USCT Image Reconstruction: Acceleration using Gauss-Newton Preconditioned Conjugate Gradient

H. Wang[1], H. Gemmeke[2], T. Hopp[2], and J. Hesser[1]

[1] *Heidelberg University, Mannheim, Germany*
E-mail: hongjian.wang@medma.uni-heidelberg.de
[2] *Karlsruhe Institute of Technology, Karlsruhe, Germany*

Abstract

Ultrasound transmission tomography offers quantitative characterization of the tissue or materials by their speed of sound and attenuation. Reconstruction of such images is an inverse problem which is solved iteratively based on a forward model of the Helmholtz equation by paraxial approximation and thus is time-consuming. Hence, developing optimizers that decrease this time, in particular reducing the number of forward propagations is of high relevance in order to bring this technology into clinical practice. In this paper, we solve the inverse problem of reconstruction in a two-level strategy, by an outer and an inner loop. At each iteration of the outer loop, the system is linearized and this linear subproblem is solved in the inner loop with a preconditioned conjugate gradient (CG). A standard Cholesky preconditioning method based on the system matrix is compared with a matrix-free Quasi-Newton update approach, where a preconditioned matrix-vector product is computed at the beginning of every CG iteration. We also use a multigrid scheme with multi-frequency reconstruction to get a convergent rough reconstruction at a lower frequency and then refine it on a higher-resolution grid. The Cholesky preconditioning reduces the number of CG iterations by approx. 70%~85%; but the computation time for determining the system matrix for the Cholesky preconditioner is dominating, offsetting the gains of the reduction of iterations. The matrix-free preconditioning method saves approx. 30% of the computation time on average for single-frequency and multi-frequency reconstruction. For the robust multi-frequency reconstruction, we test three breast-like numerical phantoms resulting in a deviation of 0.13 m/s on average in speed of sound reconstruction and a deviation of 5.4% on average in attenuation reconstruction, from the ground truth simulation.

Keywords: Ultrasound transmission tomography, Preconditioning, Multigrid reconstruction

1 Introduction

Breast cancer diagnostics based on ultrasound computed tomography (USCT) promises high specificity for early cancer detection. Some 2D or 3D USCT devices [1] measure reflection and transmission tomography at the same time. Transmission tomography offers quantitative characterization of the imaged tissue or materials by speed of sound and attenuation profiles [2]. In this paper, we focus on the image reconstruction of transmission tomography based on the *Karlsruhe USCT* system [1].

For image reconstruction of transmission tomography, we consider the wave equation in the frequency domain. The Helmholtz equation models the wave propagation of ultrasound through an acoustic background medium including refraction, diffraction and multiple scattering as

$$\Delta p + {k_0}^2(1+\eta)^2 p = 0, \tag{1.1}$$

with the frequency dependent pressure field p. The acoustic medium is described by the background wave number $k_0 = \omega/c$ and the refractive index $1+\eta$, where $\omega = 2\pi f$ accounts for the angular frequency for frequency f and speed of sound (SoS) c of the background medium, and η accounts for the deviation of the SoS from the background medium. The full solution of the Helmholtz equation poses a very high computational burden. To mitigate this limitation for medical imaging, the paraxial approximation was chosen [3][4] which is fast enough to be applied for radiological and material diagnosis at the same day. The paraxial approximation describes the ultrasound field via nearly plane waves in forward direction. We use the wide-angle parabolic equation (WAPE) [5] and perform Lie-Trotter splitting [6] to the (formal) solution of the parabolic/paraxial approximation. The forward solution on the computational grid can be calculated by

$$p_{k+1} = e^{i\Delta z k_0 \eta_k} \cdot ifft\left\{ e^{i\Delta z\sqrt{k_0^2-\xi^2}} \cdot fft(p_k) \right\}. \tag{1.2}$$

The index k at p and η denote the considered z slice, whereas the indices for the other directions are omitted. The spectral variable is denoted by ξ and the discrete Fourier transformations are denoted by *fft* and *ifft* in 1D or 2D, whether the problem is 2D or 3D respectively.

