Free-plasma-boundary solver for axisymmetric ideal MHD equilibria with flow

PAPER

Free-plasma-boundary solver for axisymmetric ideal MHD equilibria with flow

To cite this article: G. F-Torija Daza *et al* 2022 *Nucl. Fusion* **62** 126044

IOP Publishing | International Atomic Energy Agency Nuclear Fusion

Nucl. Fusion **62** (2022) 126044 (11pp) https://doi.org/10.1088/1741-4326/ac95ae

Free-plasma-boundary solver for axisymmetric ideal MHD equilibria with flow

G. F-Torija Daza[1,*], J.M. Reynolds-Barredo[1], R. Sanchez[1], A. Loarte[2] and G. Huijsmans[3,4]

[1] Departamento de Física, Universidad Carlos III de Madrid, Leganés, Spain
[2] ITER Organization, Route de Vinon-sur-Verdon, 13067 St Paul Lez Durance, France
[3] Department of Applied Physics, Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, The Netherlands
[4] Institute for Magnetic Fusion Research, CEA, 13108 St Paul Lez Durance, France

E-mail: gonferna@fis.uc3m.es

Received 13 June 2022, revised 23 September 2022
Accepted for publication 28 September 2022
Published 27 October 2022

Abstract

An efficient iterative, free-plasma-boundary solver for the Grad–Shafranov–Bernoulli system of equations, that describes the ideal MHD equilibrium of a toroidally axisymmetric plasma with flow, is presented. The code implements a numerical scheme recently developed in the context of free-plasma-boundary solvers for ideal static MHD equilibria with magnetic islands and stochastic regions for stellarators. The shape of the plasma edge is permitted to change as needed until the total net force eventually vanishes en route to the equilibrium. Complex coil configurations can be treated in the toroidally axisymmetric approximation. The code opens the possibility of quantifying the changes that plasma flows may induce on important features of a tokamak equilibrium such as the shape of the plasma edge, the plasma confining volume, the position of the magnetic axis or the position of the X-point, among others. Some examples, selected for illustrative purposes, are shown for the ITER baseline magnetic configuration.

Keywords: ideal MHD, equilibrium with flow, Grad–Shafranov equation, free-boundary, iterative solvers

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

1. Introduction

The calculation of MHD equilibria is routinely required in many contexts in magnetically confined fusion plasmas, starting at the initial phases of the design of a confining magnetic configuration and continuing during the interpretation of multiple diagnostic systems during their operation. Many key evaluations for plasma scenarios depend on having a reliable knowledge of the confining magnetic field and the geometry of the confined plasma at equilibrium is required, for instance, when assessing the consequences of adding or removing elements to/from an existing tokamak (such as control coils), or when making changes in its configuration (varying currents in the external coil set, modifying the location and strength of particle and heat sources, etc).

Another situation when a precise knowledge of the plasma equilibrium may be important is when significant poloidal and/or toroidal flows are established within the tokamak plasma due to either external (biasing, neutral beam heating, …) or internal causes (development of transport barriers…). However, ideal 2D MHD equilibria are traditionally calculated in tokamaks under the assumption of axisymmetry and negligible plasma flows, since the problem is then reduced to solving one elliptic equation for the magnetic poloidal flux function, $\psi(R, Z)$, with R and Z being the usual cylindrical coordinates. This equation, the well-known Grad–Shafranov equation [1], depends on two external functions of ψ that essentially prescribe first, the plasma pressure, $p(\psi)$, and secondly, $F(\psi) = RB_\phi(\psi)$, where B_ϕ is the toroidal magnetic field and that can be shown to be related to the net poloidal current [2]. Toroidal axisymmetry usually remains a good approximation in most experimental situations. The absence

* Author to whom any correspondence should be addressed.

of macroscopic plasma flows, however, is often not satisfied. The consideration of plasma flows in the 2D MHD equilibrium calculation, however, significantly complicates the problem. The number of equations to be solved is now two, since the (modified) Grad–Shafranov equation must be supplemented with another one, known as the Bernoulli equation [3–6]. The number of free functions to prescribe is also increased from two to five that, regretfully, are not easy to assign since experimentally available quantities, such as the plasma pressure or the density, are no longer flux functions.

Several numerical codes exist, such as FINESSE [7] or FLOW [8], that manage to deal with all this complexity and routinely carry out calculations of ideal 2D MHD tokamak equilibria with poloidal and toroidal plasma flows. A common element to all these codes, however, is that they often assume a fixed plasma boundary at which the value of magnetic poloidal flux ψ is known and assigned from input. This approach is usually referred to as a 'fixed-boundary' equilibrium solution. Calculations are then restricted to the plasma region. In many cases, this is a reasonable approach since the approximate shape of the plasma edge might be experimentally available, and the value of ψ at the edge could be inferred from the coils and measured plasma currents. There are situations, however, in which the shape of the plasma may not be known in advance. Or, more interestingly, one could perhaps like to quantify how the shape of the plasma edge changes under controlled variations of discharge parameters in the presence of plasma flows. These could be the case, for instance, while investigating the consequences of adding new elements to, or changing the nominal current density values of an existing coil configuration. A different type of code is then needed that could solve the Grad–Shafranov–Bernoulli (GSB) equations using a free-plasma-boundary approach in which the shape of the plasma edge is not prescribed *a priori*, being instead free to adapt to what the force-balance equations dictate.

There are various free-plasma-boundary solvers available for ideal static MHD equilibria, both for tokamaks and stellarators. In this paper, we rely on this expertise to build a code that can produce free-plasma-boundary solutions of the GSB system of equations for an arbitrarily complex set of external coils. This is done by the implementation, taking into account the particular features of the GSB equations, of a numerical scheme recently used to enable the SIESTA code [9], a 3D ideal MHD equilibrium code without plasma flows but with capabilities to support magnetic islands and stochastic regions, to perform free-plasma-boundary simulations. The process entails the coupling of a GSB solver with two well established numerical tools. First, the VMEC code [10], an ideal MHD solver for general 3D static equilibria that can be easily adapted to 2D problems as well. Secondly, the MAKEGRID code, a tool from the STELLOPT stellarator optimization suite [11] that calculates the vacuum magnetic field created by an arbitrarily complex external coil set at any arbitrary position. Both codes where developed in the context of free-boundary, static stellarator MHD equilibria in the 1980s and 90s.

The particular issues encountered and addressed to implement this free-plasma-boundary scheme in the GSB context are detailed in section 4, after quickly reviewing the Grad–Shafravnov–Bernoulli system of equations in section 2 and describing in section 3 the more relevant features of the GBS numerical solver used. Examples of the analysis capabilities enabled by this new free-plasma-boundary code will be illustrated in section 5 on the ITER baseline magnetic configuration. Finally, some conclusions will be drawn in section 6.

2. Review of axissymmetric ideal MHD equilibrium with flows

2.1. Ideal MHD equations

The equations of ideal MHD in the presence of plasma flows are well known [2]:

$$\text{Continuity} \quad \rightarrow \quad \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) \tag{1}$$

$$\text{Momentum} \quad \rightarrow \quad \rho \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{J} \times \mathbf{B} - \nabla p \tag{2}$$

$$\text{Entropy equation} \quad \rightarrow \quad \frac{\partial S}{\partial t} + \mathbf{v} \cdot \nabla S = 0 \tag{3}$$

$$\text{Ohm's law} \quad \rightarrow \quad \mathbf{E} + \mathbf{v} \times \mathbf{B} = 0 \tag{4}$$

$$\text{Ampere:} \quad \rightarrow \quad \nabla \times \mathbf{B} = \mu_0 \mathbf{J} \tag{5}$$

$$\text{B} - \text{divergence:} \quad \rightarrow \quad \nabla \cdot \mathbf{B} = 0 \tag{6}$$

using as variables the electric field $\mathbf{E}$, the magnetic field $\mathbf{B}$, the current density $\mathbf{J}$, the (assumed isotropic) plasma pressure p, the plasma density ρ and the plasma fluid velocity $\mathbf{v}$. The function $S \equiv p/\rho^{\gamma}$ is the entropy, with $\gamma = c_{\mathrm{p}}/c_{\mathrm{v}}$ being the ratio of specific heats. Equation (3) provides the closure for the system of equations.

