Electronic version of an article published as J. Computational Acoustics, Vol. 10, No. 2, June 2002, p. 243-264. DOI: http://dx.doi.org/10.1142/S0218396X02001644. © World Scientific Publishing Company.

A Conjugate Gradient-Neural Network Technique for Ultrasound Inverse Imaging

Xiaodong Zhang
Shira L. Broschat

School of Electrical Engineering and Computer Science
Washington State University
PO Box 642752
Pullman, WA 99164-2752
xzhang@eecs.wsu.edu
shira@eecs.wsu.edu

Patrick J. Flynn

Department of Computer Science and Engineering
384 Fitzpatrick Hall
University of Notre Dame
Notre Dame, IN 46556
flynn@nd.edu

August 9, 2001

Abstract. In this paper, a new technique for solving the two-dimensional inverse scattering problem for ultrasound inverse imaging is presented. Reconstruction of a two-dimensional object is accomplished using an iterative algorithm which combines the conjugate gradient (CG) method and a neural network (NN) approach. The neural network technique is used to exploit knowledge of the statistical characteristics of the object to enhance the performance of the conjugate gradient method. The results for simulations show that the CGNN algorithm is more accurate than the CG method and, in addition, convergence occurs more rapidly. For the CGNN algorithm, approximately 50% fewer iterations are needed to obtain the inverse solution for a signal-to-noise ratio (SNR) of 50 dB. For a smaller SNR of 35 dB, the CGNN method is not as accurate, but it still gives reasonable results.

I. INTRODUCTION

Formulation of the inverse scattering problem for ultrasound imaging leads to a nonlinear integral equation. When this integral equation is linearized using the first-order Born or Rytov approximations, the problem reduces to that of conventional diffraction tomography which works well for weak scattering [1]. However, when scattering is strong, the reconstructed image is distorted significantly. Thus, for strong scattering situations, a new approach for solving the nonlinear integral equation must be developed. Several inversion algorithms have been proposed. Chew and Wang used an iterative Born method [2] and a distorted Born iterative method [3] with the former converging more quickly and the latter more robust to noise. Liu *et al.* [4] introduced several variations to the iterative Born method that substantially reduce the computational cost—that is, the computational time required to obtain convergence. However, the iterative Born method is limited to real images and does not work well for reconstruction of complex quantities. Borup *et al.* [5] and Manry and Broschat [6] used a reconstruction technique based on the Gauss-Newton method. Manry and Broschat incorporated a material classification technique which increased the speed of convergence appreciably, but the results were sensitive to noise in the scattered data.

In recent years, several conjugate gradient (CG) based optimization approaches have been investigated. Kleinman and van den Berg proposed a modified gradient method which eliminates the

need to solve the direct scattering problem at each step of the iteration [7], [8]. In the approach of Harada *et al.* a Frechet derivative was used to determine the gradient of the cost functional [9]. Lobel *et al.* successfully applied a conjugate gradient method to several Ipswich experimental data sets [10]. To enhance the noise stability of their algorithm, an edge preservation term was added to the cost functional; this served as a regularization term, allowing smoothing of homogeneous areas while preserving discontinuities [11]. For all this work the CG method was found to be robust to noise in the scattered data.

The robustness of the CG method to noise is encouraging. However, for practical implementation, the computational cost of the CG algorithm must be reduced. Thus, the goal of this work is to accelerate the convergence of the CG method while retaining its noise robustness. The success of the material classification technique proposed by Manry and Broschat [6] in significantly increasing the rate of convergence, as well as the robustness of the CG method to noise, was the motivation for trying a new material classification technique together with the conjugate gradient method. The classifier is based on a neural network, and we refer to the new algorithm as the CGNN method. *A priori* knowledge of the statistics of the ultrasound properties of different materials is used to generate a database which is divided into a training set and a testing set. These, in turn, are used to establish a proper architecture for the neural network. Once the network has been established, it is introduced into the CG method as the last step of an iteration but only after enough iterations have been performed to allow differences in types of materials to have emerged.

To test the CGNN algorithm we apply it to simulations of a small tumor embedded in sections of breast tissue of two different sizes. The size of the first object is $4\lambda \times 4\lambda$, where λ is the wavelength in the background medium. The second object represents a larger section of breast tissue $16\lambda \times 16\lambda$ in size. For both cases, the frequency used is 400 kHz. The performance of the CGNN method in the presence of noise, with signal-to-noise ratios (SNRs) of 50 dB and 35 dB, is also investigated. For the test cases examined, the CGNN method significantly increases the rate of convergence for large SNRs. Approximately 50% fewer iterations are needed to obtain the inverse solution than are needed using the CG method. Moreover the CGNN method retains the robustness to noise of the CG method. In fact, for an SNR of 35 dB, the CGNN method successfully identifies a tumor

as small as 1.8 mm in diameter.

In the next section, formulation of the conjugate gradient-neural network method is given. The geometry and material parameters for the two test cases are described in Sec. III. Numerical simulation results are presented and discussed in Sec. IV. Finally, in Sec. V, concluding remarks are given.

II. Formulation of the Conjugate Gradient-Neural Network Technique

The scattering geometry for the imaging problem is shown in Fig. 1. Imaging is accomplished by insonifying the object using transducers distributed around it. The scattered ultrasound field is then measured around the object at the receiver locations indicated by R. The acoustic wave equation in two dimensions is expressed in integral form as [12]

$$u(\boldsymbol{\rho}) = u_{inc}(\boldsymbol{\rho}) + \int_S \int g(\boldsymbol{\rho} - \boldsymbol{\rho}') o(\boldsymbol{\rho}') u(\boldsymbol{\rho}') d\boldsymbol{\rho}' \tag{1}$$

where S is the surface of the object to be imaged, $\boldsymbol{\rho} = (x, y)$ and $\boldsymbol{\rho}' = (x', y')$ are the coordinates of the observation and source points, respectively, $u(\boldsymbol{\rho})$ is the total acoustic pressure field at $\boldsymbol{\rho}$, $u_{inc}(\boldsymbol{\rho})$ is the incident field at $\boldsymbol{\rho}$, $o(\boldsymbol{\rho})$ is the object function to be reconstructed, and $g(\boldsymbol{\rho} - \boldsymbol{\rho}')$ is the Green's function in two dimensions given by

$$g(\boldsymbol{\rho} - \boldsymbol{\rho}') = \frac{-j}{4} H_0^{(2)}(k_0 |\boldsymbol{\rho} - \boldsymbol{\rho}'|), \tag{2}$$

where $H_0^{(2)}$ is the Hankel function of the second kind, $k_0 = \frac{\omega}{c_0}$ is the acoustic wavenumber in the background medium, c_0 is the sound speed of the background medium, and ω is the angular frequency. The scattered field at a receiver located at $\boldsymbol{\rho}$ is given by

$$u_s(\boldsymbol{\rho}) = \int_S \int g(\boldsymbol{\rho} - \boldsymbol{\rho}') o(\boldsymbol{\rho}') u(\boldsymbol{\rho}') d\boldsymbol{\rho}'. \tag{3}$$

The object function satisfies the relationship $o(\boldsymbol{\rho}) = k_0^2 (\frac{k^2(\boldsymbol{\rho})}{k_0^2} - 1)$, where the term $(\frac{k^2(\boldsymbol{\rho})}{k_0^2} - 1)$ is the contrast of the object. $k(\boldsymbol{\rho})$ is the wavenumber at $\boldsymbol{\rho}$; it is related to the frequency f, attenuation α, and sound speed c by [13]

$$k(\boldsymbol{\rho}) = \frac{2\pi f}{c(\boldsymbol{\rho})} - j\alpha(\boldsymbol{\rho}). \tag{4}$$

The Method of Moments [14], with pulse-basis functions and point matching, is used to transform Eqs. (1) and (3) into matrix form

$$\vec{u}_{inc} = \Gamma O \vec{u} \tag{5}$$

$$\vec{u}_s = KU\vec{o}, \tag{6}$$

respectively. Γ is an $N \times N$ matrix and K is an $M \times N$ matrix; $\vec{u}_{inc}$ is the incident field vector of length N; $\vec{u}$ and $\vec{o}$ are the two unknown vectors of length N; $\vec{u}_s$ is a vector of length M; O and U are $N \times N$ diagonal matrices containing $\vec{o}$ and $\vec{u}$, respectively—*i.e.*, $O_{ii} = \vec{o}_i$ and $U_{ii} = \vec{u}_i$; N is the total number of cells into which the object is discretized; and M is the total number of receivers per view. The elements in Γ and K are given by

$$\Gamma_{ij} = \int_{S_j}\int g(\boldsymbol{\rho}(i) - \boldsymbol{\rho}')d\boldsymbol{\rho}'$$

where $\boldsymbol{\rho}$ is defined over the entire domain of the object, and

$$K_{ij} = \int_{S_j}\int g(\boldsymbol{\rho}(i) - \boldsymbol{\rho}')d\boldsymbol{\rho}'$$

where $\boldsymbol{\rho}$ is defined only at the receivers. S_j is the area of the jth cell.

When the object vector $\vec{o}$ is given, the total field vector $\vec{u}$ can be determined uniquely from Eq. (5); this is the forward scattering problem. When both the scattered field vector $\vec{u}_s$ and the total field vector $\vec{u}$ are given, the object function $\vec{o}$ cannot be obtained directly from Eq. (6) since it is ill-posed; this is the inverse scattering problem. For the forward scattering problem, the BICGSTAB-CG-FFT method [15],[16] was used to solve Eq. (5) for $\vec{u}$. The solution for the inverse scattering problem is described below.