The image reconstruction consists of determining the distribution of SoS and attenuation which are both derived from η. To be specific, $\eta = a + i\frac{b}{\omega}$, where $Re(\eta) = a = \frac{c_0}{c} - 1$ describes the deviation in the SoS (c is the SoS in the object and c_0 in the background medium), and $Im(\eta) = \frac{b}{\omega}$ accounts for frequency dependent attenuation. Hence, we need to estimate the parameter η according to the forward model (1.2) given the measurements taken by the USCT system. This is known as an inverse problem usually provided as a nonlinear

least-squares problem, and it is ill-posed due to the nonlinearity and the extremely large scale of the problem. Therefore, we use Newton type methods to linearize the problem together with preconditioned conjugate gradient (CG) method to solve the linearized system at every Newton iteration.

Iterative methods such as CG [8] and generalized minimal residual (GMRES) [14] are widespread to solve a sequence of linear systems that arise in Newton's framework, without forming the Jacobian matrices explicitly while the Jacobian-vector multiplications are approximated by the finite difference method. A challenge is to determine a preconditioner to accelerate the convergence of the Jacobian-free iterative methods. Existing preconditioning methods [15][16] require the full information of the Jacobian for the first outer iteration and the lower or upper triangular part of the Jacobian for each outer iteration. For large problems, it is still expensive to compute the Jacobian even for the first outer iteration. Moreover, if the Jacobian is dense or has some sparse structure, computing the lower or upper triangular part can be as expensive as computing the whole matrix [16]. On the other hand, few existing preconditioning techniques are truly "matrix-free", where neither the full nor the partial system matrix needs to be formed explicitly [17]. One example can be found in [9], where Morales and Nocedal proposed a preconditioner for CG which has the form of a limited memory Quasi-Newton matrix and is generated using information from CG iterations without requiring explicit knowledge of the system (Jacobian) matrix. Xu et al. [17] proposed an inner-iteration preconditioner based on the weighted Jacobi method, and used it for preconditioning the CGLS [18] and BA-GMRES [19] methods in a trust region framework. In this paper, we adopt the Quasi-Newton updating preconditioning techniques [9][10] for CG in our USCT image reconstruction task.

2 Methods

2.1 Gauss-Newton method

The reconstruction is defined by the least-squares problem on the squared L_2 norm of the residual

$$S(\eta) = \frac{1}{2}\|r(\eta)\|_2^2 , \tag{2.1}$$

where vector $r(\eta)\colon \mathbb{C}^N \to \mathbb{C}^M$, called *residual vector*, is given by $r(\eta) = F(\eta) - p_d$. Here, $p_d \in \mathbb{C}^M$ are the measured pressure field from the USCT system, and $F(\eta)\colon \mathbb{C}^N \to \mathbb{C}^M$ is the predicted pressure field computed according to the forward model of (1.2), i.e., $p = F(\eta)$. The unknowns $\eta \in \mathbb{C}^N$ are the parameters that we want to estimate for reconstruction. The derivatives of $S(\eta)$ can be expressed in terms of the *Jacobian* $J(\eta)$, which is the $M \times N$ matrix of the first partial derivatives of the residuals, defined by

$$J(\eta) = \left[\frac{\partial r_j}{\partial \eta_i}\right]_{\substack{j=1,2,\dots,M \\ i=1,2,\dots,N}} = [\nabla r_1(\eta)^T, \nabla r_2(\eta)^T, \dots, \nabla r_m(\eta)^T]^T. \quad (2.2)$$

The gradient and Hessian of $S(\eta)$ can then be expressed as follows:

$$\nabla S(\eta) = \sum_{j=1}^{M} r_j(\eta) \nabla r_j(\eta) = J(\eta)^T r(\eta), \quad (2.3)$$

$$\nabla^2 S(\eta) = \sum_{j=1}^{M} \nabla r_j(\eta)\, \nabla r_j(\eta)^T + \sum_{j=1}^{M} r_j(\eta) \nabla^2 r_j(\eta) = J(\eta)^T J(\eta) + \sum_{j=1}^{M} r_j(\eta) \nabla^2 r_j(\eta). \quad (2.4)$$

To minimize the nonlinear objective function $S(\eta)$ of (2.1), we use the Gauss-Newton (GN) method, which can be viewed as a modified Newton's method [7]. Instead of solving the standard Newton equations $\nabla^2 S(\eta_k) d_k = -\nabla S(\eta_k)$ for a search direction d_k (which can be overdetermined or underdetermined depending on N, M), we solve the following system, i.e. the normal equations, to obtain the search direction d_k^{GN}:

$$J_k^T J_k d_k^{GN} = -J_k^T r_k. \quad (2.5)$$

Here, the use of the approximation $\nabla^2 S_k \approx J_k^T J_k$ relieves us to compute individual residual Hessians $\nabla^2 r_j, j = 1,2, \dots, M$, which are needed in the second term in (2.4).

