Technical University of Denmark

# SIPPI: A Matlab toolbox for sampling the solution to inverse problems with complex prior information

Part 2—Application to crosshole GPR tomography

**Hansen, Thomas Mejer; Cordua, Knud Skou; Looms, Majken Caroline; Mosegaard, Klaus**

*Published in:*
Computers & Geosciences

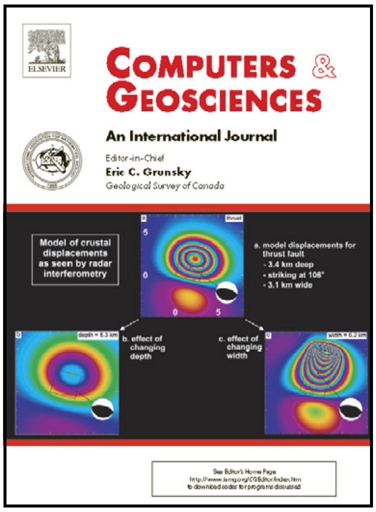
*Link to article, DOI:*
10.1016/j.cageo.2012.10.001

*Publication date:*
2013



*Citation (APA):*
Hansen, T. M., Cordua, K. S., Looms, M. C., & Mosegaard, K. (2013). SIPPI: A Matlab toolbox for sampling the solution to inverse problems with complex prior information: Part 2—Application to crosshole GPR tomography. Computers & Geosciences, 52, 481-492. DOI: 10.1016/j.cageo.2012.10.001

Author’s Accepted Manuscript




www.elsevier.com/locate/cageo

SIPPI: A Matlab toolbox for sampling the solution to inverse problems with complex prior information: Part 2 - Application to cross hole GPR tomography

Thomas Mejer Hansen, Knud Skou Cordua, Majken Caroline Looms, Klaus Mosegaard

PII: S0098-3004(12)00340-8
DOI: http://dx.doi.org/10.1016/j.cageo.2012.10.001
Reference: CAGEO3045

To appear in: *Computers & Geosciences*

Received date: 22 June 2012
Revised date: 7 September 2012
Accepted date: 10 September 2012

Cite this article as: Thomas Mejer Hansen, Knud Skou Cordua, Majken Caroline Looms and Klaus Mosegaard, SIPPI: A Matlab toolbox for sampling the solution to inverse problems with complex prior information: Part 2 - Application to cross hole GPR tomography, *Computers & Geosciences,* http://dx.doi.org/10.1016/j.cageo.2012.10.001

# SIPPI : A Matlab toolbox for sampling the solution to inverse problems with complex prior information: Part 2 - Application to cross hole GPR tomography

Thomas Mejer Hansen[a,*], Knud Skou Cordua[a], Majken Caroline Looms[b], Klaus Mosegaard[a]

[a] *Technical University of Denmark, Center for Energy Resources Engineering, DTU Informatics, Asmussens Alle, Building 305, DK-2800 Lyngby, Denmark*
[b] *University of Copenhagen, Department of Geography og Geology, Øster Voldgade 10, DK-1350 København K, Denmark*



**Abstract**

We present an application of the SIPPI Matlab toolbox, to obtain a sample from the a posteriori probability density function for the classical tomographic inversion problem. We consider a number of different forward models, linear and non-linear, such as ray based forward models that rely on the high frequency approximation of the wave-equation and 'fat' ray based forward models relying on finite frequency theory. In order to sample the a posteriori probability density function we make use of both least squares based inversion, for linear Gaussian inverse problems, and the extended Metropolis sampler, for non-linear non-Gaussian inverse problems. To illustrate the applicability of the SIPPI toolbox to a tomographic field data set we use a cross-borehole traveltime data set from Arrenæs, Denmark. Both the computer code and the data is released in the public domain using open source and open data licenses. The code has been developed to fascilitate inversion of 2D and 3D travel time tomographic data using a wide range of possible a priori models and choices of forward models.

