TRANSP/PTRANSP Isolver Free Boundary Equilibrium Solver

Robert Andre, PPPL

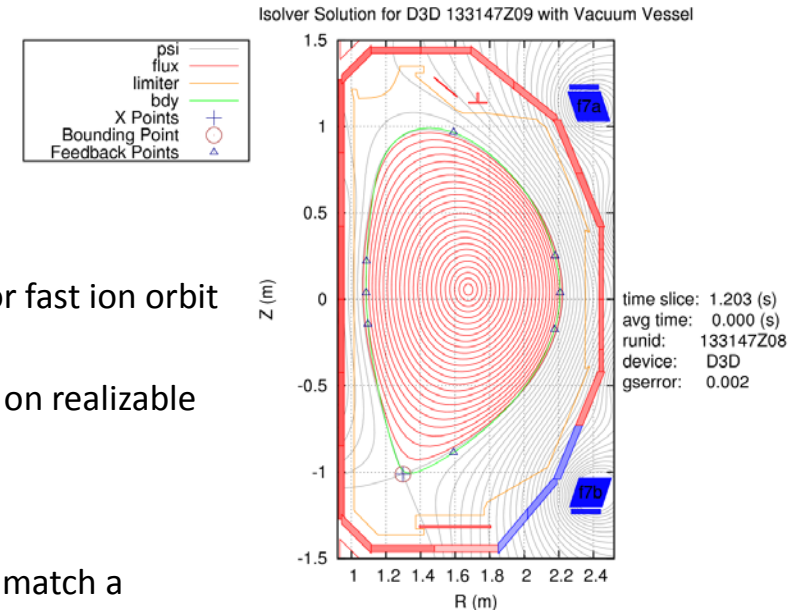
Presented at APS-DPP 2012

October 30, 2012

Supported by U.S. DOE Contract No. DE-AC02-09CH11466

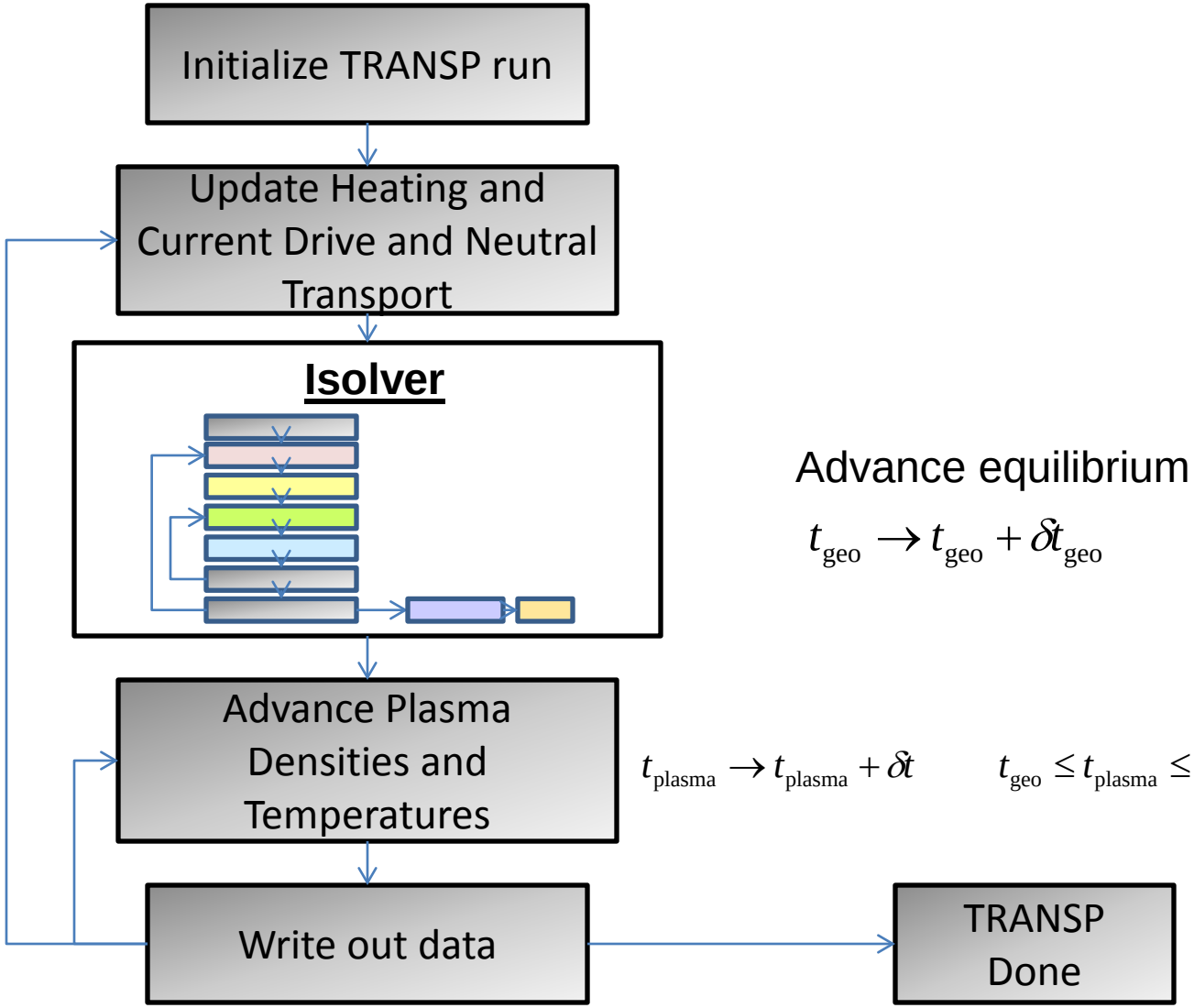
TRANSP/PTRANSP Isolver

- TRANSP Isolver is a free boundary equilibrium solver available in TRANSP in both the interpretive and the fully predictive mode (PTRANSP)
- Features:
  - **Free Boundary**
    - Magnetic field representation outside plasma is useful for fast ion orbit and RF heating codes
    - Boundary may adjust to heating and current drive based on realizable PF coil currents
  - **Coil Currents**
    - *Least Squares Mode* – coil currents selected to best match a prescribed boundary (traditional TRANSP mode)
    - *Measured Coil Currents Mode* – feedback applied to center the plasma at the prescribed boundary (analysis and predictive mode)
  - **Passive Structures** – induced currents in vacuum vessel
    - Affects plasma evolution and shape particularly during current ramp up
  - **Coupled Flux Diffusion** – time evolution of q profile coupled to equilibrium solution.
    - Self consistent solution of coupled plasma and coil circuit equations for predictive modeling
- Supported Tokamaks:
  - Coils and Vacuum Vessel: NSTX (current), D3D
  - Coils only: NSTX (upgrade), ITER, CMOD, EAST, IGTR, MAST



TRANSP/PTRANSP Execution

- Isolver is called during the TRANSP run to update the equilibrium solution



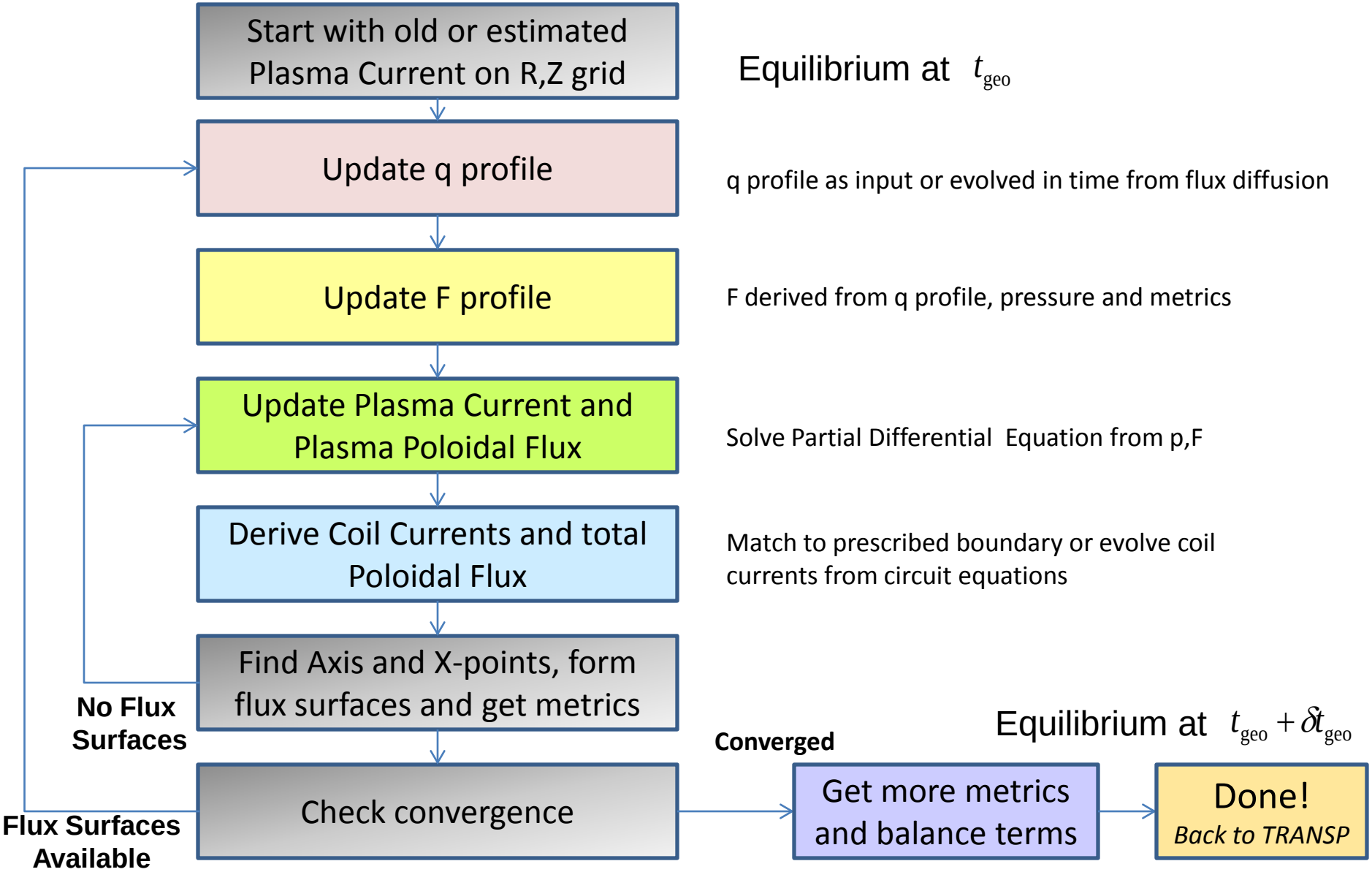
Advance equilibrium solution

$$t_{\text{geo}} \rightarrow t_{\text{geo}} + \delta t_{\text{geo}}$$

$$t_{\text{plasma}} \rightarrow t_{\text{plasma}} + \delta t \qquad t_{\text{geo}} \leq t_{\text{plasma}} \leq t_{\text{geo}} + \delta t_{geo}$$

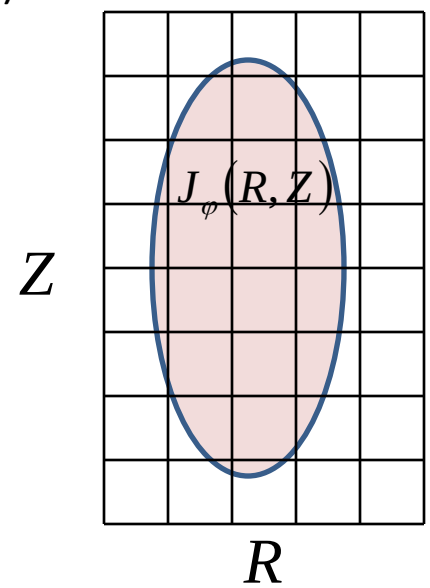
Algorithm: Isolver's Picard Loop

- Picard loop incrementally adjusts the equilibrium solution at each new geometry time point until converged



