

PAPER • OPEN ACCESS

Multiple solutions to the static forward free–boundary Grad–Shafranov problem on MAST-U

To cite this article: K. Pentland *et al* 2025 *Nucl. Fusion* **65** 086053

View the [article online](#) for updates and enhancements.

You may also like

- [Validation of the static forward Grad–Shafranov equilibrium solvers in FreeGSNKE and Fiesta using EFIT++ reconstructions from MAST-U](#)
K Pentland, N C Amorisco, O El-Zobaidi et al.
- [Neutronics analysis of spin-polarized fuel in spherical tokamaks](#)
Jin Whan Bae, Katarzyna Borowiec, Vittorio Badalassi et al.
- [Integrated modeling of SPARC H-mode scenarios: exploration of the impact of modeling assumptions on predicted performance](#)
M. Muraca, P. Rodriguez-Fernandez, N.T. Howard et al.

Multiple solutions to the static forward free–boundary Grad–Shafranov problem on MAST-U

K. Pentland^{1,*} , N.C. Amorisco¹ , P.E. Farrell^{2,3}  and C.J. Ham¹ 

¹ United Kingdom Atomic Energy Authority, Culham Campus, Abingdon, Oxfordshire OX14 3DB, United Kingdom of Great Britain and Northern Ireland

² Mathematical Institute, University of Oxford, Oxford, Oxfordshire OX2 6GG, United Kingdom of Great Britain and Northern Ireland

³ Mathematical Institute, Faculty of Mathematics and Physics, Charles University, Prague 186 75, Czech Republic

E-mail: kamran.pentland@ukaea.uk

Received 30 April 2025, revised 17 July 2025

Accepted for publication 23 July 2025

Published 1 August 2025



Abstract

The Grad–Shafranov (GS) equation is a nonlinear elliptic partial differential equation that governs the ideal magnetohydrodynamic equilibrium of a tokamak plasma. Previous studies have demonstrated the existence of multiple solutions to the GS equation when solved in idealistic geometries with simplified plasma current density profiles and boundary conditions. Until now, the question of whether multiple equilibria might exist in real-world tokamak geometries with more complex current density profiles and integral free-boundary conditions (commonly used in production-level equilibrium codes) has remained unanswered. In this work, we discover multiple solutions to the static forward free-boundary GS problem in the MAST-U tokamak geometry using the validated evolutive equilibrium solver FreeGSNKE and the deflated continuation algorithm. By varying the plasma current, current density profile coefficients, or coil currents in the GS equation, we identify and characterise distinct equilibrium solutions, including both deeply and more shallowly confined plasma states. We suggest that the existence of even more equilibria is likely prohibited by the restrictive nature of the integral free-boundary condition, which globally couples poloidal fluxes on the computational boundary with those on the interior. We conclude by discussing the implications of these findings for wider equilibrium modelling and emphasise the need to explore whether multiple solutions are present in other equilibrium codes and tokamaks, as well as their potential impact on downstream simulations that rely on GS equilibria.

* Author to whom any correspondence should be addressed.



Original Content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Keywords: multiple solutions, Grad–Shafranov, MHD equilibria, FreeGSNKE, deflated continuation, MAST-U

(Some figures may appear in colour only in the online journal)

1. Introduction

1.1. Motivation and aims

The solution to the static forward free–boundary Grad–Shafranov (GS) problem describes the magnetohydrodynamic (MHD) equilibrium state of a magnetically-confined, toroidally symmetric plasma in a tokamak fusion device. Obtaining accurate solutions to this problem is a critical requirement for experimental tokamak plasma design and operation. The GS equation is a nonlinear elliptic partial differential equation (PDE) that can be solved in different tokamak geometries with a wide range of plasma current density parametrisations. Due to its nonlinearity, the GS problem may support multiple, isolated solutions. However, computational methods used for the simulation or reconstruction of free–boundary GS equilibria almost always return only a single solution to the problem. This solution is then used in a computational plasma simulation pipeline (e.g. in stability analysis, scenario modelling, and tokamak optimisation), without consideration of other possible solutions and how they might impact subsequent analyses.

In this paper, we build upon the work of Ham and Farrell [17], which identified multiple numerical solutions to a contrived fixed-boundary GS problem. We demonstrate the existence of multiple solutions to the static forward (integral) free-boundary GS problem for real-world plasmas in the MAST-U tokamak. We aim to:

- (i) Find multiple solutions to the static forward GS problem on MAST-U using *FreeGSNKE* and the *deflated continuation* algorithm;
- (ii) Investigate how they change (and possibly bifurcate) when certain parameters in the GS equation are varied.

The presence of multiple GS solutions in real-world tokamaks could have significant implications for a number of different areas across plasma simulation. For example, this may also occur during the integrated modelling of plasma scenarios [26], such as those on ITER or STEP [9], which require the time-dependent evolution of an equilibrium alongside coil currents and plasma profile parameters (e.g. using transport codes). Missing bifurcations points and therefore the presence of multiple solutions during such a simulation could potentially undermine scenario design and operational planning. These implications highlight the need to explore and identify different solution branches during forward/inverse equilibrium simulations.

To solve the forward GS problem here, we will make use of the Python-based, dynamic free–boundary toroidal plasma equilibrium solver *FreeGSNKE* [3]. This solver has the ability

to carry out both (static/dynamic) forward and (static-only) inverse free–boundary GS equilibrium calculations. The static forward solver has previously been validated against the equilibrium codes *Fiesta* [12] and *EFIT++* [7, 23, 24] on MAST-U plasma discharges [25] and has been used to emulate plasma scenarios for plasma control [1]. Most importantly, and like almost all other equilibrium codes, *FreeGSNKE* currently returns a single solution to the GS equation upon simulation.

To systematically search for multiple solutions to the GS problem, we will use the deflated continuation algorithm [14]. This algorithm is able to identify multiple solutions to a PDE (when varying a parameter) by modifying the nonlinear problem to guarantee non-convergence to known solutions, under certain conditions. This means that when the nonlinear solver (e.g. Newton’s method) is applied again, if the solver converges then it has discovered another, distinct solution. This algorithm enables the user to construct (possibly disconnected) bifurcation diagrams, which show how the number of solutions change with the PDE parameter being varied. It has already proven successful in a number of different application areas, identifying hundreds of new stable/unstable equilibria for magnetic rotors [10] and discovering multiple experimentally-observed solutions in smectic liquid crystals [35]. We will explore the static forward GS problem, *FreeGSNKE*, and deflated continuation in more detail below.

By harnessing the capabilities of both *FreeGSNKE* and deflated continuation, we will search for multiple solutions by varying certain parameters in the (toroidal) plasma current density function of the GS equation. We also assess whether varying the current in one of the active poloidal field coils (that control the shape of the plasma) also affects the number of solutions found. We will do this for a single time slice during the flat-top (steady state) phase of a MAST-U shot and generate bifurcation diagrams for these varying parameters.

1.2. Related work

Early studies of the GS equation primarily focused on proving the existence and uniqueness of solutions in simplified domains, under restricted boundary conditions, and with reduced plasma current density profiles. Under these restrictions, one can formulate an eigenvalue problem from the GS equation with a free boundary⁴ to prove solution existence, uniqueness (for cases with small eigenvalues), and non-uniqueness (for cases with larger eigenvalues). This was work initially carried out by Temam [32, 33] and Schaeffer [27], then, in a slightly more general setting, by Ambrosetti and

⁴ These problems typically assign a free, but constant, value to the flux on the computational boundary, whereas in real-world GS problems, the flux is allowed to vary (spatially) on the boundary, see (2.2).

