ADVANCED TECHNIQUES IN THE COMPUTATION OF REDUCED ORDER MODELS AND KRYLOV RECYCLING FOR DIFFUSE OPTICAL TOMOGRAPHY

by

Meghan Jane O'Connell

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Mathematics

TUFTS UNIVERSITY

May 2016

Advisor: Misha Elena Kilmer

Abstract

Nonlinear parametric inverse problems whose forward model is described by a partial differential equation (PDE) arise in many applications, such as diffuse optical tomography (DOT). The main computational bottleneck in solving these types of inverse problems is the need to repeatedly solve the forward model, which requires solves of large-scale discretized parametrized PDEs. The main focus of this thesis is developing methods to reduce this cost.

In the context of absorption imaging in DOT, interpolatory model reduction can be employed to reduce the computational cost associated with the forward model solves. We use surrogate models to approximate both the function evaluations and the Jacobian evaluations, which significantly reduces the cost while maintaining accuracy.

We consider two methods for construction of the global basis required for the reduced model. Both methods require several full order model solves. The first method solves the fully discretized PDE for multiple right-hand sides and then uses a rank-revealing factorization to compress the basis. The second method reduces the cost of the construction of the global basis in two ways. First, we show how we exploit the structure of the matrix to rewrite the full order transfer function and corresponding derivatives in terms of a symmetric matrix. We then apply model order reduction to the new symmetric formulation of the problem. Second, we give an inner-outer Krylov approach to dynamically build the global basis while the full order systems are solved. This means that we only update the global basis with the incrementally new, relevant information eliminating the need to do an expensive rank-revealing factorization. Next, we extend the inner-outer Krylov recycling approach to solving sequences of shifted linear systems.

We show the value of the above approaches with 2-dimensional and 3-dimensional examples from DOT, however, we believe our methods have the potential to be useful for other applications as well.

In the final chapter, we explore different approaches to constructing the recycle spaces for shifted systems. We show how the use of generalized eigenvectors has the potential to be extremely useful for large shifts.

for my family, friends, and mentors.

Acknowledgments

First and foremost, I would like to thank my advisor, Misha Kilmer. I am profoundly grateful for her support, her encouragement, her mentorship, and the hours of explanations and discussions. Her passion for her research and her students motivated and inspired me throughout my time at Tufts.

I would like to thank the many professors who have taught and supported me throughout the years, specifically James Adler, Eric de Sturler, Serkan Gugercin, Xiaozhe Hu, and Scott MacLachlan for their mentorship and helpful discussions. I would also like to thank the mentors I have had at various internships, specifically Pat Quillen and Penny Anderson who encouraged me to go to graduate school and continue to offer support and advice.

Next, I would like to thank my officemates past and present for their encouragement and helpful discussions about my research. I also thank all my friends for always listening, bringing balance to my life, offering moral support, and finally for drinking tea with me.

Last but not least, I would like to thank my family for their never ending love and support. From listening to me talk about my research even though they have no idea what I am talking about to home cooked meals and everything in between, I would never have made it without you all!

Contents

List of Tables

List of Figures

Advanced Techniques in the Computation of Reduced Order Models and Krylov Recycling for Diffuse Optical Tomography

Chapter 1

Introduction

Inverse problems arise in many important applications such as medical image reconstruction [14, 24, 48, 78], geophysics [66, 82], and groundwater imaging [69, 79]. The goal of solving an inverse problem is to reconstruct an unknown input, given measured output data and a forward model. A forward model is a mathematical model that relates the unknown input to the known output data. The model can be linear, such as in X-ray tomography, image deblurring, and geophysics, or nonlinear, such as in electrical impedance tomography (EIT). In either case, solving the inverse problem requires the solution to an optimization problem for the unknown values that define the unknown quantity of interest. To solve the optimization problem requires possibly repeated calls to the forward model. Executing a forward model solve usually means solving a large-scale linear system.

The forward model solve of large-scale nonlinear inverse problems, which needs to be solved many times, can therefore be the computational bottleneck for solving the inverse problem. Reducing this cost by developing efficient methods to solve the forward problem and ultimately the inverse problem is the focus of this work.

One specific application in medical image reconstruction is diffuse optical tomography (DOT). DOT exemplifies the type of nonlinear inverse problem we want to consider, but other imaging problems such as EIT or electrical resistivity tomography (ERT) have a similar structure, and the methods developed in this thesis can be extended to those as well. In this work, we focus on developing a class of model reduction and Krylov subspace methods to address the computational bottleneck outlined above.

1.1 Thesis Outline

The structure of this thesis is as follows. In Chapter 2 background on inverse problems is given. This includes background on inverse problems in general, the problem setup for DOT, and the parametric level set method that is adopted in the context of the DOT problem to reduce the search space and offer implicit regularization. More background is provided in Chapter 3, specifically on Krylov Methods. We give the definition of a Krylov subspace, provide the details of the generalized minimal residual method as an example of a Krylov method, and also explain the basics of Krylov recycling.

In Chapter 4 we explain model order reduction in the context of the DOT problem. We begin by providing background on interpolatory parametric model reduction and explain how this is adopted for the DOT problem. We give a method for the construction of the global basis, which involves solving several full order systems and using a rank-revealing factorization. Analysis and numerical results conclude the chapter.

Next, in Chapter 5 we continue to study the construction of the global basis for reduced order modeling in DOT. We begin by showing how to restate the problem in terms of a symmetric positive definite matrix. Then, we give an algorithm for inner-outer Krylov recycling that not only solves the required full order model solves, but also constructs the global basis for the reduced model eliminating the need for a rank-revealing factorization. Algorithm analysis and numerical results are also given.

We further extend our inner-outer recycling method to shifted systems in Chapter 6. First, we consider shifted systems with a single right-hand side and then we consider shifted systems with multiple right-hand sides. The inner-outer Krylov recycling algorithm for shifted systems with multiple right-hand sides again allows us to construct the global basis during the full order model solves. Analysis and numerical results are also provided.

In Chapter 7, we explore different ways of constructing the recycle spaces used for inner-outer Krylov recycling for shifted systems. We consider different invariant subspaces that improve convergence.

Finally, concluding remarks are given in Chapter 8 and future work is provided in Chapter 9.

Chapter 2

Inverse Problems Background

2.1 Inverse Problems

The two main components of an inverse problem are the measured output data and the forward model. The forward model, $\mathcal{M}$, is a mathematical model that takes the unknown input, $\mathbf{f}$, and generates an observation vector, $\mathbf{u}$, as

$$\mathbf{u} = \mathcal{M}(\mathbf{f}). \tag{2.1}$$

In general, $\mathbf{u} \in \mathbb{C}^k$, where k is the number of measurements. The inverse problem refers to finding $\mathbf{f}$ given $\mathcal{M}$ and the observed data, $\mathbf{u}$, according to

$$\min_{\mathbf{f}} \|\mathbf{u} - \mathcal{M}(\mathbf{f})\|_2. \tag{2.2}$$

The forward model, $\mathcal{M}$, can be linear, but it is often nonlinear. When $\mathcal{M}$ is nonlinear that means that solving (2.2) requires solving a nonlinear least squares problem. In particular, updating the estimate of $\mathbf{f}$ requires executing $\mathcal{M}(\mathbf{f})$ many times.

It is well known that inverse problems are ill-posed, which means that a solution to the problem either does not exist, is not unique, or is not a continuous function of the data [39]. Thus, inverse problems are highly sensitive to noise in the data and are often underdetermined. Regularization is used to stabilize the inverse problem by damping the noise and/or forcing a unique solution. This is accomplished by using prior knowledge about the data or the problem. If explicit regularization is used, the problem becomes

$$\min_{\mathbf{f}} \{\|\mathbf{u} - \mathcal{M}(\mathbf{f})\|_2 + \mathcal{L}(\mathbf{f})\}. \tag{2.3}$$

Tikhonov and total variation regularization are widely used choices for $\mathcal{L}(\mathbf{f})$. For more details and other methods of regularization the reader is encouraged to see [2,30,70,76]. The parametric level set (PaLS) [3] approach, described in Section 2.3, for the DOT problem does not require the additional regularization term because the regularization is provided directly through the PaLS image model. In the following sections, we will describe DOT and PaLS in more detail.

2.2 Diffuse Optical Tomography

One specific example of a nonlinear inverse problem is diffuse optical tomography (DOT) [14]. A main application of DOT is breast tissue imaging. The inverse problem involves using the data captured at the detectors by illuminating the tissue with signal sources to determine optical properties of the tissue. The tissue is illuminated with near-infrared light, which is less harmful than standard imaging with X-rays. The light is transmitted and detected in arrays on each side of the tissue. Since tumors have different optical properties than normal tissue, we recover images of optical properties, such as absorption or diffusion. Figure 2.1 shows the basic set-up for DOT imaging.

The forward model for DOT is now discussed, which follows from [14]. We consider the region to be imaged as a rectangular slab, $\Omega = [a_1, b_1] \times [a_2, b_2] \times [a_3, b_3]$. The top surface, denoted $\partial\Omega_+$, is where sources and detectors are located, while the bottom surface, denoted $\partial\Omega_-$, only has detectors. We assume that both the sources and detectors are stationary and that we have n_{src} number of sources and n_{det} number of detectors. The spatial location will be denoted by $\mathbf{x} = (x_1, x_2, x_3)^T$. The input source, $g(\mathbf{x}, t)$, is defined by functions $b_j(\mathbf{x})$, which describe the transmittance field, and $u_j(t)$, which gives the pulse profile, such that $g(\mathbf{x}, t) = b_j(\mathbf{x}) u_j(t)$ for $j = 1, \ldots, n_{src}$. The observations made at the detectors, $m_i(t)$, are defined by functions, $c_i(\mathbf{x})$, which describe the response of the sensors, and $\eta(\mathbf{x}, t)$, which describes the photon flux/fluence, such that $m_i(t) = \int_{\partial\Omega} c_i(\mathbf{x}) \eta(\mathbf{x}, t)\, d\mathbf{x}$ for $i = 1, \ldots, n_{det}$. Additionally, we let $D(\mathbf{x})$ and $\mu(\mathbf{x})$ denote diffusion and absorption coefficients, $\mathcal{A}$ denote a constant related to diffusive boundary reflection (see [14, p. R50]), ξ denote

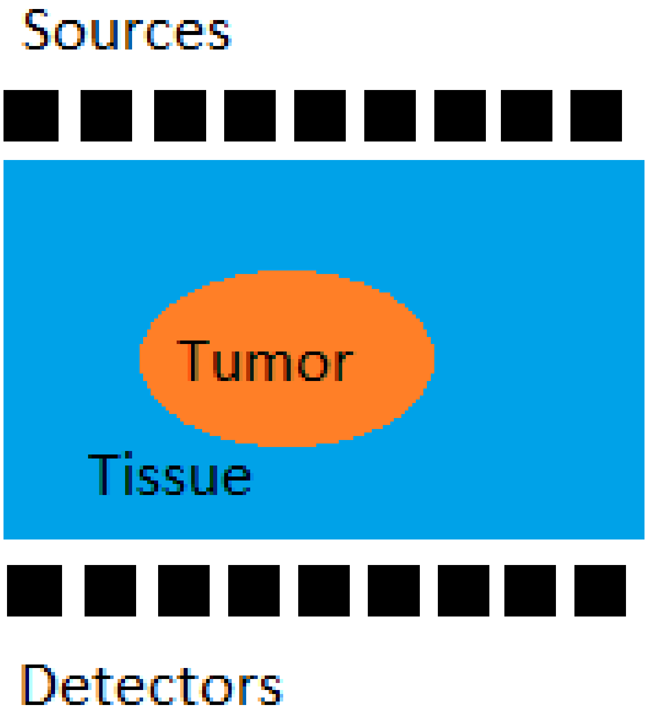


Figure 2.1: Schematic of the DOT problem.

the outward unit normal, and ν be the speed of light in the medium. We then model the photon flux/fluence, $\eta(\mathbf{x},t)$ with a time-domain diffusion model,

$$\frac{1}{\nu}\frac{\partial}{\partial t}\eta(\mathbf{x},t) = \nabla\cdot(\,D(\mathbf{x})\nabla\eta(\mathbf{x},t)\,) - \mu(\mathbf{x})\eta(\mathbf{x},t) + b_j(\mathbf{x})u_j(t), \quad \text{for } \mathbf{x}\in\Omega, \tag{2.4}$$

$$0 = \eta(\mathbf{x},t) + 2\,\mathcal{A}\,D(\mathbf{x})\,\frac{\partial}{\partial\xi}\eta(\mathbf{x},t), \quad \text{for } \mathbf{x}\in\partial\Omega_\pm, \tag{2.5}$$

$$0 = \eta(\mathbf{x},t), \quad \text{for } x_1 = a_1 \text{ or } x_1 = b_1 \text{ or } x_2 = a_2 \text{ or } x_2 = b_2, \tag{2.6}$$

$$m_i(t) = \int_{\partial\Omega} c_i(\mathbf{x})\eta(\mathbf{x},t)\,d\mathbf{x} \quad \text{for } i = 1,\ldots,n_{det} \tag{2.7}$$

(see [14, p. R56]). It should be noted that while this PDE is linear, the forward model is nonlinear since it takes the absorption, $\mu(\mathbf{x})$, and maps it to output.

While we could recover images of $D(\mathbf{x})$ and $\mu(\mathbf{x})$, we assume that $D(\mathbf{x})$ is known and attempt to recover $\mu(\mathbf{x})$. Therefore, the inverse problem involves using the data, $\boldsymbol{m}(t)$, captured at the detectors by illuminating the tissue with signal sources, $\mathbf{u}(t)$, to determine $\mu(\mathbf{x})$. In order to reduce our search space, we will adopt the PaLs approach, as in [3], to parametrize the absorption field, $\mu(\cdot)$. This means that we assume that we can express $\mu(\cdot)$ in terms of a finite set of parameters, $\mathbf{p} = [p_1,\ldots,p_{n_p}]^T$, giving us $\mu(\cdot) = \mu(\cdot,\mathbf{p})$. More on the specifics of the PaLs approach is found in Section 2.3.

The discretization of (2.4)-(2.7) and problem setup will be discussed in more detail in Section 4.2. The optimization problem we ultimately want to solve is

$$\min_{\mathbf{p}\in\mathbb{R}^{n_p}} \|\mathcal{M}(\mathbf{p}) - \mathbb{D}\|_2. \tag{2.8}$$

where $\mathcal{M}(\mathbf{p})$ is the synthetic data generated by the forward model for parameter vector $\mathbf{p}$ and $\mathbb{D}$ is the acquired data. We solve the optimization problem using the trust region algorithm with regularized model solution (TREGS) as in [44], although other nonlinear least squares methods could also be used.

2.3 Parametric Level Set Methods

Parametric level set methods fall under the umbrella of shape-based approaches. For many inverse problems, it makes sense to think about the property you are inverting being comprised of two (or a small number of) classes [3, 30, 45, 51, 63]. In the DOT setting, this means we can think about breast tissue as regions of tumor and regions of non-tumor having constant absorption within those regions. Then the goal is to find the boundary between the two regions. We will first discuss shape-based approaches in general and then parametric level set methods.

2.3.1 Shape-based Approaches

Many methods for traditional level set approaches have been developed , see [23, 30, 54, 63, 72–74]. While these methods have proved useful, they still require regularization of the inverse problem to handle the ill-posedness of noise in the problem. If the image we reconstruct only has two types of classes, then we can use characteristic functions in the following way. Given a closed domain $D \subseteq \Omega$, we define the characteristic function χ_D as

$$\chi_D(\mathbf{x}) = \begin{cases} 1 & \mathbf{x} \in D \\ 0 & \mathbf{x} \in \Omega \setminus D. \end{cases} \tag{2.9}$$

Now, we define the unknown image f over Ω as

$$f(\mathbf{x}) = f_i(\mathbf{x})\chi_D(\mathbf{x}) + f_o(\mathbf{x})(1 - \chi_D(\mathbf{x})) \tag{2.10}$$

where f_i indicates the property values inside D and f_o indicates the values outside. The boundary between the classes, ∂D, is represented as a level set of a Lipschitz continuous function, $\phi : \Omega \to \mathbb{R}$. If we consider the zero level set, we get

$$\begin{cases} \phi(\mathbf{x}) > 0 & \forall \mathbf{x} \in D \\ \phi(\mathbf{x}) = 0 & \forall \mathbf{x} \in \partial D \\ \phi(\mathbf{x}) < 0 & \forall \mathbf{x} \in \Omega \smallsetminus D. \end{cases} \tag{2.11}$$

Figure 2.2 shows an example of the different images one can get when ϕ is sliced using different level sets, in particular, the c level set. We then use a Heaviside function, $H(r) = \frac{1}{2}(1 + sign(r))$, to redefine our image, $f(\mathbf{x})$:

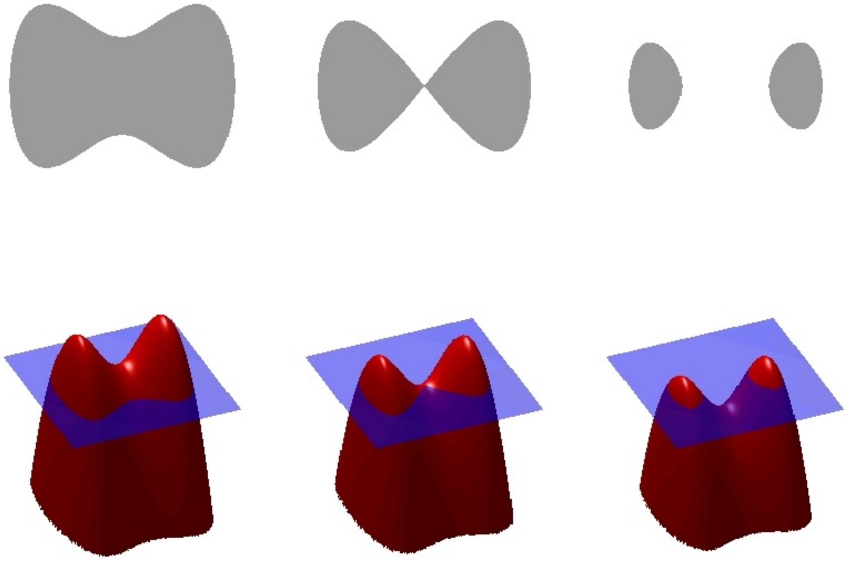


Figure 2.2: An example of the different images one can get when using different heights for the level set. Figure from https://en.wikipedia.org/wiki/Level_set_method.

$$f(\mathbf{x}) = f_i(\mathbf{x})H(\phi(\mathbf{x}) - c) + f_o(\mathbf{x})(1 - H(\phi(\mathbf{x}) - c)) \tag{2.12}$$

$$= \begin{cases} f_i(\mathbf{x}) & \phi(\mathbf{x}) \geq c \\ f_o(\mathbf{x}) & \phi(\mathbf{x}) < c, \end{cases} \tag{2.13}$$

which maps the space of unknown regions D into the space of unknown smooth functions ϕ.

2.3.2 PaLS

As was shown in [3], parametric level set methods have proved beneficial for many inverse problems. While traditional level methods require additional regularization, PaLS provides implicit regularization in addition to reducing the search space. PaLS follows the same approach as traditional level set methods, but we define the continuous Lipschitz function, $\phi : \Omega \times \mathbb{R}^m \to \mathbb{R}$, as a function of $\mathbf{x}$ and a parameter vector $\mathbf{p} = (p_1, p_2, \cdots, p_m) \in \mathbb{R}^m$,

$$\begin{cases} \phi(\mathbf{x}, \mathbf{p}) > c & \forall \mathbf{x} \in D \\ \phi(\mathbf{x}, \mathbf{p}) = c & \forall \mathbf{x} \in \partial D \\ \phi(\mathbf{x}, \mathbf{p}) < c & \forall \mathbf{x} \in \Omega \setminus D \end{cases} \tag{2.14}$$

for $c \in \mathbb{R}$. Specifically, we let

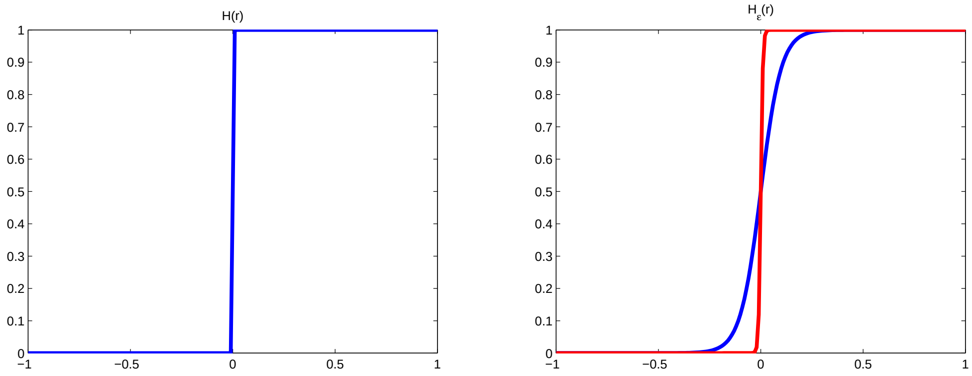
$$\phi(\mathbf{x}, \mathbf{p}) = \sum_{j=1}^{m_0} \alpha_j \psi_j \tag{2.15}$$

where $\psi_j := \psi(||\beta_j(\mathbf{x} - \boldsymbol{\chi}_j)||^\dagger)$ and the unknown parameter vector, $\mathbf{p}$, is comprised of scalars α_j, dilation factors β_j, and center locations $\boldsymbol{\chi}_j$. A smooth approximation of the Euclidean norm is used $||\mathbf{x}||^\dagger := \sqrt{||\mathbf{x}||_2^2 + \nu^2}$ where $\nu \neq 0$ is a small real number. Now, the image is described as

$$f(\mathbf{x}, \mathbf{p}) = f_i(\mathbf{x}) H_\epsilon(\phi(\mathbf{x}, \mathbf{p}) - c) + f_o(\mathbf{x})(1 - H_\epsilon(\phi(\mathbf{x}, \mathbf{p}) - c)), \tag{2.16}$$

where $H_\epsilon(r)$ denotes a continuous approximation to the Heaviside function, e.g. see [81], and c is the height of the level set. We then have to solve for the parameter vector $\mathbf{p}$. Figure 2.3 shows plots of the Heaviside function as well as smooth approximations to the Heaviside function.

The $\phi(\mathbf{x}, \mathbf{p})$ functions should be a linear combination of ψ_j functions, where the ψ_j functions have compact support, are in C^2, have radial symmetry, and create "bumps". As in [3], we use compactly supported radial basis functions (CSRBFs). Figure 2.4 shows an example of a CSRBF with $\psi(r) = \{(\max(0, 1 - r))^2(2r + 1)\}$ with $r = \sqrt{x^2 + y^2}$. An advantage of the PaLS approach is that we get high curvature geometries with a limited number of bumps since the interaction of bumps with



(a) Plot of Heaviside function.

(b) Plot of two examples of continuous approximations to the Heaviside function.

Figure 2.3: Plot of the Heaviside function and two graphical representations of $H_\epsilon(r)$, where the width of the transition is defined by ϵ.

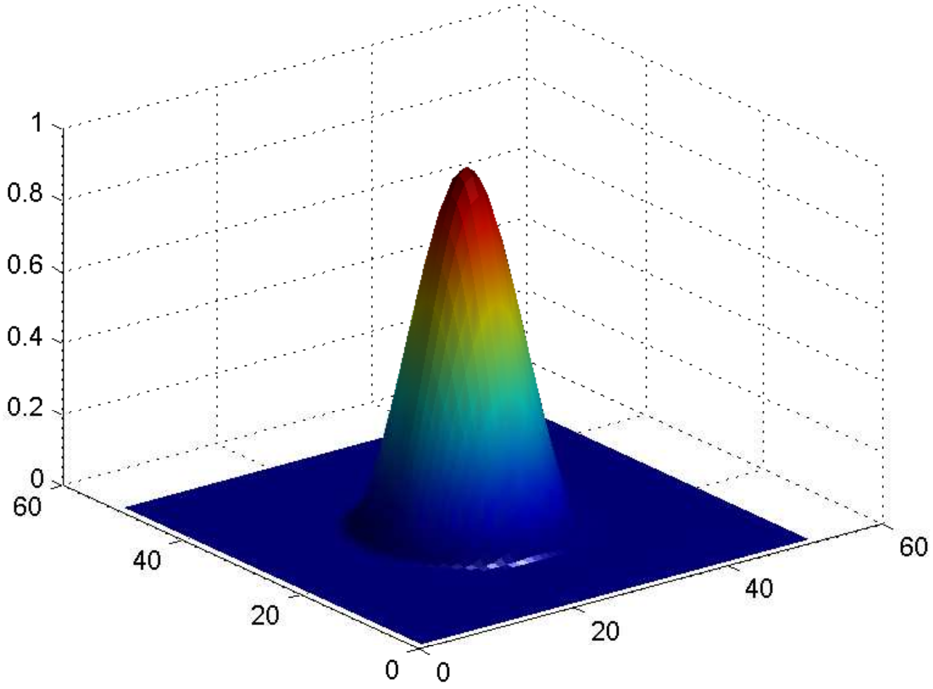


Figure 2.4: The CSRBF given by $\psi(r) = \{(\max(0,1-r))^2(2r+1)\}$ with $r = \sqrt{x^2+y^2}$.

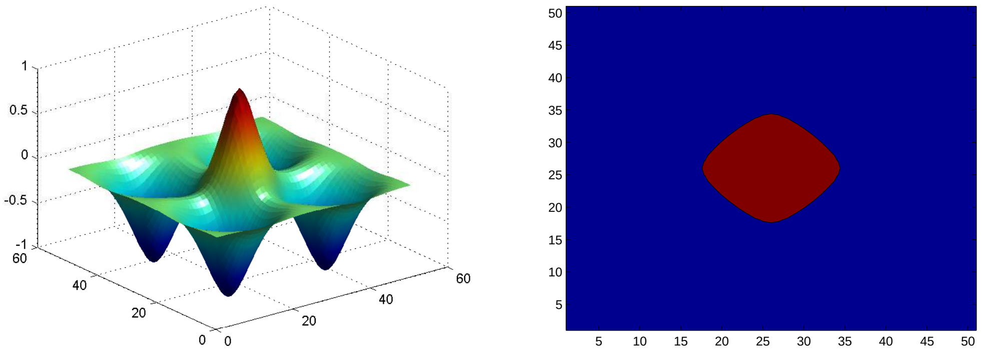


Figure 2.5: CSRBF example with opposite sign ψ functions and the image recovered with the zero level set.

opposite signs create edges. The compact support can imply that we also have sparse updates in the optimization problem. Figure 2.5 shows another example of a CSRBF and the image created with the zero level set.

Chapter 3

Solver Background

The need to solve large scale sequences of shifted linear systems with multiple right-hand sides arises in many important applications, nonlinear inverse problems among them. These systems have the following form,

$$\left(\mathbf{A}^{(k)} + \gamma_\ell \mathbf{E}\right) \mathbf{X}^{(k,\ell)} = \mathbf{B}. \tag{3.1}$$

The expense of solving these systems can be the computational bottleneck of the larger problem in which they are involved. Many approaches have been developed to address the computational cost associated with solving sequences of shifted systems. Some methods use Lanczos recurrences or Arnoldi iterations, see e.g., [26, 43, 49, 61, 64]. The work presented in this thesis focuses on designing new iterative approaches based on Krylov subspaces to solve large scale sequences of shifted linear systems with multiple right-hand sides, therefore we now provide the reader with a background on Krylov subspace methods. In this section, we explore ways to solve systems of the form

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{3.2}$$

for $\mathbf{A} \in \mathbb{C}^{n\times n}$ and $\mathbf{b} \in \mathbb{C}^n$ via Krylov subspace methods.

3.1 Krylov Subspaces

The k^{th} Krylov subspace generated by square $\mathbf{A}$ and $\mathbf{b}$ is

$$\mathcal{K}_k(\mathbf{A}, \mathbf{b}) = span\{\mathbf{b}, \mathbf{A}\mathbf{b}, \mathbf{A}^2\mathbf{b}, \dots, \mathbf{A}^{k-1}\mathbf{b}\}. \tag{3.3}$$

This has dimension less than or equal to k. Krylov subspaces have the property that they are shift invariant, meaning $\mathcal{K}_j(\mathbf{A}, \mathbf{b}) = \mathcal{K}_j(\mathbf{A} + \gamma\mathbf{I}, \mathbf{b})$. Many iterative methods for solving large systems of linear systems utilize Krylov subspaces, such as the conjugate gradient method and GMRES or MINRES. At iteration k of a Krylov method, we get an approximate solution, $\mathbf{x}_k$, from the Krylov space $\mathcal{K}_k(\mathbf{A}, \mathbf{b})$. In the next section, a brief introduction to GMRES is given as an example of a Krylov method.

3.2 GMRES

The generalized minimal residual method (GMRES) [60] is given as a Krylov subspace method example. As stated above, at the k^{th} iteration, GMRES looks for the solution $\mathbf{x}_k$ in $\mathcal{K}_k(\mathbf{A}, \mathbf{b})$ that minimizes the residual. This means GMRES looks for the $\mathbf{x}_k$ that solves the following least squares problem

$$\min_{\mathbf{z} \in \mathcal{K}_k(\mathbf{A}, \mathbf{b})} \|\mathbf{b} - \mathbf{A}\mathbf{z}\|_2. \tag{3.4}$$

We assume here that $\mathbf{x}_0 = \mathbf{0}$, although non-zero initial guesses are possible. To solve (3.4), we need a basis for the Krylov subspace. This is accomplished by using the Arnoldi method [13] to construct an orthonormal basis for $\mathcal{K}_k(\mathbf{A}, \mathbf{b})$, which will be represented by $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$. To start, define $\mathbf{v}_1 = \frac{\mathbf{b}}{\|\mathbf{b}\|}$. Arnoldi proceeds in a Gram-Schmidt fashion by letting $\hat{\mathbf{v}}_{j+1} = \mathbf{A}\mathbf{v}_j - (\mathbf{h}_{1j}\mathbf{v}_1 + \cdots + \mathbf{h}_{jj}\mathbf{v}_j)$, where $\mathbf{h}_{ij} = \mathbf{v}_i^* \mathbf{A}\mathbf{v}_j$, and $*$ denotes the conjugate transpose. The vector is then normalized to get the final new basis vector, $\mathbf{v}_{j+1} = \frac{\hat{\mathbf{v}}_{j+1}}{\|\hat{\mathbf{v}}_{j+1}\|}$. Define $\mathbf{V}_j = [\mathbf{v}_1, \cdots, \mathbf{v}_j]$. Arnoldi gives the following relation,

$$\mathbf{A}\mathbf{V}_j = \mathbf{V}_{j+1}\mathbf{H}_j, \tag{3.5}$$

where $\mathbf{H}_j$ is a $(j+1) \times j$ upper Hessenberg matrix.

Turning our attention back to the least squares problem in (3.4), if we want $\mathbf{z} \in \mathcal{K}_k(\mathbf{A}, \mathbf{b})$, this is equivalent to $\mathbf{z} = \mathbf{V}_k\mathbf{y}$ for some $\mathbf{y}$. Plugging $\mathbf{z}$ in for $\mathbf{x}$ we get $\mathbf{A}\mathbf{z} = \mathbf{A}\mathbf{V}_k\mathbf{y} = \mathbf{V}_{k+1}\mathbf{H}_k\mathbf{y}$ and $\mathbf{b} = \beta\mathbf{v}_1 = \beta\mathbf{V}_{k+1}\mathbf{e}_1$, where $\mathbf{e}_1$ denotes the first Cartesian basis vector in $\mathbb{R}^{k+1}$ and $\beta = \|\mathbf{b}\|_2$. The least squares problem in (3.4) then becomes

$$\min_{\mathbf{z}\in\mathcal{K}_k(\mathbf{A},\mathbf{b})} \|\mathbf{b}-\mathbf{A}\mathbf{z}\|_2 = \min_{\mathbf{y}} \|\beta\mathbf{e}_1 - \mathbf{H}_k\mathbf{y}\|_2. \tag{3.6}$$

In theory, if the $(k+1)^{\text{st}}$ column $\mathbf{A}^k\mathbf{b}$ is dependent on $\{\mathbf{A}^j\mathbf{b}\}_{j=0}^{k-1}$, then GMRES stops iterating because $\hat{\mathbf{v}}_{j+1} = \mathbf{0}$. In practice, GMRES is usually stopped much earlier, say when $\frac{\|\mathbf{b}-\mathbf{A}\mathbf{x}_k\|}{\|\mathbf{b}\|}$ is sufficiently small.