To solve the linearized system of (2.5), where the system matrix now corresponds to $J_k^T J_k$, we selected the conjugate gradient (CG) method [8] which does not need the system matrix explicitly but only matrix-vector products on both sides of the equation (2.5). Based on the forward model, we can formulate two iterative schemes for the evaluation of the derivative of the forward operator and its adjoint. They are respectively called the *sensitivity* operator F' and the *adjoint* operator F'^* in [4]: the former produces the product of J_k and its input vector while the latter produces the product of J_k^T and its input vector.

After the search direction d_k^{GN} is computed by the CG method, we would like to choose a step length α_k to give a substaintial reduction of S along this direction, but at the same time we do not want to spend too much time making this choice. The ideal choice would be the global minimizer of the univariate function $\phi(\alpha_k) = S(\eta_k + \alpha_k d_k^{GN})$. There is a closed-form solution for α_k if S is a simple quadratic function. In our case, however, S is much more complicated and to find even a local minimizer of ϕ requires many evaluations of S and possibly ∇S. Therefore, we use a practical backtracking line search method [7] to decide the step length α_k along the search direction d_k^{GN}. A general description of the iterative Gauss-Newton method follows.

Algorithm 2.1 (*Gauss-Newton method*)

Given initial solution η_0;
for $k = 0,1,2, \dots$
  Solve $J(\eta_k)^T J(\eta_k) d_k = -J(\eta_k)^T r_k$ for d_k by the conjugate gradient method;
  Find the step length α_k for d_k using line search;
  Set $\eta_{k+1} \leftarrow \eta_k + \alpha_k d_k$;
  if $MSE < GNtol$ **then** stop; **endif** % *MSE stands for mean squared error.*
endfor

2.2 Preconditioners for the conjugate gradient method

For large-scale applications with symmetric system matrices, CG is usually used with preconditioners. A widely used and efficient preconditioner is the incomplete Cholesky preconditioning. The *Cholesky factorization* is a decomposition of a matrix A into the form $A = LL^T$, where L is a lower triangular matrix. Incomplete Cholesky factorization is a variant in which L might be restricted to have the same pattern of nonzero elements as A; other elements of L are ignored [8].

The incomplete Cholesky preconditioning is based on the system matrix while in our forward model computing and storing the Jacobian is expensive both in time and memory consumption. Moreover, when the matrix gets large, factorization is a nontrivial task even though the matrix is sparse. Therefore, matrix-free preconditioning techniques are preferable for our application. An elaborate and recent approach is through limited memory Quasi-Newton update [9][10] given the gradient function without forming the Hessian. This approach approximates the diagonal of the Hessian via Quasi-Newton updates as preconditioners, using information gathered and updated from CG iterations. The preconditioner does not require explicit knowledge of the system matrix and is therefore suitable for our application where only products of the system matrix times a vector can be computed.

Let us denote the problem of (2.5) as $Ax = b$ (i.e., $A = J_k^T J_k$, $x = d_k^{GN}$, $b = -J_k^T r_k$), which we use the preconditioned CG method to solve for the Gauss-Newton search direction x, i.e., d_k^{GN}. It is equivalent to the following optimization problem

$$minimize \quad q(x) = \frac{1}{2}x^T Ax - b^T x. \tag{2.6}$$

The Quasi-Newton matrices are updated by means of the Broyden–Fletcher–Goldfarb–Shanno (BFGS) formula,

$$H^{(j+1)} = H^{(j)} + V_j^T H^{(j)} V_j + \theta_j y_j s_j^T \ , \tag{2.7}$$

where

$$V_j = I - \theta_j y_j s_j^T \ , \ \theta_j = 1/y_j^T s_j \ , \ s_j = x^{(j+1)} - x^{(j)} \ , \ y_j = \nabla q\left(x^{(j+1)}\right) - \nabla q\left(x^{(j)}\right). \tag{2.8}$$

The pair of vectors (s_j, y_j) is called a *correction pair*. Practically, we do not form the matrices $H^{(j)}$, but only store the correction pairs and the scalars θ_j, and then use a recursive formula as described in [11] to compute the product of $H^{(j)}$ and the CG residual vector. A general description of the preconditioned CG method with Quasi-Newton update preconditioner (PREQN) follows.

