New predictive capabilities in PTRANSP with PTSOLVER

Xingqiu Yuan, Stephen Jardin, Robert Budny, and Greg Hammett
Princeton Plasma Physics Laboratory

Gary Staebler
General Atomics

Presented at APS 2012, Providence, Rhode Island, Poster No. BP8.00131

Related poster:
UO7.00012 : Testing TGLF in ITER-demonstration plasmas and predictions for ITER
2:00 PM–5:00 PM, Thursday, November 1, 2012, Room: 556AB

Abstract

- A new implicit transport equation solver (PT-SOLVER) implemented in predictive TRANSP (PTRANSP) is used to integrate the highly nonlinear time-dependent transport equations using implicit Newton iteration methods. Capabilities are extended to include density and angular momentum prediction. The new solver allows users to choose between different transport models via a standard namelist input. A wide range of neoclassical and/or turbulent models or semi-empirical, including TGLF choices are available. Extensive benchmark test runs have been performed with PT-SOLVER using the TGLF parallelized over flux-surfaces and wavenumbers. A combined number of CPUs up to 128 have been used. The new solver is robust, efficient, and allows large time-steps to be used. PTRANSP predicted temperatures have also been compared with experimental data for various plasma regimes, and good agreement has been achieved.

PTRANSP solver (PT-SOLVER)

1): PT-SOLVER is modular, parallel, multi-regional solver for PTRANSP.

2): It integrate the highly nonlinear time-dependent equations for electron and ion temperatures, densities, and angular momentum.

3): multilevel parallelization (over flux-surface and wave number in TGLF). No longer limited by the number of zones in flux-surface. Flexible number of CPUs can used depending on the problem.

4): Inputs to PT-SOLVER are provided by “plasma state” file.

5): Upto five kinetic species can be used for TGLF.

6): available neoclassical and turbulent models are:
TGLF, GLF23, NEO, NCLASS, Chang-Hinton

Current status of PT-SOLVER development

1): Stand-alone version of PT-SOLVER with TGLF for Te & Ti prediction with two-level parallelization (over flux-surface and wave number in TGLF)

2): Implementation of PT-SOLVER into PTRANSP for Te & Ti prediction (maximum 3 regions, axial, confinement, and edge region)

3): Initial capability of ne prediction

ne predicted, ni, and nimp derived

Governing equations (1)

- Electron density

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

- Total thermal plasma density

$$\frac{\partial}{\partial t}\left[V'\sum_i n_i\right] + \frac{\partial}{\partial \rho}\left[V'\left\langle\left|\nabla\rho\right|^2\right\rangle\left(\sum_i n_i v_i - \sum_i D_i \nabla n_i\right)\right] - \xi\frac{\partial}{\partial \rho}\left[\rho V'\sum_i n_i\right] = \sum_i s_i V'$$

- Total impurity plasma density

$$\frac{\partial}{\partial t}\left[V'\sum_{j,q} n_j^q\right] + \frac{\partial}{\partial \rho}\left[V'\left\langle\left|\nabla\rho\right|^2\right\rangle\left(\sum_{j,q} n_j^q v_j^q - \sum_{j,q} D_j^q \nabla n_j^q\right)\right] - \xi\frac{\partial}{\partial \rho}\left[\rho V'\sum_{j,q} n_j^q\right] = \sum_{j,q} s_j^q V'$$

where S_e, s_i, s_j^q is source terms, which includes neutral gas source, edge source, Nubeam contributed source, and fusion reaction source terms.

D_e, D_i, D_j^q is particle diffusivity v_e, v_i, v_j^q is pinch velocity

$\rho = \sqrt{\Phi / \Phi_{\lim}}\,;\ \xi = \frac{1}{2\Phi_{\lim}}\frac{d\Phi_{\lim}}{dt}$ and $\Phi_{\lim}$ is toroidal flux enclosed by plasma

Governing equations (2)

Assumptions:

(a): constant fraction (for thermal plasma and impurities)

for thermal plasmas: $\Re_i = \dfrac{n_i}{n_1 + n_2 + \ldots + n_i + \ldots}$

for impurity plasmas: $\Re_j^q = \dfrac{n_j^q}{n_1^1 + n_1^2 + \ldots + n_2^1 + n_2^2 + \ldots + n_j^q + \ldots}$

(b): local thermodynamic equilibrium (LTE) (for impurity plasmas) $\dfrac{n_j^{q+1}}{n_j^q} = \dfrac{I_j^{q \to q+1}}{R_j^{q+1 \to q}}$

Assumptions are used to derive individual species density when total number of thermal plasma density and/or impurity density are predicted.

Constraints:

(a): charge neutrality: $n_e = \sum_i n_i + \sum_{j,q} q n_j^q$

(b): effective Z: $n_e Zeff = \sum_i n_i + \sum_{j,q} q^2 n_j^q$

Constrains are explicitly applied in density prediction which reduced the total number of unknows. For example, if electron density is predicted, total thermal density, and Impurity density can be derived when Zeff is know.

Governing equations (3)

Ion energy conservation equation

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

electron energy conservation equation

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

Angular momentum conservation equation

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

$$-\dot{\xi}\frac{\partial}{\partial\rho}\left[V'\rho\sum n_i m_i\left\langle R^2\right\rangle\omega\right]=S_\omega V'$$

where S_{te}, S_{ti}, S_ω is source terms, which includes radiation loss source, neutral gas source, edge source, Nubeam contributed source, and fusion reaction source terms, ICRF, ECRF, and LHW contributed source terms.