*Keywords:* inversion, nonlinear, tomography, sampling, a priori, a

*Corresponding author. Tel.:+45 45253086, Fax.: +45 45882673

*Email addresses:* tmeha@imm.dtu.dk (Thomas Mejer Hansen), kcor@imm.dtu.dk (Knud Skou Cordua), mcl@geol.ku.dk (Majken Caroline Looms), kmos@imm.dtu.dk (Klaus Mosegaard)

posteriori

## 1. Introduction

Tomographic inversion is used in many research fields such as geophysics and medical imaging. With this technique, images of an unknown 3D object can be obtained based on indirect observations from outside of the object. One such example is travel time inversion, that can for example be used to map the internal velocity structure of the earth, based on recordings of the arrival times of certain seismic phases generated as part of e.g. an earthquake. Another example of a tomographic data set, is that obtained by measuring the travel time delay of a seismic or electromagnetic wave travelling between a source and a receiver. Given such a set of observed travel time data the tomographic inverse problem consists of inferring information about the velocity around and in-between the sources and receivers. It is this latter problem that we will address here using the SIPPI toolbox, which is a Matlab toolbox for sampling the solution to inverse problems with complex a priori information, Hansen et al. (this issue).

We will specifically address the problem of first arrival travel time inversion using crosshole ground-penetrating radar (GPR) data. Such travel time data are sensitive to the subsurface variations in electromagnetic wave velocity, that is related to the dielectric permittivity, which is strongly influences by water moisture, Topp et al. (1980). Inversion of such travel time data thus has the potential to map subsurface moisture content.

For linear or weakly non-linear inverse problems least squares based methods are widely applied. Deterministic least squares methods is presented by e.g. Menke (1989), while a probabilistic approach is given by e.g. Tarantola and Valette (1982) and Tarantola (2005).

A probabilistic approach to linear travel time tomography, based on sequential simulation, was proposed by Hansen et al. (2006) and Hansen and Mosegaard (2008) who utilized the equivalence of classical least squares inversion (e.g. Tarantola and Valette, 1982) and kriging (e.g. Journel and Huijbregts, 1978). An application of this approach to crosshole georadar data is given in Nielsen et al. (2010). A related method based on kriging through error simulation (Journel and Huijbregts, 1978), equivalent with the probabilistic least squares approach, was proposed and applied to cross hole GPR tomographys by Gloaguen et al. (2005a,b). Recently this approach was applied for inversion of an anisotropic velocity field, Giroux and Gloaguen (2012). These methods are only strictly valid for linear inverse problems, and rely on an inherent assumption of Gaussian statistics describing both the noise model and the a priori model. Specifically the a priori model must be given in form of a Gaussian a priori model defined by a mean and a covariance model. Choosing such a Gaussian prior model may not be trivial. A number of methods have been developed to estimate this model prior to inverting the data (Asli et al. (2000); Hansen et al. (2008a); Irving et al. (2009); Looms et al. (2010)).

For examples of least squares based deterministic tomographic inversion of GPR cross hole data see e.g. Irving et al. (2007) and Dafflon et al. (2011). Examples of stochastic inversion is presented for inversion of time lapse cross hole 1D travel time data by Scholer et al. (2012) and 2D time lapse electrical resistivity data by Irving and Singha (2010). Hansen et al. (2008b) demonstrate an application of the extended Metropolis sampler (Mosegaard and Tarantola, 1995) to a nonlinear cross hole tomographic problem, where the a priori model is non-Gaussian and defined by any geostatistical method.

Here we will demonstrate the use of the SIPPI Matlab toolbox for solving the crosshole traveltime tomography inverse problem in a probabilistic framework. Initially we will briefly describe the theory describing different linear and non-linear solutions to the forward problem of computing the travel time delay between a propagating wave traveling between a source and a receiver. Then we will demonstrate how these forward models can be utilized with SIPPI. We will then make use of a reference data set obtained at Arrennæs, North Sealand, Denmark, to demonstrate all the inversion methods available in SIPPI, such as classical least squares estimation and simulation, and sampling methods such as the rejection sampler and the extended Metropolis sampler, see Hansen et al. (this issue).

## 2. Theory, first arrival travel time computation

The travel time delay of a propagating wave between a source and a receiver can be defined in a number of ways. We will consider methods based on the eikonal equation, 1st order sensitivity kernels and the Born approximation.

### *2.1. The eikonal equation*

The eikonal equation describes the arrival time along a closed curve, $u(\mathbf{x})$, travelling with the speed defined by the velocity field, $m(\mathbf{x})$ (Sethian and Popovici, 1999)

$$\mid \nabla u(\mathbf{x}) \mid m(\mathbf{x}) = 1 \tag{1}$$

Solving Eq. 1 allows locating the travel time, $d$, between a source and a receiver along the closed curve. To solve the eikonal equation we make use of an efficient implementation of the multistencil fast marching method proposed by Hassouna and Farag (2007), and made available by Dirk-Jan Kroon[1] under an open source license. This forward model is non-linear and, as the eikonal equation corresponds to a high frequency approximation to the wave equation. Therefore it is often referred to as the high frequency ray approximation.

