TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 02/13 pp545-559
Volume 20, Number 6, December 2015

Decentralized Multigrid for In-situ Big Data Computing

Goutham Kamath, Lei Shi, Edmond Chow, Wenzhan Song*, and Junjie Yang

Abstract: Modern seismic sensors are capable of recording high precision vibration data continuously for several months. Seismic raw data consists of information regarding earthquake's origin time, location, wave velocity, etc. Currently, these high volume data are gathered manually from each station for analysis. This process restricts us from obtaining high-resolution images in real-time. A new in-network distributed method is required that can obtain a high-resolution seismic tomography in real time. In this paper, we present a distributed multigrid solution to reconstruct seismic image over large dense networks. The algorithm performs in-network computation on large seismic samples and avoids expensive data collection and centralized computation. Our evaluation using synthetic data shows that the proposed method accelerates the convergence and reduces the number of messages exchanged. The distributed scheme balances the computation load and is also tolerant to severe packet loss.

Key words: distributed multigrid; cyber physical system; big data; seismic tomography; sensor network; in-network computing

1 Introduction

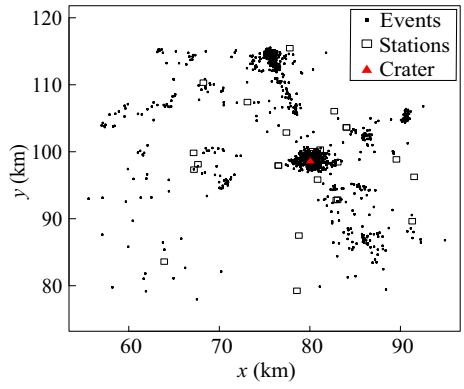
Current volcano monitoring systems lack the capability of obtaining real time information and recover the physical dynamics of seismic activity with sufficient resolution. At present, the seismic tomography process involves aggregation of raw seismic data to centralized server for post-processing and analysis. To give some perspective on the volume, the raw seismic data are sampled with 16–24 bit precision at 50–200 Hz. These high fidelity samples are generally primary (P) or secondary (S) wave, which contain information such as earthquake origin time, location, wave velocity, etc. This high frequency sampling at each node makes it extremely difficult to transmit the data over a dense sensor network due to severe limitations on energy and bandwidth. Due to these restrictions, many of the most threatening active volcanoes worldwide use fewer than 20 nodes[1]. The existing scheme also requires months to generate satisfactory tomography images. This limits our ability to understand volcano dynamics and physical processes in real-time. The centralized solution also introduces a bottleneck in computation. The risk of data loss also increases in case of node failures, especially at the base station. The centralized algorithm for these battery powered nodes, which have high risk of failures, is not suitable for volcano monitoring.

The high volume raw samples consist of sparse earthquake information, however current technology requires station to transfer all the raw samples of P and S waves to centralized station for post processing. In Ref. [2] the data collected from 1980–2004 consists only of 19 379 useful earthquake events and in addition 6916 events from October 2004–December 2005.

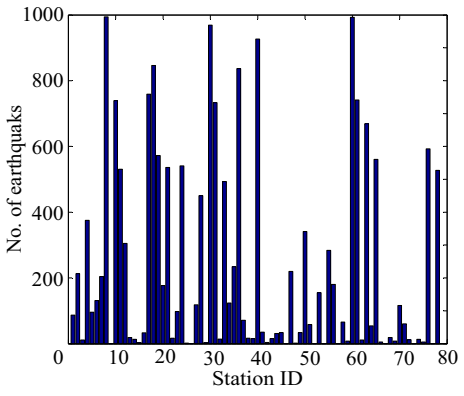
• Goutham Kamath, Lei Shi, and Wenzhan Song are with the Department of Computer Science, Georgia State University, Atlanta, GA 30303, USA. E-mail: {gkamath1, lshi1}@student.gsu.edu; wsong@gsu.edu.
• Edmond Chow is with College of Computing, Georgia Institute of Technology, Atlanta, GA 30332, USA. E-mail: echow@cc.gatech.edu.
• Junjie Yang is with School of Electric and Information Engineering, Shanghai University of Electric Power, Shanghai 200090, China.
∗ To whom correspondence should be addressed.
Manuscript received: 2015-08-01; accepted: 2015-08-19

Figure 1 shows the parse distribution of earthquake events obtained from 78 stations placed on Mt St Helens (MSH). Few stations receive as few as 10 events while others receive more than 900. This sparse feature of raw samples has led researchers to adopt distributed techniques to perform in-network processing and avoid centralized computation. The advancement in current wireless sensor technology makes it possible to deploy and maintain a large-scale network for environmental monitoring and surveillance. However, seismic tomography algorithms commonly in use today cannot be easily implemented under this distributed scenario as it relies on centralized processing. Thus, real-time volcano tomography requires a practical approach which is distributed, scalable, and efficient with respect to tomography computation.

Seismic tomography can be broadly classified into two main categories: active and passive tomographies. In active seismic tomography, earth's interior is studied by sending P wave signal through external source such as vibrator, however in passive tomography, measurements are taken based on P wave generated by



(a) Earthquake locations



(b) Event detection distribution

Fig. 1 Non-uniform distribution of rays and events at Mt St Helens.

natural sources such as earthquake. Since late 1970s, active tomographic inversion of 2-D and 3-D structures has been studied widely both theoretically and experimentally by applying it to oil field exploration and volcanoes[3]. Only in recent years, passive seismic tomography has been studied and the data obtained from few tens of nodes are being used to study seismic activities. As mentioned earlier, these inversion methods rely on centralized data gathering scheme and have been implemented on volcanoes such as Mount St. Helens[4] and Mount Rainier[5] as well as many others. The resolutions of such inversions are typically in tens of kilemeter's and higher resolutions are hard to obtain from the existing systems as the number of sensors is not sufficient to cover the entire region of interest. Deploying large sensor nodes using the current data gathering network is also not feasible as these networks do not scale and sometimes it becomes impossible due to data load. To overcome this, we developed a method called Component Average-Distributed Multiresolution Evolving Tomography (CA-DMET) which computes the tomography over sensor networks[6]. In this method each sensor nodes were responsible to calculate partial solution by solving large sparse linear equation available to them using Bayesian Algebraic Reconstruction Technique (BART)[7]. The partial solution obtained from each node was later combined with others to obtain the next iteration. Convergence of this algorithm was proved to be better than other distributed methods. In this paper we try to accelerate its convergence and improve the performance of the reconstruction using multigrid approach.

Typically when solving large sparse linear systems, iterative methods tend to reduce high-frequency (oscillatory) components directly while not lower the errors caused due to low-frequency. Multigrid methods are often used to mitigate these low-frequency errors, as they reduce them by transferring the problem to lower grids. We investigated our seismic tomography inversion problem and found that multigrid could be used to accelerate the convergence. In this paper, we propose Distributed MultiGrid Tomography (DMGT) algorithm which accelerates the convergence rate of CA-DMET. Our contribution in the proposed approach differs from our previous algorithm CA-DMET in three ways: firstly we prove that BART satisfies the smoothing property and can be used as a smoother in multigrid. Secondly, we show that multigrid with BART as smoother when applied on each node converges

faster than applying only BART as used in CA-DMET. Thirdly, we show that multigrid can be applied on each node distributedly and the intermediate result can be combined using the component average method. This paper mainly focuses on the distributed tomography algorithm, while assuming the arrival time of events at each node has been extracted from the raw seismic data by each node itself[8, 9]. The algorithm proposed here has application to fields far beyond the specifics of volcanology, e.g., oil field explorations have similar problems and needs.

The rest of the paper is organized as follows. In Section 2, we provide background on seismic tomography inversion and present the problem formulation. Section 3 presents related work on distributed least squares and distributed multigrid methods. In Section 4, we first discuss mathematical developments that lead to the design of DMGT and then present the DMGT algorithm in detail. Simulation results are shown in Section 5. Finally we conclude the paper in Section 6.

2 Problem Formulation

Seismic tomography: The methodology used in seismic tomography is borrowed from medical tomography where the travel time of elastic wave is used to probe internal structure. Although this idea is common in these two applications, there are significant differences, mainly pertaining to size of the structures and to event generation. The velocity model used in seismic tomography is non-linear and the ray path of the waves traveling through the ground may be highly curved due to the size and complexity of the volcano. Typically, the ray source in volcano tomography is an earthquake event where the distribution of the ray path is highly non-uniform unlike uniform short distance rays generated in medical imaging. These differences indicate that special care must be taken when techniques borrowed from medical tomography are applied to seismic data.

The basic principle behind 2-D or 3-D seismic tomography is to use the arrival time of the P wave to derive the internal velocity structure of the volcano. This approach is called *travel-time seismic tomography* and the model here is continuously evolving and refined as more earthquakes are recorded. Below we explain the three basic principles involved in travel-time seismic tomography.

