An Adaptive, Black-Box Model Order Reduction Algorithm Using Radial Basis Functions

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Matthew Stephanson, B.S., M.S.

Graduate Program in Electrical & Computer Engineering

The Ohio State University

2012

Dissertation Committee:

Prof. Jin-Fa Lee, Advisor
Prof. Fernando Teixeira
Prof. Ed Overman
Abstract

This project investigates reduced order modeling, as applied to electromagnetics problems. Given a linear system describing a problem and depending on one or more parameters (such as frequency or material properties), model order reduction seeks to find a small set of global basis functions that can accurately approximate the solution across the given range of parameter values. By solving the system in terms of this smaller basis, it is possible to recast the original matrix equation into one with much lower dimensionality; this smaller system can then be solved much more rapidly. Many algorithms, and refinements thereof, have been developed to accomplish this goal. Most are limited by the fact that they explicitly or implicitly assume that it is exactly known how the linear system of interest depends on the parameters. Some, such as those based on a Padé approximation, as also limited to the case of a single parameter.

The goal of this work is to adapt the model order reduction framework so that it is suitable for use with a general solution technique. That is, as long as the problem of interest can be represented as a continuously varying linear system, the algorithm should be able to perform order reduction, without having to know the detailed dependence on the parameters. To accomplish this, the process of generating a reduced order model is described in terms of simple linear algebra operations between matrices and vectors, so that it is applicable to a wide variety of problem formulations. Having
done this, the problem then becomes how to approximate the parameter dependence, since the exact behavior is unknown. This is where radial basis function interpolation comes in: it allows the reduced order system to be efficiently approximated even in the case of a multidimensional parameter space with scattered interpolation points. Furthermore, in contrast to polynomial interpolation, for example, whose basis dimension increases rapidly with order, it is possible to add radial basis function one-by-one, which allows for efficient adaptive interpolation. To this end, several combinations of sampling methods and error estimators are described and evaluated. After the most likely candidates are identified, several numerical examples are presented showing that efficient, accurate reduced order models can be generated, by the same program, for a variety problems, solver formulations, and parameter dependencies.
Acknowledgments

My thanks go foremost to my advisor, Professor Jin-Fa Lee. He introduced me to computational electromagnetics my sophomore year and has been a continuing inspiration since then. Without his patience, guidance, and support, this work could not have been accomplished.

Through the years, I have had the pleasure of working with the fellow members of the Computational Science Group. I thank Dr. Peng Zhen, Dr. Shao Yang, Dr. Petros Allilomes, Dr. Davood Ansari, Vineet Rawat, Stylianos Dosopoulus, Xiaochuan Wang, Jiangong Wei, Jue Wang, Josh Mahaffy, and Caleb Waltz for their many engaging collaborations and discussions.

I am grateful to the following for their financial support during my dissertation work: The Ohio State University (Presidential Fellowship), the Northrop Grumman Corporation (Northrop Grumman Fellowship), and the U.S. Department of Energy (contract W-7405-ENG-48). In conjunction with the latter, it was my pleasure to spend three summers at Lawrence Livermore National Laboratory, under the mentorship of Dr. Dan White. I would also like to thank Ben Fasenfest, Mark Stowell, and Dr. Kyle Lange, also of the LLNL Computational Electromagnetics Group, for their guidance and assistance.
Finally, I thank my amazing family. My parents and brother, whose love, guidance, and support have made me the person I am today. And, at the end, my wonderful wife Yuanhong, who showed me the warm sunshine I had missed.
Vita

April 28, 1984 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . Born — Pickerington, Ohio, U.S.A

March 2007 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . B.S. Electrical Engineering, The Ohio State University

December 2011 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . M.S. Electrical Engineering, The Ohio State University

November 2007 – June 2012 . . . . . . . . . . . . . . . . . . Graduate Research Associate & University Fellow, The ElectroScience Lab


Publications

Research Publications


### Fields of Study

Major Field: Electrical & Computer Engineering

Studies in:

- Computational Electromagnetics
  - Prof. Jin-Fa Lee
- Electromagnetic Theory
  - Prof. Fernando Teixeira
- High-Performance Computing
  - Prof. P. Sadayappan
3.1.3 Difference Between Two Reduced Order Models ........................................ 28
3.2 Choosing the Sample Point ........................................................................... 32
   3.2.1 Fixed Grid Selection ................................................................................. 32
   3.2.2 Optimization-Based Selection .................................................................. 35
3.3 Solver Interface ............................................................................................... 40
3.4 Parallelization .................................................................................................. 44

4. Applications of Black-Box MOR Algorithm .................................................... 47
   4.1 Oscillator Circuit in Enclosure ..................................................................... 47
      4.1.1 Problem Statement ................................................................................ 47
      4.1.2 Formulation ............................................................................................ 48
      4.1.3 Numerical Results .................................................................................. 50
   4.2 Patch Antenna ............................................................................................... 52
      4.2.1 Problem Statement ................................................................................ 52
      4.2.2 Formulation ............................................................................................ 56
      4.2.3 Numerical Results .................................................................................. 58
   4.3 Radome with Frequency Selective Surface .................................................. 66
      4.3.1 Problem Statement ................................................................................ 66
      4.3.2 Design Process and Solver for FSS Radome ........................................ 68
      4.3.3 Numerical Results .................................................................................. 71

5. Conclusion .......................................................................................................... 83

Bibliography ........................................................................................................... 86
List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Computational results for circuit board example</td>
</tr>
<tr>
<td>4.2</td>
<td>Parameters of interest for patch antenna example</td>
</tr>
<tr>
<td>4.3</td>
<td>Computational statistics for patch antenna example</td>
</tr>
<tr>
<td>4.4</td>
<td>Computational statistics for FSS</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Singular values of solution matrix for scattering from a 1 m PEC sphere, 50–300 MHz</td>
<td>2</td>
</tr>
<tr>
<td>2.1</td>
<td>Convergence of polyharmonic spline approximation of a test function interpolated on a uniform grid</td>
<td>25</td>
</tr>
<tr>
<td>3.1</td>
<td>Comparison of true and estimated error with least-squares comparison, ( B = V^T A^T(s) A(s) V )</td>
<td>31</td>
</tr>
<tr>
<td>3.2</td>
<td>Initial layout of sample points (●) and test points (⊗)</td>
<td>32</td>
</tr>
<tr>
<td>3.3</td>
<td>Distribution of sample and test points after adding samples, and tree structure relating test and sample points</td>
<td>33</td>
</tr>
<tr>
<td>3.4</td>
<td>Pseudocode for hierarchical sampling MOR algorithm</td>
<td>34</td>
</tr>
<tr>
<td>3.5</td>
<td>Patch antenna ( S_{11} ) error with hierarchical sampling</td>
<td>35</td>
</tr>
<tr>
<td>3.6</td>
<td>Pseudocode for optimized sampling MOR algorithm</td>
<td>38</td>
</tr>
<tr>
<td>3.7</td>
<td>Code listing of sample <code>solve.sh</code> file</td>
<td>41</td>
</tr>
<tr>
<td>3.8</td>
<td>Code listing of <code>mor.py</code>, containing common functions for user scripts</td>
<td>43</td>
</tr>
<tr>
<td>4.1</td>
<td>Geometry of oscillator circuit and enclosure</td>
<td>48</td>
</tr>
<tr>
<td>4.2</td>
<td>(a) Induced current on enclosure (b) top surface of enclosure removed, showing circuit board</td>
<td>51</td>
</tr>
<tr>
<td>4.3</td>
<td>Adaptive sampling grid of oscillator example</td>
<td>52</td>
</tr>
</tbody>
</table>
4.20  y-polarization reduced order model: (a) estimated and (b) actual error after 11 samples, (c) estimated and (d) actual error after 12 samples, (e) estimated and (f) actual error after 13 samples.  

76

4.21  y-polarization reduced order model: (a) estimated error and (b) condition number of $\hat{B}$ with 13 samples, (c) estimated error and (d) condition number of $\hat{B}$ with 12 samples (note change in horizontal axis).  

79

4.22  Frequency sweep of periodic boundary condition FEBI code for several incident angles, compared to HFSS results.  

80

4.23  Geometry and mesh of scaled unit cell with dielectric layer.  

81

4.24  Performance of dielectric-backed FSS. (a) peak transmission versus dielectric thickness (b) angular dependence of two-layer dielectric with thickness 0.7 mm.  

82
Chapter 1: Introduction

From a mathematical viewpoint, model order reduction (MOR) is a process for reducing the order (i.e., number of degrees of freedom) in a mathematical model of a system. Of particular interest here is the case where the model consists of $N$ linear equations for $N$ unknowns, in the form a matrix equation

\[ A(s)x = b(s). \]  

(1.1)

In most cases, the matrix $A$ or the right-hand side $b$ (or both) depend on some parameters $s$, and we are interested in solving (1.1) not once but many times, for many different values of the parameters. For example, the matrix equation could represent the SPICE model of a circuit, and we may be interested in calculating the behavior for many different values of resistance, capacitance, or inductance at many different frequencies. Or, (1.1) could be a finite element model of an antenna, and we may be interested in how the variance in material properties affect the final performance.

Despite the variety of applications, the key insight in model order reduction methods is to realize that in most cases, the model is capable of resolving more physical phenomena than we are actually interested in. For example, we may have an aircraft model that is capable of resolving the radar-scattering behavior from any incident wave direction, at frequencies from 0 to 8 GHz. If, however, we are only interested
in incident directions with $\theta = \pi/2$, $\phi \in [0, 2\pi]$, and frequencies from 4 to 8 GHz, then in a sense our model has too much detail; the solution vector $x$ is capable of representing solutions below 4 GHz, even though we are not interested in them.

As a concrete example, consider the problem of calculating the scattered field from a PEC sphere with 1 m radius and a plane wave incidence. Meshing the surface of the sphere with 1,520 triangle and applying the electric field integral equation results in a matrix equation with $N = 2,280$ unknowns. If this problem is solved over the frequency range 50–300 MHz in 1 MHz increments and the solution vectors are grouped together, the resulting $2,280 \times 251$ matrix has the singular value distribution shown in Figure 1.1. It can be seen that if we are willing to settle for an answer that is 0.001% accurate, relative to the solution of the full $N \times N$ matrix equation, then only 28 basis vectors are needed. The other 223 solutions can be adequately represented as
a linear combination of these vectors. The goal of model order reduction is to, either implicitly or explicitly, take advantage of this linear dependence to reduce the original \( N \times N \) matrix equation into a much smaller one (ideally \( 28 \times 28 \) in this example) that can be solved much more rapidly.

When applying model order reduction techniques to matrices and right-hand sides with known (often polynomial) parameter dependence, the dominant technique is some form of asymptotic waveform evaluation (AWE). These methods are based on forcing the derivatives of the reduced- and full-order solutions to match, either implicitly or explicitly.

The idea of finding a compact representation of a system goes back at least as far as the 1965 work of Kalman [3]. This work was not concerned directly with approximation, but rather with the problem of, given a matrix of transfer functions between the inputs and outputs, finding a state-space representation with minimal order. The related concepts of controllability and observability have analogies to the above examples. When the incident angle and frequency are restricted, there are certain states that the system can no longer be brought to, for example the state representing scattering from a 3 GHz wave; these states are uncontrollable and reduce the effective order of the system. Similar observation were made by Moore [4], who, building on Kalman’s irreducible representations, was among the first to use the SVD to generate reduced order approximations of linear, time-invariant systems.

Later, driven by the need to efficiently simulate high-speed circuits and microwave devices, the development of MOR techniques accelerated rapidly. Asymptotic waveform evaluation (AWE) [5] is among the first to explicitly recognize the utility of moment-matching methods, in which the \( s \)-domain transfer function is approximated...
by finding a simpler function whose derivatives match those of the full system up to a particular order. Early methods calculated these derivatives explicitly, and it was soon pointed out that this limits their stability due to, among other things, the similarity to inverse power methods and the attendant difficulty of finding poles other than the smallest ones. To combat these difficulties, methods were soon developed that made use of stable Krylov subspace methods, such as the Lanczos method [6–8]. However, due to this method’s relationship to linear eigenvalue problems, they are limited to systems with linear dependence on a single parameter. This is in contrast to generic AWE which, requiring only the calculation moments, allows for more general parameter dependencies.