$\chi_e, \chi_i, \chi_\varphi$ is thermal conductivity $v_{te}, v_{ti}, v_\varphi$ is pinch velocity

G.V.Pereverzev & G.Corrigan's algorithm

Governing equations can be written in general form as: $\frac{\partial T}{\partial t} + \nabla \cdot [vT - \Gamma \nabla T] = S$

An artificial transport coefficient is added:

$$\frac{\partial T}{\partial t} + \nabla \cdot \left[\left(v + \Gamma_{art} \frac{\nabla T}{T} \right) T - (\Gamma + \Gamma_{art}) \nabla T \right] = S$$

No change in the equation,
however, the peclet number is different: $P_{j+1/2} = \frac{\left(v_{j+1/2} + \Gamma_{art,j+1/2} \frac{\nabla T_{j+1/2}}{T_{j+1/2}} \right) (\delta x)_{j+1/2}}{\Gamma_{j+1/2} + \Gamma_{art,j+1/2}}$

Using **Power-Law** scheme to discretize the convection-diffusion terms.

$$\frac{T^{n+1} - T^n}{\Delta t} + \nabla \cdot \left[\left(v + \Gamma_{art} \frac{\nabla T^n}{T^n} \right) T^{n+1} - (\Gamma + \Gamma_{art}) \nabla T^{n+1} \right] = S$$

For explicit numerical scheme, no change
For implicit numerical scheme, difference arose due to non-cancellation of these two terms.

since $\Gamma_{art} \frac{\nabla T^n}{T^n} T^{n+1} \neq \Gamma_{art} \nabla T^{n+1}$

Apply G.V.Pereverzev & G.Corrigan's algorithm to each of the governing equations

S.Jardin & G. Hammett's 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

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

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

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

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

$$B_{11} = -1 - A_{11} - C_{11}$$

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

$$B_{21} = -2 \quad B_{22} = -1$$

$$\Theta = (\Delta\rho)^2 \frac{\partial}{\partial \rho}(V'T')/V'$$

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

$$C_{12} = -s\alpha\Phi_{j-1/2}x_{j-1/2}$$

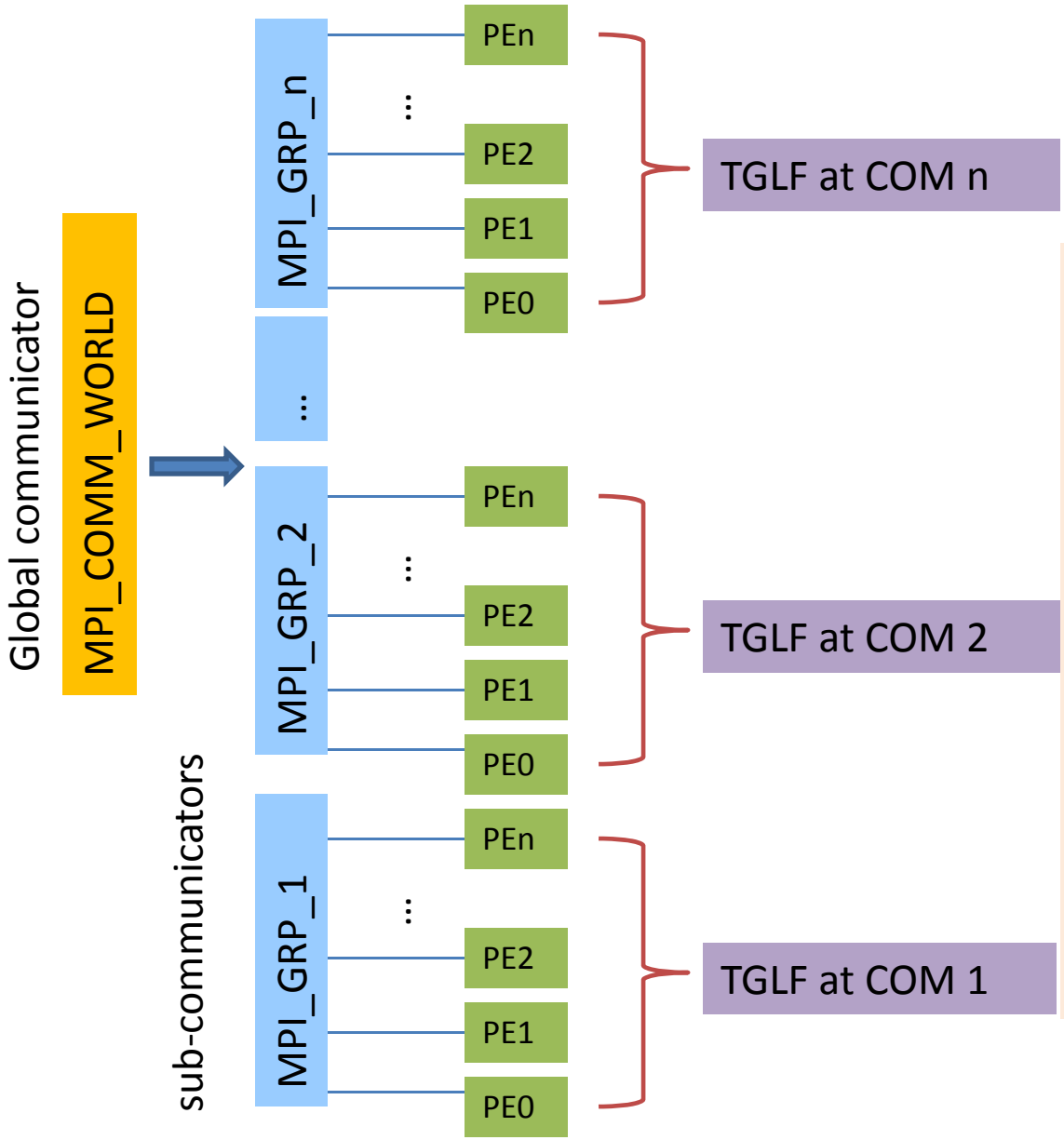
$$C_{21} = 1 \quad C_{22} = 0$$

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

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

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

Two-level parallelization
(over flux-surface, and wavenumber in TGLF)



