A multilevel algorithm for inverse problems with elliptic PDE constraints

George Biros and Günay Doğan

School of Engineering and Applied Science, University of Pennsylvania, Philadelphia, PA 19104

E-mail: biros@seas.upen.edu and gdogan@seas.upenn.edu

Abstract. We present a multilevel algorithm for the solution of a source identification problem in which the forward problem is an elliptic partial differential equation on the 2D unit box. The Hessian corresponds to a Tikhonov-regularized first-kind Fredholm equation. Our method uses an approximate Hessian operator for which, first, the spectral decomposition is known, and second, there exists a fast algorithm that can perform the spectral transform. Based on this decomposition we propose a Conjugate Gradients solver which we precondition with a multilevel subspace projection scheme. The coarse-level preconditioner is an exact solve and the finer-levels preconditioner is one step of the scaled Richardson iteration.

As a model problem, we consider the 2D-Neumann Poisson problem with variable coefficients and partial observations. The approximate Hessian for this case, is the Hessian related to a problem with constant coefficients and full observations. We can use a Fast Cosine Transform to compute the spectral transforms. We examine the effect of using Galerkin or level-discretized Hessian operators and we provide results from numerical experiments that indicate the effectiveness of the method for full and partial observations.

1. Introduction

We propose a multilevel algorithm for a certain class of linear inverse problems of the form

$$Hu = g, \tag{1}$$

were u is the inversion parameter and $H = \beta I + H_0$, with H_0 being a compact operator and β a regularization parameter. Our method is designed for source identification problems in which the state variable is the solution of the 2D variable-coefficient Poisson problem with homogeneous Neumann conditions on the unit box with full and partial observations. Extensions of this model find applications in geophysics and acoustics [10, 16].

The source identification problem is posed using a PDE-constrained optimization formulation [5]

$$\min_{y,u} \frac{1}{2}\|Qy - d\|^2 + \frac{\beta}{2}\|u\|^2, \quad \text{subject to } Jy + u = 0,$$

where y is the state variable and d is the data. If we eliminate the state variable using $y = -J^{-1}u$, (1) corresponds to the first-order optimality condition, with $H_0 = J^{-T}Q^TQJ^{-1}$, where J is the forward problem operator, J^T is the adjoint problem operator, and Q is an observation operator. In general, choosing β is difficult. Here are not concerned with the choice of β. Our objective is to develop efficient linear solvers for (1), even in the case of vanishing β. Although the case $\beta = 0$ has little practical significance, designing solvers that scale for all values of β provides a method that works well for both low and high-fidelity reconstruction problems.

Our algorithm relies on constructing an operator B that approximates H well (so that $B^{-1}H$ has clustered eigenvalues) and whose spectral decomposition can be computed using a fast algorithm. Assuming the availability of B, our scheme consists of the following steps: (1) define a subspace decomposition using subspaces defined by eigenvectors of B and use this decomposition to construct an additive multilevel preconditioner for H (see Oswald's chapter in [20]); (2) define the preconditioner by exact solves on the coarse level and one step of preconditioned (by B^{-1}) Richardson relaxation in the finer levels; and (3) use the multilevel preconditioner with Conjugate Gradients to solve (1). For our source identification inverse problem, B corresponds to the Hessian of a constant-coefficient forward problem and full observations ($Q = I$). In this case, the spectrum is trivial to compute analytically; the subspace projections can be computed in $\mathcal{O}(n \log n)$ time using the fast cosine transform (FCT). In addition, if we discretize using a Galerkin scheme and a cosine basis, then B is a diagonal operator and thus, trivial to apply.

1.1. Contributions

This paper builds on the work of King [19] and Kaltenbacher [17]. The objective of these works was to design efficient multigrid schemes for first-kind Fredholm equations.

That is, solution schemes that work well even in the case of vanishing regularization. They designed a multigrid scheme in which function analytic estimates of the lowest singular value of the operator were used.

We propose a multigrid scheme that has β-independent convergence properties. But are smoother is based on constructing an exact Hessian representation for constant coefficient elliptic PDEs on regular geometries and full observations and using it as a preconditioner for variable coefficients, partial observations, and penalty-based formulations of inverse problems on arbitrary geometries.

Our algorithm that is less general, as it requires efficient algorithms for B and its eigenspace. It has the advantage of incorporating more analytic and problem-specific structure into the preconditioner. In addition, we would like to design alternative approaches to multigrid schemes based on stencils (e.g., [6]), for the cases in problems in which the forward problem is discretized using non-local discretizations (e.g., Fourier expansions or Greens' functions). Note that multigrid methods have been proposed to solve the spectral discretization of the forward problem [21, 22, 14, 15]. In [1], Adavani and Biros proposed a V-cycle multigrid preconditioner that uses a stationary iterative solver as a smoother, combined with an approximate filtering operator that restricts the smoothing steps in the high-frequency Krylov subspace. They restricted their attention to 1D parabolic problems with smooth coefficients, and finite-difference-based discretizations. They showed that a multigrid preconditioner using orthogonal projections and analytically available spectral information results in a mesh-independent scheme; they also performed comparisons with other existing schemes. Here, we look at a slightly modified scheme for a different forward problem, a 2D elliptic PDE. We use an multilevel additive subspace preconditioner in which the the smoother is based on B and requires matvecs with H only at the coarse and fine levels, does not require eigenvalue estimates for H_0 (unlike [1]). For 3D and evolution problems, this approach can potentially offer substantial computational savings as H requires inverting the forward and adjoint operators. We conduct numerical experiments in which we examine the scalability of the solver and we compare with the unpreconditioned case, and single level diagonal scaling.

We test the method for smooth, oscillatory, and discontinuous coefficients and for full and partial observation operators. The preconditioner performs well; it reduces the number of iterations drastically resulting in a nearly β-independent scheme. *Thus, despite the restrictive assumptions we impose in order to obtain an analytic preconditioner, by combining it with a multigrid acceleration, we are able to construct a scheme that is efficient for a broader set of forward and observation operators.*

1.2. Background and Related Work

There is a rich literature on methods for ill-posed infinite-dimensional linear, least-squares problems. We are interested in so-called *"matrix-free"* methods that do not assume element-wise access to H but only the action of H on a vector. If we fix the

regularization parameter β to a positive value, H is a compact perturbation of the identity and thus, has a bounded (mesh-independent) condition number which scales as $\mathcal{O}(1/\beta)$. Using CG to solve a linear system involving H requires $\mathcal{O}(1/\sqrt{\beta})$ iterations. In a practical setting, one chooses the discretization size (for u) and β based on the range of H and the noise level in g. The more information we can recover from H and the higher the signal-to-noise ratio, the smaller the value of β and the finer the required discretization of u. Therefore, when we study algorithmic scalability of a solver for (1), we should consider β to be mesh-dependent. This implies the deterioration of the condition number of H with decreasing mesh size (and thus β), which in turn suggests the need to design β-independent preconditioners for (1).

Standard preconditioning techniques, e.g., incomplete factorizations or Jacobi relaxations cannot be used, because they require an assembled matrix [4]. Another option is to use a multigrid method. These methods have been developed mainly for linear systems arising from the discretization of elliptic PDEs and second-kind integral operators. Besides the pioneering work of [8] for differential operators and of [12] for second-kind Fredholm integral operators, there exists significant work on multigrid methods for optimal control problems [3, 6, 7, 11].

Unlike differential operators and second-kind Fredholm operators, the unregularized reduced Hessian (H_0) is an operator of the first kind. There has been little work on multigrid algorithms for such problems. In [13], multilevel and domain decomposition preconditioners were proposed for integral equations of first-kind. Multigrid solvers for Tikhonov-regularized ill-posed problems were discussed in the pioneering work of King in [18] and [19]. Such problems were further analyzed in [17] in which a V-cycle multigrid solver was proposed. A multigrid preconditioner based on that work was also used in [2] (for convection-diffusion constraints) and in [1] (for reaction-diffusion constraints).

