An adaptive Newton-based Grad-Shafranov solver for tokamak equilibrium

Qi Tang[1,2]

[1]School of Computational Science and Engineering, Georgia Institute of Technology

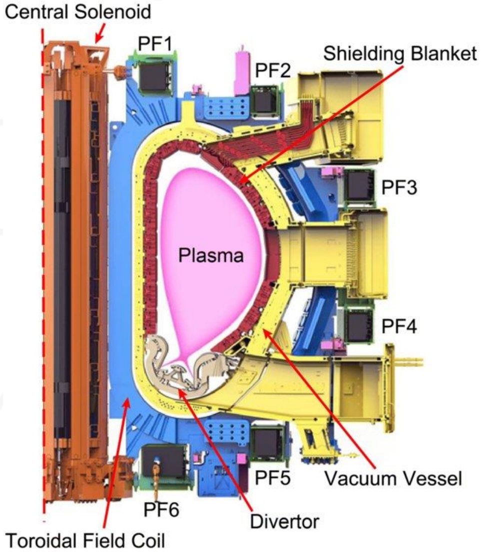
[2]Theoretical Division, Los Alamos National Laboratory

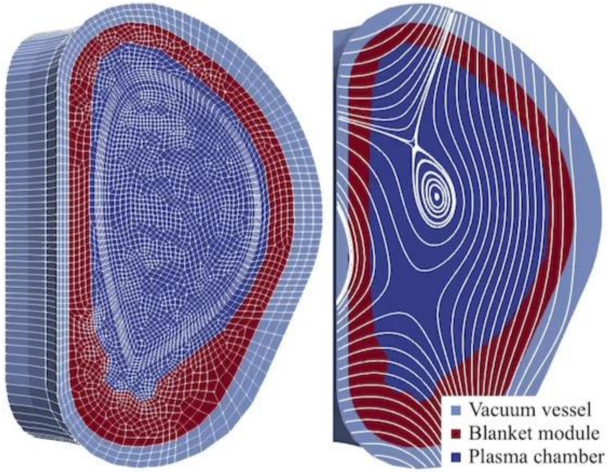
In collaboration with **Dan Serino** (LANL), Xian-Zhu Tang (LANL), Tzanio Kolev (LLNL), Konstantin Lipnikov (LANL)

Georgia Tech

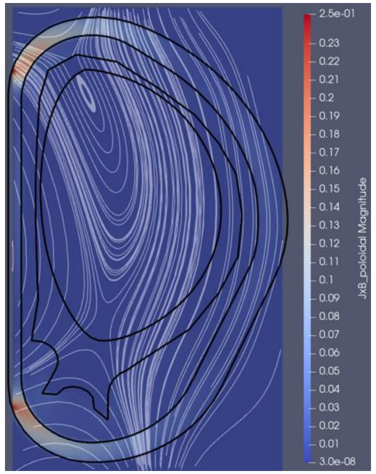
Overview of dynamical MHD and MHD equilibrium solvers developed at LANL/GT

We are working on a series of MHD solvers for tokamak simulations

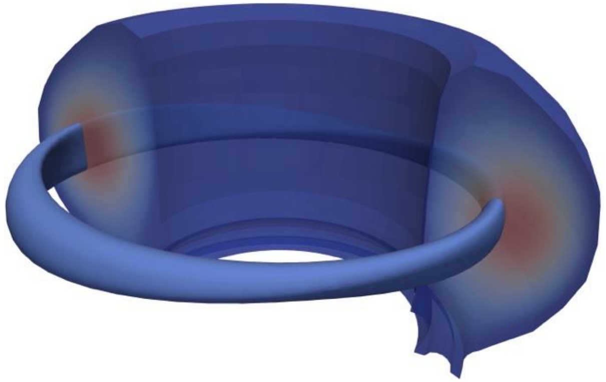




1. Compatible FEM
(G. Wimmer, et al.)

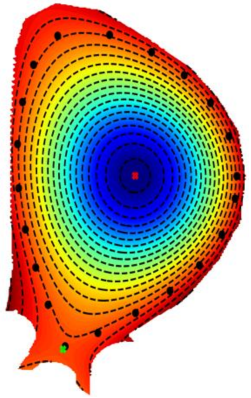


2. Mimetic FD
(Z. Jorti, et al.)

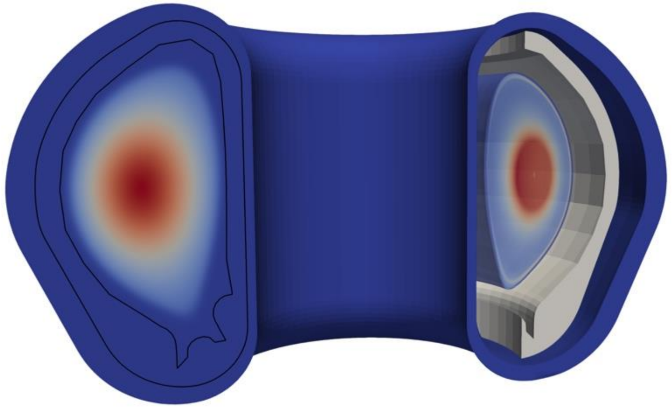


3. Stabilized FEM
(J. Bonilla, et al.)

All of them require a good initial condition: a **cut-cell Picard-based** MHD equilibrium solver



Current workflow



Issues:

- Mismatching between FD structured grid and FEM unstructured grid
- Picard fails to converge for hard cases

Axisymmetric MHD equilibrium leads to Grad-Shafranov equations

Grad-Shafranov equations are derived from MHD force balancing [in (r, φ, z)]:

Force balancing $$J \times B = \nabla p,$$

MHD approx. $$\mu J = \nabla \times B,$$

Tokamak Rep. $$B = \frac{1}{r}\nabla\psi \times e_\phi + \frac{f(\psi)}{r} e_\phi,$$

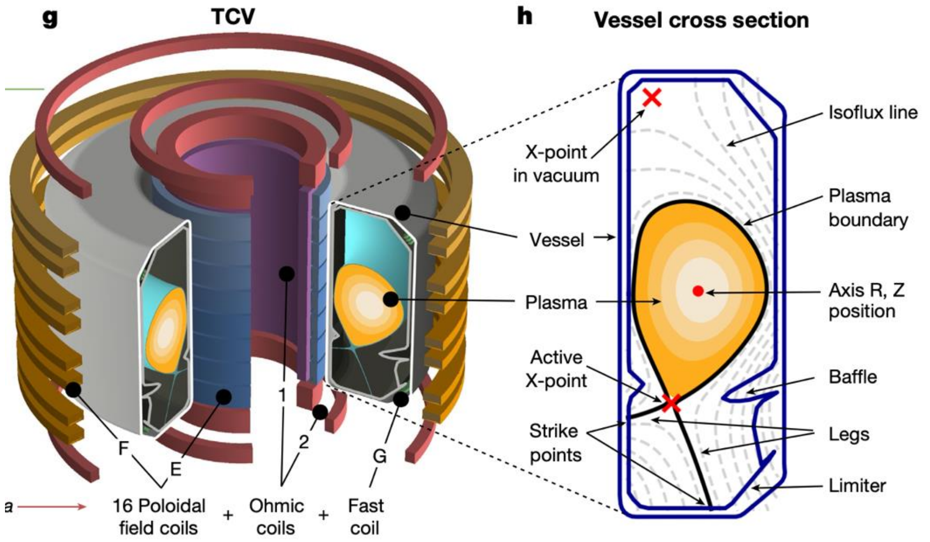
$$\Delta^*\psi := r\partial_r\left(\frac{1}{r}\partial_r\psi\right) + \partial_z^2\psi = -\mu r^2 p'(\psi) - f(\psi)f'(\psi)$$

Therefore, the governing equations become:

$$-\frac{1}{\mu r}\Delta^*\psi = \begin{cases} rp'(\psi) + \frac{1}{r}f(\psi)f'(\psi), & \text{in } \Omega_p(\psi), \\ I_i/|\Omega_{c_i}|, & \text{in } \Omega_{c_i}, \\ 0, & \text{elsewhere in } \Omega_\infty \end{cases}$$

$$\psi(0, z) = 0,$$

$$\lim_{\|(r,z)\|\to+\infty} \psi(r, z) = 0$$

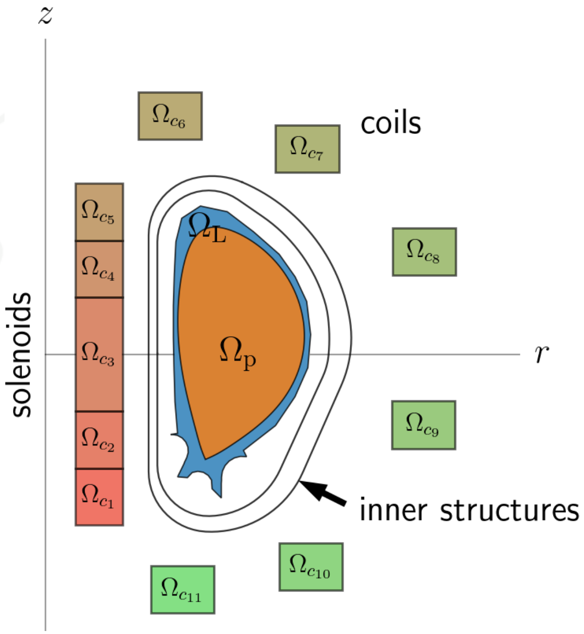


[DeepMind and EPFL, Nature, 2022]

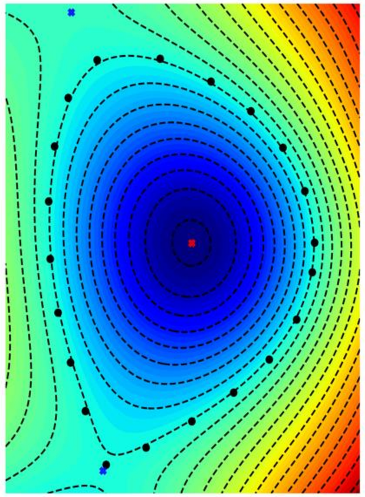
Georgia Tech

Two additional components

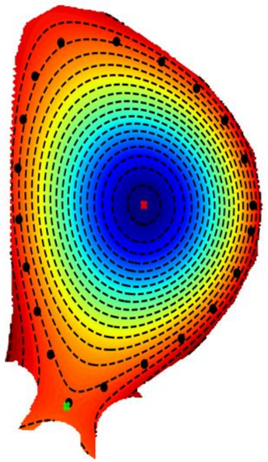
1. It is a free-boundary problem where one can make a choice on the computational boundary.