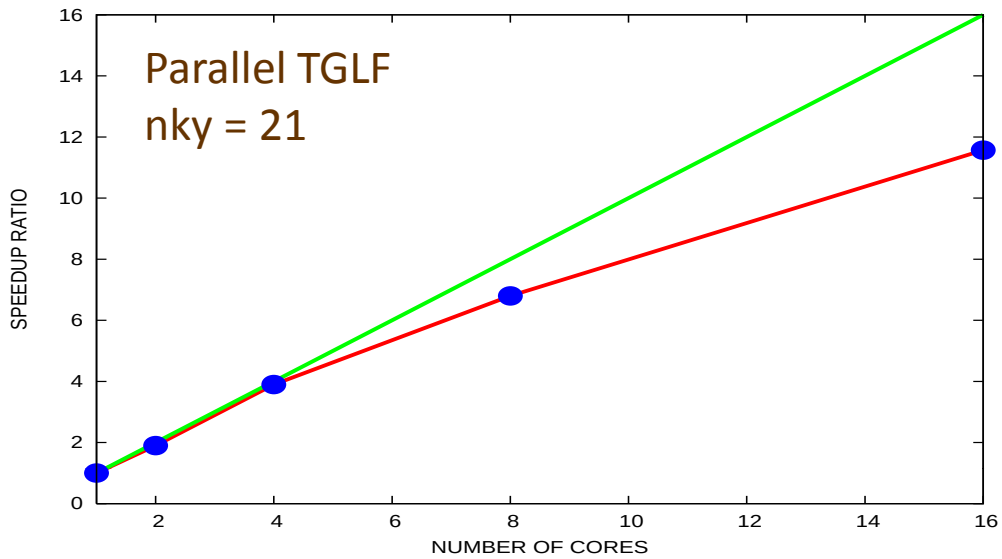
1: split MPI_COM_WORLD into several sub communicators, the number of sub communicators is dependent on the umber of node on which TGLF model will run.

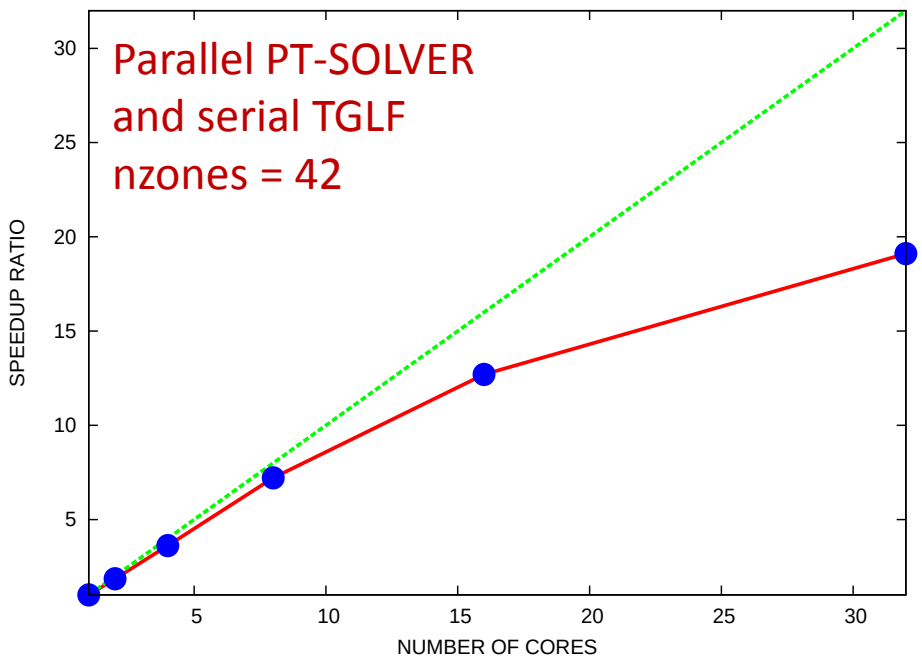
2: assign the TGLF model on the sub-communicators to calculate the transport coefficient

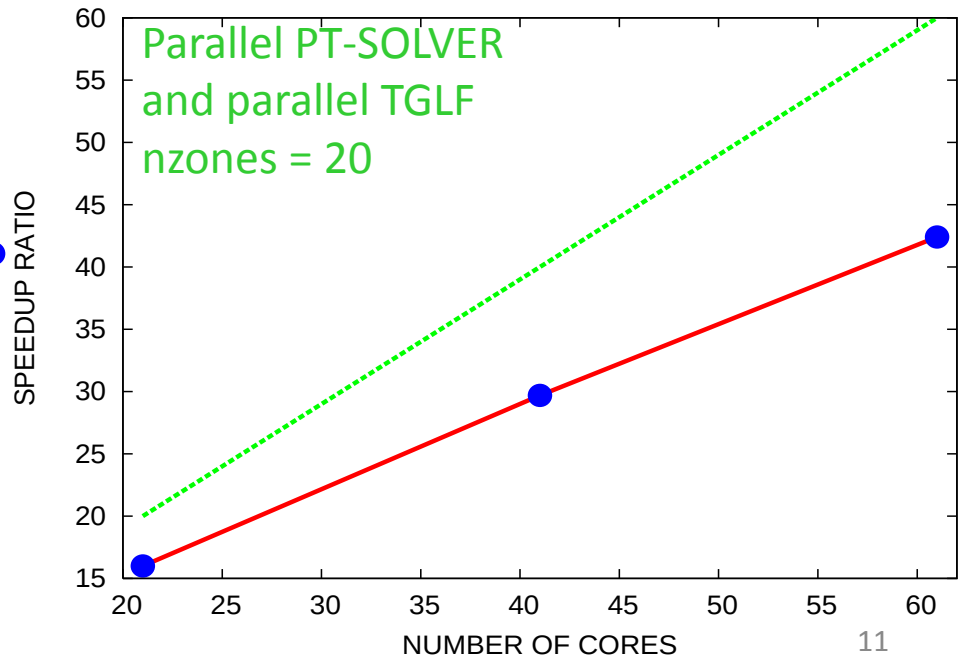
3: collect the transport coefficients for the Newton iteration solver.

parallel run benchmark

Two-level's parallelization allows
1): parallel PT-SOLVER and serial TGLF when nzones > ncpus
2): parallel PT-SOLVER and parallel TGLF when nzones < ncpus







Implementation in TRANSP

lpredictive_mode = 3

! choose predictive model
lpredict_te = 1 (Te predicted)
lpredict_ti = 1 (Ti predicted)
lpredict_pphi = 0 (omega prescribed)
lpredict_ne = 1 (ne predicted)
lpredict_nmain = 0 (thermal ion prescribed)
lpredict_nimp = 0 (impurities prescribed)

