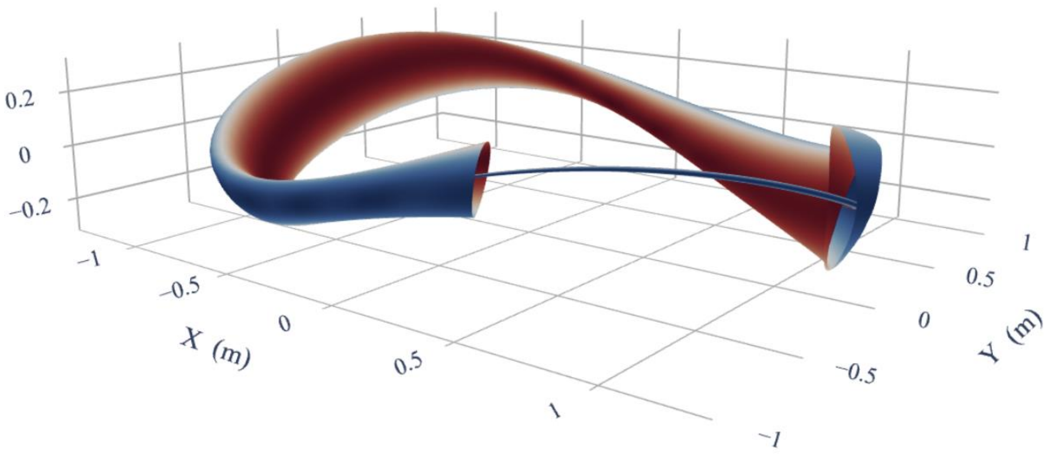
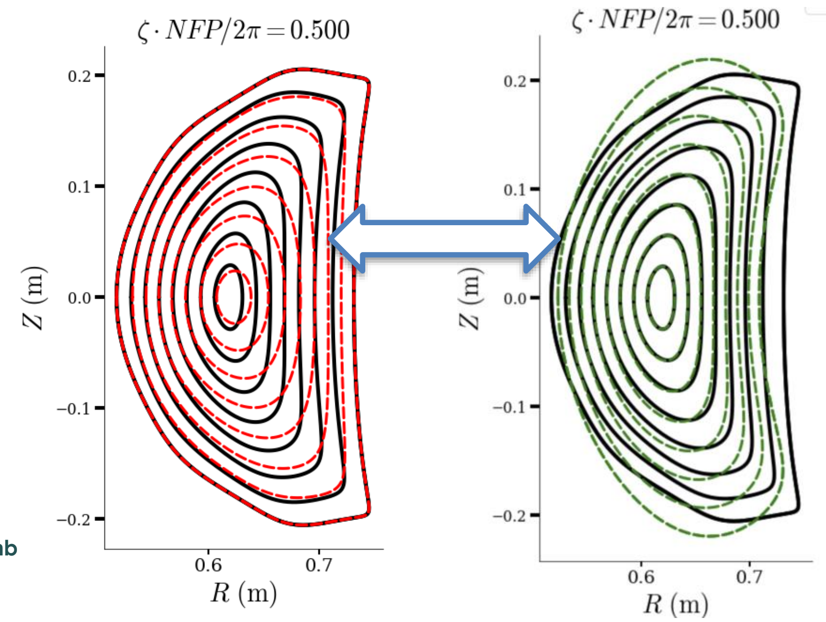
Near-Axis Constrained Equilibria with the DESC code

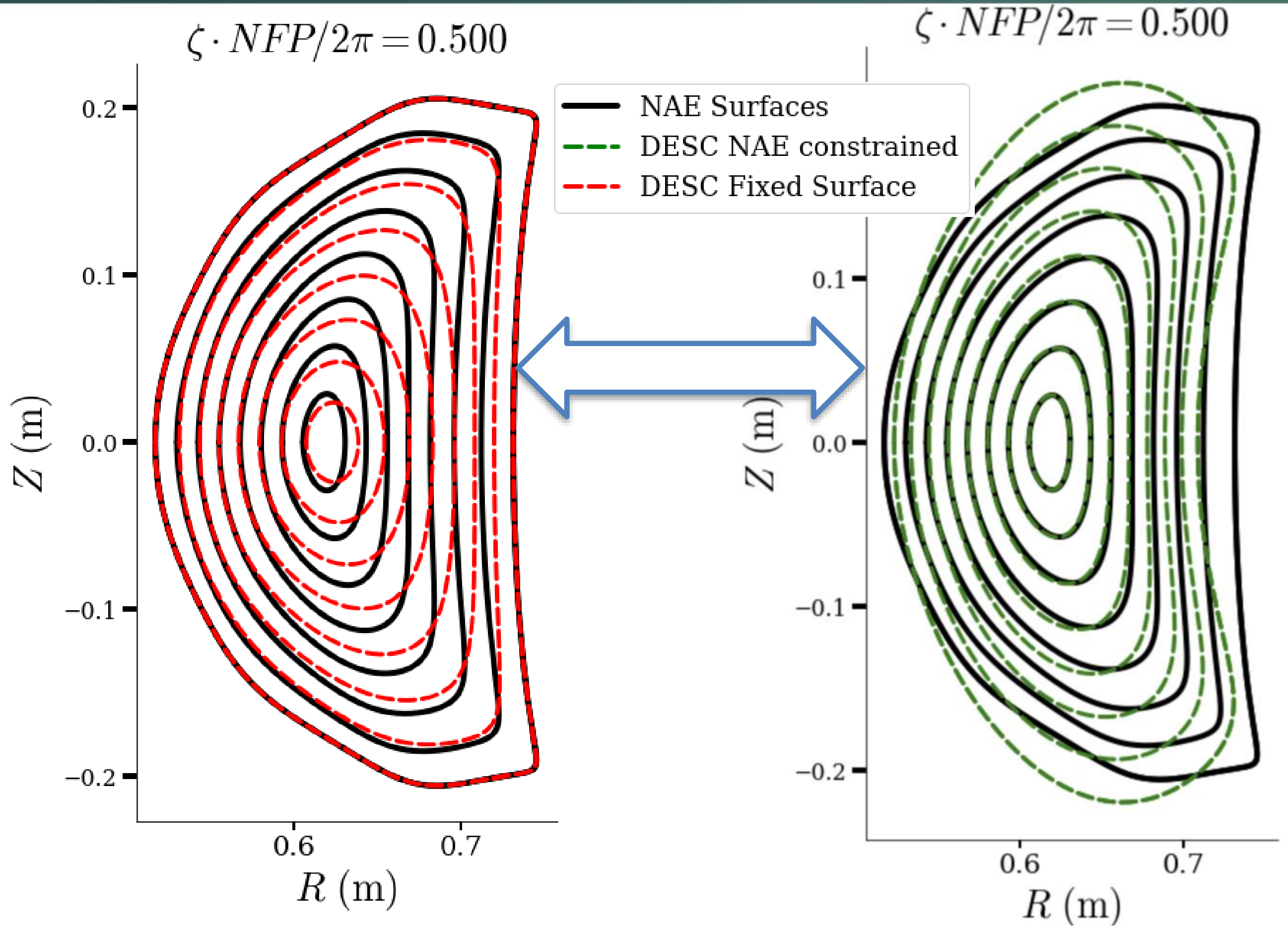
Dario Panici, Daniel Dudt, Rory Conlin, Eduardo Rodriguez, Egemen Kolemen

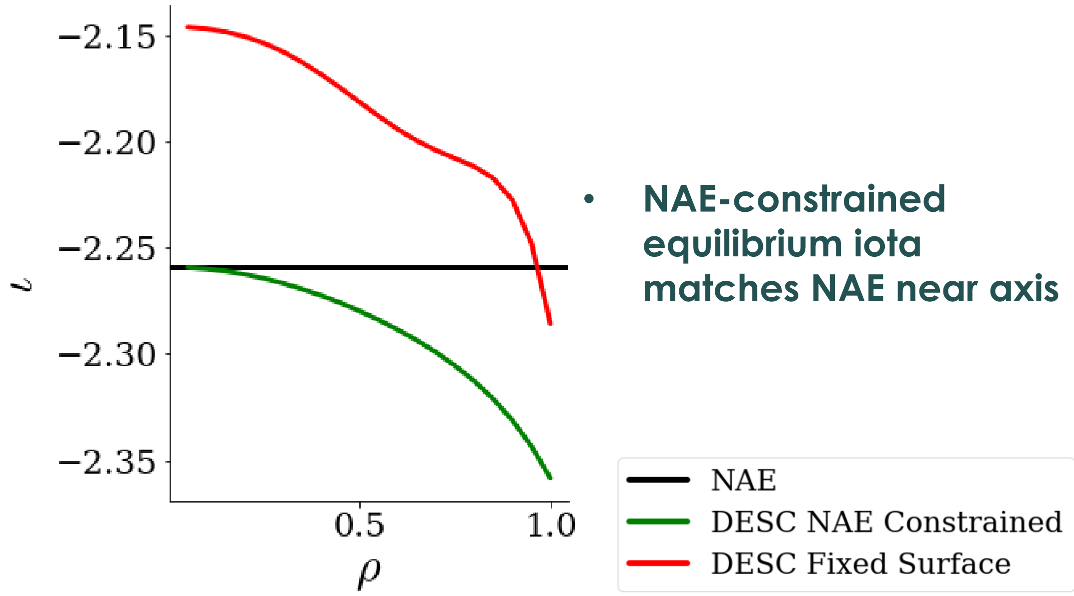




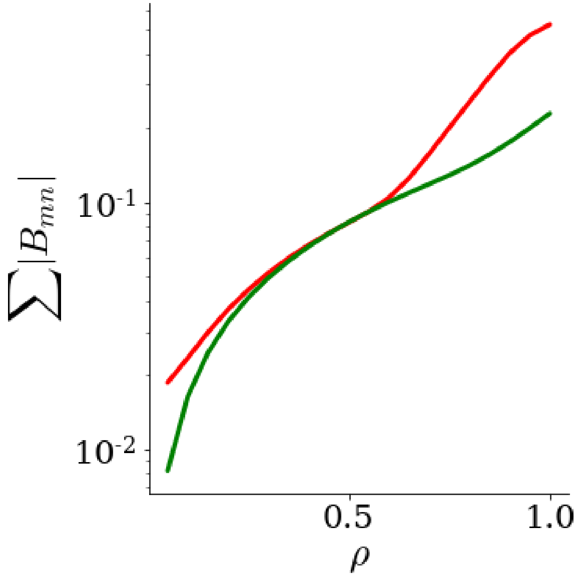
Plasma Control

Near-Axis-Expansion Constrained Equilibria in DESC





- NAE-constrained equilibrium iota matches NAE near axis



- NAE-constrained equilibrium maintains better QS near axis

- Global equilibria solutions with near-axis behavior constrained to match the NAE to O(ρ)
- Enables the connection between global MHD equilibria solutions and the existing insight on optimized stellarators

Plasma Control

Ideal MHD

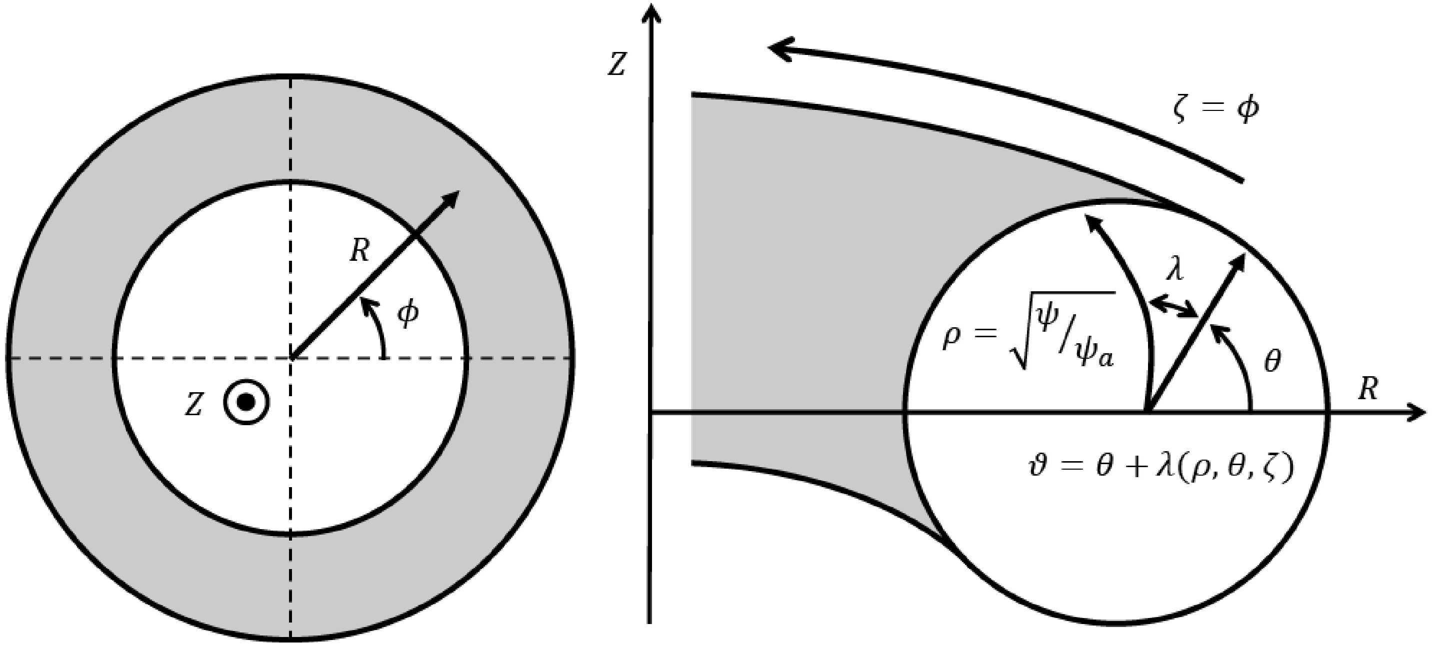
$$\boldsymbol{J} \times \boldsymbol{B} = \nabla p$$

$$\nabla \times \boldsymbol{B} = \mu_0 \boldsymbol{J}$$

$$\nabla \cdot \boldsymbol{B} = 0$$

Inverse Equilibrium

- Cast problem as solving for the locations of the flux surfaces
- Yields 2nd order PDE in (ρ, θ, ζ)



