MULTIGRID ALGORITHMS FOR INVERSE PROBLEMS WITH LINEAR PARABOLIC PDE CONSTRAINTS *

SANTI S. ADAVANI † AND GEORGE BIROS‡

Abstract. We present a multigrid algorithm for the solution of distributed parameter inverse problems constrained by variable-coefficient linear parabolic partial differential equations. We consider problems in which the inversion variable is a function of space only; for stability we use an L^2 Tikhonov regularization. The main feature of our algorithm is that its convergence rate is mesh-independent—even in the case of no regularization. This feature makes the method algorithmically robust to the value of the regularization parameter, and thus, useful for the cases in which we seek a high-fidelity reconstruction.

The problem is formulated as a PDE-constrained optimization. We use a reduced space approach. We eliminate the state and adjoint variables and we iterate in the inversion parameter space using Conjugate Gradients. We precondition with a V-cycle multigrid scheme. The multigrid smoother is a two-step stationary iterative solver that inexactly inverts an approximate Hessian by iterating exclusively in the high-frequency subspace (using a high-pass filter). We analyze the performance of the scheme for the constant coefficient case with full observations; we analytically calculate the spectrum of the reduced Hessian and the smoothing factor for the multigrid scheme. The forward and adjoint problems are discretized using a backward-Euler finite-difference scheme. The overall complexity of our inversion algorithm is $\mathcal{O}(N_t N + N \log^2 N)$, where N is number of grid points in space and N_t is the number of time steps.

We provide numerical experiments that demonstrate the effectiveness of the method for different diffusion coefficients and values of the regularization parameter. We also provide heuristics, and conduct numerical experiments for the case with variable coefficients, and partial observations. We observe the same complexity as in the constant-coefficient case. Finally, to avoid exact forward and adjoint solves far from the minimum, we combine the reduced-space algorithm with a full-space method.

Key words. Inverse problems, heat equation, reaction-diffusion equations, multigrid, regularization.

AMS subject classifications.

1. Introduction.

In this paper we present multigrid algorithms for inverse problems constrained by parabolic partial differential equations (PDEs). As a model problem we consider the one-dimensional heat equation. The inversion parameter is a heat source, which we try to reconstruct given full or partial observations of the temperature. Our method is designed for problems in which the temporal variation of the heat source is known, but the spatial variation is not. Our model is motivated by inverse medium and data assimilation problems that are constrained by reaction-convection-diffusion equations. We use an an infinite-dimensional PDE-constrained optimization formulation [5]. Although we consider only the 1D case, our algorithmic choices are designed for large-scale three-dimensional problems.

More precisely, we seek to reconstruct an unknown function $u(x)$ by solving the following minimization problem:

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

subject to:

$$\frac{\partial y(x)}{\partial t} - \nu \Delta y(x) = a(x,t)y(x) + b(x,t)u(x) \quad \text{in} \quad D,\ y = 0 \quad \text{on} \quad \partial\Omega,\ y(x,0) = 0 \quad \text{in} \quad \Omega,$$

*This work was supported by the U.S. Department of Energy under grant DE-FG02-04ER25646, and the U.S. National Science Foundation grants CCF-0427985, CNS-0540372, IIS-0431070, and DMS-0612578.

†Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, PA 19104, USA (adavani@seas.upenn.edu)

‡Departments of Mechanical Engineering and Applied Mechanics, Bioengineering, and Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104, USA (biros@seas.upenn.edu)

where D is defined as $\Omega \times (0,T]$. Here, y is the *state variable*, u is the *inversion variable*, $\nu > 0$ is the diffusion coefficient, and $\beta \geq 0$ is the regularization parameter. The objective is to reconstruct u by minimizing the misfit between the observed state y^* and the predicted state y. We assume that both $a(x,t)$ and $b(x,t)$ are known, smooth, and bounded functions.[1]

By forming a Lagrangian, introducing the adjoint variables λ, and by requiring stationarity with respect to the state, inversion, and adjoint variables, we arrive at the the first-order optimality conditions:
State

$$\frac{\partial y}{\partial t} - \nu\Delta y - ay - bu(x) = 0 \quad \text{in} \quad D, \quad y = 0 \quad \text{on} \quad \partial\Omega, \quad y(x,0) = 0 \quad \text{in} \quad \Omega.$$

Adjoint

$$-\frac{\partial \lambda}{\partial t} - \nu\Delta\lambda - a\lambda + (y - y^*) = 0 \quad \text{in} \quad D, \quad \lambda = 0 \quad \text{on} \quad \partial\Omega, \quad \lambda(x,T) = 0 \quad \text{in} \quad \Omega.$$

Inversion

$$\beta u - \int_T b\lambda \, d\, t = 0 \quad \text{in} \quad \Omega.$$

The above system of equations is also known as the *Karush-Kuhn-Tucker* optimality system or the *"KKT"* system. The corresponding linear operator can be written as

$$\begin{bmatrix} I & 0 & -\frac{\partial}{\partial t} - \nu\Delta - a \\ 0 & \beta I & -\int_0^T b \\ \frac{\partial}{\partial t} - \nu\Delta - a & -b & 0 \end{bmatrix} = \begin{bmatrix} Q & 0 & J^T \\ 0 & \beta I & C^T \\ J & C & 0 \end{bmatrix}. \tag{1.1}$$

The KKT operator corresponds to a symmetric saddle point problem. For an excellent review on linear solvers for such problems, we refer the reader to [6]. In this paper we will consider two methods, the so-called *"full space"* and *"reduced space"* [15]. In full space methods one solves directly (1.1), for example, using a Krylov iterative method. In reduced space methods one solves for u by an iterative solver on the the Schur complement of u. To derive the Schur complement, we first eliminate y and λ using the state and adjoint equations respectively, and then we substitute λ in the inversion equation. In this way we obtain

$$Hu = g. \tag{1.2}$$

The *"reduced Hessian"* H (or just *"Hessian"*) is defined by $H = C^T J^{-T} Q J^{-1} C + \beta I$. Since Q is positive semi-definite, H is a symmetric and strictly positive definite operator. The reduced gradient g is defined by $g = -C^T J^{-T} Q y^*$. We focus our attention to the design of efficient solvers for reduced space formulations. For completeness we include an example that shows how we can combine full and reduced space approaches.

Related work. Reduced space methods are quite popular because one can iterate on the adjoint and state equation in sequence, they require less storage, and the Conjugate Gradients method (CG) can be used to invert H. The KKT matrix (1.1), is indefinite, ill-conditioned, and its size is more than twice as large as that of the forward problem. Most implementations avoid using H and instead use some approximation, for example, quasi-Newton. Such approaches however, are not algorithmically scalable [1]. If H is to be used, direct solvers are not a viable option since the reduced Hessian is a non-local and thus, dense operator. The

[1] In the following we suppress the notation for the explicit dependence on x and t.

Preconditioned Conjugate Gradients (PCG) algorithm requires matrix-vector product (hereinafter, "*matvec*") operations only, and thus, can be used to solve (1.2).

If we fix the regularization parameter β to a positive value we can show that H is a compact perturbation of the identity and thus, has a bounded (mesh-independent) condition number: it scales as $\mathcal{O}(1/\beta)$. Using CG to solve a linear system involving H requires $\mathcal{O}(1/\sqrt{\beta})$ iterations. Therefore, the overall scheme does not scale with vanishing β. We claim that in mesh refinement studies and scalability analyses for inverse problem solvers, having a fixed value of β can lead to wrong conclusions.

There are two reasons that drive the need to solve problems in refined meshes. The first reason is the need to resolve the forward and adjoint equations. In that case one can use a mixed discretization in which u is discretized in a coarser grid, or one can use a large value for β. In the second case, which is pertinent to scalability of the inverse problem solver, we have high-quality observations[2] that allow for a high-resolution reconstruction of u. This implies that β cannot be fixed as we refine the mesh because we will not be able to recover the sought frequencies. Obtaining a mesh-independent scheme for vanishing β, to our knowledge, has not been addressed.

Returning to the reduced Hessian, we observe that the deterioration of the condition number with decreasing β suggests the need for a preconditioning scheme. We cannot use standard preconditioning techniques like incomplete factorizations or Jacobi relaxations, as these methods need an assembled matrix [4]. In [7] a two-step stationary iterative method that does not need an assembled matrix was used to precondition the reduced Hessian. (The two-step method will be the smoother in our scheme.)

Another alternative is to use multigrid methods. These methods have been developed mainly for linear systems arising from the discretization of elliptic and parabolic PDEs. The basic idea of multigrid is to accelerate the iterative solution of a PDE by computing corrections on a coarser grid and then interpolating them back to the original grid. The three important steps of multigrid scheme are *pre-smoothing, coarse-grid correction* and *post-smoothing*. Smoothing is equivalent to taking a few iterations of an iterative method (*"smoother"*) that should selectively remove the high-frequency error components faster than low-frequency components. Besides the pioneering work of [12] for differential operators, and of [19] for second-kind Fredholm integral operators, there exists significant work on multigrid methods for optimal control problems. For example see the work of [2] and [15] for a general discussion, and [9] and [10] for distributed control problems constrained by parabolic PDEs. An alternative to multigrid is domain decomposition; a promising work for problems similar to ours can be found in [21]. There the author proposes a space-time Schur domain decomposition preconditioner for the KKT system. A nice feature of that method is that it can be readily parallelized in time. The context however, is optimal control and not inverse problems: the value of the regularization parameter is quite large.

In our case, the unregularized reduced Hessian is a Fredholm operator of the first kind. There has been little work on multigrid algorithms for such problems. In [20] multilevel and domain decomposition preconditioners were proposed for integral equations of first-kind. Multigrid solvers for Tikhonov-regularized ill-posed problems were discussed in [27] and [26]. Such problems were further analyzed in [24] and [25]. A multigrid preconditioner based on that work was also used in [1] to solve problems with million of inversion parameters. All these methods however, require a relatively small but non-vanishing value of the regularization parameter. As we will discuss later in the paper, the methods described in [1] and [10] do not scale well in the case of a mesh-dependent regularization parameter.

[2] If the data is not in the range of the inversion operator, e.g., due to noise, vanishing data will result in blow up for u.

Contributions. Our main contribution is to derive a method for which we obtain a mesh-independent and β-independent convergence rate—including the case of $\beta = 0$. We design special smoothers that are used to built multigrid schemes for (1.2).

There are several challenges in designing multigrid schemes for the reduced space. As we mentioned, we have to design a matrix-free smoother with no access to diagonal or off-diagonal terms of the Hessian operator. The reduced Hessian (with $\beta = 0$) is a compact operator. Its dominant eigenvalues correspond to low-frequency components and for such operators standard smoothers fail. Finally, every matrix-vector multiplication with the reduced Hessian is equivalent to a forward and an adjoint solve; hence, it is important to design algorithms that require a minimum number of matvecs in the fine grid.

We first propose a multigrid solver that uses a CG smoother combined with an approximate filtering operator that restricts the CG iterations in the high-frequency Krylov subspace. We show numerical results that indicate good behavior. The method is easy to implement, but difficult to analyze. For this reason we propose a second smoother that is more expensive but for which we can provide complexity estimates.

The main components of the proposed algorithm are: (1) a reduced Hessian that is a composition of spectral filtering with an approximate Hessian operator based on inexact forward and adjoint solves; and (2) a smoothing scheme that uses a stationary second-order method targeted in the high-frequency components of u. It is crucial that the effectiveness of the smoother in the high-frequency regime is mesh independent; our method fulfills this requirement. The multigrid scheme (a V-cycle) can be used as solver or as a preconditioner for a Krylov iterative method.

The forward and adjoint problem are discretized using a backward-Euler scheme in time, and a standard three-point Laplacian (Dirichlet BCs) in space. We conduct numerical experiments to test (1) the effects of semi-coarsening (only space coarsening) and standard-coarsening; (2) different smoothing techniques; and (3) the effects of using non-Galerkin coarse-grid operators. We analyze and experimentally measure convergence factors. Also, we present results for the more general case in which multigrid is used as a preconditioner and not as a solver. In addition, we test the algorithm for the case of variable coefficients (resembling reaction-diffusion equations that admit traveling wave solutions) and partial observations. Finally, we include a discussion on full-space methods and we propose a multigrid scheme for (1.1) along with numerical results that illustrate its performance.

1.1. Organization of the paper. In Section 2 we derive the spectral properties of the analytic and discretized reduced Hessian for the case of constant coefficients. In Section 3 we discuss multigrid and we give details on the coarse-grid operator. In Section 3.2 we discuss standard smoothers and present construction of novel smoothers based on the idea of subspace decomposition; in Section 4, we present appropriate preconditioners so that PCG can be used as a smoother. Numerical results on this approach are presented in Section 4.1. In Section 5, we present our main contribution, a multigrid variant which based on exact subspace projections, and we present results for both the constant and variable coefficient case. Finally, in Section 6 we discuss full space methods.

