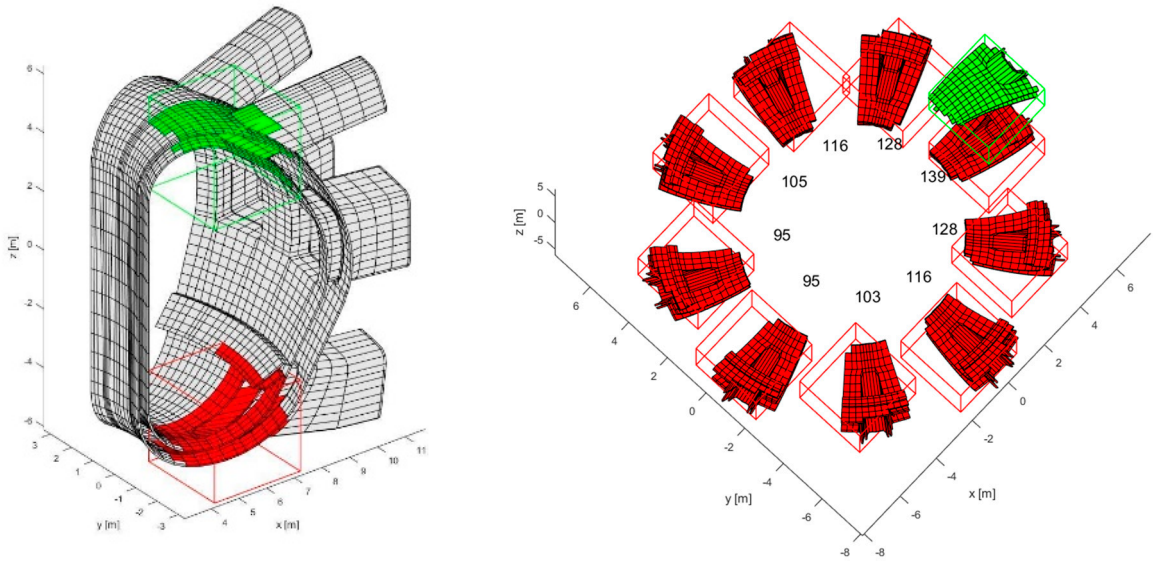
For the sake of clarity, in the following, we report a simple example in which $n_{rot} = 9$, $n_{sym} = 1$ and, hence $n_{tot} = 9$. The main mesh sector is contained in $-20^\circ < \varphi < 20^\circ$.

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 ($-20^\circ > \varphi < 20^\circ$), 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 $\mathbf{L}^*_{ij}$ 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:

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

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 $\mathbf{L}^{\mathbf{bc}}$, 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.

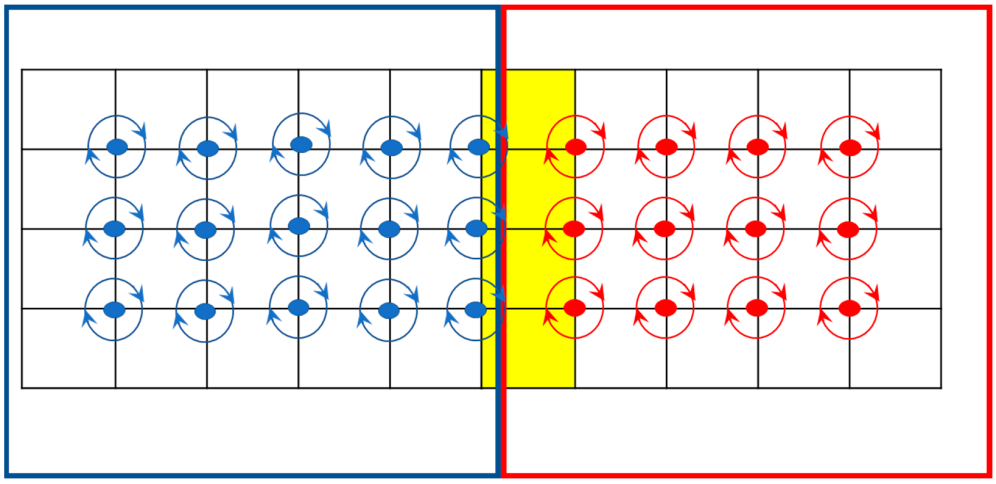


Figure 8. The DOF-based approach. Each box contains its own DOFs, here marked in red and blue. The elements belonging to the interface between the two boxes, require to be replicated in the box–box matrix assembly.

