FAST ALGORITHMS FOR INVERSE PROBLEMS WITH PARABOLIC PDE CONSTRAINTS. *

SANTI S. ADAVANI † AND GEORGE BIROS‡

Abstract. We present optimal complexity algorithms to solve the inverse problem with parabolic PDE constraints on the 2D unit box where the temporal variation of a source function is known but the spatial variation is unknown. We consider measurements on a single boundary and two opposite boundaries. The problem is formulated as a PDE-constrained optimization problem. We use a reduced space approach in which we eliminate the state and adjoint variables and we iterate in the inversion parameter space using the Conjugate Gradients algorithm. We derive analytical expressions for the entries of the reduced Hessian and propose preconditioners based on a low-rank approximation of the Hessian. We also propose preconditioners for problems with non-constant coefficient PDE constraints. We perform numerical experiments to show the effectiveness of the preconditioners for different source functions and noise levels. We observed mesh-independent and noise-independent convergence of CG with the preconditioner. The overall computational complexity of solving the inverse problem is the same as that of solving the forward problem, which is $\mathcal{O}(NN_t \log N)$, where N is the spatial discretization and N_t is the number of time steps.

1. Introduction.

We present fast algorithms to solve inverse problems with linear parabolic partial differential equations (PDEs) as constraints. We consider problems in which the temporal variation of the source function is known but the spatial variation is unknown. More precisely, we reconstruct an unknown time-independent source function $u(\mathbf{x})$ in the heat equation by solving the following PDE-constrained optimization problem:

$$\min_{y,u} \mathcal{J}(y,u) := \frac{1}{2}\|Qy - z\|^2_{L^2(\Omega)\times L^2(0,T]} + \frac{\beta}{2}\|u\|^2_{L^2(\Omega)}$$

subject to:

$$\frac{\partial y(\mathbf{x},t)}{\partial t} - \nu\Delta y(\mathbf{x},t) = a(\mathbf{x},t)y(\mathbf{x},t) + b(\mathbf{x},t)u(\mathbf{x}) \quad \text{in} \quad \Omega \times (0,T],$$
$$\nu\nabla y \cdot \mathbf{n} = 0 \quad \text{on} \quad \partial\Omega,\ y(\Omega,0) = 0 \quad \text{in} \quad \Omega,$$

where $\Omega = \{(x_1,x_2) : (x_1,x_2) \in [0,1]\times[0,1]\}$ as shown in Figure 1.1, Q is the observation operator, $z = Qy^*$ and y^* is the solution of the parabolic PDE or the *"forward problem"* for the exact source u^*. 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. We assume that $a(\mathbf{x},t)$ and $b(\mathbf{x},t) > 0$ are known, smooth and bounded function. This problem is motivated by inverse medium and data assimilation problems that are constrained by reaction-convection-diffusion equations. The problem is to reconstruct the source function $u(\mathbf{x})$ given Neumann-to-Dirichlet data.

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

$$\frac{\partial y(\mathbf{x},t)}{\partial t} - \nu\Delta y(\mathbf{x},t) = a(\mathbf{x},t)y(\mathbf{x},t) + b(\mathbf{x},t)u(\mathbf{x}) \quad \text{in} \quad \Omega \times (0,T],$$
$$\nu\nabla y \cdot \mathbf{n} = 0 \quad \text{on} \quad \partial\Omega,\ y(\Omega,0) = 0 \quad \text{in} \quad \Omega.$$

* This work is supported by the U.S. Department of Energy under grant DE-FG02-04ER25646.

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

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

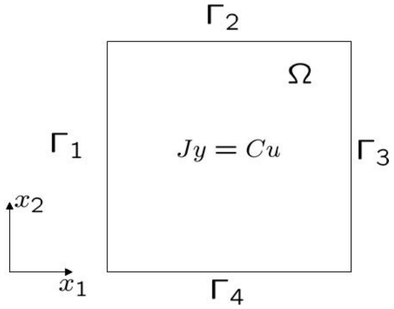


FIG. 1.1. *The forward problem $Jy = Cu$ is solved in the square domain $\Omega = \{(x_1, x_2) : (x_1, x_2) \in [0,1] \times [0,1]\}$ with zero Neumann conditions on the boundary $\partial\Omega = \cup\Gamma_i$ where $i = \{1,2,3,4\}$, where J and C are the Jacobians of the parabolic PDE with respect to the state and the inversion variables respectively.*

Adjoint

$$-\frac{\partial\lambda}{\partial t} - \nu\Delta\lambda - a(\mathbf{x},t)\lambda = -Q^T(Qy - z) \quad \text{in} \quad \Omega \times (0,T],$$
$$\nu\nabla\lambda \cdot \mathbf{n} = 0 \quad \text{on} \quad \partial\Omega,\ \lambda(\mathbf{x},T) = 0 \quad \text{in} \quad \Omega.$$

Inversion

$$\beta u(\mathbf{x}) - \int_0^T b(\mathbf{x},t)\lambda(\mathbf{x},t)\,dt = 0.$$

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

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

where J and C are the Jacobians of the constraints with respect to the state and the inversion variables respectively. 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 [9]. In this paper, we will consider the so-called *"reduced space"* method [15] where one solves for u by an iterative solver on 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

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

where $H = C^TJ^{-T}Q^TQJ^{-1}C + \beta I$ is known as the *"reduced Hessian"* (hereinafter, *"Hessian"*) and $g = -C^TJ^{-T}Q^Tz$ is the *"reduced gradient"*. Since Q is positive semi-definite, H is a symmetric and strictly positive definite operator. Each Hessian matrix-vector product (hereinafter, " *matvec*") requires one exact forward solve and one adjoint solve, which makes it expensive to solve (1.2). We focus on the design of efficient solvers for the reduced space formulations.

Terminology and Notation. We refer to the problem in which $a(\mathbf{x},t)$ is a constant and $b(\mathbf{x},t) = 1(t)$ as the *"constant coefficient problem"*. We refer to the problems in which $a(\mathbf{x},t)$ is non-constant and $b(\mathbf{x},t) \neq 1(t)$ as the *"non-constant coefficient problem"*. We use n, δ, and N_t for the number of cosine modes per spatial dimension, the time step and the total number of time steps respectively. We represent the k^{th} coefficient of the 1-D discrete cosine transform (DCT) of the function $f(x)$ with f_k and the (k_1, k_2) coefficient of the 2-D DCT of the function $f(\mathbf{x})$ with $f_{k_1 k_2}$. The subscript k_1 or k_2 represents the k_1^{th} or k_2^{th} DCT coefficient in x_1 (or x_2) direction respectively.

Related work. In [27], the authors proved that the time-independent source term of a heat equation in a rectangle can be uniquely determined from the base boundary data $y(x_1, 0, t)(0 < x_1 < 1, 0 < t < T)$. Assuming that u is restricted to a compact set in the Sobolev spaces, they also gave an estimate of $\|u\|_{L^2}$. In [17], these uniqueness results have been extended to arbitrary domains. There is not much work in the design of fast algorithms for source inversion for parabolic problems. In [4, 12], optimal algorithms for source inversion problems have been proposed in the case of domain observations. To solve (1.2), direct solvers are not a viable option since the Hessian is a non-local, dense operator. The Preconditioned Conjugate Gradients (PCG) algorithm requires matvecs only and 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 $\mathcal{O}(1/\beta)$ [4, 10]. 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 analysis 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 either a discretization in which u is discretized in a coarser grid or can use a relatively large value for β. In the second case, which is pertinent to scalability of the inverse problem solver, we have high-quality observations that allow for a high-resolution reconstruction of u. Here, we should notice that we do not need full observations to reconstruct (as opposed to the elliptic case) since we assume only spatial dependence on the source. This implies that β cannot be fixed as we refine the mesh because it will prevent recovery of the high frequencies. Thus, β has to be mesh-dependent.

