

This is a postprint version of the following published document:

Reynolds-Barredo, J. M., Peraza-Rodríguez, H., Sanchez, R., & Tribaldos, V. (2020). A novel efficient solver for Ampere's equation in general toroidal topologies based on singular value decomposition techniques. *In Journal of Computational Physics*, 406, 109214-109229

DOI: [10.1016/j.jcp.2019.109214](https://doi.org/10.1016/j.jcp.2019.109214)

© 2019 Elsevier Inc. All rights reserved



This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](https://creativecommons.org/licenses/by-nc-nd/4.0/).

A novel efficient solver for Ampere's equation in general toroidal topologies based on singular value decomposition techniques

J. M. Reynolds-Barredo, H. Peraza-Rodríguez, R. Sanchez, V. Tribaldos

Departamento de Física, Universidad Carlos III de Madrid, Leganés 28911, Spain

Abstract

A new method is proposed to solve Ampere's equation in an arbitrary toroidal domain in which all currents are known, given proper boundary conditions for the magnetic vector potential. The novelty of the approach lies in the application of singular value decomposition (SVD) techniques to tackle the difficulties caused by the kernel introduced by the curl operator. The physical origin of the kernel is on the gauge associated with the magnetic field. To increase the efficiency of the solver, the problem is represented by means of a dual finite difference-spectral scheme in arbitrary generalized toroidal coordinates, which permits to take advantage of the block structure exhibited by the matrices that describe the discretized problem. The result is a fast and efficient solver, up to three times faster than the double-curl method in some cases, that provides an accurate solution of the differential form of Ampere law while guaranteeing a zero divergence of the resulting magnetic field down to machine precision.

Keywords: Ampere's law, singular value decomposition, SVD, spectral method, finite differences, toroidal topology, magnetohydrodynamics, MHD

1. Introduction

There are many situations where the magnetic field created by a prescribed set of currents needs to be calculated. For instance, this is the case in magnetically confined, fusion toroidal plasmas such as those inside an stellarator, when
5 one often needs to calculate the magnetohydrodynamical (MHD) equilibrium of a plasma confined by a set of external coils [1]. Since the response of the plasma is to generate its own currents, finding the magnetic field of the equilibrium state is a non-linear problem that requires the use of sophisticated numerical techniques. The resulting magnetic field may have a complicated topology that
10 could include not only closed toroidal magnetic surfaces, but magnetic islands and stochastic regions as well. In addition, the treatment of the boundary between the confined plasma and the empty region that separates it from the coils is delicate. In the literature, several numerical codes can be found that address

15 this problem such as PIES [2], HINT [3] or SIESTA [4], among others. Many of these codes are often iterative in nature and thus require an initial guess for the magnetic field that needs to be sufficiently close to the solution to guarantee convergence. From there, techniques such as the Newton or Picard methods may be used to iterate until the nearest equilibrium state is found.

20 Each iterative code uses a different approach to find a suitable initial guess for the magnetic field. Since one confides in the iterative scheme to refine the initial guess, these approaches are required to be fast, even at the expense of not being terribly accurate. A path that is often followed is to calculate the vacuum field created by the external coils by integrating Biot-Savart's law over the computational volume. To this vacuum field, a guess of the magnetic field 25 generated by the plasma currents must also be added. This is often provided by other codes that yield reasonably good estimates relatively fast at the price of making some additional simplifying assumptions. An example is provided by the widely used VMEC equilibrium code [5]. Since VMEC efficiently provides the complete magnetic field inside of the plasma volume (i.e., the vacuum field 30 plus the one created by the plasma currents) by assuming that it must have a topology with closed toroidal magnetic surfaces (it might thus contain divergent shielding current sheets that should open up to yield magnetic islands, if this was not prevented by the fixed topology assumed), one only needs to estimate the magnetic field created by the plasma currents in the vacuum region and 35 added to the one created by the coils. This can be easily done by applying the so-called casing principle [6, 7], or by applying Biot-Savart equation but integrating over the plasma currents instead of the coils.

In fact, this second path could be followed from the start and be applied to the whole computational volume. Indeed, one could start with a sufficiently 40 good guess for the plasma currents as obtained by a simpler code (for instance, with VMEC) and then obtain a guess for the total magnetic field everywhere by integrating Biot-Savart equation including both the currents in the coils and within the plasma [8]. This approach provides a smoother initial guess than the one we sketched previously, since it does not patch together different solutions 45 over various domains. Integration is however neither optimal nor efficient. Although it is a linear operation in the currents that can be easily parallelized, it is still very time consuming. In addition, numerical errors may also become an issue, since the generalized toroidal coordinate systems in which most of these codes internally operate require that the result of the integrations be trans- 50 formed to the local basis, which often distorts the value of the divergence of the field that may become significant. Clearly, one could fix this problem by moving to a formulation in terms of the magnetic vector potential such as,

$$\mathbf{A}(\mathbf{r}) = \frac{\mu_0}{4\pi} \int_{\text{all currents}} \mathbf{dl}' \frac{I(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}, \quad (1)$$

from which $\mathbf{B} = \nabla \times \mathbf{A}$, but the large computational time required for the integration and the inaccuracies introduced by the subsequent transformation 55 still remain.

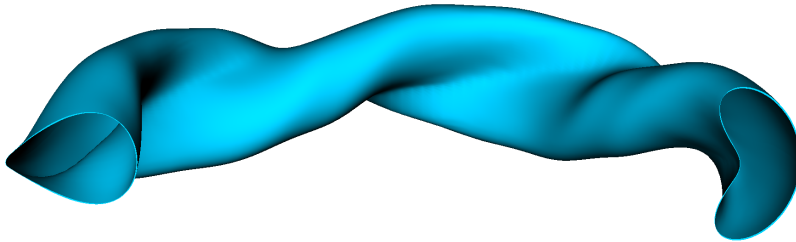


Figure 1: Section of a typical domain where the plasma magnetic field has to be calculated. The domain is limited by the vacuum chamber wall inside of which the plasma is confined by the magnetic field. This geometry corresponds to the W7-X stellarator, in operation in Greifswald (Germany) since 2016. It is always assumed that the coils that confine the plasma are always out of the computational domain.

For all these reasons, we present a different approach in this paper that relies on solving efficiently the differential form of Ampere’s law instead. That is, we solve numerically the equation:

$$\nabla \times \nabla \times \mathbf{A}(\mathbf{r}) = \mu_0 \mathbf{J}(\mathbf{r}), \quad (2)$$

inside a toroidal volume that includes the plasma and extends into the vacuum region that separates it from the coils that confine it, but that excludes the coils (see Fig. 1 for an example). Thus, only the current that flows within the plasma is included in the right hand side. The effect of the coils is felt through the boundary conditions of the problem, that must be chosen properly. In our case, they correspond to the requirement that the value of the vector potential close to the central axis and the outer boundary of the toroidal computational domain (see Fig. 2, where an example of the surfaces where the boundary conditions are enforced is shown) be equal to that obtained by integrating Eq. 1 at those locations including both the coil and plasma currents. Notice how the gauge freedom allows to add any gradient $\nabla\chi$ to the magnetic vector potential solution of Eq. 2 and still be solution. For example, the Biot-Savart integration (Eq. 1) is associated with the Coulomb gauge $\nabla \cdot \mathbf{A} = 0$. The existence of this freedom makes challenging the numerical solution of the associated linear system. When discretizing the differential problem, the free variables associated with the gauge will show up as a large kernel of one third of the total number of variables on the system to solve. It is one third because there is at least one free parameter on each position of the space (the value of χ on that position) and three unknowns on the same position (the three components of the vector potential). Also notice how, to keep consistency, the input current $\mathbf{J}(\mathbf{r})$ has to be adivergent or, in other words, should not have any component in the kernel of the curl operator.

The implementation of the algorithm that will be discussed next assumes the use of the same generalized toroidal coordinates that the VMEC code uses, since this solver was first developed to support the free-boundary extension of

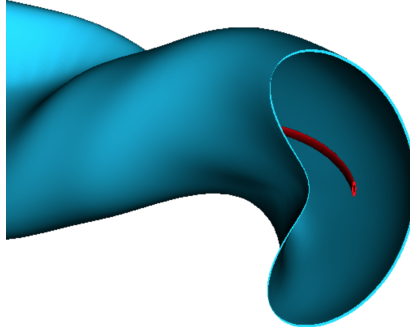


Figure 2: Inner and outer radial surfaces where the boundary conditions are implemented for the domain previously shown in Fig. 1.

the SIESTA MHD code [8], that shares (almost) the same internal coordinates than VMEC. However, this is not a necessary condition and the algorithm can be easily converted to any other generalized toroidal coordinate system. The only thing required is that a spectral representation will be used for the poloidal and toroidal directions and finite differences in the radial direction of the torus. The reason is that these choices lead to a discretized form of the problem, $\mathbf{Q} \cdot \mathbf{x} = \mathbf{b}$, in which \mathbf{Q} displays a block-tridiagonal structure. Singular value decomposition (SVD) techniques can then be used to deal with the gauge related kernel of the \mathbf{Q} matrix very efficiently. Under the SVD decomposition, the null singular values of an operator discretization are associated with the kernel of that operator, for example in the case of the curl, with any expression composed as a gradient. This is the powerful mechanism that will be used to obtain projections of the current or vector potential over the operators kernels, helping obtaining the solution in an efficient way.