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 $\mathbf{L}^{*}_{ij}$ 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

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

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

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

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 $(\mathbf{L}+\Delta t\ \mathbf{R})^{-1}\mathbf{B}^{T}$ 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, $(\mathbf{L}+\Delta t\ \mathbf{R})^{-1}\mathbf{B}^{\mathrm{T}}$ 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

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

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

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

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

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

By setting

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

the resulting system is

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

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 $\mathbf{A}\mathbf{x}$) 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.

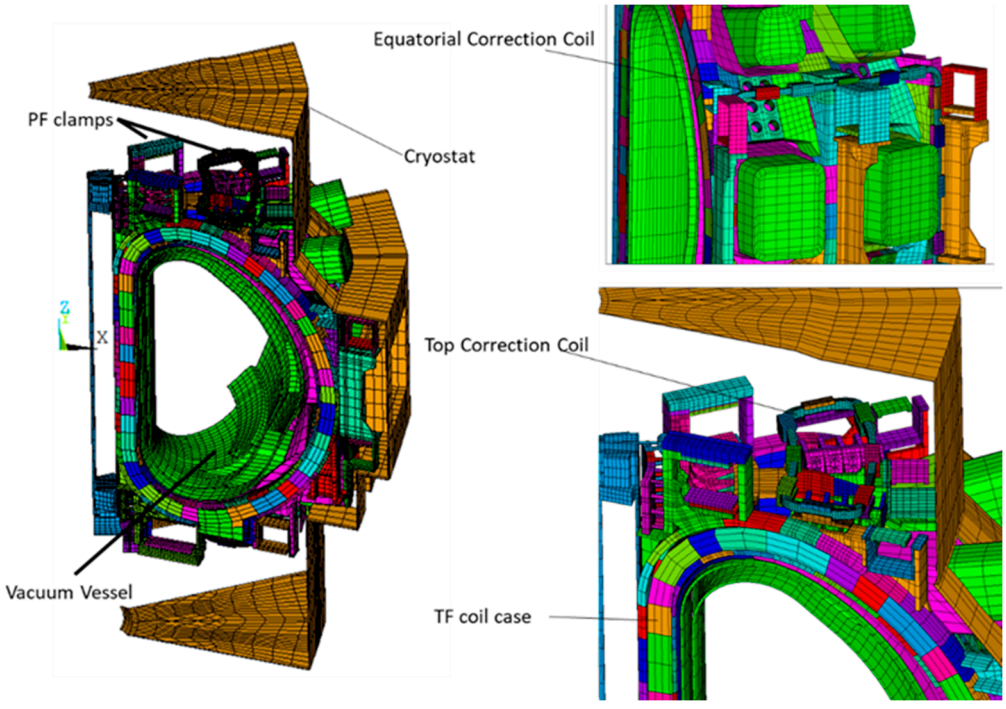
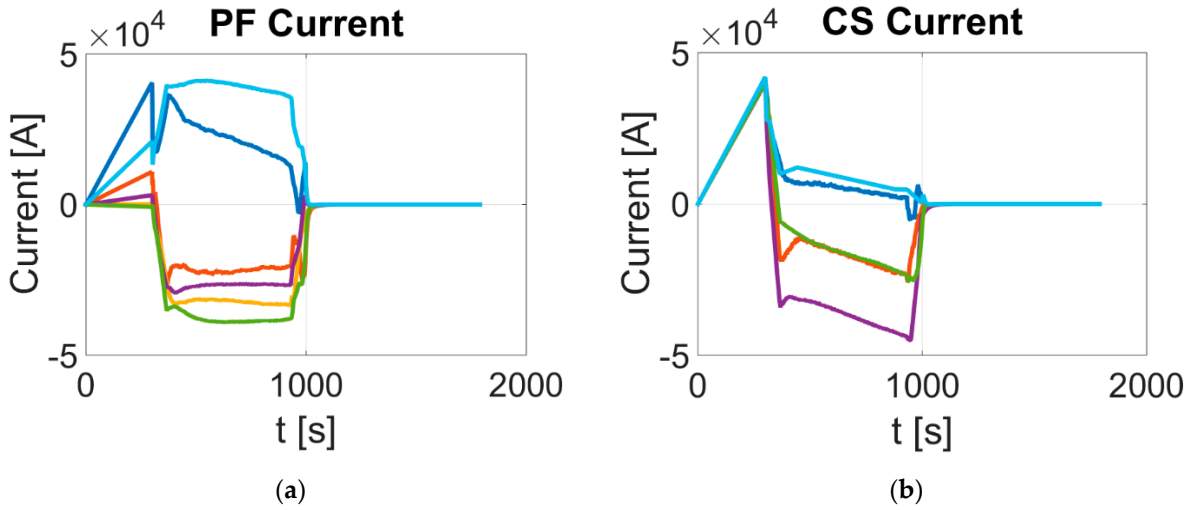


