

Presented at APS 2011, Salt Lake City, poster No. JP9.00139

A modular, parallel, multi-region, predictive transport equation solver, installed and available in PTRANSP

Xingqiu Yuan
Douglas McCune,
Stephen Jardin
Robert Budny
and Greg Hammett

Princeton Plasma Physics Laboratory,
Princeton, NJ

Thanks to J.E.Kinsey, G.M.Staebler (General Atomics)
and Wayne Solomon (Princeton Plasma Physics Laboratory)

Abstract

We introduce a modular, parallel, multi-regional, implicit transport equation solver built over the Plasma State and other publicly available (NTCC) libraries. The solver has been installed, is being tested, and will be available for use in predictive TRANSP (PTRANSF). The solver itself does not depend on PTRANSF internals and will be made available through the NTCC website. The solver is used to integrate the highly nonlinear time-dependent equations for ion, and electron temperatures and densities, and angular momentum with implicit Newton iteration methods. The user controls choice of transport models attached to the solver, with a wide range of neoclassical and/or turbulent, or semi-empirical or data driven choices available. Available turbulent transport models include: MMM series, GLF23, and TGLF. For the more CPU-intensive transport models such as TGLF, a multi-level, communicator splitting method is used to parallelize the computation of transport coefficients using MPI, allowing the code to run on a flexible number of CPUs. In order to test and benchmark the code, PTRANSF code predicted temperature profiles have been compared to experimental data, achieving good agreements.

Governing equations

Ion energy conservation equation

$$\frac{\partial}{\partial t} \left[\frac{3}{2} V' n_i k T_i \right] + \frac{\partial}{\partial \rho} \left[V' \langle |\nabla \rho|^2 \rangle n_i k \left(\chi_i \frac{\partial (T_i - \alpha \Theta_i)}{\partial \rho} - T_i v_i \right) \right] - \xi \frac{\partial}{\partial \rho} \left[\rho V' \frac{3}{2} n_i k T_i \right] = S_i V'$$

electron energy conservation equation

$$\frac{\partial}{\partial t} \left[\frac{3}{2} V' n_e k T_e \right] + \frac{\partial}{\partial \rho} \left[V' \langle |\nabla \rho|^2 \rangle n_e k \left(\chi_e \frac{\partial (T_e - \alpha \Theta_e)}{\partial \rho} - T_e v_e \right) \right] - \xi \frac{\partial}{\partial \rho} \left[\rho V' \frac{3}{2} n_e k T_e \right] = S_e V'$$

Angular momentum conservation equation

$$\frac{\partial}{\partial t} \left[V' \sum n_i m_i \langle R^2 \rangle \omega \right] + \frac{\partial}{\partial \rho} \left[V' \sum n_i m_i \langle R^2 |\nabla \rho|^2 \rangle \left(\chi_\phi \frac{\partial (\omega - \alpha \Theta_\omega)}{\partial \rho} - \omega \frac{v_\phi}{\rho} \right) \right] - \xi \frac{\partial}{\partial \rho} \left[V' \rho \sum n_i m_i \langle R^2 \rangle \omega \right] = S_\omega V'$$

where $\Theta_i = (\Delta \rho)^2 \frac{\partial}{\partial \rho} (V' T_i) / V'$ $\Theta_e = (\Delta \rho)^2 \frac{\partial}{\partial \rho} (V' T_e) / V'$ $\Theta_\omega = (\Delta \rho)^2 \frac{\partial}{\partial \rho} (V' \omega) / V'$

α is a parameter for hyper-conductivity terms. Note these term vanish in limit $\Delta \rho \rightarrow 0$

Numerical algorithm

- * Finite difference method used to discretize the governing equations
- Newton iteration method developed by S.Jardin & G. Hammett (JCP 2008) used to solve tri-diagonal finite difference equations
- * Hyper-conductivity term to damp short wave oscillations

$$\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}_j \begin{pmatrix} T_{j+1} \\ \Theta_{j+1} \end{pmatrix} + \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}_j \begin{pmatrix} T_j \\ \Theta_j \end{pmatrix} + \begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix}_j \begin{pmatrix} T_{j-1} \\ \Theta_{j-1} \end{pmatrix} + \begin{pmatrix} D_j \\ 0 \end{pmatrix} = 0$$

$$A_{11} = s\Phi_{j+1/2} \left[\chi_{j+1/2} + \left(\frac{\partial \chi}{\partial T'} \right) T'_{j+1/2}{}^{n+(i-1)/N} \right] \quad C_{11} = s\Phi_{j-1/2} \left[\chi_{j-1/2} + \left(\frac{\partial \chi}{\partial T'} \right) T'_{j-1/2}{}^{n+(i-1)/N} \right]$$

$$A_{12} = -s\alpha\Phi_{j+1/2}\chi_{j+1/2} \quad C_{12} = -s\alpha\Phi_{j-1/2}\chi_{j-1/2}$$

$$A_{21} = 1 \quad A_{22} = 0 \quad C_{21} = 1 \quad C_{22} = 0$$

$$B_{11} = -1 - A_{11} - C_{11} \quad D_j = T_j^n + \Delta t S + s\Phi_{j+1/2} T'_{j+1/2}{}^{n+(i-1)/N} \left(T_j^{n+(i-1)/N} - T_{j+1}^{n+(i-1)/N} \right) \left[\frac{\partial x}{\partial T'} \right]$$

$$B_{12} = -A_{12} - C_{12}$$

$$B_{21} = -2 \quad B_{22} = -1 \quad + s\Phi_{j-1/2} T'_{j-1/2}{}^{n+(i-1)/N} \left(T_j^{n+(i-1)/N} - T_{j-1}^{n+(i-1)/N} \right) \left[\frac{\partial x}{\partial T'} \right]$$

where $s = \Delta t / (\Delta\Phi)^2$ S is the source term Φ Metric elements in transport eqns

Plasma flows and $\vec{E} \times \vec{B}$ shear

- * No poloidal momentum equation solved in present formulation.
- * Two methods have been implemented to calculate poloidal velocities
 - * NCLASS model to calculate the poloidal velocities

$$\vec{E} \times \vec{B} \text{ Velocity: } v_{E \times B} = v_\theta - \frac{B_\theta}{B_\phi} v_\phi - \frac{1}{ZenB_\phi} \frac{\partial p}{\partial r}$$

- * R.E.Waltz's neoclassical approximation for simple circular geometry (R. E. Waltz, G.M.Staebler et al., PoP, 1997)

$$v_{E \times B} = \frac{T_i}{T_e} \left(\frac{\rho_i}{a} \right) c_s \left(\frac{a}{L_{ni}} + \alpha_{neo} \frac{a}{L_{Ti}} \right) - \sigma \left(\frac{r}{Rq} \right) v_\phi$$

$$\text{flow shear rate } \gamma_{E \times B} = \frac{r}{q} \frac{d}{dr} (qv_{E \times B} / r)$$

Parallel PT_SOLVER

- Some turbulent models (such as TGLF) use most of the computing time.
- The newton iteration solver is efficient.
- Multi-region capability allows user to choose different model for core, middle, and edge plasma region.
- Communicator split method for multi-task parallelization. It assigns different task into different group of CPUs (daughter communicator)
- Integrated, multi-scale simulation code requires multi-task parallelization.
- Two-level parallelization implemented in PT-SOLVER using communicator split method.
 - a): parallelization of flux-surface numbers
 - b): parallelization of TGLF in spectrum domain.