1.3. Organization of the paper

The rest of the paper is organized as follows. In Section 2, we give the formulation and discretization of the inverse problem (1). In Section 3, we examine the spectrum of the reduced Hessian and discuss how it allows us to design a fast solver for (1). In Section 4, we introduce the multilevel preconditioner that we use to solve the discretized system. We conclude with Section 5, where we document the effectiveness of our algorithm with several experiments.

Notation. In the following, we use regular typeface to indicate an infinite dimensional operator (upper case) or function (lower case), and we use bold typeface for discrete counterparts. For example, $\mathbf{f}$ denotes the coefficient vector for function f in the current discretization. Similarly the discretization of an operator A will be denoted by $\mathbf{A}$.

2. Formulation and Discretization

We consider the following problem:

$$\min_{y,u} \mathcal{J}(y,u) := \frac{1}{2}\int_\Omega (q(x)y(x) - d(x))^2 dx + \frac{\beta}{2}\int_\Omega u(x)^2 dx, \tag{2}$$

subject to

$$\alpha(x)y(x) - \Delta y(x) + u(x) = 0 \quad \text{in} \quad \Omega, \quad \frac{\partial y(x)}{\partial n(x)} = 0 \quad \text{on} \quad \partial\Omega. \tag{3}$$

Here y is the state variable, u is the inversion variable; $q \geq 0$ is a weight that models a partial observation operator‡; the coefficient $\alpha > 0$ is assumed to be piecewise smooth and bounded and $\Omega = [0,1]^2$. (For notational simplicity, we suppress the dependence on x in the remaining text.)

By forming a Lagrangian, introducing the adjoint variable p, and taking variations with respect to y, u and p, we obtain the first-order optimality conditions for (2):

$$\begin{aligned} &\alpha y - \Delta y + u = 0 \quad \text{in} \quad \Omega, \qquad \frac{\partial y}{\partial n} = 0 \quad \text{on} \quad \partial\Omega, \\ &\alpha p - \Delta p + q(qy - d) = 0 \quad \text{in} \quad \Omega, \qquad \frac{\partial p}{\partial n} = 0 \quad \text{on} \quad \partial\Omega, \\ &\beta u + p = 0 \quad \text{in} \quad \Omega. \end{aligned} \tag{4}$$

In abstract form, we can write

$$Jy + u = 0, \quad J^T p + Q^T(Qy - d) = 0, \quad \beta u + p = 0, \tag{5}$$

where J^T is the adjoint of J. By eliminating the state and adjoint variables, we arrive at the reduced system

$$Hu = g. \tag{6}$$

The reduced Hessian H is defined by $H = J^{-T}Q^TQJ^{-1} + \beta I$. The right hand side g is defined by $g = -J^{-T}Q^Td$. To find a numerical solution to problem (2), we discretize equation (6) and obtain the corresponding discrete linear system

$$\mathbf{Hu} = \mathbf{g}. \tag{7}$$

2.1. Discretization

We seek a solution $u^h(x) = \sum_k u_k c_k(x)$ in terms of the orthonormal cosine basis functions

$$c_k(x) = \begin{cases} 1 & \text{if} \quad k_1 = k_2 = 0, \\ \sqrt{2}\cos(k_1\pi x_1) & \text{if} \quad 1 \leqslant k_1 \leqslant n, k_2 = 0, \\ \sqrt{2}\cos(k_2\pi x_2) & \text{if} \quad k_1 = 0, 1 \leqslant k_2 \leqslant n, \\ 2\cos(k_1\pi x_1)\cos(k_2\pi x_2) & \text{if} \quad 1 \leqslant k_1, k_2 \leqslant n, \end{cases} \tag{8}$$

‡ Partial observations are typically convolutions, for example, sums of delta functions. Here, for simplicity, we assume a simple pointwise weighting.

where $x = (x_1, x_2)$, $k = (k_1, k_2)$, $0 \leqslant k_1, k_2 \leqslant n$ and n specifies the total number of frequencies that we seek to recover. We chose this basis because c_k are the eigenfunctions of the forward operator $\mathcal{L}$ when we have constant coefficient α. We have $Jc_k = \lambda_k c_k$, where

$$\lambda_k = \lambda_{k_1 k_2} = \alpha + \pi^2 k^2, \quad k^2 = k_1^2 + k_2^2. \tag{9}$$

If additionally we have $q(x) = 1$, namely full domain observations, then the functions c_k are the eigenfunctions of the reduced Hessian H as well. This fact will allow us to construct a preconditioner for the case with variable α, q.

We use the basis functions (8) to discretize the operators J, Q, and thereby, H. The matrices corresponding to J, Q are defined by $\mathbf{J_{kl}} = (Jc_k, c_l)$, $\mathbf{Q_{kl}} = (qc_k, c_l)$ where $(f, g) = h^2 \sum_{ij} f(x_i) g(x_j)$, $h = 1/n$, $x_i = (i_1 h, i_2 h)$, $0 \leq i_1, i_2 \leq n$, x_j similarly. Thus, we obtain the discretized reduced system

$$\mathbf{Hu} = \mathbf{g}, \tag{10}$$

where

$$\mathbf{H} = \beta \mathbf{I} + \mathbf{J^{-T} Q^T Q J^{-1}}, \tag{11}$$

and $\mathbf{g} = -\mathbf{J^{-T} Q^T d}$, $\mathbf{d}$ is the vector of the cosine coefficients of the domain observation function d.

We intend to solve (10) using an iterative algorithm, for which we need to define the matvec for $\mathbf{H}$. The key operation in the matvec is the inversion of the forward operator $\mathbf{J}$, that is, solving the forward problem

$$\mathbf{Ju} = \mathbf{f}. \tag{12}$$

For constant coefficient α, $\mathbf{J}$ is a diagonal matrix and inverting $\mathbf{J}$ is trivial. However for the non-constant coefficient case, it turns out to be a dense matrix. Therefore, we resort to an iterative algorithm for (12) as well and for that we need an efficient way to perform the matvec $\mathbf{J}$. Writing $\alpha(x) = \alpha_0 + \eta(x)$, where $\alpha_0 = \int_\Omega \alpha(x) dx$, we can decompose $\mathbf{J}$ into a dense matrix $\mathbf{M}$ and a diagonal matrix $\mathbf{\Lambda}$, so that $\mathbf{J} = \mathbf{M} + \mathbf{\Lambda}$, where

$$\mathbf{M_{kl}} = (\eta c_k, c_l), \quad \mathbf{\Lambda_{kl}} = (\alpha_0 + \pi^2 k^2) \delta_{kl}. \tag{13}$$

Note that we can write $\mathbf{M} = \mathbf{F} \mathfrak{N} \mathbf{F^T}$, where $\mathbf{F}, \mathbf{F^T}$ denote discrete cosine transform and inverse cosine transform respectively, $\mathfrak{N}$ is a diagonal matrix defined by $\mathfrak{N}_{ij} = \eta(x_i) \delta_{ij}$. This allows us to compute the matvec $\mathbf{J}$ in $\mathcal{O}(n \log n)$ time using FCT. So we can use PCG to solve (12) with a cost of $\mathcal{O}(n \log n)$ per iteration.

