Journal of Computational Physics 228 (2009) 1562–1572

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

A fast technique applied to the analysis of Resistive Wall Modes with 3D conducting structures

Guglielmo Rubinacci [a], Salvatore Ventre [b], Fabio Villone [b,*], Yueqiang Liu [c]

[a] Ass. EURATOM/ENEA/CREATE, DIEL, Universitá degli Studi di Napoli, Federico II, Italy
[b] Ass. EURATOM/ENEA/CREATE, DAEIMI, Universitá degli Studi di Cassino, Via Di Biasio 43, 03043 Cassino, (FR), Italy
[c] EURATOM/UKAEA Fusion Association, Culham Science Centre, Abingdon, Oxon OX14 3DB, United Kingdom

ARTICLE INFO

Article history:
Received 14 March 2008
Received in revised form 7 October 2008
Accepted 31 October 2008
Available online 17 November 2008

PACS:
52.35.Py
28.52.Av
52.55.Fa

Keywords:
Resistive Wall Modes
Integral formulations
Fast methods

ABSTRACT

This paper illustrates the development of a "fast" technique for the analysis of Resistive Wall Modes (RWMs) in fusion devices with three-dimensional conducting structures, by means of the recently developed CarMa code. Thanks to its peculiar features, the computational cost scales almost linearly with the number of discrete unknowns. Some large scale problems are solved in configurations of interest for the International Thermonuclear Experimental Reactor (ITER).

1. Introduction

Plasma evolution in toroidal devices for magnetic confinement controlled fusion can be characterized, in many situations of interest, by the presence of unstable evolution modes [1,2]. The growth time of ideal, macroscopic instabilities is typically very short – of the order of microseconds for devices currently operating or under design. Such fast time scale makes any corrective control action practically ineffective.

However, surrounding the plasma with sufficiently close conducting structures, the plasma perturbation due to certain instabilities induces eddy currents which tend to counteract the instability itself. This stabilizing effect ends on the time scale over which eddy currents decay; this intuitively explains why the growth rate is slowed down to electromagnetic times [3,4], which are of the order of milliseconds or slower for devices currently operating or under design. Such modes are hence called Resistive Wall Modes (RWMs).

In this case, an active stabilization (i.e. by means of suitably fed external actuators) is indeed possible, and necessary if one wants to operate these devices on longer time scales. Currently, very good results have been achieved in the stabilization of the axisymmetric vertical instability that can occur in plasmas with an elongated (i.e. non-circular) cross-section [3]. In this case, the stabilization action is provided by a suitable axisymmetric vertical field. These achievements have been made

* Corresponding author. Tel.: +39 07762993674; fax: +39 0817683171.
E-mail address: villone@unicas.it (F. Villone).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.10.040

possible mainly by an accurate predictive modelling of the behaviour of the overall system [5], that allowed the design of effective control systems [6].

The situation is less assessed when looking at non-axisymmetric modes, tending to deform the plasma boundary helicoidally – the so-called kink instability. Such modes can become unstable if the plasma pressure exceeds a given threshold. For their stabilization, two are the techniques that are envisaged as maximally efficient: plasma rotation and magnetic control [7]. The first technique is based on the consideration that the injection of neutral particles in the plasma through Neutral Beam Injection systems, that are routinely used for plasma heating, can provide an angular momentum to the plasma, if the injection is not purely normal to it. Indeed, it has been experimentally demonstrated that if the rotation angular velocity is above a given threshold, RWMs can be stabilized. However, it is not clear yet if on next-generation devices, like the International Thermonuclear Experimental Reactor (ITER) this technique is viable [8]. Hence, the scientific community is paying considerable attention to magnetic control, which consists of using non-axisymmetric coils to produce a suitable magnetic field coupled with the unstable mode.

From this point of view, a lot of effort has been put in developing experimental control strategies [7,9]. However, in order to extrapolate current solutions to next-generation devices like ITER, a reliable modelling is needed. Several codes have been developed in the past for studying Magneto-Hydro-Dynamic (MHD) instabilities, e.g. MARS-F [10] and KINX [11]. However, typically such codes make strong simplifications on the geometry of the resistive wall (which is assumed to be axisymmetric) and of the coils used for control (which do not have a realistic geometry). This may prevent an accurate prediction of performances of future devices.

Hence, a detailed description of the three-dimensional features of the conducting structures surrounding plasma is needed [12,13]. Of course, the inclusion of fine geometrical details may easily lead to huge memory and computational time requirements. To this purpose, in this paper we present the development of "fast" techniques, for the treatment of large computational models, within the formulation behind the CarMa code, which is able to perform a 3D analysis of RWMs [13–15] and has also been validated with experimental results [16].

The numerical model of the passive conducting structures surrounding the plasma is based on an integral formulation of the eddy current problem (described in details in [17,18] and briefly reviewed below). Integral formulations require the storage of matrices of size scaling as N^2 (N being the number of discrete unknowns), and their inversion needs a computational cost of the order of N^3, if a direct solver is used.

In order to make such methods convenient for large-scale problems, some so-called fast techniques can be used. One possibility is based on the use of the Fast Fourier Transform, introducing equivalent sources on suitable regular grids. In this way, the products needed in the iterative procedure, providing the field associated to a given estimate of the sources, can be conveniently accelerated by means of a fast convolution product [19–21].

Other approaches are based on the consideration that the source can be characterized by few parameters, if the field point is far enough. This principle has been deeply exploited in the so called Fast Multipole Method (FMM) [22–25], in which the sources are expanded in spherical harmonics and the field computation is re-organized using a suitable oct-tree structure. Another approach in this frame is based on block Singular Value Decompositions (SVD) [26–28] of the matrix approximating the integral operator, mainly in the context of the solution of the full Maxwell equations. This SVD-based method is also at the basis of our approach for the treatment of the large computational models characterizing the passive structure surrounding the plasma. It has been remarked [26] that SVD can be faster and more memory efficient than FMM, although the preprocessing phase is more computationally expensive. This conclusion has been confirmed also using the 3D eddy current integral formulation at the basis the present approach [29].

In particular, our method is based on an efficient low-rank approximation of the sub-matrices representing "far" interactions. The efficiency of this scheme is greatly enhanced by re-organizing the field computation by means of a suitable adaptive oct-tree structure. This approach, widely exploited in the frame of the Fast Multipole Method [23], has been recently generalized by defining a multidimensional block cluster tree and storing the resulting approximation of the integral operator in the form of hierarchical matrices [30–33].