2.2. Grad–Shafranov equation

In equilibrium, and in the absence of plasma flows, the ideal MHD equations reduce to:

$$\mathbf{J} \times \mathbf{B} = \nabla p \tag{7}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} \tag{8}$$

$$\nabla \cdot \mathbf{B} = 0. \tag{9}$$

These equations can be easily recast into a more manageable form by introducing cylindrical coordinates (R, ϕ, Z), assuming toroidal axissymetry and expressing the magnetic field as:

$$\mathbf{B} = \nabla \psi \times \mathbf{u}_{\phi} + B_{\phi} \mathbf{u}_{\phi}, \tag{10}$$

that trivially satisfies the divergence-free condition for the magnetic field. The function $\psi(R, Z)$ represents the poloidal magnetic flux (divided by 2π). The isosurfaces $\psi(R, Z) =$ constant define the magnetic surfaces. Introducing this representation in the Ampere equation, to get the current density vector $\mathbf{J}$, and then in the momentum equation, leads to the famous elliptic Grad–Shafranov equation [1]:

$$R^2\nabla\cdot\left(\frac{\nabla\psi}{R^2}\right)=-\mu_0R^2p'(\psi)-\frac{1}{2}(F^2(\psi))', \qquad (11)$$

where two surface functions appear. First, $p=p(\psi)$ that gives the plasma pressure as a function of ψ, and $F(\psi)=RB_\phi$ that is related with the total poloidal current flowing through a disk-shaped surface lying in the $Z=0$ plane that extends from $R=0$ to the contour ψ due to both coils and plasma [2] (indeed, $I_{\rm p}(\psi)=-2\pi F(\psi)$). The prime ($'$) is used to denote derivative with respect to ψ, as usual. Various codes exist that can solve this equation rather efficiently (for instance, the CHEASE [12]or EFIT [13] codes). Two additional ingredients are needed to have a unique solution of equation (11): (i) expressions for the profiles of p and F as a function of ψ, that can often be inferred from experimental measurements, and (ii) the value of ψ on some prescribed curve, usually taken at the plasma edge. This relation, expressed as $\psi(R,Z)=\psi_{\rm edge}$, is known as a fixed-boundary condition.

2.3. Grad–Shafranov–Bernoulli system of equations

In the case in which plasma flow is retained, the manipulation of the equilibrium equations become more involved, as shown by various authors [3–5]. The more popular version of the derivation is probably the one published by Hameiri [6]. Thanks to axisymmetry the plasma flow can be expressed in similar form to the magnetic field,

$$\mathbf{v}=\frac{1}{\rho R}\nabla\chi\times\mathbf{u}_\phi+v_\phi\mathbf{u}_\phi, \qquad (12)$$

where $\chi(R,Z)$ is the velocity streamfunction. Inserting this representation, together with equation (10) for the magnetic field, in the momentum balance equation leads to a pair of equations that must be satisfied at equilibrium. The first one, the Bernoulli equation, is obtained by projecting the momentum equation along the magnetic field, leading to:

$$\frac{(\chi'(\psi))^2}{2\rho^2}\left(\frac{|\nabla\psi|^2}{R^2}+B_\phi^2\right)-\frac{1}{2}R^2\Phi'^2(\psi)+\frac{\gamma}{\gamma-1}\rho^{\gamma-1}S(\psi)=H(\psi). \qquad (13)$$

It is useful to view this equation as a constraint to be satisfied by the combination of quantities on the left-hand side, that must become a flux function at equilibrium. This function, $H(\psi)$, is known as the Bernoulli function. The Bernoulli equation contains several other functions that also turn out to be flux functions at equilibrium. In addition to the velocity streamfunction, $\chi(\psi)$, one has the entropy function, $S(\psi)=p/\rho^\gamma$, and the electric potential $\Phi(\psi)$, whose derivative,

$$\Phi'(\psi)=\frac{1}{R}\big(v_\phi-\chi'(\psi)B_\phi/\rho\big), \qquad (14)$$

provides the radial electric field $\mathbf{E}=-\Phi'(\psi)\nabla\psi$ needed to satisfy Ohm's law.

The second equation that must be satisfied at equilibrium is a modified version of the Grad–Shafranov equation. It is obtained, as in the case without plasma flow, from the projection of the momentum equation along $\nabla\psi$. The result is:

$$\frac{1}{\mu_0}\nabla\cdot\left(\frac{(1-(M_{\rm p}^{\rm Alf})^2)}{R^2}\nabla\psi\right)=-\frac{B_\phi}{\mu_0R}K'(\psi)-\left(\frac{\chi'(\psi)}{\rho R^2}+v_\phi B_\phi\right)\chi''(\psi)-\rho H'(\psi)-R\rho v_\phi\Phi''(\psi)+\frac{\rho^\gamma}{\gamma-1}S'(\psi). \qquad (15)$$

The equation is clearly more complicated than the static one, equation (11), although the latter is recovered if one sets $\chi(\psi)=0$. Equation (15) contains one additional flux function,

$$K(\psi)\equiv RB_\phi-\mu_0R\chi'(\psi)v_\phi, \qquad (16)$$

that reduces to $F(\psi)=RB_\phi$ in the absence of plasma flow. In addition, an extra factor $(1-(M_{\rm p}^{\rm Alf})^2)$, with $M_{\rm p}^{\rm Alf}$ being the poloidal Alfven Mach number,

$$M_{\rm p}^{\rm Alf}(R,Z)=\frac{|\mathbf{v}_{\rm p}|}{|\mathbf{B}_{\rm p}|/\sqrt{\mu_0\rho}}=\left(\frac{\mu_0}{\rho}\right)^{1/2}\chi'(\psi), \qquad (17)$$

has appeared inside the differential operator in the left-hand side. This factor may drastically change the nature of the Grad–Shafranov equation for poloidal plasma flows if the condition $M_{\rm p}^{\rm Alf}<1$ is violated anywhere inside of the domain of interest [14–16].

The solution of the GSB pair of equations in the $R-Z$ plane requires the prescription of five functions of ψ. Namely, the electrostatic potential $\phi(\psi)$, the velocity streamfunction $\chi(\psi)$, the Bernoulli function, $H(\psi)$, the entropy function $S(\psi)$ and $K(\psi)$, defined by equation (16). There are several codes available that can solve this pair of equations such as, for instance, FINESSE [7] or FLOW [8]. Most of them, however, work in the same 'fixed-boundary' mode we previously described for the Grad–Shafranov equation. That is, by forcing the shape of the plasma boundary to coincide with an isosurface of the poloidal magnetic flux function, $\psi(R,Z)=\psi_{\rm edge}$, thus restricting the solution to the plasma volume. In the next section we will describe the steps taken to be able to consider a free-plasma-boundary instead.

3. Fixed-boundary solver for Grad–Shafranov–Bernoulli system of equations

The free-plasma-boundary scheme to be described in the next section requires a fixed-boundary solver of the GSB equations. This solver, however, should be able to use a fixed boundary different from the shape of the plasma boundary. In this section, we describe some features particular to the iterative solver we have built for this task. Our code uses toroidal coordinates (r,θ,ζ), related to the more usual cylindrical ones

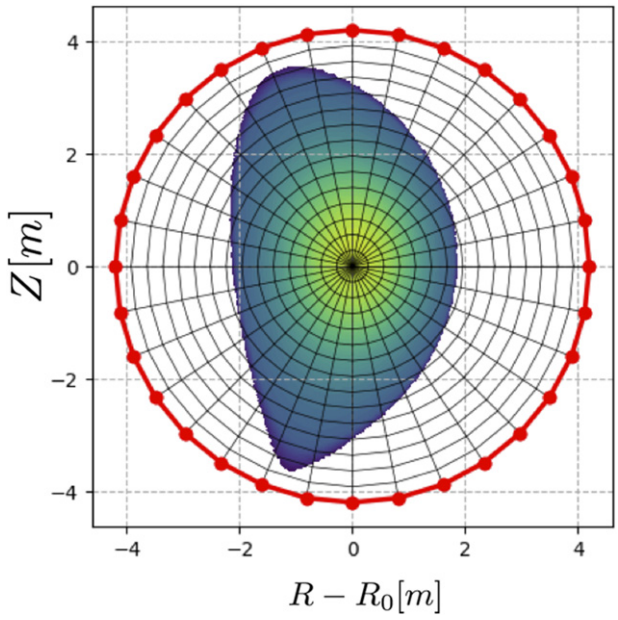

Figure 1. 2D computational domain used to solve the GSB system. Boundary conditions are enforced on an arbitrary interior circle (illustrated in red) with radius set by the user. Although this circle usually encloses the full plasma, it may also be smaller if desired. Isocurves for r and θ are shown in black.