Moreover we find $\mathbf{\Lambda}^{-1}$ to be a good preconditioner for (12) to speed up the convergence of PCG. We test this with three different coefficients α:

$$\begin{aligned} \text{smooth}(x) &= 1 + 10 \left(\frac{1}{2} + \frac{1}{2} \cos(2\pi x_1) \right), \\ \text{osc}(x) &= 1 + 10 \left(\frac{1}{4} + \left(x_1 - \frac{1}{2} \right)^2 + \left(x_2 - \frac{1}{2} \right)^2 + \frac{1}{4} \sin(6\pi x_1) \right), \end{aligned} \tag{14}$$

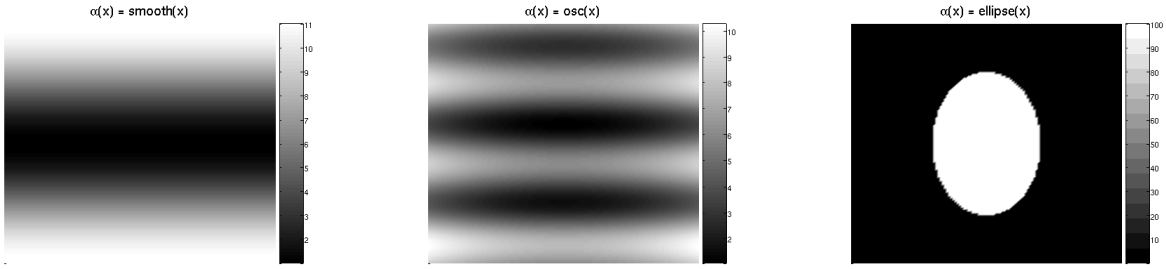

Figure 1. Variable coefficients $\alpha(\mathbf{x})$. The functions used as examples of the variable coefficient $\alpha(x)$ in the forward problem (12). Definitions of these functions are given in (14). The function smooth(x) is a relatively smooth cosine wave of amplitude ten. The function osc(x) is an oscillatory function obtained as the sum of a sine wave and a quadratic function. The function ellipse(x) is a discontinuous function, with a jump of four orders of magnitude (100 at the support of the ellipse defined in (14) and 10^{-2} elsewhere).

$$\text{ellipse}(x) = \begin{cases} 100, & \text{if } x \text{ is in the ellipse } \frac{(x_1-0.5)^2}{0.2^2} + \frac{(x_2-0.5)^2}{0.3^2} = 1, \\ 10^{-2}, & \text{otherwise,} \end{cases}$$

We compute the condition numbers for $\mathbf{\Lambda}^{-1}\mathbf{J}$ and see that they are mesh-independent. The condition numbers are equal to 1.55, 1.22, 1.25 for smooth(x), osc(x), ellipse(x), respectively. This is to be expected as $\mathbf{I} + \mathbf{\Lambda}^{-1}\mathbf{M}$ is a compact perturbation of the identity. Hence, PCG converges in a small mesh-independent number of iterations in each case. (In our experiments, it takes 8–12 iterations to converge the discretized system to machine accuracy.)

3. Spectral Properties of the Reduced Hessian

In this section, we investigate the spectrum of the reduced Hessian to get insight for the development of a fast solver for the reduced system (10). There are two key ingredients that allow us to design the fast solver:

(i) Decomposition of the spectrum of the reduced Hessian $\mathbf{H}$ into smooth and oscillatory parts, to enable a multilevel approach,

(ii) Characterization of the eigenvectors of $\mathbf{H}$ with cosine functions so that we can use FCT to switch back and forth between the real number grid and the cosine coefficients.

Looking at the reduced Hessian closely and assuming $\mathbf{Q} = \mathbf{I}$, we have

$$\mathbf{H} = \beta\mathbf{I} + \mathbf{J}^{-T}\mathbf{J}^{-1}, \tag{15}$$

and we can see that the key to understanding the spectrum of $\mathbf{H}$ is the spectrum of the forward operator $\mathbf{J}$. Assuming that the spectrum of $\mathbf{J}$ is given by the eigenvector-eigenvalue pairs $(\mathbf{v_k}, \lambda_k)$, one can easily show that the eigenvector-eigenvalue pairs of $\mathbf{H}$ are given by $(\mathbf{v_k}, \beta + \frac{1}{\lambda_k^2})$.

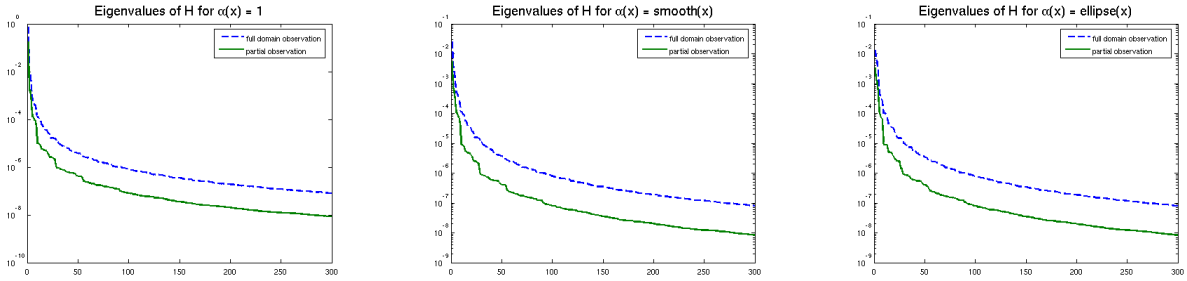

Figure 2. Spectrum of the reduced Hessian. These figures show the first three hundred eigenvalues of the reduced Hessian $\mathbf{H}$ for $\beta = 0$. The eigenvalues were computed numerically by a discretization of $\mathbf{H}$ on a 64×64 grid. The spectra are given for the cases of full domain observations and partial observations (illustrated in Figure 5). We can see that the eigenvalues decay faster when we have partial observations.

Returning to the specific forward problem, for the constant coefficient inverse problem (constant α), we have

$$\sigma_k = \beta + \frac{1}{(\alpha + \pi^2 k^2)^2}. \tag{16}$$

Consequently we can identify the smallest and largest eigenvalues of $\mathbf{H}$. These are $\sigma_{min} = \sigma_n = \beta + \frac{1}{\lambda_n^2}$, $\sigma_{max} = \sigma_1 = \beta + \frac{1}{\lambda_1^2}$ respectively. Then the condition number of the reduced Hessian is

$$\kappa(\mathbf{H}) = \frac{\sigma_1}{\sigma_n} < 1 + \frac{1}{\beta\lambda_1^2} = \mathcal{O}(\frac{1}{\beta}). \tag{17}$$

We can see that the condition number increases as we decrease β.

3.1. Spectral Decomposition of the Reduced Hessian

We would like to introduce a decomposition of the reduced Hessian $\mathbf{H}$ into two parts, $\mathbf{H_s}$ and $\mathbf{H_o}$, acting on smooth and oscillatory parts of given functions respectively. This decomposition is critical for the design of our multilevel preconditioner.