2. Spectral properties of the reduced Hessian. We start by calculating the spectrum of the reduced Hessian. We show that in its general form, the source inversion is an ill-posed problem with algebraically decaying eigenvalues. Let K be the Green's operator for the forward problem, so that K maps functions from the inversion variable space to the state variable space. Using K, we can eliminate the constraint and obtain an unconstrained varia-

tional problem for u:

$$\min \tilde{\mathcal{J}}(u) = \frac{1}{2}\int_\Omega \int_T (Ku - y^*)^2 \, d\Omega dt + \frac{\beta}{2}\int_\Omega u^2 \, d\Omega. \tag{2.1}$$

Taking variations of (2.1) with respect to u we get an Euler-Lagrange equation for u:

$$\frac{\partial \tilde{\mathcal{J}}}{\partial u}\hat{u} = \int_\Omega \int_T K^T(Ku - y^*)\hat{u} \, d\Omega dt + \beta \int_\Omega u\hat{u} \, d\Omega = 0, \forall \hat{u}, \tag{2.2}$$

where K^T is the adjoint of K. Then the strong form of the optimality conditions is given by

$$\left(K^T K + \beta I\right) u = K^T y^* \qquad \text{or} \qquad Hu = K^T y^*, \tag{2.3}$$

where H is the reduced Hessian. If $a = 0$ then K (for homogeneous Dirichlet boundary conditions) is given by

$$\begin{aligned} y &= \int_\Omega \int_0^t k(x-y, t-\tau) b(x,t) u(x) \, dy d\tau \\ &= 2\sum_{k=1}^{\infty} \int_\Omega \int_0^t e^{-k^2\pi^2(t-\tau)} S_k(x) S_k(y) b(x,t) u(x) \, dy d\tau, \end{aligned} \tag{2.4}$$

where $S_k(x) = \sin(k\pi x)$. If we assume that $b = 1(t)$, expand $u(x) = \sum_{j=1}^{\infty} u_j S_j(x)$, and use orthogonality we get

$$y = \sum_{k=0}^{\infty} S_k(x) y_k(t), \quad \text{with} \quad y_k(t) = \int_0^t e^{-k^2\pi^2(t-\tau)} 1(\tau) u_k \, d\tau. \tag{2.5}$$

The adjoint operator K^T is given by

$$\lambda(x, T-t) = K^T z = \int_\Omega \int_0^t k(x-y, t-\tau) \, z(y, T-\tau) \, dy d\tau. \tag{2.6}$$

Using (2.5) and (2.6) in (2.3), and setting $\beta = 0$, the *eigenvalues* (σ_k) and *eigenvectors* (v_k) of the reduced Hessian ($H = K^T K$) are given by

$$\sigma_k = \frac{2k^2\pi^2 T + 4e^{-k^2\pi^2 T} - 2e^{-2k^2\pi^2 T} - 3}{2k^6\pi^6} \quad \text{and} \quad v_k = S_k. \tag{2.7}$$

If we discretize in space using the three-point Laplacian approximation, the corresponding eigenvalues and eigenvectors of the reduced Hessian (H^h) are given by

$$\sigma_k = \frac{2\lambda_k T + 4e^{-\lambda_k T} - 2e^{-2\lambda_k T} - 3}{2\lambda_k^3} \quad \text{and} \quad v_k = S_k^h, \tag{2.8}$$

where $\lambda_k = 4\nu N^2 \sin^2(\frac{k\pi}{2N})$ is the k^{th} eigenvalue of the discrete Laplacian and ν is the diffusion coefficient. The discrete sine function is represented by S_k^h with N being discretization size. If we use a backward Euler scheme for time the eigenvalues of the discrete reduced Hessian (H^h) are given by

$$\sigma_k^\delta = \delta^3 \sum_{j=1}^{N_t} \sum_{l=0}^{N_t - j} \frac{\sum_{r=0}^{l+j-1} \frac{1}{(1+\lambda_k\delta)^r}}{(1+\lambda_k\delta)^{l+1}}, \tag{2.9}$$

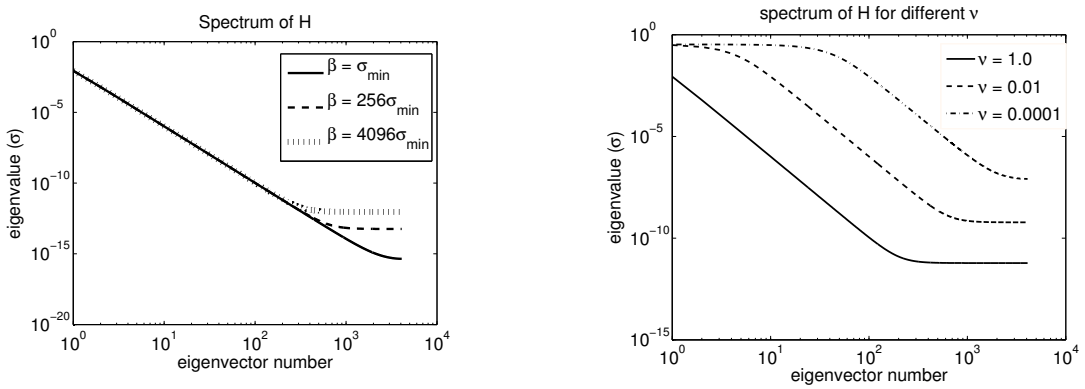

FIG. 2.1. ***Effect of regularization on the spectrum of the reduced Hessian.*** *Here we report the Spectrum of eigenvalues (σ) of the reduced Hessian H for $\nu = 1.0$ and $T = 1$. Three cases of regularization parameter $\beta = \sigma_{min}, 256\sigma_{min}, 4096\sigma_{min}$ are plotted. The right plot shows the spectrum of H for three different diffusion coefficients $\nu = 1, 0.01, 0.0001$.*

TABLE 2.1

Eigenvalues, eigenvectors of Laplacian and reduced Hessian. *The k^{th} eigenvalue and eigenvector of the operator are represented by σ_k and v_k respectively. The diffusion coefficient and total time interval are ν and T respectively. Here $S_k = \sin(k\pi x)$. Discrete operators and functions are denoted with a superscript h.*

Operator	σ_k	v_k
$-\nu\Delta$	$\lambda_k = \nu k^2\pi^2$	S_k
$-\nu\Delta^h$	$\lambda_k^h = 4\nu N^2 \sin(\frac{k\pi}{2N})$	S_k^h
H	$\sigma_k = \frac{2\lambda_k T + 4e^{-\lambda_k T} - 2e^{-2\lambda_k T} - 3}{2\lambda_k^3}$	S_k
H^h	$\sigma_k = \frac{2\lambda_k^h T + 4e^{-\lambda_k^h T} - 2e^{-2\lambda_k^h T} - 3}{2(\lambda_k^h)^3}$	S_k^h
$H^{h,\delta}$	$\sigma_k^\delta = \delta^3 \sum_{j=1}^{N_t} \sum_{l=0}^{N_t - j} \frac{\sum_{r=0}^{l+j-1} \frac{1}{(1+\lambda_k\delta)^r}}{(1+\lambda_k\delta)^{l+1}}$	S_k^h

where N_t is the number of time steps and δ the time step.

From (2.7) and (2.8) it is evident that $k \to \infty \Rightarrow \sigma_k \to 0$. Furthermore $\sigma_{max} \approx \frac{T}{(\lambda_k)^2}$, for a large-enough time-horizon T. If $\beta \neq 0$ then $\sigma_{min} = \beta$ and the *condition number* of the reduced Hessian is given by $\kappa = \frac{\sigma_{max}+\beta}{\beta}$ and it is bounded. For small β, however, the reduced Hessian is a highly ill-conditioned operator (see Figure 2.1).

TABLE 2.2

CG mesh-dependence. *Here we report the performance of CG as a function of the mesh size and the value of the regularization parameter. The number of CG iterations does not change with an increase in the problem size N_s; β is the regularization parameter and in parentheses the number of recovered frequencies;* **iters** *corresponds to number of CG iterations for a relative residual reduction $\|r\|/\|r^0\| \leq 10^{-12}$; and maximum number of iterations is $2N_s$. Two cases of regularization parameter are considered: $\beta = \sigma_{20}$ and $\beta = \sigma_{100}$. Additional parameters used in this numerical experiment are $\nu = 1$, $T = 1$. One observes that the number of CG iterations are mesh-independent only in the case of constant β.*

N_s	β ($\sigma > \beta$)		**iters**	
512	6e-08 (19)	1e-10 (99)	69	725
1024	6e-08 (19)	1e-10 (99)	70	781
2048	6e-08 (19)	1e-10 (99)	68	763
4096	6e-08 (19)	1e-10 (99)	71	713

The number of CG iterations required for convergence is proportional to the square root of the condition number of the underlying operator. Therefore, for mesh-independent condition number, we obtain a mesh-independent number of iterations. It may be the case however, that the data fidelity allows quite small regularization parameter.

In Table 2.2 we report results from a numerical experiment in which we study the number of CG iterations for two cases of the regularization parameter. One can observe that for constant β the the number of iterations is mesh-independent. This is not the case when β is related to the mesh size. The goal of the present work is to use multigrid ideas to address problem of β-independence number of CG iterations, at least for the source inversion problem for the heat equation.

3. Reduced space multigrid. In this section we summarize the algorithmic issues related to multigrid for the reduced Hessian. Here, and in the rest of the paper, we use the superscript h to denote the fine discretization level, and $2h$ the coarse level—in the case of a two-grid scheme. For example, we denote the discrete reduced Hessian at resolution h by H^h. The key steps of a multigrid algorithm for $H^h u^h = g^h$ are given in Algorithm 1. The re-

Algorithm 1 Multigrid (MG)

1: *Pre-smoothing:* Smoother iterations on $H^h u^h = g^h$
2: *Restriction:* $r^h = g^h - H^h u^h$ and $r^{2h} = I_h^{2h} r^h$
3: *Coarse-grid correction:* Solve $H^{2h} e^{2h} = r^{2h}$
4: *Prolongation:* $e^h = I_{2h}^h e^{2h}$
5: *Update:* $u^h \leftarrow u^h + e^h$
6: *Post-smoothing* Smoother iterations on $H^h u^h = g^h$

striction (I_h^{2h}), prolongation (I_{2h}^h) and coarse grid (H^{2h}) operators are important components that determine the performance of the algorithm. In Table 3.1 we summarize the spectra of several restriction and prolongation based operators. Key in a multigrid scheme is to that each

TABLE 3.1

Spectral properties of the restriction and prolongation operators. *Let* $s_k = \sin^2(\frac{k\pi x}{2})$, $c_k = \cos^2(\frac{k\pi x}{2})$ *and* $S_k = \sin(k\pi x)$ *for* $1 \le k \le N-1$. $I_h^{2h}\colon \Omega^h \to \Omega^{2h}$, $I_{2h}^h\colon \Omega^{2h} \to \Omega^h$ *and* $I - I_{2h}^h I_h^{2h}\colon V^h \to W^{2h}$ *where* V^h *is the fine space and* W^{2h} *is the space containing high frequency components.* S_k^h, S_k^{2h} *are the eigenfunctions of the discrete Laplacian and the reduced Hessian in* Ω^h *and* Ω^{2h} *respectively.* $1 \le k \le \frac{N}{2}$ *for all the rows in the Table.*

Operator	Input function	Output function
I_h^{2h}	S_k^h	$c_k S_k^{2h}$
I_h^{2h}	S_{N-k}^h	$-s_k S_k^{2h}$
I_{2h}^h	S_k^{2h}	$c_k S_k^h - s_k S_{N-k}^h$
$I - I_{2h}^h I_h^{2h}$	S_k^h	$(1-c_k^2) S_k^h + c_k s_k S_{N-k}^h$
$I - I_{2h}^h I_h^{2h}$	S_{N-k}^h	$(1-s_k^2) S_{N-k}^h + c_k s_k S_k^h$

grid level the majority of the work is in removing errors associated with high-frequencies (at the specific grid level). In addition as we move into the grid hierarchy, errors should not be reintroduced or amplified. The problems we are discussing here are pretty regular so prolongation and restriction do not present particular challenges. Below we first discuss the coarse-grid operator representation and then we discuss smoothing techniques.

3.1. Coarse-grid operator. There are two main ways to define the coarse-grid operator, given a grid-hierarchy, the Galerkin and the direct discretization. Using a variational principle

and provided that $I_h^{2h} = cI_{2h}^{h\,T}$, the *"Galerkin"* coarse-grid operator operator is defined by

$$H_G^{2h} = I_h^{2h} H^h I_{2h}^h,$$

where H_G^{2h} and H^h are the Galerkin coarse-grid operator and fine grid operators respectively [13]. Another way of defining the coarse-grid operator is by discretizing directly the forward and adjoint problems in the coarse grid:

$$H^{2h} = (C^T J^{-T})^{2h} Q (J^{-1} C)^{2h}.$$

In the classical multigrid theory for the Laplacian operator on regular grids with constant-coefficients there is no difference between the two coarse-grid operators. In the case of reduced Hessian, however, they are quite different—especially in the high-frequency region of the coarse space (Figure 3.1). The difference in the spectra can be explained from the scaling of the eigenvalues of H^h due to the eigenstructure of the standard restriction and prolongation operators (Table 3.1). Therefore, error components in certain intermediate eigenvector directions of the fine grid spectrum cannot be recovered if we use H^{2h}. So it is preferable to use the Galerkin coarse-grid operator for robustness and easily provable convergence. On the other hand every Galerkin coarse-grid matvec requires a fine-grid reduced Hessian matvec which makes it computationally expensive. Therefore, we avoid using H_G^{2h} and use H^{2h}.