Returning to the 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 [6]. In [11], a two-step stationary iterative method that does not need an assembled matrix was used to precondition the Hessian.

In our case, the unregularized Hessian is a Fredholm operator of the first kind. 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 [24]. In [23], vanishing regularization parameter has been discussed and an algorithmically optimal scheme is presented. Such problems were further analyzed in [21] and [22]. A multigrid preconditioner based on that work was also used in [5] to solve problems with million of inversion parameters with a relatively small but non-vanishing value of the regularization parameter. Here, we consider only boundary measurements unlike our previous work [4] where we proposed scalable multigrid algorithms for mesh-dependent β in the case of full domain and partial domain observations.

Approach. We use the following three-step approach to solve the inverse problem:

1. Derive analytical expressions for the entries of the Hessian.
2. Construct a low-rank approximation of the Hessian using the analytically derived Hessian.

3. Construct a preconditioner based on the low-rank approximation of the Hessian.

Contributions. *The main contributions of this work are: 1) analytical expressions for the entries of the Hessian for the constant coefficient problem on a square domain, 2) a preconditioner based on the low-rank approximation of the Hessian for the constant coefficient problem, and 3) a preconditioner for the non-constant coefficient problem using the preconditioner for the constant coefficient problem.*

Limitations. In this paper, all our derivations are based on the assumption that we have a square domain. It is non-trivial to extend these derivations to arbitrary domains. We consider L^2 Tikhonov regularization and a different choice of regularization like H^1 or total variation (TV) will affect the quality of reconstruction and behavior of the solvers significantly; these aspects; however, are beyond the scope of this paper.

1.1. Organization of the paper.

In Section 2, we discuss the spatial discretization and time marching scheme used to solve the state and the adjoint equations. In Section 3, we derive analytical expressions for the entries of the Hessian for the constant coefficient problem and propose preconditioners based on low-rank approximation of the Hessian. In Section 4, we derive preconditioners for non-constant coefficient problem from the preconditioners for the constant coefficient problem. We present numerical experiments to show the effectiveness of the preconditioners and the reconstructed solutions in Section 6. Finally, we summarize the the algorithms, results and present the future work in Section 7.

2. Parabolic PDE solver.

To solve the state and the adjoint equations, we use a Fourier-Galerkin method for spatial discretization and backward Euler scheme to march in time. The variables $y(\mathbf{x},t)$, $\lambda(\mathbf{x},t)$ and $u(\mathbf{x})$ are represented as a linear combination of the trial functions in a finite dimensional space X_n. Since we have zero Neumann boundary conditions, we choose the trial space X_n to be C_n the cosine functions. A given function $f(\mathbf{x},t)$ is approximated by using the truncated cosine series[1]

$$f(\mathbf{x},t) \approx \sum_{k_2=0}^{n-1}\sum_{k_1=0}^{n-1} f_{k_1k_2}(t)\cos(k_1\pi x_1)\cos(k_2\pi x_2). \tag{2.1}$$

In the state equation, we solve for the unknowns $y_{k_1k_2}(t)$ for $k_1,k_2 = 0,1,\ldots,n-1$. We use the backward Euler scheme for time stepping. By using the test functions in $Y_n = C_n$ and the orthonormality of the test and trial functions, we obtain

$$\begin{aligned} J^q y_{k_1k_2}(q\delta) &= (1+\delta\nu(k_1^2+k_2^2)\pi^2 - \delta F a(\mathbf{x},q\delta)F^T)y_{k_1k_2}(q\delta) \\ &= y_{k_1k_2}((q-1)\delta) + \delta F b(\mathbf{x},q\delta)F^T u_{k_1k_2}, \end{aligned} \tag{2.2}$$

where J^q is the Jacobian of the state equation at the q^{th} time step, and F and F^T are the forward and inverse DCT operators respectively. If $a(\mathbf{x},t)=0$ then the solution to (2.2) is

$$y_{k_1k_2}(q\delta) = \frac{y_{k_1k_2}((q-1)\delta) + \delta F b(\mathbf{x},q\delta)F^T u_{k_1k_2}}{1+\delta\nu(k_1^2+k_2^2)\pi^2}. \tag{2.3}$$

For variable $a(\mathbf{x},t)$, we use PCG with a spectral preconditioner [8, 13, 26]. We define the forward spectral preconditioner D to solve for $y_{k_1k_2}(q\delta)$ as:

$$D f_{k_1k_2} = \frac{f_{k_1k_2}}{1+\bar{a}+\nu\delta\pi^2(k_1^2+k_2^2)}\,\forall f_{k_1k_2}, \tag{2.4}$$

[1] For the sake of clarity, the coefficients $1, \sqrt{2}$ are absorbed into $f_{k_1k_2}$

where $\bar{a} = \int_\Omega a(\mathbf{x}, q\delta) d\Omega$. To solve a linear system of equations of the form $J^q y_{k_1k_2}(q\delta) = y_{k_1k_2}((q-1)\delta) + \delta F b(\mathbf{x}, q\delta) F^T u_{k_1k_2}$, PCG only requires the matvec $J^q y_{k_1k_2}$. Therefore, we solve (2.2) in a matrix-free manner without explicitly forming J^q. In the forward problem, the matvec of the forward operator requires one forward and one inverse DCT to perform $Fa(\mathbf{x}, q\delta)F^T y_{k_1k_2}$ in (2.2). We perform DCT using the Fast Fourier Transform (FFT) [14], which has an optimal computational complexity of $\mathcal{O}(N \log N)$ where $N = n^2$.

The adjoint PDE is backwards in time and it is driven by the boundary conditions. Let f^i be the forcing term on the boundary Γ_i(Figure 1.1). By using the orthonormality of test and trial function and integration by parts, the solution to the adjoint Equation is given by

$$\begin{aligned} J^{T^q} \lambda_{k_1k_2}(T - q\delta) &= (1 + \delta\nu(k_1^2 + k_2^2)\pi^2 - \delta F a(\mathbf{x}, T - q\delta) F^T) \lambda_{k_1k_2}(T - q\delta) \\ &= \lambda_{k_1k_2}(T - (q-1)\delta) + \delta(f^1_{k_2} + (-1)^{k_2} f^2_{k_1} + (-1)^{k_1} f^3_{k_2} + f^4_{k_1}), \end{aligned} \tag{2.5}$$

where J^{T^q} is the Jacobian of the adjoint equation at the q^{th} time step. We use PCG with the spectral preconditioner defined in (2.4) to solve the system of equations that arise at each time step.

Note. The choice of the time stepping scheme will effect the symmetry of the Hessian. If we choose second-order Crank Nicholson scheme instead of backward Euler to solve the forward and the adjoint PDEs, the Hessian will be unsymmetric and PCG cannot be used to solve (1.2).

