

Residual-Based Adaptive Refinement for Meshless Eigenvalue Solvers

Thomas Kaufmann*, Christian Engström*, Christophe Fumeaux†

*Laboratory for Electromagnetic Fields and Microwave Electronics (IFH), ETH Zurich, Switzerland

†School of Electrical and Electronic Engineering, The University of Adelaide, SA 5005, Australia

Corresponding Email: thomas.kaufmann@ifh.ee.ethz.ch

Abstract—The concept of an adaptive meshless eigenvalue solver is presented and implemented for two-dimensional structures. Based on radial basis functions, eigenmodes are calculated in a collocation approach for the second-order wave equation. This type of meshless method promises highly accurate results with the simplicity of a node-based collocation approach. Thus, when changing the discrete representation of a physical model, only node locations have to be adapted, hence avoiding the numerical overhead of handling an explicit mesh topology. The accuracy of the method comes at a cost of dealing with poorly-conditioned matrices. This is circumvented by applying a leave-one-out-cross-validation optimization algorithm to get stable results. A node adaptivity algorithm is presented to efficiently refine an initially coarse discretization. The convergence is evaluated in two numerical examples with analytical solutions. The most relevant parameter of the adaptation algorithm is numerically investigated and its influence on the convergence rate examined.

I. INTRODUCTION

A meshless method based on radial basis functions (RBFs) in a collocation scheme is known as Radial Point Interpolation Method (RPIM) [1] or Kansa's Method [2]. Meshless methods avoid the generation and handling of an explicit mesh for the numerical solution of differential equations. No computational overhead is required to take care of tetrahedron or hexahedron mesh data such as volumes, surfaces or edges. Instead, an arbitrarily distributed set of nodes is created and interpolations based on RBFs are used in a collocation scheme. This framework facilitates adaptive refinement significantly. It has previously been shown that RBF collocation methods yield very high accuracy and fast convergence [3]. A time-domain implementation of RPIM for Maxwell's equations has been previously introduced in [4], [5].

In this paper, an eigenvalue solver for the second-order wave equation is presented. In contrast to [6], the fields are calculated directly rather than solving for the shape parameter. Then a refinement algorithm, based on a residual error indicator is introduced. An alternative residual-based method was recently presented for source problems in mechanics [7]. For structured node distributions, an adaptivity algorithm has been presented in [8]. Even though the algorithm presented here is for an RBF approach, it can also be directly applied to other meshless collocation methods, such as Smooth Particle Electromagnetics (SPEM) [9]. The convergence of the calculation error obtained by applying the refinement algorithm is evaluated on two 2D numerical examples with analytical

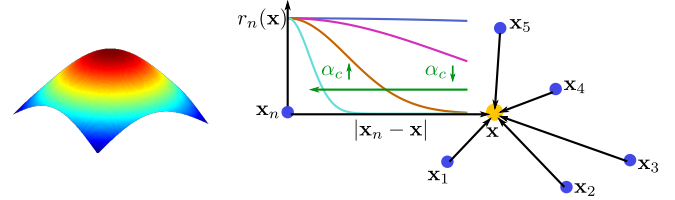


Fig. 1: Radial basis function $r_n(\mathbf{x} - \mathbf{x}_n)$ (left), and interpolation of the field value at position \mathbf{x} , based on the values at the collocation nodes $\mathbf{x}_1, \dots, \mathbf{x}_n$ (right). The parameter α_c controls the flatness of the basis function r_n .

solutions: a square and a L-shaped perfectly conducting cavity resonator. In the algorithm a parameter has to be set initially to determine the rate at which nodes are added into the domain. An investigation is performed to understand the behavior for different values of this parameter.

II. RADIAL BASIS FUNCTIONS

Interpolation of field values with RBFs gives high accuracy for arbitrarily scattered node locations. RBFs are gaining momentum in many fields of science, e.g. neural networks and computer vision, as well as for solving partial differential equations. In [10] the scheme is presented in detail, thus here only the main points are summarized.

To interpolate a field component $u(\mathbf{x})$ at the position $\mathbf{x} = (x, y)$, the following equation holds:

$$\langle u(\mathbf{x}) \rangle = \sum_{n=1}^N a_n r_n(\mathbf{x}) = \mathbf{r}(\mathbf{x})^T \mathbf{a}, \quad (1)$$

for the N nodes inside the computational domain. The RBFs of Gaussian type with a given shape parameter α_c can be expressed as

$$r_n(\mathbf{x}) = \exp \left(-\alpha_c \frac{|\mathbf{x}_n - \mathbf{x}|^2}{d_c^2} \right). \quad (2)$$

The distances to neighboring nodes $|\mathbf{x}_n - \mathbf{x}|$ are normalized by the average node distance d_c between all nodes. Fig. 1 illustrates how the shape parameter α_c controls the flatness of the basis functions.