NBI_PSERVE = 1 (parallel nubeam)
NPTR_PSERVE = 1 (parallel pt-solver)
NTORIC_PSERVE = 0 (serial toric)

! Only 1 region applied
XIMIN_CONF = 0.0
XIMAX_CONF = 0.8
XIBOUND = 0.8

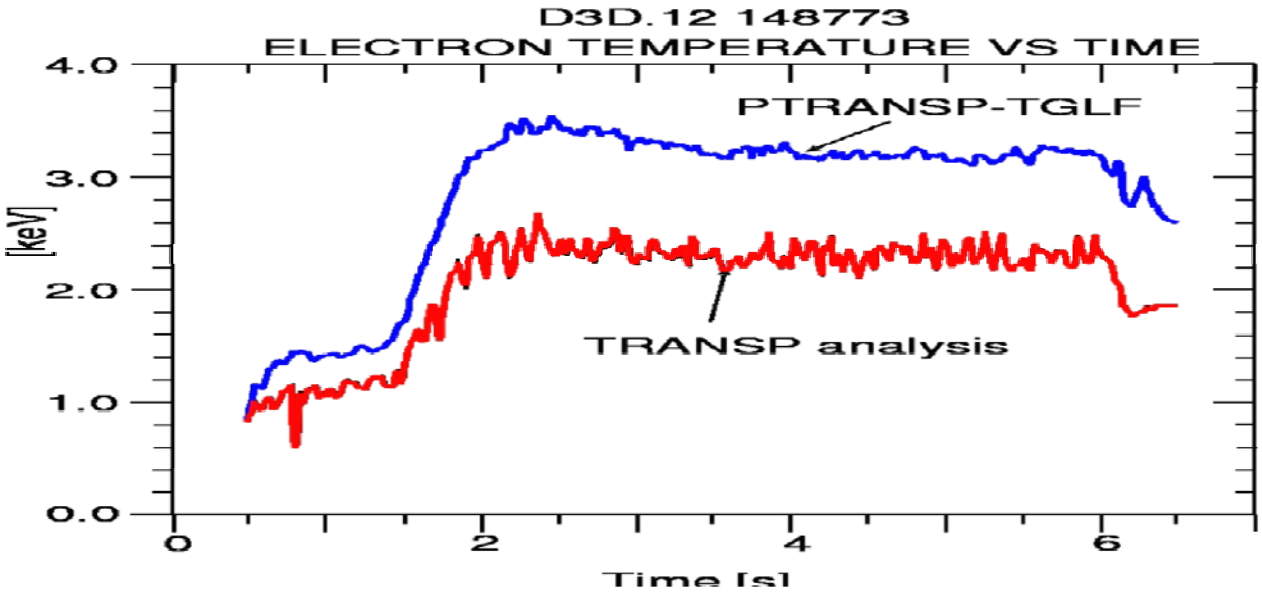
! turbulent model selected
TR_TURB_AXIAL = 'NONE'
TR_NC_AXIAL = 'NONE'
TR_EXB_AXIAL = 'NONE'
} axial region

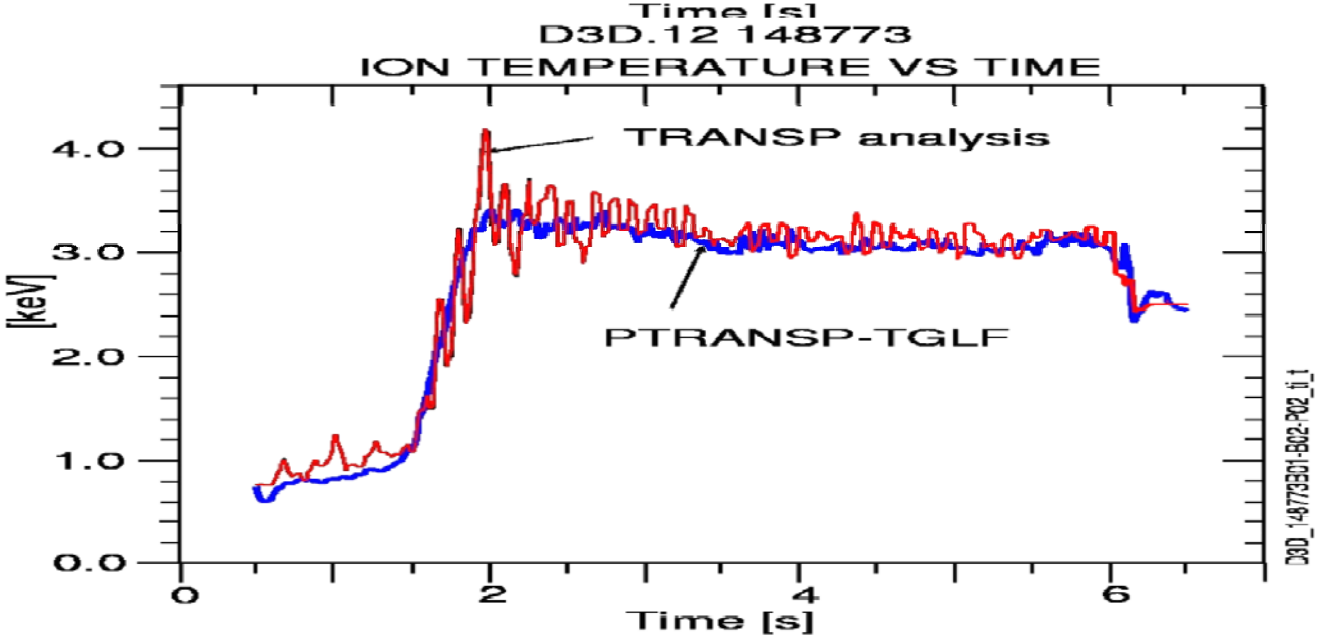
TR_TURB_EDGE = 'NONE'
TR_NC_EDGE = 'NONE'
TR_EXB_EDGE = 'NONE'
} edge region

