLASS, GLF23: F90 and C
  - Parallelization of GLF23 over wavenumbers
- **Benefits FACETS and SWIM**
- **Phase II extension would allow**
  - extension to GYRO
  - parallelization of TGLF (30x more computation than GLF23) over wavenumbers

Core: Parallel core solver developed, then improved

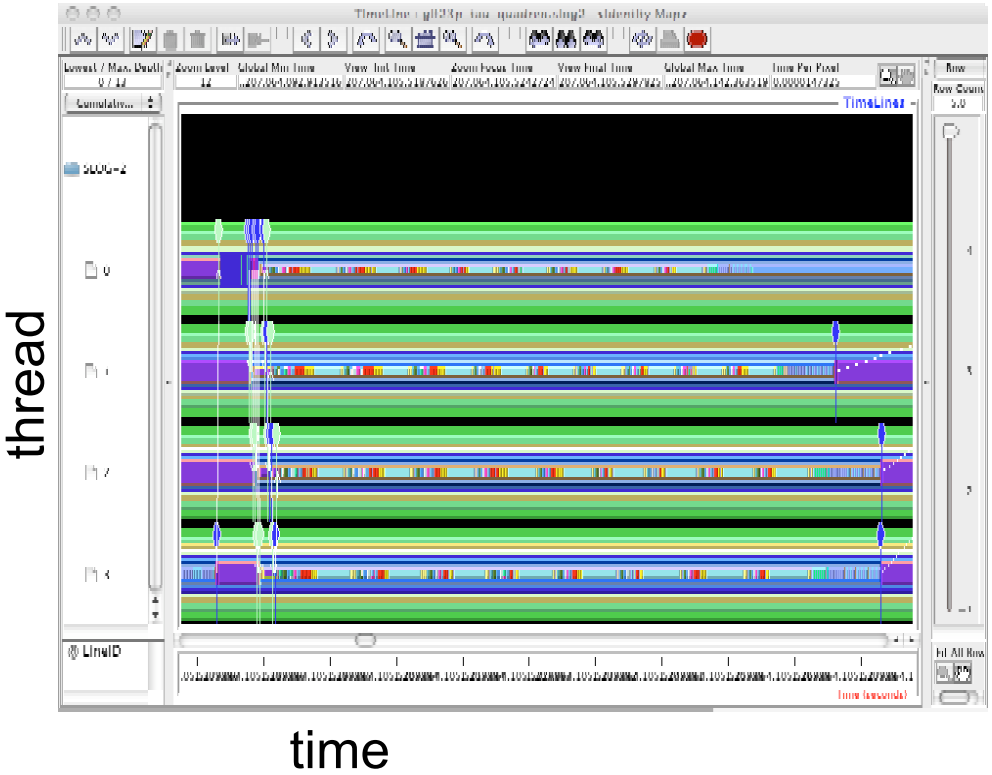
- Distributed flux calculations, matrix solves on rank 0
- Bassi
- 128 cells
- Expected 1 cell per proc, since GLF23 so compute intensive
- Initially found 20x
- Improved to 30x (45x on 256 procs)



Core: Performance analysis helping isolate implementation issues

- Better able to see on 4 procs
- 32 cells, so 31 flux calculations
- 34 calculations seen.