1.1 Review of Model Order Reduction Methods

1.1.1 Asymptotic Waveform Evaluation

The simplest form of AWE begins by explicitly calculating the derivatives (also called moments) of the solution. In the case of a single parameter \( p \), for example,

\[
\begin{align*}
\mathbf{x} &= \mathbf{x}^{(0)} = A^{-1}\mathbf{b} \\
\mathbf{x}' &= \mathbf{x}^{(1)} = A^{-1}(\mathbf{b}' - A'\mathbf{x}) \\
\mathbf{x}''/2! &= \mathbf{x}^{(2)} = A^{-1}(\mathbf{b}'' - 2A'\mathbf{x}' - A''\mathbf{x}) \\
&\vdots \\
\frac{d^n\mathbf{x}}{dp^n} \left/ n! \right. &= \mathbf{x}^{(n)} = A^{-1} \left[ \frac{d^n\mathbf{b}}{dp^n} - \sum_{k=1}^{n} \binom{n}{k} \frac{d^kA}{dp^k} \frac{d^{(n-k)}\mathbf{x}}{dp^{(n-k)}} \right] 
\end{align*}
\]

where all expressions are evaluated at an expansion point \( p = p_0 \). Once the first \( 2Q \) orders are determined, the resulting power series for each element of the solution vector

\[
x_i(p) \approx \sum_{n=0}^{2Q-1} x_i^{(n)} (p - p_0)^n
\]

(1.3)
can be converted into a Padé approximation

\[ x_i(p) \approx \frac{\sum_{n=0}^{Q-1} a_n (p-p_0)^n}{1 + \sum_{m=1}^{Q} b_n (p-p_0)^m} \]  

(1.4)

by forcing the derivatives of (1.3) and (1.4) to match up to order \(2Q-1\). Equivalently, one can consider the expression

\[
\left( \sum_{n=0}^{2Q-1} \frac{x_i^{(n)}}{n!} (p-p_0)^n \right) \left( 1 + \sum_{m=1}^{Q} b_n (p-p_0)^m \right) - \sum_{n=0}^{Q-1} a_n (p-p_0)^n
\]  

(1.5)

and require that the coefficients be zero for all orders from zero to \(2Q-1\). This leads to the systems

\[
\begin{bmatrix}
    x_i^{(Q-1)} & x_i^{(Q-1)} & \cdots & x_i^{(0)} \\
    x_i^{(Q)} & x_i^{(Q-1)} & \cdots & x_i^{(1)} \\
    \vdots & \vdots & \ddots & \vdots \\
    x_i^{(2Q-2)} & x_i^{(2Q-3)} & \cdots & x_i^{(Q-1)}
\end{bmatrix}
\begin{bmatrix}
    b_1 \\
    b_2 \\
    \vdots \\
    b_Q
\end{bmatrix}
= \begin{bmatrix}
    x_i^{(Q)} \\
    x_i^{(Q+1)} \\
    \vdots \\
    x_i^{(2Q-1)}
\end{bmatrix}
\]  

(1.6)

and

\[
a_0 = x_i^{(0)} \\
a_1 = x_i^{(1)} + x_i^{(0)}b_1 \\
a_2 = x_i^{(2)} + x_i^{(1)}b_1 + x_i^{(0)}b_2 \\
\vdots \\
a_{Q-1} = x_i^{(Q-1)} + \sum_{k=1}^{Q-1} x_i^{(Q-1-k)}b_k.
\]  

(1.7)

Once the Padé coefficients have been calculated for each entry of the solution (or a linear combination if, for example, the desired quantity is a scattering parameter), an approximate solution can rapidly be calculated for any given value of the parameter.

It should be noted that although AWE is often applied to finite element or Laplace-transformed problems, in which cases the needed derivatives are straightforward to calculate, there has also been work in applying it to integral equations methods
In these cases, the parameter dependence is much more complex, and detailed calculations are needed for each different type of parameter (e.g., frequency, material properties, and so forth).

1.1.2 Galerkin AWE

Galerkin AWE \cite{10,11} takes a different approach. Here it is assumed that a set of linearly-independent trial functions \( \{v_1, v_2, \ldots, v_m\} \) is given. By assembling these vectors into an \( N \times m \) matrix \( V = [v_1, \ldots, v_m] \), the solution can be approximated as

\[
x \approx V \tilde{x}(p),
\]

where \( \tilde{x} \) is the reduced order solution. It is determined by requiring the residual

\[
r = AV \tilde{x} - b
\]

(1.9)

to be weakly equal to zero. In particular, given a set of \( N \)-vector testing functions \( W = [w_1, w_2, \ldots, w_m] \), it is required that

\[
W^T AV \tilde{x} = W^T b
\]

(1.10)

for all parameter values. In this way, for any given parameter value \( \tilde{x}(p) \) can be calculating by solving a linear system of equations with the \( m \times m \) matrix \( W^T AV \).

As for \( W \), the so-called Galerkin method is to choose the trial and testing spaces to be the same, \( W = V \), giving Galerkin AWE (GAWE) its name.

Despite their obvious differences, there are a few similarities between AWE and GAWE. First, according to Cramer’s rule \cite{12}, the \( i \)-th entry of the reduced order solutions is

\[
\tilde{x}_i = \frac{\det(V^T AV)_i}{\det V^T AV},
\]

(1.11)
where \((V^TAV)\) is the matrix formed by replacing the \(i\)-th column of \(V^TAV\) with \(V^Tb\). Now, if the matrix is a polynomial function of the parameters, then both of these determinants will also be polynomials of the parameters. For example, this is often the case when the finite element method is applied to frequency-domain problems with mesh truncation by absorbing boundary conditions or perfectly matched layers (though it is not universally true, such as when the problem contains ferromagnetic materials or plasma). The primary difference in this respect between AWE and its Galerkin counterpart is the order of the approximation function. In AWE, matching \(2Q\) moments requires a \((Q - 1, Q)\) order rational function. On the other hand, for Galerkin AWE to match the same order requires a \(2Q\)-dimensional reduced order basis, so the matrix \(V^TAV\) is \(2Q \times 2Q\). Assuming that the entries of the system matrix \(A\) are up to quadratic functions of the parameter implies that the determinant of the reduced order matrix is a polynomial with order as high as \(4Q\).

Also, it is possible to choose \(W\) such that the GAWE solution matches moments with the true solution to up a certain order at the expansion point. In particular, suppose that the trial functions consists of the moment vectors from (1.2), that is, \(V = [x^{(0)}, x^{(1)}, \ldots, x^{(q)}]\). (1.12)

It is unsurprising that the GAWE solution matches the value of the true solution at the expansion point when the trial space contains the solution vector:

\[
\tilde{x}(p_0) = (V^T A(p_0)V)^{-1}V^T b(p_0)
\]

\[
= (V^T A(p_0)V)^{-1}V^T A(p_0)x(p_0)
\]

\[
= (V^T A(p_0)V)^{-1}V^T A(p_0)Ve_1
\]

\[
= e_1
\]

\[
V\tilde{x}(p_0) = Ve_1 = x(p_0),
\]

7
where \( \mathbf{e}_i \) is a vector that is all zero except for a one in the \( i \)-th component. More surprising is that it is also easy to prove that the first moment is matched.

\[
\tilde{x}' = [(V^T AV)^{-1} V^T b]' \\
= (V^T AV)^{-1} V^T b' - (V^T AV)^{-1}(V^T AV)'(V^T AV)^{-1} V^T b \\
= (V^T AV)^{-1} V^T [b' - A' \tilde{x}] \\
= (V^T AV)^{-1} V^T [b' - A' \tilde{x}] \\
= (V^T AV)^{-1} V^T Ax' \\
= (V^T AV)^{-1} V^T A \mathbf{e}_2 \\
(V \tilde{x})' = \mathbf{x}'.
\]

(1.14)

In fact, it has been shown that the choice of \( V \) in (1.12) results in the the derivatives of the residual vector up to order \( q - 1 \) being zero at the expansion point, which is equivalent to the derivatives of the GAWE solution matching their true values up to the same order [13].

1.1.3 Automated Multipoint GAWE

The multipoint generalization of GAWE is to require

\[
\frac{d^i r}{ds^i} \bigg|_{s=s_1} = 0 \quad \text{for } i = 0, \ldots, q_1 - 1, \quad \text{(1.15)}
\]

\[
\frac{d^i r}{ds^i} \bigg|_{s=s_2} = 0 \quad \text{for } i = 0, \ldots, q_2 - 1, \quad \text{(1.16)}
\]

\[
\vdots \quad \vdots
\]

\[
\frac{d^i r}{ds^i} \bigg|_{s=s_3} = 0 \quad \text{for } i = 0, \ldots, q_3 - 1. \quad \text{(1.18)}
\]

It turns out [10] that this is easily accomplished. It is necessary only to choose

\[
V = [V_1 \quad V_2 \quad \ldots \quad V_M],
\]

(1.19)
where $V_i$ is the $N \times q_i$ reduced order basis matrix that would be generated by the single-point GAWE algorithm matching moments up to order $q_i$ at expansion point $s_i$. Beyond the proof of this fact, one of the interesting contribution is the automated method of choosing both the expansion points and the moment-matching order [1]. When a new basis vector $v_i$ is generated at the $j$-th expansion point, but before it is orthogonalized to the previous basis vectors, the angle between the vector and the space spanned by those previous vectors could be computed. However, practically it is more efficient to compare $y_i = Av_i$ to the space $AV_j$, because many of the necessary computations are already needed for the GAWE process. The comparison is accomplished via the oblique projection

$$y^\parallel = AV_j(V_j^TAV_j)^{-1}V_j^Ty_i,$$

which can then be used to find the cosine of the angle between the current basis and the new vector:

$$\cos \vartheta = \frac{\|y_i - y^\parallel\|}{\|y_i\|}$$

When the angle $\vartheta$ is less than some use-defined tolerance $\tau_1$, it is assumed that the iterative moment-matching process has extracted the most important derivatives, and a new expansion point is chosen, if necessary. The choice of the next expansion point is made by examining the relative residual of the MGAWE solution at each of the desired frequencies. If there are any intervals where the residual is greater than another user tolerance $\tau_2$, then a new expansion point is added in the middle of the unconverged interval, except that intervals containing an endpoint of the frequency range of interest are biased toward that endpoint.
1.1.4 Impedance Matrix Interpolation

Another method that bears mentioning is matrix entry interpolation [14]. Although not a model order reduction method, it has been used in the context of fast frequency sweeps for PEC objects analyzed with the electric field integral equation (EFIE). In particular, in certain situations the amount of time needed to compute the entries of the matrix that is to be solved (often called the impedance matrix in the engineering community) may be considerably longer than the time needed to solve the matrix equation, either via LU decomposition or an iterative method. In these cases, it can be beneficial to assemble the matrix for a few frequencies and then interpolate the entries, allowing much faster evaluation at other frequencies. However, with the development of fast matrix-vector multiplication algorithms for integral equations, such as the fast multipole method [15, 16] and IE-FFT [17], the applications of integral equation methods has tended to focus on extremely large problems, for which it is impossible to store and interpolate the entire matrix. Nonetheless, all of these methods must calculate the near-neighbor interactions (i.e., the matrix elements $Z_{ij}$ must still be computed and stored when testing function $i$ and trial function $j$ are close to each other), and there has been recent work on using matrix interpolation to efficiently evaluated this so-called near-field matrix.

Given that, as stated above, matrix interpolation is not a model order reduction method, its relationship to the present work appears unclear at first. The connection is that both methods are concerned with rapidly approximating a matrix that is too costly to calculate exactly. In the case of impedance matrix interpolation, this is the large, dense $Z$ matrix, each of whose entries requires the computation of two surface
integrals with a potentially singular Green function. In this work, it is the calculation of $V^T A(s)V$ for many values of $s$, when the exact dependence of $A$ on $s$ is unknown.

### 1.2 Assessment and Plan

Considered together, it is seen that each of the previously-described MOR methods has some deficiency that limits its range of applications, often by assuming a single parameter of interest and/or that the system matrix’ parameter dependence is explicitly known. To widen the scope of applications, three goals are proposed for a general model order reduction framework:

1. Ability to handle multiple parameters.

2. Adaptivity—the user should not be required to, e.g., select how many moments to match. Rather, the method should be guided by the user’s desired accuracy of the ROM. To the maximum extent possible, the method should be autonomous, freeing the user for more productive tasks than supervising the program’s progress.

3. Generality—the program should not rely on preconceived notions of how the problem is formulated or solved, or what parameters the user may be interested. Certain situations, such as frequency or angle sweeps, may be common enough to justify optimized variants. But increasing computer system performance can be used to make programs run faster; it can never add a feature that isn’t already present.

The goal of generality, as seen from the description above, encompasses many aspects of the model order reduction algorithm’s design and implementation. To
operationalize this idea, the first major design choice is made: the MOR program should only rely on common, generic linear algebra operations. The particular, the program conceptualizes the solver program as, at its core, a procedure for generating and solving the linear system \( A(s)x = b(s) \). Therefore, it is assumed only that the solver is capable of

1. calculating the solver’s own right-hand side \( b(s) \),

2. solving the linear system with its own or a given right-hand side, and

3. performing matrix-vector multiplications with given vectors.

This design choice has many implications. Foremost, it means that higher-order moment-matching approaches are unavailable. Since any knowledge of the linear system’s derivatives is disclaimed, the best that can be achieved is the match the value of the solution at several points in parameter space.

Another aspect of generality is the types of variables that may be used as parameters. One solution to the goal of keeping the types of parameters as general as possible is to decide that the MOR program itself should know nothing at all about the parameters’ meaning. Instead, it works only with normalized values between zero and one. The conversion between these values and the physical values used by the solver is performed by user-created scripts.

The desire for adaptivity has two goals, though they somewhat overlap: user-friendliness and efficiency. The first part of this is freeing the user from having to choose the ROM expansion points beforehand, hoping that this choice will be sufficient to meet the desired accuracy and capture any interesting behavior, such as resonances. Thus, the user should only be required to specify the desired level of
accuracy, leaving the algorithm to automatically and adaptively determine the expansion points. This process, in turn, promotes efficiency by not expending resources to achieve hyperaccurate solutions that the user does not require.

Finally, the ability to handle multiple parameters constrains the choices available for function approximation (for, if the exact parameter dependence is unknown, some type of approximation of the reduced order system $V^TA(s)V$ will be necessary). For example, polynomial interpolation and Fourier harmonics may not be as useful in the multivariable, scattered data case as they are in one dimension.