3. Fast inversion algorithm.

In this section, we derive analytical expressions for the entries of the Hessian and propose a preconditioner in the case of the constant coefficient problem. This preconditioner is based on a low-rank approximation of the analytical Hessian. We derive analytical expressions for the Hessian when the data is given on the 1) left boundary, and 2) left and right boundaries. In the derivations to follow, let

$$\sigma_{k_1k_2}(t) = \frac{1 - \exp(-(k_1^2 + k_2^2)\nu\pi^2 t)}{\alpha + \nu(k_1^2 + k_2^2)\pi^2}. \tag{3.1}$$

The solution to the forward problem is given by

$$y(\mathbf{x}, t) = \sum_{k_1=0}^{n-1} \sum_{k_2=0}^{n-1} \sigma_{k_1k_2}(t) u_{k_1k_2} \cos(k_1\pi x_1) \cos(k_2\pi x_2). \tag{3.2}$$

Let $K : u \to z$ be the Green's operator that maps the functions from the inversion variable space to the observations; $K = QJ^{-1}C$. In the case of data on the left boundary, Green's operator is defined as

$$Ku = y_{k_2}(t) = \sum_{k_1=0}^{n-1} \sigma_{k_1k_2}(t) u_{k_1k_2}. \tag{3.3}$$

In the discretized form, $K = [K^1, K^2, \cdots, K^q, \cdots, K^{N_t}]$, where $K^q \in \mathcal{R}^{n \times n^2}$ and $K^q u = y_{k_2}(q\delta)$. Since K^q is a summation over a single index k_1, non-zero entries in the rows of K^q do not overlap. The Hessian $H = \sum_q K_q^T K_q$ and $K_q^T K_q$ is a block diagonal matrix with n blocks and each block is of size $n \times n$. Let $H_q = K_q^T K_q$ and the k_2^{th} block of H_q is given by

$$H_q^{k_2} = \sigma_{\cdot k_2}(q\delta) \otimes \sigma_{\cdot k_2}(q\delta), \tag{3.4}$$

where $\sigma_{\cdot k_2} = [\sigma_{0k_2}, \sigma_{1k_2}, \cdots, \sigma_{n-1k_2}]$ and the k_2^{th} block of Hessian H is given by

$$H^{k_2} \approx \int_0^T \sigma_{\cdot k_2}(t) \otimes \sigma_{\cdot k_2}(t) dt. \tag{3.5}$$

In Figure 3.1, the eigenvalues and eigenvectors of different blocks of the Hessian are shown. In all the cases, the eigenvalues decay rapidly and approach zero very fast. Also, the frequency of the eigenvectors is higher for smaller eigenvalues that makes this problem highly ill-posed. Since the eigenvalues approach zero very fast, H^{k_2} can be approximated by a low-rank matrix corresponding to the first few eigenvalues of H^{k_2}. Therefore, we compute a low-rank approximate of H^{k_2}, say $\tilde{H}^{k_2}$, and build a preconditioner using the pseudo-inverse of $\tilde{H}^{k_2}$. The preconditioner M is defined as

$$M = I + [\tilde{H}^{0^\dagger}, \tilde{H}^{1^\dagger}, \cdots, \tilde{H}^{(n-1)^\dagger}], \tag{3.6}$$

where † represents the pseudo-inverse and the shift by I is to guarantee that M is symmetric and positive definite. A shift by I will not effect the behavior of the preconditioner because H is a compact operator[2]. In the case of observations on the right boundary, the ${k_2}^{\text{th}}$ block of the Hessian $\bar{H}^{k_2} = (-1)^{i+j} H^{k_2}$ where (i, j) correspond the index of the matrix entry and H^{k_2} is given by (3.5). Therefore, the Hessian when the data is given on the left and the right boundaries is $H' = H + \bar{H}$. Similar to (3.6), the preconditioner in the case of data on both the boundaries is defined as

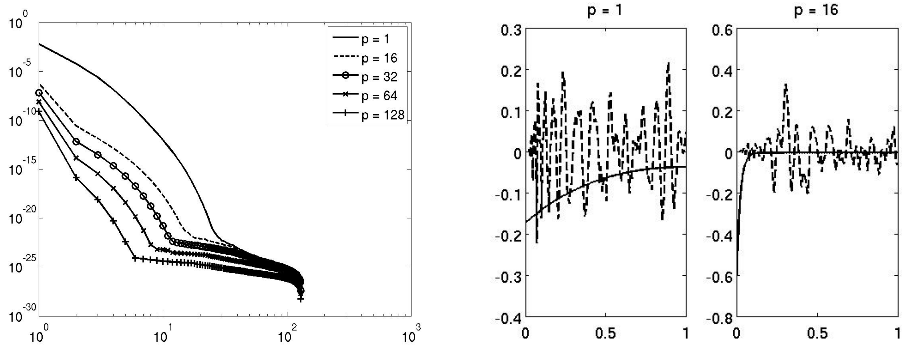


FIG. 3.1. *(Left) Eigenvalues of H^{k_2} for $k_2 = 1, 16, 32, 64, 128$, and (Right) Eigenvectors corresponding to the 1^{st} and the 32^{nd} eigenvalues of H^{k_2} for $k_2 = 1, 16$. The eigenvalues are rapidly decaying in all the cases and the frequency of the eigenvectors is higher for smaller eigenvalues that makes this problem ill-posed. Since the eigenvalues decay rapidly, each block in the Hessian can be approximated by a low-rank matrix corresponding to the first few eigenvalues of the blocks.*

$$M = I + [\tilde{H'}^{0^\dagger}, \tilde{H'}^{1^\dagger}, \cdots, \tilde{H'}^{(n-1)^\dagger}]. \tag{3.7}$$

Below, we discuss the complexity of building the preconditioner and the overall complexity of solving the problem.

Complexity. Complexity of computing the inverse of H^{k_2} is $\mathcal{O}(n^3)$ and the complexity of computing the inverse of H is $\mathcal{O}(n^4)$, which is prohibitively expensive. Instead, we compute $\tilde{H}^{k_2}$, a k-rank approximation of H^{k_2}, where $k \ll n$, using the first k eigenvalues and eigenvectors of H^{k_2}. The complexity of computing the first k eigenvalues and eigenvectors of one block is $\mathcal{O}(k^2 n)$ and computing the same for all the blocks is $\mathcal{O}(k^2 n^2)$. The complexity of applying the preconditioner M defined in (3.6) is $\mathcal{O}(kn^2)$. In Section 6, we show that PCG converges in mesh-independent number of iterations for different choices of u and β using M as the preconditioner. Therefore, the complexity of solving both the forward and inverse problems is $\mathcal{O}(n^2 N_t \log n)$.

