energies

Article

A Fast Matrix Compression Method for Large Scale Numerical Modelling of Rotationally Symmetric 3D Passive Structures in Fusion Devices

Francesca Cau [1], Andrea Gaetano Chiariello [2], Guglielmo Rubinacci [3], Valentino Scalera [4], Antonello Tamburrino [5], Salvatore Ventre [5,*] and Fabio Villone [3]

[1] Fusion for Energy (F4E), Carrer de Josep Pla, 9, 08019 Barcelona, Spain; francesca.cau@f4e.europa.eu
[2] CREATE, DI, Università degli Studi della Campania "Luigi Vanvitelli", Via Roma, 28, 80125 Aversa, Italy; andreagaetano.chiariello@unicampania.it
[3] CREATE, DIETI, Università degli Studi di Napoli Federico II, Via Claudio, 21, 80125 Napoli, Italy; rubinacci@unina.it (G.R.); fabio.villone@unina.it (F.V.)
[4] Engineering Department, University of Naples Parthenope, 80143 Napoli, Italy; valentino.scalera@uniparthenope.it
[5] CREATE, DAEIMI, Università degli Studi di Cassino e del Lazio Meridionale, Via G. Di Biasio, 43, 03043 Cassino, Italy; tamburrino@unicas.it
* Correspondence: ventre@unicas.it

Abstract: This paper illustrates the development of a recursive QR technique for the analysis of transient events, such as disruptions or scenario evolution, in fusion devices with three-dimensional conducting structures using an integral eddy current formulation. An integral formulation involves the solution, at each time step, of a large full linear system. For this reason, a direct solution is impractical in terms of time and memory consumption. Moreover, typical fusion devices show a symmetric/periodic structure. This can be properly exploited when the plasma and other sources possess the same symmetry/periodicity of the structure. Indeed, in this case, the computation can be reduced to only a single sector of the overall structure. In this work the periodicity and the symmetries are merged in the recursive QR technique, exhibiting a huge decrease in the computational cost. Finally, the proposed technique is applied to a realistic large-scale problem related to the International Thermonuclear Experimental Reactor (ITER).

Keywords: plasma fusion; integral formulations; fast methods; eddy current; QR-MGS sparsification; H-matrix

Citation: Cau, F.; Chiariello, A.G.; Rubinacci, G.; Scalera, V.; Tamburrino, A.; Ventre, S.; Villone, F. A Fast Matrix Compression Method for Large Scale Numerical Modelling of Rotationally Symmetric 3D Passive Structures in Fusion Devices. *Energies* **2022**, *15*, 3214. https://doi.org/10.3390/en15093214

Academic Editor: Hiroshi Sekimoto

Received: 25 March 2022
Accepted: 25 April 2022
Published: 27 April 2022

1. Introduction

The working principle of magnetic confinement fusion devices is fundamentally based on the electromagnetic interaction of the plasma, where fusion reactions occur with the conducting structures surrounding the plasma, and where external currents circulate. Such conductors can be both actively fed (e.g., poloidal field, PF, coils and toroidal field, TF, and coils) and passive (e.g., the vacuum vessel). The currents flowing in the former set are responsible for the magnetic field needed to keep the plasma in equilibrium, hence giving rise to the nominal desired plasma configuration. Conversely, passive conductors play a fundamental role in the stability of such an equilibrium configuration. In fact, assuming the plasma is described as a current-carrying fluid using the Magneto-Hydro-Dynamics (MHD) model, it can be demonstrated that there exist unstable modes of evolution—the so-called MHD instabilities. A huge number of such unstable modes may exist, whose triggering depends on several physical and geometrical plasma parameters, such as pressure, current, position, and shape. They are usually classified in terms of the toroidal and poloidal number, i.e., the Fourier decomposition in the toroidal and poloidal direction of the dominant plasma perturbation. In the present paper, we consider situations in which the plasma evolves in an axisymmetric way, although this is not an intrinsic limitation of the proposed approach, as we will clarify below. Practically, this means that the plasma geometrical descriptors (e.g., the centroid vertical position, gaps with respect to the first wall, and the displacement with respect to the nominal configuration) may grow exponentially in time, after an initial perturbation takes place. The time scale of such phenomenon—the so-called growth rate of the instability—may be as fast as microseconds in typical fusion devices, in the absence of passive conducting structures, being related to the inertial dynamics of the plasma, which is very fast due to its very small mass. Such abrupt movements and deformations of the current-carrying plasma induce eddy currents in surrounding passive conductors; such eddy currents, in turn, tend to counteract the plasma movement, thanks to the Lenz rule, hence slowing down the instability to the time scale over which eddy currents decay—tens or hundreds of milliseconds for typical devices. This allows an active control of the instability to be practically feasible [1].

The discussion reported above demonstrates clearly that numerical modelling of the conducting structures surrounding the plasma is of paramount importance for a macroscopic description of the behaviour of a fusion device. A comprehensive, modern, and rigorous presentation of the theory leading to the numerical modelling of such complex electromagnetic problems can be found in the book of Bossavit [2]. In its exposition, Bossavit limits the attention to the variational formulation of the field problems. In this case, the quasi-stationary approximation of the Maxwell equations are discretized by the Finite Element Method (FEM). Here, the computational domain in principle also includes the air. This approach is very well diffused (see for instance [3]) and it is at the basis of many general-purpose commercial codes. For avoiding the discretization of the air, a boundary integral equation can be coupled to the differential equations on the boundary of the conducting domain, leading to the so-called FEM-BEM (Boundary Element Method), as described for instance in [2,4,5]. The thermonuclear fusion devices present very complex geometries in which massive structures and thin shells coexist and ferromagnetic effects are usually negligible. In this respect, integral formulations are usually quite advantageous, because they allow (i) to mesh the conductors only and (ii) to implicitly take into account regularity conditions at infinity. For this reason, in the first generation of tokamaks, made almost completely of thin conducting structures, the finite element eddy current integral formulation for nonmagnetic shells proposed by Kameari [6], Bossavit [7], and Blum et al. [8] resulted in being particularly efficient, from the computational point of view. The main obstacle to the treatment of fully three-dimensional massive conductors was the lack of a general method for generating a complete set of independent, solenoidal shape functions for the current density. The proposal [9] of a numerical formulation for modelling in an efficient way these 3D massive structures was the main step leading to the implementation of the CARIDDI code [9,10], and indeed it has been extensively used for the modelling of fusion devices, such as for instance JET, EAST, COMPASS, RFX, JT-60SA, ITER, and DEMO. On the other hand, general purpose commercial codes based on differential formulations substantially improved in the course of the years [11–14]. Nowadays, they represent a valid alternative in many cases where the presence of the plasma does not pose specific challenges in the modelling.

One drawback of integral formulations—and of course CARIDDI is no exception—is that they give rise to fully populated matrices to be inverted to find the solution. This inevitably poses a limitation to the maximum number of discrete unknowns—the Degrees Of Freedom, DOF—that can be practically handled. On the other hand, the complexity of devices and the accuracy required for the analyses require very detailed models and hence huge numerical models to be considered. To tackle this problem, so-called “fast methods” can be used, ranging from FFT (Fast Fourier Transform)-based approaches [15–17] to multipole approximations [18,19], to QR-recursive compression [20,21]. Although all such methods have been successfully implemented in the CARIDDI code, our previous experience clearly show that for fusion devices, the most effective technique is the QR-recursive (see [22] for the definition of QR matrix factorization). This is the reason why, in the present paper, we focus on a number of significant extensions of such a technique, which computes the effective numerical solution of the problem with a very high efficiency, hence allowing an unprecedented level of detail in the description of fusion devices.

In the present paper, the authors starting from the previous version of the QR-recursive method, face some important numerical issues, adding new efficient enhancements. We can summarize the new features addressed by the paper in the following:

- The extension of the method when both the structure and the sources exhibit the same symmetries. This is quite important because symmetry is typical in devices for fusion applications.
- A new approach (namely the DOF-based method) is introduced for the QR-recursive method, which compared to the old one (ELEMENT-based method) is numerically more effective.
- An efficient numerical approach is used in solving the system in order to handle the "electrodes" case. This case is very usual for practical fusion device application.
- We tackle the problem of the "small boxes" in QR-recursive, which could degrade the performances of the overall method.

Finally, we applied the method to a relevant application case of the International Thermonuclear Experimental Reactor (ITER) [23].

2. Mathematical Formulation

2.1. Integral Formulation

Here we summarize the Magneto-Quasi-Static volume integral formulation at the basis of the numerical model. We refer to a conducting three-dimensional domain $\mathrm{V_c}$ excited by a time varying magnetic field and by a set of current/voltage sources applied at a subset S_E of its boundary made by a set of equipotential electrodes.

Faraday's law is automatically satisfied by assuming

$$\mathbf{E} = -\frac{\partial \mathbf{A}}{\partial \mathrm{t}} - \nabla \varphi\,, \ \text{ in } \mathbb{R}^3. \tag{1}$$

The magnetic vector potential is uniquely defined by $\nabla \times \mathbf{A} = \mathbf{B}$ with the Coulomb gauge and the regularity conditions at infinity. Consequently, it is possible to express $\mathbf{A}$ in terms of the unknown solenoidal current density, by using the Biot–Savart law:

$$\mathbf{A}(\mathbf{r},\mathrm{t}) = \frac{\mu_0}{4\pi}\int_{\mathrm{V_c}} \frac{\mathbf{J}\left(\mathbf{r}',\mathrm{t}\right)}{\left|\mathbf{r}-\mathbf{r}'\right|}\mathrm{dV}' + \mathbf{A}_\mathrm{s}(\mathbf{r},\mathrm{t})\,, \ \text{ in } \mathbb{R}^3, \tag{2}$$

where

$$\mathbf{A}_\mathrm{s}(\mathbf{r},\mathrm{t}) = \frac{\mu_0}{4\pi}\int_{\mathbb{R}^3-\mathrm{V_c}} \frac{\mathbf{J}_\mathrm{s}\left(\mathbf{r}',\mathrm{t}\right)}{\left|\mathbf{r}-\mathbf{r}'\right|}\mathrm{dV}'\,, \ \text{ in } \mathbb{R}^3, \tag{3}$$

and $\mathbf{J}_\mathrm{s}$ is the (known) current density due to the external sources.

The constitutive equation

$$\mathbf{J} = \sigma\,\mathbf{E}\,, \quad \text{in } \mathrm{V_c}\,, \tag{4}$$

where σ is the electric conductivity tensor, can be verified in an average sense by adopting a weighted residual approach. It results in the following weak form

$$\int_{V_\mathrm{c}} \mathbf{w}\cdot\left\{\sigma^{-1}\mathbf{J} + \frac{\partial}{\partial \mathrm{t}}\left[\frac{\mu_0}{4\pi}\int_{V_\mathrm{c}}\frac{\mathbf{J}\left(\mathbf{r}',\mathrm{t}\right)}{\left|\mathbf{r}-\mathbf{r}'\right|}\mathrm{dV}'\right] + \frac{\partial \mathbf{A}_\mathrm{s}}{\partial \mathrm{t}}\right\}\mathrm{dV} + \sum_{\mathrm{h=1,\,N_E}}\int_{\mathrm{S_E}} \boldsymbol{\phi}_\mathbf{h}\,\mathbf{w}\cdot\hat{\mathbf{n}}\,\mathrm{dS} = 0 \tag{5}$$

for any $\mathbf{w} \in \mathrm{S}$ and with $\mathbf{J} \in \mathrm{S}$. Here S is the set of solenoidal vector functions with zero normal component on $\partial \mathrm{V_c} \backslash \mathrm{S_E}$, being $\hat{\mathbf{n}}$ the unit normal pointing outward $\mathrm{V_c}$:

$$\mathrm{S} = \left\{ \mathbf{w} \in \mathrm{H}(\mathrm{div};\ \mathrm{V_c}) \;\middle|\; \mathrm{div}\ \mathbf{w} = 0 \text{ in } \mathrm{V_c} \text{ and } \mathbf{w} \cdot \hat{\mathbf{n}} = 0 \text{ on } \partial \mathrm{V_c} \backslash \mathrm{S_E} \right\}. \tag{6}$$

N_E is the number of electrodes (they are part of the boundary of the conducting domain), identified by the surface $\mathrm{S_E}$. At each electrode h, there is an unknown potential $\boldsymbol{\phi}_\mathbf{h}$ and a prescribed current $\mathbf{I}_\mathrm{h}(\mathrm{t})$.

In Figure 1 we report a schematic sketch showing the geometrical objects involved in Equation (5). Note that, being an integral formulation, only the sources in the conducting domain are involved in the computation of the integrals. The air regions do not bring any additional unknowns.

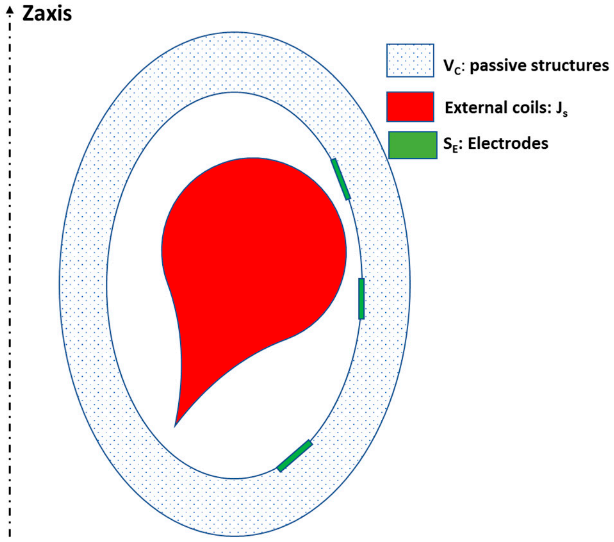


Figure 1. A schematic figure reporting the meaning of the symbols used in Equation (5).

2.2. Numerical Model

A numerical solution is obtained by applying Galerkin's method to Equation (5). The unknown is the current density vector $\mathbf{J}$, which we represent as the linear combination of N basis functions $\mathbf{J}_i \in \mathrm{S}$:

$$\mathbf{J}(\mathbf{r}, \mathrm{t}) = \sum_{i=1:N} \mathrm{I}_i(\mathrm{t})\, \mathbf{J_i}(\mathbf{r}). \tag{7}$$

According to the Galerkin's method, we choose N independent weighting functions as $\mathbf{W}_i = \mathbf{J}_i$. Condition $\mathbf{J}_i \in S$ can be satisfied by introducing the electric vector potential $\mathbf{T}$ ($\mathbf{J} = \nabla \times \mathbf{T}$) and adopting edge element shape functions for $\mathbf{T}$. The uniqueness of the vector potential is assured by the tree-cotree gauge [10]. This gauge is conveniently imposed directly on the basis functions, introducing the tree-cotree decomposition of the mesh and eliminating the degrees of freedom associated to the tree edges. Condition $\mathbf{J}_i \cdot \hat{\mathbf{n}} = 0$ on $\partial \mathrm{V_c} \backslash \mathrm{S_E}$ can also be imposed on the shape functions using the approach described in [10,24,25]. The shape functions $\mathbf{J}_i$ are therefore derived from the $\mathbf{N}_i$ edge element functions for the gauged vector potential:

$$\mathbf{J}_i = \nabla \times \mathbf{N}_i. \tag{8}$$

In this way, Galerkin's method applied to (5) gives the following linear dynamical system, for $\mathrm{t} \geq 0$:

$$\mathbf{L}\frac{\mathrm{d}\mathbf{I}}{\mathrm{dt}} + \mathbf{RI} + \mathbf{F}^\mathrm{T}\boldsymbol{\Phi} = -\frac{\mathrm{d}\mathbf{V}_\mathrm{S}}{\mathrm{dt}}, \tag{9}$$

with $\mathbf{I}(0) = \mathbf{I}_0$, where $\mathbf{I}_0$ is the prescribed initial condition.