A. The Conjugate Gradient Technique

The conjugate gradient method is an iterative optimization approach. To utilize CG optimization, a cost functional must be formed. For our problem, the cost functional is chosen to measure the difference between the measured and scattered data when an estimate of the object is used. The expression for this cost functional is given by [9]

$$\Phi(o) \;=\; \frac{1}{2} \cdot w \cdot \sum_{l=1}^{L} \| \, \vec{u}_s(l) - \vec{u}_m(l) \, \|^2$$

$$= \frac{1}{2} \cdot w \cdot \sum_{l=1}^{L} \| \vec{r}_l \|^2 \tag{7}$$

where $\| \cdot \|$ indicates the Euclidean norm, and $r_l = \vec{u}_s(l) - \vec{u}_m(l)$. L is the total number of views, and $\vec{u}_m(l)$ is the measured scattered field for illumination by the lth source. The w in Eq. (7) is a weighting factor chosen to insure that the amplitude of the incident field has no impact on the cost functional. Here the weight used is [7]

$$w = (\sum_{l=1}^{L} \| \vec{u}_m(l) \|^2)^{-1}.$$

The main equations used in the CG method are:

$$\vec{d_1} = -\vec{g}_1, \tag{8}$$

$$\vec{o}_{n+1} = \vec{o}_n + t_n \vec{d_n}, \tag{9}$$

$$\vec{d}_{n+1} = -\vec{g}_{n+1} + \gamma_n \vec{d_n}, \tag{10}$$

where $\vec{d_n}$ is the update direction of the object at the nth iteration and $\vec{d_1}$, the initial direction, is chosen to be the steepest descent direction as specified by Eq. (8); t_n is the step size along the direction $\vec{d_n}$ which in this work is found using an inexact line search with a strong Wolfe condition [17]; γ_n is chosen to insure that the direction vector is conjugate to all previous directions; and $\vec{g}_n$ is the gradient of the cost functional at the nth iteration and is obtained using

$$\vec{g}_n = w \sum_{l}^{L} \left[\mathrm{diag} \left(\mathcal{L}(O) \vec{u}_{inc} \right) \mathcal{L}(O) \right]^* K^\dagger \vec{r}_l^{\,n} \tag{11}$$

where $\mathcal{L}(O) = (I - \Gamma O)^{-1}$, $[\cdot]^*$ indicates the conjugate of $[\cdot]$, $K^\dagger$ is the conjugate transpose of the matrix K defined in Eq. (6), and $\vec{r}_l^{\,n}$ is the difference between the measured and calculated scattered data at the nth iteration when the object is illuminated by the lth source. The first-order approximation of the gradient,

$$\vec{g}_n = w \sum_{l}^{L} K^\dagger \vec{r}_l^{\,n},$$

is useful for modest phase shifts. Since Case 1 has a phase shift of less than π, the first-order approximation can be used, while Case 2, with its larger phase shift, requires use of Eq. (11).

There are a number of different choices for γ_n. In this work, we use the one most commonly used known as Polyak-Polak-Ribiere (PPR) [18],[19]

$$\gamma_n^{PPR} = \frac{< \vec{g}_{n+1}, \vec{g}_{n+1} - \vec{g}_n >}{\| g_n \|^2} \tag{12}$$

where $< \cdot >$ denotes the inner product. If the difference between the calculated scattered field and the measured scattered field satisfies a predetermined value for the relative residue error (RRE), convergence is considered to have been obtained, and the iterations are terminated. The relative residue error is defined by

$$RRE = \sqrt{\frac{\sum_{l=1}^{L} \| \vec{u}_s^n(l) - \vec{u}_m(l) \|^2}{\sum_{l=1}^{L} \| \vec{u}_m(l) \|^2}} \tag{13}$$

where the summation is over all the sources. The superscript n indicates the nth iteration.

The steps for solving the inverse scattering problem are summarized as follows:

Step 1: Set $n = 1$ and set the initial guess of the object to be $\vec{o}_1$. Then

- calculate the corresponding scattered field using Eqs. (5) and (6),
- calculate $\vec{g}_1$ using Eq. (11) or its first-order approximation,
- set $\vec{d}_1 = -\vec{g}_1$,
- obtain t_1 using an inexact line search,
- update the object $\vec{o}$ using Eq. (9), and
- increment n by 1.

Step 2: Using Eq. (11), calculate the gradient vector $\vec{g_n}$ and use Eq. (12) to obtain γ_n. Then

- use Eq. (10) to find the update direction $\vec{d_n}$,
- find the corresponding step size t_n, and
- update the object using Eq. (9).

Step 3: Solve the forward scattering problem using the updated object $\vec{o}_n$ in Eqs. (5) and (6), and check to see if the RRE satisfies the required criterion. If yes, terminate the iterations; if not, continue.

Step 4: Increment n by one; go to Step 2.

B. Three-Layer Backpropagation Neural Network Classifier

While the CG method is accurate and robust to noise, it is also computationally expensive—that is, convergence to an accurate solution requires significant CPU time. One approach to decreasing the time required to obtain a solution is to accelerate the convergence of the CG algorithm so that fewer iterations are needed. In this work, a neural network classifier, introduced between Steps 2 and 3, is used to increase the rate of convergence. The neural network is a supervised classifier which uses a database consisting of labeled patterns. This database is divided into training and testing sets. The training set is used to establish the neural network architecture, and the testing set is used to test the resultant architecture to insure that it provides the optimum classification.

Multilayer neural networks can be used to approximate almost any function and can also be applied to pattern classification problems [20]. The most widely used multilayer network is the three-layer configuration shown in Fig. 2. When a pattern, denoted by vector $[p_1, p_2, \ldots, p_R]$, is applied to this network, every feature of the pattern, *i.e.*, $p_i, i = 1, \ldots, R$, is weighted by the weight matrix W^1 of the first layer consisting of values determined during the training stage. The neuron, represented by a circle enclosing a "+" sign, sums all the weighted features and passes the sum to a filter. The output of the filter $b = f(a)$ is a function of its input a, where the filter function f is known as the activation function. The result of the first layer is the input to the next layer and so on. The activation function may differ for different neurons, and the number of neurons for each layer may vary as well. The final output of the entire network for a given input pattern is compared with predetermined classification values. These values are determined in accordance with the desired degree of classification accuracy. Euclidean distance between estimated and true material parameters is used to perform classification. When the distance between the final result of the network and one of the possible classes is less than its classification value, we label the pattern to be that class. If this label is the same as that of the class to which the pattern actually

belongs, we say the network correctly classified the pattern $[p_1, p_2, ..., p_R]$; otherwise, we say that the network misclassified the pattern. To avoid misclassification of materials, we introduce an unclassified category; when the final result of the network is larger than all the classification values, the material is not classified.

The goal of the training stage is to create a weight matrix that will maximize the number of patterns that are correctly classified. A number of methods can be used to solve this problem, but the one most widely used, and the one used in this work, is backpropagation [20]. The advantage of backpropagation is that the chain rule can be used to easily calculate the derivatives that determine the sensitivities of the layers. The sensitivity is then used to update the parameters of the neural network architecture that yield the weight matrix. The sensitivity at layer m is computed from the sensitivity at layer $m + 1$—hence the name backpropagation.

To apply the backpropagation algorithm, the activation function must be differentiable. We have chosen to use the Log-Sigmoid function given by

$$f(x) = \frac{1}{1 + e^{-x}} \tag{14}$$

because of its simplicity. In Eq. (14), x is the sum of the weighted inputs to the neuron.

Neural network training is an iterative optimization procedure. Two parameters have a strong impact on its convergence, the learning rate and the momentum. The learning rate η is the step size taken along the descent direction. A small learning rate insures stability, but the rate of convergence is slow. Increasing the learning rate improves the rate of convergence, but it also leads to instability. A large learning rate can be used with the addition of the momentum parameter which maintains stability by reducing oscillations occurring in the vicinity of a minimum. It achieves this by relating changes in the current iteration to changes in the previous iteration. The learning rate and momentum are chosen to insure fast convergence with stability. In this project, the neural network tool kit in MATLAB was used for the training process. For details on the training process, see [21].

The training stage is implemented after the configuration of the neural network has been specified and the number of neurons at each layer is known. For a given network configuration, the optimal weight matrix is unique so that different configurations result in different weight matrices.

The final architecture—that is, both the network configuration and weight matrix—is chosen by trial and error. Specifically, each network architecture with its associated weight matrix is used to classify the test set, and the percentage of misclassifications, according to the Euclidean distance between the output of the network and the labels of the test data, is calculated. The smaller the percentage of misclassification, the better the network. The network architecture giving the smallest percentage of misclassification is chosen as the final architecture.

The motivation for our development of the CGNN method is the early detection of breast tumors. Thus, for this work we generate the training database using statistical knowledge of the ultrasound parameters for the different tissues in the breast. It is assumed that the sound speeds and attenuation coefficients are independent, that the mean values for each tissue are known, and that they satisfy normal distributions. The standard deviation for the sound speed in each tissue is set to 2% of its mean value [22], and the standard deviation for the attenuation is chosen to be 10% of its mean value. Beyond this, fat and glandular tissue attenuation values start to overlap; it is assumed that no significant overlap in tissue attenuation values occurs [6]. This assumption is generally valid for tumors which have high attenuation relative to normal tissues. The data for each tissue in the database are generated using a Gaussian random number generator implementing the statistical information above.

The neural network obtained after training is used to perform classification at each iteration of the conjugate gradient method. Between Steps 2 and 3 of the CG method, the reconstructed contrast is fed into the neural network pixel by pixel, with each pixel treated as an individual pattern and its real and imaginary parts as two features which serve as the elements of the input pattern. When classification is possible, the network labels a pixel to be a specific type of material. The real and imaginary parts for that pixel are then set to the true values of the corresponding material. When a pixel is left unclassified, the reconstructed contrast remains unchanged. After the entire image has been labeled, it is used as the current object for the next iteration.

In this study the training stage resulted in a $2 - 7 - 2$ neural network architecture—that is, 2 neurons for the first layer, 7 neurons for the second layer, and 2 neurons for the output layer. As mentioned previously, the MATLAB neural network tool kit was used for the training. The

initial value for the learning rate was chosen empirically to be 0.08 and was updated by the neural network tool kit internally during the training process. The momentum was chosen empirically to be 0.7 and was not changed during the training process.

III. Test Cases

To assess the CGNN algorithm, we applied it to two test cases. For Case 1, we reconstructed a section of tissue with a small breast tumor 0.9 mm in diameter surrounded by glandular tissue with a total phase shift of approximately 0.17π. The real and imaginary parts of the contrast of the tissue model for Case 1 are shown in Fig. 3. Case 2 represents a larger section of glandular tissue with a tumor 1.8 mm in size. The total phase shift is approximately 1.6π. Figure 4 shows the real and imaginary parts of the contrast for Case 2. Figures 3 and 4 clearly demonstrate why it is difficult to use the real part to distinguish between the tumor and glandular tissue and why the imaginary part must be used. The sound speed for the glandular tissue was $c = 1520$ m/s and the attenuation was $\alpha = 0.80$ dB/cm/MHz. The sound speed for the tumor was $c = 1564$ m/s, and the attenuation was $\alpha = 1.18$ dB/cm/MHz [22]. For the background medium, the sound speed of water, $c_0 = 1490$ m/s, was used and the source frequency was chosen to be $f = 400$ kHz.