Figure 9. Testcase#1 TF magnet system structure.

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]).

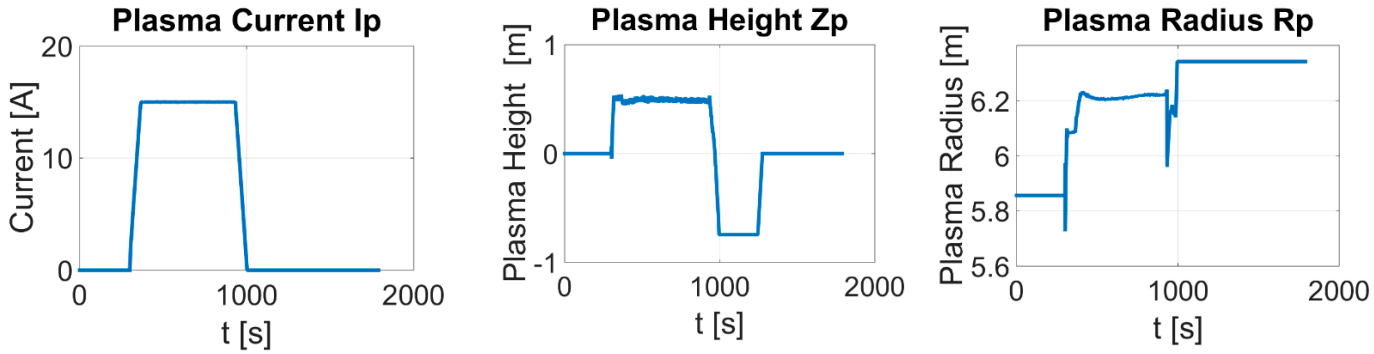
The currents flowing in the each of the six PF and CS coils during the *DINA 2016* scenario is shown in the following Figure 10.



(**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.

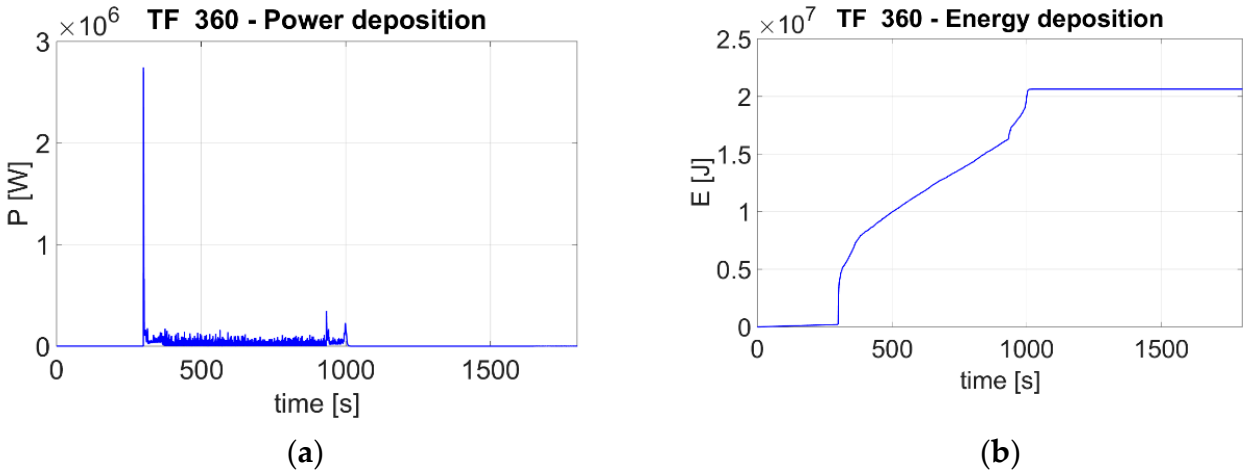


Figure 12. Testcase#1: The power (**a**) and the energy dissipation (**b**) in the TF coil case system.

