RF wave simulation for cold edge plasmas using the MFEM library

S. Shiraiwa^{1,*}, J. C. Wright¹, P. T. Bonoli¹, T. Kolev², and M. Stowell²

Abstract. A newly developed generic electro-magnetic (EM) simulation tool for modeling RF wave propagation in SOL plasmas is presented. The primary motivation of this development is to extend the domain partitioning approach for incorporating arbitrarily shaped SOL plasmas and antenna to the TORIC core ICRF solver, which was previously demonstrated in the 2D geometry [S. Shiraiwa, et. al., "HISTORIC: extending core ICRF wave simulation to include realistic SOL plasmas", Nucl. Fusion in press], to larger and more complicated simulations by including a 3D realistic antenna and integrating RF rectified sheath potential model. Such an extension requires a scalable high fidelity 3D edge plasma wave simulation. We used the MFEM [http://mfem.org], open source scalable C++ finite element method library, and developed a Python wrapper for MFEM (PyMFEM), and then a radio frequency (RF) wave physics module in Python. This approach allows for building a physics layer rapidly, while separating the physics implementation being apart from the numerical FEM implementation. An interactive modeling interface was built on π Scope [S Shiraiwa, et. al. Fusion Eng. Des. 112, 835] to work with an RF simulation model in a complicated geometry.

1. Introduction

SOL plasmas affect both antenna coupling and core heating/current drive performance through various mechanisms, including RF rectified sheath potential for the ion cyclotron range of frequency (ICRF) problem and parasitic edge absorption for the lower hybrid (LH) waves. The need of incorporating global SOL plasmas to RF heating and current drive simulations has been widely recognized in order to better understand and assess roles of SOL plasmas. Such a simulation requires theoretical and computational advancement from the past RF simulations, where the majority of the works started from separating the plasma into core and SOL/antenna regions.

The separation has been made mainly motivated by the need of capturing rather different aspects of wave physics in the two regions. In the core, spatially dispersive hot plasma conductivity is important to properly model the wave propagation and absorption, whereas, in the SOL/antenna region, reproducing the detailed 3D geometry of antenna structure is important. The issue is that, after separating the core and SOL/antenna region, a rather simplified assumption on the interfacing layer was made to define a boundary condition. Several works [1, 2, 3, 4] in the literature have attempted to eliminate or at least soften the impact of the separation. These works oriented toward constructing a monolithic solver to handle the two regions together, and they were partially successful in expanding the existing

RF simulation capability to include both core and edge. However, none of them represents a simulation approach universally applicable to all (from ICRF to EC) frequency ranges to include hot core plasma, surrounding cold SOL plasmas and a complicated 3D antenna structure all together.

Recently we tackled this issue with an alternative strategy, in which we solved core and edge regions separately, but stitch them afterwards. An initial work of this approach ("domain partitioning") was presented at the previous conference [5], and a successful simulation of H-minority ICRF heating on Alcator C-Mod was

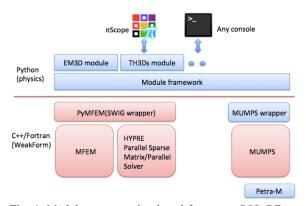


Fig. 1. Module structure developed for a new SOL RF wave simulation. The bottom layer (below the red line) is C++/Fortran libraries and their python wrappers. The modules shown in red and π Scope are already available as open source software.

¹MIT Plasma Science and Fusion Center, 190 Albany St Cambridge, 02139, USA

²Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551, USA

Corresponding author: shiraiwa@psfc.mit.edu

[©] The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (http://creativecommons.org/licenses/by/4.0/).

recently published [6]. In ref 5 and 6, the wave in the hot core plasma was solved by the TORIC spectral solver [7], and the SOL and antenna regions were discretized by unstructured triangles (HIS-TORIC: <u>Hybrid Integration of SOL to TORIC</u>) and solved using a commercial FEM package, COMSOL. We showed in 2D axisymmetric simulation that the correct final solution is obtained from the superposition of "mode solutions".

