Computers and Structures 181 (2017) 41–54

Contents lists available at ScienceDirect

Computers and Structures

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

Resolving the sign conflict problem for *hp*–hexahedral Nédélec elements with application to eddy current problems

R.M. Kynch, P.D. Ledger *

Zienkiewicz Centre for Computational Engineering, College of Engineering, Swansea University Bay Campus, Fabian Way, Crymlyn Burrows, Swansea SA1 8EN, United Kingdom

ARTICLE INFO

Article history:
Received 1 October 2015
Accepted 23 May 2016
Available online 11 August 2016

Keywords:
hp–Nédélec elements
Edge finite elements
Hexahedral discretisations
Sign conflict problem
Eddy current problems

ABSTRACT

The eddy current approximation of Maxwell's equations is relevant for Magnetic Induction Tomography (MIT), which is a practical system for the detection of conducting inclusions from measurements of mutual inductance with both industrial and clinical applications. An MIT system produces a conductivity *image* from the measured fields by solving an *inverse problem* computationally. This is typically an iterative process, which requires the *forward* solution of a Maxwell's equations for the electromagnetic fields in and around conducting bodies at each iteration. As the (conductivity) images are typically described by voxels, a hexahedral finite element grid is preferable for the forward solver. Low order Nédélec (edge element) discretisations are generally applied, but these require dense meshes to ensure that skin effects are properly captured. On the other hand, *hp*–Nédélec finite elements can ensure the skin effects in conducting components are accurately captured, without the need for dense meshes and, therefore, offer possible advantages for MIT. Unfortunately, the hierarchic nature of *hp*–Nédélec basis functions introduces edge and face parameterisations leading to sign conflict issues when enforcing tangential continuity between elements. This work describes a procedure for addressing this issue on general conforming hexahedral meshes and an implementation of a hierarchic *hp*–Nédélec finite element basis within the `deal.II` finite element library. The resulting software is used to simulate Maxwell forward problems, including those set on multiply connected domains, to demonstrate its potential as an MIT forward solver.

1. Introduction

Magnetic induction tomography (MIT) holds promise for the detection of metallic objects and inclusions in industrial and clinical imaging applications. For example, it has been proposed that the medical diagnosis of cerebral haemorrhage can be assisted by identifying the regions of higher conductivity associated with bleeding [17]. Further potential applications include lung monitoring, security screening, ensuring food safety and the monitoring of molten metals. For an indepth review of MIT and its applications we refer to [17,37,32]. In principle, an MIT system is capable of determining the distributions of the three electromagnetic parameters, permittivity, permeability and conductivity, from measurements of the electromagnetic fields exterior to the body of interest, although conductivity is usually selected as being of primary importance for imaging purposes. An MIT system works by first generating a sequence of background magnetic fields by exciting a series of coils, and voltages are then measured in a second array of coils. In the presence of a conducting body, the voltages are perturbed due to the eddy currents generated when a low-frequency magnetic field interacts with a conducting body. Finally, in order to produce a voxelated MIT conductivity image from the measured voltages, an *inverse problem* must be solved.

The computational solution of the MIT inverse problem is challenging due to its ill-posed nature and the noise in the measured data. Common solution approaches include regularised linearised single step algorithms [38] and Gauss-Newton based procedures [31], see also [8,37] for further alternatives. In this work, our focus is on the associated MIT *forward problem*, which also presents significant challenges, and its efficient and accurate solution is required at each iteration of the computational inverse solution in order to compute sensitivity and voltage information.

An MIT forward problem is the boundary value problem that is associated with the eddy current approximation of the time harmonic Maxwell equations. It is set in a domain consisting of a collection of conducting bodies and current sources located in free space, as illustrated in Fig. 1. Note that at each stage of the inverse solution algorithm, the shape and conductivity values of the bodies change and the MIT forward problem must be resolved until convergence is achieved. For the purpose of computation, the domain is truncated and, provided the boundary is placed sufficiently far from the bodies and coils, the fields are assumed to vanish at this location. To provide a simple description of the shape of the bodies, and to facilitate the

* Corresponding author.
E-mail address: p.d.ledger@swansea.ac.uk (P.D. Ledger).

http://dx.doi.org/10.1016/j.compstruc.2016.05.021
0045-7949/

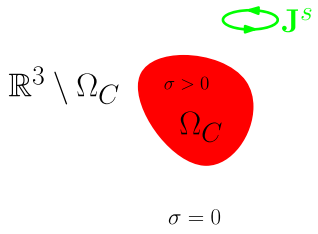


Fig. 1. An illustration of an eddy current problem showing conducting region Ω_C with a current source $\boldsymbol{J}^s$ in the surrounding unbounded region $\mathbb{R}^3 \setminus \Omega_C$.

change in their conductivity values in the MIT inverse solution, a regular voexlated grid is usually preferred for describing the conductivity distribution across the domain, which, in turn, provides a natural conductivity image. Previous modelling attempts have focused on low order Nédélec finite element discretisations (in our notation $p = 0$ (first, edge) or $p = 1$ (second) order elements, where p refers to the degree of approximation of the tangential component of the field) [31,39,38] or finite difference approaches [24]. It is also not uncommon to select the voxelated description of the conductivity to be the same as that used for computing the fields leading to coarse grids. Unfortunately, even small inaccuracies in the simulated voltages can have large effects on the conductivity reconstruction due to the ill-posed nature of the MIT inverse problem. Furthermore, inaccuracies in the sensitivity information can point Gauss–Newton iterations in the wrong direction.

Conducting objects in the eddy current regime are characterised by their skin depth, which describes the distance that induction currents in a conductor decay to $1/e$ of their surface value. When the skin depth is small, high gradients in the fields are experienced close to the surface of the conductor leading to the requirement for fine grids. Elsewhere the solution is smooth and can be well represented by high degree polynomials. *hp*–Finite elements combine the ability to use local mesh (*h*–) refinement with polynomial (*p*–) enrichment of the solution and are known to lead to exponential convergence of the solution (using *p*–enrichment when the solution is analytic and using *hp*–refinement when the solution has singularities). In particular, high order and *hp*–Nédélec finite element procedures have been shown to offer superior levels of accuracy compared to low-order finite approaches for a variety of problems in electromagnetism. For an in depth-review we refer to [9,10] and references therein as well as to [21] for an application of *hp*–finite elements to the solution of eddy current problems in three–dimensions using unstructured tetrahedral meshes.

To facilitate code-reuse within an academic environment, we have chosen to pursue our software developments within the context of an open source finite element library. A number of open source finite element libraries offer the possibility of applying Nédélec elements to high polynomial degree. The `Netgen` unstructured mesh generator, combined with the `NGsolve` library [12], is based on Schöberl and Zaglmayr's basis function sets for the complete DeRham sequence and can handle tetrahedral and hybrid/hexahedral meshes [35]. Zaglmayr has also developed basis functions for other element shapes [34]. The `Hermes` project [13] and its accompanying documentation [25] offers adaptive *hp* finite element approximation on structured meshes. However, to the best of the authors' knowledge, its Nédélec elements are currently limited to two–dimensional problems. The FEniCS library [4] also supports high order Nédélec elements on triangles and tetrahedra and a further library that also supports higher order Nédélec elements on unstructured meshes is GetDP [11]. The `deal.II` library [7] is focused on quadrilateral and hexahedral meshes and can be applied to the solution of a variety of problems including electromagnetism. Its built-in mesh generator allows for the generation of meshes for simple geometrical configurations and the library also allows meshes to be imported from external generators, such as `Cubit` (known commercially as `Trelis`) [19].

Unstructured meshes have clear benefits of meshing complex geometrical configurations; however, for the present application, a structured mesh for the voxelated solution of the MIT inverse problems is desired. Therefore, we have chosen to adopt the `deal.II` finite element framework for our MIT forward solver. A hierarchic Nédélec finite element basis is available within the library, but unfortunately this has been found to have several important limitations: In particular, in its handling of boundary conditions and limitation to regular structured meshes consisting of blocks. The former issue has been overcome by improved handling of boundary conditions but the latter problem is more severe and fails for meshes, which do not conform to `deal.II`'s prescribed standard orientation of edges and faces. For example, the relatively simple geometry of a cylinder, generated using `deal.II`'s `GridGenerator::cylinder()`, results in a mesh which is non-standard. The cause of the issue has been identified as being due to the sign–conflict issue, which arises due to the mismatch of local edge and face parameterisations on neighbouring elements. In two–dimensions this problem is easily overcome by a simple sign-flip during element assembly. For three–dimensional tetrahedral discretisations, a strategy for circumventing the sign–conflict issue has been proposed by Ainsworth and Coyle [1] in which each element is chosen to be one of two reference configurations. The situation for hexahedral discretisations is much more complex due to increased number of edge–face orientation permutations and, if a similar strategy were to be adopted, would lead to much larger number of reference configurations.

In this work, we describe an implementation of Nédélec elements on meshes comprised of either structured or unstructured hexahedra using the hierarchic basis function set proposed by Schöberl and Zaglmayr [35]. This choice offers added benefits as the set lends itself to an efficient preconditioner for solving the discretised equations for the MIT problem. To circumvent the sign conflict problem, we employ edge and face parameterisations based on the specification of global, rather than local, orientations. Consequently, the description of the basis function changes between elements and, to address this, a strategy for their efficient implementation is proposed. As illustrations of the success of the approach we apply it to a series of three–dimensional eddy current benchmark problems, relevant to our MIT application, using the previously described regularised formulation of eddy current problems [21]. The novelty of this work is a detailed description of the efficient implementation of the basis and resolution of the sign conflict problem, which allows the `deal.II` library to be applied to handle a much wider class of geometries for problems in electromagnetism. The description will be useful for users and developers of `deal.II` interested in the implementation of hierarchic basis functions sets on hexahedral meshes and, furthermore, our implementation offers wide reaching benefits to the software library and its current and future users (e.g. in geophysics [16,15]).

The organisation of the paper is as follows: In Section 2 we briefly summarise the MIT forward problem/eddy current model, which will serve as an illustration of our `deal.II` Nédélec finite element implementation. Then, in Section 3, we summarise the construction and properties of the basis functions and discuss the sign conflict issue and how it can be overcome by the use of global rather than local edge and face orientations. Section 4 describes the details of the `deal.II` implementation of the elements, and an efficient preconditioning technique for the solution of eddy current problems [21] is summarised in Section 5. Section 6 presents a series of numerical examples and includes comparisons with the previous `deal.II` Nédélec element implementation. Finally, the steps required to extend the implementation to non–uniform *hp*–refinements on non–conforming meshes are presented in Section 7.

(a) Edge 1, $p = 0$ (b) Edge 1, $p = 1$ (c) Edge 1, $p = 2$

(d) Edge 2, $p = 0$ (e) Edge 2, $p = 1$ (f) Edge 2, $p = 2$

(g) Face function, $p = 1$ (h) Face function, $p = 2$ (i) Face function, $p = 3$

Fig. 2. An ilustration of typical Nédélec edge– and face–based basis functions, restricted to a quadrilateral face. In (a)-(f) the edge–based functions are shown for two typical edges and polynomial degrees $p = 0, 1, 2$. In (g)-(i) typical face-based gradient functions are shown for $p = 1, 2, 3$.