It will be also shown that the proposed technique is faster, when many problems have to be solved under the same geometry, to the simpler but effective double-curl technique (see for example [9]). On this technique, the kernel can be eliminated by using the identity:

$$\nabla^2 \mathbf{A}(\mathbf{r}) = \nabla (\nabla \cdot \mathbf{A}(\mathbf{r})) - \nabla \times \nabla \times \mathbf{A}(\mathbf{r}) \quad (3)$$

Notice how, under the Coulomb gauge we can write

$$\nabla^2 \mathbf{A}(\mathbf{r}) = -\mu_0 \mathbf{J}(\mathbf{r}), \quad (4)$$

the kernel has vanished out. To help understand this affirmation, we can use the Helmholtz decomposition $\mathbf{A} = \nabla \chi + \nabla \times \mathbf{L}$ and notice that the kernel of the first operator on rhs of Eq. 3 is any curl and the kernel of the second operator of rhs is any gradient. The resultant system, lacking kernel, can be directly inverted either with a direct solver for small systems or with an preconditioned iterative solver for larger ones.

Finally, it is worth saying that the approach is not restricted to fusion applications, although we focus on them to illustrate its performance in this paper.

110 The paper is thus organized as follows. Sec. 2.1 gives details about the coordinate system used for the implementation and the finite difference-spectral representation of the quantities of interest. In Sec. 2.2, the result of discretizing Ampere’s Law according to these prescriptions is described. Then, in Sec. 3, the algorithm that we propose to solve the problem is presented, focusing on
 115 the use of SVD to deal with the kernel of the problem. Sec. 4 illustrates its performance by applying it to a complicated toroidal geometry and compares its performance to a direct integral implementation and to a double-curl method implementation. Finally, in Sec. 5, several conclusions are drawn.

2. Discretization scheme

120 2.1. Numerical grid and quantities of interest

Figure 1 shows a typical toroidal domain in which the magnetic field is to be calculated. By introducing a set of curvilinear, toroidal coordinates (s, θ, ϕ) , the domain becomes a (dense) set of nested toroidal surfaces (which are not necessarily magnetic surfaces, since s does not need to be a flux quantity!).
 125 Here, s is a *radial* coordinate that defines the toroidal surface, and θ and ϕ are the two 2π -periodic angles that characterize the position of any point within each toroidal surface. In the applications discussed in this paper, (s, θ, ϕ) are chosen to be the same coordinates that the VMEC code uses, but this is clearly not required. It is sufficient that the transformation between our toroidal coordinate
 130 system of choice and the usual cylindrical coordinates be known.

Since the two angles θ and ϕ are periodic, we assume a spectral representation for those directions which is completed by using finite-differences in the radial direction. This choice is very important, since it endows the matrices that will describe the problem with a block-diagonal structure that is essential for the
 135 efficiency of the algorithm that we will propose later. For simplicity, we choose ϕ to coincide with the geometrical cylindrical angle, as VMEC does. We also assume up-down symmetry at the $\phi = 0$ plane (this is a common symmetry for stellarators [10], but it can be easily removed by considering a complex Fourier representation instead), which halves the size of the problem by endowing all
 140 quantities with either a cosine or sine parity. Therefore, the aforementioned coordinate transformation to the usual cylindrical coordinates is given by:

$$\begin{aligned}
 R(s, \theta, \phi) &= \sum_{m=0}^M \sum_{n=-N}^N R_{mn}(s) \cdot \cos(m\theta + nN_p\phi) \\
 Z(s, \theta, \phi) &= \sum_{m=0}^M \sum_{n=-N}^N Z_{mn}(s) \cdot \sin(m\theta + nN_p\phi)
 \end{aligned} \tag{5}$$

where N_p is the number of toroidal periods of the domain (if not known or absent, one can simply set $N_p = 1$). The harmonics $R_{mn}(s)$ and $Z_{mn}(s)$ define

the transformation and are assumed known. The number of harmonics included
 145 in the representation is $N_{modes} = (M + 1) * (2N + 1)$. For a typical stellarator
 calculation, $M, N \sim 20$, so that $N_{modes} \sim 1,000$. On the Appendix A can
 be found details about the staggered finite difference scheme used in the radial
 direction.

2.2. Ampere's equation

150 We will express Ampere's equation in terms of the two coupled differential
 equations:

$$\begin{aligned}\nabla \times \mathbf{A}(\mathbf{r}) &= \mathbf{B}(\mathbf{r}) \\ \nabla \times \mathbf{B}(\mathbf{r}) &= \mathbf{J}(\mathbf{r})\end{aligned}\quad (6)$$

which become, in our curvilinear system of coordinates (s, θ, ϕ) ,

$$\begin{aligned}\left(\epsilon_{\alpha\beta\gamma} \frac{\partial}{\partial x_\beta} A_\gamma\right) &= \sqrt{g} B^\alpha \\ \left(\epsilon_{\alpha\beta\gamma} \frac{\partial}{\partial x_\beta} B_\gamma\right) &= \sqrt{g} J^\alpha,\end{aligned}\quad (7)$$

where remember μ_0 has been absorbed within the current and we have used
 the usual Levi-Civita symbol, ϵ_{ijk} , and assumed the repeated index summation
 155 convention. The two sets of equations are connected via the local transformation
 from covariant to contravariant components, given by,

$$B^\alpha = g^{\alpha\beta} B_\beta, \quad \alpha, \beta = s, \theta, \phi. \quad (8)$$

that requires the use of the contravariant metric tensor, $g^{\alpha\beta}$, that is easily
 computed from the coordinate transformation (Eqs. 5).

By making the discretization scheme sketched in the previous subsection
 160 and the Appendix A explicit, the equation for the magnetic potential vector
 becomes:

$$\begin{aligned}-\frac{m}{2} \left([A_\phi]_{mn}^{i-1} + [A_\phi]_{mn}^i \right) + \frac{nN_p}{2} \left([A_\theta]_{mn}^{i-1} + [A_\theta]_{mn}^i \right) &= [\sqrt{g} B^s]_{mn}^i \quad (9) \\ \frac{nN_p}{2} \left([A_s]_{mn}^{i-1} + [A_s]_{mn}^i \right) - \frac{1}{\Delta s} \left([A_\phi]_{mn}^i - [A_\phi]_{mn}^{i-1} \right) &= [\sqrt{g} B^\theta]_{mn}^i \\ \frac{1}{\Delta s} \left([A_\theta]_{mn}^i - [A_\theta]_{mn}^{i-1} \right) - \frac{m}{2} \left([A_s]_{mn}^{i-1} + [A_s]_{mn}^i \right) &= [\sqrt{g} B^\phi]_{mn}^i\end{aligned}$$

where $i \in [2, N_s]$, since the magnetic field is defined on the half mesh.

Similarly, the discretized version of the equation for the magnetic field, is
 given by:

$$\begin{aligned}-\frac{m}{2} \left([B_\phi]_{mn}^i + [B_\phi]_{mn}^{i+1} \right) + \frac{nN_p}{2} \left([B_\theta]_{mn}^i + [B_\theta]_{mn}^{i+1} \right) &= [\sqrt{g} J^s]_{mn}^i \quad (10) \\ \frac{nN_p}{2} \left([B_s]_{mn}^i + [B_s]_{mn}^{i+1} \right) - \frac{1}{\Delta s} \left([B_\phi]_{mn}^{i+1} - [B_\phi]_{mn}^i \right) &= [\sqrt{g} J^\theta]_{mn}^i \\ \frac{1}{\Delta s} \left([B_\theta]_{mn}^{i+1} - [B_\theta]_{mn}^i \right) - \frac{m}{2} \left([B_s]_{mn}^i + [B_s]_{mn}^{i+1} \right) &= [\sqrt{g} J^\phi]_{mn}^i\end{aligned}$$

165 where again $i \in [2, N_s - 1]$.