Algorithm 2.2 (*preconditioned CG method with PREQN preconditioner*)

Given $A_k, b_k, x^{(0)}$; *% Note that k means the number of iteration in the outer Gauss-Newton loop.*
$r^{(0)} = b_k - A_k x^{(0)}$;
for $j = 1,2, \ldots$
  Compute $z^{(j-1)} = H_k r^{(j-1)}$ using PREQN preconditioner routine;
  $\rho^{(j-1)} = r^{(j-1)^T} z^{(j-1)}$;
  if $j == 1$ **then** $p^{(1)} = z^{(0)}$;
  else $\beta^{(j-1)} = \rho^{(j-1)} / \rho^{(j-2)}$; $p^{(j)} = z^{(j-1)} + \beta^{(j-1)} p^{(j-1)}$; **endif**
  $\alpha^{(j)} = \rho^{(j-1)} / p^{(j)^T} A_k p^{(j)}$; $x^{(j)} = x^{(j-1)} + \alpha^{(j)} p^{(j)}$; $r^{(j)} = r^{(j-1)} - \alpha^{(j)} A_k p^{(j)}$;
  if $\left\| r^{(j)} \right\|_2 < CGtol$ **then** stop; **endif**
endfor

In Algorithm 2.2, only one call to the PREQN preconditioner routine is needed for each CG iteration. The routine is summarized as Algorithm 2.3.

Algorithm 2.3 (*PREQN preconditioner*)

Given $k, j, r^{(j-1)}, (s_{j-1}, y_{j-1})$; *% k is the number of iteration in the outer Gauss-Newton loop.*
% j is the number of iteration in the inner CG loop.
if $j > 1$ **then** decide if (s_{j-1}, y_{j-1}) is to be saved; **endif**
if $k == 1$ **then** $z^{(j-1)} = r^{(j-1)}$; *% No preconditioning for CG at the first outer iteration.*
else { **if** $j == 1$ **then** build preconditioner H_{k+1} ; **endif** } $z^{(j-1)} = H_k r^{(j-1)}$; **endif**

2.3 Multi-frequency reconstruction

Convergence for single-frequency reconstruction is only guaranteed if the typical size of the reconstructed object multiplied by the ultrasound frequency is smaller than a given constant [12]. If this condition is not fulfilled, the starting solution must be sufficiently near to the true solution; a simple starting solution like the average SoS in the object fails to converge. Therefore, if we want to reconstruct larger objects at higher frequencies, we can use a multi-frequency method. For each frequency, the problem is solved on a grid resolution that is matching the condition for the used frequency [12]. Firstly, we obtain a convergent rough reconstruction at a lower frequency. From this rough reconstruction with fewer parameters/pixels, we then interpolate a starting solution for the finer reconstruction with more parameters, which corresponds to the reconstruction at higher frequency. This starting solution should be close to the true solution for the high-frequency reconstruction, and hence using it we expect a convergent solution for the high-frequency reconstruction. A general description of our multi-frequency reconstruction follows.

Algorithm 2.4 (*Multi-frequency reconstruction*)

Given initial solution η_0, the starting frequency $f = f_{start}$, the maximum frequency f_{max};
for $i = 1,2, \ldots$
  $\eta_{i-1} = interpolation(\eta_{i-1})$; $\eta_i = reconstruction(\eta_{i-1})$;
  Increase f by $f = f * factor$;
  if $f > f_{max}$ **then** stop; **endif**
endfor

3 Results

In our simulated reconstruction tests, the measurements of pressure field p_d are modeled as $F(\eta_{exact})$ plus additive Gaussian noise characterized by the signal-to-noise ratio (SNR). By minimizing the least-squares problem (2.1), we obtain a solution η that approximates η_{exact}. Considering the robustness of reconstruction with noisy data, we also test Tikhonov regularization where instead of solving (2.5) by CG at every outer iteration (Gauss-Newton iteration), we solve $(J_k^T J_k + \lambda^2 I) d_k^{GN} = -J_k^T r_k$ where I is the identity matrix and λ is the Tikhonov regularization parameter determined by the L-curve method [13].