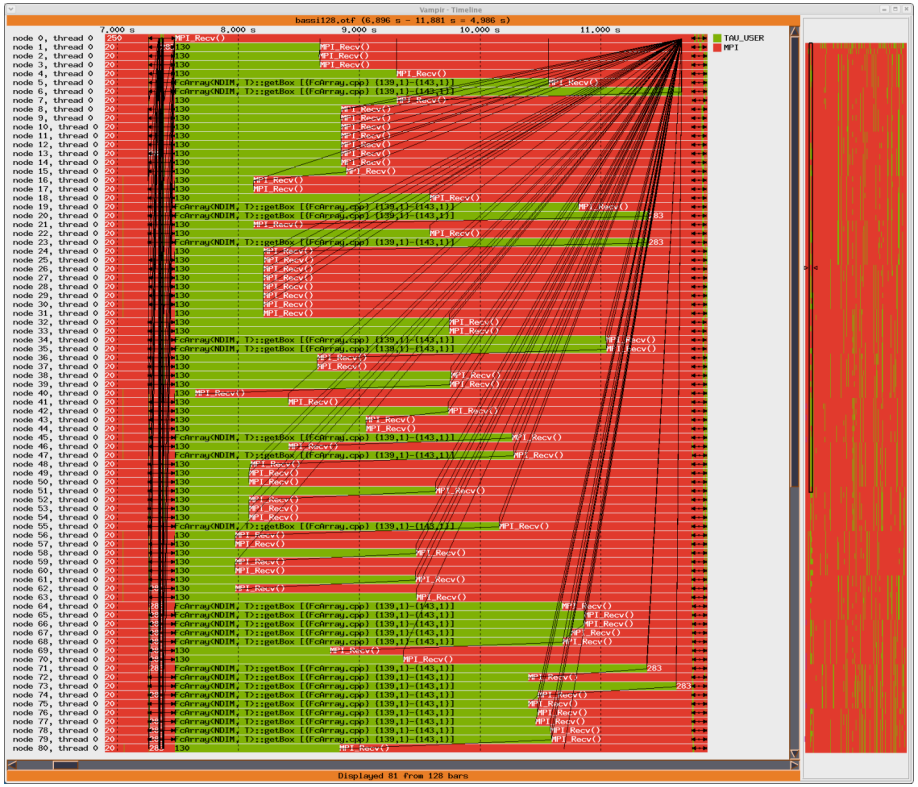
DOH! we were calculating the edge fluxes on both procs!



Jumpshot visualization of trace on 4 proc SMP
Hints of load imbalance

Core: Now looking at large proc counts to isolate problems (ParaTools)

- First 80 traces of 128.
- On a few surfaces, flux calculations taking a long time.
- All others in message state.



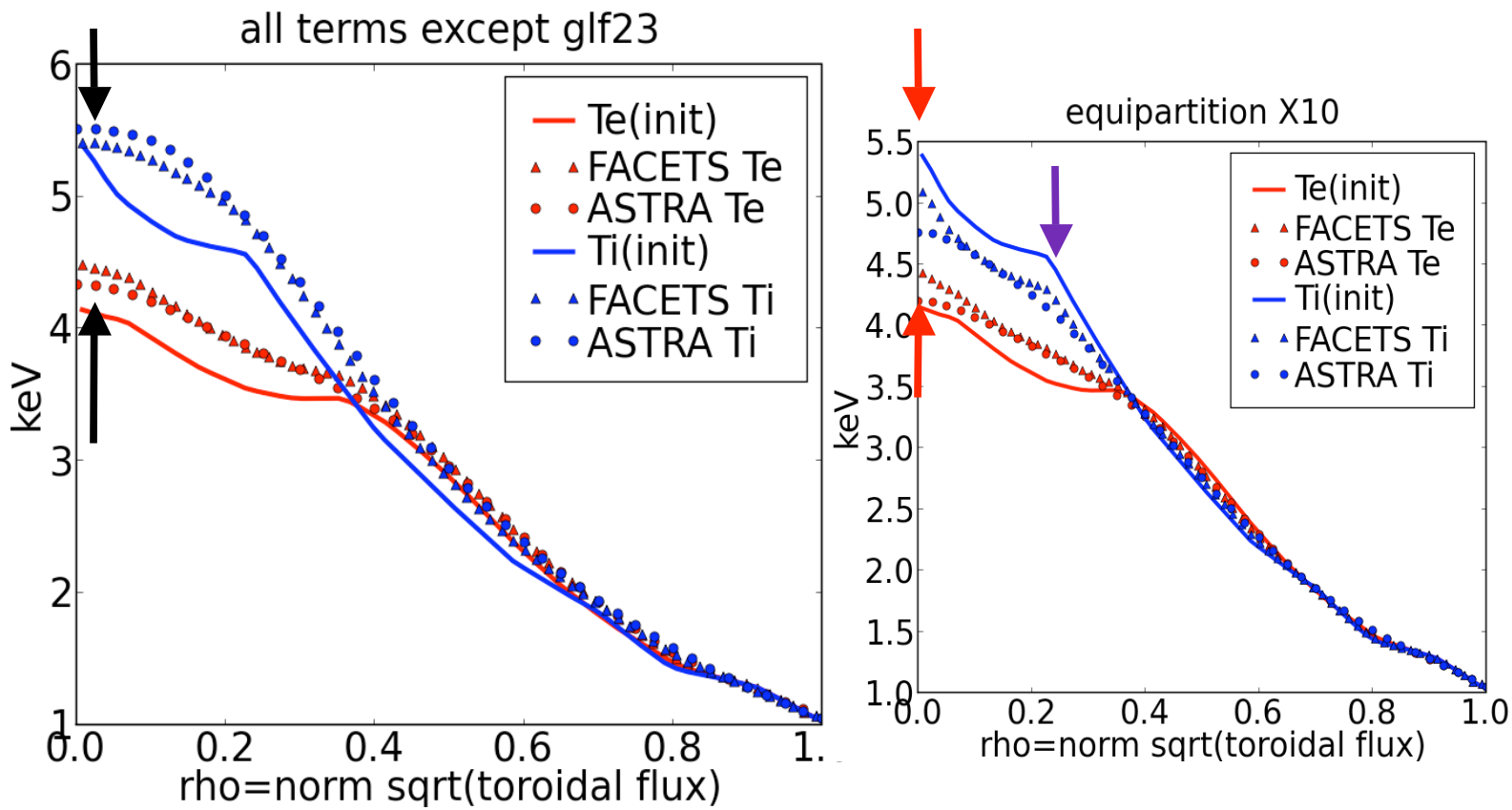
- Parallel GLF23 with load balancing should be able to effectively use 32*8 + 32*2 + 64 = 384 procs for 128 cells. Probably twice that many for 256 cell simulations (expected for ITER).

