Adaptive Workflow for Simulation of RF Heaters

Morteza H. Siboni[a,*], Mark S. Shephard[a]

[a]*Scientific Computation Research Center, Rensselaer Polytechnic Institute*
Troy, NY USA 12180

Abstract

Accurate RF simulations of fusion systems like ITER require the definition of high-fidelity analysis geometries that include detailed antenna, reactor wall, and physics regions. This paper will describe a workflow for the execution of adaptive high-performance simulations of RF fusion systems. In this workflow, the simulation input consists of a CAD model attributed with the needed analysis attributes. The analysis mesh is automatically generated and the analysis steps are executed using the time-harmonic Maxwell's equations solved using high-order Nédélec finite elements. A patch recovery-based error estimator is used to drive a conforming mesh adaptation procedure.

Keywords: RF Fusion; Maxwell's Equations; Nédélec Finite Elements; Conforming Mesh Adapt; Recovery-based Error Estimation.

1. Introduction

The development and application of advanced simulation codes are critical to the design and operation of the ITER fusion tokamak as well as the planned development of a prototype fusion power plant targeted to go online by 2040 [1]. To provide the level of simulation fidelity required, these codes must address a wide range of complex coupled physics and, in many cases, must fully account for the geometry of the various components within the tokamak systems which are often orders of magnitude smaller than the major dimension of the tokamak. In those cases where full geometric detail of components must be accounted for in the simulation, methods based on unstructured meshes are desirable due to their flexibility to account for the

[*] Corresponding author.
E-mail address: hakimm2@rpi.edu

geometry and the availability of powerful mesh generators that can automatically generate meshes. In addition to being able to account for local geometric details, unstructured mesh methods can vary their approximation resolution over the domain by grading the mesh to concentrate computational resources where they are most needed. A challenge for problems with complex physical behaviors that must be accounted for over general domains is determining where and how to grade the mesh. The goal of adaptive mesh control methods is to use *a posteriori* error measures to determine the mesh element sizes needed throughout the domain as part of the solution process.

As part of a project aimed at developing reliable high fidelity simulation capabilities for the modeling of effectiveness of fusion relevant RF actuators in tokamak fusion and other plasma-containing devices, a software infrastructure consisting of several high-performance computing components is being developed. Section 2 provides a brief overview of the different components of this software infrastructure.

The focus of the current paper is a presentation of the adaptive finite element analysis capability being developed as part of this infrastructure to provide reliable simulation results for the RF problems of interest. Since the error indication procedure used to drive the mesh adaptation is a function of the finite element discretization used, Section 3 provides a summary of the Nédélec finite elements used to discretize the inhomogeneous Maxwell's equations written in the frequency domain being used to model the RF physics. Section 4 presents the error indicator developed and indicates how the error estimates are used in conjunction with a general conforming mesh adaption procedure to produce the adapted meshes. Section 5 demonstrates the application of the developed adaptive procedure to problems of interest. In particular, we will solve the time-harmonic Maxwell's equations in the *cold plasma* approximation regime, in which the spatial dispersions from the electric properties of the plasma are eliminated and therefore one can effectively solve the equations using the finite element method. As will be seen later on, the spatial variations of the permittivity of the plasma will dictate the wavelength of the propagated wave modes. The wavelength is expected to become extremely short when some of the components of the permittivity tensor tend to zero, a phenomenon known as the *lower hybrid resonance*. When this happens, one will need to use a very fine mesh to obtain accurate finite element solutions. Using the adaptive workflow of this paper, we are able to effectively identify such regions in the domain as needed and use mesh modification operations to modify the mesh locally to achieve high levels of accuracy in regions of interest in the analysis domain.

2. Software Infrastructure for the Simulation of Fusion Relevant RF Actuators

As indicated in Figure 1, tokamak fusion reactors are geometrically complex systems made up of a large number of components that are in themselves highly detailed. The execution of an effective simulation workflow that includes a high degree of geometric fidelity and provides the level of solution accuracy required involves:

- Accessing the component geometric descriptions.
- Building the analysis model geometry from the component geometry descriptions.
- Associating the required analysis attributes (material properties, loads, boundary conditions, and initial conditions) with the analysis model geometry.
- Generating an initial mesh of the analysis model geometry and preparing the input information needed by the mesh-based analysis procedure.
- Performing the analysis on the mesh and applying adaptive mesh control to improve the mesh as needed to ensure the requested solution accuracy.

A number of software tools are available that can support aspects of this simulation workflow. Given that, a number of these tools have been integrated with specifically developed new tools to provide a software infrastructure that effectively supports the execution of high fidelity RF simulations of antenna arrays in real tokamak geometries. Figure 2 provides a graphical depiction of the steps that the developed workflow supports.

The first major step is the construction of the analysis geometry, where the analysis geometry is the geometry that will be employed by the mesh generation procedures applied. Given the geometric complexity of the desired analysis domain, the mesh generation methods integrated into this simulation workflow are fully automatic in that the only input provided is the analysis model geometry, with mesh control information applied to the geometric model entities that define the desired mesh sizes, anisotropy, and gradation

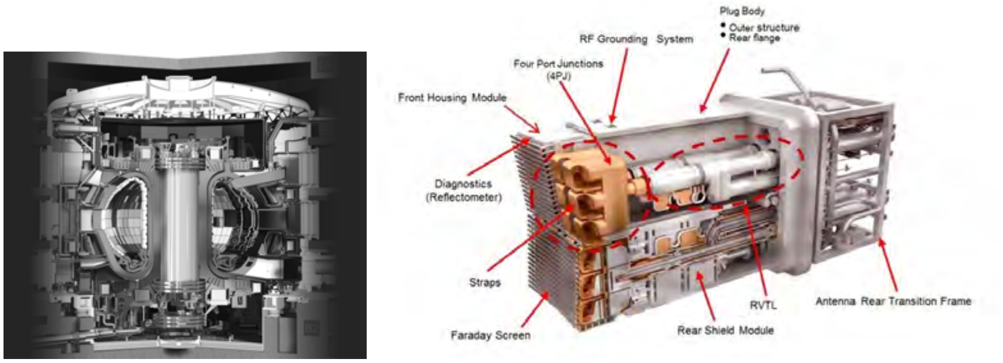


Figure 1: Left: A cutaway view of the ITER tokamak. Right: One of the RF antenna assemblies (copied with permission from the ITER Organization web page https://www.iter.org).

through the analysis model domain. Such automatic mesh generators require that the analysis model geometry be defined in terms of a complete non-manifold boundary representation. Typically a non-manifold representation of the desired analysis geometry is not readily available. What is available are different levels of model representations of the components with the key components being the antenna arrays, the tokamak walls, and magnetic flux surfaces. The component models representations range from:

- CAD models that are fully detailed to the level of including pieces not desired in the analysis models, for example, the bolts used to mount a component to a wall. There are typically CAD models of the antenna arrays.
- Faceted models that provide a raw collection of boundary segments without a complete boundary representation. The most commonly available representations of tokamak walls are faceted models.
- Discrete models that are commonly used to describe physics information that may be used in the definition of analysis model geometry. The key example of this is the selected magnetic flux surfaces as defined in an EFIT file that will be used to define selected boundaries in the analysis geometry model.

