



Optimizing Node Locations for the Method of Fundamental Solutions

Cadet Gabriel T. Bann

Advisors: LTC Randy Boucher & Dr. Theodore V. Hromadka II

Abstract

The Method of Fundamental Solutions (MFS) is a method of solving boundary value problems that is “meshless,” meaning that it does not require domain discretization. Instead, a collection of nodes located outside the problem boundary are used to collocate the boundary and develop a linear combination of basis functions that can be used to approximate the solution to the problem. A major contingency of the accuracy of the resulting approximation function is the location of nodes outside the problem boundary, and the optimization of these locations has not yet been examined in other research. This research proposes an algorithm that completes that goal, and the algorithm is tested using an example problem.

Global Problem

The governing partial differential equation (PDE) is the 3D Laplace’s equation,

$$u_{xx} + u_{yy} + u_{zz} = 0,$$

an equation used extensively in modeling physical phenomena such as ideal fluid flow or heat transfer.

Solution Technique

To solve this PDE we will use a solution function in the form:

$$\hat{w}(x, y, z) = \sum_{j=1}^n c_j g_j(x, y, z),$$

where c_j is the coefficient for the j -th basis function, and where $g_j(x, y, z)$ is that basis function evaluated at the point (x, y, z) . To solve for the coefficients there must be $2n$ known potential values. Then the following matrix equation can be created, in the form $Gc=w$.

$$\begin{bmatrix} g_{00} & g_{01} & g_{02} & \dots & g_{02n} \\ g_{10} & g_{11} & g_{12} & \dots & g_{12n} \\ g_{20} & g_{21} & g_{22} & \dots & g_{22n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{2n0} & g_{2n1} & g_{2n2} & \dots & g_{2n2n} \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ \vdots \\ c_{2n} \end{bmatrix} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_{2n} \end{bmatrix}$$

The matrix G assembles the value of each basis function at every collocation point, and w is the vector of potential values at these locations. The vector c is the vector of coefficients. This vector is solved using MATLAB’s matrix solving algorithm. Finally, the coefficients are substituted back into the approximation function.

Algorithm Description

The goal in this research is to optimize the node locations implemented in the individual basis functions. The following is a step-by-step description of the algorithm that completes this task.

1. Array a set of collocation points and a set of nodal coordinates
2. Set a maximum number of nodes that are desired for the final model.
3. Determine the optimal one-node model. Every combination of one node and one collocation point is used to solve for the coefficient that, when substituted into the one-node model with its corresponding node, minimizes Root Mean Squared Error (RMSE).

$$RMSE = \sqrt{\frac{\sum_{j=1}^n (\text{approximate}_j - \text{actual}_j)^2}{n}},$$

where n is the number of evaluation points that will be used to measure RMSE.

4. Remove the chosen node and collocation point from the initial pool of candidate locations, and fix those locations in the model.
5. Determine the optimal n -node model. Similar to Step 3, for each successive iteration choose the next node and collocation point pair that minimizes RMSE.

Basis Function Definition

The general basis function for the example problem is the inverse source function,

$$\frac{1}{R_j},$$

where R_j is the distance from a given node (x_j, y_j, z_j) , given by

$$\sqrt{(x - x_j)^2 + (y - y_j)^2 + (z - z_j)^2},$$

and the resulting approximation function will again be the sum of these basis functions combined with the coefficient c_j ,