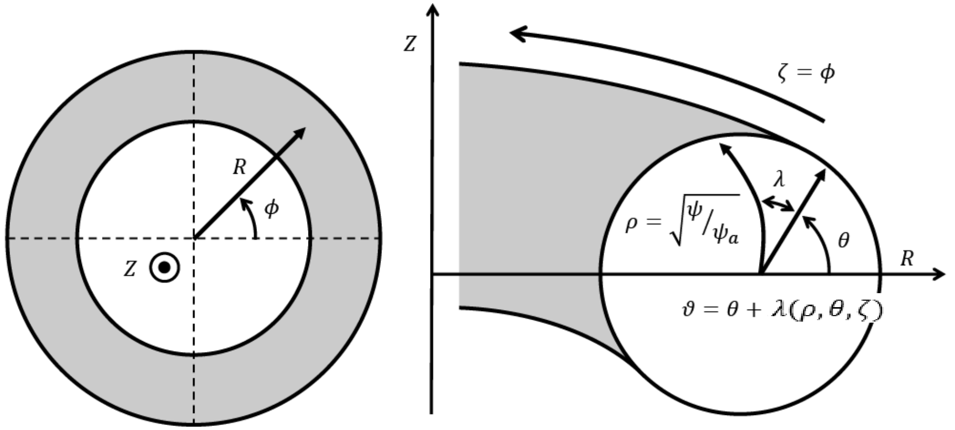
Stellarator Equilibrium - DESC

$$\boldsymbol{F} = \boldsymbol{J} \times \boldsymbol{B} - \nabla p = 0$$

(Dudt and Kolemen 2020)

- 3D Ideal MHD Equilibrium Code
- Assumes Nested Flux Surfaces
- Inverse Equilibrium Problem
- **Minimizes Force Error Directly**
- **Pseudospectral Code**

3D Spectral Representation of $\mathbf{x} = (R, \lambda, Z)$ using Fourier-Zernike Basis



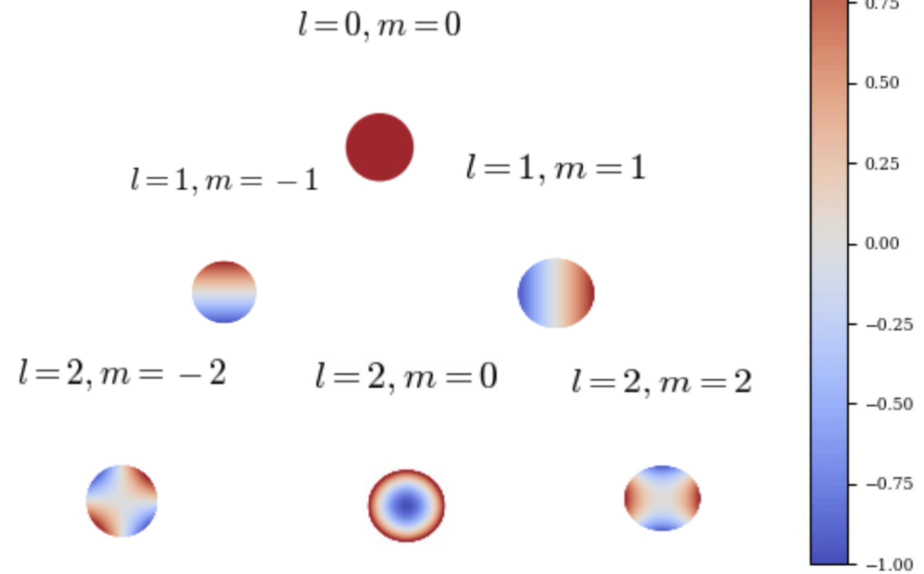
DESC - Fourier-Zernike Spectral Basis

$$R(\rho,\theta,\zeta) = \sum_{m=-M,n=-N,l=0}^{M,N,L} R_{lmn}\mathcal{Z}_l^m(\rho,\theta)\mathcal{F}^n(\zeta)$$

$$\lambda(\rho,\theta,\zeta) = \sum_{m=-M,n=-N,l=0}^{M,N,L} \lambda_{lmn}\mathcal{Z}_l^m(\rho,\theta)\mathcal{F}^n(\zeta)$$

$$Z(\rho,\theta,\zeta) = \sum_{m=-M,n=-N,l=0}^{M,N,L} Z_{lmn}\mathcal{Z}_l^m(\rho,\theta)\mathcal{F}^n(\zeta)$$

Zernike Polynomials in (ρ,θ) plane



$$\mathcal{Z}_m^l(\rho,\theta) = \begin{cases} \mathcal{R}_l^{|m|}(\rho)cos(|m|\theta) & \text{for } m \geq 0 \\ \mathcal{R}_l^{|m|}(\rho)sin(|m|\theta) & \text{for } m < 0 \end{cases}$$

Zernike Polynomials

$$\mathcal{R}_l^{|m|}(\rho) = \sum_{s=0}^{(l-|m|)/2} \frac{(-1)^s(l-s)!}{s!\,[(l+|m|)/2-s]!\,[(l-|m|)/2+s]!}\rho^{l-2s}$$

Radial Polynomial is of degree *l-2s*, and *l=m,m+2,…, L*

$$\mathcal{F}^n(\zeta) = \begin{cases} cos(|n|N_{FP}\zeta) \text{for } n \geq 0 \\ sin(|n|N_{FP}\zeta) \text{for } n < 0 \end{cases}$$

Toroidal Fourier Series [3]

Plasma Control

DESC Allows Flexible Constraints when Defining Equilibrium Problem - Fixed ρ=1 Boundary

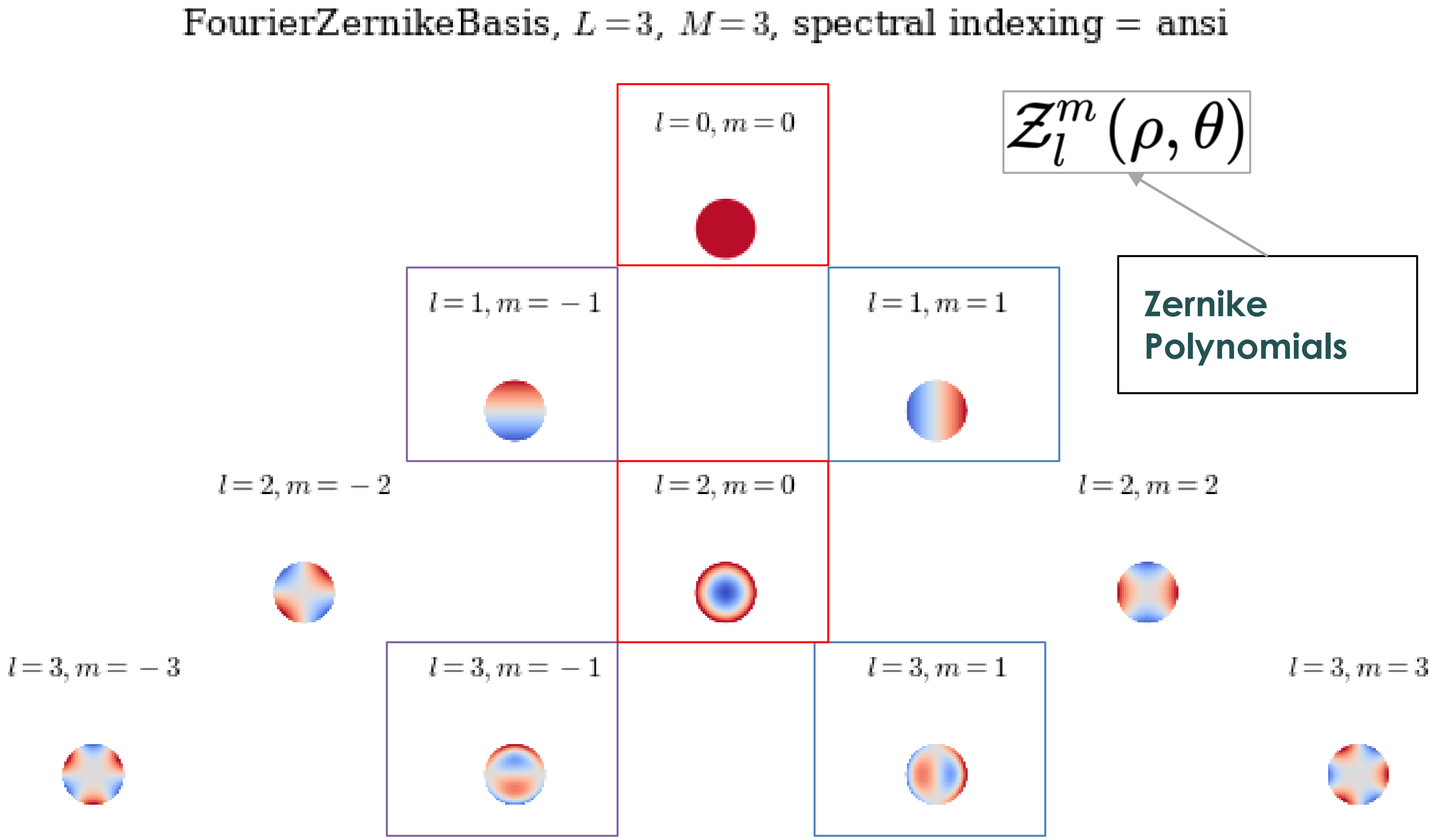
$$R^b(\theta,\zeta) = \sum_{m=0}^{M}\sum_{n=-N}^{N} R^b_{m,n} cos(m\theta - n\zeta)$$

$$Z^b(\theta,\zeta) = \sum_{m=0}^{M}\sum_{n=-N}^{N} Z^b_{m,n} sin(m\theta - n\zeta)$$

<u>Fixed-Boundary ρ=1 Constraint</u>

$$\sum_{l=0}^{L} R_{lmn}\mathcal{Z}_l^m(\rho=1,\theta) = R^b_{mn} \quad \forall m,n$$

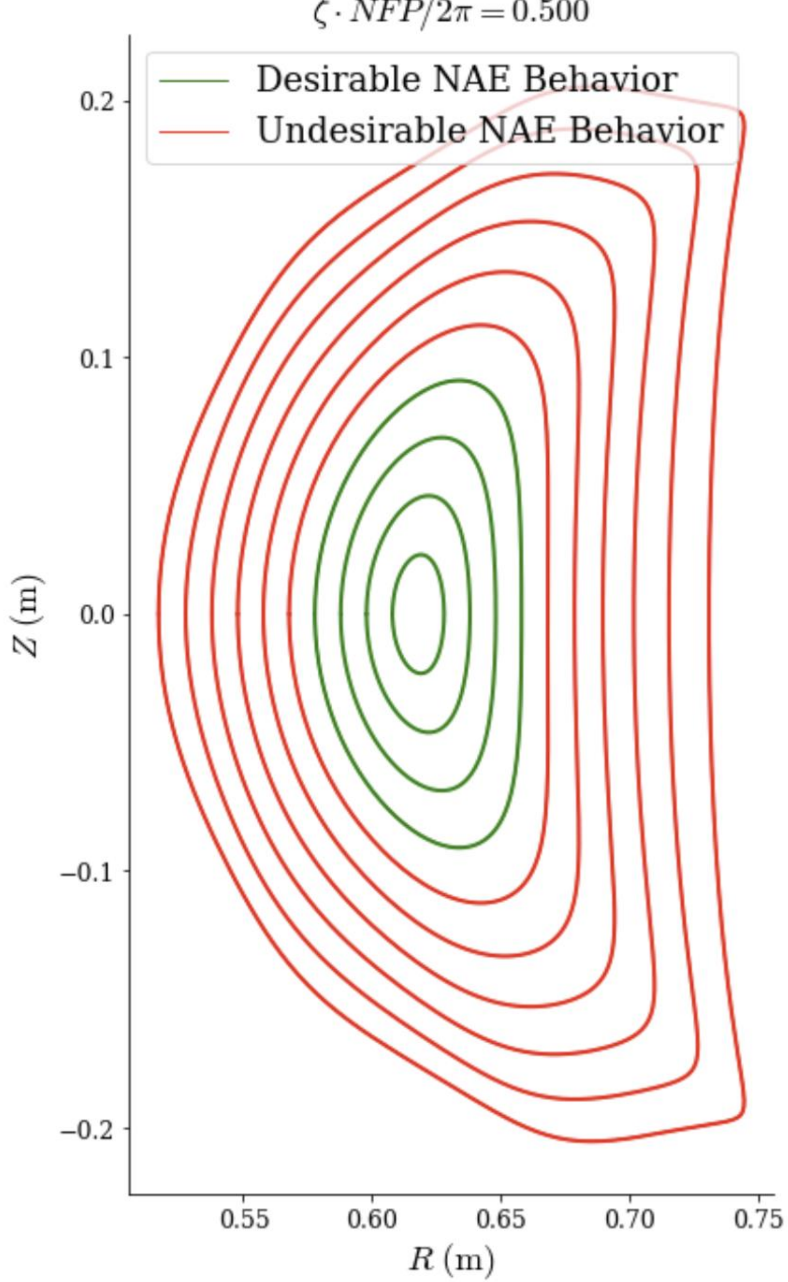
$$\sum_{l=0}^{L} Z_{lmn}\mathcal{Z}_l^m(\rho=1,\theta) = Z^b_{mn} \quad \forall m,n$$



Plasma Control

Near-Axis Expansion (NAE) Constraints in DESC (with E. Rodriguez)

- **Idea is to constrain the global equilibrium to have NAE behavior as $\rho \longrightarrow 0$**
  - **only use information from NAE where it is most valid**
  - **Avoid singular behavior present when evaluating at large r**
- **Map NAE coefficients to Fourier-Zernike modes of DESC to fix $O(\rho^0)$ (axis), $O(\rho^1)$, $O(\rho^2)$ behavior**



pyQSC equilibrium evaluated at r =0.1

O(ρ^0) (axis) Constraint in DESC

NAE axis in pyQSC given as Fourier series in cylindrical toroidal angle ϕ:

$$R = R_0 + \sum_{n=1}^{N} (R_n^C \cos m\phi + R_n^S \sin m\phi)$$

$$Z = \sum_{n=1}^{N} (Z_n^C \cos m\phi + Z_n^S \sin m\phi)$$

Constraint in DESC representation is simple: Evaluate DESC R(ρ,θ,ϕ), Z(ρ,θ,ϕ) at ρ=0 and match terms:

NAE Axis Coefficients

$$R_n^{C/S} = \sum_{k=0}^{\infty} (-1)^k R_{2k,0,\pm|n|}$$

$$Z_n^{C/S} = \sum_{k=0}^{\infty} (-1)^k Z_{2k,0,\pm|n|}$$

DESC Fourier-Zernike Coefficients

Plasma Control

$O(\rho^1)$ NAE Constraint in DESC

- After a short geometric derivation, one can derive (up to $O(\rho)$) the R,Z position of a point on a flux surface from the NAE in terms of the cylindrical angle

$$\mathbf{r} \approx \mathbf{r}_0(\phi) + \rho R_1 \hat{\mathbf{R}} + \rho Z_1 \hat{\mathbf{Z}}$$

where

$$R_1 = \mathcal{R}_{1,1}(\phi)\cos\theta + \mathcal{R}_{1,-1}(\phi)\sin\theta \qquad Z_1 = Z_{1,1}(\phi)\cos\theta + Z_{1,-1}(\phi)\sin\theta$$

- And the coefficients are functions of the NAE X,Y coefficients and the Frenet-Serret basis vectors

- Then, equating the $O(\rho)$ coefficients in the DESC Fourier-Zernike basis with the above expressions yields:

(Identical expressions for Z as well)

NAE Coefficients

$$\mathcal{R}_{1,1,n} = -\sum_{k=1}^{M}(-1)^k k R_{2k-1,1,n},$$

$$\mathcal{R}_{1,-1,n} = -\sum_{k=1}^{M}(-1)^k k R_{2k-1,-1,n},$$

DESC Fourier-Zernike Coefficients

Plasma Control

Verification of DESC Equilibrium Quantities Against the NAE

Magnetic Axis and Near-Axis Flux Surfaces

On-axis Rotational Transform

Near-Axis Confinement (f_B for QS, $\epsilon_{eff}^{3/2}$ for QI)

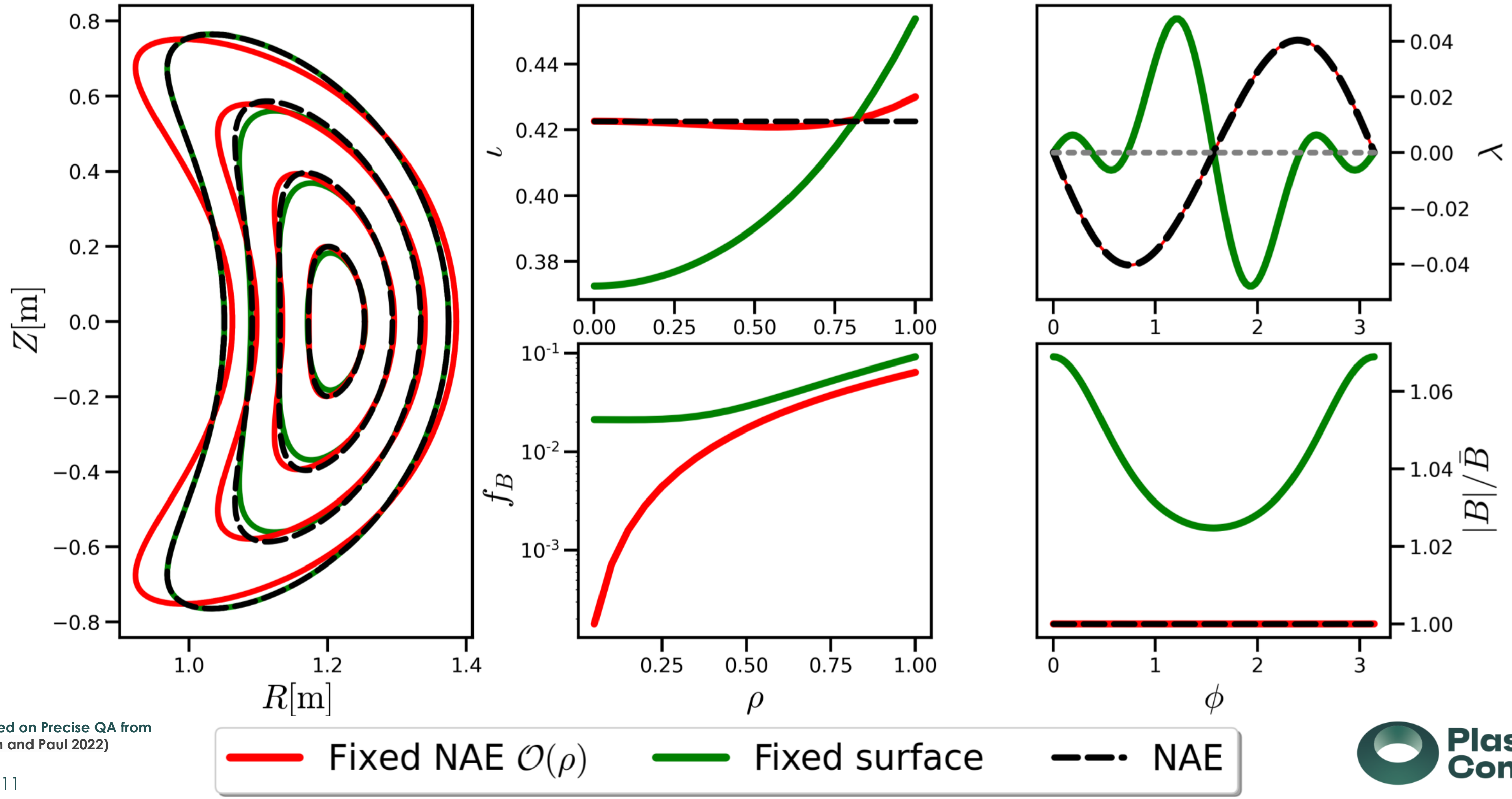
On-axis Poloidal Angle ($\theta = \theta_B$)

On-axis Magnetic Field Magnitude $|B|$

Will show 3 examples: QA, QH, and a QI NAE-constrained equilibrium

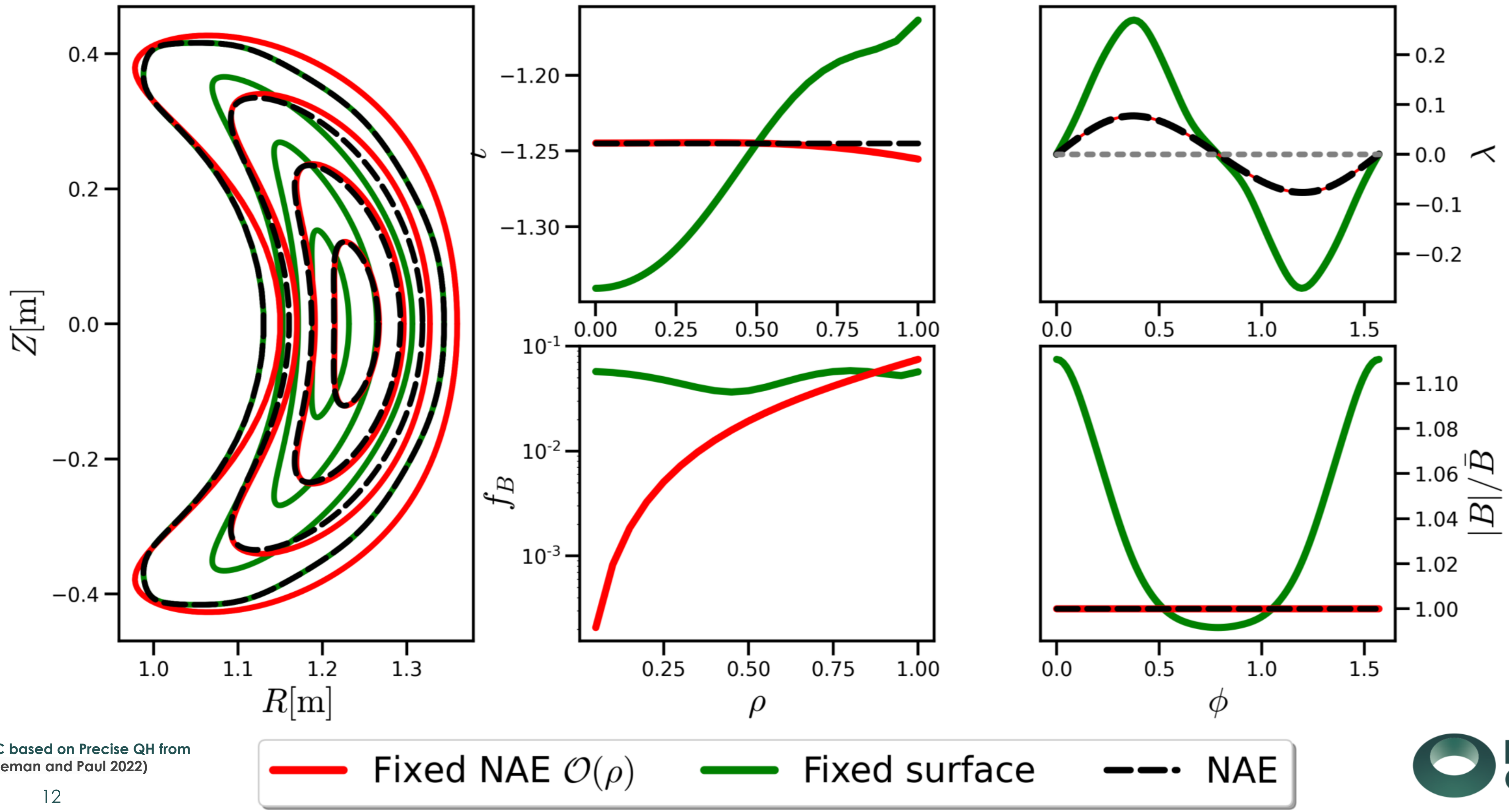
Plasma Control

QA $O(\rho^1)$ NAE-constrained Equilibrium



pyQSC based on Precise QA from (Landreman and Paul 2022)

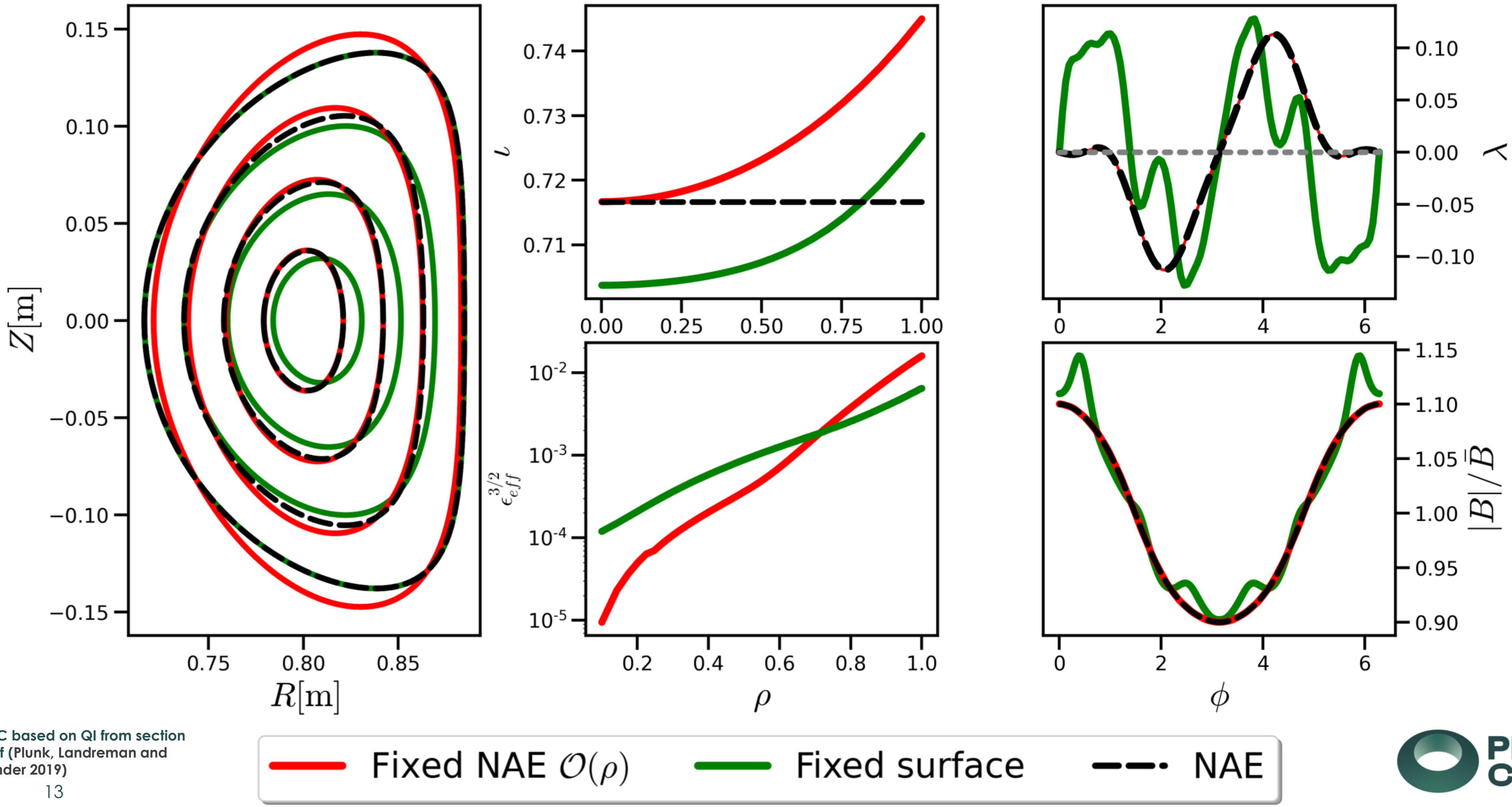
QH O(ρ^1) NAE-constrained Equilibrium



pyQSC based on Precise QH from (Landreman and Paul 2022)