Mancini [2] and Berestycki and Brezis [6]. Similar problems were later revisited by Bartolucci *et al* [5] and Jeyakumar *et al* [22].

In slightly more real-world settings, some work has been done to provide analytical derivations of multiple GS equilibria (again, under simplifying conditions and with constant unknown flux on the boundaries). For example, in a cylindrical plasma with polynomial current profiles (similar to those used in modern equilibrium codes), Turnbull [34] identified settings in which variations in certain parameters (e.g. plasma current, flux on the conducting wall, wall radius) yield up to two GS solutions.

In a similar setting, but with stepped current density profiles, Ilgisonis and Pozdnyakov [19] demonstrate that variations in the magnitude and location of the stepped profile leads to a fold bifurcation, resulting in up to three solutions. Furthermore, they suggest that bifurcations in GS equilibria may not appear in numerical simulations due to the current density normalisation process (present in almost all equilibrium solvers for numerical stability), which they say restricts the poloidal flux values that solvers can identify. In section 2, we discuss this process and provide an experiment in section 4 that demonstrates this statement might hold for Picard-based solvers but not for Newton-based solvers.

Other works include that of Solano [30] in which the criticality (bifurcation) of GS solutions is discussed for polynomial current density profiles. Similarly to Ilgisonis and Pozdnyakov [19], Schnack [28] also finds evidence of up to two GS equilibria in the case of a tall thin plasma column with stepped current density profiles.

Despite these theoretical studies and the widespread use of various equilibrium codes, there has been surprisingly little numerical investigation into the existence of multiple solutions to the GS equation for free-boundary equilibria in settings relevant to real-world tokamak operations. A step towards this goal was made by Ham and Farrell [17], in which they identify multiple solutions to a fixed-boundary GS problem numerically. Using the Firedrake finite element package [18], they are able to use deflated continuation to consider much more physically realistic plasma boundary shapes and systematically search for multiple solutions by varying parameters such as the aspect ratio, elongation, and triangularity. Despite this progress, we still lack an investigation for truly free-boundary GS formulations that incorporate the integral boundary conditions, more realistic internal plasma current profiles, and external conductor currents used in real-world tokamak experiments.

1.3. Paper structure

The rest of this paper is organised as follows. In section 2 we present the static forward GS problem, how it is solved in FreeGSNKE, and which parameters we will vary during the search for multiple solutions. In section 3 we outline the deflated continuation algorithm, remarking on a number of algorithmic parameter choices that need to be made in order to efficiently search for solutions. The multiple solutions found

using FreeGSNKE and deflated continuation will be presented and analysed in section 4. Finally in section 5 we discuss what these results mean for the future of GS equilibrium simulation and propose some ideas for future work in this area.

2. The static forward free-boundary Grad-Shafranov problem

The GS equation is a nonlinear elliptic PDE used ubiquitously in plasma equilibrium modelling for describing the (static, time-independent) balance between magnetic and plasma pressure forces in ideal MHD equilibria [16, 29]. It governs the poloidal flux $\psi(R, Z)$, which has units [Weber/2 π], within a two-dimensional cross-section of a toroidally (ϕ) symmetric tokamak device and is given by

$$\Delta^* \psi = -\mu_0 R J_\phi(\psi, R, Z), \quad (R, Z) \in \Omega, \quad (2.1)$$

where $\Delta^* := R\partial_R R^{-1}\partial_R + \partial_{ZZ}$ is a linear elliptic operator and $\mu_0 = 4\pi \times 10^{-7}$ [N/A²] is the magnetic permeability of free space. Here, (R, ϕ, Z) denotes the cylindrical coordinate system.

The poloidal flux ψ is the sum of two terms $\psi := \psi_p + \psi_c$, where ψ_p and ψ_c are contributions from the plasma and the (toroidally symmetric) conducting metal structures external to the plasma,⁵ respectively. The toroidal current density $J_\phi(\psi, R, Z) := J_p(\psi, R, Z) + J_c(R, Z)$ also contains contributions from both the plasma J_p and the external conductors J_c . The dependence on ψ is where part of the nonlinearity in the static forward GS problem originates. The computational domain is a pre-specified rectangular grid denoted by $\Omega := \Omega_p \cup \Omega'_p$, where Ω_p defines the plasma region (whose boundary is to be determined) and Ω'_p refers to its complement (see figure 1).

An integral (Dirichlet) free-boundary condition accompanies (2.1):

$$\psi \Big|_{\partial\Omega} = \int_{\Omega} G(R, Z; R', Z') J_\phi(\psi, R', Z') \, dR' dZ', \quad (2.2)$$

which gives the flux on the computational boundary $\partial\Omega$ produced by all non-zero toroidal current sources in Ω_p and Ω_c (see (2.7) for external conductor currents). The function G is the (analytically) known Green's function for the operator Δ^* and links points on the boundary with the toroidal current sources. Refer to Takeda and Tokuda [31, section 3.1] for further details on G and to Jardin [20, chapter 4.6.4] for how this integral can be calculated efficiently.

It should be noted that the plasma boundary $\partial\Omega_p$ is defined by the (last) closed (R, Z) contour of ψ that passes through the X-point closest to the magnetic axis of the plasma⁶. The (non-linear) identification of $\partial\Omega_p$ is required in order to calculate J_p

⁵ The external conductors are the active poloidal field coils (whose currents are used to shape and control the plasma) and the passive conducting structures (whose currents are induced by the plasma and the active coils).

⁶ The X-point and magnetic axis (sometimes referred to as an O-point) are identified by finding the critical points of ψ (see Jeon [21, section 5]).

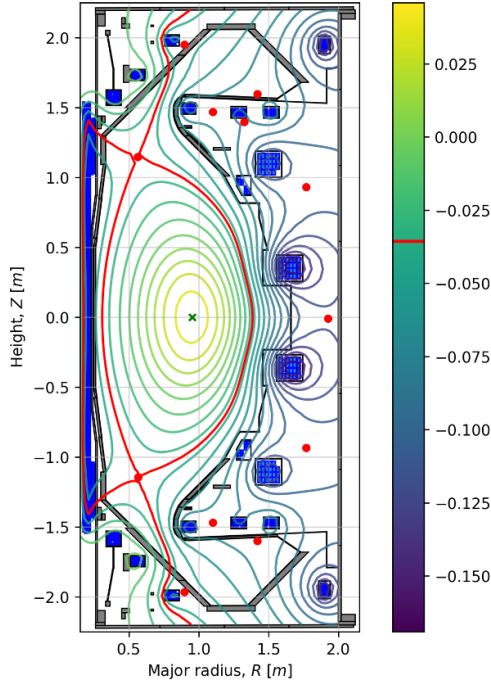


Figure 1. FreeGSNKE-simulated equilibrium of MAST-U shot 45272 ($t = 0.79854$ s) with ψ contours (see colour bar) shown in domain $\Omega = [0.06, 2] \times [-2.2, 2.2]$. Key features include the plasma region Ω_p , which is enclosed by the last closed flux surface (solid red), the X-points (red dots), and the magnetic axis (green cross). Also shown are the twelve active poloidal field coils (dark blue), the passive structures (dark grey), and the wall/limiter (solid black).

(see (2.3)) and therefore solve the forward GS problem (2.1) and (2.2).

