Validation of the static forward Grad-Shafranov equilibrium solvers in FreeGSNKE and Fiesta using EFIT++ reconstructions from MAST-U

K. Pentland, N.C. Amorsico, O. El-Zobaidi, S. Etches, A. Agnello, G. K. Holt, C. Vincent, J. Buchanan, S. J. P. Pamela, G. McArdle, L. Kogan, G. Cunningham

Science and Technology Facilities Council

9th July 2024

UK Atomic Energy Authority

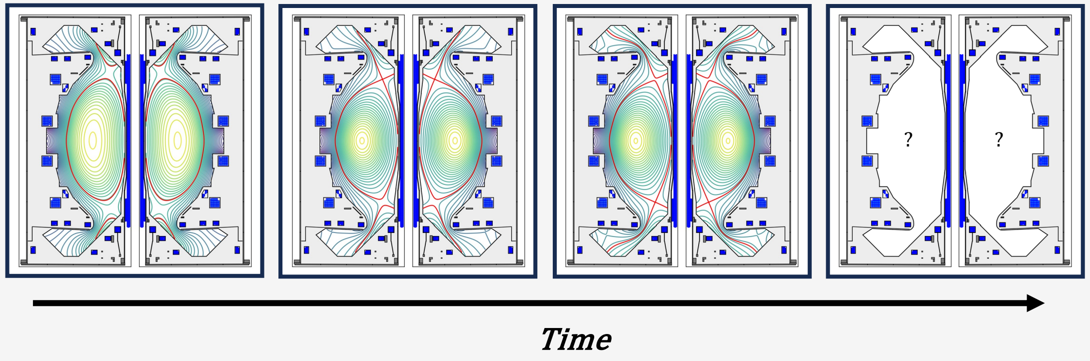
Motivation and aims

Why model plasma equilibria?

- Crucial to have fast and accurate numerical methods for simulating MHD equilibria.
- These methods used to design/analyse plasma scenarios, shapes, stability + inform plasma control/feedback systems.

Our focus:

- The static (free-boundary) forward Grad-Shafranov (GS) problem.
- Robust validation of static GS solvers, against both analytical solutions and real-world tokamak plasmas, critical for modelers that require reliable equilibrium calculations.



What are our aims?

- Validate the static GS solvers in FreeGSNKE and Fiesta by demonstrating they can reproduce equilibria obtained by magnetics-only EFIT++ reconstructions on MAST-U.
- Compare poloidal flux quantities, shape control measures (e.g. midplane radii, magnetic axes, and separatrices), and other targets (e.g. X-points and strikepoints).
- Do this for several physically different MAST-U shots.

Overview

The solvers

The static forward Grad-Shafranov problem

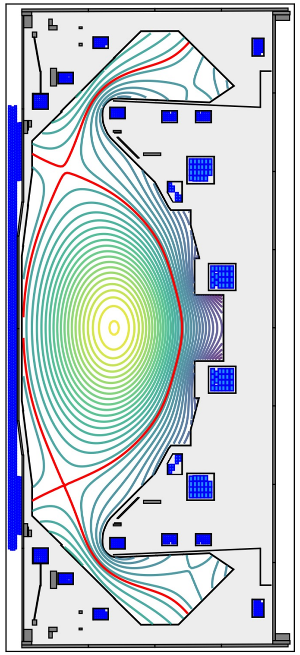
Input parameters (for MAST-U)

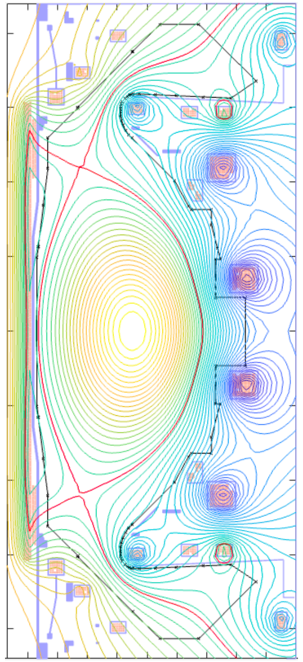
Numerical results: shot 45425 (conventional)

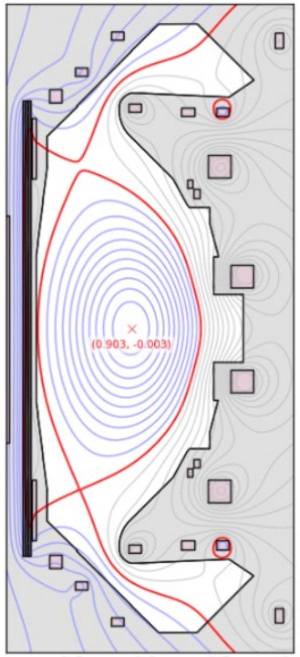
Conclusions

Additional results: shot 45292 (Super-X)

The solvers


FreeGSNKE


Fiesta



EFIT++

The solvers: FreeGSNKE

What is it?

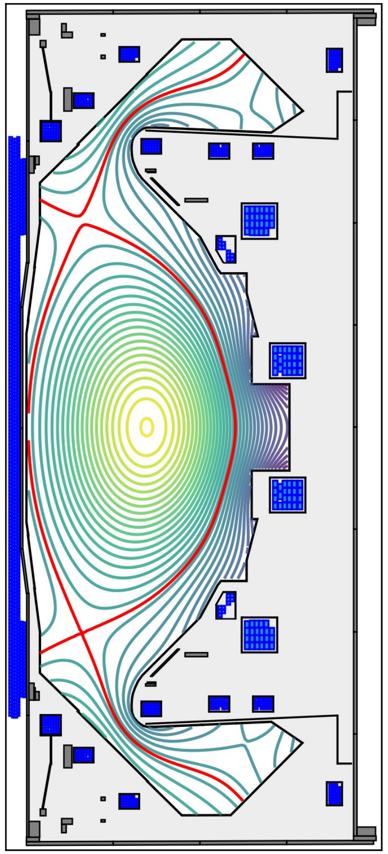
- A Python-based, finite difference, dynamic free-boundary equilibrium solver.
- Developed between Hartree centre and UKAEA (Amorsico et al., 2024).
- Extension of FreeGS (Dudson et al., 2024):
  - ➢ contains static forward/inverse GS solvers (Picard iteration based).
  - ➢ aided design of numerous tokamaks (SPARC, ARC, KSTAR, WEST, MANTRA, and more…).
- FreeGSNKE introduces Newton-Krylov method into static forward solver + introduces dynamic solver → mitigates (vertical) numerical instability affecting Picard iterations.
  - ➢ due to highly-elongated plasma shape + mathematical feature of the Picard iterations.

Progress so far?

- Emulation of scenario and control design in MAST-U-like tokamak (Agnello et al., 2024).
- Static solver validated against analytic GS solutions.
- Inverse solver identical to one in FreeGS.

What's our aim?

- Go a step further and validate static solver against EFIT++ reconstructions on (full) MAST-U machine.



FreeGSNKE

The solvers: Fiesta

What is it?

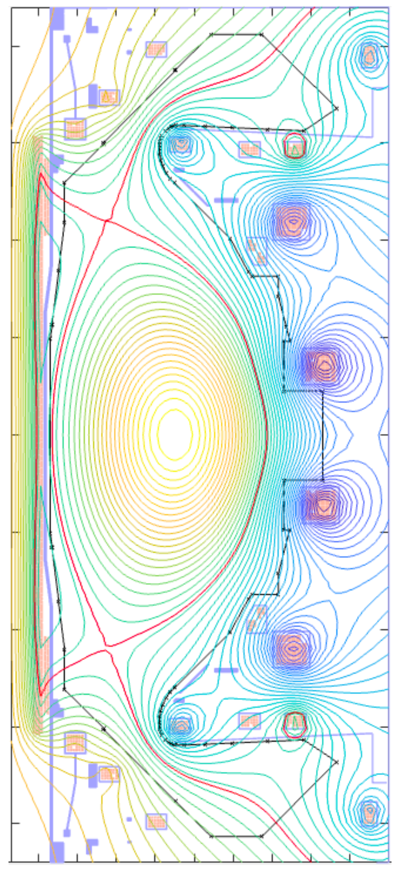
- A MATLAB-based, finite difference, static free-boundary equilibrium solver.
- Developed by Cunningham (2013) and others at UKAEA.
- Capabilities:
  - ➢ Has static forward/inverse GS solvers (Picard-based).
  - ➢ Linearised dynamic modelling using RZIp rigid plasma framework (Coutlis et al., 1999).

Progress so far?

- Validated on and aided design of numerous tokamaks (MAST-(U), JET, STEP, DIII-D, NSTX, and more…).

What is its purpose for us?

- Given historical use, we run Fiesta alongside FreeGSNKE to demonstrate they both return quantitatively similar equilibria.
- Cross-validation is also helpful for identifying differences between each implementations.



Fiesta

The solvers: EFIT++

What is it?

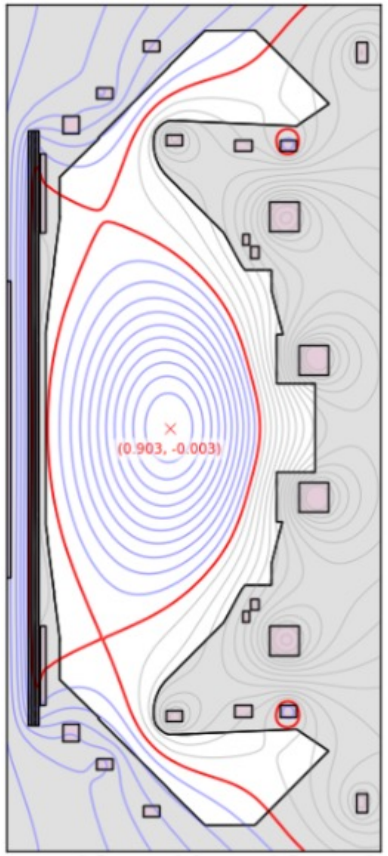
- Computational method widely used to "fit" (i.e. reconstruct) plasma equilibria using diagnostic measurements.
- Does this by solving an (ill-posed) linearised least-squares minimization problem (involving the measurement data) such that the linearised GS equation is satisfied.
- Diagnostics include: flux loops, pickup and Rogowski coils, motional Stark effect (MSE), Thompson scattering, etc.

How is it currently used?

- Different versions of EFIT validated/used on many (too many to list) tokamaks.
- Used for (real-time?) reconstruction (for intra-shot plasma control) and post-shot experimental analysis.
- Run routinely with magnetics diagnostics only (with MSE/Thompson scattering if available).

What is its purpose for us?