(R, Z, ϕ) via:

$$\begin{aligned} R &= R_0 + r\cos\theta \\ \phi &= -\zeta \\ Z &= r\sin\theta \end{aligned} \quad . \tag{18}$$

R_0 is the major radius of the geometrical torus, that should be chosen close to, but need not necessarily coincide with, the magnetic axis as discussed below. The 2D computational domain (see figure 1) can be chosen as large as desired. The boundary condition for ψ is prescribed on a circular surface selected by the user (in red in the figure) that is centered at $R = R_0$ and with arbitrary minor radius $r = r_B$. The value of the magnetic flux function on the circle, $\psi(r_b, \theta) = \psi_b(\theta)$, must be provided by the user to match the problem of interest. Note that, in general, this value is not a single number but a function of the poloidal angle. Note also that the domain need not restricted to $r < r_B$ as long as the values of ψ on that circle are kept fixed. Our choice of coordinates allows us, in addition to an easier interaction with the VMEC code, to treat all angular dependencies in θ using a spectral method, while the radial derivatives are dealt with using finite differences of second-order accuracy. This leads to very good convergence properties for the residual net force, that decreases quadratically with the number of radial points, M_r, and exponentially with the number of poloidal modes, N_θ (see figure 2).

The solver searches iteratively for the equilibrium solution. The convergence of the iterative scheme is facilitated if R_0 can be taken as close as possible to the magnetic axis, since the Fourier representation then becomes more compact, with fewer poloidal modes, that leads to smaller matrices and an

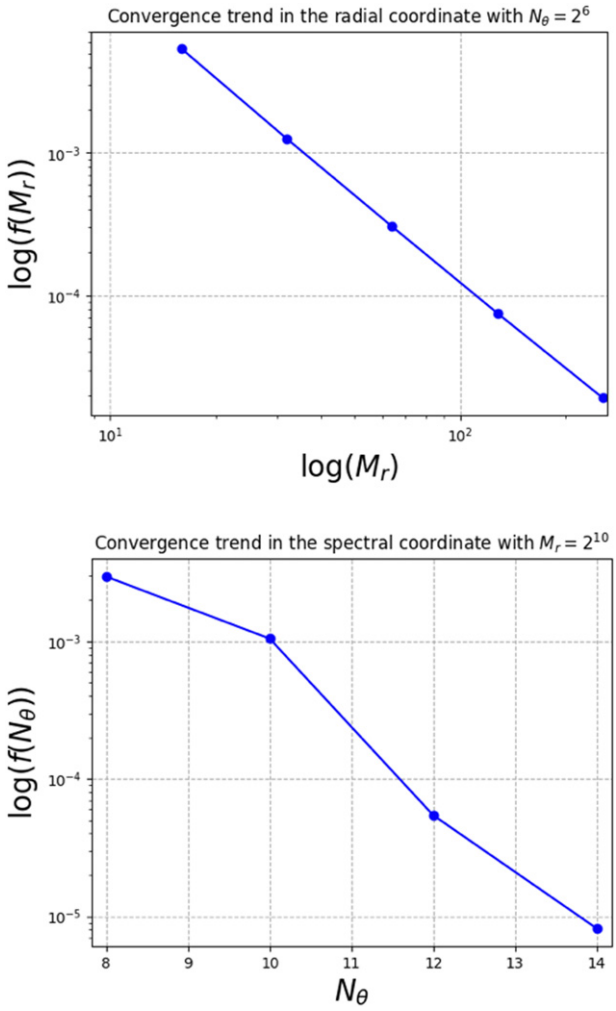

Figure 2. Convergence of the normalized residual force (defined as the average value of the ratio to absolute values $|\nabla p - \mathbf{J}\times\mathbf{B} - \rho(\mathbf{v}\cdot\nabla)\mathbf{v}|/|\mathbf{J}\times\mathbf{B}|$ over the computational domain) as a function of the number of radial points, M_r, and poloidal modes, N_θ for the Maschke–Perrin analytical equilibrium with toroidal rotation [17] for $M_{tor} = 0.8$.

overall faster convergence of the iterative scheme. To start the procedure, profiles for the five free flux functions ($K(\psi)$, $\Phi(\psi)$, $\chi(\psi)$, $H(\psi)$ and $S(\psi)$) must be given as well as an initial guess, $\psi_0(r, \theta)$, for the poloidal magnetic flux function throughout the computational box that satisfies the prescribed boundary condition, $\psi_0(r_b, \theta) = \psi_b(\theta)$. The iteration starts by first solving Bernouill's equation (equation (13)) to obtain the density using a combination of Newton and bisection methods. This is possible, as previously noted by many authors [8, 16], because the only undetermined functions in equation (13) are the density ρ and B_ϕ, that can in turn be related by combining equations (14) and (16) to yield,

$$B_\phi = \frac{K(\psi) + \mu_0 R^2 \chi'(\psi)\phi'(\psi)}{R(1 - \mu_0 \chi'^2/\rho)}. \tag{19}$$

The obtained density, together with the guess for the poloidal magnetic flux function $\psi_0(r, \theta)$ are then inserted into the right-hand side of the modified Grad–Shafranov equation, that is subsequently solved to provide an improved poloidal magnetic

flux, $\psi_1(r,\theta)$, that still satisfies the prescribed boundary condition at $r = r_\mathrm{b}$. The refined ψ_1 is then used to recalculate the density by means of the Bernoulli equation, and then inserted back into the Grad–Shafranov equation to further refine the poloidal magnetic flux function. This procedure is iterated for as many times as needed to reach the required tolerance in ψ, defined at the k-iteration as the average over the whole domain of the ratio,

$$\mathrm{tol}(k) = 2 \times \left\langle \frac{|\psi_k(r,\theta) - \psi_{k-1}(r,\theta)|}{|\psi_k(r,\theta) + \psi_{k-1}(r,\theta)|} \right\rangle_{r,\theta}. \tag{20}$$

The code handles $K(\psi)$, $\Phi(\psi)$, $\chi(\psi)$, $H(\psi)$ and $S(\psi)$ using the same prescription used by the FLOW code [8]. That is, defining them via five auxiliary functions, $D(\psi)$, $P(\psi)$, $B_0(\psi)$, $M_\mathrm{t}^{c_\mathrm{s}}(\psi)$ and $M_\mathrm{p}^{c_\mathrm{s}}(\psi)$:

$$K(\psi) \equiv R_0 B_0(\psi) \tag{21}$$

$$\chi'(\psi) \equiv \sqrt{\gamma P(\psi) D(\psi)} \left(\frac{M_\mathrm{p}^{c_\mathrm{s}}(\psi)}{B_0(\psi)} \right) \tag{22}$$

$$\phi'(\psi) \equiv \sqrt{\gamma \frac{P(\psi)}{D(\psi)}} \left(\frac{(M_\mathrm{t}^{c_\mathrm{s}}(\psi) - M_\mathrm{p}^{c_\mathrm{s}}(\psi))}{R_0} \right) \tag{23}$$

$$H(\psi) \equiv \left(\gamma \frac{P(\psi)}{D(\psi)} \right) \left[\frac{1}{(\gamma - 1)} + M_\mathrm{t}^{c_\mathrm{s}}(\psi) \cdot \left(M_\mathrm{p}^{c_\mathrm{s}}(\psi) - \frac{M_\mathrm{t}^{c_\mathrm{s}}(\psi)}{2} \right) \right] \tag{24}$$

$$S(\psi) \equiv \frac{P(\psi)}{D^\gamma(\psi)}. \tag{25}$$

A nice aspect about these choices is that $D(\psi)$, $P(\psi)$ and $B_0(\psi)$ reduce to the usual density, pressure and toroidal field profiles in the absence of plasma flow, so they can be interpreted in this fashion even if no longer flux quantities at finite plasma flow. The remaining two functions, $M_\mathrm{p}^{c_\mathrm{s}}(\psi)$ and $M_\mathrm{t}^{c_\mathrm{s}}(\psi)$ are (approximate) sonic Mach numbers in the poloidal and toroidal directions.