[2] If $M = H^{-1}$ then $(M + I)H = I + H$ has a bounded condition number

4. Inverse problem with non-constant coefficient parabolic PDE. In this section, we propose a preconditioner for inverse problems with non-constant parabolic PDE constraints. First, we consider the case in which $b(\mathbf{x},t)$ is non-constant and $a(\mathbf{x},t)=0$. Let $H_N, H \in \mathcal{R}^{n^2\times n^2}$ be the Hessians corresponding to the non-constant and constant coefficient problems respectively. Let $C_N, C \in \mathcal{R}^{n^2N_t\times n^2}$ be the Jacobians of the parabolic PDE with respect to u in the case of non-constant and constant coefficient problems respectively. From (1.2), we have $H_N = C_N^T J^{-T} Q^T Q J^{-1} C_N$ and $H = C^T J^{-T} Q^T Q J^{-1} C$. We approximate $H_N \approx P^T H P$ where $P \in \mathcal{R}^{n^2\times n^2}$ is the minimum norm solution of $\|C_N - CP\|^2$. Since C_N and C are rectangular matrices, $P = (C^TC)^{-1}C^TC_N$ and similarly $P^T = C_N^T C(C^TC)^{-1}$. In the present case, $C^TC = I$ and $P = C^TC_N$. We define the preconditioner in the case of the non-constant coefficient problem M_N as :

$$M_N = P^{-1} M P^{-T}, \tag{4.1}$$

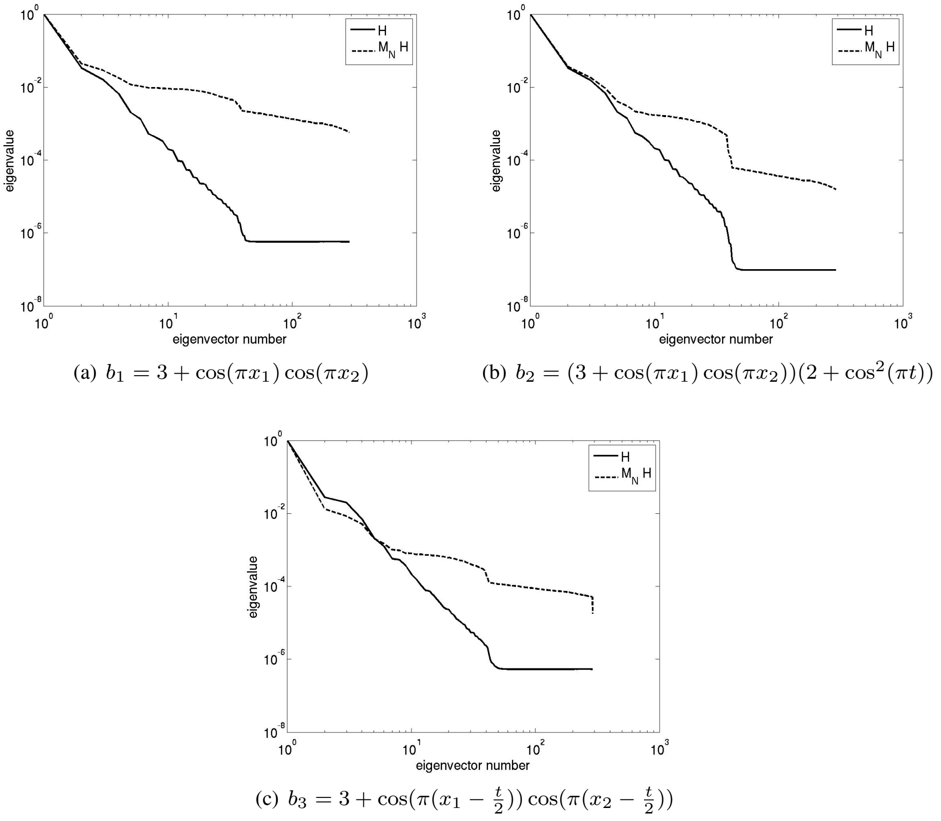
where M is the preconditioner in the case of the constant coefficient problem. We have $C = [I, I, \cdots, I]^T$, and $C_N = [F^T b(\mathbf{x},\delta)F, F^T b(\mathbf{x},2\delta)F, \cdots, F^T b(\mathbf{x},N_t\delta)F]^T$. Since $P = C^TC_N$, $P = F^T \sum_{q=1}^{N_t} b(\mathbf{x},q\delta)F$ and $P \approx F^T \int_0^T b(\mathbf{x},t)dtF$. Therefore, we precompute $\int_0^T b(\mathbf{x},t)$ and define $P^{-1} = F^T \frac{1}{\int_0^T b(\mathbf{x},t)dt} F$. In the case of $b(x,t) = 1(t)$, $M_N = M$.

We consider three cases of $b(\mathbf{x},t)$: 1) $b_1(\mathbf{x},t) = 3 + \cos(\pi x_1)\cos(\pi x_2)$, which is of the form $f(\mathbf{x})1(t)$, 2) $b_2(\mathbf{x},t) = (3+\cos(\pi x_1)\cos(\pi x_2))(2+\cos^2(\pi t))$, which is of the form $f(\mathbf{x})g(t)$, and 3) $b_3(\mathbf{x},t) = 3+\cos(\pi(x_1 - \frac{t}{2}))\cos(\pi(x_2-\frac{t}{2}))$, which is a traveling wave and of the form $f(\mathbf{x},t)$. In Figure 4.1, we show the spectrum of the Hessian (H) and the preconditioned Hessian ($M_N H$) for all the three cases of $b(\mathbf{x},t)$. We observe that the condition number of the preconditioned Hessian is much smaller than the unpreconditioned Hessian in all the three cases. In the case of b_1, which is of the form $f(\mathbf{x})1(t)$, $\|C_N - CP\| = 0$ and hence M_N is a better approximate of H_N^{-1} when compared to M_N in the case of b_2 and b_3 where $\|C_N - CP\| > 0$. In numerical experiments, this effect is more evident from the number of CG iterations with the preconditioner for the three cases.

In the case of non-zero $a(\mathbf{x},t)$, we use the same preconditioner as discussed above in which we set $a(\mathbf{x},t)=0$. In the forward and the adjoint solves, $a(\mathbf{x},t)$ perturbs the Laplacian operator at every time step. Since we assume that $a(\mathbf{x},t)$ is smooth and bounded, the effect of $a(\mathbf{x},t)$ on the spectrum of the Hessian is insignificant. We consider two cases of $a(\mathbf{x},t) := 0, b_3$ a traveling wave. In Figure 4.2, we show the spectrum of the Hessian in the case of $b(\mathbf{x},t) = 1(t)$ for the two cases of a. The effect of non-zero $a(\mathbf{x},t)$ is insignificant on the spectrum of the Hessian and hence we use the same preconditioner as in the case of $a(\mathbf{x},t)=0$.

5. Parallel implementation. In this section, we describe the parallel implementation of the parabolic source inversion problem in 3D. We discuss the implementation of the FFT, Hessian matvec, preconditioner setup, and the preconditioner matvec. Our implementation builds on PETSc [7] to manage parallel data structures, and to interface with linear solvers and preconditioners. We use SLEPc [3] for the computation of eigenvalues and eigenvectors of the blocks of the Hessian to setup the preconditioner.

FFT. We assume that the unit periodic box is divided into $n\times n\times n$ grid and the domain is partitioned into a logically regular grid of p processors. The data is handled by the Distributed Arrays (DA) that are used in conjunction to the PETSc vectors (Figure 5.1). For a detailed description of the DA refer to [7]. We use the FFT libraries developed by Sandia National Lab to perform FFT in parallel [2, 25], which uses the 3D transpose algorithm [18]. This in turn uses FFTW [1] to do FFTs on a single processor.