Algorithm: Update Plasma Current

- Isolver's solution is based **entirely** on the plasma current $J_\varphi(R,Z)$ defined on a finite difference R,Z grid (typically 65x129 or 129x129)



- As the solution iterates, the plasma current is updated from the most recent pressure and F profiles.
  - Note that this does not require a flux surface representation which enables robust start up.

$$J_\varphi^{p,F}(R,Z) = \frac{R}{\Delta\psi}\frac{dp}{d\overline{\psi}} + \frac{F}{\mu_0 R\Delta\psi}\frac{dF}{d\overline{\psi}}$$

$$F = RB_\varphi \qquad \overline{\psi} = \frac{\psi}{\Delta\psi} \quad \text{normalized poloidal flux}$$

- New plasma current is relaxed with the old solution

$$J_\varphi^{\text{new}}(R,Z) = \omega J_\varphi^{p,F}(R,Z) + (1-\omega) J_\varphi^{\text{old}}(R,Z)$$

Algorithm: Update Plasma Poloidal Flux

- A fast PDE solver is used to derive the poloidal flux due to the plasma $\psi^{\text{plasma}}(R,Z)$ from the plasma current $J_{\varphi}^{\text{new}}(R,Z)$ defined on the finite difference R,Z grid.

$$\Delta^{*}\psi^{\text{plasma}} \equiv R\frac{\partial}{\partial R}\left(\frac{1}{R}\frac{\partial\psi^{\text{plasma}}}{\partial R}\right)+\frac{\partial^{2}\psi^{\text{plasma}}}{\partial Z^{2}}=-\mu_{0}RJ_{\varphi}^{\text{new}}(R,Z)$$

- The total poloidal flux also includes the flux from the coils and passive structures (represented as coils)

$$\psi(R,Z)=\psi^{\text{plasma}}(R,Z)+\psi^{\text{coils}}(R,Z)$$

Algorithm: Update F from q profile
General Principles

- The F profile is required to satisfy

$$FV'\left\langle \frac{1}{R^2} \right\rangle = 4\pi^2 q \Delta\psi$$

$$V' = \frac{dV}{d\overline{\psi}} \quad \text{differential volume}$$

$$\Delta\psi = \text{poloidal flux enclosed by the plasma}$$

- The **F profile must be smooth** in order to be representable by $J_\varphi(R,Z)$ on the R,Z grid
- For a diverted plasma, **q and V' are singular at the edge**.
- So to provide asymptotic consistency at the edge when the q profile is provided as input or from flux diffusion, V' **was considered a derived quantity** and not a geometric metric when deriving the equilibrium equations.

Algorithm: Update F from q profile
Solving for F

- The F profile is derived from the solution of the ordinary differential equation (ODE),

$$\frac{d\left(F^2\right)}{d\overline{\psi}} = -2\left[\frac{\mu_0}{\left\langle \frac{1}{R^2} \right\rangle}\frac{dp}{d\overline{\psi}} + \frac{\Delta\psi F}{4\pi^2}\left(\frac{\alpha'}{q}\right)\right]$$

$$\alpha \equiv V'\left\langle \frac{\left|\nabla\overline{\psi}\right|^2}{R^2} \right\rangle = \frac{-2\pi\mu_0 I_\varphi(\overline{\psi})}{\Delta\psi}$$

$$p(\overline{\psi}) = \text{pressure} \quad I_\varphi(\overline{\psi}) = \text{enclosed plasma current}$$

- When substituted into the Grad-Shafranov Equation

$$\Delta^*\psi = \frac{-\mu_0}{\Delta\psi}\left(R^2 - \frac{1}{\left\langle \frac{1}{R^2} \right\rangle}\right)\frac{dp}{d\overline{\psi}} + \frac{F}{4\pi^2 q}\alpha'$$

- A flux surface average recovers the required relation $FV'\left\langle \frac{1}{R^2} \right\rangle = 4\pi^2 q\Delta\psi$ when α' is nonzero.

Algorithm: Update F from q profile
Integral of q form

- The F profile ODE can be manipulated to <u>remove the singularity of the q</u> profile in the equation

$$\left(\frac{\alpha'}{q}\right) = \frac{\frac{d\alpha}{d\bar{\phi}}}{\phi(1)}$$

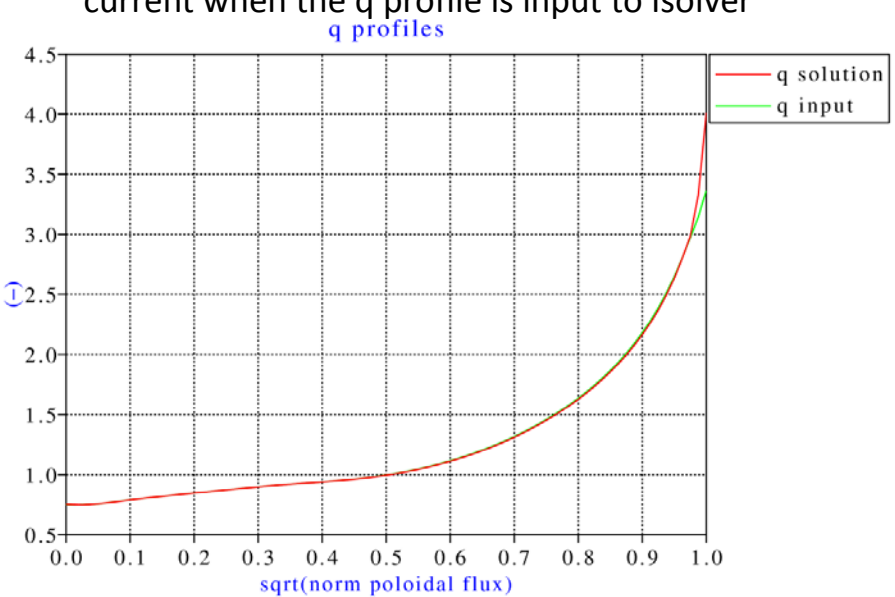
$$\phi(\bar{\psi}) \equiv \int_0^{\bar{\psi}} q(\hat{\bar{\psi}})d\hat{\bar{\psi}} = \frac{\Delta\Phi(\bar{\psi})}{2\pi\Delta\psi}$$

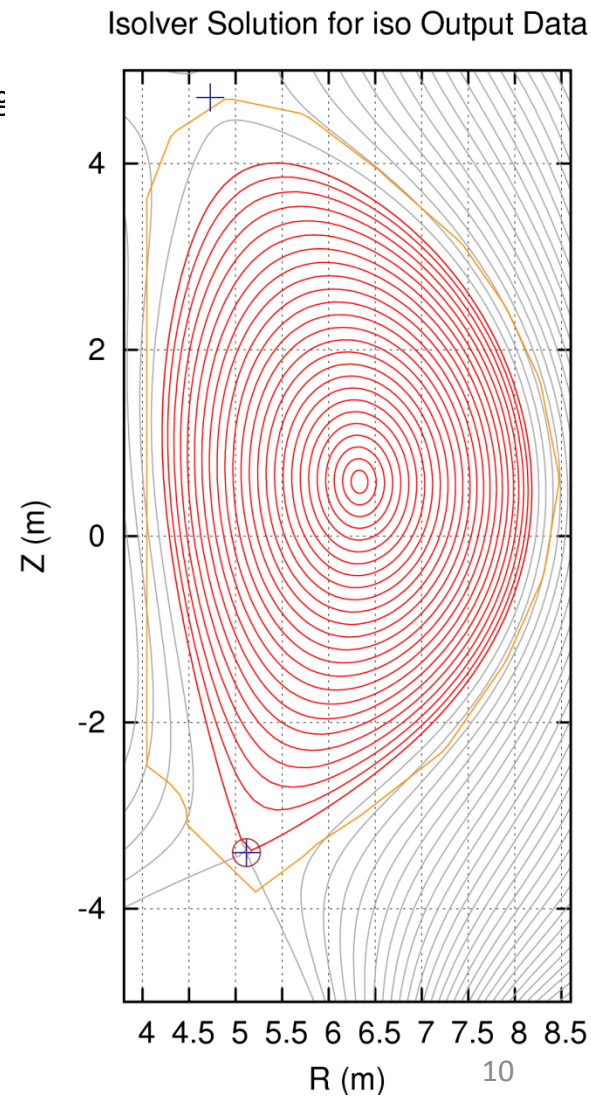
$$\alpha(\bar{\phi}) \equiv \alpha(\left[\bar{\phi}(\bar{\psi})\right]^{-1}) \quad \text{using the function inverse of } \bar{\phi} = \frac{\phi}{\phi(1)}$$

- In this form, the solution of the Grad-Shafranov equation depends <u>only</u> on the integral of the q profile $\phi(\bar{\psi})$ which represents the toroidal flux at each flux surface per total enclosed poloidal flux on a normalized poloidal flux grid.

Algorithm: Update F from q profile
Characteristic of Solution

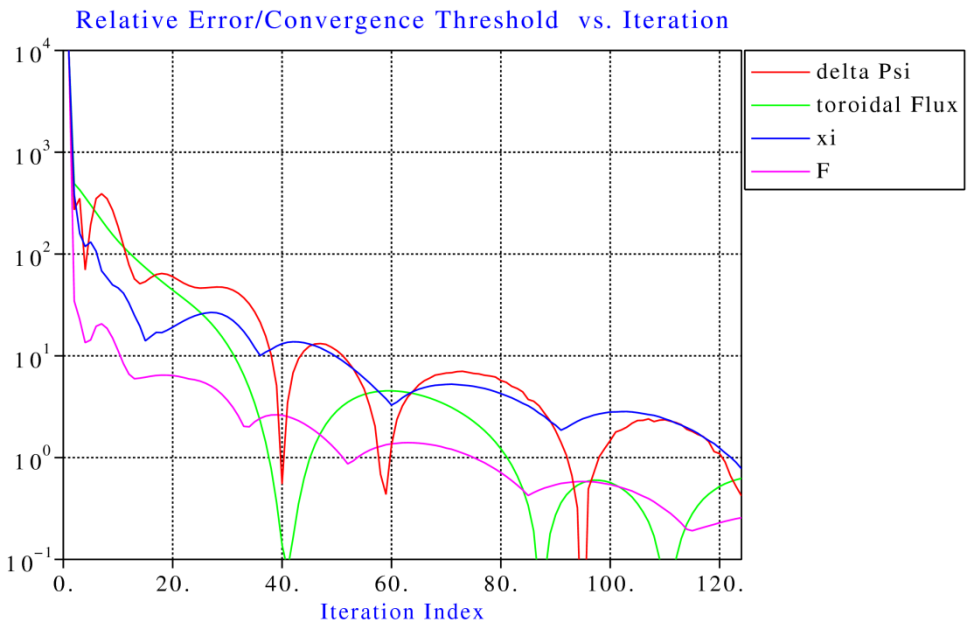
- Smoothing of α is required
  - A smoothing half-width of 1 radial zone is sufficient though 2 radial zones is used in practice – better convergence with more smoothing
  - The number of radial zones is arbitrary
- Solution can be taken to the separatrix
  - TRANSP requires the boundary be offset from the separatrix due to metric singularities
- Q profile can deviate from input near separatrix
  - Reflects disconnect between q profile singularity and V' singularity
  - Typically, the edge q profile is perturbed to match desired plasma current when the q profile is input to Isolver