Core: Verification efforts underway: comparison with ASTRA

- **Systematic approach (Pletzer/Pankin/Kruger)**
  - Case 2: GLF23 only, no neoclassical, no heating
  - Case 3: Neoclassical only, no beams
  - Case 4: Beams only
  - Case 5: Beams and radiation
  - Case 6: Equipartion terms are on
  - Case 7: Ohmic heating is on
  - Case 8: Beams, radiation, neoclassical, and equipartion
  - Case 9: Case 8 + GLF23
  - Case 10: Equipartion terms times 10
  - Case 12: GLF23 only (case 2) without smoothing, unbounded chi
  - Case 13: The same as case 12, but only one small time step (dt=1.e-10)
  - Case 14: Case 9 without smoothing (10 msec)
  - Case 15: Case 9 without smoothing (1 msec)
- **Required additional implementations**
  - Equipartition
  - Neoclassical diffusivity (Chang-Hinton, 1986)
  - Implemented UFILE reader for
    - Initial profiles
    - Boundary conditions
    - Input powers
  - Metric coefficients from EFIT/ESC
- **Multiple issues resolved**
  - ASTRA smoothing
  - Flux coord normalization
  - Definition of minor radius (midplane versus sqrt(tor flux/ πB_T)

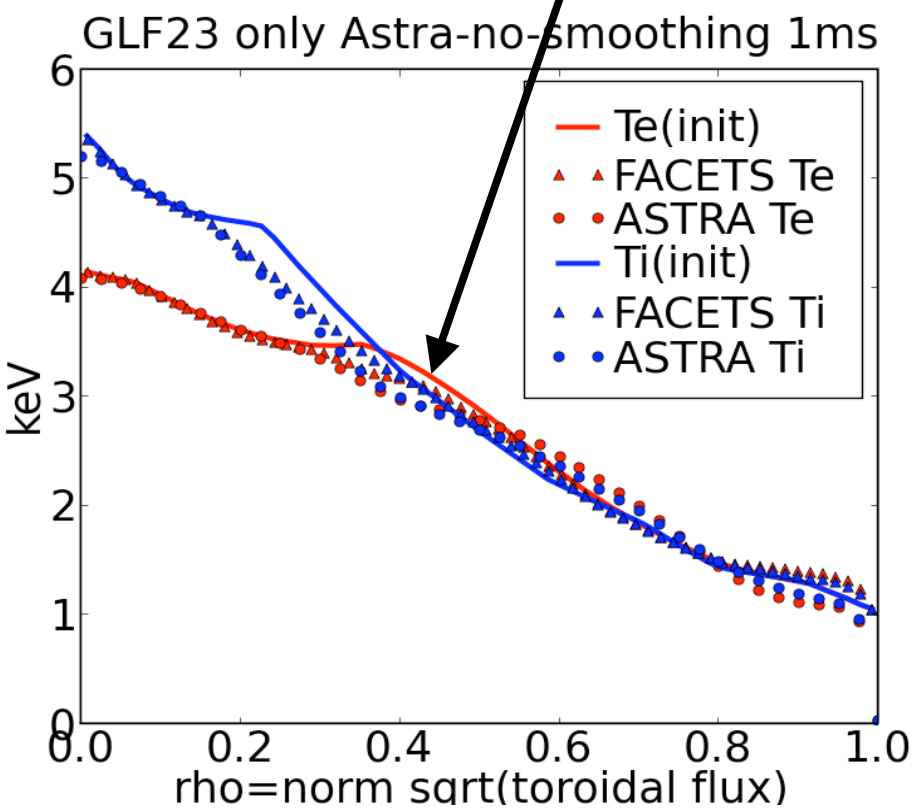
Core: Verification without GLF23 compares well except at axis