In a preprocessing step, the interpolation parameter vector \mathbf{a} is calculated by setting up a system to interpolate the field component $u(\mathbf{x})$ at all node locations $\mathbf{U}_s =$

$[u(\mathbf{x}_1), u(\mathbf{x}_2), \dots, u(\mathbf{x}_N)]^T$. The system takes the form:

$$\begin{pmatrix} u(\mathbf{x}_1) \\ \vdots \\ u(\mathbf{x}_N) \end{pmatrix} = \begin{pmatrix} r_1(\mathbf{x}_1) & \cdots & r_1(\mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ r_N(\mathbf{x}_1) & \cdots & r_N(\mathbf{x}_N) \end{pmatrix} \begin{pmatrix} a_1 \\ \vdots \\ a_N \end{pmatrix}, \quad (3)$$

or in compact notation

$$\mathbf{U}_s = \mathbf{R}_0 \mathbf{a}. \quad (4)$$

The shape parameter α_c determines the interpolation accuracy and the condition number of the interpolation matrix \mathbf{R}_0 . For larger values of α_c , the RBFs approach a delta-like shape, which means that the field value at one node does not depend on the value at other positions. Consequently, the interpolation accuracy degrades dramatically. For decreasing values of α_c , the basis functions become flatter and flatter, which means that the field value at one position is heavily weighted by the values at all surrounding nodes. It has been shown [3] that in the ideal case the accuracy increases exponentially by decreasing α_c . Numerically though, this translates into a interpolation matrix \mathbf{R}_0 with all entries almost identical. Thus the rank of the matrix decreases and the system becomes ill-conditioned. Hence, the choice of the shape parameter is crucial for accurate results. Several methods exist to circumvent this issue, e.g. [11], [12]. Here, we chose the “leave-one-out-cross-validation” algorithm [13] to find an optimized value of the shape parameter α_c for a given node distribution.

The system (4) can be numerically inverted if the interpolation matrix \mathbf{R}_0 is non-singular, i.e. if no nodes coincide and if the shape parameter is sufficiently large. The vector \mathbf{a} is then obtained as

$$\mathbf{a} = \mathbf{R}_0^{-1} \mathbf{U}_s. \quad (5)$$

and the shape functions $\Psi(\mathbf{x})$ and their derivatives are calculated as

$$\begin{aligned} \langle u(\mathbf{x}) \rangle &= \mathbf{r}^T(\mathbf{x}) \mathbf{a} = \mathbf{r}^T(\mathbf{x}) \mathbf{R}_0^{-1} \mathbf{U}_s \\ &= \Psi(\mathbf{x}) \mathbf{U}_s \end{aligned} \quad (6a)$$

$$\begin{aligned} \langle \partial_\kappa u(\mathbf{x}) \rangle &= \partial_\kappa \mathbf{r}^T(\mathbf{x}) \mathbf{a} = \partial_\kappa \mathbf{r}^T(\mathbf{x}) \mathbf{R}_0^{-1} \mathbf{U}_s \\ &= \partial_\kappa \Psi(\mathbf{x}) \mathbf{U}_s \end{aligned} \quad (6b)$$

$$\begin{aligned} \langle \partial_\kappa^2 u(\mathbf{x}) \rangle &= \partial_\kappa^2 \mathbf{r}^T(\mathbf{x}) \mathbf{a} = \partial_\kappa^2 \mathbf{r}^T(\mathbf{x}) \mathbf{R}_0^{-1} \mathbf{U}_s \\ &= \partial_\kappa^2 \Psi(\mathbf{x}) \mathbf{U}_s \end{aligned} \quad (6c)$$

with ∂_κ the spatial derivative in $\kappa = x, y$ direction.