Eqs. 9 – 10 can be easily cast into the matrix form $\mathbf{Q} \cdot \mathbf{x} = \mathbf{b}$, as will be discussed in what follows. Before proceeding any further, it is however worth noting that the divergence of the magnetic field, when calculated under this discretization scheme, is given by:

$$\begin{aligned} [\nabla \cdot \mathbf{B}]_{mn}^i &= \frac{1}{\Delta s} \left([\sqrt{g}B^s]_{mn}^{i+1} - [\sqrt{g}B^s]_{mn}^i \right) - \\ &\quad - \frac{m}{2} \left([\sqrt{g}B^\theta]_{mn}^{i+1} + [\sqrt{g}B^\theta]_{mn}^i \right) - \\ &\quad - \frac{nN_p}{2} \left([\sqrt{g}B^\phi]_{mn}^{i+1} + [\sqrt{g}B^\phi]_{mn}^i \right) = 0, \end{aligned} \quad (11)$$

170 as can be shown by substituting Eqs. 9 into Eq. 11.

3. Numerical solution

As with many other numerical problems, Eqs. 9 – 10 are ultimately reduced to an algebraic problem of the type $\mathbf{Q} \cdot \mathbf{x} = \mathbf{b}$, where the vector \mathbf{x} contains the component of the desired magnetic vector potential. Due to the presence of the double curl in the equation for the magnetic potential, \mathbf{Q} becomes a singular matrix which makes the solution of the algebraic problem highly non-trivial. Our approach to deal with it requires that \mathbf{Q} is block-diagonal and symmetric.

3.1. Structure of the problem

In order to make sure that \mathbf{Q} is block-diagonal and symmetric we choose two boundary conditions in order to close the problem in the following way (see Fig. 2). The first one is imposed at a generic internal position, $s_{i_0}^f$ ($i_0 \ll N_s$), where we require the magnetic potential to take the value computed by direct integration of Eq. 1 including all currents; the second one, by requiring the value of the magnetic potential at the boundary (i.e., $s_{N_s}^f$) to be also equal to the one resulting from the direct integration. It is worth noting that, at first glance, it might appear that the influence of the external coils could be included using just one boundary condition at the outer surface, and use a standard regularity condition at the toroidal axis (i.e., requiring $A_s = A_\theta = 0$ at $s = 0$). However, to avoid dealing with the singularity at the axis, to keep the symmetry of \mathbf{Q} and for clarity in the exposition, we prefer to keep an additional internal boundary condition, which helps the numerical solution at the (small) price of having to estimate the vector potential near the axis. That will not be a problem in our case, as will be illustrated in section 4.

In Fig. 3 we illustrate the variables involved in the problem. The unknowns vector \mathbf{x} is composed by the components $[A_\alpha]_{mn}^i$ for $i \in [i_0 + 1, N_s - 1]$. The sources vector \mathbf{b} will contain the currents values $[\sqrt{g}J_\alpha]_{mn}^i$ for $i \in [i_0 + 1, N_s - 1]$ plus the effect on the source of the boundary condition associated with the values $[A_\alpha]_{mn}^{i_0}$ and $[A_\alpha]_{mn}^{N_s}$. The number of modes is $N_{modes} = (M + 1) \cdot (2N + 1)$ and

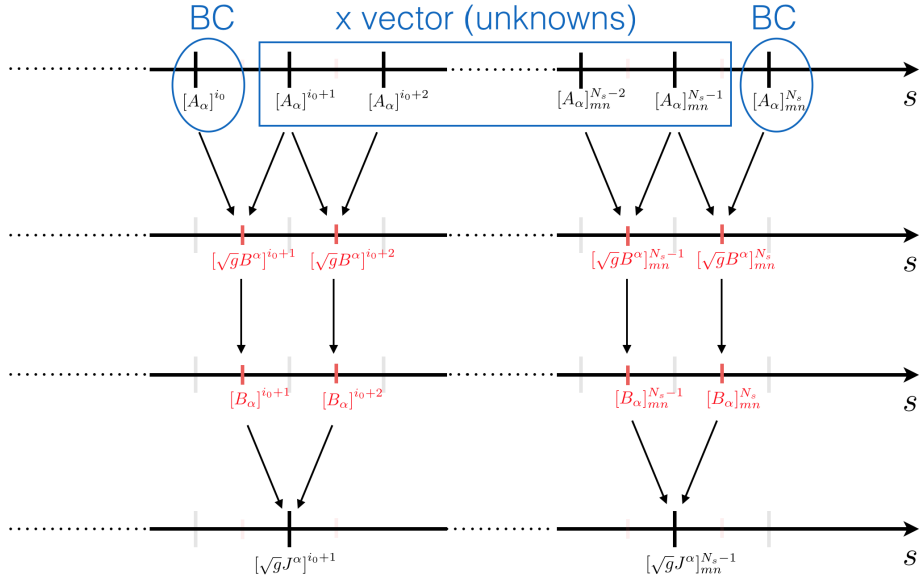


Figure 3: Representation of the different variables and their interdependence in the radial mesh. In black full mesh quantities and in red half mesh quantities.

the number of surfaces is $N_{surf} = N_s - i_0 - 1$. The total number of unknowns is thus $N_{tot} = 3N_{surf}N_{modes}$.

Next, we discuss the \mathbf{Q} matrix. After inspecting carefully Eqs. 9 – 10, it is possible to see that the matrix of the problem is obtained from the product of three matrices:

$$\mathbf{Q} = \mathbf{R}^T \cdot \mathbf{M} \cdot \mathbf{R} \quad (12)$$

whose structures are shown in Fig. 4. The matrix \mathbf{M} represents the multiplication by the jacobian in real space plus the transformation from contravariant to covariant components of the magnetic field. Since this operation is done separately at each surface, only angular modes and components are coupled. As a result, \mathbf{M} is a symmetric, square, block-diagonal matrix (see left frame of Fig. 4). The matrix \mathbf{R} , on the other hand, corresponds to the application of the curl operator. Since this implies radial (and angular) derivatives, the use of finite differences implies the coupling of each radial node to the previous and the next, but it does not couple different modes due to the spectral representation. It does however couple the same mode of different components (i.e., s , θ and ϕ), since their derivatives are combined to form the curl. Therefore, \mathbf{R} is block-banded in surfaces (see right frame of Fig. 4), but the blocks only contain a few diagonals that represent the coupling between components. In addition, \mathbf{R} is not a square matrix. Due to the features of both \mathbf{M} and \mathbf{R} , the resulting matrix \mathbf{Q} is block tridiagonal. This structure has a strong impact in the solution process, allowing it to be accelerated considerably, as will be shown

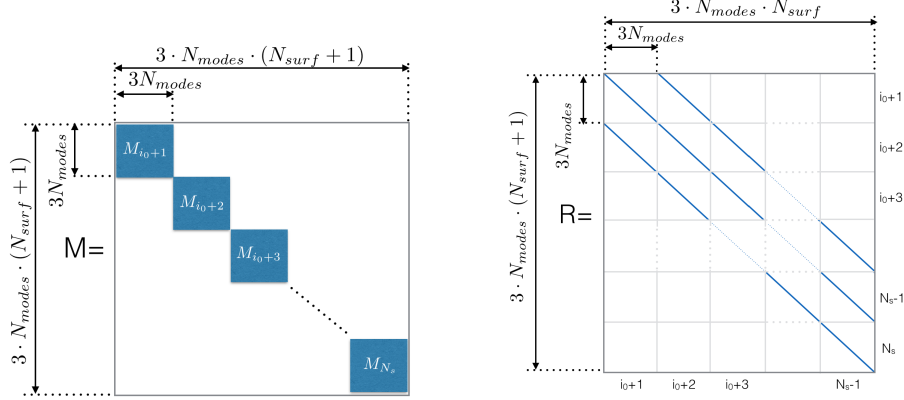


Figure 4: Structure of the \mathbf{M} matrix (left) and \mathbf{R} matrix (right). Observe that \mathbf{R} is not square.

220 in the next section.

3.2. Solution of the system of equations

The matrix \mathbf{Q} that represents the discrete problem is singular in the sense that its kernel has a dimension larger than zero. This is a consequence of the fact that the curl vanishes when applied to the gradient of any function. Therefore, special care has to be taken in order to solve it, since the problem is ill-posed. In order to illustrate the difficulties in front of us, we will start by describing a direct attempt to solve the problem by means of the eigenvalues and eigenvectors of \mathbf{Q} . Once the limitations are made apparent, we present our proposed SVD-based scheme, whose performance will be illustrated in the next section.