### *2.2. Forward models based on 1st order sensitivity kernels*

The travel time $d$ between a source and a receiver can be given by

$$d = \int G(\mathbf{x}) \frac{1}{m(\mathbf{x})} d\mathbf{x} \tag{2}$$

where $m(\mathbf{x})$ is the velocity field in which the signal travels. $G(\mathbf{x})$ is the sensitivity kernel that describes the sensitivity of each model parameter (within the Fresnell zone) to the travel time. $G(\mathbf{x})$ can be computed under a wide range of assumptions and thus defines the forward problem of computing the travel time delays in different ways.

#### *2.2.1. Ray based forward model*

Using the high frequency approximation to the wave equation results in a sensitivity kernel $G(\mathbf{x})$ that can be described by a ray connecting the source and receiver. Hence, this kernel can be obtained by solving the eikonal

[1] http://www.mathworks.com/matlabcentral/fileexchange/24531-accurate-fast-marching

equation, which provides the fastest possible forward model. We will refer to this type of forward model as ray based.

*2.2.2. Fat ray based forward model*

Using a finite frequency (band limited) approximation to the wave equation leads to a sensitivity kernel where the sensitivity of the travel time delay also appears in a zone around the fastest ray path. A number of works have defined sensitivity kernels based on geometrical rules assigning sensitivity within the first Fresnel zone. Forward models based on these types of kernels will be referred to as fat ray based forwards (Husen and Kissling, 2001; Jensen et al., 2000).

*2.2.3. Born based forward model*

The Born approximation to the wave equation (considering only 1st order scattering) is an exact analytical expression for the sensitivity kernel for a point source, which can be derived for both seismic (Dahlen et al., 2000; Spetzler and Snieder, 2004; Marquering et al., 1999; Liu et al., 2009) and electromagnetic wave propagation (Bursink et al., 2008). The Born approximation also leads to a senstivity kernel with sensitivity outside the ray approxiamtion (i.e. a fat ray). The Born approximation is only strictly valid for a homogeneous velocity field, but have in practice been used also when the velocity field has relatively small velocity contrasts. For large velocity contrast this method becomes unstable and cannot be used. Forward models based on the Bron approximation will be referred to as Born based forward models.

## 3. Cross hole GPR tomography at Arrenæs

As a case study we will demonstrate the capabilities of SIPPI for solving tomographic inverse problems. The implementation is generally applicable for travel time based tomographic problems, but here we will apply the toolbox to a cross hole GPR tomographic problem.

Initially we will present a 3D data set. Then we will demonstrate how the the different types of forward models have been implemented in `sippi_forward_traveltime` for easy utilization as part of SIPPI. Finally we demonstrate the use of SIPPI to solve the GPR cross hole tomography inverse problem using both linear and non-linear forward models, and simple and more complex a priori models.

### *3.1. Data : 3D GPR Crosshole traveltime data from Arrenæs*

As a reference data set we consider a 3D tomographic data set recorded as part of a ground penetrating radar (GPR) cross borehole survey at Arrenæs, North Sealand, Denmark. The data set we use here is identical to data presented by Looms et al. (2010), and is here made available in the public domain.

The observed data are first arrival times of electromagnetic waves propagating from a source location in one borehole to a receiver location in another borehole. Thus, the forward problem consists of estimating the travel time delay caused by the subsurface velocity field, given the recording geometry. The inverse problem is then to infer information about the subsurface velocity structure.

The subsurface at Arrenæs consists mostly of sand, with various degree of coarseness. The velocity of the subsurface is believed to represent natural moisture content. The lower the velocity the higher the moisture content, Topp et al. (1980).

Figure 1 shows the relative position of four boreholes, AM1, AM2, AM3, and AM4. Tomographic travel time delay have been recorded between boreholes AM1-AM3 and AM2-AM4, respectively. The locations of the source and receiver positions down through the boreholes are shown in Figure 1 and is marked by red dots in two of the boreholes. Note that the coloured ray like structure on Figure 1 reflect the high frequency ray kernel ralated to a constant velocity model. The colours of each ray reflect the average velocity along each of the rays, and can be used as a rough indicator of the subsurface velocity structure.

[Figure 1 about here.]