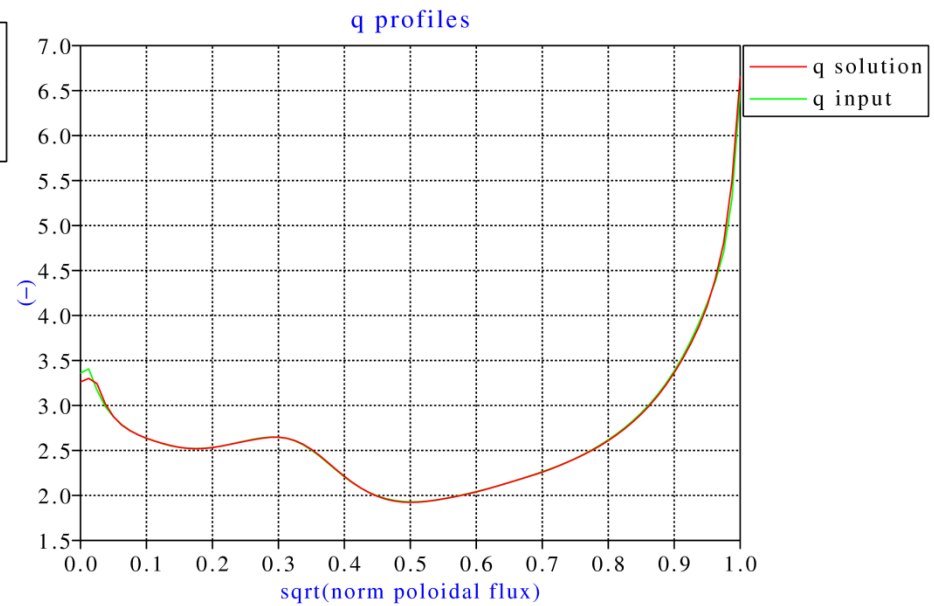




Algorithm: Update F from q profile *Characteristic of solution cont.*

- Convergence is robust but <u>slow</u>
  - Convergence can slow down as final converged solution is approached (ugh!)





$$\Delta\psi \text{ error threshold} \equiv \frac{\left|\Delta\psi_{k+1} - \Delta\psi_k\right|}{\varepsilon\Delta\psi_k} \text{ considered converged when} \le 1$$

$$\varepsilon = 10^{-6} \quad \text{typically}$$

$$\text{similarly for } \Phi_{\text{bdy}}, \xi(\overline{\psi}) = \sqrt{\frac{\Phi(\overline{\psi})}{\Phi_{\text{bdy}}}}, F(\overline{\psi})$$

Algorithm: Update F from q profile
Speeding up Convergence

- The normalized poloidal flux coordinate used during the picard iteration $\tilde{\psi}$ is only an approximation to $\overline{\psi}$
- The relationship between $\tilde{\psi}$ and ψ is given by,

$$h(\tilde{\psi}) \equiv \frac{d\psi}{d\tilde{\psi}} = V'\left\langle \frac{1}{R^2} \right\rangle \frac{\Delta\psi}{\delta\psi} \frac{F}{4\pi^2 q}$$

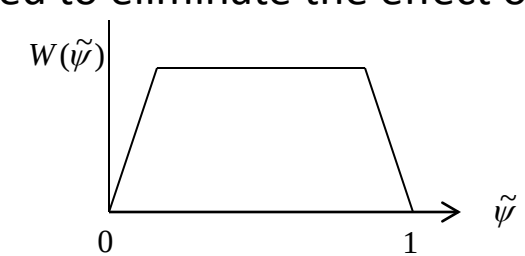
$$\delta\psi \equiv \int_0^1 h(\tilde{\psi}) d\tilde{\psi}$$ this definition removes the scale dependence of h

- This can be used to refine the solution during the picard iteration

$$\frac{d\left(F^2\right)}{d\tilde{\psi}} = -2\left[\frac{h}{\delta\psi} \frac{\mu_0}{\left\langle \frac{1}{R^2} \right\rangle} \frac{dp}{d\overline{\psi}} + \frac{\Delta\psi F}{4\pi^2} \left(\frac{h}{\delta\psi} \frac{\alpha'}{q} + \frac{\alpha}{q} \frac{h'}{\delta\psi} \right) \right]$$

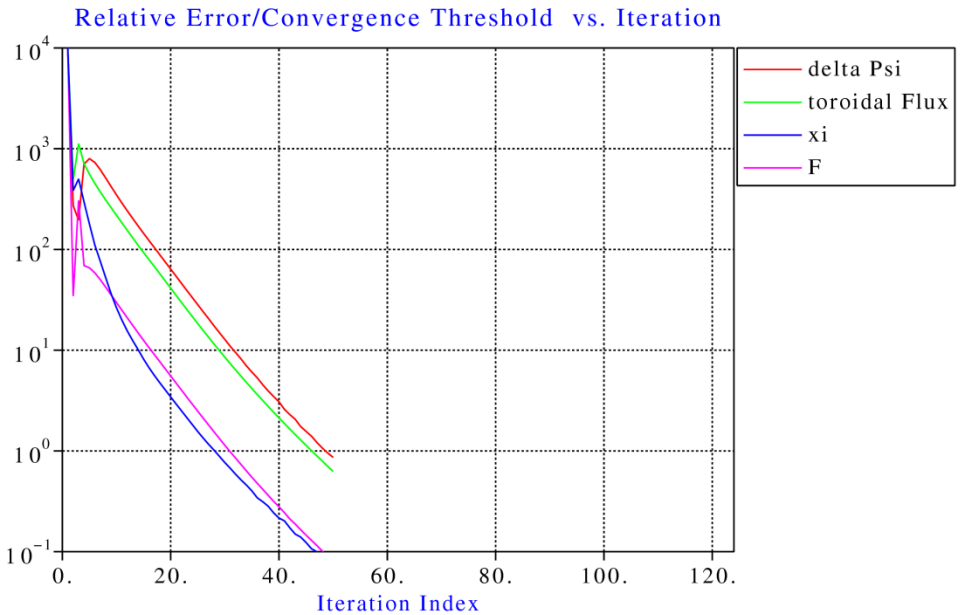
- Since $h(\tilde{\psi}) \to \Delta\psi$ a constant in the limit of the solution, $h(\tilde{\psi})$ can be windowed away from the axis and edge and heavily smoothed to eliminate the effect of taking the derivative of the q profile.

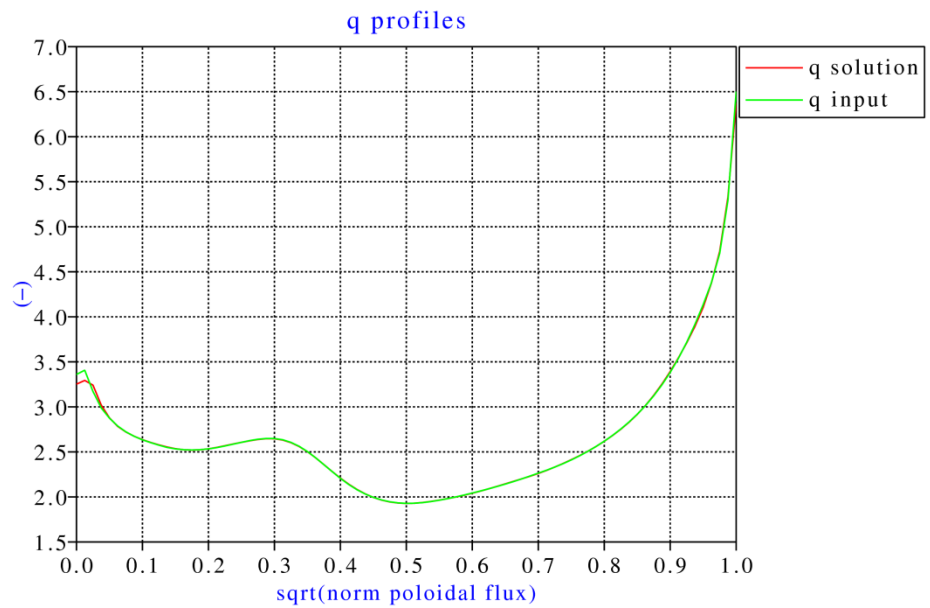
redefine $$h'(\tilde{\psi}) \equiv \frac{d}{d\tilde{\psi}} \left(V'\left\langle \frac{1}{R^2} \right\rangle \frac{F}{4\pi^2 q} \right) W(\tilde{\psi})$$



Algorithm: Update F from q profile
Speeding up Convergence cont.

- The effect of this algorithm change is significantly improved convergence
  - Slightly less robust though there are cases where the modified solution can reach convergence whereas the original will not
  - Solution is a little tighter in matching the q profile





Algorithm: Update q profile
Flux Diffusion

- The poloidal flux evolves in time to satisfy,

$$\left.\frac{\partial \psi}{\partial t}\right|_{\Phi} = -\frac{V_{\text{loop}}}{2\pi}$$

$$\frac{V_{\text{loop}}}{2\pi} = \eta_{\parallel} J^{\text{oh}} \qquad J^{\text{oh}} \equiv \frac{\left\langle \vec{J}\cdot\vec{B}\right\rangle}{\left\langle \vec{B}\cdot\nabla\varphi\right\rangle} - J^{\text{drive}} \qquad J^{\text{drive}} \equiv \frac{\left\langle \vec{J}^{\text{drive}}\cdot\vec{B}\right\rangle}{\left\langle \vec{B}\cdot\nabla\varphi\right\rangle}$$

- Where $\eta_{\parallel}$ is the plasma resistivity and $\left\langle \vec{J}^{\text{drive}}\cdot\vec{B}\right\rangle$ represents driven currents.
- When expressed as an evolution equation of the integral of the q profile $\phi(\overline{\psi},t)$ as used in the F profile ODE on a normalized poloidal flux grid, two equations must be satisfied.
  - Boundary condition at the edge for $\phi(\overline{\psi},t)$ equation provides one degree of freedom in the solution
    - Will be used to match the plasma current, surface voltage or axial poloidal flux condition
  - Boundary condition at axis is fixed to enforce regularity

$$\left.\frac{\partial \phi}{\partial t}\right|_{\overline{\psi}} = \frac{\eta_{\parallel} q}{\Delta\psi} J^{\text{oh}} + \frac{1}{\Delta\psi}\frac{d(\Delta\psi)}{dt}\left(\overline{\psi}q - \phi\right) + \frac{q}{\Delta\psi}\frac{d\psi_0^{\text{diff}}}{\text{dt}}$$

$$\frac{d\psi_0^{\text{diff}}}{\text{dt}} = -\eta_{\parallel} J^{\text{oh}}\Big|_{\overline{\psi}=0} \qquad \psi_0^{\text{diff}} \equiv \text{poloidal flux at axis}$$

Algorithm: Update q profile
Full *Flux Diffusion Equation*

- Substituting for J^{oh} gives the diffusion equation in $\phi(\overline{\psi},t)$

$$\left.\frac{\partial\phi}{\partial t}\right|_{\overline{\psi}} = \frac{\eta_{\|}\alpha}{\nu\mu_0}\left[1+\frac{\alpha(\Delta\psi)^2}{\nu F^2}\right]\frac{\partial^2\phi}{\partial\overline{\psi}^2} - \frac{\eta_{\|}}{4\pi^2\Delta\psi}\left[\frac{\alpha}{F\left\langle\frac{1}{R^2}\right\rangle}\frac{\partial p}{\partial\overline{\psi}} + \frac{F}{\mu_0}\left(1+\frac{\alpha(\Delta\psi)^2}{\nu F^2}\right)\frac{\partial\alpha_\phi}{\partial\overline{\phi}}\right]$$

$$+\frac{1}{\Delta\psi}\left[\frac{d\psi_0^{\mathrm{diff}}}{\mathrm{dt}} + \overline{\psi}\frac{d(\Delta\psi)}{dt} - \eta_{\|}J^{\mathrm{drive}}\right]\frac{\partial\phi}{\partial\overline{\psi}} - \frac{1}{\Delta\psi}\frac{d(\Delta\psi)}{dt}\phi$$

$$\frac{d\psi_0^{\mathrm{diff}}}{\mathrm{dt}} = \eta_{\|}\left.\left(\frac{\Delta\psi R_0^2}{\mu_0\frac{\partial V}{\partial\overline{\psi}}}\frac{\partial\alpha}{\partial\overline{\psi}} + J^{\mathrm{drive}}\right)\right|_{\overline{\psi}=0}$$