2. Eddy current model

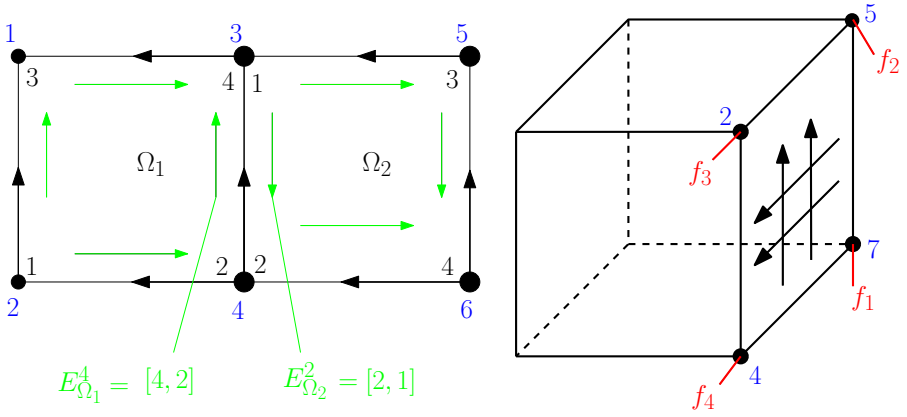
We briefly review the time harmonic eddy current model, which will serve as a basis for the numerical examples presented later in this paper. In the following, we adopt the convention that vectors valued quantities are written in bold face and matrices, whose coefficients are independent of the choice of coordinate system, are written in Roman italics. Let Ω_C denote a conducting object with uniform conductivity and permeability, which is located in an unbounded free space region $\mathbb{R}^3 \setminus \Omega_C$, as illustrated in Fig. 1. The position dependent material parameters are

$$\sigma = \begin{cases} \sigma_* & \text{in } \Omega_C \\ 0 & \text{in } \mathbb{R}^3 \setminus \Omega_C \end{cases}, \qquad \mu = \begin{cases} \mu_* & \text{in } \Omega_C \\ \mu_0 & \text{in } \mathbb{R}^3 \setminus \Omega_C \end{cases},$$

where $\mu_0 := 4\pi \times 10^{-7}$ H/m denotes the permeability of free space. Given a time harmonic divergence free current source of amplitude $\boldsymbol{J}^s$ and frequency ω, located away from Ω_C, the interaction magnetic and electric fields, $\boldsymbol{H}$ and $\boldsymbol{E}$ respectively, satisfy the eddy current equations [5]

$$\operatorname{curl} \boldsymbol{E} = -\mathrm{i}\omega\mu\boldsymbol{H}, \qquad \operatorname{curl} \boldsymbol{H} = \sigma\boldsymbol{E} + \boldsymbol{J}^s. \tag{1}$$

The interaction fields decay at appropriate rates as $|\boldsymbol{x}| \to \infty$ [5], which computationally allows us to truncate the unbounded domain at a finite distance from the object and form the bounded domain $\Omega := \Omega_C \cup \Omega_{NC}$, where Ω_{NC} is the truncated part of $\mathbb{R}^3 \setminus \Omega_C$. On $\partial\Omega$ we apply simple zero Dirichlet boundary conditions.



(a) Edge orientation (b) Face orientation

Fig. 3. Illustrations of the parameterisations for an edge $E^m = [e_1^m, e_2^m]$ and a face $F^m = [f_1^m, f_2^m, f_3^m, f_4^m]$. Edge: The local vertex numbers in elements Ω_1 and Ω_2 are shown in black, while possible global numberings are shown in blue. The standard `deal.II` edge orientations are shown by the green arrows and the globally defined edge orientations, induced by the global node numbers, are shown by the black arrows. The shared edge's orientation labelling is given in green for the two elements. Face: The global vertex numberings are shown in blue and the globally defined face orientation is shown in black. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Based on previous success in applying *hp*–finite elements to eddy current problems on multiply connected domains, we adopt an $\boldsymbol{A}$-based regularised form of the eddy current model [21], which has the associated weak form: Find $\boldsymbol{A} \in \boldsymbol{V}$ such that

$$\int_\Omega \mu_r \mathrm{curl}\, \boldsymbol{A} \cdot \mathrm{curl}\, \boldsymbol{w} \mathrm{d}\Omega + \int_\Omega \tilde{\kappa} \boldsymbol{A} \cdot \boldsymbol{w} \mathrm{d}\Omega = \mu_0 \int_{\Omega_{NC}} \boldsymbol{J}^s \cdot \boldsymbol{w} \mathrm{d}\Omega \qquad \forall \boldsymbol{w} \in \boldsymbol{V}, \tag{2}$$

where $\mu_r := \mu/\mu_0, \boldsymbol{A}$ is a vector potential defined as $\boldsymbol{B} := \mu \boldsymbol{H} = \mathrm{curl}\, \boldsymbol{A}$ and is such that $\boldsymbol{E} = -\mathrm{i}\omega \boldsymbol{A}$ in Ω_C. The parameter $\tilde{\kappa}$ is defined as

$$\tilde{\kappa} := \begin{cases} \kappa & \text{in } \Omega_C \\ \mathrm{i}\varepsilon & \text{in } \Omega_{NC} \end{cases},$$

where ε is a small regularisation parameter, $\kappa := \mathrm{i}\omega\mu_0\sigma_*$and

$\boldsymbol{V} := \{\boldsymbol{A} \in \boldsymbol{H}(\mathrm{curl}) : \boldsymbol{n} \times \boldsymbol{A} = \boldsymbol{0} \text{ on } \partial\Omega\}$.

For a review and details of alternative eddy current formulations we refer to [21,27,18] and references therein.

3. Basis functions and overcoming the sign conflict issue

We consider a hexahedral triangulation of Ω, which is described by the sets of cells, $\mathcal{H}$, faces, $\mathcal{F}$, edges, $\mathcal{E}$ and vertices, $\mathcal{V}$. On this triangulation we choose to apply the Nédélec finite element basis of Schöberl and Zaglmayr [29,34], which we denote as $\boldsymbol{V}_{h,p}$ where h denotes the mesh width and p the polynomial degree. These basis functions are hierarchic and can be decomposed as

$$\boldsymbol{V}_{h,p} := \mathcal{N}_0 \oplus \sum_{E \in \mathcal{E}} \boldsymbol{V}_p^E \oplus \sum_{F \in \mathcal{F}} \boldsymbol{V}_p^F \oplus \sum_{I \in \mathcal{H}} \boldsymbol{V}_p^I \subset \boldsymbol{H}(\mathrm{curl}), \tag{3}$$

where $\mathcal{N}_0$ denotes the set of lowest order basis functions, and $\boldsymbol{V}_p^E, \boldsymbol{V}_p^F$ and $\boldsymbol{V}_p^I$ the enrichment through the sets of higher-order basis functions associated with edges, faces and cell interiors, respectively. The construction of the basis is such that $\boldsymbol{V}_p^E$ is comprised entirely of gradient fields and $\boldsymbol{V}_p^F$ and $\boldsymbol{V}_p^I$ are comprised of both gradients and additional non–gradients in order to complete the space. The explicit definition of the basis functions can be found in [34] and is not repeated here. Instead, we illustrate vector quiver plots of sample basis functions in Fig. 2. In this figure we observe the property that the tangential components of edge–based basis functions vanish on all edges apart from the one with which they are associated and that the corresponding basis functions associated with a face vanish on all edges of the hexahedra, thus allowing tangential continuity to be enforced between neighbouring elements. Furthermore, considering a pth order edge–based basis function, we see that its tangential component on the edge with which it is associated is a polynomial of degree p. The lowest order case, $p = 0$, gives rise to a constant tangential component and this has led to these elements been commonly called *edge elements*. Indeed, for the specific choice of basis functions proposed in [34], the tangential components of the higher order edge–based basis functions ($p \geqslant 1$) reduce to

$$L_1(\xi) = \xi, \tag{4a}$$

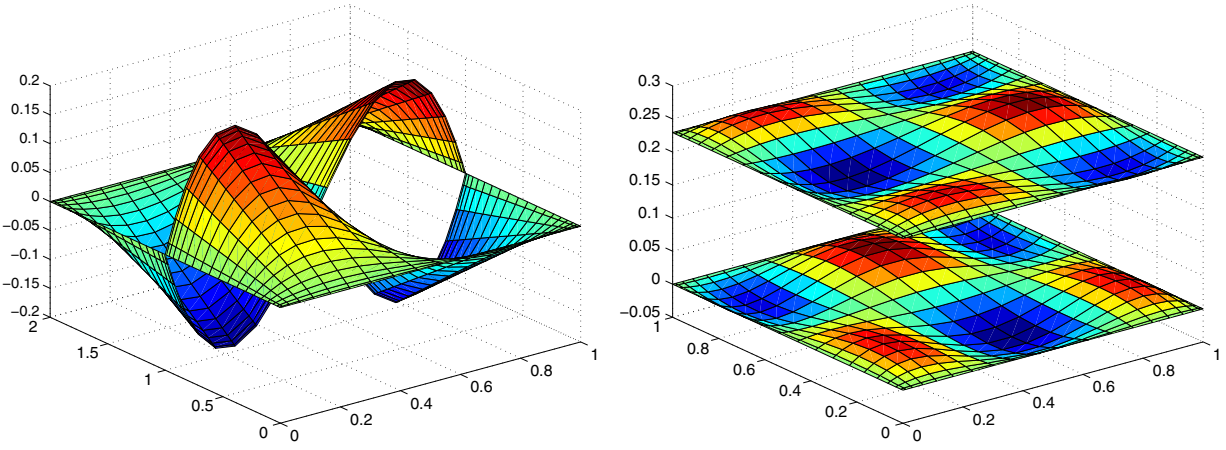
$$L_2(\xi) = \frac{1}{2}(\xi^2 - 1), \tag{4b}$$

$$L_p(\xi) = \frac{1}{p}((2p-3)\xi L_{p-1}(\xi) - (p-3)L_{p-2}(\xi)) \qquad \text{for } p \geqslant 2. \tag{4c}$$

In the above, $L_p(\xi)$ for $p \geqslant 2$ are the integrated Legendre polynomials and $\xi \in [-1, 1]$ denotes a parameterisation of the edge.