In (9) the unknowns are

- $\mathbf{I}(\mathrm{t})$ is a column vector discretizing the unknown density current $\mathbf{J}(\mathbf{r},\mathrm{t})$(i.e., $\mathbf{I}(\mathrm{t}) = [\mathrm{I}_i(\mathrm{t})]$)
- $\boldsymbol{\Phi}(t)$ is a column vector representing the unknown voltages at the electrodes (i.e., $\boldsymbol{\Phi}(t) = [\boldsymbol{\phi}_{\mathbf{h}}(t)]$)

In the RHS of (9) appears the source term $\mathbf{V}_S(\mathrm{t}) = [V_{\mathrm{sk}}(\mathrm{t})]$ given by

$$V_{sk}(t) = \int_{V_c} \nabla \times \mathbf{N}_{\mathrm{k}}(\mathbf{r}) \cdot \mathbf{A}_{\mathrm{s}}(\mathbf{r},\mathrm{t})\, \mathrm{d}V \tag{10}$$

The matrices in (9) are $\mathbf{R} = [R_{ij}]$, $\mathbf{L} = [L_{ij}]$, $\mathbf{F} = [F_{ij}]$. They are defined as

$$R_{ij} = \int_{V_c} \nabla \times \mathbf{N}_i(r) \cdot \sigma^{-1} \nabla \times \mathbf{N}_j(\mathbf{r})\, \mathrm{d}V, \tag{11}$$

$$L_{ij} = \frac{\mu_0}{4\pi} \int_{V_c} \int_{V_c} \frac{\nabla \times \mathbf{N}_{\mathrm{i}}(\mathbf{r}) \cdot \nabla \times \mathbf{N}_{\mathrm{j}}\left(\mathbf{r}'\right)}{\left|\mathbf{r} - \mathbf{r}'\right|} dV dV', \tag{12}$$

$$F_{ij} = \int_{S_i} \nabla \times \mathbf{N}_{\mathrm{j}}(\mathbf{r}) \cdot \hat{\mathbf{n}}\, \mathrm{d}S. \tag{13}$$

The dimension of the matrix $\mathbf{F}$ is $\mathrm{N}_{\mathrm{E}} \times N$. It is worth noticing that

- F_{ij} is the contribution to the current flowing in the electrode i due to the DOF j

The current C_i flowing in the *i-th* electrode, is the product of the *i*-th row ($\mathbf{F}_{\mathbf{i}}$) of **F** times **I.** That is:

$$C_i = \mathbf{F}_{\mathrm{i}}\mathbf{I}. \tag{14}$$

Rewriting the condition (14) for each electrode we have

$$\mathbf{FI} = \mathbf{I}_{\mathrm{h}}, \tag{15}$$

- F_{ij} is zero for those DOF $\boldsymbol{j}$, which do not belong to the boundary.

Hence **F** is sparse due to the local interaction of the electrodes.

Matrices **R** and **L** (whose dimensions are $N \times N$) represent the discrete counterparts of the electric constitutive relationship (Ohm's law) and of the vector potential operator, respectively. Matrices **R** and **L** are symmetric and positive definite. In addition, matrix **R** is sparse, since it arises from a local operator, whereas matrix **L** is fully populated, because it arises from the Biot–Savart integral operator.

Applying implicit time step integration to Equation (9), and keeping into account constrain (15) and the initial condition, the algebrical system to be solved is:

$$\begin{gathered}(\mathbf{L} + \Delta\mathrm{t}\, \mathbf{R})\mathbf{I}^{(\mathrm{k}+1)} + \Delta\mathrm{t}\, \mathbf{F}^{\mathrm{T}} \boldsymbol{\phi}^{(\mathrm{k}+1)} = \mathrm{L}\mathbf{I}^{(\mathrm{k})} + \mathbf{V}_{\mathrm{S}}^{(\mathrm{k}+1)} - \mathbf{V}_{\mathrm{S}}^{(\mathrm{k})} \\ \mathrm{F}\, \mathbf{I}^{(\mathrm{k}+1)} = \mathbf{I}_{\mathrm{h}}^{(\mathrm{k}+1)} \\ \mathbf{I}^{(0)} = \mathbf{I}_0 \\ \mathrm{k} \geq 1\end{gathered} \tag{16}$$

where

- Δt is the time step
- $\mathbf{I}^{(\mathrm{k})} = \mathbf{I}(\mathrm{k}\Delta\mathrm{t})$ and $\boldsymbol{\phi}^{(\mathrm{k})} = \boldsymbol{\phi}(\mathrm{k}\Delta\mathrm{t})$ are the unknowns at integration instant kΔt
- $\mathbf{I}_{\mathrm{h}}^{(\mathrm{k}+1)} = \mathbf{I}_{\mathrm{h}}((\mathrm{k}+1)\Delta t)$ are the prescribed currents flowing in the electrodes at integration instant kΔt
- and $\mathbf{V}_{\mathrm{s}}^{(\mathrm{k})} = \mathbf{V}_{\mathrm{s}}(\mathrm{k}\Delta\mathrm{t})$ are source terms at integration instant kΔt

2.3. Symmetric Periodic Boundary Condition

In many relevant situations, two proper conditions are satisfied. First, the plasma discharge is axisymmetric. Therefore, the plasma discharge can also be retained periodically, with the same period of the nuclear fusion reactor. Second, the electrodes present on each sector and used to inject electrical energy into the system, may be driven by the same waveforms. Under these two conditions, the electromagnetic fields are periodic with the period of the nuclear fusion reactor. Similar arguments can be used in the presence of other symmetries, other than rotations.

For the sake of clarity, in Figure 2 we report a simple case in which a periodic boundary condition should be set at $\varphi = -20^\circ$ and $\varphi = +20^\circ$ on a symmetry mesh. We should force the density current $\mathbf{J}(\mathbf{r}, t)$, at each boundary point at $\varphi = -20^\circ$, to be equal to the correspondent point at $\varphi = +20^\circ$.

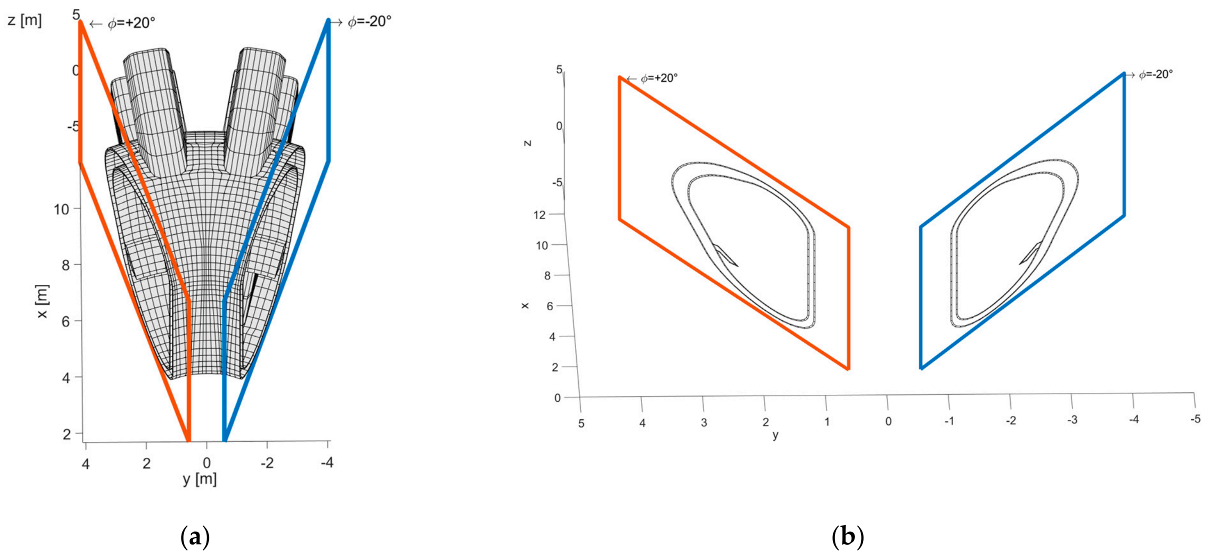


Figure 2. Forcing periodic boundary conditions. (**a**) The sector mesh ($-20^\circ < \varphi < 20^\circ$); (**b**) The boundary faces at $\varphi = -20^\circ$ and $\varphi = +20\ ^\circ$. The current flowing at each face at $\varphi = -20^\circ$ should be equal to the current flowing in the correspondent face at $\varphi = +20^\circ$.

Let α be defined as the semi-angular width of the sector. For each pair of facets f_1 and f_2 lying, respectively, at $\varphi = +\alpha$ and $\varphi = -\alpha$ and coupled by the condition imposed by the periodicity, we force the current flowing through f_1 to be equal to the one flowing through f_2. This can be easily done by the means of the matrix $\mathbf{F}$. Keeping into account definition (15) we should have

$$C_{f1} = C_{f2}, \tag{17}$$

and hence

$$(\mathbf{F}_{f1} - \mathbf{F}_{f2})\mathbf{I} = 0 \tag{18}$$

Therefore, the periodic boundary conditions act like the electrode's conditions, adding an equation such as (18) for each corresponding pair of facets.

Summing up, in the presence of both electrode currents and periodic boundary conditions, the second equation in (16) is generalized as:

$$\mathbf{B}\,\mathbf{I}^{(k+1)} = \mathbf{M}^{(k+1)}, \tag{19}$$

where

$$\mathbf{B} = \begin{bmatrix} \mathbf{F_E} \\ \mathbf{F}_{\alpha} - \mathbf{F}_{-\alpha} \end{bmatrix}, \tag{20}$$

$$\mathbf{M}^{(k+1)} = \begin{bmatrix} \mathbf{I}_h^{(k+1)} \\ 0 \end{bmatrix} \tag{21}$$

$$\mathbf{F}_E \overset{\text{def}}{=} \left\{ F_{ij} \,\middle|\, i \in S_E,\ j \in 1, N \right\}$$

$$\mathbf{F}_{\boldsymbol{\alpha}} \overset{\text{def}}{=} \left\{ F_{ij} \,\middle|\, i \in \pi_\alpha,\ j \in 1, N \right\}$$

$$\mathbf{F}_{-\alpha} \overset{\text{def}}{=} \left\{ F_{ij} \,\middle|\, i \in \pi_{-\alpha},\ j \in 1, N \right\}$$

where π_α and $\pi_{-\alpha}$ are the planes at $\varphi = +\alpha$ and $\varphi = -\alpha$, respectively.

In accordance with the two different constrains, the matrix **B** has a number of row equal to $N_E + N_p$, where

N_E is the number of electrodes on S_E where the current should be prescribed.

N_p is the number of faces at $\varphi = +\alpha$ (or at $\varphi = -\alpha$).

Finally, in place of the (16), the overall numerical system to be solved is:

$$\begin{aligned} &(\mathbf{L} + \Delta t\, \mathbf{R})\mathbf{I}^{(k+1)} + \Delta t\, \mathbf{B}^T \boldsymbol{\phi}^{(k+1)} = \mathbf{L}\mathbf{I}^{(k)} + \mathbf{V}_S^{(k+1)} - \mathbf{V}_S^{(k)} \\ &\mathbf{B}\, \mathbf{I}^{(k+1)} = \mathbf{M}^{(k+1)} \\ &\mathbf{I}^{(0)} = \mathbf{I}_0 \\ &k \geq 1 \end{aligned} \tag{22}$$

At the discrete level, the dynamical matrix is:

$$\mathbf{A} = \mathbf{R}\, \Delta t + \mathbf{L}. \tag{23}$$

The solution of large-scale problems involving a fully populated matrix poses a relevant challenge both in the assembly and in the solution of the dynamical system (22) [26]. Indeed

(i) The assembly of **L** has a cost of $O(N^2)$.

(ii) The inversion (usually factorization) of the matrix **A**, using a direct method, as well know, requires $O(N^3)$ operations.

The authors developed a parallel implementation of the direct method to solve system (22) [26,27], based on the Scalapack library [28].

3. QR-Recursive Method

3.1. Summary of QR-Recursive Approach

As explained in the previous section, the solution of (22) cost is prohibitively large when the mesh of the structure is very detailed. The compression of the stiffness matrix is recommended only for large scale problems, otherwise a direct method is preferred. When working on large scale problems, the compression of such large matrices as a whole is too expensive, from the computational point of view. Therefore, we are obliged to resort to recursive approaches, such as the QR-recursive method proposed in the present work.

In iterative methods, the matrix-vector product is the fundamental building block of the solution, and the QR-recursive method [20,21] is actually a way to evaluate efficiently the matrix-vector product.

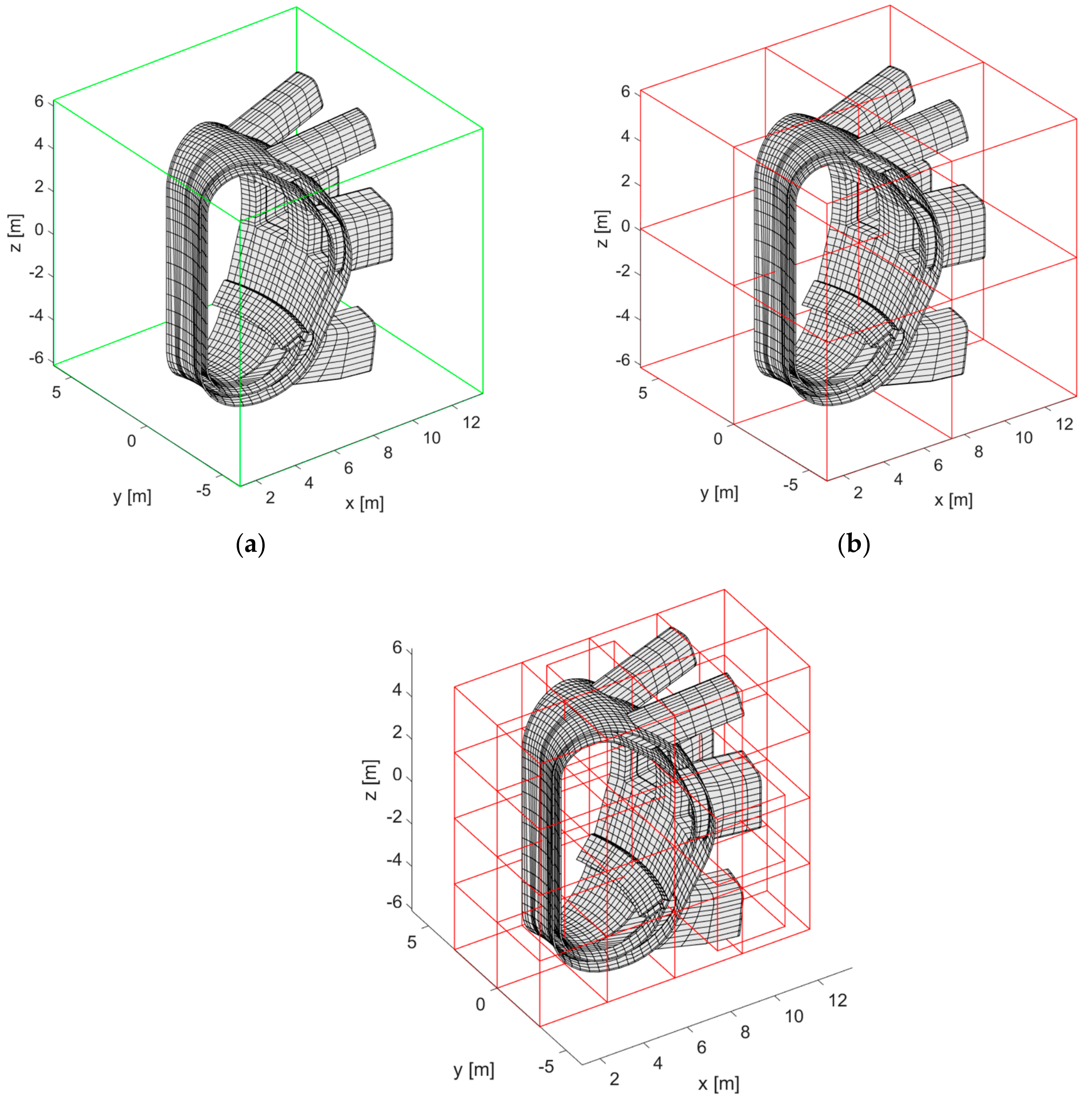
The direct computation of the matrix-vector product **Ax** (see [29]) is actually too costly, from the numerical point of view (it costs $O(N^2)$ operations); QR-based methods [30], compress the matrix and recast the product evaluation in a cheap cost (which scales $O(N)$) by the means of separation between the near and far interactions. The near interactions should be computed without approximation; on the contrary, the far interaction could be approximated.

The relevant steps of the QR-recursive algorithm are:

(1) Boxes generation
(2) Definition of DOFs associated to the boxes
(3) Definition of interaction matrix between two boxes
(4) Setting $\mathbf{L}_{\text{near}}$, $\mathbf{L}_{\text{far}}$
(5) Overall compressed operator optimization and practical considerations

3.1.1. Boxes Generation

The first step is grouping the mesh "objects" into clusters (**boxes**). The structure is recursively generated by subdividing the domain in cubic cells, halving the edge length at each subdivision. This procedure starts from the smallest cube containing the whole mesh (**b0**) and ends when a prescribed minimum number (*smin1*) of "objects" is left in the box. The resulting boxes, which should not require further division, are called childless boxes. In the Appendix A**,** we report the full algorithm used, and in Figure 3 we show an example of an application.