Assume that $\alpha(x)$ is constant and $\mathbf{Q} = \mathbf{I}$. Then, in k-space, $\mathbf{H}$ is diagonal and its spectrum is given by $(\mathbf{e^k}, \sigma_k)$ where $\mathbf{e^k}$ is the unit vector with k^{th} component equal to 1 ($\mathbf{e^k}$ represents $c_k(x)$). We can define a projection operator $\mathbf{P_s}$ as

$$\mathbf{P_s e^k} = \begin{cases} \mathbf{e^k} & \text{if } \mathrm{k}_1, \mathrm{k}_2 < \frac{\mathrm{n}}{2} \text{ in } \mathrm{k} = (\mathrm{k}_1, \mathrm{k}_2), \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

$\mathbf{P_s}$ projects given coefficient vectors to the subspace of smooth eigenvectors. Then the projection to the oscillatory subspace is given by

$$\mathbf{P_o} = \mathbf{I} - \mathbf{P_s}. \tag{19}$$

Using these we can obtain the decomposition of $\mathbf{H}$:

$$(\mathbf{P_s} + \mathbf{P_o})\mathbf{H}(\mathbf{P_s} + \mathbf{P_o}) = \mathbf{P_s H P_s} + \mathbf{P_o H P_o}. \tag{20}$$

This follows from

$$\mathbf{P_s H P_o e^k} = \mathbf{0}, \qquad \mathbf{P_o H P_s e^k} = \mathbf{0}. \tag{21}$$

So instead of solving $\mathbf{Hu} = \mathbf{g}$, we can solve

$$\mathbf{H_s u_s} = \mathbf{P_s H P_s u_s} = \mathbf{P_s g} \tag{22}$$

$$\mathbf{H_o u_o} = \mathbf{P_o H P_o u_o} = \mathbf{P_o g}, \tag{23}$$

and obtain the solution $\mathbf{u} = \mathbf{u_s} + \mathbf{u_o}$. This will be the basis of our multilevel algorithm.

Let us also remark on the case with non-constant coefficient α and $\mathbf{Q} \neq \mathbf{I}$. For non-constant α, $\mathbf{H}$ will not be diagonal and, therefore, it will not have the unit vectors $\mathbf{e^k}$ as its eigenvectors. Consequently, the condition (21) will be violated. An attempt to solve (10) using (22)–(23) will miss information about the solution. However, we are interested in using this scheme only as a preconditioner, and this inexact decomposition still serves our purposes. We will elaborate this point in the following section.

4. A Multilevel Preconditioner

In this section, we will describe a multilevel algorithm for the linear system

$$(\beta \mathbf{I} + \mathbf{J}^{-T}\mathbf{Q}^T\mathbf{Q}\mathbf{J}^{-1})\mathbf{u} = \mathbf{g}. \tag{24}$$

Multilevel algorithms can be seen as a class of iterative algorithms (see [9] for an introduction). What sets them apart from traditional iterative algorithms is that they converge in a constant number of iterations independent of mesh size n, resulting in overall time complexity of $\mathcal{O}(n)$ or $\mathcal{O}(n \log n)$. As we aim for large-scale inverse problems, this property makes the multilevel approach the appropriate foundation for our method as well. In addition, we want a scheme that is both mesh- and regularization-independent.

The main idea is to treat the oscillatory and smooth error components of an approximate solution in two different subspaces, the *fine* and *coarse*. Given an approximate solution $\mathbf{u}$, we apply an iterative *smoother* to $\mathbf{u}$ (or equivalently, we *relax* $\mathbf{u}$) in order to damp the oscillatory error components of $\mathbf{u}$. Then, the remaining smooth error components can be approximated well on the coarse subspace, where we solve for the smooth error. We use this to correct the approximate solution $\mathbf{u}$ (see Algorithm 1). The three key components of a multilevel algorithm are thus:

(i) *the intergrid transfer operators,* namely *restriction*, which gives the coarse subspace representation of vector represented on the fine subspace, and *prolongation*, which interpolates a coarse subspace representation to a fine subspace representation,

(ii) *the smoother* or *the relaxation scheme* that damps the oscillatory error components of a given approximate solution on the fine subspace, and

(iii) *the coarse-grid correction*, which gives an approximation to the smooth error component in order to correct the given approximate solution.

In the following, we describe the specific choices that we use to implement a multilevel solver for (10).

Algorithm 1 The two-grid preconditioner for $\mathbf{H^h u^h = g^h}$

1: $\mathbf{u^h} = \text{relax}(\mathbf{H^h}, \mathbf{g^h})$
2: $\mathbf{g^{2h}} = \mathbf{I_h^{2h} g^h}$
3: $\mathbf{u^{2h}} = (\mathbf{H^{2h}})^{-1}\mathbf{g^{2h}}$
4: $\mathbf{u^h} = \mathbf{u^h} + \mathbf{I_{2h}^h u^{2h}}$

4.1. Intergrid Transfer Operators

As the functions are represented by their cosine coefficients in our formulation, the restriction and prolongation operators $\mathbf{I_h^{2h}}, \mathbf{I_{2h}^h}$ are defined by their actions on given coefficient vectors. To restrict a given vector $\mathbf{v^h}$ to the coarse subspace, we keep the smooth coefficients, i.e. $\mathbf{v^h_{k_1 k_2}}$, $0 \leqslant k_1, k_2 < \frac{n}{2}$, and delete the remaining oscillatory coefficients. We store the smooth coefficients in $\mathbf{v^{2h}}$. This corresponds to filtering out the high frequencies in the function represented by $\mathbf{v^h}$. In order to prolong a vector $\mathbf{v^{2h}}$ of n^2 coefficients at the coarse subspace to the fine subspace, we pad $\mathbf{v^{2h}}$ with zeros for the entries corresponding to the oscillatory coefficients and obtain the coefficient vector $\mathbf{v^h}$ with $(2n)^2$ coefficients.

4.2. The Relaxation Scheme

The purpose of the relaxation scheme is to damp the oscillatory components of the error in the approximate solution. For this we need an iterative scheme whose action is approximately restricted to the oscillatory part of the spectrum of $\mathbf{H}$. Traditionally, with elliptic problems, iterative solvers such as Jacobi, Gauss-Seidel or Conjugate Gradient (CG) are used for this purpose. However, these methods turn out to be suboptimal for our problem. Jacobi and Gauss-Seidel require the explicit matrix, which we do not construct. CG, which does not require the explicit matrix but has the opposite effect of a smoother on our problem. It can be shown that CG damps smooth error components faster than the oscillatory components, and thus, acts as a rougher (see [1]). This is because the spectrum of $\mathbf{H}$ is such that the smoother eigenvectors have larger eigenvalues. This is in contrast to the smoothing properties of the forward operator $\mathbf{J}$.

At this point, we will find that the spectral decomposition (22)-(23) should be useful. By choosing to iterate on

$$\mathbf{H_o u = P_o g}, \tag{25}$$

with an appropriate scheme, we can improve only the oscillatory components of the error. So let us take a moment to examine the conditioning of $\mathbf{H_o}$. We define the pseudo-condition number of $\mathbf{H_o}$ by $\tilde{\kappa}(\mathbf{H_o}) = \sigma_{max}/\sigma_{min}$, the ratio of maximum and minimum positive eigenvalues of $\mathbf{H_o}$ respectively. For the case of constant coefficient α and full domain observations $\mathbf{Q} = \mathbf{I}$, we have

$$\sigma_{min} = \beta + \frac{1}{\left(\alpha + \pi^2(n^2+n^2)\right)^2} = \beta + \frac{1}{4\pi^4\left(n^2 + \frac{\alpha}{2\pi^2}\right)^2},$$

	$\alpha(x)=1$	$\alpha(x)=\text{ellipse}(x)$		$\alpha(x)=\text{osc}(x)$		$\alpha(x)=\text{smooth}(x)$	
	$\kappa(\mathbf{H_o})$	$\kappa(\mathbf{H_o})$	$\kappa(\mathbf{B^{-1}H_o})$	$\kappa(\mathbf{H_o})$	$\kappa(\mathbf{B^{-1}H_o})$	$\kappa(\mathbf{H_o})$	$\kappa(\mathbf{B^{-1}H_o})$
n=32	56.3	56.5	1.621	56.1	1.007	56.1	1.005
n=64	60.1	60.0	1.274	60.0	1.002	60.0	1.001
n=128	62.0	62.0	1.132	62.0	1.001	62.0	1.000
n=256	63.0	62.7	1.065	62.9	1.000	62.4	1.000