For the first case, the object to be imaged is defined inside a square domain with dimensions $4\lambda \times 4\lambda$, where λ is the wavelength in the background medium. The square domain is discretized into 16×16 cells. Each model is successively illuminated by $L = 16$ plane waves equally spaced around the object, and the scattered field is measured along the four edges for each source rotation. The total number of receivers per view is $M = 64$ and these are spaced 1 mm apart. Case 2 is $16\lambda \times 16\lambda$ in size, which is discretized into 64×64 cells. It is illuminated by $L = 64$ plane waves, and there are $M = 256$ receivers per view. For both cases the forward scattering problem is solved using the BISTAB-CG-FFT method. The algorithm terminates either when the RRE is less than 10^{-5} or when twenty iterations have been completed for the first case and 100 iterations for the second. If the algorithm terminates with the first condition, convergence is considered to have occurred.

The computational cost for the CGNN method consists mainly of two parts, the CG iterations

and the neural network classification. The number of operations for one complete iteration is given by

$$NOp = C_1 LN \log_2 N + C_2 N \tag{15}$$

where C_1 is a constant associated with the CG method, C_2 is a constant associated with the neural network classifier, N is the total number of cells, and L is the number of illuminations. The first term represents the CPU time for the CG iterations and the second term the CPU time for the classification. Clearly the cost of the iterations dominates the total cost. For example, for Case 2 with $N = 64 \times 64$ and $L = 64$ the cost of the CG iterations is on the order of 10^6 operations; the cost of the neural network classifier is on the order of 10^3. Thus, introduction of the neural network classifier does not significantly affect the CPU cost per iteration.

IV. Simulation Results

Computer simulations were performed for the two test cases, both with and without noise. By noise, we mean that Gaussian white noise was added to the real and imaginary parts of the synthetic scattered data. The signal-to-noise ratio (SNR) is defined by

$$SNR = 10 \log \frac{\|\vec{u}_s\|^2}{\|\vec{n}\|^2} \tag{16}$$

where $\|\vec{u}_s\|$, as given by Eq. (6), with the true object is used to simulate the exact measured data. The Euclidean norm of $\|\vec{u}_s\|$ is obtained after a summation over receivers and sources. $\vec{n}$ denotes the random noise, $\|\vec{n}\|^2 = 2LM\sigma^2$, where σ^2 is the variance used to generate the Gaussian white noise, L is the total number of sources, and M is the total number of receivers per source.

In the examples that follow, results for the imaginary part of the object for both the CGNN and CG methods are compared. The boundary between the breast and the background material is used to choose the initial values. For pixels within the boundary the average of the tumor and glandular tissue parameters is used; for pixels outside the boundary, the true value of water is used. The neural network classifier is introduced into the algorithm after visual inspection indicates that differences in tissue types have started to emerge. Simulations for the first case were performed on an HP9000 workstation with 256 MB memory, while simulations for the second case were performed on a Beowulf commodity parallel processor with 16 CPUs and 4 GB of memory.

A. Simulations for Case 1

For the noiseless example, the CGNN results are shown in Fig. 5 and the CG results in Fig. 6. The relative residue error (RRE) curves for both methods are shown in Fig. 7 and are identical until the 8th iteration when the neural network classifier is applied. As shown in Fig. 5, the tumor is detectable at the 9th iteration with the CGNN method, and at the 10th iteration the entire image is correctly classified. In contrast, after 20 iterations the CG method has still not converged to the correct reconstruction. The CGNN method requires 50% fewer iterations and 46% less CPU time and gives more accurate results than the CG method.

For the $SNR = 50$-dB example, the neural network classifier is applied at iteration 10. The reconstructed image, not shown here, is correctly classified immediately after the neural network has been applied. As shown in Fig. 8, the RRE values for the CGNN algorithm are the same from the 10th iteration to the 20th iteration. Thus, for this example, the CGNN algorithm is very stable.

For the $SNR = 35$-dB example, the reconstructed images for the CGNN and CG methods are shown in Figs. 9 and 10, respectively. For this case, the CG method almost fails to reconstruct the small tumor. For the CGNN method, the neural network classifier is applied at the 13th iteration, and the tumor is detected immediately afterward. At 20 iterations, one pixel has been misclassified, and two pixels have been left unclassified. The CGNN results do not improve after the 16th iteration, but they are much better than those of the CG method, providing much higher visual contrast.

B. Simulations for Case 2

Next we consider a much stronger scattering situation. For Case 2, the phase shift is 1.6π. For the noiseless example, the reconstruction results for the CGNN and CG methods are shown in Figs. 11 and 12, respectively. The RRE curves for both methods are shown in Fig. 13. For the CGNN algorithm, the neural network classifier is applied at the 20th iteration. The tumor is correctly classified, but several pixels of glandular tissue are misclassified as tumor. By the 60th iteration, the entire image is correctly classified. For Case 2 the RRE curves decrease very slowly compared with those of Case 1. One possible explanation is that Case 2 is more ill-posed than Case 1. The

CG results shown in Fig. 12 are much noisier than those of the CGNN method. A comparison of the RRE curves in Fig. 13 shows that the CG method requires more than double the number of iterations to obtain the same RRE. The CGNN method uses 59% less CPU time than the CG method to achieve comparable results.

When system noise of $SNR = 50$ dB is added, neither the CGNN nor the CG method performs as well. For the CGNN method, the neural network classifier is again applied at the 20th iteration. The reconstructed image is shown in Fig. 14 for 20, 60, and 100 iterations. Clearly at the 60th iteration, the tumor is classified correctly, but some pixels of glandular tissue are misclassified. Some of these misclassified pixels are actually corrected by the 100th iteration. The reconstructed images for the CG method are shown in Fig. 15, and the RRE curves for both methods are shown in Fig. 16. The RRE value at the 35th iteration for the CGNN algorithm is about the same as the RRE value at the 90th iteration for the CG algorithm. The CGNN method uses 60% fewer iterations than the CG method.

When system noise of $SNR = 35$ dB is added, both the CGNN and CG methods break down, and the overall reconstructed results, not shown, are poor. However, with the neural network classifier applied at the 40th iteration, the tumor is still detectable with the CGNN method.

V. Summary

In this paper, a neural network (NN) classification technique has been introduced into the conjugate gradient (CG) iterative imaging algorithm. The neural network classifier exploits knowledge of the statistics of the acoustic properties of materials to increase the rate of convergence of the CG iterative solution. The neural network is applied to the CG algorithm after different material types have started to emerge. The CGNN method has been tested on two simulations of breast tissue, one with a phase shift of 0.17π and the other with a phase shift of 1.6π, both with and without system noise. The results have been compared with those of the CG method. The CGNN results are more accurate at approximately 50% of the cost of the CG method. At an SNR level of 50 dB, the CGNN method correctly reconstructs and classifies the objects. At a lower level SNR of 35 dB, some pixels are misclassified and others are left unclassified. Nonetheless, minimal tumors

(about 1 mm in diameter) are still identified with the CGNN method. Thus, the CGNN technique seems promising for ultrasound inverse imaging. It increases the rate of convergence of the CG method, it is fairly robust to noise, and it can be used when weak scattering methods do not apply.

VI. Acknowledgments

This work was supported by the National Science Foundation, Grants No. ECS-9711051 and EIA-9818212, and by the Carl M. Hansen Foundation.

REFERENCES

[1] A.C. Kak and M. Slaney, *Principles of Computerized Tomographic Imaging*. New York: IEEE Press, 1988, pp. 203-273.

[2] Y.M. Wang and W.C. Chew, "An iterative solution of two-dimensional electromagnetic inverse scattering problem," *Int. J. Imaging Syst. Technol.*, vol. 1, pp. 100-108, 1989.

[3] W.C. Chew and Y.M. Wang, "Reconstruction of two-dimensional permittivity distribution using the distorted Born iterative method," *IEEE Trans. Medical Imaging*, vol. 9, pp. 218-225, June 1990.

[4] L.J. Liu, X. Zhang, and S.L. Broschat, "Ultrasound imaging using variations of the iterative Born technique," *IEEE Trans. Ultrasonics, Ferroelectrics, and Freq. Control*, vol. 46, no. 3, pp. 574-583, May 1999; vol. 45, no. 5, pp. 1331-1333, Sep. 1999.

[5] D.T. Borup , S.A. Johnson, W.W. Kim and M.J. Berggren, "Nonperturbative diffraction tomography via Gaussian-Newton iteration applied to the scattering integral equation," *Ultrasonic Imaging*, vol. 14, pp. 69-85, 1992.

[6] C.W. Manry, Jr. and S.L. Broschat, "Inverse imaging of the breast with a material classification technique," *J. Acoust. Soc. Am.*, 1998.vol. 103, no. 3, pp. 1538-1546, Mar. 1998.

[7] R.E. Kleinman and P.M. van den Berg, "A modified gradient method for two-dimensional problems in tomography," *J. Computational and Applied Mathematics*, vol. 42, pp. 17-35, 1992.

[8] R.E. Kleinman and P.M. van den Berg, "Inverse scattering—a nonlinear optimization approach," ftp : $//poseidon.math.udel.edu/pub/Wave_Center/Reports/96$.

[9] H. Harada, D.J.N. Wall, T. Takenaka, and M. Tanaka, "Conjugate gradient method applied to inverse scattering problem," *IEEE Trans. Antennas Propagat.*, vol. 43, pp. 784-792, no. 8, Aug. 1995.

[10] P. Lobel, R. Kleinman, Ch. Pichot, L. Blanc-Feraud, and M. Barlaud, "Conjugate gradient method for solving inverse scattering with experimental data," *IEEE Antennas Propagat. Magazine*, vol. 38, no. 3, pp. 48-51, Jun. 1996.