2.1. The toroidal current density

Here, we outline how the plasma and external conductor current density contributions to J_ϕ are defined and, more crucially, highlight which of their parameters we vary later on in deflated continuation to identify multiple GS equilibria.

2.1.1. Plasma current density. The plasma current density is governed by the distribution of charged particles within the plasma and generates a poloidal magnetic field that contributes to the confinement of the plasma. It is non-zero only within Ω_p , taking the form

$$J_p(\psi, R, Z) = R \frac{dp}{d\psi} + \frac{1}{\mu_0 R} F \frac{dF}{d\psi}, \quad (R, Z) \in \Omega_p, \quad (2.3)$$

where $p := p(\psi)$ is the plasma pressure profile and $F := F(\psi) = RB_\phi$ is the toroidal magnetic field profile (B_ϕ is the toroidal component of the magnetic field).

The full specification of the forward GS problem (2.1) and (2.2) requires specifying p' and FF' to determine the internal profiles of the pressure and toroidal current within the

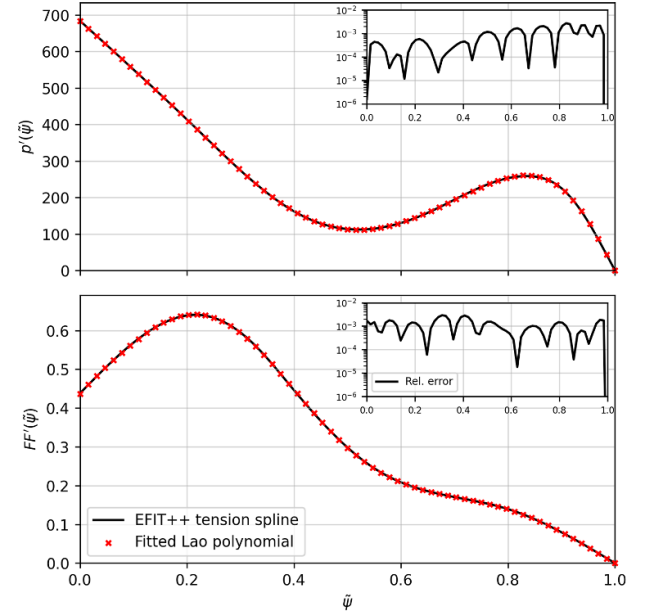


Figure 2. Plasma current density profiles p' (top) and FF' (bottom) used to simulate the equilibrium in figure 1. Shown are the original tension spline profiles obtained from the EFIT++ reconstruction (solid black) and the profiles fit using the Lao polynomial parameterisation (2.4) (red dots). The inset plots display the relative error between the two different parameterisations.

plasma⁷. For real-world tokamak plasmas, these can take various nonlinear forms, making analytical progress difficult and numerical computation challenging.

Here, we use the (Lao) polynomial profile parametrisation:

$$\begin{aligned} \frac{dp}{d\psi} &= \sum_{i=0}^{n_p} \alpha_i \tilde{\psi}^i - \bar{\alpha} \tilde{\psi}^{n_p+1} \sum_{i=0}^{n_p} \alpha_i, \\ F \frac{dF}{d\psi} &= \sum_{i=0}^{n_F} \beta_i \tilde{\psi}^i - \bar{\beta} \tilde{\psi}^{n_F+1} \sum_{i=0}^{n_F} \beta_i \end{aligned} \quad (2.4)$$

where $\alpha_i, \beta_i \in \mathbb{R}$ are coefficients, $\bar{\alpha}, \bar{\beta} \in \{0, 1\}$ are Booleans (governing edge conditions at $\tilde{\psi} = 1$), and $n_p, n_F \in \mathbb{N} \cup 0$ dictate the polynomial orders [24]. Also note

$$\tilde{\psi} = \frac{\psi - \psi_a}{\psi_b - \psi_a} \in [0, 1], \quad (2.5)$$

defines the normalised poloidal flux (ψ_a and ψ_b being the flux on the magnetic axis and plasma boundary, respectively).

In the numerical experiments with deflated continuation (to follow in section 4) we will vary the coefficients α_i, β_i . Figure 2 displays the profiles used to obtain the equilibrium in figure 1. In addition, we will vary the total plasma current I_p , whose value is not strictly required as an input to solve (2.1) and (2.2), but is often prescribed to normalise (2.3).

⁷ Fitting these profiles is part of solving the GS equilibrium reconstruction problem—see Jardin [20, section 4.7] for an introduction.

This simply involves multiplying the right hand side of (2.3) by

$$I_p \left(\int_{\Omega_p} J_p(\psi, R, Z) \, dRdZ \right)^{-1}. \quad (2.6)$$

2.1.2. External conductor current density. To close the system we must also specify the current density J_c within the metal conductors external to the plasma in the tokamak, i.e. the active poloidal field coils and the passive (non-active) structures, as shown in figure 1. These conductors generate magnetic fields for shaping and controlling the plasma position and stability. The current density produced by N_c external conductors is modelled as

$$J_c(R, Z) = \sum_{j=1}^{N_c} \frac{I_j^c(R, Z)}{A_j^c}, \quad (R, Z) \in \Omega, \\ I_j^c(R, Z) = \begin{cases} I_j^c & \text{if } (R, Z) \in \Omega_j^c, \\ 0 & \text{elsewhere,} \end{cases} \quad (2.7)$$

where Ω_j^c , I_j^c , and A_j^c are the domain region, current, and cross-sectional area of the j th conductor, respectively. Note that this term can be calculated explicitly before solving (2.1) and (2.2).

In section 4, we will investigate how the equilibrium solutions vary as we change a current in one of the active poloidal field coils in MAST-U.

2.2. Numerical solution

Solving (2.1) and (2.2) with FreeGSNKE requires a number of different inputs specific to the MAST-U machine. Firstly, we require a machine description of MAST-U that includes the position, size, orientation, and polarity of the active poloidal field coils, passive structures, and the limiter/wall (that will confine the boundary of the plasma during the simulation). Then, to simulate a specific equilibrium (at a given time slice of a shot) we require the plasma profile coefficients/parameters for (2.4), the total plasma current I_p for the normalisation in (2.6), and the currents measured in the external conductors for (2.7). The input data required to do this comes from an EFIT++ reconstruction of the plasma equilibrium.

In this paper, we use data obtained from a magnetics plus motional Stark effect EFIT++ reconstruction. This reconstruction code uses measured coil currents, plasma current, magnetic fields, and motional Stark effect data in order to find the ‘best’ fit for the aforementioned parameters [11]. We should note that this type of EFIT++ reconstruction actually fits parameters to the tension spline parameterisation of the p' and FF' profiles (see Pentland *et al* [25, appendix A]), however, these parameters are difficult to use within the deflated continuation framework. To suit our needs, we instead fit the α and β coefficients of the Lao polynomial profiles (2.4) to the tension spline profiles from EFIT++—this fit (and the relative errors) can be

seen in figure 2. For profiles of this complexity, we required polynomials up to order $n_p = n_F = 9$.

In FreeGSNKE, the static forward GS problem (2.1) and (2.2) is solved in its residual form

$$F(\psi; \lambda) \equiv \Delta^* \psi + \mu_0 R J_\phi(\psi, R, Z; \lambda) = 0, \quad (2.8)$$

where we have slightly abused notation by omitting the boundary condition (though it is indeed applied). Here λ denotes the (scalar) parameter in J_ϕ that we will vary when searching for multiple solutions with deflated continuation.