3.2.1. Eigenvalue/eigenvector approach to the solution of the $\mathbf{Q} \cdot \mathbf{x} = \mathbf{b}$ system.

In order to find a solution to the problem, due to the presence of a non-trivial kernel, one proceeds by finding first the decomposition of \mathbf{Q} in terms of its eigenvalues and eigenvectors. That is, we rewrite the system as

$$\mathbf{Q} \cdot \mathbf{x} = \mathbf{\Gamma} \cdot \mathbf{D} \cdot \mathbf{\Gamma}^{-1} \cdot \mathbf{x} = \mathbf{b} \quad (13)$$

235 where \mathbf{D} is a diagonal matrix containing the eigenvalues of \mathbf{Q} , $\{\lambda_i, i = 1, \dots, N_Q\}$, and $\mathbf{\Gamma}$ is formed by its related eigenvectors ordered in column form. Here, N_Q is the linear dimension of \mathbf{Q} . Since \mathbf{Q} is symmetric, $\mathbf{\Gamma}^{-1} = \mathbf{\Gamma}^\top$. Thus, by projecting the vectors \mathbf{x} and \mathbf{b} onto the basis formed by all eigenvectors (i.e., $\bar{\mathbf{x}} = \mathbf{\Gamma}^\top \cdot \mathbf{x}$ and $\bar{\mathbf{b}} = \mathbf{\Gamma}^\top \cdot \mathbf{b}$), the system of equations can be rewritten as:

$$\mathbf{D} \cdot \bar{\mathbf{x}} = \bar{\mathbf{b}} \quad (14)$$

240 Since \mathbf{D} is a diagonal matrix, each component of the solution satisfy a single, decoupled equation given by:

$$\lambda_i \cdot \bar{x}_i = \bar{b}_i, \quad i = 1, \dots, N_Q. \quad (15)$$

However, if the dimension of the kernel of \mathbf{Q} , $N_Q^{ker} > 0$, there are N_Q^{ker} null eigenvalues, that implies a non unique solution or even a non compatible system. To work around this issue, one usually orders the eigenvalues so that the null eigenvalues are placed at the last positions, $i \in [N_Q - N_Q^{ker} + 1, N_Q]$. The appropriate solution of the problem is then found by considering the pseudoinverse of the matrix [11], that gives as solution of Eq. 15:

$$\bar{x}_i = \begin{cases} \bar{b}_i/\lambda_i, & \text{for all } i \in [1, N_Q - N_Q^{ker}] \\ \bar{x}_i = 0, & \text{for all } i \in [N_Q - N_Q^{ker} + 1, N_Q] \end{cases} \quad (16)$$

The final solution is then built as $\mathbf{x} = \mathbf{\Gamma} \cdot \bar{\mathbf{x}}$. It must be noted, however, that this solution is exact if and only if:

$$\bar{b}_i = 0 \text{ for all } i \in [N_Q - N_Q^{ker} + 1, N_Q] \quad (17)$$

250 or, in other words, if the source vector \mathbf{b} of Eq. 13 is orthogonal to the kernel of \mathbf{Q} . Otherwise, the obtained solution would be the most compatible one in terms of the Frobenius norm. In the case of Ampere's equation, \mathbf{b} is given by the currents flowing in the interior plasma and the kernel in \mathbf{Q} is originated by the curl operator. Therefore, the physical meaning of this restriction is that the currents must be divergence free.

255 There are however very important practical difficulties that prevent applying the eigenvalue method just described. In particular, it requires to find the full eigenvalue decomposition of \mathbf{Q} . Since, in most problems of interest, \mathbf{Q} has a dimension of the order of hundreds of thousands (in a typical stellarator case, $N_Q \sim 100,000 - 200,000$), this is a huge computational task. In addition, the eigenvector matrix $\mathbf{\Gamma}$ is a dense matrix, even if \mathbf{Q} is block-tridiagonal, which would also require an enormous amount of memory to store. Thus, a more appropriate way to find and apply the pseudoinverse approach is needed. We propose to use some well-established SVD techniques, in combination with a proper factorization of the \mathbf{Q} matrix, to achieve this. We discuss the details in the next subsection.

3.2.2. Proposed methodology

270 As discussed previously, the matrix \mathbf{Q} that results from discretizing Ampere's law following our prescription is symmetric and block-tridiagonal. By writing it in terms of the \mathbf{R} and \mathbf{M} matrices, the problem to solve reads:

$$\mathbf{Q} \cdot \mathbf{x} = (\mathbf{R}^\top \cdot \mathbf{M} \cdot \mathbf{R}) \cdot \mathbf{x} = \mathbf{b} \quad (18)$$

in which \mathbf{R} is block-tridiagonal and \mathbf{M} is block diagonal. \mathbf{M} is never singular. \mathbf{R} , on the other hand, is always singular since it is a discretization of the curl operator, that has a non-zero null space.

Let us now discuss in detail, step by step, how our proposal takes advantage of this representation to improve upon the eigenvalue decomposition we described earlier. The method goes as follows:

- The SVD decomposition [12] of the \mathbf{R} matrix is found first:

$$\mathbf{R} = \mathbf{U}_{\mathbf{R}} \cdot \mathbf{S}_{\mathbf{R}} \cdot \mathbf{V}_{\mathbf{R}}^{\top} \quad (19)$$

where $\mathbf{S}_{\mathbf{R}}$ is a diagonal matrix with real and positive values (the singular values), $\mathbf{U}_{\mathbf{R}}^{\top} \cdot \mathbf{U}_{\mathbf{R}} = I$ and $\mathbf{V}_{\mathbf{R}}^{\top} \cdot \mathbf{V}_{\mathbf{R}} = \mathbf{I}$. Since \mathbf{R} is a singular matrix, the last $N_{ker} > 0$ elements on the diagonal of $\mathbf{S}_{\mathbf{R}}$ will be zero, where we assume that the dimension of the kernel of \mathbf{R} is N_{ker} . Eq. 18 can then be rewritten as:

$$\mathbf{V}_{\mathbf{R}} \cdot \mathbf{S}_{\mathbf{R}} \cdot \mathbf{U}_{\mathbf{R}}^{\top} \cdot \mathbf{M} \cdot \mathbf{U}_{\mathbf{R}} \cdot \mathbf{S}_{\mathbf{R}} \cdot \mathbf{V}_{\mathbf{R}}^{\top} \cdot \mathbf{x} = \mathbf{b}. \quad (20)$$

observe that since $\mathbf{S}_{\mathbf{R}}$ is diagonal, $\mathbf{S}_{\mathbf{R}}^{\top} = \mathbf{S}_{\mathbf{R}}$.

- The system is then back-solved in several steps. However, instead of enumerating them in the order in which they are actually carried out, we discuss them first in a more pedagogical fashion.

- By defining the unknown vector \mathbf{x}_1 as:

$$\mathbf{x}_1 = \mathbf{S}_{\mathbf{R}} \cdot \mathbf{U}_{\mathbf{R}}^{\top} \cdot \mathbf{M} \cdot \mathbf{U}_{\mathbf{R}} \cdot \mathbf{S}_{\mathbf{R}} \cdot \mathbf{V}_{\mathbf{R}}^{\top} \cdot \mathbf{x} \quad (21)$$

Eq. 20 is rewritten as:

$$\mathbf{V}_{\mathbf{R}} \cdot \mathbf{x}_1 = \mathbf{b} \quad (22)$$

whose solution is,

$$\mathbf{x}_1 = \mathbf{V}_{\mathbf{R}}^{\top} \cdot \mathbf{b}. \quad (23)$$

- By defining the unknown vector vector \mathbf{x}_2 as:

$$\mathbf{x}_2 = \mathbf{U}_{\mathbf{R}}^{\top} \cdot \mathbf{M} \cdot \mathbf{U}_{\mathbf{R}} \cdot \mathbf{S}_{\mathbf{R}} \cdot \mathbf{V}_{\mathbf{R}}^{\top} \cdot \mathbf{x}, \quad (24)$$

Eq. 21 is rewritten as:

$$\mathbf{S}_{\mathbf{R}} \cdot \mathbf{x}_2 = \mathbf{x}_1 \quad (25)$$

where \mathbf{x}_1 is known from previous step. Since the last N_{ker} singular values present in the diagonal of $\mathbf{S}_{\mathbf{R}}$ are zero (the kernel of the curl operator), we can find the solution by using the pseudoinverse of $\mathbf{S}_{\mathbf{R}}$. Since $\mathbf{S}_{\mathbf{R}}$ is diagonal, the pseudoinverse is the diagonal matrix:

$$[\mathbf{S}_{\mathbf{R}}^+]_{ii} = \begin{cases} \frac{1}{[\mathbf{S}_{\mathbf{R}}]_{ii}}, & ii = 1, \dots, N_{tot} - N_{ker} \\ 0, & ii = N_{tot} - N_{ker}, \dots, N_{ker} \end{cases} \quad (26)$$

where the superindex + denotes pseudoinverse. In Fig. 5 we represent the magnitude of $[\mathbf{S}_{\mathbf{R}}]_{ii}$ associated with a particular (m, n) mode for

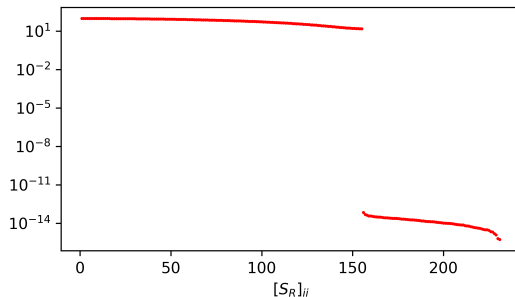


Figure 5: Singular values associated with a particular mode (m, n) mode for a realistic case.

a realistic case. These values corresponds with the singular values of the curl operator which, as has being previously pointed out, should have a kernel (any gradient quantity) of size one third of the total size of the matrix. Notice in the plot how no ambiguity exist in the cut-off when deciding which singular values corresponds with the kernel.