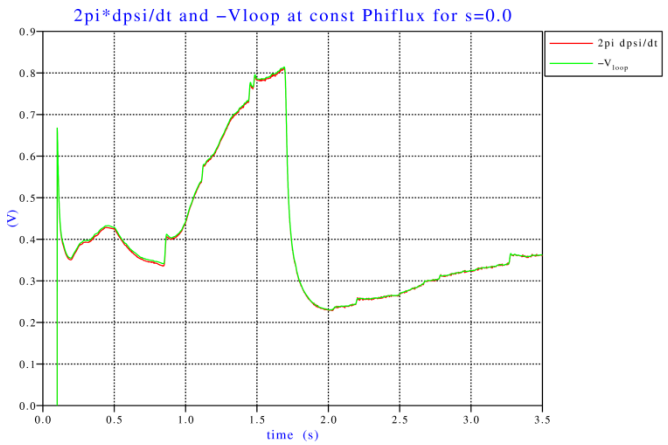
$$\alpha_\phi \equiv \frac{\partial V}{\partial\overline{\phi}}\left\langle\frac{\left|\nabla\overline{\phi}\right|^2}{R^2}\right\rangle = \frac{2\pi\Delta\psi}{\Delta\Phi}\alpha q$$

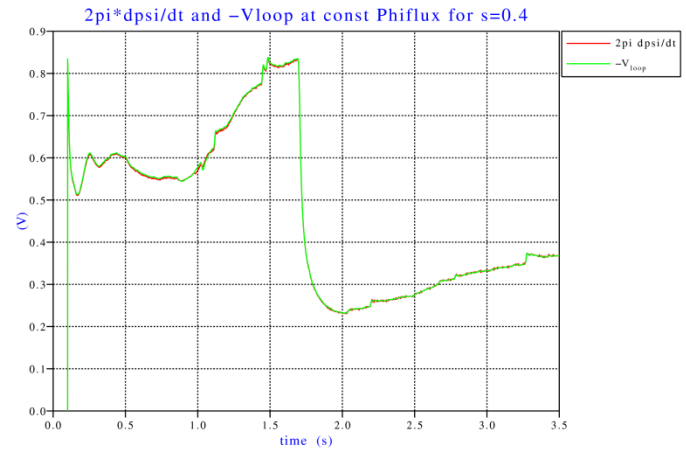
q here is from flux surface metrics

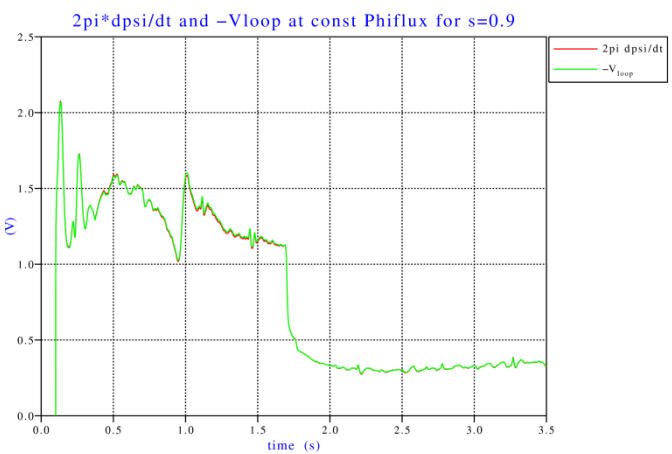
$$\nu \equiv \frac{\partial V}{\partial\overline{\psi}}\left\langle\frac{1}{R^2}\right\rangle$$

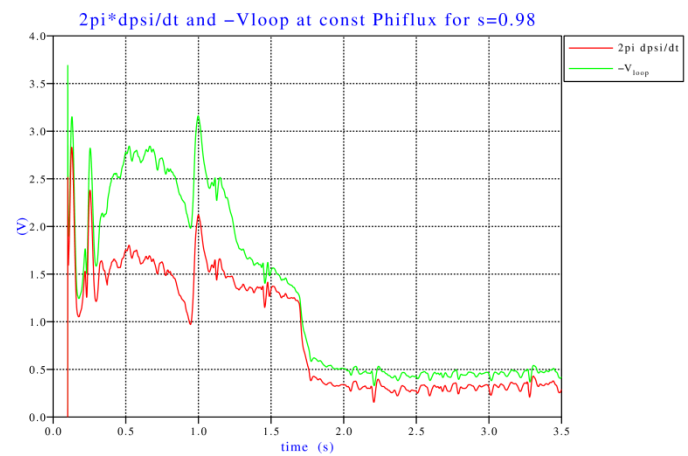
Algorithm: Update q profile
Flux Diffusion Sanity Check

- Check of $2\pi \left.\frac{\partial \psi}{\partial t}\right|_{\Phi}(s,t) = -V_{\text{loop}}(s,t)$ over time t for fixed $s \equiv \sqrt{\overline{\psi}}$ for a D3D run
  - Poor match at edge due to error in q profile match as amplified by shape of resistivity profile



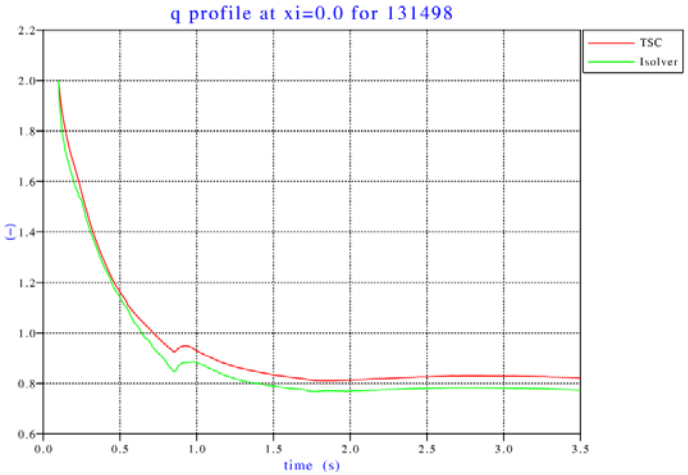


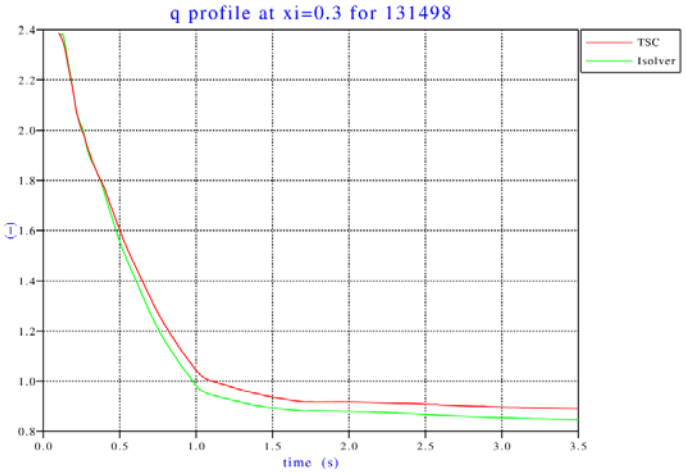


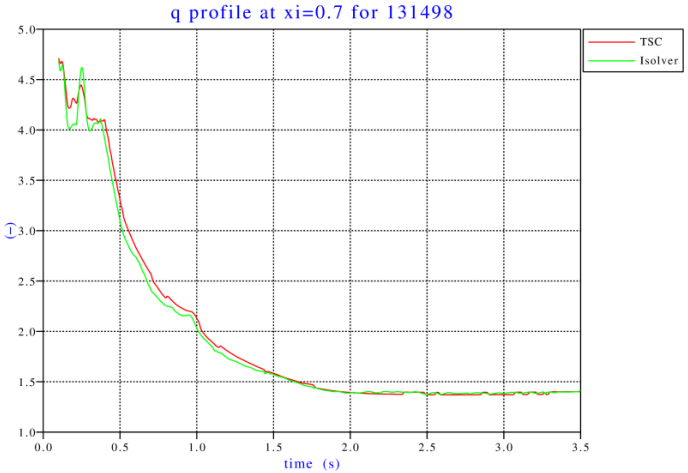


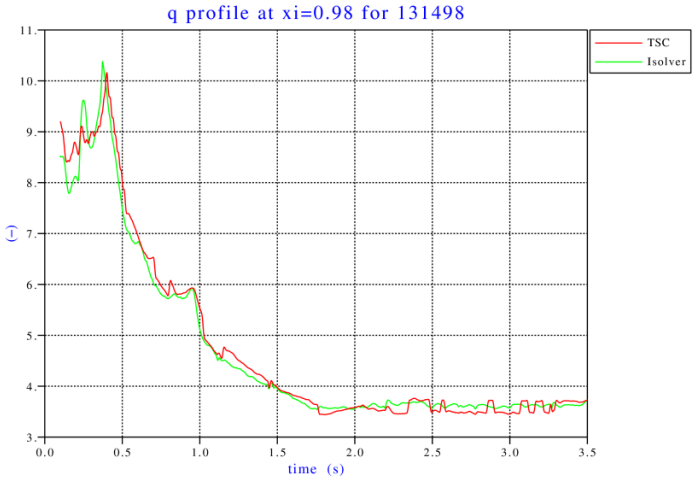
Algorithm: Update q profile
Flux Diffusion Comparison with TSC

- Comparison of radial slices of the q profile from Isolver with TSC for D3D run
  - Isolver run outside of TRANSP in plasma current prediction mode using TSC coil currents and pressure and resistivity profiles
  - Isolver driven current (bootstrap only) estimated from TRANSP run should be similar to TSC driven current









Algorithm: Update q profile
Flux Diffusion Mixed Boundary Condition

- The flux diffusion equation has the form

$$\frac{\partial \phi}{\partial t} = \sigma \frac{\partial^2 \phi}{\partial \overline{\psi}^2} - \mathrm{v} \frac{\partial \phi}{\partial \overline{\psi}} - \mathrm{c}\phi$$

- Where c and v change significantly within the picard loop. The solution of this equation can be written in terms of a function $\Theta(y,t)$ which does not change very much during the loop.

$$\phi(\overline{\psi},t) = \Theta(\overline{\psi} - \mathrm{v}t, t)e^{-\mathrm{c}t}$$

$$\phi(\overline{\psi},t+\delta t) = (1-\mathrm{c}\delta t)\Theta e^{-\mathrm{c}t} - \mathrm{v}\delta t \frac{\partial \Theta}{\partial y} e^{-\mathrm{c}t} + O(\delta t^2) \qquad \text{for } \frac{\partial \Theta}{\partial t} << \mathrm{c}, \mathrm{v} \text{ terms}$$

- Applying a boundary condition on $\Theta(y,t)$ during a finite difference time step δt is equivalent to using the following mixed boundary condition on $\phi(\overline{\psi},t)$

$$q_{\mathrm{mix}} \equiv (1+\mathrm{c}\delta t)\phi\Big|_{\overline{\psi}=1} + \mathrm{v}\delta t \frac{\partial \phi}{\partial \overline{\psi}}\Bigg|_{\overline{\psi}=1}$$

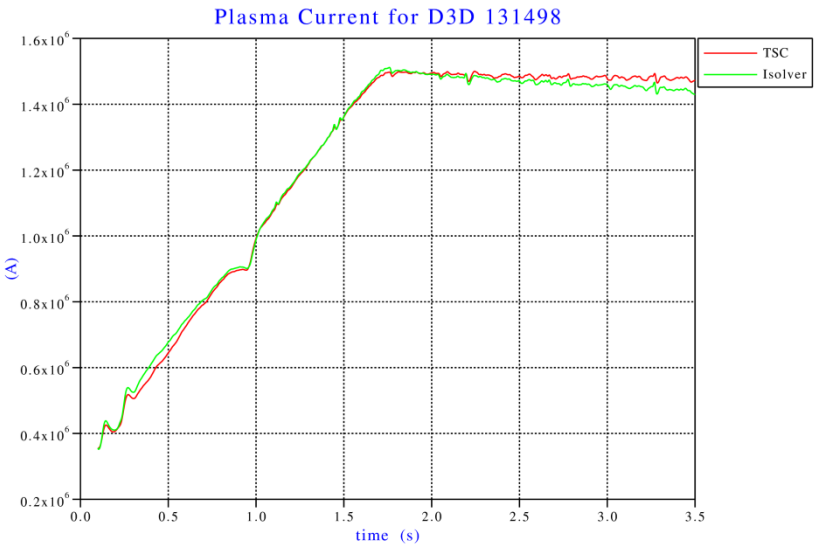
- In the case of the flux diffusion equation $\mathrm{c}, -\mathrm{v} = \frac{1}{\Delta\psi}\frac{d(\Delta\psi)}{dt}$ is used to remove the effect of the changing poloidal flux on the diffusion equation during the picard loop.