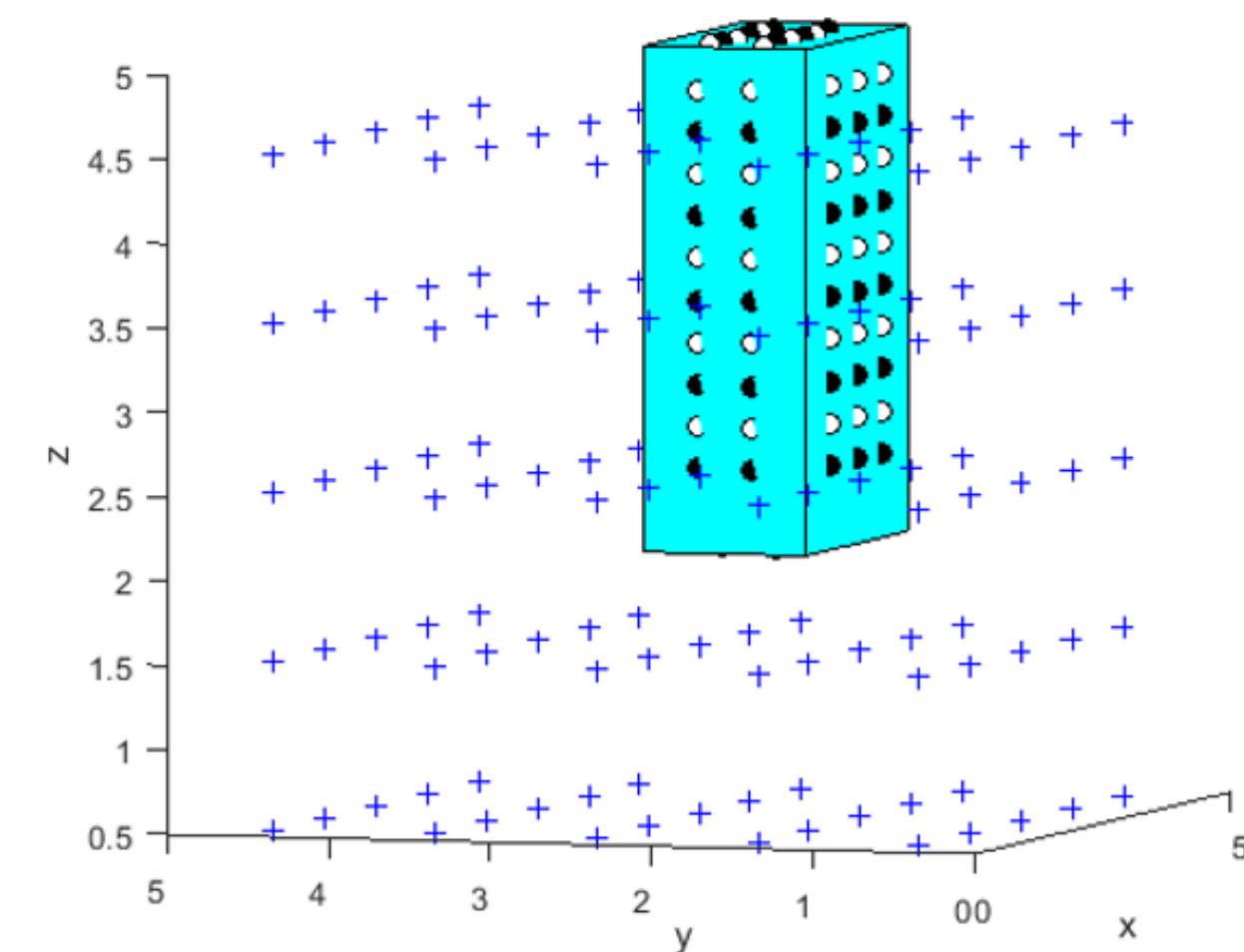
$$\hat{w}(x, y, z) = \sum_{j=1}^n \frac{c_j}{R_j}.$$

Example Problem

The example problem is to solve the 3D Laplace’s equation for the boundary conditions of

$$\frac{1}{R_j} = \frac{1}{\sqrt{x^2 + y^2 + z^2}}.$$

Although the boundary conditions are in the same form as the basis functions, this problem provides convincing evidence for the efficacy of the optimization algorithm. Below is the problem domain.