Data are available as ASCII and binary Matlab formatted files for both the two 2D data sets, `AM13_data` and `AM24_data`, and the combined 3D data set, `AM1234_data` that combines the data sets `AM13_data` and `AM24_data` .

The Matlab mat files contain the location of the sources and receivers in the `S` and `R` variables. Observed data is in the `d_obs` variable and the associated uncertainty (in form of the standard deviations) is in the `d_std` variable. A covariance model describing static like errors related to cross borehole GPR data, as given by Cordua et al. (2009), is available in the `Ct` variable.

### 3.2. The forward model - traveltime computation

As described in Hansen et al. (this issue), the only problem dependent part of using SIPPI is the implementation of a solution to the forward problem. We have implemented the m-file `sippi_forward_traveltime` that can be used to solve the forward problem of computing the travel time delay between a set of sources and receivers. All properties relating to solving the forward problem is defined in the `forward` Matlab structure. The output is the data structure `d`:

```
[d]=sippi_forward_traveltime(m,forward,prior,data);
```

To make this solution of the forward problem available for the various inversion algorithms available in SIPPI, one can either implement an m-file `sippi_forward` that simply calls `sippi_forward_traveltime`, or one can specify the m-file to be used for solving the forward problem directly using `forward.forward_function='sippi_forward_traveltime'`. Note that this m-file and the specification of the `forward` structure is specific to the tomographic travel time inverse problem, while all other parts of the SIPPI toolbox are applicable to inverse problems in general.

*Source and receiver geometry.* The locations of the sources and receivers must be provided in the `forward.sources` and `forward.receivers` fields. Both the `sources` and `receivers` must point to a matrix with a number of rows equal to the number of rows (i.e. number of data) of `data{id}.d_obs`, and a number of columns reflecting the dimension of the prior model. For example, two sets of sources and receivers defined in 3D could be given by

```
forward.sources=[1 1 5 ; 1 1 10];
forward.receivers=[5 5 5 ; 5 5 10];
```

*Forward model.* Four types of forward models are available through `sippi_forward_traveltime` by specifying the `forward.type` field to one of `eikonal`, `ray`, `fat`, or `born`.

`forward.type='eikonal'` defines a forward model based on the solution to the eikonal equation, Eq. 1. This forward model is non-linear.

The other three available forward model types, `ray`, `fat`, and `born`, refer the the ray, fat and Born based sensitivity kernels presented earlier. When `sippi_forward_traveltime` is called using any of these types of forward models, a matrix operator, reflecting the choice of forward model, is computed as `forward.G`.

One can choose either a linear or non-linear formulation for solving such forward problems by specifying the `forward.linear` field. By default a non-linear formulation is assumed, such that `forward.linear=0`. This cause `forward.G` to be recalculated for each call to `sippi_forward_traveltime`. Different velocity models will result in different sensitivity kernels, and hence different forward operators, `forward.G`. Therefore the forward problem is non-linear.

One can also choose a linear formulation, using `forward.linear=1`. In this case `forward.G` is only computed once, when `sippi_forward_traveltime` is called for the first time, and hence any subsequent calls to solve the forward model requires only a fast matrix multiplication. One can provide a velocity model for which the sensitivity kernel will be computed using `forward.linear_m`. If this is not specified the sensitivity kernel will be computed for the a priori mean model, given in `prior{1}.m0`.

`forward.type='ray'` selects the high frequency ray approximation presented earlier. This type of forward model is based on the same high frequency assumption as the `eikonal` type forward model. The difference is that here the forward operator `forward.G` is explicitly computed, which allows for a very fast forward model using `forward.linear=0`. If one would consider using the `ray` type forward model in a non-linear formulation, we suggest to use the `eikonal` type of forward model instead, which provides similar results but is computationally much more efficient. Used in the linear formulation this type forward model resemble the 'straight ray' approximation, as the the travel delay is due to the travel time delay along straight ray path that connects the source and receivers. The 'rays' on Figure 1 reflect such a linear 'ray' type forward model.

`forward.type='fat'` selects a finite frequency (band limited) approximation to the wave equation, where the travel time delay i sensitive to a zone around the fastest ray path. Specifically the `fat` type forward model uses the empirical description of the travel time sensitivity kernel as proposed by Jensen et al. (2000), which is based on 1st order Fresnel zone sensitivity. The `fat` type forward model can be used both as linear and non-linear forward model.