We firstly test the incomplete Cholesky preconditioning. We use the Matlab build-in function *ichol* to obtain the incomplete Cholesky factorization of the system matrix (Jacobian), as a preconditioner to the Matlab build-in conjugate gradient function *pcg*. We use the sensitivity operator F' to generate the complete Jacobian by giving it a series of input vectors $e_1 = [1,0,\dots]^T, e_2 = [0,1,\dots]^T, \dots, e_N$. Since the computation of the Jacobian is expensive, we do not apply it at every outer iteration, and a preconditioner generated at a given outer iteration is reused for multiple outer iterations. We test five different scenarios denoted as "preconditioner (*re*)", where *re* = {1, 5, 10, 20, and 1000} indicates that the preconditioner is updated after *re* outer iterations. So for example, "preconditioner (5)" means a new preconditioner is generated at every fifth outer iteration. Besides, we also test the situation without preconditioner. For all the six situations, we run four versions of our reconstruction algorithm: (1), (3) no regularization with data of SNR=60dB, SNR=40dB, resp.; (2), (4) Tikhonov regularization with data of SNR=60dB, SNR=40dB, resp. Therefore, we test 24 combinations in total and we report for each combination the sum of outer iterations (S_{outer}), the sum of CG iterations (S_{cg}), and the average CG iterations per outer iteration (S_{mean}), as shown in Figure 1. Note that $S_{mean} = S_{cg}/S_{outer}$. The results show that the incomplete Cholesky preconditioning can reduce about 70%~85% of the CG iterations. It can also reduce outer iterations. The preconditioner update rate, on the other hand, has almost trivial impact on CG iterations.

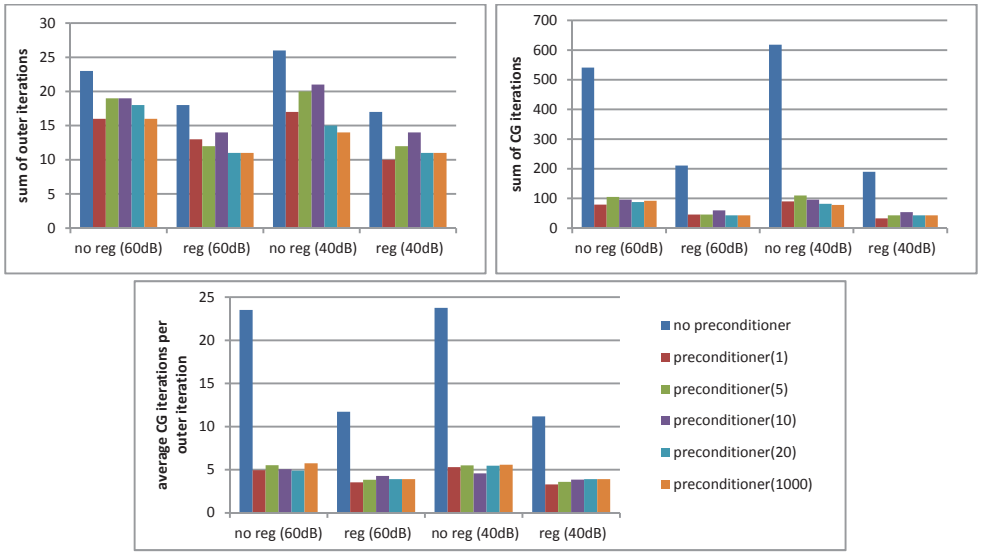

Figure 1: Test results for incomplete Cholesky preconditioning with and without regularization (reg) and at different SNR levels (60dB or 40dB). Upper left: the sum of outer iterations (S_{outer}). Upper right: the sum of CG iterations (S_{cg}). Bottom: the average CG iterations per outer iteration (S_{mean}).

Although the incomplete Cholesky preconditioning reduces CG iterations, it actually does not save computation time in our tests. This is because of the extra computation for forming the Jacobian, especially for large-scale problems where the Jacobian computation is too expensive even though we only have to do it once at the first outer iteration. In contrast, the matrix-free preconditioner PREQN does not need to calculate the Jacobian explicitly. We have rewritten the Fortran PREQN routine[1] [10] into Matlab code and then tested it in our algorithm with several configurations. Specifically, we test six configurations where the main differences are *frequency*, problem *size N* (number of parameters/pixels to be reconstructed), and *CG tolerance* (see the $CGtol$ parameter in Algorithm 2.2). The six tested configurations are:

test1 = (2.5MHz, 48×38, 0.05), test2 = (2.5MHz, 48×38, 0.01),
test3 = (2.5MHz, 96×76, 0.01), test4 = (2.5MHz, 96×76, 0.005),
test5 = (1.5MHz, 104×80, 0.01), test6 = (1.5MHz, 104×80, 0.005).