Once discretised (using fourth-order finite differences), (2.8) is solved using a Jacobian-free Newton–Krylov (NK) method (see Amorisco *et al* [3, appendix 1]) with an appropriate initial guess for the plasma flux ψ_p (recall ψ_c is known *a priori* to simulation). If no initial guess for ψ_p is provided, FreeGSNKE generates one by default with ellipse-shaped flux contours, the magnitude of which are adaptively scaled up such that the total flux produces a magnetic axis and an X-point within the confining limiter geometry. The NK method then iterates until a relative convergence tolerance

$$\frac{\max |F(\psi; \lambda)|}{\max(\psi) - \min(\psi)} < \varepsilon, \quad (2.9)$$

is met, returning a *single* solution $\psi = \psi_p + \psi_c$ to the problem.

Searching for multiple solutions to (2.8) by using a set of different initial guesses ψ_p is difficult for a number of reasons. Firstly, it is difficult to efficiently generate such a set and, even if we could, there is no way to guarantee that the input space would be well covered by a given number of them. Secondly, many of these initial guesses would most likely converge to the same solution and many may not converge at all. This task is both cumbersome and computationally inefficient, hence we now explain how we can systematically search for multiple *distinct* solutions while varying the parameter λ in (2.8) using the deflated continuation algorithm.

3. Deflated continuation

To search for multiple solutions to the static forward GS problem, as well as potential bifurcation points in parameter space, we use the deflated continuation algorithm first proposed by Farrell *et al* [14].

3.1. How it works

For simplicity we consider the discretised problem. The purpose of deflated continuation is to locate solutions $u \in \mathbb{R}^m$ to

$$F(u; \lambda) = 0, \quad (3.1)$$

where $F: \mathbb{R}^m \times \mathbb{R} \rightarrow \mathbb{R}^n$ is a nonlinear function that represents the residual of a PDE problem (e.g. (2.8)) and $\lambda \in \mathbb{R}$ is a parameter in the equations. More specifically, it will locate a set of distinct solutions $\{u_1^*, u_2^*, \dots\}$ to (3.1) for *each* of the parameter values considered, typically $L + 1$ equally spaced ($\Delta\lambda$)

values of λ in a chosen interval $[\lambda_0, \lambda_L]$. An important advantage over other methods for computing multiple solutions (e.g. pseudo-arclength continuation and branch switching) is that deflated continuation is able to compute disconnected bifurcation diagrams in which solutions on different branches may not meet at bifurcation points (see Farrell *et al* [14, figure 1.2] for an example). It can do this by combining the power of both *deflation* and *continuation*.

Suppose we have found a solution u_1^* to (3.1), for fixed λ , using a suitable nonlinear root finding method (e.g. NK). To try to identify more solutions, we can use the deflation technique [8, 13] to modify the operator (3.1) such that we instead solve

$$M(u; u_1^*) F(u; \lambda) = 0, \quad (3.2)$$

where

$$M(u; u_1^*) = \left(\frac{1}{\|u - u_1^*\|_2^p} \right) + \sigma, \quad (3.3)$$

is the *deflation operator* with power and shift parameters $p > 0$ and $\sigma > 0$, respectively ($\|\cdot\|_2$ denotes the Euclidean norm). Under mild regularity conditions, solving (3.2) ensures the nonlinear solver does not return the known solution u_1^* (using the same initial guess as before) but rather a distinct solution u_2^* (if the method converges, which is not guaranteed). The problem in (3.2) can subsequently be deflated again using known solutions $\{u_1^*, \dots, u_N^*\}$ such that

$$\prod_{i=1}^N M(u; u_i^*) F(u; \lambda) = 0, \quad (3.4)$$

is solved until no more are found within a specified number of nonlinear iterations.

To initialise the algorithm at the first λ , (3.1) is solved using the chosen nonlinear solver and the solution(s)⁸ recorded. Then, for $\lambda + \Delta\lambda$, deflated continuation carries out two separate stages for solving (3.1): continuation and exploration.

Continuation stage: Each known solution (from step λ) is used as an initial guess in the nonlinear solver to try to find the corresponding solution at $\lambda + \Delta\lambda$. For each solution successfully continued,⁹ the new solution is used to deflate the residual function as in (3.4).

Exploration stage: Again, each known solution (from step λ) is used as an initial guess to solve (3.4) to try to locate additional new solutions (at $\lambda + \Delta\lambda$). Again, if any new solutions are found, they are used to deflate (3.4) before considering the next initial guess.

⁸ If any other solutions are known or found, they can also be used at this stage.

⁹ The implicit function theorem states that a solution branch should only cease to exist if the Fréchet derivative of the residual function (at that state) is zero. If not, then it suggests that the nonlinear solver has failed due to lack of iterations, instability, etc.

All solutions found for $\lambda + \Delta\lambda$ are then stored, ready for use when considering the next value of λ .

3.2. Numerical implementation

The original implementation of deflated continuation [15] is built using the Firedrake [18] and FEniCS [4] libraries. Here, we use a purpose-built Python implementation, tailored specifically for use with FreeGSNKE. This is necessary as FreeGSNKE uses its own finite difference scheme and its own purpose-built NK solver for tackling (3.1).

Before each simulation, we need to choose a few key parameters within both deflated continuation and FreeGSNKE. When solving (3.1) (or (3.4)) with the NK solver, we use a relative convergence tolerance of $\varepsilon = 10^{-6}$ in (2.9), a maximum of 150 NK iterations, and a scaled Newton step size of 1.2 (all other settings are FreeGSNKE defaults). Given there is no systematic way of choosing the free parameters (p, σ) in (3.3) (see [13] for a discussion on this), we set them independently for each experiment in section 4 depending on which combination works well—the best default choice was found to be $(p, \sigma) = (1, 0.05)$. For each parameter λ under consideration, the interval $[\lambda_0, \lambda_L]$ and the chosen step size $\Delta\lambda$ will vary. In addition, we explore each side of the starting value λ_0 , i.e. we explore $[\lambda_{-L}, \lambda_0]$ using step $-\Delta\lambda$.

4. Searching for multiple solutions

In this section, we carry out our search for multiple solutions to the static forward GS problem laid out in section 2 using deflated continuation. Here, we solve (2.8) at time $t = 0.79854$ s of MAST-U shot 45272 using $m = 65^2 = 4425$ spatial grid points (recall the equilibrium in figure 1 and profiles used in figure 2). This is a double-null plasma with a flat-top current of approximately 750 kA (heated by two neutral beam injection systems) and a conventional divertor configuration.

The inputs required to solve for the equilibrium at this time slice are described in section 2.2 and will be used as the starting parameters (λ_0) in our deflated continuation experiments. The parameters we can vary for this equilibrium are the:

- Coefficients α_i, β_i in the 9th order Lao polynomials (2.4);
- Plasma current I_p in (2.6);
- Coil currents $I_j^c, j \in \{1, \dots, 12\}$, in (2.7).

We will investigate whether any bifurcations occur when varying some of these parameters (separately) over a suitable interval. To initialise deflated continuation at λ_0 , we use the plasma flux ψ_p found by FreeGSNKE during a standard solve (i.e. what we find in figure 1).