3.3 Convergence of GMRES

In this section, we discuss how quickly GMRES converges. First, we show how GMRES solves a polynomial approximation problem. The iterate $\mathbf{x}_k$ of GMRES is written $\mathbf{x}_k = q_k(\mathbf{A})\mathbf{b}$, where q_k is a polynomial of degree $k-1$. The residual then becomes, $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k = p_k(\mathbf{A})\mathbf{b}$, where $p_k(\mathbf{A}) = \mathbf{I} - \mathbf{A}q_k(\mathbf{A})$ and therefore p_k is a polynomial in $P_k = \{\text{polynomials } p \text{ of degree} \le k \text{ with } p(0) = 1\}$. Theorem 35.2 from [71] provides a convergence bound for GMRES as described next.

Theorem 3.3.1 ([71]) *At step k of the GMRES iteration, the residual $\mathbf{r}_k$ satisfies*

$$\frac{\|\mathbf{r}_k\|}{\|\mathbf{b}\|} \le \inf_{p_k\in P_k} \|p_k(\mathbf{A})\| \le \kappa(\mathbf{V}) \inf_{p_k\in P_k} \|p_k\|_{\Lambda(\mathbf{A})},$$

where $\Lambda(\mathbf{A})$ is the set of eigenvalues of $\mathbf{A}$, $\mathbf{V}$ is a nonsingular matrix of eigenvectors (assuming $\mathbf{A}$ is diagonalizable), and $\|p_k\|_{\Lambda(\mathbf{A})}$ is defined by $\sup_{z\in\Lambda(\mathbf{A})} |p_k(z)|$.

This upper bound will be small if $\mathbf{V}$ is well conditioned and if one can find polynomials whose size on the spectrum of $\mathbf{A}$ decreases rapidly with k.

Now, consider the case where $\mathbf{A}$ is symmetric and positive definite (SPD). For these types of matrices, we use the minimal residual method (MINRES) [55], which is a variant of GMRES designed for symmetric matrices. For this case, $\kappa(\mathbf{V}) = 1$, the polynomials have real coefficients, and the eigenvalues are real and positive. Clearly, this is beneficial for convergence since the upper bound of Theorem 3.3.1 is now only $\inf_{p_k\in P_k} \|p_k\|_{\Lambda(\mathbf{A})}$. For example, an identity matrix of dimension $n \times n$ has one eigenvalue, 1, with multiplicity n. Therefore, MINRES will converge in 1 iteration.

Another example is the following matrix which has 2 distinct eigenvalues,

$$\begin{bmatrix} \lambda & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & \mu \end{bmatrix}.$$

We would expect MINRES to converge in 2 iterations in this case. For more details and for the singular matrix case, see [42, 71].

3.4 Krylov Recycling

We will now provide a brief discussion of Krylov recycling as explained in [27]. We once again consider the linear system $\mathbf{A}\mathbf{x} = \mathbf{b}$, now with symmetric $\mathbf{A} \in \mathbb{R}^{n \times n}$ and $\mathbf{b} \in \mathbb{R}^n$. More on recycling is found in the literature, see [5, 43, 57, 77].

Let Range$(\tilde{\mathbf{U}})$, where $\tilde{\mathbf{U}} \in \mathbb{R}^{n \times n_c}$, be given as the space over which we, initially, look for solutions. Next, compute $\mathbf{A}\tilde{\mathbf{U}} = \tilde{\mathbf{K}}$. Set $\mathbf{K}$ to be the Q factor in the skinny QR factorization of $\tilde{\mathbf{K}}$ and $\mathbf{U} = \tilde{\mathbf{U}}\mathbf{R}^{-1}$, where $\mathbf{R}$ is the R factor. Now $\mathbf{K}^T\mathbf{K} = \mathbf{I}$ and note Range$(\mathbf{U})$ = Range$(\tilde{\mathbf{U}})$. Next, we assume that the solution of $\mathbf{A}\mathbf{x} = \mathbf{b}$ is in Range$(\mathbf{U})$ and find the approximate solution

$$\mathbf{x}_0 = \mathbf{U}\mathbf{K}^T\mathbf{b}, \tag{3.7}$$

giving an initial residual of $\mathbf{r} = \mathbf{b} - \mathbf{K}\mathbf{K}^T\mathbf{b}$. If this solution is not sufficiently accurate (as estimated by $\frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}$), then we expand $\mathbf{U}$ as follows. We use a Lanczos recurrence [47] with $(\mathbf{I} - \mathbf{K}\mathbf{K}^T)\mathbf{A}$ and $\mathbf{v}_1 = (\mathbf{I} - \mathbf{K}\mathbf{K}^T)\mathbf{b}/\|(\mathbf{I} - \mathbf{K}\mathbf{K}^T)\mathbf{b}\|_2$ to generate

$$\begin{aligned} (\mathbf{I} - \mathbf{K}\mathbf{K}^T)\mathbf{A}\mathbf{V}_m &= \mathbf{V}_{m+1}\underline{\mathrm{T}}_m \Leftrightarrow \\ \mathbf{A}\mathbf{V}_m &= \mathbf{K}\mathbf{K}^T\mathbf{A}\mathbf{V}_m + \mathbf{V}_{m+1}\underline{\mathrm{T}}_m \end{aligned} \tag{3.8}$$

where $\underline{\mathrm{T}}_m$ is an $(m+1) \times m$ tridiagonal matrix.* The Lanczos method is analogous

*The matrix recurrence in (3.8) is unchanged if $(\mathbf{I} - \mathbf{K}\mathbf{K}^T)\mathbf{A}$ is replaced by $(\mathbf{I} - \mathbf{K}\mathbf{K}^T)\mathbf{A}(\mathbf{I} - \mathbf{K}\mathbf{K}^T)$ since $\mathbf{A}$ is symmetric.

to Arnoldi, but for Hermitian matrices. We then compute the approximate solution in Range($[\mathbf{V}_m\ \mathbf{U}]$) by looking for the solution that minimizes $\|\mathbf{b}-\mathbf{A}(\mathbf{V}_m\mathbf{y}+\mathbf{U}\mathbf{z})\|_2$, as follows from [43]:

$$\begin{aligned}
&\min_{\mathbf{y},\mathbf{z}} \left\| \mathbf{b} - \mathbf{A}[\mathbf{U}\ \mathbf{V}_m] \begin{bmatrix} \mathbf{z} \\ \mathbf{y} \end{bmatrix} \right\|_2 \\
= &\min_{\mathbf{y},\mathbf{z}} \left\| \mathbf{b} - [\mathbf{K}\ \mathbf{V}_{m+1}] \begin{bmatrix} \mathbf{I} & \mathbf{K}^T\mathbf{A}\mathbf{V}_m \\ 0 & \underline{\mathrm{T}}_m \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{y} \end{bmatrix} \right\|_2 \\
= &\min_{\mathbf{y},\mathbf{z}} \left\| \begin{bmatrix} \mathbf{K}^T\mathbf{b} \\ \xi\mathbf{e}_1 \end{bmatrix} - \begin{bmatrix} \mathbf{I} & \mathbf{K}^T\mathbf{A}\mathbf{V}_m \\ 0 & \underline{\mathrm{T}}_m \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{y} \end{bmatrix} \right\|_2, \qquad (3.9)
\end{aligned}$$

where $\mathbf{e}_1$ denotes the first Cartesian basis vector in $\mathbb{R}^{m+1}$ and $\xi = \|(\mathbf{I}-\mathbf{K}\mathbf{K}^T)\mathbf{b}\|_2$. The solution is then found by finding the solution to the projected problem

$$\min_{\mathbf{y}} \|\underline{\mathrm{T}}_m\mathbf{y} - \xi\mathbf{e}_1\|_2,$$

then computing the $\mathbf{z}$ that satisfies (3.9), and finally, with $\mathbf{y}_m := \mathbf{V}_m\mathbf{y}$, setting $\mathbf{x} = \mathbf{y}_m + \mathbf{U}\mathbf{z}$. Since $\mathbf{V}_m\mathbf{y}$ is computed via short term recurrences (MINRES), we do not have to explicitly store $\mathbf{V}_m$ [50, 77].

We now discuss why Krylov recycling is beneficial in terms of convergence. Cauchy's interlacing theorem [68] ensures that no matter what $\mathbf{K}$ is used, the spectrum of $(\mathbf{I}-\mathbf{K}\mathbf{K}^T)\mathbf{A}(\mathbf{I}-\mathbf{K}\mathbf{K}^T)$ is within the spectrum of $\mathbf{A}$, therefore recycling is not making the convergence worse for the symmetric and positive definite case. Obviously, the goal is to converge much faster to offset the overhead cost associated with this method. If an exact invariant subspace is included in $\mathbf{U}$, then MINRES would converge as if part of the spectrum has been deflated. The right-hand side $(\mathbf{I}-\mathbf{K}\mathbf{K}^T)\mathbf{b}$ is made small across spectral components as well.

Chapter 4

Interpolatory Parametric Model Reduction for DOT-PaLS

In this chapter, we show how we use interpolatory parametric model reduction for the DOT problem. First, we provide the reader with background on model order reduction. Next, we explain how we can view the forward model in the DOT problem as a transfer function and show how the computational bottleneck of the problem is repeatedly solving the forward problem. We then show how to use interpolatory parametric model reduction to find a surrogate transfer function to reduce this computational cost. The global basis used to form the reduced order model (ROM) is constructed by solving several full order models and using a rank-revealing factorization. An analysis and numerical results for 2-dimensional and 3-dimensional problems are also given. For the use of model reduction in other optimization and inverse problem applications, we refer the reader to [8, 9, 12, 15, 31, 41, 46, 80] and the references therein.

4.1 Model Order Reduction Background

Another way to reduce the computational cost of solving large-scale systems is model order reduction (MOR), where the goal is to find a lower order model that nearly replicates the input-output behavior of the original model. There are many different parametric model order reduction methods, see [17, 21, 22, 37, 40, 53, 58, 59, 75]. We adopt a specific type of model reduction called interpolatory model reduction [11, 16, 20, 25, 32, 36]. As with other model reduction methods, the goal of interpolatory model reduction is to find lower order models, but interpolatory model reduction also requires that the lower order models transfer function interpolates the original systems transfer function at selected interpolation points.

We follow the interpolatory model reduction method as explained in [11, 16]. Consider the following input-output map, $\mathcal{S} : \mathbf{u} \to \mathbf{y}$,

$$\mathcal{S} : \begin{cases} \mathbf{E}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{cases} \quad \text{with } \mathbf{x}(0) = \mathbf{0} \tag{4.1}$$

where $\mathbf{A}, \mathbf{E} \in \mathbb{R}^{n\times n}, \mathbf{B} \in \mathbb{R}^{n\times m}, \mathbf{C} \in \mathbb{R}^{p\times n}, and\, \mathbf{D} \in \mathbb{R}^{p\times m}$. The reduced input-output map, $\mathcal{S}_r : \mathbf{u} \to \mathbf{y}_r$, is then defined as,

$$\mathcal{S}_r : \begin{cases} \mathbf{E}_r\mathbf{x}_r(t) = \mathbf{A}_r\mathbf{x}_r(t) + \mathbf{B}_r\mathbf{u}(t) \\ \mathbf{y}_r(t) = \mathbf{C}_r\mathbf{x}_r(t) + \mathbf{D}_r\mathbf{u}(t) \end{cases} \quad \text{with } \mathbf{x}(0) = \mathbf{0} \tag{4.2}$$

where $\mathbf{A}_r, \mathbf{E}_r \in \mathbb{R}^{r\times r}, \mathbf{B}_r \in \mathbb{R}^{r\times m}, \mathbf{C}_r \in \mathbb{R}^{p\times r}, and\, \mathbf{D}_r \in \mathbb{R}^{p\times m}$. The goal is to find $\mathcal{S}_r$ such that it is close to $\mathcal{S}$ and that $||\mathbf{y} - \mathbf{y}_r||$ is small. Taking the Laplace transform we get,

$$\hat{\mathbf{y}}(s) = (\mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D})\hat{\mathbf{u}}(s) \tag{4.3}$$

$$\hat{\mathbf{y}}_r(s) = (\mathbf{C}_r(s\mathbf{E}_r - \mathbf{A}_r)^{-1}\mathbf{B}_r + \mathbf{D}_r)\hat{\mathbf{u}}(s) \tag{4.4}$$

where ˆ denotes Laplace transformed quantities. The transfer functions are then defined as follows,

$$\mathbf{H}(s) = \mathbf{C}(s\mathbf{E} - \mathbf{A})^{-1}\mathbf{B} + \mathbf{D}, \tag{4.5}$$

$$\mathbf{H}_r(s) = \mathbf{C}_r(s\mathbf{E}_r - \mathbf{A}_r)^{-1}\mathbf{B}_r + \mathbf{D}_r. \tag{4.6}$$

Given left and right interpolation points, $\{\mu_i\}_{i=1}^q \in \mathbb{C}$ and $\{\sigma_i\}_{i=1}^r \in \mathbb{C}$, with corresponding left and right tangent directions, $\{\tilde{c}_i\}_{i=1}^q \in \mathbb{C}^p$ and $\{\tilde{b}_i\}_{i=1}^r \in \mathbb{C}^m$, we seek to find $\mathbf{H}_r(s)$, such that it is a tangential interpolant to $\mathbf{H}(s)$. This means we want the following to hold, $\tilde{c}_i^T\mathbf{H}_r(\mu_i) = \tilde{c}_i^T\mathbf{H}(\mu_i)$, for $i = 1, \cdots, q$ and $\mathbf{H}_r(\sigma_j)\tilde{b}_j = \mathbf{H}(\sigma_j)\tilde{b}_j$, for $j = 1, \cdots, r$.

Next, we discuss how to construct $\mathbf{H}_r(s)$ via Petrov-Galerkin projective approximation. First, choose r-dimensional right and left modeling subspaces, $\mathcal{V}_r \subset \mathbb{C}^n$ and

$\mathcal{W}_r \subset \mathbb{C}^n$ and rewrite the problem as follows,

$$\begin{aligned} \mathbf{E}\mathbf{v}(t) - \mathbf{A}\mathbf{v}(t) - \mathbf{B}\mathbf{u}(t) \perp \mathcal{W}_r \\ \mathbf{y}_r(t) = \mathbf{C}\mathbf{v}(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \quad \text{with } \mathbf{v}(t) \in \mathcal{V}_r. \tag{4.7}$$

Next, let $\mathbf{V}_r, \mathbf{W}_r \in \mathbb{C}^n$ be matrices such that $\mathcal{V}_r = \text{Range}(\mathbf{V}_r)$ and $\mathcal{W}_r = \text{Range}(\overline{\mathbf{W}_r})$. Then, $\mathbf{v}(t) = \mathbf{V}_r\mathbf{x}_r(t)$ with $\mathbf{x}_r \in \mathbb{C}^r$ and we rewrite the problem as follows,

$$\begin{aligned} \mathbf{W}_r^T(\mathbf{E}\mathbf{V}_r\mathbf{x}_r(t) - \mathbf{A}\mathbf{V}_r\mathbf{x}_r(t) - \mathbf{B}\mathbf{u}(t)) = \mathbf{0}, \\ \mathbf{y}_r(t) = \mathbf{C}\mathbf{V}_r\mathbf{x}_r(t) + \mathbf{D}\mathbf{u}(t). \end{aligned} \tag{4.8}$$

The reduced matrices are then given by, $\mathbf{E}_r = \mathbf{W}_r^T\mathbf{E}\mathbf{V}_r$, $\mathbf{B}_r = \mathbf{W}_r^T\mathbf{B}$, $\mathbf{A}_r = \mathbf{W}_r^T\mathbf{A}\mathbf{V}_r$, $\mathbf{C}_r = \mathbf{C}\mathbf{V}_r$, and $\mathbf{D}_r = \mathbf{D}$. It should be noted that since $\mathbf{D}$ is of modest size, letting $\mathbf{D}_r = \mathbf{D}$ is common practice.

We now explore how to construct $\mathbf{V}_r$ and $\mathbf{W}_r$.

Theorem 4.1.1 (U. Baur, C. Beattie, P. Benner, and S. Gugercin) *Let σ, $\mu \in \mathbb{C}$ be such that $s\mathbf{E}-\mathbf{A}$ and $s\mathbf{E}_r-\mathbf{A}_r$ are invertible for $s = \sigma, \mu$. Also, let $\mathbf{V}_r$, $\mathbf{W}_r \in \mathbb{C}^{n\times r}$ have full-rank. If $\mathbf{b} \in \mathbb{C}^m$ and $\mathbf{c} \in \mathbb{C}^\ell$ are fixed nontrivial vectors then*

(a) if $(\sigma\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}\mathbf{b} \in$ Range$(\mathbf{V}_r)$, then $\mathbf{H}(\sigma)\mathbf{b} = \mathbf{H}_r(\sigma)\mathbf{b}$;

(b) if $\left(\mathbf{c}^T\mathbf{C}(\mu\mathbf{E} - \mathbf{A})^{-1}\right) \in$ Range$(\mathbf{W}_r)$, then $\mathbf{c}^T\mathbf{H}(\mu) = \mathbf{c}^T\mathbf{H}_r(\mu)$; and

(c) if both (a) and (b) hold, and $\sigma = \mu$, then $\mathbf{c}^T\mathbf{H}'(\sigma)\mathbf{b} = \mathbf{c}^T\mathbf{H}_r'(\sigma)\mathbf{b}$ as well.

Theorem 4.1.1 tells us that given left and right interpolation points, $\{\mu_j\}_{j=1}^r$ and $\{\sigma_i\}_{i=1}^r$ as well as left and right tangential directions, $\{c_k\}_{k=1}^r \in \mathbb{C}^p$ and $\{b_k\}_{k=1}^r \in \mathbb{C}^m$, the solution of the tangential interpolation problem is found by simply constructing $\mathbf{V}_r$ and $\mathbf{W}_r$ as follows,

$$\mathbf{V}_r = [(\sigma_1\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}b_1, \cdots, (\sigma_r\mathbf{E} - \mathbf{A})^{-1}\mathbf{B}b_r]$$

$$\mathbf{W}_r^T = \begin{bmatrix} c_1^T\mathbf{C}(\mu_1\mathbf{E} - \mathbf{A})^{-1} \\ \vdots \\ c_r^T\mathbf{C}(\mu_r\mathbf{E} - \mathbf{A})^{-1} \end{bmatrix}.$$

4.2 A Systems Theoretic Perspective on Parametric Inversion

Many spatial discretizations can be applied to (2.4)-(2.7), including finite difference and finite element methods. Discretization gives the following differential algebraic equation,

$$\frac{1}{\nu}\mathbf{E}\,\dot{\mathbf{y}}(t,\mathbf{p}) = -\mathbf{A}(\mathbf{p})\mathbf{y}(t,\mathbf{p}) + \mathbf{B}\mathbf{u}(t) \quad \text{with} \quad \boldsymbol{m}(t,\mathbf{p}) = \mathbf{C}^T\mathbf{y}(t,\mathbf{p}) \tag{4.9}$$

where $\mathbf{y}$ denotes the discretized photon flux and $\boldsymbol{m} = [m_1, \ldots, m_{n_{det}}]^T$ is the vector of outputs. The columns of $\mathbf{B}$ and $\mathbf{C}$ are discretizations of the sources and detectors, respectively. We also have that $\mathbf{A}(\mathbf{p})$ only changes on the diagonal for each different set of parameters. The matrix $\mathbf{E}$ is generally singular because of the Robin boundary conditions. Following [28], we rewrite (4.9) as

$$\widehat{\mathbf{m}}(\omega,\mathbf{p}) = \boldsymbol{\Psi}(\omega,\mathbf{p})\,\widehat{\mathbf{u}}(\omega) \quad \text{where} \quad \boldsymbol{\Psi}(\omega,\mathbf{p}) = \mathbf{C}^T\left(\frac{\imath\omega}{\nu}\mathbf{E} + \mathbf{A}(\mathbf{p})\right)^{-1}\mathbf{B}, \tag{4.10}$$

where $\widehat{\mathbf{y}}(\omega,\mathbf{p})$, $\widehat{\mathbf{u}}(\omega)$, and $\widehat{\mathbf{m}}(\omega,\mathbf{p})$ denote the Fourier transforms of $\mathbf{y}(t,\mathbf{p})$, $\mathbf{u}(t)$, $\boldsymbol{m}(t,\mathbf{p})$, $\omega \in \mathbb{R}$ is the frequency, and $\boldsymbol{\Psi}(\omega,\mathbf{p})$ is known as the frequency response or the transfer function.

For the i^{th} input source, j^{th} frequency, and absorption field, $\mu(\cdot,\mathbf{p})$, $\widehat{\mathbf{m}}_i(\omega_j,\mathbf{p}) \in \mathbb{C}^{n_{det}}$ denotes the vector of predicted observations as given by the forward model in the frequency domain. If we stack these vectors for all n_{src} sources and n_ω frequencies, we obtain

$$\mathcal{M}(\mathbf{p}) = [\widehat{\mathbf{m}}_1(\omega_1,\mathbf{p})^T, \ldots, \widehat{\mathbf{m}}_1(\omega_{n_\omega},\mathbf{p})^T, \widehat{\mathbf{m}}_2(\omega_1,\mathbf{p})^T, \ldots, \widehat{\mathbf{m}}_{n_{src}}(\omega_{n_\omega},\mathbf{p})^T]^T, \tag{4.11}$$

which is a vector of dimension $n_{det} \cdot n_{src} \cdot n_\omega$. We let $\mathbb{D}$ be the corresponding vector from acquired data. This leads us to the optimization problem we ultimately want to solve,

$$\min_{\mathbf{p}\in\mathbb{R}^{n_p}} \|\mathcal{M}(\mathbf{p}) - \mathbb{D}\|_2. \tag{4.12}$$

At each step of the optimization problem we must evaluate $\mathcal{M}(\mathbf{p}) - \mathbb{D}$, which means that we must compute the following for all i and j,

$$\widehat{\mathbf{m}}_i(\omega_j, \mathbf{p}) = \mathbf{\Psi}(\omega_j)\widehat{\mathbf{u}}_i(\omega_j), \tag{4.13}$$

where $\mathbf{\Psi}(\omega, \mathbf{p})$ is the frequency response found in (4.10), $\widehat{\mathbf{u}}_i(\omega_j) = \mathbf{e}_i$, and $\mathbf{e}_i$ is the i^{th} column of the identity matrix. Therefore, a function evaluation at parameter vector $\mathbf{p}_k$ requires solving the following linear systems

$$\left(\frac{\imath\omega_j}{\nu}\mathbf{E} + \mathbf{A}(\mathbf{p}_k)\right)\mathbf{Y}_{k,j} = \mathbf{B} \qquad \mathbf{B} \in \mathbb{R}^{n \times n_{src}}, \qquad j = 1, \dots, n_\omega, \tag{4.14}$$

where

$$\mathbf{Y}_{k,j} := [\widehat{\mathbf{y}}_1(\omega_j, \mathbf{p}_k), \dots, \widehat{\mathbf{y}}_{n_s}(\omega_j, \mathbf{p}_k)].$$

In DOT, the columns of $\mathbf{B}$ are multiples of the i^{th} columns of the identity matrix, which correspond to the i^{th} source location.

In order to solve the nonlinear inverse problem (4.12) for the parameters, the Jacobian is also required and we make similar observations. The adjoint-type (or co-state) approach as in [38, 76] is adopted to construct the Jacobian. To construct the Jacobian, we use (4.10) and (4.13) and differentiate $\widehat{\mathbf{m}}_i(\omega_j, \mathbf{p})$ with respect to the k^{th} component of $\mathbf{p}$ to get

$$\frac{\partial}{\partial p_k}\widehat{\mathbf{m}}_i(\omega_j, \mathbf{p}) = \frac{\partial}{\partial p_k}\left[\mathbf{\Psi}(\omega_j, \mathbf{p})\right]\widehat{\mathbf{u}}_i(\omega_j) = -\mathbf{Z}(\omega_j, \mathbf{p})^T \frac{\partial}{\partial p_k}\mathbf{A}(\mathbf{p})\,\widehat{\mathbf{y}}_i(\omega_j, \mathbf{p}), \tag{4.15}$$

where $\mathbf{Z}(\omega_j, \mathbf{p})$ is found by solving

$$\left(\frac{\imath\omega_j}{\nu}\mathbf{E} + \mathbf{A}(\mathbf{p})\right)^T \mathbf{Z}(\omega_j, \mathbf{p}) = \mathbf{C}, \qquad \mathbf{C} \in \mathbb{R}^{n \times n_{src}}. \tag{4.16}$$

As with matrix $\mathbf{B}$, the columns of $\mathbf{C}$ are columns of the identity matrix, which correspond to the j^{th} detector location. The main computational cost of evaluating the Jacobian at a parameter vector is the cost of solving $n_{det} \times n_\omega$ linear systems in (4.16) of order equal to the number of degrees of freedom. This is because the matrices $\frac{\partial}{\partial p_k}\mathbf{A}(\mathbf{p})$ only have to be computed once for all parameters. Also, $\widehat{\mathbf{y}}_i(\omega_j, \mathbf{p})$ has already been computed for the function evaluation.

The critical bottleneck in solving the inverse problem is associated with repeatedly solving (4.14) and (4.16). Therefore, we seek a surrogate function $\boldsymbol{\Psi}_r(\omega, \mathbf{p})$ that is not only easy to evaluate, but provides a good approximation to $\boldsymbol{\Psi}(\omega, \mathbf{p})$ over parameters and frequencies of interest. Our goal fits nicely within the context of model reduction [10, 11]. In our application, the frequencies are given to us by the experimental set-up. The parameters of interest are those that the optimization routine would chose when run using the full order model. In addition, we require that $\nabla_{\mathbf{p}}\boldsymbol{\Psi}_r(\omega, \mathbf{p})$ is easy to evaluate and that $\nabla_{\mathbf{p}}\boldsymbol{\Psi}(\omega, \mathbf{p}) \approx \nabla_{\mathbf{p}}\boldsymbol{\Psi}_r(\omega, \mathbf{p})$ over the same frequencies and parameters.

4.3 Surrogate Forward Model

The surrogate forward model for the DOT problem, found via parametric model reduction, follows the method explained in Section 4.1. We seek a reduced input-output map that is able to replicate (4.9),

$$\frac{1}{\nu}\mathbf{E}_r\,\dot{\mathbf{y}}_r(t, \mathbf{p}) = -\mathbf{A}_r(\mathbf{p})\mathbf{y}_r(t, \mathbf{p}) + \mathbf{B}_r\mathbf{u}(t) \quad \text{with} \quad \boldsymbol{m}_r(t, \mathbf{p}) = \mathbf{C}_r^T\mathbf{y}_r(t, \mathbf{p}) \tag{4.17}$$

where $\mathbf{y}_r(t, \mathbf{p}) \in \mathbb{R}^r \mathbf{A}_r(\mathbf{p}), \mathbf{E}_r \in \mathbb{R}^{r\times r}, \mathbf{B}_r \in \mathbb{R}^{r\times n_{src}}, \mathbf{C}_r^T \in \mathbb{R}^{n_{det}\times r}$, and $r \ll n$. The reduced input-output map is constructed such that $\boldsymbol{m}(t, \mathbf{p}) \approx \boldsymbol{m}_r(t, \mathbf{p})$ for $\mathbf{p}$ of interest. The reduced transfer function then becomes

$$\boldsymbol{\Psi}_r = \mathbf{C}_r^T\left(\frac{\imath\omega}{\nu}\mathbf{E}_r + \mathbf{A}_r(\mathbf{p})\right)^{-1}\mathbf{B}_r. \tag{4.18}$$

Since $\mathbf{B}_r$ and $\mathbf{B}$ have the same number of columns and $\mathbf{C}_r^T$ and $\mathbf{C}^T$ have the same number of rows, $\boldsymbol{\Psi}_r$ and $\boldsymbol{\Psi}$ are of the same dimension. However, one needs to solve n dimensional linear systems to evaluate $\boldsymbol{\Psi}$, while only r dimensional linear systems are solved to evaluate $\mathbf{\Psi}_r$. Similarly, (4.16) is reduced as follows,

$$\left(\frac{\imath\omega_j}{\nu}\mathbf{E}_r + \mathbf{A}_r(\mathbf{p})\right)^T \mathbf{Z}_r(\omega_j, \mathbf{p}) = \mathbf{C}_r. \tag{4.19}$$

This means that $\nabla_{\mathbf{p}}\mathbf{\Psi}(\omega, \mathbf{p})$ is evaluated by solving n dimensional linear systems, while $\nabla_{\mathbf{p}}\mathbf{\Psi}_r(\omega, \mathbf{p})$ is evaluated by solving r dimensional linear systems. Since $r \ll n$, these reduced solves will drastically reduce the cost of the forward problem.

Projection is used to obtain the surrogate parametric model [19]. Suppose full rank matrices $\mathbf{V} \in \mathbb{C}^{n\times r}$ and $\mathbf{W} \in \mathbb{C}^{n\times r}$ are given. If we assume $\mathbf{y}(t, \mathbf{p})$ evolves near the r-dimensional subspace Range$(\mathbf{V})$, then $\mathbf{y}(t, \mathbf{p}) \approx \mathbf{V}\widehat{\mathbf{y}}(t, \mathbf{p})$ and we enforce a Petrov-Galerkin condition to obtain the reduced matrices given by

$$\mathbf{E}_r = \mathbf{W}^T\mathbf{E}\mathbf{V}, \quad \mathbf{A}_r(\mathbf{p}) = \mathbf{W}^T\mathbf{A}(\mathbf{p})\mathbf{V}, \quad \mathbf{B}_r = \mathbf{W}^T\mathbf{B}, \quad \text{and} \quad \mathbf{C}_r = \mathbf{V}^T\mathbf{C}. \tag{4.20}$$

4.4 Interpolatory Parametric Model Reduction for DOT

In this section, we explain how we use interpolatory model reduction as in Section 4.1 in the context of DOT. Through parametric model reduction for DOT, we are trying to match the original transfer function and derivatives with the reduced transfer function and derivatives, therefore, interpolatory parametric model reduction suits this goal perfectly. This means that for a parameter vector $\hat{\mathbf{p}} \in \mathbb{R}^\ell$ and a frequency $\hat{\omega} \in \mathbb{R}$, we want to use the reduced model found in (4.17) where the reduced transfer function, $\mathbf{\Psi}_r(\omega, \mathbf{p})$, satisfies

$$\mathbf{\Psi}(\hat{\omega}, \hat{\mathbf{p}}) = \mathbf{\Psi}_r(\hat{\omega}, \hat{\mathbf{p}}) \quad \text{and} \quad \nabla_{\mathbf{p}}\mathbf{\Psi}(\hat{\omega}, \hat{\mathbf{p}}) = \nabla_{\mathbf{p}}\mathbf{\Psi}_r(\hat{\omega}, \hat{\mathbf{p}}). \tag{4.21}$$

The following theorem was originally stated in [11, 16], also given in Theorem 4.1.1, and presented in [28] in the context of DOT. We present the DOT version of the theorem, which shows how to construct the projection matrices, $\mathbf{V}$ and $\mathbf{W}$.

Theorem 4.4.1 *Suppose $\mathbf{A}(\mathbf{p})$ is continuously differentiable in a neighborhood of $\hat{\mathbf{p}} \in \mathbb{R}^\ell$. Let $\hat{\omega} \in \mathbb{C}$, and both $\frac{\imath\hat{\omega}}{\nu}\mathbf{E} + \mathbf{A}(\hat{\mathbf{p}})$ and $\frac{\imath\hat{\omega}}{\nu}\mathbf{E}_r + \mathbf{A}_r(\hat{\mathbf{p}})$ be invertible.*

If $\left(\frac{\imath\hat{\omega}}{\nu}\mathbf{E}+\mathbf{A}(\hat{\mathbf{p}})\right)^{-1}\mathbf{B}\in$ *Range*$(\mathbf{V})$ *and* $\left(\mathbf{C}\left(\frac{\imath\hat{\omega}}{\nu}\mathbf{E}+\mathbf{A}(\hat{\mathbf{p}})\right)^{-1}\right)^{T}\in$ *Range*$(\mathbf{W})$ *then, the reduced parametric model of (4.20) satisfies*
$\mathbf{\Psi}(\hat{\omega},\hat{\mathbf{p}})=\mathbf{\Psi}_r(\hat{\omega},\hat{\mathbf{p}})$, $\nabla_{\mathbf{p}}\mathbf{\Psi}(\hat{\omega},\hat{\mathbf{p}})=\nabla_{\mathbf{p}}\mathbf{\Psi}_r(\hat{\omega},\hat{\mathbf{p}})$, *and* $\mathbf{\Psi}'(\hat{\omega},\hat{\mathbf{p}})=\mathbf{\Psi}_r'(\hat{\omega},\hat{\mathbf{p}})$,
where $'$ *denotes the derivative with respect to* ω.

Theorem 4.4.1 tells us that if the Range$(\mathbf{V})$ and Range$(\mathbf{W})$ contain certain vectors, the function and gradient values of the full order model and the reduced model will match exactly. This means that for the frequency and parameter interpolation points chosen to construct $\mathbf{V}$ and $\mathbf{W}$, the optimization approach will produce identical values for the reduced forward model as for the full order forward model. That is, if we were able to choose parameters vectors that the optimization algorithm would choose and use those parameters vectors as the interpolation points to construct $\mathbf{V}$ and $\mathbf{W}$, there would be no difference between using the full order forward model and the reduced forward model in the optimization algorithm.