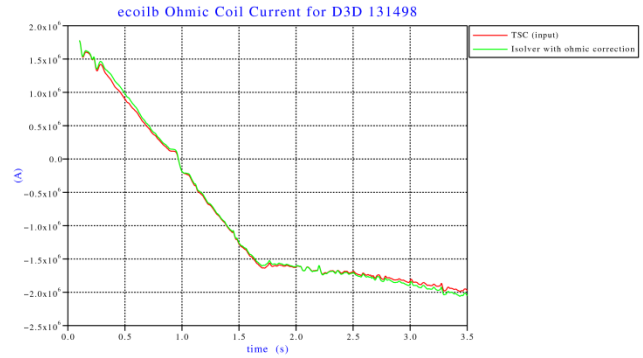
Algorithm: Update q profile
Flux Diffusion Boundary Condition and ψ_0^{diff}.

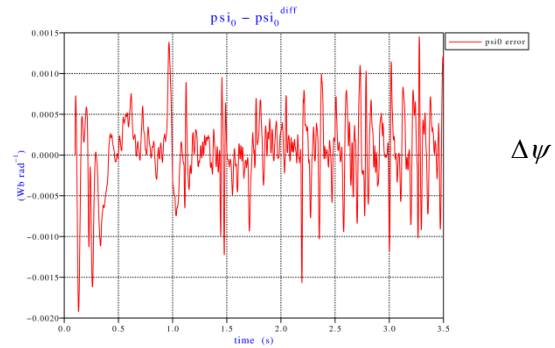
- ψ_0^{diff} represents the poloidal flux at the axis which is required to satisfy the flux diffusion requirements -- physically, it should match the axial poloidal flux due to the plasma current and coils ψ_0
- But there is no a-priori condition to force $\psi_0^{\mathrm{diff}} = \psi_0$ from the equations alone
- There are several options in Isolver for resolving this difference and setting the boundary condition q_{mix} on the flux diffusion equation:
  1. Adjust q_{mix} to match a desired plasma current (NLPCUR=.TRUE.) or surface voltage (NLVSUR=.TRUE.)
     a. **Traditional TRANSP**: Let ψ_0^{diff} and ψ_0 differ. This implies the existence of an ohmic coil which makes up the flux difference but does not affect the shape.
     b. **Ohmic Coil Current Prediction**: Apply a feedback current to the ohmic coils to match ψ_0^{diff} and ψ_0
        - Implemented with a PID (proportional –integral-differential) controller between time steps
  2. Adjust q_{mix} to force $\psi_0^{\mathrm{diff}} = \psi_0$ (NLINDUCT=.TRUE.)
     a. **Plasma Current Prediction**: Inductive coupling between the poloidal field coils and the plasma predicts the plasma current
- A PI (proportional-integral) controller is used to adjust q_{mix} for the desired boundary condition within the picard loop.
  - A loop voltage condition will match $q_{\mathrm{mix}} = \phi^- + \frac{q}{\Delta\psi}\left(\frac{\delta t V_{\mathrm{loop}}}{2\pi} + (\psi_0^+ - \psi_0^-)\right)$ for $\delta t = t^+ - t^-$

Algorithm: Update q profile
Example Boundary Condition and ψ_0^{diff}

- Comparison of TSC with Isolver run in predictive mode outside of TRANSP







$\Delta\psi \approx 0.3\,(\text{Wb/rad})$

- **Plasma Current Prediction**
  - Ohmic coil induces current in plasma
  - $\psi_0^{\text{diff}} = \psi_0$ matched exactly by flux diffusion boundary condition

- **Ohmic Coil Current Prediction**
  - Plasma current is used as flux diffusion boundary condition
  - Ohmic coil current feedback PID controller keeps $\psi_0^{\text{diff}} \approx \psi_0$ over time.

Algorithm: Poloidal Field Balance
Poloidal Field Energy

- TRANSP's poloidal field diffusion equation for evolving the q profile from within TRANSP has an associated poloidal field balance equation evaluated at each radial surface within the plasma.
- Isolver's flux diffusion tracks a similar balance and in addition volume integrates it within and outside the plasma to form a coupled plasma/coil circuit poloidal field power balance.
- The total poloidal field energy at a time slice is given by

$$\frac{1}{2\mu_0}\int\frac{|\nabla\psi|^2}{R^2}dV=\frac{1}{2}\left\{\vec{I}\cdot M\cdot\vec{I}+I_\varphi\vec{L}^p\cdot\vec{I}+2\pi I_\varphi(\psi_0+\Delta\psi)+\mu_0 V\left(\frac{I_\varphi}{L}\right)^2\ell_i^1\right\}$$

$V, L =$ volume and cross sectional arc length of the plasma at the boundary

$\vec{I} =$ coil currents

$M =$ poloidal field coil mutual inductance matrix

$\vec{L}^p =$ mutual inductance of plasma and coils

$I_\varphi =$ plasma current

$\ell_i^1 =$ plasma normalized internal inductance

Algorithm: Poloidal Field Balance
Integrated Power Balance

- The poloidal field power balance associated with the change in the poloidal field energy is given by

$$\frac{1}{2\mu_0}\frac{d}{dt}\left[\int \frac{\left|\nabla\psi\right|^2}{R^2}dV\right] = \vec{I}\cdot\vec{\Sigma} - \vec{I}\cdot R\cdot\vec{I} - \frac{1}{2}\frac{d}{dt}\left[\vec{I}\cdot M^S\cdot\vec{I}\right] \qquad \text{circuit terms}$$

$$-\int_0^1 \frac{\partial V}{\partial\overline{\psi}}\left\langle J_\varphi E_\phi\right\rangle d\overline{\psi} \qquad \text{dissipation in plasma}$$

$$+\frac{1}{2\mu_0}\int_0^1 \frac{\Delta\psi}{q}\frac{\partial}{\partial t}\bigg|_\Phi \left[\frac{\left\langle\frac{\left|\nabla\Phi\right|^2}{R^2}\right\rangle}{F\left\langle\frac{1}{R^2}\right\rangle}\right] d\overline{\psi} \qquad \text{compression in plasma}$$

$$+\frac{1}{2\mu_0}\oint_{boundary}\frac{\left|\nabla\psi\right|^2}{R^2}\hat{\mathrm{n}}\cdot\frac{\partial\vec{\mathrm{r}}}{\partial t}\bigg|_\Phi dS \qquad \text{compression of boundary}$$

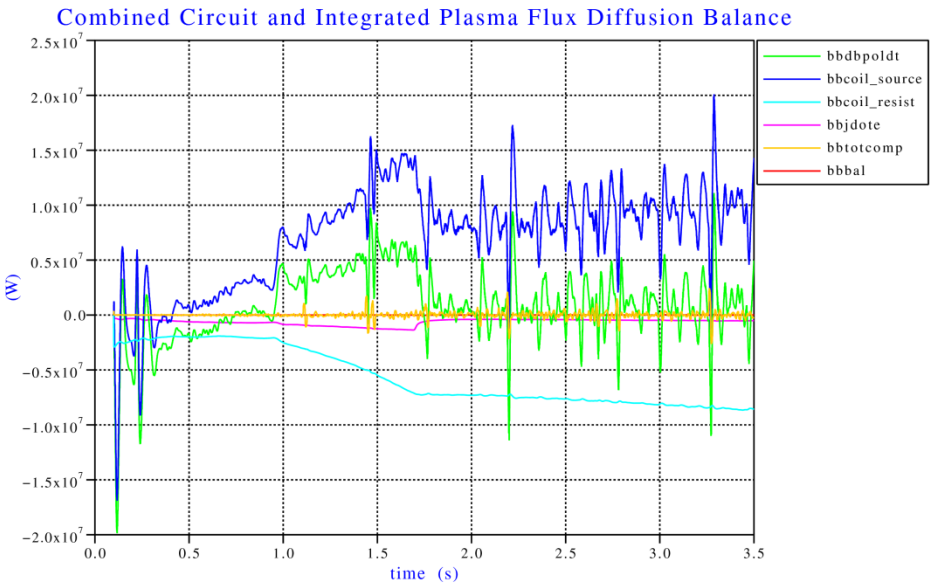
$\vec{\Sigma} = \text{coil circuit voltage and current sources}$

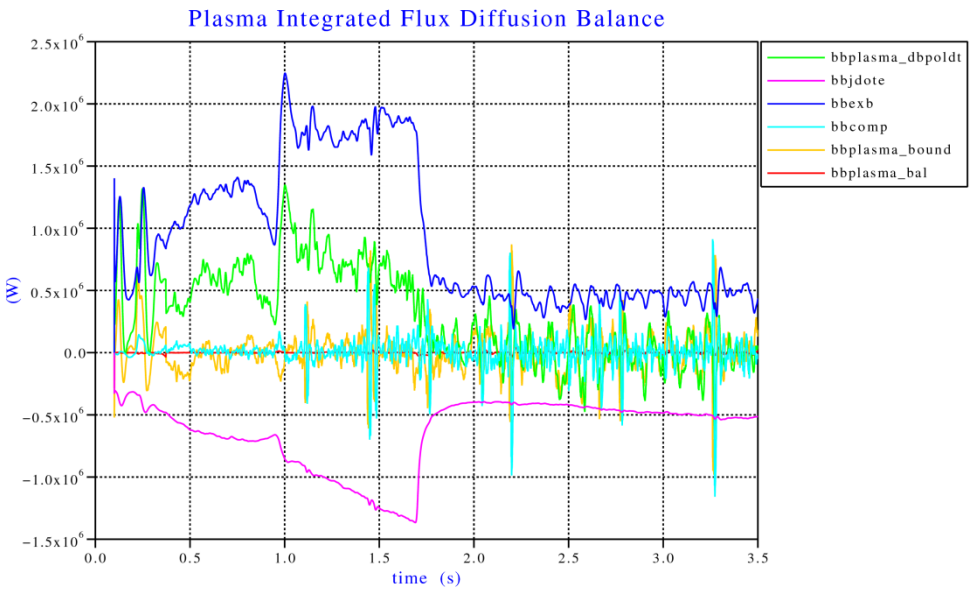
$\vec{R} = \text{coil circuit resistance matrix}$

