

The GCNMP Module Summary

- GCNMP is a “globally convergent Newton method parallel solver” module that solves the tokamak diffusion equations. The module is currently under development for use in PTRANSP.
- A priori static selection of equations to run in simulation and analysis mode, possibly changing on each time step.
- Auxiliary sources (rf, beam, etc) supplied by external modules
- Effective adaptive grid possible by changing iterdb files between calls.
- Solver demonstrates application of a standard Newton method with a line search. To avoid failure in a transport code environment additional methods using higher order curvature terms determined as part of the solution process are dynamically introduced as needed:
 - Forced positive definite Cholesky factorization
 - trust region methods: hook step, dogleg
 - steepest descent
 - Homotopy type method based on confinement model specific parameters .



Recent development

- GCNMP is now parallel (MPI)
- Netcdf version of iterb file
- Flow equation for density added.
- Equation splitting developed to allow arbitrary combination of equations to be run in simulation on alternate steps.
- Allow mixture of boundary condition specification of density/particle flux and/or temperature/energy flux at arbitrary rho locations to facilitate interface with edge codes. (Not all combinations currently work)



GCNMP Solver details

- GCNMP code solves the set of tokamak transport equations

$$\underline{\underline{M}} \frac{\partial}{\partial t} \Big|_{\zeta} \underline{u} - \frac{1}{H\rho} \frac{\partial}{\partial \rho} \left(H\rho \underline{\underline{D}} \frac{\partial}{\partial \rho} \underline{u} \right) + \frac{1}{H\rho} \frac{\partial}{\partial \rho} \left(H\rho \underline{\underline{V}} \underline{u} \right) + \underline{\underline{W}} \underline{u} = \underline{S}_{ext} \quad (1)$$

- Vector $\underline{u} \equiv [n_1, ..n_N, T_e, T_i, FGH\rho B_P, \omega]$ represent the dependent variables , $\underline{\underline{M}}$ is an $N + 4$ by $N + 4$ coefficient matrix with N ion species.
- The total flux, $\underline{\Gamma}$, consists of diffusive and convective parts :

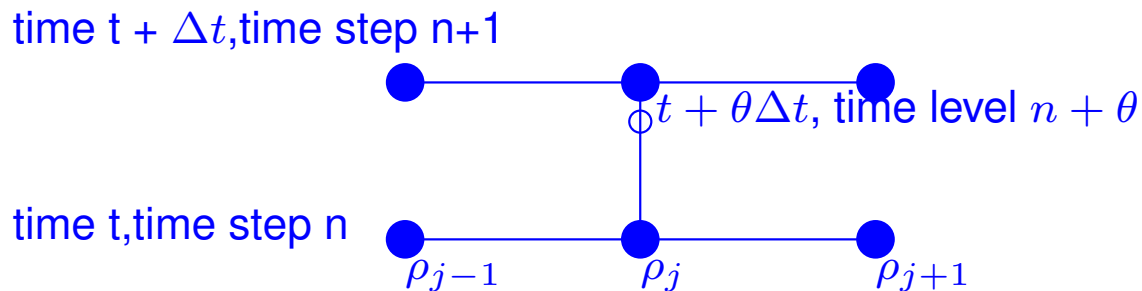
$$\underline{\Gamma} = \underline{\Gamma}_D \left(= -\underline{\underline{D}} \frac{\partial \underline{u}}{\partial \rho} \right) + \underline{\Gamma}_C \left(= -\underline{\underline{V}} \underline{u} \right)$$

- The diffusion matrix D and convection matrix V have a form which depends on the confinement models under investigation.
- The matrix \mathbf{W} is introduced for numerical stability purposes It is used to handle the electron-ion energy exchange term implicitly



Numerical considerations ...

- The partial differential equations are transformed for numerical solution by discretization in time and space.
- Space derivatives are done in central difference form for the diffusive terms
- an upwind difference form is used for the convective terms
- a two point forward difference is used for the time derivatives.



Numerical considerations ...

- The space discretization is done first. We use an integral approach to obtain the finite differences in conservative form. The volume element is $4\pi^2 R_0 H \rho d\rho$. Thus at an arbitrary interior mesh point j we have:

$$\int_{\rho_{j-\frac{1}{2}}}^{\rho_{j+\frac{1}{2}}} d\rho H \rho \underline{\underline{M}} \frac{\partial}{\partial t} \Big|_{\zeta} \underline{u} - \int_{\rho_{j-\frac{1}{2}}}^{\rho_{j+\frac{1}{2}}} d\rho \frac{\partial}{\partial \rho} \left(H \rho \underline{\underline{D}} \frac{\partial}{\partial \rho} \underline{u} \right) \quad (2)$$

$$+ \int_{\rho_{j-\frac{1}{2}}}^{\rho_{j+\frac{1}{2}}} d\rho \frac{\partial}{\partial \rho} \left(H \rho \underline{\underline{V}} \underline{u} \right) + \int_{\rho_{j-\frac{1}{2}}}^{\rho_{j+\frac{1}{2}}} d\rho H \rho \underline{\underline{W}} \underline{u} = \int_{\rho_{j-\frac{1}{2}}}^{\rho_{j+\frac{1}{2}}} d\rho H \rho \underline{\underline{S}}_{ext} \quad (3)$$



Numerical considerations ...

- The discretized equations have a “tridiagonal” structure. The matrices $\underline{\underline{P}}, \underline{\underline{Q}}, \underline{\underline{R}}$ depend on the vector \vec{u} .

$$\underline{\underline{M}}_j \frac{\partial u_j}{\partial t} - \underline{\underline{P}}_{j-1} u_{j-1} - \underline{\underline{Q}}_j u_j - \underline{\underline{R}}_{j+1} u_{j+1} = \underline{\underline{S}}_{exp,j} \quad (4)$$

- Note that Eq[4] is now a set of ODES which could be solved by method of lines.
- GCNMP uses an implicit time difference scheme to eliminate the time derivative - resulting in an algebraic set of equations.
- A zero gradient boundary condition is imposed at the magnetic axis and specified values of the dependent variables are incorporated near the plasma edge (which may be different for each variable).



Numerical considerations ...

- For a common boundary the block tridiagonal system to be solved is:

$$\begin{pmatrix} \underline{\underline{B}}_1^{n+\theta} & \underline{\underline{C}}_1^{n+\theta} & 0 & 0 & 0 \\ \underline{\underline{A}}_2^{n+\theta} & \underline{\underline{B}}_2^{n+\theta} & \underline{\underline{C}}_2^{n+\theta} & 0 & 0 \\ 0 & \underline{\underline{A}}_3^{n+\theta} & \underline{\underline{B}}_3^{n+\theta} & \underline{\underline{C}}_3^{n+\theta} & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \underline{\underline{B}}_{nj}^{n+\theta} \end{pmatrix} \begin{pmatrix} \underline{u}_1^{n+1} \\ \underline{u}_2^{n+1} \\ \underline{u}_3^{n+1} \\ \vdots \\ \underline{u}_{nj}^{n+1} \end{pmatrix} = \begin{pmatrix} \underline{g}_1^{n+\theta} \\ \underline{g}_2^{n+\theta} \\ \underline{g}_3^{n+\theta} \\ \vdots \\ \underline{g}_{nj}^{n+\theta} \end{pmatrix} \quad (5)$$

- Each sub-matrix, $\underline{\underline{A}}$, $\underline{\underline{B}}$, $\underline{\underline{C}}$ is $n+4$ by $n+4$, where n is the number of ion species and the 4 comes from the remaining dependent variables (T_e, T, B_P, ω) The vector \underline{u}_j contains the dependent variables at grid point j and we have assumed a grid of size n_j . The matrices $\underline{\underline{A}}_j, \underline{\underline{B}}_j, \underline{\underline{C}}_j$ depend on $\underline{u}(t + \theta * dt, \rho_j)$ and are combinations of the $\underline{\underline{P}}, \underline{\underline{Q}},$ and $\underline{\underline{R}}$ matrices defined above.

Fully Implicit Solver

- $\theta = 1$ defines a fully implicit solver. Eq.(5) represent a set of non-linear equations of the form

$$F_i = 0, i = 1..(n_j - 1) * (n + 4) - 1$$

to be solved for $n_j, Te_j, Ti_j, RBP_j, \omega_j$ at each grid point $r_j, j = 1, ..n_j$.

- Such sets of equations can be solved using a Newton type method enhanced with a strategy that insures “global” convergence.
- No single global strategy will work reliably with confinement models such as GIF23 (which is part of matrix D). Here three methods are used in a round robin type approach to generate the solution:
 - line search
 - and two trust region methods which change both the step size and direction:
 - dog leg
 - hook step



Fully Implicit Solver...