4.5 Projection via Global Basis Matrices

For the DOT problem, we adopt a global basis approach in the construction of $\mathbf{V}$ and $\mathbf{W}$. This means we only construct $\mathbf{V}$ and $\mathbf{W}$ once and reuse the bases at each optimization step. Therefore, we need to construct $\mathbf{V}$ and $\mathbf{W}$ with a sample of parameters such that $\mathbf{V}$ and $\mathbf{W}$ capture information about the range of parameters values the optimization routine will encounter. An alternative method to the global basis approach is the local basis method where the basis is updated as the parameters vary. For more details on these methods see, e.g., [6, 7, 29, 56].

Following Theorem 4.4.1 and referring to (4.14), we let

$$\mathbf{Y}:=[\mathbf{Y}_{1,1},\ldots,\mathbf{Y}_{1,n_\omega},\mathbf{Y}_{2,1},\ldots,\mathbf{Y}_{2,n_\omega},\ldots,\mathbf{Y}_{K,n_\omega}], \tag{4.22}$$

where $\mathbf{Y}_{i,j}=\left(\frac{\imath\omega_j}{\nu}\mathbf{E}+\mathbf{A}(\mathbf{p}_i)\right)^{-1}\mathbf{B}$ for $i=1,\ldots,K$ and $\omega_1,\ldots,\omega_{n_\omega}$ are the frequency interpolation points. For the DOT problem, the frequency interpolation points are given by experimental set-up. This is not the case in all model reduction applications and usually the optimal frequency interpolation also needs to be found, as in [16,35]. It is reasonable to assume that $\mathbf{Y}$ will have repetitive information and therefore we define $\mathbf{V}$ to be the left singular vectors corresponding to the non-zero singular values of $\mathbf{Y}$. Similar steps are taken to construct $\mathbf{W}$ from $\mathbf{Z}_{i,j}$ via (4.16) for $i = 1, \ldots, K$ and $j = 1, \ldots, n_\omega$. Constructing $\mathbf{V}$ and $\mathbf{W}$ in this way ensures that the corresponding reduced transfer function formed using (4.20) will match the transfer function evaluation at *every* $(\omega, \mathbf{p}) = (\omega_j, \mathbf{p}_i)$ for $j = 1, \ldots, n_\omega$ and $i = 1, \ldots, K$. A similar result holds for the derivative computations.

In addition, one-sided global basis construction is employed. This means that we let $\mathbf{V} \leftarrow [\mathbf{V}, \mathbf{W}]$ and $\mathbf{W} \leftarrow [\mathbf{V}, \mathbf{W}]$ in (4.20). This is done to preserve symmetry, if $\mathbf{A}(\mathbf{p}), \mathbf{E}$ are symmetric (Hermitian), the reduced counterparts are also symmetric (Hermitian).

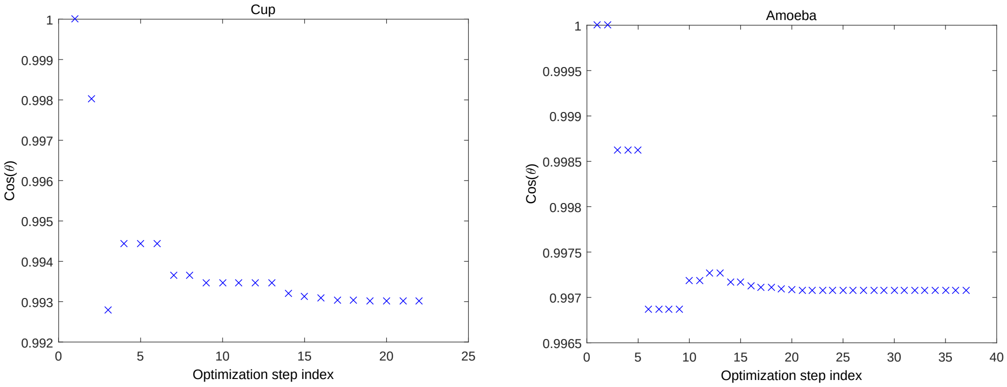
4.6 Analysis of Global Basis Projection

In this section, we give some intuition about why the approach discussed in the previous section is effective for the DOT problem. Notably, for the DOT problem, $\mathbf{V}$ and $\mathbf{W}$ are not are expected to change much from one set of parameters to the next. Therefore, $\mathbf{V}$ and $\mathbf{W}$ are constructed from a small number of interpolation points and still give a good approximation to the full order forward model. Also, we might be able to reuse $\mathbf{V}$ and $\mathbf{W}$ for different image reconstructions.

We demonstrate that $\mathbf{V}$ does not change significantly from one parameter set to the next with two numerical examples. Let $\mathbf{V}_k$ be the right projection space for parameter vector $\mathbf{p}_k$ obtained in the k^{th} optimization step,

$$\mathbf{V}_k = \left[\left(\frac{\imath \omega_1}{\nu} \mathbf{E} + \mathbf{A}(\mathbf{p}_k) \right)^{-1} \mathbf{B}, \ldots, \left(\frac{\imath \omega_{n_\omega}}{\nu} \mathbf{E} + \mathbf{A}(\mathbf{p}_k) \right)^{-1} \mathbf{B} \right].$$

$\mathbf{V}_k$ is the concatenation of the solutions to the k^{th} system for all frequencies. Figure 4.1 shows how close these projection spaces remain to the initial space $\mathbf{V}_1$ for two numerical examples. We give the cosine of the largest canonical angle at each optimization step. The cosine of the largest canonical angle tells us how close two



(a) (b)

Figure 4.1: Evolution of the subspace gap (cosine of the largest canonical angle θ) between the initial and subsequent projection spaces over the course of the optimization.

subspaces are to one another. If the cosine is 1, then the subspaces intersect one another, while if the cosine is 0 the subspaces are orthogonal. In both test problems, the cosine of the largest canonical angles remains close to 1, meaning the subspaces are quite close.

Next, we provide the reader with some intuition about how changing the parameters affects the reconstruction. We refer the reader to (2.15) for a description of the parameters discussed here. For the experiments below each set of parameters, α_j, β_j, $\chi_{j,x}$, and $\chi_{j,y}$, are shifted by the amount given. In each experiment, the shifting is repeated ten times. Figures 4.2, 4.3, 4.4, and 4.5 show the images at each step of the shifting.

In Figure 4.6 we show that the evolution of the absorption image over the first six distinct reconstructions corresponding to approximate solutions, $\mathbf{p}_k$ for one example. Figure 4.7 shows the evolution of absorption images for another example. The initial guess, i.e. the image for parameter vector $\mathbf{p}_1$, is shown in the first image. While the reconstructions can change drastically from the initial guess, the corresponding changes in the projection spaces are quite small. Similar results are found for $\mathbf{W}$ as well.

The experiments presented here suggest that a good choice for parameter vectors for use in constructing the global basis would be parameter vectors from the first

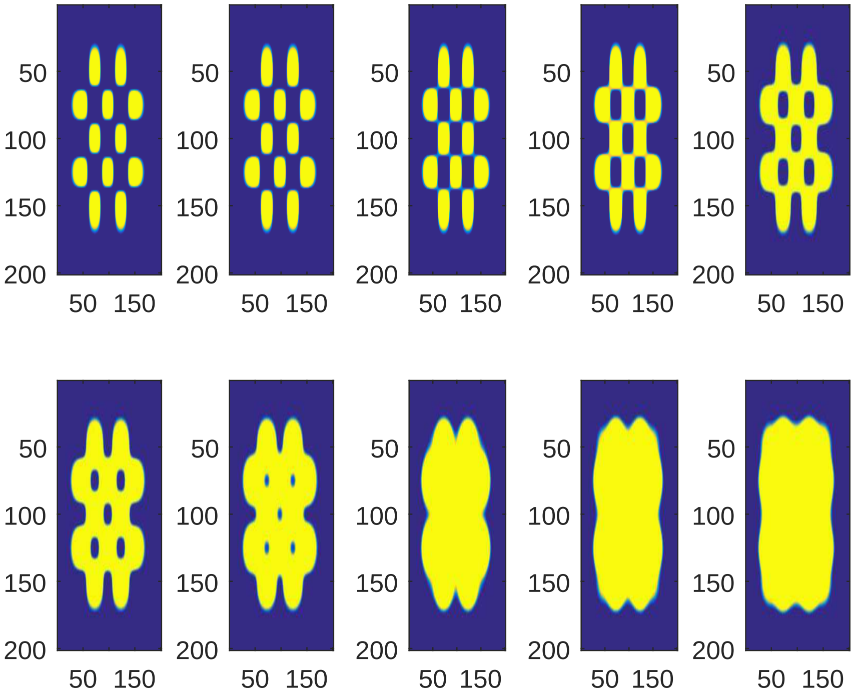


Figure 4.2: Images with all α_j changed by 0.025.

few steps of the optimization. Since these initial parameter vectors would be close to the initial guess, global bases constructed in this way will likely be effective for distinct image reconstructions as well.

4.7 Cost

The computational cost of this method can be split into an offline phase, where the model reduction bases are constructed and an online phase, where the reduced order model is used for the function and Jacobian evaluations. First in the offline phase, the parameter space needs to be sampled. In the numerical examples below, the leading $K \leq 3$ iterations of the full optimization problem are used to obtain the sample parameter points since in the previous section we saw that subsequent projection spaces remain close to the initial guess. However, any sampling technique may be used. Once the sample parameters have been selected, $\mathbf{Y}$ is found as in (4.22). Since $\mathbf{Y}$ likely has linearly dependent columns, $\mathbf{V}$ is constructed as the

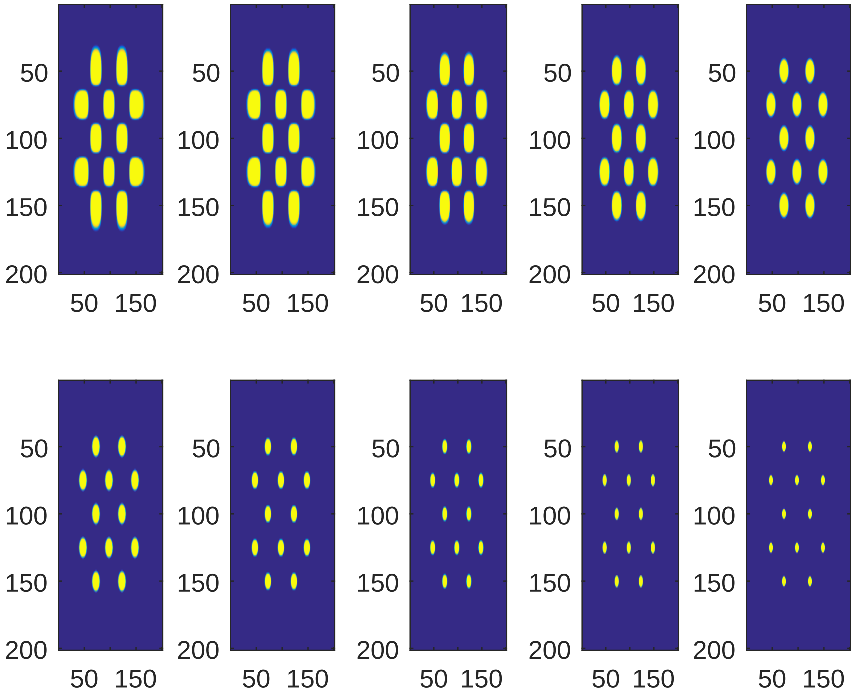


Figure 4.3: Images with all β_j changed by 0.1.

leading left singular vectors of $\mathbf{Y}$ corresponding to the singular values greater than a given tolerance. To conclude the offline phase, we construct $\mathbf{W}$ in the same way from concatenating solutions to (4.16) as in (4.22) and using the leading left singular vectors corresponding to singular values greater than a given tolerance.

The online phase consists of using the parametric reduced model in place of the expensive function and Jacobian evaluations. At each step in the optimization algorithm, that is for each $\mathbf{p}_k$, we need to find $\mathbf{E}_r = \mathbf{W}^T\mathbf{E}\mathbf{V}$, $\mathbf{A}_r(\mathbf{p}) = \mathbf{W}^T\mathbf{A}(\mathbf{p})\mathbf{V}$, $\mathbf{B}_r = \mathbf{W}^T\mathbf{B}$, and $\mathbf{C}_r = \mathbf{V}^T\mathbf{C}$. $\mathbf{E}_r$, $\mathbf{B}_r$, and $\mathbf{C}_r$ are constant and only need to be computed once. $\mathbf{A}_r(\mathbf{p})$ must be computed for each $\mathbf{p}_k$, although in our setting this can be done cheaply. As stated above, $\mathbf{A}(\mathbf{p})$ only changes on the diagonal for different sets of parameters, so we let $\mathbf{A}(\mathbf{p}) = \mathbf{A}_* + \mathbf{A}_k(\mathbf{p})$, where $\mathbf{A}_*$ is constant and $\mathbf{A}_k(\mathbf{p})$ is diagonal. Therefore,

$$\mathbf{A}_r(\mathbf{p}) = \mathbf{W}^T\mathbf{A}_*\mathbf{V} + \mathbf{W}^T\mathbf{A}_k(\mathbf{p})\mathbf{V}$$

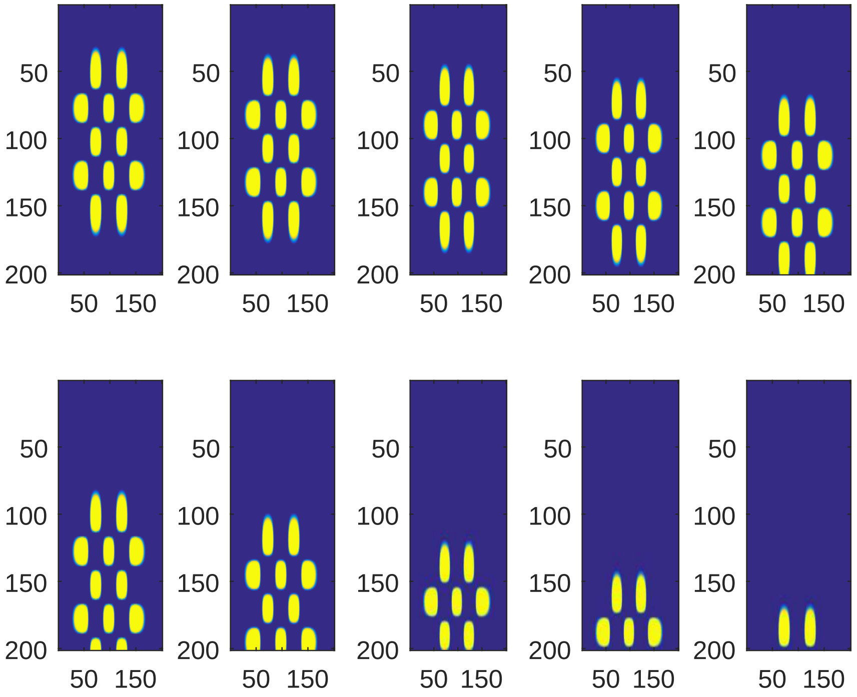


Figure 4.4: Images with all $\chi_{j,x}$ changed by 0.05.

and $\mathbf{W}^T\mathbf{A}_*\mathbf{V}$ are precomputed. Computing $\mathbf{W}^T\mathbf{A}_k(\mathbf{p})\mathbf{V}$ and then solving with $\mathbf{A}_r(\mathbf{p})$ will be very cheap compared to the alternative of solving many large, sparse linear systems for the full order model. Once these matrices are computed, the optimization algorithm proceeds with the reduced function and Jacobian evaluations, which are now $r \times r$ linear systems.

Therefore, the main computational cost comes from forming $\mathbf{V}$ and $\mathbf{W}$. Assume we have n_ω frequency interpolation points, $n_{src} = n_{det}$ sources and detectors, and K parameters samples, then in order to construct $\mathbf{V}$ and $\mathbf{W}$ we must solve $2Kn_\omega n_{src}$ large, sparse, $n \times n$ linear systems. If we were to use full order model solves throughout the optimization, we would solve $K_{fun}n_\omega n_{src} + K_{Jac}n_\omega n_{det}$ large, sparse $n \times n$ linear systems, where K_{fun} is the number of function evaluations and K_{jac} is the number of Jacobian evaluations. Since the ratio $(K_{fun} + K_{jac})/2K$ has been found to always be greater than 3 in our numerical experiments, the offline costs are amortized. In addition, since we are able to reuse $\mathbf{V}$ and $\mathbf{W}$ for different images, there are no

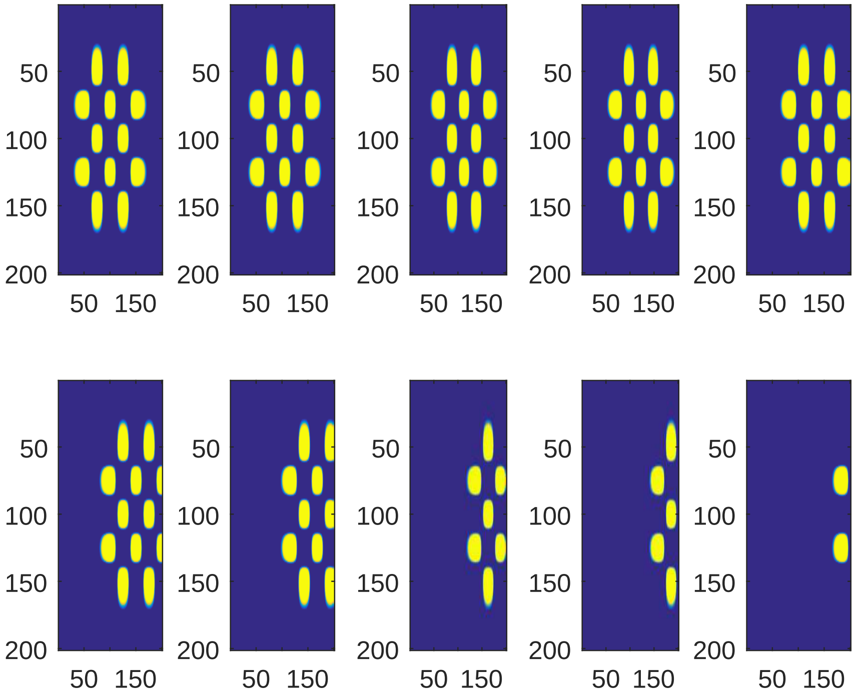


Figure 4.5: Images with all $\chi_{j,y}$ changed by 0.05.

additional offline costs for the subsequent inversions.

4.8 Numerical Experiments

In this section, we provide four numerical experiments, two 2-dimensional examples and two 3-dimensional examples. For each set of examples only one set of projection bases are used, therefore no additional large systems solves are required for the second experiment. These proof of concept examples strongly suggest that the projection bases might only need to be computed once for many different reconstructions.

All of the experiments are set up in the following way. First, we need to generate synthetic data for use in computing the measured values. We construct a $0 - 1$ image, where 0 represents healthy tissue and 1 represents anomalous tissue. Each pixel is then assigned an absorption value, which has a small normally distributed random variation, based on whether the pixel corresponds to healthy tissue or anomalous tissue. We also add 0.1% white noise to the image. The PaLS approach, as described

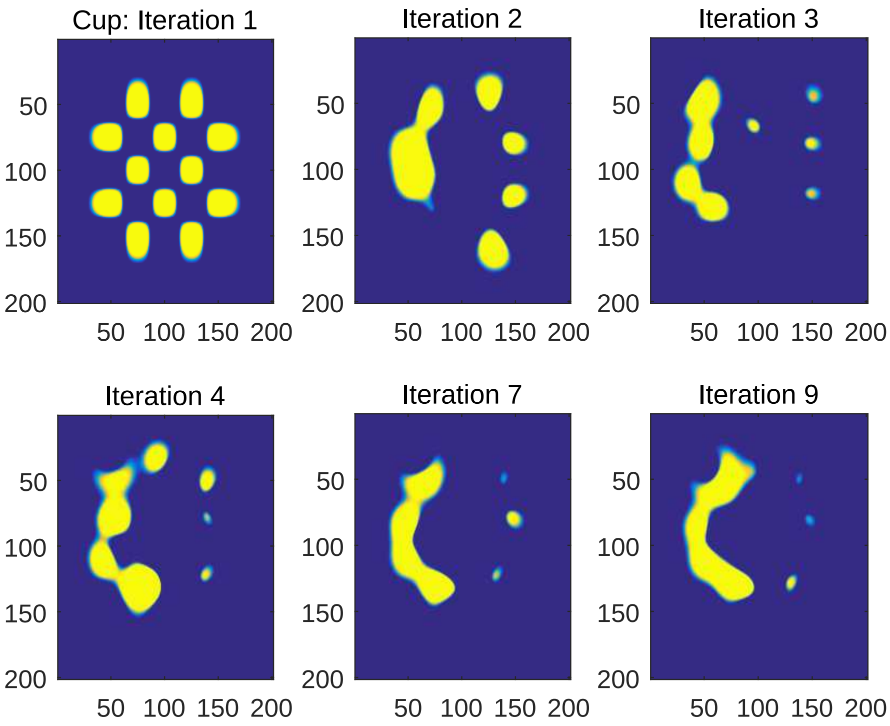


Figure 4.6: Initial image and the first five distinct reconstructions for Cup example.

in Section 2.3, is used to reconstruct the shape of the images and the optimization problems are solved using the TREGS algorithm [44]. The stopping criterion for the optimization, known as the discrepancy principle [52], is when the residual norm falls below 1.1 times the noise level. Results are reported for both using the full order model and the ROM to compute the function and Jacobian evaluations for each experiment. For each experiment, the first 3 iterations of the optimization using the full order model are used as the interpolation points for the reduced order model. All of the experiments were run using a laptop with a 3.20 GHz processor and 16.0 GB RAM using MATLAB R2015b.

4.8.1 2D Experiments

The first two experiments were solved on a 201×201 mesh, which gives us $40,401$ degrees of freedom for the forward problem. We use 32 sources and 32 detectors. Following the PaLs approach, we use 25 compactly supported radial basis functions,

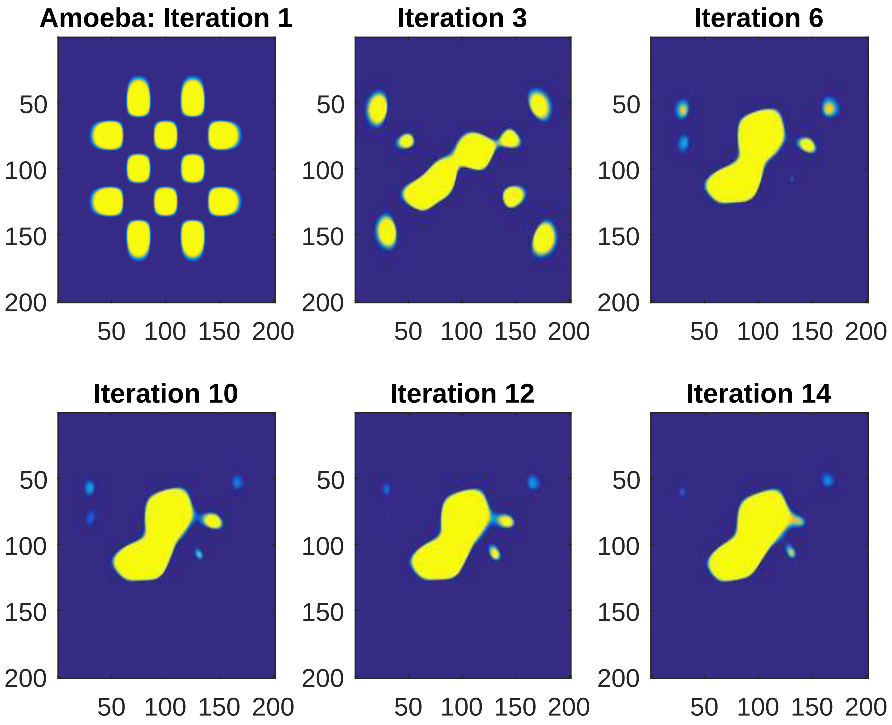


Figure 4.7: Initial image and the first five distinct reconstructions for Amoeba example.

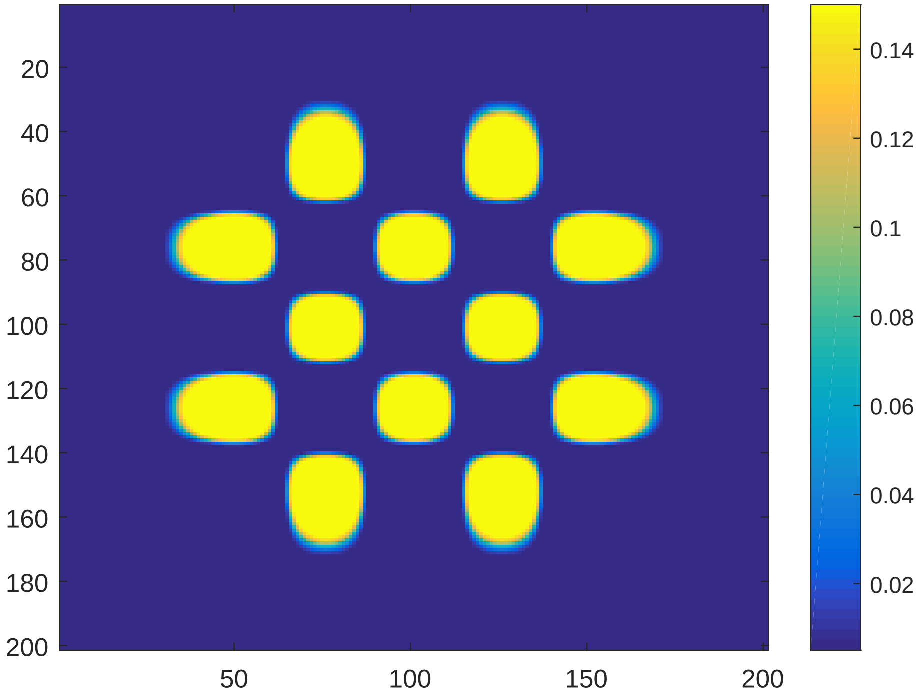
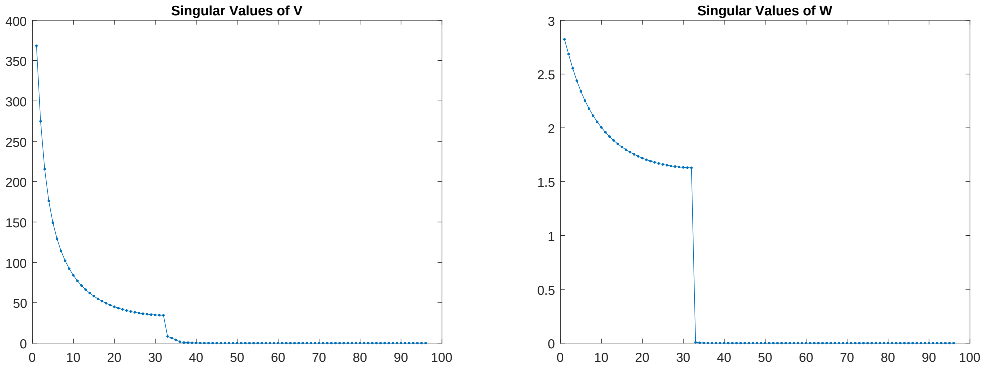


Figure 4.8: Initial absorption image with a 5×5 grid of 25 CSRBFs with alternating signs of α_j.

which results in 100 parameters for the optimization problem (4.12). Figure 4.8 shows the absorption image using the initial set of parameters. The projection bases that are used for Experiments 1 and 2 were created with the first 3 iterations of the optimization using the full order model for Experiment 1. Therefore, 192 large, sparse systems need to be solved in order to form the projection bases. After computing the SVD, the reduced bases have 100 vectors giving the reduced model order 100. Plots of the singular values of $\mathbf{V}$ and $\mathbf{W}$ before truncation are given in Figure 4.9. We decided to truncate at 50 for both $\mathbf{V}$ and $\mathbf{W}$. In this chapter, where to truncate is based on numerical experimentation that balances the number of function evaluations and quality of the image. We have found that we need to include some of the singular vectors corresponding to the singular values after the drop off to get a good reconstruction. Therefore, the reduced models require the solutions to linear systems of size 100×100, while the full order model requires solutions to linear systems of size $40,401 \times 40,401$.



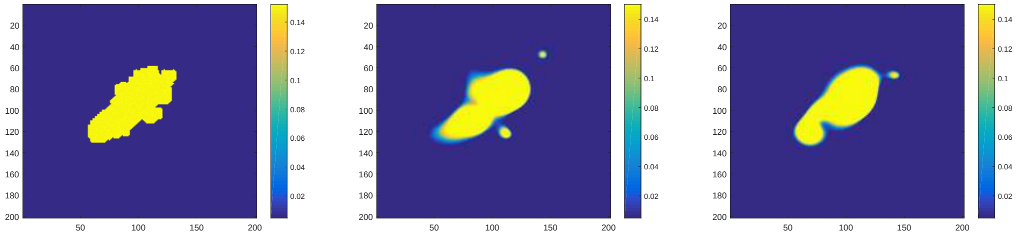
(a)

(b)

Figure 4.9: Plots of the singular values of **V** and **W** before truncation for Experiment 1.

For Experiment 1, the optimization using the full order model required 26 function evaluations and 14 Jacobian evaluations. Meanwhile, the optimization using the reduced order model for Experiment 1 required 31 function evaluations and 16 Jacobian evaluations, indicating that the use of the ROM instead of the FOM does not greatly impact the convergence rate of the optimization. Figure 4.10 shows the reconstructions for Experiment 1. Arguably, the reconstruction using the ROM is not as good as the FOM here. This observation brings up the question of how do we know when we are done building the reduced basis. If we had included more systems in the basis, then we could achieve a better reconstruction, but under the current scheme we have no way of knowing when we have constructed a sufficient basis. The method presented in Chapter 5 addresses this issue, but this is also the subject of future work.

The optimization using the full order model for Experiment 2 required 72 function evaluations and 47 Jacobian evaluations. Meanwhile, the optimization using the reduced order model for Experiment 2 required 28 function evaluations and 16 Jacobian evaluations, indicating that the ROM in this case actually speeds up the convergence of the optimization. Figure 4.11 shows the reconstructions for Experiment 2. Remember that for the reduced order model here, the same projection bases were used as in Experiment 1, so no full order model solves are required for

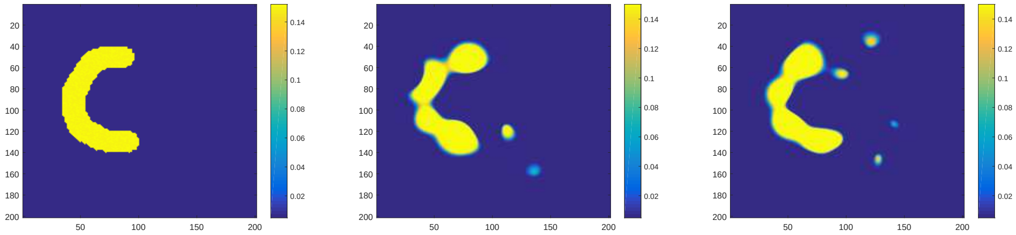


(a) Ground truth absorption image.

(b) Reconstruction using the full order model.

(c) Reconstruction using the reduced order model.

Figure 4.10: Results for Experiment 1. Reconstruction on a 201 × 201 mesh, resulting in 40, 401 degrees of freedom in the forward model and 100 degrees of freedom in the reduced order model. 32 sources, 32 detectors, and 25 basis functions were used.



(a) Ground truth absorption image.

(b) Reconstruction using the full order model.

(c) Reconstruction using the reduced order model.

Figure 4.11: Results for Experiment 2. Reconstruction on a 201 × 201 mesh, resulting in 40, 401 degrees of freedom in the forward model and 100 degrees of freedom in the reduced order model. 32 sources, 32 detectors, and 25 basis functions were used. This reconstruction used the same projection bases as in Experiment 1.

this experiment.

4.8.2 3D Experiments

The next two experiments were solved on a $32 \times 32 \times 32$ mesh, which gives us 32, 768 degrees of freedom for the forward problem. We use 225 sources and 225 detectors. Following the PaLs approach, we use 27 compactly supported radial basis functions, which results in 135 parameters for the optimization problem (4.12). Figure 4.12 shows 16 slices of the absorption image using the initial set of parameters. The projection bases that are used for Experiments 3 and 4 were created with the first 3 iterations of the optimization using the full order model for Experiment 3. Therefore, 1, 350 large, sparse systems need to be solved in order to form the projection bases.