[11] P. Lobel, Ch. Pichot, L. Blanc-Feraud, and M. Barlaud, "Conjugate gradient algorithm with edge-preserving regularization for image reconstruction from Ipswich data for mystery objects," *IEEE Antennas Propagat. Magazine*, vol. 39, no. 2, pp. 48-51, Apr. 1997.

[12] P.N.T. Wells, *Ultrasonics in Clinical Diagnosis*. Edinburgh, New York: Churchill Livingstone, 1977.

[13] E.L. Madsen, H.J. Sathoff, and J.A. Zagzebske, "Ultrasonic shear wave properties of soft tissues and tissue-like materials," *J. Acoust. Soc. Am.*, vol. 74, pp. 1346-1355, Nov. 1983.

[14] R.F. Harrington, *Field Computation by Moment Methods*. New York: MacMillan, 1968.

[15] H.A. van der Vost, "BI-CGSTAB: A fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems," SIAM *J. Sc. Stat. Comput.*, vol. 13, no. 2, pp. 631-644, 1992.

[16] D.T. Borup and O.P. Gandhi, "Calculation of high-resolution SAR distributions in biological bodies using the FFT algorithm and conjugate gradient method," *IEEE Trans. Microwave Theory and Techniques*, vol. 33, no. 5, pp. 417-419, May 1985.

[17] M. Al-Baali, "Descent property and global convergence of the Fletcher-Reeves method with inexact line search," IMA *J. Numer. Anal.*, vol. 5, pp. 121-124, 1985.

[18] B.T. Polyak, "The conjugate gradient method in extremem problems," *USSR Comput. Math. and Math. Phys.*, vol. 9, pp. 94-112, 1969.

[19] E. Polak and G. Ribiere, "Note sur la convergence de methode de directions conjuguees," *Revue Francaise d'Informatique et de Recherche Operationnelle*, vol. 16, pp. 35-43, 1969.

[20] J. Mao and A.K. Jain, “Neural networks for feature extraction and multivariate data projection,” *IEEE Trans. Neural Networks*, vol. 6, pp. 296-317, 1995.

[21] J.M. Zurada, *Introduction to Artificial Neural Network Systems*. Boston: PWS Publishing Company, 1992.

[22] E.L. Madsen, J.A. Zagzebske, G.R. Frank, J.F. Greenleaf, and P.L. Carson, “Anthropomorphic breast phantoms for assessing ultrasonic imaging system performance and for training ultrasonographers: Part two,” *J. Clinical Ultrasound*, vol. 10, pp. 91-100, Mar. 1982.

LIST OF FIGURES

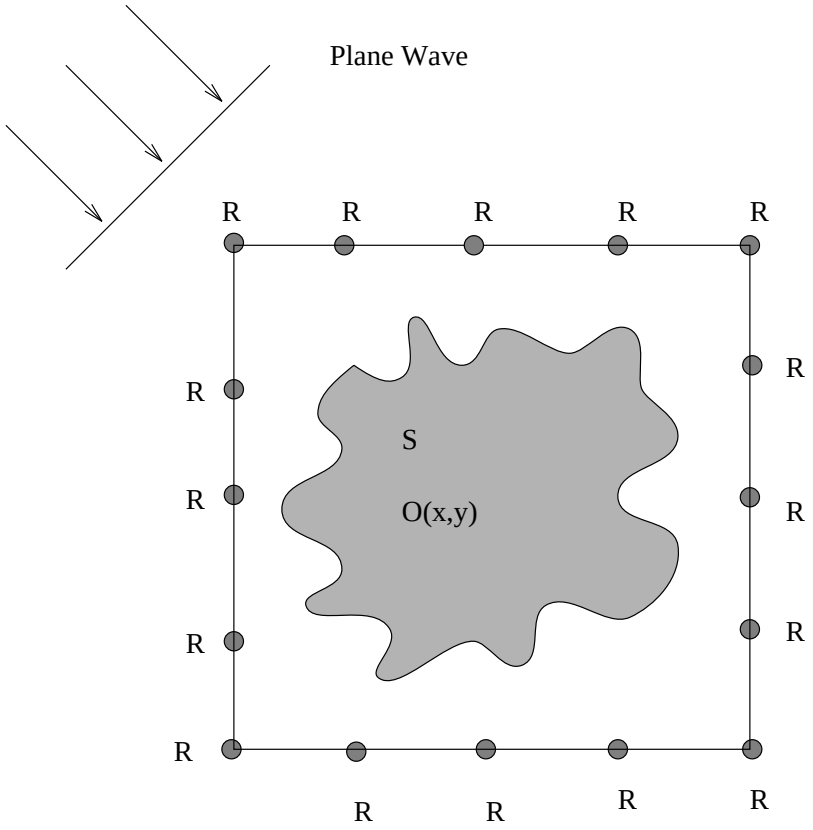

Fig. 1. Scattering geometry.

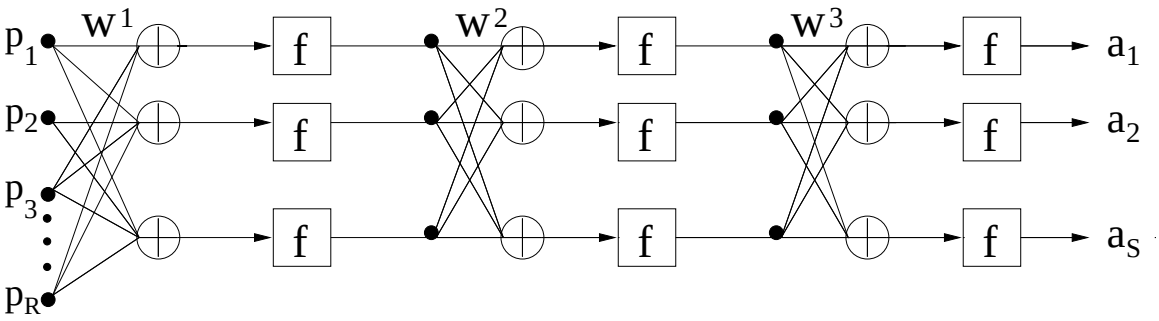

Fig. 2. Three-layer neural network.

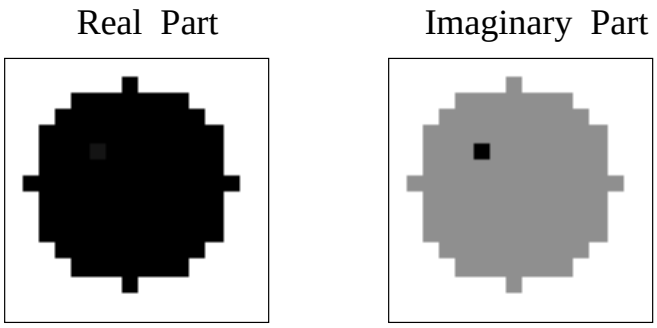

Fig. 3. Real and imaginary parts of the model for Case 1.

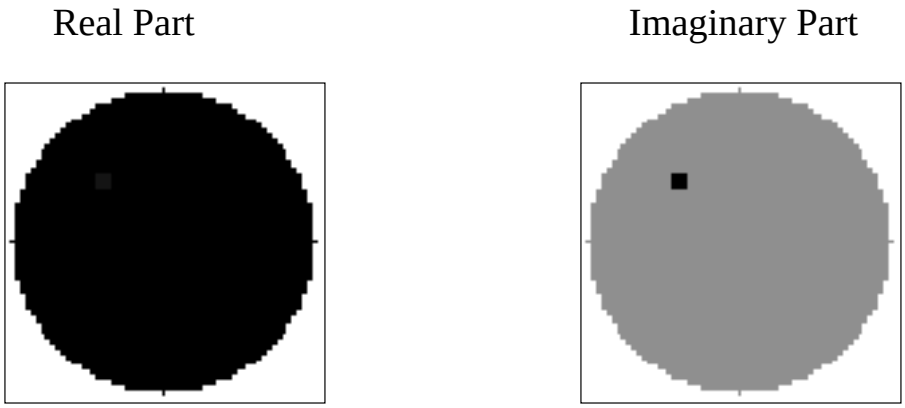

Fig. 4. Real and imaginary parts of the model for Case 2.