Step 1 Event location: Once an earthquake occurs, seismic disturbances are detected by sensor nodes and arrival times are recorded. Using these estimated arrival times, Geiger[10] introduced a technique to estimate the earthquake location and origin time. This is a classic and widely used event localization scheme generally using Gauss-Newton optimization.

Step 2 Ray tracing: This is the technique of finding the ray paths from the seismic source locations to the sensor nodes with minimum travel time. Given the source location of the seismic events and the current velocity mode of the volcano, ray tracing finds the ray paths from the event source location to the nodes as shown in Fig. 2b.

Step 3 Tomographic inversion: The ray paths traced in turn are used to estimate the velocity model of the volcano. The volcano is partitioned into small blocks as shown in Fig. 2c. This allows us to formulate the tomography problem as a system of sparse linear equations. Suppose there are N sensor nodes and E earthquakes and $\boldsymbol{x}^*$ denotes the reference slowness (reciprocal of velocity) model of the volcano with resolution M blocks (e.g., 32×32). Let $\boldsymbol{x}^*$ denote the sum of $\boldsymbol{x}^0$, unperturbed model, and $\boldsymbol{x}$, a small perturbation, i.e., $\boldsymbol{x}^* = \boldsymbol{x}^0 + \boldsymbol{x}$.

Let $\boldsymbol{b}_i^* = [b_{i1}^*, b_{i2}^*, \cdots, b_{iE}^*]^{\mathrm{T}}$, where b_{ie}^* be the travel time experienced by node i in the e-th event. Based on the ray paths traced in Step 2, the travel time of a ray is the sum of the slowness in each block times the length of the ray within that block, i.e., $b_{ie}^* = \boldsymbol{A}_i[e, m] \cdot \boldsymbol{x}^*[m]$ where $\boldsymbol{A}_i[e, m]$ is the length of the ray from the e-th event to node i in the m-th block and $\boldsymbol{x}^*$ is the slowness of the m-th block. Let $\boldsymbol{b}_i^0 = [b_{i1}^0, b_{i2}^0, \cdots, b_{iE}^0]^{\mathrm{T}}$ be the unperturbed travel times where $\boldsymbol{b}_i^0 = \boldsymbol{A}_i[e, m] \cdot \boldsymbol{x}^0[m]$. In the matrix notation we have the following equation,

$$\boldsymbol{A}_i\boldsymbol{x}^* - \boldsymbol{A}_i\boldsymbol{x}^0 = \boldsymbol{A}_i\boldsymbol{x} \tag{1}$$

where $\boldsymbol{A}_i \in \mathbf{R}^{E \times M}$. Let $\boldsymbol{b}_i = [b_{i1}, b_{i2}, \cdots, b_{iE}]^{\mathrm{T}}$ be the travel time residual such that $\boldsymbol{b}_i = \boldsymbol{b}_i^* - \boldsymbol{b}_i^0$, Eq. (1) can be rewritten as

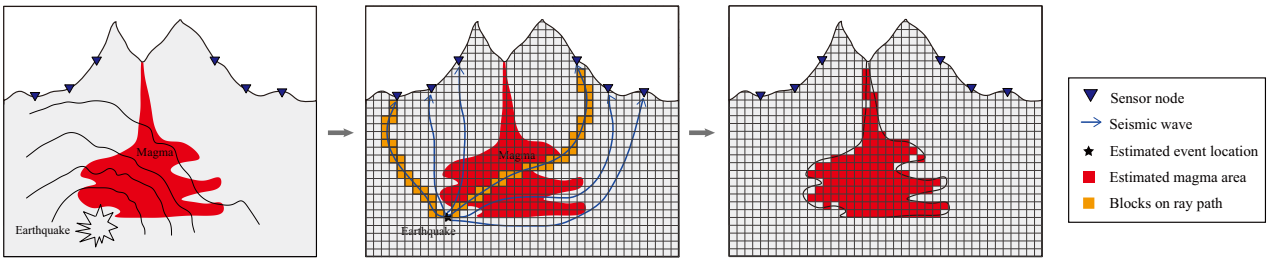
$$\boldsymbol{A}_i\boldsymbol{x} = \boldsymbol{b}_i \tag{2}$$

Since each ray path intersects the model at a small number of blocks, the design matrix, $\boldsymbol{A}_i$, is sparse. For the system with N sensor nodes, the equation of the entire system would be

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{B} \tag{3}$$

where $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, \cdots, \boldsymbol{b}_N]^{\mathrm{T}}$, $\boldsymbol{b}_i = [b_{i1}, b_{i2}, \cdots, b_{iE}]^{\mathrm{T}}$, and $\boldsymbol{A} = [\boldsymbol{A}_1, \boldsymbol{A}_2, \cdots, \boldsymbol{A}_N]^{\mathrm{T}}$.

Now from the above equation, each seismic sensor



(a) Event location (b) Ray tracing (c) Tomographic inversion

Fig. 2 Procedures of seismic tomography inversion.

$i \in (1, \cdots, N)$ contains at least E rows, i.e., earthquake events and travel time information. The column size of $\boldsymbol{A}$ denotes the resolution of the slowness model $\boldsymbol{x}$ being calculated. Our goal is to obtain the slowness model $\boldsymbol{x}$ without collecting the event information from each node in a centralized server, but only by exchanging partial slowness between the sensors.

3 Related Work

3.1 Distributed linear least squares

The tomography inversion process mainly involves solving large sparse overdetermined systems of linear equations (Eq. (3)) and iterative methods are commonly used. Several parallel and distributed iterative methods have been developed and are currently being used to solve a large variety of problems[11, 12]. Consensus-based methods are the most widely used distributed algorithm for wireless sensor networks, e.g., Ref. [13]. These algorithms use a weighted sum of local estimates to achieve consensus. Each sensor node maintains its own local estimation and exchanges information locally to achieve consensus. These methods are primarily designed for estimation of low dimensional vectors typically in a parallel environment. To achieve global convergence, consensus protocols generally require relatively high execution time and frequent communication between neighbors. In seismic tomography networks, this approach not only means high communication overhead but also longer delays involving many multi-hop communications. Therefore, the consensus-based distributed least square algorithms are not suitable for high-resolution seismic tomography in sensor networks.

Another method originally proposed for parallel computing is the multi-splitting solution of the least squares problem[14]. This method partitions the system into columns instead of rows, letting each processor compute a partial solution. These partial solutions are exchanged iteratively to obtain global convergence. Column splitting of Eq. (3) in seismic tomography means splitting the travel time $\boldsymbol{B}$. Since we only have the information of total travel time from event source to node, we cannot divide $\boldsymbol{B}$ exactly and any heuristic approach will add error in addition to existing system noise. Apart from that, this method is only linearly convergent and the communication cost is very expensive as it requires exchanging $\boldsymbol{B}$ which in our case increases with occurrences of earthquake events. Due to these reasons, column partitioning is not suitable for seismic tomography.

A popular iterative method for solving overdetermined systems was proposed by Kaczmarz (KACZ)[15], which is an alternating projection method. This method is also known under the name ART in computer tomography[16]. This algorithm does not require the full matrix to be in memory at one time and can incorporate new information (ray paths) real-time the fly. The vectors of unknowns are updated after processing each equation of the system and this cycle repeats until convergence. These iterative algorithms are distributed by averaging the boundary information, e.g., Component AVeraging (CAV)[17], Block Iterative-Component AVeraging (BI-CAV)[18], and Component-Averaged Row Projections (CARP)[19]. A survey paper comparing various block parallel methods based on their performance on GPU's is Ref. [20]. CA-DMET[6] involved modification of these algorithms for seismic tomography. The convergence of the iterative method used depended on spectral properties of the iteration matrix. Generally in iterative methods, convergence stalls once the error is smooth, i.e., high-frequency errors are reduced. Multigrid methods provide great tool to prevent stagnation by transferring smooth errors from fine grids to coarse grids, resulting in overall

acceleration of convergence[21], however, it cannot be applied to solve all the problems arising from systems of linear equations[22]. In this paper, we analyze the tomography problem carefully and develop tools such as smoothers, intergrid operator, etc., satisfying the requirements of multigrid.

3.2 Distributed multigrid

Multigrid has been parallelized on multicore computers and distributed memory clusters[23,24]. To perform multigrid in distributed networks, many new considerations arise, including high communication cost and the possibility of packet loss. For example, some existing parallel and distributed multigrid algorithms partition the multigrid levels among different cores/nodes and the intergrid operators communicate between each other to perform a multigrid cycle[25,26]. In case of seismic tomography, exchanging the rows of matrix $\boldsymbol{A}$ (ray information) between each nodes is expensive and defeats the whole purpose of the distributed approach. Thus, we cannot adopt all previous techniques for parallelizing multigrid and apply them to volcano tomography over sensor networks.