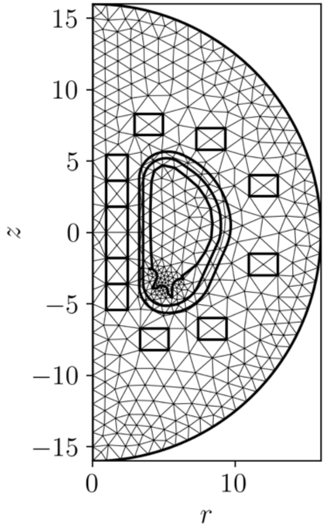
ITER configuration



Conventional FD solver on structured mesh



Our cutcell solver [SISC, 2021]



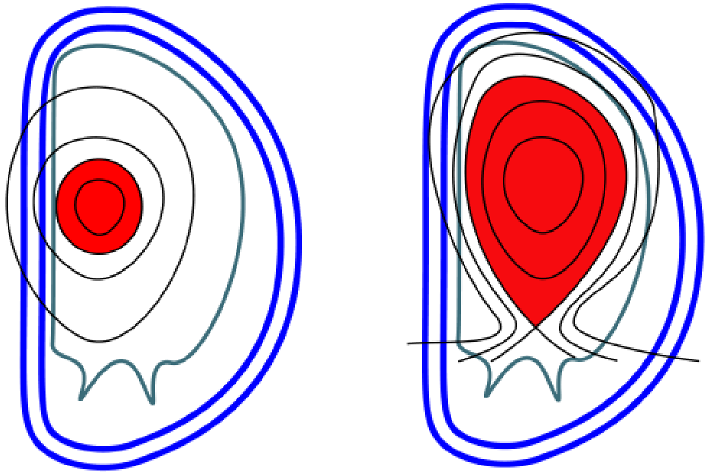
Current work: compatibility and far-field BCs

2. A shape control is needed to ensure the plasma stays inside the chamber:
   1. Control the total current Ip
   2. Control the plasma domain Ω_p by a list of pre-determined control points.

More numerical challenges

- The plasma domain Ω_p is not known a prior and depends on the solution ψ in a nonlinear way:
  1. Non-diverted case: the level set of ψ attaching the wall
  2. Diverted case: the saddle point closest to the magnetic axis

The conventional wisdom among physicists is that it is not possible to perform a Newton solve (not true!).



Two cases for the plasma domain [Heumann et al. JPP, 2015].

- We are interested in the 0-β equilibrium (more challenging):

$$p'(\psi) = 0$$

$$f(\psi) = f_{\mathrm{x}} + \alpha \bar{f}(\psi_N(\psi)),$$

where $\bar{f}$ is a known function given by some experimental measurement table and α is another tunable parameter.

High-level summary of our problem

- A PDE constrained optimization problem
- Target:
  1. Grad-Shafranov PDE
  2. Plasma shape
  3. Total plasma current $I_{\mathrm{p}} = \int_{\Omega_{\mathrm{p}}(\psi)} \left(r p'(\psi) + \frac{1}{\mu r} f f'(\psi) \right) \, dr dz.$
- Control parameters:
  1. α in the source term
  2. a list of current density in the coils: **I**={I1, I2, … }
- Non-conventional components in the solver:
  1. a saddle point search needs to be performed in each iteration (determined based on the sign changes in the local solution difference)
  2. the plasma domain is then determined (a tree search algorithm is performed, populating from the magnetic axis)
  3. analytical Jacobian requires a shape calculus (**discussed later**)
  4. The far-field boundary condition requires a dense double surface integral

Weak formulation of Grad-Shafranov equations

We aim to solve the following nonlinear problem using H1 FEM:

$$\mathrm{a}(\psi, v) = l(\mathbf{I}, v)$$

where **I** stands for the coil current density,

$$\begin{aligned} a(\psi, v) = & \int_\Omega \frac{1}{\mu r} \nabla\psi \cdot \nabla v \, drdz \\ & - \int_{\Omega_p(\psi)} \left(r p'(\psi) + \frac{1}{\mu_0 r} f f'(\psi) \right) v \, drdz \\ & + \frac{1}{\mu_0} \int_\Gamma \psi(\mathbf{x}) N(\mathbf{x}) v(\mathbf{x}) dS(\mathbf{x}) \\ & + \frac{1}{2\mu_0} \int_\Gamma \int_\Gamma (\psi(\mathbf{x}) - \psi(\mathbf{y})) M(\mathbf{x}, \mathbf{y}) (v(\mathbf{x}) - v(\mathbf{y})) dS(\mathbf{x}) dS(\mathbf{y}), \end{aligned}$$

$$l(\mathbf{I}, v) = \sum_{i=1}^{N} \frac{I_i}{|\Omega_{C_i}|} \int_{\Omega_{C_i}} v \, drdz.$$

$$M(\mathbf{x}, \mathbf{y}) = \frac{k_{\mathbf{x},\mathbf{y}}}{2\pi(\mathbf{x}_r \mathbf{y}_r)^{3/2}} \left(\frac{2 - k_{\mathbf{x},\mathbf{y}}^2}{2 - 2k_{\mathbf{x},\mathbf{y}}^2} E(k_{\mathbf{x},\mathbf{y}}) - K(k_{\mathbf{x},\mathbf{y}}) \right),$$

$$N(\mathbf{x}) = \frac{1}{\mathbf{x}_r} \left(\frac{1}{\delta_+} + \frac{1}{\delta_-} - \frac{1}{\rho_\Gamma} \right),$$

$$\delta_\pm = \sqrt{\mathbf{x}_r^2 + (\rho_\Gamma \pm \mathbf{x}_z)^2},$$

$$k_{\mathbf{x},\mathbf{y}} = \sqrt{\frac{4\mathbf{x}_r \mathbf{y}_r}{(\mathbf{x}_r + \mathbf{y}_r)^2 + (\mathbf{x}_z - \mathbf{y}_z)^2}}.$$

A custom integrator is developed to support this double integral on surface (thanks to Veselin)

