Two-Level Multiplicative Domain Decomposition Algorithm for Recovering the Lamé Coefficient in Biological Tissues

Si Liu[1] and Xiao-Chuan Cai[2]

[1] Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO 80309, USA, sliu@colorado.edu
[2] Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309, USA, cai@cs.colorado.edu

1 Introduction

Tissue stiffness is one of the qualitative properties to distinguish abnormal tissues from normal tissues, and the stiffness changes are generally described in terms of the Lamé coefficient. In this paper, an all-at-once Lagrange-Newton-Krylov-Schwarz algorithm is developed to solve the inverse problem of recovering the Lamé coefficient in biological tissues. Specifically, we propose and study a multiplicative two-level domain decomposition preconditioner in the inexact Newton step. Numerical experiments are presented to show the efficiency and scalability of the algorithm on supercomputers.

2 Recovering the Lamé Coefficient in Biological Tissues

One of the signs in many diseases is abnormal tissue, of which shear stiffness differs greatly from that of normal tissue. Therefore, it is possible for scientists and engineers to develop new techniques for disease detection and diagnosis through reconstruction of high-resolution images of shear stiffness. In this paper, we focus on the inverse problem derived from transient elastography experiments. Previous work has shown that transient elastography experiments can determine the elastic wave displacement history through scans of the target tissue [2, 3, 5]. Our goal is to identify the Lamé coefficient that describes the shear wave speed or the mechanical stiffness changes inside the target tissue from the elastic wave time-dependent displacement.

The normalized $2D$ scalar wave equation that describes the shear wave displacement has the following form

$$\nabla \cdot (c_0^2 \, \rho \, \nabla d) - d_{tt} = 0, \tag{1}$$

with the boundary condition $\frac{\partial d}{\partial y} = g(t)$ at $y = 0$, where $d(x, y, t)$ describes the local time-dependent displacement inside the tissue, c_0 represents the speed of the background shear wave, and $g(t)$ describes the boundary source. ρ is called the Lamé coefficient, representing the stiffness profile of the tissue. Practically, ρ and d are generally twice continuously differentiable.

Without losing generality, we restrict ourselves to $2D$ domain problems:

$$\Omega = \{(x, y) \in \mathbb{R} \times \mathbb{R}, -6\,(\mathrm{cm}) \le x \le 6\,(\mathrm{cm}), 0\,(\mathrm{cm}) \le y \le 12\,(\mathrm{cm})\}$$

with piecewise smooth boundary $\Gamma = \partial\Omega$ and outer unit normal n. For convenience, the boundary Γ is separated into four pieces and is named as the North ($y = 0$), South ($y = 12$), West ($x = -6$), and East ($x = 6$) boundary, respectively.

We then take the Fourier transform of (1) and obtain the following Helmholtz equation:

$$-\nabla \cdot (\rho \nabla u) - k^2 u = 0, \text{ for } y > 0 \tag{2}$$

with the boundary conditions

$$\begin{cases} \dfrac{\partial u}{\partial r} = f(\tau), & \text{at } y = 0, \\ \lim\limits_{r\to\infty} \dfrac{1}{\sqrt{r}} \left(\dfrac{\partial u_s}{\partial r} - \tilde{i} k u_s \right) = 0, & \text{for the scattered field } u^s, \end{cases} \tag{3}$$

where u and f are the Fourier transform of d and g.

Here, the spatial variable r equals $\sqrt{x^2 + y^2}$. The wave number k equals τ/c_0. u is the total field $u = u^i + u^s$, which is the sum of the incident wave u^i and the scattered field u^s. u^i equals $\frac{1}{\tilde{i}k} f e^{\tilde{i}ky}$, where $\tilde{i}$ represents the imaginary unit $\sqrt{-1}$ throughout this paper.

Furthermore, the experiments of Catheline et al. show that there exists a dominant frequency represented as τ^*, called the central frequency [2, 3, 4]. The largest contribution of the Fourier transform is at this central frequency. Consequently, we evaluate the Eqs. (2) and (3) at the central frequency τ^* and arrive at the following equations:

$$\begin{cases} -\nabla \cdot (\rho \nabla u) - k^2 u = 0, & (x, y) \in \Omega \\ \dfrac{\partial u}{\partial n} = f, & \text{North boundary} \\ \dfrac{\partial u}{\partial n} - \tilde{i} k u = 0, & \text{South boundary} \\ \dfrac{\partial u}{\partial n} - \tilde{i} k u = -f e^{\tilde{i}ky}, & \text{East boundary and West boundary.} \end{cases} \tag{4}$$

We hereby focus on the inverse problem of recovering a high resolution image of the coefficient ρ from the observed data of u and the corresponding boundary conditions in (4).