TR_TURB_CONF = 'TGLF'
TR_NC_CONF = 'NEOCH'
TR_EXB_CONF = 'DMEXB'
} conf. region

And namelists for TGLF switches (not show here)

Test TRANSP runs with PT-SOLVER





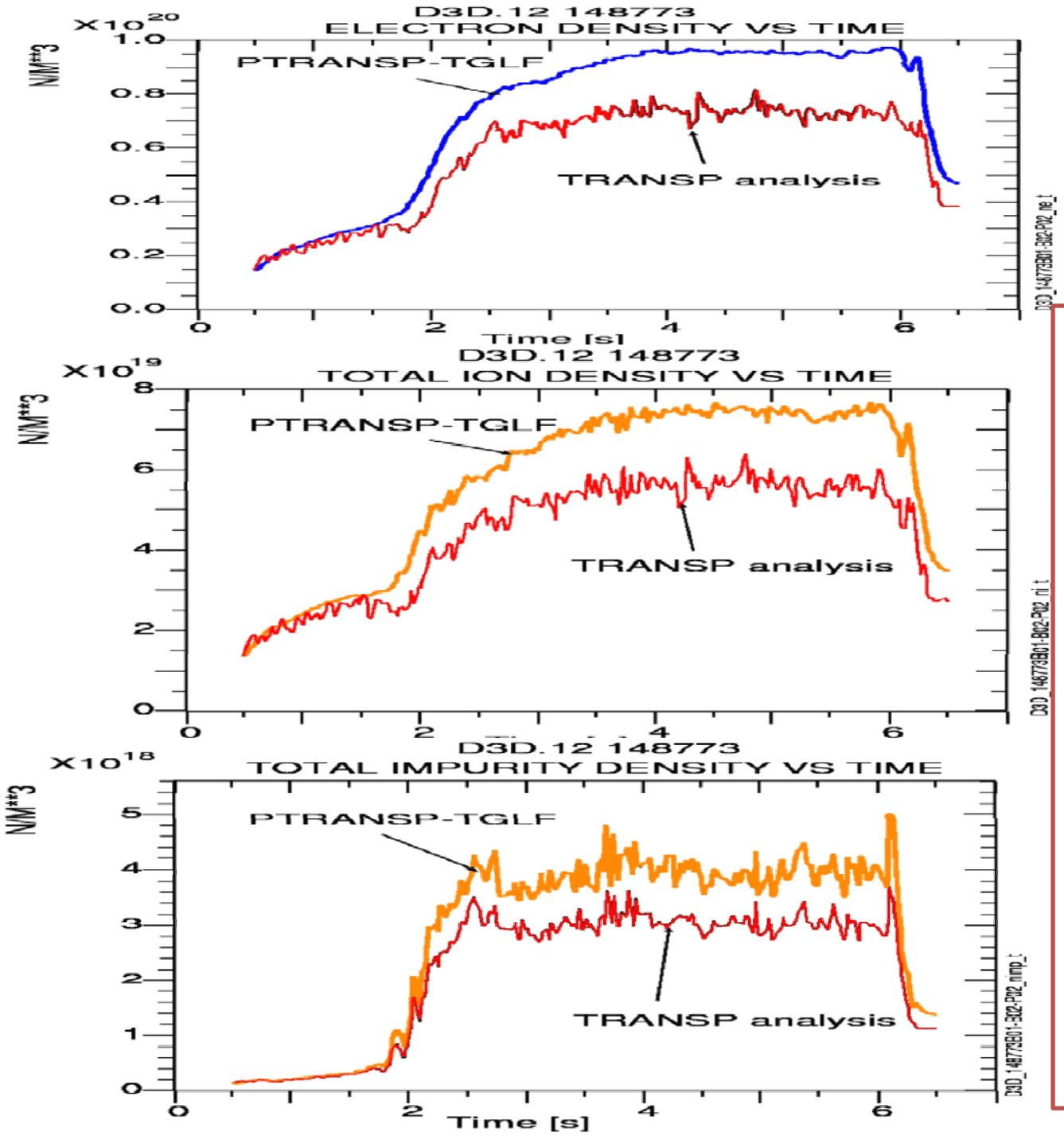
Parallel PT-SOLVER, and parallel NUBEAM used.

Te, Ti & Ne Predicted
Ni, Nimp derived from quasi-neutrality, and Zeff

3 species in TGLF (electrons, ions, and impurities)

Alpha_E = 0.5
Alpha_P = 0.5
Quench_rule used

Nzones = 50
64 cores used in dawson cluster for 7 days



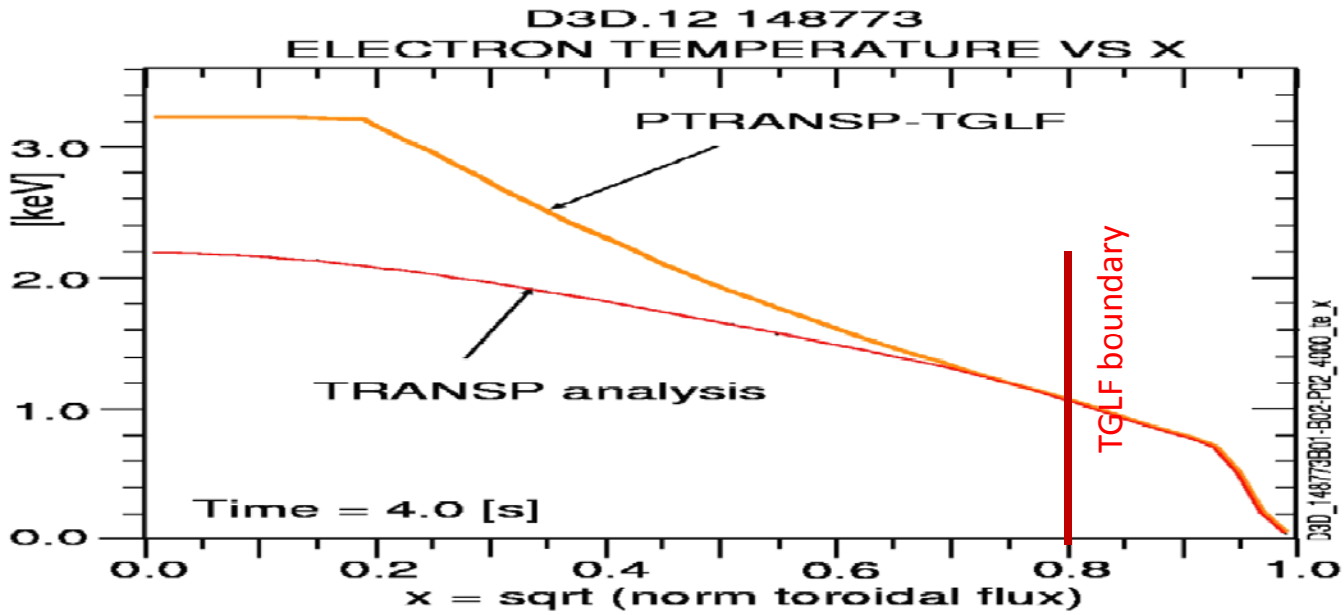
Parallel PT-SOLVER, and parallel NUBEAM used.