A typical output of the code is illustrated in figure 3 that shows the isosurfaces for the poloidal magnetic flux as obtained for the baseline ITER $Q = 10$ plasma scenario in a case that assumes only toroidal plasma rotation. The equilibrium has been calculated using $S(\psi) = 1$, $M_\mathrm{p}^{c_\mathrm{s}}(\psi) = 0$ and the profiles for pressure and toroidal magnetic field shown in figure 4. Note that these profile shapes are adopted for testing purposes and are not necessarily consistent with those from integrated modeling. The toroidal sonic Mach number profile used is the one labeled as '(qf)' in figure 5. This normalized profile has been scaled, in this run, with a global scaling factor $M_\mathrm{tor} = 0.6$. As can be seen, the toroidal plasma rotation is maximum at the magnetic axis and smoothly decreases quadratically towards zero at the plasma edge. The initial guess for the magnetic poloidal flux function ψ, as well as its value on the circle where the fixed boundary condition is imposed, are built from a solution previously calculated with the VMEC code for the same equilibrium but *in the absence of plasma flow*. The initial isocontours for ψ are shown in dashed blue (labeled as ψ_0) in the top frame of figure 3 and the final

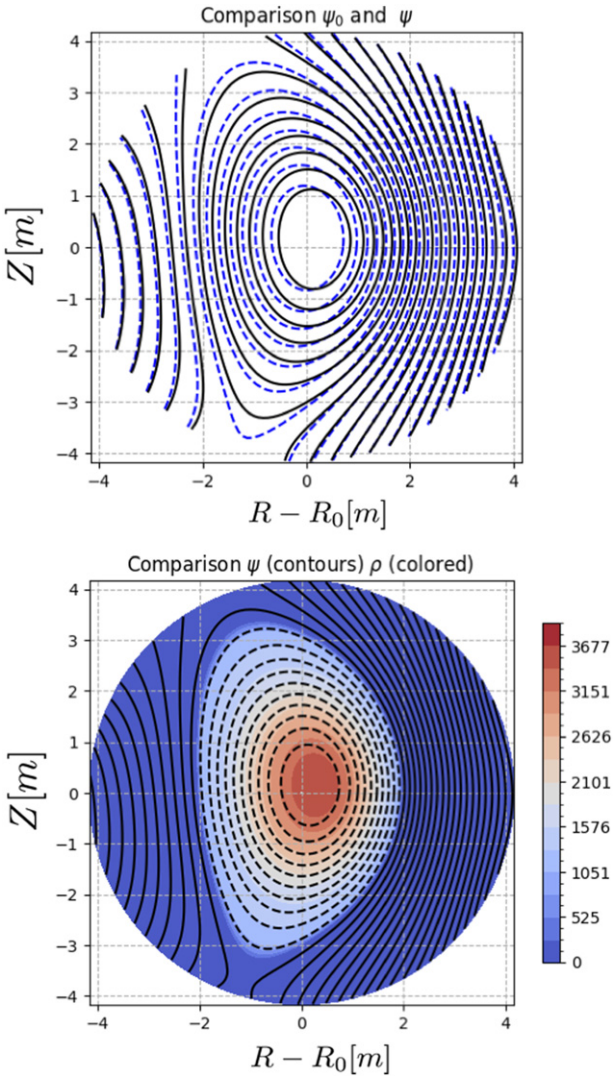

Figure 3. Top: initial and final isocontours of ψ for the ITER equilibrium examined using the '(qf)' profile for the toroidal sonic Mach number with scaling factor $M_\mathrm{tor} = 0.6$. Bottom: color map of the plasma density with overlaid ψ-isocontours that illustrate the outwards displacement of the density due to the centrifugal force.

contours obtained by our code are shown in continuous black. In the bottom frame, a color map of the final plasma density ρ is shown together with the isocontours of ψ, illustrating the expected outward radial displacement of the density resulting from the centrifugal force.

4. Free-plasma-boundary scheme

Free-plasma-boundary capabilities can be added to the previous solver by finding a suitable way to treat the boundary condition that is imposed on the value of poloidal magnetic flux function at the control circle, $\psi_\mathrm{b}(r_\mathrm{b})$. Clearly, $\psi_\mathrm{b}(r_\mathrm{b})$ is determined by two different contributions:

$$\psi_\mathrm{b}(r_\mathrm{b}) = \psi_b^V(r_\mathrm{b}) + \psi_b^P(r_\mathrm{b}). \tag{26}$$

$\psi_\mathrm{b}^V(r_\mathrm{b})$ is the contribution from the currents flowing in the external coils that define the magnetic configuration. $\psi_\mathrm{b}^P(r_\mathrm{b})$,

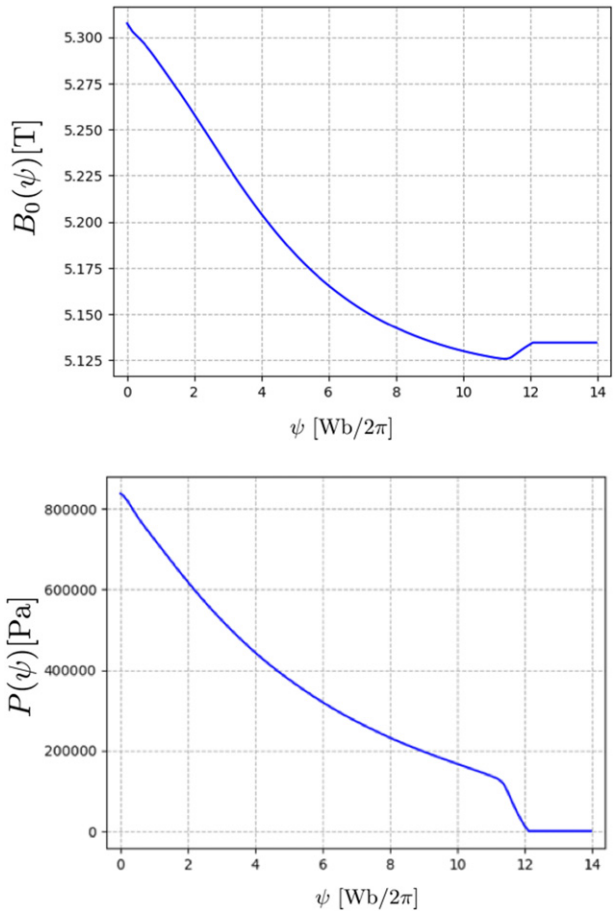

Figure 4. Toroidal magnetic field ($B_0(\psi)$) profile (top) and pressure ($P(\psi)$) profile (bottom) used for the ITER equilibrium calculation.