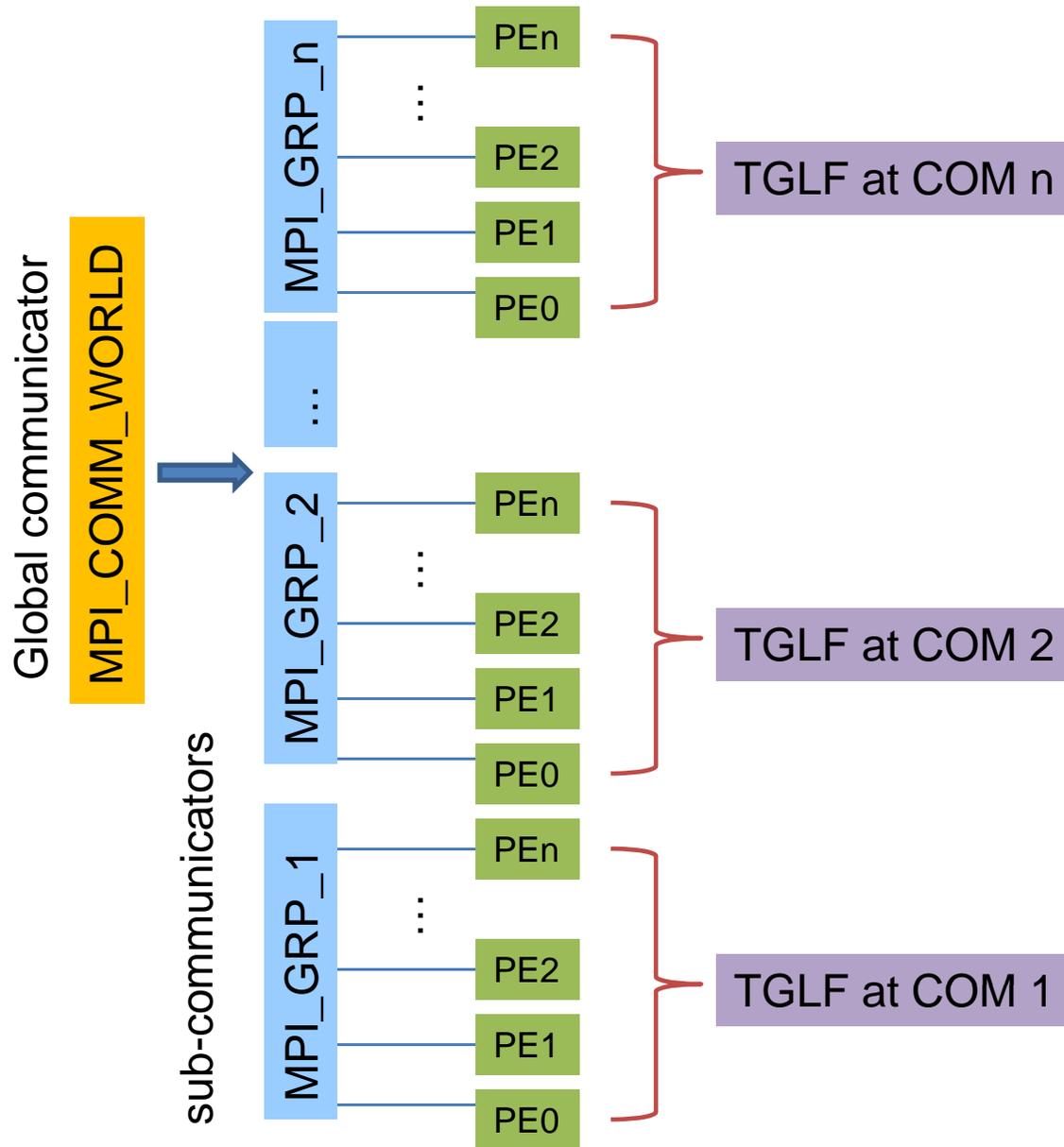
TGLF model

- The model solves for the linear eigenmodes of trapped ion and electron modes (TIM, TEM), ion and electron temperature gradient (ITG, ETG) modes and electromagnetic kinetic ballooning (KB) modes.
- The TGLF model generalizes the methods of GLF23 to a more accurate system of moment equations and an eigenmode solution method that is valid for shaped geometry and finite aspect ratio.
- The Miller equilibrium model is used in TGLF for shaped finite aspect ratio geometry.

Plasma state module

- The plasma state
 - 1): contains data for axisymmetric MHD equilibrium, plasma and source profiles (1D and 2D), and associated scalar data.
 - 2): has interface to access the data (rezone, and interpolation function).
 - 3): data in plasma state is component based fortran 90 type.
 - 4): provides a mechanism for communication between different codes.
 - 5): PT_SOLVER is based on plasma state software.
 - 6): see http://cswim.org/componentdescrip/plasma_State_V2.003.doc for more details.

Communicator split method (parallel PT_SOLVER)