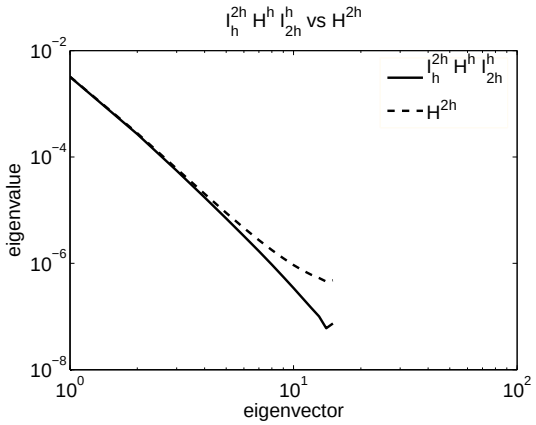

FIG. 3.1. ***Spectrum of the coarse-grid reduced Hessian.*** *Here we depict the difference between the Galerkin $I_h^{2h} H^h I_{2h}^h$ and the direct discretization of the reduced Hessian operators. We observe that H^{2h} does not satisfy the Galerkin condition, and thus, inverting it will not eliminate the low frequency components of the error. Due to this fact we use multigrid as a CG preconditioner.*

3.2. Smoothers. Classical smoothing schemes for the elliptic PDEs include iterative methods like Jacobi, Gauss-Seidel and CG. A common characteristic of these methods is that they remove error components corresponding to large eigenvalues faster than error components corresponding to small eigenvalues.[3] This property makes these methods favorable for elliptic operators that have large eigenvalues for high frequency eigenvectors. In our case, the (unregularized) reduced Hessian is a compact operator and behaves quite differently. As shown in Figure 2.1, large eigenvalues of the reduced Hessian are associated with smooth eigenfunctions and small eigenvalues are associated with rough or oscillatory eigenfunctions.

[3]CG works on both ends of the spectrum, but this is not so important in our context.

Therefore, the above smoothing methods act as *roughers*. In addition, we do not have direct access to the entries of the reduced Hessian matrix so there is no cheap way to apply smoothers like Jacobi or Gauss-Seidel.

We discuss in a greater detail why CG cannot be used as a smoother. Using (2.7) we will show that CG cannot be used as a smoother for this problem as it acts on the high energy (large eigenvalues) smooth components and acts as rougher instead. We neglect the exponential terms as they go to zero very fast. Let $\beta = 0$ and $T = \pi^4$, then the k^{th} eigenvalue of the reduced Hessian is $\sigma_k = \frac{1}{k^4}$. At the i^{th} iteration, CG constructs an $(i-1)^{\text{th}}$ degree polynomial to minimize $\|e_{(i)}\|_H$. Therefore the error at the i^{th} iteration can be expressed as $e_{(i)} = P_i(\sigma)e_{(0)}$, where $P_i(\sigma)$ is the $(i-1)^{\text{th}}$ degree polynomial, $e_{(0)}$ is the initial error and $e_{(i)}$ is the error at i^{th} iteration. $P_i(\sigma)$ is given by *Chebyshev polynomials*, where the Chebyshev polynomial $T_i(\omega)$ of degree i is $T_i(\omega) = \frac{1}{2}\left[(\omega + \sqrt{\omega^2-1})^i + (\omega - \sqrt{\omega^2-1})^i\right]$, [28]. The polynomial $P_i(\sigma)$ is given by

$$P_i(\sigma) = \frac{T_i\left(\frac{\sigma_{max}+\sigma_{min}-2\sigma}{\sigma_{max}-\sigma_{min}}\right)}{T_i\left(\frac{\sigma_{max}+\sigma_{min}}{\sigma_{max}-\sigma_{min}}\right)}.$$

We have already seen that the reduced Hessian is a compact operator. Thus, neglecting σ_{min} ($\sigma_{min} \ll \sigma_{max}$) gives $P_i(\sigma_k) = T_i\left(1 - \frac{2}{k^4}\right)$. Without loss of generality, we can assume that the initial guess has error components in all the eigenvector directions. Notice that $P_i(\sigma_k)$ is the amount of attenuation of the k^{th} eigenvector at the i^{th} CG iteration. For high-frequency error components $P_i(\sigma_k) \approx T_i(1) = 1$; for small $k \approx P_i(\sigma_k) = 0.5$. Thus, amplitude reduction of low frequency error components is greater than that of high frequency error components: CG can not be used as a smoother in the present problem.

This motivates a modification of the Hessian operator so that the low-frequency spectrum is screened out from CG. In this regard, we discuss construction of smoothers based on the idea of decomposing the finite-dimensional space into relatively high-frequency and low-frequency subspaces given in [27], [26]. This idea was further studied in [24], [25]. Similar ideas of using the subspace decomposition are also used in the construction of efficient preconditioners in [18].

Our preconditioners will be based on a fine-coarse grid decomposition of the reduced Hessian. The "coarse" space V^{2h} is embedded into the "fine" space V^h. By $P^h : V^h \to V^{2h}$ we denote the L^2-orthogonal projection, and by $I - P^h : V^h \to W^{2h}$ the projection to high frequency functions W^{2h}. We decompose $v \in V^h$ into a smooth component $v_s \in V^{2h}$, and an oscillatory component $w_o \in W^{2h}$. Then $H^h v = H^h v_s + H^h w_o$. If in addition, we assume that P^h coincides with the eigenvectors of the reduced Hessian (as it is in the case of constant coefficients) we can write H^h as

$$(I - P^h + P^h)H^h(I - P^h + P^h) = (I - P^h)H^h(I - P^h) + P^h H^h P^h \tag{3.1}$$

assuming that P^h is the exact orthogonal projection operator i.e.,

$$(I - P^h)H^h P^h u = P^h H^h (I - P^h)u = 0, \ \forall u \in V^h. \tag{3.2}$$

Therefore we can write $H^h u = g$ as,

$$(I - P^h)H^h(I - P^h)u + P^h H^h P^h u = (I - P^h)g + P^h g. \tag{3.3}$$

Hence $P^h H^h v_s = P^h g$, and

$$(I - P^h)H^h w_o = (I - P^h)g. \tag{3.4}$$

Since we are interested in removing the high-frequency error components while smoothing we solve (3.4). However, since in general P^h will not correspond to the high-frequency spectrum of the Hessian, we can use it as an approximate projection. An alternative approach is to use Chebyshev iterative methods and work on the spectrum of interest, provided we have eigenvalue estimates [3]. In principle, this method is quite similar to our approach (it is used for an entirely different problem.) It uses a number of reduced Hessian matvec operations and computes an exact decomposition. In the present case, we would like to avoid spectra calculations, if possible, as our main goal is to minimize the number of matrix-vector multiplications with the reduced Hessian. (For our smoother in Section 5 we only need the spectra range, and *not all* the eigenvalues.) Instead we approximate $I - P^h$ either by standard interpolation-restriction operators or by using Fourier transforms.

Based on these decompositions we present a smoother that uses PCG for (3.4) and $I_h^{2h} I_{2h}^h$as an approximation to P^h (Section 4). The advantage of this scheme is its straightforward implementation. However, it is hard to analyze its convergence for reasons that we will discuss in the following sections. We also present a second smoother in which we use a two-step stationary solver that acts exclusively on the high-frequency spectrum using exact frequency truncations for P^h (Section 5).

4. PCG smoother and restriction-prolongation projection. We will consider two schemes. In the first one we use a V-cycle multigrid as a solver. In the second one we use multigrid as a preconditioner for CG. In both schemes we will use a few iterations of PCG as a smoother (within multigrid). Our contribution here, is the design of appropriate preconditioners for the PCG smoothing iterations. The preconditioner will be based on an inexact inversion of the $(I - P^h)H^h$. To that end we need to approximate P^h and H^h approximations one for P^h and one for H^h; P^h will be approximated by $I_h^{2h} I_{2h}^h$ since this approach generalizes to arbitrary meshes and it is easy to implement. For H^h we will explore two approaches, one based on the regularization parameter (King preconditioner) and one based on inexact solves of the forward and adjoint solves (Pointwise preconditioner).

KING PRECONDITIONER. This approach was proposed by King in [27] where multigrid method for first kind of integral operator equations were developed. From Figure 2.1, we can see that if the regularization parameter β is sufficiently large, it can approximate most of the high frequency spectrum. Therefore, eigenvalues corresponding to the high-frequency eigenvectors will be β so that $\beta(I - P^h)I \approx (I - P^h)H^h$. Substituting this in the (3.4) we get a single-level preconditioner of the form $\beta^{-1}(I - P^h)$. In a additional approximation step we substitute the orthogonal projection by standard interpolation-restriction operators. Therefore the single-level King preconditioner is given by $\beta^{-1}(I - I_{2h}^h I_h^{2h})$.

In Table 3.1, we summarize the spectral properties of the restriction operator I_h^{2h}, prolongation operators I_{2h}^h, and the orthogonal decomposition operator $I - I_{2h}^h I_h^{2h}$.

POINTWISE PRECONDITIONER. The pointwise preconditioner is based on a pointwise approximation of the reduced Hessian, combined with the high-frequency filtering described in the previous section. The approximate reduced Hessian $\tilde{H}^h$ should approximate well the high-frequency of the true Hessian (for $\beta = 0$) and should be easy to compute. Here we propose a simple waveform-Jacobi relaxation in time. If we discretize in space using the standard three point stencil for the Laplacian on a uniform grid, and introduce a space-Jacobi splitting a matrix vector multiplication with the reduced Hessian (in the frequency domain)

is given by

$$\begin{aligned} \frac{\partial y}{\partial t} + \frac{2\nu}{N^2} y - \frac{2\nu}{N^2} \cos\frac{k\pi}{N} y = u, \qquad & \text{solve for} \quad y \\ \frac{\partial \lambda}{\partial t} + \frac{2\nu}{N^2}\lambda - \frac{2\nu}{N^2}\cos\frac{k\pi}{N}\lambda + y(T-t) = 0, \qquad & \text{solve for} \quad \lambda \\ v = \int \lambda, \qquad & v = Hu. \end{aligned}$$

Here k is the wavenumber, and y, λ and u represent the magnitude of the k^{th} eigenvector. The approximate waveform Jacobi relaxation is given by

$$\begin{aligned} \frac{\partial y_i}{\partial t} + \frac{2\nu}{N^2} y_i - \frac{2\nu}{N^2} \cos\frac{k\pi}{N} y_{i-1} = u, \qquad & i = 1 \dots M \\ \frac{\partial \lambda_i}{\partial t} + \frac{2\nu}{N^2}\lambda_i - \frac{2\nu}{N^2}\cos\frac{k\pi}{N}\lambda_{i-1} + y_M(T-t) = 0, \qquad & i = 1 \dots M \\ v = \int \lambda_M, \qquad & v = \tilde{H}u. \end{aligned}$$

The number of iterations M determines the quality of the preconditioner. So far we have only discretized in space. We use a Backward-Euler scheme to discretize in time. The number of time steps equals the number of discretization points in space. Next, we discuss numerical experiments in which we compare the two preconditioners.

TABLE 4.1

Performance of multigrid solver with PCG smoother. *We report results for a V-cycle multigrid solver. The King and pointwise preconditioners are used in PCG. Here N_s is the size of the problem, β is the regularization parameter. The number of resolved frequencies (number of eigenvalues that are greater than β are reported in brackets);* **K-PCG** *corresponds to the number of V(3, 3) cycles with King preconditioner and* **PF-PCG** *corresponds to the number of V(3, 3) cycles with pointwise preconditioner. Convergence factors, ρ_K and ρ_{PF}, are the average of convergence factors over all the V-cycles till convergence. Stopping criterion for the multigrid solver is $\|r\|/\|r^0\| \leq 10^{-12}$. Two cases of regularization parameter are considered: Case 1: $\beta = 10^{-3}h^2/\nu$ and Case 2:$\beta = 10^{-2}h^2/\nu$. '-' means that the multigrid solver hasn't converged within the 50 V-cycles. Numerical experiment are done for $\nu = 1$, $\nu = 0.01$ with $T = 1$. At the coarsest level $N_s = 16$.*

$\nu = 1$

N_s	β ($\sigma > \beta$)		K-PCG		PF-PCG		ρ_K		ρ_{PF}	
512	4e-09 (40)	4e-08 (22)	14	6	7	6	0.239	0.171	0.041	0.022
1024	1e-09 (57)	1e-08 (32)	-	7	8	6	0.493	0.174	0.076	0.058
2048	2e-10 (81)	2e-09 (45)	-	13	8	6	0.573	0.253	0.162	0.053
4096	6e-11 (114)	6e-10 (64)	-	12	9	6	0.704	0.206	0.150	0.104

$\nu = 0.01$

N_s	β ($\sigma > \beta$)		K-PCG		PF-PCG		ρ_K		ρ_{PF}	
512	4e-07 (131)	4e-06 (72)	22	7	5	5	0.359	0.294	0.007	0.004
1024	1e-07 (183)	1e-06 (102)	29	9	7	6	0.474	0.385	0.025	0.014
2048	2e-08 (257)	2e-07 (144)	-	13	8	6	-	0.164	0.074	0.033
4096	6e-09 (363)	6e-08 (203)	-	19	9	7	-	0.322	0.118	0.089

4.1. Numerical experiments for the PCG smoother. We report numerical experiments in which we compare the the effectiveness of a V-cycle multigrid for (1.2). The V-cycle uses linear finite element based interpolation and restriction operators, and preconditioned