There are various methods for approximating the integral operator represented by a fully populated matrix L. The common basis of all these approaches is in two key properties: (a) it is possible to approximate the kernel function $g(\mathbf{r},\mathbf{r}')$ of the integral operator as $\widetilde{g}(\mathbf{r},\mathbf{r}') = \sum_k v_k(\mathbf{r}) w_k(\mathbf{r}')$; (b) this form has to converge rapidly to the original kernel function [32]. In this way, the block L_{XY} of the original matrix can be approximated within a prescribed error bound by the product of two low-rank matrices as $\widetilde{L}_{XY} = A_X B_Y^T$, being X and Y two suitably defined sub-domains of the domain of definition of the discretized integral operator. These properties are satisfied when the kernel function is the Green function $\frac{1}{|\mathbf{r}-\mathbf{r}'|}$ as in our case, and naturally lead to the multipole expansion. In this case, as in many others, it is possible to obtain suitable expansions of the kernel function using a Taylor expansion or interpolations.

The singular value decomposition gives, for a prescribed error bound, the minimal rank approximation $\widetilde{L}_{XY}$ of L_{XY} [34]. This property allows to compute a low rank approximation directly on the basis of the knowledge of the entries of the already discretized matrix block. Along this line, an interesting approach is introduced in [35] under the name *pseudoskeleton approximation* or *cross approximation*. It is proved there that whenever L_{XY} is approximated by a matrix of rank r with sufficient accuracy, then there exists a cross approximation with almost the same approximation quality. This approach leads to several algorithms, like the cross approximation with full pivoting with a quadratic complexity [32] and the adaptive cross approximation (ACA) with a linear complexity [32]. However, also the ACA and its improved version (ACA+) do not guarantee

a good approximation quality in all possible cases. Moreover, in some cases these approaches can even fail to provide the desired accuracy [32].

Conversely, in our specific case, the critical issues are an efficient storage of the system matrix and a fast computation of the matrix–vector products in many cases characterized by the same geometry of the conductors. For this reason, we prefer to resort to an algorithm which guarantees the minimal rank approximation, rather than choosing a faster pre-computation. Hence we compute the low rank approximations using a *QR* factorization which relies on the modified Gram–Schmidt orthogonalization procedure.

The paper is organized as follows. Section 2 briefly describes the CarMa code, while Section 3 describes how the fast technique can be successfully applied. Section 4 presents some examples of application and Section 5 draws the conclusions.

2. The CarMa code

The CarMa computational tool is described in details in [13,14]. Here we simply report the main points.

Let us consider a toroidal plasma circumvented by a three-dimensional conducting structure V_c, with H equipotential surface electrodes $\Sigma_1, \ldots, \Sigma_H$ fed with potentials $V_1, \ldots, V_H$.

The three-dimensional conducting structure is described using the integral formulation for the eddy currents problem presented in [17,18]. We introduce the magnetic vector potential $\mathbf{A}$ and the electric scalar potential V, such that the electric field $\mathbf{E}$ is given by

$$\mathbf{E} = -\frac{\partial \mathbf{A}}{\partial t} - \nabla V \tag{1}$$

By imposing Ohm's law on V_c in weak form, using (1) we have

$$\frac{\partial}{\partial t}\int_{V_c} \mathbf{A}\cdot\mathbf{w}dV + \int_{V_c} \nabla V\cdot\mathbf{w}dV + \int_{V_c} \eta\mathbf{J}\cdot\mathbf{w}dV = 0 \quad \forall \mathbf{w} \tag{2}$$

where $\mathbf{w}$ is a suitable weight function, η is the resistivity tensor and $\mathbf{J}$ is the current density in V_c. The magnetic vector potential at a point $\mathbf{P}$ is given by the Biot–Savart integral:

$$\mathbf{A}(\mathbf{P}) = \int_{V_c} \frac{\mathbf{J}(\mathbf{Q})}{|\mathbf{P}-\mathbf{Q}|} dV_Q + \mathbf{A}_{plasma}(\mathbf{P}) \tag{3}$$

where $\mathbf{A}_{plasma}$ is the contribution of the plasma current density.

In order to impose the solenoidality of the current density $\mathbf{J}$, we introduce the electric vector potential $\mathbf{T}$, such that $\mathbf{J} = \nabla \times \mathbf{T}$, with the two-component gauge. Giving a finite elements discretization of V_c, $\mathbf{T}$ is expanded in terms of edge elements $\mathbf{N}_k$, so that

$$\mathbf{J} = \sum_{k=1}^{N} I_k \nabla \times \mathbf{N}_k \tag{4}$$

The gauge is imposed by computing a tree–cotree decomposition of the mesh and retaining only the degrees of freedom (DoF) related to the edges belonging to the cotree. A special automatic treatment is used to take into account multiply connected domains and electrodes [36], giving rise to N DoF.

Adopting the Galerkin method and using (3) and (4), Eq. (2) becomes

$$\underline{\underline{L}}\frac{d\underline{I}}{dt} + \underline{\underline{R}}\underline{I} + \frac{d\underline{U}}{dt} = \underline{\underline{F}}\underline{V} \tag{5}$$

where $\underline{I} = \{I_k\}_{k=1,\ldots,N}$, $\underline{V} = \{V_k\}_{k=1,\ldots,H}$ and

$$L_{ij} = \int_{V_c}\int_{V_c} \frac{\nabla\times\mathbf{N}_i(\mathbf{Q})\cdot\nabla\times\mathbf{N}_j(\mathbf{P})}{|\mathbf{P}-\mathbf{Q}|} dV_Q dV_P \tag{6}$$

$$R_{ij} = \int_{V_c} \nabla\times\mathbf{N}_i\cdot\eta\nabla\times\mathbf{N}_j dV \tag{7}$$

$$F_{ij} = \int_{\Sigma_j} \nabla\times\mathbf{N}_i\cdot\hat{\mathbf{n}}_\Sigma dS \tag{8}$$

$$U_i = \int_{V_c} \nabla\times\mathbf{N}_i\cdot\mathbf{A}_{plasma} dV \tag{9}$$

Here, the quantity $\hat{\mathbf{n}}_\Sigma$ is the normal to the surface electrode Σ_j, pointing outwards with respect to the conducting domain.

