# Distributed Traveltime Tomography Using Kernel-based Regression in Seismic Networks

Ban-Sok Shin and Dmitriy Shutin

***Abstract*—Distributed subsurface imaging is of high relevance for autonomous seismic surveys by multi-agent networks as envisioned for future planetary missions. The goal is to achieve a cooperative reconstruction of a subsurface image at each agent by relying on data exchange among the agents. To this end, distributed full waveform inversion for high-resolution imaging has been proposed. However, full waveform inversion always requires an initial model of the subsurface. To provide each agent in the network with such a model, we propose a distributed traveltime tomography. To this end, we integrate a distributed kernel-based regression of traveltime residuals into traveltime tomography. By that, each agent computes an approximation of all time residuals in the network and can perform a traveltime tomography to obtain a subsurface image locally. We conduct numerical evaluations for a synthetic subsurface model and the SEG salt model. The results show that each receiver indeed achieves a subsurface image that is close to the global result even for a low network connectivity.**

***Index Terms*—Distributed seismic imaging, travel time tomography, inverse problems, seismic velocity analysis, multi-agent seismic exploration**

## I. Introduction

SEISMIC data acquisition on planets has become increasingly relevant during the last decade. Current missions like InSight or Mars2020 have seismic instruments on their rovers to record seismic data from Mars [1]. These data are decisive to gain more insight into the Martian subsurface. Concepts of multiple rovers have been proposed for future planetary missions where rovers autonomously explore a subsurface on Moon or Mars [2]. Here, near-surface objects such as cavities and lava tubes are of interest since these can serve as habitats for equipment or humans. However, to enable an autonomous exploration without the dependence on a central entity, distributed subsurface imaging is required that provides each rover with an estimate of the subsurface. Using these estimates rovers can then be steered to new sampling positions to improve the image. A well-known geophysical imaging technique for near-surface problems is traveltime tomography (TT) [3]. It is used to obtain a first, usually low resolution image of the subsurface that can be refined by e.g. full waveform inversion (FWI) [4], [5]. In a classical setting, TT operates in a centralized fashion, i.e., all receiver data are collected at one central entity that performs the inversion of the data. Thus, to employ TT in a multi-agent network without a central data collector, distributed techniques are required. Distributed tomographic algorithms have been considered in the past, see e.g. [6], [7]. However, all of these works require either sharing intermediate data with a central master node or sharing of partial tomographic maps over central cluster nodes. Different from these works we proposed the adapt-then-combine full waveform inversion (ATC-FWI) in [8]. Here, the ATC technique [9] enables each receiver in the network to obtain an estimate of the global image by cooperating with neighboring receivers. However, to obtain a complete subsurface imaging scheme, the ATC-FWI still requires an appropriate starting model that needs to be provided to each receiver in a distributed fashion.

To equip each receiver with a starting model we develop a distributed traveltime tomography, where each receiver obtains an estimate of the global TT without a central entity. This can be achieved if residual data, i.e. the difference between measured and reconstructed traveltimes, are provided to each receiver. We incorporate a distributed regression into the tomography that provides each receiver with an approximation of these traveltime residuals. Numerical evaluations show that each receiver obtains an image close to the global tomography image. Even though the estimated residuals do not perfectly match the true residuals, our proposed scheme achieves model misfits similar to the global performance.

B. Shin and D. Shutin are with the Department of Communication Systems, Institute of Communications and Navigation, German Aerospace Center, Wessling, Germany. E-mail: ban-sok.shin@dlr.de, dmitriy.shutin@dlr.de.

Manuscript received ... 2022; revised ... 2022.

## II. Brief review of traveltime tomography

Traveltime tomography is an imaging technique that obtains a $P$-wave velocity model $m(\boldsymbol{x})$ as a scalar function over the spatial coordinate $\boldsymbol{x} = (x, z)$ in the subsurface domain $\Omega \subset \mathbb{R}^3$. It relies on first-arrival times of propagating waves from one source to $N_\mathrm{R}$ receivers. TT minimizes the squared residual between measured traveltimes $T_{\mathrm{obs},r}$ and synthesized traveltimes $T_{\mathrm{syn},r}(m)$ per receiver $r$ with respect to the velocity $m(\boldsymbol{x})$:

$$\min_m \mathcal{L}(m) = \frac{1}{2}\sum_{r=1}^{N_\mathrm{R}} |T_{\mathrm{syn},r}(m) - T_{\mathrm{obs},r}|^2 \qquad (1)$$