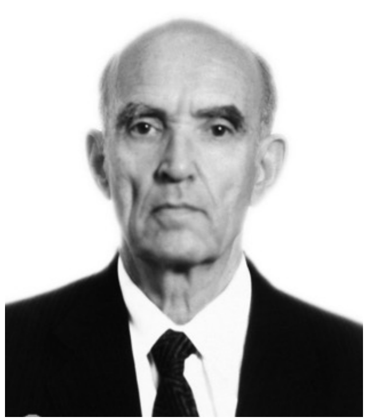
- Given widespread faith(?) in EFIT++ equilibria, we use them as trusted sources of reference equilibria against which to compare both FreeGSNKE and Fiesta.



EFIT++

The static forward Grad-Shafranov problem

Harold Grad



Vitaly Shafranov

The static forward (free-boundary) Grad-Shafranov problem

What is it/what are we solving for?

- A nonlinear elliptic PDE with (initially unknown) Dirichlet BC – solved in poloidal plane.
- Total poloidal flux ψ is sum of plasma (ψ_p) and external conductor (ψ_c) flux:

$$\psi(r,z) = \psi_p(r,z) + \psi_c(r,z).$$

Plasma current density

$$J_p(\psi, r) = r\frac{\partial p}{\partial \psi} + \frac{1}{\mu_0 r} F \frac{\partial F}{\partial \psi}, \quad (r,z) \in \Omega_p$$

$$I_j^c(r,z) = \begin{cases} I_j^c & \text{if } (r,z) \in \Omega_j^c, \\ 0 & \text{elsewhere.} \end{cases}$$

Conductor current density

$$J_c(r,z) = \sum_{j=1}^{N_c} \frac{I_j^c(r,z)}{A_j^c}, \quad (r,z) \in \Omega$$

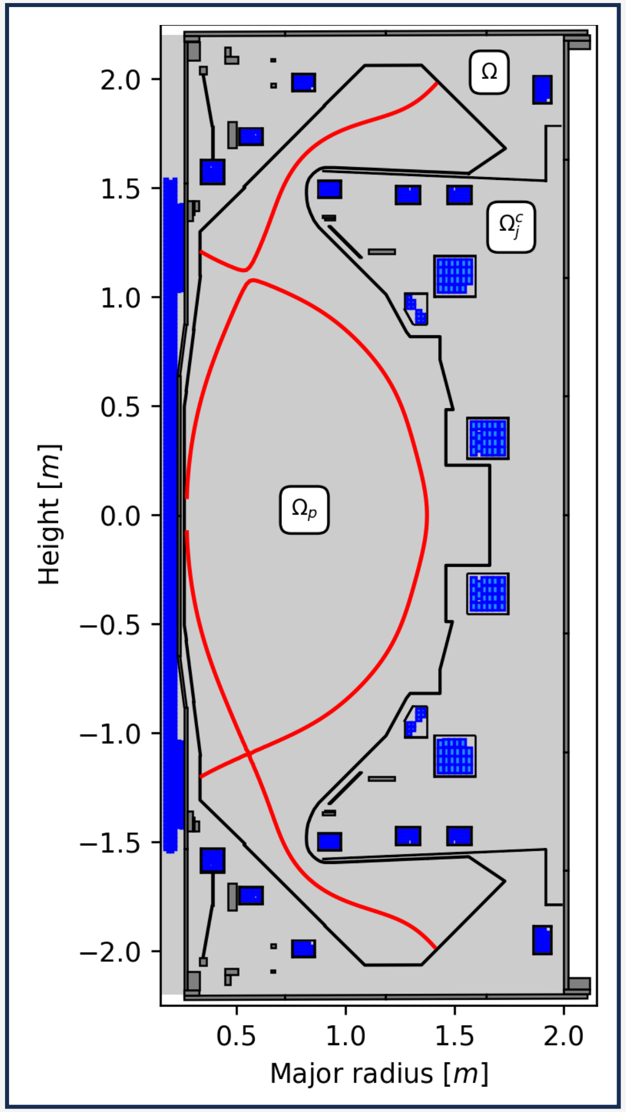
Linear elliptic operator

$$\Delta^* := r\partial_r r^{-1}\partial_r + \partial_{zz}$$

PDE: $$\Delta^* \psi = -\mu_0 r \underbrace{(J_p + J_c)}_{=J_\phi}, \quad (r,z) \in \Omega$$

BC: $$\psi\Big|_{\partial\Omega} = \int_{\Omega_p} G(r,z;r',z') J_p(\psi, r', z')\, \mathrm{d}r'\mathrm{d}z' + \sum_{j=1}^{N_c} \frac{1}{A_j^c} \int_{\Omega_j^c} G(r,z;r',z') I_j^c(r',z')\, \mathrm{d}r'\mathrm{d}z'$$

Plasma contribution Conductor contribution



The static forward (free-boundary) Grad-Shafranov problem

How to solve (rough outline):

1. Spatially discretise domain: finite differences here (FreeGSNKE = 4th order, Fiesta = 2nd order, EFIT++ = ?).
2. Iteratively solve for flux as $\psi^{(n)} = \psi_p^{(n)} + \psi_c^{(n)}$:
   - Requires an initial guess $\psi_p^{(0)}$ to begin iterations ($\psi_c^{(n)}$ is fixed and known for all n).
3. Calculate flux on $\partial\Omega$ (requires identifying plasma boundary – nonlinear):

$$\psi^{(n)}\Big|_{\partial\Omega} = \int_{\Omega_p} G(r,z;r',z')J_p(\psi^{(n)},r',z')\,\mathrm{d}r'\mathrm{d}z' + \sum_{j=1}^{N_c} \frac{1}{A_j^c}\int_{\Omega_j^c} G(r,z;r',z')I_j^c(r',z')\,\mathrm{d}r'\mathrm{d}z'$$

4. Solve GS equation iteratively (Fiesta and EFIT++) via Picard iterations:

$$\Delta^*\psi^{(n+1)} = -\mu_0 r J_\phi(\psi^{(n)},r,z), \quad (r,z)\in\Omega$$

5. Check convergence (else repeat from 2):

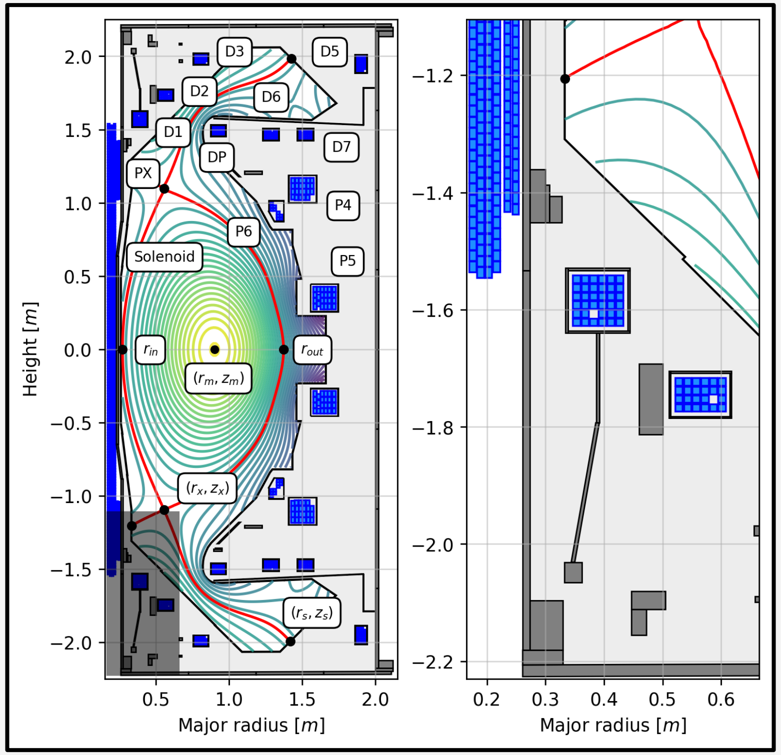
$$\frac{\max|\psi^{(n+1)} - \psi^{(n)}|}{\max(\psi^{(n)}) - \min(\psi^{(n)})} < \varepsilon$$

Note: FreeGSNKE uses NK method to directly solve for roots of:

$$\Delta^*\psi + \mu_0 r J_\phi(\psi,r,z) = 0, \quad (r,z)\in\Omega \longrightarrow \mathcal{F}(\psi) = 0$$

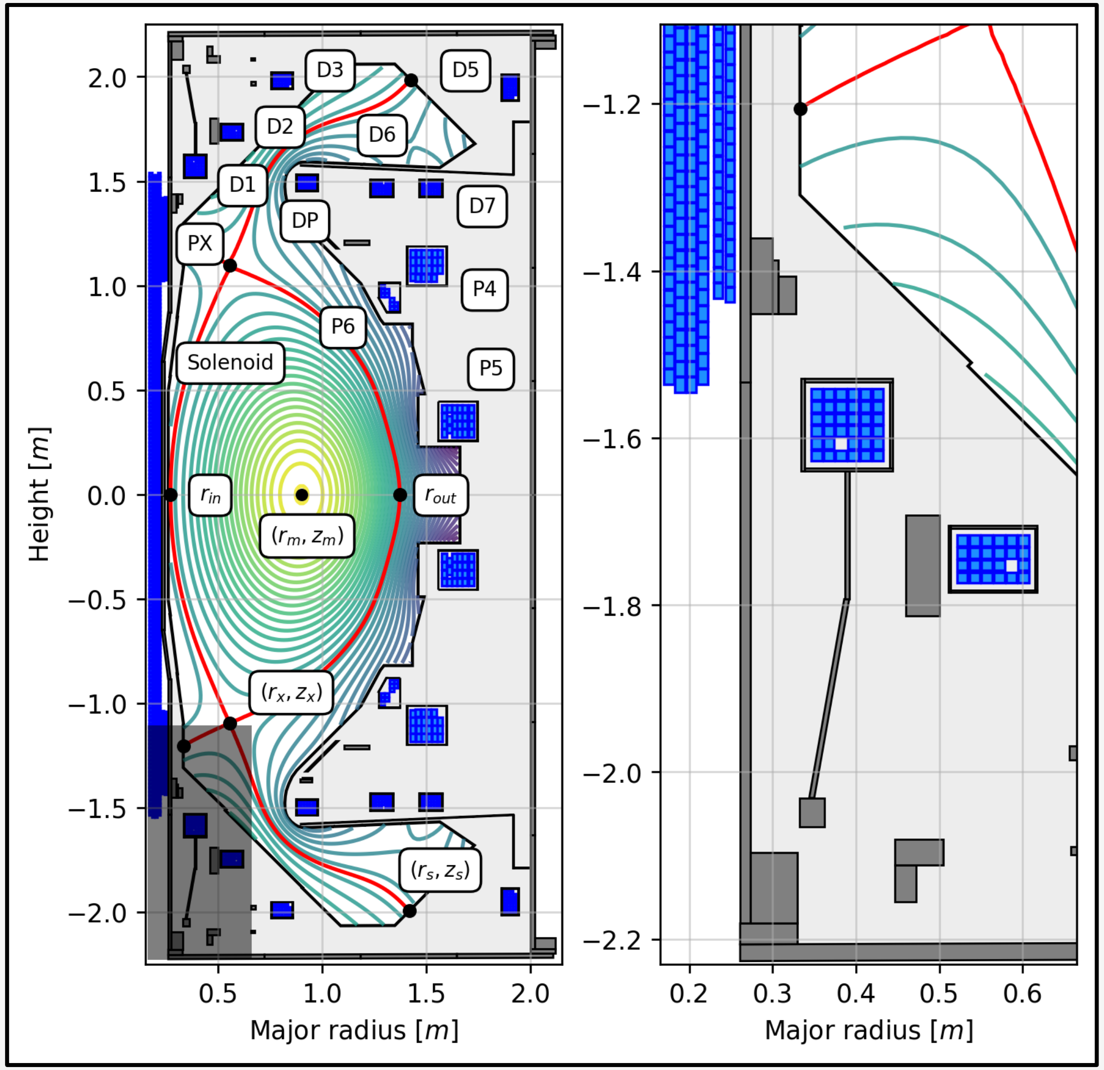
(Still requires an initial guess and the nonlinear BC calculation).