3 Lagrange-Newton-Krylov-Schwarz Algorithm

To solve the inverse problem of recovering ρ from u, we apply the Tikhonov regularization method and solve the following minimization problem:

$$\text{minimize } J(\rho,u)=\frac{1}{2}\int_{\Omega}|u-z|^2\,d\Omega+\frac{\beta}{2}\int_{\Omega}|\nabla\rho|^2\,d\Omega, \tag{5}$$

where $z(x,y)$ denotes the measured value of $u(x,y)$. This minimization problem is subject to (4), and the H^1-semi-norm is applied as the regularization because of the continuous differentiability of ρ.

To solve this constrained optimization problem, we introduce the Lagrangian functional $\mathcal{L}$ as:

$$\begin{aligned}\mathcal{L}(\rho,u,\lambda)=&\frac{1}{2}\int_{\Omega}(u_1-z_1)^2\,d\Omega+\frac{1}{2}\int_{\Omega}(u_2-z_2)^2\,d\Omega+\frac{\beta}{2}\int_{\Omega}|\nabla\rho|^2\,d\Omega\\&+\int_{\Omega}\rho\nabla u_1\cdot\nabla\lambda_1\,d\Omega-\int_{\Omega}k^2u_1\lambda_1\,d\Omega\\&-\int_{\mathrm{N}}f\rho\lambda_1\,d\Gamma+\int_{\mathrm{S}}ku_2\rho\lambda_1\,d\Gamma+\int_{\mathrm{E\ \&\ W}}(ku_2+fcos(ky))\rho\lambda_1\,d\Gamma\\&+\int_{\Omega}\rho\nabla u_2\cdot\nabla\lambda_2\,d\Omega-\int_{\Omega}k^2u_2\lambda_2\,d\Omega\\&-\int_{\mathrm{S}}ku_1\rho\lambda_2\,d\Gamma-\int_{\mathrm{E\ \&\ W}}(ku_1-fsin(ky))\rho\lambda_2\,d\Gamma,\end{aligned} \tag{6}$$

where u_1 and u_2 represent the real and imaginary parts of u, z_1 and z_2 represent the real and imaginary parts of z, and λ_1 and λ_2 are the corresponding Lagrange multipliers.

The solution of the minimization problem is then obtained by solving the following saddle-point system:

$$\begin{cases}F^{(\rho)}\equiv\dfrac{\partial\mathcal{L}}{\partial\rho}\equiv-\beta\,\Delta\rho+\nabla Re(u)\cdot\nabla Re(\lambda)+\nabla Imag(u)\cdot\nabla Imag(\lambda)=0\\F^{(u)}\equiv\dfrac{\partial\mathcal{L}}{\partial u}\equiv(u-z)-\nabla\cdot(\rho\nabla\lambda)-k^2\lambda=0\\F^{(\lambda)}\equiv\dfrac{\partial\mathcal{L}}{\partial\lambda}\equiv-\nabla\cdot(\rho\nabla u)-k^2u=0.\end{cases} \tag{7}$$

With a finite difference discretization of the saddle-point system and a fully coupled ordering of the variables and the equations, we obtain a large system of nonlinear equations $F(X)=0$ [1]. This system is then solved by an inexact Newton method, and the Newton step is computed by:

$$\begin{aligned}&X_{k+1}=X_k+\xi_k\Delta X_k,\quad k=0,1,...\\&J(X_k)\Delta X_k=-F(X_k),\end{aligned} \tag{8}$$

where X_0 is an initial approximation, $J(X_k)=F'(X_k)$ is the Jacobian matrix at X_k, and ξ_k is the steplength determined by a linesearch procedure.

The generalized minimal residual method (GMRES) is applied to solve the Jacobian system, and the key step is to employ a good preconditioner. In our algorithm, the two-level multiplicative domain decomposition preconditioner is applied as a right preconditioner, and the preconditioning matrix is defined as:

$$M_{mult}^{-1} = A^{-1}[I - (I - AM_{AS}^{-1})(I - AM_c^{-1})(I - AM_{AS}^{-1})], \tag{9}$$

where M_{AS}^{-1} represents the one-level additive preconditioning matrix [1, 6]. M_c^{-1}, equal to $I_c^f A_c^{-1} R_f^c$, is derived from the inverse Jacobian matrix defined on a coarse mesh; I_c^f and R_f^c represent the restriction and interpolation operators.