Given these varied forms of component geometric definitions, the construction of the analysis model can be viewed as a two-step process in which the

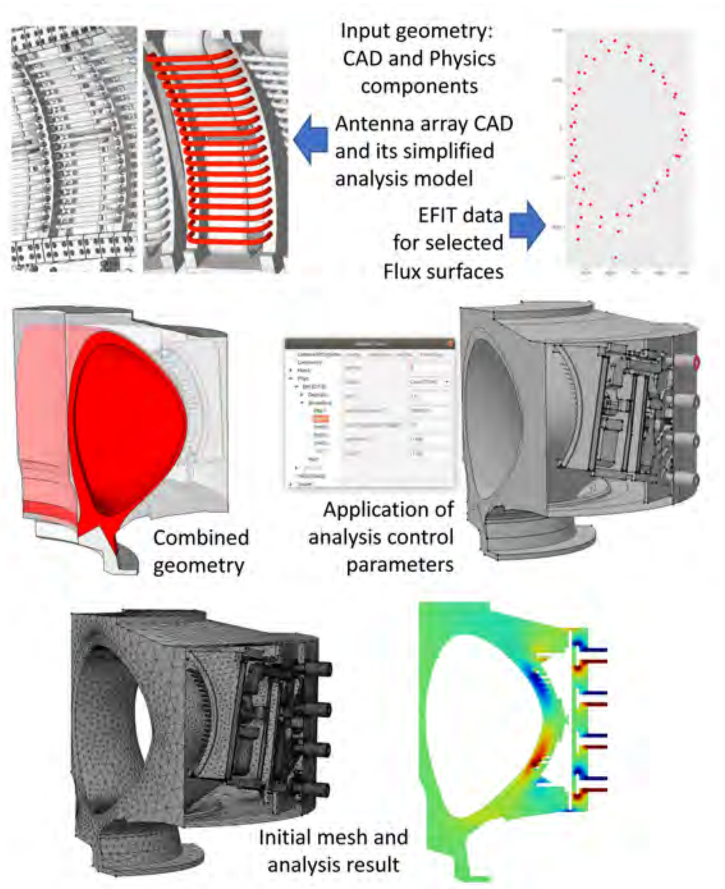


Figure 2: Key steps in the simulation workflow.

component models are first manipulated to represent the specific geometric features to be included in the analysis model, and then the component models are combined to create the required non-manifold geometric model.

Procedures to support the needed geometric modeling operations have been constructed employing two different core geometry tools, Open Cascade [2] and SimModeler [3]. In the case of using Open Cascade, the process of manipulating existing models and combining them is supported by the Petra-M user interface on modeling operations [4, 5]. SimModeler was used in the analysis geometry construction operations in the example shown in figure 2. The upper left image shows a portion of an antenna array as defined in the detailed CAD model and the analysis models created by the application of selected SimModeler geometric simplification operations. The upper right image depicts the discrete data imported into SimModeler representing two flux surfaces that will be used in the construction of the complete analysis model. The tokamak walls were obtained from a file of boundary segments that were combined onto a set of bounding edges based on specific geometric properties and then extruded to define the surfaces and associated wall faces. The left image in the middle row shows the final analysis geometry that includes the tokamak walls, the flux surfaces and the antenna array.

The analysis attributes can be applied in either SimModeler, which has a set of generalized attribute specification capabilities, or in Petra-M which has specific RF relevant attribute specification capabilities. The right image in the middle row of figure 2 shows the use on the Petra-M user interface for attribute specification. Mesh control information is also specified directly on the analysis model entities in SimModeler. From there the mesh can be automatically generated. Both MeshSim [3] and Gmsh [6] can be used to automatically generate meshes that match the specified mesh control information. The mesh in the lower left image was generated using MeshSim.

At this point the information needed for the finite element analysis is complete and the input to the finite element analysis can be provided via input files or using in-memory data transfer methods [7, 8]. The MFEM high-order finite element library [9, 10] was selected to provide the core of the finite element analysis capability used in the developed simulation workflow. The key reasons for the selection of MFEM include:

- Support of high-order finite elements that provide high rates of convergence that can be effectively taken advantage of when solving wave types of equations.

- Emphasis on performant and scalable computational methods to solve very large finite element models on the upcoming exascale computers for which almost all of their raw compute capability will be provided by accelerators with a current emphasis on GPUs.

- Ready support of the full range of weak equation forms allowing the easy development of finite elements that address RF physics and plasma physics models to which the RF simulations may be coupled.

- Ease of extensibility to include new capabilities and to couple with other components that execute additional functions.

The extensibility of MFEM was taken advantage of to tightly integrate it with Petra-M to more effectively address specific RF simulation needs and with the PUMI [11] mesh adaptation procedures used to execute the mesh modifications used to adapt the meshes based on the error indicator procedure presented in section 4.

3. Finite Element Formulation

The propagation of electromagnetic waves can be described by the time-harmonic Maxwell's equations [12]. Given a bounded domain Ω with two disjoint connected boundaries Γ and Σ, these are

$$\begin{aligned} \nabla \times \left(\mu_r^{-1} \nabla \times \mathbf{E}\right) - \kappa^2 \boldsymbol{\epsilon}_r \mathbf{E} = \mathbf{F} \quad & \text{in} \quad \Omega \\ \mathbf{n} \times \mathbf{E} = 0 \quad & \text{on} \quad \Gamma \\ \mu_r^{-1} \left(\nabla \times \mathbf{E}\right) \times \mathbf{n} - \mathbf{i}\kappa\lambda \mathbf{E}_T = \mathbf{g} \quad & \text{on} \quad \Sigma \end{aligned} \tag{1}$$

where $\mathbf{g}$ is a given tangential vector field on Σ, $\mathbf{E}$ is a vector-valued complex function of space (but not time) denoting the electric field, $\mu_r := \mu/\mu_0$ is the relative permeability, $\boldsymbol{\epsilon}_r := \left(\boldsymbol{\epsilon} + \frac{\mathbf{i}\boldsymbol{\sigma}}{\omega}\right)/\epsilon_0$ is the relative anisotropic permittivity tensor, and $\kappa := \omega\sqrt{\epsilon_0 \mu_0}$, is the wavenumber. The forcing term on the right-hand-side of equation (1) is defined in terms of the applied current density:

$$\mathbf{F} := \mathbf{i}\kappa\mu_0^{1/2} \mathbf{J}_a.$$

Finally, the quantity $\mathbf{E}_T$, which denotes the tangent electric field on the boundary Σ is defined via

$$\mathbf{E}_T := (\mathbf{n} \times \mathbf{E}) \times \mathbf{n}.$$

It is noted that since we are interested in studying the propagation of waves for a specific frequency the time-dependent part of the solution (i.e., $e^{-\mathbf{i}\omega t}$) has been factored out in the above formulation.

The relevant boundary conditions for the RF applications considered herein include the perfect electric conductor (PEC) boundary condition and the port boundary conditions (Ports).

Perfect electric conductors (PECs): When the boundary of the cavity is a perfect conductor, Ohm's law implies the electric field must be zero. For such cases, the electric field inside the cavity, and the vicinity of the boundary has to satisfy

$$\mathbf{n} \times \mathbf{E} = 0. \tag{2}$$

This means that the tangential component of the electric field must be zero in the vicinity of perfect conductors.

Port Boundary Condition (Ports): Port boundary conditions are used on the boundary, where the amplitudes and phases (also known as *modes*) of the incoming fields are known. The amplitude and phase of the outgoing *modes* are not known and are solved as part of the FEM solution. Here we use a variation of the the port boundary conditions introduced in [13]. This boils down to specifying the electric (and magnetic) field values at such boundaries. For example, for a Coax port (a hollow cylinder with inner radius a and outer radius b in the z direction) we have

$$\begin{aligned}\mathbf{E}_{\mathrm{coax}} &= A_{\mathrm{n}}\frac{\mathbf{r}}{r}e^{(kz)}\\ \mathbf{i}\omega\mathbf{H}_{\mathrm{coax}} &= A_{\mathrm{n}}\omega\sqrt{\frac{\epsilon_0\epsilon_{\mathrm{r}}}{\mu_{\mathrm{r}}\mu_0}}\frac{\mathbf{r}}{r}e^{(kz)}\end{aligned}$$

where the normalized amplitude

$$A_{\mathrm{n}}^2 = \frac{1}{\pi\ln{(b/a)}}\sqrt{\frac{\mu_{\mathrm{r}}\mu_0}{\epsilon_0\epsilon_{\mathrm{r}}}}$$

is obtained by enforcing that the average incoming wave power is equal to 1.0 Watt.