- The zeros of the function

$$f(u) = \frac{1}{2} \sum F_i(u)^2 \quad (6)$$

are possible solution(s) of EQS(5). The relative minima also present in Eq.(6) have the property that at such points

$$\nabla f = \underline{\underline{J}}^T \underline{F} = 0 \quad (7)$$

But a relative minimum has $\underline{F} \neq 0$ which implies that the columns of J must be linearly dependent at such points. The Jacobian is perturbed away from such points by adding a minimal perturbation to the diagonals that insures that J has an acceptable condition number.



Fully Implicit Solver...

- The local linear representation of the set of EQS(5):

$$\underline{F}(\underline{u}_c + \underline{s}) = \underline{F}(\underline{u}_c) + \underline{J}(\underline{u}_c) \underline{s} \quad (8)$$

leads to the Newton step \underline{s} to be taken from the current approximate solution point \underline{u}_c by solving

$$\underline{J}(\underline{u}_c) \underline{s} = -\underline{F}(\underline{u}_c) \quad (9)$$

- To introduce higher order terms in the global strategy that allow for a deviation from the Newton direction we note that the quadratic form

$$\frac{1}{2} \underline{F}(\underline{u}_c + \underline{x})^T \underline{F}(\underline{u}_c + \underline{x}) = \frac{1}{2} \underline{F}^T \underline{F} + \left(\underline{J}^T \underline{F} \right)^T \underline{x} + \frac{1}{2} \underline{x}^T \underline{J}^T \underline{J} \underline{x} \quad (10)$$

is positive for all \underline{x} except the Newton solution $\underline{x} = \underline{s}$ where its value is zero.



Fully Implicit Solver...

- The quadratic form, EQ(10) is closely related to the quadratic form of f defined in EQ(6)

$$f(\underline{u}_c + \underline{x}) = f(\underline{u}_c) + \left(\underline{J}^T \underline{F} \right)^T \underline{x} + \frac{1}{2} \underline{x}^T \underline{H} \underline{x} \quad (11)$$

where the Hessian H is $\underline{J}^T \underline{J}$ plus terms involving the second derivative of f . To minimize f we would look for the minimizer of this local representation of f .

- The dogleg and hook step trust region strategies are based on finding the minima in f using the modified form, EQ(10) subject to the constraint that the step size is limited to an a priori specified length, δ . It can be shown that any such constrained local minimizer of EQ(10) is in a direction that decreases the value of f so long as the approximation to H is positive definite.
- For the hook step the solution is

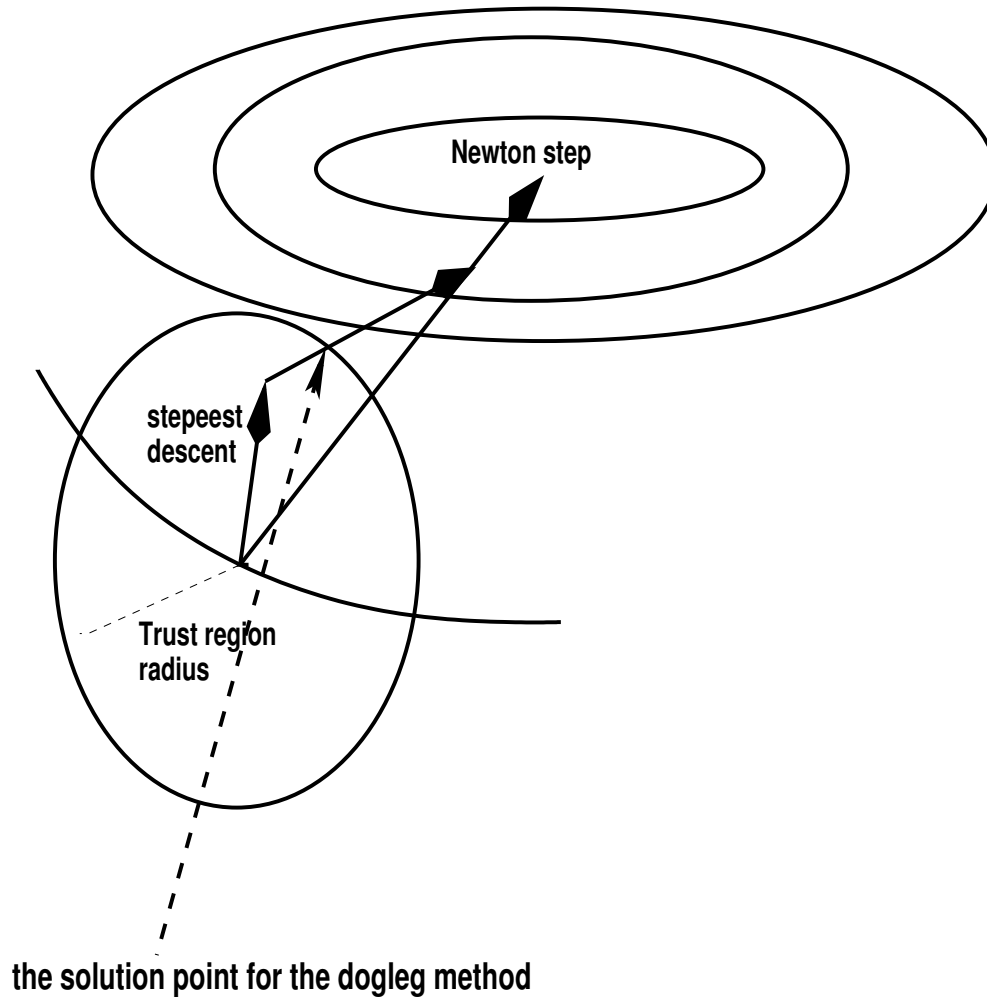
$$\underline{x} = - \left(\underline{J}^T \underline{J} + \mu \underline{I} \right)^{-1} \underline{J}^T \underline{F} \quad (12)$$

where $\mu_c \geq 0$ is found iteratively so that $\|\underline{x}\| \approx \delta$. For small μ we approach the newton direction while for large μ the steepest decent direction is approached.



Fully Implicit Solver...

Hook step and Dogleg Geometry



GCNMP Parallel Solver Module

- Multi level parallelization. Requires inter and intra communicator structure:
 - Jacobian breaks up into sets of independent columns due to structure of finite difference equations. This defines independent communicator groups. Currently we simply default to one group.
 - Within a communicator group:
 - Standard domain decomposition is applied over the rho grid.
 - Example 2 density ,Te,Ti,Rbp,w,on 51 point grid implies about 300 linearized equations to be solved.
 - Pure Newton step requires standard LU decomposition. Both LU and Cholesky should be implemented with Plapack or similar.
- The Jacobian in GCNMP has a dynamically alterable (or user selectable) bandwidth that accommodates above structure.



Parallel Jacobian Generation

- The Jacobian has same structure as EQ[5]. The banded structure leads to parallelization dependent on bandwidth. Example for tridiagonal set:

$$\underline{\underline{J}} = \begin{pmatrix} x & x & 0, & 0, & 0, & 0, & 0, & 0 \\ x & x & x, & 0, & 0, & 0, & 0, & 0 \\ 0, & x, & x, & x, & 0, & 0, & 0, & 0 \\ 0, & 0, & x, & x, & x, & 0, & 0, & 0 \\ 0, & 0, & 0, & x, & x, & x, & 0, & 0 \\ 0, & 0, & 0, & 0, & x, & x, & x, & 0 \\ 0, & 0, & 0, & 0, & 0, & x, & x, & x \\ 0, & 0, & 0, & 0, & 0, & 0, & x, & x \\ G_1 & G_2 & G_3 & G_1 & G_2 & G_3 & G_1 & G_2 \end{pmatrix} \quad (13)$$

- Independent groups (G_1, G_2, G_3) , G_i can be done on processor $i, \forall i$ simultaneously. Tridiagonal ==> limited to 3 INTER communicators. But each communicator has further parallelization due to domain decomposition over the grid ==> INTRA communications groups and parallel solvers yield another group.



Parallel Jacobian Generation...