As outlined in the introduction, the practical implementation of high order Nédélec elements requires the sign conflict problem to be addressed. This issue stems from the required introduction of edge and face parameterisations, as illustrated in Fig. 3. Note that the chosen face orientation differs from that in [34] and has been chosen to be consistent with that in `deal.II`. Given an edge $E^m := [e_1^m, e_2^m]$[1] with vertex numbers e_1^m, e_2^m the edge parameterisation is defined as $\xi_E^m := \sigma_{e_2^m} - \sigma_{e_1^m} \in [-1, 1]$, where $\sigma_{e_2^m}, \sigma_{e_1^m}$, denote the barycentric coordinates of the two vertices, which make up the edge. In a similar manner, the face parameterisation for a face $F^n := [f_1^n, f_2^n, f_3^n, f_4^n]$ is defined as $(\xi_F^n, \eta_F^n) \in [-1, 1]^2$ where $\xi_F^n := \sigma_{f_2^n} - \sigma_{f_1^n}, \eta_F^n := \sigma_{f_4^n} - \sigma_{f_1^n}$. On an edge, the tangential components of the basis reduce to polynomials in ξ_E^m and, on faces, polynomials in ξ_F^n, η_F^n. In order to apply tangential continuity between elements it is essential that the resulting polynomials between edges in neighbouring elements match. When a simple local edge and face parameterisation is introduced this is no longer guaranteed to be the case, as Fig. 4 illustrates.

[1] We use E^m to represent the mth edge and $\boldsymbol{E}$ for the electric field. From the context it should clear as to which definition applies.



(a) Edge conflict (b) Face conflict

Fig. 4. Illustrations of the sign conflict issue, which arises due to the introduction of local edge and face parameterisations. In (a) we show the tangential component of a $p = 3$ edge–based basis function, when neighbouring elements have different edge orientations. In (b) we show the non-zero tangential component of a $p = 3$ face–based function for neighbouring elements with different face orientations.

Zaglmayr [34] has described a strategy for resolving the sign conflict issue, which relies on Nédélec basis functions being defined according to global edge and face parameterisation rather than local parameterisations. We briefly summarise her strategy for choosing these global edge and face parameterisations in Algorithm 1, where $\mathcal{V}^K \subseteq \mathcal{V}, \mathcal{E}^K \subseteq \mathcal{E}$ and $\mathcal{F}^K \subseteq \mathcal{F}$ denote the restriction of the sets of vertices, edges and faces to a cell K. By adopting this strategy in each element in the mesh the sign conflict problem is addressed. This strategy can also be applied to two–dimensional problems, but without the additional complexity of introducing the global face parameterisations. An alternative strategy for obtaining edge and face parameterisations for high order Nédélec elements is presented in [22].

Algorithm 1. Zaglmayr's algorithm [34] for choosing unique global edge and face parameterisations in a element-face-edge-vertex partitioning of Ω.

In cell, K, denote the global vertex index for local index α as $v_\alpha^K \in \mathcal{V}^K \subseteq \mathcal{V}$.

On each edge $\ell = 1, \cdots, 12$ of cell K (with local vertices $\left\{\alpha_i^{E_\ell}\right\}_{i=1}^{2}$) we define the edge orientation $E_K^\ell = [e_1^\ell, e_2^\ell] \in \mathcal{E}_K$ such that

$$e_1^\ell := \arg\max_{\alpha \in \left\{\alpha_i^{E_\ell}\right\}_{i=1}^{2}} \{v_\alpha^K\}, \qquad e_2^\ell := \arg\min_{\alpha \in \left\{\alpha_i^{E_\ell}\right\}_{i=1}^{2}} \{v_\alpha^K\}$$

On each face $\ell = 1, \cdots, 6$ of cell K (with vertices $\left\{\alpha_i^{F_\ell}\right\}_{i=1}^{4}$), we define the face orientation $F_K^\ell = \left[f_1^\ell, f_2^\ell, f_3^\ell, f_4^\ell\right] \in \mathcal{F}_K$ such that

$$f_1^\ell := \arg\max_{\alpha \in \left\{\alpha_i^{F_\ell}\right\}_{i=1}^{4}} \{v_\alpha^K\}, \qquad f_3^\ell \quad \text{opposite } f_1^\ell \text{ on } F_K^\ell, \qquad v_{f_2^\ell}^K > v_{f_4^\ell}^K$$

Choosing E_K^ℓ and F_K^ℓ in this way then will ensure the orientation matches on edges and faces on the corresponding global edges and faces E^m and F^n between neighbouring elements for some m and n.

4. `deal.II` implementation

The most important part of the implementation of the Nédélec basis of Schöberl and Zaglmayr [29] in the `deal.II` framework is the generation of the global basis functions $\mathbf{\Phi}_i$ required for the evaluation of integrals over a cell K of the form

$$a_{ij}^e(\mu_r) := \int_K \mu_r^{-1} \operatorname{curl} \mathbf{\Phi}_i \cdot \operatorname{curl} \mathbf{\Phi}_j \mathrm{d}\Omega, \qquad m_{ij}^e(\tilde{\kappa}) : = \int_K \tilde{\kappa} \mathbf{\Phi}_i \cdot \mathbf{\Phi}_j \mathrm{d}\Omega, \tag{5}$$

which, when suitably assembled, form the global *curl–curl* and *mass* matrices, $A(\mu_r)$ and $M(\tilde{\kappa})$, respectively. Crucially, although Algorithm 1 describes an approach for generation of global edge and face parameterisations, the definition of their Nédélec type basis function for a given cell K still remains on a standard reference $[0,1]^3$ cube with coordinates $\hat{\boldsymbol{x}}$. Therefore, the first step is to map typical local basis functions $\hat{\mathbf{\Phi}}_i$ to their corresponding global functions and, by using standard transformations [23], this means that

$$a_{ij}^e(\mu_r) = \int_{\widehat{R}} \frac{\mu_r^{-1}}{|J|} \hat{\operatorname{curl}}\, \hat{\mathbf{\Phi}}_i \cdot \hat{\operatorname{curl}}\, \hat{\mathbf{\Phi}}_j \mathrm{d}\hat{\Omega}, \qquad m_{ij}^e(\tilde{\kappa}) = \int_{\widehat{R}} \tilde{\kappa} J^{-T} \hat{\mathbf{\Phi}}_i \cdot J^{-T} \hat{\mathbf{\Phi}}_j |J| \mathrm{d}\hat{\Omega}, \tag{6}$$

where J denotes the Jacobian of the transformation between K and the reference cell $\widehat{R}$ and the superscript T denotes the transpose. To numerically evaluate $a_{ij}^e(\mu_r)$ and $m_{ij}^e(\tilde{\kappa})$ standard Gaussian quadrature is applied, which requires the integrand to be evaluated at a series of Gauss points and then the weighted sum is taken as the integrals' value. If the integrands in Eq. (6) are polynomials (i.e. when the mapping is affine) the resulting weighted sums become exact by choosing the correct number of Gauss points. However, for other mappings the sums will always remain an approximation to the integrals' true value. The vector valued functions $\hat{\mathbf{\Phi}}_i$ and $\hat{\operatorname{curl}}\, \hat{\mathbf{\Phi}}_i$ are always polynomials, which are explicit functions of (the derivatives of) the barycentric coordinates of the element, edge extension parameters [34] and the Legendre polynomials, and can be evaluated at Gauss points. A naive approach would be to recompute all the basis functions at each Gauss point for each element, but this would quickly become prohibitively expensive. To overcome

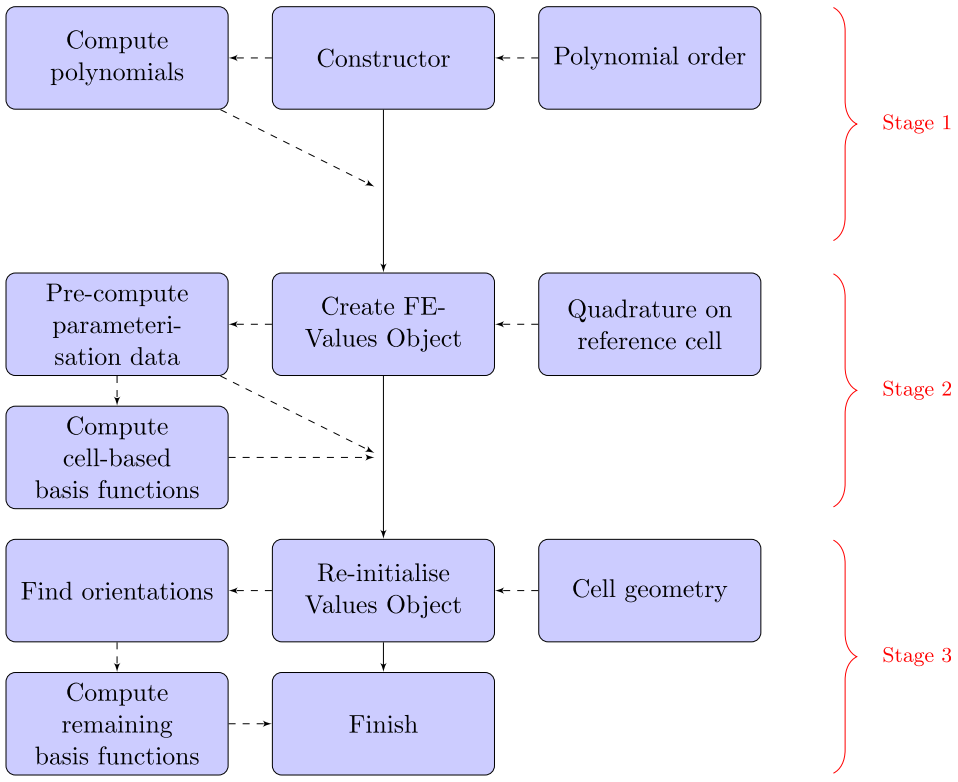


Fig. 5. Schematic demonstrating the typical workflow for computing the basis functions at quadrature points on a given mesh in `deal.II`.

this, we make an effort to pre-compute as much information about the possible edge and face parameterisations as possible. In the context of `deal.II`, we must set out three main stages in the construction of the basis functions. An overview of this process is shown in Fig. 5.

Stage 1 is the constructor of the finite element object and here we pre-compute data about the basis functions, which does not rely on knowledge of local cell information or on the degree of the quadrature rule to be applied. We also precompute the coefficients of integrated Legendre polynomials given in Eq. (4). These will later be used to form the basis function set defined in [34]. This information is then stored within the element's internal data structure.

In stage 2, an object, which provides access to the values of the basis functions, is initialised, with knowledge of the quadrature rule and the reference cell but not of the local cell (i.e. the orientation of its edges and faces is unknown at this point). The set of cell-based basis functions are independent of the orientation of the cell's edges and faces, and therefore these functions and their derivatives are evaluated at the quadrature points and stored within the internal data structure. The edge- and face-based basis functions depend on the orientation of the respective entities, which prevents them from being computed at this stage. Nonetheless, we can compute information about the possible combinations of edge and face parameterisations. The edge- and face-based basis functions require the evaluation of the integrated Legendre polynomials and their derivatives as well as the evaluation of barycentric coordinates, edge extension parameters and their derivatives. Therefore, we precompute these quantities at the quadrature points on the reference cell and store them in the internal data structure of the finite element.

Finally, in stage 3, an update call on a local cell is made once cell-specific information is available. At this point, we combine the precomputed data and local cell orientation in order to evaluate the basis functions values, any requested derivatives, the Jacobian values and quadrature weights. This may update values on a complete cell, or be restricted to a particular face or edge of a cell. The aim here is to compute only information that relies on knowledge of the local cell (i.e. a face or edge orientation) at this point and to rely on information pre-computed in the previous stages and access this information when required. We return these values on the reference cell (with the relevant orientation), which can then be transformed to the physical cell by mappings provided by the library.