`forward.type='born'` selects a forward model based on the Born approximation as presented earlier. Here we will make explicit use of the formulation of the sensitivity kernels given by Buursink et al. (2008). The `born` type forward model is only strictly valid for a homogeneous velocity field, but have in practice been used also when the velocity field has relatively small velocity contrasts. For large velocity contrasts this method becomes unstable and should not be used.

Using either `forward.type='fat'` or `forward.type='born'` the width of the sensitivity around the ray path, is related to the frequency of the propagating wave. Therefore this frequency must be set as `forward.freq`. The frequency must be specified in the inverse unit of the observed travel time data given in `data{id}.d_obs`.

As an example of choosing the `fat` type forward model in a non-linear formulation using a wavelet frequency of 0.1 GHz, where traveltime data is measured in nanoseconds, is

```
forward.type='fat';
forward.freq=0.1;
forward.linear=0;
```

### *3.3. Solving the inverse problem*

Having defined the forward problem, we will demonstrate the methods available in SIPPI for solving the inverse tomographic problem.

#### *3.3.1. 2D non-linear inversion - AM13*

Initially we will consider the 2D traveltime data set, AM13, recorded between well AM1 and AM3, using a simple Gaussian type a priori model. 702 travel time data and the position of associated source and receiver locations is available in the Matlab file `AM13_data.mat`. To use SIPPI, the `forward`, `data`, and `prior` structures need to be defined.

*Setting up the `forward` structure.* We use the high frequency ray approximation, in form of the `eikonal` type forward model, such that the `forward` data structure can be setup using

```
D=load('AM13_data.mat');
forward.sources=D.S;
forward.receivers=D.R;
forward.type='eikonal';
```

*Setting up the* ***data*** *structure.* The high frequency approximation, assumed by using the eikonal solution, will always provide the fastest travel time between a source and a receiver, and always faster than the travel time of a wave with a finite finite frequency in a inhomogeneous velocity field. Therefore we allow for a small modelization error, `Ct`, in form of a constant correlated Gaussian error of 1 $ns^2$ between all data. This will allow a small bias correction (the same for all data observations) to account for the relative high travel times caused by the use of the high frequency forward model. The data in form of 702 observed traveltimes, `d_obs`, and associated uncorrelated uncertainties, `d_std` (of 0.7 ns), is available in the Matlab file `AM13_data.mat`. The `data` structure can be setup as

```
D=load('AM13_data.mat');
id=1;
data{id}.d_obs=D.d_obs;
data{id}.d_std=D.d_std;
data{id}.Ct=1; % modelization error
```

SIPPI allows using only a subset of the available data, which can be useful to test a certain setup relatively fast. The number of data considered is given by `data{id}.i_use`. To use every 20th data one could use `data{id}.i_use=20:20:702`. If not set it is automatically set to all data. In the current case this will be `data{id}.i_use=1:1:702`.

*Setting up the **prior** structure.* Looms et al. (2010) demonstrate a method for inferring the structural parameters of a Gaussian type a priori model. They tested their method on the data we use here and find an optimal a priori model for profile AM13 and AM24 independently. Initially we will make use of the same a priori model for both profile AM13 and AM24 and, therefore, based on the findings in Looms et al. (2010), we choose to use a Gaussian type a priori model as defined by a Spherical type covariance model with an isotropic covariance model with a range of 6m, a variance of 0.0003 $m^2/ns^2$, and a mean of 0.145 $m/ns$. We make use of the `FFTMA` type a priori model. The complete definition of the a priori model can then be given as

```
im=1;
prior{im}.type='FFTMA';
prior{im}.name='Velocity (m/ns)';
prior{im}.m0=0.145;
prior{im}.Va='.0003 Sph(6)';
prior{im}.x=[-1:.2:6];
prior{im}.y=[0:.2:13];
```

A sample of the corresponding a priori model can then be generated and visualized using `sippi_plot_prior_sample(prior)` as shown in Figure 2a.

[Figure 2 about here.]

*Sampling the a posteriori pdf using the extended Metropolis algorithm.* Given the `forward`, `prior`, and `data` structures the extended Metropolis algorithm can be setup and run using e.g.

```
options.mcmc.nite=500000;
options.mcmc.i_plot=200;
options.mcmc.i_sample=250;
sippi_metropolis(data,prior,forward,options);
```

This will cause the extended Metropolis sampler to run for 500000 iterations. The currently visited model will be saved to disk for every 250 iterations as specified by `options.mcmc.i_sample`

As the Metropolis algorithm is running, some properties are visualized for every `options.mcmc.i_plot` iterations, such as the currently accepted model, the step length for each prior type, and the log-likelihood curve. Such figures are often useful in the phase where the properties of the Metropolis algorithm are selected, prior to performing a full sampling.