Now, we consider a surface S with normal $\hat{\mathbf{n}}$, in between the plasma and the conducting structures. Inside S we solve the linearized single-fluid MHD equations:

$$\begin{cases} 0 = \nabla p + \mathbf{j_p} \times \mathbf{B} + \mathbf{J_p} \times \mathbf{b} \\ p = -\xi \cdot \nabla P - \Gamma_{sh} P \nabla \cdot \xi \\ \nabla \times \mathbf{b} = \mu_0 \mathbf{j_p} \\ \mathbf{b} = \nabla \times (\xi \times \mathbf{B}) \\ \nabla \cdot \mathbf{b} = 0 \end{cases} \quad \text{inside } S \tag{10}$$

$$\mathbf{b} \cdot \hat{\mathbf{n}}|_S = h \tag{11}$$

In (10), uppercase letters represent the reference equilibrium values and lowercase letters stand for first-order perturbations; $\mathbf{b}$ is the magnetic flux density, $\mathbf{j_p}$ is the plasma current density, ξ is the plasma displacement, p is the plasma pressure and Γ_{sh} is the specific heat ratio.

We notice that in (10) we have neglected plasma mass, which is an excellent approximation on typical RWM time scales; we have also assumed that no plasma rotation takes place.

In (10) we are also considering a single toroidal Fourier harmonic with mode number n, i.e. we are assuming that the spatial dependence upon the toroidal angle φ of all plasma quantities is as $e^{jn\varphi}$. This is an exact assumption when the plasma equilibrium configuration is axisymmetric.

Eq. (10) are solved using a Fourier decomposition along the poloidal angle and a Galerkin-based finite elements method on a staggered grid along the radial direction. The problem is formulated in a curvilinear coordinate system related to the equilibrium magnetic field.

We now decompose the quantity h, defined in (11), in terms of M suitable spatial basis functions w_i on S:

$$h = \sum_{i=1}^{M} h_i w_i \tag{12}$$

One could use, for instance, M poloidal Fourier harmonics.

We solve (10) M times, imposing that the quantity h is obtained from (12) with $h_i = 1,\ h_{j \neq i} = 0$. Doing so, we compute the plasma response, which is static thanks to the fact that we have neglected plasma mass. For each of these M solutions, we evaluate the quantity (9), hence giving rise to a $N \times M$ matrix $\underline{S}$. Typically, it results $M \ll N$ due to the fact that S is a (regular) surface; moreover, M does not depend on N. The expression of the elements of matrix $\underline{S}$ is given in [14].

In addition, we compute the $M \times N$ matrix $\underline{Q}$, which provides, for each discrete degree of freedom I_k, the projection on the basis functions defined in (12) of the normal component of its magnetic flux density. Particular care is given to the fact that the solution of (10) requires the imposition of the total magnetic flux density, while the matrix $\underline{Q}$ provides only the contribution of the conducting structures, i.e. not considering the plasma [13].

To sum up, we have that

$$\underline{U} = \underline{\underline{S}}\,\underline{\underline{Q}}\,\underline{I} \tag{13}$$

and hence (5) becomes

$$\underline{\underline{L}}^* \frac{d\underline{I}}{dt} + \underline{\underline{R}}\,\underline{I} = \underline{\underline{F}}\,\underline{V} \tag{14}$$

where the $N \times N$ fully populated matrix $\underline{\underline{L}}^*$ is given by

$$\underline{\underline{L}}^* = \underline{\underline{L}} + \underline{\underline{S}}\,\underline{\underline{Q}} \tag{15}$$

If several different toroidal mode numbers n are to be considered, the related additional terms can be summed up as

$$\underline{\underline{L}}^* = \underline{\underline{L}} + \sum_n \underline{\underline{S}}_n \underline{\underline{Q}}_n \tag{16}$$

where each $\underline{\underline{S}}_n$ and $\underline{\underline{Q}}_n$ have the same dimensions as before.

3. The fast technique

One of the most important information needed for the analysis of RWMs is the value of the unstable eigenvalue of the time-domain system defined by (14) (the growth rate of the instability) and the corresponding eigenvector (the unstable mode). In order to find such quantities, we use the inverse iteration scheme, which is widely used in MHD stability computations [37,38]. Indeed, this algorithm can compute individual eigenvalues – e.g. the unstable ones, as mentioned before – taking advantage of a priori approximate knowledge of their values for a fast convergence. Both these features make the algorithm very convenient in the present context.

Given initial guesses γ_0 and $\underline{I}_0$, at step k we estimate the eigenvalue γ_k and the eigenvector $\underline{I}_k$ by solving

$$\left(\gamma_0 \underline{\underline{L}}^* + \underline{\underline{R}}\right) \underline{I}_k = -\underline{\underline{L}}^* \hat{\underline{I}}_{k-1} \tag{17}$$

$$\gamma_k = \gamma_0 + \frac{\underline{I}_k^T \hat{\underline{I}}_{k-1}}{\underline{I}_k^T \underline{I}_k} \tag{18}$$

where $\hat{\underline{I}}_{k-1} = \underline{I}_{k-1}/\|\underline{I}_{k-1}\|$. The algorithm stops when the norm of the difference between two consecutive estimates γ_k and γ_{k-1} is below a given threshold ϵ_{invit}.

When considering the whole passive structure surrounding the plasma with a detailed description of 3D features, the number of unknowns N could easily become so large that the storage and inversion of the fully-populated matrix of system (17) can become critical. In particular, the solution of the system (17) by a direct method, requiring $O(N^3)$ operations, can be unpractical.

One can resort to iterative methods, such as the preconditioned GMRES method [39]. Such algorithm minimizes in n_{GMRES} iterations (within a given tolerance ϵ_{GMRES}) the norm of the residual of the linear system to be inverted. To do so, the repetitive computation of the matrix–vector product $(\gamma_0 \underline{\underline{L}}^* + \underline{\underline{R}})\underline{I}_k$ is required. Hence, it is mandatory to develop an efficient approach to reduce the computational cost of such product, which would require $O(N^2)$ operations with the standard matrix–vector product formula.

First of all, we notice that the computation of $\underline{\underline{R}}\underline{I}_k$ is not an issue, since the matrix $\underline{\underline{R}}$ is sparse. Apart from trivial scalar multiplicative factors, we are left with the computation of

$$\underline{\underline{L}}^* \underline{I}_k = \underline{\underline{L}}\underline{I}_k + \sum_n \underline{\underline{S}}_n \underline{\underline{Q}}_n \underline{I}_k \tag{19}$$

where we can compute $\underline{q} = \underline{\underline{S}}_n \underline{\underline{Q}}_n \underline{I}_k$ as

$$\underline{p} = \underline{\underline{Q}}_n \underline{I}_k, \quad \underline{q} = \underline{\underline{S}}_n \underline{p} \tag{20}$$

The evaluation of (20) requires $2MN$ operations, i.e. a computational cost scaling linearly with the number of unknowns N.