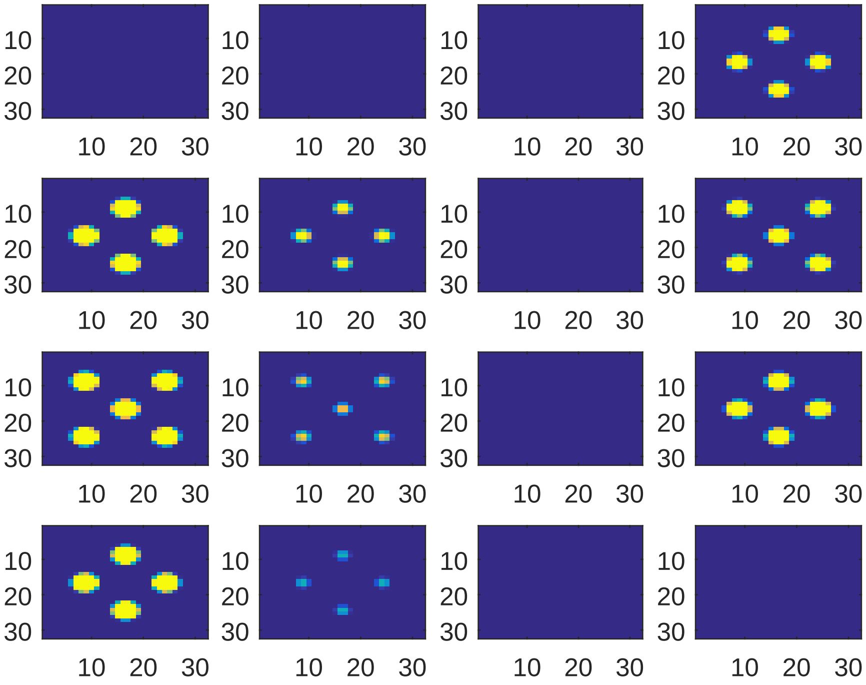


Figure 4.12: Slices of the initial absorption image with 27 CSRBFs.

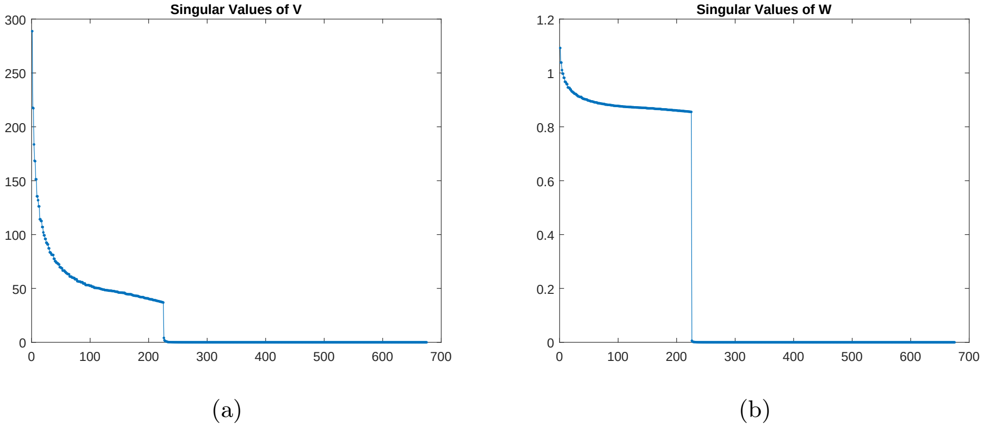
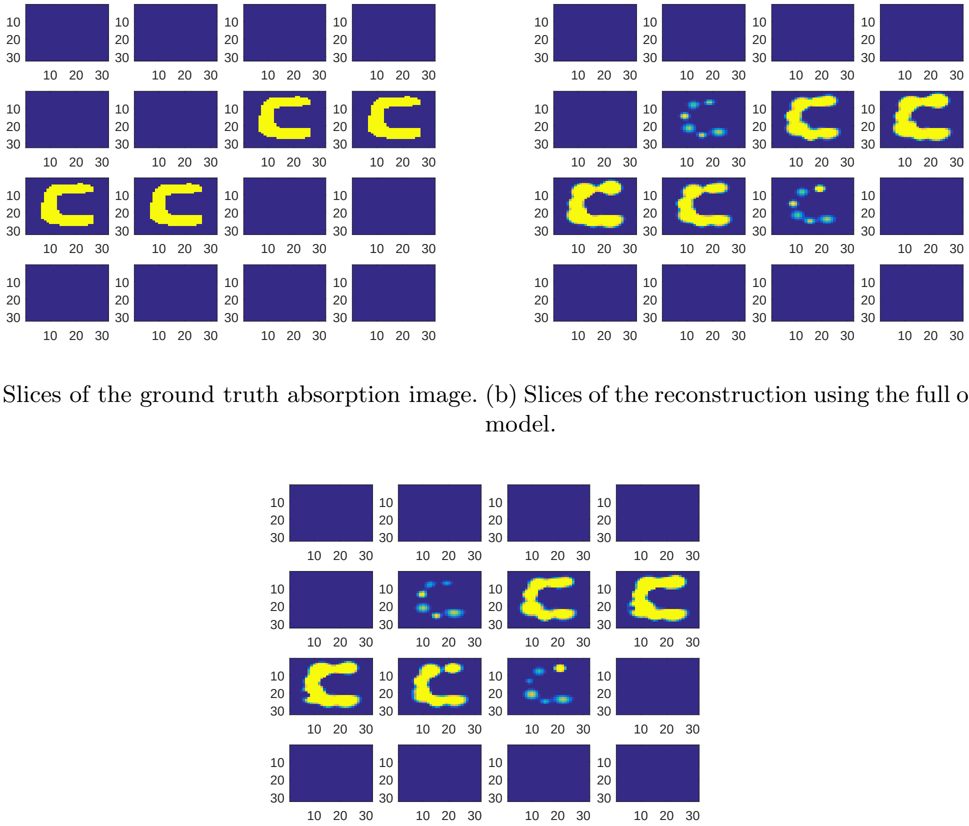


Figure 4.13: Plots of the singular values of **V** and **W** before truncation for Experiment 3.

After computing the SVD, the reduced bases have 500 vectors giving the reduced model order 500. Plots of the singular values of **V** and **W** before truncation are given in Figure 4.13. We decided to truncate at 250 for both **V** and **W**. Again, 250 was chosen based on numerical experimentation. Therefore, the reduced models



(a) Slices of the ground truth absorption image.

(b) Slices of the reconstruction using the full order model.

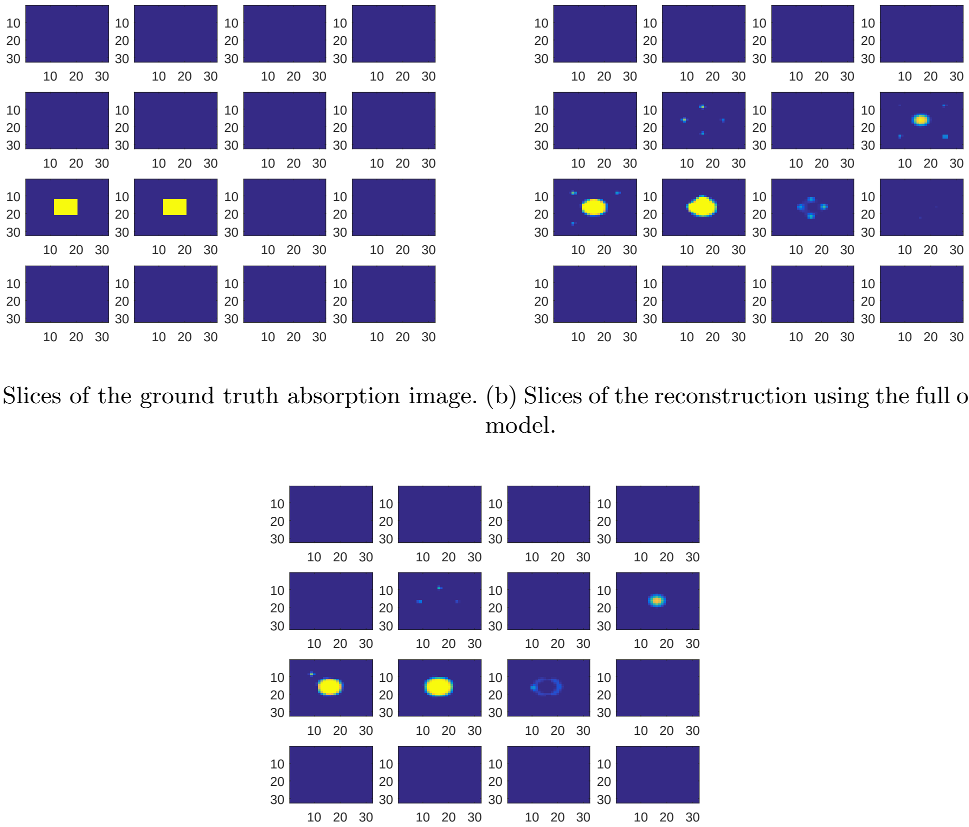
(c) Slices of the reconstruction using the reduced order model.

Figure 4.14: Results for Experiment 3. Reconstruction on a $32 \times 32 \times 32$ mesh, resulting in $32,768$ degrees of freedom in the forward model and 500 degrees of freedom in the reduced order model. 225 sources, 225 detectors, and 27 basis functions were used.

require the solutions to linear systems of size 500×500, while the full order model requires solutions to linear systems of size $32,768 \times 32,768$.

The optimization using the full order model for Experiment 3 required 28 function evaluations and 15 Jacobian evaluations. Meanwhile, the optimization using the reduced order model for Experiment 3 required 31 function evaluations and 21 Jacobian evaluations. Once again indicating that the ROM does not greatly impact the convergence rate of the optimization. Figure 4.14 shows slices of the reconstructions for Experiment 3.

For Experiment 4, the optimization using the full order model required 51 function evaluations and 29 Jacobian evaluations. Meanwhile, the optimization using the reduced order model for Experiment 4 required 44 function evaluations and 31 Jacobian evaluations. Figure 4.15 shows the reconstructions for Experiment 4. Remember that for the reduced order model here, the same projection bases were used as in Experiment 3, so no full order model solves are required for this experiment. If we use a more refined mesh or more compactly supported radial basis functions, then we potentially recover images with sharper edges. While we might be able to recover images with shaper edges, we have to weigh this benefit with the additional cost incurred.



(a) Slices of the ground truth absorption image.

(b) Slices of the reconstruction using the full order model.

(c) Slices of the reconstruction using the reduced order model.

Figure 4.15: Results for Experiment 4. Reconstruction on a $32 \times 32 \times 32$ mesh, resulting in 32, 768 degrees of freedom in the forward model and 500 degrees of freedom in the reduced order model. 225 sources, 225 detectors, and 27 basis functions were used. This reconstruction used the same projection bases as in Experiment 3.

Chapter 5

Efficient Computation of Reduced Order Models in DOT

In this chapter, we once again look at solving the DOT problem with ROM. As was shown in the previous chapter, in order to form the basis used in the ROM, several large-scale forward problems need to be solved. These solves are typically done iteratively due to the structure and size of the systems [18]. We show how one can use Krylov recycling techniques to solve these systems while dynamically building the reduced basis without redundant information. In the previous chapter, we explained how the construction of the reduced basis was sensitive to the truncation parameter, which was chosen in an ad hoc manner. The method presented in this chapter eliminates the need to perform an expensive SVD by constructing the basis as we go with only new information. The Krylov recycling method presented here also provides some insight into when we have built a basis sufficient for producing a good reduced order model. In addition, this method provides the potential for computational savings, since we solve fewer systems with fewer total matrix-vector products and eliminate the SVD computation. In recent work [4, 33, 34], the use of Krylov subspace recycling to construct the MOR basis was investigated. In [43], Krylov recycling for DOT was investigated with the goal of solving the sequence of systems throughout the inversion process. In Section 5.1, we show that the transfer function in the DOT problem can be reformulated as a transfer function of a slightly smaller symmetric problem given a particular geometry and discretization. Then, in Section 5.2 we use this property to develop an efficient algorithm to generate the reduced basis and give an analysis. Numerical results for 2-dimensional and 3-dimensional problems are shown in Section 5.3.

5.1 Rewriting the Transfer Function and Derivatives

In this section, we show how we express the transfer function and derivatives in terms of a symmetric positive definite matrix. This is useful in creating an efficient method for constructing the global basis.

Following the discretization scheme in [43], we use second-order centered differences away from the boundary and first-order differences to implement the Robin boundary condition. Looking back at the original definition of the full order transfer function in (4.10) for the 2D case, the matrix $\frac{\imath\omega}{\nu}\mathbf{E} + \mathbf{A}(\mathsf{p})$, has the following block structure:

$$\begin{bmatrix} \mathbf{G} & \mathbf{D}_1 \\ \mathbf{D}_2 & (\mathbf{F}(\mathsf{p}) + \frac{\imath\omega h^2}{\nu}\mathbf{I}) \end{bmatrix}. \tag{5.1}$$

This matrix is lexicographically ordered with the boundary nodes first followed by the internal nodes. In addition, we know the following about the blocks in (5.1),

- $\mathbf{G}$ is an invertible diagonal matrix,
- $\mathbf{D}_1$ has at most one nonzero per row, and these occur only in the first N_xN_y and last N_xN_y columns,
- $\mathbf{D}_2$, although it has different entries, has the same sparsity pattern as $\mathbf{D}_1^T$.

While the matrix $\mathbf{A}(\mathsf{p})$ is not symmetric, it was shown in [43] that the Schur complement (for the $\omega = 0$ case), given by $\mathbf{F}(\mathsf{p}) - \mathbf{D}_2\mathbf{G}^{-1}\mathbf{D}_1$, is symmetric and positive definite. We now exploit this fact to rewrite the transfer function and derivatives.

5.1.1 Transfer Function

Recall that the columns of $\mathbf{B}$ and $\mathbf{C}$ are scaled columns of an $N_xN_y \times N_xN_y$ identity matrix. The sources and detectors are only found on the boundary, therefore partitioning $\mathbf{B}$ and $\mathbf{C}$ with $\frac{\imath\omega}{\nu}\mathbf{E} + \mathbf{A}(\mathsf{p})$ as in (5.1) we obtain

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_1 \\ \mathbf{0} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{0} \end{bmatrix}.$$

It should also be noted that the sources and detectors are not co-located and therefore $\mathbf{C}_1^T \mathbf{B}_1 = 0$.

For ease of exposition, we assume that $\omega = 0$ in (5.1). For the $\omega = 0$ case, the inverse of $\mathbf{A}(\mathsf{p})$ appears in the definition of the transfer function. We also assume the matrix $\mathbf{A}(\mathsf{p})^{-1}$ has the following block structure:

$$\begin{bmatrix} \mathbf{H} & \mathbf{S}_1 \\ \mathbf{S}_2 & \mathbf{N} \end{bmatrix}.$$

Using $\mathbf{A}(\mathsf{p})\mathbf{A}(\mathsf{p})^{-1} = \mathbf{I}$, we obtain the following three expressions that are helpful in rewriting the transfer function and derivatives:

$$\mathbf{H} = [\mathbf{G} - \mathbf{D}_1^T \mathbf{F}^{-1} \mathbf{D}_2]^{-1}, \tag{5.2}$$

$$\mathbf{S}_1 = -\mathbf{G}^{-1}\mathbf{D}_1[\mathbf{F} - \mathbf{D}_2\mathbf{G}^{-1}\mathbf{D}_1]^{-1}, \tag{5.3}$$

$$\mathbf{S}_2 = -[\mathbf{F} - \mathbf{D}_2\mathbf{G}^{-1}\mathbf{D}_1]^{-1}\mathbf{D}_2\mathbf{G}^{-1}. \tag{5.4}$$

From these expressions, it is straightforward to show that

$$\mathbf{\Psi}(0, \mathsf{p}) = \mathbf{C}^T(\mathbf{A}(\mathsf{p}))^{-1}\mathbf{B} = \mathbf{C}_1^T \mathbf{H} \mathbf{B}_1.$$

We know that $\mathbf{F}(\mathsf{p})$ is SPD since $\mathbf{F}(\mathsf{p}) = \mathbf{L} + \operatorname{diag}(\mu(\mathsf{p}))$, where $\mathbf{L}$ is the discretization of the Laplacian at the internal nodes multiplied by the (constant) diffusion coefficient, and $\operatorname{diag}(\mu(\mathsf{p}))$ is a non-negative diagonal matrix. Therefore, we express $\mathbf{F}(\mathsf{p})$ in terms of its eigendecomposition, $\mathbf{F} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, so that

$$\begin{aligned} \mathbf{H} &= [\mathbf{G} - \mathbf{D}_1^T \mathbf{Q} \mathbf{\Lambda}^{-1/2} \mathbf{\Lambda}^{-1/2} \mathbf{Q}^T \mathbf{D}_2]^{-1} \\ &= \mathbf{G}^{-1} + \mathbf{G}^{-1}\mathbf{D}_1^T[\mathbf{F} - \mathbf{D}_2\mathbf{G}^{-1}\mathbf{D}_1]^{-1}\mathbf{D}_2\mathbf{G}^{-1}, \end{aligned}$$

by the Sherman-Morrison-Woodbury formula.

Using the above, the fact that the sources and detectors are not co-located, as well as the fact that $\mathbf{G}$ is diagonal, it follows that

$$\mathbf{\Psi}(0,\mathsf{p}) = \underbrace{\mathbf{C}_1^T\mathbf{G}^{-1}\mathbf{D}_1^T}_{\tilde{\mathbf{C}}^T}\underbrace{[\mathbf{F}-\mathbf{D}_2\mathbf{G}^{-1}\mathbf{D}_1]^{-1}}_{\tilde{\mathbf{A}}(\mathsf{p})^{-1}}\underbrace{\mathbf{D}_2\mathbf{G}^{-1}\mathbf{B}_1}_{\tilde{\mathbf{B}}}. \tag{5.5}$$

It is important to note that due to the structure of the matrices, $\tilde{\mathbf{C}}$ and $\tilde{\mathbf{B}}$ maintain the same structure as $\mathbf{C}_1$ and $\mathbf{B}_1$. Meaning their columns are multiples of the columns of the N_xN_y identity matrix, therefore they are considered 'effective' sources and receivers.

A similar argument is made for the $\omega \neq 0$ case to obtain,

$$\mathbf{\Psi}(\omega,\mathsf{p}) = \tilde{\mathbf{C}}^T\left(\frac{\imath\omega}{\nu}\mathbf{I}+\tilde{\mathbf{A}}(\mathsf{p})\right)^{-1}\tilde{\mathbf{B}}. \tag{5.6}$$

Therefore, the transfer function for the 0 frequency case is expressed in terms of an SPD matrix, and complex symmetric if ω is non-zero.

5.1.2 Derivative Computation

We once again use $\omega = 0$ for simplicity and therefore we define (4.11) for this case, which is an $n_{src} \times n_{det}$ matrix,

$$M(0,\mathsf{p}) = [\widehat{\mathbf{m}}_1(0,\mathsf{p}),\ldots,\widehat{\mathbf{m}}_{n_{det}}(0,\mathsf{p})].$$

Note that the "vec" command vectorizes a matrix, meaning it maps a matrix in $\mathbb{R}^{m_1\times m_2}$ to a vector in $\mathbb{R}^{m_1m_2}$ by unstacking the columns of the argument from left to right. Therefore, the vector $\text{vec}(\frac{\partial}{\partial p_k}M) \in \mathbb{R}^{n_{src}n_{det}}$ gives the k^{th} column of the Jacobian matrix. Now, using (4.14 - 4.16), we write

$$\frac{\partial}{\partial p_k}M(\mathsf{p}) = -\mathbf{C}^T\mathbf{A}(\mathsf{p})^{-1}\frac{\partial}{\partial p_k}\mathbf{A}(\mathsf{p})\mathbf{A}(\mathsf{p})^{-1}\mathbf{B}.$$

We now show how we rewrite this in terms of $\tilde{\mathbf{A}}(\mathsf{p})$.

Using the same finite-difference discretization scheme and ordering of unknowns as before and the fact that the boundary terms have no absorption, we write

$$\frac{\partial}{\partial p_k}\mathbf{A}(\mathbf{p}) = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Delta \end{bmatrix},$$

for a diagonal matrix $\Delta := \frac{\partial}{\partial p_k}\tilde{\mathbf{A}}(\mathbf{p})$. Therefore,

$$\begin{aligned} \frac{\partial}{\partial p_k}M(0,\mathbf{p}) &= -\begin{bmatrix} \mathbf{C}_1^T & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{H} & \mathbf{S}_1 \\ \mathbf{S}_2 & \mathbf{N} \end{bmatrix}\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Delta \end{bmatrix}\begin{bmatrix} \mathbf{H} & \mathbf{S}_1 \\ \mathbf{S}_2 & \mathbf{N} \end{bmatrix}\begin{bmatrix} \mathbf{B}_1 \\ \mathbf{0} \end{bmatrix} \\ &= -\mathbf{C}_1^T\mathbf{S}_1\Delta\mathbf{S}_2\mathbf{B}_1. \end{aligned}$$

Using (5.3) and (5.4), we have

$$\frac{\partial}{\partial p_k}M(0,\mathbf{p}) = -\underbrace{\mathbf{C}_1^T\mathbf{G}^{-1}\mathbf{D}_1}_{\tilde{\mathbf{C}}^T}\underbrace{[\mathbf{F}-\mathbf{D}_2\mathbf{G}^{-1}\mathbf{D}_1]^{-1}}_{\tilde{\mathbf{A}}(\mathbf{p})^{-1}}\Delta\underbrace{[\mathbf{F}-\mathbf{D}_2\mathbf{G}^{-1}\mathbf{D}_1]^{-1}}_{\tilde{\mathbf{A}}(\mathbf{p})^{-1}}\underbrace{\mathbf{D}_2\mathbf{G}^{-1}\mathbf{B}_1}_{\tilde{\mathbf{B}}}. \tag{5.7}$$

This means that for the 0 frequency case the required derivatives are computed from the same SPD matrix, $\tilde{\mathbf{A}}(\mathbf{p})$, as the transfer function.

A similar argument is made for the $\omega \neq 0$ case to obtain,

$$\frac{\partial}{\partial p_k}M(\omega,\mathbf{p}) = -\tilde{\mathbf{C}}^T\left(\frac{\imath\omega}{\nu}\mathbf{I}+\tilde{\mathbf{A}}(\mathbf{p})\right)^{-1}\Delta\left(\frac{\imath\omega}{\nu}\mathbf{I}+\tilde{\mathbf{A}}(\mathbf{p})\right)^{-1}\tilde{\mathbf{B}}. \tag{5.8}$$

Therefore, the derivatives for the 0 frequency case are expressed in terms of an SPD matrix and complex symmetric if ω is non-zero.

5.1.3 Rewriting ROM

The original transfer function and derivatives, for the 0 frequency case, are now computed using an SPD matrix of size $(N_y-2)N_x \times (N_y-2)N_x$. Therefore, we now look for a ROM for this slightly smaller problem.

As in Section 4.5, we use the one-sided global basis approach and for $\mathbf{V} \in \mathbb{C}^{n\times r}$

we define[*]

$$\tilde{\mathbf{E}}_r = \mathbf{V}^T\mathbf{V}, \qquad \tilde{\mathbf{A}}_r(\mathbf{p}) = \mathbf{V}^T\tilde{\mathbf{A}}(\mathbf{p})\mathbf{V}, \qquad \tilde{\mathbf{B}}_r = \mathbf{V}^T\tilde{\mathbf{B}}, \qquad \tilde{\mathbf{C}}_r = \mathbf{V}^T\tilde{\mathbf{C}}. \tag{5.9}$$

The reduced transfer function is therefore, $\tilde{\mathbf{\Psi}}_r = \tilde{\mathbf{C}}_r^T(\frac{\imath\omega}{\nu}\tilde{\mathbf{E}}_r + \tilde{\mathbf{A}}_r(\mathbf{p}))^{-1}\tilde{\mathbf{B}}_r$.

We use the fact that $\tilde{\mathbf{A}}$ is SPD in order to solve the forward and adjoint problems at the same time to generate $\mathbf{V}$,

$$\left(\tilde{\mathbf{A}}(\mathbf{p}_k) + \frac{\imath\omega_j}{\nu}\mathbf{I}\right)\mathbf{X}_{k,j} = [\tilde{\mathbf{B}}, \tilde{\mathbf{C}}], \tag{5.10}$$

for appropriate choices of parameters $\mathbf{p}_k$, $k = 1, \ldots, K$ and frequencies $\omega_j, j = 1, \ldots, n_\omega$.

For the remainder of this chapter, we only consider the $\omega = 0$ case. The non-trivial extension to the non-zero ω case is explained in the next chapter.

As in Section 4.2, we are seeking a surrogate transfer function, $\tilde{\mathbf{\Psi}}_r(\omega, \mathbf{p})$ that approximates $\tilde{\mathbf{\Psi}}(\omega, \mathbf{p})$ as well as ensuring $\nabla_{\mathbf{p}}\tilde{\mathbf{\Psi}}(\omega, \mathbf{p}) \approx \nabla_{\mathbf{p}}\tilde{\mathbf{\Psi}}_r(\omega, \mathbf{p})$. The following theorem, which follows from [16], is Theorem 4.4.1 rewritten for the symmetric DOT-PaLs problem in the zero frequency case, shows how to construct $\mathbf{V}$.

Theorem 5.1.1 *Suppose $\tilde{\mathbf{A}}(\mathbf{p})$ is continuously differentiable in a neighborhood of $\hat{\mathbf{p}} \in \mathbb{R}^l$. Let both $\tilde{\mathbf{A}}(\hat{\mathbf{p}})$ and $\tilde{\mathbf{A}}_r(\hat{\mathbf{p}})$ be invertible. If $\tilde{\mathbf{A}}(\hat{\mathbf{p}})^{-1}\mathbf{B}$ and $\left(\mathbf{C}\tilde{\mathbf{A}}(\hat{\mathbf{p}})^{-1}\right)^T$ are in Range$(\mathbf{V})$, then the reduced parametric model satisfies $\tilde{\mathbf{\Psi}}(0, \hat{\mathbf{p}}) = \tilde{\mathbf{\Psi}}_r(0, \hat{\mathbf{p}})$ and $\nabla_{\mathbf{p}}\tilde{\mathbf{\Psi}}(0, \hat{\mathbf{p}}) = \nabla_{\mathbf{p}}\tilde{\mathbf{\Psi}}_r(0, \hat{\mathbf{p}})$.*

If we follow the approach in the previous chapter to construct $\mathbf{V}$, it would consist of the steps outlined in Algorithm 1. While we need to compute each $\mathbf{X}_{k,0}$, because we need this to compute function and Jacobian evaluations at steps 1 through K_* of the optimization problem, some of the information that we generate and put into the concatenated matrix is redundant. This was shown in Section 4.6 with the rapid decay of the singular values of the concatenated matrix. Meaning that while generating $\mathbf{V}$ we are computing information that we do not really need and increasing the cost by requiring an expensive postprocessing rank-revealing SVD.

[*] The $\mathbf{V}$ as we generate it will typically not have orthonormal columns, hence the need to specify $\tilde{\mathbf{E}}_r$.

Algorithm 1: Generate One-Sided Global Basis via Truncated SVD

```
1. Solve the systems (5.10) for $k = 1, \dots, K_*$ for the first $\mathbf{p}_k$ parameter vectors
   produced by the optimization;
2. Concatenate the block solutions into a large block matrix
       $[\mathbf{X}_{1,0}, \mathbf{X}_{2,0}, \dots, \mathbf{X}_{K_*,0}]$.
3. Set $\mathbf{V}$ to be the matrix of the first $r$ left singular vectors of the above matrix.
   This gives a reduced order model, (5.9), of dimension $r$.
```

Since we are solving large and sparse systems with SPD matrices, it makes sense to use a Krylov subspace algorithm to solve the individual systems. In the next section, we provide an iterative Krylov subspace method which not only minimizes the work for computing only what we need for approximating the $\mathbf{X}_{k,0}(:, j)$ where it is needed for the optimization, but also generates an approximate global basis matrix by updating an initial estimate with only non-redundant information. This method eliminates the need for step 3 above in computing $\mathbf{V}$. This is because we build $\mathbf{V}$ to have r columns as we go, rather than overbuilding $\mathbf{V}$ and then compressing it to r columns.

5.2 Inner-Outer Krylov Recycling

In an attempt to keep the notation as simple as possible, we let $\mathbf{A}_k := \tilde{\mathbf{A}}(\mathbf{p}_k)$, $\mathbf{X}_k := \mathbf{X}_{k,0}$, and $\mathbf{B} := [\tilde{\mathbf{B}}, \tilde{\mathbf{C}}]$. Note that the matrices $\mathbf{A}$ and $\mathbf{B}$ have new definitions from earlier in this chapter. Therefore, we will be solving,

$$\mathbf{A}_k \mathbf{X}_k = \mathbf{B}, \tag{5.11}$$

for several values of k in order to determine the global basis via Krylov recycling. We also know that $\mathbf{A}(\mathbf{p})$ only changes on the diagonal from one system to the next since

$$\mathbf{A}(\mathbf{p}) = \underbrace{(\mathbf{L} - \mathbf{D}_2 \mathbf{G}^{-1} \mathbf{D}_1)}_{\mathbf{A}_*} + \mathrm{diag}(\mu(\mathbf{p})). \tag{5.12}$$

We remind the reader that Krylov recycling basics are given in Section 3.4. In [43], the authors showed that for systems of the form (5.11), recycling was efficient in the context of optimization for shape parameters in DOT. In that work, they used recycling, with a different recycle space for each right-hand side, to solve for all parameters selected throughout the optimization. We extend from their work to not only use recycling to solve our shorter sequence of full order problems efficiently, but also to build the global basis without redundant information. We now introduce our inner-outer recycling method. The idea is that if we have an estimate of the global basis matrix $\mathbf{V}$, it makes sense to consider using $\mathbf{V}$ as a recycling space as well. If the global basis is not a sufficient recycle space and we need to expand it, we clearly need to solve more full order model systems, which we do with recycling. The problem is that $\mathbf{V}$ might have too many columns to use it as a recycle space, so in the next sections we explain how to get around this issue.

5.2.1 Recycling on Updated ROM Equations

To keep consistent with Section 3.4, let k be fixed and set $\mathbf{A} := \tilde{\mathbf{A}}(\mathbf{p}_k)$, $\mathbf{b}_j = \mathbf{B}(:,j)$, and $\mathbf{x}_j = \mathbf{X}_k(:,j)$. We begin by assuming $\mathbf{A}\mathbf{V} = \tilde{\mathbf{K}}$, finding the QR factorization $\tilde{\mathbf{K}} = \mathbf{K}\mathbf{R}$, and then setting[†] $\mathbf{U} = \mathbf{V}\mathbf{R}^{-1}$, so that $\mathbf{A}\mathbf{U} = \mathbf{K}$.

As was shown in Section 3.4, the optimal solution in Range$(\mathbf{U})$ = Range$(\mathbf{V})$ is $\mathbf{z} = \mathbf{U}\mathbf{K}^T\mathbf{b}_j = \mathbf{V}(\mathbf{R}^{-1}\mathbf{K}^T\mathbf{b}_j)$, and the initial residual is $(\mathbf{I} - \mathbf{K}\mathbf{K}^T)\mathbf{b}_j$. If the relative initial residual is small, then we are done and have the approximate solution. If the relative initial residual is not small enough, then we need to expand $\mathbf{U}$. This is done by using Lanczos to form a basis, $\mathbf{V}_m$ for the Krylov subspace generated by the projected matrix $(\mathbf{I} - \mathbf{K}\mathbf{K}^T)\mathbf{A}$ and projected right-hand side $(\mathbf{I} - \mathbf{K}\mathbf{K}^T)\mathbf{b}_j$. An approximate solution is then found in Range$([\mathbf{U}, \mathbf{V}_m])$. This would be the ideal way to expand our search space, but in general we cannot afford to do this since the number of columns in $\mathbf{V}$, hence $\mathbf{K}$, is not expected to be small and the reorthogonalization would be too expensive. In addition, we may not need to find an approximation to $\mathbf{x}$ directly, which is the case for updating the global basis

[†] In practice, $\mathbf{U}$ is formed without inverting $\mathbf{R}$ explicitly.

approximation since we only need information *that is not already reconstructable from Range*$(\mathbf{V}) =$ *Range*$(\mathbf{U})$.