Plasma Control

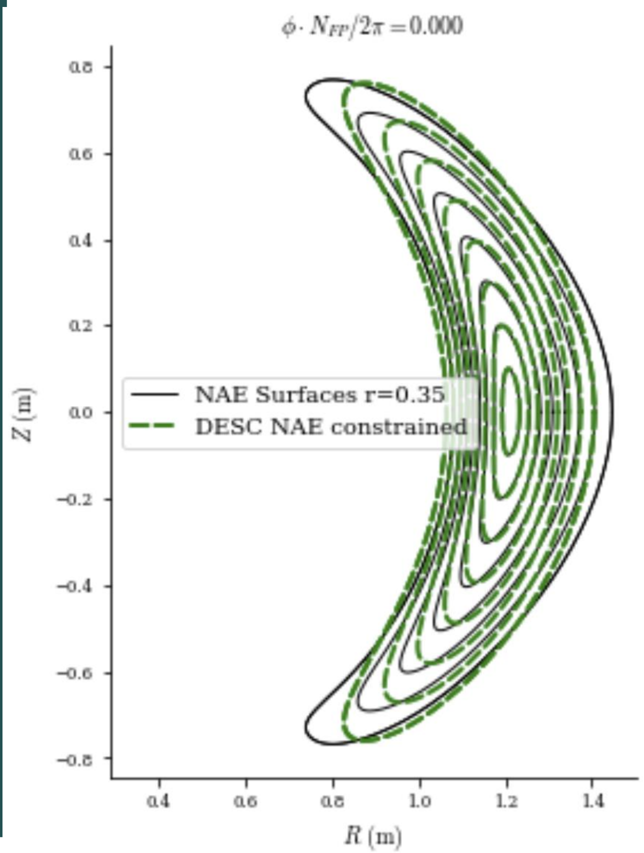
QI O(ρ^1) NAE-constrained Equilibrium



pyQIC based on QI from section 8.2 of (Plunk, Landreman and Helander 2019)

Example Python Code for NAE-Constrained Equilibria in DESC – Simple!

```
# imports
from qsc import Qsc
from desc.objectives import get_NAE_constraints
from desc.equilibrium import Equilibrium

# fit initial DESC equilibrium from desired qsc solution
qsc = Qsc.from_paper('precise QA',nphi=99)
desc_eq = Equilibrium.from_near_axis(qsc,r=0.35,L=9,M=9,N=10)

# get constraints on axis and O(r) coefficients
# to pass to eq.solve using utility function
cs = get_NAE_constraints(desc_eq, qsc, order=1)

# solve NAE-constrained equilibrium
desc_eq.solve(objective="force",constraints=cs);
```



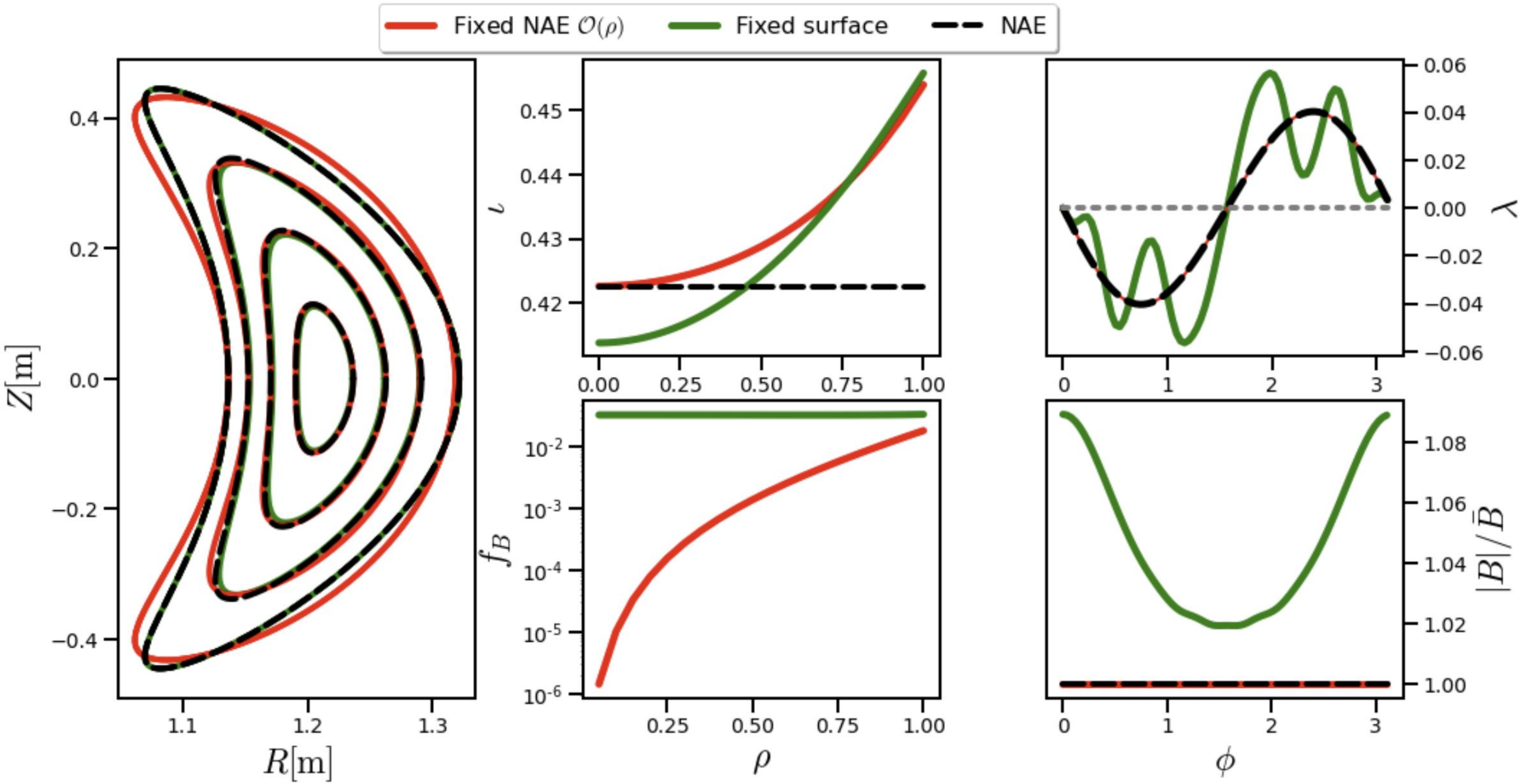
Tutorial on DESC documentation website:

desc-docs.readthedocs.io

D. Panici / December 2023

Plasma Control

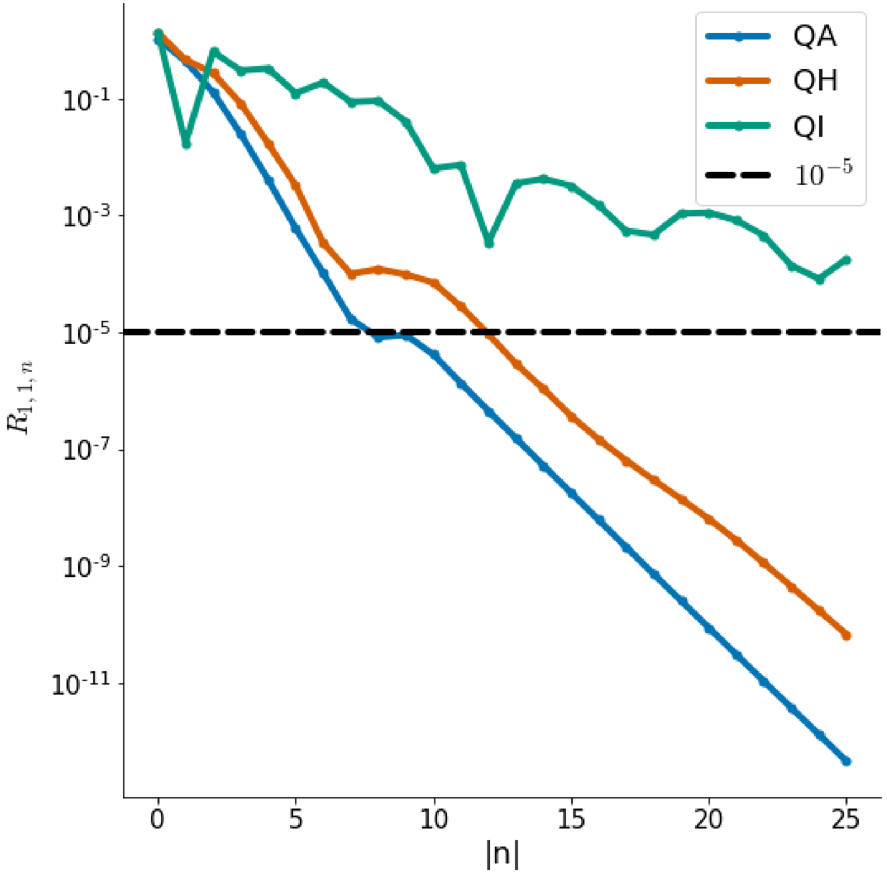
QA O(ρ^2) NAE-constrained Equilibrium



Fitting NAE Behavior with Toroidal Fourier Series – O(ρ^1)

O(ρ) Coefficient $R_{1,1,n}$

$$R_1 = \mathcal{R}_{1,1}(\phi)\cos\theta + \mathcal{R}_{1,-1}(\phi)\sin\theta$$

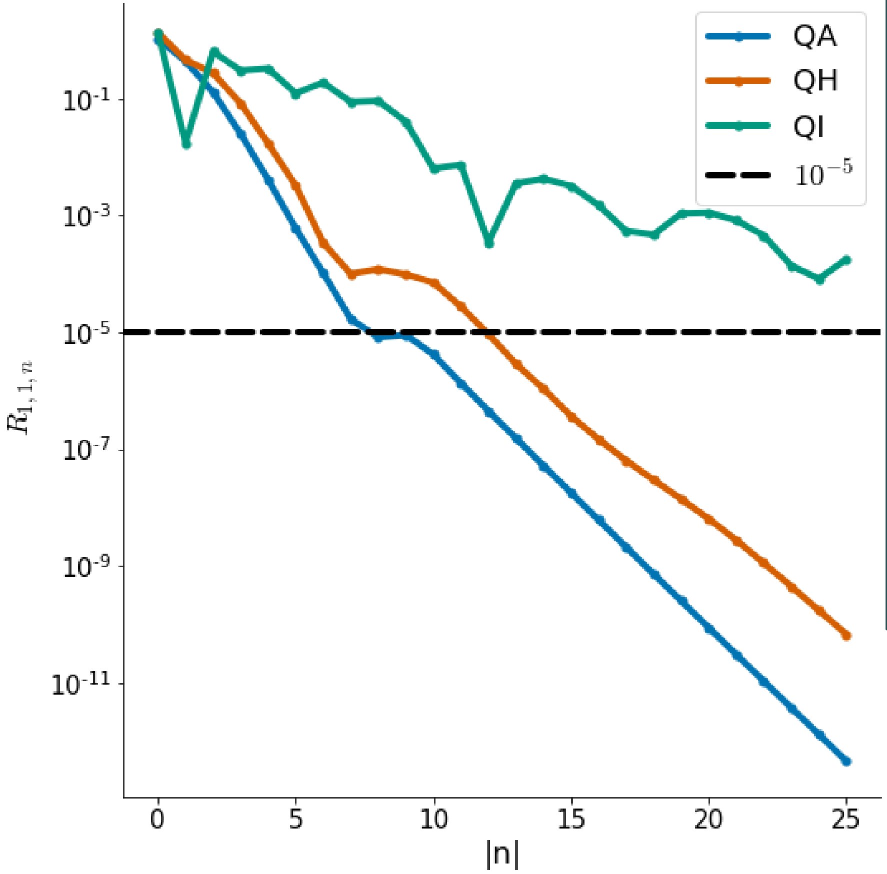


QA NAE behavior simplest to describe

QI NAE behavior very difficult to describe with cylindrical angle!

Fitting NAE Behavior with Toroidal Fourier Series – O(ρ^2)

O(ρ) Coefficient $R_{1,1,n}$

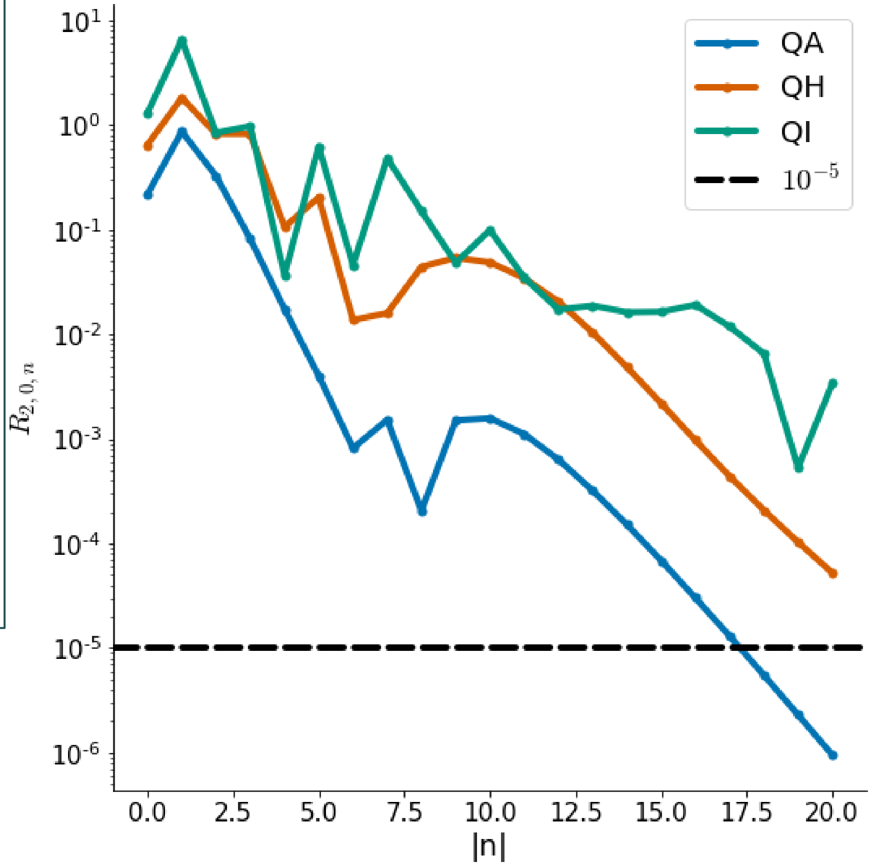


O(ρ) NAE behavior much easier to describe with cylindrical angle than O(ρ^2)

O(ρ^2) – constrained equilibria may require very high N_{tor} to capture behavior accurately

Generalized toroidal angle may help condense spectrum

O(ρ^2) Coefficient $R_{2,0,n}$



$$R_2 = \mathcal{R}_{2,0}(\phi) + \mathcal{R}_{2,2}(\phi)\cos 2\theta + \mathcal{R}_{2,-2}(\phi)\sin 2\theta$$