(**a**)

(**b**)

(**c**)

Figure 3. Boxes Generation algorithm. (**a**) The **b0** at level = 0. (**b**) The eight boxes (children) generated at level equal 1 by subdivision of the box **b0**. (**c**) Shows the third level subdivision. Note that empty boxes are deleted.

Here, we use the elements as mesh "objects". Hence, we refer to this method as the **ELEMENT-based** method.

As a final remark, we highlight that the shape of the elementary cell, the cube in this work, has to satisfy the following three properties:

(a) The cell is able to tessellate the 3D space.
(b) The cell can be exactly subdivided in eight smaller replicas.
(c) The cell should present an aspect ratio of the order of unity. In other words, the cell should not be either flattened or elongated.

These conditions, bring naturally to cubic cells.

3.1.2. Definition of DOFs Associated to the Boxes

Once the boxes are created, we can define the DOFs to insert into the boxes.

Let E and D be defined as

$$\mathrm{E}(\mathbf{b}) \overset{\mathrm{def}}{=} \text{is the set of the all elements mesh belonging to the box } \mathbf{b}$$

$$\mathrm{D}(\mathbf{b}) \overset{\mathrm{def}}{=} \text{is the set of all DOFs } \in \mathrm{E}(\mathbf{b}) \tag{24}$$

Note that the support of a given DOF $\in \mathrm{D}(\mathbf{b})$, could not necessarily belong to $\mathrm{E}(\mathbf{b})$**.** This actually occurs if the DOF lies on the boundary of the box. So, we label the DOFs as the internal DOF or boundary DOF (see Figure 4).

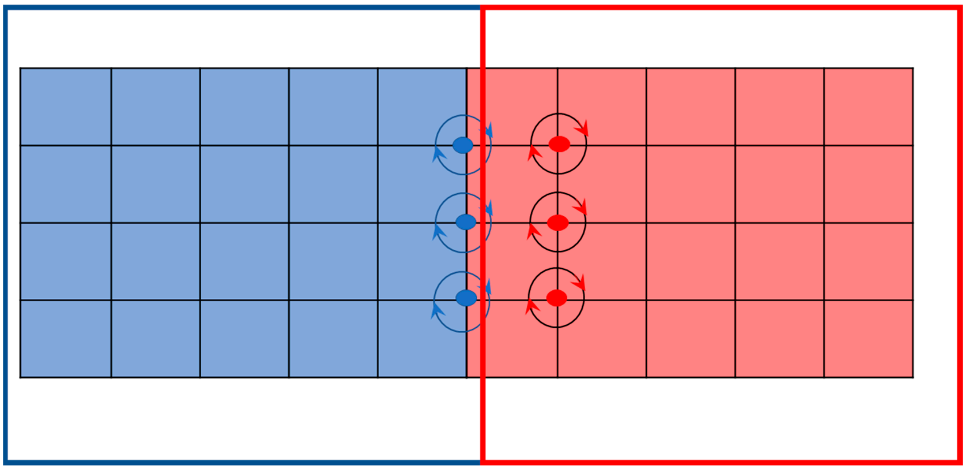


Figure 4. ELEMENT-based approach. Each element is associated to the box in which its barycenter lies. The two boxes element are marked blue and red, respectively. The blue DOFs are boundary DOF the red one are internal.

3.1.3. Definition of the Interaction Matrix between Two Boxes

Using the definition in (24), we are able to define the interactions matrix between a field **b** and source box **c**.

The matrix interaction, namely $\mathbf{L}^{\mathbf{bc}}$, is defined as

$$\mathbf{L}^{\mathbf{bc}} = \left\{ \mathbf{L}^{*}_{\mathrm{ij}} \right\}, \;\; \mathrm{i} \, \in \, \mathrm{D}(\mathbf{b}), \;\; \mathrm{j} \, \in \mathrm{D}(\mathbf{c}), \tag{25}$$

where

$$\mathbf{L}^{*}_{\mathrm{ij}} = \sum_{\mathrm{r} \in \mathrm{E}(\mathbf{b})} \sum_{\mathrm{s} \in \mathrm{E}(\mathbf{c})} \mathbf{L}_{\mathrm{i,j,r,s}}. \tag{26}$$

The terms $\mathbf{L}_{\mathrm{i,j,r,s}}$ represent the element–element contribution to the computation of $\mathbf{L}_{\mathrm{ij}}$ (see (12)). It is easy to understand that

- $\mathbf{L}^{*}_{\mathrm{ij}} \neq \mathbf{L}_{\mathrm{ij}}$ for each matrix entry of $\mathbf{L}^{\mathbf{bc}}$ for which i or j is a boundary DOF.

This is because some element–element interactions are missing.

- $\mathbf{L}^*_{\mathrm{ij}} = \mathbf{L}_{\mathrm{ij}}$ for each matrix entry of $\mathbf{L}^{\mathbf{bc}}$ for which both *i* and *j* are internal DOFs.

Hence, in the ELEMENT-based approach, the box–box interaction matrix $\mathbf{L}^{\mathbf{bc}}$ could not be a submatrix of the full matrix **L** (see Figure 4).

Nevertheless, the **L** matrix will be consistently reconstructed by superposition of all the $\mathbf{L}^{\mathbf{bc}}$ contributions.

3.1.4. Setting $\mathrm{L}_{\mathrm{near}},\ \mathrm{L}_{\mathrm{far}}$

The last step is the identification of the near and far box–box interactions.

For each box–box interaction (namely **b**, **c**) in which **b** is field and **c** is the source box, we define:

- $\mathcal{L}_{\mathrm{far}}(\mathbf{b}) \overset{\mathrm{def}}{=}$ is the set of sources boxes **c**, such that the box **b** is in far zone of the box **c.**
- $\mathcal{L}_{\mathrm{near}}(\mathbf{b}) \overset{\mathrm{def}}{=}$ is the set of sources boxes **c**, such that the box **b** is in the near of the box **c.**

The analytic definition of $\mathcal{L}_{\mathrm{far}}(\mathbf{b}), \mathcal{L}_{\mathrm{near}}(\mathbf{b})$ is explained in Appendix B.

In Figure 5, we report the near and far zone generated by box **b** acting as source.

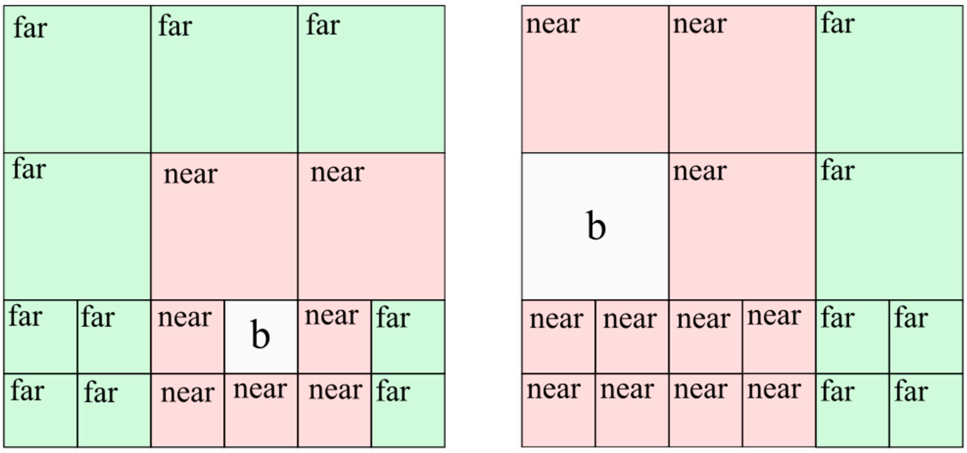


Figure 5. The box **b** is the source box. The near zone is depicted as pink, and the far zone is depicted as green.

Using such a box–box repartition, the **L** matrix is written as follows:

$$\mathbf{L} = \mathbf{L}_{\mathrm{near}} + \mathbf{L}_{\mathrm{far}}. \tag{27}$$

$\mathbf{L}_{\mathrm{near}}$ accounts for the box–box interactions, which should be computed exactly without approximation. There are two kinds of direct interactions:

- When the two boxes are classified as near, hence, they have a large rank and cannot be efficiently compressed.
- The interactions arising from non-local DOFs. These DOFs could not be compressed because they are related to nonlocal shape functions.

We call the $\mathbf{L}_{\mathrm{add}}$ matrix the sparse matrix arising from these non-local shape functions (see [31,32]).

Let $\mathrm{N}_{\mathrm{nbox}}$ be defined as the number of the boxes, the $\mathbf{L}_{\mathrm{near}}$ is

$$\mathbf{L}_{\mathrm{near}} = \mathbf{L}_{\mathrm{add}} + \sum_{\mathrm{b}=1}^{\mathrm{N}_{\mathrm{nbox}}} \sum_{\mathrm{c}=1}^{\mathbf{L}_{\mathrm{near}}(\mathbf{b})} \mathbf{L}^{\mathbf{bc}}. \tag{28}$$

$\mathbf{L}_{\mathrm{far}}$ keeps into account the far distance box–box interactions. These kinds of interactions are low rank and, hence, they can be conveniently compressed. The compression technique adopted is the modified Gram–Schmidt (MGS [22,33].). This algorithm returns an approximated QR-MGS factorization for each $\mathbf{L^{bc}}$.

$$\mathbf{L^{bc}} \cong \mathbf{Q^{bc}R^{bc}} \tag{29}$$

Hence the matrix $\mathbf{L}_{\text{far}}$ is approximated as follows:

$$\mathbf{L}_{\text{far}} = \sum_{\text{b}=1}^{\text{N}_{\text{nbox}}} \sum_{\text{c}=1}^{\mathbf{L}_{\text{far}}(\mathbf{b})} \mathbf{L^{bc}} \cong \sum_{\text{b}=1}^{\text{N}_{\text{nbox}}} \sum_{\text{c}=1}^{\mathbf{L}_{\text{far}}(\mathbf{b})} \mathbf{Q^{bc}R^{bc}}. \tag{30}$$

Doing this way, it is easy to show that both the computational cost and memory occupation are reduced. Indeed, let us consider two far boxes **b** and **c** having m and n objects, respectively. The matrix without compression (i.e., $\mathbf{L^{bc}}$) has dimension $m \times n$, whereas in its compressed matrix $\mathbf{Q^{bc}}$ has dimension $m \times r$, and $\mathbf{R^{bc}}$ has dimension $r \times n$, being r the rank of the interaction. Please note that for far interactions

$$r << m, n. \tag{31}$$

Let $\mathbf{x_c}$ be the vector representing the sources in box **c,** the direct evaluation of the product $\mathbf{L^{bc}x_c}$ has a computational cost

$$C_{dir} = m \times n, \tag{32}$$

whereas its approximated version $\mathbf{Q^{bc}}\left(\mathbf{R^{bc}x_c}\right)$ costs

$$C_{app} = (m+n) \times r. \tag{33}$$

In force of the (31) it results in that for the far interactions $\text{C}_{\text{app}} \ll \text{C}_{\text{dir}}$. This is the heart of the compression method.

For sake of clarity, in Figure 6 we report the rank distribution of the matrix $\mathbf{L^{bc}}$, where **b** is a given field box and **c** represents all the other source boxes. As expected, the rank reduces as distance increases.

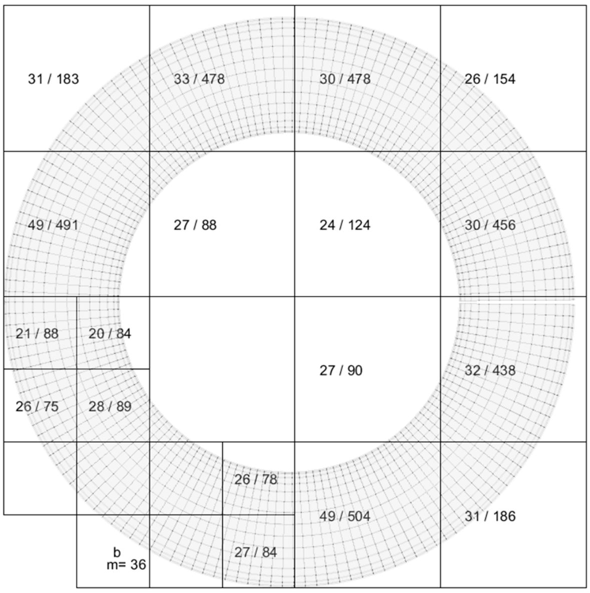


Figure 6. The rank between the field box **b** and all the other source boxes (the **c** boxes). The number of the DOFs in the field box **b** is 36. For all the far sources boxes **c** it is reported in the format X/Y format, which stands for the rank (X) of the matrix $\mathbf{L^{bc}}$ and the number of sources DOFs (Y). Note that the X/Y information is omitted for the near boxes.

Moreover, we stress that the computational cost of $\mathbf{L^{bc}x_c} \cong \mathbf{Q^{bc}R^{bc}x_c}$ is exactly equal to the memory required to store the compressed version of the interaction. Hence, all the considerations done for the memory hold for the computational cost. This is a key feature of the QR-recursive method.

Using (28) and (30), the dynamic matrix appearing in (23), can be approximated as

$$\mathbf{A} \;\approx \mathbf{A}_{\mathrm{qr}} \;= \mathbf{A}_{\mathrm{near}} + \mathbf{A}_{\mathrm{far}}, \tag{34}$$

where

$$\mathbf{A}_{\mathrm{far}} = \mathbf{L}_{\mathrm{far}}, \tag{35}$$

$$\mathbf{A}_{\mathrm{near}} = \Delta t \mathbf{R}_{add} + \mathbf{L}_{add} + \sum_{b=1}^{N_{\mathrm{nbox}}} \sum_{c=1}^{\mathbf{L}_{\mathrm{near}}(\mathbf{b})} \left(\mathbf{R^{bc}} \Delta t + \mathbf{L^{bc}} \right). \tag{36}$$

where $\mathbf{R}_{\mathrm{add}}$ is the matrix $\mathbf{R}$ evaluated in the same sparsity pattern as for $\mathbf{L}_{\mathrm{add}}$.

Moreover, note that the matrix $\mathbf{R}$, being a local sparse matrix, appears only in the near part.

We should remark that in implementation of CARIDDI code, in order to face huge cases, the parallelization is fully applied by the means of MPI (Message Passing Interface) library [34]. Indeed, the computation burden

(i) assembling of $\mathbf{L}_{\mathrm{near}}$ and $\mathbf{L}_{\mathrm{far}}$
(ii) product evaluation of $\mathbf{A}_{\mathrm{qr}}\mathbf{x}$ (i.e., $\mathbf{p}_{\mathrm{near}} = \mathbf{L}_{\mathrm{near}}\mathbf{x},\ \mathbf{p}_{\mathrm{far}} = \mathbf{L}_{\mathrm{far}}\mathbf{x}$)

Is approximatively equally distributed among processes used in computation.

As a matter of fact, this means that performances of the QR method scale linearly versus the number of processes used in computation.

Indeed, we will show (see Section 3.2) that the computational cost of the QR-recursive method, is directly connected to the time required to evaluate the compressed product $\mathbf{A}_{\mathrm{qr}}\mathbf{x}$. In turn, this time, as aforementioned, depends on the memory used. When this memory is equally redistributed among $\mathbf{N_p}$ processes, the computation cost is reduced by a factor one over $\mathbf{N_p}$

In the numerical sections, we will report some results about the numerical performances of the parallelization [28,35].

3.2. The Preconditioner and the Initial Guess Estimation

As said in previous paragraphs, the QR-recursive method uses an iterative solver for the solution of the algebraic system, which takes advantage from the compressed version of the matrix-vector product. In present work, the iterative solver adopted is GMRES [36]. As it is well known, the main operations required by an iterative solver are: (i) the computation of the matrix-by-vector product $\mathbf{Ax}$ and, (ii) the preconditioning. The computational cost of an iterative solver, then, is proportional to the number of multiplications required for the product and to the number of iterations required to achieve convergence, assuming that the preconditioning has a cheaper evaluation. The role of the preconditioner is to reduce the number of iterations and, ultimately, the computational cost. In order to understand how this preconditioner works, we apply the left preconditioner to the first equation in (22). Assuming for sake of simplicity, the system is without electrodes, we have

$$\mathbf{P}^{-1}\mathbf{A}\mathbf{I}^{(k)} = \mathbf{P}^{-1}\left(\mathbf{L}\mathbf{I}^{(k-1)} + \mathbf{V}_S^{(k+1)} - \mathbf{V}_S^{(k)}\right). \tag{37}$$

The role of the preconditioner $\mathbf{P}$ is to reduce the condition number of the stiffness matrix $\mathbf{P}^{-1}\mathbf{A}$. The “ideal” preconditioner gets this number close to unity and it is a kind of inverse of $\mathbf{A}$. A critical issue in this framework is the dependence of matrix $\mathbf{A}$ on Δt. In fast transient analysis, Δt has to be small enough to model properly the underlying dynamics. On the other hand, in slow transients Δt has to be large enough to reduce the computational cost. Therefore, the dependence of $\mathbf{A}$ on Δt yields the dependence of $\mathbf{P}$ on Δt. Specifically, for small Δt the preconditioner should be tailored on $\mathbf{L}$, whereas for large

Δt the preconditioner should be tailored on **R**, as follows from (23). In all remaining cases (Δt not too small nor too large) the preconditioner depends on **R** and **L**. The appropriate value of Δt depends on the underlying dynamics that are caused by the source, i.e., by the rate of change of $\mathbf{V}_S(\mathrm{t})$ defined in (16) and appearing in (9).