If we decompose $\mathbf{b}_j$ using the orthogonal projector $\mathbf{K}\mathbf{K}^T$ we discover something important:

$$\begin{aligned}
\mathbf{A}\mathbf{x}_j &= (\mathbf{I} - \mathbf{K}\mathbf{K}^T)\mathbf{b}_j + \mathbf{K}\mathbf{K}^T\mathbf{b}_j \\
\mathbf{A}\mathbf{x}_j - \mathbf{K}\mathbf{K}^T\mathbf{b}_j &= (\mathbf{I} - \mathbf{K}\mathbf{K}^T)\mathbf{b}_j \\
\mathbf{A}\mathbf{x}_j - \mathbf{A}\mathbf{U}\mathbf{K}^T\mathbf{b}_j &= \mathbf{r}_j \\
\mathbf{A}\underbrace{(\mathbf{x}_j - \mathbf{U}\mathbf{K}^T\mathbf{b}_j)}_{\mathbf{g}_j} &= \mathbf{r}_j. \qquad (5.13)
\end{aligned}$$

Notice that the vector $\mathbf{g}_j$ is the correction to the initial guess $\mathbf{U}\mathbf{K}^T\mathbf{b}_j$. Therefore, if we want to obtain the information not already in Range$(\mathbf{V}) =$ Range$(\mathbf{U})$ to construct the global basis matrix, we should consider an iterative solution to (5.13). We have one such system for each right-hand side, $j = 1, \ldots, n_{src} + n_{det}$, and we also have a sequence of systems. We can adopt a recycling approach if we choose our recycle space carefully.

Suppose $\mathbf{r}_j$ is not already small and we want to solve (5.13), or alternatively, we want to find

$$\min_{\mathbf{g}_j \in \mathcal{S}} \|\mathbf{r}_j - \mathbf{A}\mathbf{g}_j\|_2,$$

for suitable subspace $\mathcal{S}$[‡]. It is too expensive to use all the columns of $\mathbf{V}$ as a recycle space to generate $\mathbf{V}_m$ because of the required orthogonalization against Range(K). Due to this cost, we will instead use a subset of the columns of $\mathbf{V}$ for right-hand side j to be the recycle space, which we will expand immediately.

This means that we find $\mathbf{U}_j \in \mathbb{R}^{N \times n_j}$ where $\mathbf{U}_j \subseteq$ Range$(\mathbf{U})$ and such that $\mathbf{A}\mathbf{U}_j = \mathbf{K}_j$, where $\mathbf{K}_j^T\mathbf{K}_j = \mathbf{I}$. We now have our suitable subspace $\mathcal{S} =$ Range$([\mathbf{U}_j, \mathbf{V}_m^{(j)}])$,

[‡] Note that $\mathcal{S}$ should not be Range$(\mathbf{U})$, as then the solution is zero.

where the $\mathbf{V}_m^{(j)}$ are the Lanczos vectors for $\mathcal{K}_m\left((\mathbf{I}-\mathbf{K}_j\mathbf{K}_j^T)\mathbf{A},(\mathbf{I}-\mathbf{K}_j\mathbf{K}_j^T)\mathbf{r}_j\right)$ (compare to (3.8)) to look for a solution. That is, we want to solve

$$\min_{\mathbf{z},\mathbf{y}}\left\|\mathbf{r}_j-\mathbf{A}[\mathbf{V}_m^{(j)},\mathbf{U}_j]\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix}\right\|.$$

An important consequence of this choice for $\mathbf{U}_j$ is that if we apply the projector $(\mathbf{I}-\mathbf{K}_j\mathbf{K}_j^T)$ to both sides of (5.13), the right-hand side $\mathbf{r}_j$ remains unchanged because of the relationship between $\mathbf{K}_j$ and $\mathbf{K}$, meaning $(\mathbf{I}-\mathbf{K}_j\mathbf{K}_j^T)\mathbf{r}_j=\mathbf{r}_j$ Therefore, we have $\mathbf{v}_1^{(j)}=\mathbf{r}_j/\|\mathbf{r}_j\|$. Next, we use the Lanczos recurrence with $(\mathbf{I}-\mathbf{K}_j\mathbf{K}_j^T)\mathbf{A}$ and $\mathbf{v}_1^{(j)}$ to generate the recurrence relation

$$\begin{aligned}(\mathbf{I}-\mathbf{K}_j\mathbf{K}_j^T)\mathbf{A}\mathbf{V}_m^{(j)} &= \mathbf{V}_{m+1}^{(j)}\underline{T}_m^{(j)}\Leftrightarrow\\ \mathbf{A}\mathbf{V}_m^{(j)} &= \mathbf{K}_j\mathbf{K}_j^T\mathbf{A}\mathbf{V}_m^{(j)}+\mathbf{V}_{m+1}^{(j)}\underline{T}_m^{(j)}.\end{aligned}\tag{5.14}$$

Then $\mathbf{y},\mathbf{z}$ are found by (compare to (3.9)) solving

$$\min_{\mathbf{y},\mathbf{z}}\left\|\begin{bmatrix}0\\ \xi\mathbf{e}_1\end{bmatrix}-\begin{bmatrix}\mathbf{I} & \mathbf{K}_j^T\mathbf{A}\mathbf{V}_m^{(j)}\\ 0 & \underline{T}_m^{(j)}\end{bmatrix}\begin{bmatrix}\mathbf{z}\\ \mathbf{y}\end{bmatrix}\right\|_2.$$

Therefore, we have $\mathbf{g}_j=\mathbf{V}_m^{(j)}\mathbf{y}+\mathbf{U}_j\mathbf{z}$, where $\mathbf{z}=-\mathbf{K}_j^T\mathbf{A}\mathbf{V}_m^{(j)}\mathbf{y}$ and $\mathbf{y}_m^{(j)}:=\mathbf{V}_m^{(j)}\mathbf{y}$ is generated by a short term recurrence. We recover $\mathbf{x}_j$ by

$$\mathbf{x}_j=\mathbf{y}_m^{(j)}-\mathbf{U}_j\mathbf{K}_j^T\mathbf{A}\mathbf{y}_m^{(j)}+\mathbf{U}\mathbf{K}^T\mathbf{b}_j\tag{5.15}$$

Because we have set it up so that $\mathrm{Range}(\mathbf{U}_j)\subseteq\mathrm{Range}(\mathbf{U})=\mathrm{Range}(\mathbf{V})$, the information about $\mathbf{x}_j$ that cannot already be expressed using the columns of $\mathbf{V}$ is $\mathbf{V}_m^{(j)}\mathbf{y}$. Therefore, it would make sense to only use this information to update the global basis. This process is repeated for any right-hand side for which the initial residual $\mathbf{r}_j$ is not already small enough[§]. For each k, the maximum number of columns we

[§] The integer m for which the solution estimate is good enough will vary depending on the system, that is, $m=m_j$ – but for ease in notation we have omitted the subscript on m.

add to $\mathbf{V}$ is $n_{src} + n_{det}$, but in theory, we may add substantially fewer columns.

In the next subsections we discuss the stopping criterion and how to construct the initial global basis $\mathbf{V}$ and specify $\mathbf{U}_j$ from the columns of $\mathbf{U}$ when we do need to solve (5.13).

5.2.2 Stopping Criterion

We now explain the tolerance used as a stopping criterion for solving the projected problem. Define the residual of the projected problem as

$$\mathbf{r}_p = (\mathbf{I} - \mathbf{K}\mathbf{K}^T)\mathbf{b}_j - (\mathbf{I} - \mathbf{K}_j\mathbf{K}_j^T)\mathbf{A}\mathbf{y}_m^{(j)}. \tag{5.16}$$

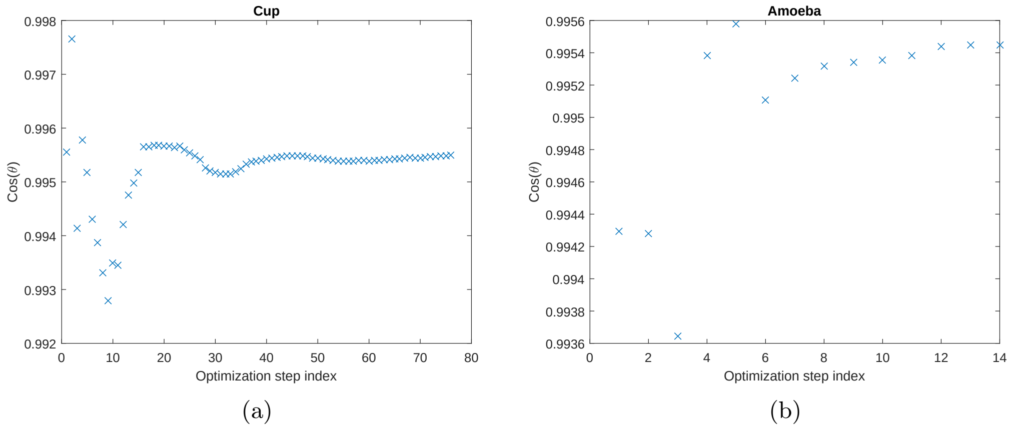
The goal is to have the following,

$$\frac{||\mathbf{b}_j - \mathbf{A}\mathbf{x}_j||}{||\mathbf{b}_j||} < tol, \tag{5.17}$$

where $\mathbf{x}_j \in \text{Range}([\mathbf{U}_j, \mathbf{V}_m^{(j)}])$ and *tol* is the desired tolerance. Using (5.15), we have that $\mathbf{r}_j = \mathbf{r}_p$, where $\mathbf{r}_j$ is the updated residual for the original problem. Therefore, to ensure (5.17) we let the stopping criterion for the projected problem be $||\mathbf{r}_p|| < tol * ||\mathbf{b}_j||$.

5.2.3 Identifying Recycle Spaces

As stated earlier, the authors in [43] use a different recycle space for each right-hand side, but each recycle space contains a common subspace. This common subspace is an approximate invariant subspace corresponding to the smallest eigenvalues because the authors observed that that subspace remained relatively unchanged throughout the optimization. Even though in [43] they work on a different parametric inverse problem for DOT, we observe the same for our problem. For our DOT problem, we also observe that the invariant subspace due to the smallest several eigenvalues for the $\tilde{\mathbf{A}}(\mathbf{p}_k)$ remains unchanged. Figure 5.1 shows how close the subspaces corresponding to the 10 smallest eigenvalues remain to the initial subspace for two numerical



(a) (b)

Figure 5.1: Evolution of the subspace gap (cosine of the largest canonical angle θ) between the initial and subsequent eigenvectors corresponding to the 10 smallest eigenvalues over the course of the optimization.

examples. We give the cosine of the largest canonical angle at each optimization step. In both test problems, the cosine of the largest canonical angles remains close to 1, meaning the subspaces are quite close. Therefore, we adopt the same approach here. We use a different recycle space $\mathbf{U}_j$ for each right-hand side, but seed each with the same invariant subspace, plus right-hand-side specific information.

In order to obtain our invariant subspace, we compute (approximate) eigenvectors of $\tilde{\mathbf{A}}_0$ that correspond to the smallest eigenvalues. Since the small eigenvalues of the $\tilde{\mathbf{A}}_k$ matrices remain close from one system to the next, this suggests that the corresponding invariant subspaces also remain close. We refer the reader to [43] for more details and theory on the invariant subspace. We have found experimentally that 10 eigenvectors is sufficient for the invariant subspace, while keeping the recycle space small. We set $\mathbf{U}_0 \in \mathbb{R}^{n\times 10}$ to contain (estimates of) those 10 vectors. This technique of finding an accurate invariant subspace, even though it might be expensive to find, so it can be deflated from the right-hand side has proven to be worthwhile in other large-scale applications, such as QCD [1, 67].

We set the initial global basis and recycle space to be $\mathbf{V} = ([\mathbf{U}_0, \mathbf{X}_0])$ and $\mathbf{U}_j = [\mathbf{U}_0, \mathbf{X}_0(:, j)]$. $\mathbf{V}$ has the invariant subspace and initial solutions to all the right-hand sides, while $\mathbf{U}_j$ has the invariant subspace and only the solution from the j^{th} right-hand side. If we need to perform recycling as outlined above, meaning $\mathbf{r}_j$ is not suitably small, then we update both $\mathbf{V}$ and $\mathbf{U}_j$ with $\mathbf{y}_m^{(j)}$. The $\mathbf{y}_m^{(j)}$ is appended to $\mathbf{V}$ every time we need to do recycling, while it is only appended to $\mathbf{U}_j$ if we are working on the j^{th} right-hand side. This ensures that $\mathbf{V}$ contains information pertinent to the entire system, while $\mathbf{U}_j$ is kept small.

5.2.4 Inner-Outer Krylov Recycling Algorithm

Algorithm 2 describes our dynamic inner-outer recycling process. Details on efficient implementation will be addressed in the next subsection.

Algorithm 2: Recycling and Global Basis Construction

```
$\mathbf{U}_0 \Leftarrow 10$ eigenvectors of $\tilde{\mathbf{A}}_0$, $\mathbf{X}_0$ solves $\tilde{\mathbf{A}}_0\mathbf{X}_0 = \mathbf{B}$
$\mathbf{V} \Leftarrow$ basis for Range$([\mathbf{U}_0, \mathbf{X}_0])$
$\mathbf{U}_j \Leftarrow [\mathbf{U}_0, \mathbf{X}_0(:, j)]$
for $i = 1 : K$ % for each interpolation point $i$ do
    for $j = 1 : nrhs$ do
        % Check if $\mathbf{V}$ is a good enough space
        $\tilde{\mathbf{K}} = \tilde{\mathbf{A}}_i\mathbf{V}$
        $[\mathbf{K}, \mathbf{R}] = qr(\tilde{\mathbf{K}}, 0)$
        $\mathbf{V} = \mathbf{V}/\mathbf{R}$ %implicit only; now $\mathbf{U}, \mathbf{V}$ same
        $\mathbf{r}_j = \mathbf{B}(:, j) - \mathbf{K}\mathbf{K}^T\mathbf{B}(:, j)$
        if $\frac{\|\mathbf{r}_j\|}{\|\mathbf{B}(:,j)\|} > tol$ then
            % MINRES recycling using $\mathbf{U}_j$
            $\tilde{\mathbf{K}}_j = \tilde{\mathbf{A}}_i\mathbf{U}_j$ % have already done this product
            $[\mathbf{K}_j, \mathbf{R}] = qr(\tilde{\mathbf{K}}_j, 0)$ % need not be done from scratch
            $\mathbf{U}_j = \mathbf{U}_j/\mathbf{R}$
            Solve $(\mathbf{I} - \mathbf{K}_j\mathbf{K}_j^T)\tilde{\mathbf{A}}_i\mathbf{y}_m^{(j)} = \mathbf{r}_j$ with MINRES
            $\mathbf{V} \Leftarrow [\mathbf{V}, \mathbf{y}_m^{(j)}]$
            $\mathbf{U}_j \Leftarrow [\mathbf{U}_j, \mathbf{y}_m^{(j)}]$
        end
    end
end
```

5.2.5 Algorithm Analysis

In this subsection, we give an analysis of Algorithm 2. Our approach is able to solve the sequence of systems efficiently while generating the global basis with only non-redundant information. We explain how this approach is an improvement to the approach given in Section 4.5 as we eliminate unnecessary solves, no longer need to find the expensive SVD, and eliminate the need to truncate the singular values and vectors in an ad hoc manner. We also explain how this approach is different from using recycling as in [43] to solve the systems and then doing an SVD to get the global basis. Additionally, we show how the cost is kept down by only updating $\mathbf{V}$ and $\mathbf{U}_j$ one column at a time.

5.2.5.1 System Solves

In Step 1 of Algorithm 1, we need to find the $\mathbf{X}_k$. Two ways this could be done are to simply use MINRES directly for each right-hand side or we could use recycling with the $\mathbf{U}_j$ for the respective right-hand side, across all the systems. This type of recycling approach is essentially what the authors did in [43], with recycle spaces having common invariant subspace information but tailored to the particular right-hand side. But they do further tuning of the recycle spaces to account for where one is in the optimization process. Recycling with just $\mathbf{U}_j$ consists of generating $\hat{\mathbf{V}}_m^{(j)}$ as a basis for $\mathcal{K}_m((\mathbf{I}-\mathbf{K}_j\mathbf{K}_j^T)\mathbf{A},(\mathbf{I}-\mathbf{K}_j\mathbf{K}_j^T)\mathbf{b}_j)$, with the intent of approximating $\mathbf{x}_j$ over Range$([\mathbf{U}_j,\hat{\mathbf{V}}_m^{(j)}])$. In contrast, the recycling approach presented in this chapter solves (5.13) and $\mathbf{V}_m^{(j)}$ is generated as a basis for $\mathcal{K}_m((\mathbf{I}-\mathbf{K}_j\mathbf{K}_j^T)\mathbf{A},(\mathbf{I}-\mathbf{K}\mathbf{K}^T)\mathbf{b}_j)$. This Krylov space differs from the previous one by the right-hand side. We also approximate $\mathbf{x}_j-\mathbf{x}_{0,j}$, with $\mathbf{x}_{0,j}=\mathbf{U}\mathbf{K}^T\mathbf{b}_j$ rather than $\mathbf{x}_j$. We argue that the new approach is an improvement over both of these alternatives. A numerical comparison of these methods is provided in Subsection 5.2.5.3.

It should be clear that recycling using $\mathbf{U}_j$ must have some advantage over using MINRES without recycling. Since $\mathbf{U}_j$ contains an approximate invariant subspace, MINRES on the projected system would converge as if part of the spectrum has been deflated. Additionally, the right-hand sides of (5.13) are residuals that are $\mathbf{b}_j$'s orthogonalized against the entire $\mathbf{K}$, not just the (much smaller) $\mathbf{K}_j$. This means that the residuals have been made small across spectral components other than just those included in the invariant subspace. We refer the reader to Section 5.2 in [43] for an argument for why $\mathbf{r}_j$ is small in norm. The argument assumes that the difference

$\tilde{\mathbf{A}}_0 - \tilde{\mathbf{A}}_1$ is small over the invariant subspace of $\tilde{\mathbf{A}}_0$ corresponding to the smallest eigenvalues, which are the smooth modes, and uses the fact that the columns of $\mathbf{X}_0$ are relatively smooth. In our setting, if we assume that the values of absorption in the object and in the background are known and optimize only for the shape parameters, $\tilde{\mathbf{A}}_0 - \tilde{\mathbf{A}}_1$ is diagonal, possibly low rank, with smooth modes made small by the operator.

5.2.5.2 Global Basis

It is natural to ask why $\mathbf{V}$ is a good global basis for the ROM problem. The global basis includes solution information and an approximate invariant subspace corresponding to the smoothest modes for $\tilde{\mathbf{A}}_0$. Following Theorem 4.1 in [43], we expect the invariant subspace consisting of eigenvectors corresponding to the smallest eigenvalues remain close, if the changes to $\tilde{\mathbf{A}}_k$ are concentrated over the high frequency modes. Since the $\mathbf{X}_k$ are expected to be smooth, it would be helpful to include this information in $\mathbf{V}$ for ROM. Looking at this numerically, Figure 5.2 shows, in log scale, the absolute values of the coefficients of the solutions $\mathbf{X}_5$ in the directions of $\mathbf{V}$ for Experiment 1 in Section 5.3. Note that these solutions were not used to build $\mathbf{V}$. We see that the solutions have large components in the directions of the invariant subspace of $\tilde{\mathbf{A}}_0$, as well as the corresponding column of $\mathbf{X}_0$.

5.2.5.3 Implementation Issues

In the final subsection of analysis, we discuss the cost of Algorithm 2 and how we keep the cost down by adding one column at a time to $\mathbf{V}$ and $\mathbf{U}_j$.

As defined above, $\mathbf{U}_0$ is a matrix of k_u columns containing a basis for the invariant subspace of $\tilde{\mathbf{A}}_0$ corresponding to the smallest eigenvalues and $\mathbf{X}_0$ is the solution to the initial system. Since the initial guess is used for many experiments, $\mathbf{X}_0$ and $\mathbf{U}_0$ are precomputed off-line and reused for other experiments. Referring to (5.12) we let $\tilde{\mathbf{A}}_k = \tilde{\mathbf{A}}_* + \Delta_k$, where $\tilde{\mathbf{A}}_*$ is fixed and $\Delta_k = \mathrm{diag}(\mu(\mathbf{p}_k))$.

We now consider the cost of solving System 1 using Algorithm 2. In Algorithm 2 line 7, $\tilde{\mathbf{K}} \coloneqq (\tilde{\mathbf{A}}_* + \Delta_1)[\mathbf{U}_0, \mathbf{X}_0]$. Since $\tilde{\mathbf{A}}_*$ is fixed, we precompute and save the

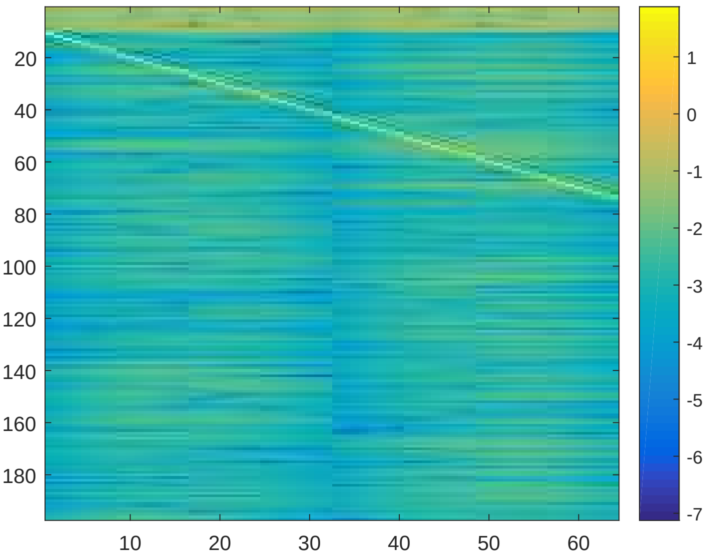


Figure 5.2: Logarithmically scaled image of the absolute values of the coefficients of solutions $\mathbf{X}_5$ in the directions of $\mathbf{V}$, where $\mathbf{V}$ has 197 columns. Recall that $\mathbf{X}_5$ has 32 columns to source positions, and 32 columns corresponding to receiver positions. The first 10 columns of $\mathbf{V}$ contain the invariant subspace of $\tilde{\mathbf{A}}_0$ corresponding to its smallest 10 eigenvalues, the next 64 columns correspond to $\mathbf{X}_0$, and the remaining columns have been constructed using the update procedure in Algorithm 2 for $K_* = 3$.

products $\tilde{\mathbf{A}}_*\mathbf{X}_0$ and $\tilde{\mathbf{A}}_*\mathbf{U}_0$, so we only have to compute $\Delta_1\mathbf{U}_0$ and $\Delta_1\mathbf{X}_0$.

Next, we need to compute the QR factorization of $\tilde{\mathbf{K}}$ in line 8. This is computed such that full re-orthogonalization is not required each time it is repeated. First, partition $\tilde{\mathbf{K}} = [\tilde{\mathbf{K}}_a, \tilde{\mathbf{K}}_b]$, where the first block corresponds to the number of columns of $\mathbf{U}_0$. Then, compute $\mathbf{Q}_1\mathbf{R}_1 = \tilde{\mathbf{K}}_a$. Next, compute $(\mathbf{I} - \mathbf{Q}_1\mathbf{Q}_1^T)\tilde{\mathbf{K}}_b = \tilde{\mathbf{K}}_b - \mathbf{Q}_1(\mathbf{Q}_1^T\tilde{\mathbf{K}}_b) = \mathbf{Q}_2\mathbf{R}_2$. So, $\tilde{\mathbf{K}}_b = \mathbf{Q}_2\mathbf{R}_2 + \mathbf{Q}_1(\mathbf{Q}_1^T\tilde{\mathbf{K}}_b)$. It follows that a QR factorization is

$$\begin{aligned}[\tilde{\mathbf{K}}_a, \tilde{\mathbf{K}}_b] &= [\mathbf{Q}_1\mathbf{R}_1, \mathbf{Q}_1(\mathbf{Q}_1^T\tilde{\mathbf{K}}_b) + \mathbf{Q}_2\mathbf{R}_2] \\ &= [\mathbf{Q}_1, \mathbf{Q}_2]\begin{bmatrix} \mathbf{R}_1 & \mathbf{Q}_1^T\tilde{\mathbf{K}}_b \\ 0 & \mathbf{R}_2 \end{bmatrix}.\end{aligned}$$

We let $\mathbf{V} = [\mathbf{U}_0, \mathbf{X}_0]\mathbf{R}^{-1}$, noting that $\mathbf{R}^{-1}$ need not be applied explicitly. We now have the required factorization, $(\tilde{\mathbf{A}}_* + \Delta_1)\mathbf{V} = \mathbf{K}$, where $\mathbf{K}$ has orthonormal columns.

Suppose we do need to do recycling for System 1 and right-hand side 1. This means that in Algorithm 2 line 13, we need to find $\tilde{\mathbf{K}}_1 \coloneqq (\tilde{\mathbf{A}}_* + \Delta_1)[\mathbf{U}_0, \mathbf{X}_0(:,1)]$. We have already formed this product above, so we just have to select the right columns of $\tilde{\mathbf{K}}$. Moreover, the QR factorization of $\tilde{\mathbf{K}}_1$ is computed from $\mathbf{Q}_1$ and $\mathbf{R}_1$. All we need to compute is $\tilde{\mathbf{K}}_1(:, k_u + 1) - \mathbf{Q}_1(\mathbf{Q}_1^T \tilde{\mathbf{K}}_1(:, k_u + 1))$ and then normalize it. The normalization constant becomes the lower right corner component of the upper triangular matrix.

We then solve the projected problem with MINRES and append the solution, $\mathbf{y}_m^{(1)}$, to $\mathbf{V}$ and $\mathbf{U}_j$. Now, we move to the second right-hand side and check to see if the newly enlarged $\mathbf{V}$ is sufficient to represent the solution. In order to do this, we compute

$$(\tilde{\mathbf{A}}_* + \Delta_1)[\mathbf{V}, \mathbf{y}_m] = [(\tilde{\mathbf{A}}_* + \Delta_1)\mathbf{V}, (\tilde{\mathbf{A}}_* + \Delta_1)\mathbf{y}_m] = [\mathbf{K}, \mathbf{z}].$$

Since $\mathbf{K}$ already has orthogonal columns, we need only to compute

$$[\mathbf{K}, \mathbf{z}] = [\mathbf{K}, \mathbf{q}] \underbrace{\begin{bmatrix} \mathbf{I} & \mathbf{K}^T\mathbf{z} \\ 0 & \rho \end{bmatrix}}_{\hat{\mathbf{R}}},$$

where $\mathbf{z} - \mathbf{K}(\mathbf{K}^T\mathbf{z}) = \rho\mathbf{q}$, so we have $\mathbf{K} \leftarrow [\mathbf{K}, \mathbf{q}], \mathbf{V} \leftarrow [\mathbf{V}, \mathbf{y}_m]\hat{R}^{-1}$.

If recycling is required for the second right-hand side, we follow the same procedure above. Therefore, for any additional right-hand sides, only incremental new calculations are needed.

5.3 Numerical Experiments

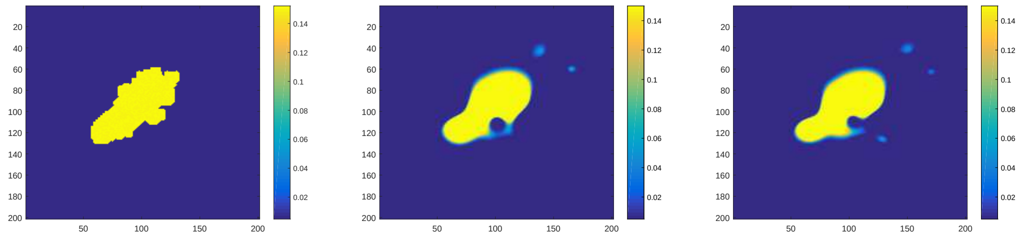
In this section, we provide four numerical experiments, three 2-dimensional examples and one 3-dimensional example. For the 2D examples, we show two experiments where the global basis is created for each problem in order to show the MINRES iteration counts. We also provide a 2D experiment where the global basis is reused, in order to show that with this new method, we can still reuse the global basis. The set-up for the experiments is the same as in Section 4.8. Results are reported for both using the full order model and the ROM to compute the function and Jacobian evaluations for each experiment. We used $K_* = 3$ systems for each experiment to create the reduced order model space. The tolerance in line 11 of Algorithm 1 was set to be 10^{-7}. All of the experiments were run using a laptop with a 3.20 GHz processor and 16.0 GB RAM using MATLAB R2015b.

5.3.1 2D Experiments

The first three experiments were solved on a 201×201 mesh, which gives us $40,401$ degrees of freedom for the forward problem. We use 32 sources and 32 detectors. Following the PaLs approach, we use 25 compactly supported radial basis functions, which results in 100 parameters for the optimization problem (2.8). Figure 4.8 shows the absorption image using the initial set of parameters. The projection bases that are used for Experiments 1 and 2 were created with the first 3 iterations of the optimization using the full order model for Experiment 1.

In Experiment 1, we needed to solve 187 large, single right-hand side systems to generate what we needed to construct the global basis matrix (note that the 64 of these corresponding to $\mathbf{X}_0$ could have been pre-computed off-line). Including the additional 10 eigenvectors of $\tilde{\mathbf{A}}_0$ that were used as the first 10 columns of $\mathbf{V}$, $\mathbf{V}$ has 197 columns and thus the reduced model has order 197. Therefore, the reduced models require solutions to linear systems of size 197×197 rather than $40,401 \times 40,401$ for the full order model.

The optimization using the full order model for Experiment 1 required 30 function evaluations and 15 Jacobian evaluations. In comparison, the optimization run using the reduced order model, once it has been generated, required 28 function evaluations and 14 Jacobian evaluations, indicating that using a ROM in place of FOM does not greatly impact convergence rate of the optimization. Figure 5.3 shows the reconstructions for Experiment 1. The bottom line is that solving the optimization using the full order model requires the solution of 1,440 systems of size $40,401 \times 40,401$. On the other hand, solving using our approach requires solution



(a) Ground truth absorption image.

(b) Reconstruction using the full order model.

(c) Reconstruction using the reduced order model.

Figure 5.3: Results for Experiment 1. Reconstruction on a 201 × 201 mesh, resulting in 40, 401 degrees of freedom in the forward model and 197 degrees of freedom in the reduced model for the forward model. 32 sources, 32 detectors, and 25 basis functions were used.

of 187 systems of size 40, 401 × 40, 401, which are used to construct $\mathbf{V}$ during the first few optimization steps. The remainder of the work is in solving systems of size 197 × 197 until the convergence tolerance for the optimization is achieved.

Table 5.1 also includes the number of (unpreconditioned) MINRES iterations for each experiment with and without recycling. Although the tables only show a sample of results, it is clear that the iterations decrease from one right-hand side to the next, and system to system, using our approach. The jump in number of iterations for right-hand-side 33 comes from the fact that we concatenated $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{C}}$ to form one right-hand-side for the symmetric transfer function, so the 33$^{\text{rd}}$ right-hand side corresponds to the first column in $\tilde{\mathbf{C}}$.

In Experiment 2, a total 188, 40, 401 × 40, 401 single-right-hand side systems were solved to compute our global basis. The reduced order model has order 198. Therefore, the reduced models require solutions to linear systems of size 198 × 198 rather than 40, 401 × 40, 401 for the full order model. The optimization using the full order model required 126 function evaluations and 78 Jacobian evaluations. The optimization run using our reduced order model took 123 function evaluations and 76 Jacobian evaluations to converge to our stopping criterion, so again, there is no negative impact on convergence rate by replacing the FOM with the ROM. The difference in the total number of large (40, 401 × 40, 401) single-right-hand side systems that need to be solved, though, is even more pronounced in this example than in

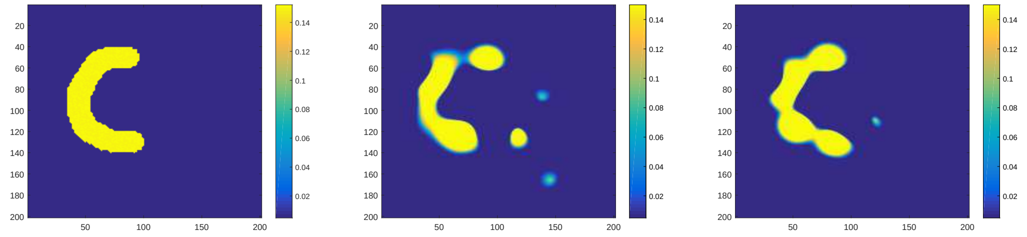
		Experiment 1		Experiment 2	
System	RHS	MINRES Its	MINRES Its with Recycling	MINRES Its	MINRES Its with Recycling
1	1	463	140	470	127
	20	541	52	506	53
	32	487	0	493	0
	33	467	124	470	120
	53	528	57	514	48
	64	489	0	494	0
2	1	474	118	501	124
	20	513	35	545	38
	32	497	0	526	5
	33	474	105	500	132
	53	526	37	567	127
	64	497	0	532	0

Table 5.1: Number of MINRES iterations for Experiments 1 and 2. Zero iterations in the table indicates that the recycle space $\mathbf{V}$ was a good enough space to look for the solution and MINRES recycling using $\mathbf{U}_j$ did not need to be performed.

the last: 6,528 are needed for the FOM approach vs. only 188 for the ROM approach. Moreover, the work involved in solving for the latter systems is reduced, since MINRES requires fewer iterations due to the recycling.