As will be seen later on in the results section we will solve the above time-harmonic equations in the cold plasma approximation regime which is a limit of zero gyroradius. The cold plasma approximation eliminates the spatial dispersion from the electric properties of the plasma and therefore can be efficiently solved using the finite element methods. In the cold plasma approximation used here, the plasma dielectric tensor is given as

$$\boldsymbol{\epsilon}_r(\omega) = \begin{pmatrix}\mathcal{S}(\omega) & -\mathbf{i}\mathcal{D}(\omega) & 0\\ \mathbf{i}\mathcal{D}(\omega) & \mathcal{S}(\omega) & 0\\ 0 & 0 & \mathcal{P}(\omega)\end{pmatrix}, \tag{3}$$

using the local *Stix* coordinate system in which the z axis is aligned to the local magnetic field. $\mathcal{S}(\omega)$, $\mathcal{D}(\omega)$, and $\mathcal{P}(\omega)$ are defined as

$$\begin{aligned} \mathcal{S}(\omega) &= 1 - \sum_{\text{s}} \frac{\omega_{\text{ps}}^2}{\omega^2 - \omega_{cs}^2} \\ \mathcal{D}(\omega) &= \sum_{\text{s}} \frac{\omega_{\text{ps}}^2 \omega_{cs}}{\omega(\omega^2 - \omega_{\text{cs}}^2)} \\ \mathcal{P}(\omega) &= 1 - \sum_{\text{s}} \frac{\omega_{\text{ps}}^2}{\omega^2}. \end{aligned} \tag{4}$$

Here ω_{ps} and ω_{cs} are the plasma and cyclotron frequencies for species s, respectively. The representation of $\boldsymbol{\epsilon}_r$ in the global coordinate system is given by a rotational transformation of equation (3).

It is worthwhile to note that the spatial variation of the functions $\mathcal{S}$, $\mathcal{D}$, and $\mathcal{P}$ dictates the wavelength of the propagated mode in the plasma. For example, *the lower hybrid resonance* that will be discussed in the results section occurs when $\mathcal{S}$ becomes zero, and thus making the wavelength extremely short.

A weak (integral) form of the above-described problem can be obtained using the Galerkin method [14], by taking the dot product of (1) with a smooth vector function $\boldsymbol{\phi}$ and integrating over the domain [see 12, for the details]. After a selected integration by parts operation, the weak form of the problem can be stated as: Given $\mathbf{F} \in \left(L^2\left(\Omega\right)\right)^3$ and $\mathbf{g} \in L_t^2\left(\Sigma\right)$, find $\mathbf{E} \in \mathcal{X}$, such that

$$\begin{aligned} \int_{\Omega} \left[\left(\mu_r^{-1} \nabla \times \mathbf{E} \right) \cdot \nabla \times \boldsymbol{\phi} - \kappa^2 \left(\boldsymbol{\epsilon}_r \mathbf{E} \right) \cdot \boldsymbol{\phi} \right] \, dV - \mathbf{i}\kappa \int_{\Sigma} \lambda \mathbf{E}_T \cdot \boldsymbol{\phi}_T \, dA \\ = \int_{\Omega} \mathbf{F} \cdot \boldsymbol{\phi} \, dV + \int_{\Sigma} \mathbf{g} \cdot \boldsymbol{\phi}_T \, dA \end{aligned} \tag{5}$$

is satisfied for all $\boldsymbol{\phi} \in \mathcal{X}$. Here the space $\mathcal{X}$ is defined as

$$\mathcal{X} := \Big\{ \mathbf{u} \in H\left(curl; \Omega\right) \ |\mathbf{n} \times \mathbf{u} = 0 \text{ on } \Gamma \ \text{ and } \ \mathbf{u}_T \in \left(L^2\left(\Sigma\right)\right)^3 \text{ on } \Sigma \Big\}. \tag{6}$$

To discretize the above equations we use a polynomial space of order p by means of *curl conforming* Nédélec finite elements [15]. We present the general Nédélec finite elements as defined in [12], and discuss thier implementation in MFEM [9]. Following the notation of [12] we can define the following vector polynomial space for order p

$$\mathcal{R}_p = \left(\mathcal{P}_{p-1}\right)^d \oplus \mathcal{S}_p, \quad \text{where} \quad \mathcal{S}_p = \left\{\mathbf{u} \in \left(\tilde{\mathcal{P}}_p\right)^d \mid \mathbf{u}.\mathbf{x} = 0\right\} \tag{7}$$

Here d denotes the dimension of the vector space, $\mathcal{P}_p$ is the set of polynomials of maximum degree p, and $\tilde{\mathcal{P}}_p$ is the set of *homogeneous* polynomials of degree p. The edge finite element spaces (e.g., the Nédélec elements) depend on the use of the vector polynomial space $\mathcal{R}_p$ defined in (7). Next, the *curl conforming* finite element for a tetrahedron element can be defined as follows (such finite elements for triangles can be defined similarly): Given the reference tetrahedron $\Omega_{\hat{r}}$, and the polynomial space $\mathcal{P}_{\Omega_{\hat{r}}} = \mathcal{R}_p$, the degrees of freedom (DOF) of a Nédélec tetrahedron, are defined by

$$M_{\hat{e}}\left(\hat{\mathbf{u}}\right) := \int_{\hat{e}} \hat{\mathbf{u}} \cdot \hat{\mathbf{t}} \hat{q} \, dS, \quad \text{for all} \quad \hat{q} \in \mathcal{P}_{p-1}\left(\hat{e}\right), \tag{8}$$

$$M_{\hat{f}}\left(\hat{\mathbf{u}}\right) := \frac{1}{\text{area}\left(\hat{f}\right)} \int_{\hat{f}} \hat{\mathbf{u}} \cdot \hat{\mathbf{q}} \, dA, \quad \text{for all} \quad \hat{\mathbf{q}} \in \mathcal{P}^3_{p-2}\left(\hat{f}\right) \quad \text{and} \quad \hat{\mathbf{q}} \cdot \hat{\mathbf{n}} = 0, \tag{9}$$

and

$$M_{\hat{r}}\left(\hat{\mathbf{u}}\right) := \int_{\hat{e}} \hat{\mathbf{u}} \cdot \hat{\mathbf{q}} \, dV, \quad \text{for all} \quad \hat{\mathbf{q}} \in \mathcal{P}^3_{p-2}\left(\hat{r}\right), \tag{10}$$

for edges, faces, and the regions, respectively. Note that the $\hat{.}$ is used to denote quantities in the reference (or parent) tetrahedron. Similar expressions can be written for the DOFs in the physical space of a given tetrahedron Ω_r by using the affine transformation between the reference tetrahedron $\Omega_{\hat{r}}$ and Ω_r. Two important properties of the selected finite element spaces are:

(a) *Curl conforming* [see Lemma 5.35, in 12, for proof]. One can show that if $\mathbf{u}$ is represented in the above form, then the tangential trace, as given by $\mathbf{u} \times \mathbf{n}$, is continuous across the face of two neighboring tetrahedra, where $\mathbf{n}$ denotes the normal to the face.

(b) *Unisolvant* [see Lemma 5.36, in 12, for proof]. If all the DOFs, as defined by equations (8)-(10), are zero then $\mathbf{u}$ is identically zero

The global Nédélec space over a tetrahedral mesh satisfies

$$\mathcal{X}_h := \left\{\mathbf{u} \in H\left(curl; \Omega\right) \;\mid\; \mathbf{u}|_{\Omega_r} \in \mathcal{R}_p \quad \text{for all} \quad \Omega_r \quad \text{in the mesh}\right\}. \tag{11}$$

The space $\mathcal{X}_h$ can be replaced with the space $\mathcal{X}$ of the weak formulation discussed earlier to form the descretized finite element space of the solution.