Input parameters (for MAST-U)



Input parameters (for MAST-U)

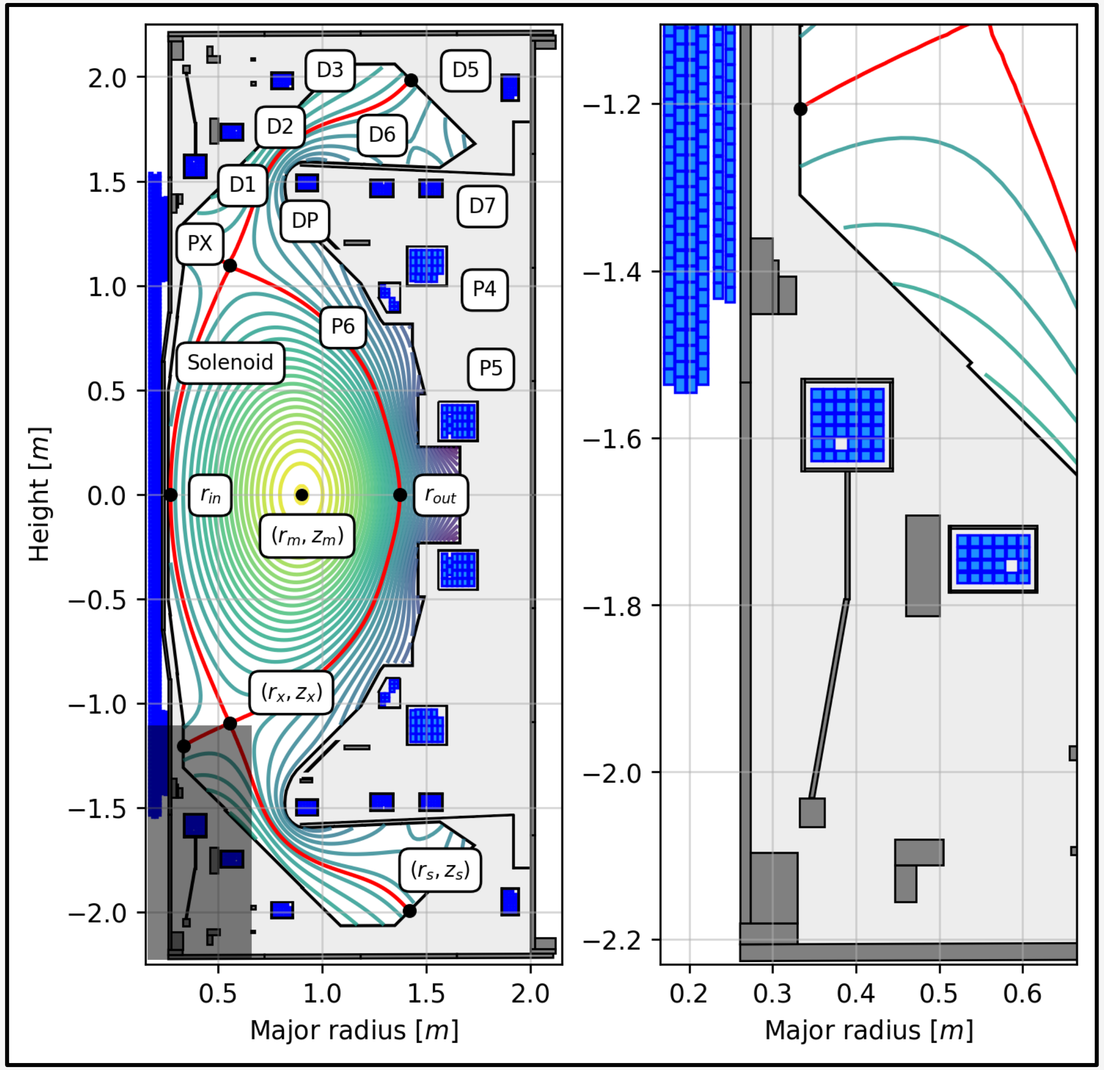
Need consistent inputs across all three codes, including:



Input parameters (for MAST-U)

Need consistent inputs across all three codes, including:

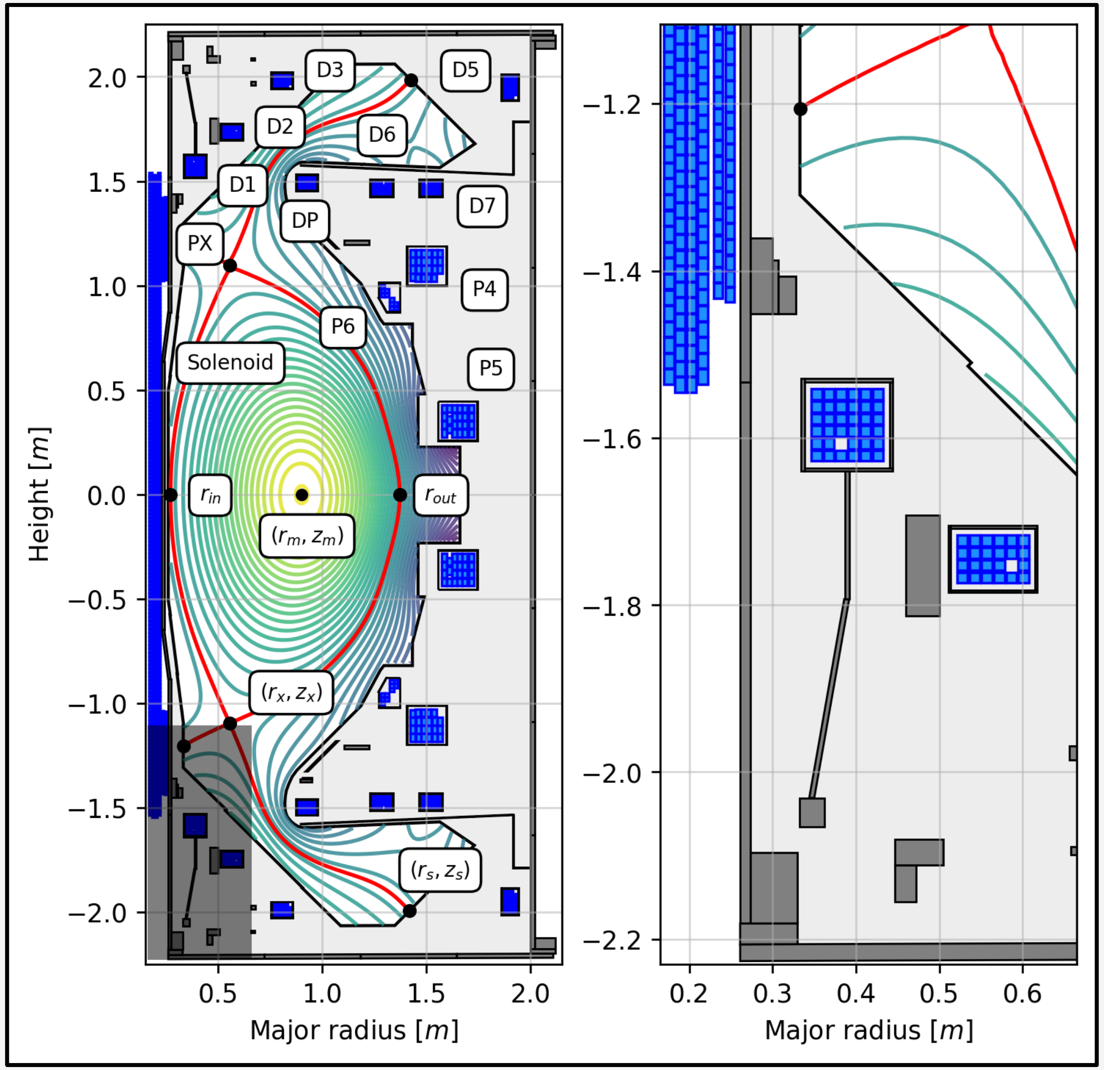
1. Accurate and representative machine description:
   a) Active poloidal field coils:
      - 12 active coils for plasma shape control.
      - Each have an upper and lower component wired in series (except for P6 coil, anti-series).
      - Each made up on filaments/winding with:
        - (r, z) position.
        - (dr, dz) width and height.
        - Polarity (1 or -1).
        - Current multiplier (solenoid only).
        - Resistivity (not required here).