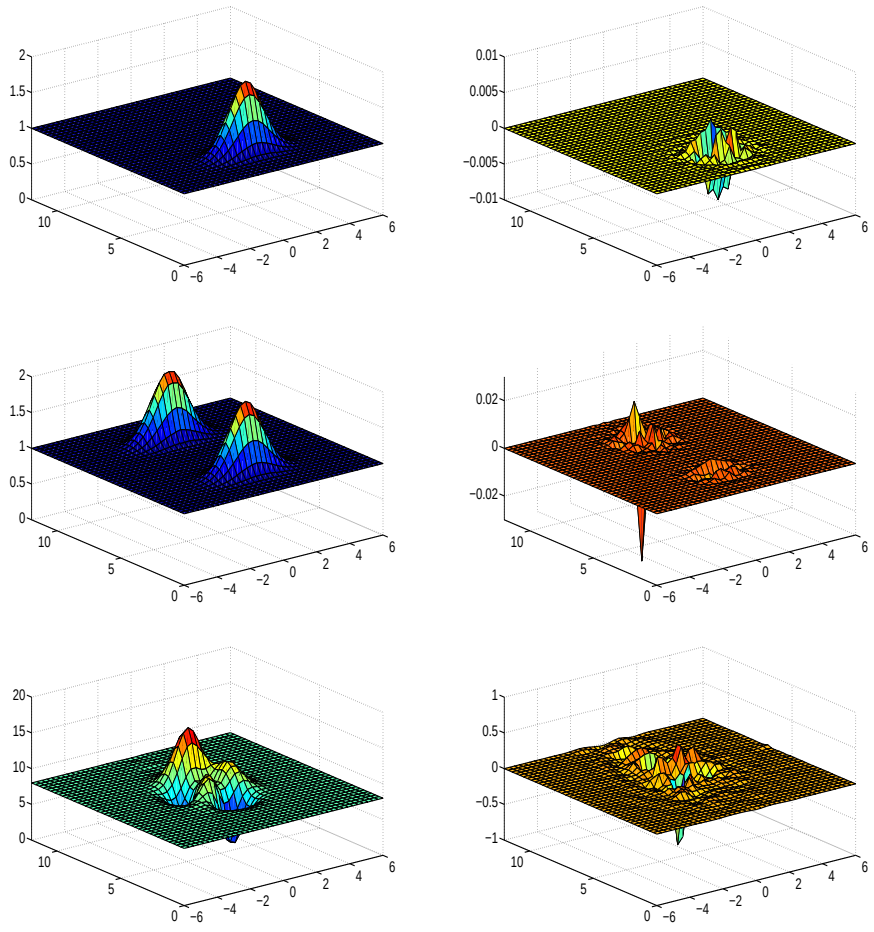

Fig. 1. This figure shows the numerical results of ρ in Test 1, 2, and 3 (from *top to bottom*) with wave number k equal to 8. The *left column* shows the numerical results of ρ, and the *right column* shows the difference between the numerical solutions and the analytic solutions.

4 Numerical Results and Discussion

Three different functions are tested in this paper. In Test 1, the Lamé coefficient to be identified is

$$\rho(x,y) = 1 + exp\left[-2(y-3)^2 - \frac{1}{2}x^2\right].$$

In Test 2, the Lamé coefficient to be identified is

$$\rho(x,y) = 1 + exp\left[-2(y-3)^2 - \frac{1}{2}x^2\right] + exp\left[-2(y-9)^2 - \frac{1}{2}x^2\right].$$

In Test 3, the Lamé coefficient to be identified is

$$\begin{aligned}\rho = {} & 3(1-x)^2 exp\left[-x^2 - (y-5)^2\right] - \frac{1}{3}exp\left[-(x+1)^2 - (y-6)^2\right] \\ & - 10\left[\frac{1}{5}x - x^3 - (y-6)^5\right] exp\left[-x^2 - (y-6)^2\right] + 8.\end{aligned}$$

To test the robustness of the algorithm, random noise is added to the observation data as $z^\delta = z\ (\ 1 + n_\epsilon\ rand(x,y)\ )$. In this paper, the noise level n_ϵ is chosen to be 0 or 1%.

We test our algorithm for different wave numbers, and the algorithm recovers the Lamé coefficient efficiently in all three test problems with wave numbers up to 15. The numerical solutions of ρ when k equals 8 are shown in Fig. 1. We also display the numerical solution of u for Test 3 when k equals 1, 8, or 15 in Fig. 2. This figure demonstrates that our algorithm recovers the Lamé coefficient well from observed data of low frequency, modest frequency, and high frequency.