- Example: Communicator group G1 gets

$$\frac{\partial F_1}{\partial u_1}, \frac{\partial F_2}{\partial u_1}, \frac{\partial F_3}{\partial u_4}, \frac{\partial F_4}{\partial u_4}, \frac{\partial F_5}{\partial u_4}, \frac{\partial F_6}{\partial u_7}, \frac{\partial F_7}{\partial u_7}, \frac{\partial F_8}{\partial u_7}, \dots \quad (14)$$

in a single call by evaluating the entire set of equations (recall there are one or more equations at each grid point and hence we use the inter communicator group for this) with the perturbed

$$\underline{u} = (u_1 + \delta u_1, u_2, u_3, u_4 + \delta u_4, u_5, u_6, u_7 + \delta u_7, \dots) \quad (15)$$

Where it is assumed that the usual forward difference scheme

$$\frac{\partial F_i}{\partial u_i} = \frac{F_i(\underline{u} + \delta u_i \underline{h}_i) - F_i(\underline{u})}{\delta u_i} \quad (16)$$

is used to approximate derivatives.



Timing for domain decomposition Only

N_p	$\rho_{nj} = 51$	$\rho_{nj} = 101$	$\rho_{nj} = 201$
1	T	2.3T	8.0T
2	0.6T	1.4T	6.4T
4	0.4T	0.9T	5.6T

- Relative timing on 64 bit Linux (Lohan4 - dual cpu - 4 cores)
- T is single processor execution time on 51 point rho grid.

16:



Adaptive Rho Grid

- We take the grid spacing in rho to be proportional to the ζ grid spacing and to a **weight function** $W(\zeta, t)$. ρ grid with required bc on $\rho(\zeta, t)$ can be taken as

$$\Delta\rho = cW\Delta\zeta \implies \frac{\partial}{\partial\zeta} \left(\frac{1}{W} \frac{\partial\rho}{\partial\zeta} \right) = 0, \rho(0, t) = 0, \rho(1, t) = \rho_a(t), \rho(\zeta, 0) = \rho_a(0)\zeta \quad (17)$$

with solution and constraint on $W(\zeta, t)$:

$$\rho(\zeta, t) = \rho_a(t) \int_0^\zeta W(\zeta, t) d\zeta, \quad W(\zeta, 0) = 1, \quad \int_0^1 W(\zeta, t) d\zeta = 1 \quad (18)$$

- This method fits in naturally with mhd derived $\rho_a(t)$
- Automated choice for **W** are based on GLF23 specific features. We are trying to find optimal choices based on physical and computational considerations.

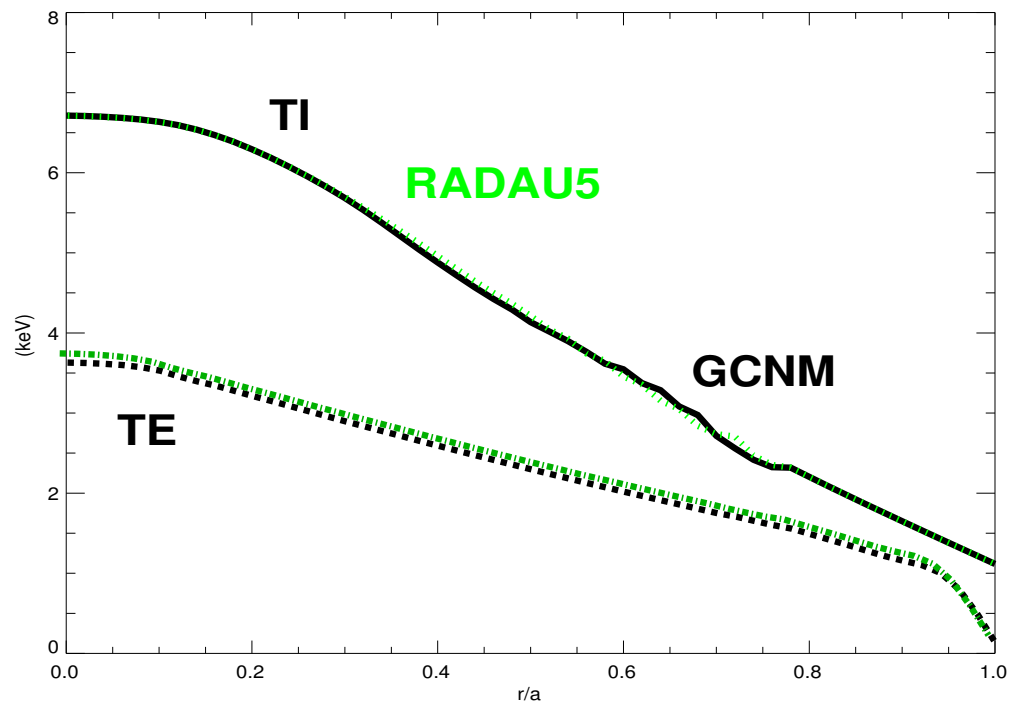
17:



GCNMP agrees with ODE solvers

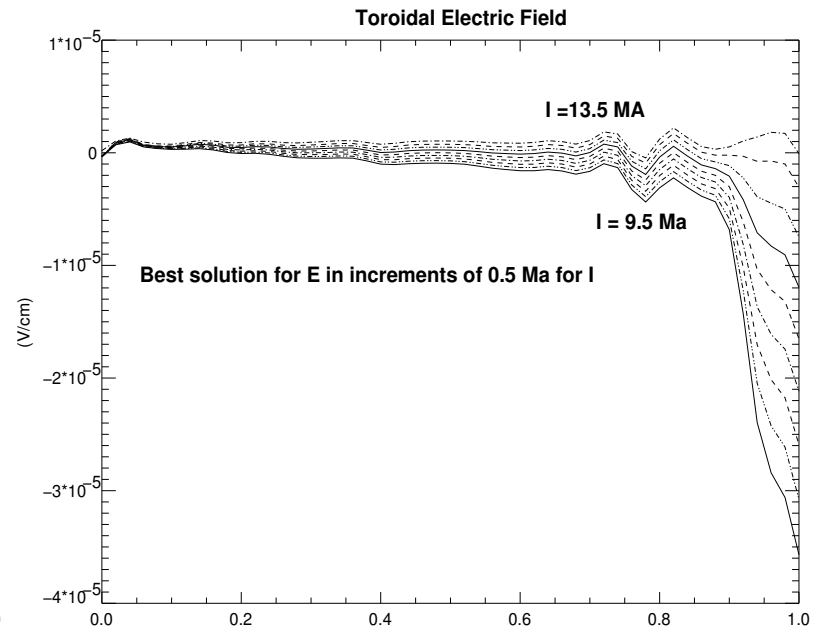
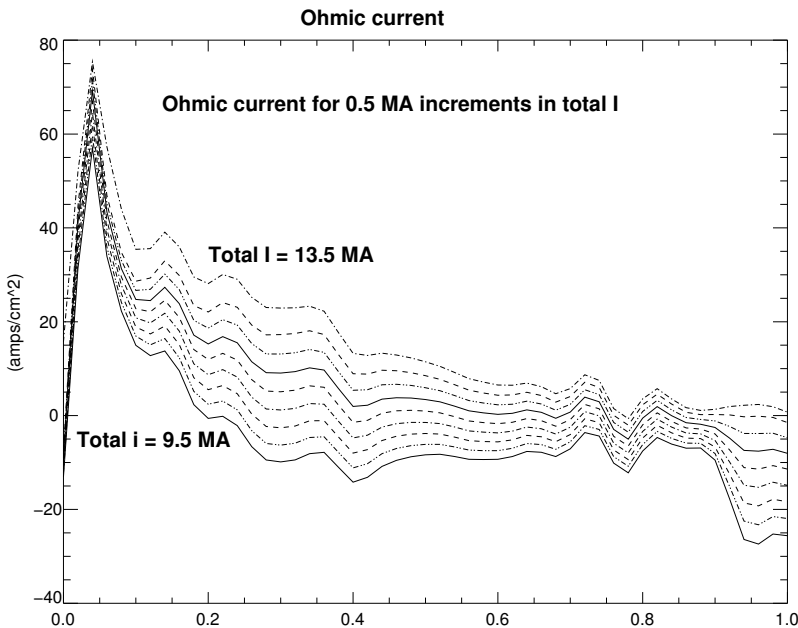
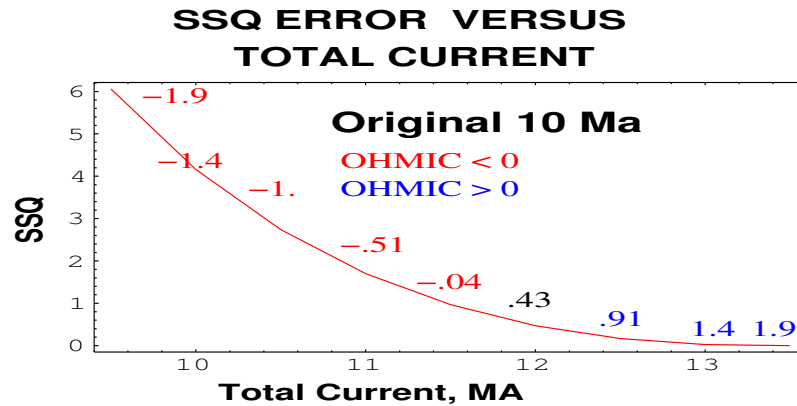
- To establish sensitivity of GLF23 to solution technique **RADAU5** and **GCNMP** were compared for typical DIII-D shots. Generally method of lines is slower.