It should be stressed that a preconditioner based on the **L** matrix is still an open problem. From our knowledge and experience any attempt to insert a "light" modified version of **L** in the preconditioner poorly fails.

In this work we set $\mathbf{P} = \mathbf{R}$. As we have said, **R** is a sparse, symmetric, and positive definite matrix. Its factorization and application are numerically very effective [37]:

- The computational cost (memory and time) of the factorization is linear versus the number of unknowns. Actually, it uses Cholesky decomposition (tailored for sparse matrix).
- The back/forward substitution (involved in the Cholesky method) is also very cheap.

The **R** preconditioner is not very effective starting from a null initial value, for the initial guess of the iterative method.

Extensive numerical experiments performed on typical meshes used in fusion devices, have shown that, if we use an estimation for the initial value of the iterative solver, the **R** preconditioner works quite fine, at least when the time variations are slow. Degradation of the preconditioner will appear only in limited time slots when abrupt transitions occur. The simple time estimation scheme that we use for the initial value is

$$\mathbf{I}_0^{(\mathrm{k})} = 2\mathbf{I}^{(\mathrm{k}-1)} - \mathbf{I}^{(\mathrm{k}-2)}, \tag{38}$$

where $\mathbf{I}_0^{(\mathrm{k})}$ is the initial guess for GMRES at the current instant k and $\mathbf{I}^{(\mathrm{k}-1)}$ and $\mathbf{I}^{(\mathrm{k}-2)}$ are the solutions at one and two previous time steps, respectively.

In the numerical results section, we report the GMRES number of iterations, proving the efficiency of the proposed strategy.

Note that, being very effective with the preconditioner, the performance of the iterative solver depends only by the computational cost of the approximated product $\mathbf{A}_{\mathrm{qr}}\mathbf{x}$.

3.3. Extension of Compression to Symmetries

In the present work, we introduce a simple approach in order to apply compression whenever symmetries are present in the structure.

Hereafter we assume that (i) the nuclear fusion reactor is a periodic structure, (ii) the plasma discharge is axisymmetric, and (iii) the electrodes present on each sector are driven by the same waveform (see Section 2.3). Assumption (i) is typical in nuclear fusion machines. Assumptions (ii) and (iii) are not restrictive at all. Indeed, if not satisfied, the approach proposed in this paper can be extended to these cases by exploiting the framework of [38]. Hereafter, for the sake of simplicity, we present the compression in the presence of symmetries, under assumptions (i)–(iii).

To describe the approach, we briefly recall how they are handled in the CARIDDI code. Two classes of symmetries are handled. One is the symmetry of reflection with respect to a co-ordinate plane; the other one is the symmetry with respect to a given rotation around the axis z. The underlying idea, in order implement such a symmetry condition, is based on two considerations:

1. Reducing the solution domain V_c only to an elementary part of the whole structure. In the following, this part of the domain is called the main sector.
2. Assuming that basis functions $\nabla \times \mathbf{N}_{\mathrm{i}}(\mathbf{r})$ automatically verify the symmetry conditions, by two suitable operators: reflection and rotation.

For instance, with reference to a system of rectangular coordinates with unit vectors $\hat{\mathbf{t}}_x$, $\hat{\mathbf{t}}_y$, and $\hat{\mathbf{n}}_z$, with $\hat{\mathbf{n}}_z$ perpendicular to the plane of symmetry, we define the following reflection operator:

$$\mathbf{S}_\mathrm{n} = \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{vmatrix}, \tag{39}$$

such that the components of $\mathbf{J}_k$ at the reflected point $\mathbf{r}_r = \mathbf{S}_\mathrm{n}\mathbf{r}$ are given by

$$\mathbf{J}_i(\mathbf{r}_\mathrm{r}) = \varepsilon\, \mathbf{S}_\mathrm{n}\, \mathbf{J}_i(\mathbf{r}), \tag{40}$$

where $\mathbf{r} = x\hat{\mathbf{t}}_x + y\hat{\mathbf{t}}_y + z\hat{\mathbf{n}}_z$ is in the integration domain $\mathrm{V_c}$, $\mathbf{r}_\mathrm{r}$ is in the reflected domain, the 3D vectors are represented as column vectors made of their Cartesian components, and ε is +1 or -1, depending on the type of symmetry. It should be noticed that the continuity of $\mathbf{J} \cdot \hat{\mathbf{n}}_z$ has to be assured on the symmetry planes. Therefore, for instance, when applying a reflection with $\varepsilon = +1$, the condition $\mathbf{J} \times \hat{\mathbf{n}}_z = 0$ is automatically guaranteed on the symmetry plane.

Similarly, the rotation of an angle α around the z axis can be represented by the usual operator of rotation $\mathbf{R}_\alpha$:

$$\mathbf{R}_\alpha = \begin{vmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{vmatrix}, \tag{41}$$

such that the components of $\mathbf{J}_\mathrm{i}$ at the rotated point $\mathbf{r}_\mathrm{r} = \mathbf{R}_\alpha\mathbf{r}$ are given by:

$$\mathbf{J}_\mathrm{i}(\mathbf{r}_r) = \mathbf{R}_\alpha\, \mathbf{J}_\mathrm{i}(\mathbf{r}). \tag{42}$$

Doing it in such a way, the solution in the main sector, along with the operators defined in (40) and (42), ensures the knowledge of the solution in the rest of the structure (that is in the whole 360° torus).

At this point, the calculation of the coefficients of **L**, **R**, and **V** is straightforward, and the evaluation of the matrices **L**, **R**, **V** are reduced only to the DOFs in main sector. Indeed, using a number of symmetries equal to Nsym and a number of rotations equal to Nrot, they have the following form:

$$L_{ij} = 2\mathrm{N}_\alpha \sum_{\mathrm{m}=0}^{\mathrm{N_{nrot}}} \sum_{\mathrm{j}=0}^{\mathrm{Nsym}} \frac{\mu_0}{4\pi} \int_{\mathrm{V_c}} \int_{\mathrm{V_v}} \frac{\nabla \times \mathbf{N}_\mathrm{i}(\mathbf{r}) \cdot \mathbf{R}_\alpha{}^{\mathrm{m}}\, \varepsilon^{\mathrm{j}}\, \mathbf{S}_\mathrm{n}{}^{\mathrm{j}} \nabla \times \mathbf{N}_\mathrm{j}\left(\mathbf{r}'\right)}{\left|\mathbf{r} - \mathbf{R}_\alpha{}^{\mathrm{m}}\mathbf{S}_\mathrm{n}{}^{\mathrm{j}}\mathbf{r}'\right|} \mathrm{d}\mathrm{r}'\mathrm{d}\mathrm{r}\,, \tag{43}$$

$$R_{ij} = -2\mathrm{N}_\alpha \int_{\mathrm{V_c}} \nabla \times \mathbf{N}_\mathrm{i}(\mathbf{r}) \cdot \times \nabla \times \mathbf{N}_\mathrm{j}(\mathbf{r})\; \mathrm{d}r, \tag{44}$$

$$V_{ij} = -2\mathrm{N}_\alpha \int_{\mathrm{V_c}} \nabla \times \mathbf{N}_\mathbf{i}(\mathbf{r}) \cdot \frac{\partial}{\partial \mathrm{t}} \mathbf{A}_\mathrm{s}(\mathbf{r}, \mathrm{t})\; \mathrm{dr}, \tag{45}$$

where $\mathrm{N}_\alpha = \frac{2\pi}{\alpha}$.

Limiting the computation to the main sector has two obvious advantages:

(i) The number of DOFs is reduced by a factor of $n_{tot} = \mathrm{Nsym} \times \mathrm{Nrot}$. This gives a huge gain in matrix storing, factorizing, and inverting.

(ii) The integration (43) can be seen as limiting the outer integral defined in (12) only to the main sector. Clearly, this reduces the matrix **L** assembly time.

We can summarize the symmetry approach, saying that the matrix **L** can be seen as the mutual inductance between DOFs. Without symmetry the field and source DOFs are the same. Using the symmetry approach, the field DOFs are only in the main sector, whereas the sources DOFs should be considered distributed all over the 360°, but they are implemented (by the means of the operator (40) and (42)) using only the solution contained in the main sector. Expanding (43) we have

$$\mathbf{L} = \mathbf{L}^{(1)} + \mathbf{L}^{(2)} + \ldots + \mathbf{L}^{(n_{tot})}, \tag{46}$$

where $\mathbf{L}^{(i-th)}$ can be seen as the effect on main sector DOFs due to the DOFs contained in the $i-th$ source rotated sector.

Now that we have explained how symmetry/rotation are implemented in CARIDDI code, we are ready to face the symmetry/rotation regarding the compression of the matrix **L**.

Let $\mathbf{L^{bc}}$ be the interaction matrix between two far boxes, **b** (field box) and **c** (source box). We denote with m and n the number of DOFs in the field and source box, respectively. In order to evaluate $\mathbf{L^{bc}}$, as said before, we need to consider the sources boxes **c** for all rotations and symmetry sectors. The resulting matrix will be obtained as superposition of

$$\mathbf{L^{bc}} = \mathbf{L^{bc}}^{\,(1)} + \mathbf{L^{bc}}^{\,(2)} + \ldots + \mathbf{L^{bc}}^{\,(n_{tot})}, \tag{47}$$

where $\mathbf{L^{bc}}^{\,(j)}$ with $\mathrm{j} = 1$ to n_{tot} represents the partial contribute of each sector/symmetry.

We have now two ways to apply the compression to $\mathbf{L^{bc}}$:

(1) Compressing each single interaction matrix $\mathbf{L^{bc}}^{\,(\mathrm{j})}$ appearing in (47).

(2) Summing the interactions matrices $\mathbf{L^{bc}}^{\,(\mathrm{j})}$ and after compressing the resulting matrix $\mathbf{L^{bc}}$.

In the first approach, being each term $\mathbf{L^{bc}}^{\,(\mathrm{j})}$ a low rank matrix, we can be applying to it the QR-MGS factorization. So, we can approximate equation (47) as:

$$\mathbf{L^{bc}} \approx \sum_{\mathrm{j}=1}^{\mathrm{n_{tot}}} \mathbf{Q}^{\mathbf{bc}(\mathrm{j})}\mathbf{R}^{\mathbf{bc}(\mathrm{j})}, \tag{48}$$

where $\mathbf{Q}^{\mathbf{bc}(\mathrm{j})}$ and $\mathbf{R}^{\mathbf{bc}(\mathrm{j})}$ have dimensions $m \times r_j$, and $r_j \times n$, respectively.

Keeping into account the (29), the computational cost (and memory) of the product $\mathbf{Q}^{\mathbf{bc}(\mathrm{j})}\mathbf{R}^{\mathbf{bc}(\mathrm{j})}\mathbf{x}$, is $(m+n) \times r_j$. Hence, the overall cost of the product $\mathbf{y} = \mathbf{L^{bc}x}$ is equal to:

$$c1 = \sum_{j=1}^{n_{tot}} (m+n) \times r_j = (m+n) \times n_{tot} \times r_m, \tag{49}$$

where $r_m = \frac{1}{n_{tot}} \sum_{j=1}^{n_{tot}} r_j$ is the medium rank.

The second approach is much simpler. The low rank nature of each $\mathbf{L^{bc}}^{\,(\mathrm{j})}$ implies that the overall matrix summation $\mathbf{L^{bc}}$ is low rank too. Although, it must be said that its rank generally could be greater than the $\mathbf{L^{bc}}^{\,(\mathrm{j})}$ rank in the sum. Hence, in the second proposed approach, we evaluate directly the QR-MGS factorization:

$$\mathbf{L^{bc}} \approx \mathbf{QR}. \tag{50}$$

Its computational cost is:

$$c2 = (m+n) \times r_{all}. \tag{51}$$

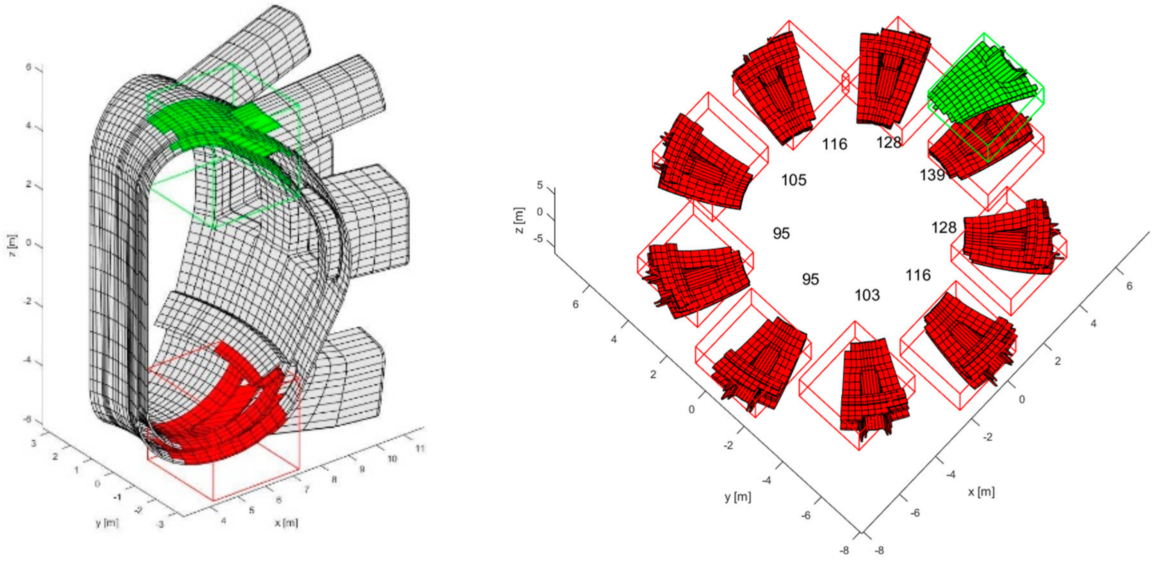
In order to compare the two approaches, we simply compute the ratio between (49) and (51) and we have

$$\frac{c1}{c2} = \frac{n_{tot} \times r_m}{r_{all}} \tag{52}$$

Extensive numerical experiments carried out on typical cases used in practical fusion devices, show that $r_m \approx r_{all}$. So as a matter of fact, $c2/c1$ results to be of the order of n_{tot}, and hence, the second approach is to be preferred.

For the sake of clarity, in the following, we report a simple example in which $n_{rot} = 9$, $n_{sym} = 1$ and, hence $n_{tot} = 9$. The main mesh sector is contained in $-20° < \varphi < 20°$.

In Figure 7, we show the mesh used, and we focus on a single far interaction between two boxes: the source (red) and field box (green). The number of DOFs in the source box m is equal 427, the number of the DOFs in the field box *n* is equal to *418*. Hence the matrix interaction $\mathbf{L^{bc}}$ size is equal to 418×427. To compute this matrix, we need to evaluate the interactions matrix between the nine replicated sources boxes (which are located in the nine sectors of the whole torus) and the field box (which is located in main sector). The rank of each box–box interaction is reported besides in Figure 7, next to the correspondent replicated mesh.



(**a**) (**b**)

Figure 7. (**a**) The main sector mesh ($-20° > \varphi < 20°$), and the source (red) and field (green) box, (**b**) For each sector (j = 1,9) we show the replicated mesh, and we report besides the rank of the matrix $\mathbf{L^{bc(j)}}$ between the source box (red) and the field box (green).

Using the first approach, we compress each single matrix $\mathbf{L^{bc}}^{\,(j)}$. It is worth noticing that, as expected, the rank decreases as the distance increases. Doing this in such a way, we obtain a computation cost equal to

$$c1 = (418 + 427) \times (139 + 128 + 116 + 105 + 95 + 95 + 103 + 116 + 128) = 866,125.$$

Using the second approach (i.e., making compression of overall matrix $\mathbf{L^{bc}}$) we obtain a rank equal to 134 and hence the computational cost is equal to

$$c2 = (418 + 427) \times 134 = 113,230.$$

Summing up, the computation gain using the second approach, defined as c2/c1, is equal to 7.6, proving the higher performance of the latter method with respect to the first one.