In chapter 2, the chosen approximation strategy, radial basis function interpolation, is described and surveyed. In particular, we look at the issues of convergence and stability, and develop mitigation strategies based on $hp$ refinement and regularization. Chapter 3 presents a detailed description of the MOR algorithm itself, as well as supporting methods such as error estimation, optimization, and parallelization. The black-box interface between the MOR program and the solver is also described. Finally, Chapter 4 examines the performance of the method as applied to several different problems and solvers: a printed circuit board (boundary integral equation), patch antenna (finite element method), and frequency selective surface (hybrid finite element/boundary integral).
Chapter 2: Radial Basis Functions

As discussed previously, an interpolation of the reduced order matrix is needed too efficiently evaluate the reduced order model. This constitutes a multi-dimensional, scattered data interpolation problem, for which radial basis functions provide an excellent solution. Therefore, this chapter gives an overview of its characteristics.

2.1 Introduction

Radial basis function (RBF) interpolation, as the latter part of the name suggests, uses a set of basis functions to interpolate scattered data as a function of one or more parameters. The essence of the method is that the basis functions depend only on the distance between the evaluation point and a fixed center point that is unique to each basis function. That is, the interpolant has the form

\[ \hat{f}(s) = \sum_{i=1}^{M} c_i \Phi(\|s - s_i\|), \]  

(2.1)

where \( s \in \mathbb{R}^n \) is the evaluation point in \( n \)-dimensional space, \( c_i \) are the interpolation coefficients, \( \{s_1, \ldots, s_M\} \subset \mathbb{R}^n \) are the set of center points, and \( \Phi \) is the RBF.

The use of radial basis functions can be traced back at least as far as the 1971 paper by Hardy [18], who used them to interpolate surveying data in order to create contour maps. This application presented a two-variable, scattered data interpolation problem, since the surveyors tended to mostly measure geographically prominent
locations, such as the peaks of hills and bottoms of valleys. Comparing several different methods, Hardy found that polynomial interpolants tended to be too smooth, while a Fourier series was too oscillatory. He found that radial basis functions gave a significantly better result, in part because his chosen basis function (the multi-quadric, described below) contains an adjustable parameter that allows one to alter its smoothness.

After this initial numerical investigation, the theory of RBF interpolation began to develop. Micchelli [19] closed a crucial gap by proving that the collocation matrix for multi-quadric basis functions, used to calculate the coefficients $c_i$, is nonsingular for any non-degenerate distribution of center points. Other researchers developed results for convergence of the interpolant, such as Powell [20], Johnson [21], and Wu & Schaback [22].

The interpolation coefficients may be found by solving the matrix equation

$$Ac = f,$$  \hspace{1cm} (2.2)

where $f_i = f(s_i)$, and the collocation matrix is $A_{ij} = \Phi(s_i - s_j)$. As one might expect, many choices for $\Phi$ have the property that they are maximum when the distance is zero and decay to zero as the distance increases. Examples include the inverse multi-quadric $\Phi(r) = 1/\sqrt{r^2 + c^2}$ and the Gaussian $\Phi(r) = \exp(-r^2/c^2)$, where $c$ is a parameter that controls the shape of the RBF. These functions are called positive definite radial basis functions, because the resulting interpolation matrix $A$ is positive definite for any set of distinct center points. On the other hand, some radial basis functions have the property of only being conditionally positive definite of degree $k$ (to be explained below). In this case, the standard RBF approximation in (2.1) is augmented by polynomial $p(s) \in P_n^{k-1}$, the space of polynomials of $n$ variables.
complete to order $k - 1$. Then the approximation becomes

$$\hat{f}(s) = \sum_{i=1}^{M} c_i \Phi(\|s - s_i\|) + p(s).$$

(2.3)

The extra degrees of freedom are handled by adding the side conditions that the RBF coefficient vector $c$ be orthogonal to the polynomial space. Specifically,

$$\sum_i c_i \cdot q(s_i) = 0 \quad \forall q \in P_{k-1}^n. \quad (2.4)$$

The radial basis function is conditionally positive definite if $A$ is positive definite for all coefficient vectors $c$ satisfying this property. In practice, we have a basis \{\(p_1, \ldots, p_{2^n}\)\} for $P_{n}^{k-1}$, and (2.4) is enforced by solving the linear system

$$\begin{pmatrix} A & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \quad (2.5)$$

where $P_{ij} = p_j(s_i)$ and $d$ is the vector of polynomial coefficients, so that $p(s) = \sum_i d_i p_i(s)$. In addition to the centers being unique, uniqueness of the interpolation solution also requires that the set of centers contain a $P_{n}^{k-1}$-resolvant subset. Examples of conditionally positive definite RBFs are the multiquadric $\Phi(r) = \sqrt{r^2 + c^2}$, of order $k = 1$, and the polyharmonic splines,

$$\Phi(r) = \begin{cases} r^{2k-n} \log r & \text{if } 2k - n \text{ is an even integer} \\ r^{2k-n} & \text{otherwise}, \end{cases} \quad (2.6)$$

of order $k$.

Consideration of the case $k = 2$ explains the origin of the polyharmonic splines. In this special case, the radial basis function is $r^3$ in one dimension and $r^2 \log r$ in two dimensions (in the first case, RBF interpolation is equivalent to cubit splines). These are the fundamental solutions of the biharmonic operator, which describes the physical phenomena of bending by thin rods and plates. Therefore, the interpolant
\( \hat{f} \) describes a thin rod or plate that is forced to pass through the given interpolation points. Mathematically, this is the solution that passes through the interpolation points and minimizes the bending energy

\[
E = \begin{cases} 
\int \left| \frac{\partial^2 \hat{f}}{\partial p_1^2} \right|^2 dp_1 & n = 1 \\
\int \int \left| \left( \frac{\partial^2}{\partial p_1^2} + 2 \frac{\partial^2}{\partial p_1 \partial p_2} + \frac{\partial^2}{\partial p_2^2} \right) \hat{f} \right|^2 dp_1 dp_2 & n = 2,
\end{cases}
\]

(2.7)
as shown in [23, 24]. In like manner, polyharmonic splines with \( k > 2 \) minimize seminorms that use higher order derivatives and correspond to fundamental solutions of

\[
\nabla^2 \circ \cdots \circ \nabla^2.
\]

(2.8)
in other words, higher-order analogues of the biharmonic operator.

### 2.2 Stability

One particular point of concern when using radial basis functions is the possibility of instability. This has been observed when the basis function becomes very flat, for example, when the parameter \( c \) in the multiquadric or Gaussian becomes large [25], or when the number of RBF centers becomes large (which is another way of saying that the RBFs are too flat). This is an interesting phenomena, which has attracted some attention among applied mathematicians, because theoretical convergence proofs suggest that in both of these limits, i.e., denser interpolation points and flatter RBFs, accuracy should increase. At first, it was discovered that part of the problem was numerical instability resulting from the condition number of the collocation matrix increasing, leading Schaback to propose an “uncertainty principle” for radial basis
functions: as the flatness of the RBF increased, it was observed that the interpolation error decreased, until the point where numerical difficulties in calculating the interpolation coefficients caused the error to increase \[26\]. Later methods of calculating the interpolant \[27, 28\], which did not rely on directly solving the collocation matrix, allowed the small-parameter regime to finally be explored. Here, it was discovered that accuracy is limited by a Runge-like phenomenon, which is not surprising since it had been shown that, in one-dimension, the RBF interpolant approaches the Lagrange polynomial as \( c \to \infty \).

Conscious of these limitations, we use two techniques to avoid instability: \( hp \) refinement and regularization. The first approach seeks to avoid instability by limiting the number of basis functions, making it less likely that the interpolation points will become too close together. In particular, our adaptive interpolation procedure begins by adding interpolation points, and therefore basis functions (\( p \) refinement), at locations where the error is estimated to be large. If this procedure does not converge before the number of basis functions reaches a prescribed maximum, then the parameter space is split in half along each dimension (\( h \) refinement), and a separate, independent interpolant is found for each subdomain. This procedure is continued recursively until a converged interpolant has been found for the entire domain.

The \( hp \) procedure described above, however, is not guaranteed to prevent a loss of linear independence in the basis functions. To guard against this situations, an SVD-based regularization is used \[29, 30\]. To do this, we must first examine how the linear independence of the basis functions might be measured. To begin with, we note that the polynomial space with which the RBFs must be augmented is entirely
independent of both the number and distribution of center points. Therefore, a suitably well-conditioned space for these functions, for example Chebyshev of Lagrange polynomials, can be chosen beforehand. Therefore, we focus on regularizing just the radial basis functions.

One possibility is to analyze the RBF point-evaluation matrix, $A$ in (2.5). However, simply looking to the eigenvalues of this matrix would be insufficient because, as discussed above, this matrix is not guaranteed to be positive-definite for classes of radial basis functions. Instead, if we choose to measure linear independence based on the point-evaluation inner product (i.e., $(f, g) = \sum_i f(s_i) \cdot g(s_i)$ for some set of points $\{s_1, s_2, \ldots \}$) then the matrix of interest is really $A^T A$. Of course, the square roots of the eigenvalues of $A^T A$ are simply the singular values of $A$, so it is sensible and convenient to use these to measure the linear independence of the basis functions. The method begins by calculating the singular value decomposition of $A$,

$$A = U \Sigma X^T,$$

(2.9)

where $U$ and $X$ are orthogonal matrices and $\Sigma = \text{diag}(\sigma_1, \sigma_2, \ldots)$ is a diagonal matrix whose entries are non-increasing, $\sigma_1 \geq \sigma_2 \geq \cdots$. Each column of $X$ can be interpreted as a set of RBF coefficients, with the corresponding column of $U \Sigma$ being the value of the resulting function at the center points. In this sense, because each column of both $U$ and $X$ is normalized, the singular value gives an indication of how close the corresponding function is to the zero function, as measured in the point-evaluation norm; smaller values indicate closeness to the zero function, which in turn implies linear dependence.
The idea of regularization, then, is to remove these functions from our set of basis functions, because they are close to zero and therefore do not contribute significantly to the interpolation. Another interpretation of this problem is that because these functions are so close to zero, it is difficult to accurately calculate their coefficients, which are sensitive to round-off errors. In effect, we would like to solve the least-squares problem using only the regularized basis functions. In practice, this is achieved by identifying the singular values that are less than some tolerance, setting them to zero, and using the Morse-Penrose pseudoinverse [12] to solve the regularized collocation equation. The details are as follows: after calculating the SVD and choosing a regularization tolerance $\epsilon$, the matrix $X$ is truncated to $\tilde{X}$ by delete the columns corresponding to singular values less than $\epsilon$. The collocation matrix, in terms of the regularized basis functions is then

\[
\begin{pmatrix}
A\tilde{X} & P \\
P^T\tilde{X} & 0
\end{pmatrix}
\begin{pmatrix}
\tilde{c} \\
d
\end{pmatrix} =
\begin{pmatrix}
f \\
0
\end{pmatrix},
\]

(2.10)

where $\tilde{c}$ is the vector of coefficients for the interpolating function in the new, regularized basis. Because this system is overdetermined, it is solved in the least-squares sense, using the GELSY Lapack function, which uses a complete orthogonal factorization.

### 2.3 Convergence

Due to the lack of similarity between radial basis functions and classical interpolation techniques such as polynomials and Fourier harmonics, it is perhaps unsurprising that convergence results for RBFs are not as well-developed.

In [22], a so-called local error estimate is given, relying on one of the more accessible RBF convergence proofs in the literature. Instead of depending of a global
mesh parameter $h$ to describe the density of RBF centers, it uses a parameter that depends on the density of centers in the neighborhood of each location,

$$h_{\rho}(s) = \max_{y \in B_{\rho}(s)} \min_{1 \leq i \leq M} \|y - s_i\|, \quad (2.11)$$

where $B_{\rho}(s)$ is the closed ball of radius $\rho$ centered at $s$. Then, there exists a positive integer $k$ such that the point-wise error in the $\mu$-th derivative ($\mu \in \mathbb{N}^n$ is a multi-index) can be bounded by

$$|\hat{f}^{(\mu)}(s) - f^{(\mu)}(s)| \leq c_f h_{\rho}^k |\mu| (2.12)$$

where $c_f$ is a constant depending on $f$ and the choice of radial basis function.