To test the computing time and scalability of our algorithm, we define the problem on a fine mesh with $1{,}601 \times 1{,}601$ mesh points. The coarse mesh is chosen to be 81×81 or 101×101. When the observed data are only available on a coarse level, we interpolate the observed data to the fine mesh using the bilinear interpolation before we solve the inverse problems.

The number of Newton iterations, the average number of linear iterations, and the computing time are shown in Tables 1 and 2. Since the choice of the coarse mesh only affects the preconditioner, the number of Newton iterations is generally not changed. The number of average linear iterations rises slightly as the number of processors increases. The 101×101 coarse mesh provides more information than the 81×81 one and improves the preconditioner in our two-level algorithm, hereby saving almost 50% of the linear iterations. However, the increasing cost on the coarse level dramatically raises the computing cost per iteration. Therefore, total computing time is not saved and the scalability is worse.

The computing time and the strong scalability of our algorithm are shown in Fig. 3, where the observed data originally come from a 801×801 mesh. Linear and super-linear scalability are achieved for up to 400 processors. When the number of processors exceeds 900, we achieve over 75% scalability of the ideal case.

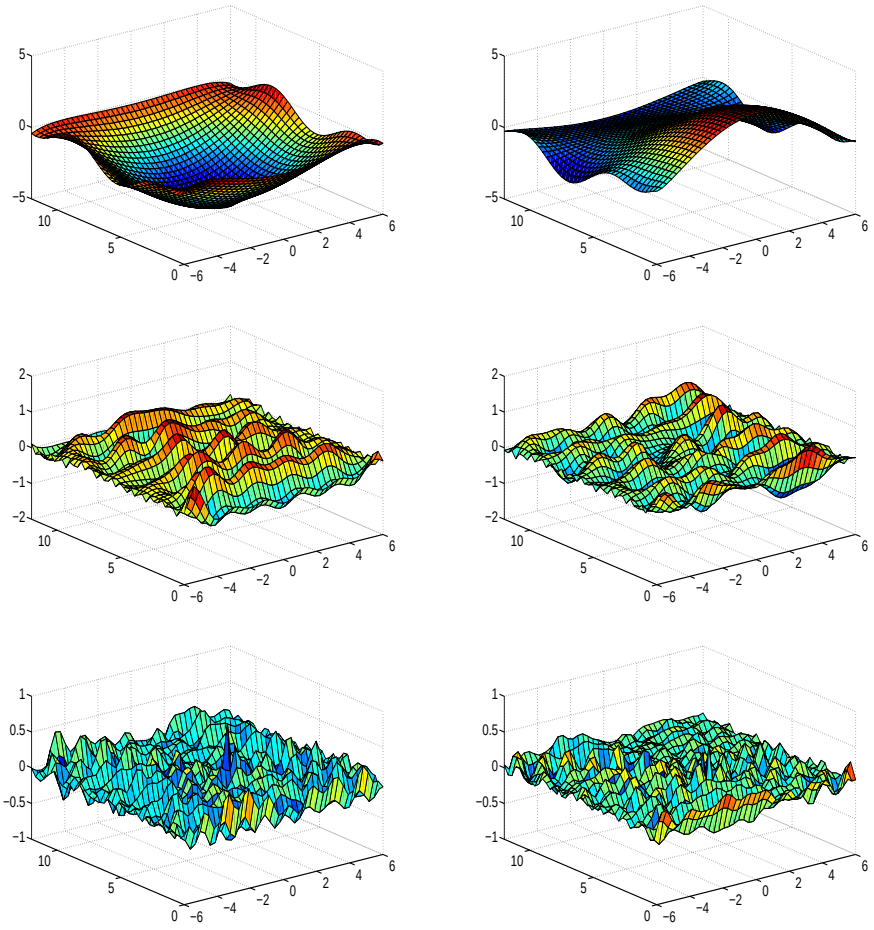

Fig. 2. This figure shows the numerical results of u in Test 3 for wave number k equal to 1, 8, and 15 from *top to bottom*. The *left column* shows the real part of the numerical solution of u, and the *right column* shows the imaginary part of the numerical solution of u.