The solution of Eq. 25 is then given by,

$$\mathbf{x}_2 = \mathbf{S}_R^+ \cdot \mathbf{x}_1 + \mathbf{x}_2^{free} \quad (27)$$

where \mathbf{x}_2^{free} is any vector that is contained in the kernel of \mathbf{S}_R , with only its last N_{ker} components different from zero. In physical terms, these components span the (discrete version of the) subspace of gradients of any function on which the curl identically vanishes, that can be used as free parameters to ensure the satisfaction of any additional constrains that might be required by the problem at hand. We will use them later. A second comment worth making is that \mathbf{x}_2 provides an exact solution of Eq. 23 only if \mathbf{x}_1 is orthogonal to the kernel. As we mentioned earlier, this should in theory be the case if \mathbf{b} comes from divergenceless currents. However, due to numerical inaccuracies, it may not be in practice. The solution provided by \mathbf{x}_2 is then the nearest solution orthogonal to the kernel that can satisfy Eq. 23.

– By defining the unknown vector \mathbf{x}_3 as

$$\mathbf{x}_3 = \mathbf{S}_R \cdot \mathbf{V}_R^T \cdot \mathbf{x} \quad (28)$$

Eq. 24 is rewritten as:

$$\mathbf{U}_R^T \cdot \mathbf{M} \cdot \mathbf{U}_R \cdot \mathbf{x}_3 = \mathbf{x}_2. \quad (29)$$

We will ignore for now that the last N_{ker} components of x_2 are still undetermined. The system is easily inverted as:

$$\mathbf{x}_3 = \mathbf{U}_R^T \cdot \mathbf{M}^{-1} \cdot \mathbf{U}_R \cdot \mathbf{x}_2 \quad (30)$$

\mathbf{M}^{-1} can be found easily, thanks to its block-diagonal character, by inverting each block separately.

– By defining the unknown vector \mathbf{x}_4 as:

$$\mathbf{x}_4 = \mathbf{V}_R^\top \cdot \mathbf{x} \quad (31)$$

Eq. 28 is rewritten as:

$$\mathbf{S}_R \cdot \mathbf{x}_4 = \mathbf{x}_3 \quad (32)$$

325 This equation is inverted by invoking again the pseudoinverse of \mathbf{S}_R , yielding:

$$\mathbf{x}_4 = \mathbf{S}_R^+ \cdot \mathbf{x}_3 + \mathbf{x}_4^{free} \quad (33)$$

330 where \mathbf{x}_4^{free} is again an undetermined vector within the kernel of \mathbf{S}_R , thus having its first $N_{tot} - N_{ker}$ components equal to zero, and the final N_{ker} components unknown. This is similar to the \mathbf{x}_2^{free} vector that we left undefined previously. In fact, it is here that we will choose the value of \mathbf{x}_2^{free} by requiring that \mathbf{x}_3 be orthogonal to the kernel of \mathbf{S}_R , that ensures that \mathbf{x}_4 is an exact solution to Eq. 32. This is done by forcing the last N_{ker} rows of Eq. 32 to be zero:

$$0 = \bar{\mathbf{U}}_R^\top \cdot \mathbf{M}^{-1} \cdot \mathbf{U}_R \cdot \mathbf{x}_2 \quad (34)$$

335 where $\bar{\mathbf{U}}_R$ contains only the last N_{ker} columns of \mathbf{U}_R (i.e., the eigenvectors associated with the kernel; in fact, we will use the overbar in what follows to denote vector and matrices that are restricted to the kernel, thus having an N_{ker} linear dimension). Introducing Eq. 27 into Eq. 34 one gets to:

$$0 = \bar{\mathbf{U}}_R^\top \cdot \mathbf{M}^{-1} \cdot \mathbf{U}_R \cdot \left(\mathbf{S}_R^+ \cdot \mathbf{x}_1 + \mathbf{x}_2^{free} \right), \quad (35)$$

that can be recast as:

$$\bar{\mathbf{D}} \cdot \bar{\mathbf{x}}_2^{free} = \bar{\mathbf{c}} \quad (36)$$

340 for the in-kernel part of \mathbf{x}_2^{free} , that we will denote as $\bar{\mathbf{x}}_2^{free}$. The matrices and right-hand-side of Eq. 36 are defined as:

$$\bar{\mathbf{D}} = \bar{\mathbf{U}}_R^\top \cdot \mathbf{M}^{-1} \cdot \bar{\mathbf{U}}_R \quad (37)$$

and

$$\bar{\mathbf{c}} = -\bar{\mathbf{U}}_R^\top \cdot \mathbf{M}^{-1} \cdot \mathbf{U}_R \cdot \mathbf{S}_R^+ \cdot \mathbf{x}_1 \quad (38)$$

345 The direct inversion of $\bar{\mathbf{D}}$ is often too computationally intensive, since it is usually a dense matrix of linear size $N_{\bar{\mathbf{D}}} = N_{surf} \cdot N_{ker} \approx N_{surf} \cdot N_{modes}$ (for typical stellarator problems around $N_{\bar{\mathbf{D}}} \approx 50,000$). This is around one third of the original matrix system size, but still large for a direct solver. Instead we use an efficient iterative solver that will be described in the next subsection. Note also that the original system was singular and, thus, not invertible. However $\bar{\mathbf{D}}$ is not singular because we have eliminated the kernel contributions.

350

- The order of application of the different steps becomes now clear. First, one builds the SVD decomposition of \mathbf{R} . Then, $\bar{\mathbf{x}}_2^{free}$ is computed by solving Eq. 36. Eq. 30 is solved to obtain \mathbf{x}_3 and, subsequently, Eq. 33 is solved to get \mathbf{x}_4 . At this point, \mathbf{x}_4^{free} is set to zero (i.e., we choose the free gauge of the magnetic vector potential). Finally, the solution \mathbf{x} is obtained as:

$$\mathbf{x} = \mathbf{V}_{\mathbf{R}} \cdot \mathbf{x}_4. \quad (39)$$

In summary, the ordered steps of the full algorithm are:

1. Carry out the SVD decomposition $\mathbf{R} = \mathbf{U}_{\mathbf{R}} \cdot \mathbf{S}_{\mathbf{R}} \cdot \mathbf{V}_{\mathbf{R}}^T$
2. $\mathbf{x}_1 = \mathbf{V}_{\mathbf{R}}^T \cdot \mathbf{b}$
3. $\bar{\mathbf{c}} = -\bar{\mathbf{U}}_{\mathbf{R}}^T \cdot \mathbf{M}^{-1} \cdot \mathbf{U}_{\mathbf{R}} \cdot \mathbf{S}_{\mathbf{R}}^+ \cdot \mathbf{x}_1$
4. Solve $\bar{\mathbf{D}} \cdot \bar{\mathbf{x}}_2^{free} = \bar{\mathbf{c}}$ iteratively to get $\bar{\mathbf{x}}_2^{free}$
5. \mathbf{x}_2^{free} is trivially built as a null vector with the last N_{ker} components set by $\bar{\mathbf{x}}_2^{free}$.
6. $\mathbf{x}_2 = \mathbf{S}_{\mathbf{R}}^+ \cdot \mathbf{x}_1 + \mathbf{x}_2^{free}$
7. $\mathbf{x}_3 = \mathbf{U}_{\mathbf{R}}^T \cdot \mathbf{M}^{-1} \cdot \mathbf{U}_{\mathbf{R}} \cdot \mathbf{x}_2$
8. $\mathbf{x}_4 = \mathbf{S}_{\mathbf{R}}^+ \cdot \mathbf{x}_3$
9. $\mathbf{x} = \mathbf{V}_{\mathbf{R}} \cdot \mathbf{x}_4$

3.3. Computational cost of the algorithm

The procedure we have just described is computationally intensive, but it is very efficient. We analyze here qualitatively the computational cost of the whole procedure. First, we will consider the SVD decomposition of the matrix \mathbf{R} . Since \mathbf{R} is block diagonal, the decomposition consists of N_{modes} independent SVD decompositions, one for each of the sub-matrices of size $3 \cdot N_{surf}$. This process can be trivially parallelized. Secondly, \mathbf{x}_1 is obtained by applying $\mathbf{V}_{\mathbf{R}}^T$, that is a block diagonal matrix. The computational cost of multiplying by a block diagonal matrix is very low. Next, in order to obtain $\bar{\mathbf{x}}_2^{free}$ one needs to build the $\bar{\mathbf{c}}$ vector, which requires the application of several matrices over \mathbf{b} (see Eq. 38 and Eq. 23). Since all these matrices are also block diagonal, either in modes or in surfaces, this step is computationally inexpensive.