5. Preconditioning and linear algebra

We form the global linear system, $Kx = (A + M)x = b$, by summing and assembling contributions from the local cells. The software library `deal.II` includes a range of direct solvers for the solution of the assembled system, but such approaches become prohibitively expensive for large problems. A range of preconditioning techniques have previously been proposed for the iterative solution of the linear system resulting from the discretisation of eddy current problems using Nédélec elements (for a review see [21] and references therein). In light of the previous success with the reduced basis gauging and a preconditioning strategy [21], which explicitly exploits the curl free nature of parts of the Schöberl–Zaglmayr Nédélec basis functions, we choose to apply the same approach here. The resulting preconditioning technique is robust with respect to $\tilde{\kappa}$, which is crucial given that its value is *small* in Ω_{NC} in order to provide regularisation to this region. We remark that the recent preconditioning strategy proposed by Grayver and Kolev [16] offers robustness with respect to h and opportunities may exist for further improvements by combining both techniques. For completeness, we briefly summarise the preconditioning strategy presented in [21] in the following.

The reduced gradient strategy involves skipping the gradient functions in Eq. (3) for cells in Ω_{NC} and only applying the regularisation term to the lowest order basis functions. Following the notation in [21] we then have the block matrix

$$K = \begin{bmatrix} K_{\mathcal{N}_0\mathcal{N}_0} & K_{\mathcal{N}_0 G} & K_{\mathcal{N}_0 N} \\ K_{G\mathcal{N}_0} & K_{GG} & K_{GN} \\ K_{N\mathcal{N}_0} & K_{NG} & K_{NN} \end{bmatrix} = \begin{bmatrix} A_{\mathcal{N}_0\mathcal{N}_0}(\mu_r) & 0 & A_{\mathcal{N}_0 N}(\mu_r) \\ 0 & 0 & 0 \\ A_{N\mathcal{N}_0}(\mu_r) & 0 & A_{NN}(\mu_r) \end{bmatrix} + \begin{bmatrix} M_{\mathcal{N}_0\mathcal{N}_0}(\tilde{\kappa}) & M_{\mathcal{N}_0 G}(\kappa) & M_{\mathcal{N}_0 N}(\kappa) \\ M_{G\mathcal{N}_0}(\kappa) & M_{GG}(\kappa) & M_{GN}(\kappa) \\ M_{N\mathcal{N}_0}(\kappa) & M_{NG}(\kappa) & M_{NN}(\kappa) \end{bmatrix} \tag{7}$$

where the subscripts $\mathcal{N}_0, G$ and N are used to distinguish between blocks derived from the lowest order Nédélec, gradient and non–gradient basis functions, respectively. The matrix K is complex symmetric but, as A is real symmetric and M is purely imaginary, we can recast $Kx = b$ as the real valued system

$$\begin{bmatrix} A & -M_{\mathrm{i}} \\ M_{\mathrm{i}} & A \end{bmatrix} \begin{bmatrix} x_r \\ x_{\mathrm{i}} \end{bmatrix} = \begin{bmatrix} b_r \\ b_{\mathrm{i}} \end{bmatrix},$$

where $x = x_r + \mathrm{i}x_{\mathrm{i}}, b = b_r + \mathrm{i}b_{\mathrm{i}}$ and $M = \mathrm{i}M_{\mathrm{i}}$, which is no longer a symmetric system. Symmetry is regained by rewriting the system as

$$\begin{bmatrix} A & -M_{\mathrm{i}} \\ -M_{\mathrm{i}} & -A \end{bmatrix} \begin{bmatrix} x_r \\ x_{\mathrm{i}} \end{bmatrix} = \begin{bmatrix} b_r \\ -b_{\mathrm{i}} \end{bmatrix},$$

Notice that each block within (7) may be rewritten in this way and, with a slight abuse of notation, we henceforth denote this as $Kx = b$. We then adopt the block Jacobi preconditioner

$$D = \begin{bmatrix} K_{\mathcal{N}_0\mathcal{N}_0} & 0 & 0 \\ 0 & D_{GG} & 0 \\ 0 & 0 & D_{NN} \end{bmatrix}, \tag{8}$$

advocated in [21] combined with the FGMRES iterative solver [28], which is available in the `deal.II` library. Still following the definitions in [21], note that the action of the inverse of this preconditioner to a vector $f := f_r - \mathrm{i}f_{\mathrm{i}}$ now involves

$$K^{-1}_{\mathcal{N}_0\mathcal{N}_0} f_{\mathcal{N}_0} = \begin{bmatrix} A_{\mathcal{N}_0\mathcal{N}_0}(\mu_r) & -M_{\mathcal{N}_0\mathcal{N}_0}(|\tilde{\kappa}|) \\ -M_{\mathcal{N}_0\mathcal{N}_0}(|\tilde{\kappa}|) & -A_{\mathcal{N}_0\mathcal{N}_0}(\mu_r) \end{bmatrix}^{-1} \begin{bmatrix} f_{\mathcal{N}_0 r} \\ -f_{\mathcal{N}_0 \mathrm{i}} \end{bmatrix},$$

$$D^{-1}_{GG} f_G = \begin{bmatrix} 0 & -\text{block diag}(M_{GG})(|\kappa|) \\ -\text{block diag}(M_{GG})(|\kappa|) & 0 \end{bmatrix}^{-1} \begin{bmatrix} f_{Gr} \\ -f_{G\mathrm{i}} \end{bmatrix},$$

$$D^{-1}_{NN} f_N = \begin{bmatrix} \text{block diag}(A_{NN}(\mu_r) + M_{GG}(|\tilde{\kappa}|)) & 0 \\ 0 & -\text{block diag}(A_{NN}(\mu_r) + M_{GG}(|\tilde{\kappa}|)) \end{bmatrix}^{-1} \begin{bmatrix} f_{Nr} \\ -f_{N\mathrm{i}} \end{bmatrix}.$$

In each case we employ a factorisation using a sparse direct solver and use this to apply the action of the inverse of preconditioner to a vector. However, for large problems, still further efficiencies can be gained by using a conjugate gradient solver to apply the action of the inverse.

6. Numerical examples

In order to establish that our method successfully overcomes the sign conflict problem, we apply the method to a simple problem with a known solution and more complicated benchmarks where comparison to derived values may be made. We begin with a simple trigonometric model solution to a vector-wave equation. This is done by comparison to the original `deal.II` implementation on a series of simple domains. To benchmark the method for eddy current problems we consider a series of examples drawn from the established TEAM (Testing of Electromagnetic Analysis Methods) benchmark problems [33]. Specifically, we present a conducting sphere in a uniform magnetic background field (TEAM benchmark problem 6) and then a similar benchmark set on a multiply-connected domain consisting of a conducting solid torus in a uniform magnetic background field. Finally, we consider the TEAM benchmark problem 7, which consists of an axisymmetric conducting plate with a hole subject to a rotational background field generated by a coil.

6.1. Trigonometric model solution

To illustrate the issues with the previous `deal.II` implementation, and establish that the sign conflict problem has been overcome, we solve a simple model problem. Consider

$$\text{curl curl } \boldsymbol{E} - \pi^2 \boldsymbol{E} = \boldsymbol{0} \quad \text{in } \Omega, \tag{9a}$$

$$\boldsymbol{n} \times \boldsymbol{E} = \boldsymbol{n} \times \boldsymbol{E}_{\text{exact}}, \quad \text{on } \partial\Omega, \tag{9b}$$

where we set

$$\boldsymbol{E}_{\text{exact}}(x_1, x_2) := \pi \begin{pmatrix} \cos(\pi x_1)\sin(\pi x_2) \\ -\sin(\pi x_1)\cos(\pi x_2) \\ 0 \end{pmatrix}. \tag{10}$$

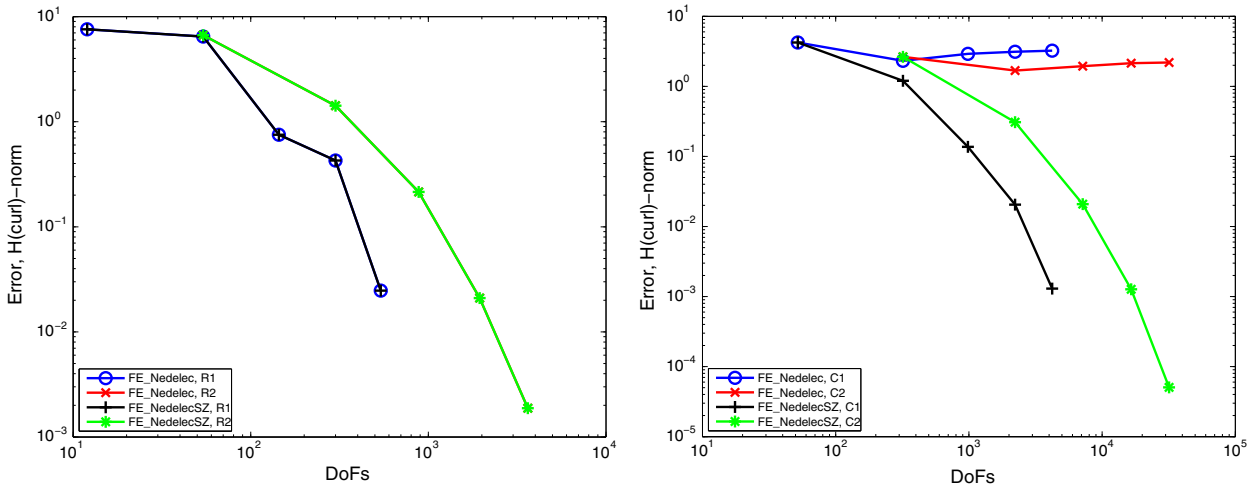
Although this is not an eddy current problem, the weak form of this problem can be obtained by setting $\boldsymbol{A} = \boldsymbol{E}$, $\boldsymbol{J}^s = \boldsymbol{0}, \mu_r = 1$, fixing $\tilde{\kappa} = -\pi^2$ in Eq. (2) and modifying the boundary condition in $\boldsymbol{V}$ accordingly.

Firstly, we consider Ω to be the rectangular domain $[0,1]^3$ and then, secondly, the cylindrical domain $\left\{(x_1,x_2) : \sqrt{x_1^2 + x_2^2} \leqslant 0.5\right\} \times [0,1]$. Here we deliberately choose coarse discretisations: in the case of rectangular domain, we construct a mesh $R1$ consisting of a single element and then a mesh $R2$ consisting of 8 elements as a result of 1 uniform refinement. For the cylindrical domain, we construct a mesh $C1$ consisting of 10 elements and then perform a single uniform refinement in order to generate a mesh $C2$ of 80 elements. In each case, we consider the application of uniform p–refinements by the application of elements of order $p = 0,1,2,3,4$, in turn, and measure the error $\boldsymbol{e} := \boldsymbol{E}_{exact} - \boldsymbol{E}_{hp}$ in the $\boldsymbol{H}(\text{curl})$ norm

$$\|\boldsymbol{e}\|_{\boldsymbol{H}(\text{curl})} := \left(\int_\Omega |\boldsymbol{e}|^2 + |\text{curl } \boldsymbol{e}|^2 \mathrm{d}\Omega \right)^{1/2},$$

on the rectangular and cylindrical domains using both the original and the new Nédélec elements in `deal.II` where $\boldsymbol{E}_{hp}$ denotes the approximate finite element solution. As the properties of the resulting linear system are different from the eddy current problem, but are all of small computational size, we employ a simple direct solver when solving the discretised version of Eq. (9).