Table 1. Conditioning of the smoother. We list the pseudo-condition numbers for $\mathbf{H_o}$ and $\mathbf{B^{-1}H_o}$ for various α and mesh sizes. We can see that the pseudo-condition number is uniformly bounded for each α and preconditioning $\mathbf{H_o}$ with $\mathbf{B}$ makes a dramatic improvement.

$$\sigma_{max} = \beta + \frac{1}{\left(\alpha + \pi^2((n/2)^2 + 0^2)\right)^2} = \beta + \frac{1}{\frac{\pi^4}{16}\left(n^2 + \frac{4\alpha}{\pi^2}\right)^2}.$$

Here we obtain σ_{min} by considering the most oscillatory eigenmode of $\mathbf{H_o}$ with $k=(n,n)$, and we obtain σ_{max} as the eigenvalue corresponding to the least oscillatory eigenmodes $k=(n/2,0)$ or $(0,n/2)$ with nonzero eigenvalues. Note that the eigenmodes with $k=(k_1,k_2)$, $0 \leqslant k_1,k_2 < n/2$ are in the null space of $\mathbf{H_o}$.
Now, for $\beta = 0$,

$$\tilde{\kappa}(H_o) = \frac{\sigma_{max}}{\sigma_{min}} = 64\left(1 - \frac{7\alpha}{2\pi^2 n^2 + 8\alpha}\right)^2.$$

We can see that $\tilde{\kappa} \to 64$ as n increases. So for large n, we achieve mesh-independent convergence of the smoother. Let us point out again this is a worst case result for $\beta = 0$. One can easily show that the pseudo-condition number $\tilde{\kappa}(\mathbf{H_o})$ improves with increasing β. Hence an iterative method applied to (25) will converge in a number of iterations independent of the mesh size, which will decrease as we set β to a positive number.

To implement the smoother, we apply Richardson iterations to (25). So the relaxation iterations are given by

$$\mathbf{u^{k+1}} = \mathbf{u^k} + \omega\mathbf{B^{-1}}(\mathbf{P_o g} - \mathbf{H_o u^k}). \tag{26}$$

A suitable choice of the preconditioner $\mathbf{B}$ significantly improves the convergence of this scheme. We define $\mathbf{B}$ by

$$\mathbf{B_{kl}} = \left(\beta + \frac{1}{(\alpha_0 + \pi^2 k^2)^2}\right)\delta_{kl}, \tag{27}$$

where α_0 is the first cosine coefficient of α. In this way, we find by numerical computation that both $\sigma_{min}(\mathbf{B^{-1}H_o})$ and $\sigma_{max}(\mathbf{B^{-1}H_o})$ are very close to 1 for various non-constant coefficients α, such as those in (14). This finding is also helpful for determining the value of ω in (26). The optimal value of ω is given by $\omega_{opt} = 2/(\sigma_{min} + \sigma_{max})$, so in our case, we set $\omega = 1$. In our experiments, we observe that one step of (26) suffices as an effective smoother.

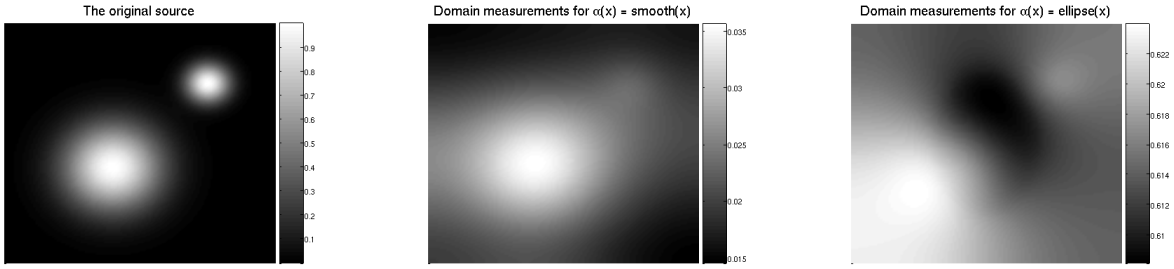

Figure 3. The source and the observations. The plot on the left shows the exact source that we used to test our method. It consists of two Gaussians, a smooth one and a sharp one. The plot in the middle and the plot on the right show the observations obtained from the source by solving the forward problem (12) with the coefficient $\alpha(x)$ set to smooth(x) and ellipse(x) respectively.

4.3. Coarse-grid Correction

Now we define the coarse-grid problem to compute the smooth component of the error. For this we need to specify the coarse-grid operator $\mathbf{H^{2h}}$ (in line 4 of Algorithm 1). Since we take the decomposition (22)-(23) as the basis of our algorithm, the correct $\mathbf{H^{2h}}$ is defined by

$$\mathbf{H^{2h}} = \mathbf{I_h^{2h}} \mathbf{H_s} \mathbf{I_{2h}^h}, \qquad \mathbf{H_s} = \mathbf{P_s} \mathbf{H^h} \mathbf{P_s}. \tag{28}$$

This is the so-called *Galerkin* operator. The disadvantage of the Galerkin operator is that the matvec defined by (28) has the same cost as $\mathbf{H^h}$, the fine-grid operator. For this reason, we seek an alternate definition, the *non-Galerkin* operator

$$\mathbf{H^{2h}} = \beta \mathbf{I} + (\mathbf{J^{2h}})^{-\mathbf{T}} (\mathbf{Q^{2h}})^{\mathbf{T}} \mathbf{Q^{2h}} (\mathbf{J^{2h}})^{-\mathbf{1}}. \tag{29}$$

The operators $\mathbf{J^{2h}}, \mathbf{Q^{2h}}$ are defined by sampling the functions α, q on the coarse grid. In this way, the cost of applying the matvec $\mathbf{H^{2h}}$ is reduced to a quarter of that for $\mathbf{H^h}$.

The definitions (28) and (29) are equivalent when we have constant coefficient α and full domain observations, i.e. $q = 1$. But this is not true when α is variable or we have partial domain observations. In that case, the coarse-grid correction via the non-Galerkin operator (29) will not recover the smooth error completely. Thus, a preconditioner that uses (29) will not be as effective (with respect number of iterations) as the one using the Galerkin operator (28) with respect to the number of iterations. However, it has a higher per-iteration cost—compared to the non-Galerkin approach. Since we are using a MATLAB implementation which has not been optimized for efficiency, we postpone precise comparisons between the two. In the next section, we present several experiments demonstrating the effectiveness of the algorithm.

5. Experiments

In this section, we present a number of experiments to test the effectiveness of our multilevel solver. As a test case, we reconstructed the source given in Figure 3

from full and partial domain observations. We considered the problem for four different coefficients: $\alpha(x) = 1, \text{smooth}(x), \text{osc}(x), \text{ellipse}(x)$. The definitions of the non-constant coefficients are given in (14). The last one, ellipse, resembles the case of a penalty approach formulation for a PDE defined on a arbitrary domain. For each case, we generated the observations by solving the forward problem at a mesh size of n=1024, then interpolating for smaller mesh sizes. Besides the interpolation error, we did not add additional noise for most of the experiments. The presence of noise deteriorates the reconstruction, but makes the solution of the linear problem in our formulation easier (as we would have to increase the value of β and make high-resolution discretizations of u unnecessary). As we are interested only in testing algorithmic scalability (we are not proposing some novel reconstruction formulation) we initially set the noise value to zero.

Full domain observations. We performed experiments for full domain observations, namely $Q = I$ and partial observations. For the former, we computed the reconstructions for n=32,64,128,256 and $\beta = \sigma_{0.05}, \sigma_{0.25}, \sigma_{0.5}$. Here $\sigma_{0.05}$ denotes the smallest of largest 5% eigenvalues of H with $\alpha = 1$. By setting $\beta = \sigma_{0.05}$, we aimed to recover roughly 5% of the eigenmodes in the solution. The reconstruction errors for each case are given in Figure 4.