The most expensive part of the algorithm is the inversion of Eq. 36. In our implementation, this is done by solving it iteratively using the BiCGSTAB iterative algorithm [13] as implemented in the standard library PetSC [14]. A simple preconditioning is applied consisting on taking the diagonal part of $\bar{\mathbf{D}}$ associated with the poloidal/toroidal modes but keeping the radial structure. This is equivalent to solve a set of decoupled small systems of dimension the number of radial grid points, each of the systems associated with the mode (m, n) . For all the stellarator cases tested in Sec. 4, that contain about a hundred radial points and a few hundred of modes, 60 to 80 iterations are required to reach convergence up to machine precision. The number of iterations seems pretty insensitive to the resolution, as can be checked on table 1 for a typical

N_s	M	N	N_{modes}	N_{iter}
80	5	5	66	46
80	5	10	121	45
80	10	10	231	47
80	10	15	336	51
20	16	20	693	45
40	16	20	693	52
80	16	20	693	57

Table 1: Number of iterations as a function of the radial resolution N_s , poloidal modes M and toroidal modes N for a typical case.

case. At each iteration there are several applications of the preconditioner and the dense matrix $\bar{\mathbf{D}}$. The preconditioner is block diagonal, and the application of $\bar{\mathbf{D}}$ is done in three steps – applying first \mathbf{U}_R , then \mathbf{M}^{-1} and finally $\bar{\mathbf{U}}_R^T$, each of them implying the multiplication by a block diagonal matrix. The inversion of the block diagonal matrix \mathbf{M} can be done separately at each surface with a computational cost of N_{surf} systems of size $3N_{modes}$. Each block inversion is done by the LAPACK library [15] as a dense matrix inversion. The inversion is done before applying the iterative method and the result is stored in memory so that, at each iteration, the cost is just that of a matrix-vector product.

Once $\bar{\mathbf{x}}_2^{free}$ is found, the remaining steps are not computationally intensive. First, \mathbf{x}_3 is obtained applying Eq. 30; then, \mathbf{x}_4 is found using Eq. 33; finally, x is found from Eq. 39. All of these steps only require multiplications by block diagonal matrices previously obtained, which can be done very efficiently.

In summary, the algorithm is very efficient, being able to solve complicated geometries (our stellarator cases have a matrix \mathbf{Q} with a linear dimension $N_Q \sim 150,000$) in a few minutes using just one core of a processor, as will be shown in next section. Since the algorithm is trivially parallelizable, due to the block structure of the matrices, we expect it to be very competitive for much larger problems. This suggests that it might be very useful for plasma real time control in experiments that require very fast equilibrium evaluations under changing plasma conditions, or for many other problems outside the realm of fusion that may require a fast solution of Ampere’s equation for a given set of currents.

4. Illustration of the method. Comparison with integral and double-curl methods.

We have chosen the toroidal geometry of the W7-X stellarator [16], which started operation in 2016 in the Institute for Plasma Physics (IPP) at the Max-Planck-Institute in Greifswald (Germany), to illustrate the performance of our Ampere solver. W7-X is one of the biggest devices of its kind. It relies on a series of modular coils to confine a plasma within a volume with a toroidal topology, although extremely convoluted (see Fig. 1). Two different tests will be

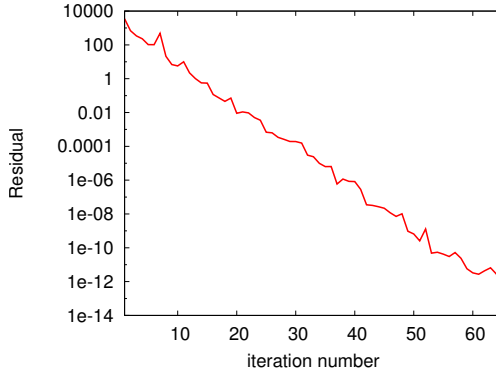


Figure 6: Relative error as a function of the BICG-STAB iteration for the case of vacuum magnetic field in log-lin scale.

carried out in this geometry. First, we will calculate the vacuum magnetic field created by the W7-X coil set inside of the vacuum vessel, assuming a certain value of the currents flowing in the coils. In the second test, the magnetic field created by the equilibrium currents flowing in the confined plasma for that W7-X configuration, as obtained by the VMEC code, will be calculated. Although the total magnetic field will naturally be the sum of both contributions, we will carry out the calculations separately since the former problem lacks any current in the calculation domain (the influence of the coils enters through the boundary conditions), whilst in the second one the magnetic field is completely generated by currents flowing inside of the calculation domain. In both cases, the results will be compared with the magnetic fields obtained through a direct integration of Biot-Savart's law.

4.1. Calculation of the vacuum magnetic field

In the absence of plasma, there are no currents inside of the domain, so that $\mathbf{b} = 0$ in Eq. 18. Since the W7-X coils are outside of our simulation domain, their influence appears via the boundary conditions that we impose at the inner and outer surfaces shown in Fig. 2. The value of the vector magnetic potential at those surfaces are provided by IPP's own MAG3D Biot-Savart code [17]. The spatial resolution we have used in this test is $N_{surf} = 79$, $M = 20$ and $N = 16$, which leads to $N_{modes} = 693$, and a linear dimension of the \mathbf{Q} matrix equal to $N_Q = 164,241$. Using a single core of a 2,4 GHz Intel Xeon E5-2630v3 processor, the amount of time required to solve Ampere's equation with the proposed method is about 255 seconds, of which 60 seconds are spent in the iterative determination of \mathbf{x}_2^{free} (i.e., in solving Eq. 36). As shown in Fig. 6, we needed 85 iterations of the BICG-STAB to reach the requested relative error of 10^{-15} . As shown in the left frame of Fig. 7, the figure-of-merit $|\nabla \times \mathbf{B} - \mathbf{J}| \approx 10^{-9}$ throughout the calculation domain. In addition, the

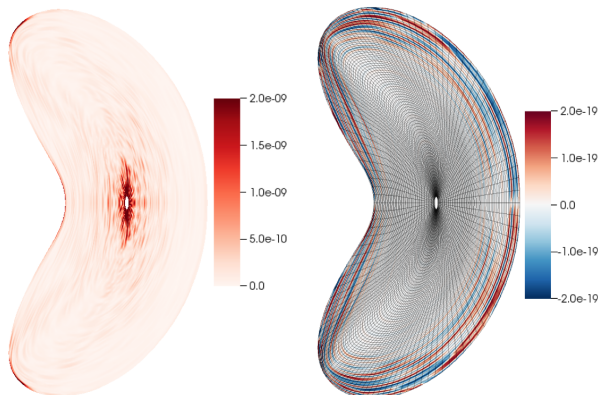


Figure 7: Left: Figure-of-merit $|\nabla \times \mathbf{B} - \mathbf{J}|$ at $\phi = 0$ toroidal plane. Since this is a vacuum case, \mathbf{J} is zero, and the residual of the solution reduces to $|\nabla \times \mathbf{B}|$. Right: Divergence of the solution, $\nabla \cdot \mathbf{B}$ at $\phi = 0$ plane. The cross section of the mesh is also represented, for illustration purposes. Both figures are normalized to B_{avg}/L , where B_{avg} is the average magnetic field and L the minor radius of the plasma.

divergence of the resulting magnetic field is of the order of machine precision all throughout the computational box (see right frame of Fig. 7), as previously advertised. A cross section of the mesh is also shown on the same figure.

We have also benchmarked our solution with that obtained from a direct integration of Biot-Savart law. In the left panel of Fig. 8, the magnitude of the vacuum magnetic field at the $\phi = 0$ toroidal cross-section can be seen, as computed with our algorithm. In the right panel of the same figure, the relative difference between our algorithm, that needs a little under ten minutes and IPP's MAG3D code, that needs about several hours to integrate the coil currents over the whole computational domain, showing a maximum discrepancy in the order of $\pm 0.5\%$.