Georgia Tech

Analytical Jacobian requires shape calculus

- Is it possible to compute the analytical Jacobian for such a complicated case (without AD)? This has been addressed in [Heumann et al. JPP, 2015].
- The idea is based on shape calculus. Consider Gateaux derivative:

$$d_\psi f(\psi, \phi) = \lim_{\epsilon \to 0} \frac{f(\psi + \epsilon\phi) - f(\psi)}{\epsilon} = \frac{d}{d\epsilon} f(\psi + \epsilon\phi)\Big|_{\epsilon=0}$$

  where f is an **integral of over the domain** $\Omega = \Omega(\epsilon)$. The evaluation of this derivative can utilize the following identity (Leibniz integral rule)

$$\frac{d}{d\epsilon} \int_{\Omega(\epsilon)} g(\mathbf{x}, \epsilon)\, dV = \int_{\Omega(\epsilon)} \frac{\partial}{\partial \epsilon} g(\mathbf{x}, \epsilon)\, dV + \int_{\partial\Omega(\epsilon)} g(\mathbf{x}, \epsilon)\mathbf{n} \cdot \mathbf{v}\, d\Gamma,$$

  where $\mathbf{v} = \frac{d\mathbf{x}}{d\epsilon}$ is the velocity of the boundary.
- **n.v** can be computed from the fact that the plasma boundary point **x** is implicitly defined by

$$\psi(\mathbf{x}) + \epsilon\phi(\mathbf{x}) = \psi(\mathbf{x}_\mathrm{x}(\epsilon)) + \epsilon\phi(\mathbf{x}_\mathrm{x}(\epsilon))$$

Georgia Tech

Plasma shape optimization constrained by PDE

- Minimize the squared distance between $\partial\Omega_\mathrm{p}$ and the control points:

$$g = \frac{1}{2}\sum_{i=0}^{N_c}(\psi_N(x_i) - 1)^2$$

note: g is nonlinear

- Regularization on currents:

$$R(I) = \frac{1}{2}\sum_{j=1}^{N_I} w_j I_j^2,$$

- The full optimization problem:

$$\begin{aligned} \min_{\psi, I} \quad & g(\psi) + R(I), \\ \text{s.t.} \quad & B(\psi) - F(I) = 0, \\ & I_p + \int_{\Omega_p(\psi)} \frac{1}{\mu_0 r} f f'(\psi)\, dr dz = 0. \end{aligned}$$

constraint: Grad-Shafranov PDE

constraint: control plasma current

Nonlinear system

- The problem can be casted into the following nonlinear system

$$\begin{aligned} \min_{y,u,\alpha} \quad & G(y) + R(u), \\ \text{s.t.} \quad & B(y,\alpha) - Fu = 0, \\ & C(y,\alpha) - I_p = 0, \end{aligned}$$

  where y stands for the numerical solution, u stands for the current, and α is the tunable scalar parameter in the RHS of f.

- The corresponding Lagrangian is

$$\mathcal{L} = G(y) + R(u) + p^T(B(y,\alpha) - Fu) + \lambda(C(y,\alpha) - I_p)$$

  where p and λ are Lagrange multipliers.

Linearized system and block factorization

We obtain the following linearized system

$$\begin{bmatrix} G_{yy}(y^n) & 0 & B_y(y^n,,\alpha^n)^T & 0 & C_y(y,\alpha) \\ 0 & H & -F^T & 0 & 0 \\ B_y(y^n,\alpha^n) & -F & 0 & B_\alpha(y,a) & 0 \\ 0 & 0 & B_\alpha(y,\alpha)^T & 0 & C_\alpha(y,\alpha) \\ C_y(y,\alpha)^T & 0 & 0 & C_\alpha(y,\alpha) & 0 \end{bmatrix} \begin{bmatrix} y^{n+1}-y^n \\ u^{n+1}-u^n \\ p^{n+1}-p^n \\ \alpha^{n+1}-\alpha^n \\ \lambda^{n+1}-\lambda^n \end{bmatrix}$$

where u is a small vector, α and λ are scalar, and y and p are **large vectors** of the same size.

The exact block vectorization leads to the following system:

$$\begin{bmatrix} B & A \\ C & B^T \end{bmatrix} \begin{bmatrix} \Delta p \\ \Delta y \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix},$$

where

$$A = G_{yy}, \qquad B = B_y^T - \frac{1}{C_\alpha} C_y B_\alpha^T, \qquad C = -FH^{-1}F^T,$$

$$c_1 = b_1 - \frac{1}{C_\alpha} C_y b_4, \qquad c_2 = b_3 + FH^{-1}b_2 - \frac{1}{C_\alpha} B_\alpha b_5.$$

Laplacian operator with rank-1 perturbation

Preconditioner idea (failed)

We seek a good preconditioner for a small rtol=1e-8.

- Idea 1:

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta p \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}.$$

Pro: symmetric (but not positive definite)

Cons: 1. A and C are very low-rank with huge values;
2. Schur complement is not SPD (with a regularization term A+βI).

- Idea 2:

$$\begin{bmatrix} A & B \\ -B^T & -C \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta p \end{bmatrix} = \begin{bmatrix} a_1 \\ -a_2 \end{bmatrix}.$$

Pro: it is a generalized saddle point problem and appears to be compatible with Hermitian and skew-Hermitian splitting (HSS) preconditioner.

Con: HSS did not work when the off-diagonal B is heavy.