Input parameters (for MAST-U)

Need consistent inputs across all three codes, including:

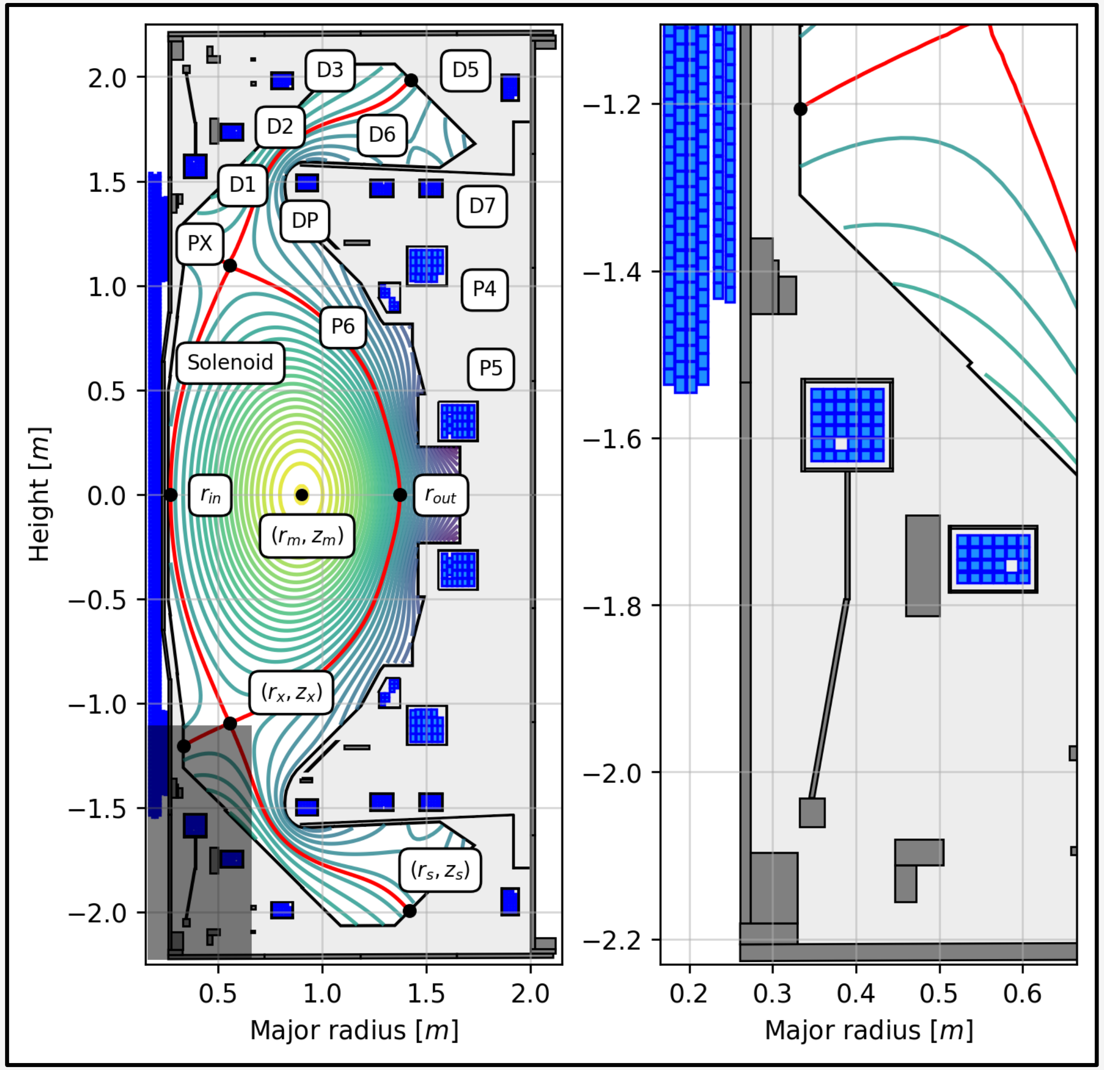
1. Accurate and representative machine description:
   a) Active poloidal field coils.
   b) Passive structures:
      - Eddy currents induced in toroidally continuous structures (from plasma and coils) greatly impacts plasma stability/modelling.
      - 150 passive structures are included (from vessel, support structures, coil cases, etc.).
      - Modelled as parallelograms via $(r, z, dr, dz, \theta_1, \theta_2)$.
      - FreeGSNKE and Fiesta uniformly distribute current density over cross-section via "refinement" (will revisit).



Input parameters (for MAST-U)

Need consistent inputs across all three codes, including:

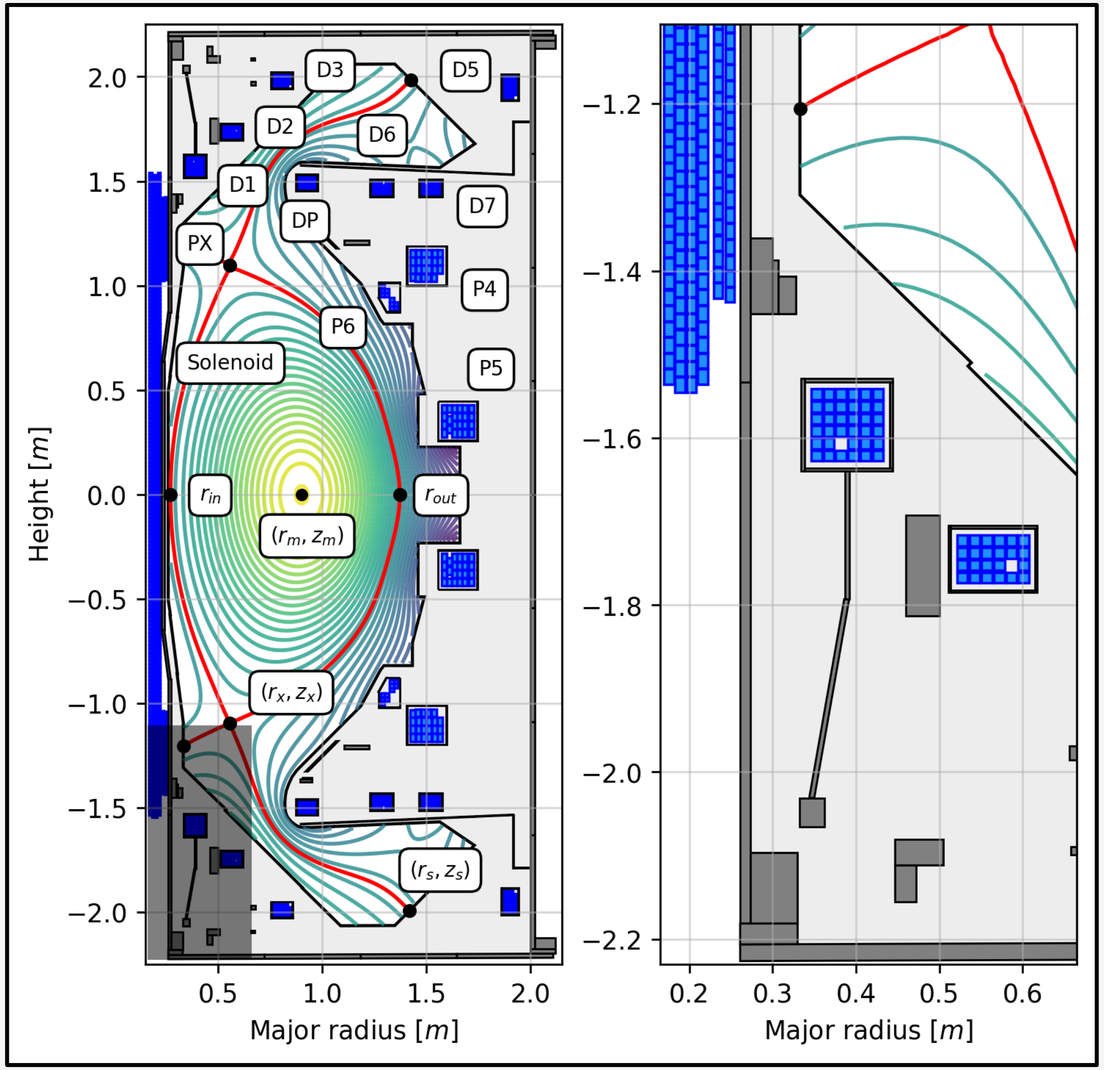
1. Accurate and representative machine description:
    a) Active poloidal field coils.
    b) Passive structures.
    c) Limiter/wall structure:
        - Confines the boundary of the plasma.
        - Modelled by 98 pairs of (r, z) coordinates.



Input parameters (for MAST-U)

Need consistent inputs across all three codes, including:

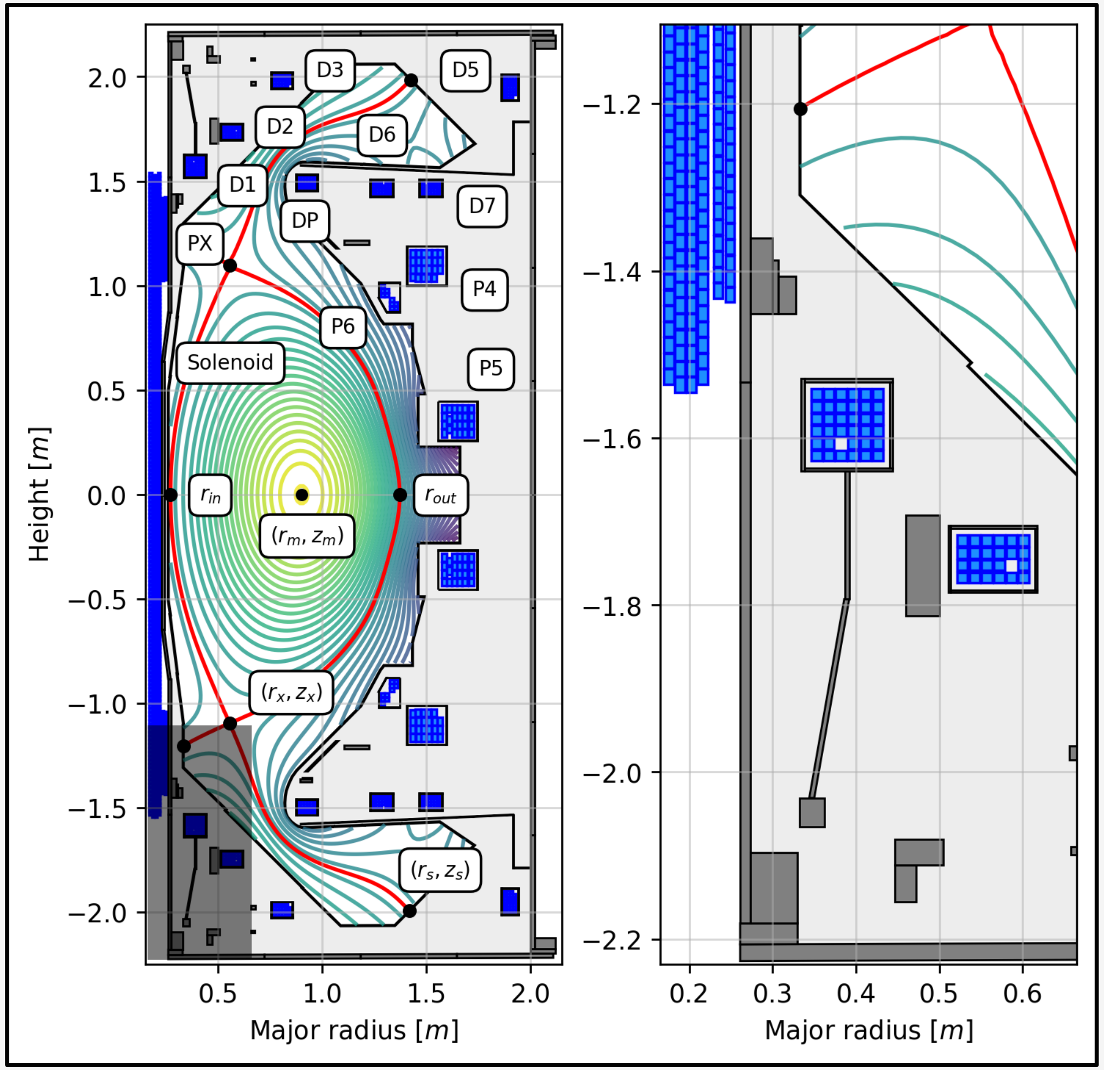
1. Accurate and representative machine description:
    a) Active poloidal field coils.
    b) Passive structures.
    c) Limiter/wall structure.
2. Fitted active coil/passive structure currents:
    - EFIT++ fits currents to upper/lower active coils independently (i.e. the coils **are not** wired in series).
    - This *non-symmetric* coil setting is adopted in FreeGSNKE/Fiesta as to recreate EFIT++ results as closely as possible.
    - We simply assign the fitted currents from EFIT++ to each coil in FreeGSNKE/Fiesta.
    - Both codes also have a *symmetric* setting, whereby upper/lower coils are wired in series.