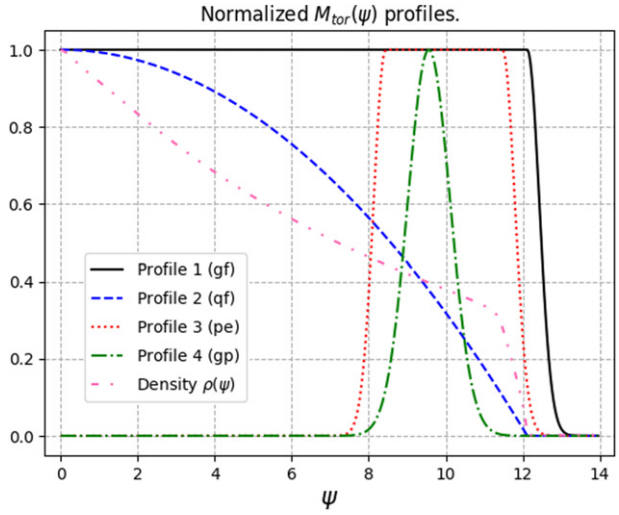

Figure 5. Normalized toroidal sonic Mach number profiles, $M_{\rm t}^{\rm cs}(\psi)$, used throughout the paper. These profiles are scaled by means of a global scaling factor, $M_{\rm tor}$, as discussed in text. The plasma density profile, $D(\psi)$ is also shown for reference.

on the other hand, is the contribution from the plasma current. The first contribution does not change once the coil currents are fixed, so it is only evaluated once. The second one, however, will change as the plasma adapts its shape and profiles on its way to the final equilibrium.

Before describing how each of these two contributions are evaluated, it is worth describing the iterative flow of the free-plasma-boundary solver. Starting from some value of $\psi_{\rm b}^{(j=1)}(r_{\rm b})$, the GSB solver described in the previous section will iteratively find the poloidal magnetic flux function, $\psi^{(j=1)}(r,\theta)$, that is consistent with that boundary condition (we will use the index j to label the (outer) iterations of the boundary condition, that should not be confused with k, that labeled the (inner) iterations within the fixed-boundary solver). However, it should be expected that the final plasma current will be different from the one used previously to determine the plasma current contribution to the boundary condition $\psi_{\rm b}^{(j=1)}(r_{\rm b})$. Therefore, the code starts a new outer iteration in which it recalculates the plasma current contribution, updates the boundary condition to $\psi_{\rm b}^{(j=2)}(r_{\rm b})$ and calls the GSB solver to find an updated poloidal magnetic flux function, $\psi^{(j=2)}(r,\theta)$, consistent with the updated boundary condition. The whole process is repeated for as many outer iterations as needed to reach the required tolerance.

Let us now describe how the boundary condition, $\psi_{\rm b}(r_{\rm b})$, is calculated. The poloidal magnetic flux function is essentially (except by a 2π factor) the flux of the magnetic field through a toroidal wedge that starts from the $Z=0$ plane at $R=R_0$ and extends radially at fixed angle θ until it intersects the control circle at $r=r_{\rm b}$. Were the control circle a magnetic surface, the result would be independent of θ, but that is not usually the case. Therefore, we need to compute the magnetic flux for wedges at multiple values of θ.

The contribution from the external coils, $\psi_{\rm b}^{V}(r_{\rm b})$, is calculated using the MAKEGRID code, that is part of the STELLOPT stellarator optimization suite [11]. MAKEGRID computes the magnetic field generated by an arbitrary set of external coils, defined as a collection of wire segments carrying a normalized unit current. The code uses the Biot–Savart law to evaluate the contribution of each coil segment to an arbitrary selection of points on a $R-Z$ grid. Each contribution is then multiplied by the actual current carried by each coil to yield the final value of the magnetic field, which is quite convenient for optimization tasks. The poloidal magnetic flux $\psi_{\rm b}^{V}(r_{\rm b})$ is then obtained numerically, using the vacuum magnetic field provided by MAKEGRID to calculate the poloidal magnetic flux through multiple toroidal wedges at fixed θ angles. It is worth noting that the vacuum poloidal magnetic field calculated by MAKEGRID is as toroidally axisymmetric as the coil set considered. However, by inserting the calculated poloidal magnetic flux into the representation given by equation (10), any toroidally non-symmetric contribution is effectively removed.

In regards to the contribution from the plasma current, $\psi^{\rm P}(r_{\rm b})$, a different approach is followed. From previous experience with the SIESTA code [18], it turns out that the most numerically efficient way is to calculate the magnetic vector potential $\mathbf{A}$ (related to the magnetic field as $\mathbf{B}=\nabla\times\mathbf{A}$) created by the plasma current density by solving the differential equation,

$$\nabla\times(\nabla\times\mathbf{A})=\mu_0\mathbf{J}^P,\quad r\leqslant r_{\rm b}, \tag{27}$$

supplemented by two boundary conditions, one at $r = \epsilon \ll 1$ and another at $r = r_b$, both of which are evaluated by integrating numerically Biot–Savart's law from the magnetic potential,

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0}{4\pi} \iiint_{\text{plasma}} \frac{\mathbf{J}^{\mathbf{P}}(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|} \mathrm{d}\mathbf{r}'. \tag{28}$$

Equation (27) is solved in toroidal coordinates by applying various singular-value decomposition techniques to tackle the difficulties caused by the kernel associated to the curl operator, and using a spectral-finite differences scheme, similar to the one used in the GSB solver, that takes advantage of the block structure of the matrices that describe the discretized problem [19]. The contribution to the poloidal magnetic flux function, $\psi_b^P(r_b)$, is computed numerically by exploiting,

$$\iint_S \mathbf{B} \cdot \mathrm{d}\mathbf{S} = \oint_{\gamma_S} \mathbf{A} \cdot \mathbf{dl}, \tag{29}$$

where γ_S represents the boundary of the same toroidal wedges used for the vacuum calculation. As mentioned, it must be recalculated at the end of each outer iteration of the free-plasma-boundary solver.

5. Examples of free-plasma-boundary capabilities of the code

We will show now some examples selected to illustrate the plasma-free-boundary capabilities of the code. The magnetic configuration used for this examples is the baseline configuration of the ITER tokamak, with $I_p \simeq 15$ MA and $B_\phi \simeq 5.3$ T. It consists of 18 toroidal coils, 6 poloidal coils and 6 central solenoid modules. In MAKEGRID, each ITER coil is modelled by means of several filaments that take into account their geometrical features in detail, as described elsewhere [20]. The pressure and poloidal current profiles considered are shown in figure 4. These same profiles will be used for all calculations in the paper. The process starts by carrying out a free-boundary-run with the VMEC code, using MAKEGRID to calculate the vacuum magnetic field created by the coil set. The VMEC run provides an initial shape of the plasma at equilibrium (see figure 1) to guide the choice of the control circle (with minor radius $r_b = 4.2$ m, as shown in red), as well as values for the magnetic field and the plasma current density in the absence of plasma flow.

The VMEC equilibrium information is used to build the boundary condition needed by the GSB solver on the control circle, $\psi_b(r_b)$, as well as an initial guess for the poloidal magnetic flux function, that we will label as $\psi_0(R, Z)$. The five profiles needed for the GSB run are chosen in the following way. The pressure $P(\psi)$ and density $D(\psi)$ functions are assigned from the VMEC pressure profile and the relation $p = \rho^\gamma$. That is, entropy is set to $S(\psi) = 1$. The plasma edge is defined by the flux iso-surface for which the pressure vanishes in the VMEC solution. Outside of this surface, the pressure no longer follows the functional dependence on ψ, but is set to a constant, albeit very small value (to avoid problems when evaluating the auxiliary functions outside of the plasma). Since there is no gradient of pressure in this outer region, the equilibrium solution is insensitive to the actual value used (that we usually set equal to the smallest pressure value in VMEC's radial mesh). Finally, $B_0(\psi)$ is also imported from the VMEC

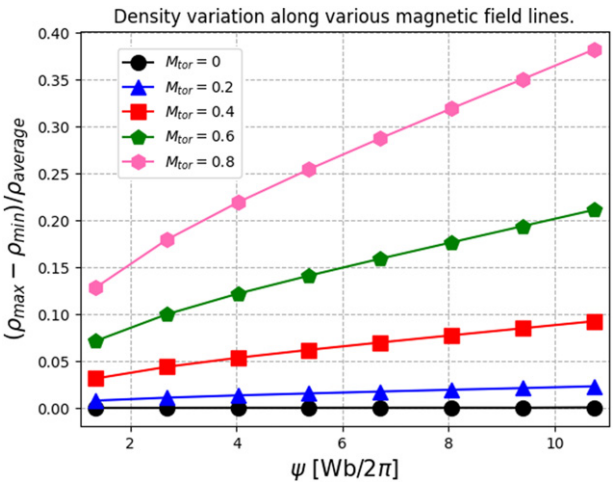

Figure 6. Variation of plasma density along various magnetic surfaces for the ITER equilibria calculated for the various scaling factors with the '(gf)' toroidal sonic Mach number profile.