In order to accelerate the computation of $\underline{\underline{L}}\underline{I}_k$ in (19), we resort to a SVD-based method [25–28]. The main idea is that the magnetic field produced by a set of sources grouped in a given region V_S, when evaluated in a different region V_E, can be described through a linear operator having a rank r decreasing as the relative separation between V_S and V_E is increased.

In other words, a submatrix $\underline{\underline{B}}$ of the matrix $\underline{\underline{L}}$, representing the interactions between sources in different regions, can be described by means of a "low" rank QR factorization. If $\underline{\underline{B}}$ is an $m_1 \times m_2$ matrix, then we can write $\underline{\underline{B}} \cong \underline{\underline{Q}}_B \underline{\underline{R}}_B$ where $\underline{\underline{Q}}_B$ is $m_1 \times r$ and $\underline{\underline{R}}_B$ is $r \times m_2$, with $r < m_1, m_2$ [26].

To improve the computational efficiency, in our procedure the QR factorization is obtained by using the Modified Gram–Schmidt (MGS) algorithm [27]. In our approach, we use the "full" MGS, which requires the pre-computation of the whole matrix $\underline{\underline{L}}$, while [27] proposes the "dual" MGS, which does not have this drawback. However, in this paper we choose full MGS, regardless the fact that the computational time for the matrix assembly scales quadratically with the number of unknowns N. Indeed, as we will show in Section 4 for typical cases of our interest, first of all this time is overwhelmed by other pre-computations, which scale linearly with N. Secondly, the full MGS approach used here provides a higher compression gain for a given approximation error.

Specifically, the matrix $\underline{\underline{Q}}_B$ consists of an orthonormal set of vectors spanning, approximately, the range of the original matrix block $\underline{\underline{B}}$. Moreover, dominant column vectors of the original block $\underline{\underline{B}}$ are incorporated in order of their relative importance, by using the strategy reported as "conventional MGS QR-Factorization algorithm with partial pivoting" in [27]. An error threshold ϵ_{tol} may be used to stop the procedure in order to obtain the smallest rank r for a given approximation error.

The original MGS QR-Factorization algorithm has been further improved by using a strategy adopted by the adaptive multilevel approach used to reduce the computational cost of the FMM [22]. In particular, in previous approaches as [26], a binary-tree splits and merges is proposed, with an ordering of the unknowns such that close-in-space points correspond to close indices in the ordering; the matrix compression is achieved by expressing each relevant submatrix by its low rank representation.

Conversely, in our approach we superimpose to the finite element mesh a regular grid consisting of cubic cells. Then, we apply the MGS QR-Factorization method to the contributions to the $\underline{\underline{L}}$ matrix arising from well-separated elements, i.e. elements that belong to non-adjacent cells. Two cells are non-adjacent if there is at least one cell between the two. Then, the reference cell and all its adjacent cells are again subdivided and the procedure is iteratively applied. The subdivision stops when the number of elements in a given cell is smaller than a prescribed value s. The interactions among the elements in this cell and its adjacent cells are computed with no approximations.

The data structure is adaptively built, following an approach somehow similar to the one described in [23]. We start with a computational box B_0 including the whole mesh. Then we recursively subdivide a box, that is consequently a *parent box*, into its eight children, if it contains more than s elements. Otherwise, this box is considered as childless and, accordingly, it is not subdivided further. The set of all non-empty boxes at level l is denoted by B_l. The maximum number of levels l_{max} is hence implicitly defined by the choice of s.

For each box where the magnetic field has to be computed, we define the following interaction lists, in the stream of [23]:

- *List 1* of a childless box b, denoted by $L_1(b)$: the set made of b and all the childless boxes adjacent to b.
- *List 2* of a box b, denoted by $L_2(b)$: the set of all boxes at the same level of b that are well separated from b, i.e not adjacent to b.
- *List 3* of a childless box b, denoted by $L_3(b)$: the set of all descendants of b's colleagues that are not adjacent to b, but whose parent boxes are adjacent to b.
- *List 4* of a box b, denoted by $L_4(b)$: set of boxes c such that $b \in L_3(c)$.

Define the box B_0 containing all the mesh.

Choose:
- the minimum number s of elements contained in a box;
- the relative precision ε_{tol} for the SVD low rank approximation of each matrix block;
- the maximum number of elements m in a given box.

Build the tree structure and create the four lists for each box.

Set the initial level l_0 such as there are no boxes containing more than m elements.

```
do for levels l = l_0 : l_max
    do for each box b of level l
        do for each box c ∈ L_2(b)
            Compute and store the low rank approximation of the block matrix A_{b,c}
        end do
    end do
end do

do for each childless box b
    do for each box c ∈ L_3(b)
        Compute and store the low rank approximation of the block matrix A_{b,c}
    end do
end do

do for each childless box b
    do for each box c ∈ L_1(b) ∪ L_4(b)
        Compute by direct integration and store the block matrix A_{b,c}
    end do
end do
```

Fig. 1. Fast algorithm.

With these definitions, the adaptive algorithm is reported in Fig. 1. Here, for a given set of elements belonging to the boxes b and c, we denote with $A_{b,c}$ the submatrix of the matrix $\underline{\underline{L}}$, made of the terms L_{ij} where i (resp. j) is the generic discrete degree of freedom associated to any element belonging to box b (resp. c). The computational cost of such algorithm for the calculation of $\underline{\underline{L}}\underline{I}_k$ in (19) can be expected to be slightly worse than linear with the number N of unknowns, as reported in [26,27] for similar approaches.

To sum up, the use of the inverse iteration scheme, the exploitation of the features of the matrices involved and the combination of the MGS QR-Factorization method and of the multilevel adaptive procedure, leads to a computational cost for the evaluation of one eigenvalue of the overall system (14) that increases almost linearly with N, provided that the number of iterations of the GMRES and of the inverse iteration scheme do not depend on N.

4. Applications

The first example is a benchmark, aimed at assessing the reliability of the procedures and the actual scaling of computational times.

A plasma with a circular cross-section is considered, surrounded by a thin 8 mm copper shell with a circular concentric section of major radius 2 m and a minor radius 0.5049 m. The shell has a poloidal and a toroidal gap of around 2 cm of width.