The Nédélec (vector) shape functions of order p in MFEM [9, 16] are constructed hierarchically using 1D Chebyshev polynomials of order p (see pseudo-code 1). Assuming the necessary smoothness conditions for the electric field $\mathbf{E}$, and for a given node j the element DOFs are then defined as

$$M^j(\mathbf{E}) := \hat{\mathbf{t}}_j \cdot \left(J^T(\xi_j)\, \mathbf{E}(\xi_j)\right) \tag{12}$$

where $\mathbf{t}_j$ is the tangent vector corresponding to node j, ξ_j is the location of node j in the parent (reference) tetrahedron, and $J(\xi)$ is the Jacobian of the affine transformation between the parent and physical tetrahedron. For an example, consider Figure 3 which depicts the node locations and the tangent vectors for a cubic tetrahedron. Note that for each face node location there would be two tangents, and therefore two DOFs are associated with each node location. Similarly, there would be three DOFs at each internal node location of a tetrahedron. Having defined the vector shape functions $\mathbf{N}^j(\xi)$ and the DOFs, the vector functions over the Nédélec tetrahedron Ω_r are

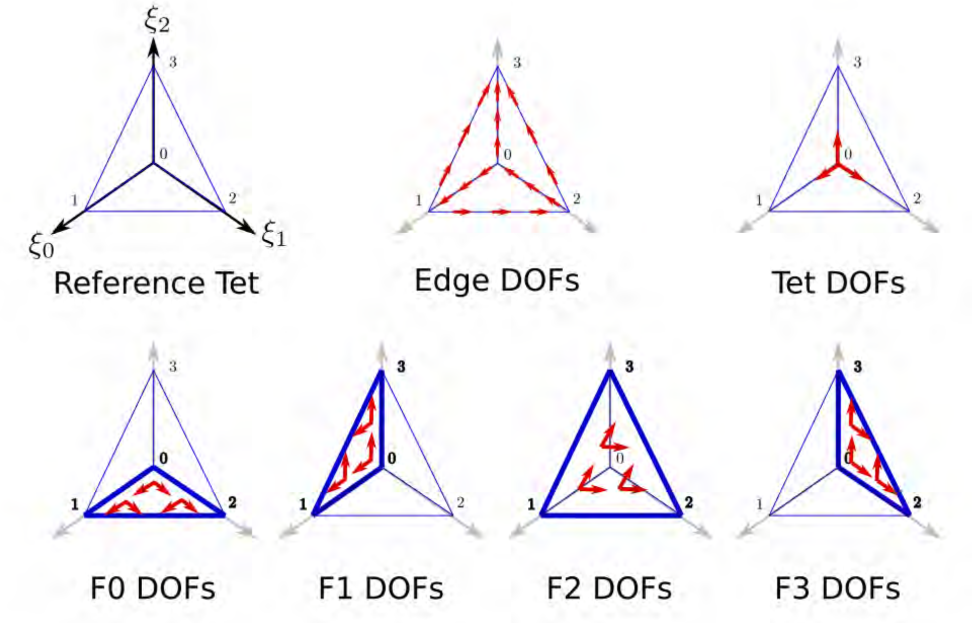


Figure 3: Schematic of a cubic Nédélec tetrahedron.

Pseudo-code 1 Implementation of Nédélec shape functions in MFEM. The vector shape functions are the rows of the returned matrix, $shape_{(\#dof \times 3)}$. The inverse transformation Ti, is used to make sure the DOFs have a simple form given in equation (12). See pseudo-code (2) in Appendix A for the details of computing Ti.

```
function GetNedelecShapes($p$, $\xi$)
    shape_x ← Chebyshev($\xi_0$, $p-1$)
    shape_y ← Chebyshev($\xi_1$, $p-1$)
    shape_z ← Chebyshev($\xi_2$, $p-1$)
    shape_w ← Chebyshev($1-\xi_0-\xi_1-\xi_2$, $p-1$)
    $o \leftarrow 0$
    $c \leftarrow 1/4$
    for $k = 0$ to $p-1$ do
        for $j = 0$ to $p-1-k$ do
            for $i = 0$ to $p-1-k-j$ do
                $s \leftarrow$ shape_x[$i$].shape_y[$j$].shape_z[$k$].shape_w[$p{-}1{-}i{-}j{-}k$]
                $u[o][:] \leftarrow [s, 0, 0]$, $o \leftarrow o+1$
                $u[o][:] \leftarrow [0, s, 0]$, $o \leftarrow o+1$
                $u[o][:] \leftarrow [0, 0, s]$, $o \leftarrow o+1$
            end for
        end for
    end for
    for $k = 0$ to $p-1$ do
        for $j = 0$ to $p-1-k$ do
            $s \leftarrow$ shape_x[$p{-}1{-}j{-}k$].shape_y[$j$].shape_z[$k$]
            $u[o][:] \leftarrow [s(\xi_1-c), -s(\xi_0-c), 0]$, $o \leftarrow o+1$
            $u[o][:] \leftarrow [s(\xi_2-c), 0, -s(\xi_0-c)]$, $o \leftarrow o+1$
        end for
    end for
    for $k = 0$ to $p-1$ do
        $s \leftarrow$ shape_y[$p{-}1{-}k$].shape_z[$k$]
        $u[o][:] \leftarrow [0, s(\xi_2-c), -s(\xi_1-c)]$, $o \leftarrow o+1$
    end for
    $Ti \leftarrow$ GetInverseTransformation($p$)
    $shapes \leftarrow$ MatrixMatrixMultiply($Ti$, $u$)
    return $shapes$
end function
```

interpolated by means of the following sum

$$\mathbf{E}|_{\Omega_r}(\xi) = \sum_j M^j(\mathbf{E})\,\mathbf{N}^j(\xi) \tag{13}$$

4. Error Indication and Adaptive Mesh Control

Mesh discretization errors are a result of using finite-dimensional solution spaces. For fixed order elements, the discretization errors can be controlled by altering the mesh sizes using mesh adaptation, a process know as h-adaptivity. In this work, we make use of the recovery-based *a posteriori* procedure developed in [17], specialized for the Maxwell's equations of this study.

The elemental errors (in the L_2 norm sense) for the electric field can be defined as

$$\|e\|_{\Omega_e} := \left[\int_{\Omega_e} (\mathbf{E} - \mathbf{E}_h)^2\, d\Omega\right]^{1/2} \tag{14}$$

where $\mathbf{E}$ and $\mathbf{E}_h$, are the exact and finite element approximation for the electric field, respectively. Since the exact solution is unknown, one can obtain an estimate of the error by replacing the exact solution with an improved approximation of the solution ($\mathbf{E}^*$). Methods for obtaining the improved approximation, include employing an enriched basis [18, 19, 20], and recovery-based methods [21, 17]. In this paper, we use a recovery-based method known as the patch recovery scheme originally proposed in [17].

The main idea behind the patch recovery scheme is to obtain a C^0 field for an appropriate solution field for which the finite element solution is a C^{-1} discontinuous field. This is done by using a patch-wise least squares fit to the finite element field to obtain the nodal values for the C^0 recovered field.

To ensure a reliable error estimate, the nodal values of the recovered field need to satisfy the "super-convergence" criteria [see, for example, reference 22, for the definition of super-convergence]. In short, this means showing that the recovered field, when compared to the finite element solution, converges with order $O\left(h^{1+\delta}\right)$ (for some $\delta > 0$). In other words, the recovered field convergences to the exact solution at a faster rate than the finite element solution itself. This justifies the use of the recovered field $\mathbf{E}^*$ in expression (14) to obtain an estimate for the error.

The proof of "super-convergence" for a given recovery procedure is rather involved and is skipped here for brevity. Interested readers are encouraged to see [23] where it is shown that super-convergence can be achieved for linear "edge" elements of Nédélec type on tetrahedron meshes with an averaging recovery operator (wherein the recovered field at each node is defined by averaging the interpolated field at the center of the adjacent tetrahedra). A more recent study [24], provides the super-convergence analysis for arbitrary order edge elements for Maxwell's equations.

One way to extend the averaging recovery method mentioned in [23] to higher orders is to replace the recovery operator based on the averages with a least-square fit of field values over carefully chosen sample points in a patch. For the Maxwell problem of this paper, the electric field is represented in Nédélec basis for which the normal components are known to be discontinuous at element boundaries. Thus, given the finite element solution $\mathbf{E}_h$, we sample this field at integration points in the patch of elements that the node of interest bounds and fit a continuous polynomial of a given order to obtain the recovered value of $\mathbf{E}^*$ at that node. Specific care does need to be exercised in the process to ensure that number of integration points available for fitting is sufficient to support a least-squares fit for the order of polynomial selected to be fit. If there is an insufficient number of integration points in the patch, the patch can be expanded to include additional neighboring elements or the polynomial order of least-square fit can be reduced. Using this patch recovery method we obtain the following approximation for the element error

$$\|e\|_{\Omega_e} \approx \|e^*\|_{\Omega_e} := \left[\int_{\Omega_e} (\mathbf{E}^* - \mathbf{E}_h)^2 \, d\Omega\right]^{1/2} \tag{15}$$

The total error over the domain as well as the percentage error can then be defined as follows

$$\|e\|_{L_2} \approx \left(\sum_{e=1}^{n} \|e^*\|^2_{\Omega_e}\right)^{1/2} \quad \text{and} \quad \eta := \frac{\|e\|_{L_2}}{\|\mathbf{E}^*\|_{L_2}} \tag{16}$$

where n is the total number of elements in the mesh.