Iterative methods such as Jacobi, Gauss-Seidel, and SOR for many problems have the property of smoothing the error and are used as the "smoother" in multigrid methods[21]. However, for solving overdetermined systems, it is more natural to use Kaczmarz or ART as the smoother. This appears to be first considered in Refs. [27, 28] for multigrid in medical image tomography in a centralized setup. For inconsistent overdeteremined systems, Extended Kacmarz (EK) was introduced[29] which performs column operations at each iteration to manipulate the right hand side of the linear equation. However in our case, since information over sensors is split row wise, column operations over the entire network will add significant communication. In this paper we propose DMGT which accelerates the convergence of seismic tomography inversion over a network and balances the computation cost with reduced communication. DMGT uses BART as a smoother and we prove that BART satisfies the smoothing property. We also show that DMGT is applicable to seismic tomography. To the best of our knowledge, this work is the first attempt to distribute the multigrid computation of seismic tomography in sensor networks.

4 Distributed Algebraic Multigrid for Tomography

4.1 Mathematical developments

4.1.1 Bayesian ART

The tomography inverse problem involves finding a solution $\boldsymbol{x}$ which satisfies Eq. (3). Typically, the seismic tomography equation is quasi-overdetermined, inconsistent, and contains measurement noise. Therefore, we need to use some form of regularization to avoid strong, undesired influence of small singular values dominating the solutions. This can be achieved by using a regularization parameter for the least-squares solution x_{LS} , i.e.,

$$x_{\mathrm{LS}} = \arg\max_{\boldsymbol{x}} \|\boldsymbol{B} - \boldsymbol{A}\boldsymbol{x}\|^2 + \lambda^2 \|\boldsymbol{x}\|^2 \qquad (4)$$

where λ is the trade-off parameter that regulates the relative importance we assign to models that predict the data versus models that have a characteristic, a priori variance.

A variant of ART called Bayesian ART can be used for solving Eq. (3) by minimizing Eq. (4). Suppose the system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is inconsistent, then we have $\boldsymbol{A}\boldsymbol{x}+\boldsymbol{y} = \boldsymbol{b}$ where $\boldsymbol{y}$ is chosen from any given $\boldsymbol{x}$. Then the system is transformed to a well-posed problem. Now $\boldsymbol{x}$ and $\boldsymbol{y}$ can be solved simultaneously using the iterative algorithm 1, as shown in Algorithm 1, where $\boldsymbol{e}_i$ is a unit vector with the i-th component equal to one, and λ is the regularization parameter.

Note that in the Bayesian ART method, we need an additional vector $\boldsymbol{y}$ of length E, but in the k-th step only one component of $r^{(k)}$ needs to be updated. This method has been used for seismic imaging in Ref. [7].

4.1.2 Multigrid

Multigrid methods are among the most efficient methods for solving the very large sparse system of linear equations[21,22]. The core idea of multigrid is to reduce the error via transferring the problem between multiple levels and solving them over these levels. The residual equation is transferred to coarser grids and its solution is used to correct the finer

Algorithm 1 Bayesian ART

1: **for** $k \leftarrow 0$ until convergence or maximum number of iteration **do**
2: $\quad k \leftarrow i \mod m + 1$
3: $\quad d^{(k)} = \rho^{(k)} \frac{\lambda b_i - (y_i^{(k)} + \lambda \boldsymbol{a}_i^{\mathrm{T}} \cdot \boldsymbol{x}^{(k)})}{1+\lambda^2 \|\boldsymbol{a}_i\|^2}$
4: $\quad \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \lambda d^{(k)} \boldsymbol{a}_i$
5: $\quad \boldsymbol{y}^{(k+1)} = \boldsymbol{y}^{(k)} + d^{(k)} \boldsymbol{e}_i$
6: **end**

resolution solution. This is performed recursively until the convergence is met. The idea of multigrid aligns with multi-resolution techniques and we have shown in Ref. [6] that multi-resolution is essential in estimating volcano tomography.

The main components of multigrid are the smoother and prolongation and restriction operators, and wide variety of these are used in different scenarios. These components are chosen based on the type of the problem to optimize the convergence. Prolongation and restriction operators mainly decide the construction of finer and coarser grids. In case of tomography the grids are constructed based on the principle of ray tracing and here we will show that ray tracing can be used for prolongation and restriction in multigrid. Prolongation and restriction are generally termed as intergrid operators as they define the transfer process between the grids. As mentioned earlier, the tomography problem has a geometric structure and here we exploit this structure to define the intergrid operators. However, these intergrid operators must have certain properties and in this section we will show that our ray tracing satisfies these properties.

Let n be the number of columns in $\boldsymbol{A}$ and suppose that $n = 4p$ where p denotes the number of pixels given in Fig. 3a and let $P_1, \cdots, P_n$ be the pixels on the *fine grid*. The *coarse grid* is obtained by combining its 4 adjacent pixels of the fine grid as shown in Fig. 3a. Let $S(j), j \in 1, \cdots, p$ be the set of indices of the fine grid that form the coarse grid P_j^{H}, i.e.,

$$S(j) = \{j_1, j_2, j_3, j_4\} \quad \forall j = 1, \cdots, p,$$

where

$$j_1 < j_2 < j_3 < j_4$$

such that

$$P_j^{\mathrm{H}} = \{P_{j1} \cup P_{j2} \cup P_{j3} \cup P_{j4}\}.$$

From the above equation the coarse grid matrix $\boldsymbol{A}_p$ will be

$$A_p^{ij} = \sum_{k \in S(j)} A_{ik}, \forall i = 1, \cdots, m, j = 1, \cdots, p \quad (5)$$

Now the interpolation operator $\boldsymbol{I}_p^n$ is given by

$$\boldsymbol{I}_p^n = \begin{cases} 1, & \text{if } i \in S(j); \\ 0, & \text{if } i \notin S(j) \end{cases} \quad (6)$$

We now see that $\boldsymbol{A} = \boldsymbol{A}_p \times \boldsymbol{I}_p^n$ satisfying the interpolation property. We also observe that $\boldsymbol{I}_p^n$ has full column rank.

Remark 1 Notice that the interpolation operator only increases the number of columns in matrix $\boldsymbol{A}$. We can also consider a similar operator which also reduces the rows by weighting them, however this is beyond the scope of this paper.

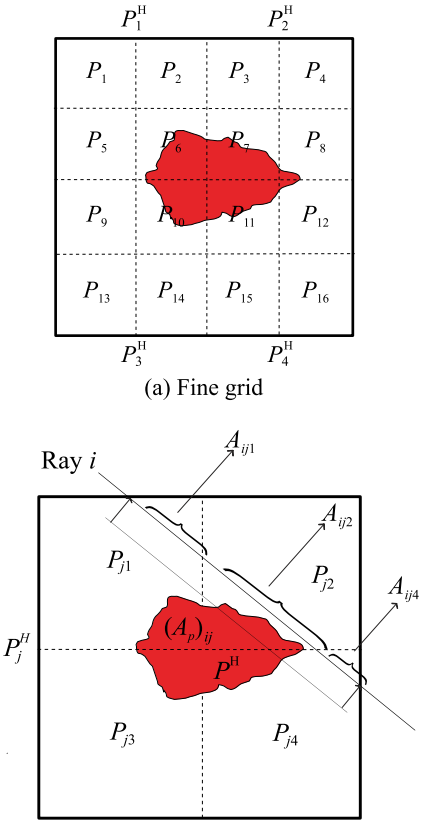


Fig. 3 Relation between fine and coarse grids.

Remark 2 The above multigrid operators are designed for 2-D cases, however 3-D case can be easily derived using $n = 8p$, i.e., cuboid.

We have now shown that the interpolation operator formed by using the property of ray tracing can be used as an intergrid operator in multigrid. We also saw that BART can be used for solving tomography problems. Next we show how BART can act as a smoother and prove it satisfies the smoothing property of multigrid.

4.1.3 Bayesian ART as a smoother

For seismic tomography, BART is commonly used rather than ART or Gauss-Seidel. The problem being inconsistent and ill-posed, BART outperforms other standard iterative algorithms in terms of convergence and solution[7]. However, BART has not been proven as a smoother in a multigrid setup and in this section we will prove that BART satisfies the smoothing property.

Definition 1 The **smoothing property** is satisfied by the relaxation scheme if there exists a constant $\alpha > 0$

(independent of size or eigenvalues of $\boldsymbol{A}$) such that

$$\|\bar{\boldsymbol{e}}\|_{\boldsymbol{A}}^2 \leqslant \|\boldsymbol{e}\|_{\boldsymbol{A}}^2 - \alpha\|\boldsymbol{r}\|_{\boldsymbol{D}^{-1}}^2 + \|\boldsymbol{y}\|_{\boldsymbol{D}^{-1}}^2 \tag{7}$$

where, $\boldsymbol{e} = \boldsymbol{x} - \boldsymbol{x}^*$, $\boldsymbol{r} = \boldsymbol{Ae} = \boldsymbol{Ax} - \boldsymbol{b}$, $\bar{\boldsymbol{e}} = \bar{\boldsymbol{x}} - \boldsymbol{x}^*$, $\boldsymbol{D} = \text{diag}(\boldsymbol{A})$, $\|\boldsymbol{x}\|_{\boldsymbol{A}} = \sqrt{\langle \boldsymbol{Ax}, \boldsymbol{x}\rangle}$, and $\|\boldsymbol{r}\|_{\boldsymbol{D}^{-1}} = \sqrt{\langle \boldsymbol{D}^{-1}\boldsymbol{r}, \boldsymbol{r}\rangle}$.