The proof begins by noting that, by solving (2.5) repeatedly for $f = (1, 0, \ldots, 0)$, $(0, 1, 0, \ldots, 0)$, etc., one can obtain a Lagrange-like representation

$$\hat{f}(s) = \sum_{i=1}^M u_i(s) f(s_i), \quad (2.13)$$

whose basis function have the property

$$u_i(s_j) = \delta_{ij}. \quad (2.14)$$

In this way, the error can be rewritten as

$$|\hat{f}(s) - f(s)|^2 = \left| \frac{1}{(2\pi)^n} \int_{\mathbb{R}^n} \left( \sum_{i=1}^M u_i^{(\mu)}(s)e^{js \cdot t} - (j t)^\mu e^{js \cdot t} \right) \tilde{f}(t) \, dt \right|^2, \quad (2.15)$$

where $\tilde{f}(t)$ is the Fourier transform of $f(s)$. Using the Cauchy-Schwartz inequality, this can be factored into a first part, which only depends on the distribution of RBF
centers, and a second part that depends only on the interpolated function:

\[
|\hat{f}(s) - f(s)|^2 \leq \left( \frac{1}{2\pi} \right)^n \int_{\mathbb{R}^n} \left( \sum_{i=1}^{M} u_i^{(\mu)}(s)e^{j\mu \cdot t} - (j t)^{\mu} e^{j\mu \cdot t} \right)^2 \hat{\Phi}(t) \frac{\tilde{f}(t)}{\tilde{\Phi}(t)} \, dt
\]

\[
\leq \left( \frac{1}{2\pi} \right)^n \int_{\mathbb{R}^n} \sum_{i=1}^{M} u_i^{(\mu)}(s)e^{j\mu \cdot t} - (j t)^{\mu} e^{j\mu \cdot t} \right)^2 \hat{\Phi}(s) \, dt \
\times \int_{\mathbb{R}^n} \tilde{f}(t)^2 /\Phi(s) \, dt
\]

\[
= (\kappa^{(\mu)}(s))^2 c_f^2,
\]

where \(\tilde{\Phi}\) is the Fourier transform of the chosen RBF (considered here as the multivariable function \(\Phi(\|s\|)\)), and \(c_f\) is the norm of \(f\) in the Hilbert space defined by the inner product

\[
\langle f_1, f_2 \rangle := \int_{\mathbb{R}^n} f_1(t) f_2^*(t) /\Phi(t) \, dt.
\]

Note that this step implicitly identifies the class of interpolated functions to which the convergence proof applied: those for which \(\|f\|_\Phi < \infty\). The function \(\kappa^{(\mu)}(s)\), called the Kriging function in [22], is defined by the variational problem

\[
(\kappa^{(\mu)}(s))^2 := \min_{U \in K^{(\mu)}(s)} \{ U^T A U - 2U^T R^{(\mu)}(s) + \Phi^{(2\mu)} \},
\]

where \(K^{(\mu)}(s)\) is the set of vectors \(U\) with the property of reproducing the \(\mu\)-th derivative of all polynomials up to order \(k\) (this is the same order that renders the RBF conditional positive definite). Formally,

\[
K^{(\mu)} := \left\{ U \in \mathbb{R}^m \mid \sum_{i=1}^{M} u_i p(s_i) = p^{(\mu)}(s) \quad \forall p \in P^{k-1}_n \right\}.
\]

It can be shown that the vector whose entries are the Lagrange-like basis, \(U = (u_1^{(\mu)}(s), \ldots, u_m^{(\mu)})\) is the minimizing vector for (2.20). As such, the expression for the kriging function in (2.17) is simply the Fourier transform of (2.20), along with certain

\footnote{It appears that there may be an error in [22]. In equation (3.2) of that paper, the authors find that the Fourier transform of \(\Phi^{(\mu)}(0)\) is equal to \(1/(2\pi)^m \int_{\mathbb{R}^n} (j t)^{\mu} e^{j\mu \cdot t} \tilde{\Phi}(t) \, dt\). This is actually the}
assumptions for the asymptotic behavior of $\tilde{\Phi}(t)$ as $\|t\|$ goes to zero and infinity, to ensure that the Fourier transform integrals are well-defined.

The final major step consists of constructing a particular, non-optimal $\bar{U} \in K^{(\mu)}$, such that

$$\left( \kappa^{(\mu)}(s) \right)^2 \leq \frac{1}{(2\pi)^n} \int_{\mathbb{R}^n} \left| \bar{u}^{(\mu)}(s) e^{is \cdot t} - (jt)^\mu e^{is \cdot t} \right|^2 \tilde{\Phi} \, dt. \quad (2.22)$$

This choice allows one to finally obtain the bound

$$\kappa^{(\mu)}(s) \leq C h_r^{s_{\infty} - |\mu|}(s) \quad (2.23)$$

for $h_r$ smaller than some constant $h_0$. The exponent $s_{\infty}$ describes the decay of the radial basis function’s Fourier transform, in particular

$$\tilde{\Phi}(t) < c \|t\|^{-n - s_{\infty}} \quad \text{for } \|t\| \to \infty \quad (2.24)$$

for some positive constant $c$.

Some specific applications of this result are as follows. The Gaussian RBF $\Phi(r) = e^{-cr^2}$ has the Fourier transform

$$\tilde{\Phi}(t) = 2^{-n}(c\pi)^{-n/2} e^{-\|t\|^2/4c}, \quad (2.25)$$

yielding $s_{\infty}$ arbitrarily large. Thus, interpolation with Gaussian RBFs can converge exponentially. Unfortunately, as states earlier, the proof requires that $f$, the function being interpolated, have a Fourier transform that is “dominated” by that of the radial basis function, in the sense that

$$\int_{\mathbb{R}^n} \frac{|\tilde{f}|^2}{\tilde{\Phi}} \, dt < \infty. \quad (2.26)$$

Fourier transform of $(-1)^{|\mu|} \Phi^{(2\mu)}(0)$. However, this does not appear to affect the validity of the proof, because a change in the additive constant does not affect the solution of (2.20).
In other words, $f$ must be infinitely differentiable for the proof to apply. Similar results apply for the multiquadric $\phi(r) = (r^2 + c^2)^{2/\alpha}$, whose Fourier transform is proportional to the Bessel function of the second kind and thus also decays exponentially for large values of $\|t\|$.

In contrast, the polyharmonic splines have Fourier transforms proportional to $\|t\|^{-2k}$. Therefore, the smoothness requirement on the interpolated function is

$$\int_{\mathbb{R}^n} |\hat{f}|^2 \cdot \|t\|^{2k} \, dt,$$

which is equivalent to $f$ having its $k$-order derivatives be square-integrable. Then the rate of convergence is $k - n/2$.

We note, however, that the above result applies when the domain is the entire space $\mathbb{R}^n$. In the case of polyharmonic splines, results [21] are available for more general domains:

**Theorem 2.3.1** Let $\Omega \subset \mathbb{R}^n$ be bounded, open, and satisfying the interior cone condition, and $f \in H^p(\Omega)$ be supported in $\Omega$. Then for any $\Xi \subset \Omega$ that contains a $\mathbb{R}^{p-1}_d$-unisolvent subset, the interpolant $\hat{f}$ on $\Xi$, using the RBF

$$\Phi(r) = \begin{cases} r^{2p-n} \log r, & \text{if } 2p - n \text{ is an even integer} \\ r^{2p-n}, & \text{if } 2p - n \text{ is not an even integer} \end{cases}$$

satisfies the error estimate

$$\|\hat{f} - f\|_{2,\Omega} \leq Ch^p,$$

with $h = \sup_{x \in \Omega} \inf_{\xi \in \Xi} |x - \xi|$ and $C$ independent of $h$.

An example of this can be seen in Fig. 2.1 which clearly shows the higher-order convergence for a smooth test function.
Figure 2.1: Convergence of polyharmonic spline approximation of a test function interpolated on a uniform grid
Chapter 3: Adaptive MOR Process

3.1 Error Estimator

Error estimator is very important for the finite element method and also for MOR, since an accurate error estimator will indicate the direction for optimization. For finite element method and other adaptive analysis, where the approximation is based on a successful refinement to approach the real solution. It is crucial for practical design and analysis.

3.1.1 Residual

Given an approximate solution \( x_{appx} \), one of the most natural and straightforward error estimates is the residual

\[
\mathbf{r} = A x_{appx} - \mathbf{b}.
\]  

(3.1)

This estimate is also somewhat rigorous, in the sense that it can be bounded in terms of the true error \( \mathbf{e} \):

\[
\mathbf{r} = A (x_{appx} - \mathbf{x})
\]

\[
= A \mathbf{e}
\]

\[
\mathbf{e} = A^{-1} \mathbf{r}
\]  

(3.2)

\[
\|\mathbf{e}\| \leq \|A^{-1}\| \cdot \|\mathbf{r}\|.
\]
However, the fact that the norm of the system matrix $A$ depends on the problem formulation, discretization, etc., means that this bound is not terribly useful in practice. However, if a good preconditioner $M$ is available, such that $MA \approx I$, then this bound becomes tighter. A more important disadvantage is that calculation of residual requires a matrix-vector multiplication. While not as costly as solving the linear system, this nonetheless represents a significant burden because dozens, even approaching 100, evaluations of the error estimator at different points will be necessary in each iteration of the adaptive process. Therefore, we will focus on more efficient estimators that can still offer useful guidance for an adaptive sampling procedure.

### 3.1.2 Change in MOR Solution

Another option is to look at the change in the MOR solution after a sample point is added. That is, after adding the $m$-th sample point, the reduced order basis matrix is

$$V_m = [V_{m-1} \ v_m] \quad (3.3)$$

These two basis matrices in turn give rise to two reduced order models, the order $m-1$ system before adding the $m$-th sample point,

$$\hat{V}_{m-1}^T A V_{m-1} x_{m-1} = V_{m-1}^T b, \quad (3.4)$$

and the order $m$ system

$$\hat{V}_m^T A V_m x_m = V_m^T b, \quad (3.5)$$

where a hat $\hat{}$ indicates RBF interpolation of the underlying quantity. The error estimate used to determine the $(m+1)$-th sample point is the difference between
these solutions,

\[ EE_1(s) = \frac{\|V_m x_m - V_{m-1} x_{m-1}\|}{\|V_m x_m\|} \]  

(3.6)

\[ = \frac{\|V_m \left[ x_m - \begin{pmatrix} x_{m-1} \\ 0 \end{pmatrix} \right]\|}{\|V_m x_m\|} \]  

(3.7)

\[ = \frac{\|x_m - \begin{pmatrix} x_{m-1} \\ 0 \end{pmatrix}\|}{\|x_m\|}, \]  

(3.8)

where the final line makes use of the fact that \( V_m \) has orthonormal columns, so that \( \|V_m x\| = \|x\| \).

This is similar to other error estimators in use, such as that of [31], which uses the change in the \( S \) parameters of a microwave circuit to determine when to stop the adaptive mesh refinement process. However, it is not so useful by itself, as will be shown in more detail in section 3.2.1. This is because although (3.8) may be a good guide as to when the MOR process has converged, it is not a good guide for choosing the next sample point.

### 3.1.3 Difference Between Two Reduced Order Models

Instead of examining two reduced order models in which one is a lower-dimensional version of the other, we might instead compare two models with the same order. For although we have been using a Galerkin procedure with \( V^T A V \), there is no reason in theory why a different set of testing functions cannot be used, \( W^T A V \). The resulting error estimator has the form

\[ EE_2 = \frac{\| (V^T A V)^{-1} V^T \hat{b} - (W^T A V)^{-1} W^T \hat{b} \|}{\| (V^T A V)^{-1} V^T \hat{b} \|}. \]  

(3.9)
We can demonstrate that this estimator is differentiable everywhere or almost everywhere in the parameter space. For ease of notation, only the single parameter case is shown here; since the parameters are independent, the calculation can be repeated for each parameter to obtain the corresponding component of the gradient in the multidimensional case. The two solutions are written as

\[ \hat{x} = (V^T AV)^{-1} V^T \hat{b} \]
\[ \hat{y} = (W^T AV)^{-1} W^T \hat{b}. \]

Then we have

\[ \frac{d}{ds} \| \hat{x} - \hat{y} \|^2 = \frac{d}{ds} \left[ (\hat{x} - \hat{y})^H (\hat{x} - \hat{y}) \right] \]
\[ = 2 \Re \left[ (\hat{x} - \hat{y})^H \frac{d}{ds} (\hat{x} - \hat{y}) \right] \]
\[ = 2 \Re \left[ (\hat{x} - \hat{y})^H \left( \frac{d\hat{x}}{ds} - \frac{d\hat{y}}{ds} \right) \right] \]

The derivative of the reduced order model solution \( \hat{x} \) can be written as

\[ \frac{d\hat{x}}{ds} = \frac{d}{ds} \left[ (V^T AV)^{-1}(V^T \hat{b}) \right] \]
\[ = (V^T AV)^{-1} V^T \frac{d\hat{b}}{ds} + \frac{d(V^T AV)^{-1}}{ds} V^T \hat{b} \]
\[ = (V^T AV)^{-1} V^T \frac{d\hat{b}}{ds} + (V^T AV)^{-1} \frac{dV^T AV}{ds} (V^T AV)^{-1} V^T \hat{b} \]

Likewise,

\[ \frac{d\hat{y}}{ds} = (W^T AV)^{-1} W^T \frac{d\hat{b}}{ds} + (W^T AV)^{-1} \frac{dW^T AV}{ds} (W^T AV)^{-1} W^T \hat{b}. \]

Because the individual entries of \( \hat{b}, V^T AV, \) and \( W^T AV \) are simple linear combinations of radial basis function and polynomials, their derivatives exist wherever the derivatives of the RBFs exist. In the case of Gaussians \((c \neq 0)\), multi-quadrics \((c \neq 0)\),
and the thin-plate spline \( r^2 \log r \), the RBF is differentiable everywhere. Other cases, most notably the linear function \(|r|\), are differentiable everywhere except the center points. Thus, it has been shown that the error estimator of this section is differentiable everywhere in parameter space, with the possible exception of the RBF center points.