A plasma configuration which is unstable to both $n = 1$ and $n = 3$ modes is considered. The standard CarMa code has been applied to such configuration on a rather coarse mesh, finding two pairs of unstable eigenvalues – one pair corresponding to $n = 1$ modes and the other pair corresponding to $n = 3$ modes, the latter being the most unstable. In particular, we focus our fast technique on finding one of the $n = 1$ unstable eigenvalues, whose estimated eigenvalue is around 15.4 s^{-1}.

We then have produced a number of discretizations, representing exactly the same geometry but with an increasing number of elements and DoF. Evidently, the estimated growth rate should remain practically unchanged.

In Fig. 2(a), for a coarse discretization, we show the boxes belonging to the various lists defined in Section 3. Here, b stands for the reference source childless box, while the other boxes are labelled with the number corresponding to the list they belong to. This figure applies to a given vertical position; obviously, a different box pattern arises at other positions.

In Fig. 2(b) we report the dimensions of the submatrices describing the interactions among the various boxes involved: the number reported inside the box b is the quantity m_1 defined in Section 3; the first number in all the other boxes is the rank r of the interaction, while the second number is the quantity m_2. For instance, the interaction of box b with the box in the upper right corner is described by a 62×154 matrix, but the fast technique uses the cascade of two matrices of dimensions 62×19 and 19×154, respectively. Hence, the speed-up of the fast technique can be quantified by the quantity

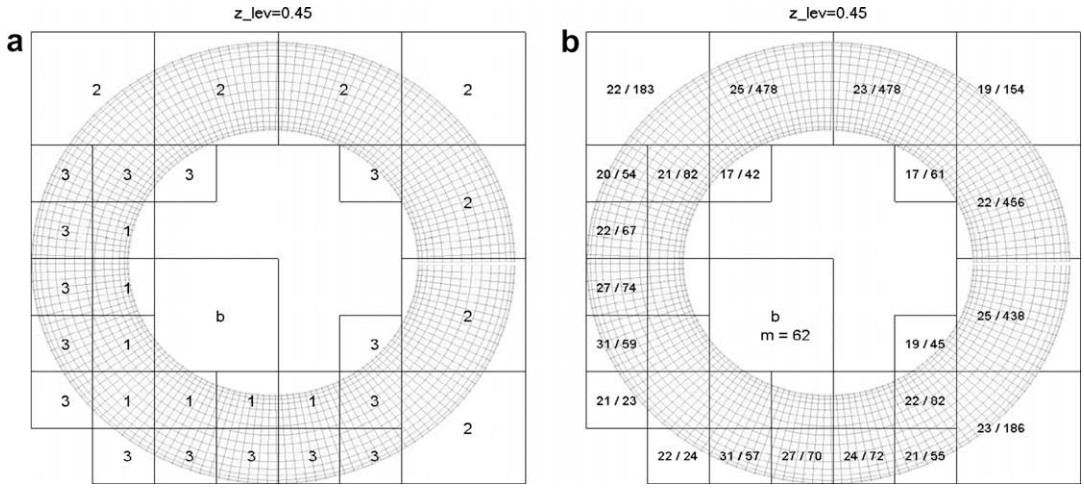

Fig. 2. Fast algorithm: (a) definition of boxes and (b) rank of interactions.

$g = (m_1 m_2)/[(m_1 + m_2)r]$, which is reported in Fig. 3 for the finest discretization used. In this case, the gain g can go up to a factor above 60, for the most expensive interactions (i.e. corresponding to a higher $m_1 m_2$). The overall gain for the whole matrix–vector product (i.e. considering all the interactions among all the boxes) is in this case 13.7.

In Table 1 we report the CPU time needed for the computation of the desired selected eigenvalue (around 15.4 s^{-1}), using a starting guess of 50 s^{-1} (with a constant starting guess for the eigenvector) in the inverse iteration scheme. This time accounts for the execution of the inverse iteration scheme, without including the time needed for pre-processing (i.e. operations that can be done once for all for a given geometry and need not be repeated for each eigenvalue), which is instead reported in the following. Three sets of parameters were used, as Table 2 illustrates. The main idea is to fix a desired accuracy ϵ_{invit} for the calculation of the eigenvalue and then guarantee that $\epsilon_{tol} < \epsilon_{GMRES} < \epsilon_{invit}$, since the matrix–vector product should have the best accuracy.

The preconditioner used in these cases is the submatrix of the matrix to be inverted, made by all the entries computed by direct integration, as detailed in Fig. 1. The overall gain for the whole matrix–vector product ranges from 2.16 to 7.11 when using parameter set #3 of Table 2.

The results show clearly that the growth rate estimate is satisfactory, being practically unchanged and very close to the preliminary estimate. We also notice that, for a given discretization, the growth rate estimate is practically unchanged with all the parameters sets of Table 2 – the variations are below 0.1%. This clearly indicates that the level of approximation of the matrix, although in principle can affect the eigenvalues, practically has a very small effect. Fig. 6 shows the second finest mesh used (with 38,019 unknowns), together with a pictorial representation of the eigenmode found by the inverse iteration scheme.

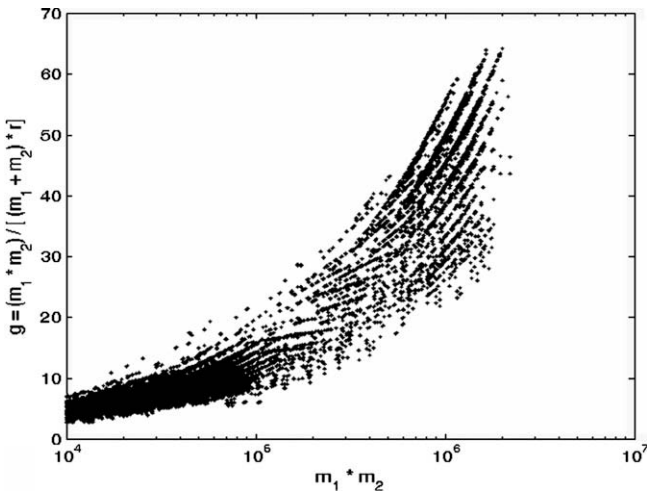

Fig. 3. Speed-up of the fast technique.

Table 1
Computational times needed for the evaluation of one unstable growth rate.

