SMWD 2023, KTH Royal Institute of Technology, Stockholm, $23^{rd} - 25^{th} May 2023$

An Accelerated Finite Element-Boundary Integral Code Developed using Open Source Software

Niklas Wingren[(1)] and Daniel Sjöberg[(1)]*
(1) Department of Electrical and Information Technology, Lund University, Lund, Sweden

Summary

A finite element-boundary integral code accelerated by the adaptive cross approximation is presented. The code can use two different formulations, both of which are accelerated in the same way. The iterative solution process is described together with the preconditioners which are necessary for convergence. The code is based on open source software, with the most significant packages being listed and described.

1 Introduction

When simulating scattering problems with complicated geometries and materials, the finite element-boundary integral (FE-BI) method can be a good choice of computational method [1]. However, it suffers from the computational costs due to dense matrices inherent to the BI part (method of moments). For this reason, acceleration schemes are needed for simulation of problems with a large number of unknowns.

2 Method Description

The FE-BI method is based on a FE part in a computation volume and a BI part on the boundary of the volume. The FE formulation used in our code is based on the electric field and can be used for inhomogeneous materials, with possibilities for many different material models. The formulation will be somewhat different if a bi-anisotropic model is used compared to a simpler anisotropic model. Both result in similar blocks in the final system matrix though. For the BI formulations on the other hand, there are two formulations implemented where the effect on the final system matrix is more pronounced. Formulation 1 uses the electric field integral equation on the boundary, and corresponds to the TE formulation in [2]. Formulation 2 uses both the electric and magnetic field integral equations, and corresponds to the S-VT [E,J] formulation in [3]. Formulation 2 results in a symmetric system matrix and does not suffer from interior resonances, as opposed to formulation 1.

Regardless of which formulation is used, the system matrix is constructed from sparse FE blocks and dense BI blocks. The dense blocks are compressed by a multilevel ACA. An octree is used to subdivide the geometry in a multilevel fashion. On each level, the ACA is used to assemble compressed blocks of source and observation degrees of freedom (DoFs) corresponding to far interacting octree groups. These blocks are stored in outer-product form, that is on the form UV^{H} where the inner dimensions of U and V^{H} are significantly smaller than the outer dimensions. Blocks are further compressed using the QR and SVD as described in [4].

Iterative solvers are based on matrix-vector multiplication, which is a natural operation to perform for compressed blocks on outer-product form. It is also a good choice of solver as the sparse nature of FE blocks can be preserved. In our case, LGMRES [5] is used to avoid some convergence issues with the standard restarted GMRES. Preconditioning is necessary for convergence of the solution, and for both formulations sparsification of the dense BI blocks is used. Blocks are sparsified by only considering basis and testing functions in the same triangle, as described in [6]. In formulation 1, blocks are also scaled as described in [7]. Right preconditioning is used, and the preconditioner is applied by solving the sparsified system for the vector to be preconditioned as the right-hand side. The solution can either be direct using a sparse LU decomposition, or iterative using BiCGSTAB in turn preconditioned by an incomplete LU decomposition.

3 Code Details

The code is written in Python with appropriate bindings to other languages when necessary for integration of certain packages. The FE part of the code is based on FEniCSx (`https://fenicsproject.org`), consisting of DOLFINx, FFCx, Basix and UFL. BI matrix entries are computed using mesh and basis functions from DOLFINx and Basix, and evaluation of singular integrals using DEMCEM (`https://github.com/thanospol/DEMCEM`) is used. For the multilevel ACA, balanced octrees are built using AdaptOctree (`https://github.com/Excalibur-SLE/AdaptOctree`. Geometry and meshing is done using gmsh (`https://gmsh.info`). SciPy (`https://scipy.org`) is used for iterative solvers and preconditioning is done using its bindings to SuperLU (`https://portal.nersc.gov/project/sparse/superlu`). Code written in Python for the BI part uses Numba (`https://numba.pydata.org`) for just-in-time compilation to speed up many tasks.

4 Conclusion

We have given an overview of a FE-BI code based on open source software. The main parts of the method have been described, as well as the most important open source packages that the code is based on. A more in-depth description of the code architecture as well as numerical examples demonstrating the capabilities of the code will be presented at the event.

5 Acknowledgements

This work was supported in part by the Swedish Armed Forces, in part by the Swedish Defence Materiel Administration, in part by the National Aeronautics Research Program and in part by the Swedish Governmental Agency for Innovation Systems.

References

[1] J. Jin, *Theory and Computation of Electromagnetic Fields*. Hoboken, NJ, USA: John Wiley & Sons, Ltd, 2010.

[2] X.-Q. Sheng, J.-M. Jin, J. Song, C.-C. Lu, and W. C. Chew, "On the formulation of hybrid finite-element and boundary-integral methods for 3-D scattering," *IEEE Transactions on Antennas and Propagation*, vol. 46, no. 3, pp. 303–311, 1998.

[3] M. M. Botha and J.-M. Jin, "On the variational formulation of hybrid finite element-boundary integral techniques for electromagnetic analysis," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 11, pp. 3037–3047, 2004.

[4] M. Bebendorf and S. Kunis, "Recompression techniques for adaptive cross approximation," *The Journal of Integral Equations and Applications*, vol. 21, no. 3, pp. 331–357, 2009.

[5] A. H. Baker, E. R. Jessup, and T. Manteuffel, "A technique for accelerating the convergence of restarted GMRES," *SIAM Journal on Matrix Analysis and Applications*, vol. 26, no. 4, pp. 962–984, 2005.

[6] X. An and Z.-Q. Lü, "An efficient finite element-boundary integral method solving electromagnetic scattering problems," *Microwave and Optical Technology Letters*, vol. 51, no. 9, pp. 2065–2071, 2009. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/mop.24538

[7] J. Guan, S. Yan, and J. Jin, "An accurate and efficient finite element-boundary integral method with gpu acceleration for 3-D electromagnetic analysis," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 12, pp. 6325–6336, 2014.