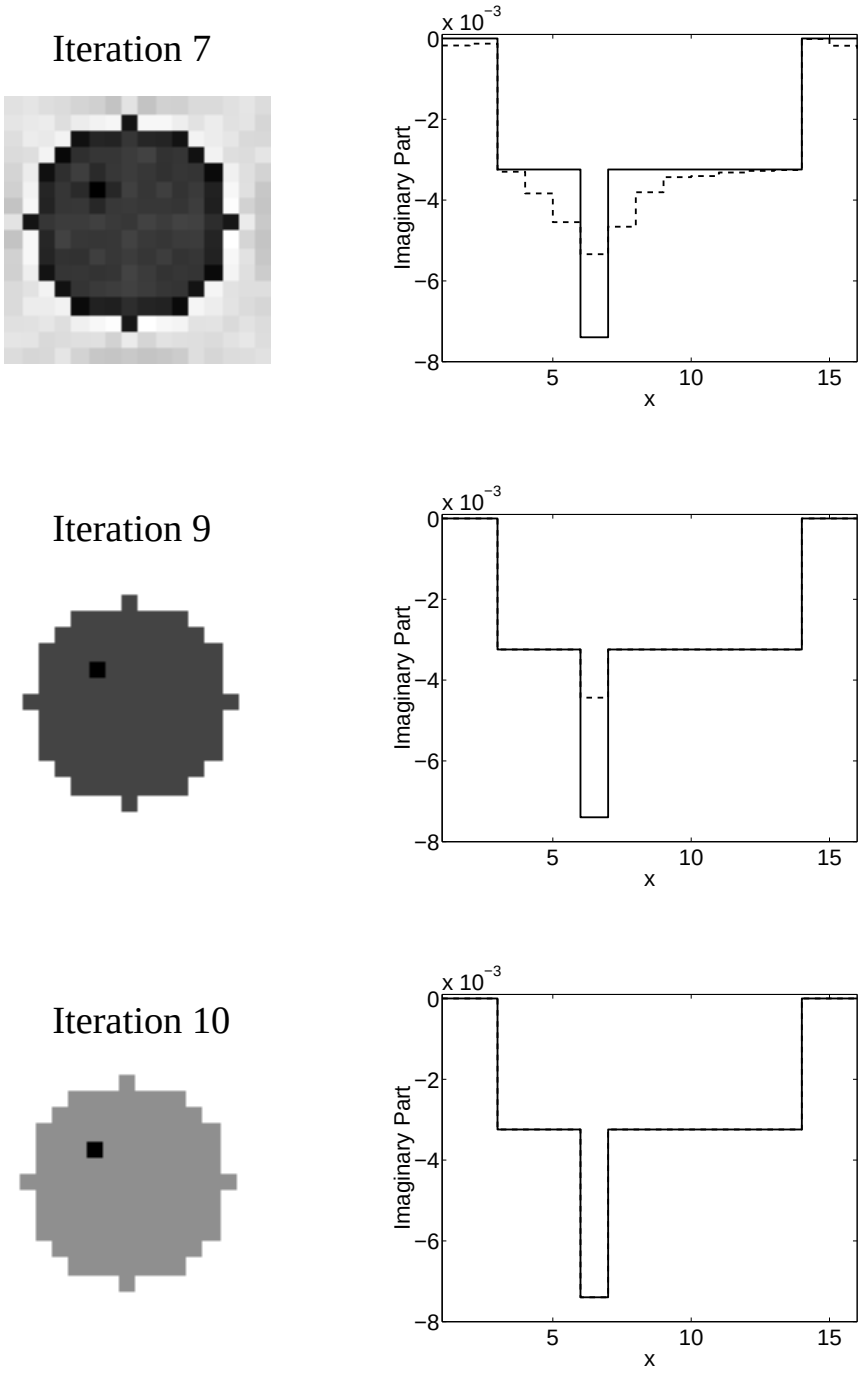

Fig. 5. Reconstructed results for the CGNN method for noiseless data (Case 1). The left column shows reconstructions at Iterations 7, 9 and 10. In the right column the solid lines are the true contrast along the line bisecting the tumor; the dashed lines are the reconstructed values for Iterations 7, 9, and 10.

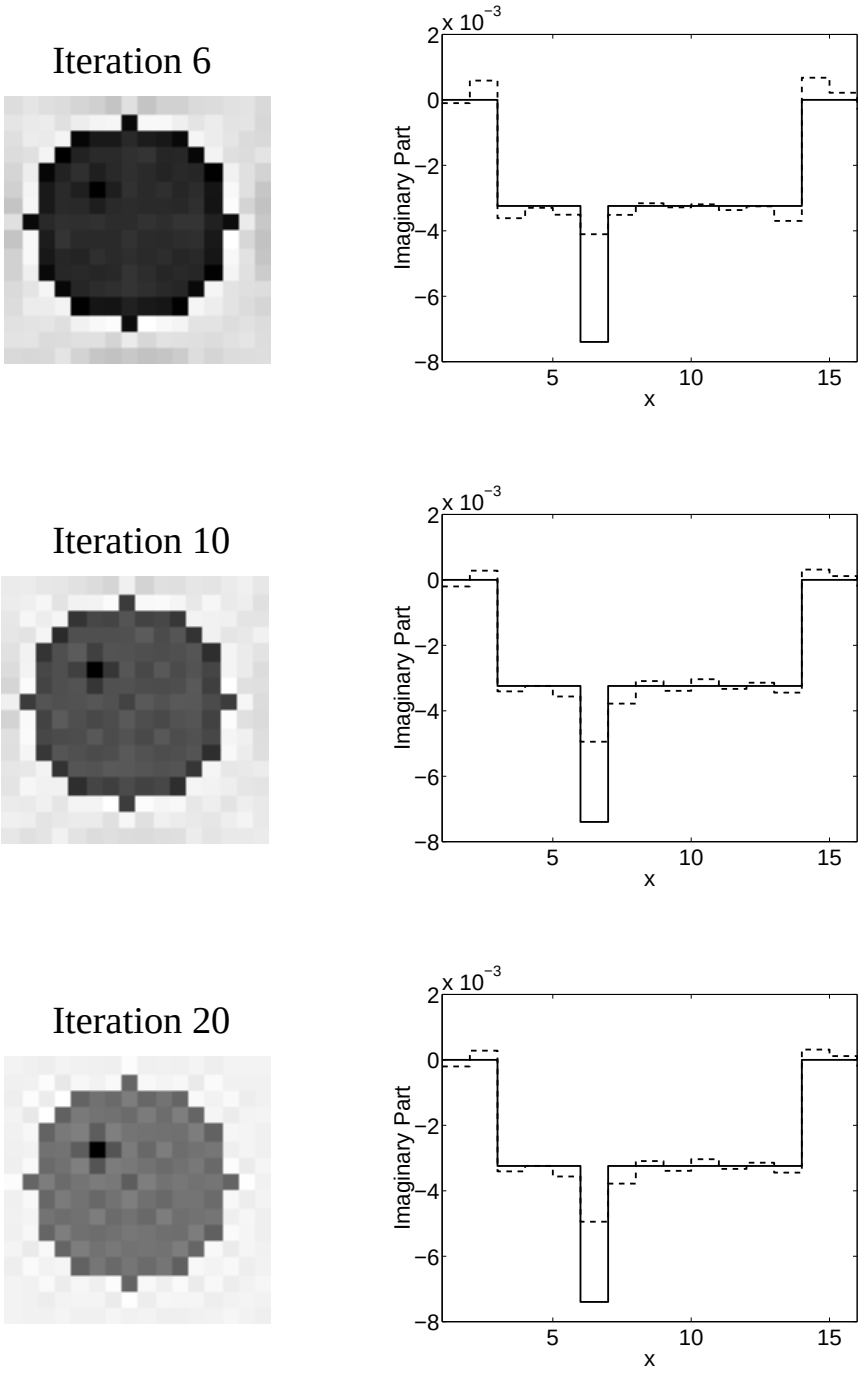

Fig. 6. Reconstructed results for the CG method for noiseless data (Case 1). The left column shows reconstructions at Iterations 6, 10, and 20. In the right column the solid lines are the true contrast along the line bisecting the tumor; the dashed lines are the reconstructed values for Iterations 6, 10, and 20.

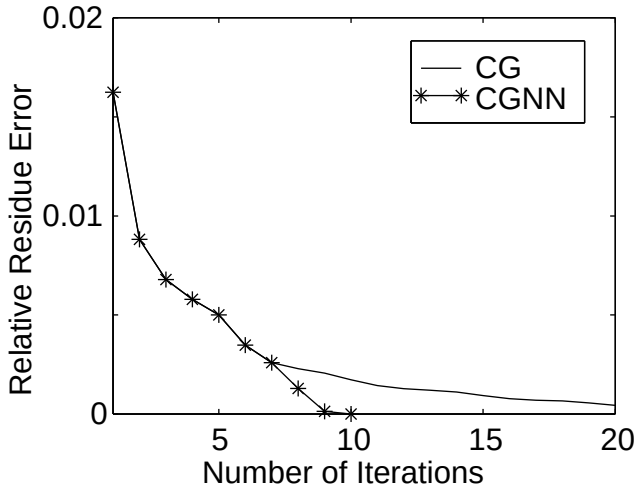

Fig. 7. RRE curves for both the CG and CGNN methods for noiseless data (Case 1).

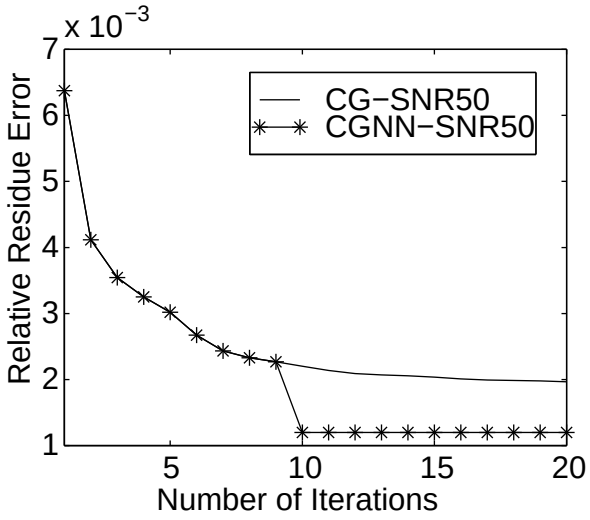

Fig. 8. RRE curves for both the CG and CGNN methods for $SNR = 50$ dB (Case 1).