To plot bifurcation diagrams for vector solutions (e.g. ψ), it is common to calculate scalar-valued *functionals* related to the solution. While many functionals of ψ are available (e.g. inner/outer midplane radii, strikepoints,

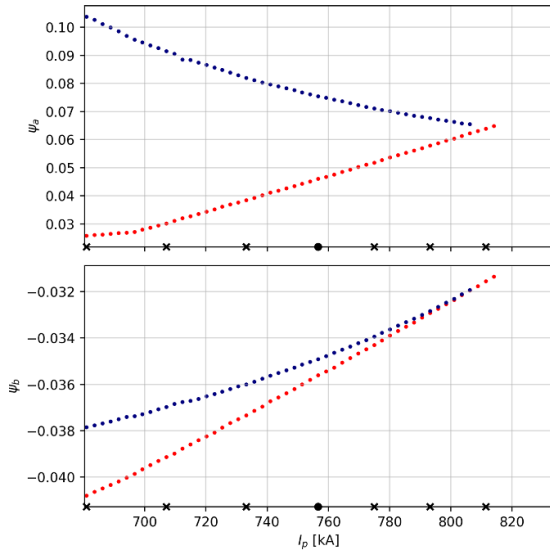


Figure 3. Bifurcation diagrams for ψ_a (top) and ψ_b (bottom) when varying plasma current I_p . Different solution branches are indicated by different colours and the initial λ_0 value is indicated by the black dot on the x -axis. The solutions in figure 4 are plotted in the same colours at the values of I_p indicated by the black crosses (and the dot) on the x -axis.

X-points), we find that the most informative are ψ_a and ψ_b (recall (2.5)).

4.1. Experiments

4.1.1. Varying I_p . The first parameter we test with deflated continuation is $\lambda = I_p$. Starting with the initial value $\lambda_0 \approx 756$ kA, we run deflated continuation both ‘forward’ and ‘backward’ to cover a range of I_p values up to 10% either side of λ_0 . In figure 3, we can see that deflated continuation reveals two distinct solution branches that diverge from one another as I_p decreases away from λ_0 and converge together above it. The lower panel (displaying ψ_b), shows the branches merging around $I_p \approx 806$ kA and the remaining solution branch being lost beyond $I_p \approx 814$ kA.

This process can be seen more clearly when we plot the separatrices of the solutions ψ at a number of different I_p values in figure 4. For λ_0 , the centre panel displays both a deeply confined diverted plasma equilibrium (i.e. the one found initially by FreeGSNKE in figure 1) and a more shallowly confined limited plasma on the outboard side. Each separatrix corresponds to the respective solution branch colour shown in figure 3. As I_p increases from left to right in figure 4, we see the size of each plasma core increasing dramatically until the outboard limited solution merges into the diverted solution causing a bifurcation. Beyond this point, the diverted equilibrium exists only for a small increase in I_p , expanding further until it can no longer be contained by the limiter/wall and is lost. For $I_p \lesssim 700$ kA, the deeply confined equilibrium switches from a diverted plasma to a limited one.

We should note here that we also ran the same experiment (not shown) without the plasma current normalisation process

(recall (2.6)) and found identical results to those shown here. This demonstrates that multiple solutions can indeed persist with or without the current normalisation process when using a Newton-based solver.

4.1.2. Varying α_i or β_i . In figure 5, we vary the coefficient α_2 in the p' profile (2.4) by up to 175%—notice the large magnitude of α_2 in the left panels. As before, we can see the two co-existing solutions we saw when varying I_p (see right panel) and we observe that the outboard limiter solution is lost for $\alpha_2 \lesssim -0.03 \times 10^6$. We can also see the upward curve on the lower branch (of the ψ_a panel) for $\alpha_2 \lesssim 0.25 \times 10^6$ indicating the transition of the diverted solution into an inboard limited plasma. In addition, we see that the two solutions co-exist with almost exactly the same ψ_b value at $\alpha_2 \approx 1 \times 10^6$ (in the ψ_b panel). Note that we do not see any bifurcations in this case.

In figure 6, we vary the β_5 parameter and, again, see the two same solutions coexisting before bifurcating in a manner very similar to what we saw in with I_p in figure 3 (though the solutions merge and are then lost as β_5 is decreased).

4.1.3. Varying a coil current I_j^c . In this experiment, we run a deflated continuation simulation in which we instead vary the current in one of the poloidal field coils, specifically the D1 coil—see Pentland *et al* [25, figure 1]. As shown in figure 7, the two solutions are present once again but do not vary significantly, even when varying the coil current by up to 175%. In other experiments (not shown), varying currents in the other coils had a similar (lack of) effect on the two solutions.

4.1.4. Picard-based solvers and I_p normalisation. Many widely used equilibrium codes rely on (possibly stabilised) Picard iterations rather than Newton-based solution schemes. It is therefore worth carrying out a brief investigation into whether the two solution branches identified here can be detected using a Picard scheme. Here we do not use deflated continuation and instead fix the plasma profile parameters and coil currents (as in section 2). Figure 8 displays the Euclidean norm of the residual (3.1) against the iteration number when using either FreeGSNKE’s NK method (solid lines) or a Picard scheme (dashed lines). For the NK method, we initialise the solver with two different guesses for ψ_p , each of which converges to either the diverted (left panels) or limited solution branch (right panels). We also explore the impact of enabling (upper panels) or disabling (lower panels) I_p normalisation, a typical setting in most equilibrium codes which fixes the total plasma current—recall (2.6). In all four cases, we see that the NK method converges to a solution, in line with the previous results. We use intermediate solutions provided by the NK method to initialise the Picard method (black crosses). Given an initial guess ψ^0 , the k -th Picard iteration is given by

$$\psi^{k+1}(R, Z) = \psi^k(R, Z) - \frac{1}{2} [F(\psi^k(R, Z)) + F(\psi^k(R, -Z))],$$

where we symmetrise the residual update to mitigate the typical instability of the Picard iterations.

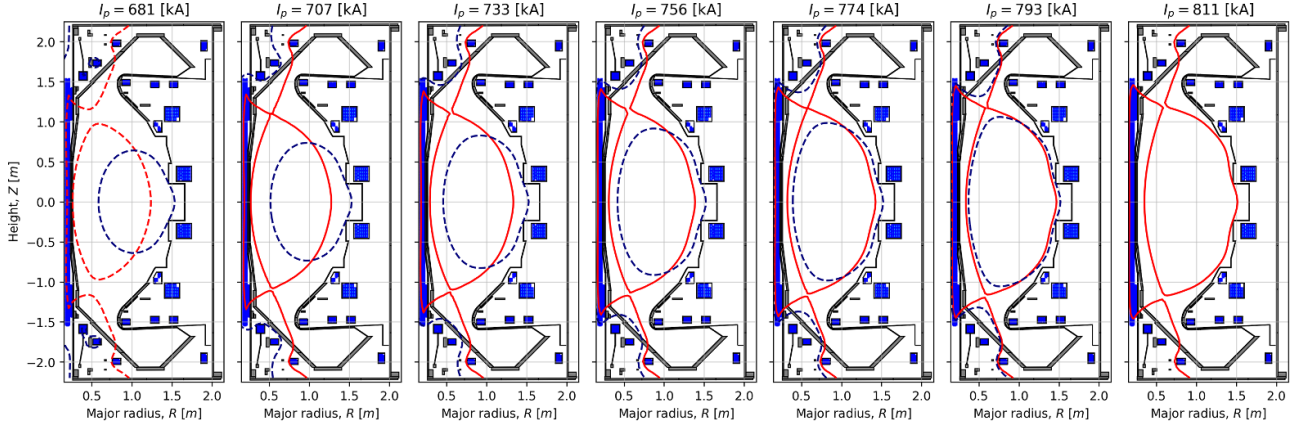


Figure 4. Separatrices of the multiple equilibria (red and blue) at increasing values of I_p (whose values are indicated by the black crosses and dot on the x -axis of figure 3). A dashed separatrix line indicates that the plasma is limited (i.e. touching the wall) while a solid line indicates it is diverted.