DoF	γ [s^{-1}]	Time [s] (Set#1)	Time [s] (Set#2)	Time [s] (Set#3)
9417	15.43	95.2	48.3	33.3
18849	15.42	204.0	107.5	74.8
38019	15.41	574.3	288.5	198.7
75891	15.41	NA	NA	461.3

Table 2
Parameters used.

Parameter	Set#1	Set#2	Set#3
s	70	70	70
ϵ_{tol}	1.e−6	1.e−5	1.e−4
ϵ_{GMRES}	1.e−5	1.e−4	1.e−3
ϵ_{invit}	1.e−4	1.e−3	1.e−2

The computational time scales almost linearly with the number of DoF, as expected. The deviation from the linear trend is partly due additional iterations n_{GMRES} of the GMRES algorithm required to get to the desired accuracy with increasing N. Conversely, the number of iterations of the inverse iteration scheme is the same in all cases. The calculations have been performed on a Core Duo 2.40 GHz 8 GB RAM computer.

We notice that for the finest case only the third set of parameters could be used, since the first two gave rise to a poor compression of the matrix and hence to a memory overflow. For the same reason the standard CarMa code could be run only on the first two meshes.

We have compared the performance of the approach proposed in this paper (full MGS) with similar techniques reported in literature and described in the introduction: full pivot cross approximation, adaptive cross approximation (ACA) and its improvement ACA+, dual-MGS. We consider the interaction between the two blocks b_1 and b_2 depicted in Fig. 4(a); similar considerations hold for many of all the possible interacting blocks. For given values of the relative error made by the aforementioned methods in approximating the part of the $\underline{\underline{L}}$ matrix related to these blocks, we compute the compression gain $g = (m_1 m_2)/[(m_1 + m_2)r]$ defined above. The results reported in Fig. 4(b) show that, for this particular choice of interacting blocks, the full MGS method used in this paper generally provides the highest gain for a given approximation error. This is achieved at the expense of a higher computational time needed in the computation of the full matrix $\underline{\underline{L}}$, scaling quadratically with the number of unknowns. In fact, we show in Fig. 5 the times (t_L, t_Q) needed for the pre-computation of the matrices $\underline{\underline{L}}$ and $\underline{\underline{Q}}$, respectively. The dashed line is an extrapolation. Evidently, for all cases of interest of this paper, the pre-computation time is dominated by t_Q, which scales linearly with the number of unknowns. The quadratically scaling time t_L will become a limiting factor only for problem of dimensions above 3×10^5 discrete unknowns, i.e. substantially greater than those presently solved. In these cases, we should resort to different methods, such as ACA+.

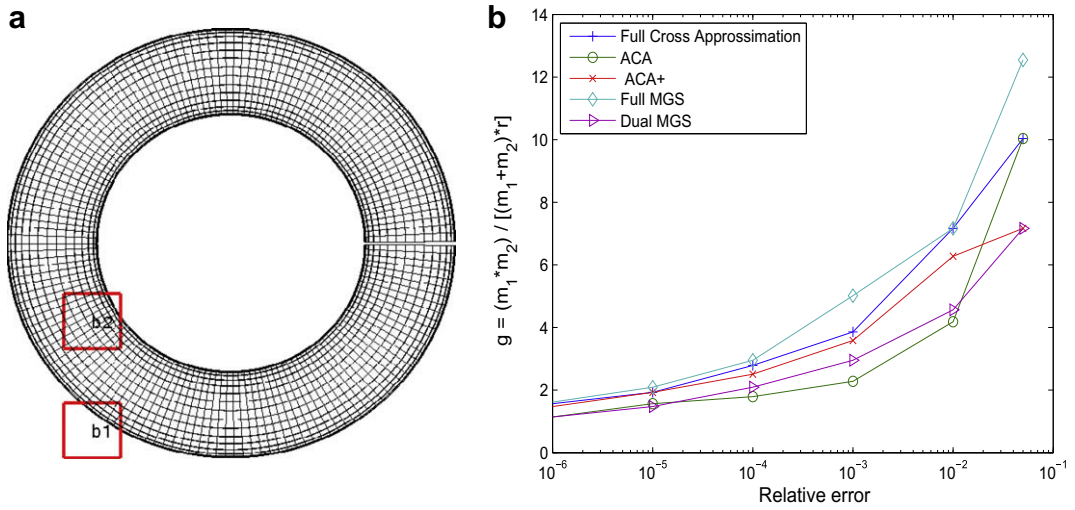

Fig. 4. Comparison of various methods: (a) interacting blocks, (b) compression gain as a function of relative error.

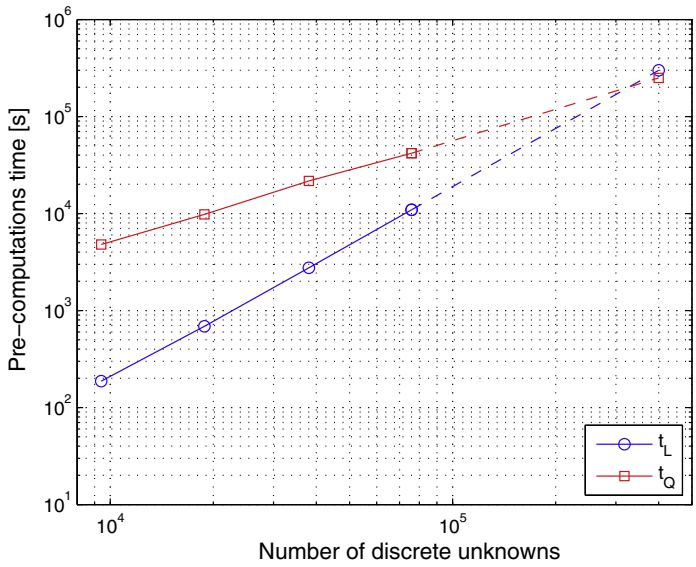

Fig. 5. Pre-computation times for the calculation of the matrices $\underline{\underline{L}}$ (t_L) and $\underline{\underline{Q}}$ (t_Q).

We would like to stress that the times reported in Fig. 5 refer to pre-processing, i.e. to calculations that must not be repeated when different eigenvalues must be estimated for the same discretization, even changing the plasma response. This is particularly important in our context, since it often happens that the eigenvalues of interest are a few tens [16] or even more, and that several different plasma responses must be considered on the same geometry [13].

Now we present another example of application to the geometry of ITER. The equilibrium configuration has two $n = 1$ almost coinciding unstable eigenvalues, corresponding to two unstable modes of evolution which are practically identical apart from a rotation of $\pi/2$ in the toroidal direction [13].