The choice of \( W \) is, at this time, heuristic. Although there is some guidance about optimal choices when one is interested in an output variable that is a linear functional of the solution (e.g., scattering parameters or radar cross section in one or more direction) [13], in this work the focus is on the overall error of the solution vector. Several choices have been investigated. One initially appealing idea is to perform RBF interpolation on the matrix \( V^T A^T(s) A(s) V \), to obtain the least-squares solution using the reduced order basis. The problem observed with this approach, however, is that the normal equations tend to be more difficult to interpolate than the original reduced order system, due to the term \( A^T(s) A(s) \). Consider, for example, that if \( A \) has a polynomial form, then the order of the normal equations is twice that of the original matrix; similarly, if there is a trigonometric dependence then the bandwidth is doubled. So, by the time enough sample points have been accumulated for \( V^T A^T(s) A(s) V \) to be interpolated accurately, and therefore serve as a valid basis for comparison, the reduced order system that we are really interested in has been drastically oversampled. This is illustrated in Fig. 3.1, which shows the actual and estimated error for a test problem (described full in section 4.2.3).

Among choices that lead to a constant \( W \) matrix, such as \( W = A^T(0.5, \ldots , 0.5)V \) and \( W = A^{-1}(0.5, \ldots , 0.5)V \), we have experimentally found that the latter provides
Figure 3.1: Comparison of true and estimated error with least-squares comparison, \( \hat{B} = V^T A^T(s) A(s) V \).
good correspondence with the true error. This is illustrated in detail in the numeric example of section 4.2.

3.2 Choosing the Sample Point

3.2.1 Fixed Grid Selection

One approach to adaptively choosing the next sample point begins with a single sample point in the middle of the parameter space. We imagine splitting the space in half along each dimension and placing a “test point” at the center of each child box, as shown in Figure 3.2. The error estimator is evaluated at each of the test points, and whichever has the greatest error becomes the next sample point. Furthermore, the box containing the new sample point is split in half along each dimension, and yet more test points are added to the centers of the new grandchild boxes. Thus, as seen
Figure 3.3: Distribution of sample and test points after adding samples, and tree structure relating test and sample points

in Figure 3.3, a hierarchical tree structure is generated with test points represented as the leaves and sample points by the interior vertices.

The pseudocode for the hierarchical sampling algorithm is shown in Fig. 3.4. Although this procedure is easy to implement, it has a number of shortcomings. One is that location of the sample points is strictly limited to the interior of the parameter space and can only asymptotically approach the boundary. This has serious implications for the radial basis function interpolation procedure that is used, because
RBFs are not well-suited for extrapolation, since they asymptotically approach either infinity or zero.

An even more serious flaw comes from limiting the potential sample point sites to a finite set. Figure 3.5 shows the error in the reflection coefficient for the patch antenna problem of section 4.2 as a function of frequency with the material properties fixed at $\epsilon_{r1} = 4$ and $\epsilon_{r2} = 3$. It can be seen that the model reduction process, for the most part, is error-controllable; a decrease in the model tolerance by an order of magnitude generally results in the true error also decreasing by an order of magnitude. The obvious exceptions are the two resonance points around 4.3 and 6.7 GHz. The root cause of this problem is the fact that, due to the fixed grid of testing points, it is not possible to sample right at a resonance point. Indeed, there isn’t even a test point placed close enough to resonance to detect the lack of convergence.
3.2.2 Optimization-Based Selection

Because of the limitations of the fixed-grid sampling described above, the natural evolution is to use an optimization-based procedure that searches for the exact point that maximizes the error estimate. Some care is required, however, because not all error estimators are suitable for this procedure.

Consider, for example, the estimator of section 3.1.2 which looks at the difference between two successive reduced order models. Under ideal circumstances, when a sample point is added it is at the point with maximum error. When the optimizer looks for where to add the next sample point, it is looking for the location where the error has changed the most. This is in fact the previous sample point: the error has gone from (presumably) the maximum to zero, since the reduced order model interpolates the solution at the sample points. Therefore, maximizing the error estimate of section 3.1.2 tends to result in sampling the same, or very nearby, locations twice.

Figure 3.5: Patch antenna $S_{11}$ error with hierarchical sampling
That said, we move on to the optimization algorithm. Because the error estimator is differentiable, it is desirable to take advantage of this information by using a gradient-based or quasi-Newton method to speed convergence. A powerful approach for nonlinear problems with differential objective functions is the trust region method \[32\]. Like gradient descent and Newton’s method, it is an iterative algorithm that generates a sequence of steps that, hopefully, converge to an extremum. It begins by assuming a quadratic approximation for the objective function,

\[
f(p + s) \approx f(p) + g^T s + \frac{1}{2} s^T H s,
\]

where \( g = \nabla f \) and \( H \) is the Hessian matrix \( H_{ij} = \partial^2 f / \partial p_i \partial p_j \), or an approximation thereof. Newton’s method would calculate the step size as

\[
H s = -g,
\]

which does indeed lead to a stationary point if \(3.19\) is exact. For nonlinear problems, however, this is unlikely to be effective. Therefore, in the trust region method, the step \( s \) is constrained to be within a certain neighborhood of the current guess \( p \); this ensures that the quadratic approximation is sufficiently accurate that the algorithm continues to make progress towards a stationary point. Thus, the problem solved at each step of the trust region method is

\[
\min \left\{ g^T s + \frac{1}{2} s^T H s : \|s\| \leq \Delta \right\},
\]

where \( \Delta > 0 \) is the size of the trust region. This problem is solved by means of a Lagrange multiplier.

After each step is computed, the predicted change in \( f, g^T s + \frac{1}{2} s^T H s \), is compared to the actual change \( f(p + s) - f(p) \). Depending on whether these two values are
“very close”, “somewhat close”, or “not close”, the size of the trust region is enlarged, kept the same, or reduced. In this way, a balance is struck between maintaining the accuracy of the quadratic approximation and allowing larger step sizes. Note also that for highly nonlinear problems, which require a small trust region size to ensure an accurate approximation, the method approaches gradient descent, since the quadratic term in the objective function becomes less and less important as the allowable step size becomes smaller.

Some additional steps are necessary to account for the upper and lower bounds of the parameter space \[33\]. Because the parameter values have been normalized, the overall problem can be stated as

\[
\min \{ f(p) : 0 < p < 1 \}. \tag{3.22}
\]

This problem is solved via a series of trust region subproblems, but with a slight change. Rather than an isotropic spherical constraint \( \|s\| \leq \Delta \), a diagonal scaling matrix \( D(p) \) is introduced. In the special case of a normalized range \([0, 1]\), the entries of this matrix are

\[
D_{ii}(p) = \begin{cases} 
1/\sqrt{1-p_i}, & g_i < 0 \\
1/\sqrt{p_i}, & g_i \geq 0 .
\end{cases} \tag{3.23}
\]

Then the trust region has the shape \( \|Ds\| \leq \Delta \). The purpose of this scaling is to bias the step away from the boundary. For example, if \( g_i \) is negative, i.e. the function can be decreased by moving in the positive direction along the \( i \)-th axis, then \( D_{ii} \) grows as the upper boundary is approached, limiting the size of the step that can be taken in that direction, while leaving the allowable step size parallel to the boundary unaffected.
1: function MOR-Main
2: Sample \((p+1)^d\)-point Cartesian grid
3: Calculate \(V, W = A^{-1}(0.5, \ldots, 0.5)V, \tilde{A}, \tilde{B}\)
4: Perform RBF interpolation of \(A, B, \) and \(b\)
5: \(i \leftarrow (p+1)^d\)
6: repeat
7: \(i \leftarrow i + 1\)
8: \(\{s_i, error\} \leftarrow \text{ERROR\_ESTIMATE}\)
9: \(x \leftarrow A^{-1}(s_i)b(s_i)\)
10: \(v \leftarrow \text{MGS}(V, x)\)
11: \(w \leftarrow A^{-1}(0.5, \ldots, 0.5)v\)
12: \(V \leftarrow (V v)\)
13: \(W \leftarrow (W w)\)
14: for \(j \leftarrow 1, i\) do
15: \(\tilde{A}_j \leftarrow V^T A(s_j)V\)
16: \(\tilde{B}_j \leftarrow W^T A(s_j)V\)
17: end for
18: \(\tilde{A}_i \leftarrow V^T A(s_i)V\)
19: \(\tilde{B}_i \leftarrow W^T A(s_i)V\)
20: Perform RBF interpolation of \(\tilde{A}, \tilde{B},\) and \(b\)
21: until \(error < \tau\)
22: end function

Figure 3.6: Pseudocode for optimized sampling MOR algorithm

After computing each trust region step, the scaled gradient direction \(-D^{-2}g\) is also calculated, and a line search is performed along each of these two directions to determine which minimizes the quadratic approximation \((3.19)\) inside the parameter space. Whichever step is predicted to generate the most change becomes the trial step, and, as described above, the true change in the objective function by using this step is compared to the change predicted by the quadratic model in order to adjust the trust region size. With this improved adaptive sampling scheme, the pseudocode for the MOR process is shown in Fig. 3.6. In the next chapter, we will introduce some
examples to validate the MOR algorithm and also the efficiency of the optimization-based sampling scheme.

The procedure begins by placing sample points on an \( d \)-dimensional uniformly spaced grid, with \( \min(p+1,2) \) points along each dimension, where \( p \) is the order of the supplemental polynomial that is added to the radial basis functions. The minimum of 2 aids robustness by ensuring that all of the corners of parameter space are sampled, so that the algorithm is always interpolating and never extrapolating. Next, the procedure enters the main loop. The error estimator is evaluated to determine if the user-specified tolerance has been met and, if not, where to add the next sample point.

After the solution and right-hand side are calculated at this sample point, the reduced order matrices \( \tilde{A} \) are calculated at each sample point. At the most recent sample point, this matrix must be calculated from scratch, as indicated by the explicit multiplication \( V^T A V \). At the previous sample points, however, this is not necessary because the only change to \( V \) is the addition of a column. Thus, the only change to \( V^T A V \) is the addition of one column and one row. The required updates for the old sample points can be described as follows. Suppose that there are \( m \) sample points, including the one just added, and partition the \( V \) matrix as \( V = (V_{m-1} \ v_m) \), where \( V_{m-1} \) consists of the first \( m-1 \) columns of \( V \) and \( v_m \) is the right-most column. Then the new, updated, reduced order matrix is

\[
\begin{pmatrix}
V_{m-1}^T A V_{m-1} & V_{m-1}^T A v_m \\
v_m^T A V_{m-1} & v_m^T A v_m
\end{pmatrix}.
\] (3.24)

On its face, this requires \( m \) matrix-vector multiplications and \( 2m - 1 \) dot products. Several possibilities for optimization are available at this point, depending on the how much memory the computer running the MOR program has and the capability of the solver being interfaced with. If sufficient memory is available, the quantity \( A V_{m-1} \) can
be stored at each sample point, reducing the required number of additional matrix-vector multiplications to one. On the other hand, if the solver is also capable of performing a transposed matrix-vector multiplication, then the term $v_m^T A V_{m-1}$ can be calculated as

$$v_m^T A V_{m-1} = [V_{m-1}^T (A^T v_m)]^T.$$  \hspace{1cm} (3.25)

In this case, no additional memory is required to store the partial result $AV_{m-1}$, and the cost is only two matrix-vector multiplications, one transposed and one not. Similar observations apply, of course, to the computation of $\tilde{B} = W^T A V$.

There is one final point to note about the algorithm in Fig. 3.6. It is presented with the assumption that a reduced order model is being generated for the entire normalized parameter space $[0, 1]^d$. Since $hp$ refinement is used for the RBF interpolation, this creates the possibility of generating an independent reduced order model for each domain, with its own sample points, $V$, $W$, etc. This would be beneficial because these models can be generated simultaneously. In such cases, locations in parameter space such as $s = (0.5, \ldots, 0.5)$ should be understood as being scaled to the domain under consideration and not as global coordinates.

### 3.3 Solver Interface

The interface between the solver and the MOR code is of some importance, given the goal of generality. As discussed previously, it is desired that the MOR code be able to easily interface with a variety of solvers, and to that end it is based on performing simple linear algebra operations. We now describe this interface in some detail.

Because the MOR program exists as a separate executable, it needs to call the solver program to perform the desired tasks. In order to hide the details of the
#! /usr/bin/python

import os
import sys
import mor

id = sys.argv[1]
mor.ParseCommandLine(sys.argv)
mor.EnsureDirectory(id)
os.chdir(id)
os.system("mv ../param_solve_rhs-"+id+"./param_solve_rhs")
mor.RunSolver("solve")
os.system("mv param_solution ../param_solution-"+id)
os.system("rm param_solve_rhs")

Figure 3.7: Code listing of sample solve.sh file.

Solver executable name, command line arguments, etc., this is accomplished by using three user-generated scripts: solve.sh, mvm.sh, and rhs.sh. Note that despite the sh extension, these files need not be shell scripts and could instead be written or Python or Perl, for example. A sample solve.sh file in shown in Figure 3.7. These scripts take, as command line arguments, a non-negative integer, followed by the normalized parameter values. The purpose of the first is to uniquely identify each instance of the solver that is running. That is, there are some portions of the MOR algorithm that are inherently parallel, such as the updating of $\tilde{A}$ and $\tilde{B}$ in lines 14–17 of Figure 3.6. However, since different instances of the solver are performing matrix multiplications with $A(s)$ at different locations in parameter space, it is possible that they will have different input files and certain that their outputs will be different. To
facilitate parallel operation under these circumstances, the MOR program provides each instance with a unique integer identification number. This number could be passed directly to the solver program, in order to allow it to create unique input and output file names. However, depending on how many such files the program reads and writes, this could require relatively burdensome modifications. An alternative, used for this work, is to simply use the number to create a unique directory for each instance of the solver to run in. This is demonstrated on lines 10 and 11 of Figure 3.7. The `EnsureDirectory` function, shown in lines 14–17 of Figure 3.8, checks to see if the unique directory already exists and, if it doesn’t, creates the directory and makes a copy of the necessary solver input files. After the solver right-hand side is copied (if it exists) into the directory and the solver is run, the output file `param_solution` has the id number appended to the end and is copied back to the main working directory.