Table 1. The table shows the numerical results of Test 3, when β equals 10^{-5} and n_ϵ equals 0%. The observed data are measured on a 801×801 mesh.

np	Newton	Average linear	Time(s)	Newton	Average linear	Time(s)
	Coarse mesh: 81×81			Coarse mesh: 101×101		
100	36	81.1	18,005.3	36	40.4	11,767.5
144	36	86.6	13,756.4	36	43.7	9,715.8
256	36	82.1	9,239.0	36	43.8	7,517.2
400	36	88.5	6,590.4	36	44.7	4,326.1
900	36	91.0	4,219.3	36	50.0	3,390.1
1,600	36	83.4	2,276.5	36	59.0	2,742.8

Table 2. The table shows the numerical results of Test 3, when β equals 10^{-5} and n_ϵ equals 1%. The observed data are measured on a 801×801 mesh.

np	Newton	Average linear	Time(s)	Newton	Average linear	Time(s)
	Coarse mesh: 81×81			Coarse mesh: 101×101		
100	38	74.2	17,730.9	38	39.0	12,077.8
144	38	68.9	12,014.1	38	39.6	9,856.3
256	38	70.3	8,573.7	38	41.7	7,664.3
400	38	77.8	6,245.4	38	43.0	4,449.3
900	38	72.8	3,697.6	38	47.4	3,453.4
1,600	38	71.2	2,137.4	38	52.5	2,679.8

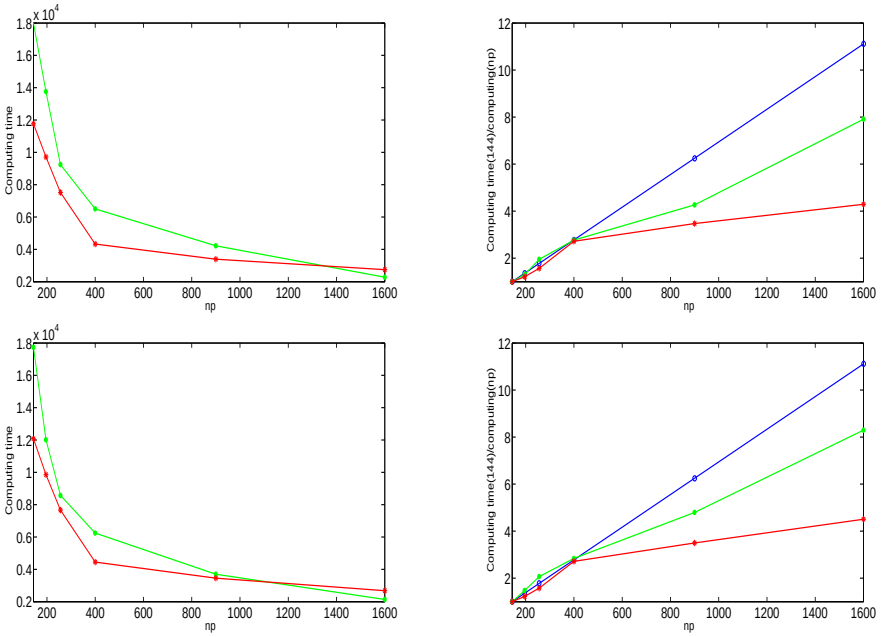

Fig. 3. This figure shows the computing time on the left and speedup curve on the right of Test 3. The noise level n_ϵ equals 0%(*top*) and 1%(*bottom*). $\circ$ represents the scalability of the ideal case. $\bullet$ and $*$ represent the results when the coarse level mesh is 81×81 and 101×101, respectively.

5 Concluding Remarks

In this paper, a two-level multiplicative domain decomposition algorithm is developed to solve this inverse problem of recovering the Lamé coefficient, which is usually difficult, expensive, and noise-sensitive. The algorithm can solve the inverse problem accurately and efficiently, when the observed data have random noise or are only available on a coarse mesh. The algorithm is fairly scalable considering the lin-

ear and nonlinear iteration numbers. Relatively scalable computing time is observed on supercomputers with up to $1{,}600$ processors.

References

1. X.-C. Cai, S. Liu, and J. Zou. Parallel overlapping domain decomposition methods for coupled inverse elliptic problems. *Commun. Appl. Math. Comput. Sci.*, 4:1–26, 2009.
2. S. Catheline, M. Tanter, F. Wu, and M. Fink. Diffraction field of a low frequency vibrator in soft tissues using transient elastography. IEEE Trans. Ultrason. Ferroelectn. Freq. Control, 46(4):1013–1019, 1999.
3. L. Ji and J. McLaughlin. Recovery of the Lamé parameter μ in biological tissues. *Inverse Probl.*, 20(1):1–24, 2004.
4. J. McLaughlin and D. Renzi. Shear wave speed recovery in transient elastography and supersonic imaging using propagating fronts. *Inverse Probl.*, 22(2):681–706, 2006.
5. L. Sandrin, M. Tanter, S. Catheline, and M. Fink. Shear modulus imaging with 2-d transient elastography. IEEE Trans. Ultrason. Ferroelectn. Freq. Control, 49(4):426–435, 2002.
6. A. Toselli and O. Widlund. *Domain Decomposition Methods—Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer, Berlin, 2005.