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.

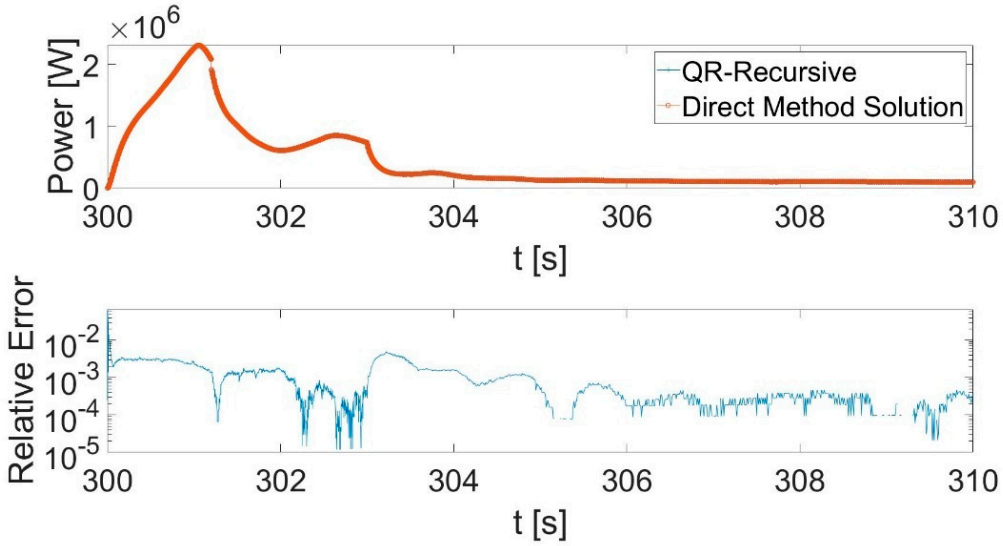


Figure 13. Testcase#1: (**Top**) the power dissipated during the transient in all passive structures. (**Bottom**) the relative error of the QR-recursive based method with respect to the Direct computation.

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.

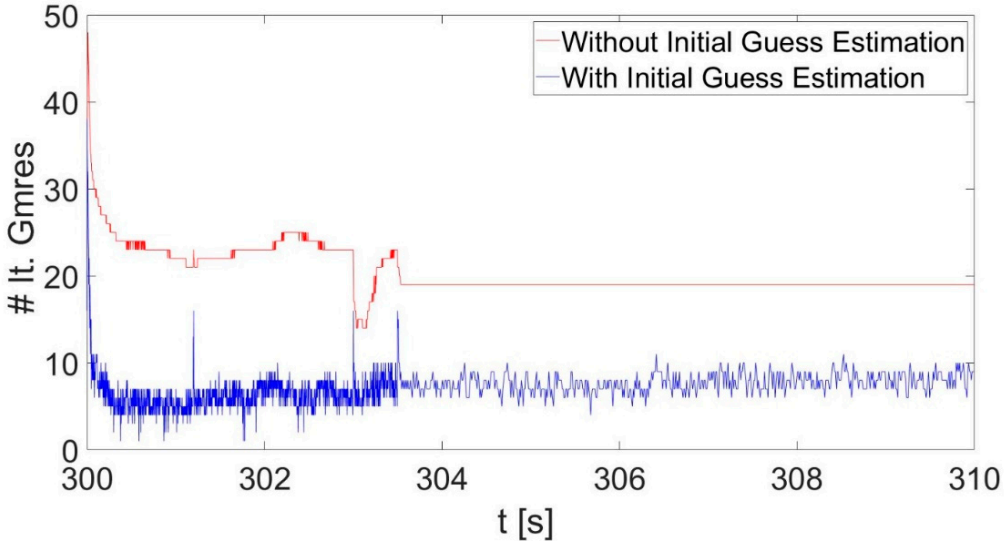


Figure 14. Testcase#1. The number of GMRES iterations required to converge to a relative error equal to 1×10^{-5}, versus time, with and without initial guess estimation.