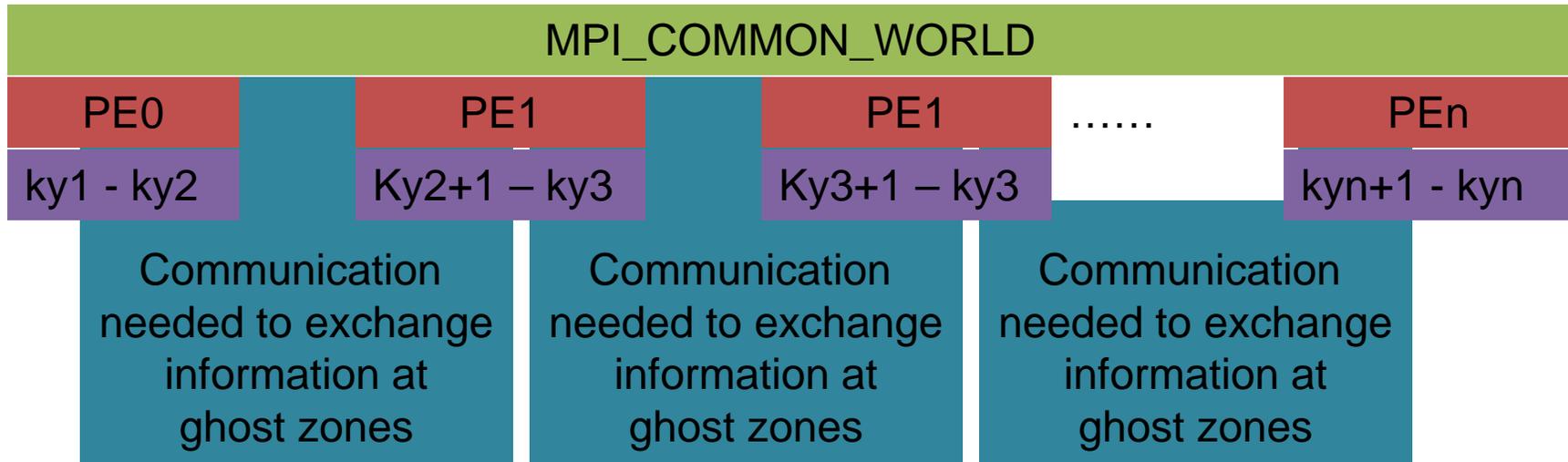


1: split MPI_COMM_WORLD into several sub-communicators, the number of sub-communicators is dependent on the number of nodes on which the TGLF model will run.

2: assign the TGLF model on the sub-communicators to calculate the transport coefficient

3: collect the transport coefficients for the Newton iteration solver.

Spectrum domain decomposition(parallel TGLF)

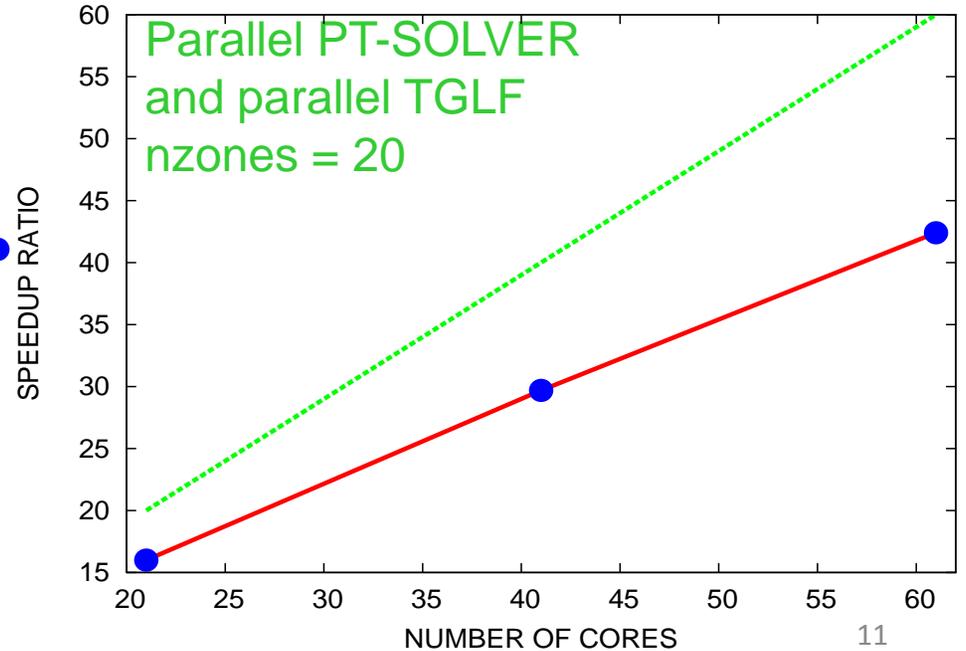
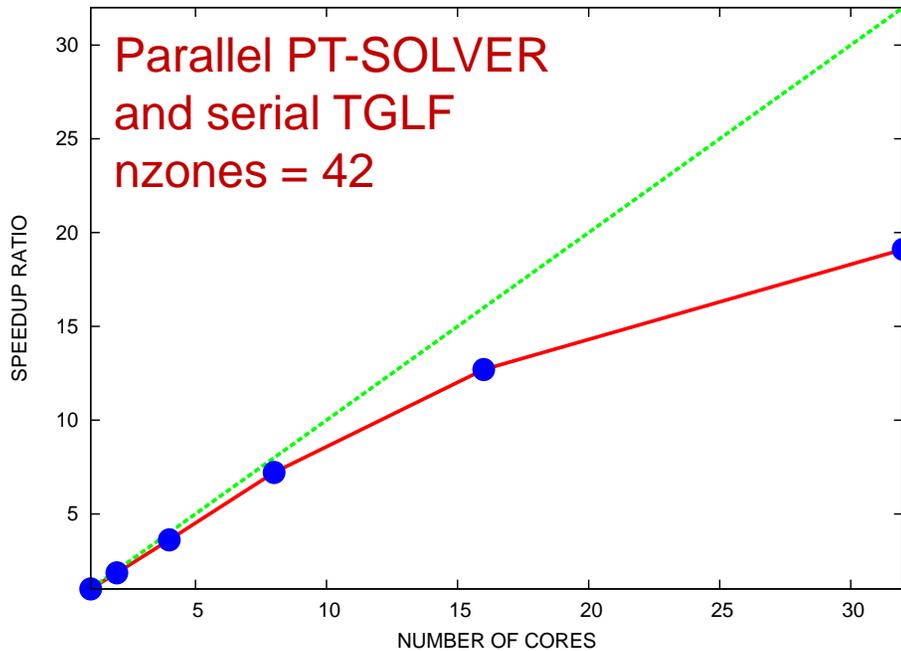
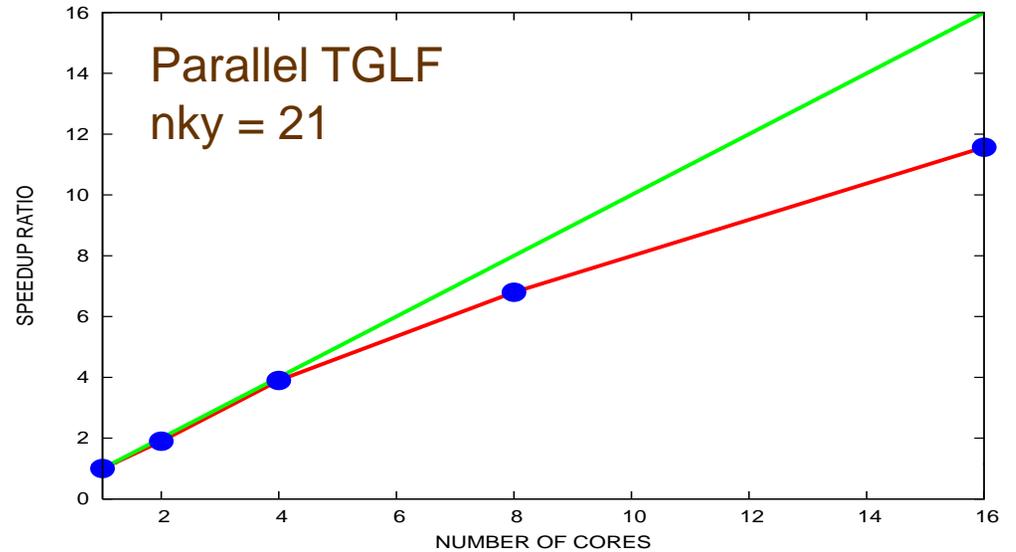


The number of CPU is limited by NKY in TGLF code.