Input parameters (for MAST-U)

Need consistent inputs across all three codes, including:

1. Accurate and representative machine description:
    a) Active poloidal field coils.
    b) Passive structures.
    c) Limiter/wall structure.
2. Fitted active coil/passive structure currents:
    - EFIT++ fits currents to **some** passive structures directly (e.g. coil cases).
    - Others are generated by electromagnetic induction (EM) model in groups – no measurements/reduces dofs.
    - Group currents assigned to individual passive structures proportionally by fraction of cross-sectional area.
    - Both codes can "refine" passives into smaller filaments for improved EM modelling.
    - Filaments are roughly same size and receive proportional current.
    - FreeGSNKE and Fiesta do this in slightly different ways.



Input parameters (for MAST-U)

Need consistent inputs across all three codes, including:

1. Accurate and representative machine description:
   a) Active poloidal field coils.
   b) Passive structures.
   c) Limiter/wall structure.
2. Fitted coil/passive structure currents.
3. Form of plasma profile functions and coefficients:
   - Same profiles/coefficients as EFIT++.
   - Low-order polynomials for magnetics-only fits.

"Lao" plasma profile functions

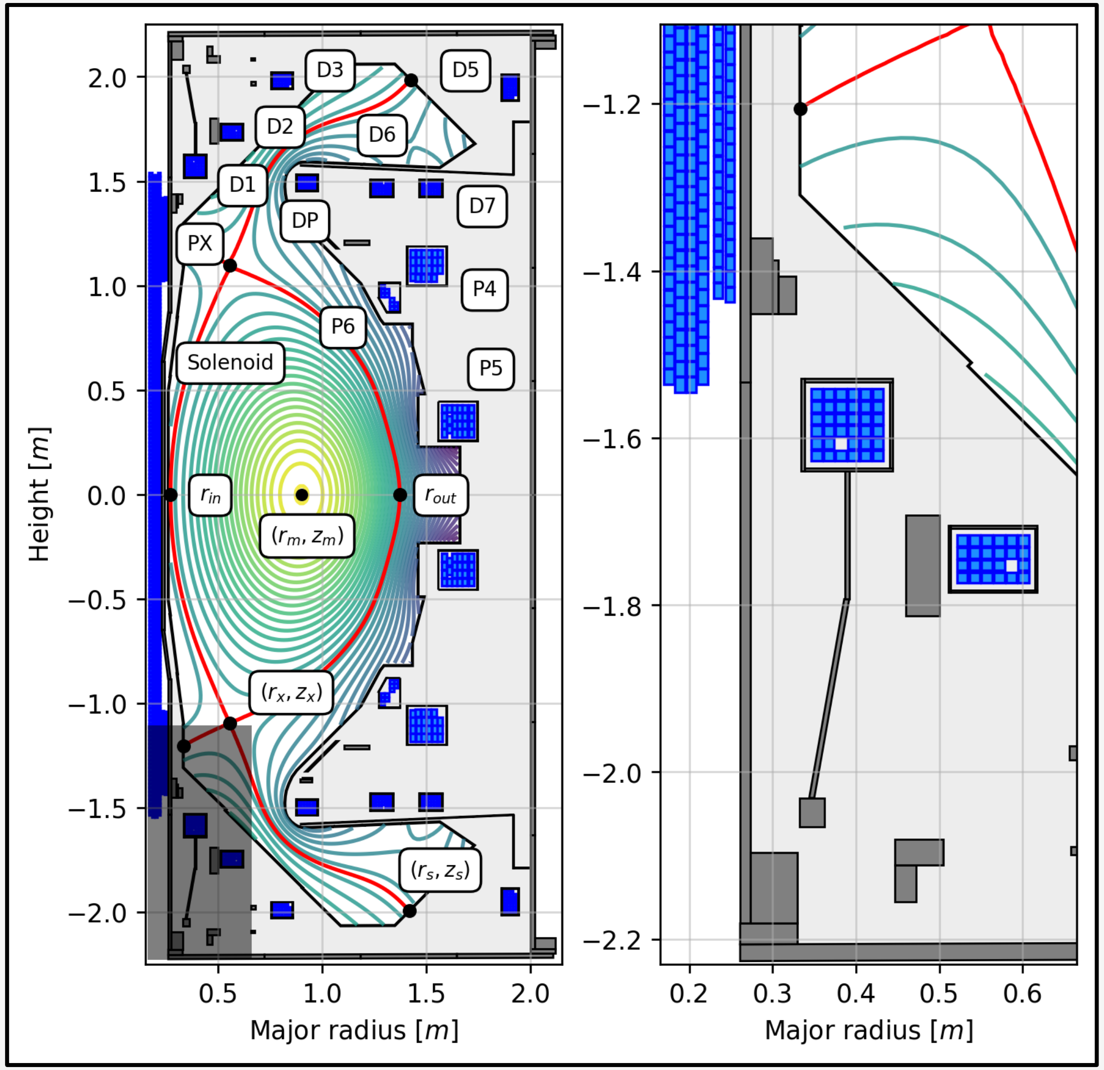
$$\frac{\partial p}{\partial \tilde{\psi}} = \sum_{i=0}^{n_p} \alpha_i \tilde{\psi}^i - \bar{\alpha}\tilde{\psi}^{n_p+1} \sum_{i=0}^{n_p} \alpha_i$$

$$F\frac{\partial F}{\partial \tilde{\psi}} = \sum_{i=0}^{n_F} \beta_i \tilde{\psi}^i - \bar{\beta}\tilde{\psi}^{n_F+1} \sum_{i=0}^{n_F} \beta_i$$

$$\tilde{\psi} = \frac{\psi - \psi_a}{\psi_b - \psi_a} \in [0, 1],$$

$$\bar{\alpha} = \bar{\beta} = 1$$

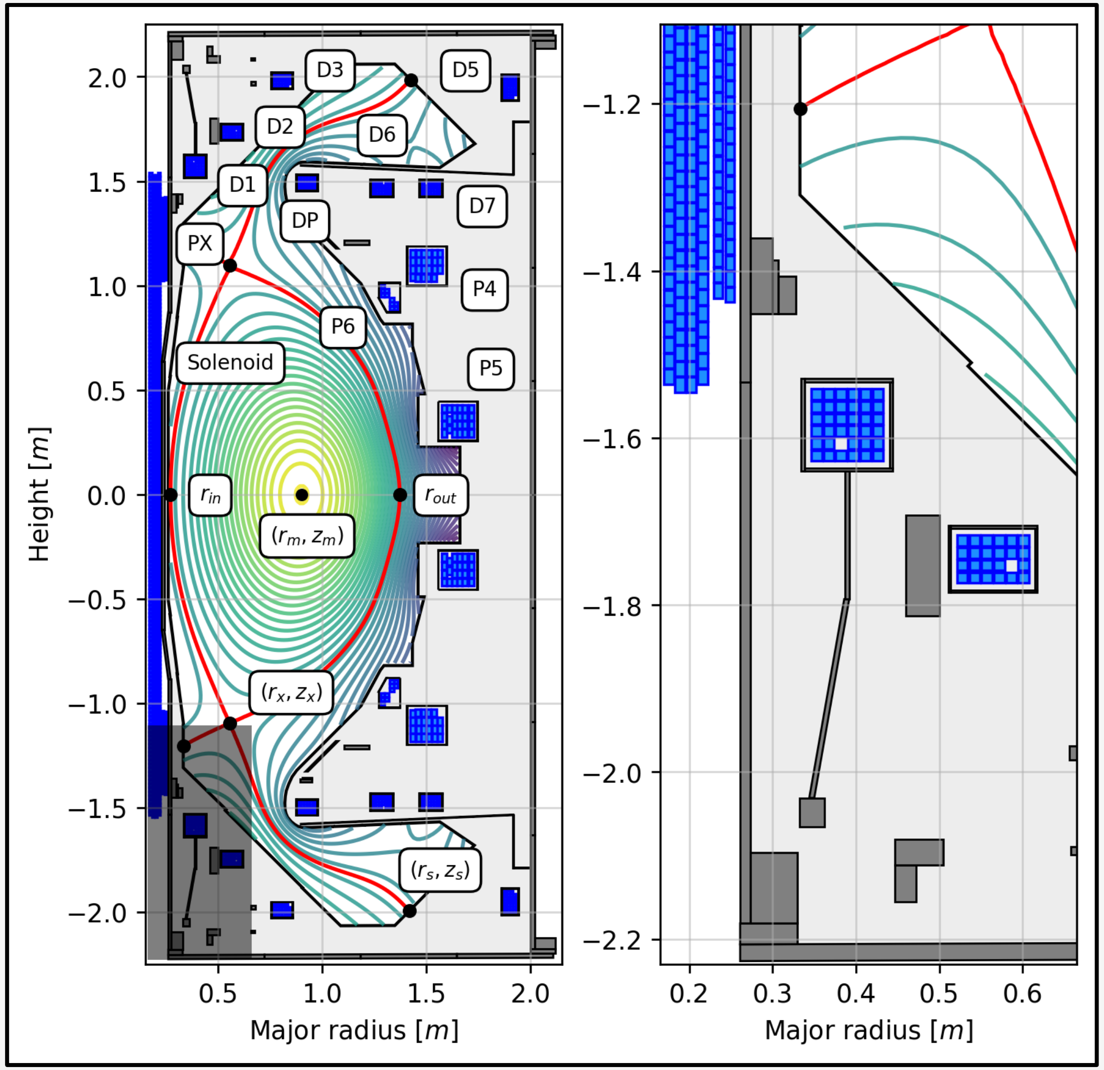
$$n_p = n_F = 1$$



Input parameters (for MAST-U)