Figure 5.4 shows the reconstructions for Experiment 2. Again, Figure 5.1 shows the number of unpreconditioned MINRES iterations for each experiment with and without our inner-outer recycling approach.

In Experiment 3, we used the same problem as in Experiment 2, but with the ROM formed from the basis made in Experiment 1. Therefore, for this experiment

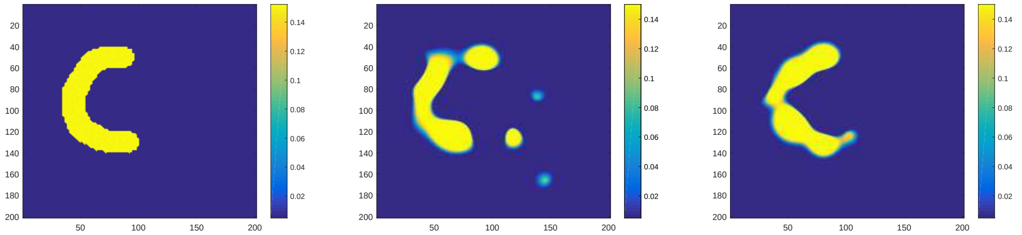


(a) Ground truth absorption image.

(b) Reconstruction using the full order model.

(c) Reconstruction using the reduced order model.

Figure 5.4: Results for Experiment 2. Reconstruction on a 201×201 mesh, resulting in $40,401$ degrees of freedom in the forward model and 198 degrees of freedom in the reduced order model. 32 sources, 32 detectors, and 25 basis functions were used.



(a) Ground truth absorption image.

(b) Reconstruction using the full order model.

(c) Reconstruction using the reduced order model.

Figure 5.5: Results for Experiment 3. Reconstruction on a 201×201 mesh, resulting in $40,401$ degrees of freedom in the forward model and 198 degrees of freedom in the reduced order model. 32 sources, 32 detectors, and 25 basis functions were used.

no additional full order model solves were required. The optimization using the full order model required 126 function evaluations and 78 Jacobian evaluations. The optimization run using our reduced order model took 32 function evaluations and 15 Jacobian evaluations to converge to our stopping criterion. For this particular experiment the reduced basis significantly reduced the number of function and Jacobian evaluations required. A possible explanation is that the full order model was getting stuck in the optimization and this basis eliminated this problem. Figure 5.5 shows the reconstructions for Experiment 3.

5.3.2 Value of Inner-Outer Recycling

There is a significant benefit to using two levels of recycling information. To see this, consider Algorithm 1 to construct the global basis. We could solve the full order model systems in Step 1 (e.g. systems in 5.11) with the unpreconditioned MINRES recycling approach in [43]. It is important to note that the recycle spaces would be different than those used in our new method. Furthermore, in the new method we solve the correction equations (5.13) as opposed to solving (5.11). For $j > 1$, the recycle spaces for the [43] approach do not incorporate information from other systems corresponding to other right-hand sides. In contrast, since we augment $\mathbf{V}$ from information about right-hand side j, we update $\mathbf{K}$. The update in $\mathbf{K}$ then causes updates to $\mathbf{r}_{j+1}, \ldots, \mathbf{r}_{n_{src}+n_{det}}$, which are the right-hand sides in (5.13).

Table 5.2 compares the recycling of [43] with our new approach. The results show that with our approach, the number of iterations and the relative residuals decrease as you move from one right-hand side to the next and also as you move from system to system. The jump at right-hand side 33 is due to the fact that you are moving to the second half of the concatenated right-hand sides, so these correspond to solving the adjoint problem. Using the approach in [43], however, does not speed up convergence across right-hand sides. In our approach, the reduced global basis, $\mathbf{V}$, is already constructed when we are done with the full order model solves. We note there is a big difference in total number of MINRES iterations to squeeze all information from systems 1 and 2. It took our approach $5,006$ iterations, while it took $22,659$ iterations for the recycling method in [43].

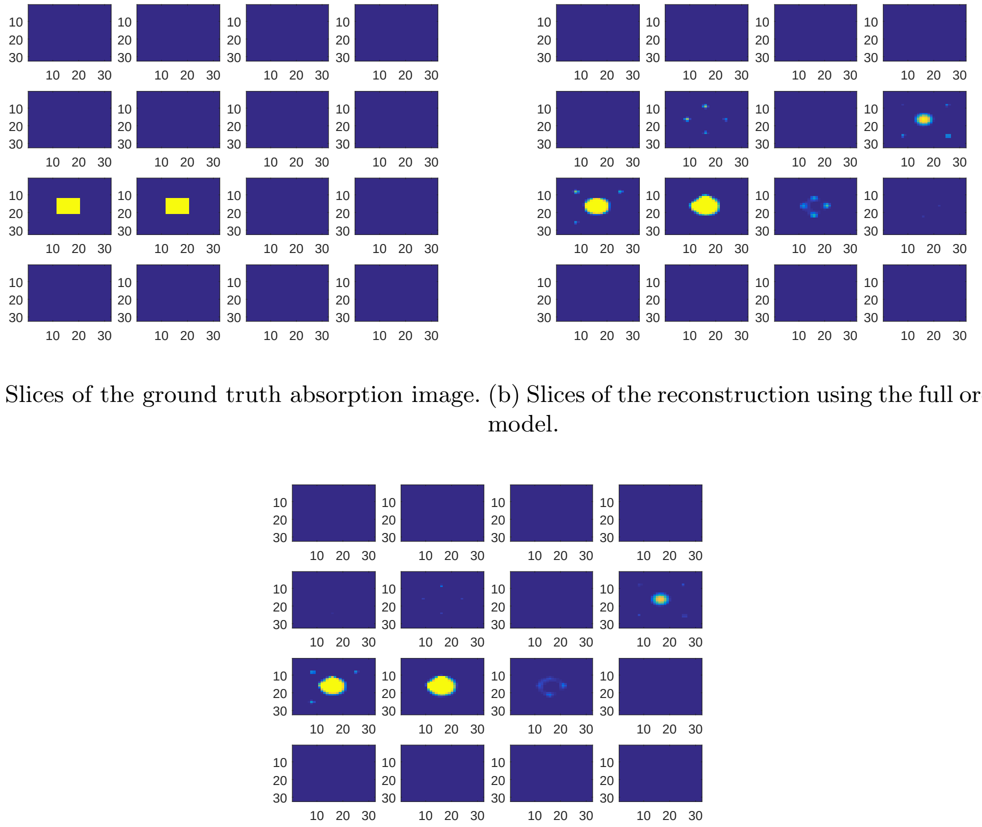
		Our Approach		Recycling from [43]	
System	RHS	Its	Initial Relative Residual	Its	Initial Relative Residual
1	1	140	7.523115e-05	140	7.523329e-05
	20	52	1.077645e-06	185	6.584893e-04
	32	0	8.866398e-08	152	1.164802e-04
	33	124	4.776975e-05	127	5.213653e-05
	53	57	1.692149e-06	191	6.114295e-04
	64	0	9.960690e-08	151	1.450962e-04
2	1	118	4.673235e-05	131	5.091877e-05
	20	35	6.653493e-07	190	6.171251e-04
	32	0	8.754708e-08	153	1.051619e-04
	33	105	2.588292e-05	129	3.388392e-05
	53	37	8.454303e-07	188	6.405050e-04
	64	0	7.960101e-08	151	9.958796e-05

Table 5.2: Comparison of MINRES recycling using $\mathbf{U}_j$ as described in [43] vs. the inner-outer approach using both $\mathbf{V}$ and $\mathbf{U}_j$ as described in Algorithm 2. Note that the two approaches lead to different choices for $\mathbf{U}_j$ as well as different systems to solve.

5.3.3 3D Experiments

The next experiment was solved on a $32\times32\times32$ mesh, which gives us $32,768$ degrees of freedom for the forward problem. We use 225 sources and 225 detectors. Following the PaLs approach, we use 27 compactly supported radial basis functions, which results in 135 parameters for the optimization problem. Figure 4.12 shows 16 slices of the absorption image using the initial set of parameters. The projection bases that are used for Experiment 3 were created with the first 3 iterations of the optimization using the full order model for Experiment 3. Due to the large number of right-hand sides in the 3D problem, we had to modify how to construct the reduced basis. The basis includes the approximate invariant subspace, the solution to the initial system, and the new information to solutions from every 8 sources and detectors from systems 2 and 3 found via recycling. We also had to solve the forward and adjoint problems separately, therefore we orthogonalized the concatenated solutions at the end. Ultimately, the reduced basis had 572 columns giving the reduced model order 572. Therefore, the reduced models require the solutions to linear systems of size 572×572, while the full order model requires solutions to linear systems of size $32,768 \times 32,768$.

The optimization using the full order model for Experiment 3 required 51 function evaluations and 29 Jacobian evaluations. Meanwhile, the optimization using the reduced order model for Experiment 3 required 57 function evaluations and 34 Jacobian evaluations. Once again indicating that the ROM does not greatly impact the convergence rate of the optimization. Figure 5.6 shows slices of the reconstructions for Experiment 3.



(a) Slices of the ground truth absorption image.

(b) Slices of the reconstruction using the full order model.

(c) Slices of the reconstruction using the reduced order model.

Figure 5.6: Results for Experiment 3. Reconstruction on a 32×32×32 mesh, resulting in $32,768$ degrees of freedom in the forward model and 572 degrees of freedom in the reduced order model. 225 sources, 225 detectors, and 27 basis functions were used.

Chapter 6

Inner-Outer Krylov Recycling for Shifted Systems

The need to solve large scale sequences of shifted linear systems with multiple right-hand sides arises in many important applications. As demonstrated in Section 4.2, this arises, for example, in DOT if the data is collected for multiple frequencies. The expense of solving these systems is the computational bottleneck of the larger problem in which they are involved. Many approaches have been developed to address the computational cost associated with solving sequences of shifted systems. Some methods use Lanczos recurrences or Arnoldi iterations, see e.g., [26, 43, 49, 61, 64].

Imagine solving an optimization problem where the function evaluation requires the solution of a linear shifted system with multiple right-hand sides over the course of the optimization. These system solves become the dominant computational cost of the optimization. We were first motivated to develop the method presented in this chapter by solving the optimization problem that arises in the nonlinear inverse problem in the context of diffuse optical tomography (DOT) [14, 28]. The goal of solving an inverse problem is to find an image of an unknown quantity. In order to find this image when the forward problem is nonlinear in the unknowns describing the desired quantity, we must solve an optimization problem, which in turn requires solving the forward problem multiple times. The forward model valuation, which requires the solution of a sequence of shifted systems, links the measured data to the unknown image we seek. These forward model solves are the main computational bottleneck of these image reconstruction problems.

In this chapter, we extend the 2-level Krylov recycling method found in Chapter 5 to systems with shifts in order to reduce this computational cost. First, we only consider a single right-hand side and then develop a method for multiple right-hand sides.

We are looking to solve shifted systems of the form

$$\left(\mathbf{A}^{(k)} + \gamma_\ell \mathbf{E}\right) \mathbf{x}^{(k,\ell)} = \mathbf{b}, \tag{6.1}$$

for symmetric $\mathbf{A}^{(k)}$ and $\mathbf{E}$ and several values of k and ℓ via recycling. The method we describe works for more general applications, but we will demonstrate on two examples from DOT.

Next, we are looking to solve shifted systems of the form

$$\left(\mathbf{A}^{(k)} + \imath\gamma_\ell \mathbf{I}\right) \mathbf{X}^{(k,\ell)} = \mathbf{B}, \tag{6.2}$$

for symmetric $\mathbf{A}^{(k)}$ and several values of k and ℓ via recycling for multiple right-hand sides. This is the form of system that arises in the DOT setting for the complex symmetric problem (5.6) and in this setting, we once again have two goals for the new method: solve the shifted systems efficiently, while constructing the global basis for use in ROM.

Both methods described in this chapter are used as efficient solvers and also to construct reduced order models from the output of the solvers. We remind the reader that Krylov recycling basics are given in Section 3.4 and inner-outer recycling is explained in Section 5.2. More on recycling is found in the literature, see [5,43,57,77].

6.1 Inner-Outer Recycling for Shifted Systems with a Single Right-Hand Side

The extension to shifted systems needs to address the following: how to recover the solution to the shifted problem and what information from the shifted solution to include in the recycle spaces. We consider solving

$$\left(\mathbf{A}^{(k)} + \gamma_\ell \mathbf{E}\right) \mathbf{x}^{(k,\ell)} = \mathbf{b}, \tag{6.3}$$

with a variant of inner-outer recycling, as we now describe.

6.1.1 The Method

Without loss of generality, we assume that the first value of γ_ℓ we wish to solve for is 0, and we need to solve (6.3) for L values of γ_ℓ, between $\gamma_1 = 0$ and γ_L. This is due to the fact we can always create a "zero shift" by letting $\widehat{\mathbf{A}^{(k)}} = (\mathbf{A}^{(k)} + \gamma_1 \mathbf{E})$ for the smallest shift, γ_1, we want to solve, and then we write the remaining shifts relative to γ_1: $\gamma_\ell = \gamma_1 + \widehat{\gamma_\ell}$, and solve the remaining systems by rewriting the matrices as $(\widehat{\mathbf{A}^{(k)}} + \widehat{\gamma_\ell}\mathbf{E})$.

As before, assume $\mathbf{U} \in \mathbb{R}^{N\times n_c}$ is generated from the recycle space $\tilde{\mathbf{U}}$ such that $\mathbf{A}^{(k)}\mathbf{U} = \mathbf{K}$ and $\mathbf{K}^T\mathbf{K} = \mathbf{I}$. Select $\tilde{\mathbf{U}}_\ell \in \mathbb{R}^{N\times n_\ell}$ from among the columns of $\tilde{\mathbf{U}}$. Generate $\mathbf{U}_\ell$ from $\tilde{\mathbf{U}}_\ell$, such that $(\mathbf{A}^{(k)} + \gamma_\ell \mathbf{E})\mathbf{U}_\ell = \mathbf{K}_\ell$ and $\mathbf{K}_\ell^T \mathbf{K}_\ell = \mathbf{I}$. For the shifted system, $\mathbf{U}_\ell$ is a shift specific recycle space instead of a right-hand side specific recycle space as in Section 5.2.

First, we estimate that the solution is in Range$(\mathbf{U})$ (i.e. it is $\mathbf{Uq}$ for some $\mathbf{q}$),

$$\begin{aligned} \mathbf{r}^{(k,\ell)} &= \mathbf{b} - \left(\mathbf{A}^{(k)} + \gamma_\ell \mathbf{E}\right)\mathbf{Uq} \\ &= \mathbf{b} - (\mathbf{Kq} + \gamma_\ell \mathbf{EUq}). \end{aligned}$$

Using Petrov-Galerkin projection, we put $\mathbf{K}^T\mathbf{r}^{(k,\ell)} = 0$, and find the estimate

$$\mathbf{x}^{(k,\ell)} \approx \mathbf{U}\left(\mathbf{I} + \gamma_\ell \mathbf{K}^T\mathbf{EU}\right)^{-1}\mathbf{K}^T\mathbf{b}. \tag{6.4}$$

When $\gamma_\ell = 0$, this gives the same initial solution as (3.7)

As in (5.13), we would like to solve for the incremental change from this initial guess, if the initial residual $\mathbf{r}^{(k,\ell)}$ is not small enough. Using $\mathbf{x}^{(k,\ell)} = \mathbf{g} + \mathbf{x}_i$, where $\mathbf{x}_i$ is the initial guess found on the right of (6.4), the goal is to recover $\mathbf{g}$. Therefore, the problem we want to solve is

$$\left(\mathbf{A}^{(k)} + \gamma_\ell \mathbf{E}\right)\mathbf{g} \quad = \quad \mathbf{b} - \left(\mathbf{A}^{(k)} + \gamma_\ell \mathbf{E}\right)\mathbf{x}_i$$

$$\begin{aligned} &= \mathbf{b} - (\mathbf{K} + \gamma_\ell \mathbf{E}\mathbf{U})\left(\mathbf{I} + \gamma_\ell \mathbf{K}^T\mathbf{E}\mathbf{U}\right)^{-1}\mathbf{K}^T\mathbf{b} \\ &= (\mathbf{I} - \mathbf{K}\mathbf{K}^T)\left(\mathbf{b} - (\mathbf{E}\mathbf{U})(\frac{1}{\gamma_\ell}\mathbf{I} + \mathbf{K}^T\mathbf{E}\mathbf{U})^{-1}\mathbf{K}^T\mathbf{b}\right). \qquad (6.5) \end{aligned}$$

The last line comes from the fact that $\mathbf{K}^T\mathbf{r}^{(k,\ell)} = 0$ and is of value in terms of illustrating the relationship to the residual for the $\gamma_\ell = 0$ system.

If the initial relative residual norm is not small enough, we solve

$$\min_{\mathbf{g}\in\mathcal{S}} \|\mathbf{r}^{(k,\ell)} - \left(\mathbf{A}^{(k)} + \gamma_\ell\mathbf{E}\right)\mathbf{g}\|_2,$$

over an appropriate $\mathcal{S}$. We want $\mathcal{S}$ to contain Range$(\mathbf{U}_\ell)$ and to generate the space with which to augment $\mathbf{U}_\ell$, we use the Krylov subspace generated by $(\mathbf{I} - \mathbf{K}_\ell\mathbf{K}_\ell^T)\left(\mathbf{A}^{(k)} + \gamma_\ell\mathbf{E}\right)$ and

$$\mathbf{v}_1 = \left(\left(\mathbf{I} - \mathbf{K}_\ell\mathbf{K}_\ell^T\right)\mathbf{r}^{(k,\ell)}\right)/\|\left(\mathbf{I} - \mathbf{K}_\ell\mathbf{K}_\ell^T\right)\mathbf{r}^{(k,\ell)}\|_2.$$

We call the reader's attention to the following important fact, which makes this inner-outer recycling approach different than the approach in Chapter 5: $\mathbf{K}$ is in Range$(\mathbf{A}^{(k)})$, while $\mathbf{K}_\ell$ is in Range$(\mathbf{A}^{(k)} + \gamma_\ell\mathbf{E})$. We obtain

$$\begin{aligned} (\mathbf{I} - \mathbf{K}_\ell\mathbf{K}_\ell^T)\left(\mathbf{A}^{(k)} + \gamma_\ell\mathbf{E}\right)\mathbf{V}_m &= \mathbf{V}_{m+1}\underline{T}_m \Leftrightarrow \\ \left(\mathbf{A}^{(k)} + \gamma_\ell\mathbf{E}\right)\mathbf{V}_m &= \mathbf{K}_\ell\mathbf{K}_\ell^T\left(\mathbf{A}^{(k)} + \gamma_\ell\mathbf{E}\right)\mathbf{V}_m + \mathbf{V}_{m+1}\underline{T}_m. \end{aligned}$$

We then find $\mathbf{y}, \mathbf{z}$ by solving

$$\min_{\mathbf{y},\mathbf{z}} \left\| \begin{bmatrix} \mathbf{K}_\ell^T\mathbf{r}^{(k,\ell)} \\ \xi\mathbf{e}_1 \end{bmatrix} - \begin{bmatrix} \mathbf{I} & \mathbf{K}_\ell^T\left(\mathbf{A}^{(k)} + \gamma_\ell\mathbf{E}\right)\mathbf{V}_m \\ 0 & \underline{T}_m \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{y} \end{bmatrix} \right\|_2, \qquad (6.6)$$

where $\xi = \|\left(\mathbf{I} - \mathbf{K}_\ell\mathbf{K}_\ell^T\right)\mathbf{r}^{(k,\ell)}\|_2$. Therefore, $\mathbf{g} = \mathbf{y}_m + \mathbf{U}_\ell\mathbf{z}$, where $\mathbf{y}_m = \mathbf{V}_m\mathbf{y}$, $\mathbf{z} = \mathbf{K}_\ell^T\mathbf{r}^{(k,\ell)} - \mathbf{K}_\ell^T\left(\mathbf{A}^{(k)} + \gamma_\ell\mathbf{E}\right)\mathbf{V}_m\mathbf{y}$ and we estimate the solution as

$$\mathbf{x}^{(k,\ell)} = \mathbf{y}_m + \mathbf{U}_\ell\mathbf{K}_\ell^T\mathbf{r}^{(k,\ell)} - \mathbf{U}_\ell\mathbf{K}_\ell^T\left(\mathbf{A}^{(k)} + \gamma_\ell\mathbf{E}\right)\mathbf{y}_m + \mathbf{x}_i. \qquad (6.7)$$

The $\mathbf{y}_m$ is used in a special way, described in the next subsection.

We have described how we use this method as an efficient solver to recover the desired estimate of the solution. If we wanted to also construct the reduced order model using this method, we could solve the necessary systems using this method and use $\mathbf{U}$ as the reduced basis. If the problem has multiple right-hand sides, then we could use this method for each right-hand side or use the method described in Section 6.2 to construct the reduced basis.

6.1.2 Identifying and Updating Recycle Spaces

Now that we have found the solution to the shifted problem via recycling, we discuss how to construct the recycle spaces. In Chapter 5, the initial recycle spaces included an invariant subspace consisting of 10 eigenvectors corresponding to the smallest eigenvalues and solutions to the initial system. This was because the corresponding subspace was found to be nearly invariant across all $\mathbf{A}^{(k)}$. In addition, $\mathbf{U}$ was seeded with solutions to the initial system for all right-hand sides, while $\mathbf{U}_j$ contained only the approximate invariant subspace and the solution to the j^{th} right-hand side. We adopt an analogous strategy here, but $\mathbf{U}$ has solutions to the initial system for all shifts and $\mathbf{U}_\ell$ has only the initial solution to the system corresponding to the shift γ_ℓ.

If the initial residual for shift ℓ is such that we need to perform the inner recycling, we append information from $\mathbf{y}_m$ to both $\mathbf{U}$ and $\mathbf{U}_\ell$. This ensures that $\mathbf{U}$ contains information pertinent to the entire set of shifted systems, while $\mathbf{U}_\ell$ is kept small. This is important for keeping computational costs down.

Note that appending a column to $\mathbf{U}$ means that $\mathbf{K}$ must also be increased by one column. We must (a) compute $\mathbf{A}^{(k)}\mathbf{y}_m$ and (b) orthogonalize the result against the previous columns of $\mathbf{K}$ to get our new $\mathbf{K}$. We then (c) update the initial residual estimate *for any shifted systems we have not yet solved* to reflect the projection onto the space as increased in dimension by 1. In this way, new information about the systems that are "close by" in terms of neighboring γ_ℓ is used to improve the current solution space.

The overhead involved in appending information to $\mathbf{U}_\ell$ is the cost of computing $(\mathbf{A}^{(k)} + \gamma_\ell \mathbf{E})\mathbf{y}_m$ (but note that $\mathbf{A}^{(k)}\mathbf{y}_m$ has already been computed) and then orthogonalizing the result against the previous columns of $\mathbf{K}_\ell$. Note again that we only append columns when there is a system for which the inner Krylov recycling became necessary to reduce the residual.

The next subsection outlines our algorithm (but note that the sequential updating approaches noted here have been replaced by more expensive calls to QR factorizations to keep the algorithm looking more tidy and make the general idea easier to follow).

6.1.3 The Algorithm

Algorithm 3 describes our inner-outer recycling process for shifted systems with one right-hand side.

6.1.4 Complex Identity Shift

We will now consider the special case of a complex identity shift, that is, $\mathbf{E} = \mathbf{I}$ and the shift is $\imath\gamma_\ell$. Therefore, the problem is

$$\left(\mathbf{A}^{(k)} + \imath\gamma_\ell \mathbf{I}\right)\mathbf{x}^{(k,\ell)} = \mathbf{b}. \tag{6.8}$$

The algorithm provided in the previous subsection works for this special case, but we need to be careful about how we construct and update $\mathbf{U}$ and $\mathbf{U}_\ell$. We initialize and update $\mathbf{U}$ and $\mathbf{U}_\ell$ just as in Algorithm 2, but if we are dealing with a complex shift, we only append the imaginary component of $\mathbf{x}^{(0,\ell)}$ and $\mathbf{y}_m$. This means that $\mathbf{U}$ and $\mathbf{K}$ remain real. Let $\tilde{\mathbf{U}}_\ell = \tilde{\mathbf{U}}(:, i_1 : i_2)$, and $[\mathbf{K}_\ell, \mathbf{R}] = (\mathbf{A}^{(k)} + \imath\gamma_\ell \mathbf{I})\tilde{\mathbf{U}}_\ell$, so $\mathbf{U}_\ell = \tilde{\mathbf{U}}_\ell \mathbf{R}^{-1}$. It is clear that $\mathbf{K}_\ell$, $\mathbf{U}_\ell$, and $\mathbf{V}_m$ are complex. In the case when many shifts are used, it may be necessary to add information from the real component of some $\mathbf{x}^{(0,\ell)}$ to the initial $\mathbf{U}$. This is because as you move farther away from the “zero shift” the solutions can change significantly and only adding the complex component is not sufficient. We also note that letting $\mathbf{E} = \mathbf{I}$ provides some savings in (6.5) and (6.6)

Algorithm 3: Krylov Recycling for Shifted Systems

```
$\mathbf{U}_0 \Leftarrow 10$ eigenvectors of $\mathbf{A}^{(0)}$
$\mathbf{X}^{(0)} \Leftarrow$ solutions to $\left(\mathbf{A}^{(0)} + \gamma_\ell \mathbf{E}\right)\mathbf{X}^{(0)} = \mathbf{b}$ for all $\gamma_\ell$
$\mathbf{U} \Leftarrow$ basis for Range($[\mathbf{U}_0, \mathbf{X}^{(0)}]$)
$\mathbf{U}_\ell \Leftarrow [\mathbf{U}_0, \mathbf{X}^{(0)}(:,\ell)]$
for $k = 1 : K$ do
    for $\ell = 1 : L$ do
        % Check if $\mathbf{U}$ is a good enough space
        $\tilde{\mathbf{K}} = \mathbf{A}^{(k)}\mathbf{U}$
        $[\mathbf{K}, \mathbf{R}] = qr(\tilde{\mathbf{K}}, 0)$
        $\mathbf{U} = \mathbf{U}/\mathbf{R}$
        $\mathbf{r}^{(k,\ell)} = \mathbf{b} - (\mathbf{K} + \gamma_\ell \mathbf{E}\mathbf{U})\left(\mathbf{I} + \gamma_\ell \mathbf{K}^T\mathbf{E}\mathbf{U}\right)^{-1}\mathbf{K}^T\mathbf{b}$
        if $\frac{\|\mathbf{r}^{(k,\ell)}\|}{\|\mathbf{b}\|} > tol$ then
            % MINRES recycling using $\mathbf{U}_\ell$
            $\tilde{\mathbf{K}}_\ell = \left(\mathbf{A}^{(k)} + \gamma_\ell \mathbf{E}\right)\mathbf{U}_\ell$
            $[\mathbf{K}_\ell, \mathbf{R}] = qr(\tilde{\mathbf{K}}_\ell, 0)$
            $\mathbf{U}_\ell = \mathbf{U}_\ell/\mathbf{R}$
            Solve (6.6)
            Find $\mathbf{x}^{(k,\ell)}$ by solving (6.7)
            $\mathbf{r}^{(k,\ell)} = (\mathbf{I} - \mathbf{K}_\ell\mathbf{K}_\ell^T)(\mathbf{r}^{(k,\ell)} - \left(\mathbf{A}^{(k)} + \gamma_\ell \mathbf{E}\right)\mathbf{y}_m)$
            $\mathbf{U} \Leftarrow [\mathbf{U}, \mathbf{y}_m]$
            $\mathbf{U}_\ell \Leftarrow [\mathbf{U}_\ell, \mathbf{y}_m]$
        end
    end
end
```

since we no longer need to find $\mathbf{E}\mathbf{U}$ or $\mathbf{E}\mathbf{y}_m$.

6.1.5 Algorithm Analysis

We now discuss the computational costs associated with Algorithm 3. We include an invariant subspace in $\mathbf{U}$ and $\mathbf{U}_\ell$ formed from eigenvectors corresponding to the smallest eigenvalues from our initial non-shifted system. While Krylov solvers can usually take advantage of the shift invariant property [65], our method of recycling cannot take advantage of this property since the presence of the shift changes the space where $\mathbf{U}_\ell$ gets mapped. Even though the invariant subspace may not deflate the spectrum for the shifted systems, it will deflate part of the spectrum for the non-shifted problem. Additionally, the right-hand side of (6.5) for the non-shifted and shifted problem is made small across spectral components in the $\mathbf{K}$ direction. This is the large $\mathbf{K}$ and not the smaller, shift specific $\mathbf{K}_\ell$.

There is a cost associated with the QR factorizations done to compute both $\mathbf{K}$ and $\mathbf{K}_\ell$. However, the initial work is done up front and saved for many inverse problems if the same $\mathbf{E}$ is used. Once inside the loop over shifts, we are only adding one column at a time to both $\mathbf{U}$ and $\mathbf{U}_j$. As was shown in Chapter 5, we compute $\mathbf{K}$ and $\mathbf{K}_\ell$ such that we do not need to do a full re-orthogonalization every time they are computed. An efficient alternative for computing the initial $\mathbf{K}_\ell$ is given in Section 3.2 of [62], but there is a potential trade off in accuracy.

For problems with many shifts, the overhead cost associated with our method is offset by the decreasing number of iterations. The overhead cost in the initialization is computed off-line for the DOT problem, since $\mathbf{A}^{(0)}$ is the same across many images. Therefore, the eigenvectors are precomputed and reused for many problems. If the same shifts are used across problems as well, the basis for the $\text{Range}([\mathbf{U}_0, \mathbf{X}^{(0)}])$ is also precomputed. The overhead costs for the main loop of the algorithm include, updating $\mathbf{K}$ and finding the initial residual and solution. If we have to do recycling, we have to update $\mathbf{K}_\ell$, solve (6.6), and update the solution and residual. We argue that the small number of MINRES iterations required offsets these costs.

6.1.6 Numerical Experiments

In this section, we look at how we use the inner-outer recycling for shifted systems in the DOT setting for two different shifts. Although this application requires that we solve each system for both multiple shifts and multiple right-hand sides, we can use the approach outlined above for each right-hand side. An extension of our method to handle multiple right-hand sides is considered in Section 6.2. All of the experiments were run using a laptop with a 3.20 GHz processor and 16.0 GB RAM using MATLAB R2015b.

6.1.6.1 Experiment 1

For the DOT problem, we discretize the diffusion equation in the frequency domain. A single function evaluation amounts to solving $\left(\mathbf{A}^{(k)} + \imath\gamma_\ell \mathbf{I}\right)\mathbf{X}^{(k,\ell)} = \mathbf{B}$ for all ℓ at which we have collected data, then computing $\mathbf{C}^T\mathbf{X}^{(k,\ell)}$, where $\mathbf{C}$ represents the detectors and $\mathbf{B}$ the sources. Though this application requires we solve each system for both multiple shifts and multiple right-hand sides, we compute the solution for each across all right-hand sides, independently, using our new approach.

In this numerical experiment we consider a 201×201 mesh, and discretize with finite differences as in [28], which gives us $40,401$ degrees of freedom for the forward problem. We adopt the technique in Chapter 5 to identify $\mathbf{A}^{(k)}$ with a SPD matrix, which now has $39,999$ degrees of freedom. We use one right-hand side from the DOT problem, which is a multiple of a column of the identity matrix. The shift in this application is, $\imath\gamma_\ell = \imath\frac{2\pi 10^6\omega_\ell}{\nu}$, where ω_ℓ is the frequency and ν is the speed of light in the medium.

We solve the problem for 21 frequencies, $0 : 10 : 200$ MHz, and 10 systems. As was stated above, since we are using many shifts we add the real component, as well as the imaginary component, of the initial solution for $\ell = 6$, 11, and 16 to the initial $\mathbf{U}$. This is because as you move farther away from the "zero shift" the solutions can change drastically and only including the real component of the "zero shift" is not sufficient. We include the real component from every fifth shift because we found that to be sufficient experimentally. Table 6.1 shows the number of (unpreconditioned) MINRES iterations for a sample of systems and shifts for this experiment with and without recycling. It also shows the initial relative residuals and the number of columns of $\mathbf{U}$ and $\mathbf{U}_\ell$. We used a tolerance of 10^{-7}. Figure 6.1 shows the number of MINRES iterations for each shift and all systems. It is clear that the iterations generally decrease from one shift to the next, and system to system, using our approach. Obviously at some point $\mathbf{U}$ will get very large if there are many systems and right-hand sides to solve and the overhead associated with this method will become too large. This is the subject of ongoing research. One idea to overcome this issue is binning the shifts into groups and having a **U** associated with each bin.