A general requirement is then to have the percentage error below a certain threshold, i.e.,

$$\eta \leq \eta_{max}. \tag{17}$$

Furthermore, the error contribution due to individual elements in the mesh are expected to be approximately equal in an optimal mesh. Using these criteria we can then define the desired mesh size for each element in the mesh as follows

$$h_{desired} = h_{current} \times r_e, \tag{18}$$

where $h_{current}$ is the current size of the element and r_e is the element size factor defined as follows [25]

$$r_e = \|e^*\|_{\Omega_e}^{-\frac{2}{2p+d}} \left(\frac{\eta_{max}^2 \|\mathbf{E}^*\|_{L_2}^2}{\sum_{e'=1}^{n} \|e^*\|_{\Omega_{e'}}^{\frac{2d}{2p+d}}} \right)^{\frac{1}{2p}} \tag{19}$$

where d is the dimension of the mesh and p is the polynomial order of the finite element basis.

Once the new mesh size field has been calculated the mesh needs to be modified such that it matches that mesh size field. MFEM supports two mesh adaption approaches. The first is based on a non-conforming element subdivision [9]. This approach is most commonly applied and is most effective when hexahedral and quadrilateral elements are used. Since fully automatic hexahedral mesh generators that can create graded meshes on general 3D domains do not exist, the non-conforming MFEM adaptation procedures are not used in the current simulation workflow.

Instead, the current simulation workflow employs the second mesh adaptation approach which is a conforming mesh adaptation procedure based on local mesh modification. This approach is well suited to the simplex element (tetrahedra and triangles) meshes generated by the automatic mesh generators integrated into the simulation workflow.

The conforming mesh adaptation procedure used in this work is based on the PUMI [11, 26] mesh adaptation procedures [27] that have been, and continue to be, generalized to work with curved meshes [28].

5. Results

In this section, we provide results demonstrating the application of the above-described adaptive workflow to the simulation of *cold plasma* in RF heaters. As mentioned earlier, when some of the components of the permittivity tensor in the plasma become small the wavelength of the propagated waves become very short. Using a uniformly fine mesh in the whole domain to capture the solution accurately can be very costly in the simulation of realistic problems. Here we demonstrate the use of the adaptive workflow of this paper to effectively identify such regions in the analysis domain and make the mesh finer in those regions as necessary and thus reducing the computational cost, significantly.

5.1. Lower Hybrid Waves in Alcator C-Mod Plasma

In this subsection, we will apply the adaptive workflow to the full-wave simulation of *Lower Hybrid Waves* in Alcator C-Mod plasma as described in [29, 30, 31] and references therein. Figure 4 shows the problem domain. We will solve axis-symmetric Maxwell's equations with an out-of-plane wave number of 200 in the vacuum and plasma regions, and 0 in the waveguides. The in-plane components of the electric field (E_r and E_z) are discretized using high order Nédélec finite element basis, while the out-of-plane component (E_ϕ) is discretized using a high-order nodal finite element basis. The order of the finite element spaces used for the results that follow is $p = 2$.

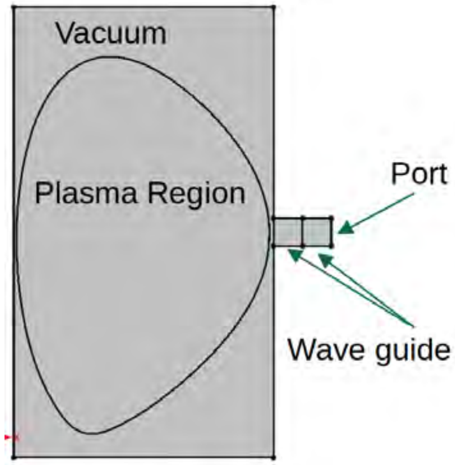
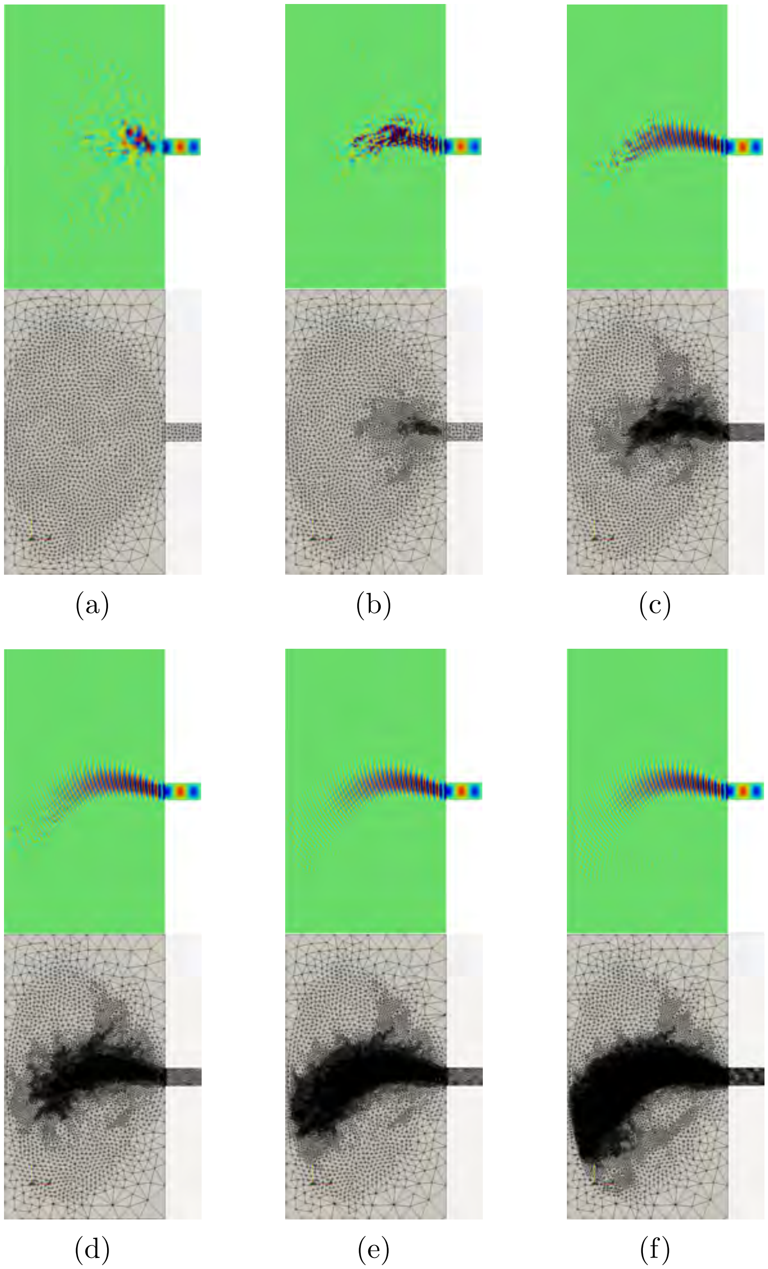


Figure 4: This figure shows the setup for the axis-symmetric full-wave simulation of lower hybrid waves in Alcator C-Mod plasma.

Figure 5 show the meshes and solutions for the starting mesh, as well as, five successive adapted meshes. As can be seen in these figures, the solution on the starting coarse mesh is not resolved at all. As the mesh is adapted (using the above-described error indication procedures), we see substantially improved resolutions in the results in areas of the domain where the solution features are of most importance. Figure 6 shows the zoomed version of the initial and final meshes/solutions.



(a) (b) (c)

(d) (e) (f)

Figure 5: Shows the initial (a) and adapted (b-f) meshes and solutions field, E_r.