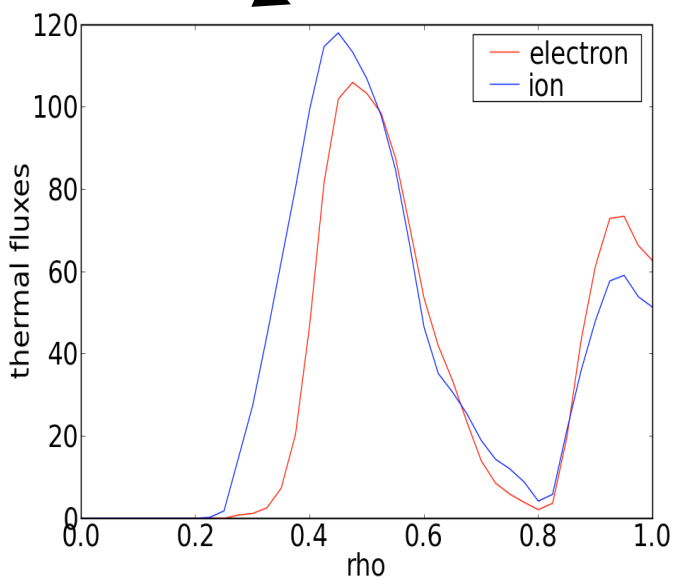
- **Disagreement for non-GLF23 near axis**
- **ASTRA appears to lack energy conservation near the axis**
  - Sum not conserved
  - Shape changes



Core: With GLF23, could be improved

- Had to turn off smoothing in ASTRA (evolution sensitive to smoothing)
- Marginal ETG/ITG stability points coincide (good)
- Max discrepancy @ ρ ~ 0.4 (~7%)
- Why does ASTRA evolve at extremum of flux?





SSGKT SAPP: Embedded turbulence calcs will provide good answers for flattop

- For flattop, need only find steady state
- Acceptable to have 1 simulation hour to find.
- SSGKT (embedded SAPP) developing a steady-state, embedded turbulence transport calculation
- Initial calculations outside of FACETS framework; will be moved into FACETS
- Lessons learned will also feed back into FACETS

Core: FACETS developed copy-free solution to access deep Fortran derived types from C/C++

- **Compatible with BABEL, used for interfacing to Plasma State**
- **Developed 3-step approach to make Fortran90 APIs C/C++ friendly**
  - Starting point: a Fortran routine dostuff with a derived type argument myfoo
  - Step 1: encapsulate derived type myfoo into myfoo_ptr
  - Step 2: serialize myfoo_ptr into an array of integers
  - Step 3: create new, C-friendly dostuffC routine that takes only primitive types

Exposing Fortran derived types to C and other languages, A Pletzer, D McCune, S Muszala, V Vadlamani, and S Kruger, to appear in Computing in Science & Engineering (2008)

```
subroutine dostuff(myfoo, ...)
 type(foo) :: myfoo
 ...
subroutine dostuffC(handle, ...)
 integer :: handle(12)
 type(foo_ptr) :: myfoo_ptr
 myfoo_ptr = transfer(handle, myfoo_ptr)
 call dostuff(myfoo_ptr%ptr, ...)
 ...
```