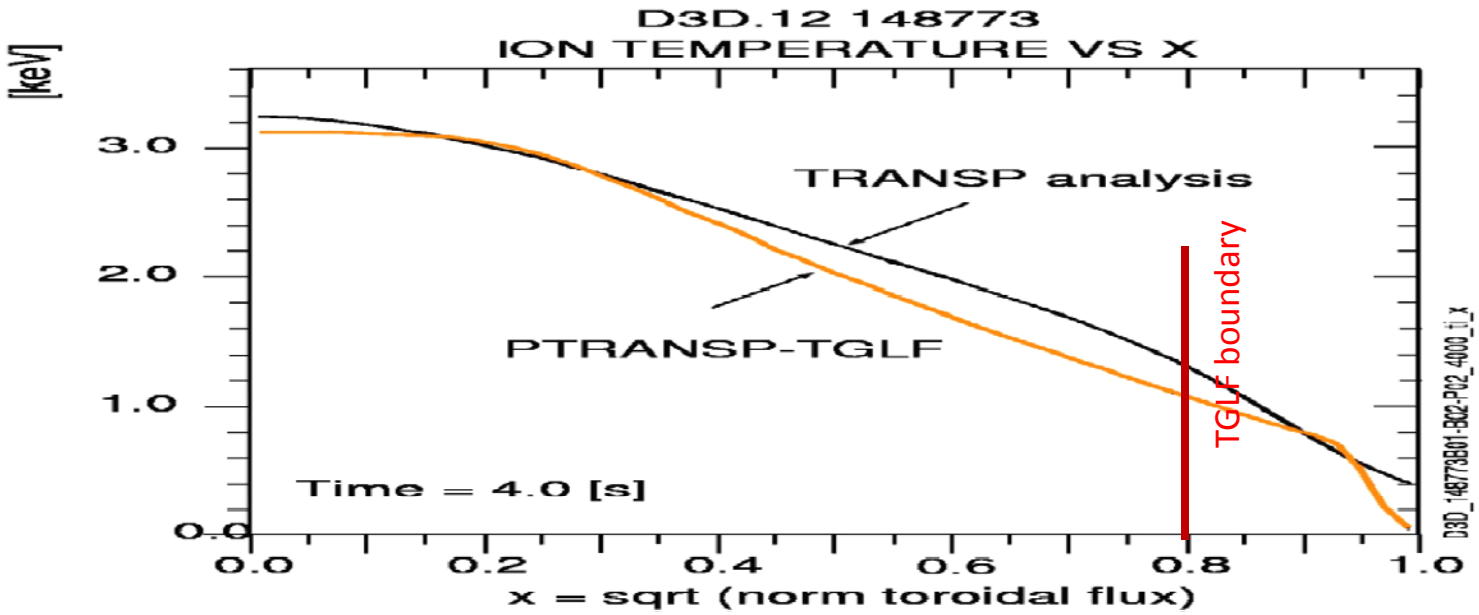
Te, Ti & Ne Predicted
Ni, Nimp derived from quasi-neutrality, and Zeff

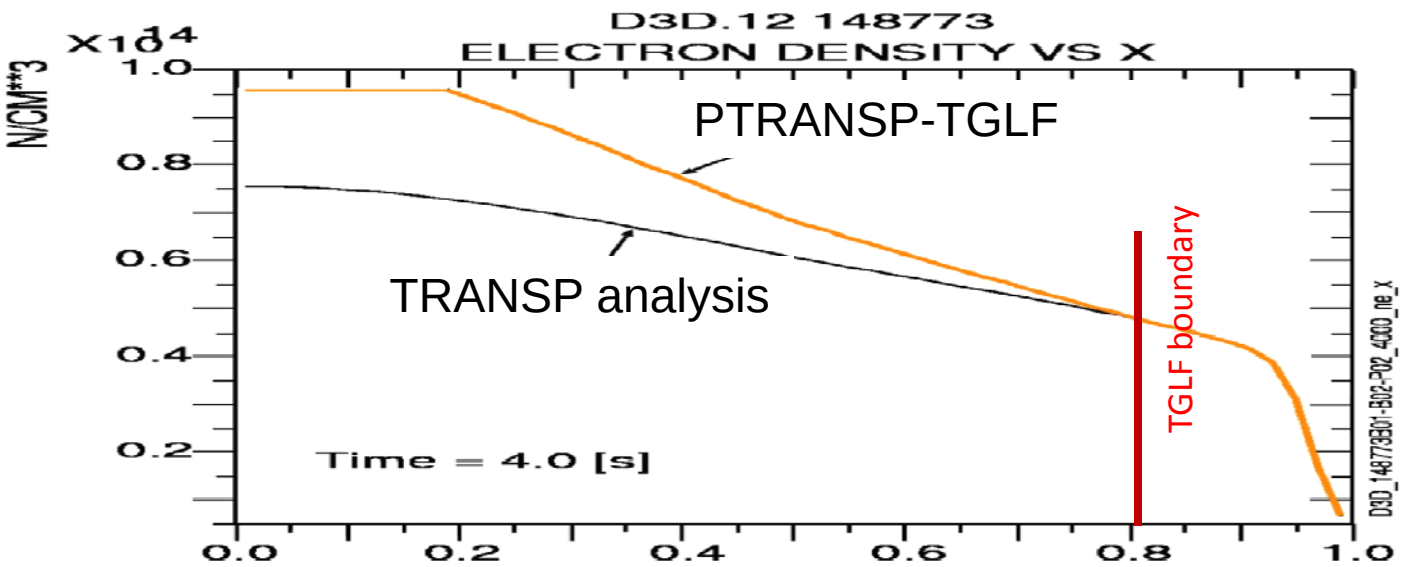
3 species in TGLF (electrons, ions, and impurities)