$M^S = \text{coil circuit source mutual inductance}$

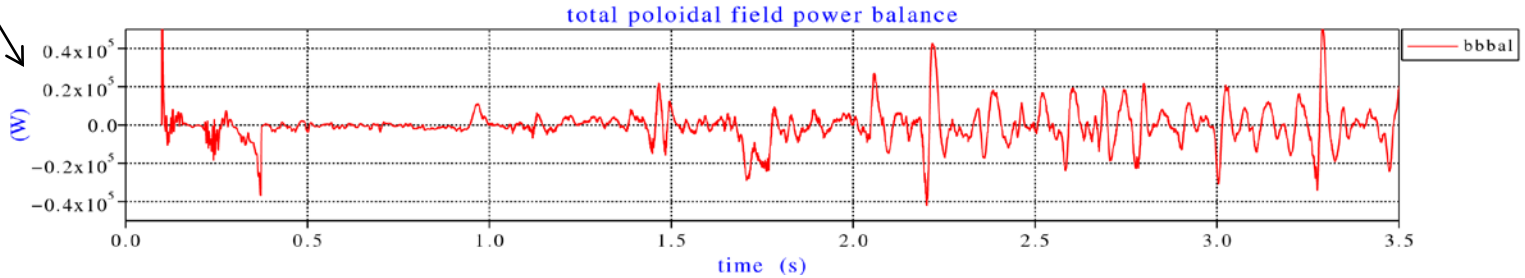
Algorithm: Poloidal Field Balance
Example D3D

- Below are examples of the total and plasma integrated poloidal field power balance for a D3D run
  - The plasma only integrated power also includes an ExB term and boundary movement term





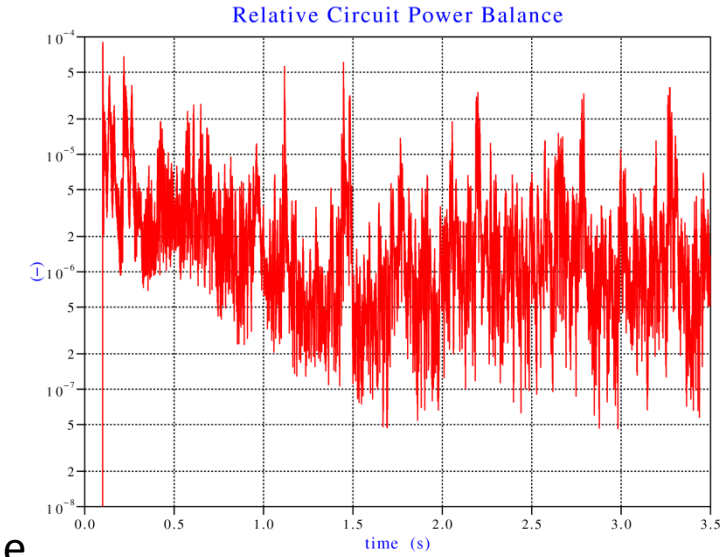
Balance only



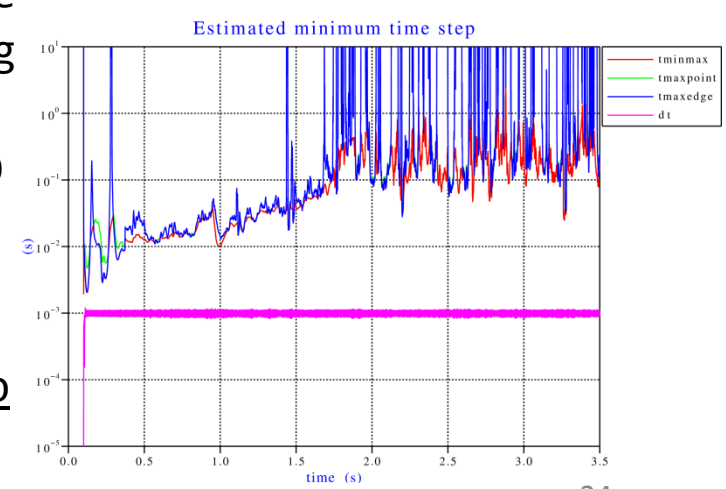
Algorithm: Miscellaneous Output

- The circuit equations have a separate balance equation where the plasma acts as an additional circuit source.
  - Circuit balance can be used to limit the time step

$$\frac{1}{2}\frac{d}{dt}\left[\vec{I}\cdot(M+M^{s})\cdot\vec{I}\right]=\vec{I}\cdot\vec{\Sigma}-\vec{I}\cdot R\cdot\vec{I}-\vec{I}\cdot\frac{d}{dt}\left[I_{\varphi}\vec{L}^{p}\right]$$



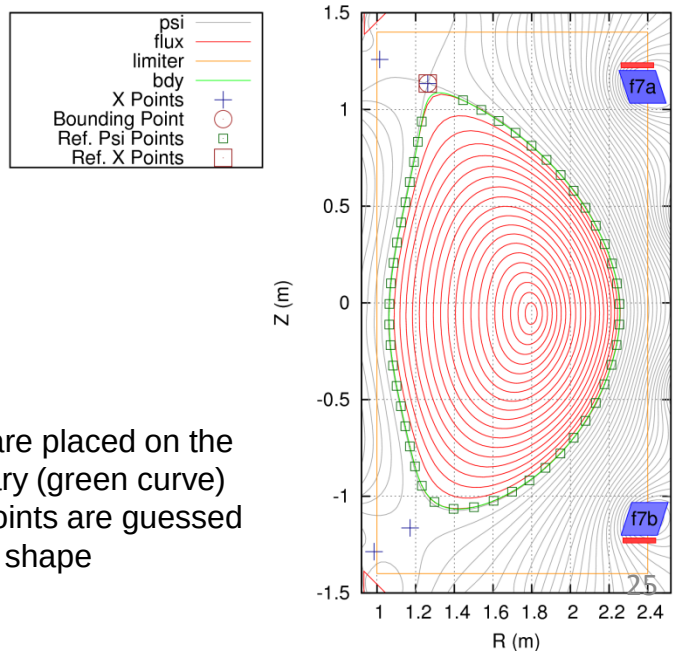
- The flux diffusion algorithm will fail if the time step is too large
- One of the outputs is an estimate of the largest time step which may be taken next time while accounting for a safety factor.
  - Time step must be short enough to ensure a monotonic $\phi(\overline{\psi})$
  - Use of the mixed boundary condition q_{mix} introduces a courant type of time step condition at the edge
- While not evident in this case, the use of the poloidal flux axis in Isolver <u>can unfortunately lead to sub millisecond time steps</u> especially during the early time part of the current ramp up.



Algorithm: Coil Currents

Least Squares Match to Prescribed Boundary

- The **Least Squares Mode** of running Isolver returns a set of coil currents which provides the best match to a prescribed boundary.
  - This mimics the traditional mode of running TRANSP which assumed a fixed boundary solver
- To stabilize the picard iteration a match is also made to the nearest X points offset from the prescribed boundary.
  - Given just the boundary, the X points are guessed from the points on the boundary with the smallest radius of curvature and offset from the boundary by the radius of curvature.
  - The weight assigned to guessed X points decreases exponentially as the square of the distance from the boundary.
- Optionally, the coil currents may also be constrained to a fixed value or included in the least squares weighting.
- The least squares solution is computed using a singular value decomposition. Small eigenvalues are effectively dropped while maintaining solution continuity.

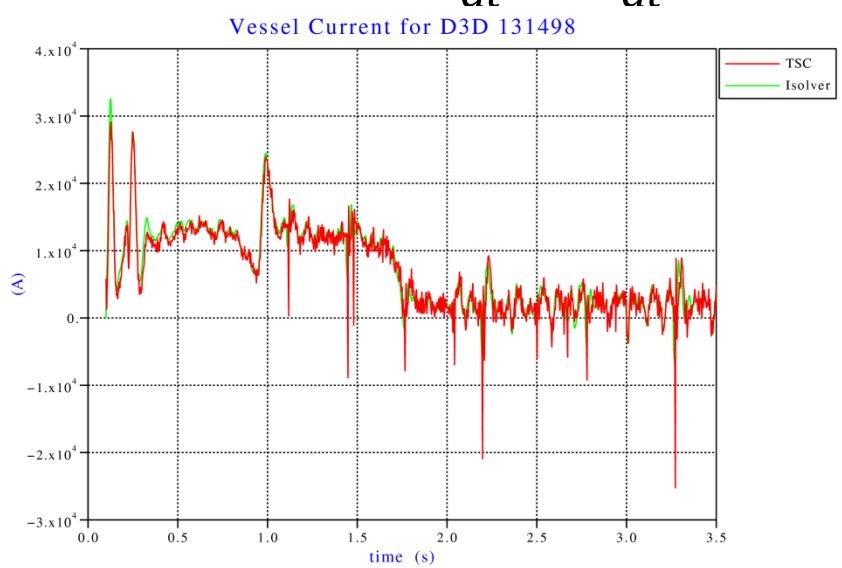


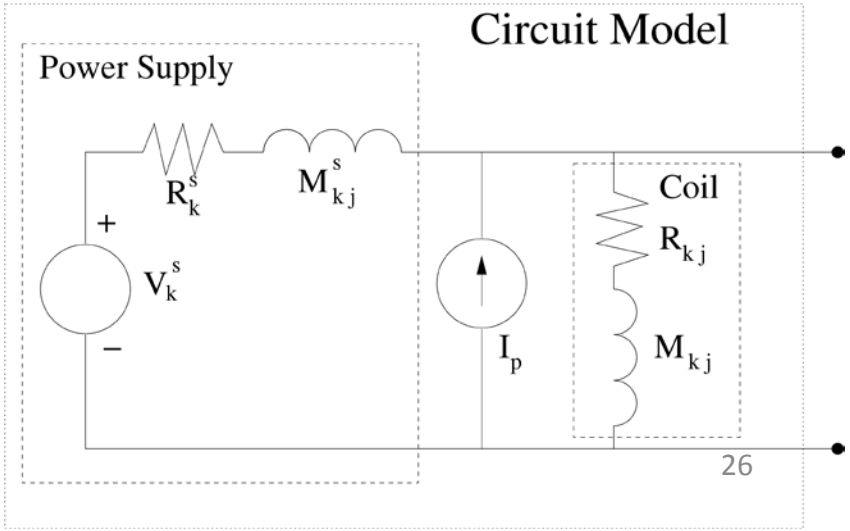
Reference points are placed on the prescribed boundary (green curve) and reference X points are guessed from the boundary shape

Algorithm: Coil Currents
Circuit Model

- In **Circuit Equation Mode**, a circuit drives each coil to advance the coil currents in time.
- Selection of circuit parameters determines coil and drive.
  - **Passive Structure**: no sources $R^s = 0$
  - **Current driven PF coil**: $I_p \neq 0 \quad R^s = \infty$
  - **Voltage driven PF coil**: $I_p = 0 \quad V^s, R^s$ set by power supply Not currently working
- Cross coupled coil resistance R_{kj} allows the modeling of toroidal cuts in the vessel.
- Isolver advances the coil current in time through a loop current circuit matrix.

$$\vec{\mathbf{V}} = \mathbf{R}\vec{I} + \mathbf{M}\frac{d\vec{I}}{dt} + \frac{d(I_\varphi \vec{L}^{\mathrm{p}})}{dt}$$





Algorithm: Coil Currents
Circuit Feedback for Centering the Plasma

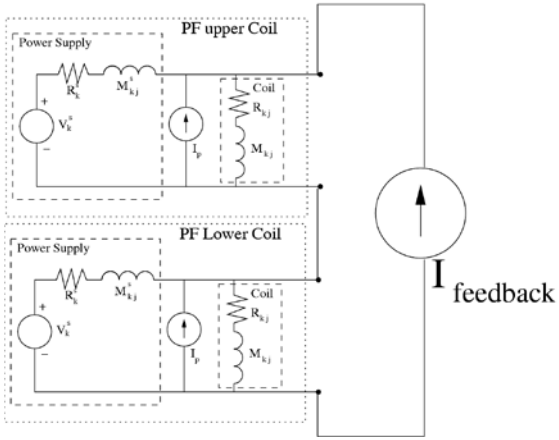
- As part of **Circuit Equation Mode,** a feedback current source can drive several coils to keep the plasma centered
- Several points are selected on the prescribed boundary to provide a feedback which forces the weighted poloidal flux difference between the feedback points to zero.