To demonstrate the advantage of using the adaptive workflow, table 1 shows the comparison for the number of DOFs (for $p = 2$ this is simply

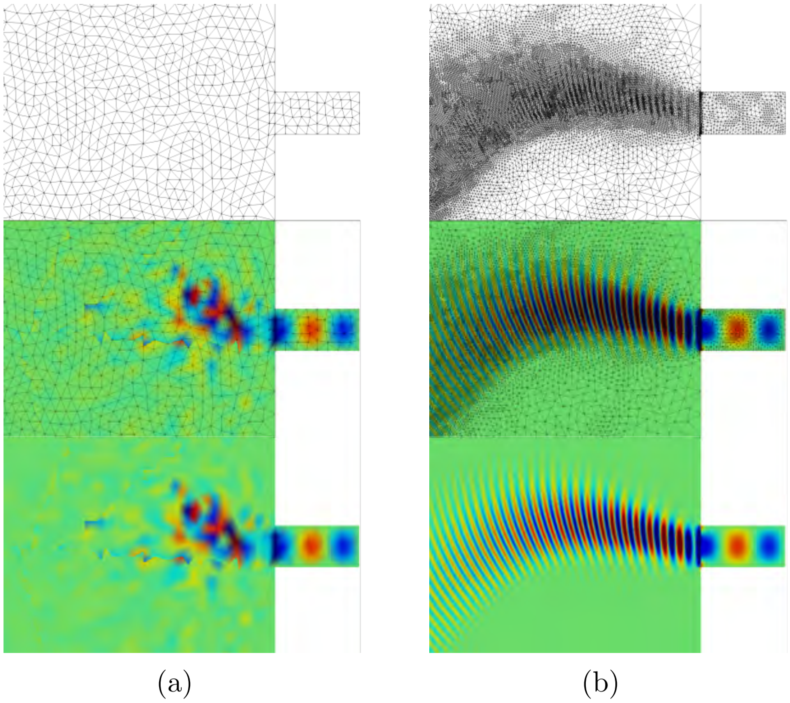


Figure 6: Shows the zoomed version of the initial (a) and final adapted (b) meshes and solutions field, E_r.

$2\,(e + f)$ where e and f are the total number of edges and faces in the mesh, respectively) for the adapted meshes shown in figure 5 and uniform meshes with the same minimum size in the plasma regions. As can be seen in this

mesh	#dofs	min size	#dofs*
(a)	17,316	8.00E-03	17,316
(b)	25,434	2.80E-03	126,238
(c)	67,688	1.29E-03	414,436
(d)	113,284	8.36E-04	463,344
(e)	176,284	7.56E-04	1,070,378
(f)	284,250	5.85E-04	2,298,062

Table 1: Comparison of the DOFs in the adapted meshes and uniform meshes with the same minimum size in the plasma regions. The first column lists the number of DOFs in the adapted meshes. The second column lists the minimum edge length in the plasma region of the adapted meshes. The last column lists the number of DOFs for uniform meshes with the same size as the minimum size in the corresponding adapted mesh.

table, using a uniform mesh to achieve the same level of accuracy for the solutions shown in Figure 5-f will require $\sim$ 2.3M DOFs while the number of DOFs in the corresponding adapted mesh is $\sim$ 284K. In other words, by using the adaptive workflow we are achieving a 10x reduction in the size of the problem.

5.2. ICRF antenna of the Large Plasma Device

Next, we consider a fully 3D simulation of the ICRF antenna model of the Large Plasma Device (LAPD). This example consists of a single strap antenna, as can be seen in Figure 7. The plasma region has an anisotropic permittivity that varies radially. The input current is applied as a port boundary condition at the *coax* port shown in the picture. Figure 8 shows the initial and adapted meshes and the solution (phase) fields. As can be seen in this figure the solution field is not properly resolved on the initial coarse mesh. However, as the mesh gets more refined in the slow-wave region of the plasma domain, more features of the solution are resolved.

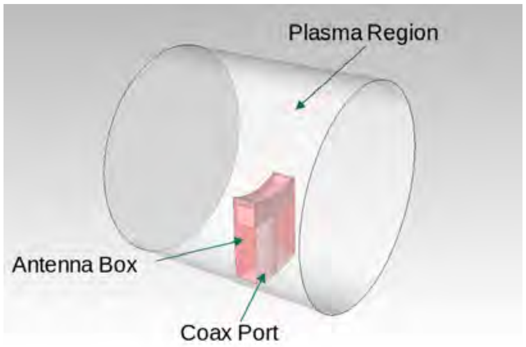


Figure 7: This figure shows the setup for the 3D LAPD example.

Table 2 shows the comparison for the number of DOFs for the adapted meshes shown in Figure 8 and uniform meshes with the same minimum size in the plasma regions. Similar to the 2D results, we see that by using the adaptive workflow we are able to achieve the desired levels of accuracy with much fewer number of DOFs, when compared to uniform meshes. For example, the number of DOFs in the final adapted mesh in Figure 8 is approximately 5x smaller than the corresponding uniform mesh.

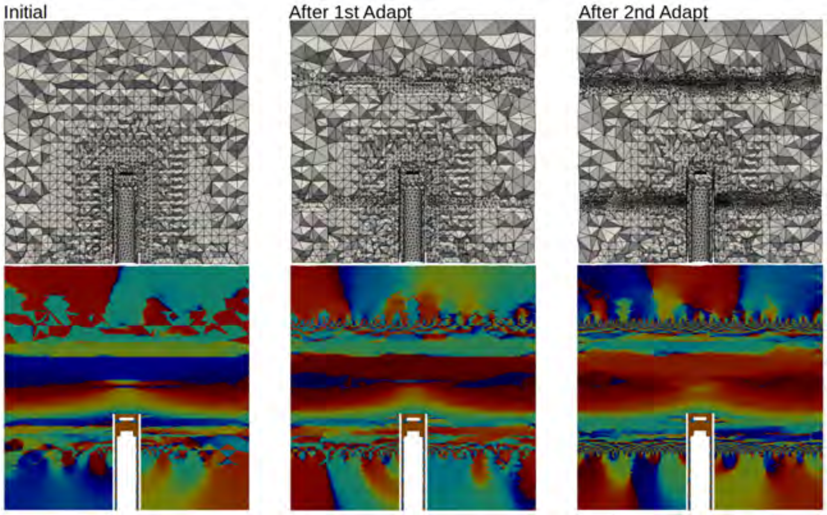


Figure 8: This figure shows the initial and adapted meshes and solution fields for the LAPD example.

mesh	#dofs	min size	#dofs*
(a)	1,088,678	2.67E-03	1,088,678
(b)	2,129,006	2.28E-03	7,860,478
(c)	12,626,818	1.11E-03	61,959,826

Table 2: Comparison of the DOFs in the adapted meshes and uniform meshes with the same minimum size in the plasma regions. The first column lists the number of DOFs in the adapted meshes. The second column lists the minimum edge length in the plasma region of the adapted meshes. The last column lists the number of DOFs for uniform meshes with the same size as the minimum size in the corresponding adapted mesh.

6. Conclusions

This paper presented an adaptive simulation workflow for RF heaters in fusion applications. This workflow takes advantage of, and carefully integrates, a number of software components including CAD modelers, automatic mesh generators, custom problem set-up tools, high-order finite element solvers, discretization error estimators, and conforming mesh adaptation procedures. The example results demonstrate the ability of the developed simulation workflow to effectively obtain highly accurate simulation results for general antenna and reactor geometries. It is important to note that an effective simulation workflow must address all aspects of the simulation process; from the specification of the simulation, to the generation of the analysis model discretization, to the solution of the discretized system, to the improvement of the discretization to attain the desired level of solution accuracy.

Even though the results shown in this paper are for order 2 Nédélec spaces, the same workflow can be utilized for higher-order finite element spaces. Efforts to support the higher-order finite element spaces require the ability of the mesh adaptation procedures to execute on higher than quadratic mesh geometry. The mesh adaptation procedures are currently being extended to deal with such higher-order mesh geometry. These developments and their application will be reported in future publications.

7. Acknowledgments

This research is supported by the Exascale Computing Project (17-SC-20-SC), a collaborative effort of U.S. Department of Energy Office of Science and the National Nuclear Security Administration, under contract DE-AC02-06CH11357, and by the U.S. Department of Energy, Office of Science Scientific Discovery through Advanced Computing program through grant DE-SC0018275 (Unstructured Mesh Technologies for Fusion Simulation Codes).