We set the stop condition of the Gauss-Newton reconstruction (see the $GNtol$ parameter in Algorithm 2.1) to 1e-5. The sum of outer iterations, sum of CG iterations, average CG iterations per outer iteration, and the computation time are reported in Figure 2. The program was executed under *MATLAB R2017a* on a laptop equipped with *Intel Core i7-6700HQ* (4

[1] http://users.iems.northwestern.edu/~nocedal/preqn.html

cores, 2.6GHz) CPU and 16 GB RAM. As shown from the last chart of Figure 2, the PREQN preconditioning does save computation time for all the tests. For smaller problems or larger CG tolerances, the time saved by PREQN preconditioning is not impressive. This implies that the cost of applying preconditioner is similar to the time saved by the reduction in the number of CG iterations. However, for larger problems such as test5 and test6, the PREQN preconditioner saves about 50% computation time. On average, the computation time saved is about 30% for the six tests.

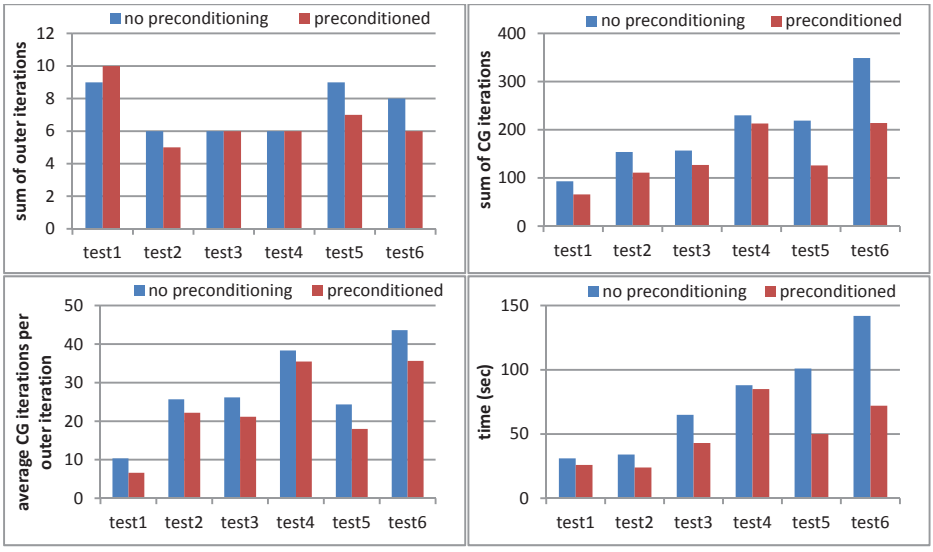

Figure 2: Results of PREQN preconditioning.

The above six reconstruction tests we have reported are for small objects, where the radius of region of interest (ROI) is about 10~20 mm. The radius of measuring device for *Karlsruhe USCT II* is 130 mm, and we have used scaling factors of 0.14 and 0.25 for the previous tests. For reconstruction without scaling, we use the multi-frequency reconstruction method. We start at 250kHz and increase the frequency with a factor of 1.1 each time, reaching the max. frequency 2.5MHz after 26 frequency steps. For all the frequencies before the final one, we use a relatively loose tolerance for reconstruction, in order to get a quick approximate solution which is then interpolated as the starting solution of the next higher frequency. We test multi-frequency reconstruction with three numerical phantoms under *MATLAB R2016a* running on a standard node of *bwForCluster*[2], which is equipped with 2×*Intel Xeon E5-2630v3* (Haswell) (8 cores, 2.4 GHz) CPU and 64 GB Memory. The phantom 1 is a simple simulation of a breast where the background is water, and from outside to inside are skin, fat,

[2] https://www.bwhpc-c5.de/wiki/index.php/Main_Page

gland, and tumor. The phantom 2 is with more shapes allowing identifying sharp edges. The phantom 3 contains the structure of a breast as segmented from a clinical MRI image. The results are reported in Figure 4, where the values are accumulated values of 26 frequencies from 250kHz to 2.5MHz. The PREQN preconditioning reduces both the outer iterations and the CG iterations for multi-frequency reconstruction. The computation time saved is about 30% on average for the three tested phantoms. At the final frequency of 2.5MHz, the number of parameters/pixels is 344×270 with the pixel size of 0.59 mm (also the ultrasound wavelength).