3.4. DOF-Based Approach

Another strategy we implemented is based on assuming the individual DOFs as the fundamental "objects" in Section 3.1.1.

We remark that in the ELEMENT-based approach, where the fundamental objects were the individual elements of the FEM (Finite Element Method), and the box–box interaction matrix is not a submatrix of the whole matrix $\mathbf{L}$. Therefore, interaction $\mathbf{L}^{*}_{\mathrm{ij}}$ for boundary edges will involve more than one box–box interaction matrices. When the number of this boundary DOFs is quite large, as for complex meshes of irregular shapes, we have a reduced efficiency of the overall compression method.

This loss of efficiency calls for a different approach, based on the DOFs rather than the individual elements of the FEM. Specifically, in the DOFs-based strategy, we assign an individual DOFs to a box, if its barycenter belongs to the box. Hereafter, we refer to this method as DOF-based method.

The specific changes in the QR-recursive algorithm are:

(1) In the Boxes Generation Algorithm, we assume the object to be individual DOFs.
(2) In Equation (24) we change the definition of the set E as follows:

$$\mathrm{E}(\mathbf{b}) \overset{\text{def}}{=} \text{is the set of elements having their DOFs support in the box } \mathbf{b}. \tag{53}$$

The rest of the algorithm is unchanged.

In force of the new definition of E, all element–element interactions are kept into account and, hence, matrix $\mathbf{L^{bc}}$, defined in (26), is a submatrix of the complete matrix $\mathbf{L}$. This will result in a higher efficiency on compression of the matrix, if compared to the ELEMENT-based approach. Figure 8 shows an example of subdivision of DOFs between two adjacent boxes. It is worth noting that the yellow elements are shared among DOFs in different boxes and hence, the related calculation is duplicated in the two different box–box interactions.

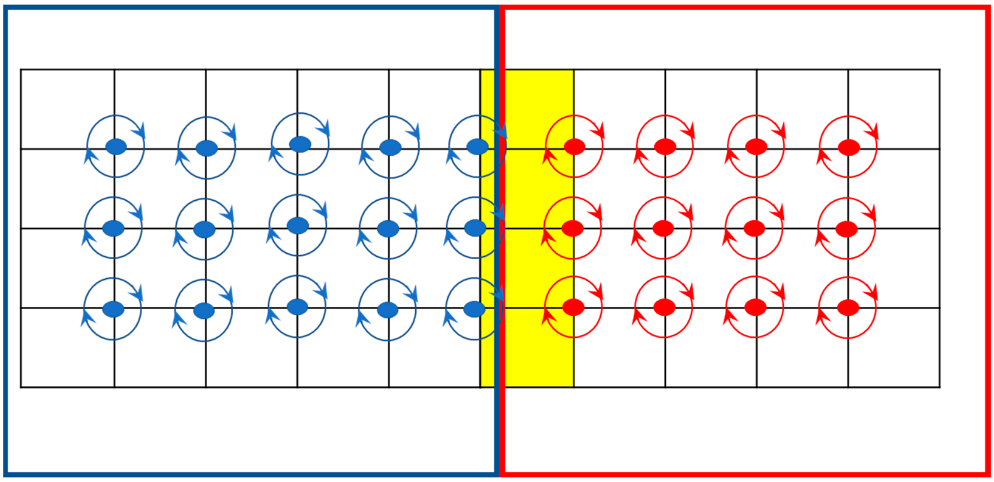


Figure 8. The DOF-based approach. Each box contains its own DOFs, here marked in red and blue. The elements belonging to the interface between the two boxes, require to be replicated in the box–box matrix assembly.

From a general perspective, the QR-recursive method requires the recursive subdivision of the elementary sources, i.e., the objects, which may be the DOFs, as a whole or the elements of the FEM.

The performances for these two choices will be compared in terms of:

- time required to assembly the compressed operator $\mathbf{L}$,
- the compression versus accuracy, signature.

Specifically, we expect the DOF-based approach to be more effective in term of Compression/Accuracy signature. Indeed, as previously explained, in the DOF-based strategy we have no replication of the $\mathbf{L}^*_{\mathrm{ij}}$ evaluating the box–box interaction matrices, whereas it does in the ELEMENT-based approach. Therefore, for a given accuracy, we expect the Compression of the DOF-based approach larger than one obtained using the ELEMENT-based approach. Consequently, for a prescribed accuracy, the DOF-based strategy will result in a shorter computational time when solving an individual time-step of the transient.

Summing up, in choosing the proper strategy, we have to consider two aspects: (i) the time required to assembly the compress operator and, (ii) the time needed for the solution of the transient. Regarding (i) the ELEMENT-based approach outperforms the DOF-based one, and vice versa, concerning (ii) the DOF-based approach outperforms the ELEMENT-based one. Therefore, in short-transients the assembly time is dominant and ELEMENT-based approach is more efficient. On the contrary, in long-transients the DOF-based approach is more effective.

3.5. Handling Small Size Boxes

In the present work, a relevant modification of boxes construction has been developed with respect to the previous work, which will result in increasing efficiency of the overall QR-recursive method.

As explained in (see Appendix A), the boxes subdivision is carried out until the number of objects (element or edge) in each box is smaller than a chosen threshold (*smin1*). During this process, a small number of objects may occur in the resulting childless boxes.

This is obvious and due to the fact that (i) we use a cube which is recursively equally subdivided in eight children and, (ii) the shape of the mesh (which is quite arbitrary) does not match the cube. As matter of fact, a fragmented number of objects could result in the childless, which might be very different from prescribed *smin1*.

These "small" childless have a drawback for all the interactions between and with the other boxes. Indeed, the QR factorization is quite inefficient, because the resulting rank will be approximately equal to the minimum matrix dimension (which is linked to the number of objects in the underlying box). So, for this kind of interaction matrix, no compression is achieved, and the overall efficiency of the method will be degraded.

To face the problem, one could think of changing the shape of the boxes, trying to match somehow the box shape with the one of the current meshes. This approach is not viable, because the cube is the simplest shape which is able to tessellate the 3D space and ensures well-separate box interactions. This is due to the underlying integral operator which is $1/r$ decaying, being r the distance between the source and field point.

In order to overcome this limitation, we fuse the childless boxes having a number of objects (elements or DOFs) smaller than a prescribed threshold, into larger boxes. The larger box for this fusion is selected by considering both the geometrical distance from the small box and the total number of resulting objects. The latter criterion is to better balance the computational load among processes. The detail of the fusion algorithm is provided in Appendix C.

3.6. Handling Electrodes

Electrodes are of great interest in fusion application devices (e.g., in computing the halo currents [29]). As explained in Section 2.3, the CARIDDI code makes use of electrodes to impose simultaneously symmetric periodic boundaries and prescribed currents at the boundaries. So, it is important to understand the impact, from the numerical point of view, of the electrodes on the solution of the overall algebraic system.

As we have shown in Section 2.3, in presence of electrodes, the resulting algebraic system is given by (9) and discretized in time domain as in (22).

As a matter of principle, two simple approaches could be used to solve such an algebraic system:

1. Apply an iterative method to solve the augmeted system

$$\begin{bmatrix} \mathbf{L}+\Delta\mathrm{t}\,\mathbf{R} & \Delta\mathrm{t}\,\mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{I}^{(\mathrm{k}+1)} \\ \boldsymbol{\phi}^{(\mathrm{k}+1)} \end{bmatrix} = \begin{bmatrix} \mathbf{L}\mathbf{I}^{(\mathrm{k})}+\mathbf{V}_{\mathrm{S}}^{(\mathrm{k}+1)}-\mathbf{V}_{\mathrm{S}}^{(\mathrm{k})} \\ \mathbf{M}^{(\mathrm{k}+1)} \end{bmatrix}. \tag{54}$$

2. Elimining $\mathbf{I}^{(\mathrm{k}+1)}$ by substituing the first equation of the system (22) into the second one. This subsitution yields the resulting linear system

$$-\Delta\mathrm{t}\,\mathbf{B}\,(\mathbf{L}+\Delta\mathrm{t}\,\mathbf{R})^{-1}\mathbf{B}^{\mathrm{T}}\boldsymbol{\phi}^{(\mathrm{k+1})}=\mathbf{M}^{(\mathrm{k+1})}-(\mathbf{L}+\Delta\mathrm{t}\,\mathbf{R})^{-1}\Big(\mathbf{L}\mathbf{I}^{(\mathrm{k})}+\mathbf{V}_{\mathrm{S}}^{(\mathrm{k+1})}-\mathbf{V}_{\mathrm{S}}^{(\mathrm{k})}\Big). \quad (55)$$

Both these methods have heavy limitations.

In the first approach, the main problem is related to the preconditioner. Indeed, it is not possible to use the resistive preconditioner $\mathbf{P}=\begin{bmatrix}\mathbf{R} & 0\\ 0 & 0\end{bmatrix}$, since it is trivially singular. Other choices for preconditioners are not trivial.

In the second approach, the evaluation of the matrix $(\mathbf{L}+\Delta\mathrm{t}\,\mathbf{R})^{-1}\mathbf{B}^{T}$ results in a large computational burden because it requires a number of inversions equal to the number of columns of $\mathbf{B}^{\mathrm{T}}$, i.e., equal to N_E+N_p. Indeed, in practical fusion device applications, this number is of the order of ten thousand. In addition, $(\mathbf{L}+\Delta\mathrm{t}\,\mathbf{R})^{-1}\mathbf{B}^{\mathrm{T}}$ is definitively computed by means of iterative inversion, introducing an error on the resulting matrix, which may badly affect the accuracy of the overall solution.

We implement an efficient method that guarantees both low computation costs and accuracy, described in the following. We make use of the following change of variable

$$\mathbf{I}=\mathbf{K}\mathbf{Z}+\mathbf{I}_0, \quad (56)$$

where the columns of $\mathbf{K}$ span the null space of the matrix $\mathbf{B}$, It can be shown that is possible to obtain a $\mathbf{K}$ sparse matrix. $\mathbf{I}_0$ is an arbitrary solution of the second equation in (22). For instance, one can chose for each instant

$$\mathbf{I}_0^{(\mathrm{k+1})}=\mathbf{B}^{+}\mathbf{I}_{\mathrm{h}}^{(\mathrm{k+1})}, \quad (57)$$

where $\mathbf{B}^{+}$ is the Moore–Penrose pseudoinverse of $\mathbf{B}$. Moreover, the computational cost for evaluating (57) is low, because $\mathbf{B}^{+}$ can be computed once and for all.

Using (56), system (22) can conveniently be rewritten as

$$\mathbf{K}^{\mathrm{T}}(\mathbf{L}+\Delta\mathrm{t}\mathbf{R})\mathbf{K}\mathbf{Z}^{(\mathrm{k+1})}=\mathbf{N}^{(\mathrm{k+1})}, \quad (58)$$

where $\mathbf{N}^{(\mathrm{k+1})}$ is the known term:

$$\mathbf{N}^{(\mathrm{k+1})}=\mathbf{K}^{T}\Big(\mathbf{L}\mathbf{Z}^{(\mathrm{k})}+\mathbf{V}_{\mathrm{S}}^{(\mathrm{k+1})}-\mathbf{V}_{\mathrm{S}}^{(\mathrm{k})}\Big)\mathbf{K}. \quad (59)$$

By setting

$$\mathbf{A}^{'}=\mathbf{K}^{\mathrm{T}}(\mathbf{L}+\Delta\mathrm{t}\mathbf{R})\mathbf{K}, \quad (60)$$

the resulting system is

$$\begin{gathered}\mathbf{A}^{'}\mathbf{Z}^{(\mathrm{k+1})}=\mathbf{N}^{(\mathrm{k+1})}\\ \mathbf{Z}^{(0)}=0,\mathbf{I}_0^{(0)}=\mathbf{B}^{+}\mathbf{I}_{\mathrm{h}}^{(0)}\\ \mathrm{k}\geq 1\end{gathered} \quad (61)$$

where the second equation in (61) condition matches the initial prescribed current distribution at the electrodes.

We note that the matrix $\mathbf{K}^{T}\mathbf{R}\mathbf{K}$ is sparse and positively defined. This matrix can be used as new preconditioner and, similar to $\mathbf{R}$, can be easily factorized and applied.

For the solution of Equation (61) we can still make use of the QR-recursive method, in the same form developed without the electrodes. Indeed, the matrix product $\mathbf{A}^{'}\mathbf{X}$ can be easily obtained by the same ordinary product (namely $\mathbf{Ax}$) using a pre- and post-multiplication for the sparse matrix $\mathbf{K}$.

4. Results

4.1. Testcase #1

The system consists of a sector (one-ninth) of the ITER magnet system structures, which is composed by 18 Nb3Sn Toroidal Field (TF) coils, 6 Poloidal Field (PF) NbTi coils, 6 Central Solenoid (CS) Nb3Sn Coils, and by their supporting structures [39–41]. There are also 18 superconducting Correction Coils located between the TF and PF coils whose aim is to correct field errors due to manufacturing tolerances. All the coils are made of superconducting strands cooled with supercritical Helium flowing at a temperature of 4.5 K. During the ITER operation, the superconducting coils shall work at a temperature lower than their critical temperature in order not to lose their superconducting state; therefore, it is very important to assess the thermal loads acting on the structures and make sure they are not compromising the functioning of the system.

The thermal loads on the superconducting coils are mainly due to the neutron heating coming from the plasma. AC loss in the superconducting cables and eddy currents in the surrounding metallic structures are due to a changing magnetic field. Several analyses have been done in the past with the direct method of the code CARIDDI to compute eddy currents and Joule losses in the cold metallic structures during normal operation plasma scenarios and during instability events such as disruptions.

A 40-degree Finite Element Model of the ITER magnet system has been built, the model includes the metallic structure of two TF coils (TF Stainless Steel case and cover and radial plates), the supports of the PF coils, the Correction coil Rails and supports, as well as the vacuum vessel thermal shield, cryostat, and vacuum vessel (see Figure 9). The mesh has 117,371 elements and 203,084 nodes. The mesh number of DOFs is 154,514. In this testcase no injected currents are present, hence N_E is equal to zero, and periodic symmetric boundaries are applied: the number of boundary faces (i.e., N_p) present at π_α are 2304 (see Section 2.3). All the components are made of Stainless Steel 316 LN working at 4.5 K, except the supports of the CC coils where some copper (Cu) sheets used as a thermal sink are present; the electrical resistivity is assigned to each component as a constant value evaluated at its working temperature.