(a) $b_1 = 3 + \cos(\pi x_1)\cos(\pi x_2)$

(b) $b_2 = (3 + \cos(\pi x_1)\cos(\pi x_2))(2 + \cos^2(\pi t))$

(c) $b_3 = 3 + \cos(\pi(x_1 - \frac{t}{2}))\cos(\pi(x_2 - \frac{t}{2}))$

FIG. 4.1.. *Spectrum of the Hessian (H) and the preconditioned Hessian ($M_N H$) for different cases of* $b(\mathbf{x}, t)$. *In all the three cases, we observed that the condition number of $M_N H$ is smaller than H. In the case of b_1, which is of the form $f(\mathbf{x})1(t)$, $\|C_N - CP\| = 0$ and hence M_N is a better approximate of H_N^{-1} when compared to M_N in the case of b_2 and b_3 where $\|C_N - CP\| > 0$.*

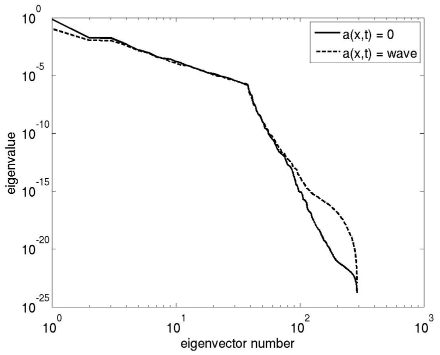


FIG. 4.2. *Spectrum of the Hessian is shown in the case of $b(\mathbf{x}, t) = 1(t)$ for $a(\mathbf{x}, t) = 0$ and a traveling wave (b_3). The effect of non-zero $a(\mathbf{x}, t)$ on the spectrum of the Hessian is insignificant because of the assumption that $a(\mathbf{x}, t)$ is smooth and bounded. Hence, in the case of non-zero $a(\mathbf{x}, t)$ we use the same preconditioner as in the case of $a(\mathbf{x}, t) = 0$ to accelerate the convergence of CG.*

FIG. 5.1. *Data is partitioned into a logically regular grid. This partitioning is handled by the Distributed Arrays (DA), which are used in conjunction to the PETSc vectors. In this figure, we show the data partitioned across eight processors. For the parallel FFT operation, two transpose operations are done using an* all-to-all personalized *communication. In the Hessian matvec, the MPI communicator is divided into subgroups to perform the Q^TQ operation. Here, the 8 processors are divided into 4 groups - $\{0,1\}$, $\{2,3\}$, $\{4,5\}$, and $\{6,7\}$ and the groups are shown in different colors.*

Hessian matvec. In Algorithm 1, we describe the steps involved in the Hessian matvec. Here, we describe the Q^TQ operation in step 2 of the Hessian matvec in more detail. The Q matvec requires summation of $y_{\mathbf{k}}$ over a single index k_1 (3.3). We split the communicator into subgroups so that the summation over k_1 can be done without communication between the groups. If there are p_1, p_2, p_3 processors along x_1, x_2, x_3 directions respectively then p_2p_3 subgroups are formed. In Figure 5.1, the four groups are $\{0,1\}$, $\{2,3\}$, $\{4,5\}$, and $\{6,7\}$. We perform an `MPI_Allreduce` operation within each group to compute the summation over a single index. The Q^T matvec requires stacking of a 2-D vector n times to make a 3-D vector. This is done within each subgroup as every processor in the group has the result of the summation.

Algorithm 1 Hessian matvec ($H\hat{u}_{k_1k_2}$)

```
1: Solve $J^q\hat{y}_{\mathbf{k}}(q\delta) = F^Tb(\mathbf{x}, q\delta)F\hat{u}_{\mathbf{k}}, \quad 1 < q < N_t$        $\mathcal{O}(N_tN\log N)$
2: Solve $J^{T^q}\hat{\lambda}_{\mathbf{k}}(q\delta) = -Q^TQ\hat{y}_{\mathbf{k}}, \quad 1 < q < N_t$        $\mathcal{O}(N_tN\log N)$
3: $H\hat{u}_{\mathbf{k}} = \sum_{q=1}^{N_t} F^Tb(\mathbf{x}, q\delta)F\hat{\lambda}_{\mathbf{k}}(q\delta)$        $\mathcal{O}(N_tN\log N)$
```

Preconditioner. To setup the preconditioner, we need to build the blocks of the Hessian (3.5). Each processor forms $\frac{n^2}{p}$ dense matrices of size $n \times n$, which has a complexity of $\mathcal{O}(\frac{n^4}{p})$. We compute the first k eigenvalues of these blocks in an embarrassingly parallel fashion using SLEPc with a complexity is $\mathcal{O}(\frac{k^2n^3}{p})$. We store the k eigenvalues and eigenvectors of the Hessian blocks locally. In order to apply the preconditioner, we make sure that the indexing of the local data is consistent with the indexing with which the low-rank approximation of the Hessian is evaluated. The indexing with which the vectors are created using the DA (`DAGlobal` in PETSc) and the order in which the Hessian blocks are formed (`DANatural`) is different. Therefore, we perform a scatter operation (`DANaturalToGlobal` and `DAGlobalToNatural`) so that the indexing is consistent for the preconditioner matvec. The complexity of computing the preconditioner matvec is $\mathcal{O}(\frac{kn^3}{p})$. In the case of the non-constant coefficient problem, we precompute $\int_0^T b(\mathbf{x}, t)dt$ and the complexity of applying P^{-1} is $\mathcal{O}(\frac{kn^3\log n}{p})$.

6. Numerical Experiments.

In this section, we present numerical experiments to show the effectiveness of the preconditioners based on the low-rank approximation of the blocks of the Hessian. We consider two source functions for u: $u_1 = \exp(-32((x_1-0.25)^2 + (x_2-0.25)^2) + \exp(-32((x-0.75)^2 + (y-0.75)^2)$, and a discontinuous function u_2 that is a sum of two square shaped step functions (Figure 6.1(e),6.1(f)). In all the experiments, we generate synthetic data by solving the forward problem using second order Crank-Nicholson scheme in time and exact source. We set $\nu = 1, T = 1$, $N_t = n$ and $k = 10$ to build $\tilde{H}^p$ and the preconditioner M in (3.7).

Regularization parameter. We explicitly compute the SVD of the Hessian for $n = 17$ in all the experiments and use these singular values to set β. In order to recover the components of the solution along the first r eigenvectors of the Hessian, we need to take $\beta \leq \sigma_r$ where σ_r is the r^{th} eigenvalue of the Hessian. We set $\beta = \sigma_r \frac{17^2}{n^2}$, where σ_r is the r^{th} eigenvalue of the Hessian for $n = 17$. As we refine the mesh the value of β is reduced by a factor of ≈ 0.25 and the accuracy of the recovered solution is improved. This choice of β agrees with our claim that in mesh-refinement studies β has to be mesh-dependent. If we choose too small a β, it would result in numerical instability because of the ill-posedness of the problem. Hence, the choice of β is very crucial in the study of inverse solvers.