III. EIGENVALUE SOLVER

An eigenvalue solver for a second-order problem was presented in [6], where the problem was solved for the interpolation parameter \mathbf{a} . Here we directly compute the electric field \mathbf{E}_z pointing perpendicularly to the computational domain. The eigenvalues $\lambda = \omega^2/c_0^2$ are calculated from

$$-\mathcal{L}^h \mathbf{E}_z = \lambda \mathbf{E}_z \quad \text{in } \Omega, \quad (7)$$

$$\mathbf{E}_z = 0 \quad \text{PEC on } \partial\Omega. \quad (8)$$

The stiffness matrix \mathcal{L}^h contains the Laplace operator, discretized with the second-order derivative of the RBF interpo-

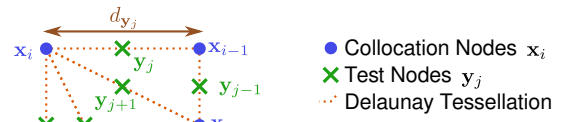


Fig. 2: Delaunay Tessellation of a set of collocation nodes. The test nodes are placed on the edges of the Delaunay cells with length d_{y_j} .

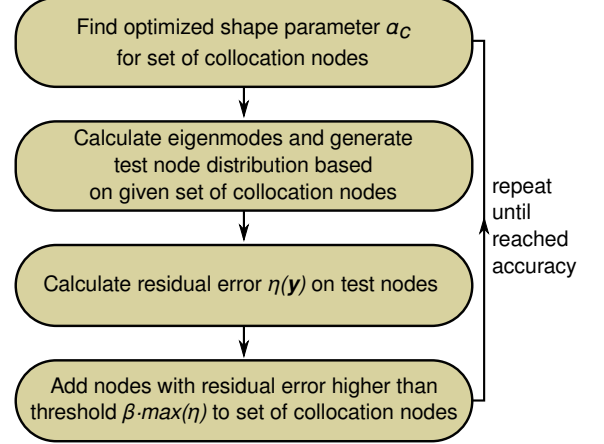


Fig. 3: Flow-chart of the refinement algorithm

lation shape functions

$$\mathcal{L}^h(\mathbf{x}_i) = \partial_x^2 \Psi(\mathbf{x}_i) + \partial_y^2 \Psi(\mathbf{x}_i). \quad (9)$$

The eigenvalue solver package ARPACK [14] for full matrices is used to calculate the eigenpairs $\{\mathbf{E}_z^l, \lambda^l\}$.

IV. REFINEMENT ALGORITHM

In the refinement algorithm, the l^{th} eigenvector \mathbf{E}_z^l and eigenvalue λ^l are calculated at an initial set of collocation nodes. A set of test nodes is then generated via a Delaunay tessellation [7] (Fig. 2). The residuals

$$\begin{aligned} \eta(\mathbf{y}_j) &= \nu_1 d_{y_j}^2 |\mathcal{L}^h(\mathbf{y}_j) \mathbf{E}_z^l - \lambda^l \Psi(\mathbf{y}_j) \mathbf{E}_z^l|^2 \quad \text{Internal Node} \\ &+ \nu_2 d_{y_j}^2 |\Psi(\mathbf{y}_j) \mathbf{E}_z^l|^2 \quad \text{Boundary Node} \end{aligned} \quad (10)$$

are then calculated on all test nodes. The weighting factors ν_1, ν_2 are used to adjust the different scales of the internal and boundary test node residuals. ν_2 is several orders of magnitude larger than ν_1 . Nodes with a residual error larger than a predefined threshold

$$\eta(\mathbf{y}_j) \geq \beta \max_l \eta(\mathbf{y}_l), \beta \in [0, 1] \quad (11)$$

are subsequently added to the set of collocation nodes. The choice of the threshold parameter β determines how many nodes are added at each iteration. For every new node distribution, another optimized shape parameter α_c is obtained, and the eigenpair $\mathbf{E}_z^l, \lambda^l$ is computed again. This results in an adaptation scheme that combines both spatial discretization and shape adaptivity. The algorithm is summarized in Fig. 3.

V. NUMERICAL EXPERIMENTS

The refinement algorithm has been evaluated in two numerical experiments. The transverse magnetic (TM) eigenmodes