GCNM and Radau5 Solutions



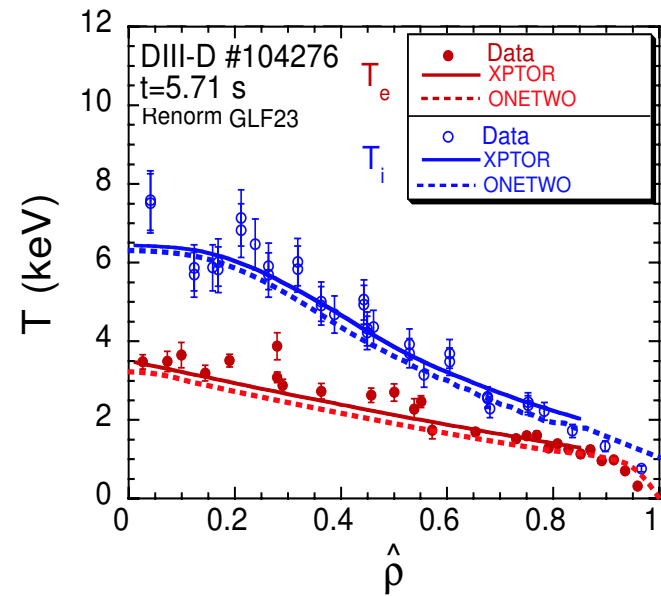
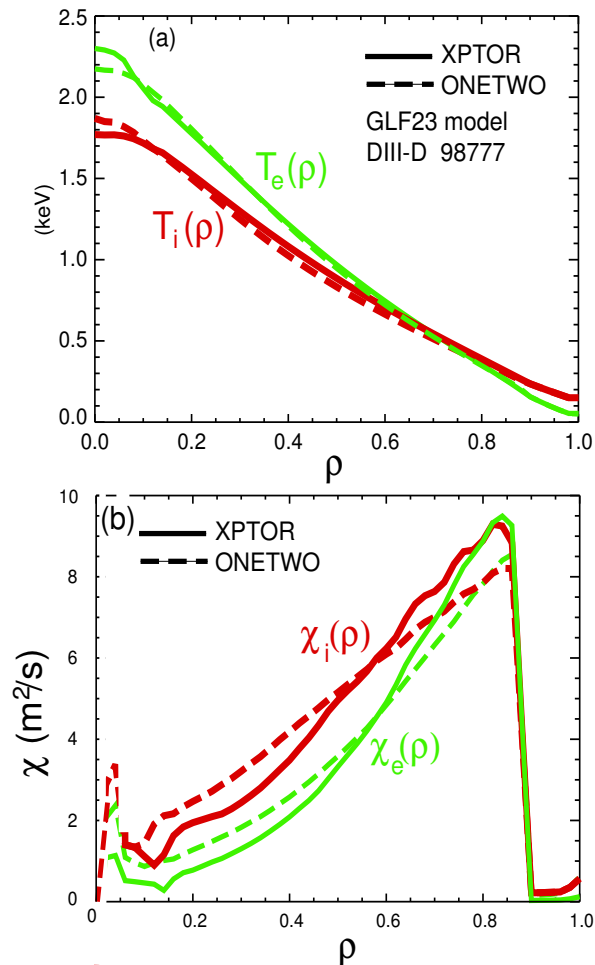
Steady State Solution May Not Exist

- Steady state self consistent solution with fixed I_p may not exist



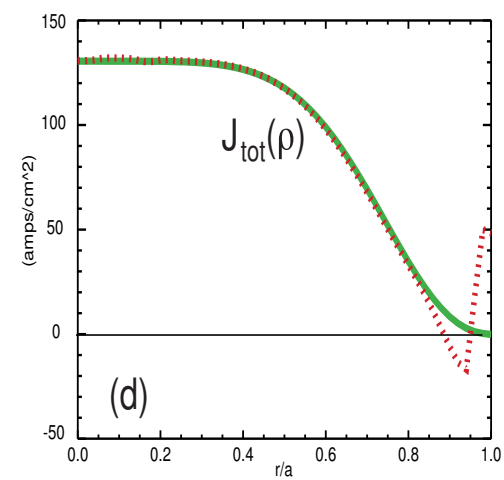
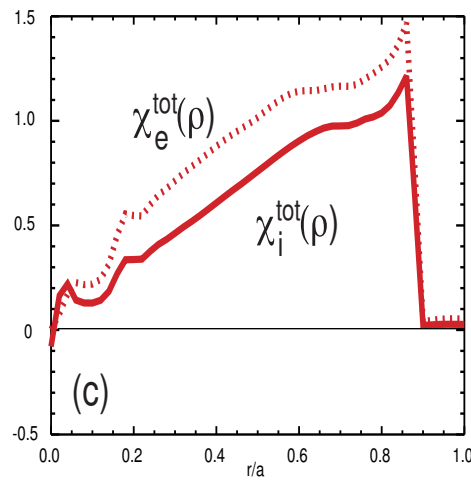
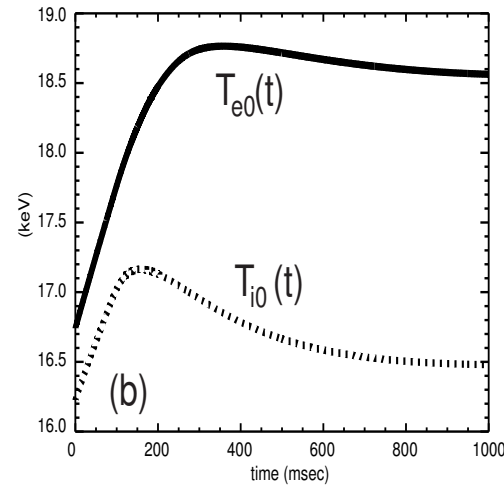
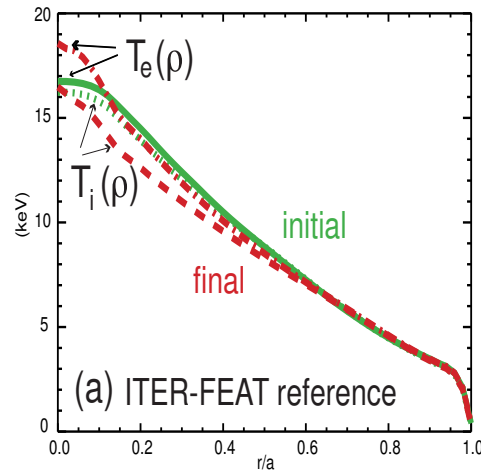
Solver Benchmarks DIII-D

- GCNMP solver module in Onetwo benchmarked against Xptor code using DIII-D data.