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**

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

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

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

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

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

The accuracy is defined as

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

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].

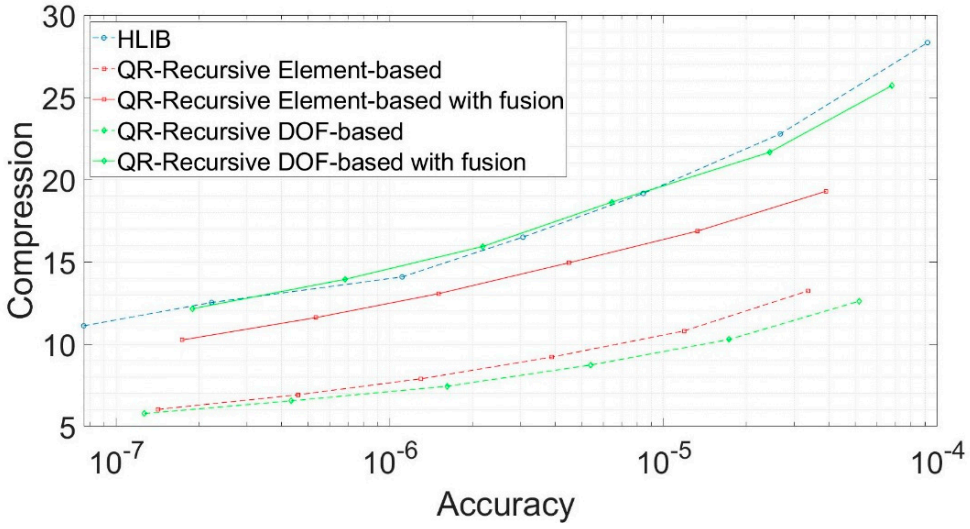


Figure 15. Testcase#1. The Compression versus the Accuracy for various proposed techniques.

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 ($\mathbf{L}_{\text{near}} + \mathbf{L}_{\text{far}}$) load versus the MPI processes involved in the computation. Figure 16 reports proper balancing among different MPI processes.