With respect to the above definitions and notation, Theorem 6 in Ref. [27] shows that Kaczmarz relaxation for consistent systems satisfies the following smoothing property,

$$\|\bar{\boldsymbol{e}}\|^2 \leqslant \|\boldsymbol{e}\|^2 - \tilde{\gamma}\|\tilde{\boldsymbol{D}}^{\frac{1}{2}}\boldsymbol{r}\|^2 \tag{8}$$

where

$$\tilde{\boldsymbol{D}}^{\frac{1}{2}} = \text{diag}\left(\frac{1}{\|\boldsymbol{A}_1\|^2}, \cdots, \frac{1}{\|\boldsymbol{A}_n\|^2}\right) \tag{9}$$

$$\tilde{\gamma} = \frac{1}{(1 + \tilde{\gamma_-}(\boldsymbol{A}))(1 + \tilde{\gamma_+}(\boldsymbol{A}))} \tag{10}$$

and

$$\tilde{\gamma}_-(\boldsymbol{A}) = \max_{1 \leqslant i \leqslant n} \sum_{j \leqslant i} \frac{|\langle \boldsymbol{A}_i, \boldsymbol{A}_j\rangle|}{\|\boldsymbol{A}_i\|^2};$$

$$\tilde{\gamma}_+(\boldsymbol{A}) = \max_{1 \leqslant i \leqslant n} \sum_{j \geqslant i} \frac{|\langle \boldsymbol{A}_i, \boldsymbol{A}_j\rangle|}{\|\boldsymbol{A}_i\|^2}.$$

Theorem 1 Bayesian ART (Algorithm 1) as a relaxation scheme for inconsistent system (3) satisfies the smoothing property if there exists

$$\|\bar{\boldsymbol{e}}\|^2 \leqslant \|\boldsymbol{e}\|^2 - \tilde{\gamma}\|\tilde{\boldsymbol{D}}^{\frac{1}{2}}\boldsymbol{r}\|^2 + \|\tilde{\boldsymbol{D}}^{\frac{1}{2}}\boldsymbol{y}\|^2 \tag{11}$$

Proof Shown in Appendix 6 ■

4.1.4 Three-grid V-cycle

Here we describe the three-grid correction scheme used in our algorithm, as shown in Algorithm 2. If the finest resolution of our system to solve is of dimension 32×32, then resolution 16×16 is used as an intermediate grid and resolution 8×8 the coarsest grid. The coarsest grid is solved directly as the dimension is small, however we can also solve it by certain sweeps/iteration of BART. Later, the fine grid correction step is applied. The total number of iterations for one three-grid V-cycle will be equal to $4 \times l_1$. The three-grid V-cycle scheme is represented diagrammatically in Fig. 4.

Algorithm 2 $v^h \leftarrow$ **V-cycle**(v^h, b^h)

1: $v^h = \text{BART}(A^h, b^h, v^h)$ % Relax using l_1 sweeps of BART
2: $r^h = b^h - A^h v^h$ % Compute fine-grid residual
3: $r^{2h} = I_h^{2h} r^h$ % Restrict the residual to coarse grid
4: $v^{2h} = \text{BART}(A^{2h}, r^{2h}, 0)$
5: $r^{4h} = r^{2h} - A^{2h} v^{2h}$
6: $r^{4h} = I_{2h}^{4h} r^{2h}$
7: $A^{4h} u^{4h} = r^{4h}$ % Solve directly
8: $e^{4h} = (A^{4h})^{-1} r^{4h}$
9: $e^{2h} = I_{4h}^{2h} e^{4h}$ % Interpolate coarse grid error to fine grid
10: $v^{2h} = v^{2h} + e^{2h}$ % Correct the fine-grid approximation
11: $e^{2h} = \text{BART}(A^{2h}, b^{2h}, v^{2h})$ % Relax using l_1 sweeps of BART
12: $e^h = I_{2h}^h e^{2h}$
13: $v^h = v^h + e^h$
14: $v^h = \text{BART}(A^h, b^h, v^h)$

4.2 Design of DMGT algorithm

In Section 4.1, we discussed separately the components of multigrid suitable for tomography. In this subsection we will put these ideas together to design a distributed multigrid scheme that can balance the computation load and compute the least-square solution for seismic tomography inversion over a sensor network. The seismic sensors are deployed on top of the volcano and each sensor gathers ray information after detecting earthquake events and forms a partial set of linear equations. Later, each sensor performs DMGT locally to obtain the partial slowness model ($\bar{x}^k$) which is then combined with the partial slowness model obtained from other nodes using component averaging as shown in Fig. 5 to obtain the next iteration (x^{k+1}). This process is repeated until it converges to a threshold after which we obtain the global slowness model ($\boldsymbol{x}$). Here, we first show how component averaging can be used to combine the partial slowness from each node to form the next iteration. Later we discuss the working of distributed multigrid algorithm in detail.

Suppose there are N sensor nodes in the network and E ray paths are traced on each sensor node, following some earthquake events. From Section 2 the seismic tomography model will be of the form,

$$\boldsymbol{Ax} = \boldsymbol{B} \tag{12}$$

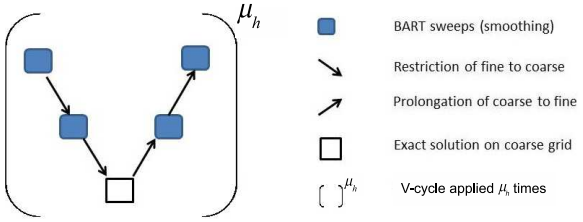


Fig. 4 V-cycle scheme for three levels.

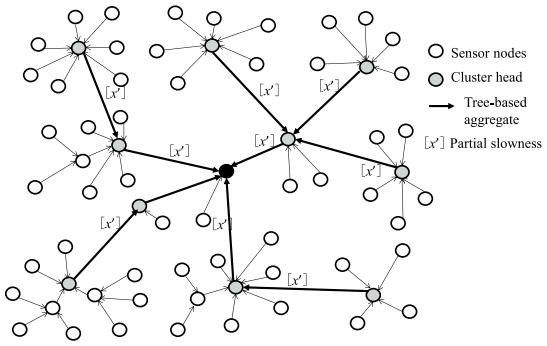


Fig. 5 Communication pattern for component averaging.

Let the size of $\boldsymbol{A}$ be $m \times n$, where $\sqrt{n}$ denotes the resolution we are calculating in case of 2-D. Let $\boldsymbol{A}_1, \boldsymbol{A}_2 \cdots, \boldsymbol{A}_N$ each contain $m_1, m_2, \cdots, m_E$ number of rows. Now in each node, we calculate the number of non-zero coefficients $\forall j$, where $1 \leqslant j \leqslant n$. Let I_j denote the index set of the blocks that contain an equation with a non-zero coefficient of x_j. Let $s_j = |I_j|$ (size of I_j).

We first show how the partial slowness obtained from each node can be combined with others using an averaging lemma. Let $\boldsymbol{A} = \boldsymbol{A}_1, \boldsymbol{A}_2, \cdots, \boldsymbol{A}_N$ and $\bar{x}_j^i$ denote the j-th component of partial slowness obtained from i-th node. The component averaging operator relative to $\boldsymbol{A}$ is transfer operator $\mathrm{CA}_{\boldsymbol{A}} : (\mathbb{R}^n)^N \to (\mathbb{R}^n)$ and defined as follows: Let $\bar{\boldsymbol{x}}^1, \cdots, \bar{\boldsymbol{x}}^N \in \mathbb{R}^n$ be partial solution from all N sensor nodes. Then $\mathrm{CA}_{\boldsymbol{A}}(\bar{\boldsymbol{x}}^1, \cdots, \bar{\boldsymbol{x}}^N)$ is the point in $\mathbb{R}^n$ whose j-th component is given by

$$\mathrm{CA}_{\boldsymbol{A}}(\bar{\boldsymbol{x}}^1, \cdots, \bar{\boldsymbol{x}}^N) = \frac{1}{s_j} \sum_{t=1}^{N} \bar{x}_j^t.$$

Assume that for some $1 \leqslant r \leqslant n$ the partial slowness $\bar{x}_1, \cdots, \bar{x}_r$ are shared by two or more nodes, i.e., $s_1, \cdots, s_r \geqslant 2$ and $s_{r+1}, \cdots, s_n = 1$. For simplicity, denote $\boldsymbol{y}$ as the components of $\mathbb{R}^s$, and index vector of $\mathbb{R}^s$ is given by

$$\boldsymbol{y} = (y_{1,1}, \cdots, y_{s,s_1}, \cdots, y_{r,1}, \cdots, y_{r,s_r}, y_{r+1}, \cdots, y_n).$$

Now we map the space from $E : \mathbb{R}^n \to \mathbb{R}^s$:

$$E(\bar{x}_1, \cdots, \bar{x}_n) =$$
$$(y_{1,1}, \cdots, y_{s,s_1}, \cdots, y_{r,1}, \cdots, y_{r,s_r}, y_{r+1}, \cdots, y_n).$$

We can see from the above equation that $(y_{1,1}, \cdots, y_{s,s_1}$ contains s_1 elements, $(y_{r,1}, \cdots, y_{r,s_r})$ contains s_r elements. Now after taking averages component-wise, we have our new update as follows:

$$\bar{x}_1 = \frac{1}{s_1}(y_{1,1} + \cdots + y_{s,s_1}),\ \bar{x}_r = \frac{1}{s_r}(y_{r,1} + \cdots + y_{r,s_r}),$$
$$\bar{x}_{r+1} = y_{r+1},\ \bar{x}_n = y_n.$$

Remark 3 We notice that, number of nodes N in component average scheme theoretically has no upper limit and can be very large. It should be noted that increasing N will increase the communication cost to carry out the summation over the network. This might also effect the rate of convergence and is shown in the simulation result.