Plasma Control

(REG)Coil Optimization In DESC

D. Panici / December 2023

Plasma Control

REGCOIL Algorithm

- **Using surface current distributions on a specified winding is an efficient approach to the coil-finding problem[4,5]**

$$\boldsymbol{K} = \boldsymbol{n} \times \nabla\Phi \qquad \Phi(\theta', \zeta') = \Phi_{sv}(\theta', \zeta') + \frac{G\zeta'}{2\pi} + \frac{I\theta'}{2\pi}$$

Surface Current Density | Unit Normal to Winding Surface | Current Potential | G: Surface Net Poloidal Current | I: Surface Net Toroidal Current

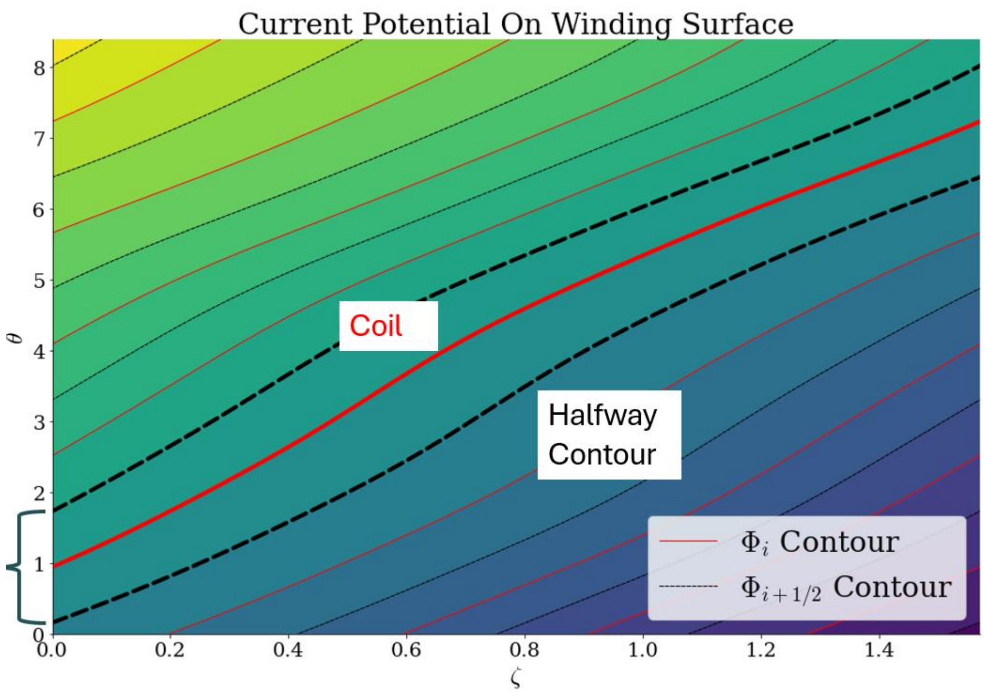
- **Then minimization of quadratic flux becomes a linear (in Φ_{sv}) least-squares problem, after expanding in Fourier Series (I,G, and other terms are known)**

$$\chi_B^2 = \int d^2a \; B_{\text{normal}}^2 \qquad \Phi_{sv} = \sum_{m,n} \Phi_{sv}^{mn} sin(m\theta' - n\zeta') \qquad B_n = B_n^{ext} + B_n^{pl} + B_n^{GI} + A\Phi_{sv}^{mn}$$

- **However, can lead to poor solutions without regularization -> REGCOIL adds regularization to the problem**

$$\chi_K^2 = \int d^2a' \; K(\theta', \zeta')^2.$$

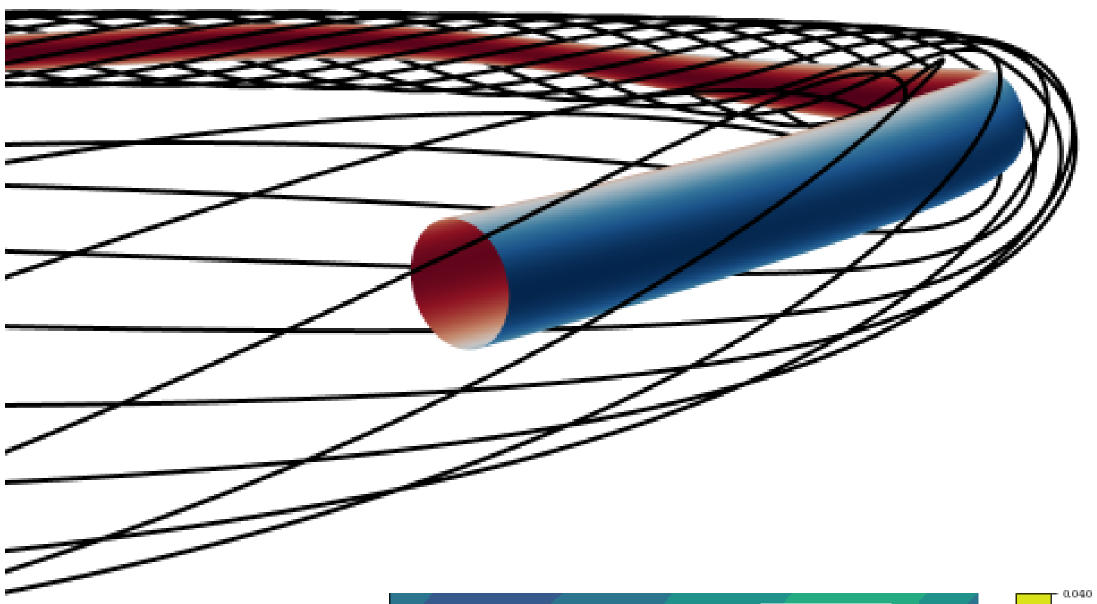
Helical Coilset in DESC

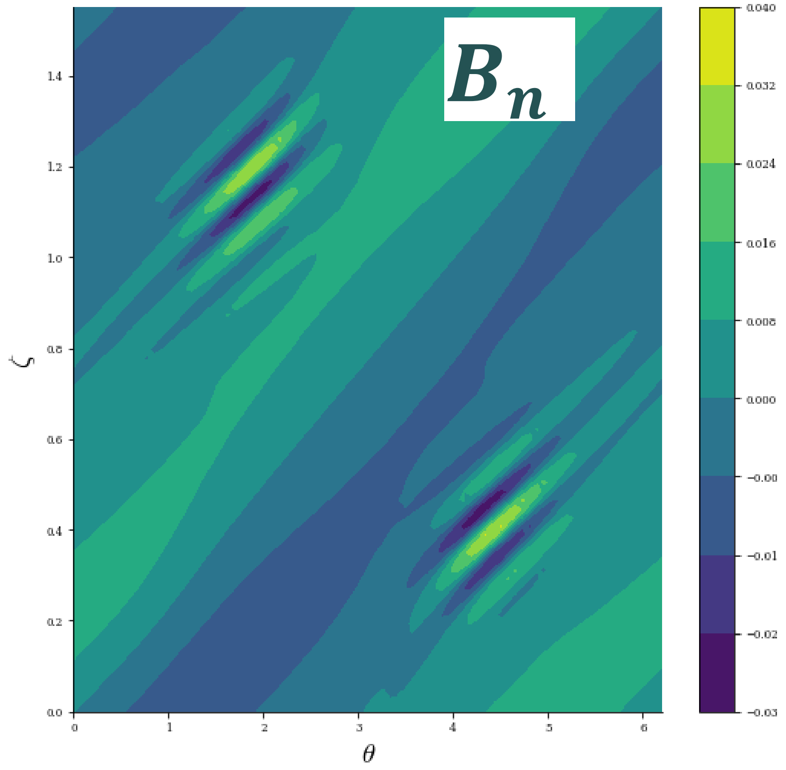


REGCOIL Algorithm implemented in DESC to find surface currents

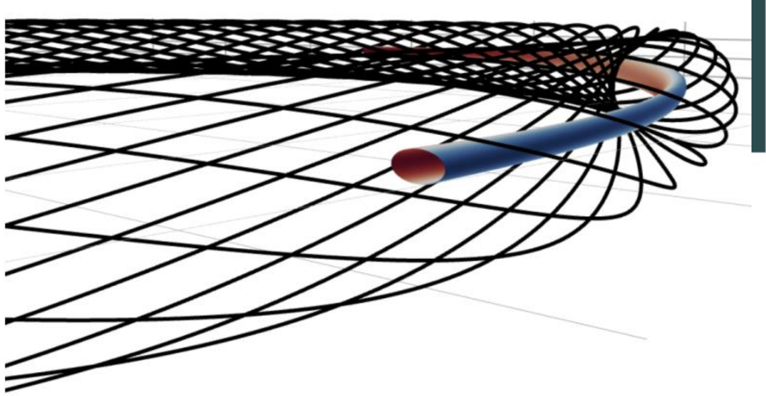
Helical coil-cutting capabilities also implemented to discretize into helical coils

Helical Coilset

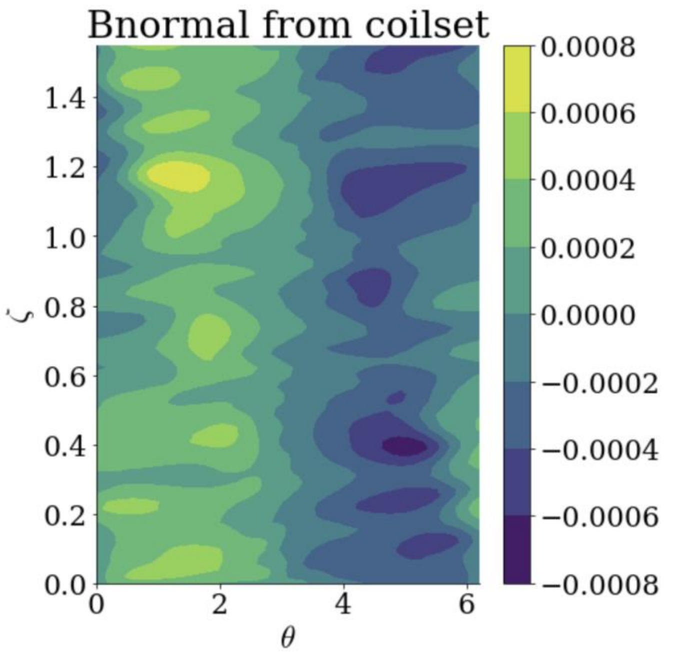




Example Python Code to create Helical Coilset

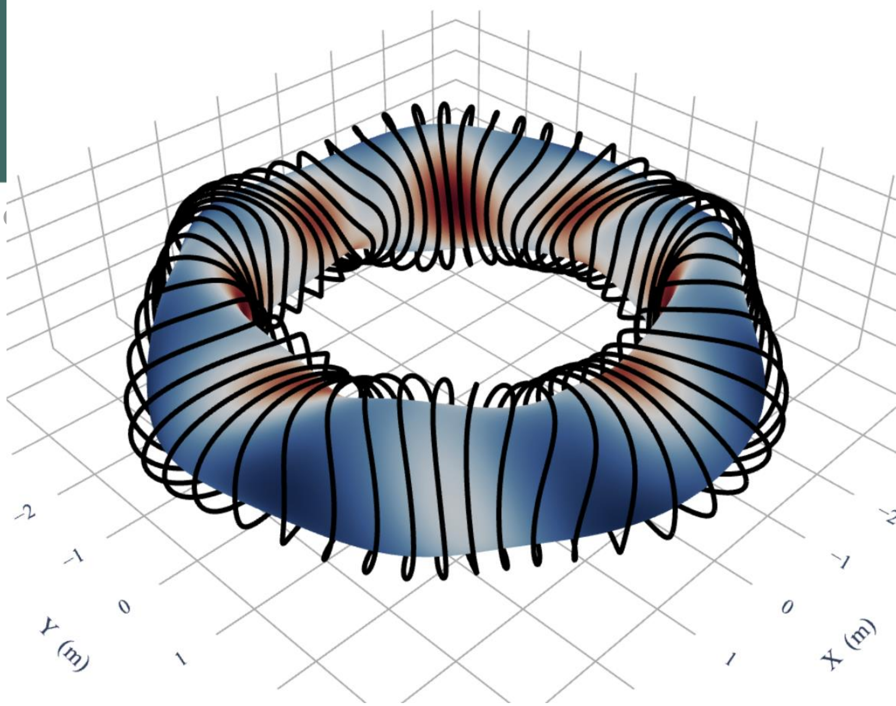


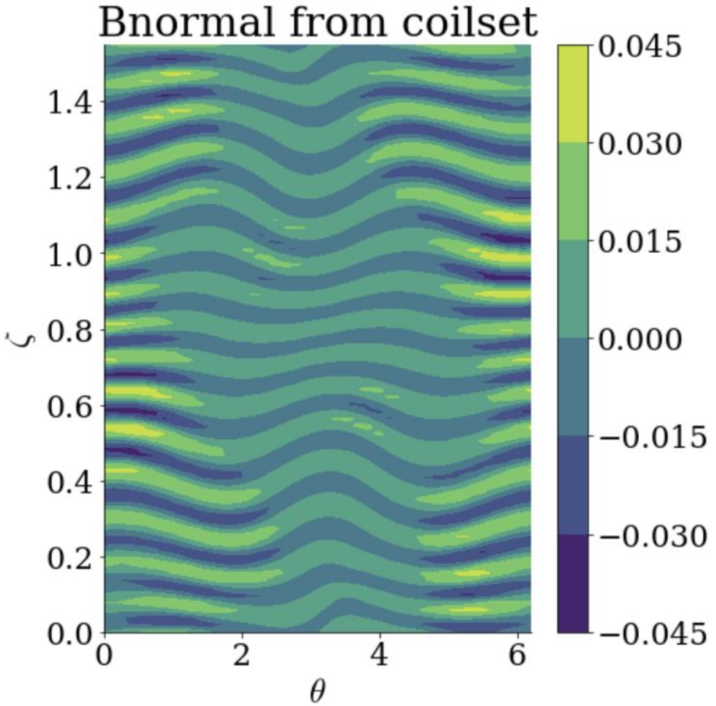
```python
# load equilibrium, this case is a simple vacuum rotating ellipse
eqname = "./tests/inputs/ellNFP4_init_smallish.h5"
eq = load(eqname)

# get the surface current which minimizes Bn with REGCOIL algorithm
(surface_current_field, _, _, _, _,) = run_regcoil(
    eqname=eq,
    # resolutions of plasma surface grid upon which Bn is minimized
    eval_grid_M=20,eval_grid_N=20,
    # resolutions of source grid for calculating Bn
    source_grid_M=40, source_grid_N=80,
    alpha=1e-15, # regularization parameter
    # ratio of I/G, 0 for modular, integer for helical coils
    helicity_ratio=-1)

# discretize into helical coils using utility function
numCoils = 15 # we want 15 helical coils
coilset2 = find_helical_coils(surface_current_field, eqname, desirednumcoils=numCoils)
```