The measured traveltimes $T_{\mathrm{obs},r}$ are picked either automatically or by hand from the seismogram at each receiver $r = 1, \ldots, N_\mathrm{R}$. $T_{\mathrm{syn},r}(m)$ is the synthetically computed traveltime at receiver $r$ using the estimated subsurface model $m$. To obtain $T_{\mathrm{syn},r}(m)$ the *eikonal equation* can be used [10]:

$$|\nabla T(\boldsymbol{x})|^2 = \frac{1}{m(\boldsymbol{x})^2}, \quad \text{s.t.} \quad T(\boldsymbol{x}_s) = 0, \boldsymbol{x} \in \Omega, \qquad (2)$$

where $\boldsymbol{x}_s \in \Omega$ is the source position and $\nabla$ is the spatial gradient operator. The solution to the eikonal equation provides an estimate of traveltimes $T(\boldsymbol{x})$ as a function over the

spatial coordinate $\boldsymbol{x}$. The initial condition at source location $\boldsymbol{x}_s$ ensures that the traveltime will be zero there. To solve (2) various numerical methods exist [11], [12]. In our implementation, we use the fast sweeping method due to its simplicity in implementation.

To minimize cost (1) subject to (2), TT employs an iterative, gradient-based minimization scheme. To compute the gradient $d\mathcal{L}(m)/dm$, traditionally ray tracing methods have been used that linearize the forward calculation of traveltimes. However, ray tracing techniques can become very cumbersome to implement. Therefore, the use of the *adjoint state method* for gradient computation has been proposed [13]. This method is a general tool to compute gradients in optimization problems constrained by a parameterized equation. This is achieved by introducing a new variable - the adjoint-state - that accounts for the constraint, cf. [14]. Following [10], in our case the adjoint state variable $\lambda$ is obtained by solving

$$\nabla \cdot \lambda(\boldsymbol{x}) \nabla T(\boldsymbol{x}) = 0 \tag{3}$$

with additional boundary condition at receiver positions $\boldsymbol{x}_r$

$$\begin{aligned}(\boldsymbol{n} \cdot \nabla T(\boldsymbol{x}_r))\lambda(\boldsymbol{x}_r) = T_{\mathrm{syn},r}(m) - T_{\mathrm{obs},r}, \\ r = 1, \ldots, N_{\mathrm{R}},\end{aligned} \tag{4}$$

where $\boldsymbol{n}$ is the unit vector normal to the surface of domain $\Omega$. According to (4) we initialize the adjoint state variable with a scaled version of the time residuals $T_{\mathrm{syn},r}(m) - T_{\mathrm{obs},r}$ at each receiver position $\boldsymbol{x}_r$. To compute $\nabla T(\boldsymbol{x}_r)$ one can use one-sided finite differences for the discretization of the gradient operator. Then (3) is solved for the inner part of $\Omega$ by using the fast sweeping method as detailed in [10], [13].

After obtaining the adjoint state $\lambda$ the gradient of $\mathcal{L}(m)$ can be computed as $-\lambda(\boldsymbol{x})/m(\boldsymbol{x})^3$ using the Lagrangian multiplier method, cf. [10]. However, in [13] the authors that such gradient computation leads to instabilities due to non-convexity of the cost function. To obtain a better conditioned gradient, the authors derive the following gradient calculation in [13]:

$$(I - \nu\Delta) m_\Delta = -\left(\frac{\lambda}{m^3}\right) \tag{5}$$

where $m_\Delta$ is a smoothed gradient of the cost $\mathcal{L}(m)$, $I$ is the identity operator, $\nu \geq 0$ is a parameter that controls smoothness of the solution and $\Delta$ is the spatial Laplace operator. Again, to solve (5) wrt. $m_\Delta$, e.g. finite differences can be used to discretize the Laplace operator. We point out that the gradient calculation (5) is valid for one source only. In case of $N_{\mathrm{S}}$ sources a separate adjoint state variable $\lambda_s$ needs to be computed for each source $s$ according to (3) and (4). Then the gradient $m_\Delta$ can be computed as a solution to

$$(I - \nu\Delta) m_\Delta = -\sum_{s=1}^{N_{\mathrm{S}}} \left(\frac{\lambda_s}{m^3}\right). \tag{6}$$

With the gradient $m_\Delta$, an iterative scheme such as gradient descent can be used to update the model $m^{[k]}$ at iteration $k$:

$$m^{[k+1]} = m^{[k]} - \mu \cdot m_\Delta \tag{7}$$

with step size $\mu > 0$. In each iteration $k$, traveltimes $T_{\mathrm{syn},r}$, adjoint state $\lambda_s$, and gradient $m_\Delta$ are computed based on $m^{[k]}$.