In order to quantify the 3D effects, three different meshes have been considered. The first mesh (Mesh#1) has an axisymmetric vacuum vessel (although represented with a 3D finite elements discretization), while the second one (Mesh#2) shows a 3D vacuum vessel with ports and tubular extension. The third one (Mesh#3) has also Toroidal Field (TF) coil cases, which are described as volumetric (thick) conductors, which is a unique feature of our computational tool as compared, e.g to [12]. Mesh#3 is presented in Fig. 7; the number N of discrete DoF is equal to 58,385 in this case. The parameters used for the fast procedure are those of set #3 in Table 2. The standard CarMa code cannot be applied to such mesh due to memory overflow; the overall gain of the fast technique is 5.0.

Table 3 reports the results for one ITER equilibrium with $\beta_N = 3.02$ (this parameter quantifies the plasma pressure). First of all, we notice that the fast procedure provides a result very close to the standard CarMa code when used on the same mesh

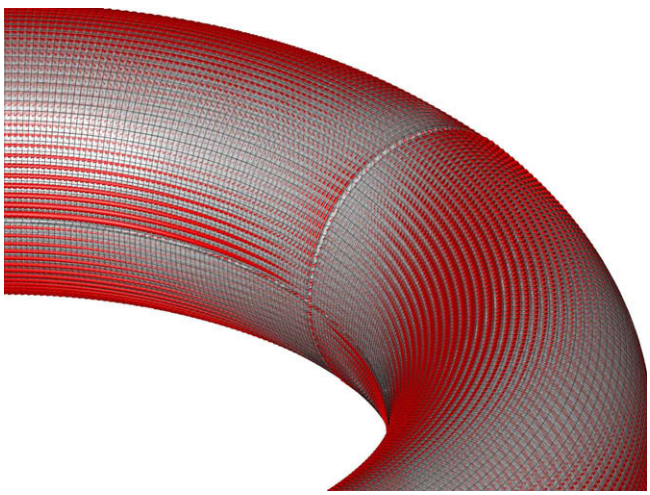

Fig. 6. Detail of the three-dimensional current density pattern corresponding to the unstable eigenvalue.

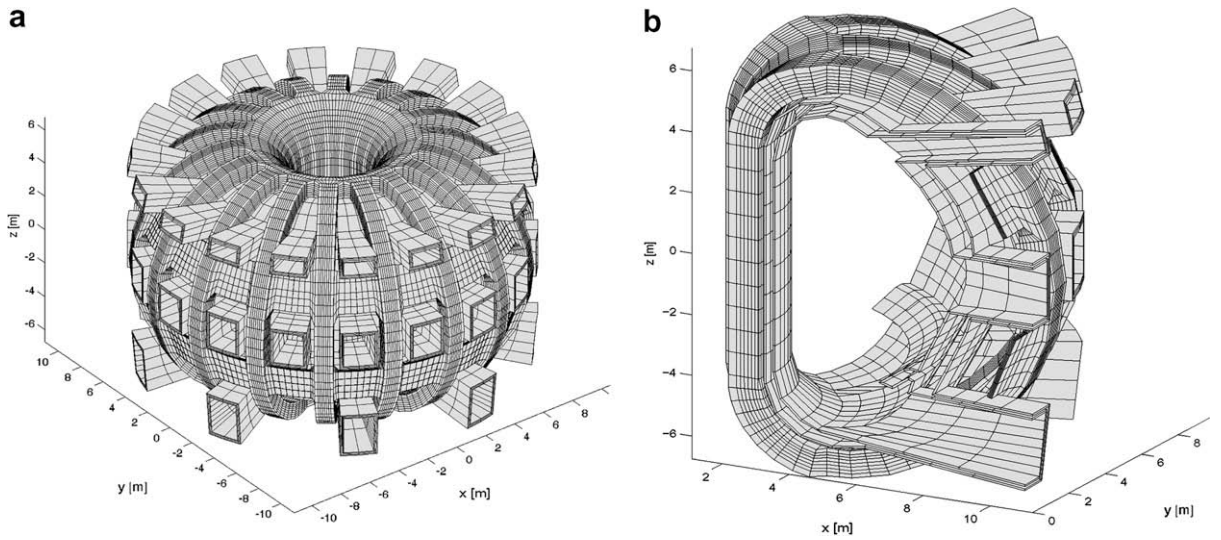

Fig. 7. Mesh of ITER: (a) overall view and (b) cutaway details.

Table 3
Estimated growth rate for ITER.

	Mesh#1 (CarMa)	Mesh#2 (CarMa)	Mesh#2 (fast)	Mesh#3 (fast)
γ [s^{-1}]	11.44	13.83	13.89	13.68

– this validates our approach once again. Secondly, we can say that the detrimental effect on stability of ports and tubular extensions can be quantified around 20%, while the stabilizing effect of the TF coil case can be assumed as negligible, being less than 2%.

5. Conclusions

In this paper we have presented a "fast" technique for the analysis of Resistive Wall Modes in fusion devices, taking into account the three-dimensional features of the conducting structures surrounding plasma with an integral formulation of the eddy currents problem. An almost linear scaling of the computational time with the number of discrete unknowns has been achieved, thanks to the peculiar features of the surface-based coupling between plasma and conductors, and to a SVD-based sparsification of the matrix describing the electromagnetic long-range interactions. This allows the solution of problems with an unprecedented level of details. Some test cases have shown the effectiveness of the proposed method; a specific example on the ITER geometry has been presented.

Acknowledgments

G. Rubinacci and F. Villone gratefully acknowledge useful discussions with Prof. R. Albanese (CREATE) and Dr. A. Portone (EFDA), who also kindly provided the geometrical data of ITER. Y.Q. Liu acknowledges the support from UK EPSRC and Euratom/UKAEA. This work was supported in part by Italian MIUR under PRIN grant and by Consorzio CREATE.