Example Python Code to create Modular Coilset

```
# load equilibrium, this case is a simple vacuum rotating ellips
eqname = "rotating_ellipse_5_aspect_ratio.h5"
eq = load(eqname)
winding_surf= load("rotating_ellipse_wind_surf.h5")
# get the surface current which minimizes Bn with REGCOIL alg
(surface_current_field, _, _, _, _,) = run_regcoil(
    eqname=eq, basis_M=16, basis_N=16,
# resolutions of plasma surface grid upon which Bn is minimized
    eval_grid_M=60,eval_grid_N=60,
    # resolutions of source grid for calculating Bn
    source_grid_M=100, source_grid_N=100,
    alpha=1e-16, # regularization parameter
    # ratio of I/G, 0 for modular, integer for helical coils
    helicity_ratio=0, winding_surf = winding_surf)

# discretize into modular coils using utility function
numCoils = 60 # we want 60 modular coils, (15 per field period)
coilset2 = find_modular_coils(surface_current_field,
                              eqname, desirednumcoils=numCoils)
```



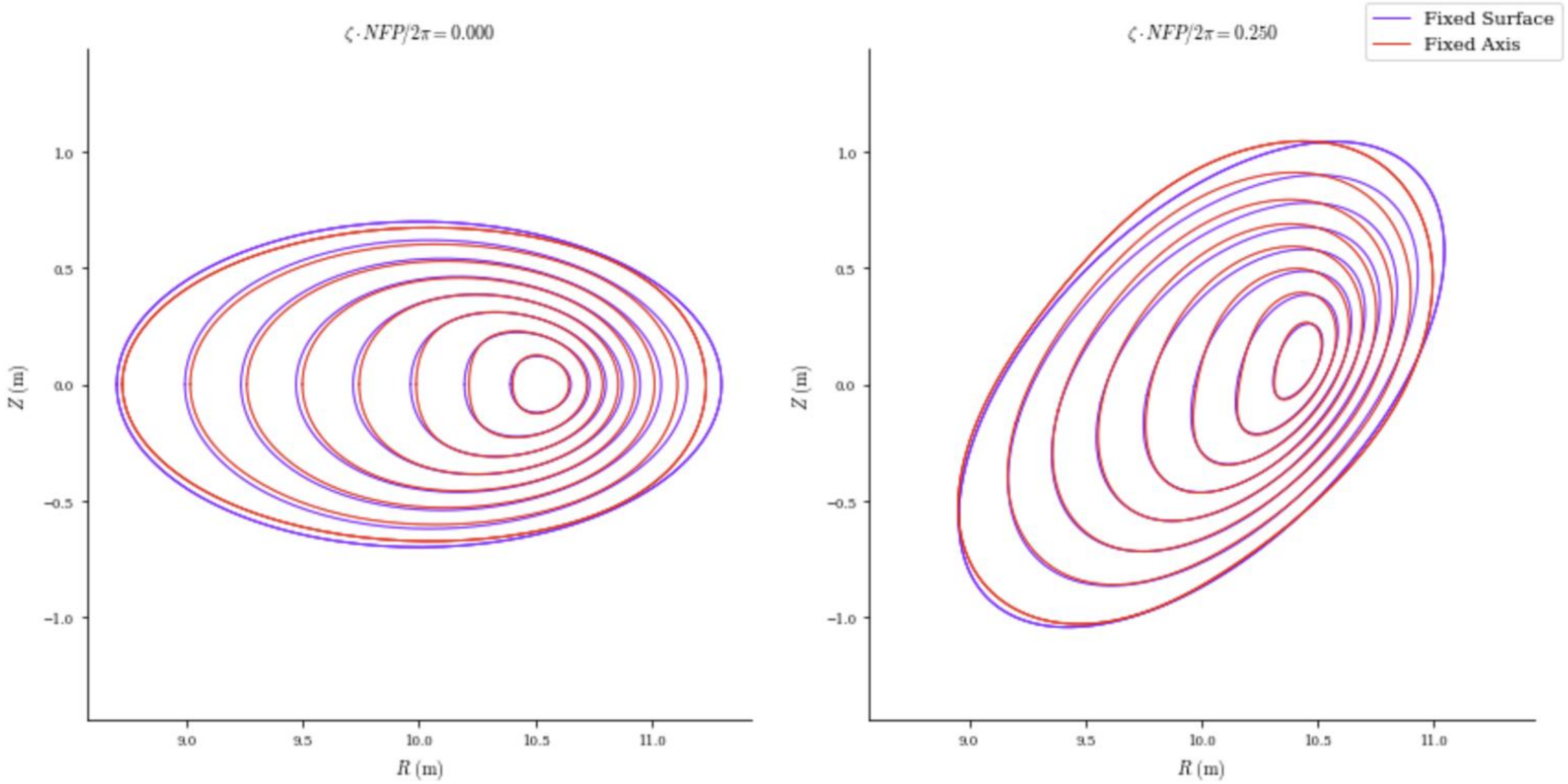


DESC Offers Unique Flexibility in Constraints that Open New Possibilities

- <u>**NAE Constrained Equilibrium Solve**</u>
  - **Can offer connection between rich NAE+QS theory and global solutions**
  - **Allow global solutions to be found matching NAE axes that otherwise could not be found traditionally**
  - **Verified against pyQSC and pyQIC for O(rho) constraints**
  - **Ongoing verification of 2nd order constraints**
    - Can allow geometrically constraining on-axis Mercier stability, for example
  - **Can be used with DESC constrained optimization**
  - **Available to use now in main DESC code**
- <u>**DESC Coil Optimization**</u>
  - **REGCOIL algorithm implemented in Python**
  - **Modular and Helical Coil-cutting algorithms implemented**
  - **Written in JAX, so can be used with optimization, combined with other objectives**
    - Single-stage optimization?
  - **Will be available soon in main DESC code**

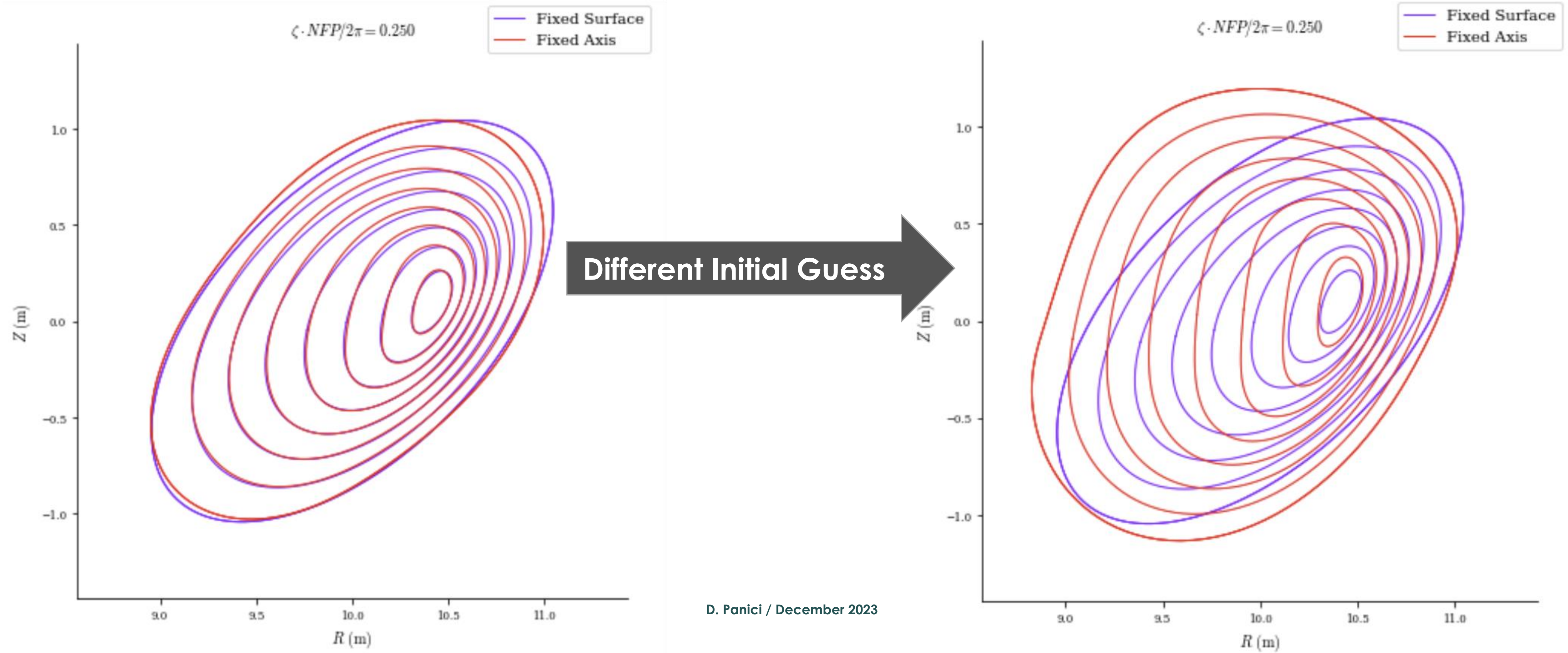
Backup

O(ρ^0) (axis) Constraint in DESC - Example Solve



Plasma Control

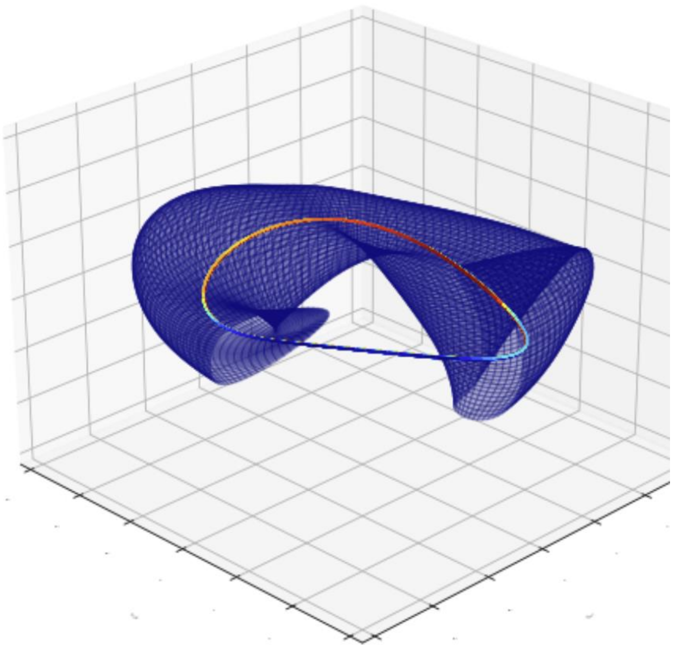
O(ρ^0) (axis) Constraint in DESC - Under-constrained Problem, Finds Closest Equilibrium



Physical Insights Yield Constraints on XS or near Axis

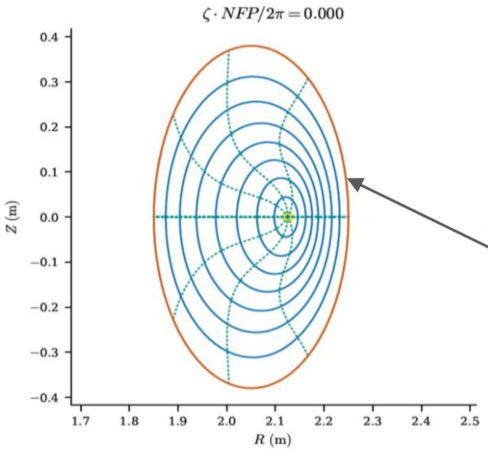
Axis + Near-Axis Behavior

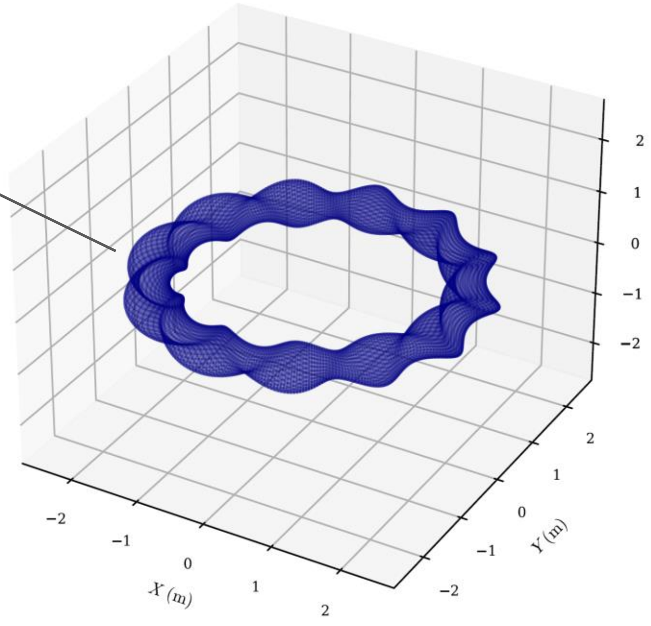
Near-Axis Expansion (NAE) yields what asymptotic behavior of equilibrium should be near the axis, and what the axis shape should be