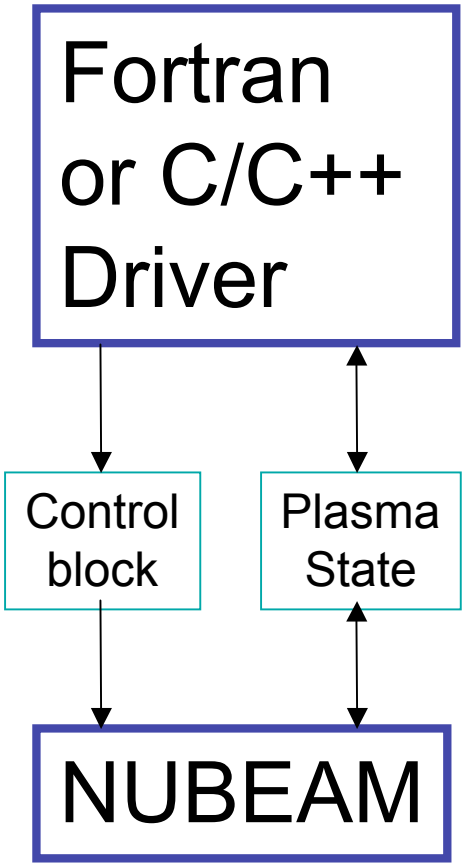
```
Type foo_ptr
   type(foo), pointer :: ptr
end type
type(foo_ptr) :: myfoo_ptr
allocate(myfoo_ptr%ptr)
```

```
integer :: handle(12)
call transfer(myfoo_ptr, handle)
```

Progress towards integrating neutral beam sources into the core: development of Plasma State based NUBEAM driver

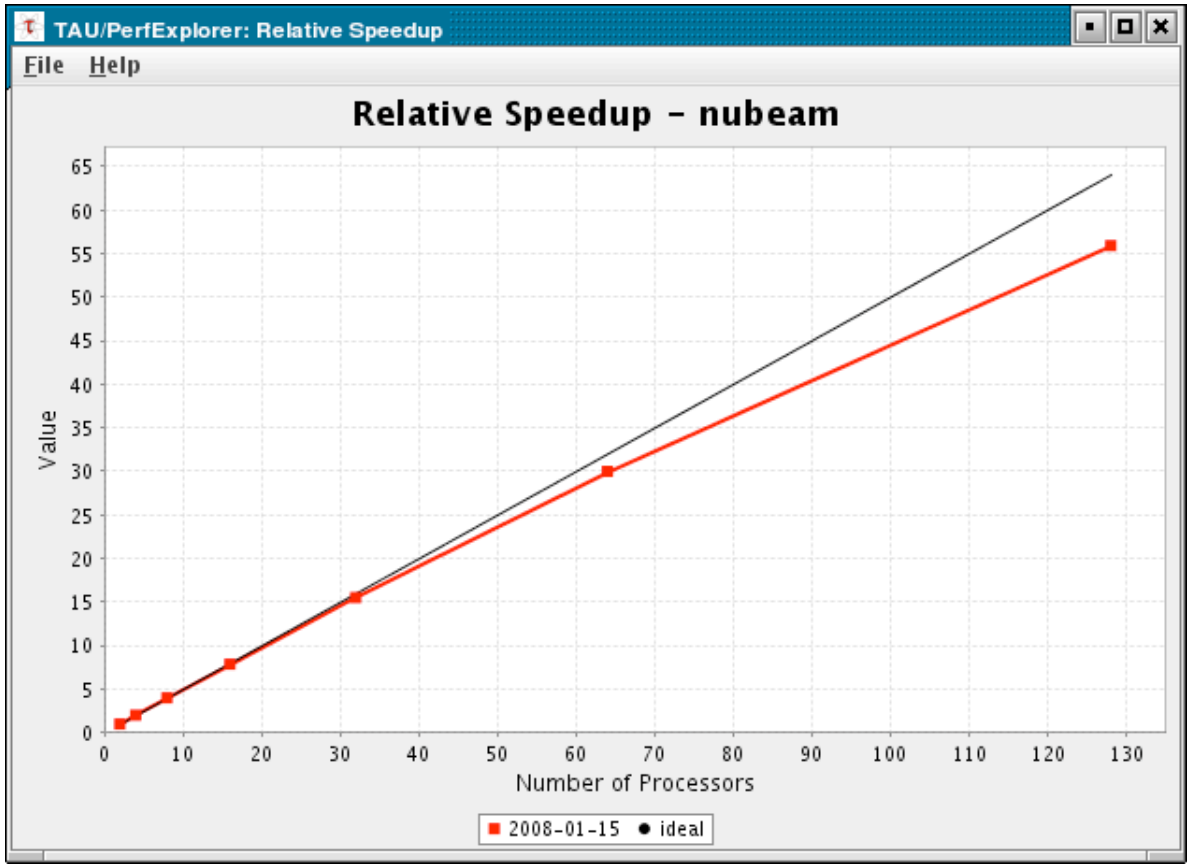
Driver Method Compatible with C/C++ (McCune, Indireshkumar, Pletzer)

- Fortran driver OK in TRANSP/PTRANSP.
- Control Block:
  - One for initialization
  - One for step advance
  - NUBEAM-specific numerical controls
  - Sequenced, static elements, directly instantiatable in C/C++.
- Plasma State physics component interface:
  - Neutral beam geometry
  - Beam powers and voltages
  - Target plasma MHD equilibrium
  - Target plasma temperatures and densities
  - Fast ion density, heating and current drive profiles returned
  - C/C++ python-generated set/get interface using “opaque handle” method for structure with dynamic elements (Tech-X & PPPL collab.).