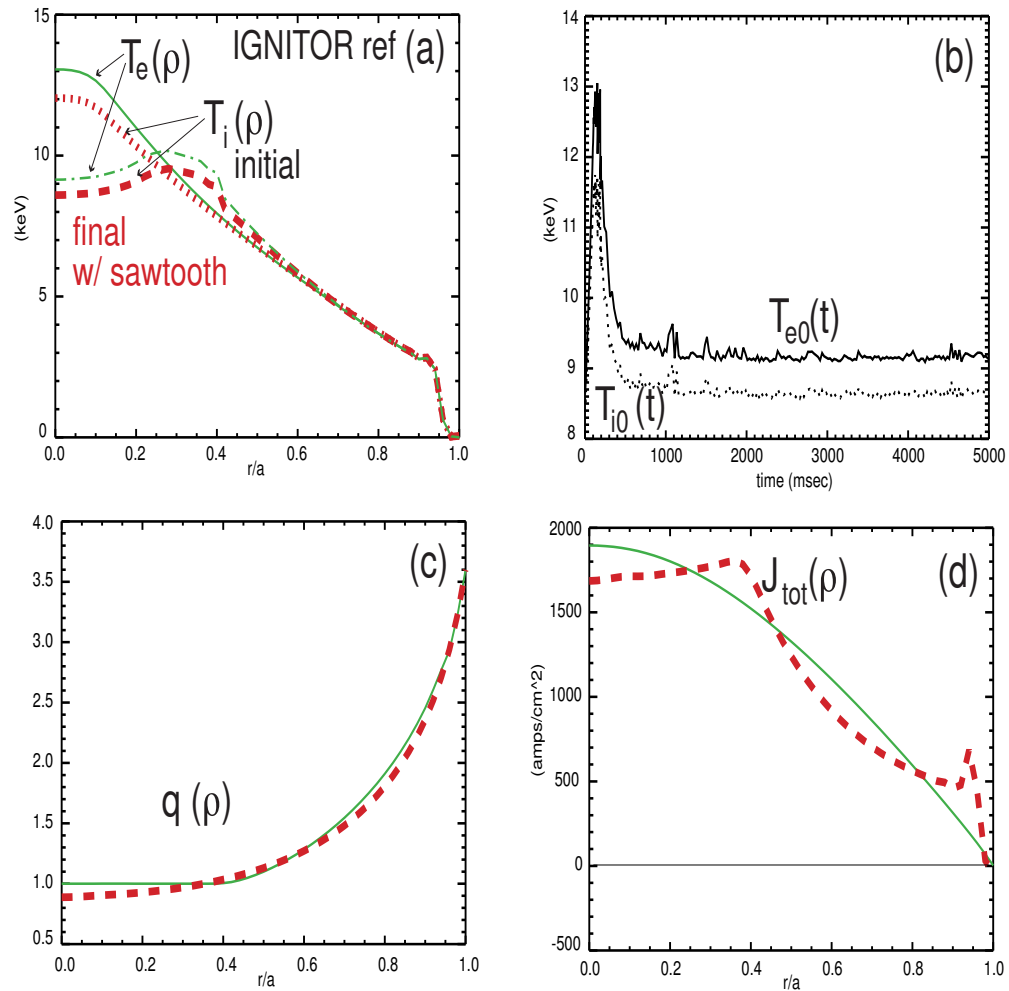
Solver Benchmark ITER

- GCNMP solver module in Onetwo compared with Xptor for Iter-FEAT:



Solver Benchmark, Ignitor with sawteeth

- GCNMP solver module in Onetwo compared with Xptor for Ignitor::



GCNMP Application: Fusion development Facility

- $B_T = 6T, I = 6.7Ma, R = 2.49m, a = 0.69m, 170\text{GHZ ECH}, 120\text{ Kev NBI}$
- Solution for T_e, T_i, ω out to $\rho = 0.93, (\Psi_N = 0.95)$
- Simulations with densities consistent with ITPA database for high performance discharges $\frac{n_e(0)}{\langle n_e \rangle} \approx 1.3 - 1.6$ and H mode confinement.
- Coupled to stable edge pressure gradient determined to be at the peeling-ballooning mode limit by Elite.
- Consistent with DIII-D observations, we set background $\chi_\omega = \chi_{ion-neo}$
- Assumed both background Te and Ti diffusivities are ion neoclassical
- Evolved for up to 3 sec. Then into steady state with constant $\langle \vec{E} \cdot \vec{B} \rangle$.
- All synchrotron radiation is assumed reabsorbed. Radiative loss is due to primary and impurity bremsstrahlung plus radiative recombination and line radiation.
- Results depend on MHD alpha stabilization in GLF23.

23:



FDF predictive modeling, $P_{Beam} =$

$20Mw, P_{ECH} = 30 + 1Mw$

$$\frac{n_e(0)}{\langle n_e \rangle} = 1.3$$

$$H89p = 2.7$$

$$H98y2 = 1.5$$

$$\beta_t = 6.3$$

$$\beta_N = 3.9$$

$$I_{ECH} = 580Ka$$

$$I_{beam} = 365Ka$$

$$I_{ohm} = 1020Ka$$

$$I_{boot} = 4750Ka$$

$$Q_{DT} = 4.4$$

$$P_{fus} = 223Mw$$

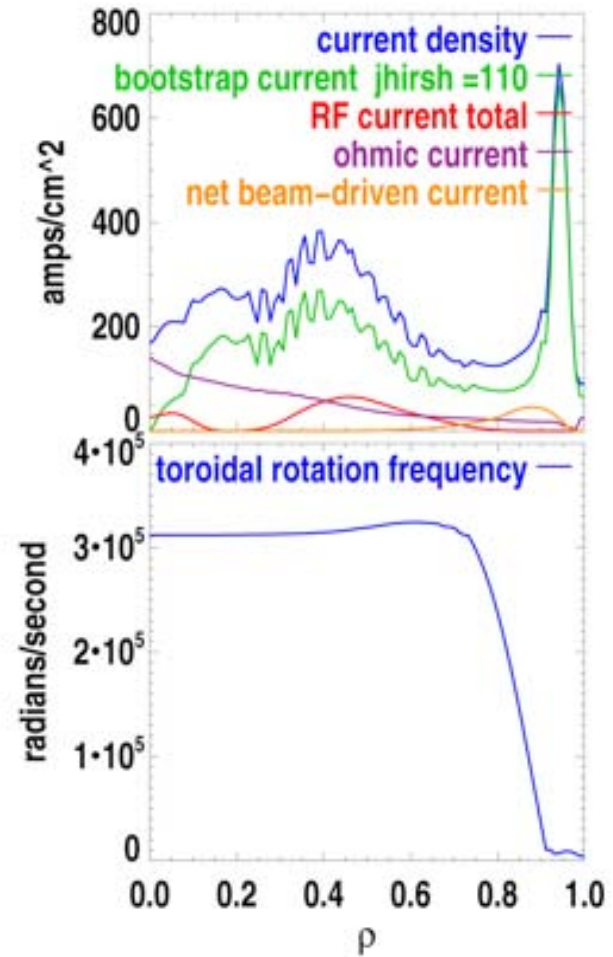
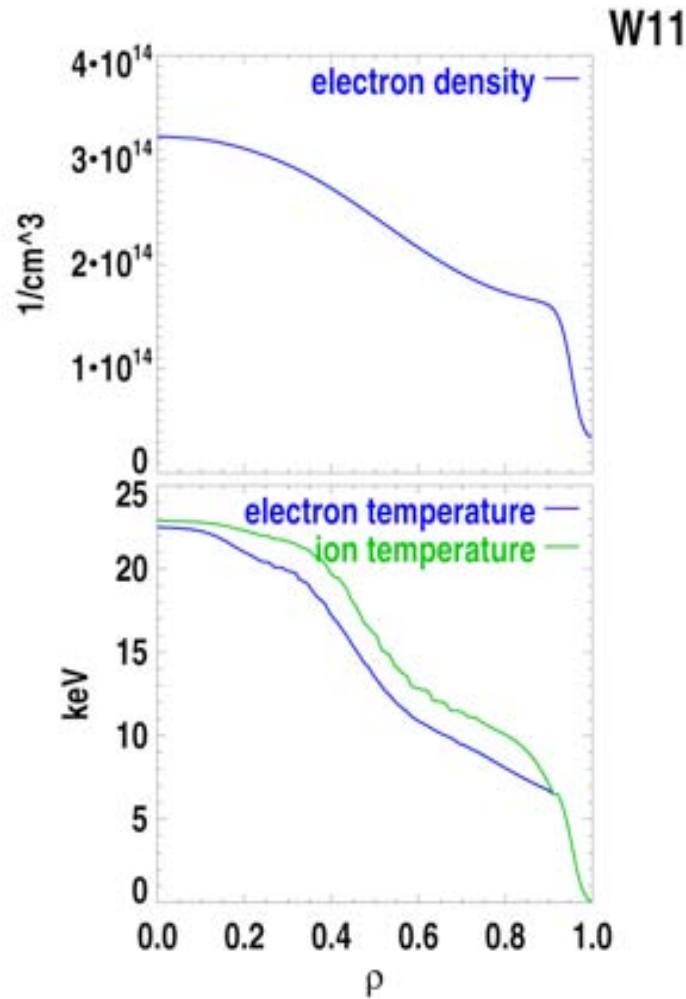
$$P_{rad} = 7Mw$$

$$P_{ECH} = 30 + 1Mw$$

$$P_{Beam} = 20Mw$$

$$W = 60MJ$$

$$q(0) = 2.5$$



FDF predictive modeling, $P_{beam} =$

30MW, $P_{ECH} = 13MW$:

$$\frac{n_e(0)}{\langle n_e \rangle} = 1.6$$

$$H89p = 3.0$$

$$H98y2 = 1.5$$

$$\beta_t = 8.1$$

$$\beta_N = 5.1$$

$$I_{ECH} = 180Ka$$

$$I_{beam} = 477Ka$$

$$I_{ohm} = 149Ka$$

$$I_{boot} = 5910Ka$$

$$Q_{DT} = 9.3$$

$$P_{fus} = 400Mw$$

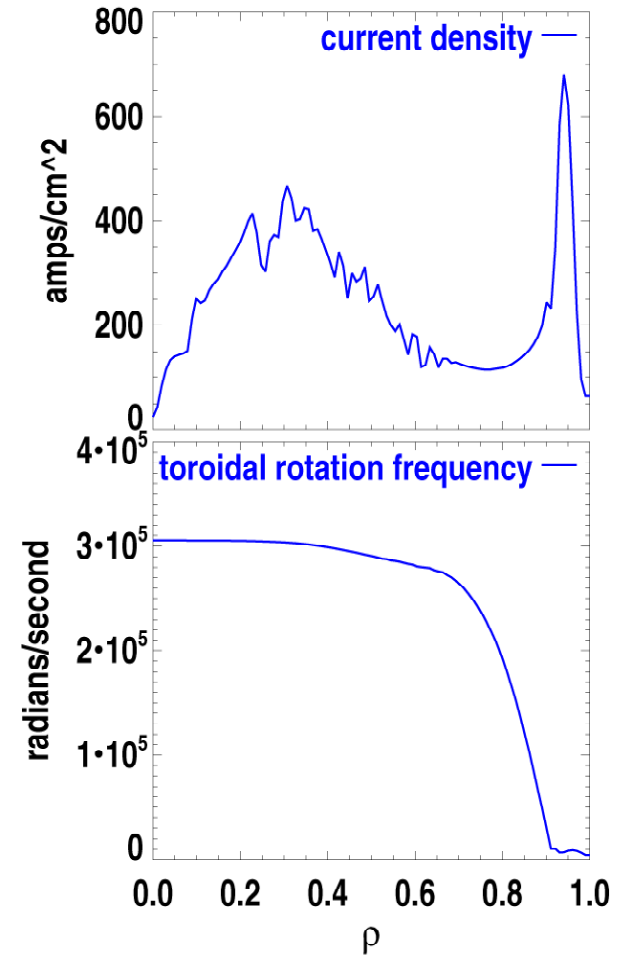
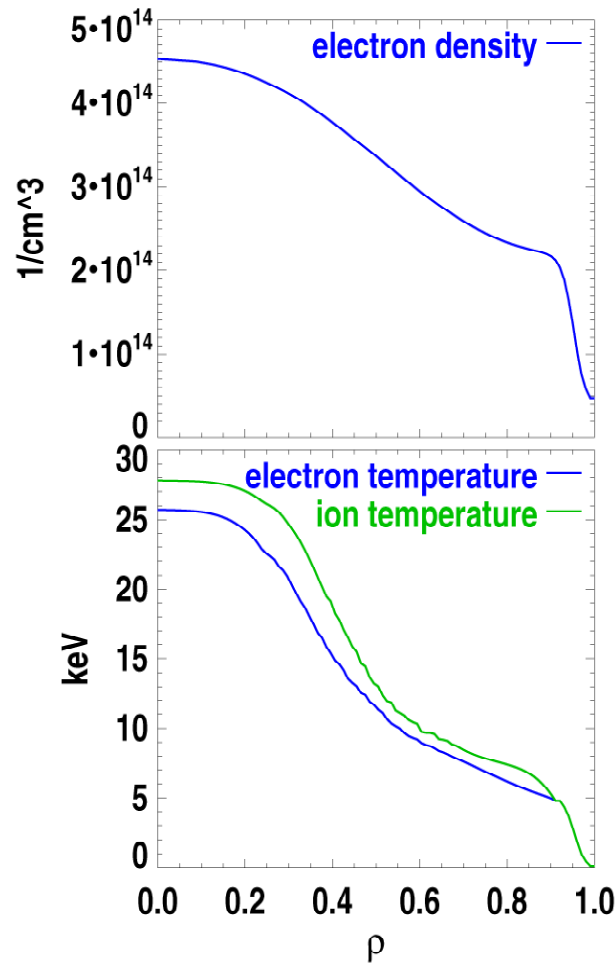
$$P_{rad} = 14Mw$$

$$P_{ECH} = 13Mw$$

$$P_{Beam} = 30Mw$$

$$W = 77MJ$$

$$q(0) = 29.$$



FDF predictive modeling, $P_{Beam} = 30Mw$

$$\frac{n_e(0)}{\langle n_e \rangle} = 1.6$$

$$H89p = 3.0$$

$$H98y2 = 1.4$$

$$\beta_t = 6.8$$

$$\beta_N = 4.2$$

$$I_{ECH} = 0Ka$$

$$I_{beam} = 474Ka$$

$$I_{ohm} = 1290Ka$$

$$I_{boot} = 4950Ka$$

$$Q_{DT} = 10$$

$$P_{fus} = 301Mw$$

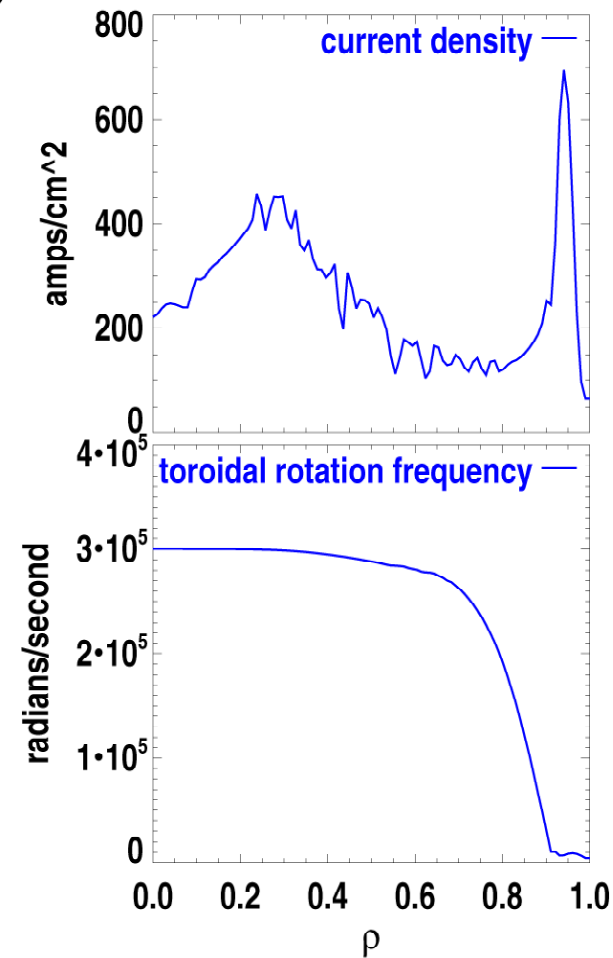
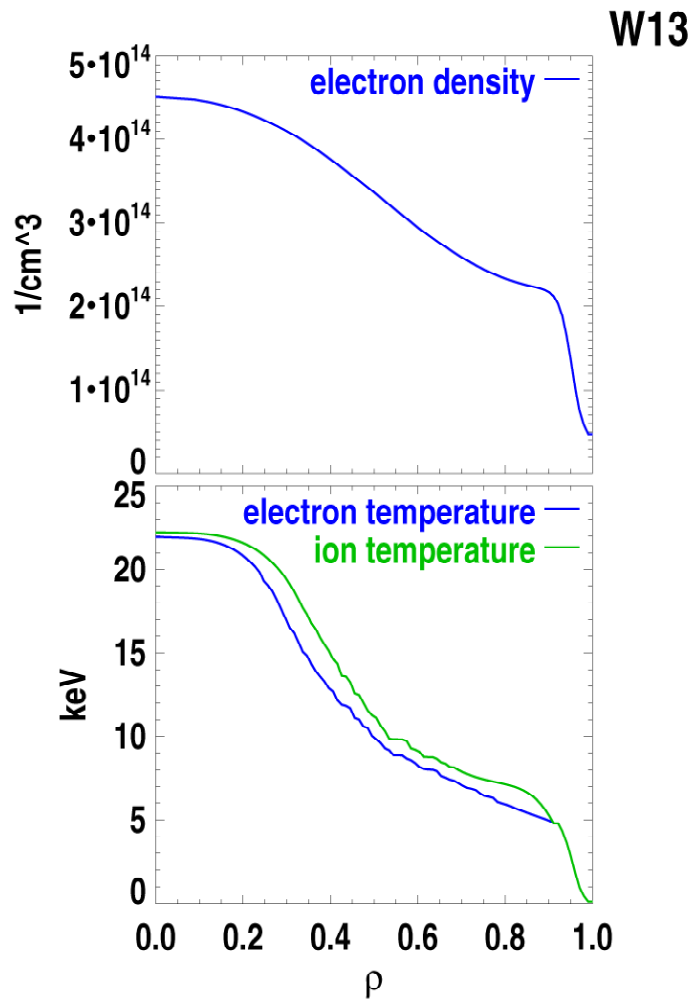
$$P_{rad} = 13Mw$$