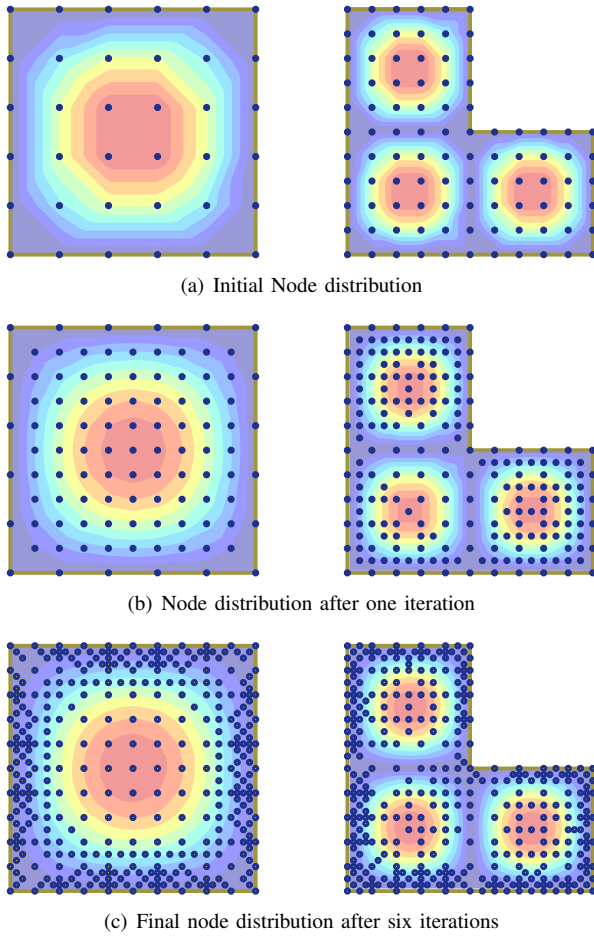


Fig. 4: Node distributions at different stages of the refinement iteration for the square domain (I) with TM_{11} mode (left) and L-shaped domain (II) for the 1st smooth eigenmode (right).

of a square cavity and the first smooth eigenmodes of an L-shaped domain, both enclosed with perfect electric conductors, have been computed. Both problems have smooth non-singular sinusoidal solutions. Starting with a coarse node distribution, the presented adaptive algorithm was applied to increase the accuracy. The chosen parameters for the weighting of the internal and boundary test nodes (10) were $\nu_1 = 1, \nu_2 = 10^4$.

The two computational domains with node distributions at different times of the adaptive algorithm (11) with $\beta = 5 \cdot 10^{-3}$ are displayed in Fig. 4. In the case of the square cavity, the adapted node distributions are illustrated for the TM_{11} eigenmode. In the case of the L-shaped domain, the adapted mode shown is the first smooth eigenmode. In both cases, the final node distributions show that the added nodes tend to cluster close to the boundary. This correlates with the findings in [15] where it was observed that computations using RBFs become more accurate when increasing the node density close to boundaries.

The eigenvalues of the first six distinct modes for the square domain (I) and the first two smooth eigenmodes for the L-shaped domain (II) were compared to the analytical values from [16] (I) and to the results of the benchmark by Dauge

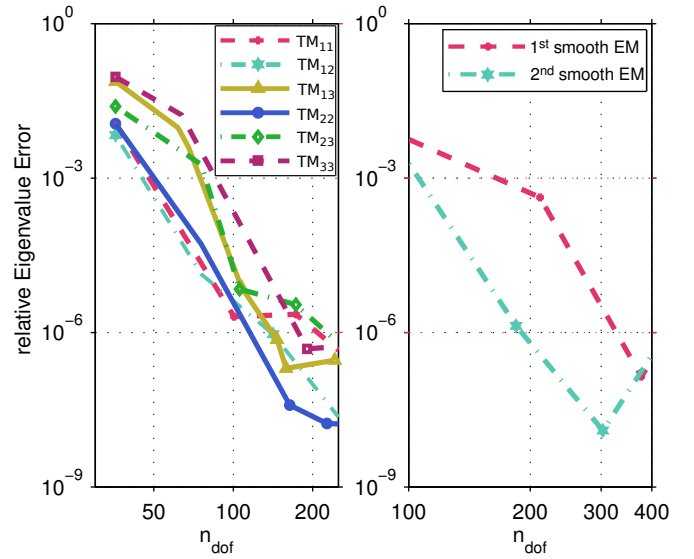


Fig. 5: Convergence of several eigenvalues of a square (left) and L-shaped domain (right).