4.2. Magnetic field created by plasma currents

Now we proceed to calculate the magnetic field generated by the equilibrium plasma currents, obtained by the VMEC code, for a plasma confined by the vacuum fields previously calculated. The analysed equilibrium corresponds to the so called standard configuration of the W7-X with a small bootstrap current on the axis and has a $\beta = 0.86\%$ (this figure-of-merit is defined as $\beta = \langle 2\mu_0 p / B^2 \rangle$). For this configuration, the island chain corresponding to a rotational transform $\iota = d\Psi/d\Phi = 5/5$, being Ψ and Φ the poloidal and toroidal magnetic fluxes respectively, is located just at the boundary of the plasma, ensuring a proper divertor operation [18]. We have used the same radial resolution and number of modes as in the vacuum calculation. The boundary conditions have now been obtained by integrating Biot-Savart law at the two boundaries considering only the plasma currents.

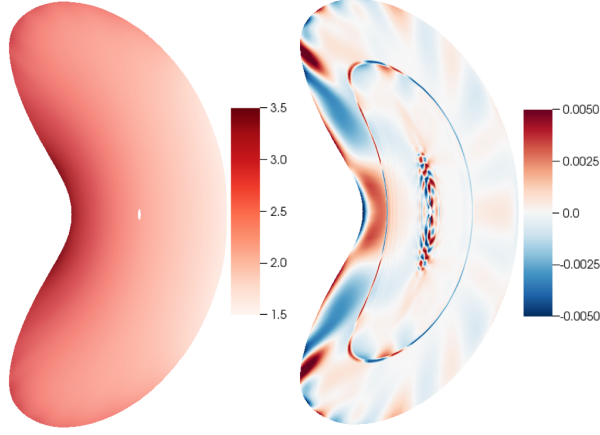


Figure 8: Left: Magnitude of the vacuum magnetic field (in Teslas) at $\phi = 0$ toroidal plane by using the algorithm proposed in this paper; Right: relative difference of the proposed algorithm and the integral solver solutions.

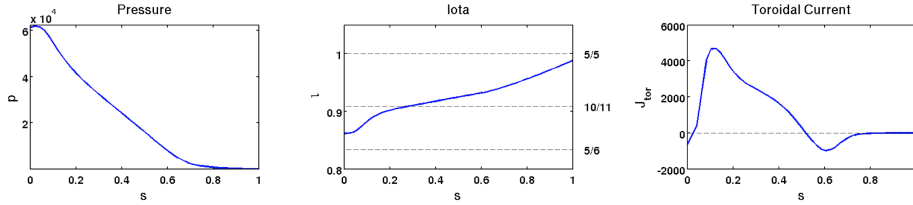


Figure 9: Profiles of pressure (left), iota (center) and toroidal current (right) corresponding to the studied equilibrium, all of them in VMEC units.

The solution of Ampere equation is obtained in 237 seconds (using the same hardware as the one in the vacuum case), with 57 seconds expend in the 69 iterations required by the solver. The magnitude of the plasma magnetic field is shown in the left panel of Fig. 10. As we did earlier, the right panel of the figure shows the relative difference between the magnetic fields obtained by our solver (less than five minutes in one core) and integrating Biot-Savart directly (ten hours in sixteen cores), which is in good agreement, showing discrepancies in the order of $\pm 5\%$. The larger discrepancy when compared with the vacuum case is probably due to the approximation as linear elements when integrating Biot-Savart on the plasma current. Even if the discrepancy is slightly larger, the vacuum magnetic field is near one hundred times larger than the plasma magnetic field, so the absolute discrepancy remains small.

As a final but important test, the vacuum and plasma magnetic field are added up and a Poincare plot is obtained by following the field lines. This test reveals if the topology of the magnetic structure is maintained (closed magnetic

surfaces and islands). As shown in Fig. 11, little difference can be found between the solutions, indicating that the topological structure is properly resolved.

4.3. Computational cost against the double-curl method

490 The proposed method, based on SVD decomposition, can be also compared with an efficient implementation of the double-curl (DC) method. An optimized version of the DC method has been implemented where the tridiagonal block system associated with the vector laplacian operator is solved with a preconditioned iterative BICG-STAB solver, equivalent to that used on the SVD method, 495 what is faster than solving the tridiagonal matrix system using a direct method. Thus, both versions of the solvers (the SVD method proposed here and the DC method) were implemented as efficient as we were able, and using the same libraries and programming language. This allows us to do a very confident comparison of the computational times. In the vacuum case, 255 seconds and 85 500 iterations were required for the SVD method while the DC method requires 328 seconds and 124 iterations. Thus, the SVD method is 1.29 times faster. This is a modest speedup but, if many problems are required to be solved under the same mesh geometry, the SVD method shows its real advantage. If we split the computational cost of the method as:

- 505 • *costA*: the computational time associated with intermediate matrices that can be calculated only once and reused for many plasma current cases, if the mesh geometry does not vary.
- *costB*: the computational time required for the inversion of the system, that will be specific to each plasma current case.

510 For the SVD vacuum case we found $costA_{SVD} = 195$ seconds and $costB_{SVD} = 60$ seconds. On the other hand, for the DC method we find $costA_{DC} = 142$ seconds and $costB_{DC} = 186$ seconds. This clearly indicates that, in the limit of many plasma current cases (where *costB* dominates the computational effort), the SVD method can be up to three times faster than the DC method. The 515 underlying reason is clear: as previously described, the dimensionality of the problem to be solved is reduced to one third in the case of the SVD method. The double-curl method is simpler to implement, but it cannot take advantage of the dimensionality reduction. This speedup can be useful in codes where the magnetic field is required to be calculated many times from the plasma 520 currents inside of an iterative loop, may be for time advance or for equilibrium convergence. A particular example can be the solution of the non-linear MHD equilibrium equation solved by SIESTA or PIES codes. There, the same grid (i.e. the same discretization) is used along the iterative process where, on each iteration, a linear problem has to be solved. Thus, the solver for obtaining the 525 magnetic field or the magnetic vector potential from the currents will be used many times, at least one for each iteration, with the same initialization matrices along the whole process, taking advantage of the 3x speedup.

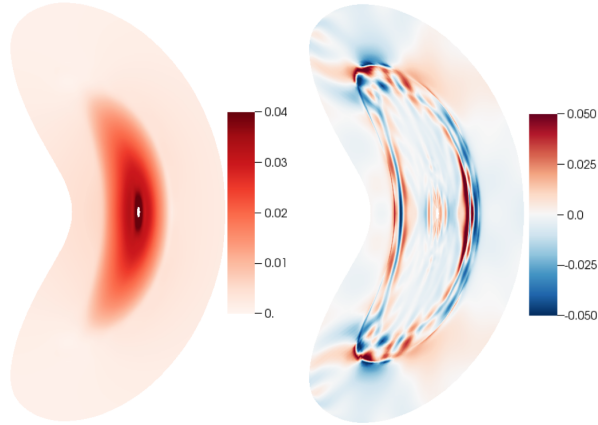


Figure 10: Left: Magnitude of the plasma magnetic field (in Teslas) at $\phi = 0$ toroidal plane by using the algorithm proposed in this paper; Right: relative difference of the proposed algorithm and the integral solver solutions.

5. Conclusions

In this paper, a novel technique has been presented that provides a very efficient way to calculate the magnetic field created by an arbitrary set of currents and coils in general toroidal geometry. The approach is differential instead of integral, and combines a dual finite-difference/spectral discretization and SVD techniques to deal with the sizeable non-trivial kernel of the matrix representation of Ampere's equation. It also deals with the magnetic potential, instead of the magnetic field, which guarantees a divergence-free solution down to machine precision. The resulting solver is between one and two orders of magnitude faster than any direct integral approaches, at least in terms of total computational time. Although it is true that the integral method could be parallelized with an almost perfect linear scaling, the differential approach proposed here also admits an easy parallelization, since the most computationally expensive part of the algorithm is the inverse of a block diagonal matrix, which could be done in parallel for the different blocks.