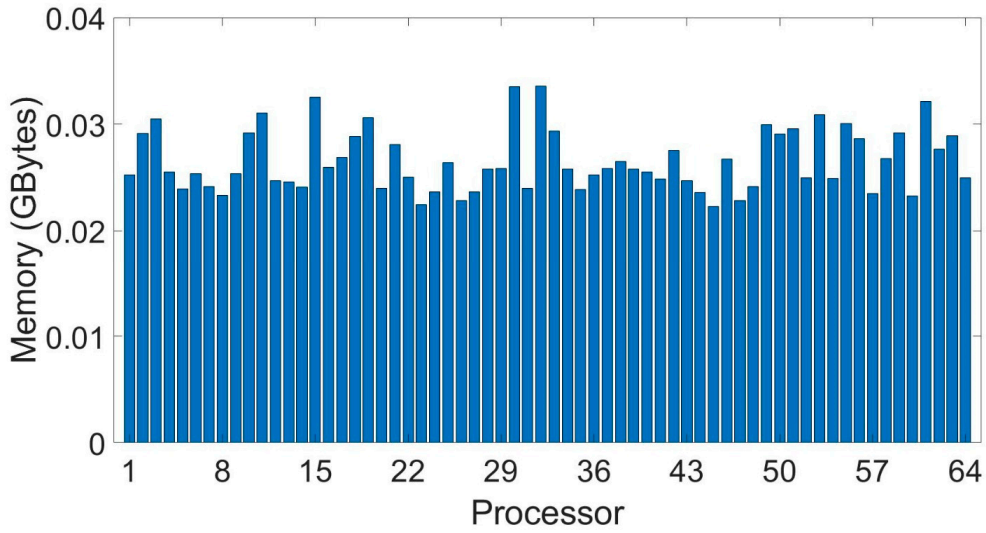


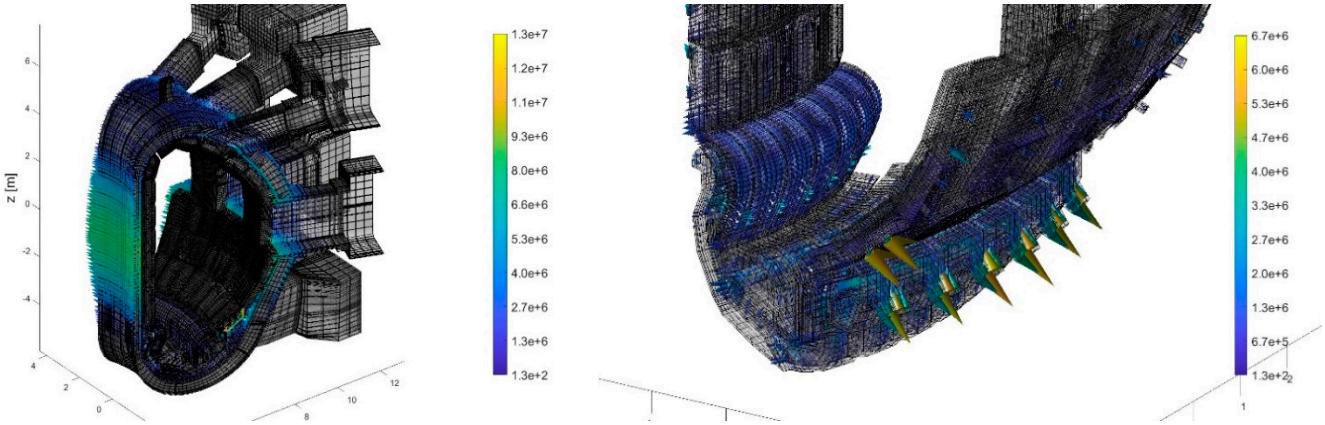
Figure 16. Testcase#1. The Memory/Computation load versus MPI processors used in computation.

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.

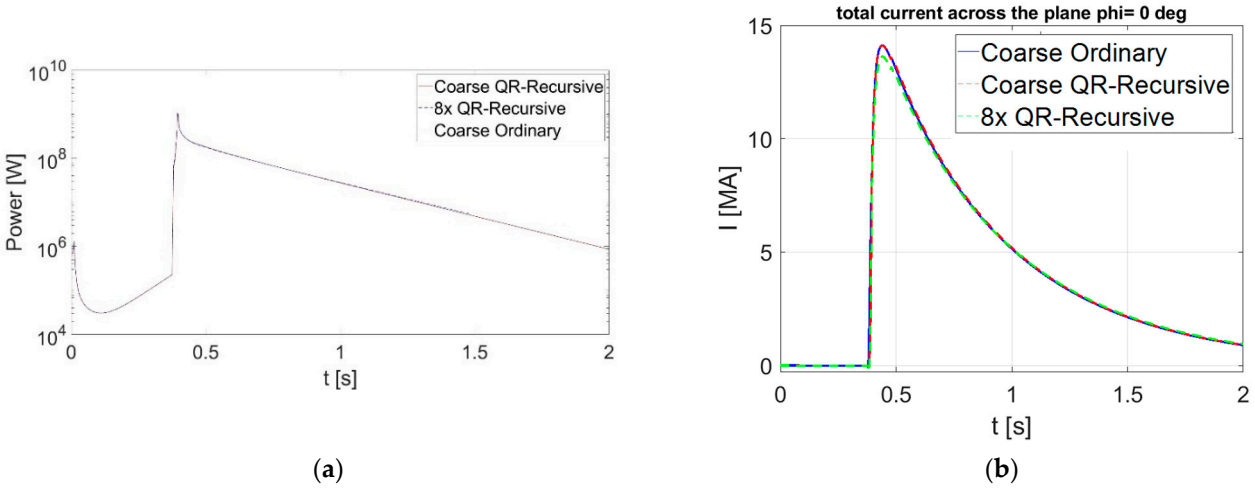
The number of mesh nodes is 504,879, and the number of elements is 346,160. In addition, in this testcase no injected currents are present, hence N_E is equal to zero, and periodic symmetric boundaries are applied; the number of boundary faces (i.e., N_p) present at π_α are 4563 (see Section 2.3). The 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.