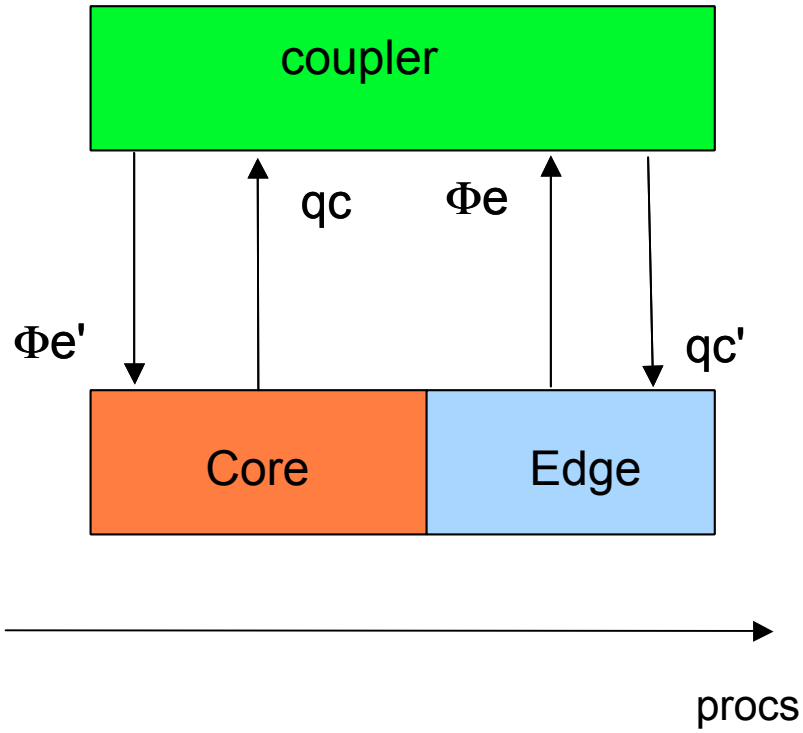
NUBEAM parallelized

- 10^6 particles
- 128 procs max



Initial ad-hoc core-edge coupling achieved

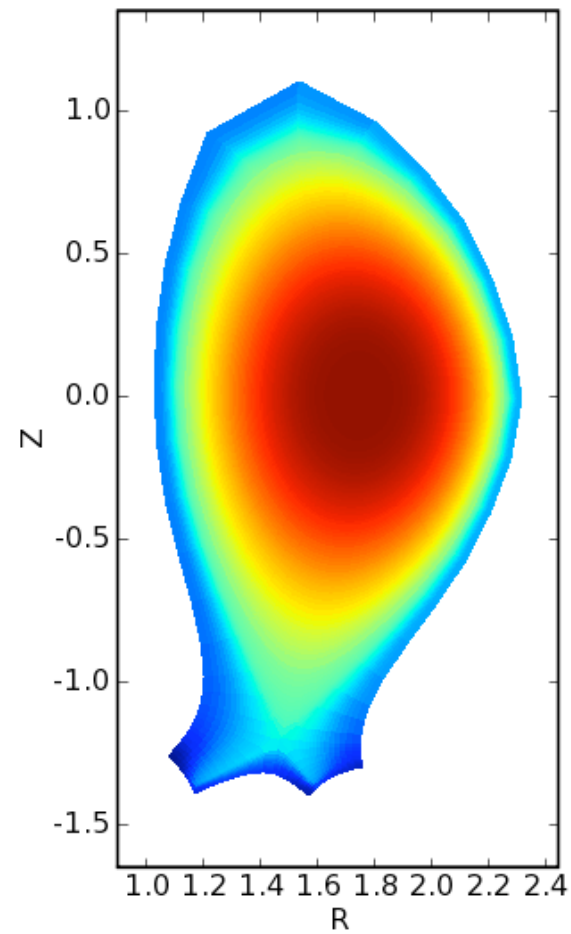
- We have coupled the core component to edge component using explicit coupling scheme
- Only temperature evolution equations coupled as initial test case
- The individual components exchange data via the framework for setting core-edge boundary conditions
- Core sends fluxes to edge, edge sends temperatures to core via framework
- A common set of names for exchanged variables makes coupling more variables simpler



https://facets.txcorp.com/facets/wiki/InterfacesAndNamingScheme

For initial coupling used D-IIID shot 118898.