Need consistent inputs across all three codes, including:

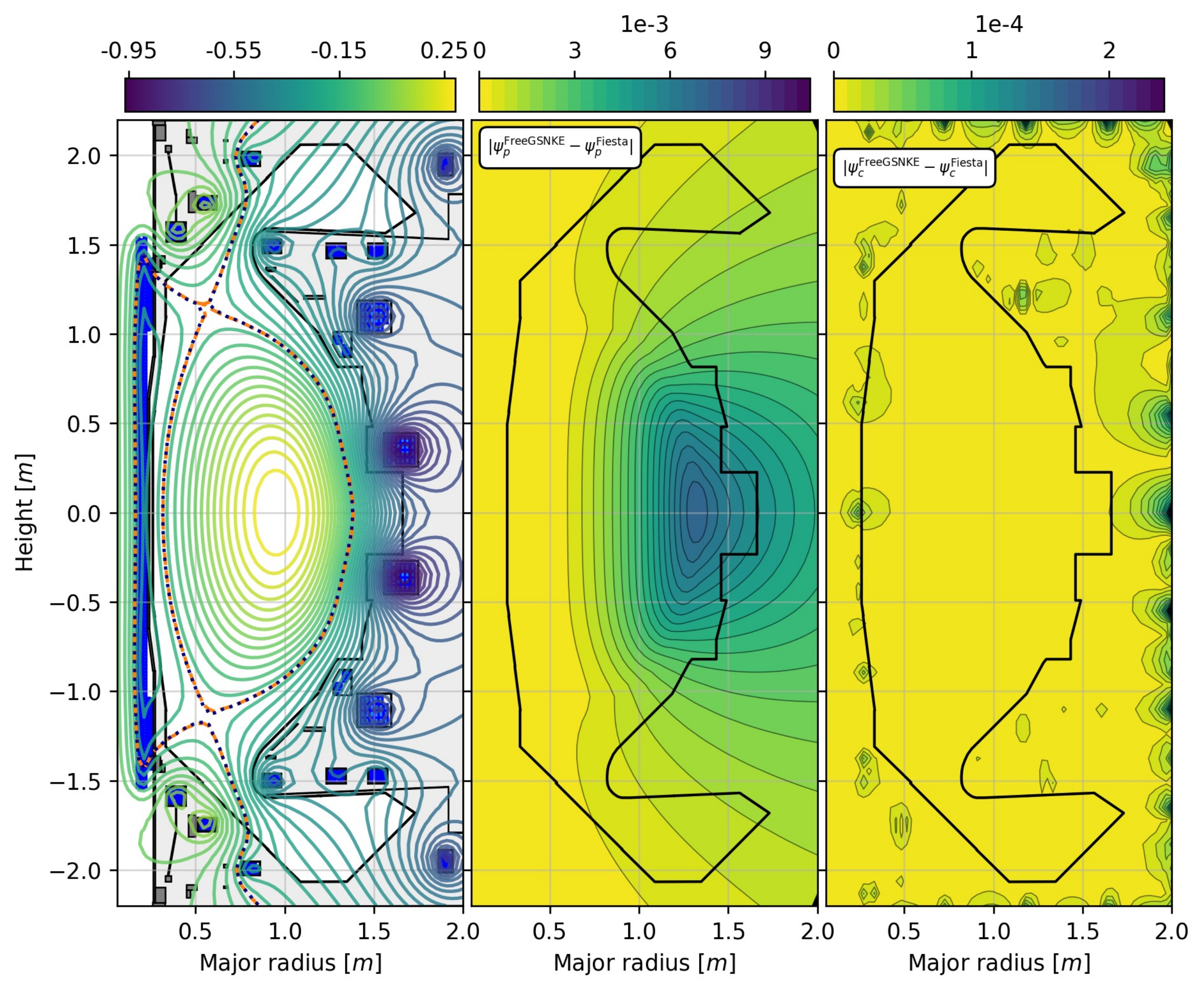
1. Accurate and representative machine description:
   a) Active poloidal field coils.
   b) Passive structures.
   c) Limiter/wall structure.
2. Fitted coil/passive structure currents.
3. Form of plasma profile functions and fitted coefficients.
4. Any additional code-specific parameters:
   - Fiesta requires a *feedback* object to mitigate numerical instability in Picard solver.
     - Uses second nonlinear solver loop to stabilize vertical position (Yoshida et al., 1986).
   - Toroidal field prescription (*irod* in Fiesta, f_{vac} in FreeGSNKE), doesn't affect equilibrium calculation.
   - Fiesta occasionally struggles to converge for some equilibria (poor initial guess/solver instability) so it uses J_p from EFIT++ as initial guess.



Numerical results

Numerical results: shot 45425

- MAST-U shot 45425:
  - Flat-top current 750kA.
  - Double-null shape with conventional divertor configuration.
  - Plasma heated with two NBI (up to 2.5MW).
  - H-mode confinement for most of shot.
- EFIT++ equilibria/targets are the reference.
- FreeGSNKE vs. Fiesta (t = 0.7s):
  - Left: flux contours from FreeGSNKE, separatrices from both match nicely (more later).
  - Centre: abs. diff. in plasma flux:
    - Driven by differences in routines calculating LCFS and J_p.
    - Corner cells by Fiesta BC implementation (see next slide).
  - Right: abs. diff. in conductor flux:
    - Active coil flux difference was $O(10^{-15})$.
    - Difference driven by how codes distribute current in passive structure cross-sections (minor).

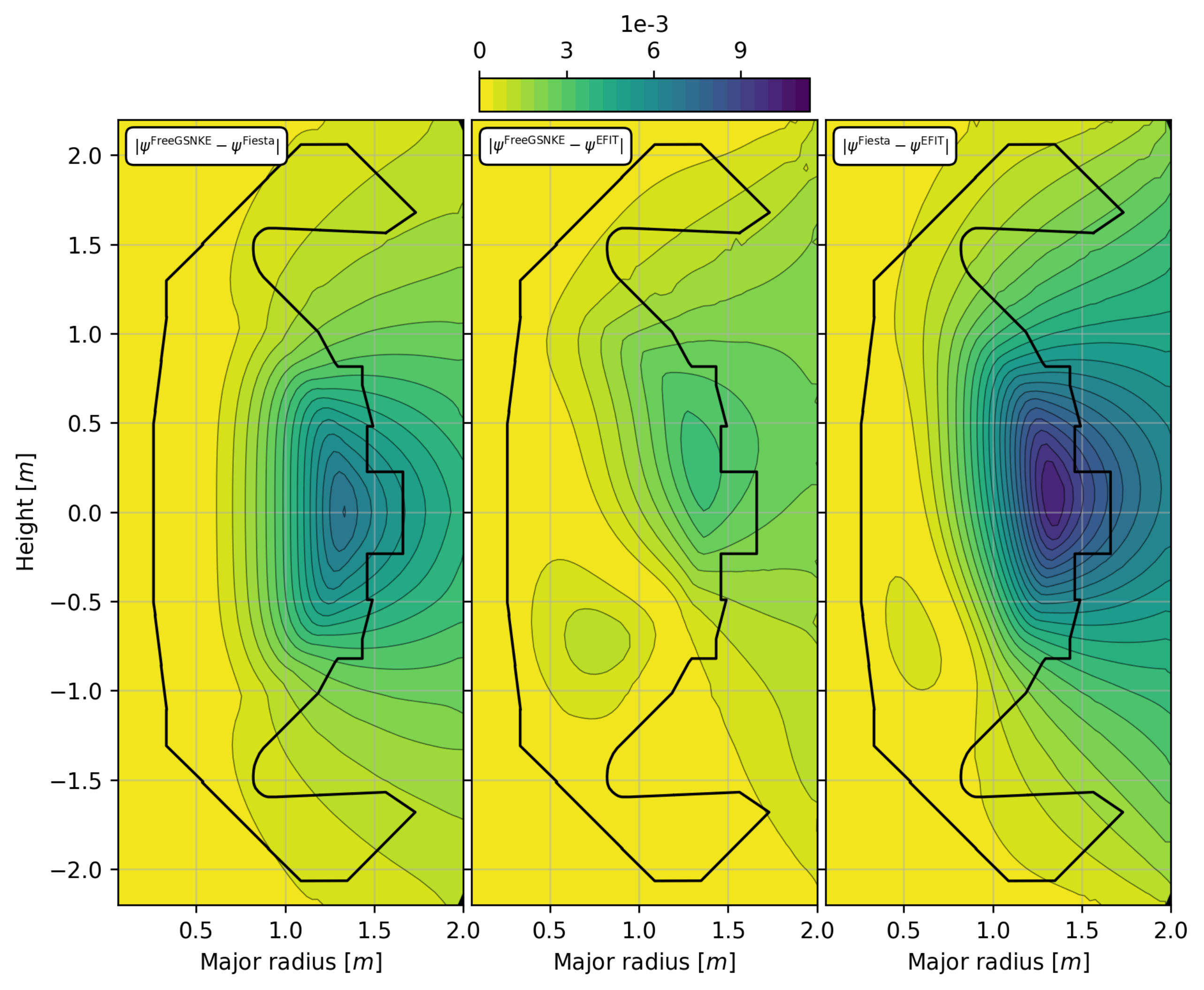


Numerical results: shot 45425

- FreeGSNKE vs. Fiesta **vs. EFIT++** (t = 0.7s):
  - Absolute differences in total flux.
  - Left:
    - similar to what we just saw (ψ_p driven).
  - Centre:
    - Slight asymmetry vs. EFIT++.
    - Again, could be differences in plasma core routines/passive structure implementation.
  - Right:
    - Similar structure to centre as expected.
    - Slightly higher magnitude.
    - Corner cell issue still present.