Another purpose of the interface scripts is to translate the normalized parameters into physical values. Because this functionality is needed by all three scripts, it is best if it can be located in a common module that can be shared. In this example, Figure 3.8, lines 8–12, the parameter is frequency, and its range is 100–1200 MHz. The calculated value is stored in a global module variable so that it can later be passed as a command line argument to the solver program. In other cases, however, such as when a parameter represents a material property, it will be necessary for the user to write a small program or script to modify the input files as needed after copying them to the unique directory that the solver will run in. Similarly, geometrical changes will require that the coordinates of the nodes in the mesh be changed. Once the parameter values have been calculated and the input files updated, the solver is finally called...
#!/usr/bin/python

import os
import string

# Global variable for frequency

def ParseCommandLine(argv, offset=0):
    global gFreq
    x = float(string.atof(argv[2 - offset]))
    gFreq = 100 + 1100 * x

def EnsureDirectory(id):
    if not os.access(id, os.F_OK):
        os.mkdir(id)
        os.system("cp cavity.tri ./" + str(id))

def RunSolver(mode, output = "/dev/null"):
    eigerExe = "efie"
    geo = "cavity"
    theta = 45
    phi = 0

    if (mode == "solve"):
        numMode = 2
    elif (mode == "matvec"):
        numMode = 0
    elif (mode == "rhs"):
        numMode = 1

    command = "%s%s%f%f%f%f%0%d%>s" % (eigerExe, geo, gFreq, theta, phi, numMode, output)
    os.system(command)

Figure 3.8: Code listing of mor.py, containing common functions for user scripts.
via the `RunSolver` function (Figure 3.8 lines 19–33). Based on the `mode` argument, a flag is passed to the solver program, indicating the desired operation.

### 3.4 Parallelization

It was hinted earlier that one of the design considerations for the solver interface was the ability to run multiple instances of the solver simultaneously. This section outlines in more detail the parallelization of the presented MOR algorithm. Note that in the following discussion, the term “process” will be used to denote a set of computer resources capable of running a single instance of the solver program. In the case of a serial solver, this will match with the usual meaning of the word. In the case of parallel solver, however, it will be different. Thus, if the solver is MPI-parallel, will be run with 2 nodes, and 6 nodes are available in total, then the user would indicate to the MOR program that 3 “processes” are available. Likewise, an OpenMP-parallel code using 4 cores on an 8-core machine would present two “processes”.

The first step is to analyze the algorithm in Fig. 3.6 to ascertain which calculations can be performed in parallel. The first candidate is line 2, sampling on the initial \((p + 1)^d\)-point grid. This step entails solving the linear system at the known sample points, a task that can easily be distributed among the available processes. Also, once the solution vectors have been orthogonalized to form \(V\), it is easy to distribute the work of multiplying \(AV\) at every sample point, a necessary step in calculating \(\tilde{A}\) and \(\tilde{B}\); these matrix-vector multiplications can also be performed concurrently with the solves needed to calculate \(W\). It is evident that this initial sampling stage offers excellent parallelization opportunities.
The adaptive loop, however, presents fewer chances. The main impediment is that only two solves are need per iteration, in lines 8 and 10, and they must be performed sequentially because of the data dependence (the result of the first solve is used to form the input of the second). All is not lost, however, as the loop to multiply the new $\mathbf{v}$ vector by $A$ (and possibly $A^T$ also) at all of the previous sample points can clearly be done in parallel, including the multiplication of the entire $V$ matrix by $A$ at the newest sample point.

The possibility of $h$ refinement of the parameter space means that there may be multiple independent reduced order models that can be generated simultaneously. Therefore, a strategy must be developed to match idle processes with domains. One possibility is to simply run one domain at a time, allocating all process to it. This is not optimal, however, because of the inherently sequential nature of many parts of the ROM-generation algorithm. Instead, one should attempt to maximum the number of domains being worked on at any one time. In the limit that the number of domains is greater than or equal to the number of process, this results in nearly perfect parallel efficiency. This is achieved by assigning one process to each region until there are no idle processes available. Otherwise, the next available domain is given $\lfloor P/D \rfloor$ processes, where $P$ is the current number of idle processes and $D$ the number of domains available for work. As the calculation for each domain finishes, due to either convergence or $h$ refinement, processes become idle again and are allocated to other domains, and the procedure repeats until all domains have converged.

In the following chapter, containing numerical examples of the MOR algorithm described herein, and mix of the two parallelization strategies will be shown, with
some problems using parallel solvers and others relying on multithreading of the MOR algorithm itself.
Chapter 4: Applications of Black-Box MOR Algorithm

4.1 Oscillator Circuit in Enclosure

4.1.1 Problem Statement

This problem consists of the PEC traces on the printed circuit board of an oscillator circuit, shown in Figure 4.1a, placed inside a perfectly conducting enclosure with an aperture on the top, Figure 4.1b. Excitation is provided by a plane wave incident from the top of the enclosure. The two parameters are the lateral displacements of the circuit board, and the outputs of interest are the currents through two particular wires.

Although this system can be simulated in any given configuration using the finite element method, the large amount of displacement that the board undergoes makes it impractical to apply model order reduction to the problem. This is because reduced order modeling seeks to approximate the solution of a matrix equation with a linear combination of a small number of global basis functions. In order for this approach to work, the degrees of freedom must have the same meaning for all the basis vectors, for example, the electric field parallel to a particular mesh edge or the electric current flowing across an edge. If the problem were to be remeshed for different configurations, the resulting solution vectors would not be compatible representations of the solution.
This, in turn, implies that geometrical parameters must be handled by perturbing the
nodes of a mesh without changing its connectivity.

4.1.2 Formulation

When using the finite element method, the empty space inside the enclosure must be
meshed. This seriously limits the distance that objects can be moved before the
mesh becomes too distorted to provide accurate results. Therefore, an integral
equation method is used to analyze the problem. The great benefit of this approach is
that only the surfaces of objects needs to be meshed, allowing disjoint bodies to be
moved arbitrarily with respect to each other without any mesh distortion. Because the
problem consists of conducting surfaces that are open, the electric field integral
equation is employed.
The EFIE begins with the representation formula for the electric field radiated by a surface current distribution,

\[
E_{\text{rad}}(x) = jk\eta \int_G \frac{e^{-jk|x-x'|}}{4\pi|x-x'|} J(x') \, dx'^2 + \frac{j}{k\eta} \nabla \int_G \frac{e^{-jk|x-x'|}}{4\pi|x-x'|} \nabla' \cdot J(x') \, dx'^2
\]  

(4.1)

\[
= jk\eta A_k(J) + \frac{j}{k\eta} \nabla A_k(\nabla' \cdot J)
\]  

(4.2)

Clearly, (4.1) shows that both \(J\) and its divergence should be square integrable. This is also desirable on physical grounds, since it is equivalent to requiring that the energy in both the current \(J\) and the charge \(\rho = \nabla \cdot J / j\omega\) be finite. Either way, the conclusion is that we seek a solution in the divergence-conforming function space

\[
H^{(-1/2)}(\text{div}, \Gamma) := \left\{ v \in H^{-1/2}(\Gamma) \mid \int_\Gamma \|v\|^2 + \|\nabla \cdot v\|^2 \, dx^2 < \infty \right\}. 
\]  

(4.3)

The boundary condition is that on the surface of the PEC, the tangential component of the total electric field \(E\), consisting of the sum of the radiated field and the incident field \(E_{\text{inc}}\), must be zero. Therefore the residual is

\[
R_{\Gamma} := \pi_t(E_{\text{rad}}) - jk\eta \pi_t[A_k(J)] - \frac{j}{k\eta} \pi_t[A_k(\nabla' \cdot J)] \in H^{-1/2}(\text{curl}, \Gamma).
\]  

(4.4)

This residual has the physical meaning of a surface electric field. Therefore, duality pairing suggests that it should be testing with a surface electric current, in order to obtain a \(J \cdot E\) energy density. This again results in a Galerkin formulation: seek \(J \in H^{-1/2}(\text{div}, \Gamma)\) such that

\[
jk\eta \langle w, A_k(J) \rangle_\Gamma + \frac{j}{k\eta} \langle w, \nabla A_k(\nabla' \cdot J) \rangle_\Gamma = -\langle w, E_{\text{inc}} \rangle_\Gamma
\]  

(4.5)

for all \(w \in H^{-1/2}(\text{div}, \Gamma)\). Applying integration by parts to the second term on the left-hand side of (4.5) and taking advantage of the fact that \(w\) and \(J\) are tangential to the boundary gives

\[
jk\eta \langle \nabla \cdot w, A_k(J) \rangle_\Gamma + \frac{j}{k\eta} \langle \nabla \cdot w, A_k(\nabla' \cdot J) \rangle_\Gamma = -\langle w, E_{\text{inc}} \rangle_\Gamma.
\]  

(4.6)
Table 4.1: Computational results for circuit board example

To discretize this equation, Rao-Wilton-Glisson basis functions \([34]\) are used. This results in a matrix equation

\[ Zx = v, \quad (4.7) \]

where

\[
Z_{ij} = jk\eta \int \int \frac{e^{-jk|x-x'|}}{4\pi||x-x'||} v_i(x) \cdot v_j(x') \, dx'^2 \, dx^2 
\]

\[
+ \frac{j}{k\eta} \int \int \frac{e^{-jk|x-x'|}}{4\pi||x-x'||} \nabla \cdot v_i(x) \cdot \nabla' \cdot v_j(x') \, dx'^2 \, dx^2 
\]

\[ v_i = \int_G v_i(x) \cdot E_{inc}(x) \, dx^2 \quad (4.10) \]

Clearly, on account of the presence of \( k \) in the Green function, there is no simple decomposition of the \( Z \) matrix as a function of frequency, as was the case when using the finite element method.

4.1.3 Numerical Results

Table 4.1 summarizes the computational results for the circuit board examples. Note that the times for the model creations and for a single, full-order solution, are using two 8-core nodes. The simulation program used is EIGER \([35]\), and the matrix equation is solved using a parallel LU factorization. The breakeven point is
[27:15/0:42] = 39 samples. In other words, if the number of samples need is greater than or equal to 39, then it is faster to create and evaluate the reduced order model, rather than repeatedly solving the full-order system. Once the reduced order model has been created, the incremental cost for an additional solution is approximately 320,000 times faster than the full-order solution.

Figure 4.2 shows the induced current on the enclosure and the circuit board. The adaptive sampling grid, using the hierarchical procedure described in Section 3.2.1, is shown in Fig. 4.3. The induced current on the circuit board is shown in Fig. 4.4 for several different locations of the board. It can easily be seen that the location has a significant effect, with the induced current being much greater in configurations (b) and (d). Finally, Fig. 4.5 shows several more details of the solution.
4.2 Patch Antenna

4.2.1 Problem Statement

The next example is a cavity-backed patch antenna previously analyzed in [1, 13]. The geometry of the antenna is shown in Fig. 4.6. It consists of a PEC patch sandwiched between two dielectric slabs and fed from beneath by a coaxial cable, and is designed to resonate between 3 and 7 GHz, depending on the permittivity of the dielectric blocks. Table 4.2 shows the parameters of interest.
Figure 4.4: Induced current with four different locations of circuit board
Figure 4.5: Solution details of oscillator circuit board
Figure 4.6: Cavity-backed patch antenna layout. All dimension in mm: $a = 47.45$, $b = 47.45$, $h = 18.15$, $l = 13.5$, $s = 1.75$, $d = 2.42$, $e = 2.6$, $r_i = 0.64$, $r_o = 2.05$ (from [1]).
The discretized problem has 84,718 unknowns using mixed-order, curl-conforming basis functions \[36\]. An absorbing boundary condition is used on the air region above the patch, as well as the region where the coaxial cable’s inner dielectric intersects the boundary. Excitation is provided by using the cable’s TEM mode as the incident field on the cable’s part of the ABC.