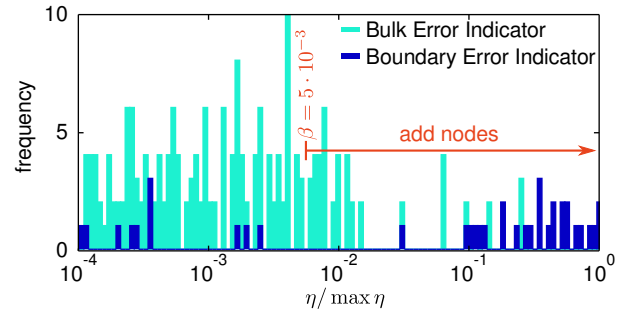


Fig. 6: Histogram of the distribution of the error indicators $\eta(y_j)$ at the second iteration step for the 2nd smooth eigenmode of the L-shaped domain. Nodes with an error function larger than $\beta \max_i \eta(y_i)$ are added to the set of collocation nodes.

[17] (II). In Fig. 5 the error is plotted against the number of degrees of freedom (n_{dof}), which is equal to the number of nodes. The relative error of the eigenvalues decreases rapidly by a factor of at least 10^{-3} for doubling the number of nodes. For all computed modes, the eigenvalue error decreased below at least 10^{-6} after less than six iterations.

A study is performed on the influence of the threshold parameter $\beta \in [0, 1]$. A histogram of a typical distribution of the error indicator $\eta(y_j)$ is illustrated in Fig. 6. In the test case, all nodes with an error indicator larger than $\beta = 5 \cdot 10^{-3}$ times the maximum error are added to the set of collocation nodes. As an illustration, Fig. 7 shows the adaptivity algorithm performed for the TM_{22} mode of the square domain for different values of β . The parameter determines how many nodes are added at each iteration. A high value of the threshold parameter, i.e. $\beta > 0.1$ signifies that only a few nodes are added per iteration. This means that many iterations have to be performed to have a significant gain in accuracy. On the other hand, if the threshold parameter is small, e.g. $\beta = 10^{-7}$, most of the test nodes are added at each iteration which corresponds to a regular

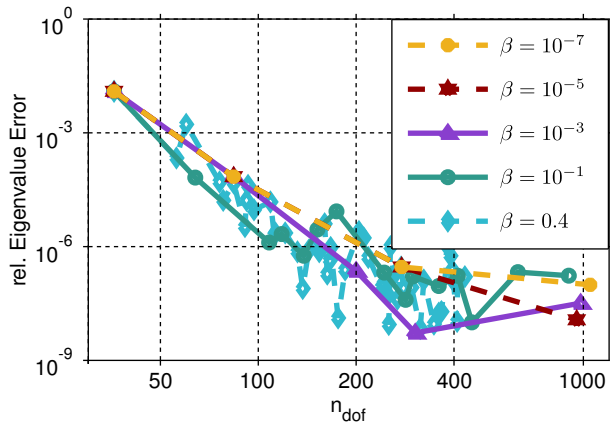


Fig. 7: Influence of the threshold parameter β on the convergence rate. For a large value, only few nodes are added at each iteration, meaning a large number of iteration steps to achieve the required accuracy. For small values of β a large number of nodes are added with each iteration, also at positions where not necessary.

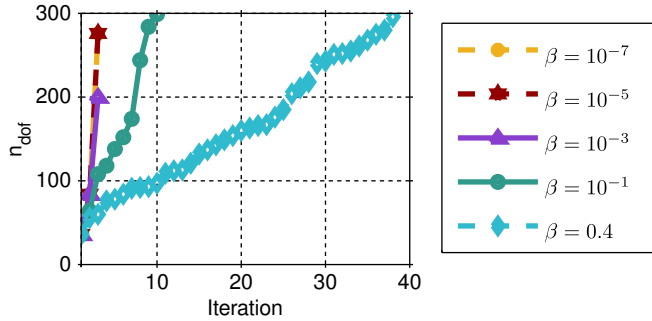


Fig. 8: Development of the number of nodes over the iterations for different values of β in the example of the refinement of a TM_{22} mode in a square domain. Each dot represents an iteration step.

refinement. This leads to many collocation nodes that do not contribute to increasing the accuracy. Thus the convergence rate is lower in that case. Depending on the choice of β , the errors can fluctuate, i.e. the error can grow sporadically when adding nodes at non-optimal locations before eventually continuing to decrease.

The evolution of the number of nodes over many iterations is shown in Fig. 8. A smaller value of β leads to a steeper curve, requiring less iterations to increase the number of nodes in the domain, but placing some at unnecessary locations. A trade-off between those two effects leads to best results, thus a value of $\beta = 5 \cdot 10^{-3}$ has been chosen for these simulations. This choice of parameter is quite robust, as the convergence rate remains stable over a large bandwidth of values of β .