## III. Distributed traveltime tomography

Traditional TT is a centralized scheme: it assumes availability of measured traveltimes $T_{\mathrm{obs},r}$ of all receivers at one central entity to solve (3) and (4) for $\lambda$. In the following, we propose a method that computes $m(\boldsymbol{x})$ at each receiver without access to all traveltimes but by data exchange among receivers.

### A. Seismic network model

First, we introduce the seismic network model to define topology of the seismic network of geophones. We consider a seismic network of $N_{\mathrm{R}}$ geophone receivers. We describe the topology of the network by a graph $\mathcal{G} = \{\mathcal{R}, \mathcal{E}\}$ with a set of nodes $\mathcal{R} = \{1, 2, \ldots, N_{\mathrm{R}}\}$ for the receivers and a set of edges $\mathcal{E} = \{(r, \ell) | r, \ell \in \mathcal{R}\}$ for wireless connections among the receivers. The graph $\mathcal{G}$ is undirected and strongly connected, i.e., each receiver can be reached by any other receiver in the network over multiple hops. Each receiver $r$ has a neighborhood set $\mathcal{N}_r$ that contains all receivers that can exchange data with receiver $r$ including receiver $r$ itself. Moreover, each receiver $r$ is located at a fixed Cartesian position denoted by $\boldsymbol{x}_r = (x_r, z_r), r = 1, \ldots, N_{\mathrm{R}}$ with the $x$- and $z$-coordinate. Fig. 1 illustrates the seismic network for a line topology common in 2D seismic surveys.

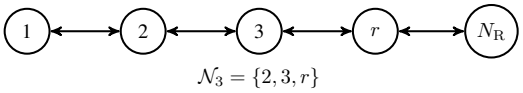


Fig. 1: Network of $N_{\mathrm{R}}$ nodes with neighborhood set $\mathcal{N}_3$ of node 3.

### B. Distributed kernel-based regression of time residuals

To obtain a subsurface model $m$ the residuals $\tau_r = T_{\mathrm{syn},r}(m) - T_{\mathrm{obs},r}$ of all receivers are decisive. If these are available locally at each receiver $r$, the adjoint state $\lambda$ in (3) and the gradient $m_\Delta$ in (5) can be computed locally and by that, a tomographic model update can be performed. To make these time residuals accessible by each receiver we use a distributed regression scheme that provides each receiver with an approximation of the time residuals only by data exchange among connected receivers. The distribution of time residuals over the receiver position $\boldsymbol{x}_r$ is highly nonlinear. Thus, we employ the kernel distributed consensus-based estimation (KDiCE) algorithm proposed in [15] that enables a distributed regression of nonlinear functions.

In kernel-based regression a linear combination of kernels $\kappa(\cdot, \cdot)$ is used to approximate a nonlinear function. Among various kernels the Gaussian kernel is the most popular one due to its universal approximation property [16]. For two input samples $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ the Gaussian kernel is defined as $\kappa(\boldsymbol{x}_1, \boldsymbol{x}_2) = \exp\left(-||\boldsymbol{x}_1 - \boldsymbol{x}_2||_2^2/(2\sigma^2)\right)$ where $\sigma$ is the bandwidth that controls the width of the kernel shape. To approximate a nonlinear function, a linear combination of kernel evaluations for a set of input samples is used. In our case, the input samples are the receiver positions $\{\boldsymbol{x}_r\}_{r=1}^{N_{\mathrm{R}}}$. Then a kernel-based approximation at input $\boldsymbol{x}$ is performed via the inner product $\boldsymbol{g}(\boldsymbol{x})^{\mathsf{T}}\boldsymbol{w}$ where $\boldsymbol{w} \in \mathbb{R}^{N_{\mathrm{R}}}$ contains combination

weights and $\boldsymbol{g}(\boldsymbol{x}) = [\kappa(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, \kappa(\boldsymbol{x}, \boldsymbol{x}_{N_\mathrm{R}})]^\mathsf{T} \in \mathbb{R}^{N_\mathrm{R}}$ consists of kernel evaluations between a position $\boldsymbol{x}$ and receiver positions $\{\boldsymbol{x}_r\}_{r=1}^{N_\mathrm{R}}$.