- As we will see, these levels of difference have negligible impact on shape control targets.

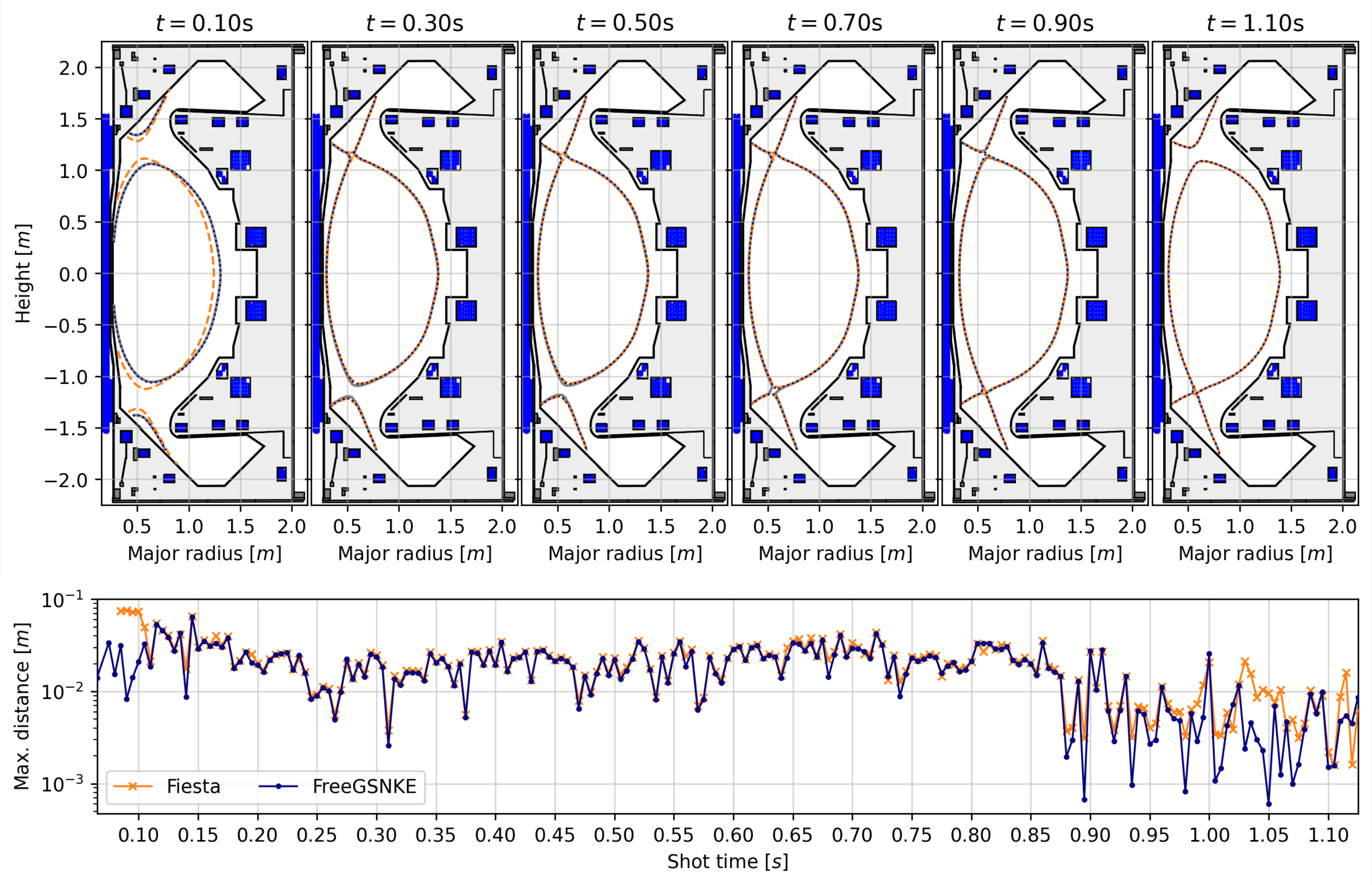
- Note: EFIT++ did not provide breakdown into ψ_p and ψ_c for detailed comparison.



Numerical results: shot 45425

Core separatrices over the shot:

- Top: excellent qualitative match of separatrices over shot.
- Bottom: excellent match in max. distance between core separatrices:
  - FreeGSNKE: below 3.5cm in 95% of time slices.
  - Fiesta: below 4.2cm in 95% of time slices.



How is this calculated:

- Find 360 (r, z) points on core separatrix of each code (these are evenly spaced in poloidal angle, centered on magnetic axis).
- Measure largest difference between corresponding points.

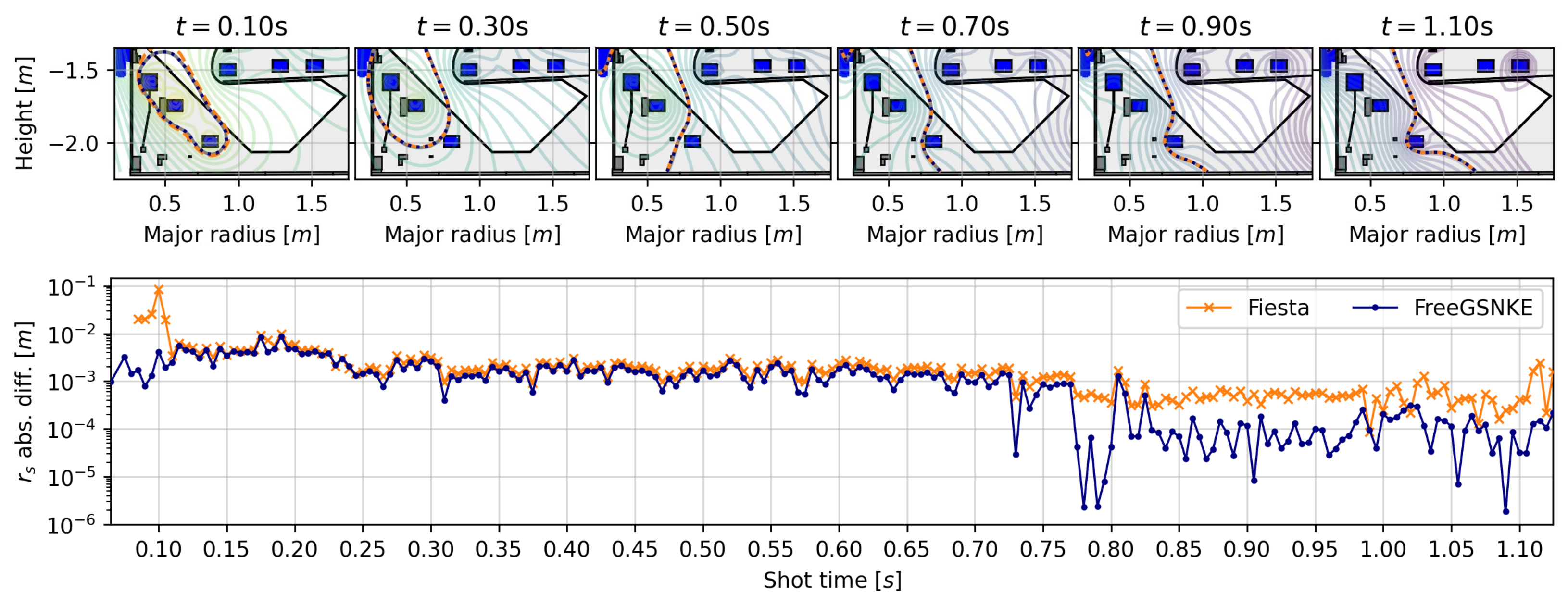
Numerical results: shot 45425

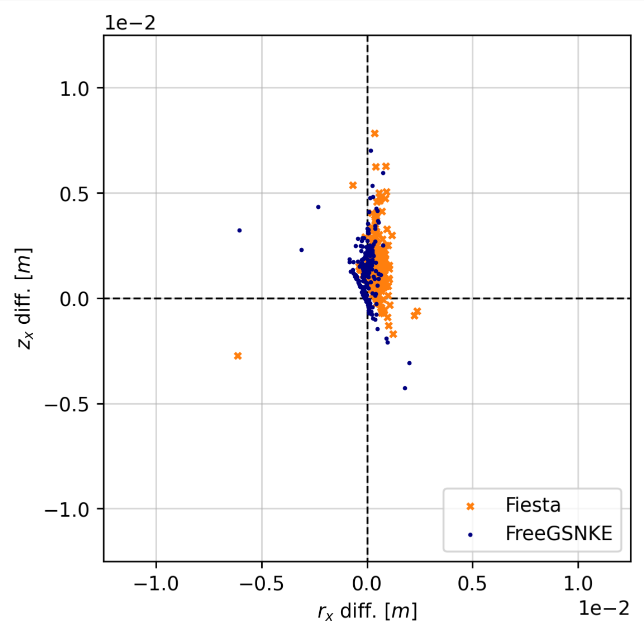
Strikepoints:

- Top: excellent match between lower divertor strikepoints on T1, T2, and T3 tiles (similar for upper).
- Bottom: sub-centimetre radial accuracy over the shot.
- Same for z_s coordinate (not shown, looks identical to r_s).

X-points:

- Sub-centimetre match in lower core chamber X-points (r_X, z_X) (small vertical bias).
- 98% (FreeGSNKE) and 93% (Fiesta) are within 0.5cm of EFIT++ over shot (similar for upper X-point).