Stopping criterion. If $N_t = \mathcal{O}(n)$ then the error due to discretization is $\mathcal{O}(1/n^2)$. If $\tilde{z}$ is the noisy data then $\|z - \tilde{z}\| \leq \eta$, where η is the Euclidean norm of the noise in the data, and η is $\mathcal{O}(1/n^2)$ due to the spatial and temporal discretization error. We use the discrepancy principle to set a stopping criterion for CG. According to the discrepancy principle [19, 16], we can obtain a solution u such that $\|Ku - \tilde{z}\| \leq \tau\eta$, where $\tau > 1$. Using this principle, we set the stopping criterion to be $\|r\| \leq 2\eta$. In these experiments, the noise is due to the discretization error and we set $\eta = \beta$, so that the β offers a balance between the accuracy of the reconstructed solution and numerical stability.

Results. In Table 6.1, we study the effectiveness of the preconditioner when the data is given on the left and right boundaries for different u and $a(\mathbf{x}, t) = 0, b_3$. We show the number of CG iterations with and without the preconditioner for the constant coefficient case and for $b(\mathbf{x}, t) = b_1, b_2, b_3$. We observed mesh-independent and β-independent number of CG iterations with the preconditioner in the constant coefficient case and b_1. In the case of b_2 and b_3, though the number of CG iterations is not mesh-independent, convergence of CG is accelerated using the preconditioner. Therefore, the overall complexity of solving the inverse problem in the constant coefficient case and b_1 is equal to a constant number of PDE solves independent of the problem size and β. In the case of b_2 and b_3, the inverse solver with the preconditioner is faster than the unpreconditioned Hessian. In Tables 6.2 and 6.3, we show the number of CG iterations for 3D constant coefficient and non-constant coefficient problems respectively. In both the cases, the data is assumed to be given on the planes $x_1 = 0$ and $x_1 = 1$. We observed mesh-independent acceleration of the convergence of CG in both the cases.

In Figure 6.1, we show the reconstructed sources u_1, u_2 using the data on the left and the right boundaries in the case of $a(\mathbf{x}, t) = 0, b_3$ and $b(\mathbf{x}, t) = 1(t), b_3$. We reconstructed the source using the data on the left and the right boundaries. Since we reduce β as we refine the mesh, we have observed a reduction in the L^2 error between the reconstructed solution and the exact source for finer meshes. We have also observed that the error in the reconstructed solution in the constant coefficient case is more than that of the error in the non-constant coefficient case. In Figure 6.2, we show the iso-surfaces of the exact sources used in generating the boundary data and reconstructed source obtained by solving the constant coefficient inversion problem.

7. Discussion. We presented a three-step approach to solve with parabolic source inversion problems. We derived analytical expressions for the entries of the Hessian in the constant coefficient problem and proposed a preconditioner based on the low-rank approximation of each block of the Hessian. We presented numerical experiments in which we show mesh-independent and β-independent convergence of PCG with the preconditioner for the constant coefficient problem. We derived a preconditioner for the non-constant coefficient problem that is based on the preconditioner for the constant coefficient problem. We peformed numerical experiments to show the effectiveness of the preconditioner for different cases of $a(\mathbf{x}, t)$ and $b(\mathbf{x}, t)$. The number of CG iterations in the case of the non-constant coefficient problem is not mesh-dependent though the convergence of CG is accelerated.

¡¡¡¡¡¡¡ parabolic2d.bbl

TABLE 6.1

Effectiveness of the preconditioner based on the low-rank approximation. *The inverse problem is solved for different problem sizes and $b(\mathbf{x}, t)$ in the case of $u = u_1, u_2$, and $a(\mathbf{x}, t) = 0, b_3$. We report the problem size $n \times n$ and the number of CG iterations without a preconditioner (***none***) and with the preconditioner (***preco***). We consider the* **constant coefficient** *problem i.e., $b(\mathbf{x}, t) = 1(t)$ (***const***) and the* **non-constant coefficient** *problem $b(\mathbf{x}, t) = b_1, b_2, b_3$. We set $\beta = \sigma_{30}\frac{17^2}{n^2}$, where σ_{30} is the 30^{th} eigenvalue of the Hessian for $n = 17$. The stopping criterion for CG is either $\|r\| < 2\beta$ or if the number of iterations exceeds 100. In all the cases, the number of CG iterations without a preconditioner increased with increase in n. In the constant coefficient case and in the case of b_1 we observed almost mesh-independent convergence of CG with the preconditioner. In the case of b_2 and b_3 though the number of CG iterations are not mesh-independent we observed an acceleration in the convergence of CG. Therefore, if $b(\mathbf{x}, t)$ is of the form $f(\mathbf{x})1(t)$ the proposed preconditioner results in an almost mesh-independent convergence and if $b(\mathbf{x}, t)$ is of the form $f(\mathbf{x})g(t)$ or $f(\mathbf{x}, t)$ the preconditioner results in an acceleration in the convergence of CG and the number of iterations are mesh-dependent. The number of CG iterations with the preconditioner are independent of the source function u except in the case of $n = 129, a(\mathbf{x}, t) = b_3, b(\mathbf{x}, t) = b_2$, and $u = u_1$ (3^{rd} table).*

$a(\mathbf{x}, t) = 0, u_1$

	const		b_1		b_2		b_3	
$n \times n$	**none**	**preco**	**none**	**preco**	**none**	**preco**	**none**	**preco**
17 × 17	4	2	4	3	4	4	7	6
33 × 33	5	3	6	4	6	4	9	8
65 × 65	6	3	7	4	7	8	15	10
129 × 129	21	4	13	7	21	12	28	16
257 × 257	-	6	44	9	21	17	49	23