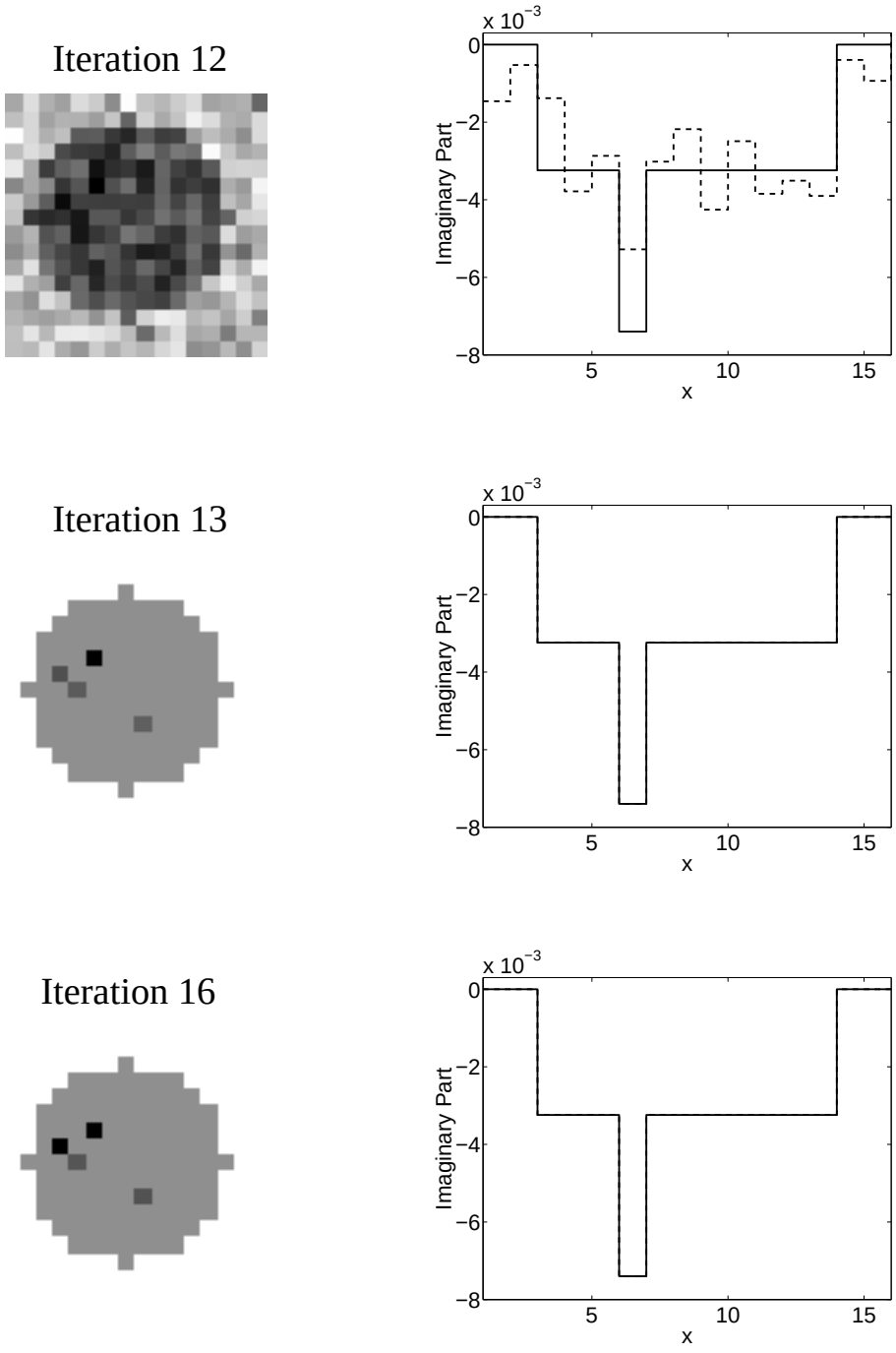

Fig. 9. Reconstructed results for the CGNN method for $SNR = 35$ dB (Case 1). The left column shows the reconstructions at Iterations 12, 13, and 16. In the right column the solid lines are the true contrast along the line bisecting the tumor; the dashed lines are the reconstructed values for Iterations 12, 13, and 16.

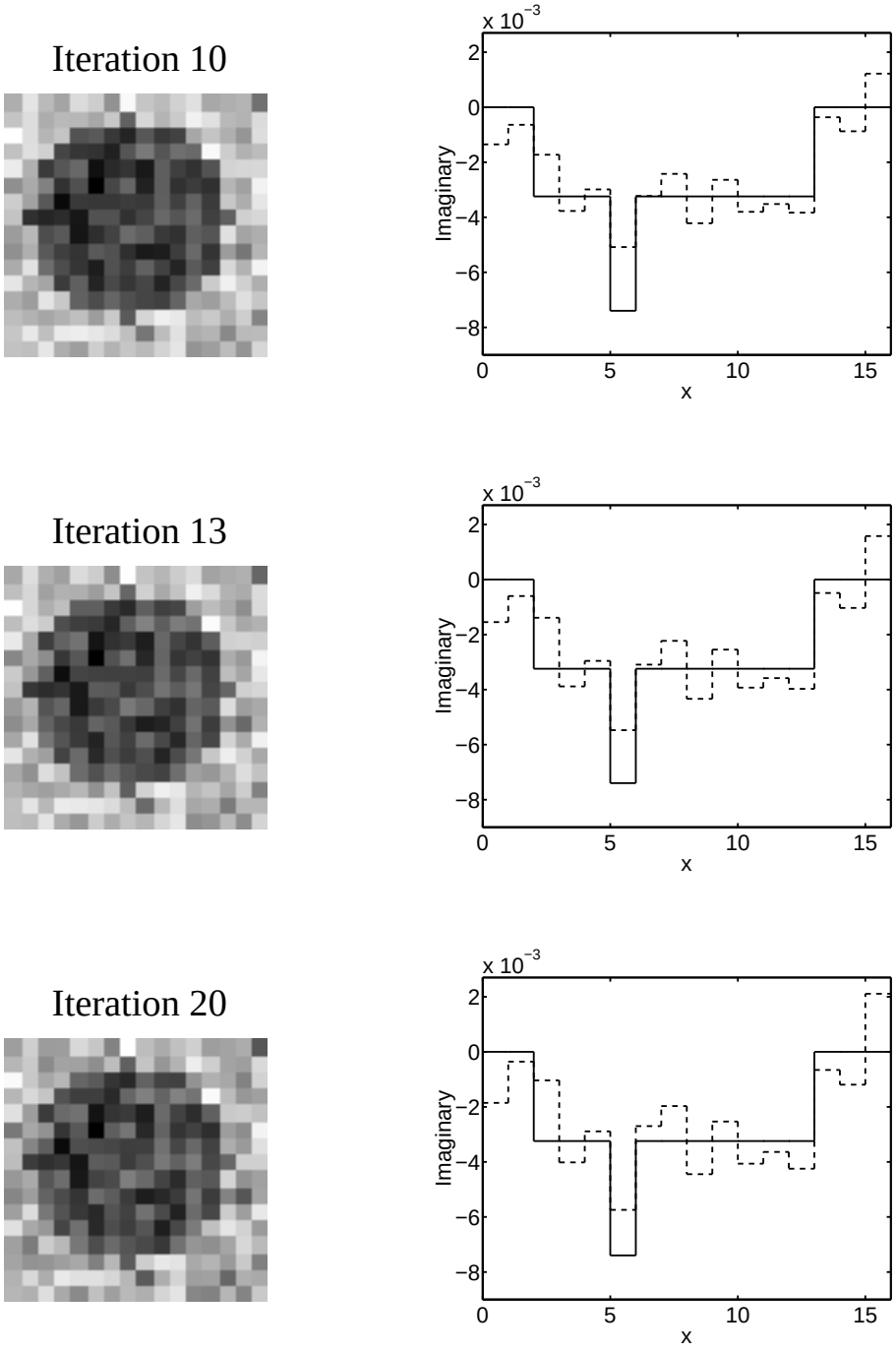

Fig. 10. Reconstructed results for the CG method for $SNR = 35$ dB (Case 1). The left column shows the reconstructions at Iterations 10, 13, and 20. In the right column the solid lines are the true contrast along the line bisecting the tumor; the dashed lines are the reconstructed values for Iterations 10, 13, and 20.

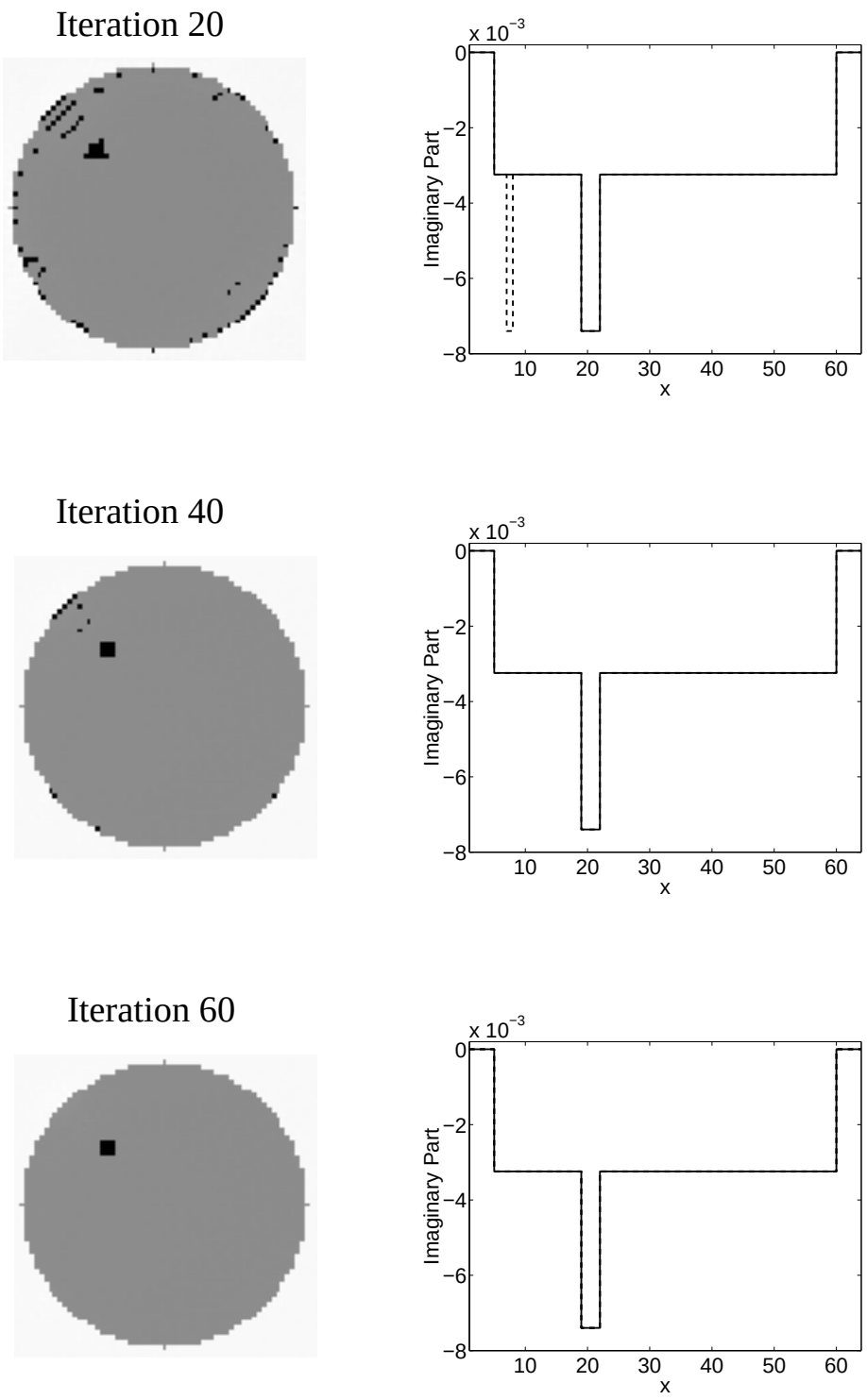

Fig. 11. Reconstructed results for the CGNN method for noiseless data (Case 2). The left column shows the reconstructions at Iterations 20, 40, and 60. In the right column the solid lines are the true contrast along the line bisecting the tumor; the dashed lines are the reconstructed values for Iterations 20, 40, and 60.