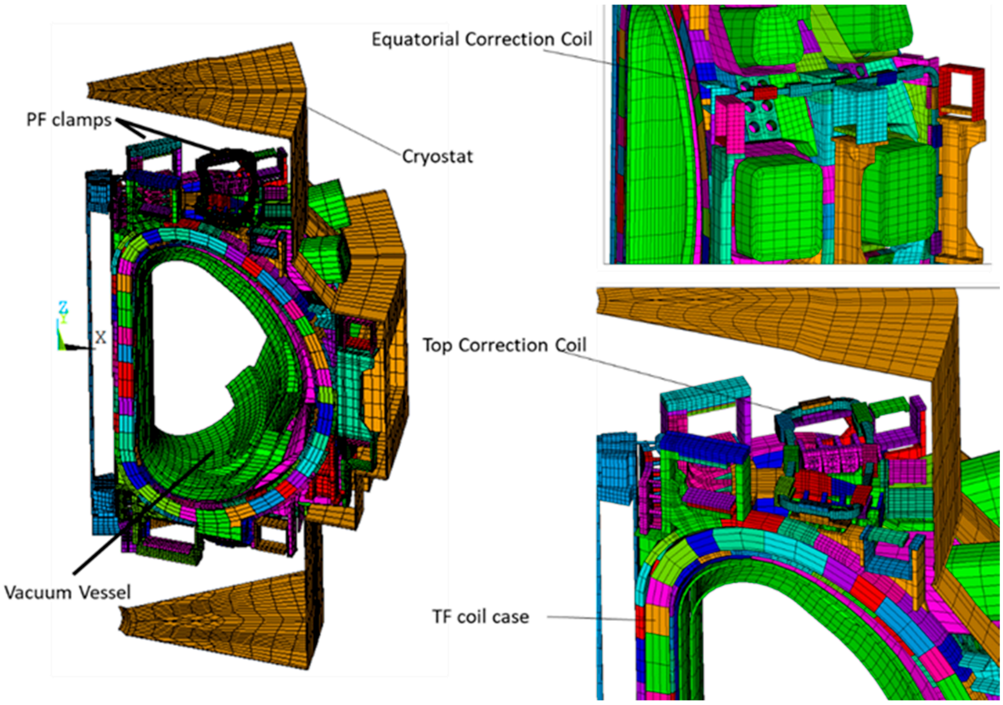
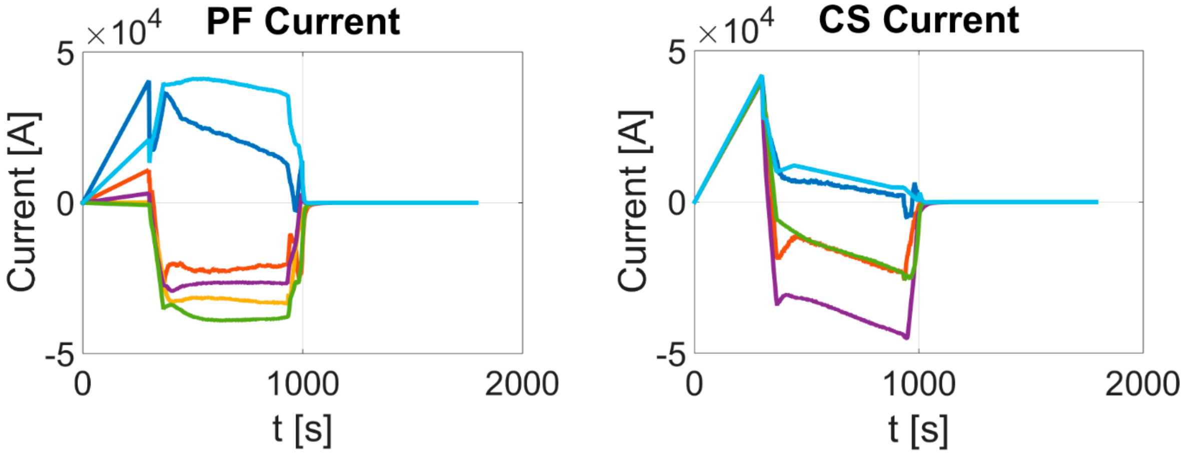


Figure 9. Testcase#1 TF magnet system structure.

During normal operation, eddy currents can develop in the metallic structures due mainly to ramp ups and ramp downs of the plasma current. The TF coils are fed with a constant current; therefore, they do not contribute to the changing magnetic field. The other coils such as the PF coils, the CS, and the plasma itself are axisymmetric sources that are not meshed but they are represented as external sources.

The scenario that has been analyzed is identified as *DINA-2016* (DINA is the ITER reference plasma equilibrium and evolution code used to analyze the plasma behavior during normal operation and instabilities [42]).

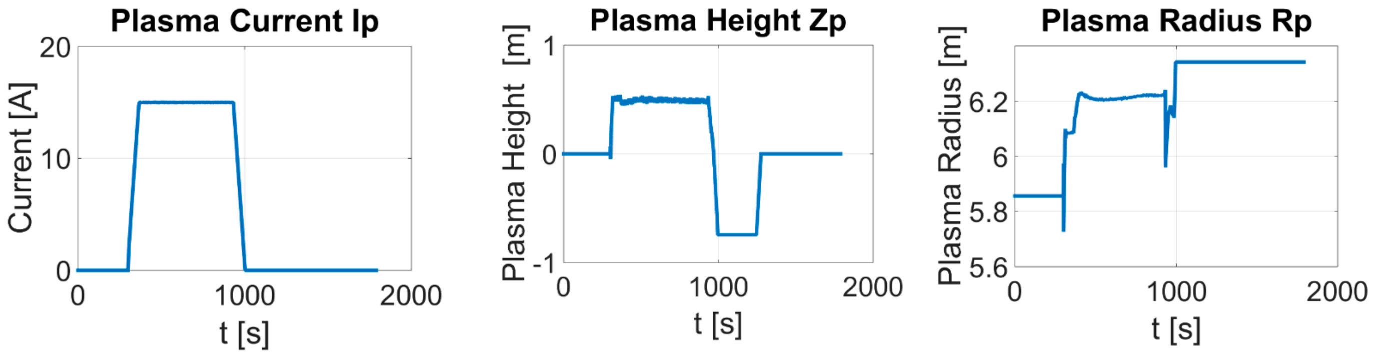
The currents flowing in the each of the six PF and CS coils during the *DINA 2016* scenario is shown in the following Figure 10.



(**a**) (**b**)

Figure 10. Testcase#1. The PF (**a**) and CS (**b**) excitation current during the plasma event (different colors are used for different coils).

In Figure 11, we report the evolution of plasma currents (Figure 11a), the radius (Figure 11b), and the height (Figure 11c) of the barycenter of the plasma during the scenario. It can be already noted from these signals that the plasma height is affected by a high-frequency noise due to the plasma control.



(**a**) (**b**) (**c**)

Figure 11. Testcase#1. The plasma current (**a**), the radius (**b**), and the height (**c**) of the barycenter of the plasma during the scenario.

As an example of results obtained with the CARIDDI code, the power and the energy dissipation in the TF coil case system is shown in Figure 12. As can be noted, the energy is also increasing during the plasma flat top (time interval from 300 s to 1000 s) due to the mentioned control noise.

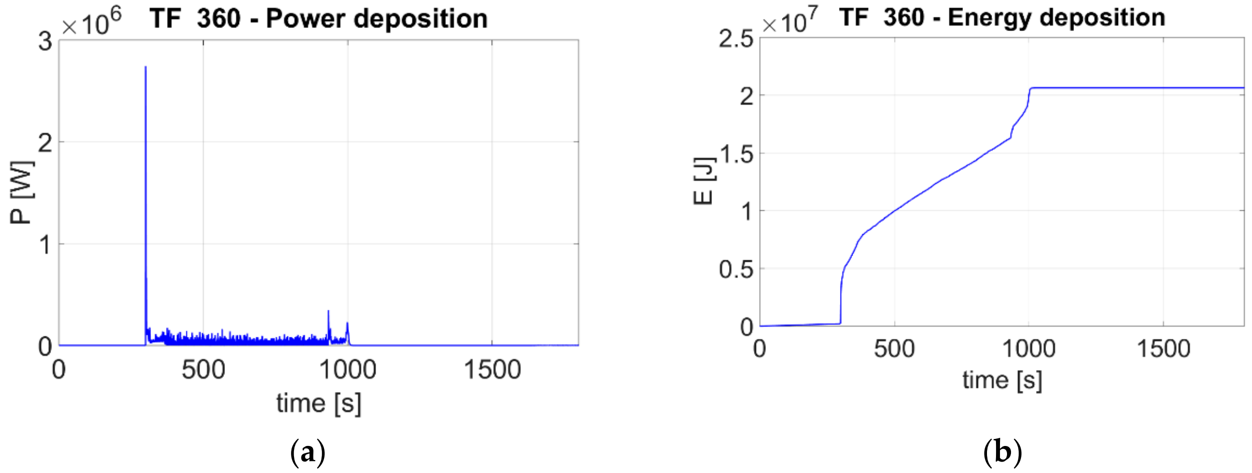


Figure 12. Testcase#1: The power (**a**) and the energy dissipation (**b**) in the TF coil case system.

We use this case to check the performances of the new features introduced for the QR-recursive method. The number of DOFs is limited, to allow the validation of the accuracy of the method against the direct solution.

The finite element mesh uses both hexahedral and pentahedral elements: mixed element types are very usual in meshing generators. To prove the accuracy of the QR-recursive method, we show in Figure 13 the ohmic loss dissipated in all the passive structures during the event.

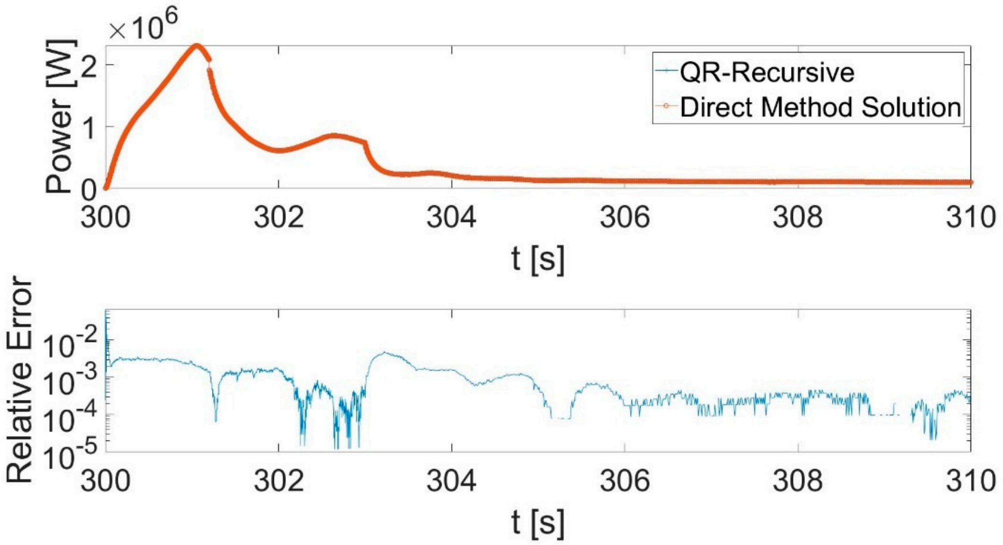


Figure 13. Testcase#1: (**Top**) the power dissipated during the transient in all passive structures. (**Bottom**) the relative error of the QR-recursive based method with respect to the Direct computation.

In order to evaluate the efficiency of the preconditioner **R** together with the interpolation scheme (see Section 3.2), in Figure 14 we show the number of iterations required to converge to the relative error equal to 1×10^{-5}, versus time. Please note that the initial guess is effective: indeed, the total number of iterations is reduced by a factor of about four compared to the approach without initial guess estimation.

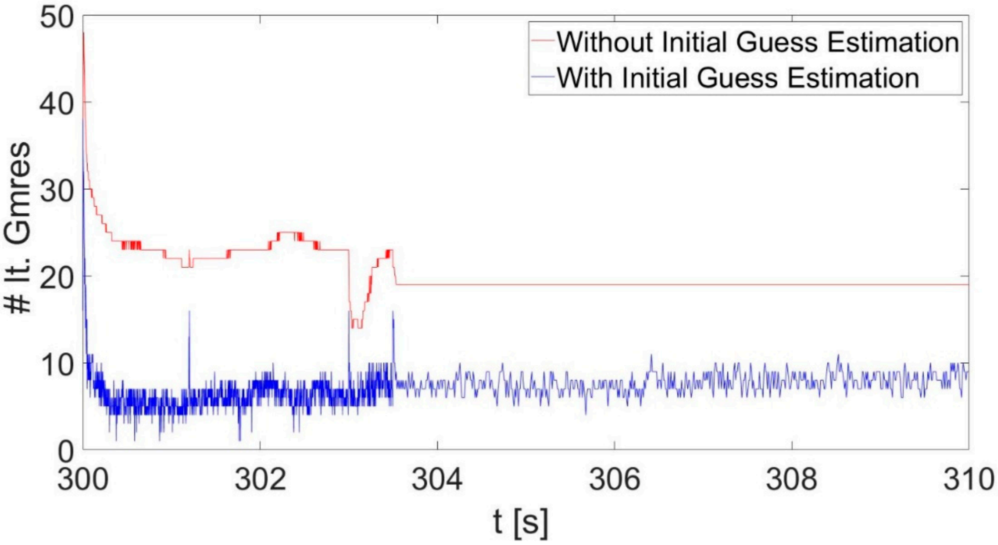


Figure 14. Testcase#1. The number of GMRES iterations required to converge to a relative error equal to 1×10^{-5}, versus time, with and without initial guess estimation.

In order to evaluate the performances of the iterative method, we plot the achieved compression versus the accuracy. Specifically, the compression is defined as:

$$\text{Compression} = \frac{\text{Dimension of full matrix } (\mathbf{A})}{\text{Dimension of the compressed matrix } (\mathbf{A}_{\text{qr}})}. \tag{62}$$

It measures the efficiency of the overall compression method.

The memory required to store either the compressed or the original matrices is proportional to the number of required multiplications to evaluate the matrix-by-vector product. Hence, this parameter is indeed the gain of computational cost.

The accuracy is defined as follows. First, we compute the fully populated matrix **L** and, hence, the dynamical matrix **A**

$$\mathbf{A} = \mathbf{R}\,\Delta\mathrm{t} + \mathbf{L}. \tag{63}$$

By setting vector **x** equal to all one's, we compute the product without error

$$\mathbf{P}_{\mathrm{c}} = \mathbf{A}\mathbf{x}\,. \tag{64}$$

Using the QR-recursive method we compute the approximated product:

$$\mathbf{P}_{\mathrm{a}} = \mathbf{A}_{\mathbf{qr}}\mathbf{x}\,. \tag{65}$$

The accuracy is defined as

$$\text{Accuracy} = \frac{\|\mathbf{P}_{\mathrm{c}} - \mathbf{P}_{\mathrm{a}}\|}{\|\mathbf{P}_{\mathrm{c}}\|}\,. \tag{66}$$

Figure 15 highlights the overall performances of the various proposed techniques for the QR-recursive method. Moreover, it proposes a comparison with a state-of-the-art method based on the hierarchical matrices (H-matrices) [25,43–45] and implemented in the state-of-the-art library. To this end, we made for the CARIDDI code the interface for Hlibpro, which is a library implementing H-matrix algorithms for the approximation of a dense matrix arising from an integral formulation. See [46].

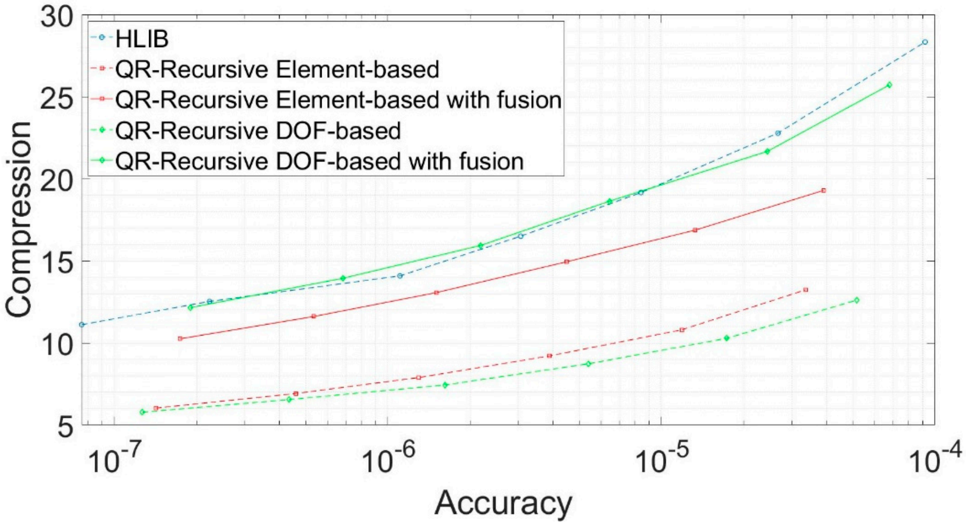


Figure 15. Testcase#1. The Compression versus the Accuracy for various proposed techniques.

Figure 15 clearly proves that DOF-based with fusion is the most effective compression method among those proposed in this paper and that it is in line with the state-of-the-art one (H-matrices). Moreover, the boxes' small size fusion is effective both for DOF-based and for ELEMENT-based techniques.

From a computational point of view, to improve the efficiency, it is also mandatory to balance the load among the processes in parallel computation. The coding strategy implemented with these recursive QR-recursive approaches is in line with this goal. In Figure 16 we report the Memory/Computation ($\mathbf{L}_{\mathrm{near}} + \mathbf{L}_{\mathrm{far}}$) load versus the MPI processes involved in the computation. Figure 16 reports proper balancing among different MPI processes.