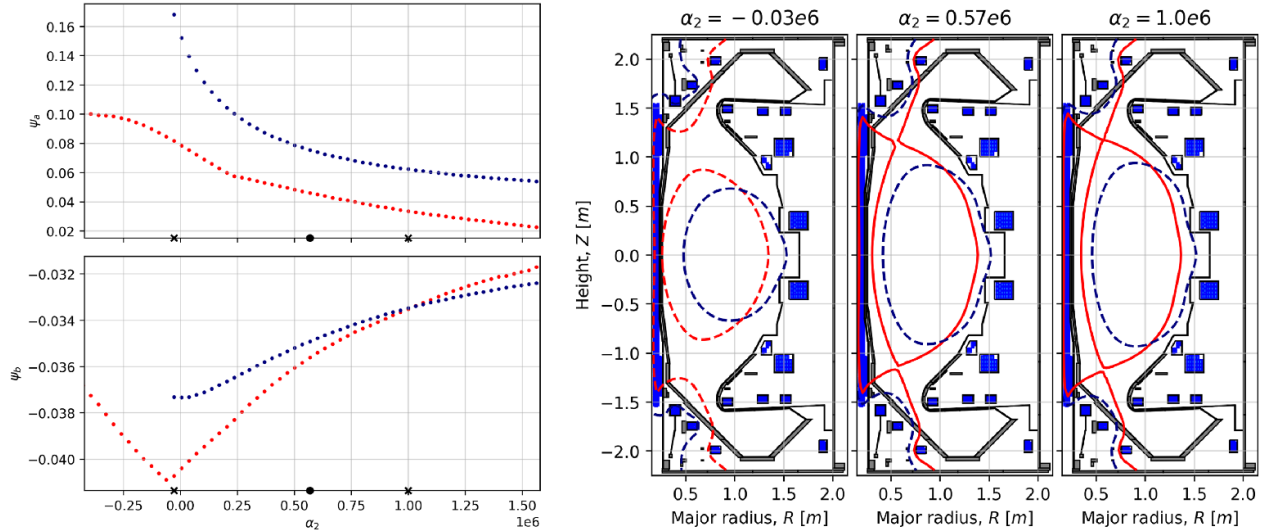


Figure 5. Left: bifurcation diagrams for ψ_a and ψ_b (left) when varying α_2 . Different solution branches are indicated by different colours and the initial λ_0 value is indicated by the black dot on the x -axis. Right: separatrices of the multiple equilibria (red and blue) at increasing values of α_2 (whose values are indicated by the black crosses and dots on the x -axis of the left panel). A dashed separatrix line indicates that the plasma is limited (i.e. touching the wall) while a solid line indicates it is diverted.

While not an exhaustive study, figure 8 suggests that only one of the two branches is a stable fixed point of the Picard iterations. With I_p normalisation enabled, the Picard method consistently converges to the diverted solution for a range of left-biased guesses but fails to converge to the limited solution when the initial guess is biased to the right. Conversely, when I_p normalisation is disabled, the Picard method instead converges to the limited solution and fails to find the diverted one. This supports earlier studies [19] (recall section 1), which had mentioned the role of I_p normalisation in Picard iterations. This also suggests that, in practice, the impact of multiple solutions may vary across studies, depending on the equilibrium code and the nonlinear solver in use.

5. Discussion and future work

We have demonstrated that the static forward GS problem can exhibit multiple solutions when solved in a physically-relevant setup, with an integral free-boundary condition, realistic plasma current density profiles, and external conductor currents in the MAST-U tokamak. By utilising both FreeGSNKE and deflated continuation, our numerical experiments revealed that two distinct solution branches exist when varying parameters such as the plasma current (I_p), plasma current density profile coefficients (α_i , β_i), or coil currents (I_j^c). The solutions identified had significantly different shapes and positions, with one being more deeply confined (and for the most

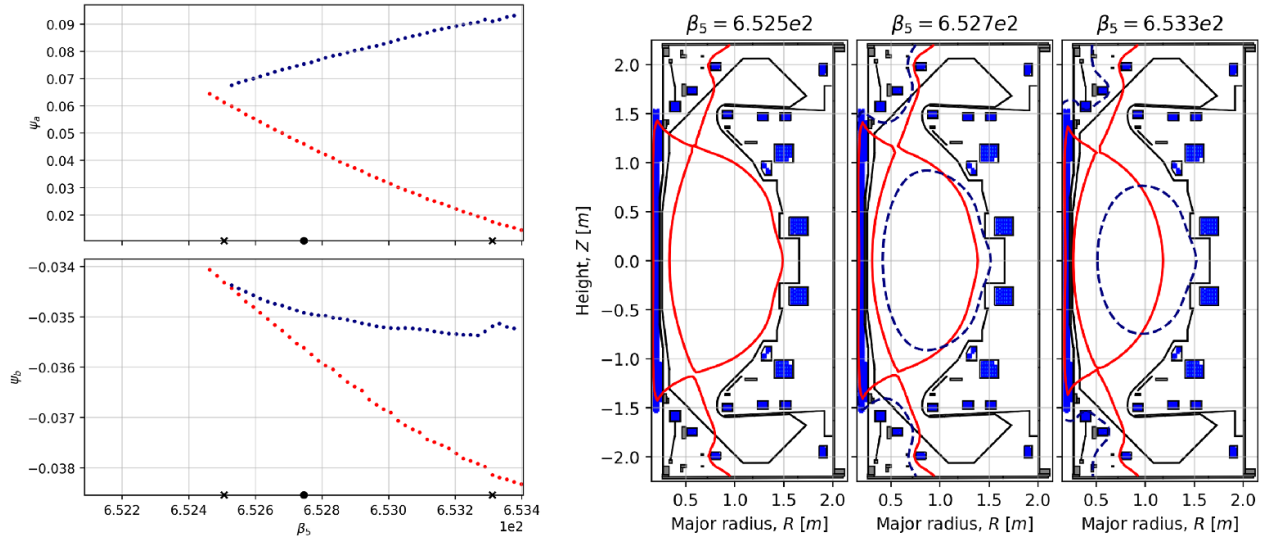


Figure 6. Left: bifurcation diagrams for ψ_a and ψ_b (left) when varying β_5 . Different solution branches are indicated by different colours and the initial λ_0 value is indicated by the black dot on the x-axis. Right: separatrices of the multiple equilibria (red and blue) at increasing values of β_5 (whose values are indicated by the black crosses and dot on the x-axis of the left panel). A dashed separatrix line indicates that the plasma is limited (i.e. touching the wall) while a solid line indicates it is diverted.