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

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.

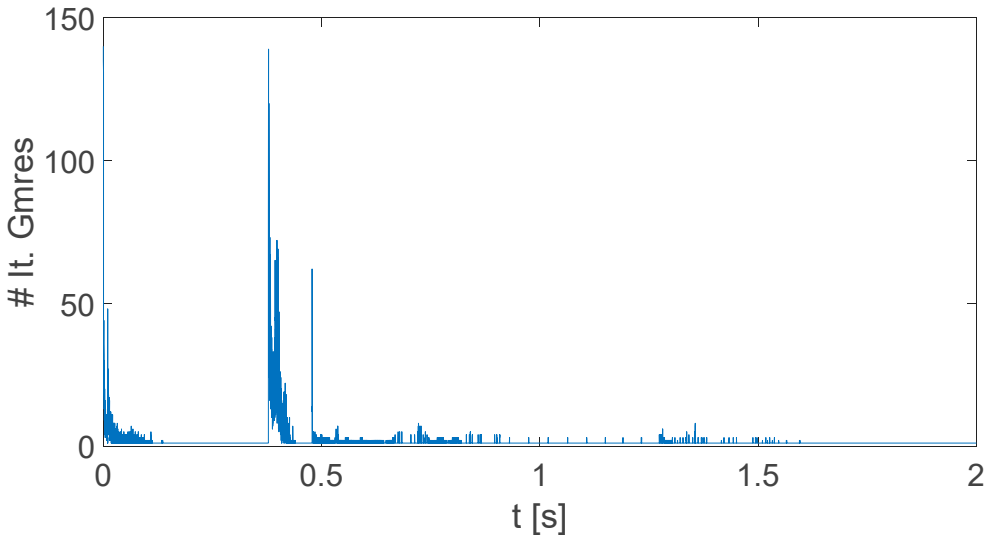


Figure 19. Testcase#2. The number of iterations required by GMRES to converge to a relative tolerance is 1×10^{-5}, versus time.

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.

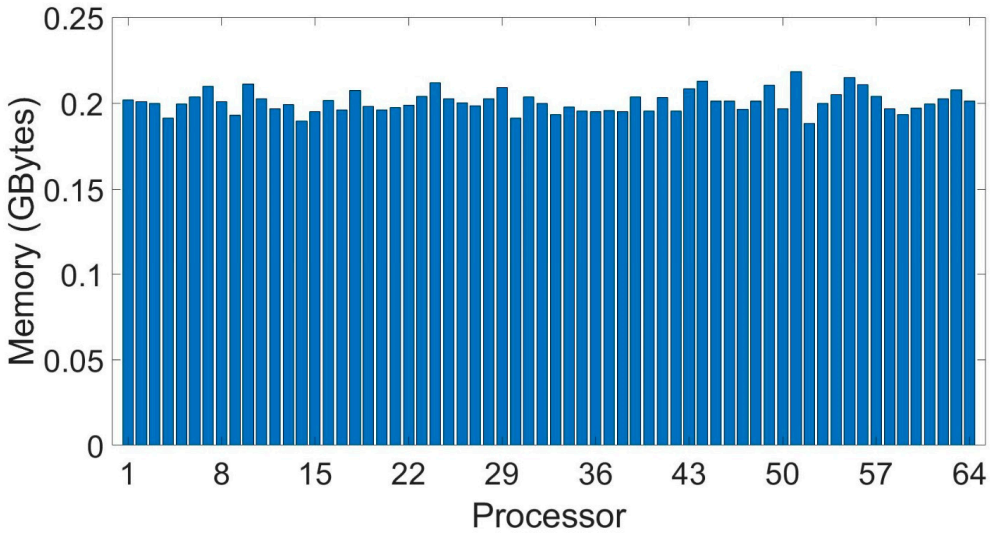


Figure 20. Testcase#2. The Memory/Computation load versus MPI processors used in computation.