TABLE 4.2

Performance of PCG as a solver with multigrid preconditioner with PCG as smoother in multigrid. *N is the size of the problem, none corresponds to number of PCG iterations without preconditioner. K-PCG corresponds to the PCG iterations with multigrid preconditioner V(3, 3), with King preconditioner in PCG smoother. PF-PCG corresponds to the number of PCG iterations with multigrid preconditioner V(3, 3) with Pointwise preconditioner in PCG smoother. Values in the brackets represent equivalent number of matvecs done at finest level. Stopping criterion for PCG is $\|r\|/\|r^0\| \leq 10^{-12}$. Two cases of regularization parameter: Case 1 is $\beta = 10^{-3}h^2$ and Case 2 is $\beta = 10^{-2}h^2$ are considered. Coarsest level is 16. Parameters used are $\nu = 1$ and $\nu = 0.01$ with $T = 1$.*

$\nu = 1$

N	$\beta(\sigma > \beta)$		none		K-PCG		PF-PCG	
512	3. 81e-09 (40)	3. 81e-08 (22)	167	81	9 (63)	4(28)	5 (35)	4(28)
1024	9. 54e-10 (57)	9. 54e-09 (32)	267	117	14 (98)	5(35)	6 (42)	4(28)
2048	2.38e-10 (81)	2.38e-09 (45)	516	192	22 (154)	8(56)	6 (42)	5(35)
4096	5. 96e-11 (114)	5. 96e-10 (64)	1007	350	28 (196)	11(77)	6 (42)	5(35)

$\nu = 0.01$

N	$\beta(\sigma > \beta)$		none		K-PCG		PF-PCG	
512	3. 81e-07 (131)	3. 81e-06 (72)	959	329	24 (168)	6(42)	4 (28)	4(28)
1024	9. 54e-08 (183)	9. 54e-07 (102)	1780	642	41 (287)	7(49)	5 (35)	5(35)
2048	2.38e-08 (257)	2.38e-07 (144)	3453	1201	99 (693)	9(63)	6 (42)	5(35)
4096	5. 96e-09 (363)	5. 96e-08 (203)	7245	2522	498 (3486)	13(91)	7 (49)	6(42)

Conjugate Gradients as a smoother. Two cases of regularization parameter are considered[4]: $\beta = \frac{10^{-3}h^2}{\nu}, \frac{10^{-2}h^2}{\nu}$. We also study the effect of the diffusion coefficient. We consider two cases of diffusion coefficient: $\nu = 1, 0.01$ are considered in the spectral domain. In all the numerical results given below, *convergence factor* is defined as the average of the ratio of the residuals resulting from two V-cycles. In all experiments $\tilde{H}$ is constructed using 20 waveform-Jacobi iterations for the adjoint and forward problems. In Table 4.1, results are given for multigrid solver. The PCG smoother with pointwise preconditioner converges in all the cases where as King preconditioner fails to converge in case of $\beta = 10^{-3}h^2/\nu$. The pointwise preconditioner is faster than the King preconditioner in case of $\beta = 10^{-2}h^2/\nu$. Despite the fact that they have almost the same effect on the reduced Hessian at the finest level they behave differently at coarser levels. In Figure 4.1 the effect of these two preconditioners on the reduced Hessian is shown for coarser levels.

The pointwise preconditioner has the same effect on the reduced Hessian even at coarser levels unlike King preconditioner. From Figure 4.1 for $\beta = 10^{-3}h^2$, KH has little or no clustering of eigenvalues near high frequency eigenvectors at level 8 whereas PFH has significant clustering. In case of $\beta = 10^{-2}h^2$, both KH and PFH have significant clustering of eigenvalues near high frequency eigenvectors and PFH has more clustering than KH which makes PFH faster. Therefore, the effect of the preconditioner on the reduced Hessian at different levels in multigrid is important to predict the performance of the preconditioner in PCG.

Since we are not using an exact coarse-grid operator, we also test multigrid as a preconditioner within a PCG solver (Table 4.2). In both the cases, we can solve the problem

[4]The regularization parameter is chosen to trade off stability and fidelity to the data. In the present problem, the discretization error is of $\mathcal{O}(h^2)$ and acts as a noise to the problem. In these synthetic experiments (in which we commit several "inverse crimes", [14]) we know the exact spectrum of the reduced Hessian, the level of noise, and our reconstructed solution is expected to be smooth. So the choice regularization is not an issue. In the general case, the choice of regularization is of paramount importance. But this is beyond the scope of this paper.

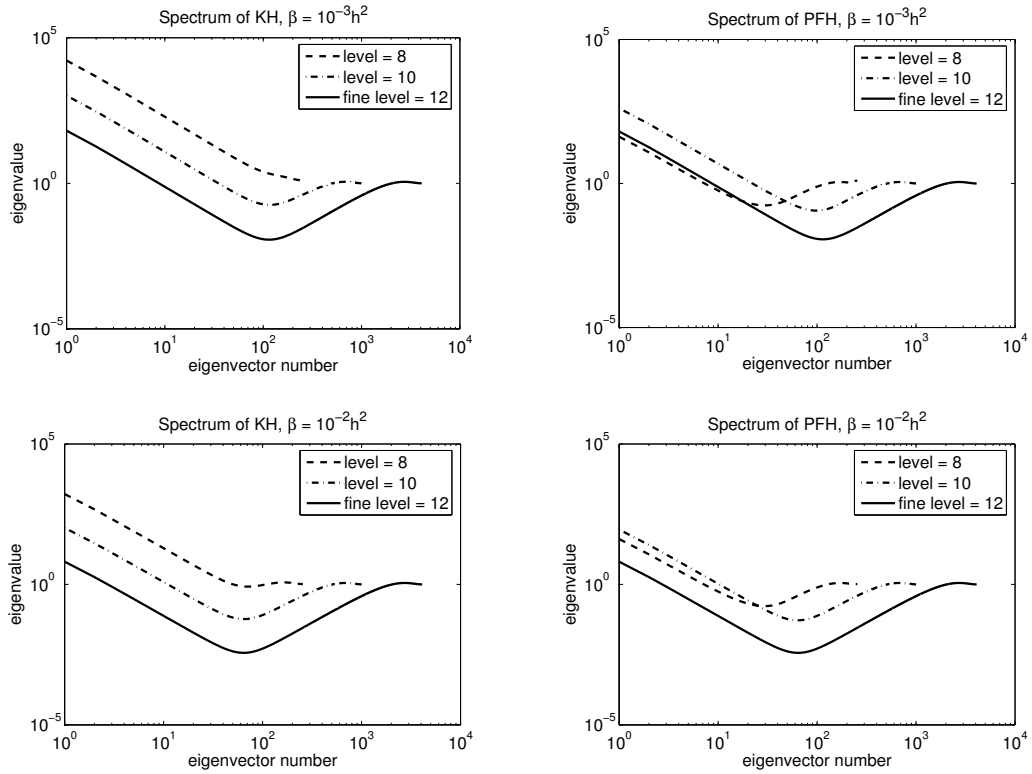

FIG. 4.1. ***Comparison of the spectrum of the preconditioned reduced Hessian***. *We report the discrete spectrum of the preconditioned reduced Hessian with King preconditioner (KH) and pointwise (PFH) preconditioner in spectral domain for the finest level and different coarser levels. The PFH preconditioner has similar clustering of eigenvalues at high-frequency region at finer and coarser levels for different values of regularization parameter. KH does not show similar trend at finer and coarser level for different values of regularization parameter. This is the reason for the robust performance of PFH and failure of KH for smaller regularization parameters.*

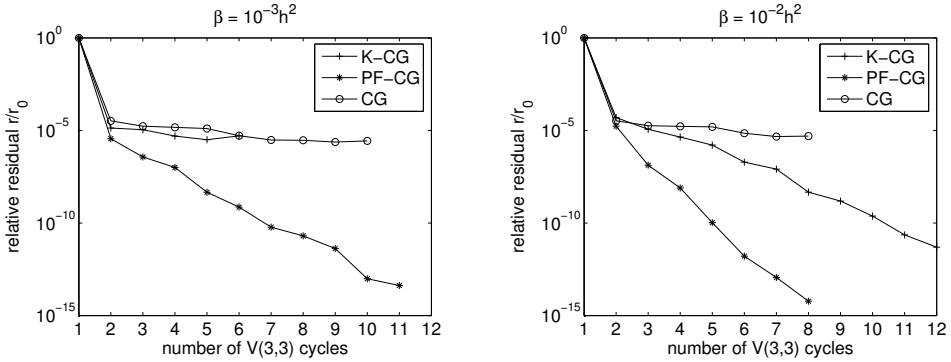

FIG. 4.2. ***V-cycle residual wrt regularization parameter.*** *Relative residual vs V(3, 3) cycles with PCG smoother for two cases of regularization parameter are shown. In this Figure, we compare the rate at which the residual decreases for different preconditioners used in PCG smoother for two cases of regularization parameters. Case 1: $\beta = 10^{-3}h^2$ is shown on the left and Case 2: $\beta = 10^{-2}h^2$ is shown on the right where $h = 1/4096$. King's preconditioner (K-CG) converges in Case 2 and fails to converge in Case 1. Pointwise (PF-CG) preconditioner converges in both the cases. CG fails as a smoother in both the cases.*

in $\mathcal{O}(1)$ iterations as shown in the numerical results. This is equivalent to solving the forward and adjoint problems a constant number of times independent of the mesh size and the regularization parameter.

The King preconditioner has negligible computational cost when compared to the actual reduced Hessian matvec. In case of pointwise preconditioner there is an overhead associated in computing $\tilde{H}^{-1}$ every iteration. For a given residual reduction, it takes of constant number of CG iterations to invert $\tilde{H}^h$. Since the residual reduction is close to machine accuracy, $(\tilde{H}^h)^{-1}$ is a linear operator and creates no convergence problems in the smoother. The computational cost of evaluating $\tilde{H}^{-1}$, however, is much higher than the cost associated with applying the King preconditioner. When the regularization parameter is large the pointwise preconditioner is not necessary. Overall however, the latter is more robust. As seen from the numerical results, pointwise preconditioner converges for different cases of regularization parameters and diffusion coefficients unlike King preconditioner, which works only for larger regularization parameters. Therefore, pointwise preconditioner can be used in general though it has more computational overhead than the King preconditioner. Solving the forward and adjoint problems has a computational complexity [5] of $\mathcal{O}(N_s^2)$ using multigrid algorithms to solve the linear algebraic system of equations at each time step in case of linear problems, where N_s is the number of grid points.

5. Two-step stationary scheme as smoother and FFT filtering. As discussed above, the King preconditioner fails in the case of smaller regularization parameters and the pointwise preconditioner, though robust, has an overhead of computing the inverse of the approximate reduced Hessian at every iteration. The combination of multigrid with PCG and the pointwise preconditioner performs well, at least for the simple model. Our target application ultimately will involve variable coefficient problems and partial observations. In those cases we expect a higher number of iterations. Although we can use multigrid as a solver it would be preferable to combine it with an outer PCG acceleration. Due to the non-stationarity of our scheme, however, this cannot be done.

As an alternative we propose to use an iterative two-step stationary scheme [16] (algorithm 2) as a smoother. Then, in the constant-coefficient case, one can derive exact smoothing factors. As in classical multigrid theory [12, 19], the analysis becomes approximate in the case of variable coefficients. One disadvantage of the two-step solver is that it requires estimates of extreme eigenvalues. To avoid computing eigenvalues we use a the spectral cutoff and analytic spectrum estimates. In this manner the smoother is forced to iterate on the high-frequency regime. In the following we present the algorithm in detail, analyze its convergence factor, and conduct numerical experiments to test our hypothesis.