We have tested the performance of the algorithm by calculating the solution of Ampere's equation for a distribution of currents from a configuration of the W7-X stellarator, which requires a complex and convoluted toroidal domain with very good results. The solution is obtained much faster and it compares very well with the results obtained by the direct integral method. Also, it is computationally more advantageous than the double-curl method, up to three times faster when many applications of the solver are required under the same geometry. It must be noted, however, that although the code has been illustrated with an application within magnetic-confinement fusion, its possible uses are much broader. Indeed, it could certainly be used to provide magnetic fields in any application that requires a fast solution of Ampere's law in a toroidal

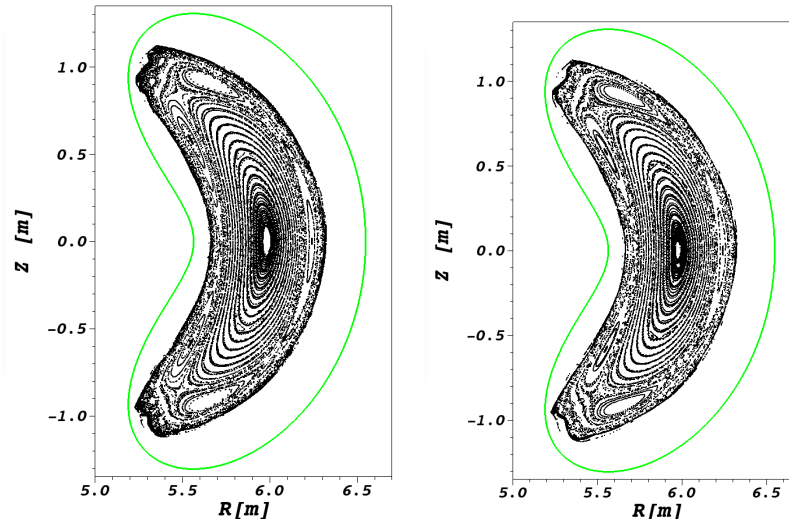


Figure 11: Poincaré plots corresponding to the proposed algorithm (left) and the integral method (right). In green the domain boundary, corresponding with the vacuum chamber.

domain, as might be the case in accelerator physics or other areas where toroidal
 555 fields are of interest.

6. Acknowledgments

This research has been sponsored in part by Spanish National Research
 Project No. ENE2015-68265. Use have also been made of *Uranus*, a supercom-
 560 puter cluster located at Universidad Carlos III de Madrid (Spain) funded jointly
 by EU FEDER funds and by the Spanish Government via the National Research
 Project Nos. UNC313- 4E-2361, ENE2009-12213-C03-03, ENE2012-33219, and
 ENE2012-31753.

Appendix A. Discretization details

In the radial direction, two staggered radial grids are introduced (see Fig.
 565 A.12). The first one, referred to as the *full mesh*, has nodes at the radial
 positions defined as $s_i^f = (i - 1) \cdot \Delta s$ where $i \in [1, N_s]$; the second one, known as
 the *half mesh*, has nodes at the intermediate radial positions: $s_i^h = \Delta s/2 + (i -$
 $2) \cdot \Delta s$ where $i \in [2, N_s]$. Δs is the spacing of the grid. The number of surfaces
 considered for a typical stellarator problem is $N_s \sim 100 - 200$.

570 The harmonics of all quantities of interest will be defined on either the full or
 half grid, depending on numerical considerations. The harmonics $R_{mn}(s)$ and
 $Z_{mn}(s)$ are both defined on the full grid. Regarding the remaining quantities

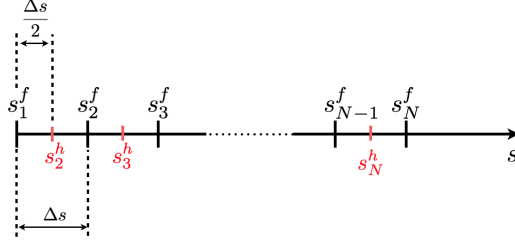


Figure A.12: full mesh nodes (black) and half mesh nodes (red) in the radial coordinate s .

of interest for solving Ampere's equation, their parities and the radial mesh on which they are defined are the covariant components of the magnetic vector potential:

$$\begin{aligned}
 A_s(s_i^f, \theta, \phi) &= \sum_{m=0}^M \sum_{n=-N}^N [A_s]_{mn}^i \sin(m\theta + nN_p\phi) & (\text{A.1}) \\
 A_\theta(s_i^f, \theta, \phi) &= \sum_{m=0}^M \sum_{n=-N}^N [A_\theta]_{mn}^i \cos(m\theta + nN_p\phi) \\
 A_\phi(s_i^f, \theta, \phi) &= \sum_{m=0}^M \sum_{n=-N}^N [A_\phi]_{mn}^i \cos(m\theta + nN_p\phi)
 \end{aligned}$$

and similarly discretized, the covariant components of the magnetic field:

$$B_s(s_i^h, \theta, \phi), \quad B_\theta(s_i^h, \theta, \phi), \quad B_\phi(s_i^h, \theta, \phi) \quad (\text{A.2})$$

the contravariant components of the magnetic field:

$$\sqrt{g}B^s(s_i^h, \theta, \phi), \quad \sqrt{g}B^\theta(s_i^h, \theta, \phi), \quad \sqrt{g}B^\phi(s_i^h, \theta, \phi) \quad (\text{A.3})$$

and the contravariant components of the current:

$$\sqrt{g}J^s(s_i^f, \theta, \phi), \quad \sqrt{g}J^\theta(s_i^f, \theta, \phi), \quad \sqrt{g}J^\phi(s_i^f, \theta, \phi) \quad (\text{A.4})$$

where \sqrt{g} is the Jacobian of the transformation defined by Eqs. 5.

References

- [1] J.P. Freidberg. *Ideal magnetohydrodynamics*, Plenum, New York (1987).
- [2] A.H. Reiman and H.J. Greenside. "Computation of zero- β three-dimensional equilibria with magnetic islands", *J. Comput. Phys.* **87**, 349 (1990)

- 585 [3] Y. Suzuki, Y. Nakamura, K. Kondo, T. Hayashi, S. Lloyd and H. Gardner. *MHD equilibrium of a low-shear helical axis Heliotron*, Plasma Phys. Contr. Fus. **45**, 971 (2003).
- [4] S.P. Hirshman, R. Sanchez and C.R. Cook. *SIESTA: A scalable iterative equilibrium solver for toroidal applications*, Phys. Plasmas **18**, 062504
590 (2011).
- [5] S.P. Hirshman and J.C. Whitson. *Steepest-descent method for three-dimensional magnetohydrodynamic equilibria*, Phys. Fluids **26** 3553 (1983)
- [6] V.D. Shafranov and L.E. Zakharov. *Use of the virtual-casing principle in calculating the containing magnetic field in toroidal plasma systems*, Nucl. Fusion **12** 599 (1972).
595
- [7] M. Drevlak, D. Monticello and A. Reiman. *PIES free boundary stellarator equilibria with improved initial conditions*, Nucl. Fusion **45** 731 (2005).
- [8] H. Peraza-Rodriguez, J.M. Reynolds-Barredo, R. Sanchez, J. Geiger, V. Tribaldos, S.P. Hirshman and M. Cianciosa. *Extension of the SIESTA MHD equilibrium code to free-boundary problems*, Phys. Plasmas **24**, 082516
600 (2017).
- [9] L. Chacón, G. Chen, *A curvilinear, fully implicit, conservative electromagnetic PIC algorithm in multiple dimensions* J. Comput. Phys. **316**, 578 (2016).
- 605 [10] K. Miyamoto. *Plasma Physics and Controlled Nuclear Fusion*. Springer Verlag, Heidelberg (2004).
- [11] S.L. Campbell and C.D. Meyer. *Generalized Inverses of Linear Transformations*. Dover Publishers, New York (1991).
- [12] G.H. Golub and C. Reinsch. *Singular Value Decomposition and Least Square Solutions*, Numerical Mathematics **14**, 403 (1970).
610
- [13] H.A. Van der Vorst, *BI-CGSTAB: Fast and Smoothly Converging Variant of the biconjugate gradient method for the Solution of Non-symmetric Linear Systems*, SIAM J. Sci. and Stat. Comput. **13** 631 (1992).
- [14] S. Balay, S. Abhyankar, M.F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V.Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. Curfman-McInnes, K.Rupp, B.F. Smith, S. Zampini, H. Zhang and Hong Zhang. PETSC web page: <http://www.mcs.anl.gov/petsc> (2016).
615
- [15] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, J., A. Greenbaum, S. Hammarling, A. McKenney and D., *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia (1999).
620

- [16] C. Beidler, G. Grieger, F. Herrnegger, E. Harmeyer, J. Kisslinger, W. Lotz, H. Maassberg, P. Merkel, J. Nührenberg, F. Rau, J. Sapper, F. Sardel, R. Scardovelli, A. Schluter and H. Wobig. *Physics and Engineering Design for Wendelstein 7-X*, Fusion Technology **17** 148 (1990).
625
- [17] J Svensson and A Werner. *Large Scale Bayesian Data Analysis for Nuclear Fusion Experiments*, Proceedings of the IEEE Workshop on Intelligent Signal Processing (2007).
- [18] J. Geiger, C.D. Beidler, M. Drevlak, H. Maaßberg, C. Nührenberg, Y. Suzuki, Y. Turkin, *Effects of Net Currents on the Magnetic Configuration of W7-X*, Contributions to Plasma Physics, **50** 770 (2010).
630