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*
$BoxSet \stackrel{\text{def}}{=}$ is the set of all not empty boxes
$ChildlessSet \stackrel{\text{def}}{=}$ 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 **b**
$ChildrenSet(b, 1:8)$ returns the eight children of the box **b**
Create first box **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 **F** $\in$ ParentBoxesList // loop over previous level father box **F**
$NumObjectsF = \#(\mathbf{F})$ // set the number of objects in box **F**
If $(NumObjectsF > smin1)$
// Subdivide father **F** in eight children **C**
For i = 1,8
C = $subdivide(F, i)$ // get i-th children **C**
$NumObjects = \#(\mathbf{B})$ // get the number of objects in box **B**
If $(NumObjects > 0)$
// Update Sets
$Children(\mathbf{F}, i) = \mathbf{C}$ // **C** is the son of **F**
$FatherSet(Box) = \mathbf{F}$ // **F** is the father of **C**
$BoxSet = \{BoxSet \cup \mathbf{C}\}$// **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.

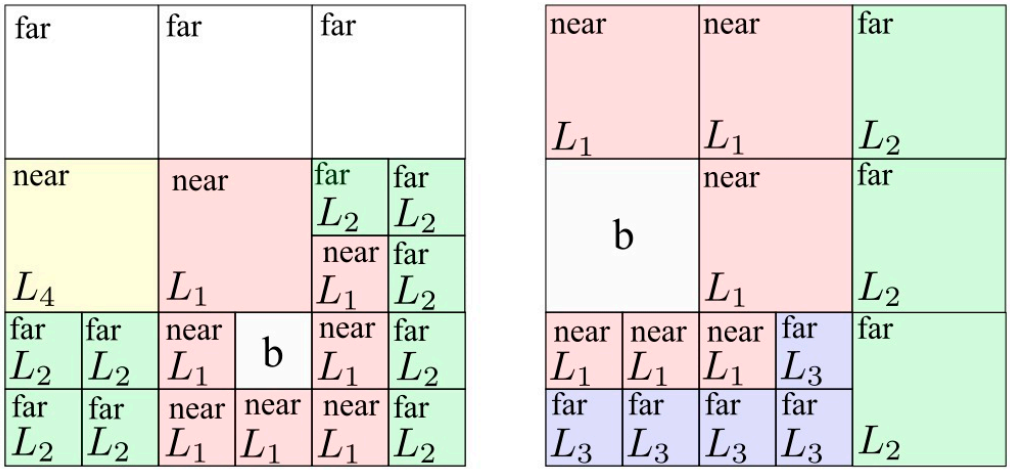


Figure A1. The four lists used to define $\mathcal{L}_{\text{near}}(\mathbf{b})$, $\mathcal{L}_{\text{far}}(\mathbf{b})$ associated for a given field box **b**. The sources boxes are depicted as: (i) pink and yellow for near interactions, and (ii) green and violet for the far interactions. Note that 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

*smin*2 is the minimum number of objects

$fsmin$ (> 1) is a parameter controlling maximum distance

$ChildLessSet \stackrel{\text{def}}{=}$ is the set of the childless boxes

Output

$ListObject(\mathbf{b}) \stackrel{\text{def}}{=}$ is the set of objects in the box **b**

$ListRetain = \{\varnothing\}$ // $\stackrel{\text{def}}{=}$ is the set of the boxes to be retained

$ListToFuse = \{\varnothing\}$ // $\stackrel{\text{def}}{=}$ is the set of the boxes to be fused

For **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 **b** $\in$ *ListToFuse* // loop over the boxes to be fused

  //compute the distance from **b**

  For **x** $\in$ *ListRetain*

    d(**x**) = distance(**b**,**x**) // distance between **b** and **x**

  End

  // Compute the minimum distance

Algorithm A2. *Cont.*

$\min\_d = \min_{\mathbf{x}} d(\mathbf{x})$
// set the maximum allowed distance
$\max\_d = \text{fsmin} \stackrel{\text{def}}{=} \min\_\text{d}$
// Create the set of all boxes whose distance from **b** is less than max_*d*
$Nearest = \{s \in ListRetain : d(\mathbf{s}) \leq \max\_d\}$
// 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 **b** into the **z** box
$ListObject(\mathbf{z}) = \{ListObject(\mathbf{z}) \cup ListObject(\mathbf{b})\}$
// the objects of the box **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).