Algorithm 2 Standard two-step stationary iterative scheme (Solve $Ad^T = d_{in}$)

1: $\sigma_1 = \sigma_{\min}(A)$ and $\sigma_n = \sigma_{\max}(A)$
2: $\rho = \frac{1-\sigma_1/\sigma_n}{1+\sigma_1/\sigma_n}$, $\alpha = \frac{2}{1+(1-\rho^2)^{1/2}}$, $\xi_0 = \frac{2}{\sigma_1+\sigma_n}$, $\xi = \frac{2\alpha}{\sigma_1+\sigma_n}$,
3: $r = -d_{in}$, $d_0 = 0$, $d_1 = \xi_0 r$
4: **for** $i = 1 \dots L$ **do**
5: $\quad r = Ad_1 - d_{in}$
6: $\quad d = \alpha d_1 + (1-\alpha)d_0 - \xi r$
7: $\quad d_0 = d_1,\ d_1 = d$
8: **end for**

Since we are interested in removing the high-frequency error components while smooth-

[5] $N_t = \mathcal{O}(N_s)$

ing, we iterate on (3.4) in the smoothing step. In (3.4) the projection operator $I - P^h$ can be defined as a filter which removes the eigenvector components corresponding to small wave numbers. Let us denote the filtering operation by $W = I - P^h$. In the present problem, the eigenvectors are sines. Therefore, we can use discrete sine transforms to filter the low-frequency components of an input vector (algorithm 3).[6]

Algorithm 3 Projection using Sine Transform

1: Let u be the input vector and let $v = Wu$
2: u_k = IDST(u) $\quad 1 < k < N-1$ — transform into spectral domain
3: $u_k = 0 \quad 1 < k < \frac{N-1}{2}$ — filtering in spectral domain
4: v = DST(u_k) — transform back to spatial domain

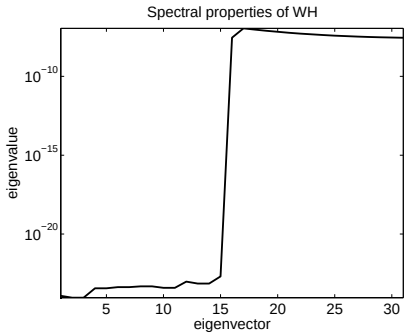

FIG. 5.1. ***Eigenvalues for the spectrally filtered reduced Hessian.*** *Here we report the magnitude of the eigenvalues of the reduced Hessian. Here the W operator represents an exact high-pass filter. During multigrid smoothing, the composite operator WH is inexactly inverted using a two-step stationary iterative solver.*

The problem that we solve during the smoothing iterations is

$$WH^h u = Wg. \tag{5.1}$$

Since the null space of W is non-trivial (5.1) is singular. However, it is proved that a positive semi-definite system of the form (5.1) can be solved by preconditioned conjugate-gradient method (PCG) as long as the right-hand side is consistent [23]. The two-step iterative scheme requires that all the eigenvalues of the matrix (WH^h) be positive (see section 5.2.3 in [4]). Let $W = ZZ^T$ where $Z = [v_{\frac{N+1}{2}}, \ldots, v_{N-1}]$ in which v_k correspond to the k^{th} eigenvector of H^h. The subspace spanned by the eigenvectors $[v_1, \ldots, v_{\frac{N-1}{2}}]$ is invariant and does not influence the convergence rate of the two-step iterative solver.

We define one smoothing step as one iteration of the two-step scheme. In case of non-zero initial guess, the error e^l after l smoothing iterations is given by

$$e^l = ((\alpha - \frac{2\alpha WH}{\sigma_1 + \sigma_n})(1 - \frac{2WH}{\sigma_1 + \sigma_n}) + (1-\alpha))^l e^0, \tag{5.2}$$

where e^0 is the initial error and α is defined in Algorithm 2; σ_1 and σ_n in (5.2) are defined later. Let $e^0 = \Sigma_{k=1}^{N-1} m_k v_k$ where v_k are the eigenvectors of H and m_k are the corresponding error magnitudes. Assuming that W results in an exact decomposition (3.2) eigenvalues of WH are $\sigma(WH) = \{0, 0, \ldots, \sigma_{N+1/2}, \ldots, \sigma_{N-1}\}$, where σ_k correspond to the k^{th} eigenvalue of the reduced Hessian H. Substitute e^0 in (5.2) and take one smoothing iteration, we get $e_k^1 = m_k v_k, \ \forall \ 1 < k < \frac{N-1}{2}$, where e_k^1 is the error component in the k^{th} eigenvector

[6] this is true only in the present case.

direction after one smoothing iteration. Similarly,

$$e_k^1 = \underbrace{((\alpha - \frac{2\alpha\sigma_k}{\sigma_1+\sigma_n})(1-\frac{2\sigma_k}{\sigma_1+\sigma_n}) + (1-\alpha))}_{\mu_k} m_k v_k, \ \forall \ \frac{N+1}{2} < k < N-1.$$

Here μ_k is the amplification factor of the error component in the k^{th} eigenvector direction. The eigenvalues $\sigma_1, \ldots, \sigma_{N-1/2}$ do not affect the iteration. The smoothing factor μ is given by $\max_k(\mu_k)$. To estimate μ we need estimates of σ_1 and σ_n. For the constant coefficient case we have computed these values analytically by (2.9): we fix the values of σ_1 and σ_n to be σ_N and $\sigma_{(N+1)/2}$ respectively. Then, since $\sigma_1 \leq \sigma_k \leq \sigma_n$ we have $\mu_k < 1 \ \forall \ \frac{N+1}{2} < k < N-1$. Using the exact spectrum we can also show that the ratio $\sigma_1/\sigma_n = 1/4$ and it is mesh independent (for $\beta = 0$). In the variable coefficient case we use a heuristic. We estimate σ_n of the unregularized Hessian using a Krylov method on the reduced Hessian. Then, guided by the constant coefficient case, we set $\sigma_1 = \sigma_n/4$. For this ratio the smoothing factor μ is 0.288 for $\nu = 1, 0.01$. In two-level V(2,2) cycles if we use the Galerkin coarse-grid operator (H_G^{2h}),

TABLE 5.1

Convergence for zero regularization parameter. *Number of two-grid V(2,2)-cycles to get a relative residual of 10^{-8}, $V_1, V_{0.01}$ for $\nu = 1, 0.01$ are given when $\beta = 0$. The size of the problem is 2^{level}. The initial guess $u^0 = u^* + \Sigma_k sin(k\pi x)$, where u^* is the exact solution. The theoretical smoothing factor is 0.288 and the numerical is 0.29, and it is mesh independent.*

level	V_1	$V_{0.01}$
4	16	22
5	22	27
6	25	32
7	26	34
8	24	33

some low frequency error components are eliminated in the first V-cycle. In Figure 5.2 we can see that relative residual drops suddenly in the first V-cycle and maintains a constant ratio thereafter. Whereas, the reduction in the error is constant which is expected. The sudden drop in the relative residual is because of the coarse-grid correction where the low-frequency error components are removed.

According to the spectrum of H^h low-frequency error components correspond to large eigenvalues and since $r = He$, there is a sudden drop in the residual. After the first V-cycle, the reduction in the residual is less than the first V-cycle. We report the number of V-cycles to get a relative residual of 10^{-8} in Table 5.1 for different mesh-sizes and two diffusion coefficients $\nu = 1, 0.01$. The number of V-cycles is mesh-independent.

5.1. Multigrid preconditioner. As we have mentioned, one difficulty in designing a multigrid scheme for the reduced Hessian operator is the choice of the coarse grid operator. If we use H^{2h} instead H_G^{2h} we cannot remove certain error components that belong to the intermediate frequency range (of the fine-grid). These error components are neither removed by the 2-step scheme nor by the coarse-grid correction. Therefore, we use multigrid as a preconditioner in PCG so that PCG removes the error components that are not removed by multigrid.

We denote the multigrid preconditioner by M^{-1} (algorithm 4) and the smoothing operator by $S(A, f, u)$ where A, f, u are the matvec operator, the right hand side and the initial guess respectively. We denote inexact multigrid preconditioner by $\tilde{M}^{-1}$ in which exact reduced Hessian H^h is replaced by inexact reduced Hessian $\tilde{H}^h$ in smoothing. $\tilde{H}^h$ is obtained

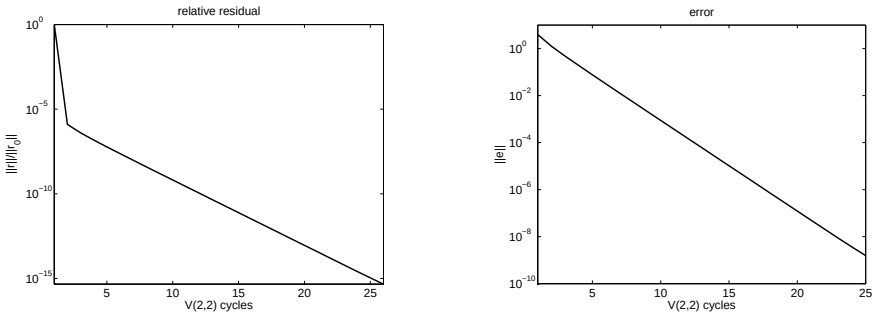

FIG. 5.2. ***Residual reduction in multigrid.*** *Relative residual and error vs number of two-level V(2,2) cycles for $\nu = 1$ and level six and using H_G^{2h} as the coarse-grid operator.*

by replacing exact forward and adjoint solves to do one reduced Hessian matvec by inexact forward and adjoint solves. In order to do this, we use a fixed number of Jacobi waveform relaxation iterations [29]. The number of waveform relaxation steps have to be increased in order to get a completely scalable algorithm because of the convergence properties of the Jacobi waveform relaxation method.

Algorithm 4 Two-level exact multigrid Preconditioner ($u^h = M^{-1} f^h$)

1: $u^h = S(WH^h, Wf^h, 0)$	pre-smoothing with H^h
2: $r^h = f^h - Hu^h$	residual evaluation
3: $r^{2h} = I_h^{2h} r^h$	restriction
4: $e^{2h} = (H^{2h})^{-1} r^{2h}$	coarse-grid correction
5: $e^h = I_{2h}^h e^h$	prolongation
6: $u^h = u^h + e^h$	correction
7: $u^h = S(WH^h, Wf^h, u^h)$	post-smoothing with H^h

In the Jacobi waveform relaxation, we solve ordinary differential equations at every spatial grid point, thus removing the spatial-coupling that arises from the discretization of the Laplacian operator. This is different from the standard spatial weighted Jacobi scheme. The (high-frequency) convergence factors of weighted Jacobi method are mesh-independent—unlike the Jacobi waveform relaxation that gives rise to mesh-dependent convergence factor [29].[7]

5.2. Results and discussion. In this section, we present results for the constant and variable coefficient case, as well as the case in which we have partial observations. We report PCG iterations with multigrid preconditioners M^{-1} and $\tilde{M}^{-1}$. We also show the sensitivity of number of Jacobi waveform relaxation steps on the number of PCG iterations. We present numerical results that interrogate the sensitivity of the scheme on the diffusion coefficient, the number of waveform relaxations, and the coarsening strategy (semi-coarsening or space only vs standard space-time coarsening).

5.2.1. Constant coefficients case. Results for two cases of diffusion coefficients $\nu = 1, 0.01$ are given in Table 5.2. The convergence of PCG is mesh independent. We are reporting results for both two-level and multiple V-cycle preconditioners. The cost of the preconditioner depends on the approximation of the reduced Hessian. Exact matvecs are more

[7] One could use standard time marching schemes in which exact inversions of the spatial operator can be replaced by an inexact solve, like weighted Jacobi.

expensive. For fixed number of waveform relaxations the quality of the approximate Hessian deteriorates with increasing mesh size. The condition number of H^h is $\kappa = \mathcal{O}(N^4)$ where N is the size of the problem. Therefore, without a preconditioner number of CG iterations will be $\mathcal{O}(N^2)$. By using multigrid preconditioner number of CG iterations is mesh-independent $\mathcal{O}(1)$. Using a Backward-Euler time-stepping combined with an optimal spatial solver for the forward and adjoint problems the amount of work done for each reduced Hessian matvec is $\mathcal{O}(NN_t + N\log^2 N)$ where the first term comes from the forward and adjoint solve with N_t time steps, and the second part comes from the multigrid sweeps (the square in the logarithm is related to the fast sine transforms). Therefore, the total amount of work done to solve the system is brought down from $\mathcal{O}(N^4)$ to $\mathcal{O}(N^2)$. *To solve the inverse problem we need to solve the forward problem a constant number of times independent of the regularization parameter and the mesh size.*

TABLE 5.2

***PCG convergence using the exact high-frequency spectrum of the reduced Hessian**. Number of PCG iterations with two-level multigrid preconditioner with exact reduced Hessian in the smoother M^{-1} and inexact reduced Hessian in the smoother $\tilde{M}^{-1}$. Semi-coarsening in space (subscript* sec*) and standard coarsening (subscript* stc*) in space and time are considered. CG is terminated when $\|r\|/\|r_0\| < 10^{-8}$ or when the number of iterations is $2N_s$ where N_s is the size of the problem. The values in the brackets are the number of eigenvectors not filtered by the regularization. Here 16 Jacobi waveform relaxation steps are done for forward and adjoint solves to do one matrix-vector operation of $\tilde{H}$.*

$\nu = 1$

N_s	β	M_{sec}^{-1}	$\tilde{M}_{\text{sec}}^{-1}$	M_{stc}^{-1}	$\tilde{M}_{\text{stc}}^{-1}$
31	5e-07 (31)	10	10	10	10
63	1e-07 (44)	9	9	9	9
127	3e-08 (63)	6	6	7	7
255	7e-09 (89)	4	4	4	4

$\nu = 0.01$

N_s	β	M_{sec}^{-1}	$\tilde{M}_{\text{sec}}^{-1}$	M_{stc}^{-1}	$\tilde{M}_{\text{stc}}^{-1}$
31	5e-03 (31)	14	14	13	13
63	1e-03 (44)	13	13	13	13
127	3e-04 (63	10	10	10	10
255	7e-05 (89)	7	7	7	7

5.2.2. Non-constant coefficient case. We extend the above ideas to solve inverse problems in parabolic problems with non-constant coefficients:

$$\frac{\partial y}{\partial t} - \Delta y = ay + bu \quad \text{in} \quad D,\ y = 0 \quad \text{on} \quad \partial\Omega,\ y(\Omega, 0) = 0 \quad \text{in} \quad \Omega.$$

Equations of this kind are obtained when a non-linear reaction-diffusion equation is linearized. In this case, sines are not the eigenvectors of the reduced Hessian. We assume that a, b are smooth and bounded. Therefore, the Fourier coefficients of a, b decay to zero relatively fast. From this assumption, the contribution of a, b to the spectrum of the forward problem in the high-frequency region is negligible. Using this observation and considering the computational cost of constructing the exact high-frequency eigenspace of the reduced Hessian, we use sine transforms to decompose the finite dimensional space to get acceptable convergence. The numerical results, that we next discuss, indicate that our assumption is reasonable. The reconstructed source is depicted in Figure 5.5.