Computational resources were provided by the Rensselaer Polytechnic Institute Center for Computational Innovations that is partly supported by a National Science Foundation through MRI grant number 1828083.

We also thank Dr. Syun'ichi Shiraiwa at Princeton Plasma Physics Laboratory for his help with the integration of PUMI adaptive tools and Petra-M and also providing us with application examples to test the workflow.

Appendix A. Inverse Transformation Matrix

The pseudo-code for the computation of the inverse transformation, Ti, mentioned in pseudo-code (1) is given below. Pseudo-codes (3)-(5), detail the computation of edge, face, and region nodes/tangents for tetrahedron elements.

Pseudo-code 2 Implementation of the inverse transformation Ti

```
function GetInverseTransformation($p$)
    $enodes, etans \leftarrow$ GetEdgeNodesAndTangents($p$)
    $fnodes, ftans \leftarrow$ GetFaceNodesAndTangents($p$)
    $rnodes, rtans \leftarrow$ GetRegionNodesAndTangents($p$)
    $nodes \leftarrow$ Concatenate($enodes$, $fnodes$, $rnodes$)            ▷ all nodes
    $tans \leftarrow$ Concatenate($etans$, $ftans$, $rtans$)            ▷ all tangents
    for $m = 0$ to $non - 1$ do            ▷ $non$ is total number of nodes
        $x \leftarrow nodes[m][0]$, $y \leftarrow nodes[m][1]$, $z \leftarrow nodes[m][2]$, $t \leftarrow tans[m]$
        $\text{shape}_x \leftarrow$ Chebyshev($x$), $\text{shape}_y \leftarrow$ Chebyshev($y$)
        $\text{shape}_z \leftarrow$ Chebyshev($z$)
        $\text{shape}_w \leftarrow$ Chebyshev($1 - x - y - z$)
        $o \leftarrow 0$
        for $k = 0$ to $p - 1$ do
            for $j = 0$ to $p - 1 - k$ do
                for $i = 0$ to $p - 1 - k - j$ do
                    $s \leftarrow \text{shape}_x[i].\text{shape}_y[j].\text{shape}_z[k].\text{shape}_w[p-1-i-j-k]$
                    $T[o][m] \leftarrow s.t[0]$, $o \leftarrow o + 1$
                    $T[o][m] \leftarrow s.t[1]$, $o \leftarrow o + 1$
                    $T[o][m] \leftarrow s.t[2]$, $o \leftarrow o + 1$
                end for
            end for
        end for
        for $k = 0$ to $p - 1$ do
            for $j = 0$ to $p - 1 - k$ do
                $s \leftarrow \text{shape}_x[p-1-j-k].\text{shape}_y[j].\text{shape}_z[k]$
                $T[o][m] \leftarrow s\{(y - c).t[0] - (x - c).t[1]\}$, $o \leftarrow o + 1$
                $T[o][m] \leftarrow s\{(z - c).t[0] - (x - c).t[2]\}$, $o \leftarrow o + 1$
            end for
        end for
        for $k = 0$ to $p - 1$ do
            $s \leftarrow \text{shape}_y[p-2-k].\text{shape}_z[k]$
            $T[o][m] \leftarrow s\{(z - c).t[1] - (y - c).t[2]\}$, $o \leftarrow o + 1$
        end for
    end for
    return MatrixInverse($T$)
end function
```

Pseudo-code 3 Implementation of edge nodes and tangents.

```
function GetEdgeNodesAndTangents($p$)
    $pm_1 \leftarrow p - 1$
    ep $\leftarrow$ GaussLobatto($pm_1$)                    ▷ to construct edge nodes
    $o \leftarrow 0$
    for $i = 0$ to $p - 1$ do                                     ▷ Edge (0,1)
        $nodes[o][:] \leftarrow [\text{ep}[i], 0, 0]$
        $tans[o][:] \leftarrow [1, 0, 0]$, $o \leftarrow o + 1$
    end for
    for $i = 0$ to $p - 1$ do                                     ▷ Edge (1,2)
        $nodes[o][:] \leftarrow [\text{ep}[pm_1 - i], \text{ep}[i], 0]$
        $tans[o][:] \leftarrow [-1, 1, 0]$, $o \leftarrow o + 1$
    end for
    for $i = 0$ to $p - 1$ do                                     ▷ Edge (2,0)
        $nodes[o][:] \leftarrow [0, \text{ep}[pm_1 - i], 0]$
        $tans[o][:] \leftarrow [0, -1, 0]$, $o \leftarrow o + 1$
    end for
    for $i = 0$ to $p - 1$ do                                     ▷ Edge (0,3)
        $nodes[o][:] \leftarrow [0, 0, \text{ep}[i]]$
        $tans[o][:] \leftarrow [0, 0, 1]$, $o \leftarrow o + 1$
    end for
    for $i = 0$ to $p - 1$ do                                     ▷ Edge (1,3)
        $nodes[o][:] \leftarrow [\text{ep}[pm_1 - i], 0, \text{ep}[i]]$
        $tans[o][:] \leftarrow [-1, 0, 1]$, $o \leftarrow o + 1$
    end for
    for $i = 0$ to $p - 1$ do                                     ▷ Edge (2,3)
        $nodes[o][:] \leftarrow [0, \text{ep}[pm_1 - i], \text{ep}[i]]$
        $tans[o][:] \leftarrow [0, -1, 1]$, $o \leftarrow o + 1$
    end for
    return $nodes$, $tans$
end function
```

Pseudo-code 4 Implementation of face nodes and tangents.

```
function GetFaceNodesAndTangents($p$)
    $pm_2 \leftarrow p - 2$
    $\mathrm{fp} \leftarrow$ GaussLobatto($pm_2$)                ▷ to construct face nodes
    $o \leftarrow 0$
    for $j = 0$ to $p - 2$ do                                  ▷ Face (0,1,2)
        for $i = 0$ to $p - 2 - j$ do
            $w \leftarrow \mathrm{fp}[i] + \mathrm{fp}[j] + \mathrm{fp}[pm_2 - i - j]$
            $nodes[o][:] \leftarrow [\mathrm{fp}[i], \mathrm{fp}[j], 0]/w$, $tans[o][:] \leftarrow [1, 0, 0]$, $o \leftarrow o + 1$
            $nodes[o][:] \leftarrow [\mathrm{fp}[i], \mathrm{fp}[j], 0]/w$, $tans[o][:] \leftarrow [0, 1, 0]$, $o \leftarrow o + 1$
        end for
    end for
    for $j = 0$ to $p - 2$ do                                  ▷ Face (0,1,3)
        for $i = 0$ to $p - 2 - j$ do
            $w \leftarrow \mathrm{fp}[i] + \mathrm{fp}[j] + \mathrm{fp}[pm_2 - i - j]$
            $nodes[o][:] \leftarrow [\mathrm{fp}[i], 0, \mathrm{fp}[j]]/w$, $tans[o][:] \leftarrow [1, 0, 0]$, $o \leftarrow o + 1$
            $nodes[o][:] \leftarrow [\mathrm{fp}[i], 0, \mathrm{fp}[j]]/w$, $tans[o][:] \leftarrow [0, 0, 1]$, $o \leftarrow o + 1$
        end for
    end for
    for $j = 0$ to $p - 2$ do                                  ▷ Face (1,2,3)
        for $i = 0$ to $p - 2 - j$ do
            $w \leftarrow \mathrm{fp}[i] + \mathrm{fp}[j] + \mathrm{fp}[pm_2 - i - j]$
            $nodes[o][:] \leftarrow [\mathrm{fp}[pm_2 - i - j], \mathrm{fp}[i], \mathrm{fp}[j]]/w$
            $tans[o][:] \leftarrow [-1, 1, 0]$, $o \leftarrow o + 1$
            $nodes[o][:] \leftarrow [\mathrm{fp}[pm_2 - i - j], \mathrm{fp}[i], \mathrm{fp}[j]]/w$
            $tans[o][:] \leftarrow [-1, 0, 1]$, $o \leftarrow o + 1$
        end for
    end for
    for $j = 0$ to $p - 2$ do                                  ▷ Face (0,2,3)
        for $i = 0$ to $p - 2 - j$ do
            $w \leftarrow \mathrm{fp}[i] + \mathrm{fp}[j] + \mathrm{fp}[pm_2 - i - j]$
            $nodes[o][:] \leftarrow [0, \mathrm{fp}[i], \mathrm{fp}[j]]/w$, $tans[o][:] \leftarrow [0, 1, 0]$, $o \leftarrow o + 1$
            $nodes[o][:] \leftarrow [0, \mathrm{fp}[i], \mathrm{fp}[j]]/w$, $tans[o][:] \leftarrow [0, 0, 1]$, $o \leftarrow o + 1$
        end for
    end for
    return $nodes$, $tans$
end function
```