$a(\mathbf{x}, t) = 0, u_2$

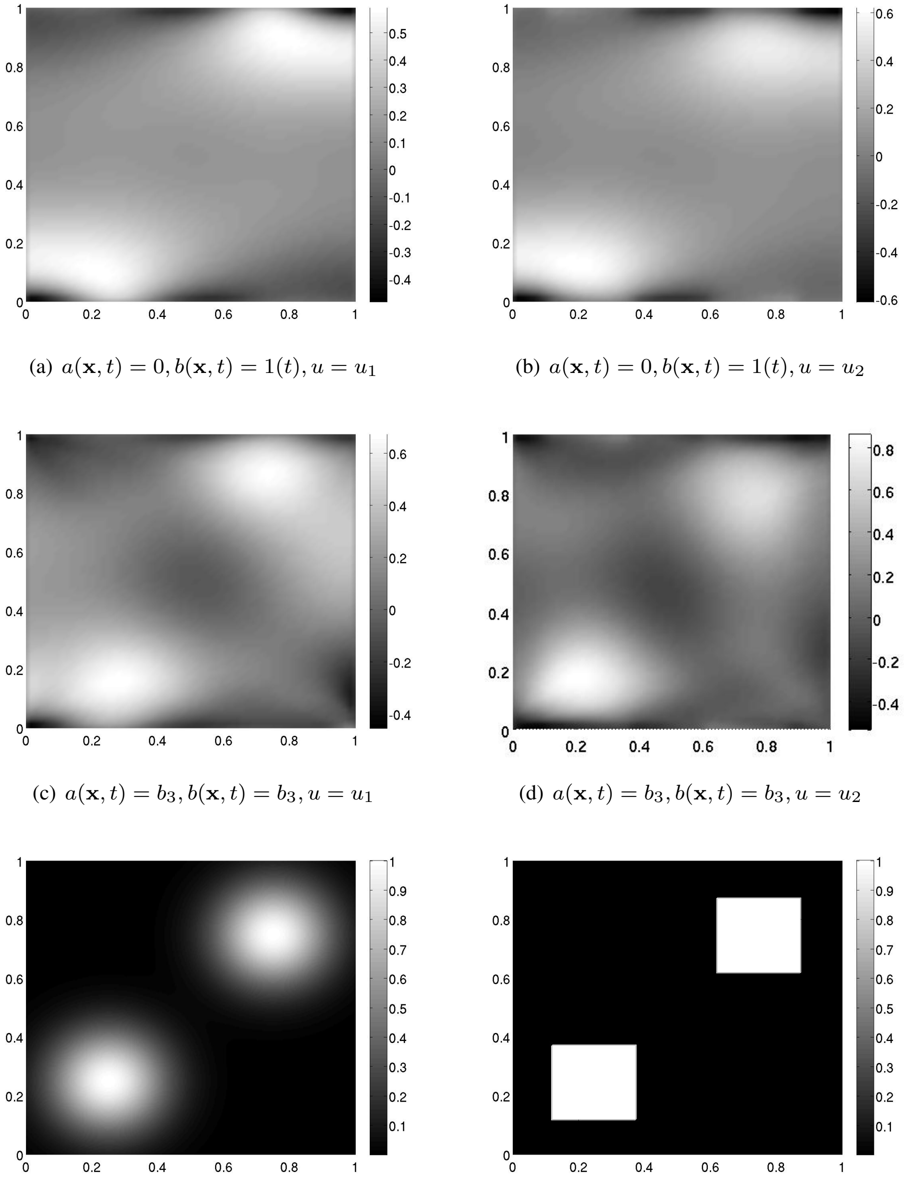
	const		b_1		b_2		b_3	
$n \times n$	**none**	**preco**	**none**	**preco**	**none**	**preco**	**none**	**preco**
17 × 17	8	3	8	6	8	8	9	6
33 × 33	9	4	9	6	9	8	10	8
65 × 65	12	5	14	9	20	12	16	10
129 × 129	39	7	23	10	33	18	28	18
257 × 257	-	8	62	14	34	24	69	25

$a(\mathbf{x}, t) = b_3, u_1$

	const		b_1		b_2		b_3	
$n \times n$	**none**	**preco**	**none**	**preco**	**none**	**preco**	**none**	**preco**
17 × 17	7	10	6	6	6	6	6	6
33 × 33	8	13	9	7	8	8	9	10
65 × 65	13	13	12	13	12	13	17	19
129 × 129	28	18	41	17	45	30	36	28
257 × 257	-	20	95	30	88	54	-	46

$a(\mathbf{x}, t) = b_3, u_2$

	const		b_1		b_2		b_3	
$n \times n$	**none**	**preco**	**none**	**preco**	**none**	**preco**	**none**	**preco**
17 × 17	8	9	8	6	8	6	8	7
33 × 33	11	13	9	9	9	9	14	10
65 × 65	17	13	12	13	12	13	18	15
129 × 129	58	15	39	14	53	13	61	26
257 × 257	-	20	-	32	-	44	-	39



(a) $a(\mathbf{x}, t) = 0, b(\mathbf{x}, t) = 1(t), u = u_1$

(b) $a(\mathbf{x}, t) = 0, b(\mathbf{x}, t) = 1(t), u = u_2$

(c) $a(\mathbf{x}, t) = b_3, b(\mathbf{x}, t) = b_3, u = u_1$

(d) $a(\mathbf{x}, t) = b_3, b(\mathbf{x}, t) = b_3, u = u_2$

(e) u_1

(f) u_2

FIG. 6.1. **Reconstructed solution for different cases of $a(\mathbf{x}, t), b(\mathbf{x}, t)$ and u.** *In all the cases, the sources are reconstructed using the left and the right boundary data on a 65×65 spatial grid and $N_t = 64$. We set $\beta \approx \sigma_{30}/16$ where σ_{30} is the 30^{th} eigenvalue of the Hessian for $n = 17$. As we refine the mesh, we reduce β and hence we recover more frequencies in the solution. We have observed a reduction in the L^2 error between reconstructed solution and the exact source as we refine the mesh. We have also observed that in the case of $a(\mathbf{x}, t), b(\mathbf{x}, t) = b_3$ the reconstructed solutions are more accurate than the constant coefficient case. In the case of $u = u_1$, the L^2 errors are $0.189, 0.169$ for the constant coefficient and the non-constant coefficient problems respectively. Similarly, in the case of $u = u_2$, the errors are $0.296, 0.247$ for the constant and the non-constant coefficient problems respectively.*

TABLE 6.2

*Effectiveness of the low-rank Hessian approximation based preconditioner in the 3D constant coefficient problem. The parabolic source inversion problem is solved for different problem sizes in 3D when $a(\mathbf{x}, t) = 0$, $b(\mathbf{x}, t) = 1(t)$ and $u(\mathbf{x})$ is a Gaussian as shown in Figure 6.2(a). We report the problem size $n \times n \times n$ and the number of CG iterations without a preconditioner (**none**) and with the preconditioner (**preco**). We set $\beta = 10^{-8}\frac{16^2}{n^2}$ so that β is mesh-dependent. The stopping criterion for CG is either $\|r\|/\|r_0\| < 10^{-8}$. In all the cases, the number of CG iterations without a preconditioner increased with increase in n and the convergence of CG with the preconditioner is mesh-independent and β-independent.*

$n \times n \times n$	**none**	**preco**
$16 \times 16 \times 16$	51	10
$32 \times 32 \times 32$	82	10
$64 \times 64 \times 64$	102	11
$128 \times 128 \times 128$	165	10

TABLE 6.3

*Effectiveness of the low-rank Hessian approximation based preconditioner in the 3D constant coefficient problem. The parabolic source inversion problem is solved for different problem sizes in 3D when $a(\mathbf{x}, t)$, $b(\mathbf{x}, t)$ are travelling waves, and $u(\mathbf{x})$ is a Gaussian as shown in 6.2(a). We report the problem size $n \times n \times n$ and the number of CG iterations without a preconditioner (**none**) and with the preconditioner (**preco**). We set $\beta = 10^{-4}\frac{16^2}{n^2}$ so that β is mesh-dependent. The stopping criterion for CG is either $\|r\|/\|r_0\| < 10^{-8}$. In all the cases, the number of CG iterations without a preconditioner increased with increase in n and the convergence of CG with the preconditioner is mesh-independent and β-independent.*

$n \times n \times n$	**none**	**preco**
$16 \times 16 \times 16$	37	24
$32 \times 32 \times 32$	67	28
$64 \times 64 \times 64$	114	30
$128 \times 128 \times 128$		

REFERENCES

[1] *Uniqueness and stable determination of forcing terms in linear partial differential equations with overspecified boundary data*, Inverse Problems, 10 (1994), pp. 1253–1276.

[2] OWE AXELSSON, *Iterative Solution Methods*, Cambridge University Press, 1994.

[3] ALEXANDRA BANEGAS, *Fast poisson solvers for problems with sparsity*, Mathematics of Computation, 32 (1978), pp. 441–446.