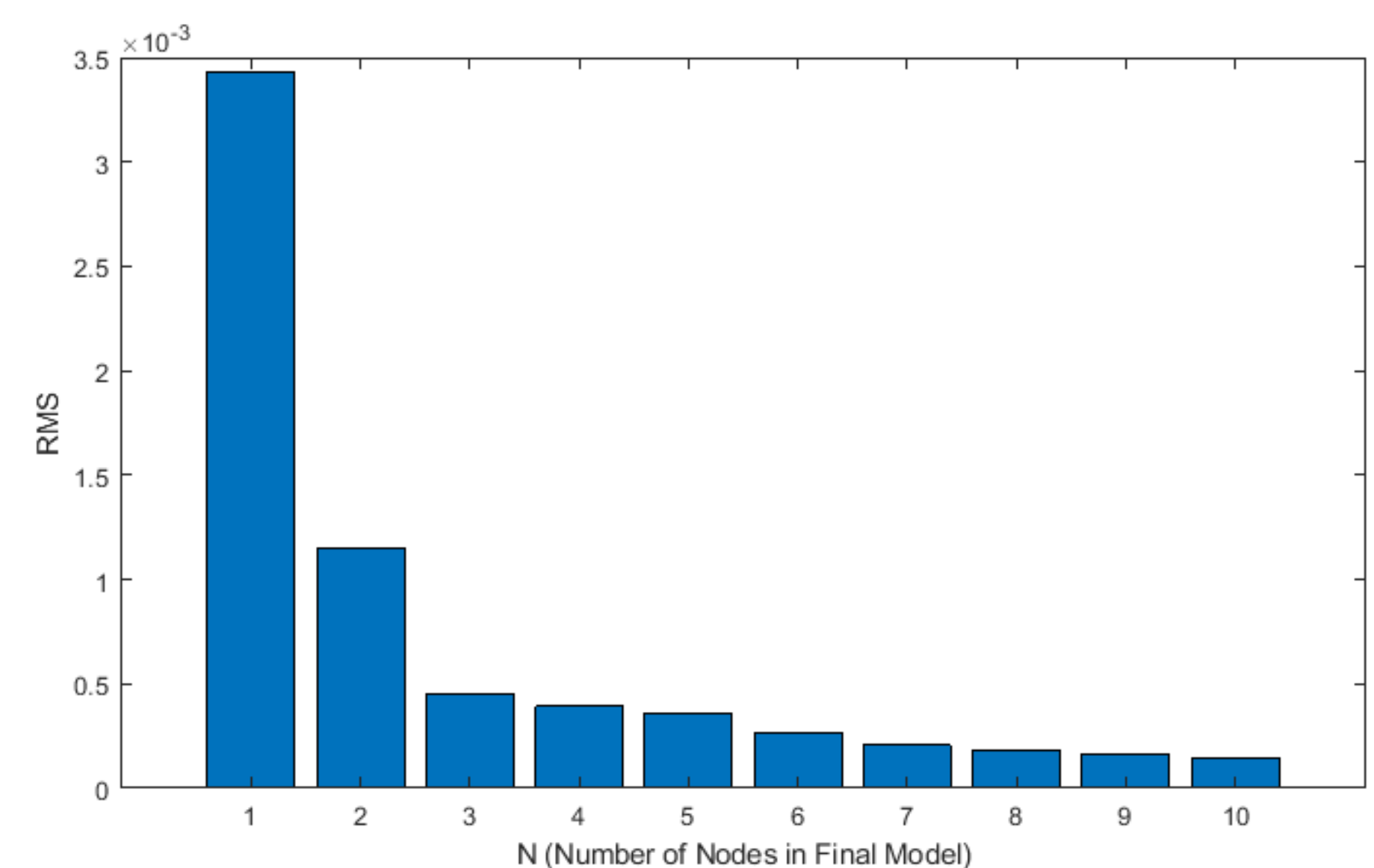
The white dots are the evaluation points, the black dots are the collocation points, and the blue plus signs are the nodal locations. In total there are 62 evaluation points, 62 collocation points, and 119 nodal locations. The domain has a length of 2, a height of 1, and a width of 3.

Problem Solution

Using the algorithm, a ten-node model was created. Tables 1, 2, and 3 report the chosen collocation points, chosen node locations, and resulting model coefficients in the order chosen and calculated, respectively.

Collocation Point Locations				Node Locations				Model Coefficients
Order	x	y	z	Order	x	y	z	
1	4.5	2	2.5	1	0.5	0.5	0.5	1.4995
2	3.5	2.333	2	2	1.5	1.5	1.5	-0.7537
3	4	3	3	3	3.5	0.5	1.5	-0.0151
4	3.5	3	2.5	4	4.5	3.5	4.5	2.3058e-04
5	4.5	3	3.5	5	2.5	2.5	2.5	-7.8448e-04
6	3	2.667	2.5	6	2.5	1.5	1.5	0.1588
7	4	2.333	2	7	3.5	1.5	1.5	-0.0090
8	3.5	3	3.5	8	1.5	2.5	2.5	0.1607
9	4	2	4.5	9	1.5	3.5	3.5	-0.0324
10	3.5	2	2.5	10	3.5	2.5	1.5	-0.0021

As expected, once substituting these coefficients into the approximation function using the basis functions created using the chosen node locations, the result is not only a highly accurate model, but also one that improves with each iteration. The table below shows the RMSE values for each model from one node to ten nodes.



Results and Conclusions

Figure 2 shows that increasing the number of nodes in the model improves RMSE much more drastically at first, but it always improves error. In the future, researchers can use these results to determine the number of nodes that will create an RMSE within a certain magnitude. Lowering error with fewer nodes ultimately has the significant benefit of avoiding sources of error from reaching the limits of matrix solvability with large matrices. Nevertheless, there is much more room to continue improving the efficiency and accuracy of MFS approximation functions.