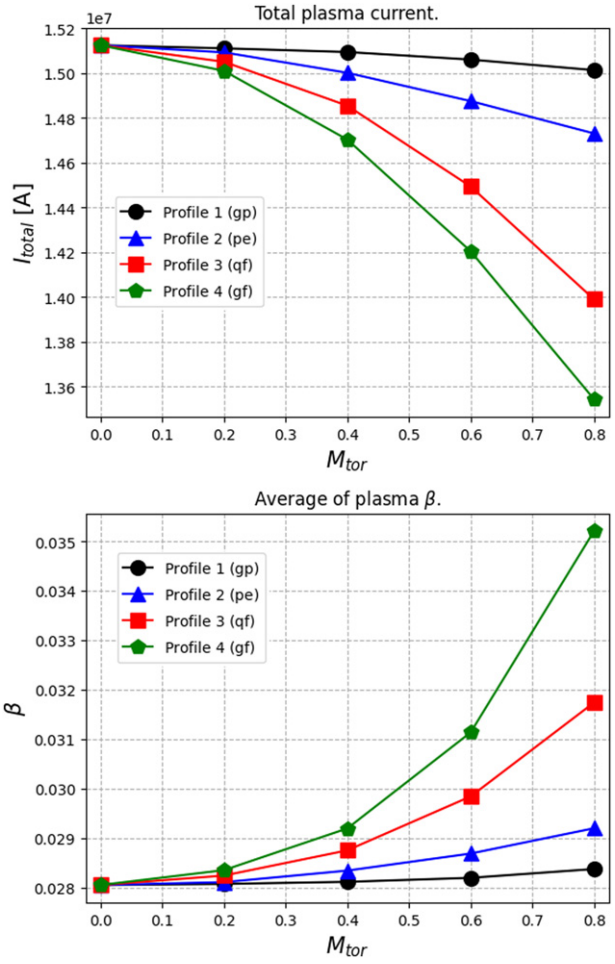

Figure 7. Total toroidal plasma current (top) and average plasma β for each of the ITER equilibria calculated for the various scaling factors used to scale the toroidal sonic Mach number profiles.

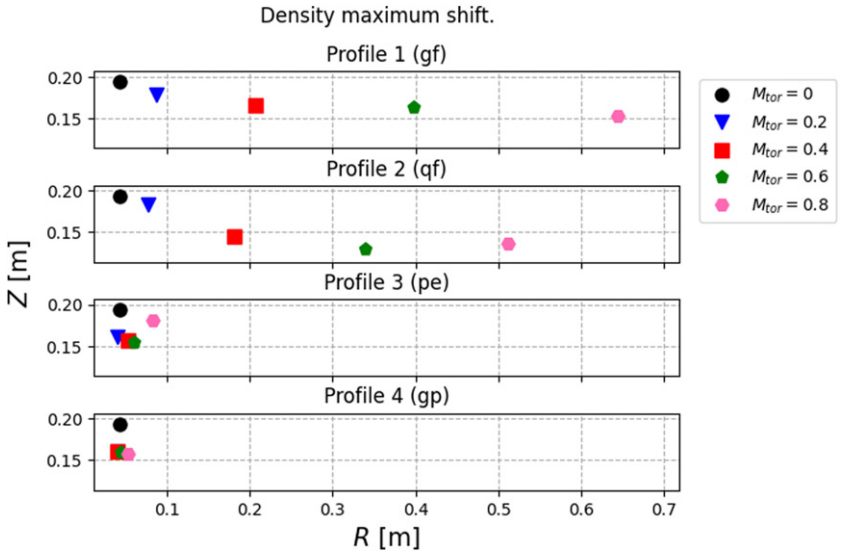

Figure 8. Position of the maximum of the plasma density in the ITER equilibria calculated for the various scaling factors used to vary the toroidal sonic Mach number profiles.

run. All numerical calculations have been done using a mesh with $M_r = 128$ radial points and $N_\theta = 122$ poloidal modes. The convergence in the spectral dimension is, as expected, much faster than in the radial direction.

All the examples shown are restricted on purpose to purely toroidal flows, although our GSB code can also handle poloidal flows. The reason is that toroidal runs are faster to run, allow for a more efficient debugging and illustrate the code capabilities equally well. The cases included here have all been obtained in a standard desktop computer and each takes between 6 to 10 h to run, the longest ones corresponding to the cases where the departure of the equilibrium from the initial guess is the largest. The four toroidal sonic Mach profiles considered, chosen solely for illustrative purposes, are shown in figure 5. The remaining free functions are kept the same. Each profile is normalized to its maximum and is identified with a unique label. Two of them, labelled as '(gf)' and '(qf)' have their maximum at $r = 0$, with '(qf)' decreasing quadratically with radius towards the edge, whilst '(gf)' is flat throughout most of the plasma and suddenly decreases at the very plasma edge. The other two correspond to profiles with toroidal rotation being only significant near the plasma edge, with '(gp)' being the most peaked of the two. At run time, the selected Mach number profile is multiplied by a positive global scaling factor $M_{tor} < 1$, so that flows remain subsonic everywhere in the domain. For each of the profiles considered, we have made runs using the scaling factors $M_{tor} = 0$, 0.2, 0.4, 0.6 and 0.8. Naturally, $M_{tor} = 0$ essentially recovers the VMEC solution, although some minor differences are found due to the solution being (re)-computed on a different computational mesh. Indeed, VMEC uses a non-geometrical angle for θ and the magnetic toroidal flux as radial coordinate, which leads to worse[better] radial resolution near the magnetic axis[edge], and a non-uniform angular mesh spacing. These differences are barely noticeable to the naked eye, though, being of order $(\Delta r)^2$ since both codes are second-order accurate in the radial direction. At non-zero toroidal Mach numbers, the global macroscopic quantities of the equilibria naturally change a bit as illustrated in figure 7, that shows the total toroidal plasma current and the average plasma beta (with $\beta \equiv 2\mu_0 p/B^2$ as usual) as a function of M_{tor}. The net toroidal plasma current is reduced as the plasma flow speed is increased, being the change smaller for profiles with edge-centered rotation and larger for profile with central rotation (i.e., (gf) and (qf)). The plasma beta, on the other hand, tends to increase as the plasma pressure becomes more peaked and its maximum shifted out towards regions with weaker field, being the effect again smaller for the edge-centered rotation profiles.

We illustrate now some of the capabilities of the free-plasma-boundary solver by monitoring and quantifying various changes of the ITER baseline equilibria as a function of the scaling factor M_{tor} for the various toroidal rotation profiles considered. Density is no longer a surface quantity in the presence of plasma flows, as illustrated in figure 6, that shows the variation of the density across various magnetic surfaces for various scaling factors using the '(gf)' toroidal sonic Mach number profile. Thus, a first interesting figure-of-merit is the position of the maximum of the plasma density in $(R - R_0, Z)$ space, shown in figure 8. As expected, the density maximum is pushed radially outwards by the centrifugal force associated to the toroidal rotation. The effect on the density is important if, at the location of the maximum for no flow, there is significant rotation. This is indeed the case for the '(gf)' and '(qf)' profiles. On the other hand, for the profiles with significant toroidal rotation in the outer part of the plasma, the maximum plasma density barely moves. It is worth to note that, although it might be argued that shifts of the maximum density are also observable and quantifiable in fixed-boundary runs, significant inaccuracies might occur if the plasma flow happens to deform the plasma boundary significantly with respect to the fixed boundary assumed.