The quantity to be approximated at each receiver is the vector of all residuals in the network: $\boldsymbol{\tau} = [\tau_1, \tau_2, \ldots, \tau_{N_\mathrm{R}}]^\mathsf{T} \in \mathbb{R}^{N_\mathrm{R}}$. To apply kernel-based regression for the distributed approximation of $\boldsymbol{\tau}$, we introduce a weight vector $\boldsymbol{w}_r \in \mathbb{R}^{N_\mathrm{R}}$ per receiver $r$. This weight vector defines the receiver-specific combination weights for the regression, i.e., each receiver obtains its own approximation of the time residuals $\boldsymbol{\tau}$. Following [15] we formulate the distributed regression problem

$$\{\boldsymbol{w}_r^\star | r \in \mathcal{R}\} = \underset{\{\boldsymbol{w}_r | r \in \mathcal{R}\}}{\operatorname{argmin}} \; ||\boldsymbol{\tau} - \boldsymbol{G}\boldsymbol{w}_r||_2^2 \tag{8a}$$

$$\text{s.t.} \quad \boldsymbol{w}_r = \boldsymbol{w}_\ell, \quad r \in \mathcal{R}, \ell \in \mathcal{N}_r. \tag{8b}$$

Matrix $\boldsymbol{G} \in \mathbb{R}^{N_\mathrm{R} \times N_\mathrm{R}}$ is a symmetric Gram matrix where the $r$-th row is the vector $\boldsymbol{g}(\boldsymbol{x}_r)^\mathsf{T} = [\kappa(\boldsymbol{x}_r, \boldsymbol{x}_1), \ldots, \kappa(\boldsymbol{x}_r, \boldsymbol{x}_{N_\mathrm{R}})]$ with $r = 1, \ldots, N_\mathrm{R}$. Problem (8a) aims at approximating $\boldsymbol{\tau}$ in the least-squares sense via adaptation of $\boldsymbol{w}_r$. The matrix-vector product $\boldsymbol{G}\boldsymbol{w}_r$ then gives an approximation of the residuals $\boldsymbol{\tau}$. Constraint (8b) enforces equality between weights of receiver $r$ and of neighboring receiver $\ell$. It is a consensus constraint ensuring that all weight vectors converge to the same solution. Problem (8) can be solved with the KDiCE algorithm proposed in [15]. It is based on the alternating direction method of multipliers (ADMM) to achieve a distributed solution of $\boldsymbol{w}_r$. The algorithm consists of the following update equations (see [15] for the derivation):

$$\boldsymbol{z}_r^{[k+1]} = \frac{\varepsilon}{|\mathcal{N}_r|} \sum_{\ell \in \mathcal{N}_r} \frac{\boldsymbol{w}_\ell^{[k]}}{\varepsilon} - \boldsymbol{u}_{\ell r}^{[k+1]}, \tag{9a}$$

$$\boldsymbol{u}_{r\ell}^{[k+1]} = \boldsymbol{u}_{r\ell}^{[k]} - \left(\boldsymbol{w}_r^{[k]} - \boldsymbol{z}_\ell^{[k+1]}\right)/\varepsilon, \tag{9b}$$

$$\boldsymbol{w}_r^{[k+1]} = \left[\boldsymbol{g}(\boldsymbol{x}_r)\boldsymbol{g}(\boldsymbol{x}_r)^\mathsf{T} + \frac{|\mathcal{N}_r|}{\varepsilon}\boldsymbol{I}_{N_\mathrm{R}}\right]^{-1} \times \left(T_{\mathrm{obs},r}\boldsymbol{g}(\boldsymbol{x}_r) + \sum_{\ell \in \mathcal{N}_r} \frac{\boldsymbol{z}_\ell^{[k+1]}}{\varepsilon} + \boldsymbol{u}_{r\ell}^{[k+1]}\right). \tag{9c}$$

Parameter $\varepsilon > 0$ is a positive step size of the ADMM that controls priority to achieving consensus in the network. Variable $\boldsymbol{z}_r^{[k]}$ is an auxiliary variable that acts as an intermediate estimate of the weight vector $\boldsymbol{w}_r^{[k]}$ while $\boldsymbol{u}_{ri}^{[k]}$ is a Lagrange multiplier that ensures convergence to a consensus solution over the network. In each iteration $k$, receiver $r$ needs to exchange variables $\boldsymbol{z}_r^k, \boldsymbol{w}_r^{[k]}$ and $\boldsymbol{u}_{ri}^{[k]}$ with receivers in its neighbor set $\mathcal{N}_r$. With (9a) - (9c) all weight vectors $\boldsymbol{w}_r$ converge to the central least squares solution of (8a). After obtaining weight vectors $\boldsymbol{w}_r^{[k]}$, we can approximate the time residuals in $\boldsymbol{\tau}$ via $\widehat{\boldsymbol{\tau}}_r = \boldsymbol{G}\boldsymbol{w}_r^{[k]}$ per receiver $r$. With the approximated time residuals, each receiver $r$ is now able to compute a gradient $m_{\Delta,r}$ and update its subsurface model $m_r$ locally. For parameter selection, the kernel bandwidth $\sigma$ needs to be small enough to capture variations of the residuals over the receiver positions. Here, values in the range of $\sigma = 1$ give a sufficient approximation. For step size $\varepsilon$ in ADMM, values in the area of 100 are necessary to give a good regression. Lower values give priority to satisfying the consensus constraint first, cf. (9b), and might slow down convergence.