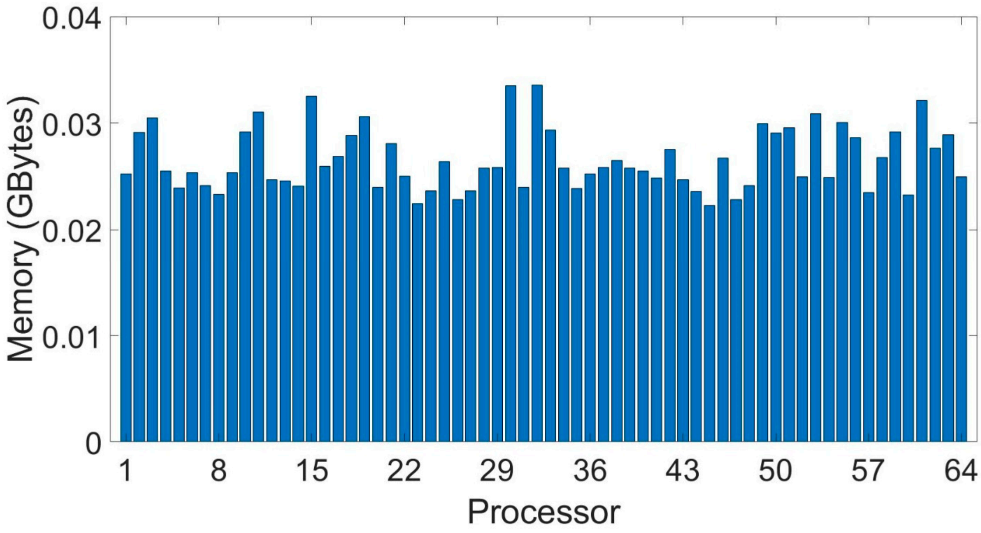


Figure 16. Testcase#1. The Memory/Computation load versus MPI processors used in computation.

4.2. Testcase #2

The second test case we present is again related to the ITER device. In particular, we analyse a fast transient due to an abrupt loss of plasma magnetic and kinetic energy—a so-called *disruption*. In particular, following the loss of vertical position control, the plasma drifts downwards with its physical parameters largely unchanged, until at the time instant 0.28 s the Thermal Quench (TQ) takes place, i.e., the loss of kinetic energy of the plasma in 1 ms. Immediately after, the Current Quench (CQ) occurs, i.e., the loss of plasma current and magnetic energy exponentially with a time constant of 16 ms, hence up to around 0.4 s, when the plasma current is negligible [29]. It is assumed that the plasma keeps axisymmetry throughout the whole time evolution; however, this is not an intrinsic limitation of the proposed numerical approach, which can easily deal also with non-axisymmetric plasmas. The consequences of disruptions may be particularly challenging for both the plasma facing components and for the conducting structures, hence requiring a careful numerical analysis of such events. In particular, the interaction of current density induced in the structures with the magnetic field present in the device gives rise to electromagnetic loads (Lorentz force density), whose effects must be carefully evaluated through a dedicated subsequent mechanical and structural analysis. In the case of an axisymmetric plasma event, the main net force is usually in the vertical direction, but in non-axisymmetric cases horizontal (sideways) forces may also be present and may pose a specific threat to the integrity of the structures. In addition, direct plasma-wall contact may give rise to additional problems, mainly on plasma-facing components, such as the injection of halo currents, the possibility of electric arches between adjacent structures, the occurrence of significant heat and particle loads, and the damage produced by the impact of suprathermal (runaway) electrons.

The number of mesh nodes is 504,879, and the number of elements is 346,160. In addition, in this testcase no injected currents are present, hence N_E is equal to zero, and periodic symmetric boundaries are applied; the number of boundary faces (i.e., N_p) present at π_α are 4563 (see Section 2.3). The resulting number of DOFs is 559,574 (see Figure 17). The present test case is hence complementary to the previous one. First of all, it allows us to evaluate the performances of the QR-recursive compression on a very fast transient occurring on the milliseconds time scale, as opposed to the previous case, where the time scale is of the order of thousands of seconds. Secondly, the number of DOFs is so high that other methods, used for reference in the previous test case (direct, non-parallel Hlibpro), cannot be applied. As a consequence, no detailed analyses on the numerical performances of the method are reported. To have a term of reference also in this case, a coarse mesh, featuring eight times less elements on exactly the same geometry, has been developed and solved with the direct method.

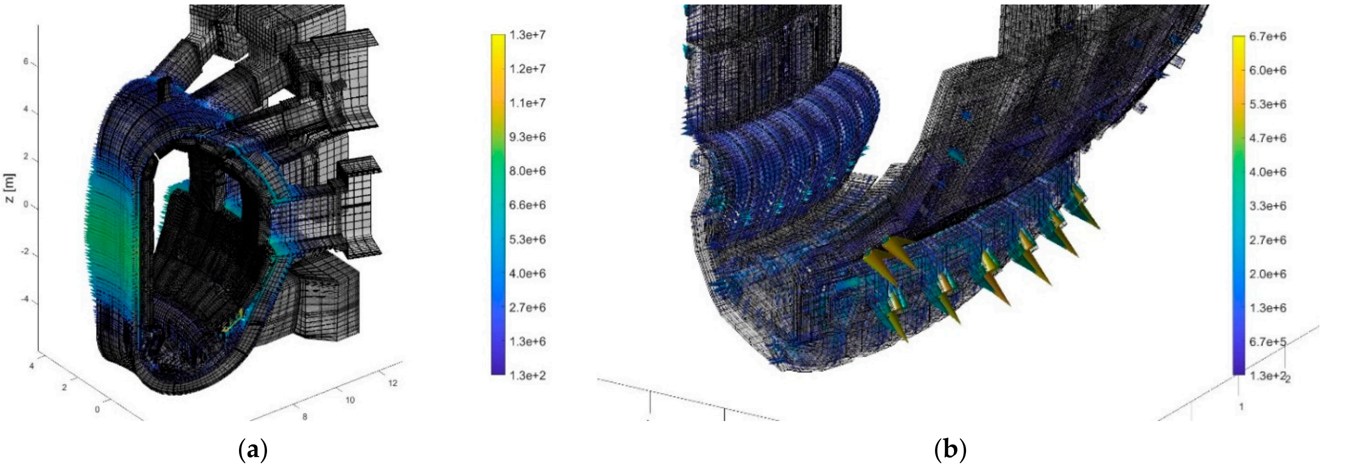
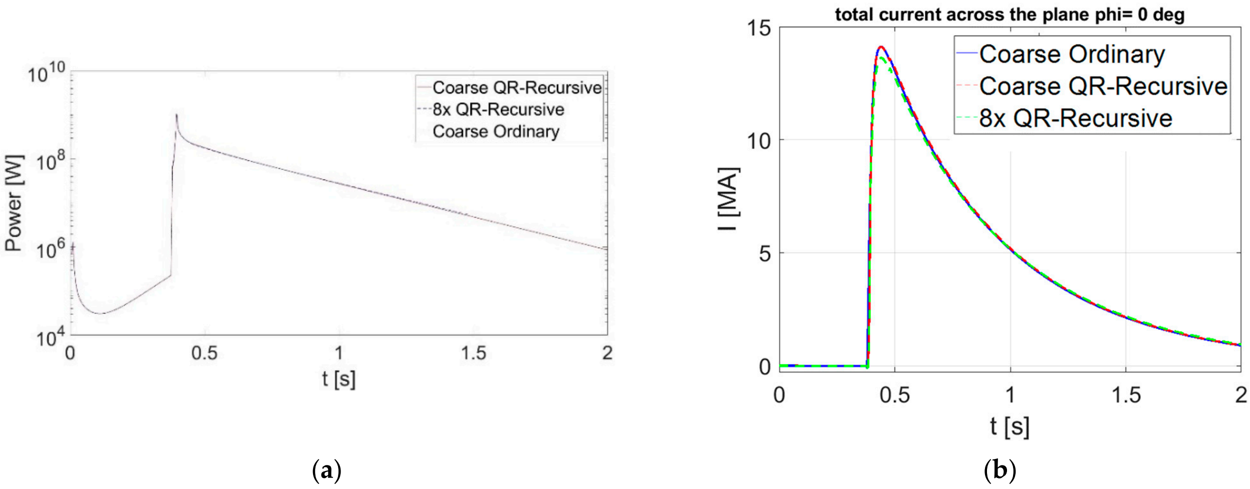


Figure 17. Testcase#2. Mesh (**a**) and pictorial representation of the current density (**b**).

Our approach also provides satisfactory results on this very challenging test case, hence showing the potential of our methods on cases out of reach for other techniques. Specifically, Figure 17 shows some pictorial representations of the current density patterns on specific components of the tokamak, while Figure 18 reports a good agreement with the reference coarse mesh.



(**a**) (**b**)

Figure 18. Testcase#2. The overall ohmic loss (**a**) and the total toroidal current (**b**) versus time.

From the numerical point of view, the compression gain achieved is about 15 and, using 64 MPI processes, the time required to assemble the compressed operator is 10 h and 20 min. The total transient time required to simulate the transient is about 4 days. The Server used in computation in our lab is based on 2 × AMD Epyc 32-Core 7452, 1 TB RAM.

The number of GMRES iterations required is reported in Figure 19. The number always keeps very low with respect to the number of unknowns, hence showing a satisfactory behaviour of the preconditioner, even in the time window around 0.3–0.4 s, where the fast transient occurs.

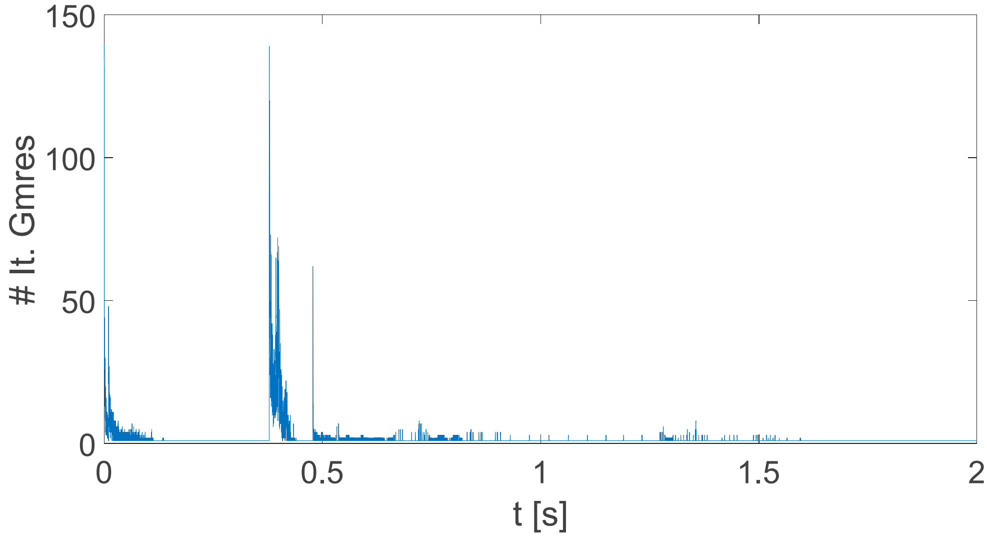


Figure 19. Testcase#2. The number of iterations required by GMRES to converge to a relative tolerance is 1×10^{-5}, versus time.

Finally, in Figure 20, we report the Memory/Computation load versus the MPI processes involved in the computation. The good load balancing proves the efficency of the overall code parallelization.

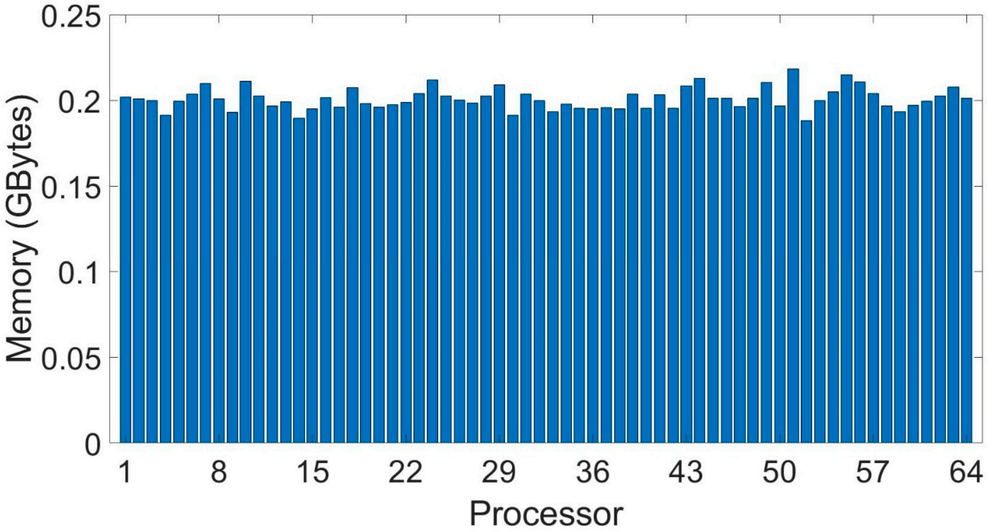


Figure 20. Testcase#2. The Memory/Computation load versus MPI processors used in computation.

5. Discussion and Conclusions

In this paper we have presented a comprehensive extension of a QR-recursive compression technique applied to the CARIDDI code. A number of improvements (the treatment of rotations and symmetries, and the introduction of electrodes for halo current injections) and optimizations (DOF-based algorithm, small boxes treatment) have been implemented with respect to the past.

The numerical performances of the proposed technique are remarkably in line with those of state-of-the-art methods such as Hlibpro, and the results are successfully compared with exact (direct) methods. In addition, the features of the proposed QR-recursive compression technique allow the favourable scaling of its properties to cases where direct methods and other state-of-the-art techniques such as Hlibpro cannot be applied, due to the high number of discrete unknowns. The test cases reported in the paper clearly demonstrate this point.

With the implementation of this QR-recursive technique, CARIDDI can be hence considered as one of the reference codes for the numerical electromagnetic analysis of fusion devices with realistic geometries.

Author Contributions: Conceptualization, S.V., G.R., F.V., A.T., A.G.C., V.S. and F.C.; Data curation, F.V., A.G.C., V.S. and F.C.; Formal analysis, G.R.; Investigation, S.V., G.R. and A.T.; Methodology, G.R. and A.G.C.; Software, S.V. and G.R.; Validation, S.V., G.R., F.V. and A.T.; Writing—original draft, S.V. and G.R.; Writing—review & editing, F.V., A.T., A.G.C., V.S. and F.C. All authors have read and agreed to the published version of the manuscript.

Funding: The work was partially supported by the Projects: “Smart Distributed Systems” under the program “Dipartimenti di Eccellenza 2018–2022”, and the program PRIN, grant # 20177BZMAH, funded by MIUR, Italian Ministry of University and Research, and by the project “FH2CPS”, 5th Marconi Fusion cycle.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors wish to thank Riccardo Torchio for his support to interface HLIB library to CARIDDI code.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Algorithm A1. Boxes Generation

```
Input
    Choose the "objects" (elements or DOF)
     $smin1$ is the minimum number of objects in a box
Output
    $BoxSet, FatherSet, ChildrenSet$
$BoxSet \overset{\text{def}}{=}$ is the set of all not empty boxes
$ChildlessSet \overset{\text{def}}{=}$ is the set of all box having a number of objects number of objects less equal to $smin1$
$FatherSet(b)$ returns the father of the box $\mathbf{b}$
$ChildrenSet(b, 1:8)$ returns the eight children of the box $\mathbf{b}$
Create first box $\mathbf{b0}$ // the smallest cube containing the mesh $ChildLessSet = \{\varnothing\}$
 $ChildrenSet = \{\varnothing\}$
 $BoxSet = \{\mathbf{b}0\}$
level = 0 // current level starting from 0
ParentBoxesList = $\{\mathbf{b}0\}$ // temporary
$ChildrenList = \{\mathbf{b}0\}$ // temporary
While $ParentBoxesList \neq \{\varnothing\}$ // Loop util there are still fathers
  level = level + 1 // current level
$ChildrenList = \{\varnothing\}$ //current level children list
For $\mathbf{F} \in$ ParentBoxesList // loop over previous level father box $\mathbf{F}$
 $NumObjectsF = \#(\mathbf{F})$ // set the number of objects in box $\mathbf{F}$
 If $(NumObjectsF > smin1)$
      // Subdivide father $\mathbf{F}$ in eight children $\mathbf{C}$
      For $i$ = 1,8
        $\mathbf{C}$ = $subdivide(F, i)$ // get $i$-th children $\mathbf{C}$
        $NumObjects = \#(\mathbf{B})$ // get the number of objects in box $\mathbf{B}$
        If $(NumObjects > 0)$
          // Update Sets
          $Children(\mathbf{F}, i) = \mathbf{C}$ // $\mathbf{C}$ is the son of $\mathbf{F}$
          $FatherSet(Box) = \mathbf{F}$ // $\mathbf{F}$ is the father of $\mathbf{C}$
          $BoxSet = \{BoxSet \cup \mathbf{C}\}$// $\mathbf{C}$ will be next level fathers
          $ChildrenList = \{ChildrenList \cup \mathbf{C}\}$
        End if
      End for
    Else
      // father F is childless
      $ChildLessSet = \{ChildLessSet \cup \mathbf{F}\}$
    End if
End while
 // Generation Update: current sons will be next level fathers
$ParentBoxesList = ChildrenSet$
End Algorithm
```

Appendix B

$\mathcal{L}_{\text{near}}(\mathbf{b})$, $\mathcal{L}_{\text{far}}(\mathbf{b})$ analytic definition

Given a field box **b**, in [20] we define four lists. Here we reported their definition for completeness.