Now we give the formal description of DMGT algorithm, see Algorithm 3.

Initialize lines 1–4: Suppose there are N sensors and each sensor initializes its ID and starting resolution d.

Algorithm 3 Distributed Multigrid Tomography

Initialize

1: Node ID id,
2: Initialize the starting resolution dimension d
3: Initialize the number of seismic sensors N
4: Current resolution dimension $Q = d \times d$
5: Initial slowness model for ray tracing $\boldsymbol{x}^l$

Repeat

1: **Upon the detection of an event**
2: Trace the ray path a_e for every node
3: **Upon the reception of $\vec{a}_e$ and b_e at each node start performing**
4: **calculation at each node**
5: For each $1 \leqslant j \leqslant Q$, calculate s_j
6: Where $s_j = |I_j| = \{1 \leqslant t \leqslant N | x_j$ has nonzero
7: coefficient in some equation of node N
8: $k \leftarrow 0, x^k \leftarrow \mathbf{0}$
9: **while** not converged **do**
10: In every node t for $1 \leqslant t \leqslant N$ do in parallel
11: $\bar{x}^t \leftarrow$ V-cycle(x^k, b^t)
12: Aggregate the partial slowness $\bar{x}^t$
13: from all nodes and find the next iterate:
14: $x_j^{(k+1)} = \begin{cases} \bar{x}_j^t, & \text{if } s_j = 1; \\ \frac{1}{s_j} \sum_{t=1}^{N} \bar{x}_j^t, & \text{if } s_j > 1. \end{cases}$
15: Send $x_j^{(k+1)}$ to all the node N
16: $k \leftarrow k + 1$
17: **end while**
18: $\mathbf{x}^l \leftarrow x^{(k-1)}$
19: **Upon the convergence obtaining final $\mathbf{x}^l$**
20: Update slowness model: $\mathbf{x}^{(l+1)} = \mathbf{x}^l$
21: TERMINATE

Let $Q = d \times d$ be the current tomography resolution where d is the initial resolution dimension. A slowness model $\boldsymbol{x}^l$ of resolution Q is used as an initial guess for ray tracing.

Repeat lines 1 and 2: After the initialization, each node will perform specific tasks based on the event detection and message reception. Once an event is detected by some node, the node will perform the ray tracing algorithm (assuming each node is aware of event location) and obtain the ray path. Then each node will compute the ray information forming a set of linear equation,

$$[\boldsymbol{A}_1, \boldsymbol{A}_2, \cdots, \boldsymbol{A}_N] \cdot [x_1, x_2, \cdots, x_Q]^{\mathrm{T}} = [b_1, b_2, \cdots, b_N],$$

where rows in $\boldsymbol{A}_i$ represents the ray information in each node, b_i represents the travel time residual of rays obtained by that node and x_j denotes the slowness of the j-th grid in a 2-D tomographic cube of dimension Q.

Repeat lines 3–18: Once all nodes compute the ray information, a parameter s_j for all $1 \leqslant j \leqslant Q$ is calculated by each of them. s_j is the number of nodes which has nonzero coefficient of particular x_j. Once s_j is calculated, each node simultaneously performs some finite number of Multigrid V-cycle (Algorithm 2). The next iteration is determined by component averaging

technique given by $x_j^{(k+1)} = \frac{1}{s_j} \sum_{t=1}^{N} \bar{x}_j^t$. Here a tree-based aggregation protocol is used to calculate the sum and broadcast back $x_j^{(k+1)}$ to all the nodes. The updated $x_j^{(k+1)}$ is used as an initial guess for the next iteration. A stopping criteria is used to stop distributed multigrid and the final slowness is sent to all the sensor nodes.

Repeat lines 19–21: Once a sensor node receives the final slowness model $\boldsymbol{x}^l$ from all N nodes it will update the previous slowness model $\boldsymbol{x}^{(l+1)} = \boldsymbol{x}^l$. The algorithm will TERMINATE if obtained result is satisfactory for the volcanologists to interpret. Otherwise the process is repeated with $\boldsymbol{x}^{(l+1)}$ as the slowness model to do ray tracing with same resolution or with higher depending upon the quality needed.

4.3 Communication cost

From the above algorithm we see that the actual communication in the network occurs in lines 12–15 which involves aggregation of partial slowness data of size n from all the nodes and then broadcasts the component averaged result back to each node. The communication scheme is shown in Fig. 5. Let $\boldsymbol{x}^i$ for $1 \leqslant i \leqslant N$ be the partial solution of node i. Let $|\boldsymbol{x}^i|$ denote the size of $\boldsymbol{x}^i$ given by n. Then the worst case communication cost involved would be $N \sum_i \dim(\boldsymbol{x}^i) = Nn$. Similarly after calculation of the component average, each node needs to flood the information to all other nodes which involves another Nn communication. Since this algorithm converges after k iterations, the worst case communication cost will be $2knN$.

In case of centralized computation we need to transfer the information from all the nodes to centralized server. Let m be average size of rows in each node, then the worst case communication cost involved for transferring data over the networks will be Nmn. The average number of events each node detects, i.e., m, will be of the order thousands or tens of thousands. Moreover, m is not constant and increases with occurrence of earthquake, therefore we see that $m \gg 2k$. Also, in lines 12–15 the communication involved is the summation of partial slowness over the network and the size of $\boldsymbol{x}^i$, i.e., n, remains constant and is cheaper than passing all the node information to a base station. Moreover, in centralized scenario if a node close to base station fails then lots of packets will be dropped making reconstruction impossible. While node failure in distributed case will only lead to loss of part of $\boldsymbol{A}$ and the reconstruction is possible using remaining data.

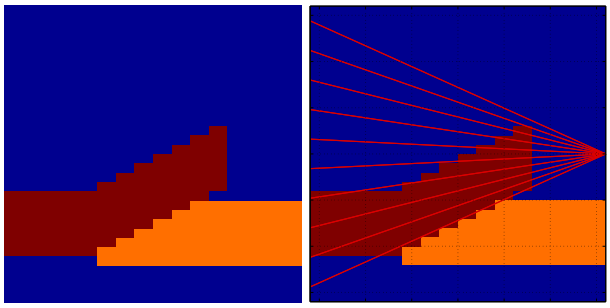
5 Evaluation and Validation

In this section, we evaluate the DMGT algorithm and present the simulation results. Typically, to test tomography inversion algorithm a synthetic model is used. This serves two purposes: (1) the real data set such as from Mt. St Helens does not have a ground truth and it is still uncertain which model is reliable. (2) The simulation using synthetic model enables us to investigate individually various phenomena which cannot be separated physically. For example, P wave data always contain noise due to measurement and scattering, but simulation can indicate the specific effect separately. For this reason, we adopt a synthetic data of a fault model from Ref. [30] which has been widely used for cross bore-hole tomography[31]. This fault model is created with velocities of 0.75 km/s for the right fault and 1.0 km/s for the left fault as shown in Fig. 6a. We perform the simulation in a customized simulator where we have implemented event detection, ray tracing, etc., for the fault model. The test cases and convergence measure used in computerized tomography are adopted to measure the volcano tomography as these two processes are similar.

5.1 Synthetic fault model

Our experiment setup has a network of 64 nodes which detects the earthquake event and traces the ray as shown in Fig. 6b. A total of 512 earthquake events are generated at random and a data generator traces the ray to obtain the travel time at each node. In practice these processes are independent and can be performed at each node distributedly. After this, we obtain $\boldsymbol{A}$ and $\boldsymbol{b}$ on each sensor node and we add Gaussian noise to $\boldsymbol{b}$ to simulate the measurement noise. We perform experiments for the finest resolution of dimension 32×32 with the three-grid V-cycle scheme. For the iterative methods, the selection of relaxation and regularization parameters ρ and λ respectively are critical and in all of our experiments these parameters remain constant throughout the iterations, i.e., $\rho^k = \rho = 0.25$ and $\lambda^k = \lambda = 5$ for all $k \geqslant 0$.