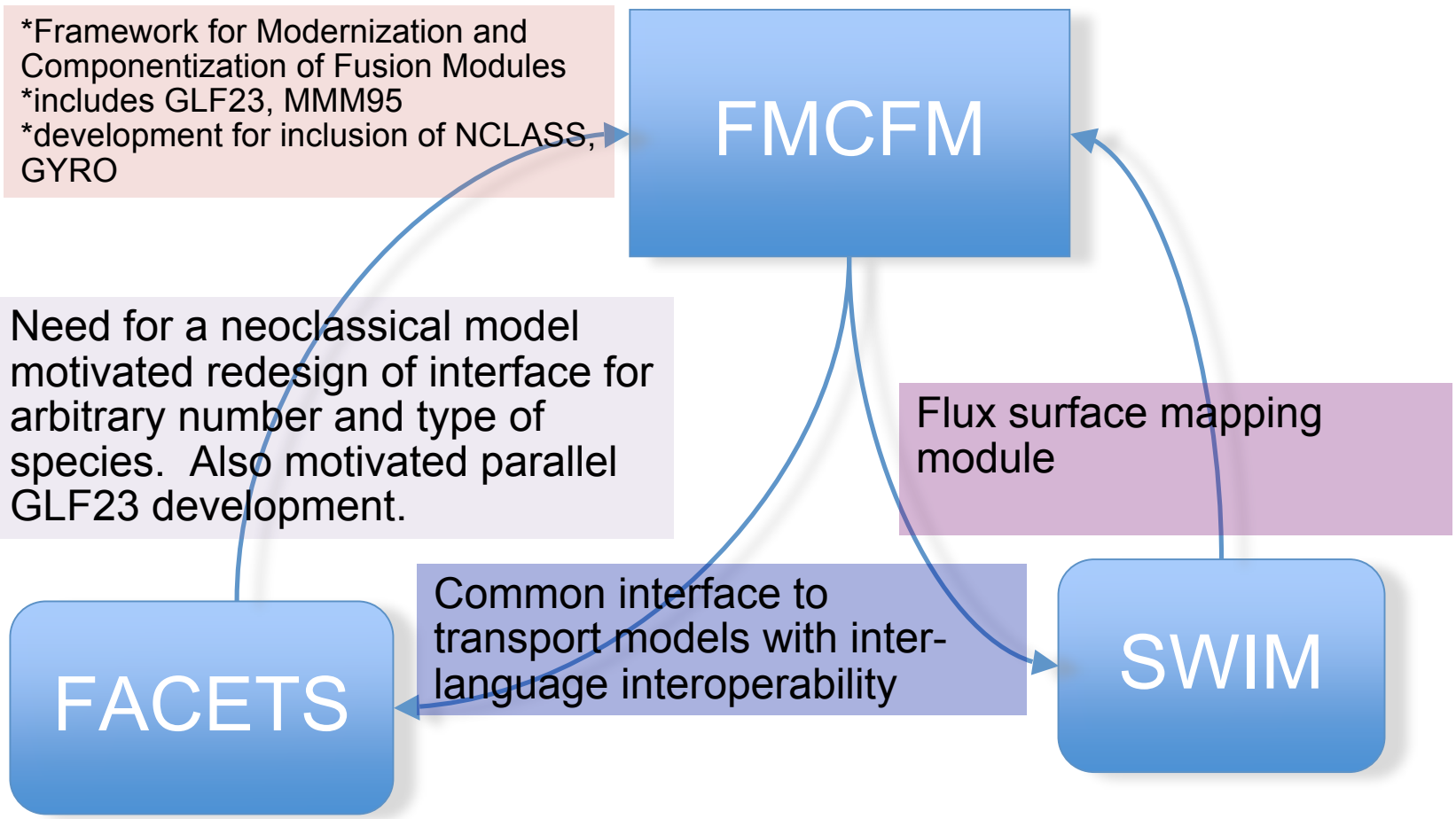
- Used eqdsk files to generate both core and edge geometry.
- Initial profiles were matched to get consistent temperatures and density at core-edge boundary
- Simulation was evolved using explicit scheme. Core runs on one set of processors, edge on one processor.
- Explicit scheme places restriction on coupling time step to 1 microsecond. This will be relaxed when implicit coupling will be used.
- Shows successful tight coupling of components from within the framework.
- Current work focuses on getting realistic coupling simulations.