While the formulation for connecting two regions derived in Ref 6 is not restricted to 2D geometry, extending the work to 3D [8] requires solving a much larger linear problem and better automation of the simulation workflow, both of which are difficult to realize when using commercial software. Furthermore, we are envisioning the need of integrating additional physics to improve the reality of simulation. Such physics includes the RF rectified sheath potential on the material surface [9] and the interaction of RF waves with edge turbulence. Introducing any hot conductivity effect in the SOL region would make a linear system matrix denser, which would potentially require massively parallel computation. Driven by this long-term research direction, we developed a new FEM framework and an edge RF wave simulation environment based on an open source scalable FEM library, MFEM [10].

In this paper, we discuss the software stack we developed in Sec 2. Initial result of applying it to a HISTORIC simulation is discussed in Sec 3, and Sec 4 summarizes the work.

2 Module structure

Figure 1 shows the module structure we developed to implement a new SOL wave simulation. We use the MFEM (modular finite element method) library for the finite element discretization. MFEM is a relatively compact C++ library developed in LLNL and is available as open source software. It uses the Hypre linear solver library [11] and allows for directly filling a distributed sparse matrix in a ParCSR (dual-CSR, compressed row storage) format. The library is written for a generic FEM analysis, not specifically for solving the Maxwell's equations. However, it is equipped with basis functions and linear/bilinear integrators required to formulate a linear system corresponding to a Maxwell problem.

2.1 PyMFEM wrapper

We developed a python wrapper layer (PyMFEM [12]) for MFEM and implemented physics modules discussed in the next section using Python. Such a hybrid approach is advantageous for rapid development and has been used in other open source FEM projects such as the FEniCS project [13].

The main role of the PyMFEM layer is to provide an access to the MFEM library objects from Python. The wrapper layer is generated mostly by SWIG (simple wrapper interface generator). A user can instantiate a C++ MFEM object and call its methods from Python. We designed the wrapper so that a user can perform

most of the operations using Python's native objects such as List or Numpy array.

An important feature implemented in PyMFEM is a wrapper class of C++ function pointer used in the FunctionCofficient and other similar objects in MFEM. In MFEM, coefficients appearing in an inner product, such as $\frac{1}{\mu_0}$ in Eq. 1, is given as a C++ function pointer, where the function arguments are spatial coordinates. PyMFEM allows for using a regular Python class method instead of a C++ function. This way, we can define material properties such as a cold plasma tensor in Python.

The wrapper layer also provides an easy access to the ParCSR matrix. A user can fill the local portion of ParCSR matrix in each processor using a standard method available in the scipy.sparse module and combine them to make a final distributed ParCSR matrix. It is also possible to decompose a ParCSR matrix into scipy.sparse matrices distributed among processors.

2.2 Petra-M (<u>Physics Equation</u> <u>Translator for MFEM)</u>

In order to solve a physics problem using MFEM, a user needs to translate physics equations, often given in the form of partial differential equations (PDEs), into a weak form. It would be much more convenient if a user can use regular physics terminologies when configuring a simulation. For this purpose, we developed a physics equation translation layer (Petra-M), which corresponds to the blue part of Fig. 1. Petra-M consists of two layers. The lower layer is a generic infrastructure providing routines to define a multi-physics problem, where a solution is expressed using a combination of multiple finite element function spaces. The support for a complex valued problem is also realized in this layer. For a complex valued problem such as a frequency domain simulation, Petra-M fills the real part and imaginary part separately and generates a final linear system matrix either as a complex value matrix, or 2x2 real value block matrix. Differences between serial and parallel (MPI) environments are absorbed in this layer,

On the upper layer of Petra-M, we define the PDE system we want to solve. A module to solve a 3D frequency domain Maxwell problem is implemented here. This module fills a linear system for the following weak form of Maxwell's equation [14, 15]

$$\left(\nabla \times F, \frac{1}{\mu_0} \nabla \times E\right) - (F, (\omega^2 \epsilon - j\sigma \omega)E) + \langle F, n \frac{1}{\mu_0} \nabla \times E \rangle = -j\omega(F, J_{\text{ext}}),$$

where E and F are test and trial functions in the H(curl) finite element space, and two inner products are defined as $(u,v) \stackrel{\text{def}}{=} \int uv \, d\Omega$ and $\langle u,v \rangle \stackrel{\text{def}}{=} \int uv \, d\partial\Omega$, corresponding to volumetric and surface contribution,