In the implementation, 5 sweeps of BART are performed at each level except the coarsest where it is solved directly. This adds up to a total of 20 iterations for a single V-cycle. For fair comparison we run 20 iterations at each node for CA-DMET. We use the relative slowness updates of the estimation between



(a) Magma/fault model (b) Ray traversal

Fig. 6 Synthetic model.

the two sweeps (one sweep means that all partial slowness are averaged to calculate the next iteration) as the stopping criteria. Convergence rates of different algorithms are compared using relative updates (ϕ), residuals (χ), and absolute error (ϵ) given by

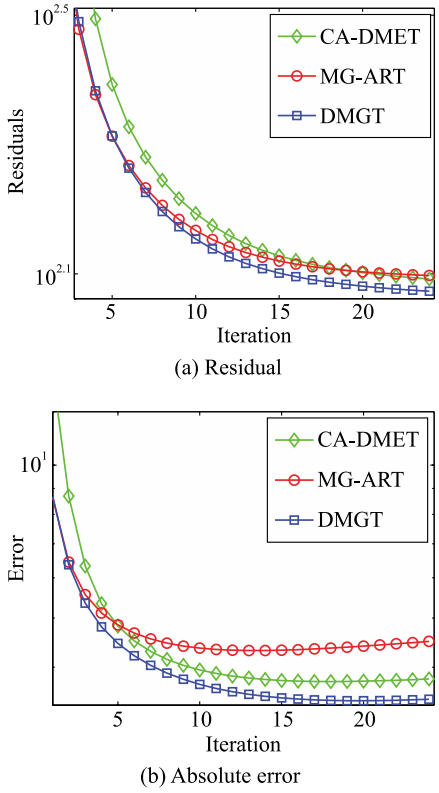
$$\begin{aligned} \phi &= |\boldsymbol{x}^{k+1} - \boldsymbol{x}^k| / |\boldsymbol{x}^k|, \\ \chi &= \|\boldsymbol{A}\boldsymbol{x}^k - \boldsymbol{b}\|, \\ \epsilon &= \|\boldsymbol{x}^* - \boldsymbol{x}^k\|, \end{aligned}$$

where $\boldsymbol{x}^*$ is the ground truth.

5.2 Correctness and accuracy

Firstly, we compare the relative performance of DMGT with two different algorithms: CA-DMET and MG-ART[28]. We use residuals and absolute error as the parameters for comparison and results are shown in Fig. 7. These plots demonstrate that there is a difference in the initial convergence behavior in these algorithms. Although the residuals of CA-DMET and MG-ART decrease at a similar rate, the absolute error of MG-ART tends to diverge from ground truth. This behavior is due to the lack of regularization parameter in this algorithm to handle inconsistent systems, whereas BART in DMGT takes care of this using appropriate λ. The iterations on x-axis denote the number of component averages required over a network, i.e., k as discussed earlier. We can see that DMGT converges faster (lesser k) compared to CA-DMET which means it requires lesser communication over the network.

A visual verification of these three algorithms is shown in Fig. 8. All the algorithms are run for the same number of iterations. The reconstructed images from different algorithms reveal that DMGT is able to obtain better reconstruction compared to other algorithms. We also observed that CA-DMET and DMGT algorithms continued to improve its image reconstruction with further increase in iterations, however MG-ART's reconstruction deteriorated with increase in iterations. This is also because of the inconsistent system as



(a) Residual

(b) Absolute error

Fig. 7 Comparing CA-DMET, MG-ART, and DMGT.

mentioned earlier.

5.3 Loss tolerance and performance

In the next set of experiments, loss tolerance and robustness of DMGT are evaluated. The algorithm runs with the same configuration for two different packet loss ratios of 10% and 40% in the simulator and the results are tabulated in Table 1. Figure 9 gives the 2-D tomography with packet loss and we can see that with 10% or even 40% packet loss, there is no significance difference in terms of the image reconstruction when compared to the results with no packet loss. Since the computation is distributed and all the nodes are involved in slowness calculation, the proposed algorithm is tolerant to a severe packet loss.

We also compare the efficiency of the algorithm by creating partitions and varying its size. Simulation results shown in Fig. 10 are run for total of 64 nodes, with partition number varying from 8 (each partition

Table 1 Robustness of DMGT.

Case	Relative error (ϕ)	Absolute error (ϵ)
No packet loss	0.0052	3.4606
10% packet loss	0.0386	3.5281
40% packet loss	0.0612	3.7411

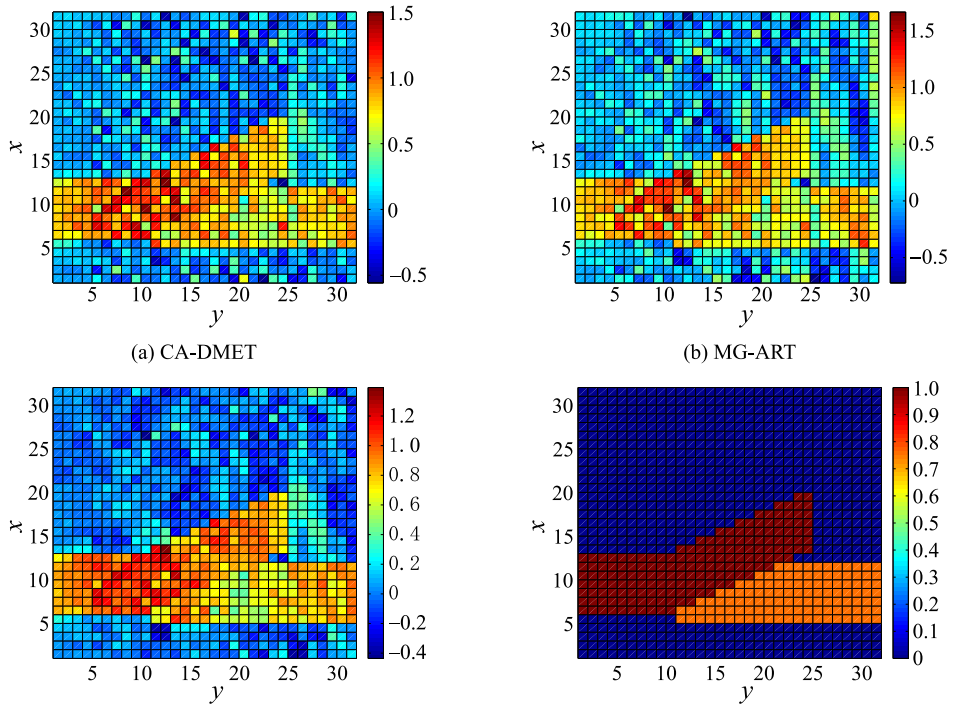


Fig. 8 Final tomography model obtained from different algorithms. The values along *x* and *y* axis denote the pixel count. The value at each pixel denotes the velocity (km/s) at that region and is given by the color bar on the right.

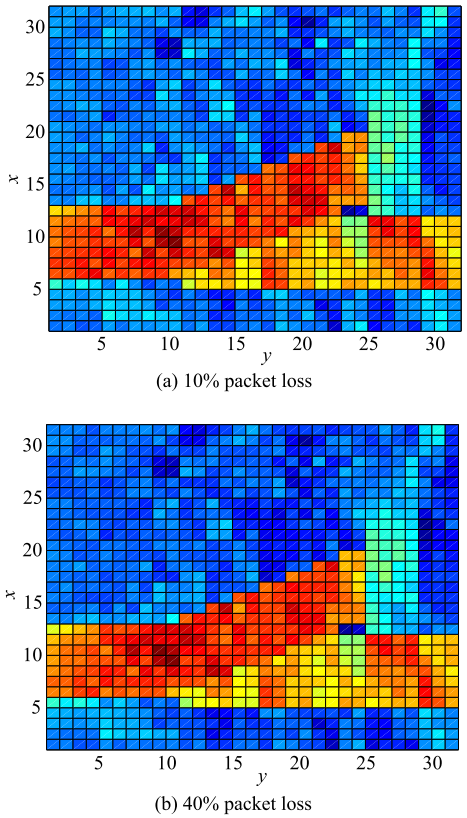


Fig. 9 DMGT under packet loss. *x* and *y* axes denote the pixel count.