Pseudo-code 5 Implementation of region nodes and tangents.

```
function GetRegionNodesAndTangents($p$)
    $pm_3 \leftarrow p - 3$
    rp $\leftarrow$ GaussLobatto($pm_3$)                ▷ to construct region nodes
    $o \leftarrow 0$
    for $k = 0$ to $p - 3$ do
        for $j = 0$ to $p - 3 - k$ do
            for $i = 0$ to $p - 3 - k - j$ do
                $w \leftarrow \mathrm{rp}[i] + \mathrm{rp}[j] + \mathrm{rp}[k] + \mathrm{rp}[pm_3 - i - j - k]$
                $nodes[o][:] \leftarrow [\mathrm{rp}[i], \mathrm{rp}[j], \mathrm{rp}[k]]/w$
                $tans[o][:] \leftarrow [1, 0, 0]$, $o \leftarrow o + 1$
                $nodes[o][:] \leftarrow [\mathrm{rp}[i], \mathrm{rp}[j], \mathrm{rp}[k]]/w$
                $tans[o][:] \leftarrow [0, 1, 0]$, $o \leftarrow o + 1$
                $nodes[o][:] \leftarrow [\mathrm{rp}[i], \mathrm{rp}[j], \mathrm{rp}[k]]/w$
                $tans[o][:] \leftarrow [0, 0, 1]$, $o \leftarrow o + 1$
            end for
        end for
    end for
    return $nodes$, $tans$
end function
```

References

[1] Committee on the Key Goals and Innovation Needed for a U.S. Fusion Pilot Plant, Bringing fusion to the U.S. grid (2021), Tech. rep., The National Academies of Sciences, Engineering and Medicine (2021). doi:https://doi.org/10.17226/25991.

[2] Open Cascade community Edition, Open cascade web page (2020).
URL `https://github.com/tpaviot/oce`

[3] Simmetrix, Simmetrix: Enabling simulation-based design.
URL `http://www.simmetrix.com/`

[4] S. Shiraiwa, J. Wright, P. Bonoli, T. Kolev, M. Stowell, Rf wave simulation for cold edge plasmas using the mfem library, in: EPJ Web of Conferences, Vol. 157, EDP Sciences, 2017, p. 03048.

[5] S. Shiraiwa, T. Fredian, J. Hillairet, J. Stillerman, πscope: Python based scientific workbench with mdsplus data visualization tool, Fusion Engineering and Design 112 (2016) 835–838.

[6] Gmsh Team, Gmsh web page.
URL `https://gmsh.info/`

[7] W. F. Godoy, N. Podhorszki, R. Wang, C. Atkins, G. Eisenhauer, J. Gu, P. Davis, J. Choi, K. Germaschewski, K. Huck, ..., S. Klasky, Adios 2: The adaptable input output system. a framework for high-performance data management, SoftwareX 12 (2020) 100561.

[8] C. W. Smith, B. Granzow, G. Diamond, D. Ibanez, O. Sahni, K. E. Jansen, M. S. Shephard, In-memory integration of existing software components for parallel adaptive unstructured mesh workflows, Concurrency and Computation: Practice and Experience 30 (18) (2018) e4510.

[9] R. Anderson, J. Andrej, A. Barker, J. Bramwell, J.-S. Camier, J. Cerveny, V. Dobrev, Y. Dudouit, A. Fisher, T. Kolev, et al., Mfem: a modular finite element methods library, Computers & Mathematics with Applications 81 (2021) 42–74.

[10] MFEM Team, Mfem web page.
URL `https://mfem.org/`

[11] D. A. Ibanez, E. S. Seol, C. W. Smith, M. S. Shephard, Pumi: Parallel unstructured mesh infrastructure, ACM Transactions on Mathematical Software (TOMS) 42 (3) (2016) 1–28.

[12] P. Monk, et al., Finite element methods for Maxwell's equations, Oxford University Press, 2003.

[13] J.-M. Jin, The finite element method in electromagnetics, John Wiley & Sons, 2015.

[14] P. G. Ciarlet, The finite element method for elliptic problems, SIAM, 2002.

[15] J.-C. Nédélec, Mixed finite elements in $\mathbb{R}^3$, Numerische Mathematik 35 (3) (1980) 315–341.

[16] A. V. Grayver, T. V. Kolev, Large-scale 3d geoelectromagnetic modeling using parallel adaptive high-order finite element methodem modeling with high-order fem, Geophysics 80 (6) (2015) E277–E291.

[17] O. C. Zienkiewicz, J. Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. part 1: The recovery technique, International Journal for Numerical Methods in Engineering 33 (7) (1992) 1331–1364.

[18] M. Ainsworth, J. T. Oden, A posteriori error estimation in finite element analysis, Computer methods in applied mechanics and engineering 142 (1-2) (1997) 1–88.

[19] I. Babuška, W. C. Rheinboldt, A-posteriori error estimates for the finite element method, International Journal for Numerical Methods in Engineering 12 (10) (1978) 1597–1615.

[20] I. Babuška, T. Strouboulis, C. Upadhyay, A model study of the quality of a posteriori error estimators for linear elliptic problems. error estimation in the interior of patchwise uniform grids of triangles, Computer Methods in Applied Mechanics and Engineering 114 (3-4) (1994) 307–378.

[21] O. C. Zienkiewicz, J. Z. Zhu, A simple error estimator and adaptive procedure for practical engineerng analysis, International journal for numerical methods in engineering 24 (2) (1987) 337–357.

[22] A. Naga, Z. Zhang, A posteriori error estimates based on the polynomial preserving recovery, SIAM Journal on Numerical Analysis 42 (4) (2004) 1780–1800.

[23] Y. Huang, J. Li, C. Wu, W. Yang, Superconvergence analysis for linear tetrahedral edge elements, Journal of Scientific Computing 62 (1) (2015) 122–145.

[24] L. Wang, Q. Zhang, Z. Zhang, Superconvergence analysis and ppr recovery of arbitrary order edge elements for maxwell's equations, Journal of Scientific Computing 78 (2) (2019) 1207–1230.

[25] J. Wan, An automated adaptive procedure for three-dimensional metal forming simulations, Rensselaer Polytechnic Institute, 2006.

[26] E. S. Seol, M. S. Shephard, Efficient distributed mesh data structure for parallel automated adaptive analysis, Engineering with Computers 22 (3-4) (2006) 197–213.

[27] X. Li, M. S. Shephard, M. W. Beall, 3d anisotropic mesh adaptation by mesh modification, Computer methods in applied mechanics and engineering 194 (48) (2005) 4915–4950.

[28] Q. Lu, M. S. Shephard, S. Tendulkar, M. W. Beall, Parallel mesh adaptation for high-order finite element methods with curved element geometry, Engineering with Computers 30 (2) (2014) 271–286.

[29] S. Shiraiwa, O. Meneghini, R. Parker, P. Bonoli, M. Garrett, M. C. Kaufman, J. C. Wright, S. Wukitch, Plasma wave simulation based on a versatile finite element method solver, Physics of Plasmas 17 (5) (2010) 056119.

[30] O. Meneghini, S. Shiraiwa, Full wave simulation of lower hybrid waves in iter plasmas based on the finite element method, Plasma and Fusion Research 5 (2010) S2081–S2081.

[31] N. Bertelli, E. Valeo, D. Green, M. Gorelenkova, C. Phillips, M. Podestà, J. Lee, J. Wright, E. Jaeger, Full-wave simulations of icrf heating regimes in toroidal plasma with non-maxwellian distribution functions, Nuclear Fusion 57 (5) (2017) 056035.