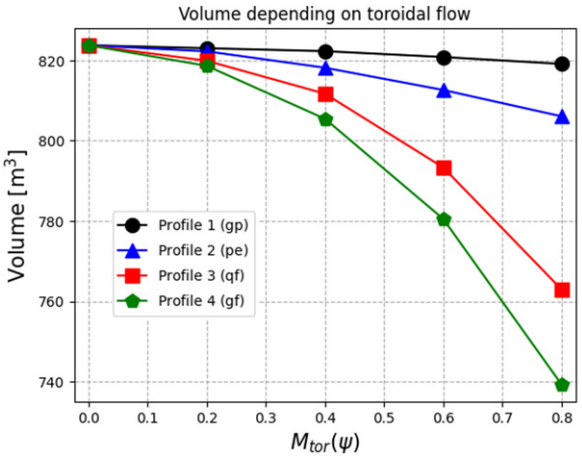

Figure 9. Plasma volume of the ITER equilibria calculated for the various scaling factors used to scale the toroidal sonic Mach number profiles.

A second figure-of-merit that does require full free-plasma-boundary capabilities to be evaluated is the determination of the confined plasma volume. It is calculated as the volume inside the pressure isosurface that defined the plasma edge in the VMEC solution. The change of the plasma volume of the ITER equilibria with $M_{\rm tor}$ is shown in figure 9 for the four rotation profiles considered. As with the maximum density shift, the change is pretty small for the toroidal rotation profiles confined to the plasma edge, but begins to become significant for the '(gf)' and '(qf)' profiles at $M_{\rm tor} \gtrsim 0.4$, reaching reductions of the confined plasma volume as large as 10% and 7%, respectively, at $M_{\rm tor} = 0.8$.

We move now to some figures-of-merit that characterise instead changes in the magnetic configuration, driven by the plasma flows, that can only be captured by free-plasma-boundary simulations. An interesting one is the displacement (if any) of the X-point at the bottom of the plasma, that separates the closed flux surfaces to the region where magnetic lines are open, and permits to uncouple the plasma from the reactor walls as well as to direct the heat and plasma exhaust towards the divertor. An illustration of this displacement is shown in figure 10, that magnifies the region around the X-point for two runs done using the '(gf)' toroidal sonic Mach number flow profile for scaling factors $M_{\rm tor} = 0.4$ (top frame) and $M_{\rm tor} = 0.8$ (bottom frame). As seen in the figure, the X-point is displaced about 30 cm upwards and 15 cm to the right for the $M_{\rm tor} = 0.8$ scaling factor with respect to the position in the absence of flow (show in dashed blue). For the more moderate rotation, $M_{\rm tor} = 0.4$, the displacement of the X-point is about 8 cm upwards and 5 cm to the right. Values of the displacements of the X-point have been collected in figure 11 or all the runs considered, showing again that edge rotation has little effect on the X-point location, whilst the core-centered torodial rotation profiles lead to significant displacements both in the vertical (upwards) and radial (outwards) direction, that always push the X-point away from the divertor, for $M_{\rm tor} \gtrsim 0.4$.

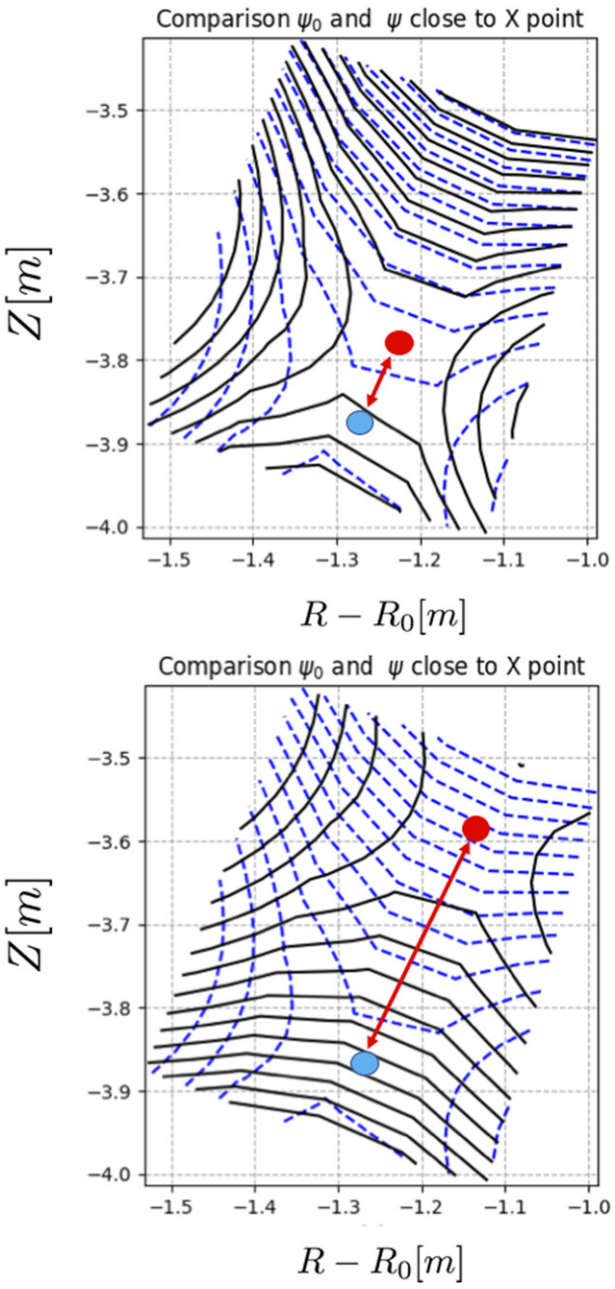

Figure 10. Displacement of the X-point for the '(gf)' toroidal sonic Mach number flow profile in figure 5 with scaling factors $M_{\rm tor} = 0.4$ (top) and $M_{\rm tor} = 0.8$ (bottom) relative to the position in the initial guess (shown in dashed blue).

As a last figure-of-merit, we have also looked at the change in the position of the magnetic axis, that is shown for all toroidal rotation profiles in figure 12. Again, this shift becomesonly significant for core-centered torodial rotation profiles and $M_{\rm tor} \gtrsim 0.4$. As was the case of the maximum density shift, it could also be argued here that the magnetic axis shift could also be captured in fixed-boundary-runs. But as in that case, significant inaccuracies might occur if the plasma flow happens to deform the plasma boundary significantly with respect to the fixed boundary assumed.

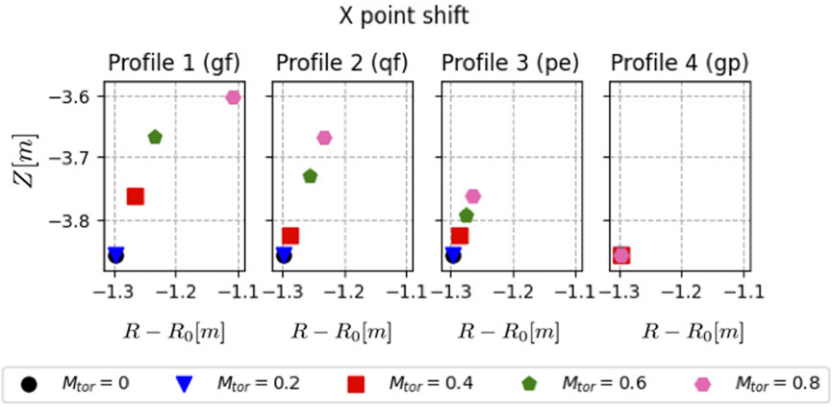

Figure 11. *X*-point position for the series of ITER equilibria calculated for the various scaling factors used to scale the toroidal sonic Mach number profiles.