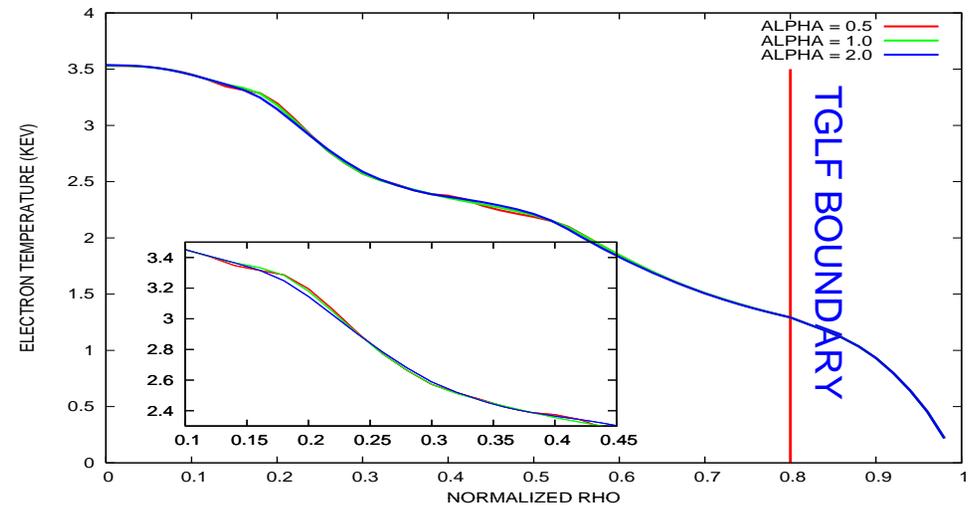
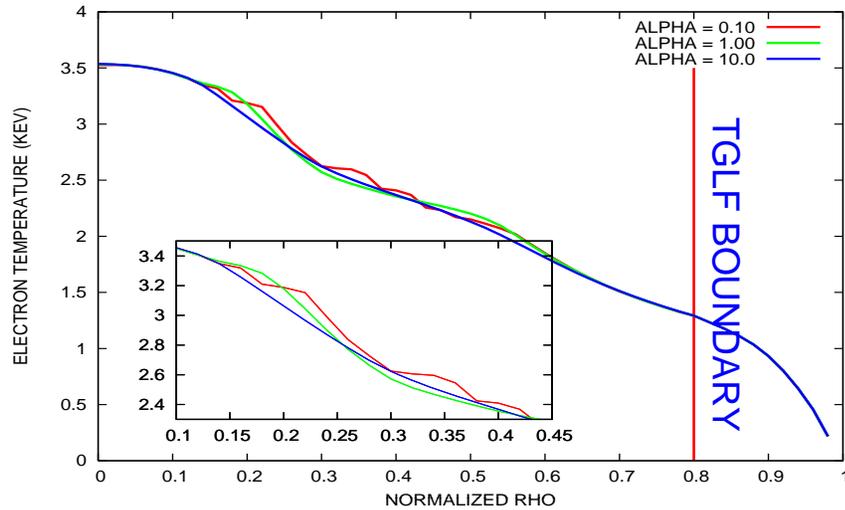
The default value of NKY is 12

parallel run benchmark

Two-level's parallelization allows
1): parallel PT-SOLVER and serial TGLF when $n_{\text{zones}} > n_{\text{cpus}}$
2): parallel PT-SOLVER and parallel TGLF when $n_{\text{zones}} < n_{\text{cpus}}$

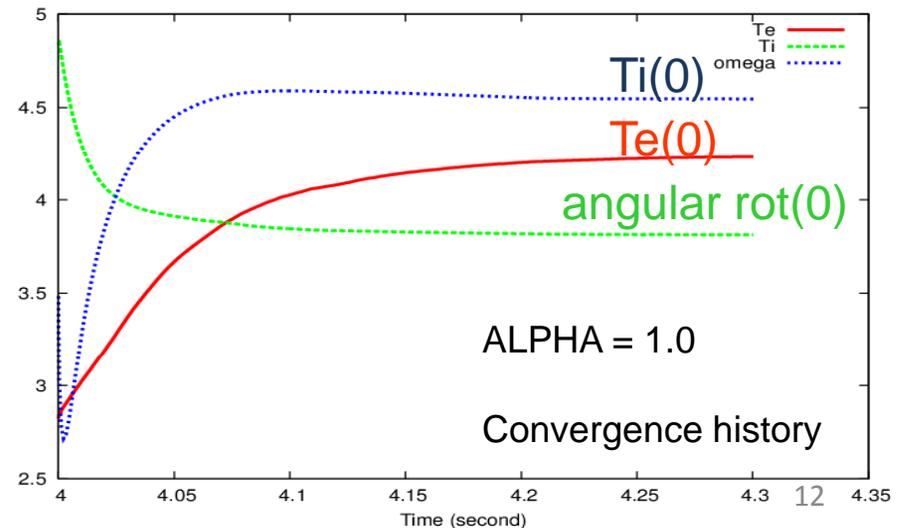
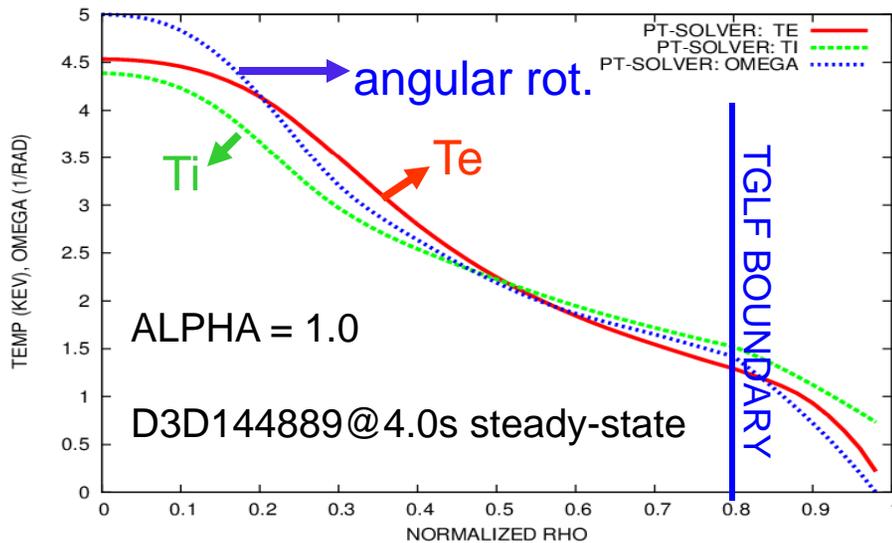