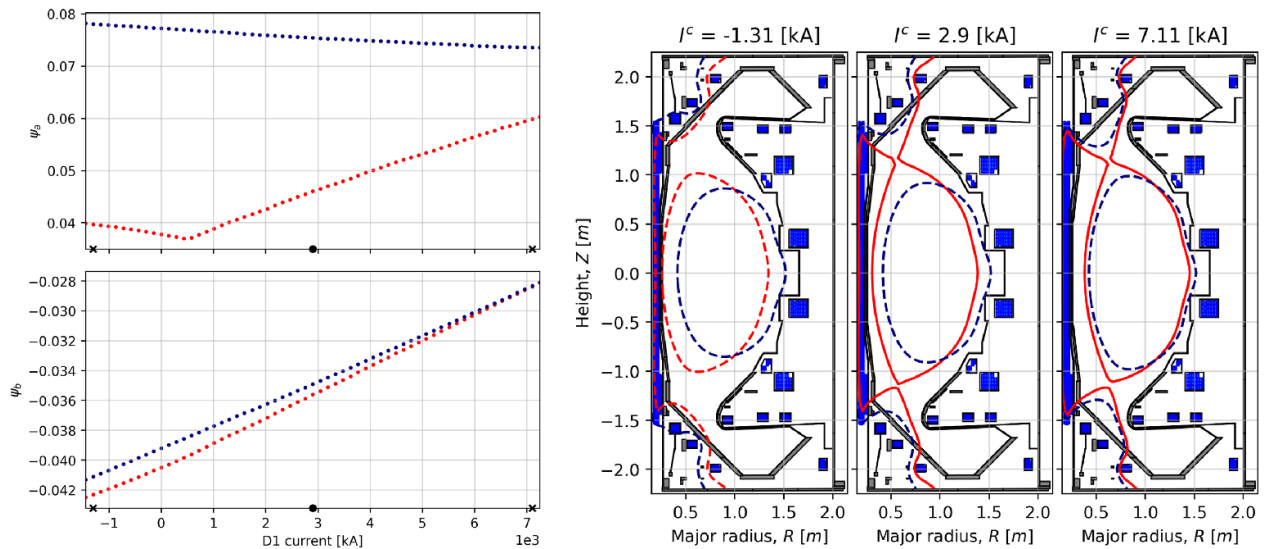


Figure 7. Left: bifurcation diagrams for ψ_a and ψ_b (left) when varying current in the D1 coil. Different solution branches are indicated by different colours and the initial λ_0 value is indicated by the black dot on the x-axis. Right: separatrices of the multiple equilibria (red and blue) at increasing values of D1 current (whose values are indicated by the black crosses and dot on the x-axis of the left panel). A dashed separatrix line indicates that the plasma is limited (i.e. touching the wall) while a solid line indicates it is non-limited.

part diverted) while the other was more shallowly confined (and always limited).

One key difference between the results presented here and in prior studies investigating the presence of multiple GS solutions is the restriction imposed by the integral boundary condition (2.2). Unlike boundary conditions where the solution takes a constant (but free) value on the domain boundary, (2.2) globally couples the boundary flux values with those on the domain's interior. Consequently, this significantly constrains the solution space for ψ and not only restricts the boundary flux values but also strongly influences the internal structure of possible solutions. This perhaps makes the emergence of

even more equilibrium solutions more difficult, though we would not rule out their presence without wider study on free-boundary equilibria in other tokamaks (and perhaps using other forward equilibrium codes).

These findings suggest that care must be taken when using forward GS solvers as they do not currently account for the presence of multiple equilibria and may converge to either of the equilibria we have seen here (depending on the inputs and the initial plasma flux guess). As hinted in section 4.1.4, only one solution branch appears to be a stable fixed point for Picard-based solvers in our experiments. While, on the one hand, this suggests that Picard solvers may be less prone to

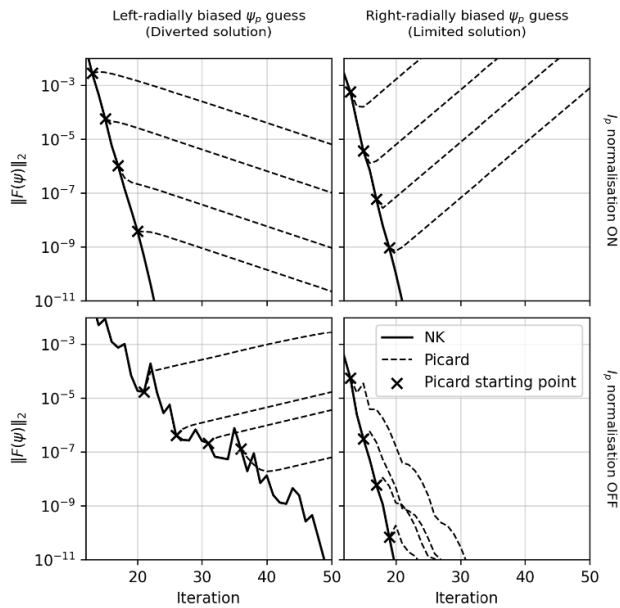


Figure 8. Euclidean norm of GS residual (3.1) against iteration number when using FreeGSNKE’s NK method (solid black) or a Picard scheme (dashed black). The NK simulations are initialised using a ψ_p guess biased to the left (left panels) or biased to the right (right panels). The Picard simulations are initialised using a ψ_p guess from an intermediate iteration of the NK simulation (black crosses). The simulations either have I_p normalisation switched on (top panels) or off (lower panels).

multiple solutions, it also further highlights that care is needed when comparing solutions between different codes, as, in the presence of multiple solutions, the adopted solver may affect which solution is identified. In particular, while the outboard limited equilibrium may not appear to be a ‘physically valid’ solution, the GS solver by itself cannot determine this. One way to select the appropriate equilibrium could be to ‘bias’ the initial plasma flux guess more toward the inboard side (in this particular case), however, this is not systematic and is not guaranteed to return the inboard diverted solution. A more complex but robust approach would be to incorporate some form of deflation into GS solvers to identify any possible equilibria before selecting the ‘correct’ one based on experimental measurement data (e.g. from magnetic probes or fluxloops). We note this process may, however, become more challenging in situations when the two solutions obtained become less distinguishable (for instance, at larger I_p as in figure 4) and in predictive modelling when experimental data is unavailable.

Along similar lines, the presence of distinct solution branches may also affect time-dependent equilibrium calculations in which the poloidal flux is evolved alongside currents in the conducting metal structures. These evolutive solvers play an important role in predictive ‘feed-forward’ and ‘feedback’ plasma scenario and control modelling. Depending on the algorithm being used (for example, the one implemented in FreeGSNKE), each time step of the evolution may require solving tens of static GS problems. If any of these internal GS solver were to select the ‘incorrect’

branch—or ‘flicker’ back and forth between branches—it could undermine the stability/validity of the overall simulation.

Having identified multiple solution branches in MAST-U with FreeGSNKE, we think it would be of interest in the future to investigate whether or not these (and perhaps other) branches exist when using alternative equilibrium codes, tokamak geometries, and plasma conditions. Further study will aid in the development of more robust algorithms for plasma control and scenario forecasting and help avoid convergence issues in future equilibrium calculations.

Data availability statement

The code scripts and data used in this paper can be found on the [UKAEA OpenData Register](#).

A simple Python implementation of deflated continuation, deployed on a few polynomial and ordinary differential equation examples [can be found on GitHub](#).

Conflict of interest

The authors have no conflicts of interest to declare.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments on improving this manuscript.

This work was funded by the EPSRC Energy Programme (Grant number EP/W006839/1) by EPSRC Grants EP/R029423/1 and EP/W026163/1, and by the Donatio Universitatis Carolinae Chair ‘Mathematical modelling of multicomponent systems’.

For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any author accepted manuscript version arising from this submission. To obtain further information, please contact publicationsmanager@ukaea.uk.