The rectangular meshes $R1, R2$ have been constructed using `deal.II`'s built-in `GridGenerator` functionality. This results in elements whose orientation conforms to the standard expected in `deal.II`. Therefore, we expect that performing p–refinement on these meshes should produce a downward sloping curve indicating that exponential convergence is being obtained. This is indeed illustrated for both the original and new Nédélec elements in Fig. 6(a) with curves for both elements being almost indistinguishable from each other on the chosen scale. However, although the meshes $C1, C2$ are also constructed using `deal.II`'s in-built `GridGenerator` functionality, they no longer conform to the standard expected in `deal.II`. The issue here is that by wrapping the elements around a circular cylinder there is a sign-conflict for horizontal edge–based and face–based basis functions, which is not resolved by the software. On comparing the performance of p–refinement for the meshes $C1, C2$ shown in Fig. 6(b) we observe how this issue translates into a lack of exponential convergence for the old Nédélec element. However, the new Nédélec element, in which the sign conflict issue has been addressed, exhibits the



(a) Cubic domain (b) Cylindrical domain

Fig. 6. Trigonometric model solution: comparison of the error measured in the $\boldsymbol{H}(\mathrm{curl})$-norm for p–refinement on (a) cubic and (b) cylindrical domains.

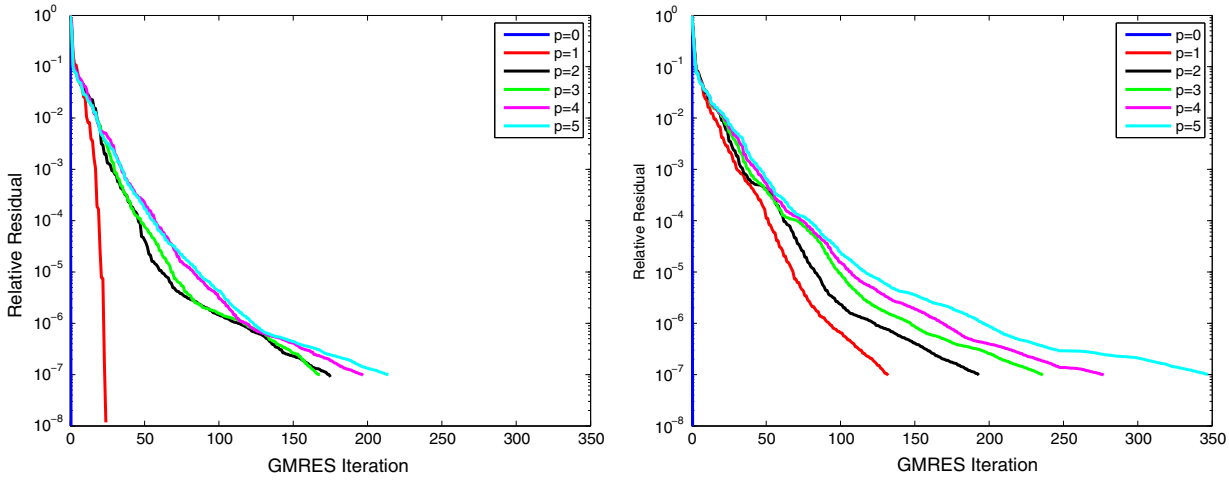
correct expected exponential convergence behaviour. Henceforth, we employ the new Nédélec element for the remaining examples presented in this work.

6.2. Conducting sphere in a uniform magnetic field

We next consider a conducting sphere of radius $R = 0.05$ m placed in a uniform background field with magnetic flux density $\boldsymbol{B}_0 = \mu_0 \boldsymbol{H}_0 = (0,\ 0,\ 1)^T$ and angular frequency $\omega = 100\pi$ rad s^{-1}, as defined by the TEAM benchmark problem 6 [33]. The sphere has material parameters $\sigma_* = 10^7$ S m^{-1}, $\mu_* = 20\mu_0$ and is placed in an unbounded region of free space. The corresponding solution to this problem has a known analytical solution [30], which we use for comparison in the following.

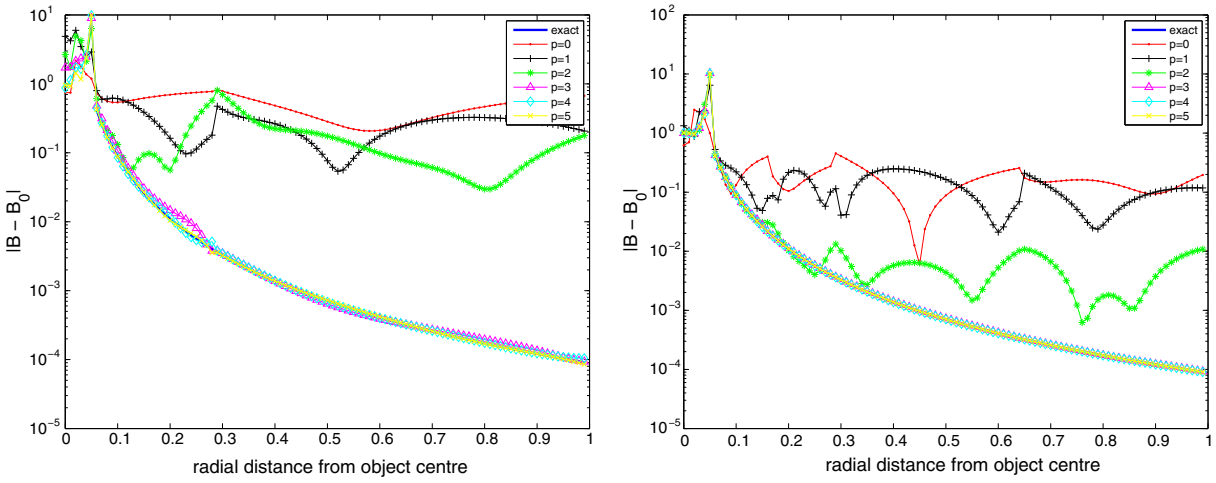
Computationally, we set up a domain Ω, which is truncated at a radius of 1m and on $\partial\Omega$ we apply the known analytical solution. By using `deal.II`'s `GridGenerator` tools, we generated the meshes $S1$ and $S2$ consisting of 19 and 151 hexahedral elements, respectively, to discretise the problem. On each of these meshes we employ uniform p–refinement by considering elements of order $p = 0, 1, 2, 3, 4, 5$, in turn, and consider the convergence history of the preconditioned GMRES solver in Fig. 7 when a regularisation parameter $\varepsilon = 10^{-6}$ is employed. In this figure, we exhibit a fast convergence of the preconditioned solver down to relative residual of $TOL = 10^{-7}$, which is the specified tolerance for this problem. The growth in number of iterations for increasing polynomial degree is moderate. The convergence behaviour and number of iterations is similar to that presented in [21].

Next, in Fig. 8, we show $|(\boldsymbol{B}_{hp} - \boldsymbol{B}_0)(\boldsymbol{x})|$, where $\boldsymbol{B}_{hp} = \mu \boldsymbol{H}_{hp}$, along a radial axis defined by $0 \leqslant \sqrt{x_1^2 + x_2^2 + x_3^2} \leqslant 0.5$m, computed on the meshes $S1$ and $S2$ for p–refinement and make comparisons with the known analytical solution. We obverse that, for low p on these coarse meshes, the solution is inaccurate; however, by increasing p, the solution rapidly converges towards the analytical result. In the case of mesh $S1$, which is extremely coarse, there are still some small differences between the numerical solution and the analytical result, even at high p. This is due to the fact that the sphere geometry has been represented approximately using quadratic geometry interpolation functions, which prevents the solution being captured more accurately with only 19 elements. However, by increasing the number of elements to 151 for mesh $S2$, the computed numerical result for $p \geqslant 3$ is indistinguishable from the exact on this scale.



(a) Mesh $S1$ (b) Mesh $S2$

Fig. 7. Conducting sphere in a uniform magnetic field (TEAM 6): comparison of the GMRES residual history.



(a) Mesh $S1$ (b) Mesh $S2$

Fig. 8. Conducting sphere in a uniform magnetic field (TEAM 6): comparison of the magnitude of the perturbed magnetic field, $|(\boldsymbol{B}_{hp} - \boldsymbol{B}_0)(\boldsymbol{x})|$, along the radial axis $0 \leqslant \sqrt{x_1^2 + x_2^2 + x_3^2} \leqslant 0.5$ m for meshes $S1$ and $S2$ and $p = 0, 1, 2, 3, 4, 5$.

6.3. Conducting solid torus in the presence of a uniform magnetic field

For the next eddy current example we consider a multiply connected domain consisting of a conducting solid torus of minor and major axes 0.01 m and 0.02 m, respectively, in the same uniform background field, $\boldsymbol{B}_0$ as specified in Section 6.2 with angular frequency $\omega = 133.5$ rad s^{-1}. The torus has material parameters $\sigma_* = 5.96 \times 10^7$ S m^{-1}, $\mu_* = \mu_0$ and is placed in an unbounded region of free space. Although this problem does not have an analytical solution, an asymptotic formula [6,20] is available for the perturbed magnetic field $(\boldsymbol{H} - \boldsymbol{H}_0)(\boldsymbol{x}) = \mu_0^{-1}(\boldsymbol{B} - \boldsymbol{B}_0)(\boldsymbol{x})$ at positions $\boldsymbol{x}$ away from small conducting objects, which permits us to make comparisons.

We truncate the unbounded domain in the form of a cube $[-0.25, 0.25]^3$ m^3 centred about the object and the origin. On $\partial\Omega$ we use the aforementioned asymptotic formula to generate appropriate (approximate) boundary conditions for $\boldsymbol{n} \times \mu_r^{-1}\text{curl}\, \boldsymbol{A}|_{\partial\Omega} = \mu_0 \boldsymbol{n} \times \boldsymbol{H}|_{\partial\Omega}$. The commercial hexahedral mesh generation tool `Cubit` is used to generate the meshes $T1$ and $T2$, which consist of 1480 and 4600 elements, respectively. The generation of such meshes is a non–trivial task and requires careful dividing of the domain using symmetry. We cut the domain along all axes of symmetry and then work only with one corner of the domain. In this corner, we cut through the centre radius of the torus, in the x_3–direction, with a cylinder to connect the surface of the cylinder to the boundary. We then cut with another cylinder, in the same direction, a distance away from the torus. The curved surfaces of the cylinders are used to provide a grading outwards towards the boundary by setting the target spacing lower on the central cylinder, higher on the outer cylinder and highest on the boundary surfaces. We employ `Cubit`'s polyhedral meshing scheme across the entire volume and use smoothing to avoid poorly formed elements (which typically lie within the region near the inside of the torus). An illustration of the mesh $T2$ is shown in Fig. 9. Note that aforementioned cuts are only for the purposes of mesh generation and are not required as part of the *hp*–finite element procedure for approximating $\boldsymbol{A}$ on multiply connected geometries. On each of these meshes, we employ uniform p–refinement by considering elements of order $p = 0, 1, 2, 3, 4$, in turn, and consider the convergence history of the preconditioned GMRES solver in Fig. 10 when a regularisation parameter $\varepsilon = 10^{-6}$ is employed. The exhibited convergence behaviour is similar to that shown in Fig. 8 and we again employ the same tolerance.