**Algorithm 1** Distributed Traveltime Tomography (D-TOMO)

```
Set parameters σ, ε, μ, ν
Initialize starting models m_1^[0] = m_2^[0] = ... = m_{N_R}^[0]
for TOMO iteration k ← 1, N_TOMO do
    for receiver r ← 1, N_R do
        Get traveltimes T_{syn,r} per source with (2)
        Get residual τ_r = T_{syn,r} − T_{obs,r} per source
    end for
    for source s ← 1, N_S do                ▷ Distributed regression
        for KDiCE iteration [k] ← 1, N_KDiCE do
            Compute z_r^[k] per receiver, exchange with N_r
            Compute u_rℓ^[k] per receiver, exchange with N_r
            Compute w_r^[k] per receiver, exchange with N_r
        end for
        for receiver r ← 1, N_R do
            Approximate residuals τ̂_r = G w_r^[N_KDiCE]
            Store residuals τ̂_r per source
        end for
    end for
    for receiver r ← 1, N_R do              ▷ Local model update
        Compute adjoints λ_{r,s} per source with (3) and (4)
        Compute gradient m_{Δ,r} with (10a)
        Update model: m_r^[k+1] = m_r^[k] − μ · m_{Δ,r}
    end for
end for
return Local model m_r^[N_TOMO], ∀r ∈ R
```

### C. Receiver-specific traveltime tomography

We now describe the key steps of the proposed algorithm with the pseudo-code summarized in Algorithm 1. Based on the cooperatively estimated residuals $\widehat{\boldsymbol{\tau}}_r$ we can compute a local adjoint state variable $\lambda_{r,s}(\boldsymbol{x})$ for each shot position $\boldsymbol{x}_s$ at receiver $r$. Each receiver computes a traveltime map $T_{\mathrm{syn},r}(\boldsymbol{x})$ over the complete domain $\Omega$ based on its local subsurface model $m_r$. For the initial state we need to assume a starting model for $m_r$, e.g. a vertical gradient between a minimum and maximum velocity. Then each receiver determines its local time residual $\tau_r = T_{\mathrm{syn},r} - T_{\mathrm{obs},r}$ and the distributed regression of these residuals is performed following (9a) - (9c). After that, each receiver initializes its adjoint state variable $\lambda_r$ according to (4) where the right-hand side of the equation now employs the estimated time residuals $\widehat{\boldsymbol{\tau}}_r$. Then the complete adjoint $\lambda_r$ is computed using the fast sweeping algorithm as detailed in [10]. For multiple source positions, we compute one adjoint state variable $\lambda_{r,s}$ per source $s$. Then for $N_\mathrm{S}$ sources gradient computation and local model update per receiver $r$ follow

$$(I - \nu\Delta)m_{\Delta,r} = -\sum_{s=1}^{N_\mathrm{S}} \frac{\lambda_{r,s}}{m_r^3}, \tag{10a}$$

$$m_r^{[k+1]} = m_r^{[k]} - \mu \cdot m_{\Delta,r}. \tag{10b}$$

After the local model update, each receiver synthesizes new traveltimes $T_{\mathrm{syn},r}(\boldsymbol{x})$ and updates its local residual $\tau_r$. This initiates a new cycle with a distributed regression of traveltime residuals and a local tomography update. To guarantee stable convergence, step size $\mu$ needs to be chosen according to

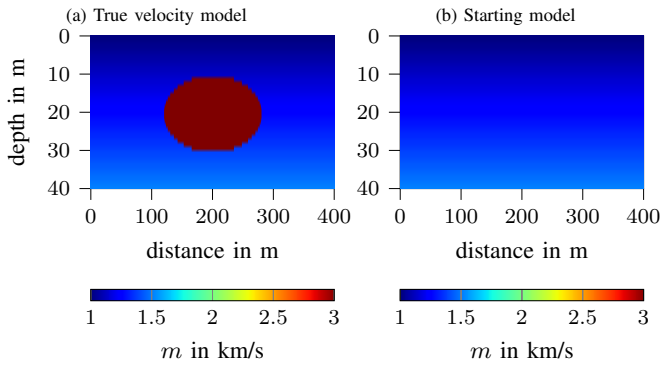


Fig. 2: (a) True velocity model and (b) background velocity as starting model.

the underlying $P$-wave velocity model. Since values in the gradient $m_{\Delta,r}$ can be very small, normalization to $[-1, 1]$ is helpful. Then $\mu$ can be in the range of 100 - 500 if $P$-wave velocities reach several $1000\,\mathrm{m/s}$. For $\nu$ we found that values in the range of $[10^3, 10^6]$ give good results for the gradient.