We compared four different algorithms for full domain observations: first, we computed the solutions using PCG with no preconditioning; second, with diagonal preconditioning by $\mathbf{B}$ as defined in (27); third, with a two-level preconditioner; and fourth, with the multilevel preconditioner. For the multilevel preconditioner, the coarse-grid operator was defined by the non-Galerkin operator (29), and the coarse level was taken to be n=16 (in all of the experiments). We terminated PCG when the relative residual $\frac{|r|}{|r_0|}$ dropped below 10^{-12}.

We can see in Table 2 that our multilevel algorithm gives mesh-independent and β-independent performance. Interestingly, this is true for the diagonal preconditioner as well. Roughly speaking, the smoother part (which is bounded) of the spectrum is the one that differs significantly between the variable and constant-coefficient cases, whereas the vanishing eigenvalues approach the diagonal scaling with increasing mesh size. For this reason, the diagonal preconditioner gives almost mesh-independent convergence in practice. Of course this spectral alignment is true only when the eigenvectors of $\mathbf{B}$ are a good approximation of the eigenvectors of $\mathbf{H}$, especially in the case of $Q = I$. Both approaches perform favorably compared to the unpreconditioned system. The multilevel method converges in significantly fewer iterations than PCG with diagonal preconditioning. On the other hand, we see in Table 2 that the two-level scheme outperforms the multilevel scheme in all instances. This is to be expected because we perform an exact solve at the coarse level.

Partial observations. The difference between the multilevel method and diagonal preconditioning is greater when we change the observation operator to partial observations (see Figure 5). For this case, we sought reconstruction at mesh levels $n = 32, 64, 128$ for $\beta = \sigma_{0.05}$ and $\sigma_{0.1}$. The termination criterion for PCG was the same as

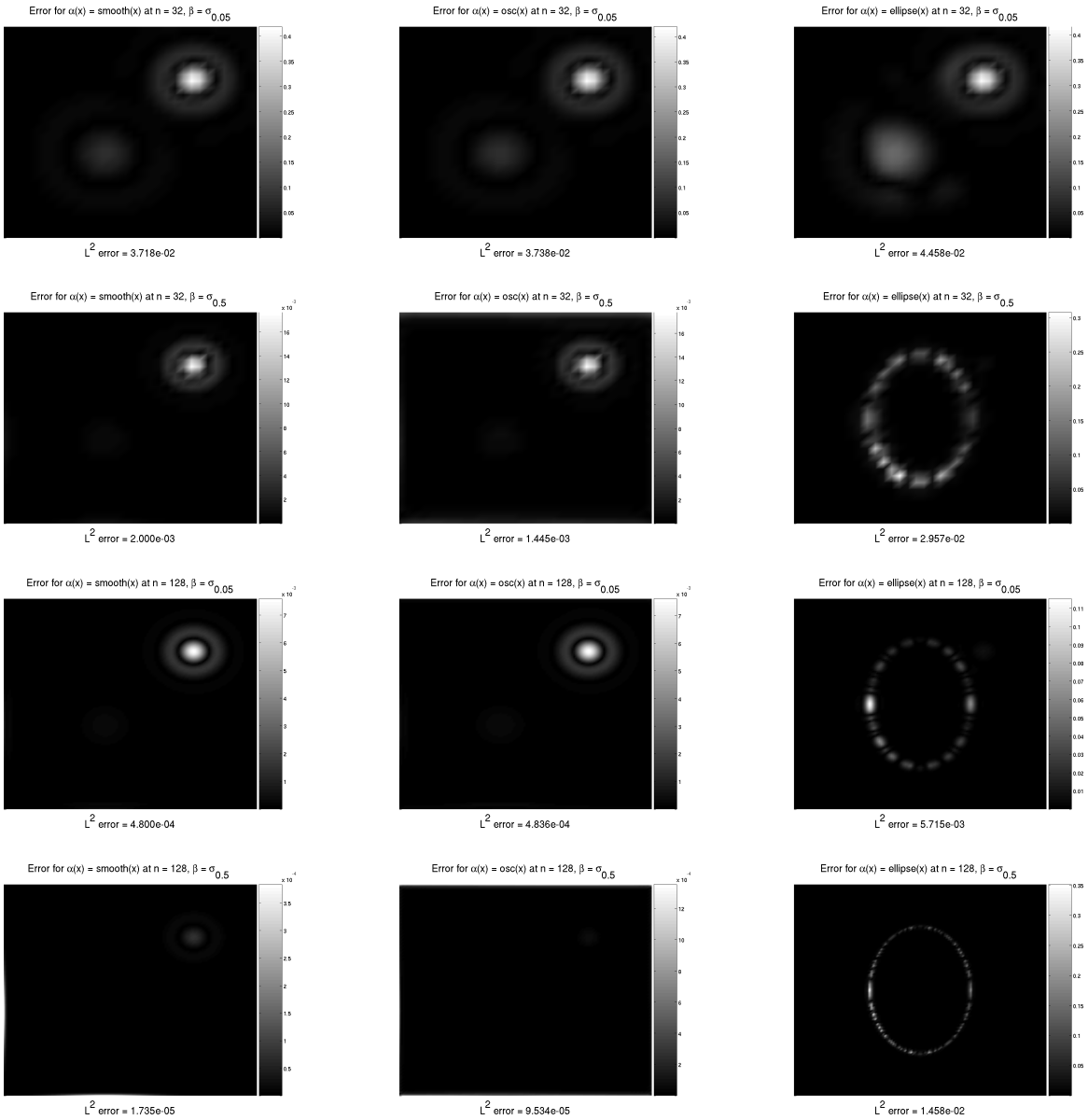

Figure 4. Reconstructions with full domain observations. These plots show the error between the true source and the one computed numerically for various choices of parameters and variables. We examine the cases with full domain observations when the coefficient $\alpha(x)$ is equal to one of smooth(x), osc(x), ellipse(x) (the left, middle and right columns respectively). We compute the reconstructions at mesh sizes n=32 (first two rows) and n=128 (last two rows). We set the regularization parameter $\beta = \sigma_{0.05}$ (first and third rows) and $\beta = \sigma_{0.5}$ (second and last rows). In this set of experiments, where we have high quality observations with no noise, we can see that the combination of a high resolution grid and a small regularization parameter enables us to obtain high quality reconstructions, such as the ones in the last the last row.

PCG without preconditioning

	$\alpha(x)=1$			$\alpha(x)=\text{smooth}(x)$			$\alpha(x)=\text{osc}(x)$			$\alpha(x)=\text{ellipse}(x)$		
n	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$
32	49	135	179	75	226	330	68	227	332	75	300	499
64	111	220	291	196	527	760	191	568	846	246	945	$1e{-10}$
128	211	296	294	467	981	$2e{-12}$	463	$2e{-12}$	$4e{-12}$	833	$2e{-9}$	$5e{-9}$
256	292	340	309	975	$1e{-12}$	$3e{-12}$	960	$2e{-12}$	$2e{-12}$	$6e{-10}$	$2e{-10}$	$4e{-9}$

PCG preconditioned with diagonal preconditioner

	$\alpha(x)=1$			$\alpha(x)=\text{smooth}(x)$			$\alpha(x)=\text{osc}(x)$			$\alpha(x)=\text{ellipse}(x)$		
n	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$
32	1	1	1	12	12	12	10	11	11	29	29	29
64	1	1	1	12	12	12	11	11	11	29	29	29
128	1	1	1	12	12	12	11	11	11	29	29	29
256	1	1	1	12	12	12	11	11	11	29	29	29