ORCID iDs

K. Pentland 0000-0001-9530-1890
 N.C. Amorisco 0000-0002-3205-5335
 P.E. Farrell 0000-0002-1241-7060
 C.J. Ham 0000-0001-9190-8310

References

- [1] Agnello A., Amorisco N.C., Keats A., Holt G.K., Buchanan J., Pamela S.J.P., Vincent C. and McArdle G. 2024 Emulation techniques for scenario and classical control design of tokamak plasmas *Phys. Plasmas* **31** 043901
- [2] Ambrosetti A. and Mancini G. 1980 A free boundary problem and a related semilinear equation *Nonlinear Anal. Theory Methods Appl.* **4** 909–15
- [3] Amorisco N.C., Agnello A., Holt G., Mars M., Buchanan J. and Pamela S.J.P. 2024 FreeGSNKE: a python-based

- dynamic free-boundary toroidal plasma equilibrium solver *Phys. Plasmas* **31** 042517
- [4] Baratta I.A., Dean J.P., Dokken J.S., Habera M., Hale J.S., Richardson C.N., Rognes M.E., Scroggs M.W., Sime N. and Wells G.N. 2023 DOLFINx: the next generation FEniCS problem solving environment *Zenodo* (<https://doi.org/10.5281/zenodo.10447666>)
- [5] Bartolucci D., Hu Y., Jevnikar A. and Yang W. 2021 Generic properties of free boundary problems in plasma physics *Nonlinearity* **35** 411
- [6] Berestycki H. and Brezis H. 1980 On a free boundary problem arising in plasma physics *Nonlinear Anal. Theory Methods Appl.* **4** 415–36
- [7] Berkery J.W., Sabbagh S.A., Kogan L., Ryan D., Bialek J.M., Jiang Y., Battaglia D.J., Gibson S. and Ham C. 2021 Kinetic equilibrium reconstructions of plasmas in the MAST database and preparation for reconstruction of the first plasmas in MAST upgrade *Plasma Phys. Control. Fusion* **63** 055014
- [8] Brown K.M. and Gearhart W.B. 1971 Deflation techniques for the calculation of further solutions of a nonlinear system *Numer. Math.* **16** 334–42
- [9] Chapman I.T., Cowley S.C. and Wilson H.R. 2024 The spherical tokamak for energy production: theme issue introduction *Phil. Trans. R. Soc. A* **382** 20230416
- [10] Cisternas J. and Concha A. 2024 Searching nontrivial magnetic equilibria using the deflated Newton method *Chaos Solitons Fractals* **179** 114468
- [11] Conway N.J., De Bock M.F.M., Michael C.A., Walsh M.J., Carolan P.G., Hawkes N.C., Rachlew E., McCone J.F.G., Shibaev S. and Wearing G. 2010 The MAST motional Stark effect diagnostic *Rev. Sci. Instrum.* **81** 10D738
- [12] Cunningham G. 2013 High performance plasma vertical position control system for upgraded MAST *Fusion Eng. Des.* **88** 3238–47
- [13] Farrell P.E., Birkisson A. and Funke S.W. 2015 Deflation techniques for finding distinct solutions of nonlinear partial differential equations *SIAM J. Sci. Comput.* **37** A2026–45
- [14] Farrell P.E., Beentjes C.H.L. and Birkisson A. 2016 The computation of disconnected bifurcation diagrams (arXiv:1603.00809)
- [15] Farrell P.E., Pollard J., Kirby R.C., Blechta J., Croci M., Sime N.J.C. and Boullé N. 2016 defcon Bitbucket repository (available at: <https://bitbucket.org/pefarrell/defcon/src/master/>)
- [16] Grad H. and Rubin H. 1958 Hydromagnetic equilibria and force-free fields *Proc. 2nd United Nations Int. Conf. on the Peaceful Uses of Atomic Energy: Theoretical and Experimental Aspects of Controlled Nuclear Fusion (Geneva, Switzerland, 1 September 1958)* vol 31 p 400 (available at: <https://inis.iaea.org/records/5wg99-qq804>)
- [17] Ham C.J. and Farrell P.E. 2024 On multiple solutions of the Grad–Shafranov equation *Nucl. Fusion* **64** 034001
- [18] Ham D.A. et al 2023 Firedrake User Manual (<https://doi.org/10.25561/104839>)
- [19] Ilgisonis V.I. and Pozdnyakov Y.I. 2004 Bifurcation of the equilibrium of a current-carrying plasma column *Plasma Phys. Rep.* **30** 988–94
- [20] Jardin S. 2010 *Computational Methods in Plasma Physics* (CRC Press) (<https://doi.org/10.1201/EK1439810958>)
- [21] Jeon Y.M. 2015 Development of a free-boundary tokamak equilibrium solver for advanced study of tokamak equilibria *J. Korean Phys. Soc.* **67** 843–53
- [22] Jeyakumar S., Pfefferlé D., Hole M.J. and Qu Z.S. 2021 Analysis of the isotropic and anisotropic Grad–Shafranov equation *J. Plasma Phys.* **87** 905870506
- [23] Kogan L. et al (MAST Upgrade Team) 2022 First MAST-U equilibrium reconstructions using the EFIT++ code *48th Eur. Physical Society Conf. on Plasma Physics (Maastricht, Netherlands, 27 June 2022)* (available at: <https://research.tue.nl/en/publications/first-mast-u-equilibrium-reconstructions-using-the-efit-code>)
- [24] Lao L.L., John H.S., Stambaugh R.D., Kellman A.G. and Pfeiffer W. 1985 Reconstruction of current profile parameters and plasma shapes in tokamaks *Nucl. Fusion* **25** 1611
- [25] Pentland K. et al 2024 Validation of the static forward Grad–Shafranov equilibrium solvers in FreeGSNKE and Fiesta using EFIT++ reconstructions from MAST-U *Phys. Scr.* **100** 025608
- [26] Romanelli M. et al (EFDA-JET Contributors) 2014 JINTRAC: a system of codes for integrated simulation of tokamak scenarios *Plasma Fus. Res.* **9** 3403023
- [27] Schaeffer D.G. 1977 Non-uniqueness in the equilibrium shape of a confined plasma *Commun. PDE* **2** 587–600
- [28] Schnack D.D. 2009 *Lectures in Magnetohydrodynamics: With an appendix on Extended MHD (Lecture Notes in Physics vol 780)* (Springer) <https://doi.org/10.1007/978-3-642-00688-3>
- [29] Shafranov V.D. 1958 On magnetohydrodynamical equilibrium configurations *Sov. J. Exp. Theor. Phys.* **6** 545 (available at: <https://www.osti.gov/biblio/4305963>)
- [30] Solano E.R. 2004 Criticality of the Grad–Shafranov equation: transport barriers and fragile equilibria *Plasma Phys. Control. Fusion* **46** L7
- [31] Takeda T. and Tokuda S. 1991 Computation of MHD equilibrium of tokamak plasma *J. Comput. Phys.* **93** 1–107
- [32] Temam R. 1975 A non-linear eigenvalue problem: the shape at equilibrium of a confined plasma *Arch. Ration. Mech. Anal.* **60** 51–73
- [33] Temam R. 1977 Remarks on a free boundary value problem arising in plasma physics *Commun. PDE* **2** 563–85
- [34] Turnbull A. 1984 Bifurcation of a free boundary equilibrium *J. Plasma Phys.* **31** 1–6
- [35] Xia J., MacLachlan S., Atherton T.J. and Farrell P.E. 2021 Structural landscapes in geometrically frustrated smectics *Phys. Rev. Lett.* **126** 177801