References

[1] J.P. Freidberg, Ideal Magnetohydrodynamics, Plenum Press, 1985.
[2] G. Bateman, MHD Instabilities, MIT press, 1978.
[3] E.A. Lazarus, J.B. Lister, G.H. Neilson, Nucl. Fusion 30 (1990) 111–141.
[4] R. Albanese, E. Coccorese, G. Rubinacci, Nucl. Fusion 29 (1989) 1013-.
[5] R. Albanese, F. Villone, Nucl. Fusion 38 (1998) 723–738.
[6] M. Ariola, G. Ambrosino, J.B. Lister, A. Pironti, F. Villone, P. Vyas, A modern plasma controller tested on the TCV tokamak, Fusion Technol. 36 (2) (1999) 126–138.
[7] M. Okabayashi et al, Stabilization of the RWM in DIII-D by plasma rotation and magnetic feedback, Plasma Phys. Contr. Fus. 44 (2002) B339–B355.
[8] Y.Q. Liu et al, Stabilization of RWMs in ITER by active feedback and toroidal rotation, Nucl. Fusion 44 (2004) 232–242.
[9] S. Martini et al, Nucl. Fusion 47 (2007) 783–791.
[10] Y.Q. Liu et al, Phys. Plasmas 7 (2000) 3681.
[11] L. Degtyarev et al, Comp. Phys. Comm. 103 (1997) 10–27.

[12] J. Bialek et al, Phys. Plasmas 8 (2001) 2170–2180.
[13] F. Villone, R. Albanese, Y.Q. Liu, A. Portone, G. Rubinacci, 3D effects of conducting structures on RWMs control in ITER, in: Proceedings of the 34th EPS Conference, Warsaw, Poland, July 2007.
[14] A. Portone, F. Villone, Y. Liu, R. Albanese, G. Rubinacci, Linearly perturbed MHD equilibria and 3D eddy current coupling via the control surface method, Plasma Phys. Contr. Fusion 50 (2008) 085004.
[15] R. Albanese, Y.Q. Liu, A. Portone, G. Rubinacci, F. Villone, Coupling between a 3D integral eddy current formulation and a linearized MHD model for the analysis of resistive wall modes, IEEE Trans. Mag. 4 (2008) 1654–1657.
[16] F. Villone, Y.Q. Liu, R. Paccagnella, T. Bolzonella, G. Rubinacci, Effects of three-dimensional electromagnetic structures on resistive-wall-mode stability of reversed field pinches, Phys. Rev. Lett. 100 (2008) 255005.
[17] R. Albanese, G. Rubinacci, Integral formulation for 3D eddy current computation using edge-elements, in: IEE Proceedings, vol. 135, Part A, No. 5, September 1988, pp. 457–462.
[18] R. Albanese, G. Rubinacci, Finite element methods for the solution of 3D eddy current problems, Adv. Imaging Electron Phys. 102 (1998) 1–86.
[19] J.R. Phillips, J.K. White, A precorrected-FFT method for electrostatic analysis of complicated 3d structures, IEEE Trans. Computer-Aided Design Integrated Circuits 16 (1997) 1059–1072.
[20] G. Rubinacci, A. Tamburrino, S. Ventre, A FFT integral formulation using edge elements for eddy current testing, Int. J. Appl. Electromagn. Mech. 11 (2001) 141–162.
[21] G. Rubinacci, A. Tamburrino, S. Ventre, F. Villone, A fast algorithm for solving 3D eddy current problems with integral formulations, IEEE Trans. Magn. 37 (2001) 3099–3103.
[22] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, J. Comput. Phys. 73 (1987) 325–348.
[23] H. Cheng, L. Greengard, V. Rokhlin, A fast adaptive multipole algorithm in three dimensions, J. Comput. Phys. 155 (1999) 468–498.
[24] G. Rubinacci, A. Tamburrino, S. Ventre, F. Villone, A fast 3-D multipole method for eddy-current computation, IEEE Trans. Mag. 40 (2) (2004) 1290–1293.
[25] R. Albanese, G. Rubinacci, A. Tamburrino, S. Ventre, F. Villone, A fast 3D eddy currents integral formulation, COMPEL 20 (2) (2001) 317–331.
[26] S. Kapur, D.E. Long, IES3: Efficient electrostatic and electromagnetic simulation, IEEE Comput. Sci. Eng. Mag. 5 (1998) 60–67.
[27] R.J. Burkholder, Jin-Fa Lee, Fast dual-MGS block-factorization algorithm for dense MoM matrices, IEEE Trans. Antennas Propag. 52 (7) (2004) 1693–1699.
[28] D. Gope, V. Jandhyala, Efficient solution of EFIE via low-rank compression of multilevel predetermined interactions, IEEE Trans. Antennas Propag. 53 (10) (2005).
[29] A. Maffucci, G. Rubinacci, A. Tamburrino, S. Ventre, F. Villone, Fast low Frequency Impedence Extraction using a volumetric three-dimensional integral formulation, in: Proceedings of 23rd Annual Review of Progress in Applied Computational Electromagnetics, March 19–23, Verona, Italy, 2007, pp. 1652–1657.
[30] W. Hackbusch, A sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices, Computing (62) (1999) 89–108.
[31] W. Hackbusch, B.N. Khoromskij, A sparse H-matrix arithmetic, part II: Application to multidimensional problems, Computing (64) (2000) 21–47.
[32] S. Börm, L. Grasedyck, W. Hackbusch, Hierarchical matrices Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, Germany, Lecture note 21/2003, revised version June 2006, <http://www.mis.mpg.de/publications/other-series/ln/lecturenote-2103.html>.
[33] J. Smajic, Z. Andjelic, M. Bebendorf, Fast BEM for eddy-current problems using H-Matrices and adaptive cross approximation, IEEE Trans. Magn. 43 (2007) 1269–1272.
[34] C.F. Van Loan, G.H. Golub, Matrix Computations, Johns Hopkins University Press, London, 1996.
[35] S.A. Goreinov, E.E. Tyrtyshnikov, N.L. Zamarashkin, A theory of pseudoskeleton approximations, Lin. Alg. Appl. 261 (1997) 1–22.
[36] G. Rubinacci, A. Tamburrino, F. Villone, Circuits/fields coupling and multiply connected domains in integral formulations, IEEE Trans. Mag. 38 (2) (2002) 581–584.
[37] R. Gruber, F. Troyon, D. Berger, L.C. Bernard, S. Rousset, R. Schreiber, W. Kerner, W. Schneider, K.V. Roberts, Erato stability code, Comput. Phys. Comm. 21 (1981) 323–371.
[38] A. Bondeson, G. Vlad, H. Luetjens, Computation of resistive instabilities in toroidal plasmas, in: Proceedings of the IAEA Technical Committee Meeting on Advances in Simulation and Modeling of Thermonuclear Plasmas, Montreal, June 1992, pp. 306–315.
[39] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.