Georgia Tech

Preconditioner ideas (succeeded)

This reordered system behaves better

$$\begin{bmatrix} \boxed{B} & A \\ C & \boxed{B^T} \end{bmatrix} \begin{bmatrix} \Delta p \\ \Delta y \end{bmatrix} = \begin{bmatrix} a_2 \\ a_1 \end{bmatrix}.$$

- Idea 3:

$$\mathcal{P}^{-1} = \mathrm{AMG}\left(\begin{bmatrix} B & A \\ C & B^T \end{bmatrix}\right)$$

Averaged: 90 outer linear iterations (90 V-cycles with a large system)

- Idea 4:

$$\mathcal{P}^{-1} = \begin{bmatrix} \mathrm{AMG}(B) & 0 \\ 0 & \mathrm{AMG}(B^T) \end{bmatrix}$$

Averaged: 40 outer linear iterations (80 V-cycles); Woodbury formula can help a little

Georgia Tech

Preconditioner ideas (succeeded)

- Idea 5:

$$\mathcal{P}^{-1} = \begin{bmatrix} \left(B - A(B^T)^{-1}C\right)^{-1} & 0 \\ 0 & (B^T)^{-1} \end{bmatrix}$$

Averaged results: 30 outer linear iterations BUT with a lot more expensive inner solver ~ 7000+ V-cycles

- Idea 6:

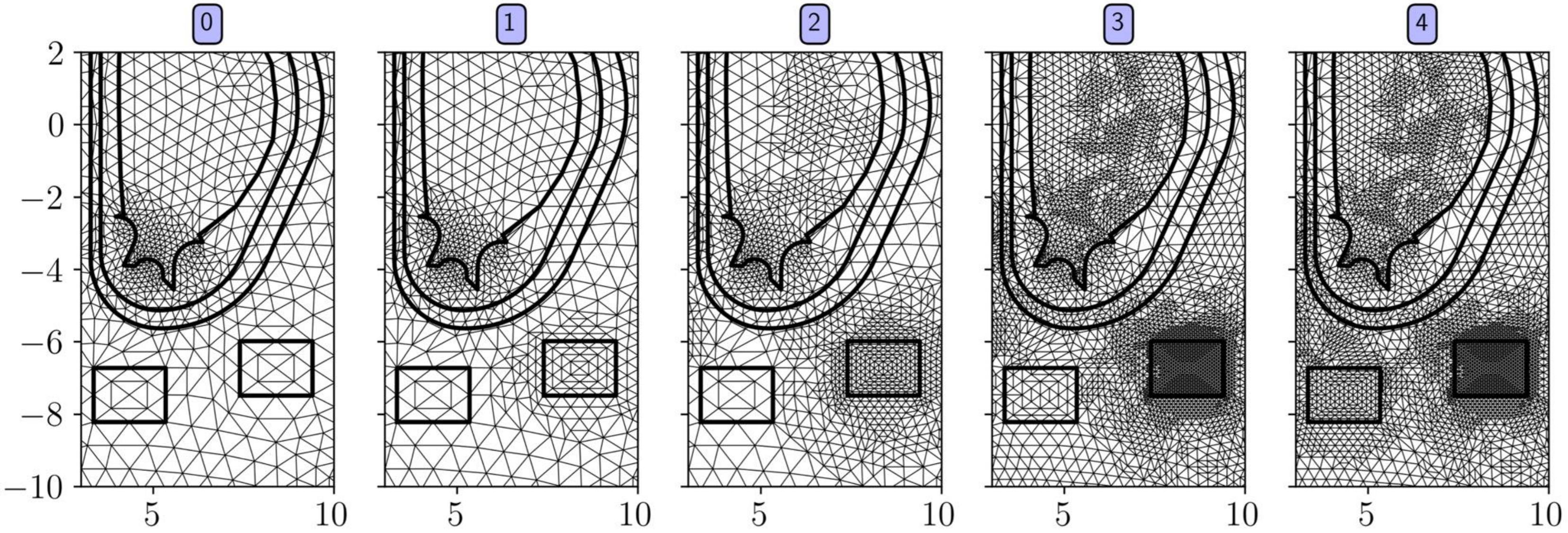
$$\mathcal{P}^{-1} = \begin{bmatrix} \mathrm{AMG}(B) & D \\ 0 & \mathrm{AMG}(B^T) \end{bmatrix},$$

where $D = -\mathrm{AMG}(B)\, A\, \mathrm{AMG}(B^T)$

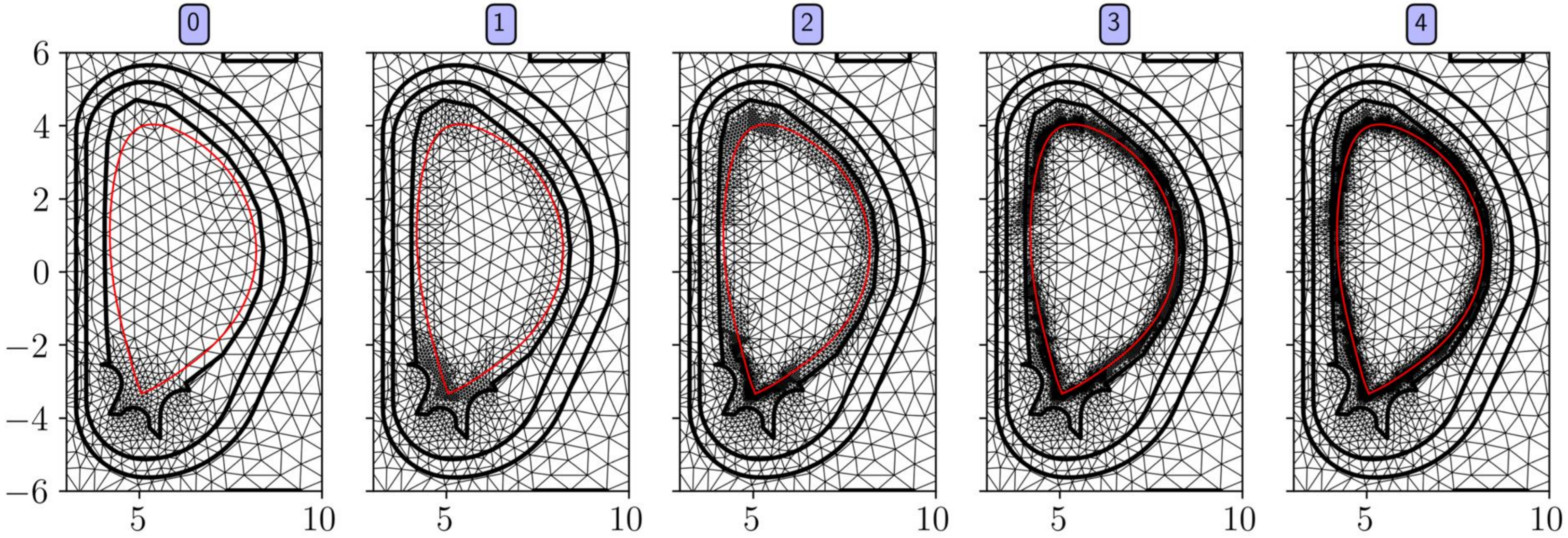
Averaged results: 30 outer linear iterations (60 V-cycles)