Effects of hyper-viscosity coefficients (ALPHA)



Results are smooth and independent of ALPHA for ALPHA ~ 1, dependence on ALPHA vanishes for $\Delta\rho \rightarrow 0$

Test with Te, Ti and preliminary angular momentum prediction



Code test(1)

1): PT-SOLVER run in TRANSP

In TRANSP analysis mode, the profile data (T_e , T_i , angular momentum and density profiles) are used as input

PT-SOLVER predicts T_e , T_i and angular momentum profiles

EETR_OBS is calculated in TRANSP as:

$$\begin{aligned} EETR_OBS &= \int_0^{\rho} \left[S_e V' - \frac{\partial}{\partial t} \left(\frac{3}{2} V' n_{e,obs} k T_{e,obs} \right) + \xi \frac{\partial}{\partial \rho} \left(\rho V' \frac{3}{2} n_{e,obs} k T_{e,obs} \right) \right] d\rho \\ &= V' \langle |\nabla \rho|^2 \rangle n_{e,obs} k \left(\chi_{e,obs} \frac{\partial T_{e,obs}}{\partial \rho} - \frac{3}{2} T_{e,obs} v_{e,obs} \right) \end{aligned}$$

EETR_MOD is calculated in PT-SOLVER as:

$$EETR_MOD = V' \langle |\nabla \rho|^2 \rangle n_{e,mod} k \left(\chi_e \frac{\partial T_{e,mod}}{\partial \rho} - \frac{3}{2} T_{e,mod} v_{e,mod} \right)$$

Where χ_e and T_e profiles are calculated from ETG turbulence model so that $EETR_MOD = EETR_OBS$

As a consistency check, we compare EETR_OBS (TRANSP) with EETR_MOD (PT-SOLVER)

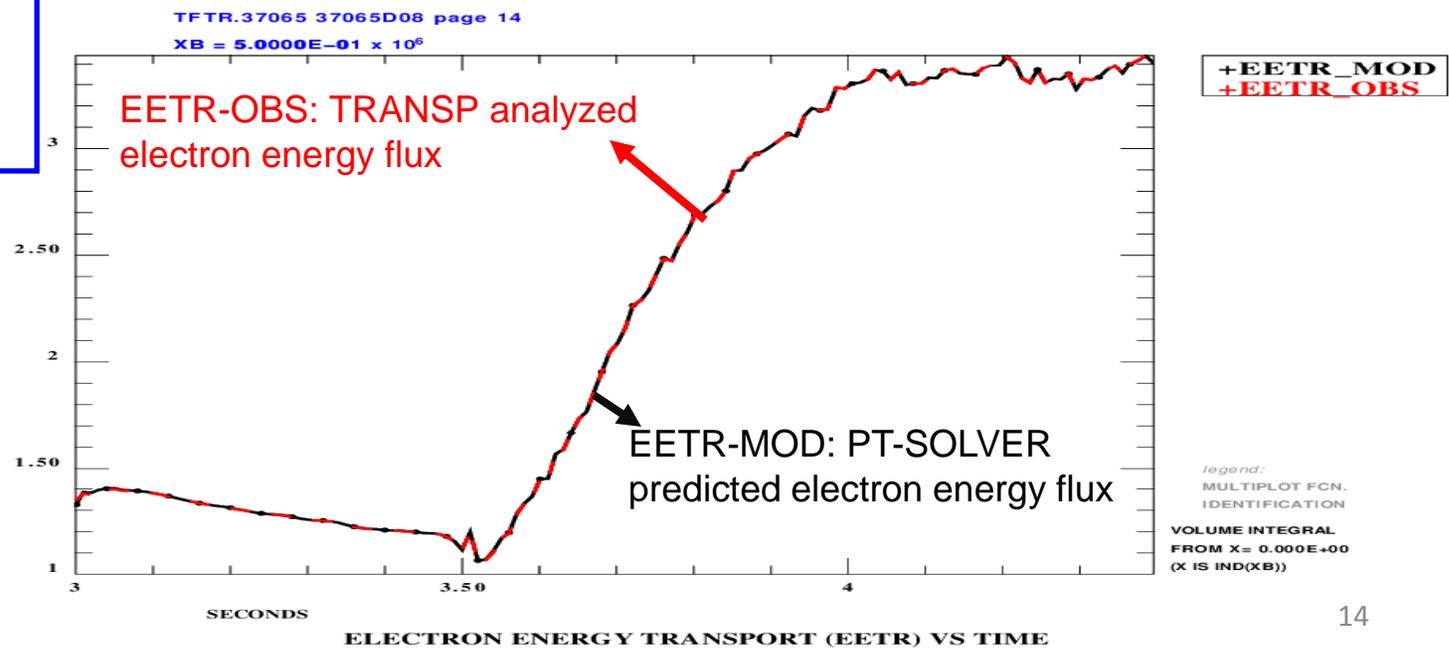
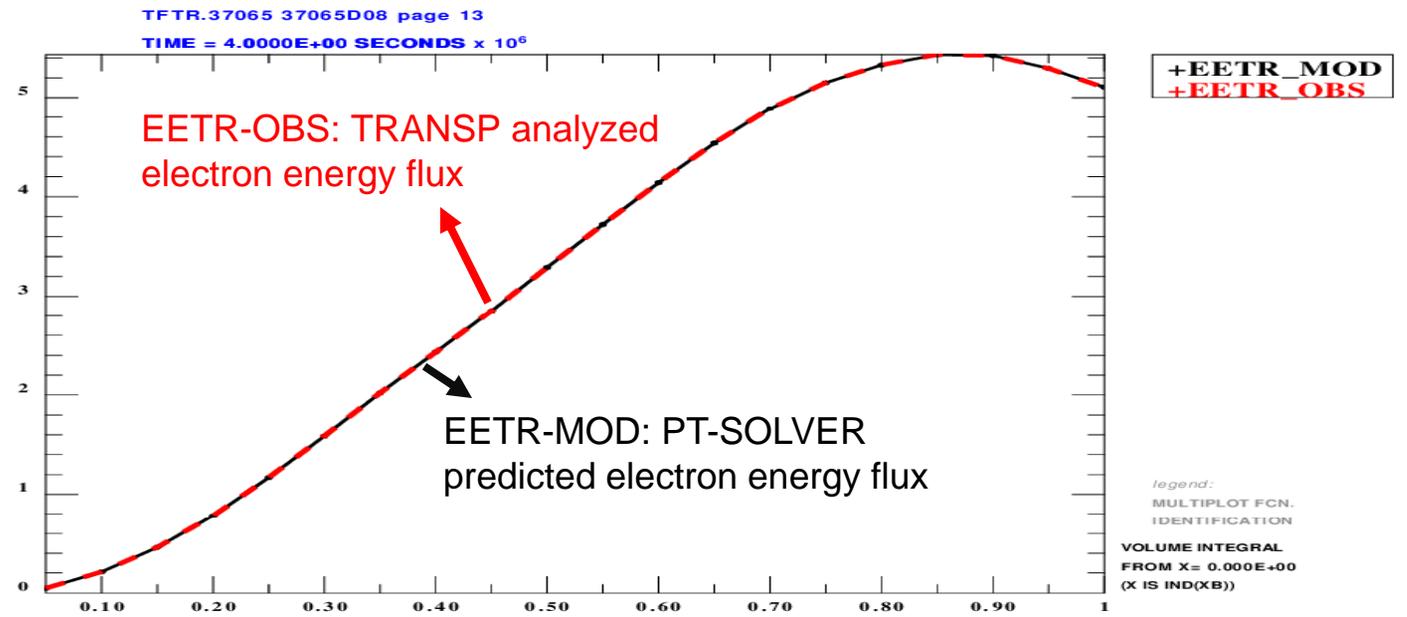
2): PT-SOLVER compare with XPTOR code