PCG preconditioned with the two-level scheme

	$\alpha(x)=1$			$\alpha(x)=\text{smooth}(x)$			$\alpha(x)=\text{osc}(x)$			$\alpha(x)=\text{ellipse}(x)$		
n	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$
32	1	1	1	2	2	3	5	6	6	10	13	15
64	1	1	1	1	1	1	5	5	5	10	12	13
128	1	1	1	1	1	1	4	4	5	11	11	12
256	1	1	1	1	1	1	4	4	4	8	10	10

PCG preconditioned with the multilevel scheme

	$\alpha(x)=1$			$\alpha(x)=\text{smooth}(x)$			$\alpha(x)=\text{osc}(x)$			$\alpha(x)=\text{ellipse}(x)$		
n	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$	$\sigma_{0.05}$	$\sigma_{0.25}$	$\sigma_{0.5}$
32	1	1	1	2	2	3	5	6	6	10	13	15
64	1	1	1	2	3	3	6	6	6	14	16	17
128	1	1	1	3	3	3	6	6	6	16	16	16
256	1	1	1	3	3	3	6	6	6	16	16	16

Table 2. Convergence of PCG preconditioned with the multilevel scheme (full observations). Here we document the superior performance of the multilevel algorithm. We test the method with a sample reconstruction problem, where we try to recover the source given in Figure 3. We perform the experiments for the constant coefficient case $\alpha(x) = 1$ and the variable coefficient case with $\alpha(x)$ equal to one of smooth(x),osc(x), ellipse(x). We repeat at mesh levels $n = 32, 64, 128, 256$ for regularization parameter $\beta = \sigma_{0.05}, \sigma_{0.25}, \sigma_{0.5}$. The maximum number of iterations is set to 1000. If this is exceeded, we terminate and report final relative residual for the experiment. We compare the multilevel method with unpreconditioned PCG and PCG preconditioned with diagonal preconditioning. Preconditioned PCG performs much better than unpreconditioned PCG and interestingly it yields mesh-independent and β-independent performance with diagonal preconditioning. However PCG preconditioned with the multilevel scheme gives considerably better results than PCG with diagonal preconditioning.

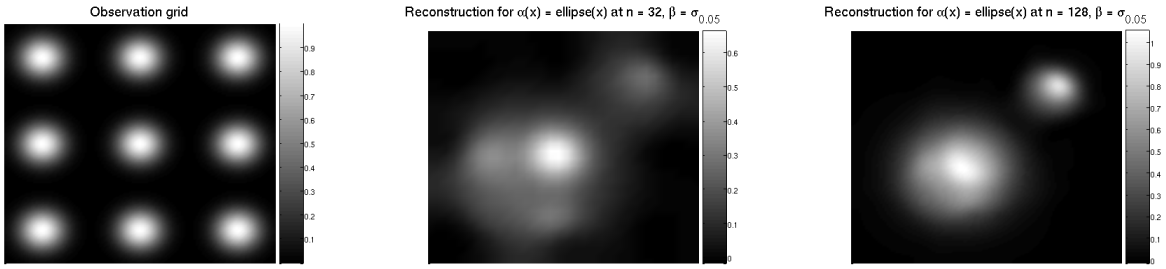

Figure 5. Reconstruction with partial domain observations. The reconstructions obtained from partial domain observations are given. The 3×3 grid of Gaussians showing the spatial distribution of the observations is given on the left. The reconstructions were obtained for $\alpha(x) = \text{ellipse}(x)$ using PCG preconditioned by the multilevel scheme with Galerkin coarse-grid operator (28). The regularization parameter β was $\sigma_{0.05}$. The middle plot shows the reconstruction for $n = 32$. The plot on the right shows the reconstruction for $n = 128$.

the previous set of experiments with full domain observations. We compared PCG with no preconditioning, PCG with diagonal preconditioning and PCG preconditioned with the multilevel preconditioner, where the coarse-grid operator was the Galerkin operator (28) or the non-Galerkin operator (29). We observed a deterioration in performance of all algorithms (see Table 3). Nevertheless we see that the multilevel algorithm still results in considerably fewer number of iterations compared to the case with no preconditioning or diagonal preconditioning, but the mesh-independence property in the convergence of the multilevel scheme is lost. In particular, it takes more iterations to converge, compared to the case with full domain observations. There are two reasons for this change in the performance of the multilevel algorithm: 1) as we change the observation operator to one different from identity, the preconditioner $\mathbf{B}$ defined by (27) is not as good an approximation to $\mathbf{H_o}$ as it is for $Q = I$; therefore, the smoother (26) is less effective, 2) additionally, the spectral decomposition of $\mathbf{H}$ is not accurate any more; the cross-terms (21) are not small, creating error not addressed by the multilevel preconditioner.

Noisy observations. Finally we performed a set of experiments checking the effect of noisy data in the reconstructions. We fixed the regularization parameter β and varied the amount of noise. We did not expect the performance of the multilevel algorithm to be effected by noise, as the noise will change the right hand side of the linear system, not the coefficient matrix. The results in Table 4 confirm our expectations. On the other hand, as we fixed β the quality of the reconstructions deteriorated with increasing noise.

6. Conclusions

We proposed an algorithm for solving problems with Tikhonov-regularized first-kind Fredholm operators. We considered the 2D Neumann Poisson problem on the unit box.

	PCG without preconditioning				PCG with diagonal preconditioning			
	$\alpha(x) = \text{smooth}(x)$		$\alpha(x) = \text{ellipse}(x)$		$\alpha(x) = \text{smooth}(x)$		$\alpha(x) = \text{ellipse}(x)$	
n	$\sigma_{0.05}$	$\sigma_{0.1}$	$\sigma_{0.05}$	$\sigma_{0.1}$	$\sigma_{0.05}$	$\sigma_{0.1}$	$\sigma_{0.05}$	$\sigma_{0.1}$
32	40	62	40	69	40	57	44	63
64	104	169	109	180	77	103	91	128
128	306	$2e{-9}$	325	$4e{-12}$	137	175	171	231

	PCG with Gal. multilevel scheme				PCG with non-Gal. multilevel scheme			
	$\alpha(x) = \text{smooth}(x)$		$\alpha(x) = \text{ellipse}(x)$		$\alpha(x) = \text{smooth}(x)$		$\alpha(x) = \text{ellipse}(x)$	
n	$\sigma_{0.05}$	$\sigma_{0.1}$	$\sigma_{0.05}$	$\sigma_{0.1}$	$\sigma_{0.05}$	$\sigma_{0.1}$	$\sigma_{0.05}$	$\sigma_{0.1}$
32	6	8	7	9	6	9	10	12
64	11	19	14	24	15	30	19	37
128	31	$1e{-12}$	42	$4e{-10}$	$1e{-10}$	$2e{-7}$	$5e{-8}$	$2e{-7}$

Table 3. Convergence of PCG preconditioned with the multilevel scheme (partial observations). Here we list the iteration numbers resulting from the experiments with partial observations. The 3×3 observation grid consisting of nine Gaussians is given in Figure 5. We test our method with the same sample reconstruction problem as the one we used for the full domain observation case. We try to recover the source given in Figure 3. We perform the experiments for the variable coefficient case with $\alpha(x)$ equal to one of smooth(x), ellipse(x). We repeat at mesh levels $n = 32, 64, 128$ for regularization parameter $\beta = \sigma_{0.05}, \sigma_{0.1}$. The maximum number of iterations is set to 500 for PCG, 50 for PCG with the multilevel algorithm. If the maximum is exceeded, we terminate and report final relative residual for the experiment. We compare the multilevel algorithm with unpreconditioned PCG and PCG preconditioned with diagonal preconditioning. We see that diagonal preconditioning does not perform as well as in the case of full domain observations. The difference between no preconditioning and diagonal preconditioning is not significant. Compared to both case, the multilevel algorithm preconditioning does very well. We run two versions of the multilevel scheme with the Galerkin (28) and non-Galerkin (29) coarse-grid operators. Using H^G gives the best results at the cost of increased computation time.