As for visualized results, we report in Figure 3 the Matlab program (running on the laptop used before) snapshots for phantom 3. The multi-frequency reconstruction begins from a starting frequency of 0.5MHz, reaching the final frequency 2.5MHz after 8 frequencies, where the number of parameters/pixels is 110×68 at the final frequency. In Figure 3, the top row 2D images are reconstructed SoS and attenuation respectively. The bottom row 1D profiles focus on the parameters/pixels at the pink dotted lines, where the reconstructed profiles are given in red and their simulated reference in blue. For the 1D profiles, the standard deviation from the simulation for SoS is 0.001 m/s, while the standard deviation for attenuation is 3.4%.

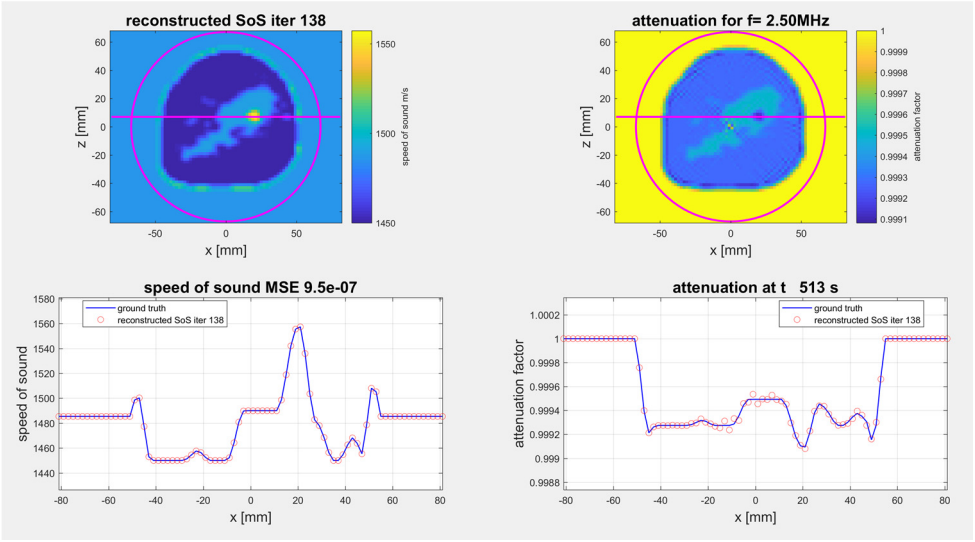

Fig. 3: Visulization of reconstructed speed of sound (blue image) and attenuation (yellow imag) of phantom 3.

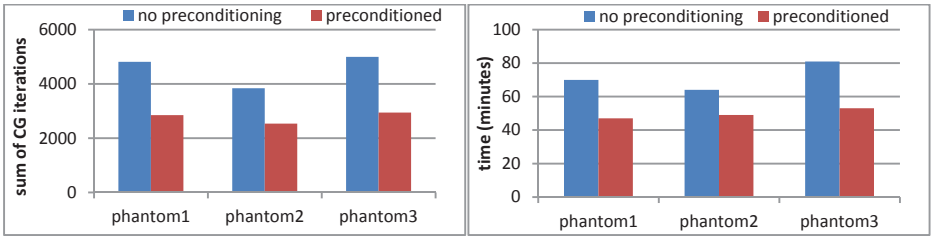

Figure 4: Results of multi-frequency reconstruction with PREQN preconditioning.