Leverage MFEM's conforming AMR

- Error-indicator-based:

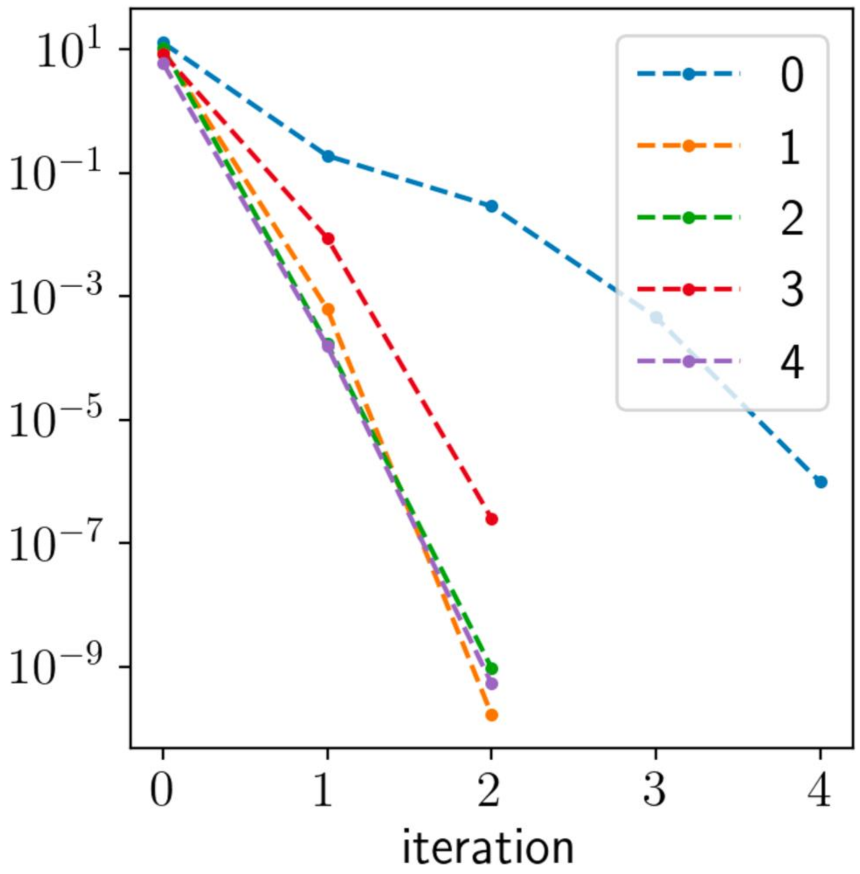


- Feature-based:

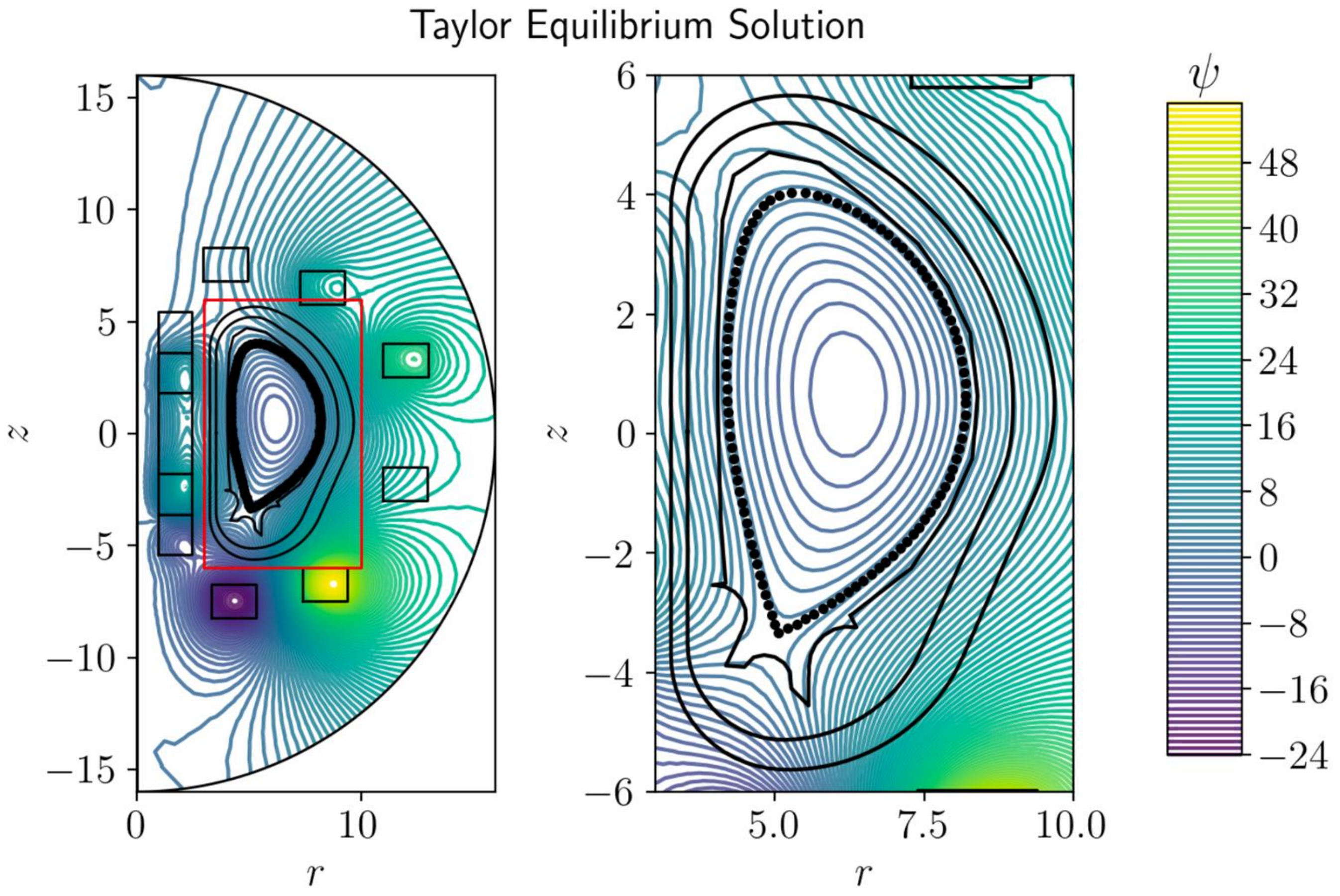


Georgia Tech

Taylor state equilibrium (0-beta)



AMR is found to help the Newton iterations



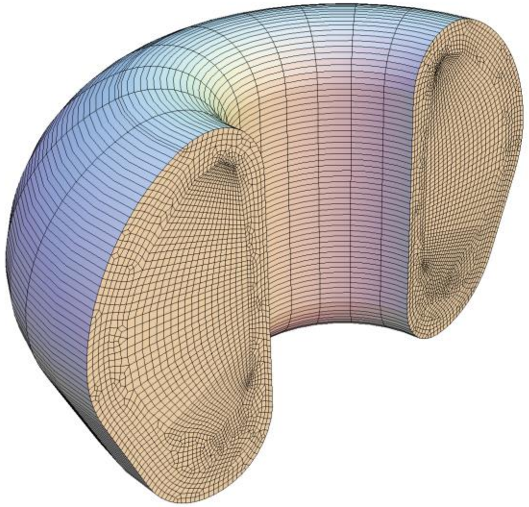
Numerical solution and its zoom-in

Georgia Tech

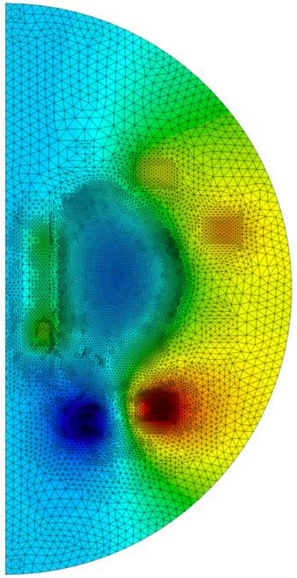
Load the Grad-Shafranov solution to a dynamical MHD solver

The current workflow to build $B = \frac{1}{r}\nabla\psi \times e_\phi + \frac{f(\psi)}{r} e_\phi,$

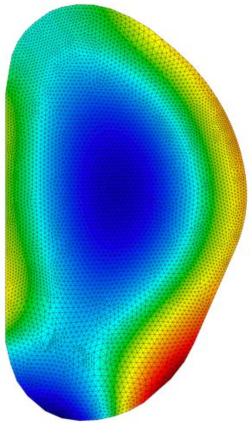
1 **extruder.cpp**



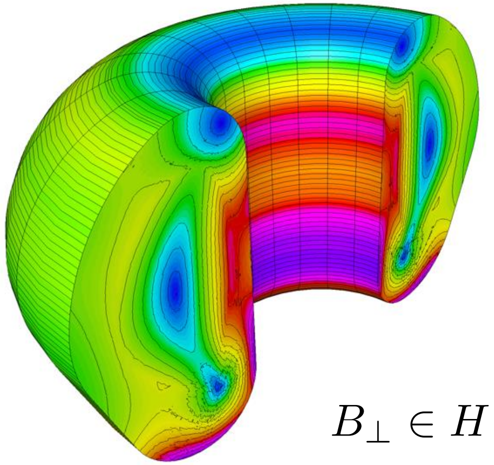
2



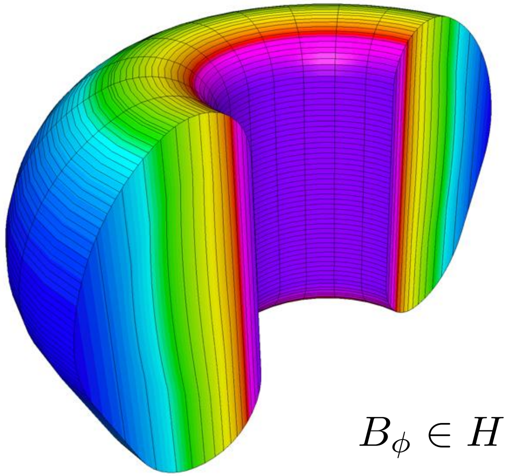
trimmer.cpp
field-interp.cpp



3 **gslib** with compatible FEM
(thanks to Ketan)