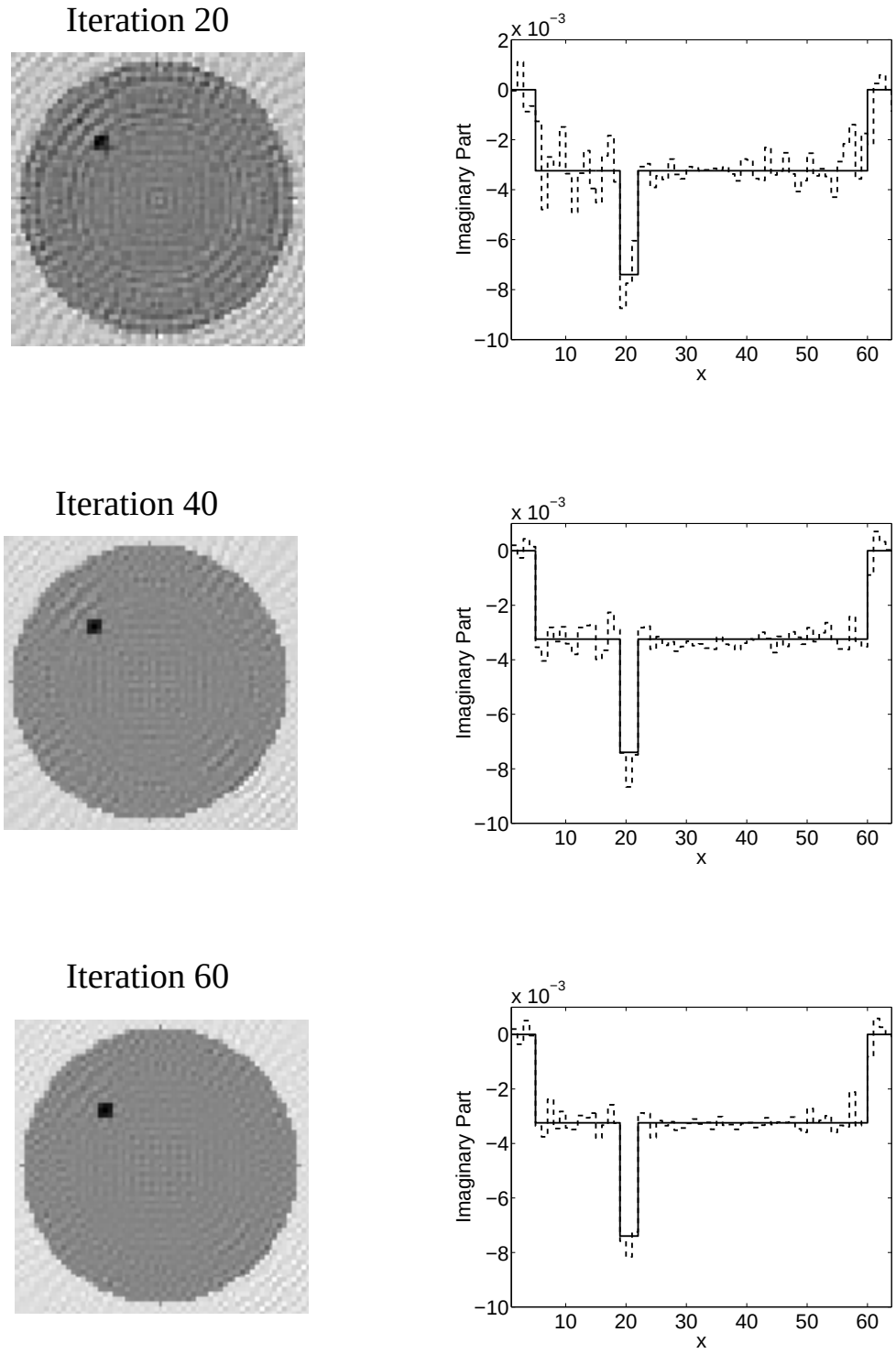

Fig. 12. Reconstructed results for the CG method for noiseless data (Case 2). The left column shows the reconstructions at Iterations 20, 40, and 60. In the right column the solid lines are the true contrast along the line bisecting the tumor; the dashed lines are the reconstructed values for Iterations 20, 40, and 60.

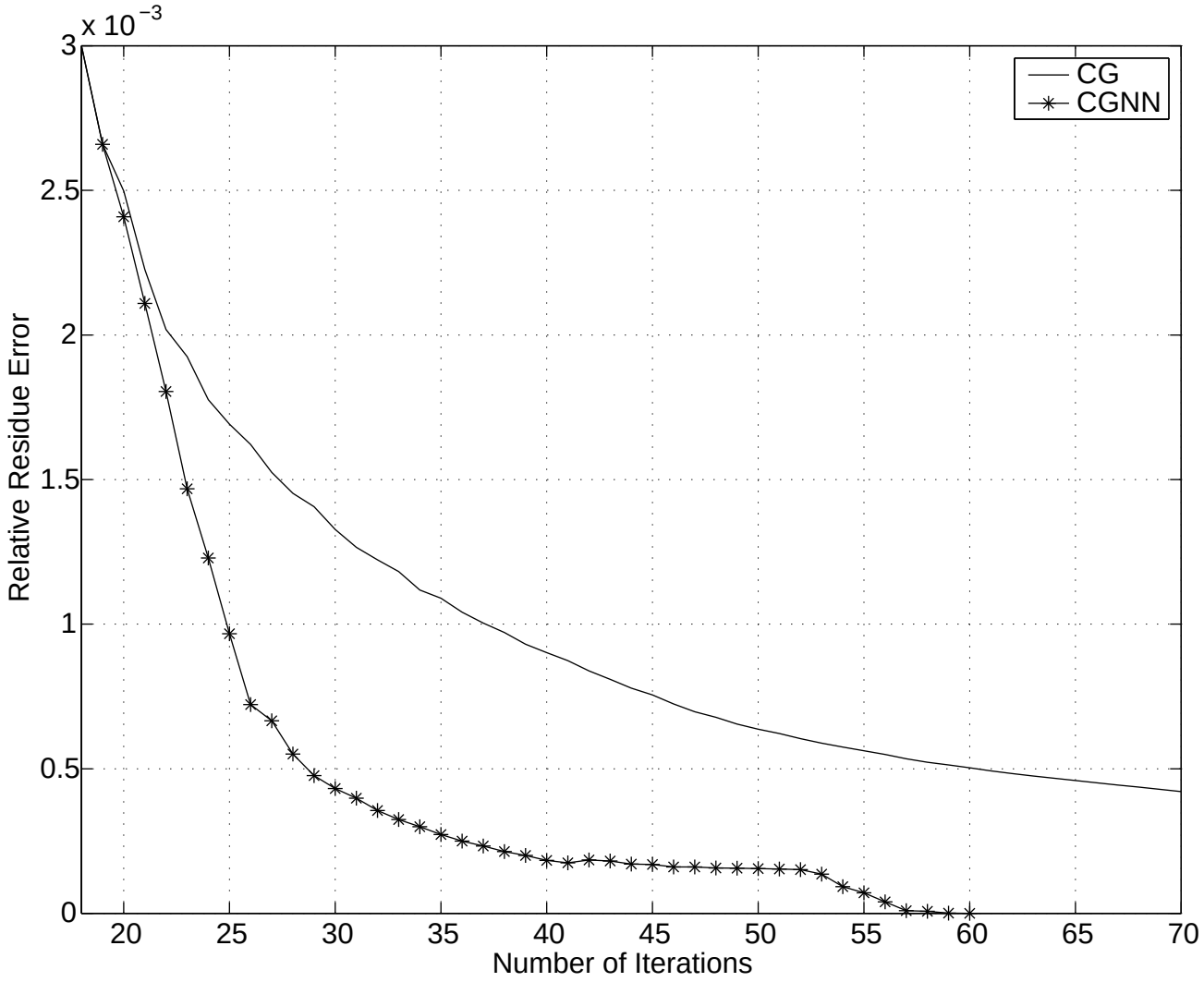

Fig. 13. RRE curves for both the CG and CGNN methods for noiseless data (Case 2).