### 4.2.2 Formulation

This problem is analyzed with the finite element method, which begins with the differential form of the Maxwell equations,

\[
\nabla \times \mathbf{E} = -j\omega \mu \mathbf{H} \quad (4.11)
\]

\[
\nabla \times \mathbf{H} = j\omega \epsilon \mathbf{E} + \mathbf{J}. \quad (4.12)
\]

Taking the curl of (4.11) and substituting (4.12) for \( \nabla \times \mathbf{H} \) results in the vector wave equation

\[
\nabla \times \frac{1}{\mu_r} \nabla \times \mathbf{E} - k_0^2 \epsilon_r \mathbf{E} = -j\omega \mu_0 \mathbf{J}, \quad (4.13)
\]

which is to hold in the entire problem domain \( \Omega \). \( \mu_r = \mu/\mu_0 \) is the relative permeability, \( \epsilon_r = \epsilon/\epsilon_0 \) is the relative permittivity, and \( k_0 = \omega \sqrt{\mu_0 \epsilon_0} \) is the free-space wavenumber. The statement of the boundary value problem is completed by adding the boundary conditions

\[
\mathbf{e} = 0 \quad \text{on } \partial \Omega_{pec} \quad (4.14)
\]

\[
\mathbf{j} = \frac{j\omega \mu_0}{\eta} \mathbf{e} - \frac{j\omega \mu_0}{\eta} \mathbf{e}_{inc} + \mathbf{j}_{inc} \quad \text{on } \partial \Omega_{abc} \quad (4.15)
\]

where \( \mathbf{j} = \hat{n} \times \frac{1}{\mu_r} \nabla \times \mathbf{E}, \mathbf{e} = \hat{n} \times (\mathbf{E} \times \hat{n}), \mathbf{E}_{inc} \) is the incident electric field, and \( \eta = \sqrt{\mu/\epsilon} \) is the wave impedance of the material on the unbounded side of the absorbing boundary condition.
To determine the weak form of (4.13), the residual
\[ R := \nabla \times \frac{1}{\mu_r} \nabla \times E - k_0^2 \epsilon_r E + j\omega \mu_0 J \] (4.16)
is examined. It can be interpreted as a volume electric error current, giving rise to
the difference between the true solution and the numerical approximation. The weak
form is generated by choosing testing functions so as to form a proper dual pairing.
In other words, because \( R \) represents a volume electric current, it should be tested
with an electric field in order to generate a \( J \cdot E \) energy density. This results in a
Galerkin formulation, in which both the trial and testing space should be chosen from
the function space
\[ \mathbf{H}(\text{curl}, \Omega) := \left\{ v \in \mathbf{H}^0(\Omega) \mid \int_\Omega \| v \|^2 + \| \nabla \times v \|^2 dx^3 < \infty \right\} \] (4.17)
where \( \mathbf{H}^s(\Omega) \) is the Sobolev space of vector-valued functions having domain \( \Omega \), whose
derivatives up to order \( s \) are square-integrable. The resulting statement is that, for
all \( w \in \mathbf{H}(\text{curl}, \Omega) \), we seek to find \( E \in \mathbf{H}(\text{curl}, \Omega) \) such that
\[ \int_\Omega v \cdot \left( \nabla \times \frac{1}{\mu_r} \nabla \times E \right) dx^3 - k_0^2 \int_\Omega w \cdot \epsilon_r E dx^3 + j\omega \mu_0 \int_\Omega w \cdot J dx^3 = 0 \] (4.18)

\[ \left( w, \nabla \times \frac{1}{\mu_r} \nabla \times E \right)_{\Omega} - k_0^2 (w, \epsilon_r E)_\Omega + j\omega \mu_0 (w, J)_\Omega = 0. \] (4.19)

Because the testing function \( w \) is curl-conforming, we may perform integration my
parts on the first term, resulting in
\[ \left( \nabla \times w, \frac{1}{\mu_r} \nabla \times E \right)_{\Omega} + \langle w, e \rangle_{\partial \Omega} - k_0^2 \langle w, \epsilon_r E \rangle_{\Omega} - j\omega \mu_0 (w, J) = 0, \] (4.20)
where \( \langle a, b \rangle_{\partial \Omega} = \int_{\partial \Omega} a \cdot b dx^2 \). Next, (4.15) can be used to expand the surface term,
giving
\[ \left( \nabla \times w, \frac{1}{\mu_r} \nabla \times E \right)_{\Omega} + \frac{j\omega \mu_0}{\eta} \langle w, e \rangle_{\partial \Omega} - \frac{j\omega \mu_0}{\eta} \langle w, e_{\text{inc}} \rangle_{\partial \Omega} - k_0^2 \langle w, \epsilon_r E \rangle_{\Omega} - j\omega \mu_0 (w, J) = 0. \] (4.21)
By partitioning the problem domain into regions $\Omega_1$ for the bottom dielectric layer, $\Omega_2$ for the top layer, and $\Omega_3$ for the remainder, the matrix equation that results from the discretization of (4.21) can be parameterized as

\[
(S + \omega D - \omega^2 \epsilon_{r1} T_1 - \omega^2 \epsilon_{r2} T_2 - \omega^2 T_3)x = \omega b_1,
\]

which clearly demonstrated the polynomial parameter dependence common in the finite element method. If the trial and testing functions are denoted $\{v_1, v_2, \ldots, v_N\} \subset H(\text{curl}, \Omega)$, then the matrices may be expressed as

\[
S_{ij} = \left(\nabla \times v_i, \frac{1}{\mu_r} \nabla \times v_j\right)_\Omega \tag{4.23}
\]
\[
D_{ij} = \frac{j\mu_0}{\eta} (v_i, v_j)_{\partial \Omega_{abc}} \tag{4.24}
\]
\[
(T_1)_{ij} = \mu_0 \epsilon_0 (v_i, v_j)_{\Omega_1} \tag{4.25}
\]
\[
(T_2)_{ij} = \mu_0 \epsilon_0 (v_i, v_j)_{\Omega_2} \tag{4.26}
\]
\[
(T_3)_{ij} = \mu_0 \epsilon_0 (v_i, \epsilon_r v_j)_{\Omega_3} \tag{4.27}
\]

and the source vectors are

\[
(b_1)_i = -j\mu_0 \left(v_i, \frac{1}{\epsilon_{inc}} - \mathbf{e}_{inc} - \hat{n} \times \mathbf{H}_{inc}\right)_{\partial \Omega_{abc}} \tag{4.28}
\]

### 4.2.3 Numerical Results

Note that because the excitation comes entirely from the waveguide port on the coaxial cable, there is no volumetric source current, i.e., $J = 0$. The parallel FEM solver FEMSTER [37] is used to so solve the problem, using the Maxwell equation plugin [38]. The magnitude of the input reflection coefficient with $\epsilon_{r2} = 3$ and the other parameters variable is shown in Fig 4.7. As a first step, the hierarchical sampling procedure described in section 3.2.1 is used, with a tolerance of $\tau = 10^{-1}$ and $\tau = 10^{-2}$. 58
These two results, along with those of Farle and Dyczij-Edlinger [13] are shown in Fig. 4.7 and the computational statistics can be seen in Table 4.3. It can be seen that the 10% result, with 22 sample points, already captures most of the qualitative details of the main resonance. The 1% result, with an additional 17 sample, is visually indistinguishable from the 165-dimensional reduced order model. For a number of reasons, this is not as efficient as traditional model order reduction techniques. Most significantly, the combination of adaptive sampling and the black-box approach means
that the stored value of $V^T A(s)V$ must be updated after each sample point is added. Thus, the total time spend update these matrices is quadratic with respect to the number of sample points, though this effect can be mitigated by the $hp$ procedure described in section 2.2, which limits the maximum number of sample points being considered at any one time. The benefit of this approach comes from the generality, as demonstrated by the use of the same model order reduction program with several different solvers, in contrast with traditional techniques that are highly tuned to their particular application.

As discussed previously, the hierarchical sampling procedure that was initially developed has flaws, in that the sampling points are constrained to a hierarchical grid and unable to adapt to the features of the problem at hand, such as resonance points. This is illustrated in Fig. 4.8 which shows the $S_{11}$ error for the two reduced order models. $\epsilon_{r1} = 4$ and $\epsilon_{r2} = 3$. It is clear that away from the resonance points, the error has decreased by approximately one order of magnitude, as desired, by the error around the resonance point is unaffected. To address this problem, the error estimator of section 3.1.3 and optimization-based sampling of section 3.2.2 are used. To verify this procedure, we repeat the patch antenna problem, with $\epsilon_{r2} = 3$ and the other two parameters variable, and compare the error estimator to the actual
Figure 4.8: Patch antenna reflection coefficient and error with $\epsilon_{r1} = 4$, $\epsilon_{r2} = 3$
error in the solution vector. In Fig. 4.9, we see the results of the initial $3 \times 3$ grid of sample points, and it is clear that the actual and estimated error are zero at the sample points, as desired. Equally important are the three prominent ridges where the error is largest. Comparing the error plots in (a) and (c) to the actual reflection coefficient in part (b), it is clear that two of these ridges correspond to the main and higher-order resonances modes of the antenna, which the initial sample points are not able to capture. Examining part (d) suggests that the third ridge is due to a false resonance in the reduced order model. It is notable that there is considerable

Figure 4.9: Comparison of true and estimated error with 9 sample points
agreement between the estimated and actual error, in terms of both the magnitude and qualitative behavior. Figure 4.10 shows that after adding four sample points, mostly around the main and false resonances, this agreement continues. The ability to sample directly where the error is estimated to be largest, unconstrained by a fixed grid of potential sample points, has significantly improved the quality of the solution, particularly in the vicinity of the false resonance. Finally, Fig. 4.11 shows the results of adding four more sample points. Again, we see that the estimated and actual errors agree quite well, and the error continues to decrease and flatten, indicating
Figure 4.11: Comparison of true and estimated error with 17 sample points
Figure 4.12: Three-parameter patch antenna problem with improved sampling, \( \tau = 10^{-1} \), 29 samples

and absence of oversampling (as opposed to a single-point, moment-matching MOR algorithm, for example, where we would expect to see a deep valley in the error function around the sample point).

Finally, the full three-parameter problem is run using the new sampling procedure. For \( \tau = 10^{-1} \), the number of samples increases from 22 to 29. However, Fig. 4.12 shows that the accuracy is much improved, with the \( S_{11} \) less than the desired 10% for all the parameter values checked, and the true error similarly low except at a few isolated points. Fig. 4.13 shows the result with a tolerance of \( \tau = 10^{-2} \). Unfortunately these is not a great deal of improvement in the relative error, except in the vicinity of \((f, \epsilon_{r1}) = (4 \text{ GHz}, 1)\) and \((f, \epsilon_{r1}) = (6 \text{ GHz}, 7)\), suggesting that the optimizer is having difficulty locating the global maximum. The error in the reflection coefficient shows slightly better behavior, being less than 1% and the checked points, as desired.
Figure 4.13: Three-parameter patch antenna problem with improved sampling, $\tau = 10^{-2}$, 36 samples

4.3 Radome with Frequency Selective Surface

4.3.1 Problem Statement

The final example consists of a frequency selective surface, intended to form a coating on a radome. Frequency selective surfaces are periodic structures that are designed to have very different reflection and transmission characteristics at different frequencies and have found a number of practical applications since the 1960’s [2]. One, shown in Figure 4.14 consists of a parabolic reflector and two feed antennas. The first, $f_1$ is located at the focus of the parabola. The second feed is located at the focus of the hyperbolic subreflector. By making this subreflector transparent at the frequency of the first feed and reflective at the frequency of the second, one obtains a structure that is much lighter, smaller, and less expensive compared to having two separate antennas for each frequency.
A second application, and the focus of this example, is coating an aircraft radome with an FSS, as shown in Figure 4.15. The purpose of the radome is to protect the antenna from weather and the windstream while at the same time not drastically affecting the antenna’s performance. At first, radomes meet this goal by simply being as transparent as possible. Later, it was found experimentally that the antenna contributed greatly to the aircraft’s radar cross section (i.e., how large an object appears on radar). To improve the radar cross section, an FSS can be used that is transparent at the operating frequency of the aircraft’s own radar, but is opaque to the frequencies used by the enemy’s radar, thus shielding the antenna from view by hostiles.
4.3.2 Design Process and Solver for FSS Radome

To begin, the FSS is designed by assuming an array of unit cells, repeated ad infinitum in the \( x \) and \( y \) directions. For analysis purposes a single unit cell, shown in Figure 4.16 is modeled and periodic boundary conditions are applied to account for the infinite repetition. The starting point for the design is the tripole array found in [39], which has arms that are \( L = 0.16\lambda \) long and \( W = 0.07\lambda \) apart. These are the two parameters of interest, and each is allowed to vary \( \pm 0.01\lambda \). However, to account for the rectangular patterning, the spacing is slightly increased, to \( 0.40\lambda \) horizontally and \( 0.37\lambda \) vertically.

As mentioned in the circuit board problem, in order to solve the problem with different parameter values, it must be possible to automatically create meshes corresponding to the given values. Furthermore, the necessities of model order reduction
dictate that the mesh connectivity be unchanged. This problem is more complex than the former, however, because we are not merely performing linear transformations to disconnected geometries. Instead, the geometry is completely connected and the desired transformation is nonlinear (even considering a single tripole arm in isolation) because the width of the slot is to remain the same. Our approach, then, is to first mesh the geometry using the largest admissible values for $L$ and $W$, as shown in Fig. 4.17. In this way, the nodes that do not touch the slot or the PEC patch inside the slot do not need to be moved. Each arm is transformed individually by rotation by $0$, $2\pi/3$, or $-2\pi/3$, to place it in a standard position, apply a sequence of transformations, and rotating back. The inner PEC patch and the straight regions of the slot
are easily handled by translation and anisotropic scaling, to give the desired length and width for each part. The circular part of the slot is modified by performing a radial scaling with respect to each parts center. That is, for each node the original, as-meshed radius $R$ can be calculated, and the original and desired arm width, $W$ adn $W'$ are given. Then a new radius $R'$ is calculated as

$$R' = R + \frac{W - W'}{2}.$$  \hspace{1cm} (4.29)

The node’s $x$ and $y$ coordinated are then scaled by the ratio $R'/R$. This smoothly maps the interior and boundary nodes of the slot to achieve the desired arms width, while maintaining the slot width unchanged.