4 Conclusions

We use the Gauss-Newton method for USCT image reconstruction by minimizing a nonlinear least-squares problem defined according to measurements data and the forward split-step formulation of the wide-angle parabolic equation (WAPE). The system is linearized in the form of a Jacobian and we choose the conjugate gradient (CG) method to solve the normal equation inside the Gauss-Newton loop, since CG does not need the explicit Jacobian matrix but only matrix-vector products. The commonly used incomplete Cholesky preconditioning for CG can reduce about 70%~85% CG iterations but the computation time for the Jacobian matrix is dominating, and as a result, it fails to reduce the overall computation time significantly. The matrix-free preconditioner via Quasi-Newton update, on the other hand, does not need to form the Jacobian explicitly and saves about 30% of the computation time on average. Reconstruction for large-size problems with high ultrasound frequencies requires starting solutions which are near to the true solutions. We use a multigrid scheme to firstly get a convergent rough reconstruction at lower frequency and then interpolate it for the starting solution with higher frequency. Together with the matrix-free preconditioning, multi-frequency reconstruction gives decent SoS reconstruction for 344×270 parameters at 2.5MHz in reasonable computation time. In future work, more effective problem-dependent preconditioning techniques will be studied for our USCT image reconstruction. Especially for the multigrid reconstruction framework, the non-uniform convergence rates for coarse scale features and fine scale features should be taken into consideration.

Acknowledgements

The research was supported by the Deutsche Forschungsgemeinschaft (DFG) under grants no. HE 3011/37-1 and HO 5565/2-1.

References

[1] H. Gemmeke and N. V. Ruiter, “3D Ultrasound Computer Tomography for medical imaging”, NIM Nucl. Instr. & Meth. In Phys. Res. Vol. Ax, pp. 1057-65, 2007.

[2] J. Greenleaf and R. C. Bahn, “Clinical imaging with transmissive ultrasonic computerized tomography”, IEEE Trans. Biomedical Engineering, vol. 4, no. 4, pp. 177-85, Feb., 1981.
[3] A. Fichtner, “Full seismic waveform modeling and inversion”, Springer, 2011, pp. 86-123.
[4] L. Althaus, “On acoustic tomography using paraxial approximations”, M.S. thesis, Darmstadt University of Technology, Department of Mathematics, Darmstadt, Germany, 2016.
[5] M. D. Feit and J. A. Fleck, “Light propagation in graded-index optical fibers”, Appl.Opt. 17(24), 3990-98, Dec., 1978.
[6] H. F. Trotter, “On the product of semi-groups of operators”, Proc. Am. Math. Soc., 10, 545-551, 1959.
[7] J. Nocedal and S. Wright, “Numerical Optimization (Second Edition)”, Springer-Verlag New York, 2006.
[8] J. R. Shewchuk, “An introduction to the conjugate gradient method without the agonizing pain”, 1994.
[9] J. L. Morales and J. Nocedal, “Automatic preconditioning by limited memory quasi-Newton updating”, SIAM Journal on Optimization 10.4 (2000): 1079-1096.
[10] J. Nocedal, “Algorithm PREQN: Fortran 77 Subroutines for Preconditioning the Conjugate Gradient Method”, (2000).
[11] J. Nocedal, “Updating quasi-Newton matrices with limited storage”, Mathematics of computation 35.151 (1980): 773-782.
[12] F. Natterer and F. Wubbeling, “A propagation-backpropagation method for ultrasound tomography”, Inverse problems 11.6 (1995): 1225.
[13] P. C. Hansen, “Regularization tools: A Matlab package for analysis and solution of discrete ill-posed problems”, Numerical algorithms 46 (2007): 189-194.
[14] Y. Saad and M. H. Schultz, “GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems”, SIMA J. Sci. Stat. Comput. 7, 856–869 (1986)
[15] J. D. Tebbens and M. Túma, “Efficient preconditioning of sequences of nonsymmetric linear systems”, SIAM J. Sci. Comput. 29, 1918–1941 (2007)
[16] W. Xu and T. Coleman, “Efficient (partial) determination of derivative matrices via automatic differentiation”, SIAM J. Sci. Comput. 35, A1398–A1416 (2013)
[17] W. Xu, N. Zheng and K. Hayami, “Jacobian-Free Implicit Inner-Iteration Preconditioner for Nonlinear Least Squares Problems”, Journal of Scientific Computing 68.3 (2016): 1055-1081.
[18] Y. Saad, “Iterative methods for sparse linear systems”, 2nd edn. SIAM, Philadelphia (2003)
[19] K. Morikuni and K. Hayami, “Inner-iteration Krylov subspace methods for least squares problems”, SIAM J. Matrix Anal. Appl. 34, 1–22 (2013)