## IV. Numerical Evaluation

### A. Synthetic elliptic velocity model

We show numerical results of our proposed distributed tomography (D-TOMO) algorithm for synthetic traveltime data. To this end, we generate an isotropic subsurface model for the $P$-wave velocity that contains an elliptic anomaly while the background consists of a vertical velocity gradient from $1-1.5\,\mathrm{km/s}$, see Fig. 2a. For the starting model we use the background velocity with the vertical gradient, see Fig. 2b. Each receiver in the network has the same starting model available locally. To image the subsurface we use a line array of $N_\mathrm{R} = 20$ receivers and $N_\mathrm{S} = 20$ shot positions. Each receiver is connected to a maximum of two neighboring receivers, i.e., only the first and last receiver in the line topology have one neighbor to their left- and right-hand side. All receivers and sources are located on the surface $z = 0\,\mathrm{m}$. For the numerical implementation we use a finite-difference scheme with a grid-spacing of $\Delta x = \Delta z = 1\,\mathrm{m}$. We compare D-TOMO to its central variant (TOMO). For both methods we use $N_\mathrm{TOMO} = 20$ iterations where D-TOMO uses an additional $N_\mathrm{KDiCE} = 100$ iterations in each cycle for the distributed regression of the residuals $\boldsymbol{\tau}$. This number of iterations for KDiCE was necessary to obtain a good regression of the residuals. A lower number would lead to worse imaging results. However, since the regression is done in each tomography iteration, this number directly influences the communication load within the network. Furthermore, we normalize the gradient in the model update (7) to $[-1, 1]$ and use an exponentially decaying step size over the iterations starting at $\mu = 250$. In the distributed regression stage, we set $\varepsilon = 100$ and the kernel bandwidth to $\sigma = 1$. For gradient computation in (6) we set $\nu = 10^6$.

Fig. 3 depicts the imaging results at three receivers in the line array. As reference, we show the image of the centralized tomography. The images by D-TOMO are very close to the central result. For receiver 1 and 20 slight deviations can be

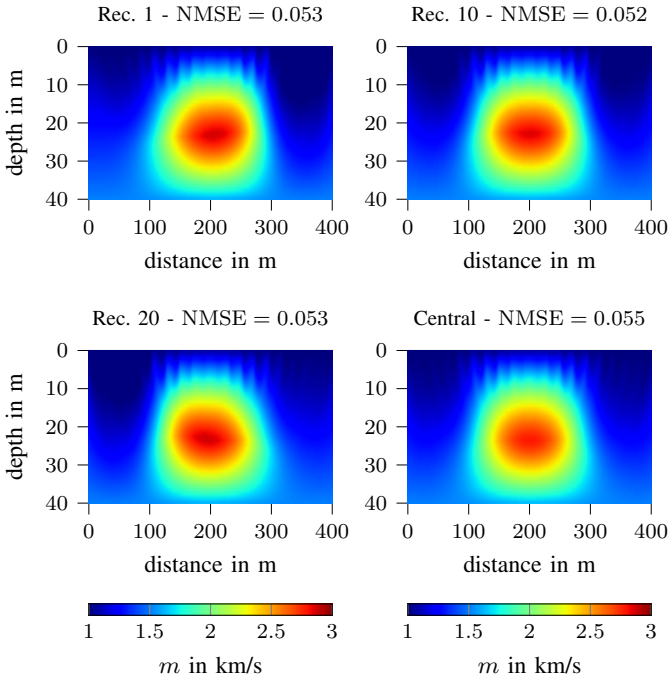


Fig. 3: Imaging results of D-TOMO at receivers $1, 10, 20$ in the line array and image of centralized tomography as reference after $N_\mathrm{TOMO} = 20$ iterations.