$$P_{ECH} = 0Mw$$

$$P_{Beam} = 30Mw$$

$$W = 65MJ$$

$$q(0) = 1.9$$



FDF predictive modeling, $P_{Beam} = 15Mw$

$$\frac{n_e(0)}{\langle n_e \rangle} = 1.6$$

$$H89p = 3.3$$

$$H98y2 = 1.5$$

$$\beta_t = 7.1$$

$$\beta_N = 4.5$$

$$I_{ECH} = 0Ka$$

$$I_{beam} = 273Ka$$

$$I_{ohm} = 1550Ka$$

$$I_{boot} = 48990Ka$$

$$Q_{DT} = 22$$

$$P_{fus} = 329Mw$$

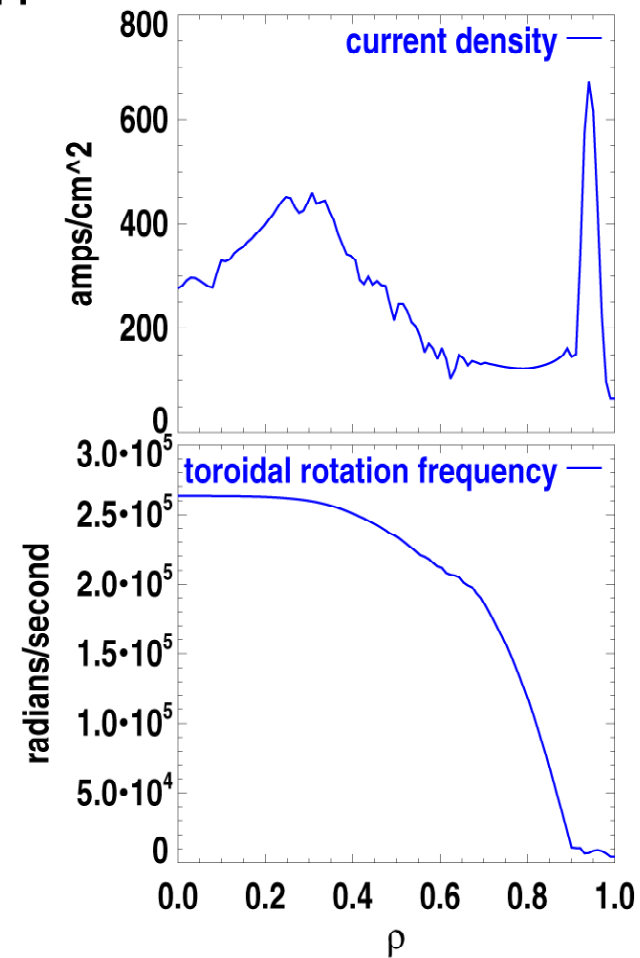
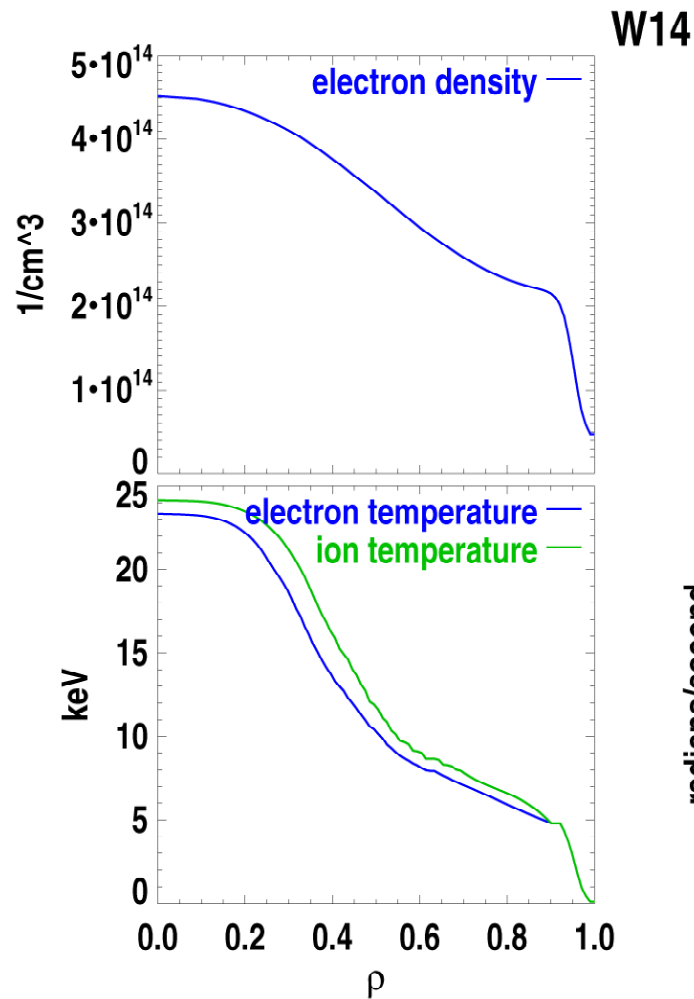
$$P_{rad} = 13Mw$$

$$P_{ECH} = 0Mw$$

$$P_{Beam} = 15Mw$$

$$W = 68MJ$$

$$q(0) = 1.5$$

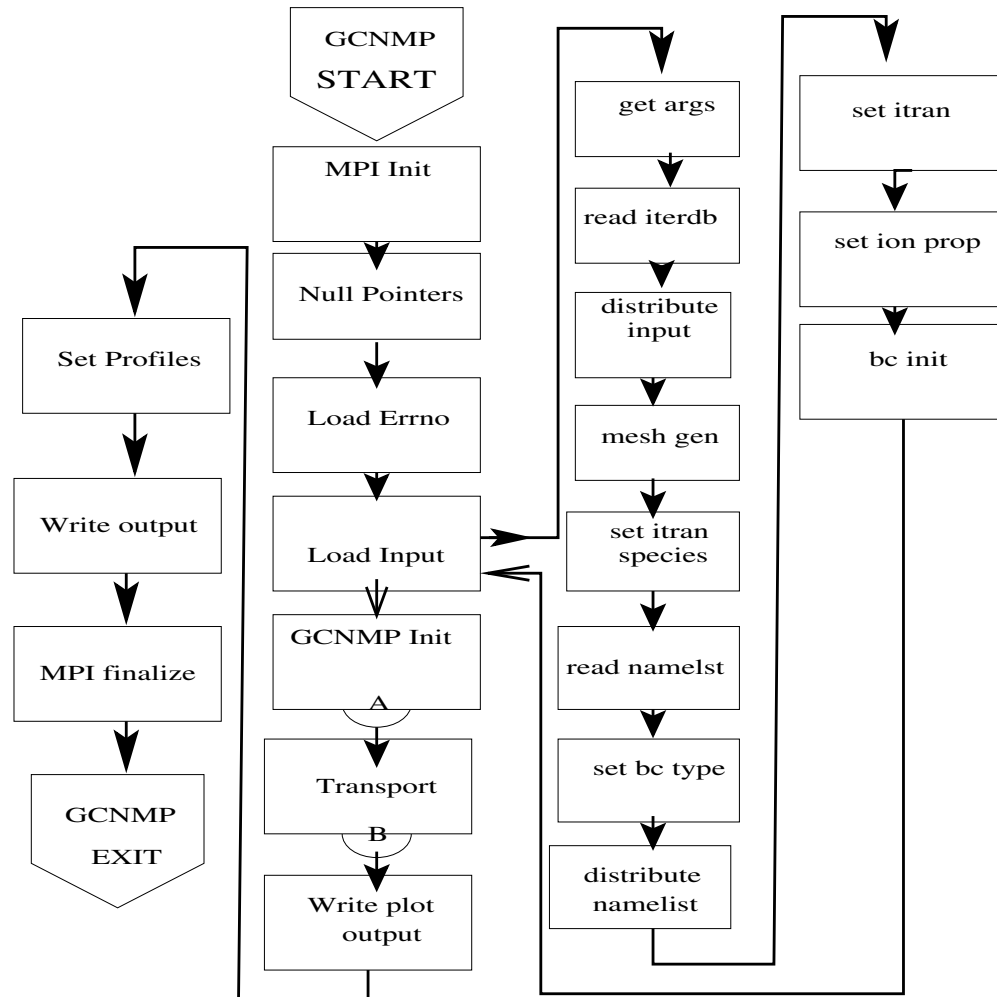


GCNMP FDF Simulations For Peaked Density Profiles

	w9/a	w13/a	w14/a	w11
ECH(170Ghz,MW)	13/14	0	0	30+1
BEAM MW	30	30	15	20
H89p	3.0/3.4	3.0/3.3	3.3/3.7	2.7
H98y2	1.5/1.7	1.4/1.7	1.5/1.8	1.5
β_T	8.1/9.3	6.8/8.9	7.1/9.6	6.3
I_{ech} Ka	180/195	0	0	580
I_{ohm}	149/-556	1290/-254	1550/-532	1020
I_{beam}	477/440	474/443	273/263	365
I_{boot}	5910/6830	4950/6520	4890/6980	4750
Q_{DT}	9.3/16	10/15	22/32	4.4
P_{fus} MW	400/481	301/448	329/501	223
P_{rad} MW	14/14	13/14	13/15	7
W Mj	77/88	65/84	68/91	60
$q(0)$	29./1.9	1.9/1.9	1.5/1.9	2.52