PT-SOLVER and XPTOR use fixed density and angular momentum profiles, predict T_e , and T_i profiles with TGLF turbulent model

Code test(2)

TRANSP run in analysis model using profile data

PT-SOLVER predicts T_e , T_i , and omega profiles using TRANSP sources.

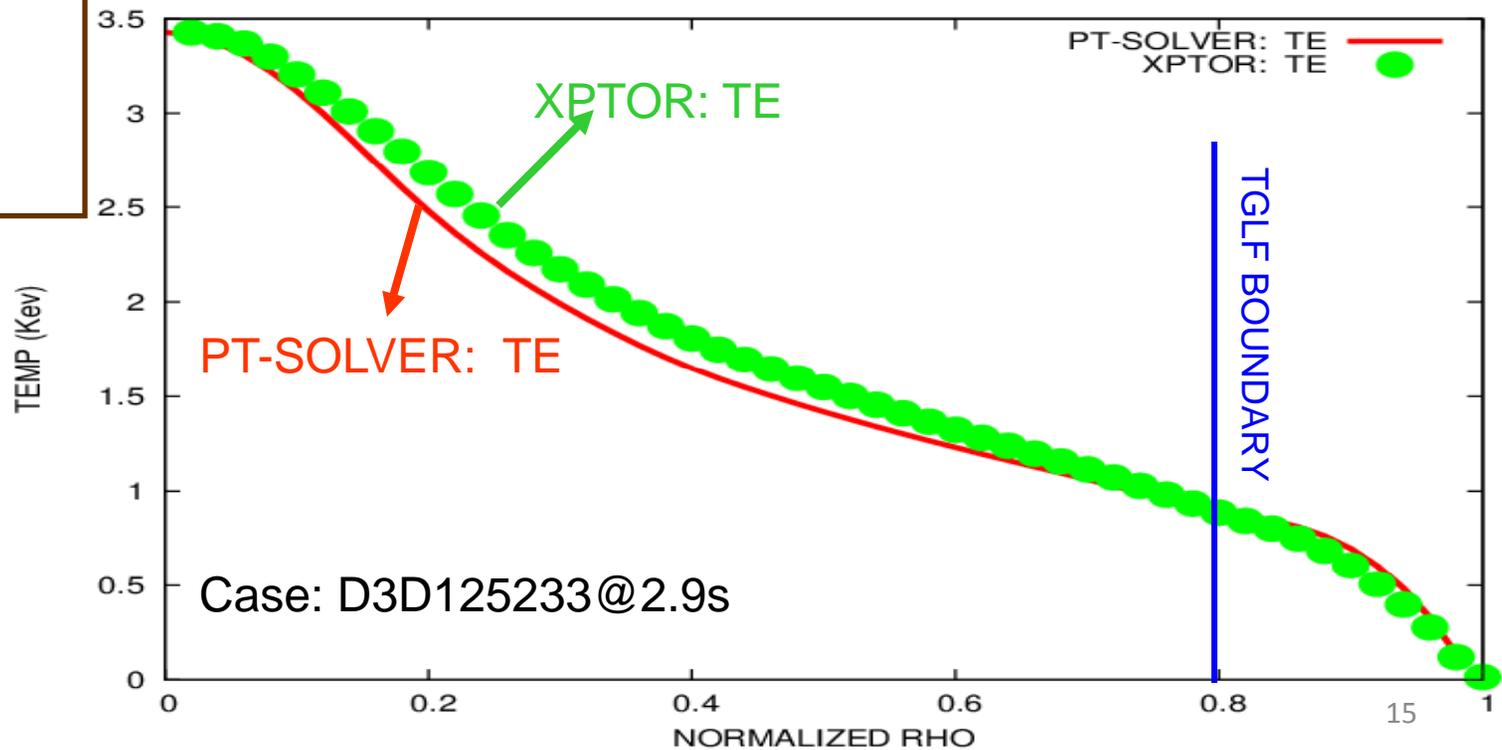
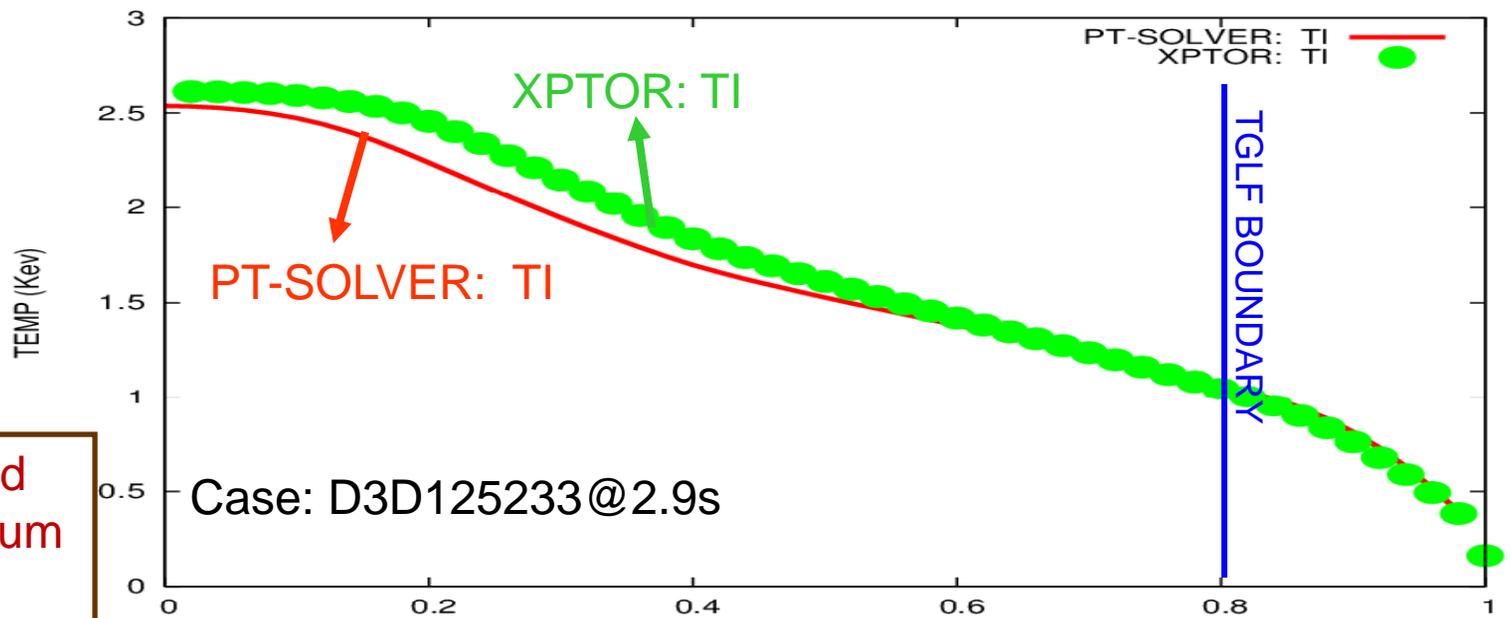