VI. CONCLUSION

An implementation of a second-order Laplace adaptive eigenvalue solver based on radial basis functions has been presented. The implemented refinement algorithm automatically adapts the node distribution for given eigenmodes. In combination with the “leave-one-out-cross-validation” algorithm, the automatic refinement is a powerful tool to increase the computation accuracy and the convergence rate of the solution.

A study was performed on the influence of the threshold

parameter β , and it was observed that this value can be chosen as a robust trade-off between adding too many nodes, including nodes at unnecessary locations, and too few nodes, leading to a high number of iterations to achieve a required accuracy. The proposed algorithm leads to a fast convergence rate of the eigenvalue error over a small number of iterations. Relative errors of less than 10^{-6} can be achieved with only tripling the number of nodes for the demonstrated refinement in the square and L-shaped domains.

Since the refinement algorithm only depends on the shape functions at the test node locations, this algorithm is suitable for any meshless collocation method.

REFERENCES

- [1] G. Liu and Y. Gu, *An Introduction to Meshfree Methods and their Programming*. Dordrecht, The Netherlands: Springer, 2005.
- [2] E. Kansa, “Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics - II solutions to parabolic, hyperbolic and elliptic partial differential equations,” *Computers & Mathematics with Applications*, vol. 19, no. 8-9, pp. 147–161, 1990.
- [3] A. H.-D. Cheng, M. A. Golberg, E. J. Kansa, and G. Zammito, “Exponential convergence and h -c multiquadric collocation method for partial differential equations,” *Numerical Methods for Partial Differential Equations*, vol. 19, no. 5, pp. 571–594, 2003.
- [4] T. Kaufmann, C. Fumeaux, and R. Vahldieck, “The meshless radial point interpolation method for time-domain electromagnetics,” in *IEEE MTT-S Int. Microwave Symposium*. Atlanta, GA, USA: IEEE, June 2008, pp. 61–65.
- [5] Y. Yu and Z. D. Chen, “A 3-d radial point interpolation method for meshless time-domain modeling,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 57, no. 8, pp. 2015–2020, August 2009.
- [6] P. Kowalczyk and M. Mrozowski, “Mesh-free approach to Helmholtz equation based on radial basis functions,” *Journal of Telecommunications and Information Technology*, 2005.
- [7] B. B. T. Kee, G. R. Liu, G. Y. Zhang, and C. Lu, “A residual based error estimator using radial basis functions,” *Finite Elem. Anal. Des.*, vol. 44, no. 9-10, pp. 631–645, 2008.
- [8] T. Rabczuk and T. Belytschko, “Adaptivity for structured meshfree particle methods in 2d and 3d,” *International Journal for Numerical Methods in Engineering*, 2005.
- [9] G. Ala, E. Francomano, A. Tortorici, E. Toscano, and F. Viola, “A smoothed particle interpolation scheme for transient electromagnetic simulation,” *IEEE Transactions on Magnetics*, vol. 42, no. 4, p. 647, 2006.
- [10] M. D. Buhmann, *Radial Basis Functions*. New York, NY, USA: Cambridge University Press, 2003.
- [11] G. E. Fasshauer and J. G. Zhang, “Preconditioning of radial basis function interpolation systems via accelerated iterated approximate moving least squares approximation,” *Computational Methods in Applied Sciences*, vol. 11, pp. 57–75, 2009.
- [12] G. Fasshauer and J. Zhang, “On choosing “optimal” shape parameters for RBF approximation,” *Numerical Algorithms*, vol. 45, no. 1, pp. 345–368, August 2007.
- [13] G. F. Fasshauer, *Meshfree Approximation Methods with MATLAB*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2007.
- [14] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *Arpack User's Guide: Solution of Large-Scale Eigenvalue Problems With Implicitly Restarted Arnoldi Methods (Software, Environments, Tools)*. Soc. for Industrial & Applied Math, 1998.
- [15] B. Fornberg, T. A. Driscoll, G. Wright, and R. Charles, “Observations on the behavior of radial basis function approximations near boundaries,” *Computers and Mathematics with Applications*, vol. 43, no. 3, pp. 473–490, February 2002.
- [16] C. A. Balanis, *Advanced Engineering Electromagnetics*. New York, USA: John Wiley & Sons, 1989.
- [17] M. Dauge, “Benchmark computations for maxwell equations for the approximation of highly singular solutions,” <http://perso.univ-rennes1.fr/monique.dauge/benchmax.html>.