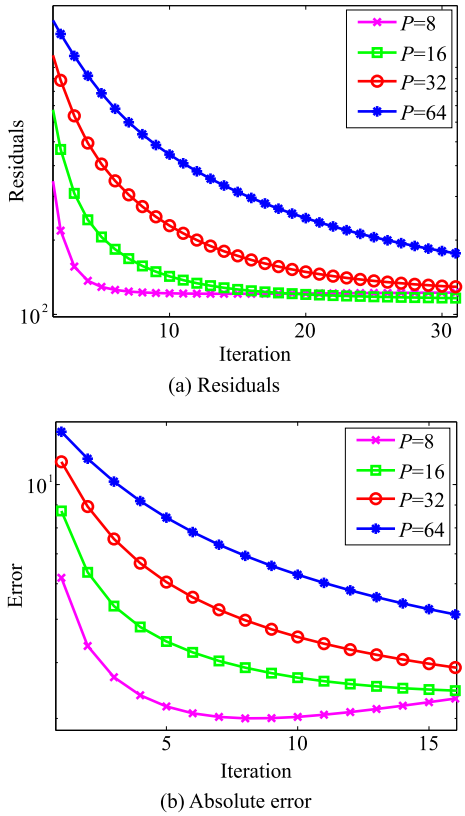
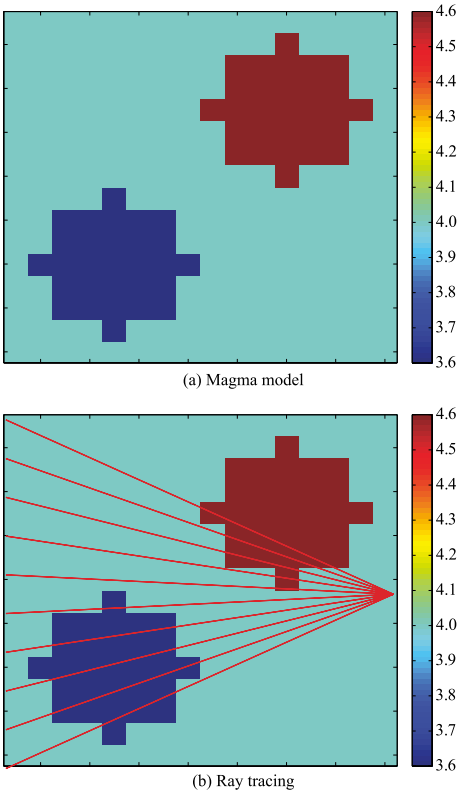


Fig. 10 DMGT with different partition sizes.

having 8 nodes) through 64 (each partition having one node). We can see that as partition number increases the convergence rate decreases. This is mainly due to the type of linear system each node has and the coefficient shared among the nodes. From this we can conclude that there is an optimal partition for a given set of nodes and given set of events. Also, in Fig. 10b we notice that for $P = 8$ case the solution diverges from ground truth. This phenomena is due to over smoothing/relaxation and to overcome this we need to dynamically select the parameters such as λ and ρ for a given partition size. We address these questions in our future work and it is beyond the scope of this paper.

5.4 Tomography with Magma model

Finally, we test the performance of DMGT using different synthetic models as shown in Fig. 11a. In this model we have Magma with velocity 4.5 km/s on top right, 3.5 km/s in the bottom left, and 4.0 km/s everywhere else. The difference in slowness is kept 15% compared to its surrounding value. This is because even in real nature the slowness does not vary more than $10\% - 15\%$ and difference more than this is treated to be an anomaly by geophysicists. Similar network comprising of 64 nodes is deployed which detects the earthquake event and traces the ray as shown in Fig. 11b. We compare the performance of DMGT with CA-DMET and CAV and the results are shown in Fig. 12. CAV was chosen over MG-ART as we saw earlier that MG-ART was not suitable in case of inconsistent systems. These plots again demonstrate that DMGT converges faster than CA-DMET and CAV because of the V-cycle scheme in DMGT. The convergence of absolute errors also show that DMGT gives result close to the ground truth. By testing DMGT on two different synthetic models and comparing its performance with other distributed algorithms we can say that it has accelerated convergence and requires relatively less number of communication to achieve similar results.



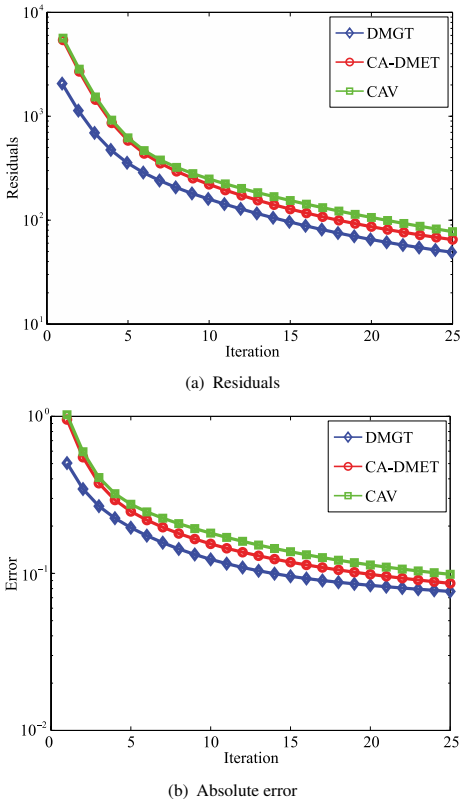
(a) Magma model

(b) Ray tracing

Fig. 11 Different synthetic models. Different synthetic model with velocities ranging from 3.5–4.5 km/s given by the color bar on the right hand side.

6 Conclusions

In this paper, we presented a new algorithm to solve the seismic tomography problem over the sensor network. We also proved that BART satisfies the smoothing property and can also be used as smoother in multigrid. We have also described the novel



(a) Residuals

(b) Absolute error

Fig. 12 DMGT with different synthetic models.

technique of performing multigrid in a distributed manner altogether forming the DMGT algorithm. This algorithm can distribute and balance the tomographic inversion computation load over the network, while computing real-time high-resolution tomography. The experimental evaluation also showed that our proposed method balances the computation load and is tolerant to data loss. Further enhancement of this algorithm can be done by applying the Full Approximation Scheme (FAS) and Full MultiGrid (FMG). Dynamic partitioning of clusters based on number of nodes and events can also be considered to improve the existing algorithm. With the introduction of embedded devices like raspberry pi and beagleboards which have computational power equivalent to a computer, we are now able to run these complex algorithms easily. Until now we have managed to build a mesh network with 20 sensors (beagleboards) and test the basic version of this algorithm and in future we will focus on using real data set for validation.

References

[1] W. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, and R. Lahusen, Air-dropped sensor network for real-time high fidelity volcano monitoring, in *The 7th Annual International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2009.

[2] G. P. Waite and S. C. Moranb, VP structure of Mount St. Helens, Washington, USA, imaged with local earthquake tomography, *Journal of Volcanology and Geothermal Research*, vol. 182, nos. 1&2, pp. 113–122, 2009.

[3] H. M. Iyer and P. B. Dawson, *Imaging Volcanoes Using Teleseismic Tomography*. Chapman and Hall, 1993.

[4] J. M. Lees, The Magma system of Mount St. Helens: Non-linear high-resolution P wave tomography, *Journal of Volcanology and Geothermal Research*, vol. 53, pp. 103–116, 1992.

[5] S. C. Moran, J. M. Lees, and S. D. Malone, P wave crustal velocity structure in the greater Mount Rainier area from local earthquake tomography, *Journal of Geophysical Research*, vol. 104, no. B5, pp. 10775–10786, 1999.

[6] G. Kamath, L. Shi, and W. Song, Component-average based distributed seismic tomography in sensor networks, in *IEEE DCOSS*, 2013.

[7] J. M. Lees and R. S. Crosson, Bayesian art versus conjugate gradient methods in tomographic seismic imaging: An application at Mount St. Helens, Washington, *Institute of Mathematical Statistics*, vol. 20, pp. 186–208, 1991.

[8] R. Sleeman and T. van Eck, Robust automatic P-phase picking: An on-line implementation in the analysis of broadband seismogram recordings, *Physics of the Earth and Planetary Interiors*, vol. 113, nos. 1–4, pp. 265–275, 1999.

[9] R. Tan, G. Xing, J. Chen, W. Song, and R. Huang, Quality-driven volcanic earthquake detection using wireless sensor networks, in *the 31st IEEE Real-Time Systems Symposium (RTSS)*, San Diego, CA, USA, 2010.

[10] L. Geiger, Probability method for the determination of earthquake epicenters from the arrival time only, *Bull. St. Louis. Univ*., vol. 8, pp. 60–71, 1912.

[11] M. T. Heath, E. Ng, and B. W. Peyton, Parallel algorithms for sparse linear systems, *SIAM Review*, vol. 33, no. 3, pp. 420–460, 1991.

[12] D. P. Bertsekas and J. N. Tsitsiklis, Some aspects of parallel and distributed iterative algorithms—A survey, *Automatica*, vol. 27, no. 1, pp. 3–21, 1991.

[13] I. D. Schizas, G. Mateos, and G. B. Giannakis, Distributed LMS for consensus-based in-network adaptive processing, *IEEE Transactions on Signal Processing*, vol. 57, no. 6, pp. 2365–2382, 2009.

[14] R. A. Renaut, A parallel multisplitting solution of the least squares problem, *Numerical Linear Algebra with Applications*, vol. 5, no. 1, pp. 11–31, 1998.