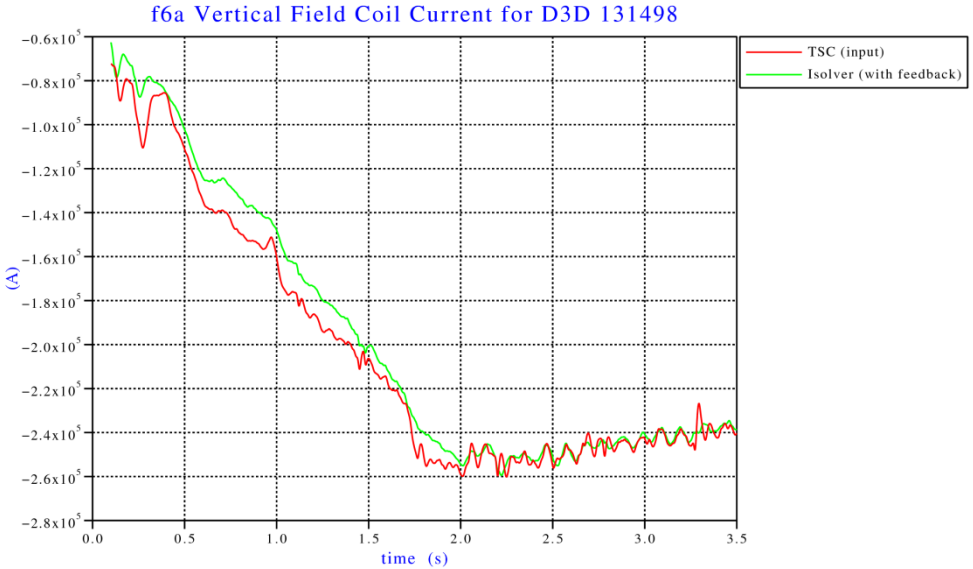
$$I_{\text{feedback}} = \frac{-\beta}{\psi^F} \sum_{k \text{ points}} w_k \psi(\vec{r}_k)$$

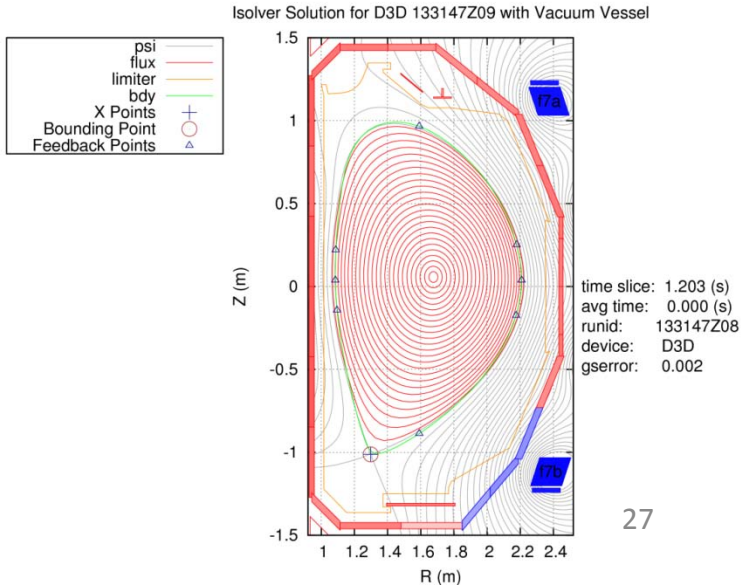
$$\psi^F = \text{no plasma response from unit } I_{\text{feedback}}$$

$$\beta = \text{feedback}$$

Vertical/Horizontal Feedback

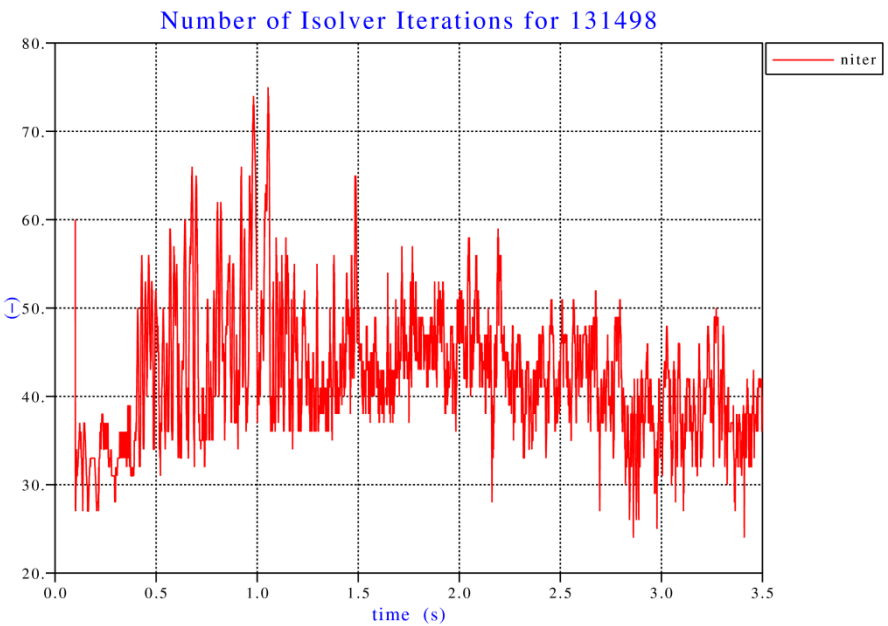


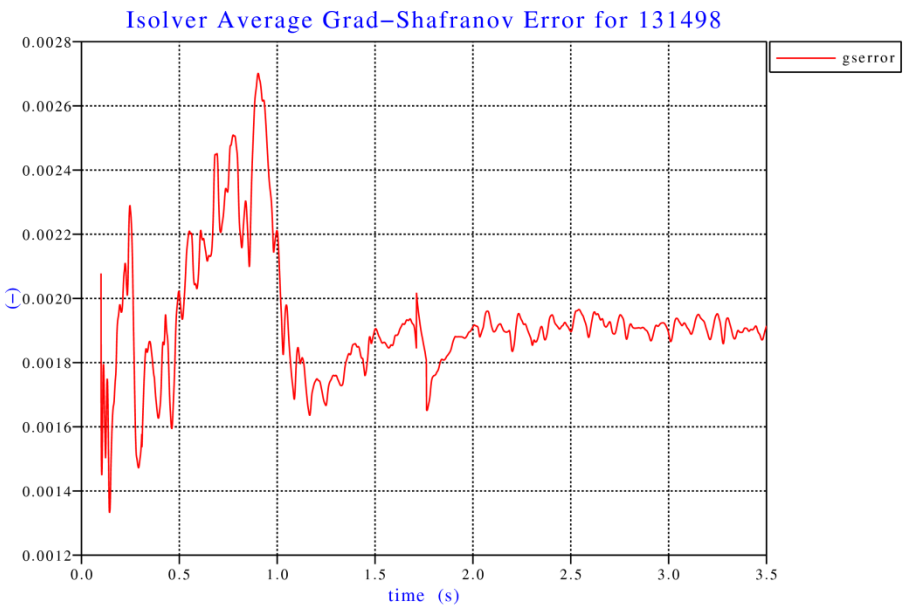




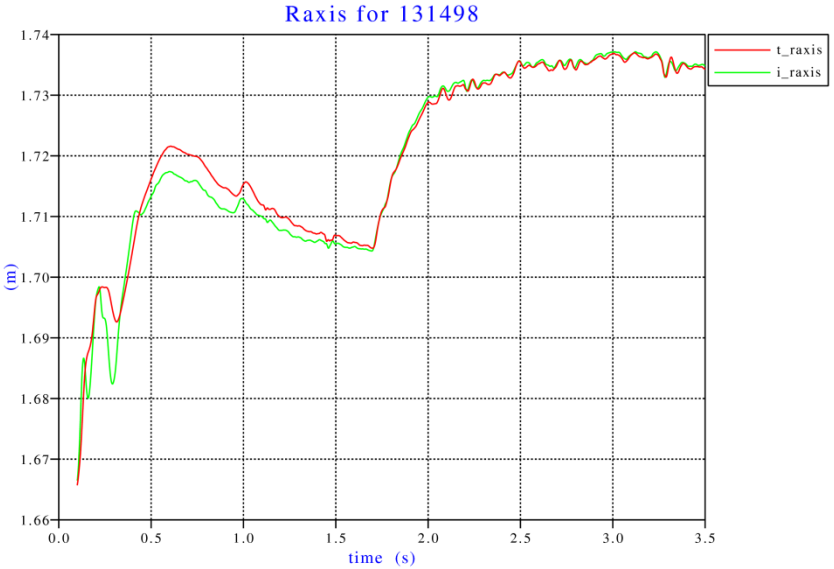
TSC, Isolver comparison D3d 131498

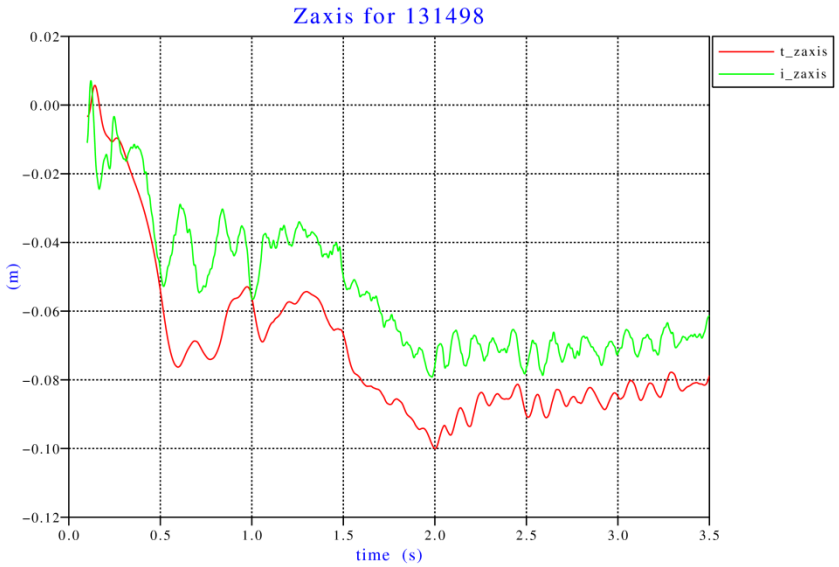
- Various results comparing a TSC run of D3D 131498 with Isolver have been given.
  - Isolver was run outside of TRANSP in plasma current prediction mode using TSC's pressure and resistivity profiles and TSC coil currents as inputs to Isolver
  - The driven currents were estimated from a TRANSP run
- Below are some additional outputs for comparison.