On the solver side, similarly to the patch antenna, the FEM is used to calculate the electric field on the interior of the FSS [40]. Now, however, various periodicity
requirements are added:

\[ \nabla \times \frac{1}{\mu_r} \nabla \times \mathbf{E} - k_0^2 \varepsilon_r \mathbf{E} = -j\omega\mu_0 \mathbf{J} \]  \hspace{1cm} (4.30)

\[ \varepsilon_r(x + mD_x \hat{x} + nD_y \hat{y}) = \varepsilon_r(x) \quad \forall m, n \in \mathbb{Z} \]  \hspace{1cm} (4.31)

\[ \mu_r(x + mD_x \hat{x} + nD_y \hat{y}) = \mu_r(x) \quad \forall m, n \in \mathbb{Z} \]  \hspace{1cm} (4.32)

Floquet’s theorem then says that if, in addition, the excitation is periodic,

\[ \mathbf{E}_{inc}(x + mD_x \hat{x} + nD_y \hat{y}) = \alpha \mathbf{E}_{inc}, \]  \hspace{1cm} (4.33)

where \( \alpha = \exp[-j k \cdot (mD_x \hat{x} + nD_y \hat{y})] \) is a constant of proportionality and \( k \) is the wavevector of the incident field, then the solutions will have the same periodicity,

\[ \mathbf{E}(x + mD_x \hat{x} + nD_y \hat{y}) = \mathbf{E}. \]  \hspace{1cm} (4.34)

In addition, a boundary integral equation is used to exactly enforce the radiation boundary condition on the open sides of the unit cell (those through which the incident field enters and the reflected and transmitted field exits). The resulting weak form and its discrete counterpart are relatively complex; details can be found in [40]. The final result is that the expression for \( A(s) \) contains a number of terms, some with polynomial dependence due to the FEM part, and others with more complicated expressions, similar to (4.8), from the BEM part.

### 4.3.3 Numerical Results

For design purposes, we would like a reduced order model that is accurate to 5%. Also, since there are two possible polarizations of the incident field and the unit cell does not have reflection symmetry across both coordinate axes, it is necessary to generate two different reduced order models, one for each polarization. In theory,
Polarization | Samples | Time (min) | Breakeven Samples
--- | --- | --- | ---
$x$ | 12 | 39.5 | 144
$y$ | 13 | 44.3 | 161

Table 4.4: Computational statistics for FSS

one could include the polarization as a third continuous parameter (e.g., the angle between the incident $E$ field and the $x$-axis), but the fact that superposition allows the solution for an arbitrary polarization to be generated from the solutions of two polarizations suggests that it may be more efficient to create two different ROMs. This is particularly true in this case, because the models are independent and can therefore be generated in parallel, thus saving wall-clock computation time. Finally, note that this example uses a serial solver with the multithreading framework described in section 3.4, an 8-core machine is used.

Figure 4.18 shows the main results of the two models. The computational statistics are presented in Table 4.4: the number of breakeven samples is based on a solve time of 2.2 minutes. Although the initial FSS design are good, with a transmission coefficient of approximately $-0.1$ dB, it can be seen that there is some room for improvement. Therefore, the trust region minimization algorithm is used to find the optimum values for $L$ and $W$, based on the objective function

$$-10 \log(T_x^2 + T_y^2),$$

(4.35)

where $T_x$ and $T_y$ are the transmission coefficients for the two polarizations. This yields the optimized values $L = 0.164\lambda$ and $W = 0.068\lambda$. The procedure takes 9 steps to find the minimum, a process that would take at least 19.8 minutes if the full-order solver were used. Although this is shorter than the time taken to generate the reduced
Figure 4.18: Results for first-stage FSS design: (a) $x$-polarized transmission coefficient (b) sample points for $x$-polarized model (c) $y$-polarized transmission coefficient (d) sample points for $y$-polarized model.
order models, it should be noted that the they provide additional benefits beyond this one-off optimization, such as the ability to investigate the global behavior of the transmission, revealing, for example, that the transmission coefficient is significantly more sensitive to the tripole arm length than to the width.

As a further study of the error estimator, we again compare the error estimator and true error for the two models. Figure 4.19 shows the results of the \( x \)-polarization model after 9, 11, and 12 samples. Initially there is substantial agreement between the estimated and actual errors. However, after 10 (not shown) and 11 samples, this agreement is mostly gone, only to reappear after the 12th sample point is added. This suggests that more workkk is necessary to develop a more rigorous estimator. A different situation is illustrated in Fig. 4.20 which shows the comparison for the \( y \)-polarization model. In this case, there is substantial agreement between the estimated and true error, but the optimization procedure has falsely indicated convergence, finding only a local maximum in the vicinity of \( s = (3.3 \text{ mm}, 1.35 \text{ mm}) \). It has missed regions of greater estimated error in the upper- and lower-right corners. Examining the program’s log file reveals the cause: the small number of random points (30) examined for possible optimizer seeds was insufficient to place a point close enough to the 0.05 regions. Since the error estimator can be rapidly evaluated, however, there is very little penalty in increasing the number of potential seeds by an order of magnitude (furthermore, for robustness the number should at least depend on the number of parameters, so that the spacing remains relatively constant).

Making this change, the error estimator finds the maximum in the upper-right corner, but the log file reveals another interesting phenomenon. Unlike the previous iterations, the error estimator’s gradient does not approach zero, and the optimizer
Figure 4.19: $x$-polarization reduced order model: (a) estimated and (b) actual error after 9 samples, (c) estimated and (d) actual error after 11 samples, (e) estimated and (f) actual error after 12 samples.
Figure 4.20: $y$-polarization reduced order model: (a) estimated and (b) actual error after 11 samples, (c) estimated and (d) actual error after 12 samples, (e) estimated and (f) actual error after 13 samples.
instead terminates after reaching the maximum number of steps. It is also quite curious that the estimated error is 99%, considering that Fig. 4.20 shows that the difference between the true solution and the reduced order model with $V^TAV$ is less than 5%. This suggests that the problem is related to the matrix $W^TAV$ or its interpolation. To investigate this, the condition number of $\hat{B}$, the RBF interpolation of $W^TAV$ is examined, both before and after adding a 14th sample point in the upper-right corner. Figure 4.21 shows that the error estimator does indeed peak in the region where the condition number is largest, indicating that the reduced order solution is particularly sensitive to an errors in calculating or interpolating $W^TAV$ or the right-hand side. Furthermore, it can be seen that even after a sample point is added, the resulting interpolation can still generate poorly-conditioned matrices nearby. Therefore, it would be advantageous to modify the matrix interpolation procedure so that the conditioning of the result is less sensitive to interpolation errors. One possibility, used in [41], is to apply a transformation so that the interpolation always has the desired property, in this case invertibility. This could be achieved, for examples, by interpolating the matrix logarithm of $\tilde{A}$ and $\tilde{B}$. After evaluating the RBF interpolation and taking the matrix exponential, the result is guaranteed to be nonsingular.

To further examine the performance of the design, and verify the accuracy of the periodic boundary condition solver, the FSS is simulated across a range of frequencies for several incident angles, and the results are compared to HFSS. The results can be seen in Fig. 4.22. There are two conclusions from these results. The first is the excellent agreement between the two solvers. The second is the considerable angle dependence on the bandwidth of the FSS filter, a characteristic common to FSSs that
consist only of a thin PEC sheet, with no dielectric backing on either side. The usual remedy to this situation is to apply a layer of dielectric material to one or both sides of the PEC. To this end, we investigate the effect of adding one or two layers of dielectric glass with \( \epsilon_r = 4.8 - j0.0096 \). Because of the large magnitude of the real part, it is necessary to scale the design in order to maintain the resonance frequency at 14 GHz. Therefore, all of the dimensions of the unit cell except for the slot width are scaled by a factor of 0.5, as shown in Fig. 4.23, and different dielectric thickness are examined to find the optimum value. Figure 4.24(a) shows that the two-layer design, with a total thickness of 0.7 mm provides optimum transmission at normal incidence. Furthermore, Fig. 4.24(b) demonstrates that the transmission at 14 GHz remains high for both polarizations as the incident angle changes.
Figure 4.21: $y$-polarization reduced order model: (a) estimated error and (b) condition number of $\hat{B}$ with 13 samples, (c) estimated error and (d) condition number of $\hat{B}$ with 12 samples (note change in horizontal axis).
Figure 4.22: Frequency sweep of periodic boundary condition FEBI code for several incident angles, compared to HFSS results. $\theta = 0$ incident field, (a) FEBI and (b) HFSS; $\theta = \pi/6$ incident field, (c) FEBI and (d) HFSS; $\theta = 2\pi/3$ incident field, (e) FEBI and (f) HFSS.
Figure 4.23: Geometry and mesh of scaled unit cell with dielectric layer

\[ \varepsilon_r = 4.8 - j0.0096 \]

on one (-z) or both sides of FSS

free space
Figure 4.24: Performance of dielectric-backed FSS. (a) peak transmission versus dielectric thickness (b) angular dependence of two-layer dielectric with thickness 0.7 mm.
Chapter 5: Conclusion

This dissertation has presented a black-box model order reduction algorithm, applicable to a wide variety of parameterized linear systems, by formulating the model generation procedure in terms of simple linear algebra operations. These systems, in turn, represented a range of physical problems, such as electromagnetic scattering and radiation, formulated with numerous methods, such as finite elements and integral equations. The cost of this abstraction, however, was the loss of information about exactly how the linear system of interest depended on the parameters. Radial basis functions were used to interpolate the reduced order matrices $V^TAV$, thus allowing the dependence to be approximated and permitting simple extension to multivariable cases. Unfortunately, it was found that RBFs are susceptible to instability. A two-step process was developed to mitigate this possibility. First, $hp$ refinement was used to limit the number of basis functions under consideration at any one time. Second, singular value decomposition-based regularization was used to find, if necessary, and use a linearly independent subset of the RBFs.

Equally important, it was shown how, with the use of an error estimator, the interpolation process can be made adaptive. Thus, the user of the developed MOR program is not forced to choose the sample points a priori, but instead need only specify the desired accuracy of the reduced order model. Numerical examples showed
that the resulting MOR program is capable, unmodified, of generating accurate and efficient reduced order models for radiation and scattering problems using a variety of formulations.

Looking forward, there are a several avenues for further improvement. Although polyharmonic splines were used for this work, there are of course many different radial basis functions, and the ability use the shape tuning parameter \( c \) may make some of them more versatile. Beyond RBFs, there are also other promising interpolation methods for scattered data. Of particular note, statistical techniques such as design and analysis of computer experiments (DACE) and kriging are attractive because they come with their own uncertainty estimates. This makes it possible to easily separate the problems of sampling to improve the reduced order basis \( V \) and sampling to improve the interpolation of the reduced order matrix \( \tilde{A} \). In this work the two have been intertwined: sampling in parameter space adds one basis function to \( V \) and one interpolation point for \( \tilde{A} \), but they are actually separate approximation problems.

In the subject of interpolation, we also mention an emerging technique, the empirical interpolation method (EIM) [42–45]. Recall that one of the motivations for interpolating the reduced order matrices is the lack of an affine decomposition, that is, a sum of terms consisting of a constant times a known function depending only the parameters. EIM seeks to automatically create such a decomposition that approximates the original function dependence in an error-controllable manner. For example, consider the EFIE, whose matrix entries contained the Green function

\[
\frac{e^{jk\|x-x'\|}}{4\pi\|x-x'\|} = \frac{e^{-jkR}}{4\pi R} = G(k;R),
\]

with \( k = \omega \sqrt{\mu \epsilon} \) as the only parameter. The difficulty for traditional reduced order methods is the presence of both a parameter \( k \) and another variable \( R \) inside the
exponential. EIM, in contrast, will generate an approximation of the form

\[ G(k; R) \approx \sum_i \alpha_i(k)q_i(R). \] (5.2)

By replacing the Green function with the \( q_i \) and integrating out the \( R \) dependence, a set of constant matrices \( Z_1, Z_2, \ldots \) is obtained. For a given \( k \) value, the impedance matrix can then be approximated as

\[ Z(k) \approx \sum_i \alpha_i(k)Z_i. \] (5.3)

The coefficient functions \( \alpha \) are determined by solving the linear system \( q_j(R_i)\alpha_j(k) = G(k, R_i) \), where the \( R_i \) are a set of sample points determined by the EIM algorithm. These, and the \( q_i \), are generated with a greedy sampling procedure. The \( N \)-th function is generated by first calculating the error function using the \( N - 1 \) previous samples,

\[ e_N(k; R) = G(k; R) - \sum_{i=1}^{N-1} \alpha_i(k)q_i(R). \] (5.4)

Then the sample points are determined by a two-step optimization,

\[ k_N = \arg\sup_k \sup_R |e_N(k, R)| \] (5.5)
\[ R_N = \arg\sup_R |e_N(k_N, R)|. \] (5.6)

The new basis function is then \( q_N(k) = e_N(k, R_N)/|e_N(k_N, R_N)| \). This process continues until the peak of the error function is less than a user-specified tolerance, resulting in a seperable expansion that interpolates the original non-affine function at the sample points.

Finally, we note that the error estimators used in this work were heuristic in nature. there is significant room for improvement in this area and a tremendous need, in this application and others, for efficient and rigorous error estimation.
Bibliography


88


90