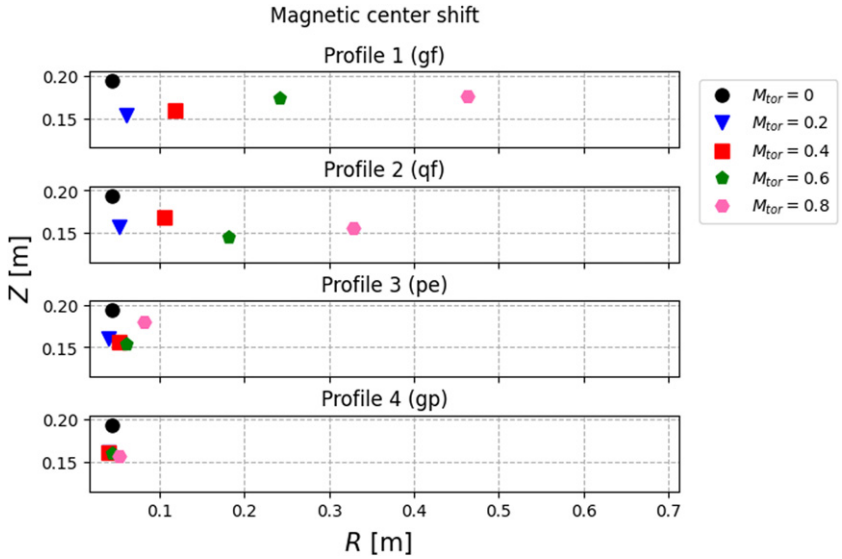

Figure 12. Magnetic axis position for the series of ITER equilibria calculated for the various scaling factors used to scale the toroidal sonic Mach number profiles.

6. Conclusions

It has been shown that a fixed-boundary solver for the GSB system of equations can be successfully coupled to a scheme, initially developed for the calculation of static MHD equilibria in stellarators, to enable it to efficiently compute free-plasma-boundary 2D equilibria in the presence of plasma flows. The key to success is provided by the decoupling of the surface where the boundary conditions are imposed form the shape of the plasma edge. Once this is done, tokamak magnetic configurations with arbitrarily complex coil sets can be easily treated within the framework by coupling the iterative solver to the VMEC and MAKEGRID codes.

In its current version, though, the surface where boundary conditions are imposed must be a circle. This is imposed by our choice of coordinates. We are fully aware of the fact that this is inadequate for 2D tokamak configurations with very elongated plasmas (spherical tokamaks, for instance), for which it might impossible to trace a circle that leaves all coils outside and the plasma inside. For that reason, we are already planning a development of the current code to enable the use of generalized coordinates instead of the current cylindrical coordinates. Building on the principles shown to work in this paper, the idea we will try to pursue is to replace R and θ by non-standard radial and angular definitions to allow to impose the boundary conditions on control surfaces of arbitrary shape. In order to further facilitate the integration with VMEC, we will explore the use of the same generalized coordinates from VMEC. Using the toroidal flux (or a function of it) of the VMEC equilibrium with no plasma flow as a fixed coordinate system should allow to define control surfaces that closely follow the shape of the plasma, making possible to draw a clear separation from the coils, as required by the principles illustrated in this paper.

Finally, we have shown in this paper that free-plasma-boundary capabilities opens up some interesting possibilities. Some have been illustrated in this paper, in which we have evaluated the effects of various toroidal rotation profiles on the

plasma and magnetic properties of the ITER baseline $Q = 10$ plasma scenario, although those are sizeable at $M_{\text{tor}} \gtrsim 0.4$, values which are on the high end of what is expected for ITER. It is worth noting that the scheme presented is not restricted to toroidal plasma flows, since the GSB solver deals with both toroidal and poloidal flows. A similar study to the one presented here but with poloidal flows is planned for the near future.

Acknowledgments

This work has been carried out within the work programme of the ITER Scientist Fellow Network. The authors would like to thank Professor V. Tribaldos for useful comments. This research has been partially sponsored by the Spanish National Research Project No. PID2019-110734RB-I00. This work has been partially supported by Comunidad de Madrid under the agreement with UC3M in the line of Excellence of University Professors (EPUC3M14). Use has been made of *Uranus*, a supercomputed cluster located at the Universidad Carlos III de Madrid (Spain) and funded jointly by EU FEDER funds and by the Spanish Government via national projects UNC313-4E-23612, ENE2009-12213-C03-03, ENE2012-33219, ENE2012-31753 and ENE2015-68265. Disclaimer: ITER is the Nuclear Facility INB No. 174. The views and opinions expressed herein do not necessarily reflect those of the ITER Organization.

ORCID iDs

J.M. Reynolds-Barredo https://orcid.org/0000-0002-7412-9419
A. Loarte https://orcid.org/0000-0001-9592-1117

References

[1] Shafranov V.D. 1966 Plasma equilibrium in a magnetic field *Rev. Plasma Phys.* **2** 103
[2] Freidberg J.P. 2014 *Ideal MHD* (Cambridge: Cambridge University Press)
[3] Woltjer L. 1959 Hydromagnetic equilibrium IV. Axisymmetric compressible media *Astrophys. J.* **130** 405
[4] Morozov A.I. and Soloviev L.S. 1963 *Sov. Phys. Dokl.* **8** 243
[5] Zehrfeld H.P. and Green B.J. 1972 Stationary toroidal equilibria at finite beta *Nucl. Fusion* **12** 569–75
[6] Hameiri E. 1983 The equilibrium and stability of rotating plasmas *Phys. Fluids* **26** 230–7
[7] Beliën A.J.C., Botchev M.A., Goedbloed J.P., van der Holst B. and Keppens R.K. 2002 FINESSE: axisymmetric MHD equilibria with flow *J. Comput. Phys.* **182** 91–117
[8] Guazzotto L., Betti R., Manickam J. and Kaye S. 2004 Numerical study of tokamak equilibria with arbitrary flow *Phys. Plasmas* **11** 604–14
[9] Hirshman S.P., Sanchez R. and Cook C.R.C. 2011 SIESTA: a scalable iterative equilibrium solver for toroidal applications *Phys. Plasmas* **18** 062504
[10] Hirshman S.P. and Whitson J.C. 1983 Steepest-descent moment method for three-dimensional magnetohydrodynamic equilibria *Phys. Fluids* **26** 3553–68
[11] Spong D.A. *et al* 1983 Physics issues of compact drift optimized stellarators *Nucl. Fusion* **41** 711–6
[12] Lütjens H., Bondeson A. and Sauter O. 1996 The Chease code for toroidal MHD equilibria *Comput. Phys. Commun.* **97** 219–60
[13] Lao L.L., St John H., Stambaugh R.D., Kellman A.G. and Pfeiffer W. 1985 Reconstruction of current profile parameters and plasma shapes in tokamaks *Nucl. Fusion* **25** 1611
[14] Goedbloed J.P. and Lifschitz A. 1997 Stationary symmetric magnetohydrodynamic flows *Phys. Plasmas* **4** 3544
[15] Gunderson L.M. 2020 Solar equilibrium à la Grad–Shafranov *PhD Thesis* Princeton University
[16] Betti R. and Freidberg J.P. 2000 Radial discontinuities in tokamak magnetohydrodynamic equilibria with poloidal flow *Phys. Plasmas* **7** 2439–48
[17] Maschke E.K. and Perrin H.J. 1984 An analytic solution of the stationary MHD equations for a rotating toroidal plasma *Phys. Lett.* A **102** 106–8
[18] Peraza-Rodriguez H., Reynolds-Barredo J.M., Sanchez R., Geiger J., Tribaldos V., Hirshman S.P. and Cianciosa M. 2017 Extension of the siesta MHD equilibrium code to free-plasma-boundary problems *Phys. Plasmas* **24** 082516
[19] Reynolds-Barredo J.M., Peraza-Rodríguez H., Sanchez R. and Tribaldos V. 2020 A novel efficient solver for Ampere's equation in general toroidal topologies based on singular value decomposition techniques *J. Comput. Phys.* **406** 109214
[20] Mitchell N., Devred A., Libeyre P., Lim B. and Savary F. 2012 The ITER magnets: design and construction status *IEEE Trans. Appl. Supercond.* **22** 4200809