Flow Charts: Overall



Plans for 2007-8

- GCNMP Improvements:
 - Continue development of transport equations based on particle, energy and momentum flows rather than the standard diffusion/convection equations. GCNMP is ideal vehicle for this (eq. no change in solver due to change in equations (parabolic to hyperbolic, etc.)
 - The current version of GCNMP allows arbitrary grids and we will extend this capability to include time dependent dynamic allocation of grids in response to internal and external triggering mechanism such as saw teeth, ELM and local beam and rf deposition.
 - Multi level parallelization. Requires inter and intra communicator structure: Jacobian breaks up into sets of independent columns due to structure of finite difference equations. This defines independent communicator groups. Currently we only use one group.
 - Introduce parallel/sparse/iterative versions of required linear algebra routines. (if number of equations rises above about 800)

30:



Plans for 2007-8...

- PTRANSP Enhancements and Validation.
 - Create and Validate PTRANSP/GCNMP coupling. Apply to DIII-D experimental data analysis for.
 - Incorporation of TGLF into PTRANSP/GCNMP and its subsequent verification using DIII-D data. This will allow for shaped magnetic geometry, strong negative magnetic shear and inclusion of the edge pedestal region in transport modeling. (It is expected that due to the additional computational burden of TGLF that a significant effort is required in this area - TGLF coupling can only occur only after the TGLF module is released).



One dimensional transport module must solve

Electron Energy

$$\frac{3}{2} \left(T_e \sum_{i=1}^{nion} \left(n_i \frac{\partial Z_i}{\partial T_e} \Big|_{\zeta} \right) + n_e \right) \frac{\partial T_e}{\partial t} + \frac{3}{2} T_e \sum_{i=1}^{nion} Z_i \frac{\partial n_i}{\partial t} \Big|_{\zeta} + \frac{1}{H\rho} \frac{\partial}{\partial \rho} \left(H\rho \left(q_e + \frac{5}{2} \Gamma_e T_e \right) \right) = Q_e + S_{T_e}^{2D} \quad (19)$$

Ion Energy Equation

$$\sum_{i=1}^{nion} \left\{ \frac{3}{2} n_i \frac{\partial T}{\partial t} \Big|_{\zeta} + \frac{\partial n_i}{\partial t} \Big|_{\zeta} \left(\frac{3}{2} T + \frac{1}{2} m_i \omega^2 \langle R^2 \rangle \right) \right\} + \sum_{i=1}^{nion} m_i n_i \omega \langle R^2 \rangle \frac{\partial \omega}{\partial t} \Big|_{\zeta} + \frac{1}{H\rho} \frac{\partial}{\partial \rho} \left\{ H\rho \left(\sum_{i=1}^{nion} \left(q_i + \frac{5}{2} \Gamma_i T \right) + \Gamma_T^\omega + \Pi\omega \right) \right\} = Q_i + S_T^{2D} \quad (20)$$



1-D transport module ...

Faraday's Law

$$\begin{aligned}
 & \frac{1}{FG(H\rho)^2\alpha} \frac{\partial(FGH\rho B_{p0})}{\partial t} - \frac{1}{H\rho} \frac{\partial}{\partial \rho} \left(H\rho \left(d_{4,1} \frac{\partial n_i}{\partial \rho} + d_{4,2} \frac{\partial T_e}{\partial \rho} + d_{4,3} \frac{\partial T}{\partial \rho} \right. \right. \\
 & \left. \left. + d_{4,4} \frac{\partial FGH\rho B_{p0}}{\partial \rho} \right) \right) - \frac{1}{H\rho} \frac{\partial}{\partial \rho} \left(\frac{\partial \rho}{\partial t} \Big|_{\zeta} B_{p0} \right) = - \frac{1}{H\rho} \frac{\partial}{\partial \rho} \left(\eta_{\parallel} cH \left\langle J_{aux}^{\vec{}} \cdot \frac{\vec{B}}{B_{t0}} \right\rangle \right) \\
 & + \frac{1}{H\rho} \frac{\partial}{\partial \rho} \left(H\rho (D_f^e + D_f^b) \frac{\partial n_f}{\partial \rho} \right) + \frac{B_{p0}}{H\rho} \frac{\partial}{\partial t} \left(\ln FGH\rho \right) - \frac{B_{p0}}{H\rho} \frac{\partial}{\partial \rho} \left(\frac{\partial \rho}{\partial t} \Big|_{\zeta} \right)
 \end{aligned} \tag{21}$$

Primary and impurity ions (analysis mode only)

$$\frac{\partial n_i}{\partial t} \Big|_{\zeta} + \frac{1}{H\rho} \frac{\partial}{\partial \rho} (H\rho \Gamma_i) = S_i + S_i^{2D} \tag{22}$$

1-D transport module ...

Toroidal Momentum Equation

$$\sum_{i=1}^{nprim} m_i n_i \langle R^2 \rangle \frac{\partial \omega}{\partial t} \Big|_{\zeta} + \omega \sum_{i=1}^{nprim} m_i \langle R^2 \rangle \frac{\partial n_i}{\partial t} \Big|_{\zeta} + \frac{1}{H\rho} \frac{\partial}{\partial \rho} (H\rho\Gamma\omega) = S_{\omega} + S_{\omega}^{2D} \quad (23)$$

Rho grid motion effects are represented by red terms

Above equations are solved by :

- Crank- Nicholson predictor - corrector
- Method of lines
- GCNMP solver



Numerical considerations ...

- The upwind difference effect is conveniently achieved by defining matrices which contain the positive and negative diagonal parts of \mathbf{V}

$$\underline{\underline{V}}_{j+\frac{1}{2}} u_{j+\frac{1}{2}} = \underline{\underline{V}}_{j+\frac{1}{2}}^+ u_j + \underline{\underline{V}}_{j+\frac{1}{2}}^- u_{j+1} \quad (24)$$

- Putting all the pieces together we get

$$\begin{aligned} \underline{\underline{M}}_j \frac{\partial u_j}{\partial t} - \frac{1}{(H\rho)_j \Delta\rho_j} \left(\frac{(H\rho)_{j-\frac{1}{2}} \underline{\underline{D}}_{j-\frac{1}{2}}}{\Delta\rho_{j-\frac{1}{2}}} + (H\rho)_{j-\frac{1}{2}} \underline{\underline{V}}_{j-\frac{1}{2}}^+ \right) u_{j-1} \\ - \frac{1}{(H\rho)_j \Delta\rho_j} \left(-\frac{(H\rho)_{j+\frac{1}{2}} \underline{\underline{D}}_{j+\frac{1}{2}}}{\Delta\rho_{j+\frac{1}{2}}} - \frac{(H\rho)_{j-\frac{1}{2}} \underline{\underline{D}}_{j-\frac{1}{2}}}{\Delta\rho_{j-\frac{1}{2}}} - (H\rho)_{j+\frac{1}{2}} \underline{\underline{V}}_{j+\frac{1}{2}}^+ + (H\rho)_{j-\frac{1}{2}} \underline{\underline{V}}_{j-\frac{1}{2}}^- - (H\rho)_j \underline{\underline{W}}_j \Delta\rho_j \right) u_j \\ - \frac{1}{(H\rho)_j \Delta\rho_j} \left(\frac{(H\rho)_{j+\frac{1}{2}} \underline{\underline{D}}_{j+\frac{1}{2}}}{\Delta\rho_{j+\frac{1}{2}}} - (H\rho)_{j+\frac{1}{2}} \underline{\underline{V}}_{j+\frac{1}{2}}^- \right) u_{j+1} = \underline{\underline{S}}_{exp,j} \quad (25) \end{aligned}$$