FACETS supported development of new 1D wall mode (WallPSI)

- WallPSI completed and committed to svn repo
- Autoconf build system
- In the process of being modified to conform to FACETS superstructure initialize/setData/run/getData interface
- Will need to distinguish between into the wall and out of the wall fluxes

FACETS provides a positive feedback loop to other projects: example



Improvements to PETSc, Babel also resulted

FACETS has a working multilanguage system

- **Direct memory calls**
  - Framework: C++
  - Core: C++
  - Core sources: F90
  - Edge: Python-MPPL (pre F90) mix
  - Wall: C
- **Analysis**
  - python/matplotlib plotting
  - Python input file processing

FACETS has moved to address issues raised last time

- Interaction with experimenters
- Team rearranged to have a dedicate modeler. (ORNL replace Kuehn with Cobb)
- Not currently dealing with deep ELMs, but if we have to
  - Stop integration at ELM instability
  - Use MHD/other model to get through ELM
  - Resume with transport code

Response to PSACI Charge 1

- **Progress toward achieving its scientific targets with respect to clear deliverables during this second year of the project:**
  - Developed a framework for composition of an arbitrary number of components connected through tight coupling. This includes interface definitions that facilitate composition as well as infrastructure for component development.
  - Developed a core modeling component that is parallel over 10's of processors now and has an improved core solver that decreases computational time by a factor of 10-50 without parallelism.
  - Componentized and parallelized UEDGE as the standard edge coupling component.
  - Coupled the core and UEDGE in the framework.
  - Began on componentizing the wall model, which was also further developed.
  - Developed visualization capability for coupled simulations.

Response to PSACI Charge 2

- **Significant interactions with respect to collaborations with the Computer Science and Math communities, such as the SciDAC Centers for Enabling Technologies (CET) and Scientific Applications Partnership Projects (SAPP's):**
  - The PETSc team assisted in the proper use of the PETSc solvers, which were needed by the core component.
  - PETSc members worked directly with the LLNL team to install PETSc in UEDGE, thus facilitating switching between solvers for improving performance and restoration of parallelism.
  - The assessment of the limitations of core component parallelization was carried out by a collaborative effort between the core component group at Tech-X and the computer scientists at ParaTools.
  - Framework and software engineering being developed by computer scientists and bridge computational scientists.
  - The Python wrapped UEDGE was brought into the FACETS framework via Babel, the interlanguage operability tool. An LLNL/CCA computer scientist developed the intercomponent interface in SIDL. Computational fusion scientists are coding to that interface to provide the edge implementation to the framework.
  - Error analysis for cell-centered finite volume algorithm has been developed and is being applied to edge modeling.

Response to PSACI Charge 3

- **How leadership-class computing resources have enabled the achievement of the targeted scientific goals as well as demonstrated the scalability of the science with the number of processors on these platforms – including what new science is enabled at the terascale and then at the petascale:**
  - The FACETS framework is able to distribute components out over different processor groups, as is needed for effective use of large numbers of processors.
  - The core solver has demonstrated scaling to 30-50 processors for 128-256 cells.
  - The neutral beam source module (NUBEAM) has shown scaling to 128 processors.
  - An SBIR is developing a per-flux-surface parallel version of GLF23. Phase II enhancement funding would allow this to be continued, including incorporation of TGLF.
  - Parallelization has been restored to edge component (UEDGE).
  - The framework builds and runs on Franklin (Cray XT4) and Bassi. UEDGE runs on Bassi, but not Franklin, as Franklin cannot accommodate shared objects on the compute nodes.
  - Complete writeups of the installation process are available on a public wiki.

Response to PSACI Charge 4

- **Status of verification and validation efforts:**
  - Comparisons of FACETS core transport with Astra are now being performed. Verification is by a systematic process involving the successive inclusion of effects: simple transport models through GLF23, temperature equilibration, removal of numerical artifacts, such as commonly-used smoothing, etc.
  - Engaged Rich Groebner (GA) in discussions to define first core-edge coupled simulations.
  - Augmented personnel to include John Cobb, who is tasked with setting up simulations for users.