Two cases of coarsening strategies are implemented : 1) semi-coarsening in space and 2) standard-coarsening is space and time. Mesh-independent convergence of PCG with multi-

TABLE 5.3

Convergence comparisons for PCG using inexact approximations of the reduced Hessian. *We report the number of PCG iterations using a multigrid preconditioner that employs (in the smoother) either an exact reduced Hessian M^{-1}, or an inexact reduced Hessian $\tilde{M}^{-1}$. Semi-coarsening in space (subscript* sec*) and standard coarsening (subscript* stc*) in space and time are considered. PCG is terminated when $\|r\|/\|r_0\| < 10^{-8}$. The values in the brackets in the column β are the number of reconstructed eigenvectors (not filtered by the regularization). The size of the coarsest level problem is 15. Here 16 Jacobi waveform relaxation steps are done for forward and adjoint solves to do one matrix-vector operation of $\tilde{H}$.*

$\nu = 1.0$

N_s	β	M^{-1}_{sec}	$\tilde{M}^{-1}_{\text{sec}}$	M^{-1}_{stc}	$\tilde{M}^{-1}_{\text{stc}}$
31	5e-07 (31)	10	10	10	10
63	1e-07 (44)	13	13	13	17
127	3e-08 (63)	13	14	14	16
255	7e-09 (89)	13	19	13	16
511	2e-09 (127)	15	18	15	17
1023	5e-10 (180)	15	17	15	17

$\nu = 0.01$

N_s	β	M^{-1}_{sec}	$\tilde{M}^{-1}_{\text{sec}}$	M^{-1}_{stc}	$\tilde{M}^{-1}_{\text{stc}}$
31	5e-03 (31)	14	14	13	13
63	1e-03 (44)	17	17	17	17
127	3e-04 (63)	20	20	19	21
255	7e-05 (89)	23	24	21	23
511	2e-05 (127)	24	26	24	27
1023	5e-06 (180)	25	28	25	30

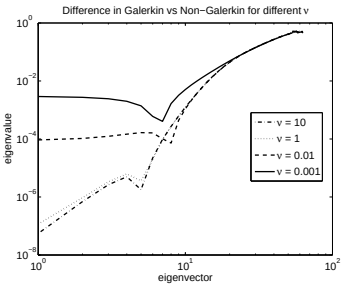

FIG. 5.3. ***Deterioration of non-Galerkin coarse-grid operator.*** *In this figure we depict the deterioration of the non-Galerkin coarse grid operator as a function of the diffusion coefficient. As the diffusion reduces, the pollution from the prolongation and restriction becomes dominant. If we discretize directly in the coarse grid the spectrum of H^{2h} approaches that of the identity operator. On the other hand, the Galerkin operator H^{2h}_G approaches that of $(I^h_{2h} I^{2h}_h)$. (Of course the high-frequency regime of the spectrum is always different.)*

grid preconditioner is observed in case of M^{-1}, whereas performance of $\tilde{M}^{-1}$ slightly deteriorates with mesh-size. Standard coarsening does not perform as well as semi-coarsening. This can be explained by the fact that the convergence factors of the Jacobi waveform relaxation are mesh dependent, given by 1 - $\mathcal{O}(h^2)$ and convergence factors using standard-coarsening are worse than semi-coarsening [22]. If we increase the number of Jacobi waveform relaxation steps with the mesh size then we could observe that the number of PCG iterations with $\tilde{M}^{-1}$ preconditioner will tend to the number of iterations taken by M^{-1}. Results

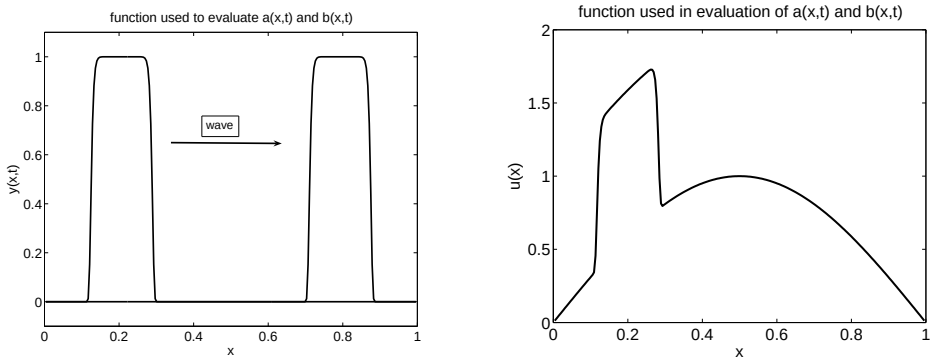

FIG. 5.4. ***Parabolic PDE with variable coefficients.*** *We have constructed a traveling wave solution to emulate solutions to reaction-diffusion equations. The function $\hat{y}(x,t)$ is used to evaluate $a(x,t)$ and $b(x,t)$ which are then used in numerical experiments. The inversion parameter u is depicted on the left panel.*

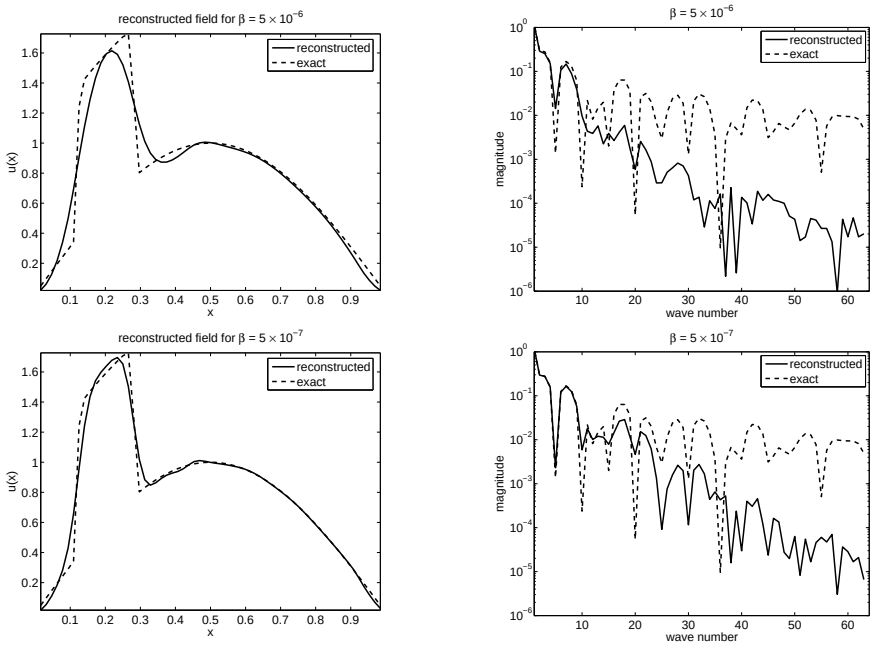

FIG. 5.5. ***Reconstructed source.*** *Here we show the reconstructed curves in the real (left column) and frequency domains (right column) for N=64 and two values of the regularization parameter.*

of PCG with multigrid preconditioners is shown in (Table 5.5 and Table 5.4). The sensitivity of number of PCG iterations with increase in number of Jacobi waveform relaxation steps is reported in Table 5.5. The number of PCG iterations taken by $\tilde{M}^{-1}$ decrease with increase in Jacobi waveform relaxation steps. A lower bound to the number iterations taken by $\tilde{M}^{-1}$ is the number of iterations taken by M^{-1}. The overall computational complexity in using M^{-1} and $\tilde{M}^{-1}$ differ only by a constant if we use a sufficient number of Jacobi waveform relaxation steps in $\tilde{M}^{-1}$.

In Table 5.6, we report the number of PCG iterations when the data is given at seven equally spaced points in space at all the time steps. Exact multigrid preconditioner with standard-coarsening of the exact reduced Hessian and approximate multigrid preconditioner

TABLE 5.4

Multigrid performance for the variable coefficient case. *Number of CG iterations for two-level preconditioner with exact reduced Hessian in the smoother M^{-1} and inexact reduced Hessian in the smoother $\tilde{M}^{-1}$. Semi-coarsening in space (subscript* sec*) and standard coarsening (subscript* stc*) in space and time are considered. CG is terminated when $\|r\|/\|r_0\| < 10^{-8}$ or when the number of iterations is $2N_s$ where N_s is the size of the problem. Case I has the $a = \hat{u}$ and $b = \hat{y}$ and Case II has $a = 2\hat{y}\hat{u}$ and $b = \hat{y}^2$ where $\hat{y}$ is a traveling wave with a Gaussian shape (Figure 5.4) and $\hat{u} = \text{Gaussian}(0.2) + \sin(\pi x)$ (0.2 is the center of the Gaussian). Here 8 Jacobi waveform relaxation steps are done in all the cases in $\tilde{M}^{-1}$.*

CASE I : $a = \hat{u}, b = \hat{y}$

N_s	β	M^{-1}_{sec}	$\tilde{M}^{-1}_{\text{sec}}$	M^{-1}_{stc}	$\tilde{M}^{-1}_{\text{stc}}$
31	2e-06	12	15	13	16
63	5e-07	11	11	13	13
127	1e-07	10	10	12	12
255	3e-08	8	9	10	10

CASE II : $a = 2\hat{y}\hat{u}, b = \hat{y}^2$

N_s	β	M^{-1}_{sec}	$\tilde{M}^{-1}_{\text{sec}}$	M^{-1}_{stc}	$\tilde{M}^{-1}_{\text{stc}}$
31	2e-06	13	15	14	15
63	5e-07	11	12	15	15
127	1e-07	11	11	11	11
255	3e-08	9	9	10	10

TABLE 5.5

Dependence on the fidelity of the reduced Hessian approximation. *Number of CG iterations for multi-level preconditioner with exact reduced Hessian in the smoother M^{-1} and inexact reduced Hessian in the smoother $\tilde{M}^{-1}$. Semi-coarsening in space (subscript* sec*) and standard coarsening (subscript* stc*) in space and time are considered. CG is terminated when $\|r\|/\|r_0\| < 10^{-8}$ or when the number of iterations is $2N_s$ where N_s is the size of the problem. Case I has the $a = \hat{u}$ and $b = \hat{y}$ and Case II has $a = 2\hat{y}\hat{u}$ and $b = \hat{y}^2$ where $\hat{y}$ is a traveling wave with a Gaussian shape (Figure 5.4) and $\hat{u} = \text{Gaussian}(0.2) + \sin(\pi x)$ (0.2 is the center of the Gaussian). Number of Jacobi waveform relaxation steps used in $\tilde{M}^{-1}$ is given in brackets.*

CASE I : $a = \hat{u}, b = \hat{y}$

N_s	β	M^{-1}_{sec}	$\tilde{M}^{-1}_{\text{sec}}(8)$	$\tilde{M}^{-1}_{\text{sec}}(16)$	$\tilde{M}^{-1}_{\text{sec}}(32)$	M^{-1}_{stc}	$\tilde{M}^{-1}_{\text{stc}}(8)$	$\tilde{M}^{-1}_{\text{stc}}(16)$	$\tilde{M}^{-1}_{\text{stc}}(32)$
31	2e-06	12	15	14	15	13	16	15	15
63	5e-07	13	16	15	14	14	16	16	16
127	1e-07	14	27	24	18	17	40	30	30
255	3e-08	18	52	37	26	23	-	-	-

CASE II : $a = 2\hat{y}\hat{u}, b = \hat{y}^2$

N_s	β	M^{-1}_{sec}	$\tilde{M}^{-1}_{\text{sec}}(8)$	$\tilde{M}^{-1}_{\text{sec}}(16)$	$\tilde{M}^{-1}_{\text{sec}}(32)$	M^{-1}_{stc}	$\tilde{M}^{-1}_{\text{stc}}(8)$	$\tilde{M}^{-1}_{\text{stc}}(16)$	$\tilde{M}^{-1}_{\text{stc}}(32)$
31	2e-06	13	15	15	15	14	15	15	15
63	5e-07	14	16	16	16	16	21	21	20
127	1e-07	16	27	28	21	18	-	30	23
255	3e-08	19	140	60	32	20	-	-	-

with semi-coarsening of the approximate reduced Hessian are considered. Results in Table 5.6 show that the multigrid preconditioners presented here are robust even in practical situations when the data is sparse.