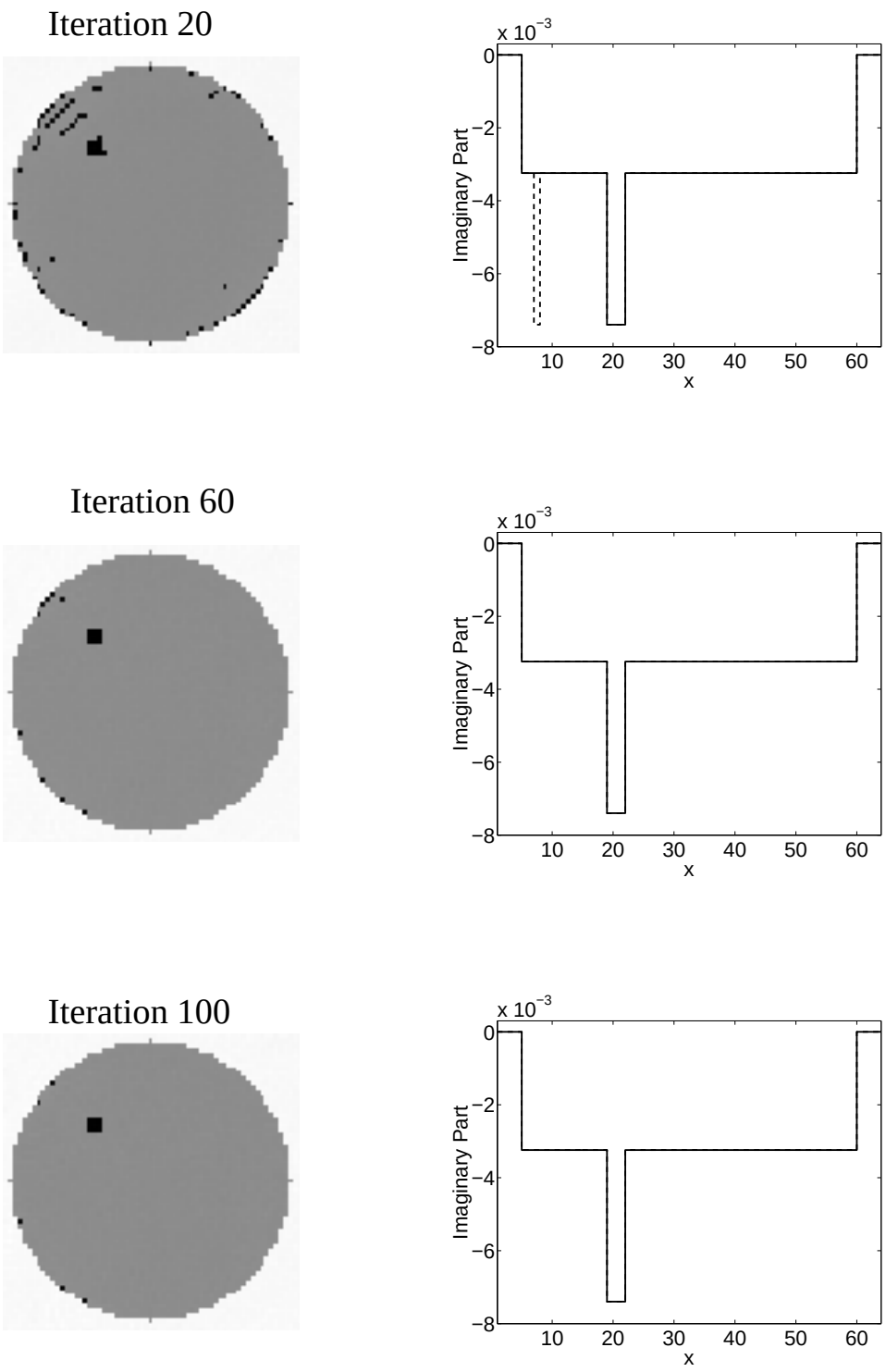

Fig. 14. Reconstructed results for the CGNN method for $SNR = 50$ dB (Case 2). The left column shows the reconstructions at Iterations 20, 60, and 100. In the right column the solid lines are the true contrast along the line bisecting the tumor; the dashed lines are the reconstructed values for Iterations 20, 60, and 100.

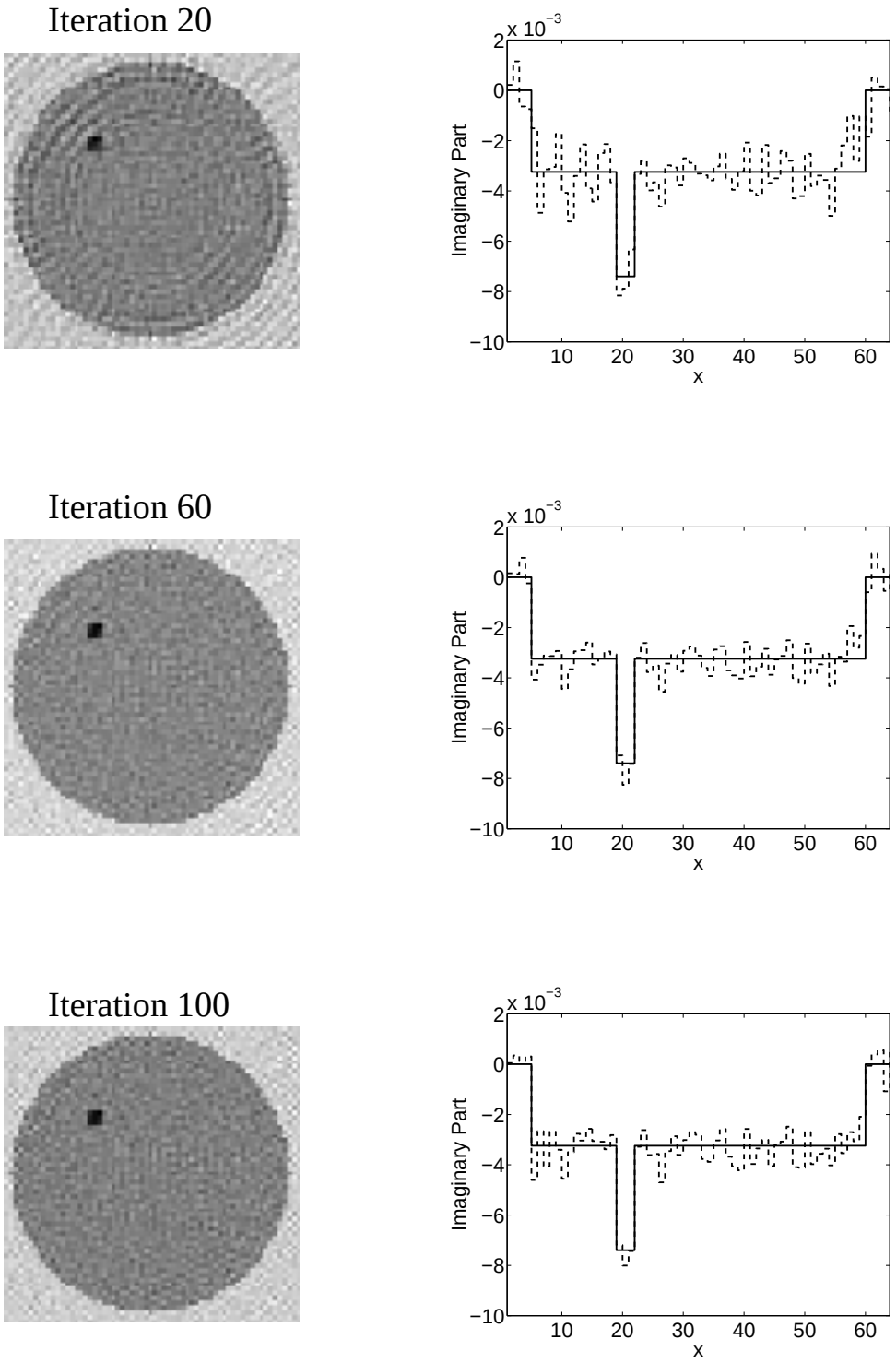

Fig. 15. Reconstructed results for the CG method for $SNR = 50$ dB (Case 2). The left column shows the reconstructions at Iterations 20, 60, and 100. In the right column the solid lines are the true contrast along the line bisecting the tumor; the dashed lines are the reconstructed values for Iterations 20, 60, and 100.

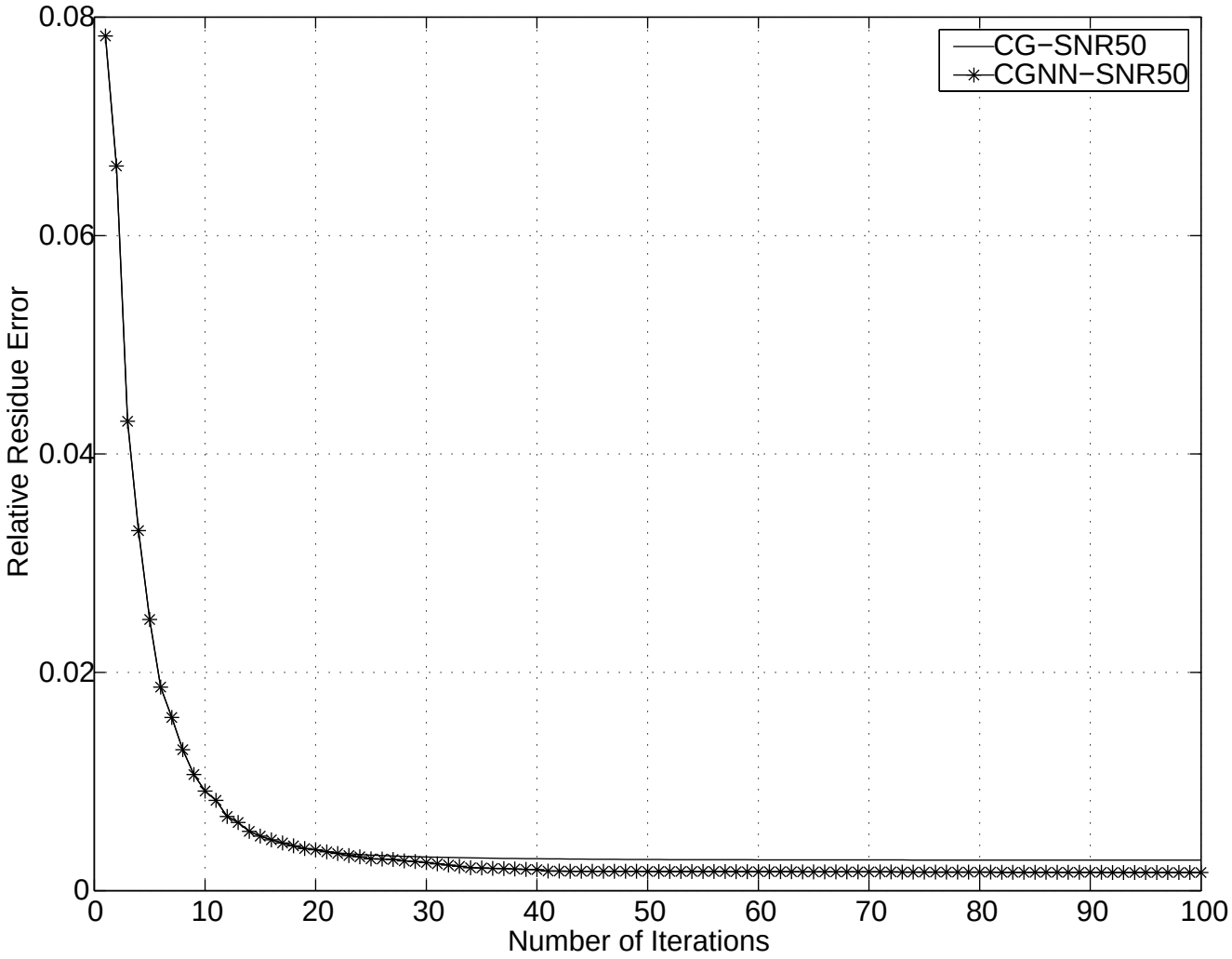

Fig. 16. RRE curves for both the CG and CGNN methods for $SNR = 50$ dB (Case 2).