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

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

[6] C.D.DIMITROPOULOS AND A.N.BERIS, *An efficient and robust spectral solver for non-separable elliptic equations*, Journal of Computational Physics, 133 (1997), pp. 186–191.

[7] JAMES W COOLEY AND JOHN W. TUKEY, *An algorithm for the machine calculation of complex fourier series*, Mathematics of Computation, 19 (1965), pp. 297–301.

[8] THOMAS DREYER, BERND MAAR, AND VOLKER SCHULZ, *Multigrid optimization and applications*, Journal of Computational and Applied Mathematics, 120 (2000), pp. 67–84.

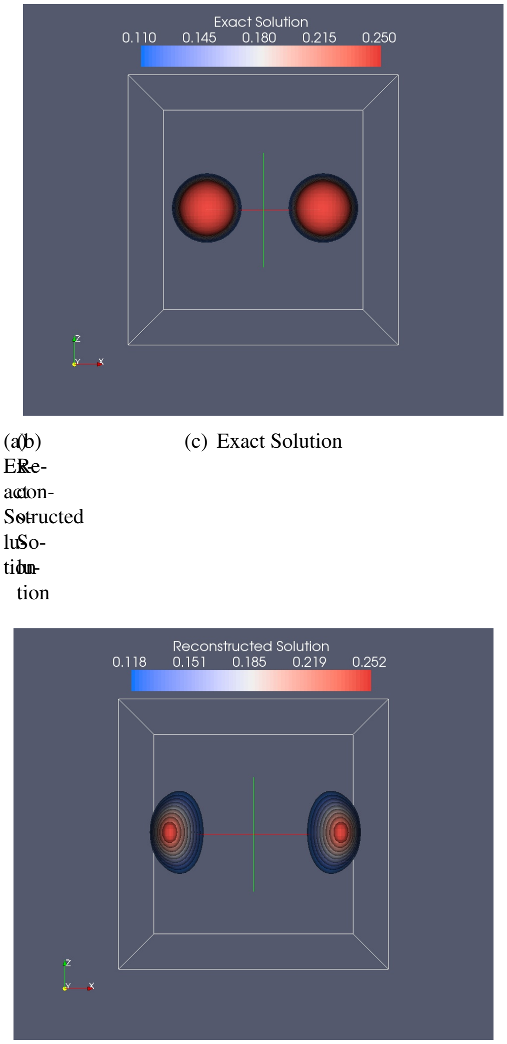
[9] HEINZ W. ENGL, MARTIN HANKE, AND ANDREAS NEUBAUER, *Regularization of Inverse Problems*, Kluwer Academic Publishers, Netherlands, 1996.

[10] U. HAMARIK AND R. PALM, *On rules for stopping the conjugate gradient type methods in ill-posed problems*, Mathematical Modelling and Analysis, 12 (2007), pp. 61–70.

[11] JOHN STRAIN, *Fast spectrally-accurate solution of variable coefficient elliptic problems*, Proceedings of American Mathematical Society, 122 (1994), pp. 843–850.

[12] CURTIS R. VOGEL, *Computational methods for inverse problems*, SIAM, 1987.

[13] M. YAMAMOTO, *Conditional stability in determination of force terms of heat equations in a rectangle*, Mathematical and computer modeling, 18 (1993), pp. 79–88.



(a) Exact Solution (b) Reconstructed Solution (c) Exact Solution

(d) Reconstructed Solution

FIG. 6.2. *In this figure, we show the iso-surfaces of the exact and reconstructed solutions in 3D. The constant coefficient problem is solved for a spatial discretization of* $64 \times 64 \times 64$ *and* 64 *time steps. Though we get an estimate of the location of the source, the width of the reconstructed source is not close to the width of the exact source. This is because of the limited temporal resolution of the boundary data (64 time steps) and discretization noise in the data. The quality of reconstruction will improve if we increase the temporal resolution of the data and reduce the noise in the data.*

=======

REFERENCES

[1] *FFTW Webpage*. http://www.fftw.org/.

[2] *Parallel FFT Webpage*. http://www.sandia.gov/ sjplimp/docs/fft/README.html.

[3] *SLEPc Web page*. http://www.grycap.upv.es/slepc/.

[4] SANTI S ADAVANI AND GEORGE BIROS, *Multigrid algorithms for inverse problems with linear parabolic PDE constraints*, SIAM Journal of Scientific Computing, to appear.

[5] Volkan Akcelik, George Biros, Andrei Draganescu, Omar Ghattas, Judith Hill, and Bart Van Blomen Waanders, *Dynamic data driven inversion for terascale simulations: Real-time identification of airborne contaminants*, Proceedings of the 2005 ACM/IEEE conference on Supercomputing, (2005).

[6] Owe Axelsson, *Iterative Solution Methods*, Cambridge University Press, 1994.

[7] Satish Balay, Kris Buschelman, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Barry F. Smith, and Hong Zhang, *PETSc Web page*, 2001. http://www.mcs.anl.gov/petsc.

[8] Alexandra Banegas, *Fast poisson solvers for problems with sparsity*, Mathematics of Computation, 32 (1978), pp. 441–446.

[9] Benzi, M. and Golub, G. H. and Liesen, J., *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), p. 1.

[10] George Biros and Gunay Dogan, *A multilevel algorithm for inverse problems with elliptic pde constraints*, Inverse Problems, (2008), p. *to appear*.

[11] George Biros and Omar 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.

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

[13] C.D.Dimitropoulos and A.N.Beris, *An efficient and robust spectral solver for non-separable elliptic equations*, Journal of Computational Physics, 133 (1997), pp. 186–191.

[14] James W Cooley and John W. Tukey, *An algorithm for the machine calculation of complex fourier series*, Mathematics of Computation, 19 (1965), pp. 297–301.

[15] Thomas Dreyer, Bernd Maar, and Volker Schulz, *Multigrid optimization and applications*, Journal of Computational and Applied Mathematics, 120 (2000), pp. 67–84.

[16] Heinz W. Engl, Martin Hanke, and Andreas Neubauer, *Regularization of Inverse Problems*, Kluwer Academic Publishers, Netherlands, 1996.

[17] Heinz W Engl, Otmar Scherzer, and Masahiro Yamamoto, *Uniqueness and stable determination of forcing terms in linear partial differential equations with overspecified boundary data*, Inverse Problems, 10 (1994), pp. 1253–1276.

[18] Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, *Introduction to Parallel Computing*, Addison-Wesley, 2003.

[19] U. Hamarik and R. Palm, *On rules for stopping the conjugate gradient type methods in ill-posed problems*, Mathematical Modelling and Analysis, 12 (2007), pp. 61–70.

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

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

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

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

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

[25] P.Dmitruk, L.P.Wang, W.H.Matthaeus, R.Zhang, and D.Seckel, *Scalable parallel fft for spectral simulations on beowulf cluster*, Parallel Computing, 27 (2001), pp. 1921–1936.

[26] John Strain, *Fast spectrally-accurate solution of variable coefficient elliptic problems*, Proceedings of American Mathematical Society, 122 (1994), pp. 843–850.

[27] M. Yamamoto, *Conditional stability in determination of force terms of heat equations in a rectangle*, Mathematical and computer modeling, 18 (1993), pp. 79–88.

¿¿¿¿¿¿¿ 1.5