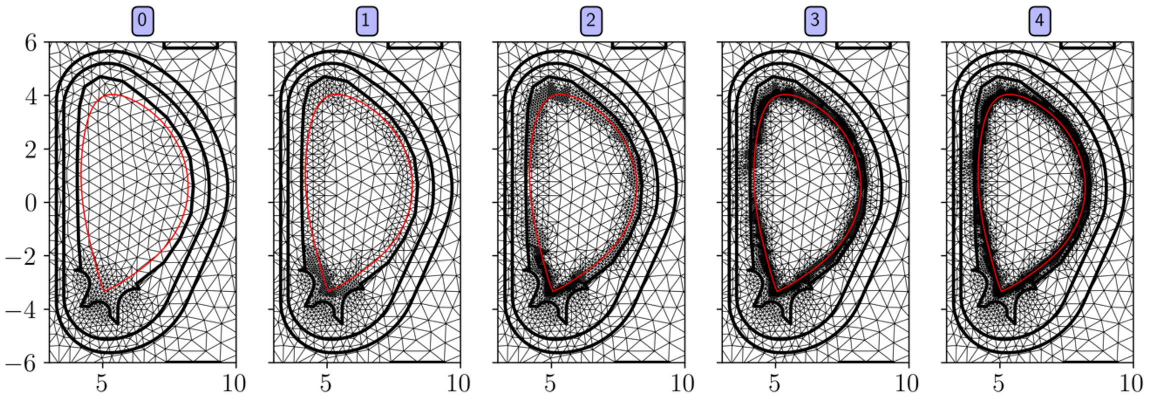
$B_\perp \in H(\mathrm{curl})$

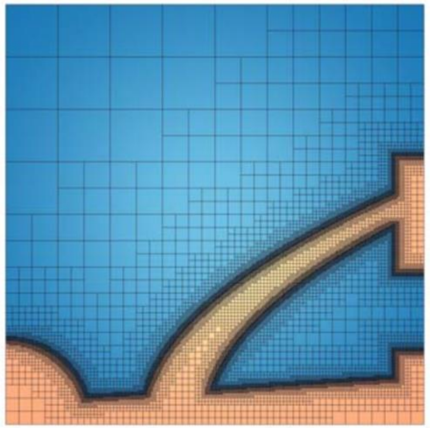


$B_\phi \in H(\mathrm{curl})$

Future improvement

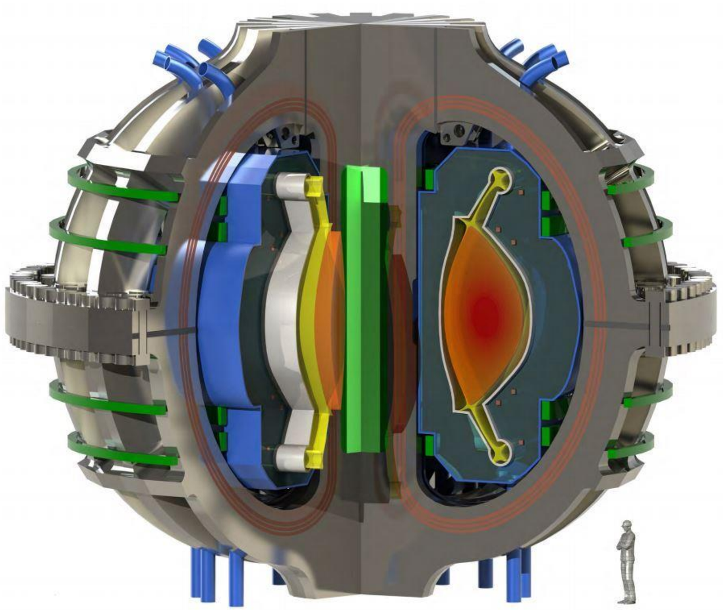
- The mesh should be aligned with the separatrix:

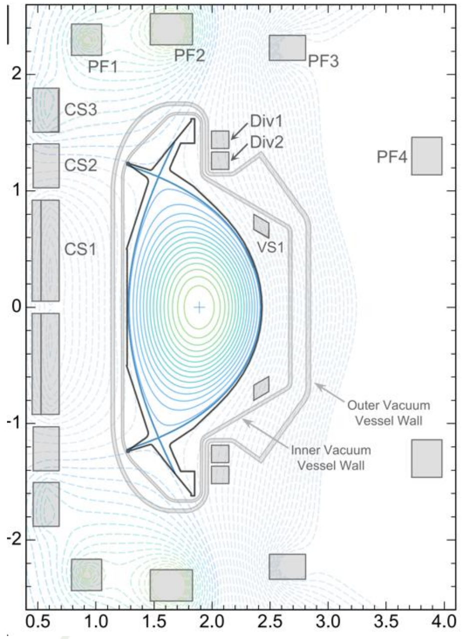




Internal interface fitting on the fly

- **Inequality constraints** may help plasma control in practice. See DeepMind's RL work [Nature 2022].
  E.g.:
  R constraint: good=0.02, bad=0.05
  Z constraint: good=0.02, bad=0.2
- More complicated shape control in the SPARC tokamak:





Conclusions

- We develop an adaptive Newton-based Grad-Shafranov solver for the shape control to seek 0-β tokamak equilibriums.
- Newton solver is much more effective than the Picard-based solver.
- Effective preconditioners for the linearized system have been explored.
- The algorithm is deployed on MFEM with conforming AMR and its flexible solver interface.
- Future work: more complicated shape control (SPARC), requiring a workflow incorporating solvers, AMR, and a **meshing** capability.

Georgia Tech