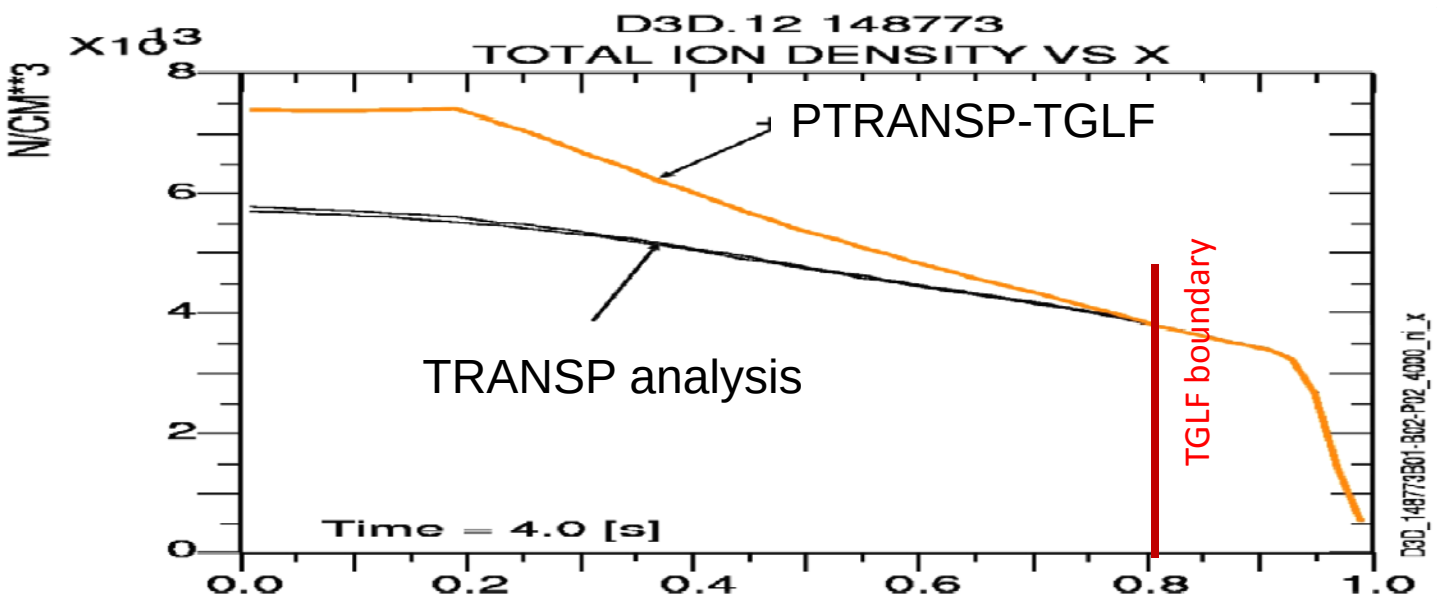
Alpha_E = 0.5
Alpha_P = 0.5
Quench_rule used