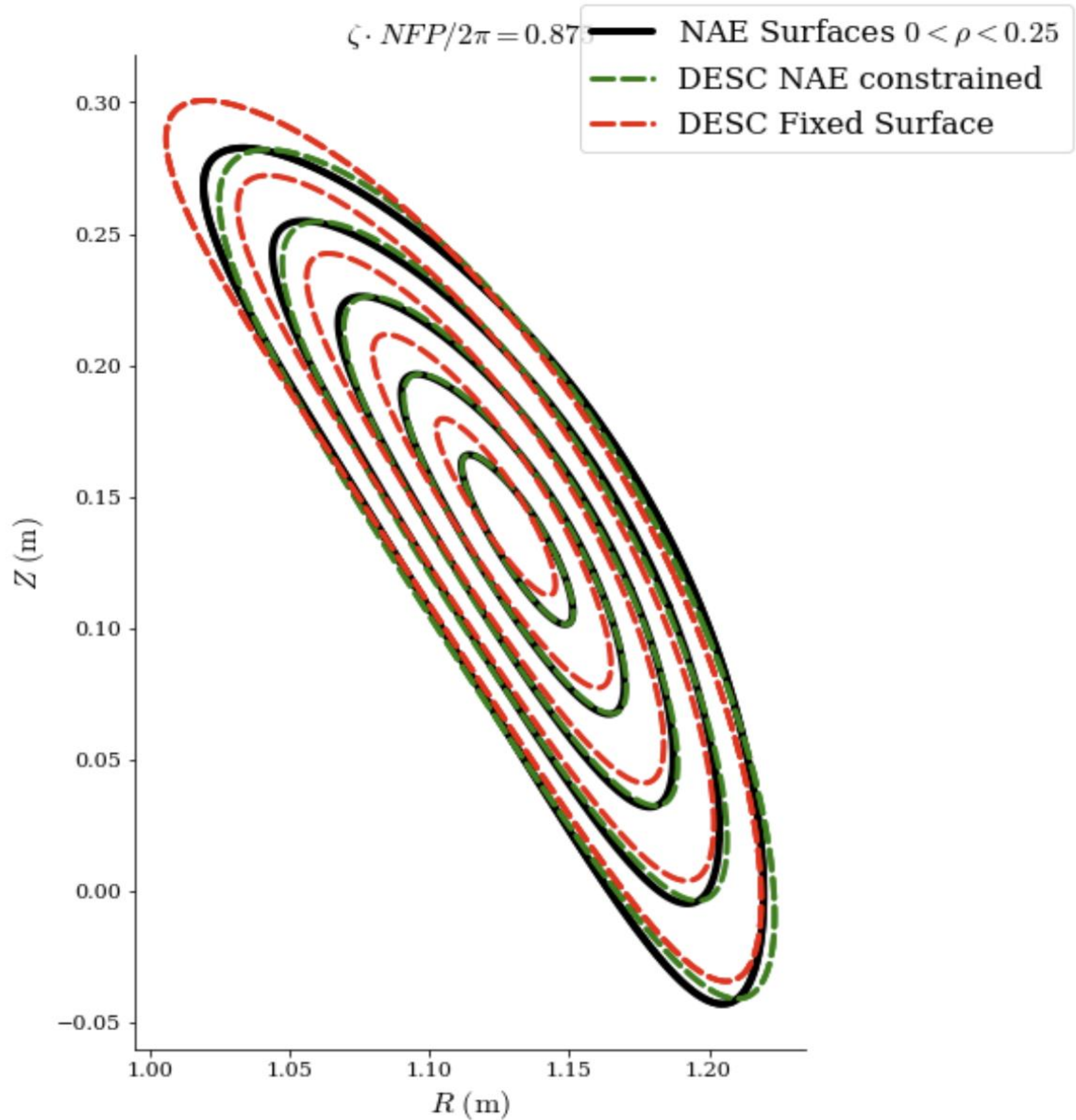
Poincare Section

Desire to avoid magnetic islands, and decoupling poloidal and toroidal resolution



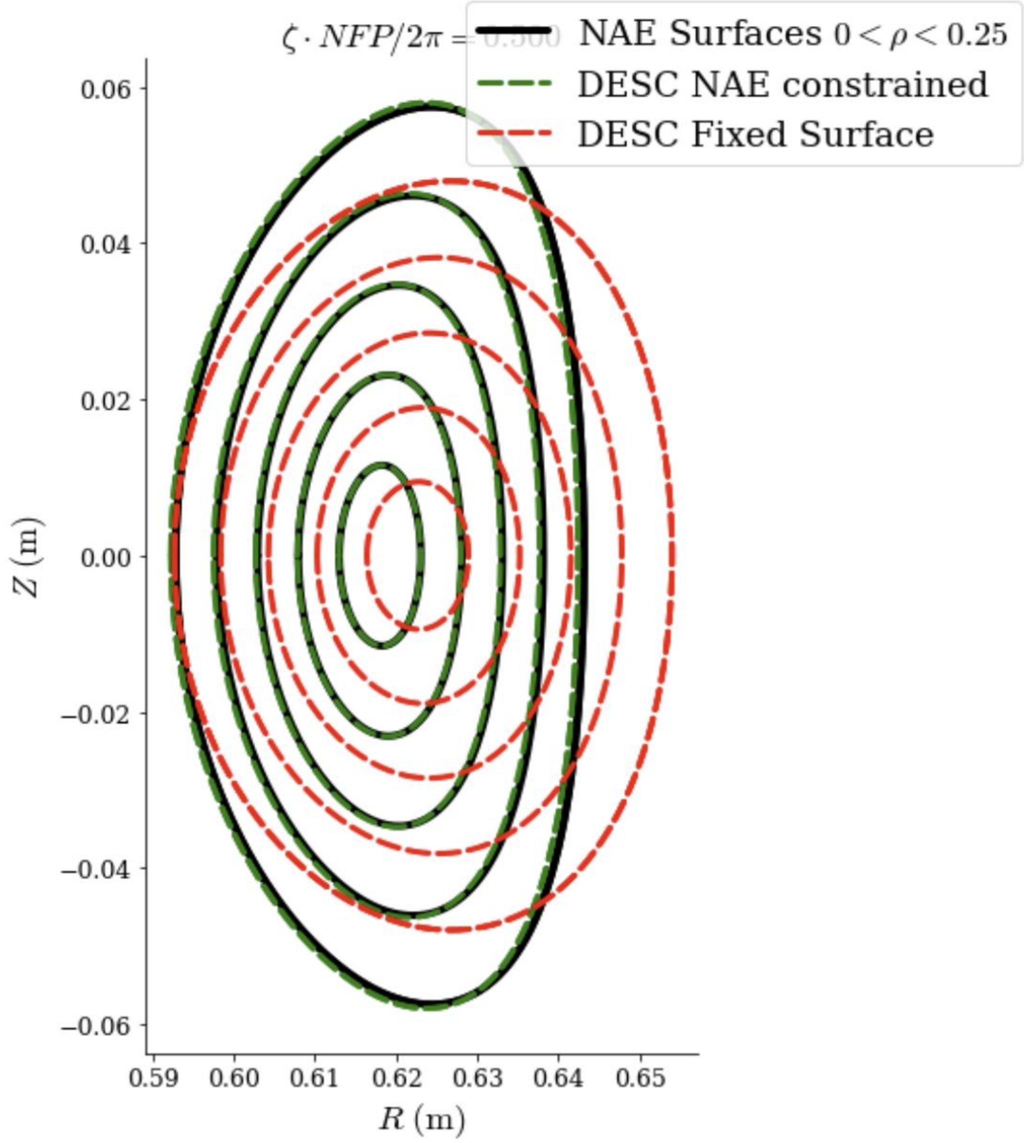


Closer look at flux surfaces near axis for Precise QA



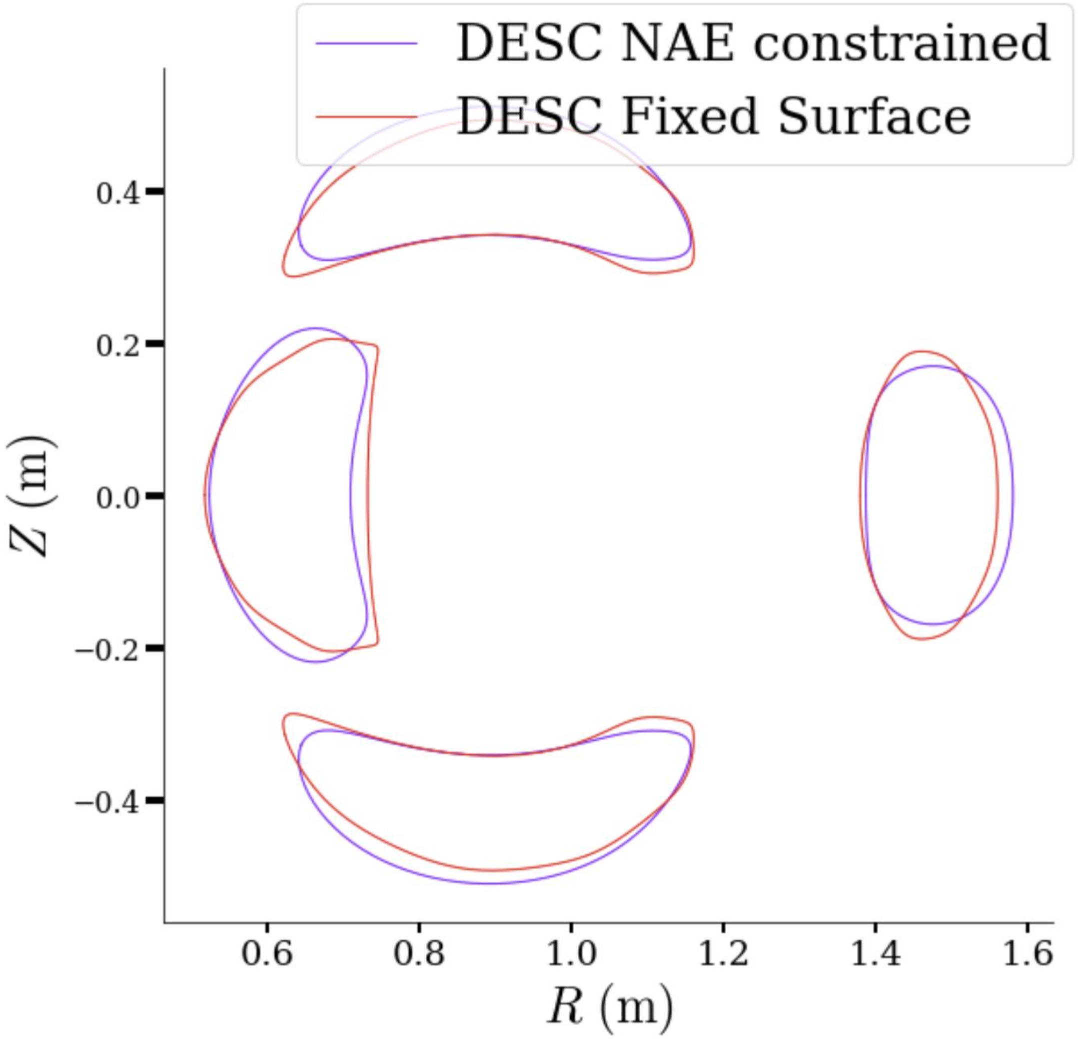
Closer look at flux surfaces near axis for difficult NAE

```
rc = [1, 0.426, 0.044, -6.3646383583351e-11,
2.851584586653665e-05, 3.892992983405039e-08]

zs = [0.0, 0.4110168175146285, 0.04335427796015756,
6.530936323433338e-05, 1.3623898672936873e-05,
1.1620514629503932e-05]

etabar=1.64209358
B2c = 0.11293987662545873
B0=1
nfp = 4

qsc = Qsc(rc=rc, zs=zs, B0=B0, nfp=nfp, I2=0, B2c = B2c,
etabar=etabar, order = "r1", nphi = 201)

desc_eq= Equilibrium.from_near_axis(qsc,r=
r,L=9,M=9,N=N,ntheta=ntheta)
```



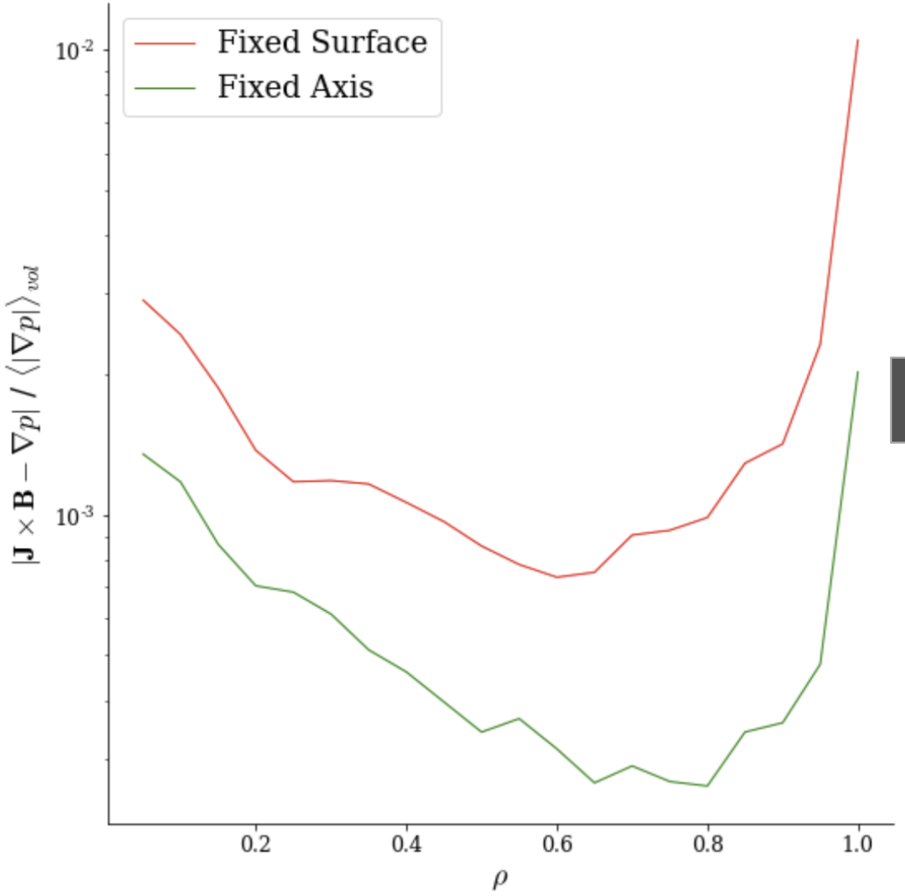
Closer look at LCFS for difficult NAE

```
rc = [1, 0.426, 0.044, -6.3646383583351e-11,
2.851584586653665e-05, 3.892992983405039e-08]

zs = [0.0, 0.4110168175146285, 0.04335427796015756,
6.530936323433338e-05, 1.3623898672936873e-05,
1.1620514629503932e-05]

etabar=1.64209358
B2c = 0.11293987662545873
B0=1
nfp = 4

qsc = Qsc(rc=rc, zs=zs, B0=B0, nfp=nfp, I2=0, B2c = B2c,
etabar=etabar, order = "r1", nphi = 201)

desc_eq= Equilibrium.from_near_axis(qsc,r=
r,L=9,M=9,N=N,ntheta=ntheta)
```