ELECTRON ENERGY TRANSPORT (EETR) VS TIME

Code test(3)

Fixed density and angular momentum Profiles

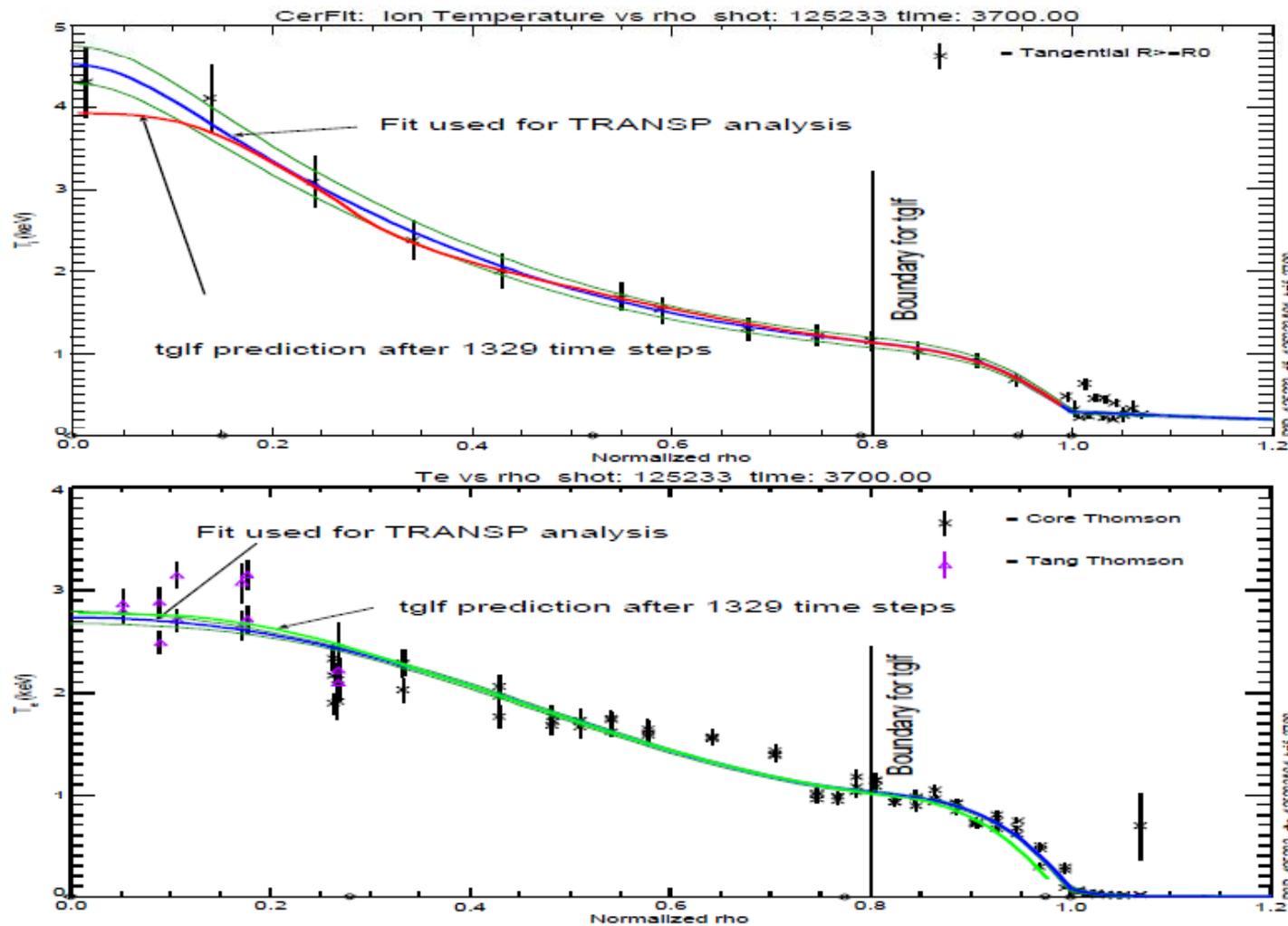
PT-SOLVER and XPTOR Predict Te and Ti



Comparison with experimental data

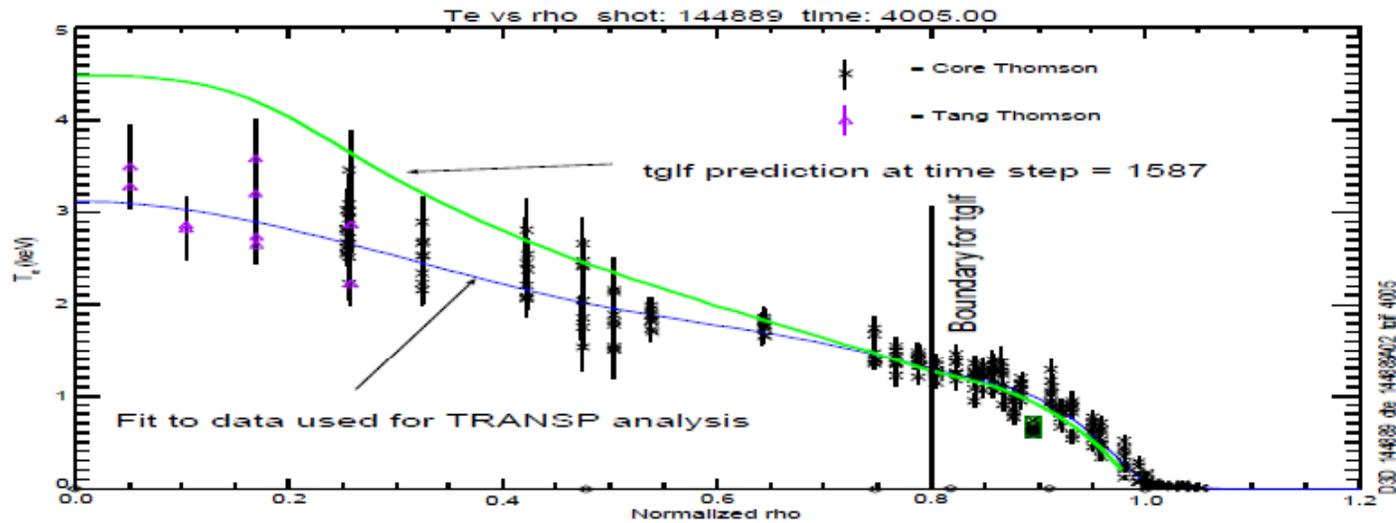
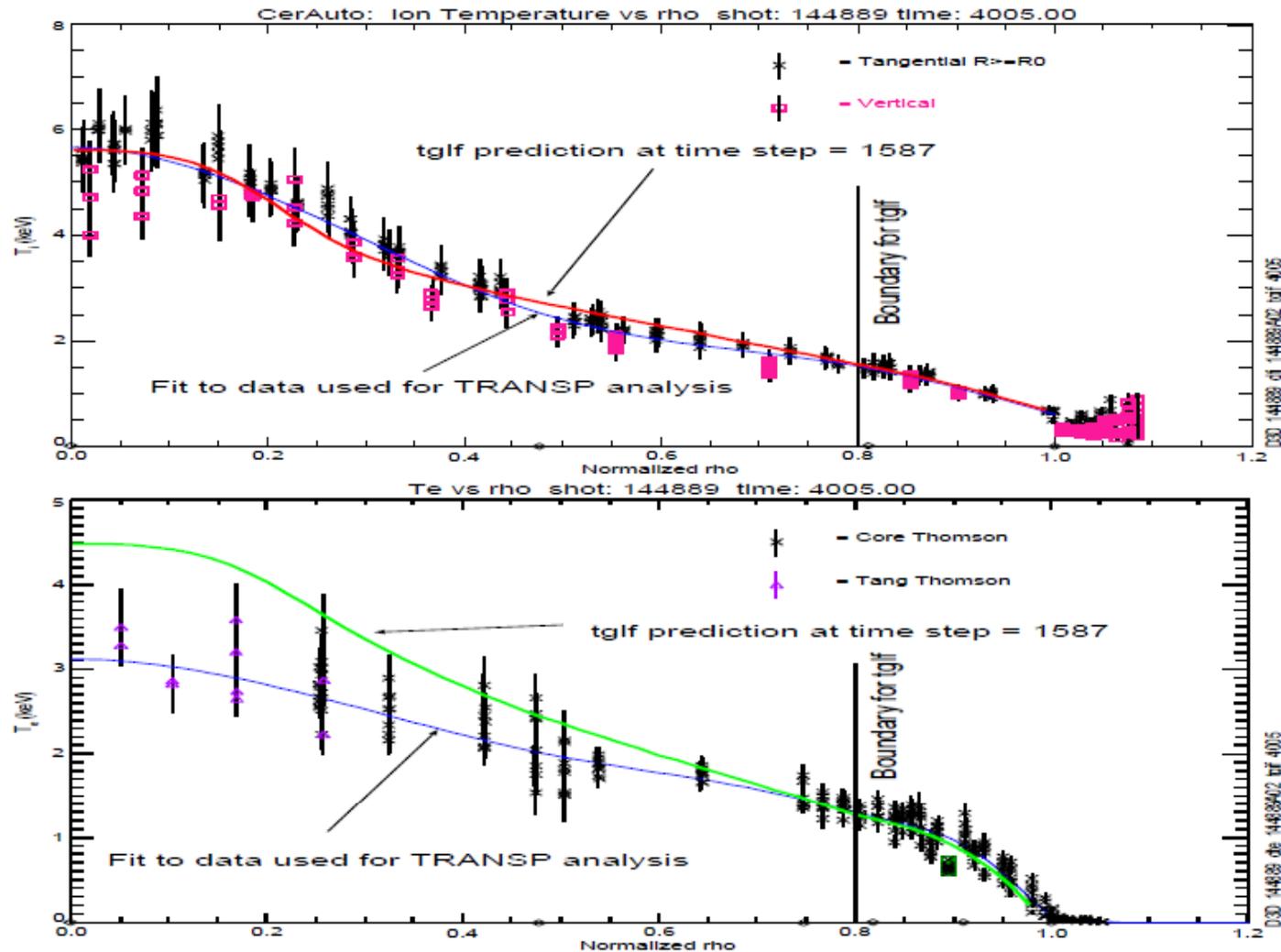
DIII-D shot from torque scan

- Compare predicted T_i and T_e with data and fits used for the TRANSP run



Another DIII-D shot from torque scan

- Compare predicted T_i and T_e with data and fits used for the TRANSP run at time with near balanced NBI



Summary and future work

- A modular, parallel, multi-regional, implicit transport equation solver built over the Plasma State and other publicly available (NTCC) libraries has been developed.
- Wide range of neoclassical, turbulent, or data driven choices models, including MMM series, GLF23, and TGLF.
- Two level parallelization has been implemented in PT_SOLVER with MPI library
- Smooth convergent solution obtained using combination of Newton iteration and hyper-conductivity.
- Initial test of PT_SOLVER in TRANSP gives consistent results with TRANSP analysis
- Good agreement with XPTOR code
- PT-SOLVER with TGLF model used to compare with experimental data
- More benchmark cases are necessary to test the code's robustness and correctness.
- Density prediction capabilities in the future

More info? Please leave your email...