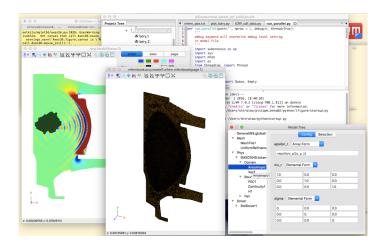


Fig 2. GUI developed in πScope. 3D mesh geometry, RF physics parameter setting, and a solution of RF field is shown. The right bottom panel shows how parameters for anisotropic media are specified, showing that the relative dielectric constant is given as "=epsilonr_pl(x,y,z)", where epsilonr_pl is a user defined python function. The dark region in the left most window indicates the typical size of domain solved by one CPU (total 16 CPU was used in this case).

respectively. The later term is yielded from a partial integration.

The first and second terms in the above form correspond to the curl-curl operator and the conductivity, while the last term is the contribution from volumetric external currents. Presently, the module supports boundary conditions (BCs) including, PEC (perfect electric conductor), PMC (perfect magnetic conductor), coax and waveguide ports BC, Floquet periodic BC, surface current BC, surface magnetic field BC, electric field BC, and impedance BC.

Optionally, a user can add a divergence-free constraint for the current. When using this constraint, the solution space is expanded to contain two finite element spaces, one for the electric field and the one for the time derivative of charge. Petra-M discretizes the second one using the H¹ element and formulates a linear system in the saddle point format. This constraint is used in the vacuum region in the simulations shown in the next section.

2.3 User interface layer

Setting up a 3D RF simulation in a realistic geometry is not a trivial task. A graphical user interface is prepared in π Scope [16], an open-source python based data analysis environment, in which a user can set domain and boundary conditions as discussed above. A screen capture of the user interface is shown in Fig 2.

Additionally, Petra-M supports generating a python script from the model settings prepared by using the GUI. The generated script is easy to modify, allowing for a user to setup, for example, a parametric scan of simulation parameters quickly from an existing simulation.

3. Application to HIS-TORIC

The original HIS-TORIC simulation involves cumbersome steps including running multiple COMSOL simulations programmed by Matlab and executing separate TORIC simulations. We re-implemented the edge RF wave physics model in HIS-TORIC using Petra-M, which moved the most of simulation workflow under π Scope control and allows for extending HIS-TORIC to the 3D geometry [8]. Here, as a verification case, we discuss the simulation of the H-minority heating scenario for an Alcator C-Mod discharge, the same simulation as shown in Ref 3, but ran with TORIC and Petra-M is shown in Fig. 3.

This model simulates an ICRF wave with the n=10 mode using the axis-symmetry. A single toroidal mode in 3D is imposed in a similar way as discussed in Ref. 16, namely by using extremely thin (0.1 deg) toroidal sector and imposing a Floquet boundary condition. On the left figure, the model geometry and boundary conditions are shown. The SOL region has 2.5 cm thickness surrounding the core TORIC region (not shown). Outside the SOL plasma is set to vacuum, and we imposed a divergence-free constraint in this vacuum region. The ICRF wave at 80MHz is excited using surface currents on the low field side curved surface. The left figure also shows the size of mesh used in the simulation. The FEM problem is assembled using Petra-M and solved using MUMPS direct sparse linear solver [18].

The right figure shows the η component ($\sim B \times \nabla \psi$) of the RF electric field after combining TORIC and Petra-M regions. It can be seen that the wave smoothly propagates through the interface between SOL and TORIC regions towards the core and is strongly absorbed in the core. Note that there is also observed a short wavelength ICW wave near $R = R_0$, where R_0 is the major radius, which wouldn't exist if a cold plasma approximation were used in the core.

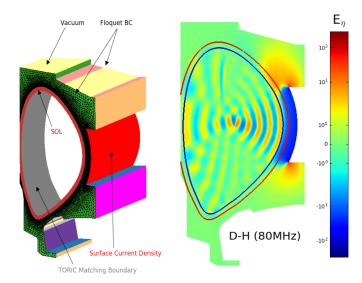


Fig 3: (Left) a simulation geometry to implement axis-symmetric simulation of ICRF wave propagation in Alcator C-Mod. (Right) a simulation result using the same input parameter presented in Ref. 3. Note that the actual geometry is extremely thin in the toroidal direction. The left figure is enlarged in the toroidal direction for better visualization.