		Our Approach				MINRES
System	Shift	Initial Relative Residual	Cols **U**	Cols $\mathbf{U}_\ell$	Its	Its
1	1	6.997652e-05	34	11	138	463
	5	1.488217e-07	38	11	11	463
	13	1.249818e-07	46	11	6	460
	21	1.703870e-07	54	11	12	455
3	1	3.110010e-05	76	13	117	491
	5	1.273549e-07	80	13	8	490
	13	1.388263e-07	88	13	7	487
	21	1.185384e-07	96	13	4	480
5	1	1.903012e-05	118	15	92	480
	5	1.228727e-07	122	15	5	479
	13	1.555352e-07	130	15	8	476
	21	1.192817e-07	138	15	4	471
7	1	4.306398e-06	160	17	49	481
	5	1.130465e-07	163	17	3	481
	13	1.089243e-07	171	17	2	478
	21	1.210036e-07	179	17	4	473
9	1	2.146463e-06	200	19	29	481
	5	1.063270e-07	201	19	1	481
	13	1.020278e-07	208	19	1	478
	21	1.236055e-07	216	19	4	473

Table 6.1: Comparison of the inner-outer approach for shifted systems as described in Algorithm 2 vs. MINRES for Experiment 1.

6.1.6.2 Experiment 2

In Experiment 2, we use the same $\mathbf{A}^{(k)}$ matrices and the same right-hand side as in Experiment 1. **E** will be a real valued, positive, diagonal matrix representing a random perturbation of the background of the image of the absorption coefficient. The shifts will be 0, .01, .02, .03, .04. We might encounter this scenario if we were to try to gather statistical or sensitivity information during the optimization process. Table 6.2 shows the number of (unpreconditioned) MINRES iterations for Experiment 2 with and without recycling. We used a tolerance of 10^{-7}. Once again, the iterations generally decrease from one shift to the next, and system to system, using our approach.

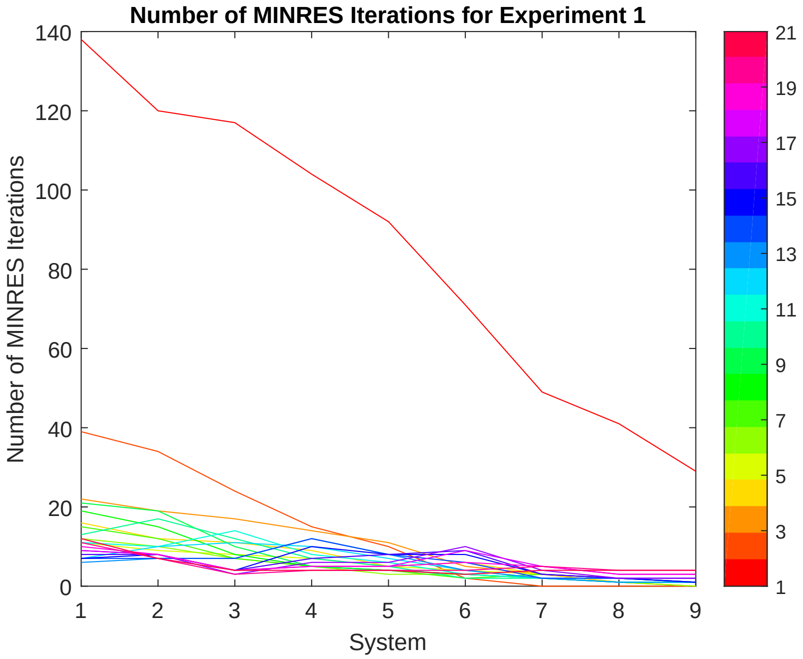


Figure 6.1: The number of MINRES iterations for Experiment 1 for all 21 shifts, which are represented in the colorbar. The iteration counts for the “zero shift" are found in the red curve above the rest. Shifts 2 and 3 are the next two largest curves for the earlier systems, while shift 21 is the next largest by system 9.

6.2 Inner-Outer Recycling for Shifted Systems with Multiple Right-Hand Sides

We now give our extension of inner-outer recycling for shifted systems with multiple right-hand sides. This extension to multiple right-hand sides needs to address how to recover the solution to the shifted problem while keeping the recycle spaces reasonable sizes. For the multiple right-hand side case, we only consider $\mathbf{E} = \mathbf{I}$, therefore we consider solving

$$\left(\mathbf{A}^{(k)} + \imath\gamma_\ell \mathbf{I}\right)\mathbf{X}^{(k,\ell)} = \mathbf{B}, \tag{6.9}$$

with a variant of inner-outer recycling. In addition to using it as an efficient solver, we would like to use this method to construct the basis used in the ROM for DOT. This was done in Chapter 5 for the 0 frequency case and we now extend that method to the multiple frequency case. We first discuss ROM for DOT for the multiple frequency case and then explain the method.

		Our Approach				MINRES
System	Shift	Initial Relative Residual	Cols $\mathbf{U}$	Cols $\mathbf{U}_\ell$	Its	Its
1	1	7.451061e-05	15	11	140	463
	2	8.910013e-07	16	11	45	458
	3	1.816563e-07	17	11	17	452
	4	1.429132e-07	18	11	14	447
	5	1.104739e-07	19	11	3	443
2	1	4.933852e-05	20	12	128	474
	2	6.396233e-07	21	12	42	469
	3	1.951564e-07	22	12	19	464
	4	1.477872e-07	23	12	16	460
	5	1.176433e-07	24	12	8	455
3	1	3.723381e-05	25	13	128	491
	2	4.602222e-07	26	13	36	484
	3	2.086688e-07	27	13	25	477
	4	1.347959e-07	28	13	9	471
	5	1.173205e-07	29	13	6	465
4	1	2.747551e-05	30	14	115	485
	2	3.213505e-07	31	14	22	479
	3	1.722536e-07	32	14	17	473
	4	1.285942e-07	33	14	8	468
	5	1.207813e-07	34	14	8	463

Table 6.2: Comparison of the inner-outer approach for shifted systems as described in Algorithm 2 vs. MINRES for Experiment 2.

6.2.1 Rewriting ROM

Adopting the technique from Section 5.1 to rewrite the DOT transfer function and derivatives in terms of a complex symmetric matrix gives us a shifted problem of the form being considered in this section. As was shown in Subsection 5.1.3, we use the one-sided global basis approach and for $\mathbf{V} \in \mathbb{C}^{n \times r}$ define

$$\mathbf{E}_r = \mathbf{V}^T\mathbf{V}, \qquad \mathbf{A}_r^{(k)}(\mathbf{p}) = \mathbf{V}^T\mathbf{A}^{(k)}\mathbf{V}, \qquad \mathbf{B}_r = \mathbf{V}^T\mathbf{B}, \qquad \mathbf{C}_r = \mathbf{V}^T\mathbf{C}. \tag{6.10}$$

The reduced transfer function is therefore, $\mathbf{\Psi}_r = \mathbf{C}_r^T \left(\frac{\imath\omega}{\nu}\mathbf{E}_r + \mathbf{A}_r^{(k)}\right)^{-1}\mathbf{B}_r$.

We once again use the fact that $\mathbf{A}$ is SPD in order to solve the forward and adjoint problems at the same time to generate $\mathbf{V}$,

$$\left(\mathbf{A}^{(k)} + \frac{\imath\omega_\ell}{\nu}\mathbf{I}\right)\mathbf{X}^{(k,\ell)} = [\mathbf{B}, \mathbf{C}] \tag{6.11}$$

for appropriate choices of parameters $\mathbf{p}_k$, $k = 1, \dots, K$ and frequencies $\omega_\ell, \ell = 1, \dots, n_\omega$.

As in Subsection 5.1.3, we are seeking a surrogate transfer function, $\mathbf{\Psi}_r(\omega; \mathbf{p})$ that approximates $\mathbf{\Psi}(\omega; \mathbf{p})$ as well as ensuring $\nabla_{\mathbf{p}} \mathbf{\Psi}(\omega; \mathbf{p}) \approx \nabla_{\mathbf{p}} \mathbf{\Psi}_r(\omega; \mathbf{p})$. The following theorem, which follows from [16], is Theorem 5.1.1 rewritten for the multiple frequency DOT-PaLs problem shows how to construct $\mathbf{V}$.

Theorem 6.2.1 *Suppose* $\mathbf{A}(\mathbf{p})$ *is continuously differentiable in a neighborhood of* $\hat{\mathbf{p}} \in \mathbb{R}^l$. *Let* $\hat{\omega} \in \mathbb{C}$, *and both* $\frac{\imath\hat{\omega}}{\nu}\mathbf{I} + \mathbf{A}(\hat{\mathbf{p}})$ *and* $\frac{\imath\hat{\omega}}{\nu}\mathbf{E}_r + \mathbf{A}_r(\hat{\mathbf{p}})$ *be invertible.*
If $\left(\frac{\imath\hat{\omega}}{\nu}\mathbf{I} + \mathbf{A}(\hat{\mathbf{p}})\right)^{-1}\mathbf{B} \in$ *Range*$(\mathbf{V})$ *and* $\left(\mathbf{C}\left(\frac{\imath\hat{\omega}}{\nu}\mathbf{I} + \mathbf{A}(\hat{\mathbf{p}})\right)^{-1}\right)^T \in$ *Range*$(\mathbf{V})$ *then,*
the reduced parametric model of (6.10) satisfies
$\mathbf{\Psi}(\hat{\omega}, \hat{\mathbf{p}}) = \mathbf{\Psi}_r(\hat{\omega}, \hat{\mathbf{p}})$, $\nabla_{\mathbf{p}}\mathbf{\Psi}(\hat{\omega}, \hat{\mathbf{p}}) = \nabla_{\mathbf{p}}\mathbf{\Psi}_r(\hat{\omega}, \hat{\mathbf{p}})$, *and* $\mathbf{\Psi}'(\hat{\omega}, \hat{\mathbf{p}}) = \mathbf{\Psi}'_r(\hat{\omega}, \hat{\mathbf{p}})$,
where $'$ *denotes the derivative with respect to* ω.

We would like to use the method presented in this subsection to construct the global basis, $\mathbf{V}$, for DOT.

6.2.2 The Method

As in Subsection 6.1.1, we assume the first value of γ_ℓ we wish to solve for is 0. Once again we assume $\mathbf{U} \in \mathbb{R}^{N \times n_c}$ is generated from the recycle space $\tilde{\mathbf{U}}$ such that $\mathbf{A}^{(k)}\mathbf{U} = \mathbf{K}$ and $\mathbf{K}^T\mathbf{K} = \mathbf{I}$. Select $\tilde{\mathbf{U}}_j \in \mathbb{R}^{N \times n_j}$ from among the columns of $\tilde{\mathbf{U}}$. Generate $\mathbf{U}_j$ from $\tilde{\mathbf{U}}_j$, such that $\mathbf{A}^{(k)}\mathbf{U}_j = \mathbf{K}_j$ and $\mathbf{K}_j^T\mathbf{K}_j = \mathbf{I}$. $\mathbf{U}_j$ is once again a right-hand side specific space as in Section 5.2, but now also contains information about the shifted system as described below.

To keep notation consistent, we let $\mathbf{b}_j = \mathbf{B}(:, j)$ and $\mathbf{x}_j^{(k,\ell)} = \mathbf{X}^{(k,\ell)}(:, j)$. First, we estimate that our solution is in Range$(\mathbf{U})$ for all shifts. When $\gamma_\ell = 0$, the optimal solution in Range$(\mathbf{U})$ is $\mathbf{x}_j^{(k,\ell)} = \mathbf{U}\mathbf{K}^T\mathbf{b}_j$ and the initial residual is $(\mathbf{I} - \mathbf{K}\mathbf{K}^T)\mathbf{b}_j$ as in Section 3.4. When $\gamma_\ell \neq 0$, we follow Subsection 6.1.1 and get $\mathbf{x}_j^{(k,\ell)} = \mathbf{U}(\mathbf{I} + \imath\gamma_\ell\mathbf{K}^T\mathbf{U})^{-1}\mathbf{K}^T\mathbf{b}_j$ as the optimal solution in Range$(\mathbf{U})$ with initial residual $\mathbf{r}_j^{(k,\ell)} = \mathbf{b}_j - (\mathbf{K} + \imath\gamma_\ell\mathbf{U})(\mathbf{I} + \imath\gamma_\ell\mathbf{K}^T\mathbf{U})^{-1}\mathbf{K}^T\mathbf{b}_j$.

If the initial residual $\mathbf{r}_j^{(k,\ell)}$ is not small enough, we once again solve for the incremental change from the initial guess. When $\gamma_\ell = 0$, we follow Section 5.2 to expand the recycle space and find the solution. When $\gamma_\ell \neq 0$ we proceed as follows. Using $\mathbf{x}_j^{(k,\ell)} = \mathbf{g}_j + \mathbf{x}_i$, where $\mathbf{x}_i$ is the initial guess, the problem we want to solve is

$$\min_{\mathbf{g}\in\mathcal{S}} \|\mathbf{r}_j^{(k,\ell)} - \left(\mathbf{A}^{(k)} + \imath\gamma_\ell\mathbf{I}\right)\mathbf{g}_j\|_2,$$

over an appropriate $\mathcal{S}$. We want $\mathcal{S}$ to contain Range($\mathbf{U}_j$) and to generate the space with which to augment $\mathbf{U}_j$, we use the Krylov subspace generated by $\left(\mathbf{I} - \mathbf{K}_j\mathbf{K}_j^T\right)\left(\mathbf{A}^{(k)} + \imath\gamma_\ell\mathbf{I}\right)$ and

$$\mathbf{v}_1 = \mathbf{r}_j^{(k,\ell)}/\|\mathbf{r}_j^{(k,\ell)}\|_2.$$

We note that $\mathbf{K}$ and $\mathbf{K}_j$ are both in Range($\mathbf{A}^{(k)}$). We obtain

$$\begin{aligned}
(\mathbf{I} - \mathbf{K}_j\mathbf{K}_j^T)\left(\mathbf{A}^{(k)} + \imath\gamma_\ell\mathbf{I}\right)\mathbf{V}_m &= \mathbf{V}_{m+1}\underline{\mathrm{T}}_m \Leftrightarrow \\
\left(\mathbf{A}^{(k)} + \imath\gamma_\ell\mathbf{I}\right)\mathbf{V}_m &= \mathbf{K}_j\mathbf{K}_j^T\left(\mathbf{A}^{(k)} + \imath\gamma_\ell\mathbf{I}\right)\mathbf{V}_m + \mathbf{V}_{m+1}\underline{\mathrm{T}}_m \\
&= \mathbf{K}_j\mathbf{K}_j^T\mathbf{A}^{(k)}\mathbf{V}_m + \mathbf{V}_{m+1}\underline{\mathrm{T}}_m.
\end{aligned}$$

Therefore, we want to solve

$$\min_{\mathbf{y},\mathbf{z}}\left\|\mathbf{r}_j^{(k,\ell)} - \left(\mathbf{A}^{(k)} + \imath\gamma_\ell\mathbf{I}\right)[\mathbf{V}_m, \mathbf{U}_j]\begin{bmatrix}\mathbf{y}\\ \mathbf{z}\end{bmatrix}\right\|.$$

Following [43], we observe that,

$$\begin{aligned}
(\mathbf{A}^{(k)} + \imath\gamma_\ell\mathbf{I})[\mathbf{V}_m\ \mathbf{U}_j] &= \begin{bmatrix}\mathbf{V}_{m+1}\underline{\mathrm{T}}_m + \mathbf{K}_j\mathbf{K}_j^T\mathbf{A}^{(k)}\mathbf{V}_m & \mathbf{K}_j + \imath\gamma_\ell\mathbf{U}_j\end{bmatrix} \\
&= \begin{bmatrix}\mathbf{V}_{m+1} & \mathbf{K}_j & \mathbf{U}_j\end{bmatrix}\begin{bmatrix}\underline{\mathrm{T}}_m & 0 \\ \mathbf{K}_j^T\mathbf{A}^{(k)}\mathbf{V}_m & \mathbf{I} \\ 0 & \imath\gamma_\ell\mathbf{I}\end{bmatrix}
\end{aligned}$$

$$= \begin{bmatrix} \mathbf{V}_{m+1} & \mathbf{K}_j & \hat{\mathbf{U}}_j \end{bmatrix} \begin{bmatrix} \mathbf{I} & 0 & \mathbf{V}_{m+1}^T \mathbf{U}_j \\ 0 & \mathbf{I} & \mathbf{K}_j^T \mathbf{U}_j \\ 0 & 0 & \mathbf{N} \end{bmatrix} \begin{bmatrix} \underline{\mathrm{T}}_m & 0 \\ \mathbf{K}_j^T \mathbf{A}^{(k)} \mathbf{V}_m & \mathbf{I} \\ 0 & \imath\gamma_\ell \mathbf{I} \end{bmatrix}, \tag{6.12}$$

where the last line orthogonalizes $\mathbf{U}_j$ against $\begin{bmatrix} \mathbf{V}_{m+1} & \mathbf{K}_j \end{bmatrix}$, so that $\begin{bmatrix} \mathbf{V}_{m+1} & \mathbf{K}_j & \hat{\mathbf{U}}_j \end{bmatrix}$ has orthonormal columns. This is done with a reduced QR decomposition, therefore $\mathbf{N}$ is a diagonal scaling matrix involving the norms of the columns of $\mathbf{U}_j$. Therefore, we write the least squares problem as

$$\min_{\mathbf{y},\mathbf{z}} \left\| \begin{bmatrix} \xi \mathbf{e}_1 \\ 0 \\ 0 \end{bmatrix} - \begin{bmatrix} \underline{\mathrm{T}}_m & \imath\gamma_\ell \mathbf{V}_{m+1}^T \mathbf{U}_j \\ \mathbf{K}_j^T \mathbf{A}^{(k)} \mathbf{V}_m & \mathbf{I} + \imath\gamma_\ell \mathbf{K}_j^T \mathbf{U}_j \\ 0 & \imath\gamma_\ell \mathbf{N} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{z} \end{bmatrix} \right\|_2, \tag{6.13}$$

where $\mathbf{e}_1$ denotes the first Cartesian basis vector in $\mathbb{R}^{m+1}$ and $\xi = \|\mathbf{r}_j^{(k,\ell)}\|_2$. We have $\mathbf{g}_j = \underbrace{\mathbf{V}_m \mathbf{y}}_{\mathbf{y}_m} + \mathbf{U}_j \mathbf{z}$ and we reconstruct the solution estimate by setting,

$$\mathbf{x}_j^{(k,\ell)} = \mathbf{g}_j + \mathbf{x}_i. \tag{6.14}$$

In addition to solving the full order systems, we also use this method to construct the basis used in the ROM for DOT. Several steps of the optimization problem would be solved with this method and we would let the basis, $\mathbf{V}$, be defined as the large recycle space, $\mathbf{U}$. This is because $\mathbf{U}$ contains information from all shifts and all right-hand sides.

6.2.3 Identifying and Updating Recycle Spaces

We now discuss how to construct the recycle spaces. As stated above, $\mathbf{U}$ contains information from all right-hand sides and all shifts and, if desired, ultimately becomes the global basis for ROM, while $\mathbf{U}_j$ is right-hand side specific but still contains information about all shifts. As before, we include an invariant subspace consisting of 10 eigenvectors corresponding to the smallest eigenvalues of $\mathbf{A}^{(0)}$ to both $\mathbf{U}$ and $\mathbf{U}_j$. In addition, $\mathbf{U}$ is seeded with solutions to the initial system for all right-hand sides and all shifts, while $\mathbf{U}_j$ is seeded with only the solution to the j^{th} right-hand side for all shifts. Therefore, $\mathbf{U}$ has $(10 + n_j * n_\ell)$ columns and $\mathbf{U}_j$ has $(10 + n_\ell)$ columns, where n_j is the number of right-hand sides and n_ℓ is the number of shifts. Since we are dealing with a complex shift, we only use the imaginary component of the solutions for the shifted systems as described in Subsection 6.1.4. If there are not many shifts, this ensures that $\mathbf{U}_j$ is kept small.

If the initial residual is such that we need to perform the inner recycling, we append information from $\mathbf{y}_m$ to both $\mathbf{U}$ and $\mathbf{U}_j$. Again, we only add the imaginary component of $\mathbf{y}_m$ for the shifted systems. This ensures that $\mathbf{U}$ contains information pertinent to the entire set of shifted systems, while $\mathbf{U}_j$ is kept small. We only update the recycle spaces after we have found the solutions to all shifts for a given right-hand side, which means that we add a maximum of the number of shifts columns to our recycle spaces.

6.2.4 The Algorithm

Algorithm 4 describes our inner-outer recycling process for shifted systems with multiple right-hand sides.

6.2.5 Algorithm Analysis

We now discuss the computational costs associated with Algorithm 4. Once again, including an invariant subspace in $\mathbf{U}$ and $\mathbf{U}_j$ formed from eigenvectors corresponding to the smallest eigenvalues from our initial non-shifted system deflates the spectrum. Additionally, the right-hand side is made small across spectral components in the $\mathbf{K}$ direction. This is the large $\mathbf{K}$ and not smaller, right-hand side specific $\mathbf{K}_j$.

We use the techniques mentioned in Subsection 5.2.5.3 and Subsection 6.1.5 to reduce the cost of the initial QR factorization and the updates. Also, since we only recalculate $\mathbf{K}$ and $\mathbf{K}_j$ after we have computed all the shifts, we store $\mathbf{K}_j^T\mathbf{A}^{(k)}$ and $\mathbf{K}_j^T\mathbf{U}_j$ for use in (6.13) for all shifts.

Algorithm 4: Recycling and Global Basis Construction

```
$\mathbf{U}_0 \Leftarrow 10$ eigenvectors of $\mathbf{A}^{(0)}$
$\mathbf{X}_0 \Leftarrow$ solutions to $\left(\mathbf{A}^{(0)} + \gamma_\ell \mathbf{I}\right) \mathbf{X}^{(0,\ell)} = \mathbf{B}$ for all $\gamma_\ell$
$\mathbf{U} \Leftarrow$ basis for Range$([\mathbf{U}_0, \mathbf{X}_0])$
$\mathbf{U}_j \Leftarrow [\mathbf{U}_0, \mathbf{X}_0(:, j)]$
for $k = 1 : nsys$ do
    for $j = 1 : nrhs$ do
        % Form $\mathbf{K}$ and $\mathbf{K}_j$ to use for all shifts $\tilde{\mathbf{K}} = \mathbf{A}^{(k)} \mathbf{U}$
        $[\mathbf{K}, \mathbf{R}] = qr(\tilde{\mathbf{K}}, 0)$
        $\mathbf{U} = \mathbf{U}/\mathbf{R}$
        $\tilde{\mathbf{K}}_j = \mathbf{A}^{(k)} \mathbf{U}_j$
        $[\mathbf{K}_j, \mathbf{R}] = qr(\tilde{\mathbf{K}}_j, 0)$
        $\mathbf{U}_j = \mathbf{U}_j/\mathbf{R}$
        for $\ell = 1 : nshifts$ do
            % Check if $\mathbf{U}$ is a good enough space
            $\mathbf{r}^{(k,\ell)} = \mathbf{b} - (\mathbf{K} + \gamma_\ell \mathbf{U}) \left(\mathbf{I} + \gamma_\ell \mathbf{K}^T \mathbf{U}\right)^{-1} \mathbf{K}^T \mathbf{b}$
            if $\frac{\|\mathbf{r}^{(k,\ell)}\|}{\|\mathbf{b}\|} > tol$ then
                % MINRES recycling using $\mathbf{U}_j$
                Solve (6.13)
                Find $\mathbf{x}_j^{(k,\ell)}$ by solving (6.14)
                $\mathbf{U} \Leftarrow [\mathbf{U}, \mathbf{y}_m]$
                $\mathbf{U}_j \Leftarrow [\mathbf{U}_j, \mathbf{y}_m]$
            end
        end
    end
end
```

Since we reuse $\mathbf{A}^{(0)}$ for many reconstructions, the initial recycle spaces are computed off-line for the DOT problem. The overhead costs for the main loop of the algorithm include updating $\mathbf{K}$ and finding the initial residual and solution. If we have to do recycling, we must update $\mathbf{K}_j$ and solve (6.13) for all shifts. We argue that the small number of MINRES iterations required offsets these costs.

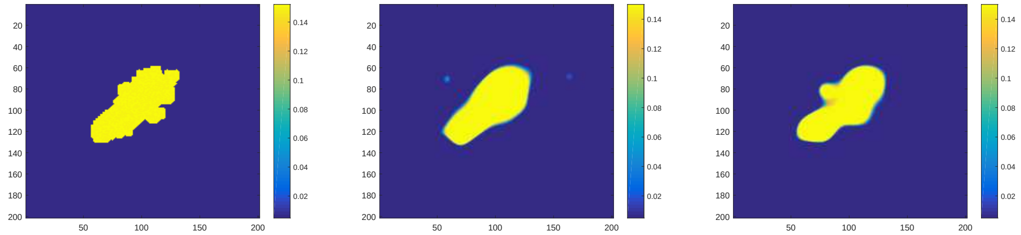
6.2.6 Numerical Experiments

In this section, we provide two 2-dimensional numerical experiments in the DOT setting. The goal in this setting is to solve the sequence of shifted systems efficiently while constructing the global basis for reduced order modeling. We show two experiments where the global basis is reused, in order to show that with this new method, we can still reuse the global basis. The set-up for the experiments is the same as in Section 5.3, except now we are using two frequencies. In this application we are to solving systems of the form $\left(\mathbf{A}^{(k)} + \imath\gamma_\ell \mathbf{I}\right)\mathbf{X}^{(k,\ell)} = \mathbf{B}$, where $\gamma_\ell = \frac{2\pi 10^6 \omega_\ell}{\nu}$, ω_ℓ is the frequency and ν is the speed of light in the medium. For both experiments we use two frequencies, 0 and 10 MHz. Results are reported for both using the full order model and the ROM to compute the function and Jacobian evaluations for each experiment. The tolerance in line 16 of Algorithm 4 was set to be 10^{-7}. All of the experiments were run using a laptop with a 3.20 GHz processor and 16.0 GB RAM using MATLAB R2015b.

The experiments were solved on a 201×201 mesh, which gives us 40,401 degrees of freedom for the forward problem. We use 32 sources and 32 detectors. Following the PaLs approach, we use 25 compactly supported radial basis functions, which results in 100 parameters for the optimization problem. Figure 4.8 shows the absorption image using the initial set of parameters. The projection basis that are used for Experiments 1 and 2 were created with the first 3 iterations of the optimization using the full order model for Experiment 1.

In Experiment 1, we needed to solve 373 large, single right-hand side systems to generate what we needed to construct the global basis matrix (note that the 128 of these corresponding to $\mathbf{X}^{(0,\ell)}$ could have been pre-computed off-line). Including the additional 10 eigenvectors of $\mathbf{A}^{(0)}$ that were used as the first 10 columns of $\mathbf{V}$, $\mathbf{V}$ has 383 columns and thus the reduced model has order 383.

The optimization using the full order model for Experiment 1 required 82 function evaluations and 49 Jacobian evaluations. In comparison, the optimization run using the reduced order model, once it has been generated, required 36 function evaluations and 18 Jacobian evaluations, indicating that using a ROM in place of FOM does not impact the convergence rate of the optimization. Figure 6.2 shows the reconstructions for Experiment 1. The bottom line is that solving the optimization using the full order model requires the solution of 8,384 systems of size $40,401 \times 40,401$. On the other hand, solving using our approach requires solution of 373 systems of size $40,401 \times 40,401$, which are used to construct $\mathbf{V}$ during the



(a) Ground truth absorption image.

(b) Reconstruction using the full order model.

(c) Reconstruction using the reduced order model.

Figure 6.2: Results for Experiment 1. Reconstruction on a 201×201 mesh, resulting in $40,401$ degrees of freedom in the forward model and 383 degrees of freedom in the reduced model for the forward model. 32 sources, 32 detectors, and 25 basis functions were used.

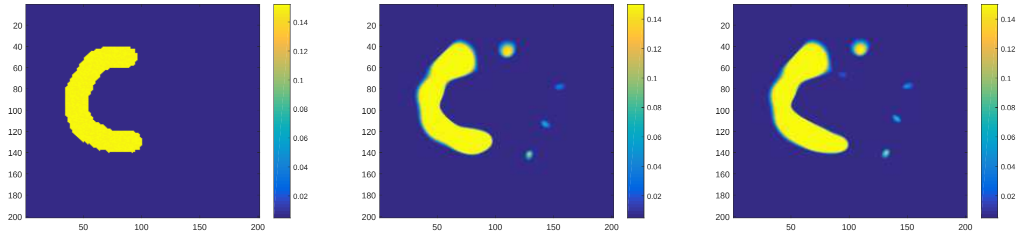
first few optimization steps. The remainder of the work is in solving systems of size 383×383 until the convergence tolerance for the optimization is achieved.

Table 6.3 also includes the number of (unpreconditioned) MINRES iterations with and without recycling. Although the table only shows a sample of results, it is clear that the iterations decrease from one right-hand side to the next, and system to system, using our approach. The jump in number of iterations for right-hand-side 33 comes from the fact that we concatenated $\tilde{\mathbf{B}}$ and $\tilde{\mathbf{C}}$ to form one right-hand-side for the symmetric transfer function, so the 33^{rd} right-hand side corresponds to the first column in $\tilde{\mathbf{C}}$.

In Experiment 2, we used a different problem than in Experiment 1, but with the ROM formed from the basis made in Experiment 1. Therefore, for this experiment no additional full order model solves were required. The optimization using the full order model required 29 function evaluations and 16 Jacobian evaluations. The optimization run using our reduced order model took 32 function evaluations and 22 Jacobian evaluations to converge to our stopping criterion. Figure 6.3 shows the reconstructions for Experiment 2.

			Our Approach				MINRES
System	Shift	RHS	Initial Rel. Res.	Cols $\mathbf{U}$	Cols $\mathbf{U}_j$	Its	Its
2	1	1	7.394391e-05	138	12	140	463
	2	1	7.393480e-05	138	12	140	463
	1	20	7.717954e-07	176	12	45	541
	2	20	7.770250e-07	176	12	46	541
	1	32	8.564147e-08	200	12	0	487
	2	32	8.993545e-08	200	12	0	487
	1	33	2.618185e-05	200	12	105	467
	2	33	2.617891e-05	200	12	105	467
	1	53	1.014723e-06	240	12	38	528
	2	53	1.017215e-06	240	12	38	528
	1	64	9.980185e-08	260	12	0	489
	2	64	1.002432e-07	260	12	1	488
3	1	1	4.164954e-05	261	14	112	474
	2	1	4.164502e-05	261	14	112	474
	1	20	4.591432e-07	299	14	22	513
	2	20	4.666139e-07	299	14	22	513
	1	32	7.825542e-08	321	12	0	497
	2	32	8.248123e-08	321	12	0	497
	1	33	1.163491e-05	321	14	81	474
	2	33	1.163392e-05	321	14	81	474
	1	53	4.317199e-07	361	14	18	526
	2	53	4.322702e-07	361	14	18	526
	1	64	8.008862e-08	383	13	0	497
	2	64	8.136813e-08	383	13	0	497

Table 6.3: Comparison of the inner-outer approach for shifted systems as described in Algorithm 4 with $\gamma_\ell = 0, \frac{2\pi 10^6 \omega_\ell}{\nu}$, where $\omega_\ell = 10$ MHz is the frequency and ν is the speed of light in the medium vs. MINRES.



(a) Ground truth absorption image.

(b) Reconstruction using the full order model.

(c) Reconstruction using the reduced order model.

Figure 6.3: Results for Experiment 2. Reconstruction on a 201 × 201 mesh, resulting in 40,401 degrees of freedom in the forward model and 383 degrees of freedom in the reduced order model. 32 sources, 32 detectors, and 25 basis functions were used.

Chapter 7

Recycle Spaces for Inner-Outer Recycling

In this chapter we explore the recycle spaces used in inner-outer recycling for shifted systems in more detail. Once again, we are looking to solve shifted systems of the form

$$\left(\mathbf{A}^{(k)} + \gamma_\ell \mathbf{E}\right) \mathbf{x}^{(k,\ell)} = \mathbf{b}, \tag{7.1}$$

for symmetric $\mathbf{A}^{(k)}$ and $\mathbf{E}$ and several values of k and ℓ via recycling. In the previous chapter, we used eigenvectors that correspond to the smallest eigenvalues of the initial system, $\mathbf{A}^{(0)}$, as an invariant subspace for the recycle space. As the shift increases and the shifted system gets further away from $\mathbf{A}^{(0)}$ we should consider if this invariant subspace is still relevant. In this chapter we explore the idea of using generalized eigenvectors of $\mathbf{A}^{(0)}$ and $\mathbf{E}$ as an invariant subspace for the recycle space used in inner-outer recycling.