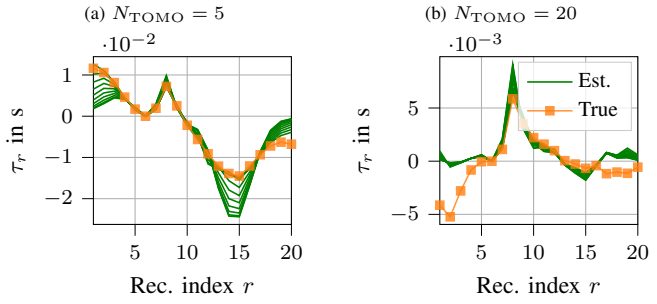


Fig. 4: True and estimated time residuals of 10 receivers in the line array after 5 and 10 iterations.

observed due to having only one neighboring receiver. Nevertheless, their imaging result is still remarkably close to the central result. This is also reflected in the similar misfit numbers that compares the estimated model to the true one computed via $\mathrm{NMSE} = \frac{1}{N_\mathrm{R}} \sum_{r=1}^{N_\mathrm{R}} ||m_r^{[k]} - m_\mathrm{true}||_2^2 / ||m_\mathrm{true}||_2^2$. Fig. 4 depicts the estimated and true time residuals over the receiver index for 10 receivers after 5 and 20 iterations. In both cases,

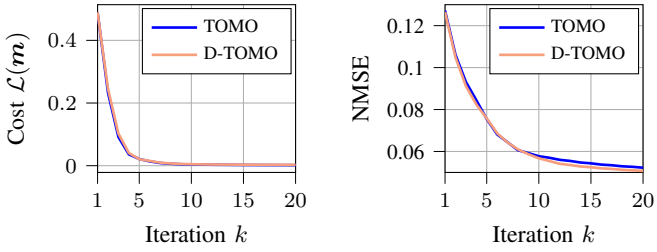


Fig. 5: Cost (left) and normalized misfit (right) of distributed (D-TOMO) and centralized tomography (TOMO).

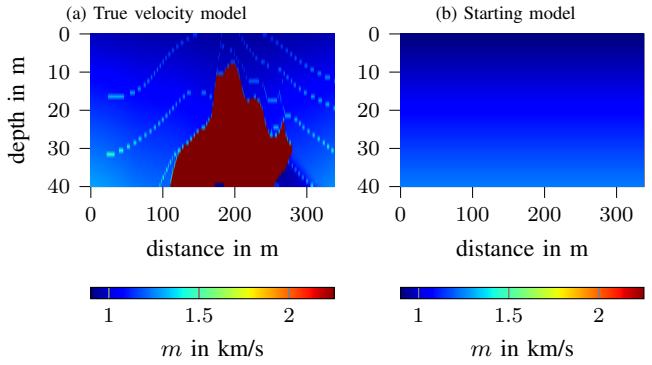


Fig. 6: (a) True velocity model and (b) background velocity as starting model.

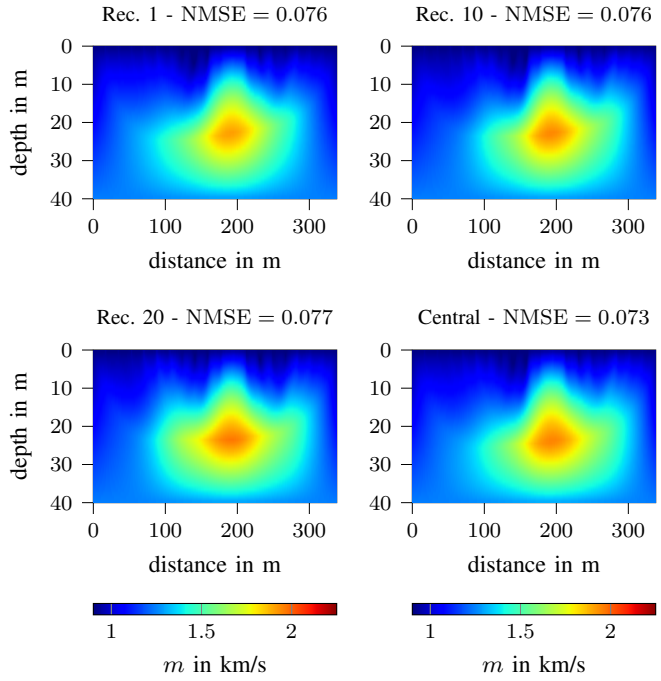


Fig. 7: Imaging results of D-TOMO at receivers $1, 10, 20$ in the line array and image of centralized tomography as reference after $N_{\mathrm{TOMO}} = 20$ iterations.

the residuals are estimated with high accuracy for receivers lying in the middle of the array. For residuals at the borders of the array the approximation degrades. This is due to a decreasing network connectivity where residuals are not taken into account in the regression process with the required significance. After 20 iterations the network achieves consensus over the estimated residuals in contrast to 5 iterations. Fig. 5 shows the cost and normalized model misfit of both methods over the iteration $k$. For both curves the performance of D-TOMO lies very close to that of TOMO. The results demonstrate that D-TOMO successfully achieves images close to the centralized result although time residuals are not estimated perfectly.