[15] S. Kaczmarz, Angenäherte Auflösung von Systemen linearer Gleichungen, *Bulletin International de l'Académie Polonaise des Sciences et des Lettres*, vol. 35, pp. 355–357, 1937.

[16] G. T. Herman, *Reconstruction from Projections: The Fundamentals of Computerized Tomography*. Academic Press, 1980.

[17] Y. Censor, D. Gordon, and R. Gordon, Component averaging: An efficient iterative parallel algorithm for large and sparse unstructured problems, *Parallel Computing*, vol. 27, no. 6, pp. 777–808, 2001.

[18] Y. Censor, D. Gordon, and R. Gordon, BICAV: A block-iterative parallel algorithm for sparse systems with pixel-related weighting, *IEEE Transaction on Medical Imaging*, vol. 20, no. 10, pp. 1050–1060, 2001.

[19] D. Gordon and R. Gordon, Component-averaged row projections: A robust, block-parallel scheme for sparse linear systems, *SIAM Journal on Scientific Computing*, vol. 27, pp. 1092–1117, 2005.

[20] J. M. Elble, N. V. Sahinidis, and P. Vouzis, GPU computing with Kaczmarz's and other iterative algorithms for linear systems, *Parallel Computing*, vol. 36, pp. 215–231, 2010.

[21] U. Trottenberg, C. Oosterlee, and A. Schuller, *Multigrid*. Academic Press, 2001.

[22] W. L. Briggs, V. E. Henson, and S. F. McCormick, A multigrid tutorial, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[23] U. M. Yang, *Parallel Algebraic Multigrid Methods—High Performance Preconditioners*, vol. 51. Springer-Verlag, 2006, pp. 209-236.

[24] E. Chow, R. D. Falgout, J. J. Hu, R. S. Tuminaro, and U. M. Yang, A survey of parallelization techniques for multigrid solvers, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2005.

[25] B. Smith, P. Bjorstad, and W. Gropp, *Domain Decomposition: Parallel Multi-level Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.

[26] H. Yserentant, On the multi-level splitting of finite element spaces, *Numer. Math.*, vol. 49, pp. 379–412, 1986.

[27] C. Popa, Algebraic multigrid smoothing property of Kaczmarz's relaxation for general rectangular linear systems, *Electronic Transactions on Numerical Analysis*, vol. 29, pp. 150–162, 2008.

[28] H. Kostler, C. Popa, and U. Rude, Algebraic multigrid for general inconsistent linear systems: The correction step, Technical report, Lehrstuhl für Informatik 10 (Systemsimulation), FAU Erlangen-N¨urnberg, 2006

[29] C. Popa and R. Zdunek, Kaczmarz extended algorithm for tomographic image reconstruction from limited-data, *Mathematics and Computers in Simulation*, vol. 65, pp. 579–598, 2004.

[30] P. C. Hansen and M. Saxild-Hansen, AIR tools—A MATLAB package of algebraic iterative reconstruction methods, *Journal of Computational and Applied Mathematics*, vol. 236, no. 8, pp. 2167–2178, 2012.

[31] A. Curtis and R. Snieder, Reconditioning inverse problems using the genetic algorithm and revised parameterization, *Geophysics*, vol. 62, no. 5, pp. 1524–1532, 1997.

Appendices

Proof of Theorem 1

Proof From the previous notations we have:

$$\boldsymbol{e}^k = \boldsymbol{x}^k - s(\boldsymbol{x}^0), k = 1, 2, \cdots, \bar{\boldsymbol{e}} = \boldsymbol{e}^m,$$

$$\boldsymbol{e}^k = \boldsymbol{x}^{(k-1)} + \lambda\rho^{(k-1)} \frac{(\lambda b_i - (y_i^{(k-1)} + \lambda \boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{x}^{(k-1)}))\boldsymbol{a}_i}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} - s(\boldsymbol{x}^0),$$

$$\boldsymbol{e}^k = \boldsymbol{e}^{(k-1)} - \lambda^2\rho^{(k-1)} \frac{r_i^k}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i - \lambda\rho^{(k-1)} \frac{y_i^{(k-1)}}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i,$$

$$\|\boldsymbol{e}^k\|^2 = \|\boldsymbol{e}^{k-1} - \lambda^2\rho^{(k-1)} \frac{r_i^{k-1}}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i\|^2 + |\lambda\rho^{(k-1)}|^2 \frac{y_i^{(k-1)}}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i - 2\langle \boldsymbol{e}^{(k-1)} - \lambda^2\rho^{(k-1)} \frac{r_i^{(k-1)}}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i, \lambda\rho^{(k-1)} \frac{y_i^{(k-1)}}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i \rangle =$$

$$\|\boldsymbol{e}^{k-1}\|^2 - 2\langle \boldsymbol{e}^{(k-1)}, \lambda^2\rho^{(k-1)} \frac{r_i^{k-1}}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i \rangle + (\lambda^2\rho^{(k-1)})^2 \frac{(r_i^{(k-1)})^2}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i^2 + (\lambda\rho^{(k-1)})^2 \frac{(y_i^{(k-1)})^2}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i^2 - 2\langle \boldsymbol{e}^{(k-1)}, \lambda\rho^{(k-1)} \frac{y_i^{(k-1)}}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i \rangle + 2\langle \lambda^2\rho^{(k-1)} \frac{r_i^{(k-1)}}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i, \lambda\rho^{(k-1)} \frac{y_i^{(k-1)}}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i \rangle =$$

$$\|\boldsymbol{e}^{k-1}\|^2 - (\lambda^2\rho^{(k-1)})^2 \frac{(r_i^{(k-1)})^2}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i^2 + (\lambda\rho^{(k-1)})^2 \frac{(y_i^{(k-1)})^2}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} \boldsymbol{a}_i^2 = \|\boldsymbol{e}^{(k-1)}\|^2 - \|\tilde{\boldsymbol{D}}^{\frac{1}{2}} \lambda^2 r_i^{(k-1)}\|^2 + \|\tilde{\boldsymbol{D}}^{\frac{1}{2}} y_i^{(k-1)}\|^2$$

where: $\tilde{\boldsymbol{D}} = \mathrm{diag}\left(\frac{\rho^{(k-1)} \boldsymbol{a}_i^2}{1 + \lambda^2 \|\boldsymbol{a}_i\|^2} \right)$

Since, $\|\tilde{\boldsymbol{D}}^{\frac{1}{2}} y_i^{(k-1)}\|^2 \geqslant 0$, from Eq. (8)

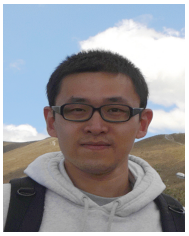
we know, $\|\bar{\boldsymbol{e}}\|^2 \leqslant \|\boldsymbol{e}\|^2 - \tilde{\gamma} \|\tilde{\boldsymbol{D}}^{\frac{1}{2}} r\|^2$.

Therefore

$$\boldsymbol{e}^k \leqslant \|\boldsymbol{e}^{(k-1)}\|^2 - \|\tilde{\boldsymbol{D}}^{\frac{1}{2}} \lambda^2 r_i^{(k-1)}\|^2 + \|\tilde{\boldsymbol{D}}^{\frac{1}{2}} y_i^{(k-1)}\|^2$$

We have now shown that BART satisfies the smoothing property and can be used as smoother in multigrid to solve the tomography problem. ■

Goutham Kamath received his BEng degree from India in 2009 and MS degree in electrical engineering from University of Wyoming in 2012. He is currently pursuing his PhD degree in the Department of Computer Science, Georgia State University. His research interests include wireless sensor networks, distributed systems, and big data computing.



Lei Shi received his BS degree from Tongji University in 2007, and the MS degree in computer science from Shanghai Jiao Tong University in 2010. He is currently pursuing the PhD degree in the Department of Computer Science at Georgia State University. His research interests include wireless sensor networks, in-network processing, and distributed systems.

Edmond Chow got his BASc degree from University of Waterloo in 1993 and PhD degree from University of Minnesota in 1997. After graduating he worked in LLNL until 2005 and later worked at D. E. Shaw Research until 2010. He is currently an associate professor at Georgia Tech. His research interest includes developing and applying numerical methods and high-performance computing to solve large-scale scientific problems.

Junjie Yang received his BEng and MS degrees both from Changchun University of Science and Technology in 1998 and 2001, respectively, and the PhD degree from Shanghai Jiao Tong University in 2005. He is currently a full professor of College of Electronics and Information Engineering, Shanghai University of Electric Power. His research interest includes wireless sensor networks, optical networks, and smart grid.

Wenzhan Song received his BS and MS degrees both from Nanjing University of Science and Technology in 1997 and 1999, respectively, and the PhD degree from Illinois Institute of Technology in 2005. He is currently a full professor in Georgia State University. His research mainly focuses on sensor web, smart grid, and smart environment where sensing, computing, communication and control play a critical role and need a transformative study. His research has received 6 million+ research funding from NSF, NASA, USGS, Boeing, etc. since 2005, and resulted in 80+ journal articles, conference articles, and book chapters in this area.