7.1 Generalized Eigenvalue Problem

The generalized eigenvalue problem is the problem of finding a vector $\mathbf{w} \neq 0$ that satisfies,

$$\mathbf{A}\mathbf{w} = \lambda \mathbf{E}\mathbf{w}, \tag{7.2}$$

where $\mathbf{A}$ and $\mathbf{E}$ are matrices. The vector $\mathbf{w}$ is called the generalized eigenvector of $\mathbf{A}$ and $\mathbf{E}$ and λ is called the generalized eigenvalue of $\mathbf{A}$ and $\mathbf{E}$. The generalized eigenvalue problem can be written as a standard eigenvalue problem if $\mathbf{E}$ is invertible,

$$\mathbf{E}^{-1}\mathbf{A}\mathbf{w} = \lambda \mathbf{w}. \tag{7.3}$$

If $\mathbf{A}$ and $\mathbf{E}$ are both Hermitian matrices, in most cases performing this inversion will destroy that structure, that is, $\mathbf{E}^{-1}\mathbf{A}$ is not necessarily Hermitian.

From (7.2) we observe the following property,

$$\begin{aligned}(\mathbf{A}+\gamma\mathbf{E})\mathbf{w} &= \mathbf{A}\mathbf{w}+\gamma\mathbf{E}\mathbf{W}\\ &= \lambda\mathbf{E}\mathbf{w}+\gamma\mathbf{E}\mathbf{w}\\ &= (\lambda+\gamma)\mathbf{E}\mathbf{w}.\end{aligned}$$

Therefore, the generalized eigenvectors have a "shift invariance" property. This property is our motivation to use them as an invariant subspace in the recycle spaces.

Additionally, perturbation to **A** would give the following generalized eigenvalue problem,

$$(\mathbf{A}+\mathbf{F})\hat{\mathbf{w}} = \hat{\lambda}\mathbf{E}\hat{\mathbf{w}}. \tag{7.4}$$

Again, if **E** is invertible we rewrite this as follows,

$$\mathbf{E}^{-1}(\mathbf{A}+\mathbf{F})\hat{\mathbf{w}} = \hat{\lambda}\hat{\mathbf{w}}. \tag{7.5}$$

We are interested in the relationship between **w** and $\hat{\mathbf{w}}$.

7.2 Invariant Subspaces

In this section we explore the generalized eigenspaces of $\mathbf{A}^{(k)}$ and **E** given a small perturbation to **A**. For the DOT problem, the eigenvalues are not nicely clustered. Theorem 4.1 in [43], says that for an SPD matrix, **A**, under certain conditions for perturbation matrix, **F**, the invariant subspace of **A** that is associated with the smallest eigenvalues stays nearly an invariant subspace for the perturbed problem, even if those smallest eigenvalues are not well separated from the next largest ones.

We rewrite our generalized eigenvalue problem using a similarity transform as follows,

$$\mathbf{E}^{-\frac{1}{2}}\mathbf{A}\mathbf{E}^{-\frac{1}{2}}\underbrace{\mathbf{E}^{\frac{1}{2}}\mathbf{w}}_{\mathbf{q}} = \lambda\underbrace{\mathbf{E}^{\frac{1}{2}}\mathbf{w}}_{\mathbf{q}}. \tag{7.6}$$

We note that this symmetrized eigenvalue problem has the same eigenvalues as the original eigenvalue problem, $\mathbf{E}^{-1}\mathbf{A}\mathbf{w} = \lambda\mathbf{w}$. Therefore, we apply Theorem 4.1 from [43] to get a bound on the eigenvalues. Let us establish some notation first. Let $\mathbf{E}^{-\frac{1}{2}}\mathbf{A}\mathbf{E}^{-\frac{1}{2}}$ have the following eigendecomposition,

$$\mathbf{E}^{-\frac{1}{2}}\mathbf{A}\mathbf{E}^{-\frac{1}{2}} = [\mathbf{Q}_1\ \mathbf{Q}_2\ \mathbf{Q}_3]diag\left(\Lambda_1, \Lambda_2, \Lambda_3\right)[\mathbf{Q}_1\ \mathbf{Q}_2\ \mathbf{Q}_3]^T, \tag{7.7}$$

where $\mathbf{Q} = [\mathbf{Q}_1\ \mathbf{Q}_2\ \mathbf{Q}_3]$ is an orthogonal matrix and $\Lambda_i = diag(\lambda_1^{(i)}, \ldots, \lambda_{ki}^{(i)})$. The eigenvalues have also been ordered from smallest to largest. We consider the perturbation matrix $\mathbf{E}^{-\frac{1}{2}}\mathbf{F}\mathbf{E}^{-\frac{1}{2}}$, where the projection of $\mathbf{E}^{-\frac{1}{2}}\mathbf{F}\mathbf{E}^{-\frac{1}{2}}$ into the Range($[\mathbf{Q}_1\mathbf{Q}_2]$) is small, therefore $||\mathbf{E}^{-\frac{1}{2}}\mathbf{F}\mathbf{E}^{-\frac{1}{2}}\mathbf{Q}_1\ \mathbf{Q}_2]||_F \leq \epsilon$. It does not need to be assumed that $sep(\Lambda_1, \Lambda_2)$ is large. It is also assumed that

$$\delta \equiv min(\lambda_1^{(2)} - \epsilon, \lambda_1^{(3)} - \eta) - 2\epsilon - (\lambda_{k1}^{(1)} + \epsilon) \gg \epsilon, \tag{7.8}$$

$$\hat{\delta} = \delta\left(1 - \frac{2\epsilon^2}{\delta^2}\right) + \lambda_{k1}^{(1)} + \epsilon. \tag{7.9}$$

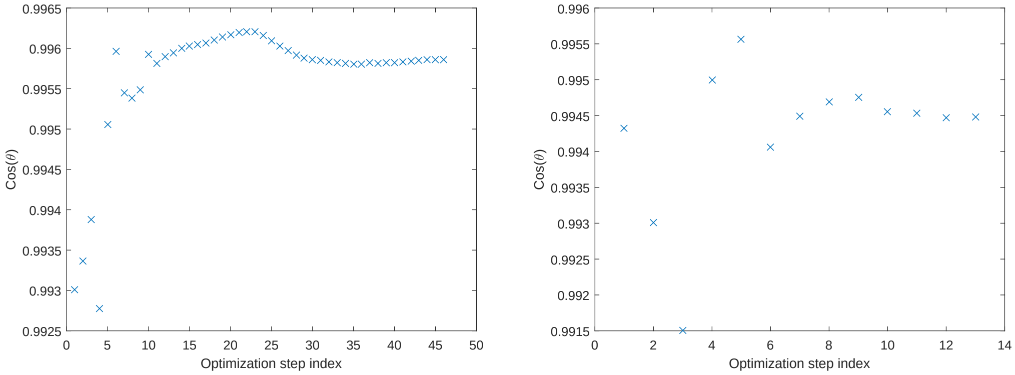
Applying the theorem means that for each eigenvalue $\hat{\lambda}_j^{(1)}$ of $\mathbf{E}^{-\frac{1}{2}}(\mathbf{A}+\mathbf{F})\mathbf{E}^{-\frac{1}{2}}$ corresponding to the invariant subspace Range($\hat{\mathbf{Q}}_1$), there exists an eigenvalue $\lambda_i^{(1)}$ such that $|\hat{\lambda}_j^{(1)} - \lambda_i^{(1)}| \leq \epsilon + \frac{2\epsilon^2}{\delta}$. Since the eigenvalues for this symmetric problem are the same for our original problem, this bound on the eigenvalues holds in that case as well. The theorem also provides the following bound,

$$\tan\theta_1\left(\text{Range}(\mathbf{Q}_1), \text{Range}(\hat{\mathbf{Q}}_1)\right) \leq \frac{\epsilon}{\hat{\delta}}. \tag{7.10}$$

Since $\mathbf{W} = \mathbf{E}^{-\frac{1}{2}}\mathbf{Q}$ it is not clear that we can use this bound to develop a bound for the relationship between $\mathbf{W}_1$ and $\hat{\mathbf{W}}_1$.

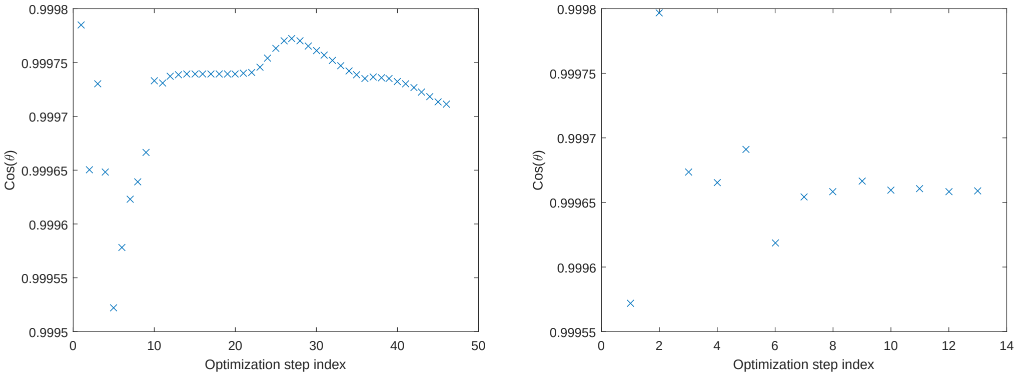
We look at the one column of $\mathbf{Q}$, $\mathbf{E}^{-\frac{1}{2}}\mathbf{q}_1 = \mathbf{E}^{-\frac{1}{2}}\hat{\mathbf{q}}_1 + \mathbf{E}^{-\frac{1}{2}}\zeta$, where ζ represents the difference between the invariant subspaces. If $\mathbf{E}^{-\frac{1}{2}}\zeta$ does not magnify the difference, then we would assume that $\mathbf{w}_1$ is close to $\hat{\mathbf{w}}_1$.

Next, we look at the invariant subspaces numerically. For both examples, we looked at the generalized eigenvector corresponding to the 8 smallest generalized



(a) (b)

Figure 7.1: Evolution of the subspace gap (cosine of the largest canonical angle θ) between the initial generalized eigenvectors and subsequent generalized eigenvectors over the course of the optimization.



(a) (b)

Figure 7.2: Evolution of the subspace gap (cosine of the smallest canonical angle θ) between the initial generalized eigenvectors and subsequent generalized eigenvectors over the course of the optimization.

eigenvalues. Figures 7.1 and 7.2 show how close the generalized eigenvectors of $\mathbf{A}^{(k)}$ and $\mathbf{E}$ remain to the generalized eigenvectors of $\mathbf{A}^{(0)}$ and $\mathbf{E}$ for two numerical examples. We give the cosine of the largest canonical angle at each optimization step in Figure 7.1 and the cosine of the smallest canonical angle at each optimization step in Figure 7.2. In both test problems, the cosine of the largest and smallest canonical angles remain close to 1, meaning θ is quite small and entire generalized eigenspace corresponding to the smallest 8 generalized eigenvalues remains close.

7.3 Numerical Experiments

We would like the invariant subspace associated with the generalized eigenvectors to improve the convergence of MINRES when used as a recycle space in Algorithm 3. The interlacing theorem [68] says that the nonzero eigenvalues of $(\mathbf{I}-\mathbf{K}\mathbf{K}^T)\left(\mathbf{A}^{(k)}+\gamma_\ell\mathbf{E}\right)(\mathbf{I}-\mathbf{K}\mathbf{K}^T)$ are bounded by those of $\left(\mathbf{A}^{(k)}+\gamma_\ell\mathbf{E}\right)$. Therefore, we know that we will not make the convergence worse, but ultimately we would like to make the convergence better.

Two numerical experiments show that this is indeed the case. Table 7.1 compares three methods for solving $\left(\mathbf{A}^{(k)}+\gamma_\ell\mathbf{E}\right)\mathbf{x}^{(k,\ell)}=\mathbf{b}$, MINRES and Algorithm 3 with two different initial recycle spaces. We initialize $\mathbf{U}$ and $\mathbf{U}_\ell$ with 8 generalized eigenvectors corresponding to the smallest eigenvalues and 8 eigenvectors corresponding to the smallest eigenvalues of $\mathbf{A}^{(0)}$. The recycle spaces are updated as usual. $\mathbf{E}$ is a random diagonal matrix with positive entries, $\mathbf{A}^{(k)}$ is from the DOT problem, and $\gamma_\ell = 0:0.01:0.04$. The reader will notice that for both method of forming the recycle spaces, there is a decrease in number of MINRES iterations required for inner-outer recycling versus MINRES. However, the number of MINRES iterations is similar across both recycle spaces. This is not unexpected since this is a small shift.

Next, we increase the shift. Table 7.2 compares three methods for solving $\left(\mathbf{A}^{(k)}+\gamma_\ell\mathbf{E}\right)\mathbf{x}^{(k,\ell)}=\mathbf{b}$, MINRES and Algorithm 3 with two different initial recycle spaces. We initialize $\mathbf{U}$ and $\mathbf{U}_\ell$ with 8 generalized eigenvectors corresponding to the smallest eigenvalues and 8 eigenvectors corresponding to the smallest eigenvalues of $\mathbf{A}^{(0)}$. The recycle spaces are updated as usual. $\mathbf{E}$ is a random diagonal matrix, $\mathbf{A}^{(k)}$ is from the DOT problem, and $\gamma_\ell = 0:0.02:0.08$. For this slight increase in shift, the 8 eigenvectors corresponding to the smallest eigenvalues of $\mathbf{A}^{(0)}$ are no longer a sufficient recycle space for this problem. In fact, the maximum number of MINRES iterations is achieved for the first shift of the initial system. Based on the results we have seen so far, we argue that for large shift, the generalized eigenvectors are beneficial.

		Our Approach				MINRES
System	Shift	Cols $\mathbf{U}$	Cols $\mathbf{U}_\ell$	Gen. Evec. Its	Evec. Its	Its
0	1	8	8	274	274	467
	2	9	8	138	152	461
	3	10	8	29	27	455
	4	11	8	33	32	450
	5	12	8	29	27	444
1	1	13	9	155	153	467
	2	14	9	53	54	461
	3	15	9	24	25	455
	4	16	9	23	25	450
	5	17	9	16	12	444
2	1	18	10	140	142	467
	2	19	10	52	53	461
	3	20	10	25	22	455
	4	21	10	16	18	450
	5	22	10	11	6	444
3	1	23	11	135	135	467
	2	24	11	45	45	461
	3	25	11	17	17	455
	4	26	11	14	15	450
	5	27	11	7	9	444
4	1	28	12	117	117	467
	2	29	12	31	31	461
	3	30	12	17	18	455
	4	31	12	15	14	450
	5	32	12	5	8	444

Table 7.1: Comparison of the inner-outer approach for shifted systems as described in Algorithm 2 with $\gamma_\ell = 0 : 0.01 : 0.04$ and only generalized eigenvectors of $\mathbf{A}^{(0)}$ and $\mathbf{E}$ as initial recycle space vs. inner-outer approach for shifted systems with only eigenvectors of $\mathbf{A}^{(0)}$ as initial recycle space vs. MINRES.

		Our Approach				MINRES
System	Shift	Cols **U**	Cols $\mathbf{U}_\ell$	Gen. Evec. Its	Evec. Its	Its
0	1	8	8	274	274	467
	2	9	8	152	1000	455
	3	10	8	49	60	444
	4	11	8	41	666	434
	5	12	8	30	23	425
1	1	13	9	153	153	463
	2	14	9	76	88	451
	3	15	9	17	16	441
	4	16	9	21	22	431
	5	17	9	17	13	423
2	1	18	10	142	143	474
	2	19	10	71	72	463
	3	20	10	18	18	453
	4	21	10	14	16	443
	5	22	10	15	11	434
3	1	23	11	132	133	491
	2	24	11	59	59	475
	3	25	11	17	21	462
	4	26	11	12	14	450
	5	27	11	9	12	440
4	1	28	12	120	120	485
	2	29	12	43	49	472
	3	30	12	16	19	459
	4	31	12	15	13	448
	5	32	12	6	11	439

Table 7.2: Comparison of the inner-outer approach for shifted systems as described in Algorithm 2 with $\gamma_\ell = 0 : 0.02 : 0.08$ and only generalized eigenvectors of $\mathbf{A}^{(0)}$ and $\mathbf{E}$ as initial recycle space vs. inner-outer approach for shifted systems with only eigenvectors of $\mathbf{A}^{(0)}$ as initial recycle space vs. MINRES.

Chapter 8

Conclusions

In this thesis we investigated methods of reducing the computational cost of forward model solves that arise in nonlinear parametric inverse problems. Specifically, we looked at reducing this computational cost in the context of diffuse optical tomography. First, we showed how we reduced the computational cost of the forward model by using interpolatory parametric model reduction. This means that we create a surrogate function to use instead of the full order model. The surrogate model is constructed using a global basis. The global basis was originally constructed by solving several full order model solves and then using a rank-revealing factorization. While this proved an effective method for reducing the computational cost while producing negligible degradation in the quality of the images, the global basis was still expensive to construct.

Next, we developed a method of inner-outer recycling to not only solve the required full order model solves efficiently, but also to build the global basis for model order reduction. We established that the transfer function and derivatives could be written in terms of a SPD matrix for the zero frequency case. This new structure was then used to develop an inner-outer Krylov recycling method that was able to build the global basis on the fly, updating with only new information.

In the DOT problem, more than one frequency is typically used, therefore we extended the inner-outer Krylov recycling to the shifted system case. We showed an approach for solving sequences of shifted systems with a single right-hand side. Also, we developed an approach for solving sequences of shifted systems with multiple right-hand sides while building the global basis for ROM.

Finally, we explored different ways of constructing the recycle spaces for sequences of shifted systems with a single right-hand side. If the shifts are large, then an invariant subspace of the initial system may no longer be helpful. Therefore, we looked at using generalized eigenvectors as an invariant subspace.

Many numerical experiments for DOT were provided to illustrate the success of all of the above methods. However, we believe our methods have the potential to be useful in other similar applications as well, such as EIT or ERT.

Chapter 9

Future Work

There are many new directions of this research we would like to investigate further. First, the performance of our recycling methods that construct the global basis depends on the residual tolerance chosen and the number of full order model systems solved. If we drop the tolerance slightly, we have found that the number of system solves, and therefore the reduced model order, was even further reduced, without too much degradation in the reconstruction. Similarly, solving more full order model systems gave slightly larger reduced order models, but with no improvement in the quality of the reconstruction. We would like to investigate the trade-offs in performance due to these selections. We have also found that solving the systems corresponding to different right-hand sides in a different ordering may also reduce model order and we would like to investigate this phenomenon further.

In addition, we are currently investigating different ways to construct our recycle spaces. We continue to investigate the generalized eigenvectors and their impact on convergence. For the complex identity shift, we are looking at better selection strategies for adding the real component from the initial solution to $\mathbf{U}$. Since our recycle spaces could get very large for multiple right-hand sides with multiple shifts, we are looking at ways to identify when we might need to purge or refresh the information in our recycle spaces. An alternative to purging information that we are considering is binning information for shifts that are close.

Bibliography

[1] A. M. Abdel-Rehim, R. Morgan, and W. Wilcox, *Deflated bicgstab for linear equations in QCD problems*, Proceedings of Science LAT2007, (2007), pp. 026/1–026/7.

[2] R. Acar and C. R. Vogel, *Analysis of bounded variation penalty methods for ill-posed problems*, Inverse Problems, 10 (1994), pp. 1217–1229.

[3] A. Aghasi, M. Kilmer, and E. Miller, *Parametric level set methods for inverse problems*, SIAM J. Imaging Sci., 4 (2011), pp. 618–650.

[4] K. Ahuja, P. Benner, E. de Sturler, and L. Feng, *Recycling bicgstab with an application to parametric model order reduction*, SIAM J. Sci. Comput., 37 (2015), pp. S429–S446.

[5] K. Ahuja, E. de Sturler, S. Gugercin, and E. Chang, *Recycling BiCG with an application to model reduction*, SIAM J. Imaging Sci., 34 (2012), pp. A1925–A1949.

[6] D. Amsallem and C. Farhat, *Interpolation method for the adaptation of reduced-order models to parameter changes and its application to aeroelasticity*, AIAA Journal, 46 (2008), pp. 1803–1813.

[7] ——, *An online method for interpolating linear parametric reduced-order models*, SIAM J. Sci. Comput., 33 (2011), pp. 2169–2198.

[8] H. Antil, M. Heinkenschloss, and R. H. W. Hoppe, *Domain decomposition and balanced truncation model reduction for shape optimization of the Stokes system*, Optimization Methods and Software, 26 (2011), pp. 643–669.

[9] H. Antil, M. Heinkenschloss, R. H. W. Hoppe, C. Linsenmann, and A. Wixforth, *Reduced order modeling based shape optimization of surface acoustic wave driven microfluidic biochips*, Mathematics and Computers in Simulation, 82 (2012), pp. 1986–2003.

[10] A. Antoulas, *Approximation of Large-Scale Dynamical Systems (Advances in Design and Control)*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2005.

[11] A. Antoulas, C. Beattie, and S. Gugercin, *Interpolatory model reduction of large-scale dynamical systems*, in Efficient Modeling and Control of Large-Scale Systems, J. Mohammadpour and K. Grigoriadis, eds., Springer-Verlag, 2010.

[12] E. Arian, M. Fahl, and E. W. Sachs, *Trust-region proper orthogonal decomposition models by optimization methods*, in Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, 2002, pp. 3300–3305.

[13] W. E. Arnoldi, *The principle of minimized iteration in the solution of the matrix eigenvalue problem*, Quart. Appl. Math., 9 (1951), pp. 17–29.

[14] S. R. Arridge, *Optical tomography in medical imaging*, Inverse Problems, 15 (1999), pp. R41–R93.

[15] O. Bashir, K. Willcox, O. Ghattas, B. van Bloemen Waanders, and J. Hill, *Hessian-based model reduction for large-scale systems with initial condition inputs*, International Journal for Numerical Methods in Engineering, 73 (2008), pp. 844–868.

[16] U. Baur, C. Beattie, P. Benner, and S. Gugercin, *Interpolatory projection methods for parameterized model reduction*, SIAM J. Sci. Comput., 33 (2011), pp. 2489–2518.

[17] U. Baur and P. Benner, *Model reduction for parametric systems using balanced truncation and interpolation*, at-Automatisierungstechnik, 57 (2009), pp. 411–420.

[18] C. Beattie, S. Gugercin, and S. Wyatt, *Inexact solves in interpolatory model reduction*, Linear Algebra and its Applications, 436 (2012), pp. 2916–2943.

[19] P. Benner, S. Gugercin, and K. Willcox, *A survey of projection-based model reduction methods for parametric dynamical systems*, SIAM Review, 57 (2015), pp. 483–531.

[20] B. Bond and L. Daniel, *Parameterized model order reduction of nonlinear dynamical systems*, in IEEE/ACM Internat. Conf. on Computer-Aided Design, 2005. ICCAD-2005, 2005, pp. 487–494.

[21] T. Bui-Thanh, K. Willcox, and O. Ghattas, *Model reduction for large-scale systems with high-dimensional parametric input space*, SIAM J. Sci. Comput., 30 (2008), pp. 3270–3288.

[22] ——, *Parametric reduced-order models for probabilistic analysis of unsteady aerodynamic applications*, AIAA Journal, 46 (2008), pp. 2520–2529.

[23] M. Burger and S. Osher, *A survey on level set methods for inverse problems and optimal design*, European Journal of Applied Mathematics, 16 (2005), pp. 263–301.

[24] J. T. Bushberg, J. A. Seibert, E. M. Leidholdt Jr., and J. M. Boone, *The essential physics of medical imaging*, Lippincott Williams & Wilkens, 2002.

[25] L. Daniel, O. C. Siong, L. S. Chay, K. H. Lee, and J. White, *A multiparameter moment matching model reduction approach for generating geometrically parameterized interconnect performance models*, IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, 23 (2004), pp. 678–693.

[26] D. Darnell, R. Morgan, and W. Wilcox, *Deflated GMRES for systems with multiple shifts and multiple right-hand sides*, Linear Algebra and its Applications, 429 (2008), pp. 2415–2434.

[27] E. de Sturler, *Nested Krylov methods based on GCR*, J. Comput. Appl. Math., 67 (1996), pp. 15–41.

[28] E. de Sturler, S. Gugercin, M. Kilmer, S. Chaturantabut, C. Beattie, and M. O'Connell, *Nonlinear parametric inversion using interpolatory model reduction*, SIAM J. Sci. Comput., 37 (2015), pp. B495–B517.

[29] J. Degroote, J. Vierendeels, and K. Willcox, *Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis*, International Journal for Numerical Methods in Fluids, 63 (2010), pp. 207–230.

[30] O. Dorn and D. Lesselier, *Level set methods for inverse scattering*, Inverse Problems, 22 (2006), p. R67.

[31] V. Druskin, V. Simoncini, and M. Zaslavsky, *Solution of the time-domain inverse resistivity problem in the model reduction framework Part I. One-dimensional problem with SISO data*, SIAM J. Sci. Comput., 35 (2013), pp. A1621–A1640.

[32] L. Feng and P. Benner, *A robust algorithm for parametric model order reduction based on implicit moment matching*, in Appl. Math. Mech., 2008.

[33] L. Feng, P. Benner, and J. G. Korvink, *Parametric model order reduction accelerated by subspace recycling*, in Proceedings of the 48th IEEE Conference on Decision and Control, 2009.

[34] ——, *Subspace recycling accelerates the parametric macromodeling of MEMS*, Int. J. Numer. Meth. Engng., 94 (2013), pp. 84–110.

[35] S. Gugercin, A. Antoulas, and C. Beattie, *H2 model reduction for large-scale linear dynamical systems*, SIAM J. Matrix Anal. & Appl., 30 (2008), pp. 609–638.

[36] P. K. Gunupudi, R. Khazaka, M. S. Nakhla, T. Smy, and D. Celo, *Passive parameterized time-domain macromodels for high-speed transmission-line networks*, IEEE Trans. Microwave Theory ad Techniques, 51 (2003), pp. 2347–2354.

[37] B. Haasdonk and M. Ohlberger, *Efficient reduced models and a posteriori error estimation for parametrized dynamical systems by offline/online decomposition*, Mathematical and Computer Modelling of Dynamical Systems, 17 (2011), pp. 145–161.

[38] E. Haber, U. M. Ascher, and D. Oldenburg, *On optimization techniques for solving nonlinear inverse problems*, Inverse Problems, 16 (2000), pp. 1263–1280.

[39] J. Hadamard, *Lectures on CauchyâĂŹs Problem in Linear Differential Equations*, Yale University Press, New Haven, 1923.

[40] A. Hay, J. T. Borggaard, and D. Pelletier, *Local improvements to reduced-order models using sensitivity analysis of the proper orthogonal decomposition*, Journal of Fluid Mechanics, 629 (2009), pp. 41–72.

[41] M. Hinze and S. Volkwein, *Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control*, in Dimension Reduction of Large-Scale Systems, Springer, 2005, pp. 261–306.

[42] I. C. F. Ipsen and C. D. Meyer, *The idea behind Krylov methods*, American Mathematical Monthly, 105 (1997), pp. 889–899.

[43] M. Kilmer and E. de Sturler, *Recycling subspace information for diffuse optical tomography*, SIAM J. Sci. Comput., 27 (2006), pp. 2140–2166.

[44] ——, *A regularized Gauss-Newton trust region approach to imaging in diffuse optical tomography*, SIAM J. Sci. Comput., 33 (2011), pp. 3057–3086.

[45] R. Kress and W. Rundell, *Inverse obstacle scattering using reduced data*, SIAM J. Appl. Math., 59 (1998), pp. 442–454.

[46] K. Kunisch and S. Volkwein, *Proper orthogonal decomposition for optimality systems*, ESAIM: Mathematical Modelling and Numerical Analysis, 42 (2008), pp. 1–23.

[47] C. Lanczos, *An iterative method for the solution of the eigenvalue problem of linear differential and integral*, (1950).

[48] A. K. Louis, *Medical imaging: State of the art and future development*, Inverse Problems, 8 (1992), pp. 709–738.

[49] K. Meerbergen, *The solution of parameterized symmetric linear systems*, SIAM J. Matrix Anal. & Appl., 24 (2003), pp. 1038–1059.

[50] L. A. M. Mello, E. de Sturler, G. H. Paulino, and E. C. N. Silva, *Recycling Krylov subspaces for efficient large-scale electrical impedance tomography*, Computer Methods in Applied Mechanics and Engineering, 199 (2010), pp. 3101–3110.

[51] E. Miller, M. Kilmer, and C. Rappaport, *A new shape-based method for object localization and characterization from scattered field data*, IEEE Transactions on Geoscience and Remote Sensing, 38 (2000), pp. 1682–1696.

[52] V. A. Morozov, *On the solution of functional equations by the method of regularization*, Soviet Math. Dokl., 7 (1966), pp. 414–417.

[53] N. C. Nguyen, A. T. Patera, and J. Peraire, *A best points interpolation method for efficient approximation of parametrized functions*, Int. J. Numer. Meth. Engng., 73 (2008), pp. 521–543.

[54] S. Osher and J. Sethian, *Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations*, Journal of Computational Physics, 79 (1988), pp. 12–49.

[55] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM J. Numer. Anal., 12 (1975).

[56] H. Panzer, J. Mohring, R. Eid, and B. Lohmann, *Parametric model order reduction by matrix interpolation*, at-Automatisierungstechnik, 58 (2010), pp. 475–484.

[57] M. Parks, E. de Sturler, G. Mackey, D. Johnson, and S. Maiti, *Recycling Krylov subspaces for sequences of linear systems*, SIAM J. Sci. Comput., 28 (2006), pp. 1651–1674.

[58] C. Prud'homme, D. Rovas, K. Veroy, L. Machiels, Y. Maday, A. T. Patera, and G. Turinici, *Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods*, Journal of Fluids Engineering, 124 (2002), pp. 70–80.

[59] G. Rozza, D. B. P. Huynh, and A. T. Patera, *Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations: application to transport and continuum mechanics*, Arch. Comput. Methods Eng., 15 (2008), pp. 229–275.

[60] Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.

[61] A. Saibaba, T. Bakhos, and P. Kitanidis, *A flexible Krylov solver for shifted systems with application to oscillatory hydraulic tomography*, SIAM J. Sci. Comput., 35 (2013), pp. A3001–A3023.

[62] A. Saibaba, M. Kilmer, E. Miller, and S. Fantini, *Fast algorithms for hyperspectral diffuse optical tomography*, SIAM J. Sci. Comput., 37 (2015), pp. B712–B743.

[63] F. Santosa, *A level-set approach for inverse problems involving obstacles*, ESAIM: Control, Optimisation and Calculus of Variations, 1 (1996), pp. 17–33.

[64] V. Simoncini, *Restarted full orthogonalization method for shifted linear systems*, BIT Numerical Mathematics, 42 (2003), pp. 459–466.

[65] V. Simoncini and D. Szyld, *Recent computational developments in Krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., 14 (2007), pp. 1–59.

[66] R. Snieder and J. Trampert, *Inverse problems in geophysics*, Springer, 1999.

[67] A. Stathopoulos, A. M. Abdel-Rehim, and K. Orginos, *Deflation for inversion with multiple right-hand sides in QCD*, Journal of Physics: Conference Series, 180 (2009).

[68] G. W. Stewart and J.-g. Sun, *Matrix Perturbation Theory*, Academic Press, Inc., 1990.

[69] N.-Z. Sun, *Inverse problems in groundwater modeling*, Springer Netherlands, 1999.

[70] A. N. Tikhonov, V. Y. Arsenin, and F. John, *Solutions of ill-posed problems*, VH Winston Washington, DC, 1997.

[71] L. N. Trefethen and D. Bau III, *Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.

[72] K. van den Doel and U. Ascher, *On level set regularization for highly ill-posed distributed parameter estimation problems*, Journal of Computational Physics, 216 (2006), pp. 707–723.

[73] ——, *Dynamic level set regularization for large distributed parameter estimation problems*, Inverse Problems, 23 (2007), pp. 1271–1288.

[74] K. van den Doel, U. Ascher, and A. Leitao, *Multiple level sets for piecewise constant surface reconstruction in highly ill-posed problems*, Journal of Scientific Computing, 43 (2010), pp. 44–66.

[75] K. Veroy, C. Prud'homme, D. Rovas, and A. T. Patera, *A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations*, in 16th AIAA Computational Fluid Dynamics Conference, 2003.

[76] C. R. Vogel, *Computational methods for inverse problems*, 2002.

[77] S. Wang, E. de Sturler, and G. Paulino, *Large-scale topology optimization using preconditioned Krylov subspace methods with recycling*, Int. J. Numer. Meth. Engng., 69 (2007), pp. 2441–2468.

[78] A. G. Webb, *Introduction to biomedical imaging*, Wiley-IEEE Press, 2003.

[79] W. Yeh, *Review of parameter identification procedures in groundwater hydrology: The inverse problem*, Water Resources Research, 22 (1986), pp. 95–108.

[80] Y. Yue and K. Meerbergen, *Accelerating optimization of parametric linear systems by model order reduction*, SIAM J. Optim., 23 (2013), pp. 1344–1370.

[81] H. K. Zhao, T. Chan, B. Merriman, and S. Osher, *A variational level set approach to multiphase motion*, Journal of Computational Physics, 127 (1996), pp. 179–195.

[82] M. Zhdanov, *Geophysical inverse theory and regularization problems*, Elsevier Science Ltd., 2002.