4. Conclusions

In this paper, we discussed the implementation details of a new edge SOL plasma simulation environment. We built our simulation model on the top of the open source MFEM C++ FEM library. Petra-M is developed using the Python wrapper for MFEM (PyMFEM) to set up a MFEM simulation using physics (not weak from) terminologies, and a 3D frequency domain RF simulation model is then developed as a part of Petra-M. A graphical user interface is developed on $\pi Scope$ to configure an RF simulation with a complicated realistic geometry.

The cold plasma wave simulation model used in the previous HIS-TORIC simulation was re-implemented using Petra-M. The model was tested by reproducing the Alcator C-Mod H minority heating simulation presented in Ref. 3. With Petra-M, the model can be extended in 3D as presented at this conference [8], and the HIS-TORIC simulation workflow can be fully automated. This open-soruce software based platform will be used to integrate more advanced physics such as an RF rectified potential discussed in [9] in our future work.

Work supported by the U.S. Department of Energy, Office of Science, Office of Fusion Energy Sciences, using User Facility Alcator C-Mod, under Award Number DE-FC02-99ER54512 and by US DoE Contract DE-FC02-01ER54648 under a Scientific Discovery Through Advanced Computing Initiative. The work at Lawrence Livermore National Laboratory was performed under the auspices of DoE under Contract DE-AC52-07NA27344, LLNL-PROC-703397. The simulations presented in this paper were performed on the MIT-PSFC partition on the Engaging cluster at the MGHPCC (www.mghpcc.org).

References

- 1. T. G. Jenkins, D.N. Smithe and R. I. Pinsker, 2015 AIP Conf. Proc. 030003
- D. L. Green, L. Berry, G. Chen, P. Ryan, J. Canik and E. Jaeger, 2011 Phys. Rev. Lett. 107, 145001

- O. Meneghini, S. Shiraiwa and R. Parker, Phys. Plasmas 16, 090701 (2009).
- S. Shiraiwa, J. Ko, O. Meneghini, et. al, Phys. Plasmas 18, 080705 (2011).
- J. C. Wright and S. Shiraiwa, AIP Conf. Proc., 1689, 050007 (2015).
- 6. S. Shiraiwa, J.C. Wright, J.P. Lee, and P.T. Bonoli, Nucl. Fusion 57 086048 (2017).
- 7. M. Brambilla, 1999 Plasma Phys. Control. Fusion **41**, 1
- 8. J. C. Wright and S. Shiraiwa, "Antenna to Core: A New Approach to RF Modelling" in 22nd Topical Conference on Radio Frequency Power in Plasmas, European Phys. J. Web of Conferences, this issue, Inv-13 (2017).
- 9. J. R. Myra and H. Kohno, "A post-processing method to simulate the generalized RF" in 22nd Topical Conference on Radio Frequency Power in Plasmas, European Phys. J. Web of Conferences, this issue, Inv-13 (2017).
- 10. MFEM (http://mfem.org)
- 11. HYPRE : Scalable Linear Solvers and Multigrid (http://llnl.gov/casc/hypre)
- 12. PyMFEM: Python wrapper for MFEM (https://github.com/MFEM/PyMFEM)
- 13. FEniCS project (https://fenicsproject.org)
- 14. J. Jin, Finite Element Method in Electromagnetics (New York: Wiley) (1993)
- 15. A. Bondeson A., T. Rylander and P. Ingelström Computational Electromagnetics (Berlin: Springer) (2000)
- 16. S Shiraiwa, T Fredian, J Hillairet, J Stillerman, Fusion Engineering and Design **112**, 835-838
- 17. S Shiraiwa, O Meneghini, R Parker, G Wallace, J Wilson, 2009 AIP Conf. Proc. **1187**, 363
- 18. MUMPS: a parallel sparse direct solver (http://mumps.enseeiht.fr/)