6. Full space methods. A disadvantage of a reduced space approach is the need to solve the forward and adjoint problems far from the optimum. In this section, we discuss full space methods where the optimality system is solved for state, adjoint and inversion variables in one shot. The main advantage in solving the problem in full space is that we

TABLE 5.6

Variable coefficients and partial observations. *Number of PCG iterations for multi-level preconditioner for seven observations on the spatial grid. Semi-coarsening in space is represented by subscript* sec *and standard coarsening is represented by subscript* stc*). CG is terminated when* $\|r\|/\|r_0\| < 10^{-8}$*. Case I has the* $a = \hat{u}$ *and* $b = \hat{y}$ *and Case II has* $a = 2\hat{y}\hat{u}$ *and* $b = \hat{y}^2$ *where* $\hat{y}$ *is a traveling wave with a Gaussian shape (Figure 5.4) and* $\hat{u} = \text{Gaussian}(0.2) + \sin(\pi x)$ *(0.2 is the center of the Gaussian).*

CASE I : $a = \hat{u}, b = \hat{y}$			
N_s	β	M_{stc}^{-1}	$\tilde{M}_{\text{sec}}^{-1}(32)$
31	2e-04	13	13
63	5e-05	15	15
127	1e-05	18	22
255	3e-06	19	23

CASE II : $a = 2\hat{y}\hat{u}, b = \hat{y}^2$			
N_s	β	M_{stc}^{-1}	$\tilde{M}_{\text{sec}}^{-1}(32)$
31	2e-04	13	13
63	5e-05	15	15
127	1e-05	18	21
255	3e-06	18	23

can avoid solving the forward and adjoint problems at each iteration which is required in reduced space. On the other hand, the KKT system is more than twice that of the forward problem, it is ill-conditioned, and indefinite. For such systems Krylov solvers are slow to converge. Therefore a good preconditioner is required to make the full space method efficient. A Lagrange-Newton-Krylov-Schur preconditioner (LNKS) has been proposed in [7], [8] in the context of solving optimal control problems with elliptic PDE constraints. In this section we discuss LNKS variants that can be used in the context of inverse problems with parabolic PDE constraints.

Space-time multigrid methods for a parabolic PDE have been considered in literature [22]. In the present problem we have two coupled PDEs with opposite time orientation which provide significant challenge to design a smoother. These issues have been considered in [9] and a time-split collective Gauss-Seidel method (TS-CGS) has been proposed. The optimality condition provided a scalar relation between the control and Lagrange multipliers; in the present problem the control equation is an algebraic-integral equation. Here we discuss the TS-CGS method for our particular problem. (We follow the notation in [9].)

6.1. Lagrange-Newton-Krylov-Schur method (LNKS). In this section we briefly discuss the LNKS method proposed in [7], [8]. LNKS method is based on block factorization of the KKT system which is shown below. (Please refer to [7] for further details.)

$$K = \begin{bmatrix} I & 0 & J^T \\ 0 & \beta I & C^T \\ J & C & 0 \end{bmatrix} = \begin{bmatrix} J^{-1} & 0 & I \\ 0 & I & C^T J^{-T} \\ I & 0 & 0 \end{bmatrix} \begin{bmatrix} J & C & 0 \\ 0 & H & 0 \\ 0 & -J^{-1}C & J^T \end{bmatrix} \tag{6.1}$$

The KKT preconditioner P is then defined as

$$\tilde{P} = \begin{bmatrix} 0 & 0 & I \\ 0 & I & C^T \tilde{J}^{-T} \\ I & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{J} & C & 0 \\ 0 & B & 0 \\ 0 & 0 & \tilde{J}^T \end{bmatrix} \tag{6.2}$$

In $\tilde{P}$, exact forward J^{-1} and adjoint solves J^{-T} are replaced by inexact solves $\tilde{J}^{-1}$ and $\tilde{J}^{-T}$ respectively. The preconditioned KKT matrix is $\tilde{P}^{-1}K$ where

$$\tilde{P}^{-1} = \begin{bmatrix} \tilde{J}^{-1} & -\tilde{J}^{-1}CB^{-1} & 0 \\ 0 & B^{-1} & 0 \\ 0 & 0 & \tilde{J}^{-T} \end{bmatrix} \begin{bmatrix} 0 & 0 & I \\ -C^T\tilde{J}^{-T} & I & 0 \\ I & 0 & 0 \end{bmatrix}. \tag{6.3}$$

A popular method to solve large symmetric indefinite systems is MINRES. One major disadvantage of MINRES is that it requires a symmetric positive definite preconditioner, despite

the fact that the KKT is indefinite. Alternatively, the symmetric Quasi-minimum residual method (SQMR) can be used with indefinite preconditioners but it requires two matvecs per Krylov iteration and it does not take advantage of the fact that the KKT system is symmetric [17]. In all the numerical experiments with LNKS we use SQMR. .

We now discuss a multigrid scheme for the full KKT matrix. We use a V-cycle, with standard restriction and prolongation, and one application $\tilde{P}$ as smoother. The goal is to remove high frequency error components in the state, Lagrange and inversion variables in each step of the smoother without doing exact forward or adjoint solves. Therefore, we use the waveform Jacobi method. To update the inversion variables we use pointwise preconditioner discussed in section 4.

Algorithm 5 LNKS smoother

1: Given y, u, $\boldsymbol{\lambda}$ and $f = [f_y, f_u, f_\lambda]$
2: Evaluate $\tilde{f}_y = y + J^T\boldsymbol{\lambda} - f_y$, $\tilde{f}_u = u + C^T\boldsymbol{\lambda} - f_u$, $\tilde{f}_\lambda = c - f_\lambda$
3: where $c = Jy + Cu$
4: $\tilde{H}p_u = \tilde{f}_u$ — $\tilde{H}$: pointwise preconditioner
5: $\tilde{J}p_y = \tilde{f}_y - Cp_u$ — Inexact forward solve
6: $\tilde{J}^T p_\lambda = \tilde{f}_\lambda - p_y$ — Inexact adjoint solve
7: $y = y - p_y$, $u = u - p_u$, and $\boldsymbol{\lambda} = \boldsymbol{\lambda} - p_\lambda$. — Update

6.2. Time-split Collective Gauss-Seidel (TS-CGS). In this method, we eliminate the inversion variables using the inversion equation (6.4),

$$\beta u - \int_T \lambda \, dt = 0 \quad \text{in} \quad \Omega. \tag{6.4}$$

(Obviously this cannot be done for $\beta = 0$.)

Therefore, we can rewrite the KKT system as :

$$\begin{aligned} &\frac{\partial y}{\partial t} - \Delta y = \frac{1}{\beta}\int_T \lambda \, dt, \quad y(\Omega, 0) = y_0, \quad y(\partial\Omega, t) = 0, \\ &-\frac{\partial \lambda}{\partial t} - \Delta\lambda = -(y - y^*), \quad \lambda(\Omega, 0) = 0, \lambda(\partial\Omega, t) = 0. \end{aligned} \tag{6.5}$$

Using finite differences for Laplacian and backward Euler scheme in time (6.5) the above system can be written as

$$[1 + 2\gamma]y_{im} - \gamma[y_{i-1m} + y_{i+1m}] - y_{im-1} = \frac{\delta t^2}{\beta}\sum_{k=1}^{N_t} \lambda_{ik} \tag{6.6}$$

$$[1 + 2\gamma]\lambda_{im} - \gamma[\lambda_{i-1m} + \lambda_{i+1m}] - \lambda_{im+1} = -\delta t(y_{im} - y^*_{im}), \tag{6.7}$$

where $\gamma = \frac{\delta t}{h^2}$, and i, m represent the spatial and temporal indices of the variables respectively. In case of a collective Gauss-Seidel iteration let us denote the variables as $\phi_k = (y_k, \lambda_k)$ at each grid point. We can write (6.6), (6.7) as $E(\phi_{im}) = [f - A(\phi_{im})] = 0$, at the grid point im. Let E' be the Jacobian of E with respect to (y_k, λ_k). One step of the collective Gauss-Seidel scheme is given by $\phi^1_{im} = \phi^0_{im} - [E'(\phi^0_{im})]^{-1}E(\phi^0_{im})$. This scheme performs well for steady state problems [11] but it diverges in the case of an optimal control of a parabolic PDE because of opposite time orientation of the state and adjoint equations. In

Algorithm 6 Time-split collective Gauss-Seidel method (TS-CGS)

1: Set $\phi^0 = \tilde{\phi}$
2: **for** m = $1, ..., N_t$ do **do**
3: **for** i in lexicographic order do **do**
4: $y^1_{im} = y^0_{im} - [E'(\phi_{im})]^{-1} E(\phi_{im})|_y$
5: $\lambda^1_{iN_t-m} = \lambda^0_{iN_t-m} - [E'(\phi_{im})]^{-1} E(\phi_{im})|_\lambda$
6: **end for**
7: **end for**

order to overcome this problem, time-split collective Gauss-Seidel (TS-CGS) iteration was proposed in [9] (algorithm 6). Following [9] we use Fourier mode analysis to analyze the convergence properties of the two-grid version of the inverse solver. Let the smoothing operator be S_k, and let the coarse-grid correction be given by CG_k^{k-1}. Fourier symbols are represented with a hat on the symbol of the operator. On the fine grid consider the Fourier components $\phi(j, \boldsymbol{\theta}) = e^{ij\cdot\boldsymbol{\theta}}$ where i is the imaginary unit, $j = (j_x, j_t) \in \mathcal{Z} \times \mathcal{Z}$, $\theta = (\theta_x, \theta_t) \in [\pi, \pi)^2$ and $j \cdot \theta = j_x\theta_x + j_t\theta_t$. The frequency domain is spanned by $\boldsymbol{\theta}^{(0,0)} := (\theta_x, \theta_t)$ and$\boldsymbol{\theta}^{(1,0)} := (\bar{\theta}_x, \theta_t)$ $(\theta_x, \theta_t) \in ([-\pi/2, \pi/2) \times [-\pi, \pi))$ and $\bar{\theta_x} = \theta_x - sign(\theta_x)\pi$. Let $E_k^\theta = \text{span}[\phi_k(\cdot, \boldsymbol{\theta}^\alpha) : \alpha \in \{(0, 0), (1, 0)\}$. Assuming all multigrid components are linear and that A_{k-1}^{-1} exists, let the Fourier symbol of the two grid operator TG_k^{k-1} on the space $E_k^\theta \times E_k^\theta$ is given by

$$\hat{TG}_k^{k-1}(\boldsymbol{\theta}) = \hat{S}_k(\boldsymbol{\theta})^{m_2} \hat{CG}_k^{k-1}(\boldsymbol{\theta}) \hat{S}_k(\boldsymbol{\theta})^{m_1}, \tag{6.8}$$

where m_1 and m_2 are the number of pre- and post- smoothing iterations respectively. Using (6.6) and (6.7) Fourier symbol of the smoothing operator is given by

$$\hat{S}(\boldsymbol{\theta}) = \text{diag}\{\sigma(\boldsymbol{\theta}^{(0,0)}), \sigma(\boldsymbol{\theta}^{(1,0)}), \sigma(\boldsymbol{\theta}^{(0,0)}), \sigma(\boldsymbol{\theta}^{(1,0)})\},$$

where

$$\sigma(\boldsymbol{\theta}^{(p,q)}) = \frac{\beta\gamma(2\gamma+1)e^{i\theta_x^p}}{\delta t^3 \sum_{k=1}^{N_t} e^{i(k-m)\theta_t^q} + \beta(2\gamma+1)[1 + 2\gamma - \gamma e^{-i\theta_x^p} - e^{-i\theta_t^q}]}.$$

The smoothing property of the operator S_k is analyzed assuming a perfect coarse-grid correction that removes all low frequency error components and leaves the high frequency error components unchanged. The smoothing property of S_k is defined by

$$\mu = \max\{r(\hat{P}_k^{k-1}(\boldsymbol{\theta}) S_k(\boldsymbol{\theta})) : \boldsymbol{\theta} \in ([-\pi/2, \pi/2) \times [-\pi, \pi))\},$$

where r is the spectral radius and P_k^{k-1} is the projection operator defined on E_k^θ by

$$P_k^{k-1}\phi(\boldsymbol{\theta}, \cdot) = \begin{cases} 0 & \text{if } \boldsymbol{\theta} = \boldsymbol{\theta}^{(0,0)} \\ \phi(\cdot, \boldsymbol{\theta}) & \text{if } \boldsymbol{\theta} = \boldsymbol{\theta}^{(1,0)} \end{cases}.$$

The Fourier symbol for a full-weighting restriction operator is given by

$$\hat{I}_k^{k-1} = \frac{1}{2} \begin{bmatrix} 1 + \cos(\theta_x) & 1 - \cos(\theta_x) & 0 & 0 \\ 0 & 0 & 1 + \cos(\theta_x) & 1 - \cos(\theta_x) \end{bmatrix},$$

and the linear prolongation operator is given by $\hat{I}_{k-1}^{k}(\boldsymbol{\theta}) = \hat{I}_{k}^{k-1}(\boldsymbol{\theta})^T$. The symbol of the fine grid operator is

$$\hat{A}_k(\boldsymbol{\theta}) = \begin{bmatrix} a_y(\boldsymbol{\theta}^{(0,0)}) & 0 & -\delta t^2/\beta & 0 \\ 0 & a_y(\boldsymbol{\theta}^{(1,0)}) & 0 & -\delta t^2/\beta \\ \delta t & 0 & a_p(\boldsymbol{\theta}^{(0,0)}) & 0 \\ 0 & \delta t & 0 & a_p(\boldsymbol{\theta}^{(1,0)}) \end{bmatrix},$$

where

$$a_y(\boldsymbol{\theta}^{(p,q)}) = 2\gamma\cos(\theta_x^p) - e^{-i\theta_t^q} - 2\gamma - 1 \text{ and } a_p(\boldsymbol{\theta}^{(p,q)}) = 2\gamma\cos(\theta_x^p) - e^{i\theta_t^q} - 2\gamma - 1,$$

and the coarse grid correction factor is given by

$$\hat{A}_{k-1}(\boldsymbol{\theta}) = \begin{bmatrix} b_y(\boldsymbol{\theta}^{(0,0)}) & -\delta t^2/\beta \\ \delta t & b_p(\boldsymbol{\theta}^{(0,0)}) \end{bmatrix},$$

where

$$b_y(\boldsymbol{\theta}^{(p,q)}) = \gamma\cos(2\theta_x^p)/2 - e^{-i\theta_t^q} - \gamma/2 - 1 \text{ and } b_p(\boldsymbol{\theta}^{(p,q)}) = \gamma\cos(2\theta_x^p)/2 - e^{i\theta_t^q} - \gamma/2 - 1.$$