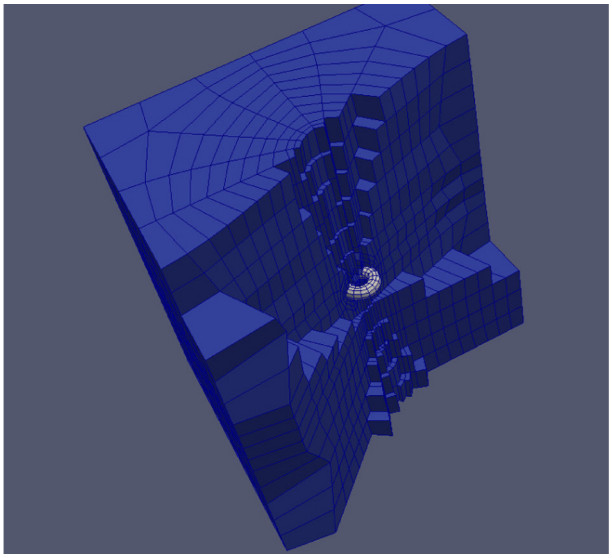
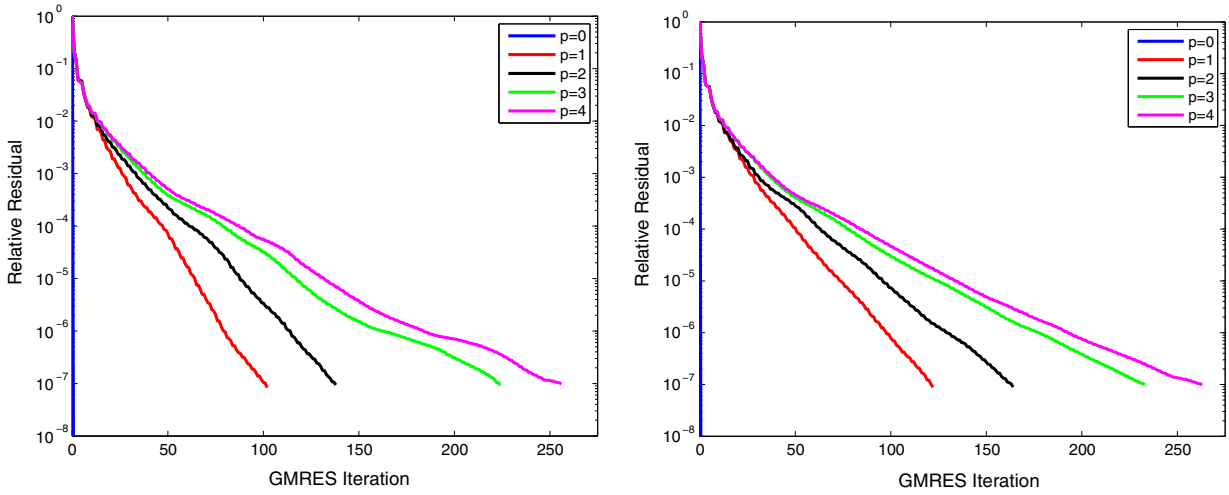


Fig. 9. Conducting solid torus in a uniform magnetic field: an illustration of the mesh $T2$ with 4600 elements.

In Fig. 11, we show $|(\boldsymbol{B}_{hp} - \boldsymbol{B}_0)(\boldsymbol{x})|$, along radial axis defined by $0 \leqslant \sqrt{x_1^2 + x_2^2 + x_3^2} \leqslant 0.25$ m for the meshes $T1$ and $T2$ and p–refinement. In this figure, we also make comparisons with the asymptotic formula for small objects, denoted by PT, outside the object (since it is only for points away from the object). By performing p–refinement, we observe that the computed solution rapidly converges towards the PT line. However, $T1$ does not converge fully to the prediction of $|(\boldsymbol{B} - \boldsymbol{B}_0)(\boldsymbol{x})|$ from the asymptotic formula, whereas $T2$ converges convincingly to the prediction. This can be explained by the poor capture of the geometry in $T1$, with only 8 points capturing the circular boundary of the torus, whereas for $T2$ there are 16. Moreover, on $T2$ for $p \geqslant 2$, the solution is almost indistinguishable in both cases from the asymptotic formula on the chosen scale.

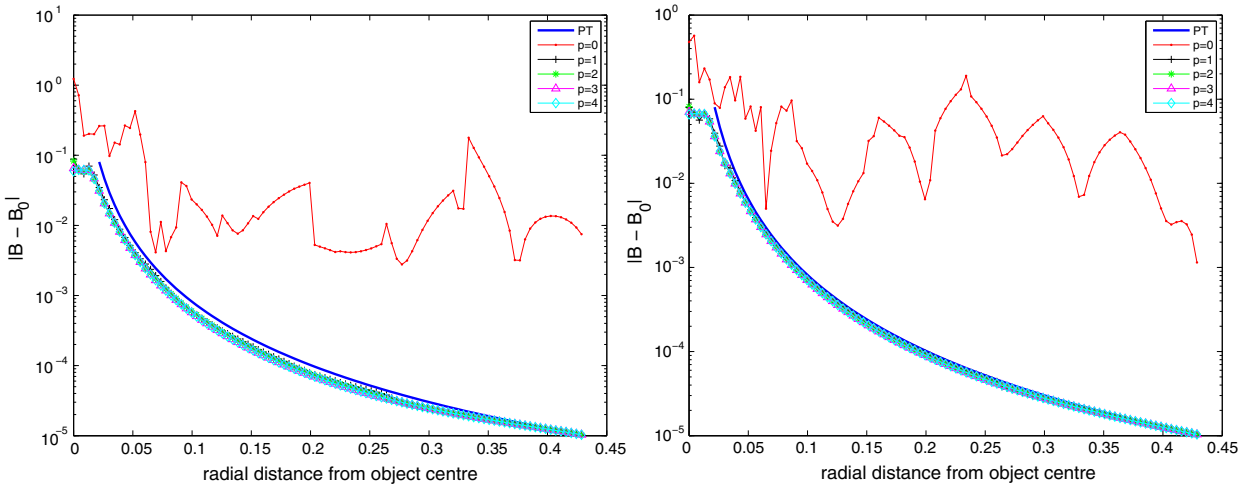
6.4. Conducting plate with a hole in a rotational field

Finally, we consider the solution of the TEAM benchmark problem 7. This problem has a multiply connected domain consisting of a conducting plate with a handle (hole), which is placed eccentri-



(a) Mesh $T1$ (b) Mesh $T2$

Fig. 10. Conducting torus in a uniform magnetic field: comparison of the GMRES residual history.



(a) Mesh $T1$ (b) Mesh $T2$

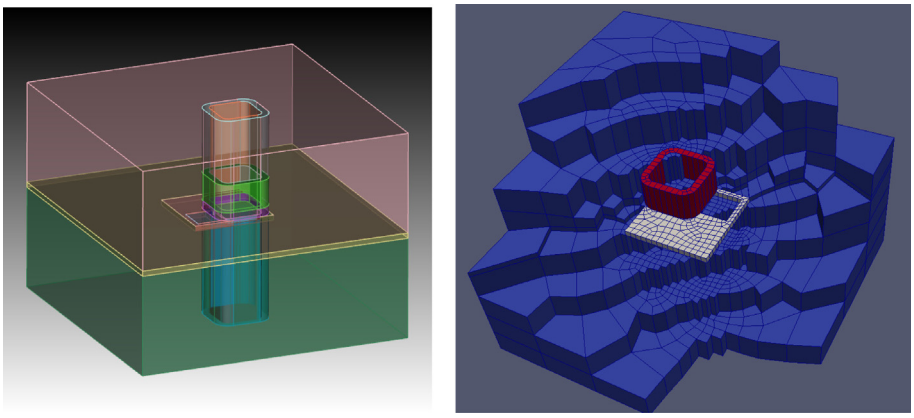
Fig. 11. Conducting torus in a uniform magnetic field: comparison of the magnitude of the perturbed magnetic field, $|(\boldsymbol{B}-\boldsymbol{B}_0)(\boldsymbol{x})|$, along the radial axis $0 \leqslant \sqrt{x_1^2+x_2^2+x_3^2} \leqslant 0.25$ m for meshes $T1$ and $T2$ and $p = 0, 1, 2, 3, 4$.

cally and is set asymmetrically in a rotational magnetic field. The plate dimensions have been previously quoted in the literature [14] and has material parameters $\sigma_* = 3.256 \times 10^7$ S m^{-1} and $\mu_* = \mu_0$. The rotational background field is generated with a coil placed in an unbounded region of free space with Ampère turn 2742 and the frequencies of interest are $\omega = 100\pi$ rad s^{-1} and $\omega = 400\pi$ rad s^{-1}.