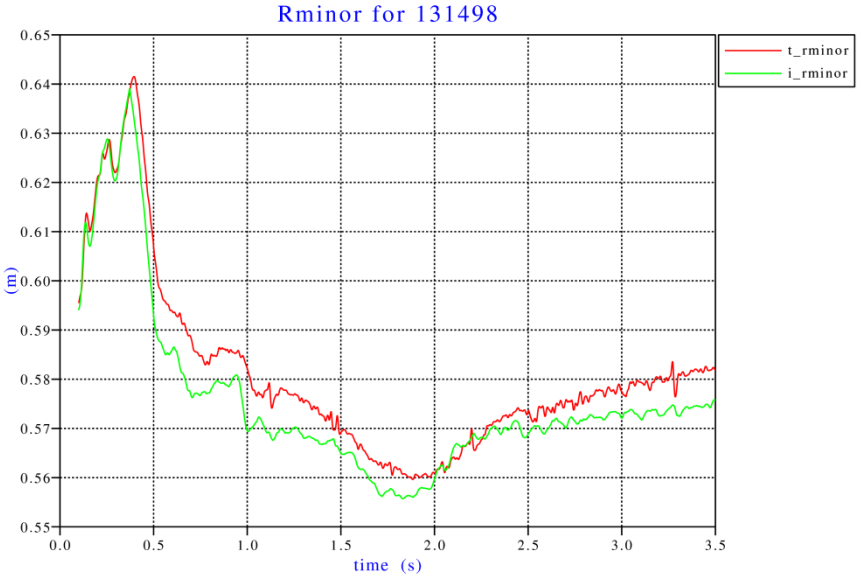




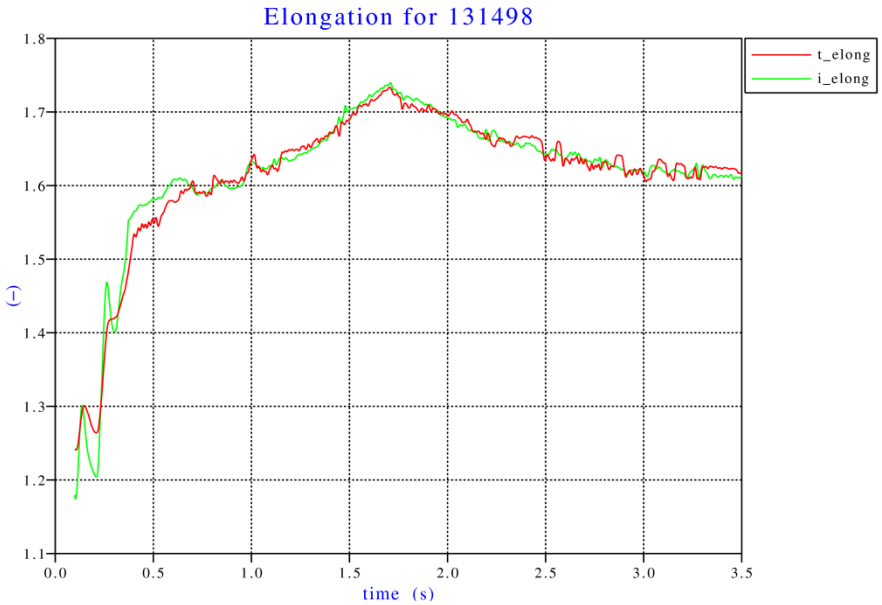
TSC, Isolver comparison D3d 131498

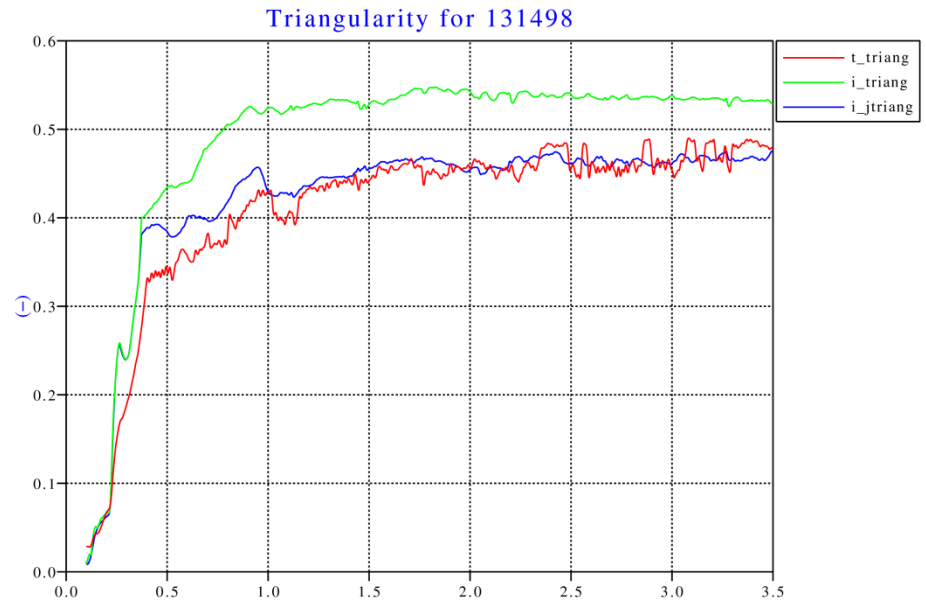


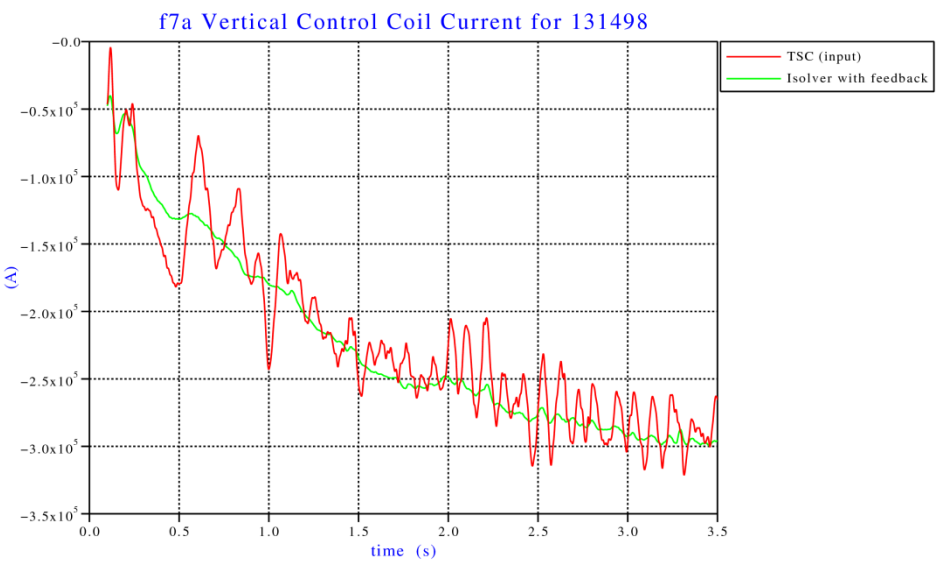


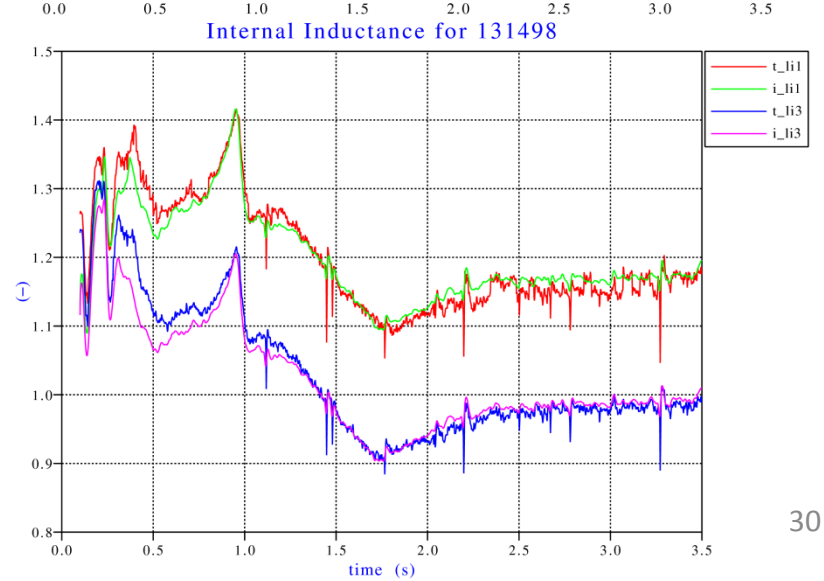


TSC, Isolver comparison D3d 131498

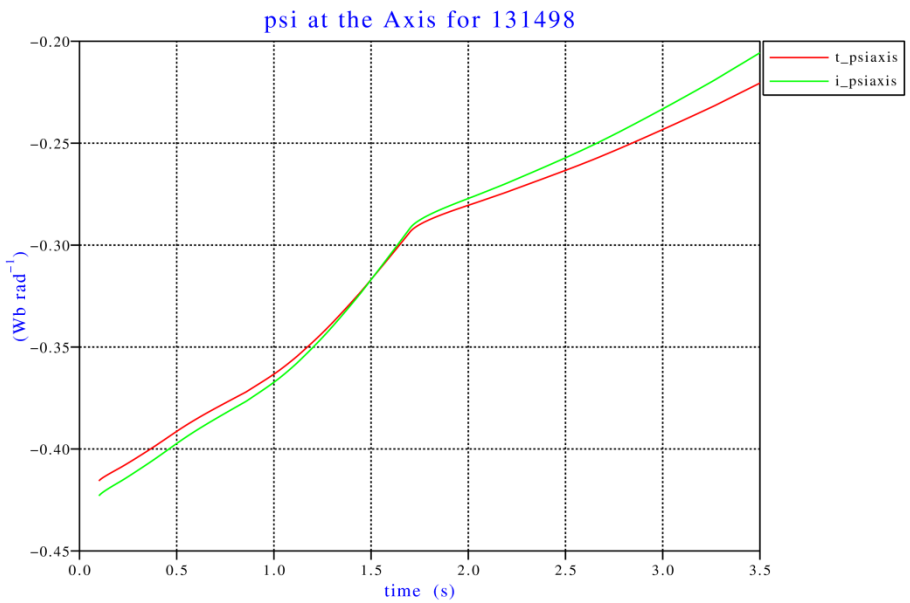


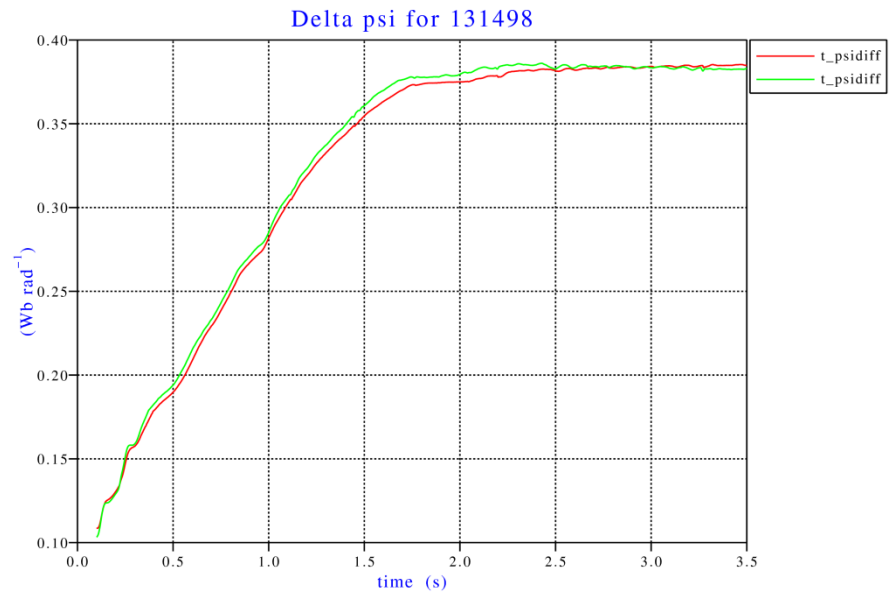






TSC, Isolver comparison D3d 131498





Summary of Isolver in TRANSP

- Isolver free boundary solution based only on $J_{\varphi}(R,Z)$ on an R,Z grid
  - An inverse solver is not used
  - Unable to model current sheets due to R,Z grid resolution
- Isolver uses an algorithm which avoids the singularity in the q profile
  - Allows the advancement of the q profile within the picard loop
  - q profile matches well to the input except near the boundary where F must remain smooth
- Flux diffusion advances the q profile
  - Boundary condition can be chosen to match the plasma current, surface loop voltage or for inductive coupling which will predict the plasma current
- Coil currents can be chosen to best match a prescribed boundary or as the solution of a circuit equation which drives the poloidal field coils
  - Vessel currents modeled as passive coils in circuit equation mode
- Balance equations check the fidelity of the flux diffusion and circuit equation solutions
- Comparison with a TSC D3D run shows very good agreement
  - Comparison of plasma current prediction in an NSTX shot did not look as good – under investigation