Nzones = 50
64 cores used in dawson cluster for 7 days

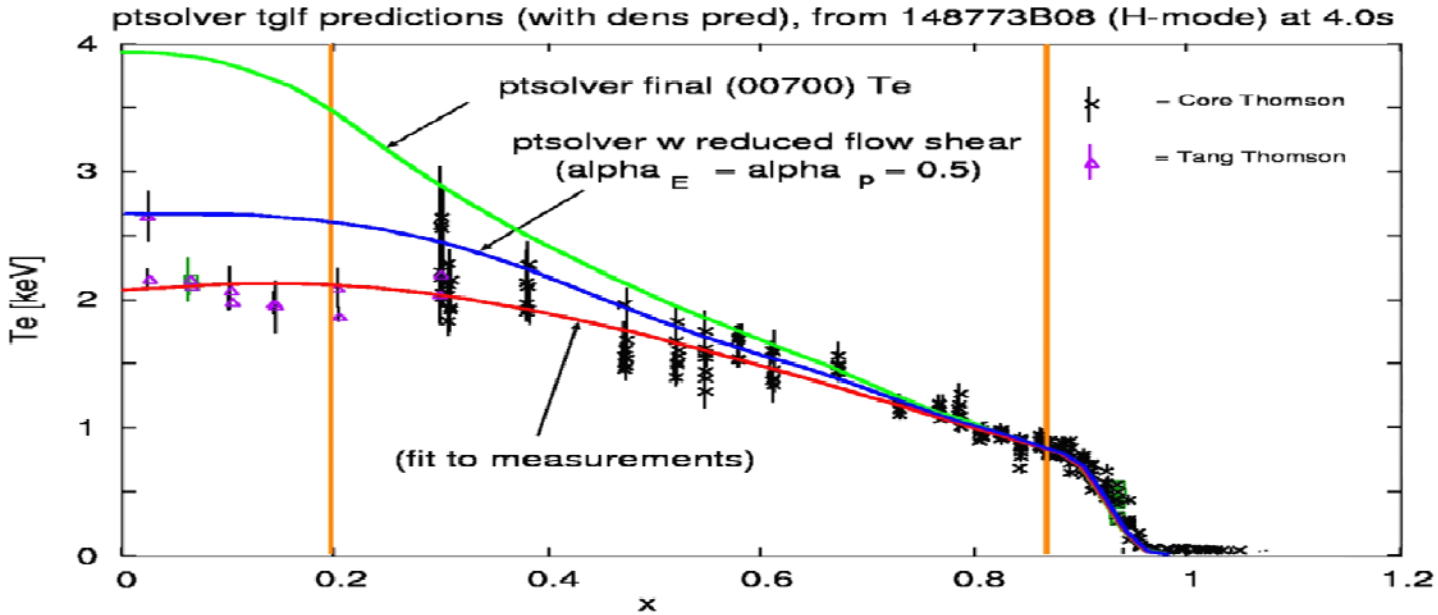

D3D.12 148773
ELECTRON TEMPERATURE VS X
[keV]
3.0
2.0
1.0
0.0
PTRANSP-TGLF
TRANSP analysis
TGLF boundary
Time = 4.0 [s]
0.0
0.2
0.4
0.6
0.8
1.0
x = sqrt (norm toroidal flux)
D3D_148773B01-B02-Pt2_4000_te_x

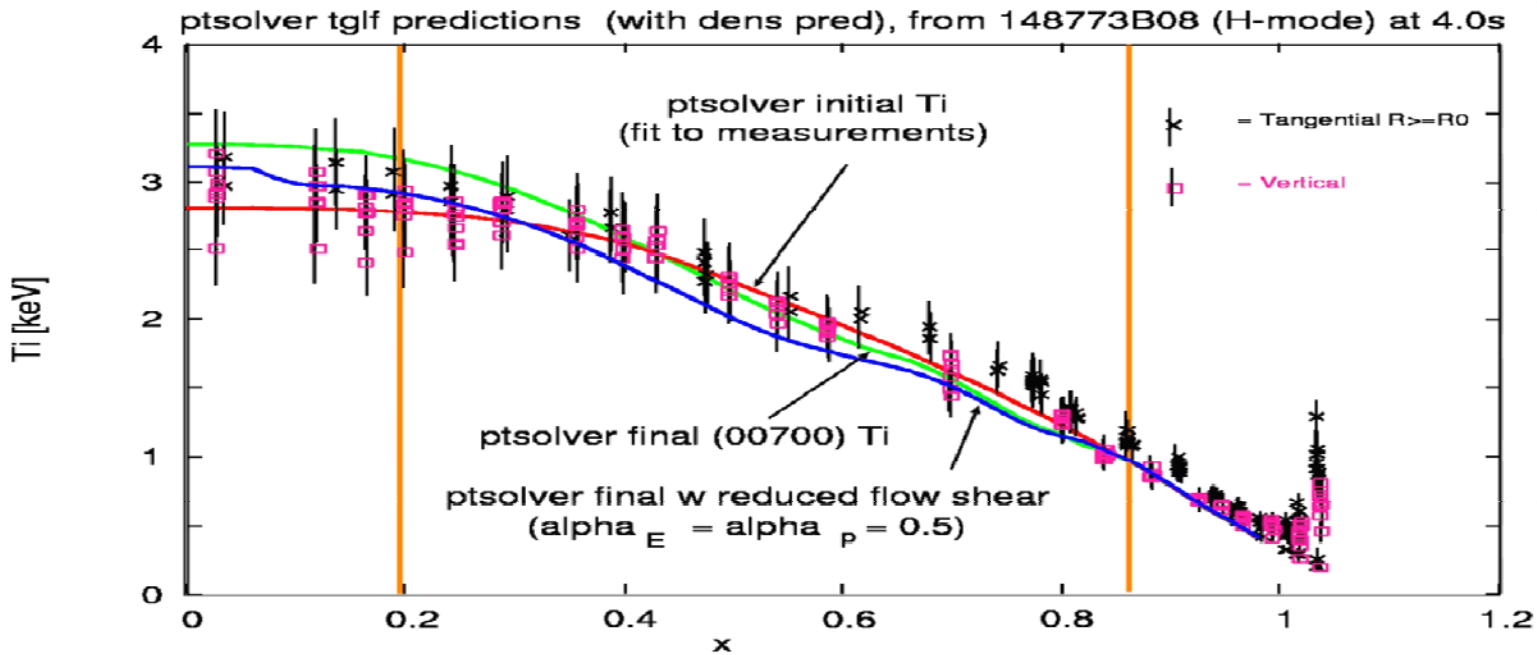

D3D.12 148773
ION TEMPERATURE VS X
[keV]
3.0
2.0
1.0
0.0
TRANSP analysis
PTRANSP-TGLF
TGLF boundary
Time = 4.0 [s]
0.0
0.2
0.4
0.6
0.8
1.0
x = sqrt (norm toroidal flux)
D3D_148773B01-B02-Pt2_4000_ti_x


D3D.12 148773
ELECTRON DENSITY VS X
X10^14
N/CM**3
1.0
0.8
0.6
0.4
0.2
0.0
0.2
0.4
0.6
0.8
1.0
PTRANSP-TGLF
TRANSP analysis
TGLF boundary
Time = 4.0 [s]
D3D_148773B01-B02-P02_4000_ne_x


D3D.12 148773
TOTAL ION DENSITY VS X
X10^13
N/CM**3
8
6
4
2
0
0.0
0.2
0.4
0.6
0.8
1.0
PTRANSP-TGLF
TRANSP analysis
TGLF boundary
Time = 4.0 [s]
D3D_148773B01-B02-P02_4000_ni_x

Test runs with standalone PT-SOLVER





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.
- Several turbulent, neoclassical, or data driven choices models, including GLF23, TGLF, NEO, NCLASS, and Chang-Hilton
- Two level parallelization has been implemented in PT_SOLVER with MPI library
- PT-SOLVER has been successfully implemented in TRANSP.
- Development and benchmarking density prediction capability.
- Development and benchmarking angular momentum prediction.
- Install and test more turbulent, and neoclassical models.
- Development and testing more robust nonlinear stiff solver
- Testing more cases in TRANSP with the combination of different components

More info? Please leave your email…