### B. EAGE/SEG salt model

As another example we investigate the SEG/EAGE salt model which serves as benchmark model in seismic imaging. We slice out a 2D section of the 3D model and adapt it to distances over meters to obtain a near-surface model, see Fig. 6a. For the imaging we use the same setup as before. However, we change the initial value of the step size to $\mu = 200$ and set $\nu = 10^4$. As starting model we again assume a vertical velocity gradient resembling the background of the true model, see Fig. 6b. The imaging results can be seen in Fig. 7. Since TT relies on first-arrival traveltimes we cannot expect highly resolved images. Therefore, the images appear rather blurry. Nevertheless, we observe that D-TOMO obtains images close to the centralized result, in particular for the middle receiver $r = 10$. At the first and last receiver, again deviations can be observed due to low graph connectivity.

## V. Conclusion

We propose a distributed traveltime tomography to obtain initial $P$-wave velocity models of a subsurface at each receiver in a seismic network. We demonstrate that distributed imaging results of a synthetic model and the EAGE/SEG salt model by our method are close to the centralized result. Therefore, our proposed method can be used to obtain a $P$-wave velocity model at each receiver that serves as starting model for a distributed full waveform inversion such as the ATC-FWI [8]. However, since our method relies on traditional TT, the same imaging limitations of TT apply to our method and images are rather limited in spatial resolution.

## References

[1] B. Knapmeyer-Endrun, M. P. Panning, F. Bissig *et al.*, "Thickness and structure of the martian crust from insight seismic data," *Science*, vol. 373, no. 6553, pp. 438–443, 2021.
[2] S. W. Courville, N. E. Putzig, P. C. Sava, M. R. Perry, and D. Mikesell, "ARES: An autonomous roving exploration system for planetary active-source seismic data acquisition," in *AGU Fall Meeting*, Dec. 2018.
[3] D. J. White, "Two-Dimensional Seismic Refraction Tomography," *Geophysical Journal International*, vol. 97, no. 2, pp. 223–245, 1989.
[4] A. Fichtner, *Full Seismic Waveform Modelling and Inversion*. Springer, Berlin, Heidelberg, 2009.
[5] N. Alaei, M. Soleimani Monfared, A. Roshandel Kahoo, and T. Bohlen, "Seismic imaging of complex velocity structures by 2D pseudo-viscoelastic time-domain full-waveform inversion," *Applied Sciences*, vol. 12, no. 15, 2022.
[6] W. Song, F. Li, M. Valero, and L. Zhao, "Toward creating a subsurface camera," *MDPI Sensors*, vol. 19, no. 2, pp. 1–20, 2019.
[7] S. Wang, F. Li, M. Panning *et al.*, "Ambient Noise Tomography with Common Receiver Clusters in Distributed Sensor Networks," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 6, pp. 656–666, 2020.
[8] B. S. Shin and D. Shutin, "Adapt-then-combine full waveform inversion for distributed subsurface imaging in seismic networks," in *IEEE ICASSP*, Jun. 2021, pp. 4700–4704.
[9] A. H. Sayed, S. Tu, J. Chen *et al.*, "Diffusion strategies for adaptation and learning over networks," *IEEE Signal Process. Mag.*, 2013.
[10] C. Taillandier, M. Noble, H. Chauris, and H. Calandra, "First-arrival traveltime tomography based on the adjoint-state method," *Geophysics*, vol. 74, no. 6, 2009.
[11] E. Treister and E. Haber, "A fast marching algorithm for the factored eikonal equation," *J. Comput. Phys.*, vol. 324, pp. 210–225, 2016.
[12] Y.-H. R. Tsai, L.-T. Cheng, S. Osher, and H.-K. Zhao, "Fast sweeping algorithms for a class of hamilton–jacobi equations," *SIAM Journal on Numerical Analysis*, vol. 41, no. 2, 2003.
[13] S. Leung and J. Qian, "An adjoint state method for three dimensional refraction tomography," *Commun Math Sci*, vol. 4, no. 1, 2006.
[14] R. E. Plessix, "A review of the adjoint-state method for computing the gradient of a functional with geophysical applications," *Geophysical Journal International*, vol. 167, no. 2, pp. 495–503, 2006.
[15] B.-S. Shin, H. Paul, and A. Dekorsy, "Distributed kernel least squares for nonlinear regression applied to sensor networks," in *EUSIPCO*, 2016.
[16] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, vol. 36, no. 3, Jun. 2008.