Using (6.8) for the definition of the two grid operator we can evaluate the convergence factor by

$$\eta(TG_k^{k-1}) = \sup\{r(\hat{TG}_k^{k-1}(\boldsymbol{\theta})) : \boldsymbol{\theta} \in ([-\pi/2, \pi/2) \times [-\pi, \pi))\}. \tag{6.9}$$

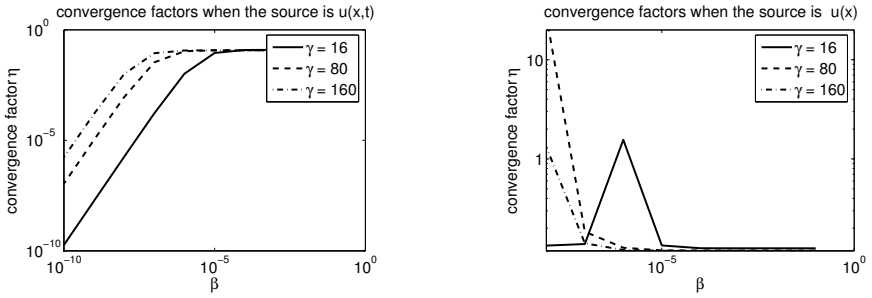

FIG. 6.1. ***Convergence factors for TS-GCS.*** *Convergence factor as a function of β and γ when (left) the source term is u [9] and (right) for inverse problem when the source term is u. When the source term is a function of space, convergence factors are greater than 1 for certain range of β and γ.*

In [9] Fourier mode analysis was carried for a spatiotemporal course time and the convergence factors were less sensitive to γ and β. In the present problem, η this is not the case: for small values of β the method fails to converge (Figure 6.1).

6.3. Numerical results. In Table 6.1 SQMR iterations using LNKS preconditioner P and multigrid preconditioner MG with $\tilde{P}$ smoother for three regularization parameters are reported. SQMR converges to the required tolerance in constant iterations using P and MG. SQMR with MG preconditioner takes less iterations than with P preconditioner. In P one exact forward and adjoint solve are done at every iteration. Whereas in MG only inexact forward and adjoint solves are done at every iteration at different levels of multigrid. One

TABLE 6.1

LNKS preconditioner. *Performance of preconditioned SQMR solver for a constant regularization parameter. P represents the number of SQMR iterations with P version of the LNKS preconditioner. MG corresponds to SQMR iterations using multigrid preconditioner with $\tilde{P}$ as smoother in multigrid. Stopping criterion for SQMR is $\|r\|/\|r_0\| \leq 10^{-8}$. Three cases of regularization are considered $\beta = 10^{-2}, 10^{-4}, 10^{-6}$.*

$N_s \times N_t$	β			P		
17 x 8	1e-02	1e-04	1e-06	10	21	36
33 x 16	1e-02	1e-04	1e-06	13	21	51
65 x 32	1e-02	1e-04	1e-06	14	21	54
129 x 64	1e-02	1e-04	1e-06	14	21	56
$N_s \times N_t$	β			MG		
17 x 8	1e-02	1e-04	1e-06	5	7	12
33 x 16	1e-02	1e-04	1e-06	7	8	16
65 x 32	1e-02	1e-04	1e-06	10	10	17
129 x 64	1e-02	1e-04	1e-06	12	12	17

TABLE 6.2

TS-CGS results for inverse problem. *We observe that the convergence is not sensitive to decrease in regularization parameter for larger regularization parameter. For smaller regularization parameter multigrid solver diverged ($\beta < 10^{-4}$) which agrees with the Fourier mode analysis Figure 6.1.*

$N_s \times N_t$	β	ρ	r_y	r_λ
17 x 8	1e-02	0.091	5e-10	1e-10
33 x 16	1e-02	0.101	3e-10	1e-10
65 x 32	1e-02	0.127	4e-09	5e-10
129 x 64	1e-02	0.130	5e-09	7e-10
17 x 8	1e-04	0.127	5e-08	3e-10
33 x 16	1e-04	0.134	2e-08	1e-10
65 x 32	1e-04	0.130	1e-08	7e-11
129 x 64	1e-04	0.131	2e-08	1e-10

major advantage of solving the problem in full space and using multigrid preconditioner is that we avoid any forward or adjoint solves which are inevitable in reduced space methods. Even in this case the computational complexity is $\mathcal{O}(N_s^2)$ as we need to do a KKT matvec at every iteration.

In Table 6.2 convergence factors and residuals of the multigrid solver using TS-CGS smoother are given. Multigrid solver converges for $\beta = 10^{-2}, 10^{-4}$ and diverges in the case of $\beta = 10^{-6}$. This agrees with the convergence factors estimates obtained from Fourier mode analysis which show that multigrid solver using TS-CGS smoother has convergence factors greater than 1 for certain combination of β and γ.

7. Conclusions. In this paper, we presented multigrid algorithms for inverse problems with linear parabolic PDE constraints. Our algorithms are designed for the case in which the inversion variable depends only in space. Although there is prior work on multigrid for optimization problems, there is no work on algorithms for vanishing regularization parameters. Assuming that we have sufficient information in the data and we need accurate reconstructions, existing schemes will not have mesh-independent convergence rates. Motivated by this observation, our main aim was to construct schemes that are robust to vanishing regularization parameter and allow fast high-fidelity reconstructions.

The key component in our scheme is the multigrid smoother. We use a high-pass filter

that allows an iterative solver to work exclusively in the high frequency regime. The second component is to accelerate the computation by using appropriate inexact versions of the reduced Hessian. By using an exact high-pass filter and a two-step stationary iterative solver as a preconditioner we were able to analyze the behavior of the algorithm. The overall scheme uses a V-cycle multigrid to accelerate a CG solver that iterates in the reduced space. In addition, we examined alternative smoothing strategies that use cheaper high-pass filters, the effects of the diffusion, and the effect of the coarse-grid operator. The high-frequency projections are preferable, but are limited to the cases in which Fourier-type expansions can be carried through fast transforms.

Our numerical experiments gave promising results and justified the extension of our scheme to problems with variable-coefficient PDE constraints and partial measurements. Finally, we combined the reduced space with a full-space solver so that we avoid solving a forward and an adjoint problem at each optimization iteration.

All the implementations were in `MATLAB` and no effort was made to optimize the code. So we refrained from reporting wall-clock times. We should emphasize, however, that, although the method has optimal complexity, the associated constants can be high. In fact, if the number of the sought frequencies in the reconstructed field is small, then the regularization parameter should be set to a relatively large value. In that case one can use much cheaper solvers, for example schemes based on the King preconditioner and inexact L^2 projections.

We would like to caution the reader that we have committed several "inverse crimes" buy choosing attainable observations, the simplest possible regularization, and by having zero noise (besides discretization noise). These parameters significantly change the quality of the reconstruction and can potentially alter the behavior of the solvers. These topics, however, are beyond the scope of the present paper.

The extension of the results to higher-dimensions are straightforward. The implementation, however, is not. Further complexity analysis and algorithmic tuning are required to implement an efficient and parallelizable scheme. Most important an optimal method is highly problem dependent. In the case of sparse partial observations for example, the full space method has much higher storage requirements than the reduced space approach (this is a reason we pursued reduced space methods). Extension to higher dimensions is ongoing work and will be reported elsewhere. The method can be used for nonlinear problems, for example within a Newton multigrid context. Alternatively, nonlinear multigrid methods can be considered. The notion of iterating on the high-frequency spectrum using exact projections and pointwise approximations can be potentially extended to the nonlinear full approximation multigrid case.

Acknowledgments. We would like to thank Alfio Borzì and Omar Ghattas for their insightful comments on multigrid and inverse problems.

REFERENCES

[1] V. AKCELIK, G. BIROS, A. DRAGANESCU, O. GHATTAS, J. HILL, AND B. V. B. WAANDERS, *Dynamic data driven inversion for terascale simulations: Real-time identification of airborne contaminants*, Proceedings of the 2005 ACM/IEEE conference on Supercomputing, (2005).

[2] E. ARIAN AND S. TA'SAAN, *Multigrid one shot methods for optimal control problems: Infinite dimensional control*, NASA-CR-194939, Technical report, (1994).

[3] M. ARIOLI AND D. RUIZ, *A Chebyshev-based two-stage iterative method as an alternative to the direct solution of linear systems*, Technical Report RAL-TR-2002-021, Rutherford Appleton Laboratory, Atlas Center, Didcot, Oxfordshire, England, (2002).

[4] O. AXELSSON, *Iterative Solution Methods*, Cambridge University Press, 1994.

[5] H. T. BANKS AND K. KUNISCH, *Estimation Techniques for Distributed Parameter Systems*, Birkhauser, 1989.

[6] BENZI, M. AND GOLUB, G. H. AND LIESEN, J., *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), p. 1.

[7] G. BIROS AND O. GHATTAS, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I : The Krylov-Schur solver*, SIAM Journal of Scientific Computing, 27 (2005), pp. 687–713.

[8] ———, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part II : The Lagrange-Newton solver and its application to optimal control of steady state flows*, SIAM Journal of Scientific Computing, 27 (2005), pp. 714–739.

[9] A. BORZÌ, *Multigrid methods for parabolic distributed optimal control problems*, Journal of Computational and Applied Mathematics, 157 (2003), pp. 365–382.

[10] A. BORZÌ AND R. GRIESSE, *Experiences with a space-time multigrid method for the optimal control of a chemical turbulence model*, International journal for numerical methods in fluids, 47 (2005), pp. 879–885.

[11] A. BORZÌ AND K. KUNISCH, *The numerical solution of the steady state solid fuel ignition model and its optimal control*, SIAM Journal of Scientific Computing, 1 (2000), pp. 263–284.

[12] A. BRANDT, *Multi-level adaptive solutions to boundary-value problems*, Mathematics of Computation, 31 (1977), pp. 333–390.

[13] W. BRIGGS, V. E. HENSON, AND S. F. MCCORMICK, *A Multigrid tutorial*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[14] D. COLTON AND R. KRESS, *Inverse Acoustic and Electromagnetic Scattering Theory, 2nd Edition*, Applied Mathematical Sciences, Springer, 1998.

[15] T. DREYER, B. MAAR, AND V. SCHULZ, *Multigrid optimization and applications*, Journal of Computational and Applied Mathematics, 120 (2000), pp. 67–84.

[16] S. P. FRANKEL, *Convergence rates of iterative treatments of PDEs*, Math. Tables Aids Comput, 4 (1950), pp. 65–75.

[17] R. W. FREUND AND N. M. NACHTIGAL, *An implementation of the QMR method based on coupled two-term recurrences*, SIAM Journal of Scientific Computing, 15 (1994), pp. 313–337.

[18] L. GIRAUD, D. RUIZ, AND A. TOUHAMI, *A comparative study of iterative solvers exploiting spectral information of SPD systems*, SIAM Journal of Scientific Computing, 27 (2006), pp. 1760–1786.

[19] W. HACKBUSCH, *Multigrid methods and applications*, Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1985.

[20] M. HANKE AND C. R. VOGEL, *Two-level preconditioners for regularized inverse problems I. Theory*, Numerische Mathematik, 83 (1999), pp. 385–402.

[21] HEINKENSCHLOSS, M., *A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems*, Journal Of Computational And Applied Mathematics, 173 (2005), pp. 169–198.

[22] G. HORTON AND S. VANDEWALLE, *A space-time multigrid method for parabolic partial differential equations*, SIAM Journal of Scientific Computing, 16 (1995), pp. 848–864.

[23] E. KAASSCHIETER, *Preconditioned conjugate gradient for solving singular systems*, Journal of Computational and Applied mathematics, 24 (1988), pp. 265–275.

[24] B. KALTENBACHER, *On the regularizing properties of a full multigrid method for ill-posed problems*, Inverse problems, 17 (2001), pp. 767–788.

[25] ———, *V-cycle convergence of some multigrid methods for ill-posed problems*, Mathematics of Computation, 72 (2003), pp. 1711–1730.

[26] J. T. KING, *On the construction of preconditioners by subspace decomposition*, Journal of Computational and Applied mathematics, 29 (1990), pp. 195–205.

[27] ———, *Multilevel algorithms for ill-posed problems*, Numerische Mathematik, 61 (1992), pp. 311–334.

[28] J. R. SHEWCHUK, *An introduction to the conjugate gradient method without agonizing pain*, CMU-CS-94-125,Technical Report, (1994).

[29] S. VANDEWALLE AND R. PIESSENS, *Efficient parallel algorithms for solving initial-boundary value problems and time periodic parabolic PDEs*, SIAM Journal of Scientific and Statistical Computing, 13 (1992), pp. 1330–1346.