We create a computational domain Ω by truncating the unbounded region of free space in the form of the rectangular domain $[-0.353, 0.647] \times [-0.353, 0.647] \times [-0.365, 0.384]$ m^3 with the plate centred inside. The commercial mesh generation tool `Cubit` is then employed to generate a mesh $P1$ of 4815 hexahedral elements for this complex multiply connected configuration, as shown in Fig. 12(b). This requires some care in the generation of the mesh and guidance for the mesher is required. The domain is cut, by extruding the coil in the positive and negative x_3-directions, cutting through the plate and imprinting onto the boundaries of the domain. An illustration of this is shown in Fig. 12(a). Additionally, we bound the plate by introducing cuts through the entire volume along its top and bottom surfaces. This then allows the use of the sweep scheme provided by `Cubit`. We also grade the mesh by using a larger target spacing on the boundary surfaces than that in the plate and coil volumes. We emphasise that these cuts are only for the purpose of mesh generation. In a similar manner to that previously described in [21], which used an unstructured tetrahedral mesh, we employ uniform p–refinement and consider elements of order $p = 0, 1, 2, 3, 4$, in turn (Fig. 13). In Fig. 14, we plot the quantity

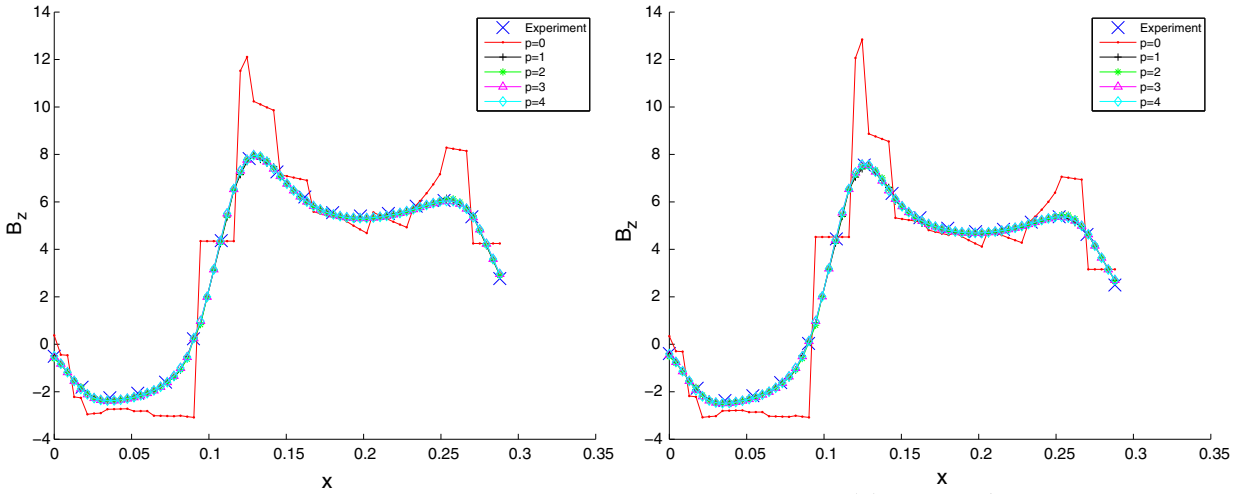
$$\chi = \text{sign}\sqrt{\chi_R^2 + \chi_I^2}, \tag{11}$$

where χ_R, χ_I are applied to the real and imaginary parts of $J_y^e = \sigma E_y$ and B_z, along the line defined by $y = 72 \times 10^{-3}$ m, $z = 19 \times 10^{-3}$ m. This line lies on the surface of the conductor and allows us to make comparisons with the measured values in the benchmark [14]. We observe that increasing p leads to a rapid convergence of the solution towards the measured values and follows a similar trend to that shown in [21] where the agreement of performing these refine-



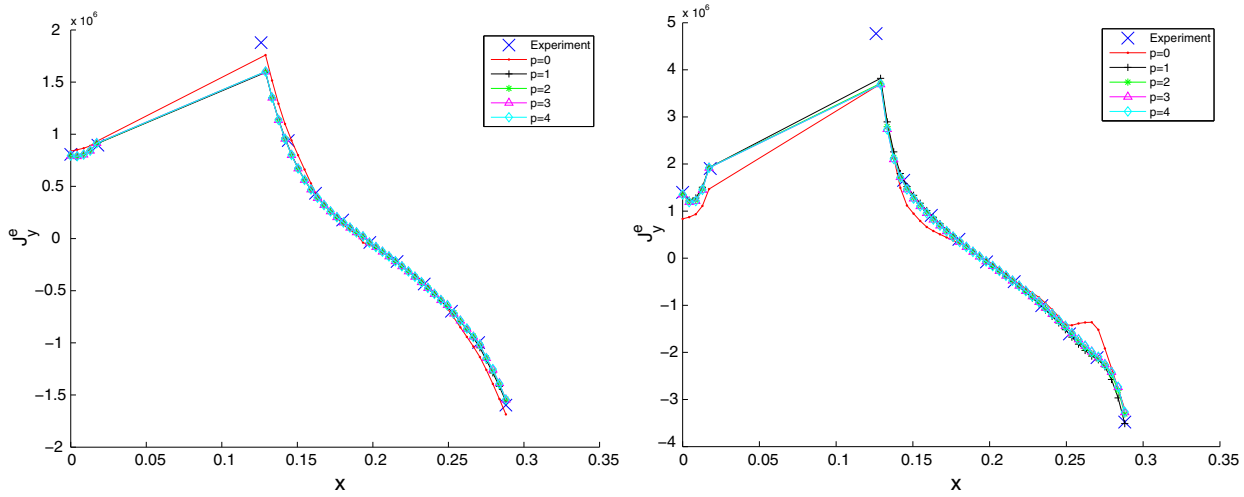
(a) Cuts (b) Mesh

Fig. 12. Conducting plate with a hole (TEAM 7): an illustration of the construction of mesh $P1$ with 4815 elements. (a) Shows the cuts required by Cubit. (b) Shows resulting mesh with the plate displayed in grey, the coil in red and the background region in blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



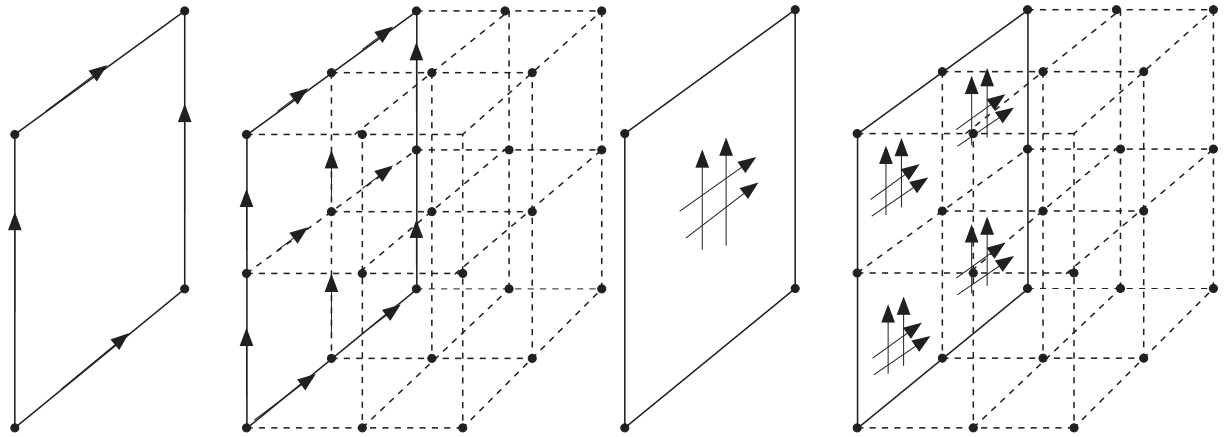
(a) 100πrad/s (b) 400πrad/s

Fig. 13. Conducting plate with a hole (TEAM 7): convergence of B_z on the line $y = 72 \times 10^{-3}$ m, $z = 34 \times 10^{-3}$ m for increasing p. Comparison is made against reference experimental values [14].



(a) 100πrad/s (b) 400πrad/s

Fig. 14. Conducting plate with a hole (TEAM 7): convergence of J_y^e on the line $y = 72 \times 10^{-3}$ m,$z = 19 \times 10^{-3}$ m for increasing p. Comparison is made against reference experimental values [14].



(a) Inheritance of Edge Parameterisation (b) Inheritance of Face Parameterisation

Fig. 15. Mesh h–refinement resulting in a non-conforming discretisation: Inheritance of edge and face parameterisation from parent to child elements on exterior entities.

ments was also found to be excellent. For this example, we have chosen not to show the convergence history for the preconditioned GMRES solver as the behaviour is similar to the previous examples. In particular, for the TEAM benchmark problem, a maximum of 375 iterations is required at 100π rad s^{-1} and 1100 iterations at 400π rad s^{-1} in order to reach the desired tolerance of $TOL = 10^{-7}$ for $p = 4$ elements.

7. Extensions to non–uniform *hp*–refinement

Following a finite element analysis, an automated or user-defined mesh refinement can be generated in order to provide an improved discretisation for a future simulation. The automatic (adaptive) selection of elements for refinement may be achieved by applying error estimators (or error indicators) in order to identify suitable regions. In the context of the *hp*–finite elements described in this work, possibilities exist for not only the subdivision (or h–refinement) of elements but also the for the local polynomial (or p–) enrichment of elements. There is a large literature on the development of error estimators, error indicators and procedures for deciding whether either a h– or a p–refinement of a particular element should be performed: for further details see [2,9,10] and references therein.

The use of hierarchic finite element basis functions offers significant advantages for the treatment of local p–refinements leading to the discretisations with non–uniform polynomial degree. In such cases, once elements have been tagged for polynomial enrichment, the *minimum rule* [3,25] can be applied to restrict the polynomial degree associated with an edge or face. This principle immediately carries over to the Nédélec *hp*–finite element basis of Schöberl and Zaglmayr as well as the edge and face parameterisation procedure described in Section 3.

When elements are selected for h-refinement, the approach in unstructured meshes is to also refine neighbouring elements to eliminate any hanging nodes that have been introduced. This process may, where appropriate, use a range of element shapes as well as different splitting patterns to ensure the elements remain of *good* quality and to minimise the number of elements subdivided [36]. However, for structured hexahedral meshes, this is not practical since a single element subdivision would result in a uniform global h–refinement. Instead, non-conforming meshes obtained from the subdivision of a (parent) element into either 8, 4 or 2 smaller hexahedra (called children), in three dimensions, without the refinement of adjacent elements, are used.[2] Constraints are then introduced to deal with the hanging nodes, edges and faces that are introduced as a result [3,25].

In Nédélec elements the constraints for non-conforming meshes must be developed for matching the tangential components of the basis functions. This means the tangential components of the edge based basis functions associated with the hanging edges must be matched with the edge based basis functions of the un-refined neighbours. Similarly, the tangential components of the face based basis functions on the hanging faces must be matched with the face based basis functions of the un-refined neighbours [25,9,10]. We illustrate this in Fig. 15 for the case of subdivision of a parent reference hexahedron into 8 children. By matching tangential constraints in a reference configuration, constraints containing weights can be developed which can also be applied to more general shaped hexahedra (or quadrilaterals) through an affine or isoparametric coordinate transformation [26]. Note that it is necessary to develop new sets of constraints depending on the type of element subdivision in the reference configuration (eg subdivision of a parent element into 8, 4 or 2 children). The resulting constraints then inform the assembly procedure of how to weight the degrees of freedom in the sub-divided element. The previous Nédélec implementation in `deal.II` constructed weights for meshes consisting of rectangular blocks where the edge and face parameterisations can all be pre-assigned and the constraints pre-computed.

In the case of a general structured/unstructured hexahedral mesh, the constraints will also depend on the edge and face orientations. In Algorithm 2 we propose an extension to Algorithm 1 for generating edge and face orientations in a general non-conforming hexahedral mesh. This will ensure the edge and face orientations on the exterior edges and faces of the children match those of the parent's neighbour, greatly simplifying the generation of the constraints. We expect that the constraints for the edge-based basis functions can be pre–computed (in a similar manner to that detailed by [26]) for different configurations (up to some specified p). The constraints for the face-based basis functions will depend on the orientations of the (hanging) edges and so we anticipate that it will be necessary to compute these at run-time.