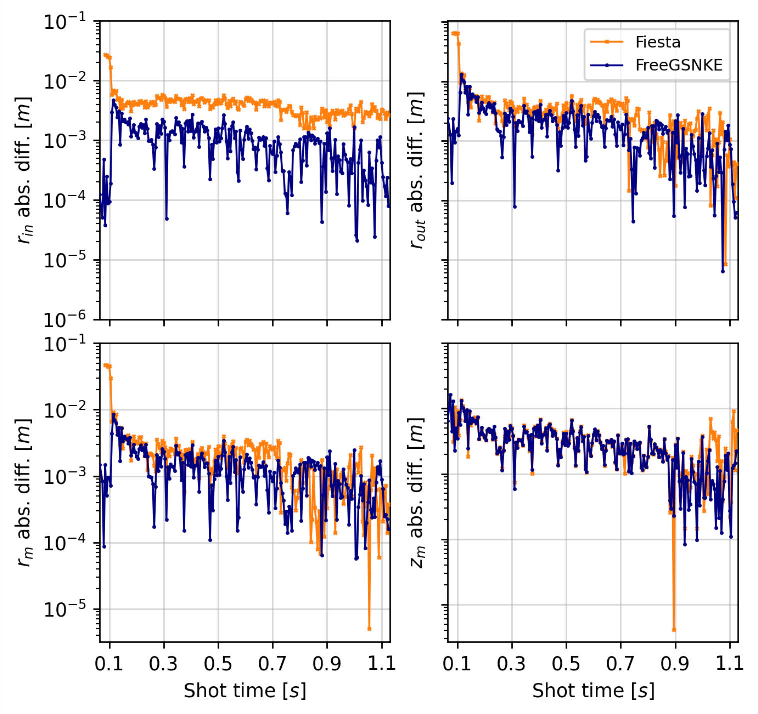
Numerical results: shot 45425

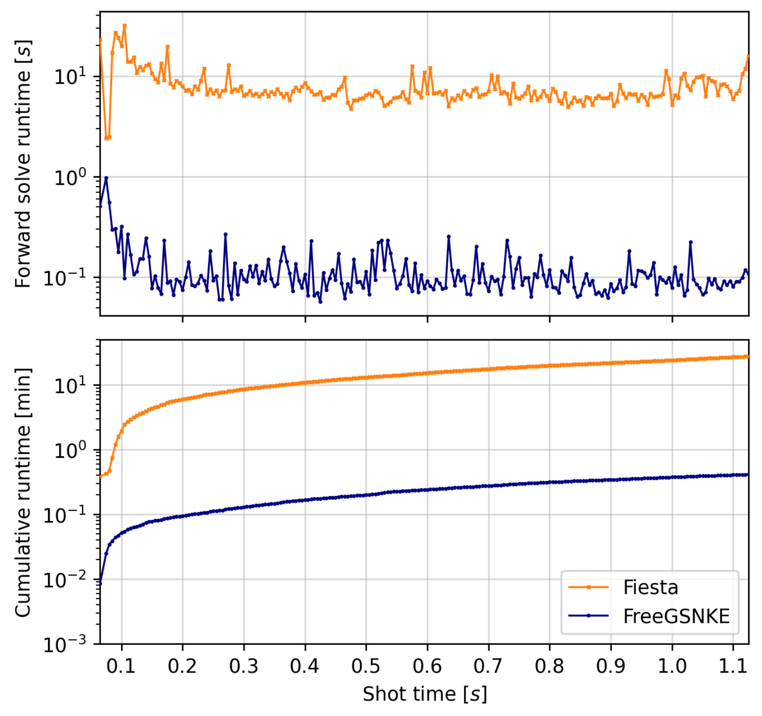
Shape targets:

- ➢ Top: midplane inner (left) and outer (right) radii match to millimetre precision (for both codes).
- ➢ Bottom: magnetic axis (r_m, z_m) are similar.

Runtime comparison:

- ➢ Top: runtime per slice (median is 6.7s for Fiesta, 0.09s for FreeGSNKE).
- ➢ Bottom: cumulative runtime (27 mins Fiesta, 25 secs FreeGSNKE).





Conclusions

Conclusions

Key takeaways:

- Demonstrated that the static forward GS solvers in FreeGSNKE and Fiesta can accurately reproduce equilibria generated by magnetics-only EFIT++ reconstructions on MAST-U.
- We collected/described inputs required for solving static forward problem on MAST-U:
  - ➢ MAST-U machine description now available in FreeGSNKE for modelling.
  - ➢ Both symmetric and non-symmetric active coil setups available.
  - ➢ Option to refine passive structures for better EM modelling.
- Should enable others to benchmark their equilibrium codes against MAST-U equilibria (and our codes).

Future work?

- FreeGSNKE static solver now validated on analytic and real-world equilibria → validate the dynamic solver?
- Can we do some probabilistic (uncertainty-aware) equilibrium reconstruction by combining FreeGSNKE and data assimilation techniques?

Acknowledgements

Co-authors: (UKAEA + Hartree)

N.C. Amorsico, O. El-Zobaidi, S. Etches, A. Agnello, G. K. Holt, C. Vincent, J. Buchanan, S. J. P. Pamela, G. McArdle, L. Kogan, G. Cunningham

Discussions on Fiesta: Stephen Dixon and Oliver Bardsley (UKAEA)

Discussions on FreeGS: Ben Dudson (LLNL)

Funders: FARSCAPE + EPSRC (EP/W006839/1)

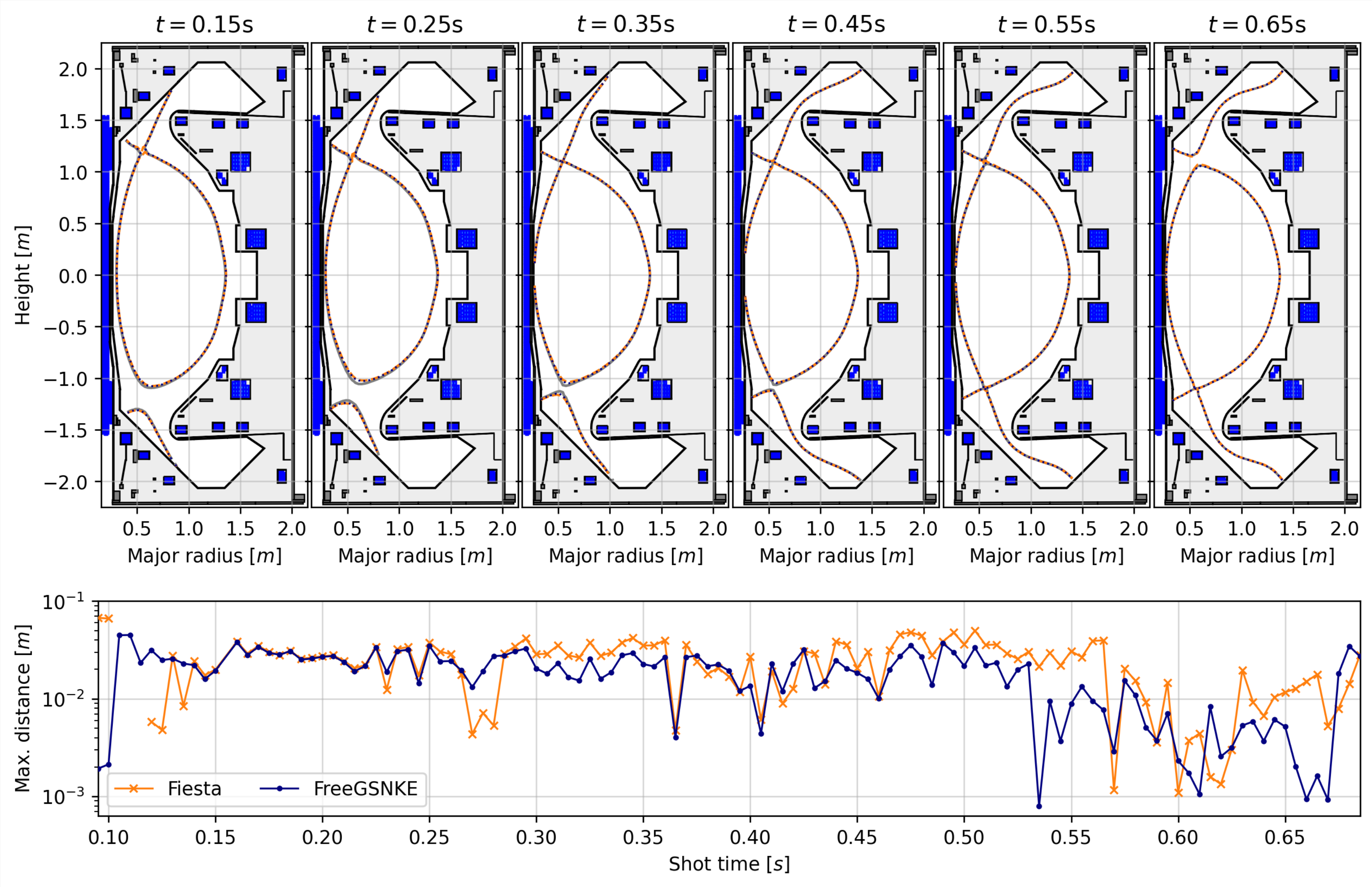
Thanks for listening!

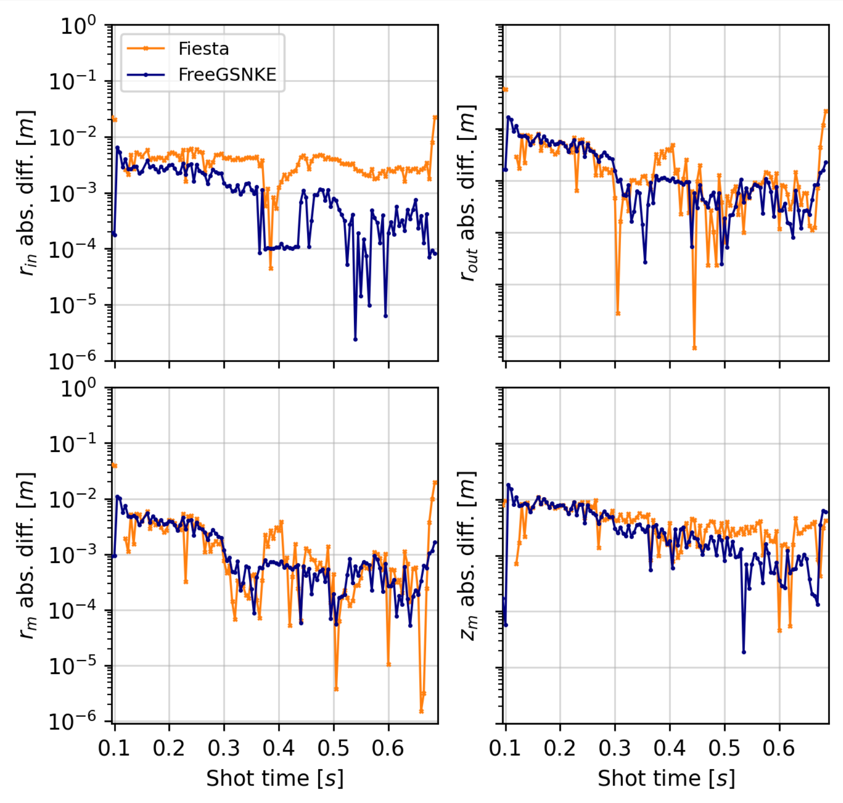
Paper available on Pinboard:

- https://pinboard.ukaea.uk/pinboard-entry/?id=23381&tab=upload
- All feedback welcome!

Additional slides

Numerical results: shot 45292 (Super-X)





Numerical results: shot 45292 (Super-X)