The ideas here extend to Dirichlet, and periodic boundary conditions and in general to all problems in which B, the approximate Hessian, can be diagonalized by a sparse transform. Our method relies on the availability of a fast algorithm for the spectral decomposition of B, which is then used to define the nested multilevel subspace, and the preconditioner. We considered the case of L^2 regularization. Typical cases in which such preconditioners are regular geometries (box, cylindrical, spherical, half-spaces) and for full-boundary or full-domain observations. Our method can be easily extended to H^1 regularization, since the regularization operator has the same eigenvectors as H_0 and thus B is still diagonal. We observed mesh-independence for the full-observation case. Also we observed order-of-magnitude savings for the partial-observation case but mesh-dependent results.

The algorithm is insensitive on the coefficient jumps on the forward problem (works well for four orders of magnitude jumps), but much more sensitive on the observation operator.

Noisy reconstruction error

	$\alpha(x) = \text{smooth}(x)$				$\alpha(x) = \text{ellipse}(x)$			
n	$\gamma = 0$	$\gamma = 100$	$\gamma = 300$	$\gamma = 1000$	$\gamma = 0$	$\gamma = 100$	$\gamma = 300$	$\gamma = 1000$
32	3.72e-2	3.79e-2	4.56e-2	1.01e-1	4.46e-2	4.52e-2	5.31e-2	1.13e-1
64	6.25e-3	6.67e-3	9.27e-3	2.35e-2	9.27e-3	9.56e-3	1.15e-2	2.45e-2
128	4.82e-4	7.39e-4	1.74e-3	5.58e-3	5.71e-3	5.73e-3	5.92e-3	7.90e-3

Number of iterations

	$\alpha(x) = \text{smooth}(x)$				$\alpha(x) = \text{ellipse}(x)$			
n	$\gamma = 0$	$\gamma = 100$	$\gamma = 300$	$\gamma = 1000$	$\gamma = 0$	$\gamma = 100$	$\gamma = 300$	$\gamma = 1000$
32	2	2	2	3	11	11	11	11
64	2	2	2	3	14	14	14	14
128	3	3	3	3	16	16	16	16

Table 4. Effect of noise on the performance. We examine the effect of noisy data on the performance of algorithm. Given full domain observations, we try to reconstruct the source using PCG preconditioned with the multilevel scheme. We fix the regularization parameter $\beta = \sigma_{0.05}$ and add random noise of magnitude $\gamma\sigma_{0.05}\|y\|_\infty$ to the observations. We vary $\gamma = 0, 100, 300, 1000$ and examine the impact of increasing noise. We perform the tests for variable coefficients $\alpha(x) = \text{smooth}(x), \text{ellipse}(x)$ and mesh sizes $n = 32, 64, 128$. We report the number of iterations and the L^2 error in the reconstructions. We see that noise in data does not change the number of iterations that the multilevel scheme takes to converge. This is natural, because the impact of noise is on the right hand side of the linear system, not on the coefficient matrix of the system (which determines the convergence behavior). On the other hand, increasing the amount of noise in the data deteriorates the reconstructions for fixed β as expected and the L^2 error between the reconstructed solution and true solution increases.

This suggests the need for additional improvements in the preconditioner. There are several options: changing the additive preconditioner to a multiplicative one, adding more smoothing steps, changing the preconditioner to W-cycle multigrid, replacing the Richardson scheme with the Frankel scheme [1], and precomputing approximations to the spectrum of the reduced Hessian with partial observations. A more interesting case is that in which the forward problem is given by $Jy + Cu = 0$; for our source identification problem, this corresponds to having a source term of the form $c(x)u(x)$ where $c(x)$ is a known function. This problem is related to linearizations (Born approximations or Newton steps) of the inverse medium problem. For this problem the unregularized reduced Hessian takes the form $C^T H_0 C$. We are currently working on extending our solver to this case; we will report results elsewhere. Finally, let us note that in the case of nonlinear reconstruction formulations, for example using bounded variation regularization, the situation is quite different as the regularization operator is not diagonalizable and our preconditioner will become less effective.

References

[1] S. S. Adavani and G. Biros. Multigrid algorithms for inverse problems with linear parabolic pde constraints. In review, 2007.

[2] V. Akcelik, G. Biros, A. Draganescu, H. Hill, O. Ghattas, and B. V. Van Bloemen Waanders. Dynamic Data-Driven Inversion for Terascale Simulations:Real-Time Identification of Airborne Contaminants, 2005.

[3] E. Arian and S. Ta'asan. Multigrid one shot methods for optimal control problems: infinite dimensional control. Technical Report ICASE 94-52, ICASE, NASA Langley Research Center, July 1994.

[4] O. Axelsson. *Iterative solution methods*. Cambridge University Press, New York, NY, USA, 1994.

[5] H. T. Banks and K. Kunisch. *Estimation Techniques for Distributed Parameter Systems*. Birkhauser, 1989.

[6] A. Borzí. High-order discretization and multigrid solution of elliptic nonlinear constrained optimal control problems. *Journal Of Computational And Applied Mathematics*, 200(1):67–85, 2007.

[7] A. E. Borzì. Multigrid methods for optimality systems. Technical report, University of Graz, Austria, 2003. Habilitation Thesis.

[8] A. Brandt. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation*, 31(138):333–390, 1977.

[9] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2000.

[10] D. Colton and R. Kress. *Inverse Acoustic and Electromagnetic Scattering Theory, 2nd Edition*. Applied Mathematical Sciences. Springer, 1998.

[11] T. Dreyer, B. Maar, and V. Schulz. Multigrid optimization and applications. *Journal of Computational and Applied Mathematics*, 120:67–84, 2000.

[12] W. Hackbusch. *Multigrid methods and applications*, volume 4 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 1985.

[13] M. Hanke and C. R. Vogel. Two-level preconditioners for regularized inverse problems. I. Theory. *Numerische Mathematik*, 83(3):385–402, 1999.

[14] W. Heinrichs. Line relaxation for spectral multigrid methods. *J. Comput. Phys.*, 77(1):166–182, 1988.

[15] W. Heinrichs. A 3D spectral multigrid method. *Appl. Math. Comput.*, 41(2, part II):117–128, 1991.

[16] V. Isakov. *Inverse Problems for Partial Differential Equations*. Springer, 1998.

[17] B. Kaltenbacher. V-cycle convergence of some multigrid methods for ill-posed problems. *Mathematics of Computation*, 72(244):1711–1730, 2003.

[18] J. T. King. On the construction of preconditioners by subspace decomposition. *Journal of Computational and Applied mathematics*, 29:195–205, 1990.

[19] J. T. King. Multilevel algorithms for ill-posed problems. *Numerische Mathematik*, 61(3):311–334, 1992.

[20] U. Trottenberg, C. W. Oosterlee, and A. Schuller. *Multigrid*. Academic Press Inc., San Diego, CA, 2001.

[21] T. A. Zang, Y. S. Wong, and M. Y. Hussaini. Spectral multigrid methods for elliptic equations. *J. Comput. Phys.*, 48(3):485–501, 1982.

[22] T. A. Zang, Y. S. Wong, and M. Y. Hussaini. Spectral multigrid methods for elliptic equations. II. *J. Comput. Phys.*, 54(3):489–507, 1984.