[2] The splitting is performed on the (parent) reference hexahedron where we allow at most one subdivision in each coordinate direction with 8, 4 or 2 children resulting from splitting in three, two or one coordinate direction, respectively. The split reference element is then mapped to the (parent) physical element. In two dimensions, a (parent) reference quadrilateral can similarly be refined into 4 or 2 smaller children by splitting in either both or one coordinate direction, respectively.

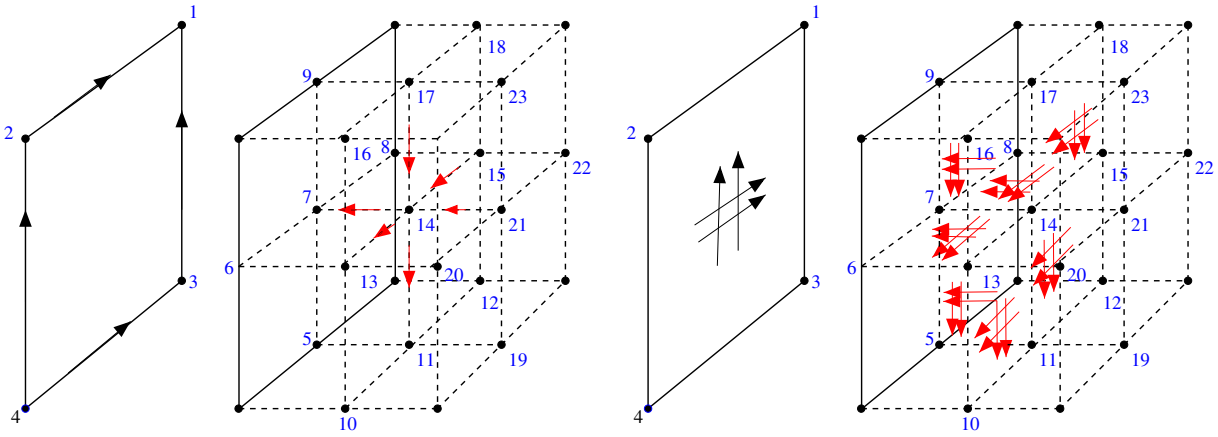


Fig. 16. Mesh *h*–refinement resulting in a non-conforming discretisation: generation of (a) edge and (b) face parameterisations on interior entities using Algorithm 1. Global vertex numbers are shown in blue and the chosen edge and face parameterisations on the interior entities are shown in red. Note that only some of the face parameterisations are shown for clarity. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Algorithm 2. An Algorithm for the construction of edge and face parameterisations on a general non-conforming structured/unstructured hexahedral mesh

1. Generate an initial conforming structured/unstructured hexahedral mesh
2. Apply Algorithm 1 to generate a set of unique global edge and face parameterisations for the conforming mesh
3. Use a user-defined (or automated) procedure to select elements for *h*-refinement
4. Elements tagged for refinement are subdivided into (8, 4 or 2) children. Vertex numbers introduced for new hanging nodes
5. The orientations of the exterior edge and faces of the children are inherited from the parent as shown in Fig. 15
6. Interior edge and face parameterisations in the sub-divided elements are chosen according to Algorithm 1 as shown in Fig. 16
8. If further *h*-refinements are desired goto to step 3

Such an extension would represent a further significant amount of implementation, which unfortunately is beyond the time constraints of the funding in our current project. We hope that the summary included here will be beneficial to software developers wishing to pursue full *hp*–refinement using Nédélec elements on general structured and unstructured non-conforming hexahedral meshes in `deal.II`.

8. Conclusions

This work has described an approach for resolving the sign conflict issue associated with *hp*–Nédélec finite element basis functions on conforming meshes of hexahedra and presented an implementation in the open source finite element library `deal.II`. These additions greatly enhance the capability of the software to solve electromagnetics problems on a wide class of geometries and, as an illustrations of this, we have included the solution of a series of eddy current benchmark problems including two taken from the TEAM problem set. In each case, we are able to demonstrate that the new Nédélec element implementation in `deal.II` is able to achieve high levels of accuracy and overcome the limitations of the old `FE_NEDELEC` class. Moreover, with the implementation of the preconditioner from [21], we are able to compute solutions to these challenging problems rapidly in a small number of iterations, comparable to the previous *hp*–Nédélec finite element tetrahedral discretisation. The implementation will be made publicly available as part of the `deal.II` library under the name of `FE_NedelecSZ` in recognition of the efforts of Schöberl and Zaglmayr. Futhermore, we have set out how implementation can be extended to non–uniform *hp*–refinements on non–conforming meshes. It is hoped that this will be of use for researchers and users of the software for applications in electromagnetism.

Acknowledgement

The authors gratefully acknowledge the financial support received from EPSRC – United Kingdom in the form of the Grant EP/K023950/1.

References

[1] Ainsworth M, Coyle J. Hierarchic finite element bases on unstructured tetrahedral meshes. Int J Numer Methods Eng 2003;58:2103–30.
[2] Ainsworth M, Oden JT. A-posteriori error estimation in finite element analysis. Wiley; 2000.
[3] Ainsworth M, Senior B. Aspects of an adaptive *hp*-finite element method: adaptive strategy, conforming approximation and efficient solvers. Comput Methods Appl Mech Eng 1997;150:65–87.
[4] Alnæs MS, Blechta J, Hake J, Johansson A, Kehlet B, Logg A, et al. The FEniCS project version 1.5. Arch Numer Softw 2015;3(100).
[5] Ammari H, Buffa A, Nédélec JC. A justification of eddy currents model for the Maxwell equations. SIAM J Appl Math 2000;60:1805–23.
[6] Ammari H, Chen J, Chen Z, Garnier J, Volkov D. Target detection and characterization from electromagnetic induction data. J Math Pures Appl 2014;101:54–75.
[7] Bangerth W, Heister T, Heltai L, Kanschat G, Kronbichler M, Maier M, et al. The `deal.II` library, version 8.2. Arch Numer Softw 2015;3.
[8] Cheney M, Isaacson D, Newell JC. Electrical impedance tomography. SIAM Rev 1999;41:85–101.
[9] Demkowicz L. Computing with *hp*-adaptive finite elements: volume 1: one and two dimensional elliptic and Maxwell problems. Boca Raton FL, USA: Chapman and Hall; 2007.
[10] Demkowicz L, Kurtz J, Pardo D, Paszynski M, Rachowicz W, Zdunek A. Computing with *hp*-adaptive finite elements: volume 2 frontiers three dimensional elliptic and Maxwell problems with applications. Boca Raton FL, USA: Chapman and Hall; 2007.
[11] Dular P, Geuzaine C. GetDP reference manual: the documentation for GetDP, a general environment for the treatment of discrete problems. <http://www.geuz.org/getdp/>.
[12] Schöberl J et al. NGSolve finite element library. <http://sourceforge.net/projects/ngsolve/> [accessed 7/8/2015].
[13] Solin P et al. Hermes - higher-order modular finite element system (user guide). <http://hpfem.org/> [accessed 7/8/2015].

[14] Fujiwara K, Nakata T. Results for benchmark problem 7 (asymmetrical conductor with a hole). COMPEL: Int J Comput Math Electr Electron Eng 1990;9:137–54.
[15] Grayver AV. Parallel three-dimensional magnetotelluric inversion using adaptive finite-element method. Part I: theory and synthetic study. Geophys J Int 2015;202:584–603.
[16] Grayver AV, Kolev TV. Large-scale 3D geoelectromagnetic modeling using parallel adaptive high-order finite element method. Geophysics 2015;80.
[17] Griffiths H. Magnetic induction tomography. In: Holder DS, editor. Electrical impedance tomography - methods, history and applications. Bristol: IOP Publishing; 2005.
[18] Hiptmair R. Boundary element methods for eddy current computation. Lect. notes comput. sci. eng., vol. 28.
[19] Sandia National Laboratories. CUBIT Toolkit; 2015. <https://cubit.sandia.gov> [accessed 7/8/2015].
[20] Ledger PD, Lionheart WRB. Characterising the shape and material properties of hidden targets from magnetic induction. IMA J Appl Math 2015;80:1776–98.
[21] Ledger PD, Zaglmayr S. *hp* finite element simulation of three-dimensional eddy current problems on multiply connected domains. Comput Methods Appl Mech Eng 2010;199:3386–401.
[22] Marsic N, Geuzaine C. Efficient finite element assembly of high order Whitney forms. IET Sci Measure Tech 2015;9.
[23] Monk P. The finite element method for Maxwell's equations. Oxford: Clarendon Press; 2003.
[24] Morris A, Griffiths H, Gough W. A numerical model for magnetic induction tomographic measurements in biological tissues. Physiol Measure 2001;22:113–9.
[25] Dolezel I, Solin P, Segeth K. Higher-order finite element methods. Chapman and Hall, CRC Press; 2003.
[26] Rachowicz W, Demkowicz L. An hp-adaptive finite element method for electromagnetics part II: a 3D implementation. Int J Numer Methods Eng 2002;53:147–80.
[27] Rodriguez AA, Valli A. Eddy current approximation of Maxwell equations: theory algorithms and applications. Springer; 2010.
[28] Saad Y. A flexible inner-outer preconditioned GMRES algorithm. SIAM J Sci Comput 1993;14(2):461–9.
[29] Schöberl J, Zaglmayr S. High order Nédélec elements with local complete sequence properties. Int J Comput Math Electr Electron Eng (COMPEL) 2005;24:374–84.
[30] Smythe WR. Static and dynamic electricity. New York: McGraw-Hill; 1950.
[31] Soleimani M, Lionheart WRB. Absolute conductivity reconstruction in magnetic induction tomography using a nonlinear method. IEEE Trans Med Imaging 2006;25:1521–30.
[32] Soleimani M, Lionheart WRB, Peyton AJ. Image reconstruction for high contrast conductivity imaging in mutual induction tomography for industrial applications. IEEE Trans Instrument Measure 2007;56:2024–32.
[33] Turner LR, Davey K, Emson CRI, Miya K, Nakata T, Nicolas A. Problems and workshops for eddy current code comparison. IEEE Trans Magnet 1988;24 (1):431–4.
[34] Zaglmayr S. High order finite element methods for electromagnetic field computation PhD thesis. Austria: Institut für Numerische Mathematik, Johannes Kepler Univerität Linz; 2006.
[35] Zaglmayr S, Schöberl J. High order Nédélec elements with local complete sequence properties. Int J Comput Math Electr Electron Eng (COMPEL) 2005;24:374–84.
[36] Zienkiewicz OC, Taylor RL, Nithiarasu P. The finite element method for fluid dynamics. 6th ed. Amsterdam: Elsevier; 2005.
[37] Zolgharni M. Magnetic induction tomography for imaging cerebral stroke PhD thesis. Swansea University; 2010.
[38] Zolgharni M, Ledger PD, Armitage DW, Holder DS, Griffiths H. Imaging cerebral haemorrhage with magnetic induction tomography: numerical modelling. Physiol Measure 2009;30:187–200.
[39] Zolgharni M, Ledger PD, Griffiths H. Forward modelling of magnetic induction tomography: a sensitivity study for detecting haemorrhagic cerebral stroke. Med Biol Comput 2009;47:1301–13.