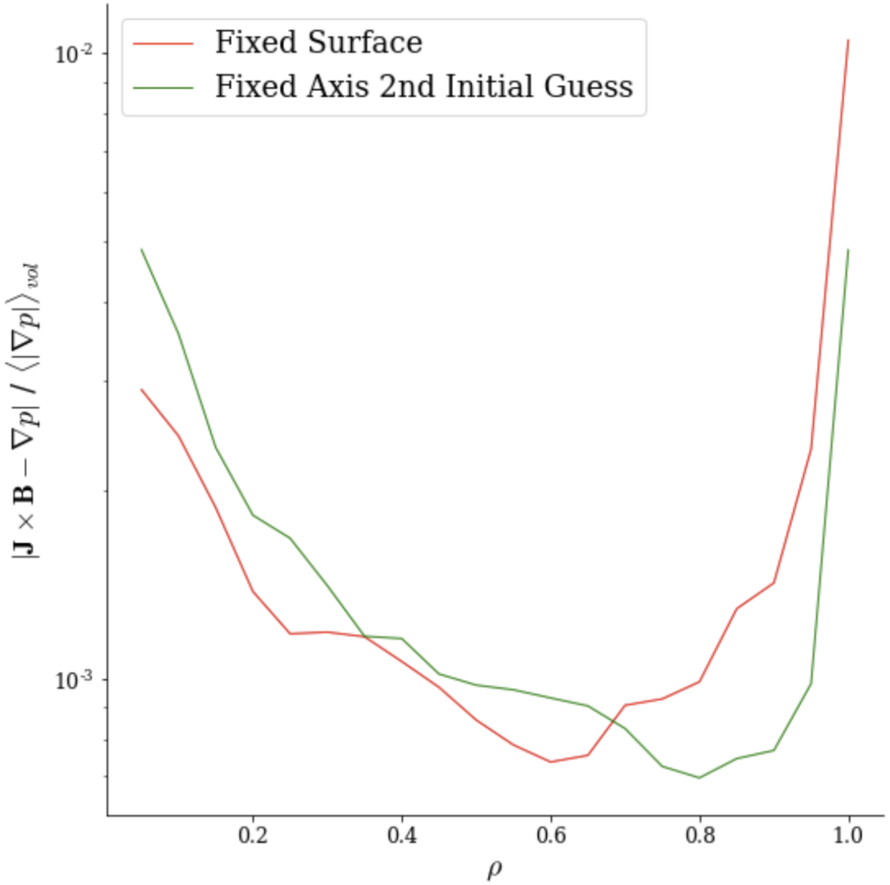


Plasma Control

O(ρ^0) (axis) Constraint in DESC - Under-constrained Problem



Different Initial Guess



Near-Axis Expansion

- **Quantities are expanded in form**

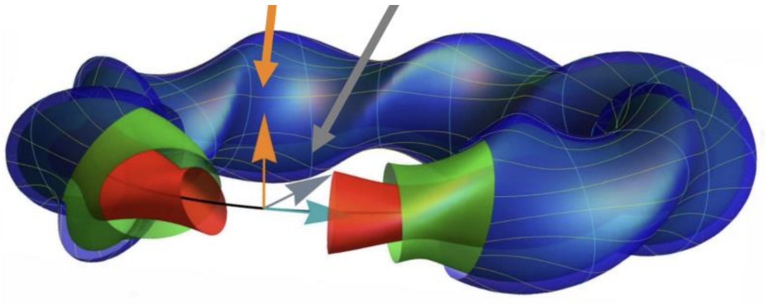
$$B_1(\vartheta,\varphi) = B_{1s}(\varphi)\sin(\vartheta) + B_{1c}(\varphi)\cos(\vartheta),$$
$$B_2(\vartheta,\varphi) = B_{20}(\varphi) + B_{2s}(\varphi)\sin(2\vartheta) + B_{2c}(\varphi)\cos(2\vartheta)$$

$$B(r,\vartheta,\varphi) = B_0(\varphi) + rB_1(\vartheta,\varphi) + r^2B_2(\vartheta,\varphi) + r^3B_3(\vartheta,\varphi) + \dots.$$

- **Inputs for $O(r^2)$ solutions**
  - **Axis Shape (R(phi), Z(phi))**
  - $\bar{\eta} = \frac{B_{1c}}{B_0}$ **Measure of magnetic field variation**
  - σ_0 **Deviation from stellarator symmetry at $\phi = 0$**
    - **Taken as 0 for most cases**
  - I_2 **Current Density on axis**
  - p_2 **Pressure near axis**
  - B_{2c} **magnetic field $O(r^2)$ poloidal variation**

$$B = B_0\left[1 + r\bar{\eta}\cos\vartheta + O(r^2)\right]$$

Outputs:
- **Flux surface shapes in neighborhood of axis**
- ι_0 **rotational transform on-axis**
- B_{20} **magnetic field variation on-axis**



(Jorge 2022)

$$\boldsymbol{r}(r,\vartheta,\varphi) = \boldsymbol{r}_0(\varphi) + X(r,\vartheta,\varphi)\boldsymbol{n}(\varphi) + Y(r,\vartheta,\varphi)\boldsymbol{b}(\varphi) + Z(r,\vartheta,\varphi)\boldsymbol{t}(\varphi)$$

$$X(r,\vartheta,\varphi) = rX_1(\vartheta,\varphi) + r^2X_2(\vartheta,\varphi) + r^3X_3(\vartheta,\varphi) + \dots.$$

(Landreman 2022)

DESC Allows Flexible Constraints when Defining Equilibrium Problem - Fixed ρ=1 Boundary

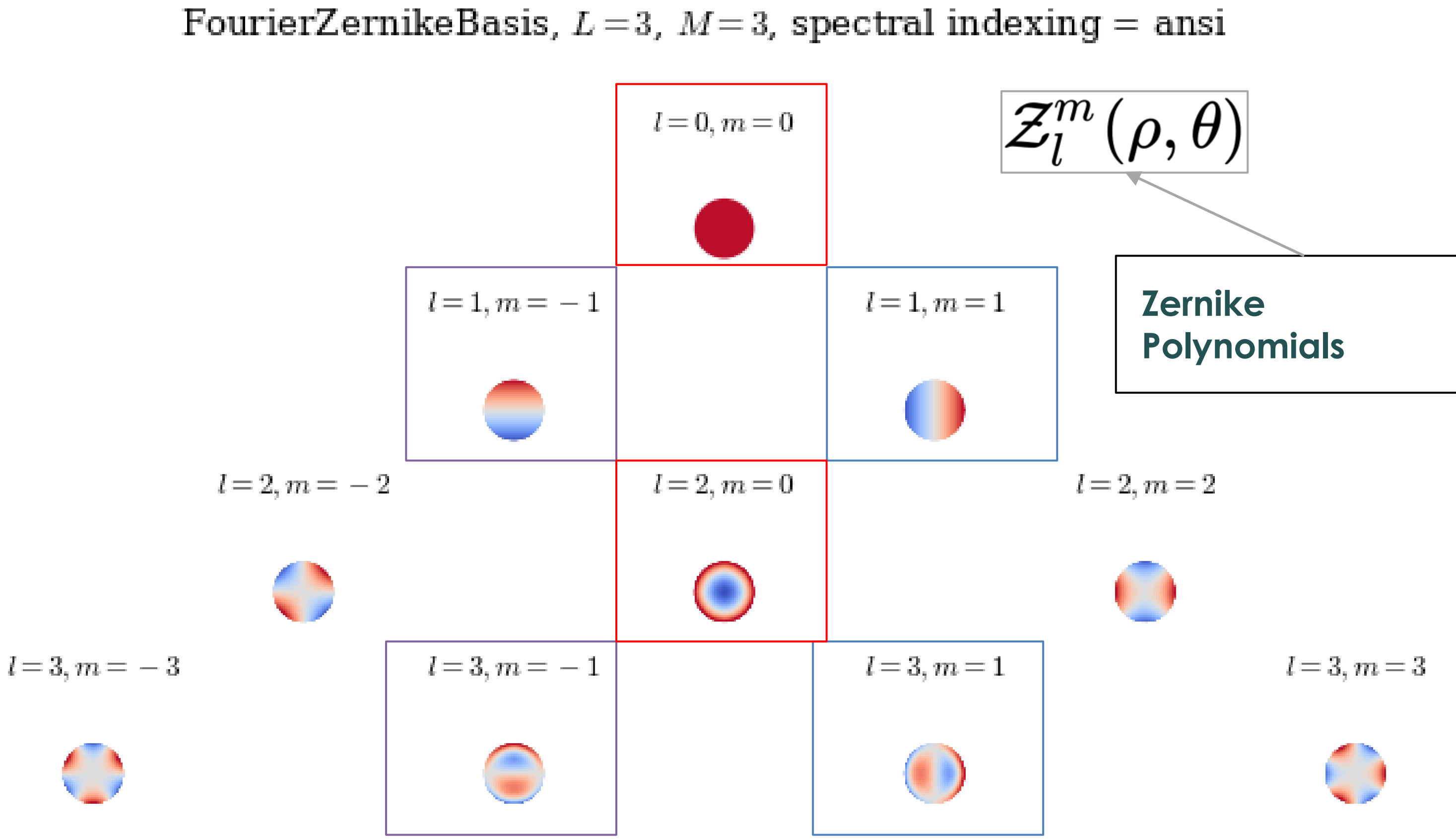
$$R^b(\theta,\zeta) = \sum_{m=0}^{M} \sum_{n=-N}^{N} R^b_{m,n} cos(m\theta - n\zeta)$$

$$Z^b(\theta,\zeta) = \sum_{m=0}^{M} \sum_{n=-N}^{N} Z^b_{m,n} sin(m\theta - n\zeta)$$

Fixed-Boundary ρ=1 Constraint

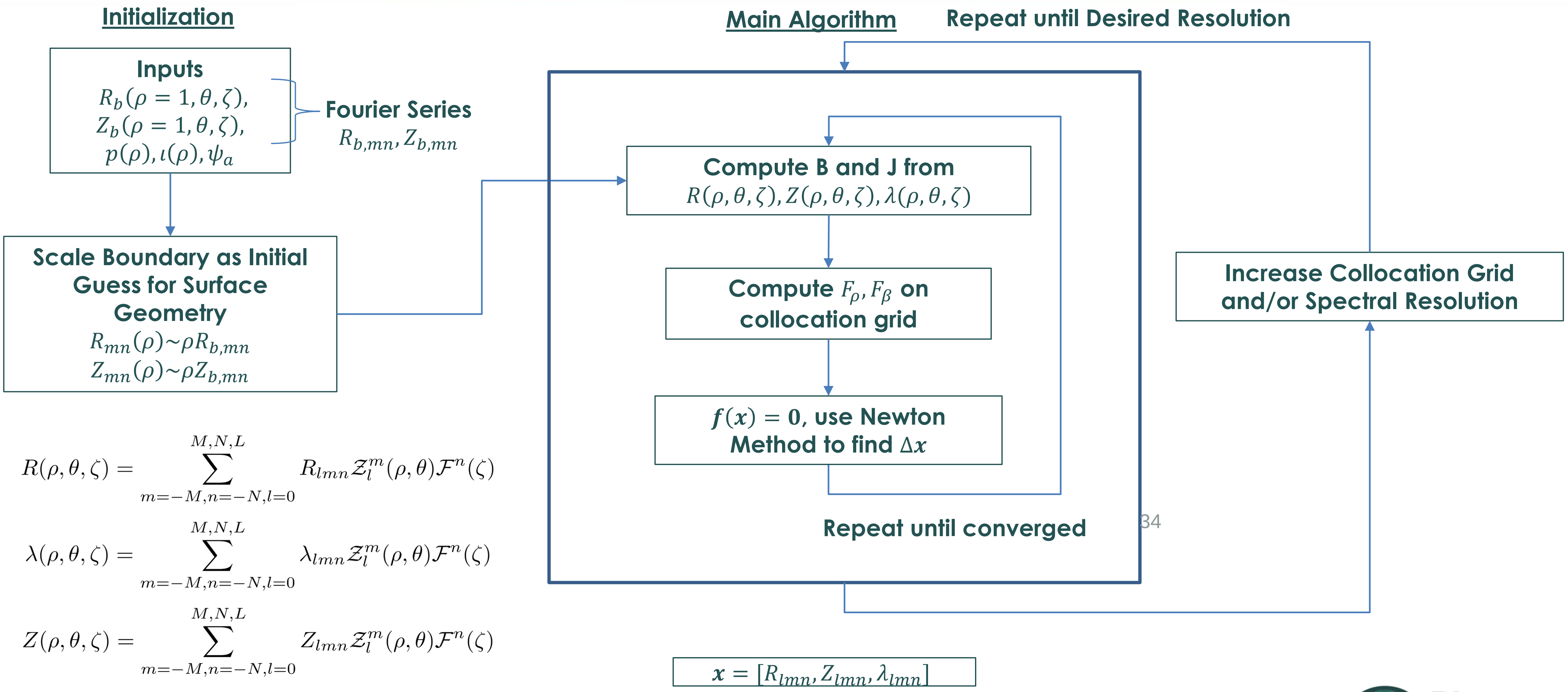
$$\sum_{l=0}^{L} R_{lmn} \mathcal{Z}_l^m(\rho=1,\theta) = R^b_{mn} \quad \forall m,n$$

$$\sum_{l=0}^{L} Z_{lmn} \mathcal{Z}_l^m(\rho=1,\theta) = Z^b_{mn} \quad \forall m,n$$



Plasma Control

DESC Algorithm



$$R(\rho,\theta,\zeta) = \sum_{m=-M,n=-N,l=0}^{M,N,L} R_{lmn} \mathcal{Z}_l^m(\rho,\theta) \mathcal{F}^n(\zeta)$$

$$\lambda(\rho,\theta,\zeta) = \sum_{m=-M,n=-N,l=0}^{M,N,L} \lambda_{lmn} \mathcal{Z}_l^m(\rho,\theta) \mathcal{F}^n(\zeta)$$

$$Z(\rho,\theta,\zeta) = \sum_{m=-M,n=-N,l=0}^{M,N,L} Z_{lmn} \mathcal{Z}_l^m(\rho,\theta) \mathcal{F}^n(\zeta)$$