- $\mathbf{L}_1(\mathbf{b})$ is the set of boxes made by **b** and all childless boxes adjacent to **b**.
- $\mathbf{L}_2(\mathbf{b})$ is the set of boxes not adjacent to **b** at the same level of **b** and are well separated by **b.**
- $\mathbf{L}_3(\mathbf{b})$**,** for a childless box **b,** is the set of descendent of **b**'s colleagues that are not adjacent to **b**, but whose parent boxes are adjacent to **b**.
- $\mathbf{L}_4(\mathbf{b})$ is the set of **c** such that $\mathbf{b} \in \mathbf{L}_3(\mathbf{c})$.

Using these four lists, we can define

$$\mathcal{L}_{\text{near}}(\mathbf{b}) = \mathbf{L}_1(\mathbf{b}) \cup \ \mathbf{L}_4(\mathbf{b})$$

$$\mathcal{L}_{\text{far}}(\mathbf{b}) = \mathbf{L}_2(\mathbf{b}) \cup \ \mathbf{L}_3(\mathbf{b})$$

In Figure A1**,** for a given field box **b,** we give two examples showing how these various lists are shaped.

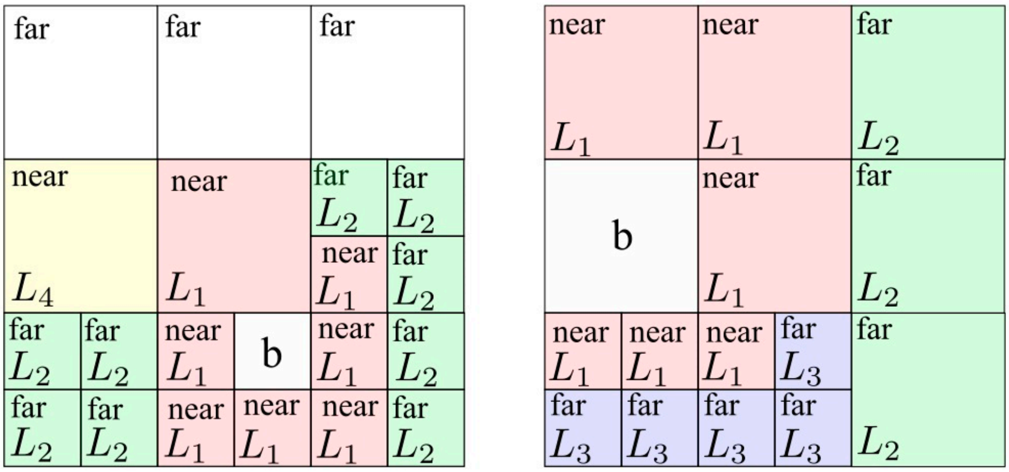


Figure A1. The four lists used to define $\mathcal{L}_{\text{near}}(\mathbf{b})$, $\mathcal{L}_{\text{far}}(\mathbf{b})$ associated for a given field box **b**. The sources boxes are depicted as: (i) pink and yellow for near interactions, and (ii) green and violet for the far interactions. Note that in left figure there are some empty source boxes. These boxes refer to the previous level far box–box interaction (indeed with the father of **b**). So actually, they are not present in $\mathcal{L}_{\text{far}}(\mathbf{b})$ list.

Appendix C

Algorithm A2. Small Boxes Fusion

```
Input
    $smin2$ is the minimum number of objects
    $fsmin$ $(> 1)$ is a parameter controlling maximum distance
                        $ChildLessSet \overset{\text{def}}{=}$ is the set of the childless boxes
Output
    $ListObject(\mathbf{b}) \overset{\text{def}}{=}$ is the set of objects in the box $\mathbf{b}$
$ListRetain = \{\varnothing\}$ // $\overset{\text{def}}{=}$ is the set of the boxes to be retained
$ListToFuse = \{\varnothing\}$ // $\overset{\text{def}}{=}$ is the set of the boxes to be fused
For $\mathbf{b} \in ChildLessSet$ // loop over all childless boxes
  $ObjectsNumber = \#ListObject(\mathbf{b})$ // get number of objects in the box b
  If $ObjectsNumber > smin2$
      $ListRetain = \{ListRetain \cup \mathbf{b}\}$ // put b in list of boxes to be retained
  Else
    $ListToFuse = \{ListToFuse \cup \mathbf{b}\}$ // put b into list of boxes to be fused
  End if
End for
For $\mathbf{b} \in ListToFuse$ // loop over the boxes to be fused
  //compute the distance from b
  For $\mathbf{x} \in ListRetain$
    d(x) = distance(b,x) // distance between b and x
  End
  // Compute the minimum distance
```

Algorithm A2. *Cont.*

```
    $\text{min}\_d = \min_{\mathbf{x}} d(\mathbf{x})$
    // set the maximum allowed distance
    $\text{max}\_d = \text{fsmin} \overset{\text{def}}{=} \text{min\_d}$
    // Create the set of all boxes whose distance from $\mathbf{b}$ is less than $\text{max}\_d$
    $Nearest = \{s \in ListRetain : d(\mathbf{s}) \leq \text{max}\_d\}$
    // get the number of objects for this set
    For $\mathbf{g} \in Nearest$
        $NumObjectsNearest(\mathbf{g}) = \#Nearest(\mathbf{g})$
    End for
    // Choose in the set $Nearest$, the one having the minimum number of objects
    $\mathbf{z} = \min_{\mathbf{g}} NumObjectsNearest(\mathbf{g})$
    // fuse all objects of the box $\mathbf{b}$ into the $\mathbf{z}$ box
    $ListObject(\mathbf{z}) = \{ListObject(\mathbf{z}) \cup ListObject(\mathbf{b})\}$
    // the objects of the box $\mathbf{b}$ must be deleted
    $ListObject(\mathbf{b}) = \{\varnothing\}$
End for
End of the algorithm
```

References

1. Villone, F.; Albanese, R.; Liu, Y.Q.; Portone, A.; Rubinacci, G. 3D effects of conducting structures on RWMs control in ITER. In Proceedings of the 34th EPS Conference, Warsaw, Poland, 2–6 July 2007.
2. Bossavit, A. *Computational Electromagnetism: Variational Formulations, Complementarity, Edge Elements*; Academic Press: Cambridge, MA, USA, 1998.
3. Biro, O.; Preis, K. An edge finite element eddy current formulation using a reduced magnetic and a current vector potential. *IEEE Trans. Magn.* **2000**, *36*, 3128–3130. [CrossRef]
4. Bossavit, A.; Verite, J. The "TRIFOU" Code: Solving the 3-D eddy-currents problem by using H as state variable. *IEEE Trans. Magn.* **1983**, *19*, 2465–2470. [CrossRef]
5. Kurz, S.; Fetzer, J.; Lehner, G.; Rucker, W.M. A novel formulation for 3D eddy current problems with moving bodies using a Lagrangian description and BEM-FEM coupling. *IEEE Trans. Magn.* **1998**, *34*, 3068–3073. [CrossRef]
6. Kameari, A. Transient eddy current analysis on thin conductors with arbitrary connections and shapes. *J. Comput. Phys.* **1981**, *42*, 124–140. [CrossRef]
7. Bossavit, A. On the numerical analysis of eddy-current problems. *Comput. Methods Appl. Mech. Eng.* **1981**, *27*, 303–318. [CrossRef]
8. Blum, J.; Leloup, C.; Dupas, L.; Thooris, B. Eddy current calculations for the Tore Supra Tokamak. *IEEE Trans. Magn.* **1983**, *19*, 2461–2464. [CrossRef]
9. Albanese, R.; Rubinacci, G. Integral formulation for 3D eddy-current computation using edge elements. *IEE Proc. A Phys. Sci. Meas. Instrum. Manag. Educ. Rev.* **1988**, *135*, 457–462. [CrossRef]
10. Albanese, R.; Rubinacci, G. Finite Element Methods for the Solution of 3D Eddy Current Problems. *Adv. Imaging Electron. Phys.* **1998**, *102*, 1–86. [CrossRef]
11. Roccella, R.; Boccaccini, L.; Meyder, R.; Raff, S. Assessment of EM loads on the EU HCPB TBM during plasma disruption and normal operating scenario including the ferromagnetic effect. *Fusion Eng. Des.* **2008**, *83*, 1212–1216. [CrossRef]
12. Testoni, P.; Albanese, R.; Lucca, F.; Roccella, M.; Portone, A.; Rubinacci, G.; Ventre, S.; Villone, F. Status of the EU domestic agency electromagnetic analyses of ITER vacuum vessel and blanket modules. *Fusion Eng. Des.* **2013**, *88*, 1934–1937. [CrossRef]
13. Zhu, J.; Bykov, V.; Pagani, I.; Lucca, F.; Wegener, L.; Bosch, H.-S. Specific features of eddy current analysis with ANSYS® for fast plasma current decay event in W7-X. *Fusion Eng. Des.* **2021**, *163*, 112136. [CrossRef]
14. Kwon, S.; Hong, S.-H. Estimation of electromagnetic loads including eddy and halo current on the major disruption for the K-DEMO divertor. *Fusion Eng. Des.* **2021**, *168*, 112592. [CrossRef]
15. Albanese, R.; Rubinacci, G.; Tamburrino, A.; Ventre, S.; Villone, F. A fast 3D eddy current integral formulation. *COMPEL Int. J. Comput. Math. Electr. Electron. Eng.* **2001**, *20*, 317–331. [CrossRef]
16. Phillips, J.R.; White, J.K. A precorrected-FFT method for electrostatic analysis of complicated 3-D structures. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **1997**, *16*, 1059–1072. [CrossRef]
17. Tamburrino, A.; Ventre, S.; Rubinacci, G. An FFT integral formulation using edge-elements for Eddy Current Testing. *Int. J. Appl. Electromagn. Mech.* **2000**, *11*, 141–162. [CrossRef]
18. Greengard, L.; Rokhlin, V. A fast algorithm for particle simulations. *J. Comput. Phys.* **1987**, *73*, 325–348. [CrossRef]
19. Cheng, H.; Greengard, L.; Rokhlin, V. A Fast Adaptive Multipole Algorithm in Three Dimensions. *J. Comput. Phys.* **1999**, *155*, 468–498. [CrossRef]

20. Rubinacci, G.; Ventre, S.; Villone, F.; Liu, Y. A fast technique applied to the analysis of Resistive Wall Modes with 3D conducting structures. *J. Comput. Phys.* **2008**, *228*, 1562–1572. [CrossRef]
21. Maffucci, A.; Rubinacci, G.; Tamburrino, A.; Ventre, S.; Villone, F. Fast Low-Frequency Impedance Extraction using a Volumetric Three-Dimensional Integral Formulation. In Proceedings of the 23rd Annual Review of Progress in Applied Computational Electromagnetics (ACES 2007), Verona, Italy, 19–23 March 2007.
22. Van Loan, C.F.; Golub, G.H. *Matrix Computations*, 3rd ed.; Johns Hopkins University Press: London, UK, 1996.
23. EFDA, European Fusion Development. The ITER Project. Available online: www.iter.org (accessed on 15 September 2019).
24. Rubinacci, G.; Tamburrino, A.; Villone, F. Circuits/fields coupling and multiply connected domains in integral formulations. *IEEE Trans. Magn.* **2002**, *38*, 581–584. [CrossRef]
25. Börm, S.; Grasedyck, L.; Hackbusch, W. Hierarchical Matrices. Institut für Mathematik in den Naturwissenschaften. Available online: https://www.researchgate.net/publication/277293203_Hierarchical_Matrices (accessed on 24 March 2022).
26. Ventre, S.; Cau, F.; Chiariello, A.; Giovinco, G.; Maffucci, A.; Villone, F. Fast and Accurate Solution of Integral Formulations of Large MQS Problems Based on Hybrid OpenMP–MPI Parallelization. *Appl. Sci.* **2022**, *12*, 627. [CrossRef]
27. Fresa, R.; Rubinacci, G.; Ventre, S. An eddy current integral formulation on parallel computer systems. *Int. J. Numer. Methods Eng.* **2005**, *62*, 1127–1147. [CrossRef]
28. Scalable Linear Algebra PACKage. Available online: www.scalapack.org (accessed on 12 November 2019).
29. Gruber, O.; Lackner, K.; Pautasso, G.; Seidel, U.; Streibl, B. Vertical displacement events and halo currents. *Plasma Phys. Control. Fusion* **1993**, *35*, B191–B204. [CrossRef]
30. Kapur, S.; Long, D.E. N-body problems: IES3: Efficient electrostatic and electromagnetic simulation. *IEEE Comput. Sci. Eng.* **1998**, *5*, 60–67. [CrossRef]
31. Voltolina, D.; Torchio, R.; Bettini, P.; Specogna, R.; Alotto, P. Optimized cycle basis in volume integral formulations for large scale eddy-current problems. *Comput. Phys. Commun.* **2021**, *265*, 108004. [CrossRef]
32. Rubinacci, G.; Tamburrino, A. Automatic Treatment of Multiply Connected Regions in Integral Formulations. *IEEE Trans. Magn.* **2010**, *46*, 2791–2794. [CrossRef]
33. Burkholder, R.J.; Lee, J.-F. Fast Dual-MGS Block-Factorization Algorithm for Dense MoM Matrices. *IEEE Trans. Antennas Propag.* **2004**, *52*, 1693–1699. [CrossRef]
34. The MPI Forum. MPI: A Message Passing Interface. In Proceedings of the 1993 ACM/IEEE Conference on Supercomputing. Supercomputing ‘93, Portland, OR, USA, 15–19 November 1993.
35. Barney, B. *Introduction to Parallel Computing*; Livermore National Laboratory: Livermore, CA, USA, 2015.
36. Saad, Y.; Schultz, M.H. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. Sci. Stat. Comput.* **1986**, *7*, 856–869. [CrossRef]
37. The HLS Mathematical Software Library, Symmetric Sparse Matrix: Compute elimination orderings. Available online: https://www.hsl.rl.ac.uk/specs/hsl_mc68.pdf (accessed on 24 March 2022).
38. Rubinacci, G.; Tamburrino, A.; Ventre, S.; Villone, F. Application of n-Fold Rotational Symmetries to Eddy Currents Integral Model in the Time Domain. *IEEE Trans. Magn.* **2019**, *56*, 1–4. [CrossRef]
39. Cau, F.; Bessette, D.; Bauer, P.; Gauthier, F.; Portone, A.; Testoni, P.; Ventre, S. Update of Joule Losses Calculation in the ITER Cold Structures during Fast Plasma Transients. *IEEE Trans. Appl. Supercond.* **2020**, *30*, 1–4. [CrossRef]
40. Cau, F.B.; Bessette, D.; D'Amico, G.; Portone, A.; Rubinacci, G.; Testoni, P.; Ventre, S.; Villone, F. Joule Losses in the ITER Cold Structures during Plasma Transients. *IEEE Trans. Appl. Supercond.* **2016**, *26*, 1–5. [CrossRef]
41. Foussat, A.; Libeyre, P.; Mitchell, N.; Gribov, Y.; Jong, C.T.J.; Bessette, D.; Gallix, R.; Bauer, P.; Sahu, A. Overview of the ITER Correction Coils Design. *IEEE Trans. Appl. Supercond.* **2010**, *20*, 402–406. [CrossRef]
42. Bachmann, C.; Sugihara, M.; Roccella, R.; Sannazzaro, G.; Gribov, Y.; Riccardo, V.; Hender, T.; Gerasimov, S.; Pautasso, G.; Belov, A.V.; et al. Specification of asymmetric VDE loads of the ITER tokamak. *Fusion Eng. Des.* **2011**, *86*, 1915–1919. [CrossRef]
43. Börm, S.; Grasedyck, L.; Hackbusch, W. Introduction to hierarchical matrices with applications. *Eng. Anal. Bound. Elem.* **2003**, *27*, 405–422. [CrossRef]
44. Hackbusch, W. A sparse matrix arithmetic based on H-matrices. Part I: Introduction to H-matrices. *Computing* **1999**, *62*, 89–108. [CrossRef]
45. Hackbusch, W.; Khoromskij, B.N. A Sparse $\mathcal{H}$-Matrix Arithmetic, Part II: Application to Multidimensional Problems. *Computing* **2000**, *64*, 21–47. [CrossRef]
46. H-Lib Pro Documentation. Available online: https://www.hlibpro.com/ (accessed on 21 March 2022).