Figure 3 shows the log-likelihood value as function of the iteration number. The Metropolis algorithm has reached burn-in after about 2000 iterations as it reaches the plateau of log-likelihood values of approximately -90.

[Figure 3 about here.]

Recall that the way the sequential Gibbs sampler works, is controlled by the `prior{1}.seq_gibbs` structure, Hansen et al (this issue). Here we make use of the default settings

```
prior{1}.seq_gibbs.i_update_step=50
prior{1}.seq_gibbs.i_update_step_max=1000
prior{1}.seq_gibbs.P_target=0.3000
```

This means that the step length of the Metropolis sampler is adjusted for every 50 iterations with the goal of achieving an acceptance rate of 0.3. After 1000 iterations the step length will be kept constant.

Figure 4 shows the step length of the sequential Gibbs sampler as well as the acceptance rate in the first 3000 iterations. In the first 1000 iterations the step length is allowed to vary, and after 1000 iterations the step length stabilize around $10^{-3}$. Initially the acceptance rate is about 0.2. Then it decreases rapidly until the step length is gradually adjusted, such that the acceptance rate ends up around 0.3, just as requested by `prior{1}.seq_gibbs.P_target`. Recall that while the step length is being changed, and until the Metropolis algorithm has reached burn-in, the a posteriori pdf is not sampled, Cordua et al. (2012).

[Figure 4 about here.]

Figure 2b shows 5 independent realizations of the a posteriori pdf, obtained after the Metropolis algorithm has reached burn-in. Comparing the realizations of the a posteriori pdf to the realization of the a priori pdf, Figure 2a, reveals that the apparent scales and spatial structures visible in the a priori realizations are also present in the a posteriori realizations. The location of these structures is not resolved in the a prior realizations. But in the a posteriori realizations it is clear that relative high velocity structures dominate in the lower right corner while areas of lower velocity dominate the upper part of the model. Features such as these, that appear on many realizations of the a posteriori pdf are well resolved features, Mosegaard (1998).

Once the extended Metropolis sampler has finished a number of plots for quality control can be generated using `sippi_plot_posterior`. First a figure visualize a sample of the a posteriori pdf, as in Figure 2a. Second, a figure shows the acceptance ratio and step length as a function of iteration number, as in Figure 4. Third, a figure shows the distribution of data residuals, i.e. the difference between observed and simulated travel time data, corresponding to number a realizations of the a posteriori pdf, as in Figure 5. Note how the distribution is very close to Gaussian, as defined in the noise model. Note also how the distribution is not entered around 0 ns, but has a mean value (i.e. a bias) of about -1.5 ns. This is due to allowing a constant modelization error of 1 $\text{ns}^2$, that was applied in order to account for the use of the eikonal type forward model, that will always provide the fastest possible travel time between a source and a receiver. This is correctly reflected in the negative bias correction.

[Figure 5 about here.]

Finally `sippi_plot_posterior` provides a figure that illustrates the correlation coefficient of the currently accepted model in the last iteration to any of the other models sampled from the a posteriori pdf. This is used to estimate the number of iterations between independent realizations of the a posteriori pdf, e.g. Cordua et al. (2012). An example generated for the present example, is shown in Figure 6. The correlation coefficient between the current model at iteration 500000 and the models close to iteration number 500000 is close to 1, and such models are not statistically independent. However, in a number of iterations away from the last considered model, the correlation coefficient decreases, until it reached a level of around 0.7. We use this level of the correlation coefficient to determine the approximate number of iterations between independent realizations of the a posteriori pdf obtained by the Metropolis algorithm. For the present case this was estimated to be around 10000 iterations between independent realizations.

[Figure 6 about here.]

*Sampling the a posteriori pdf using the rejection sampler.* Sampling the a posteriori pdf for the tomographic inverse problem using rejection sampling, can in principle be performed using

```
options.mcmc.nite=500000;
sippi_rejection(data,prior,forward,options);
```

The maximum a posteriori likelihood $L_{max}$ is set to 1, if not, as here, specifically set using `options.mcmc.Lmax`, see Hansen et al. (this issue). Figure 7 (green bars) shows a histogram of the likelihood of all the a posteriori accepted models using the extended Metropolis algorithm as considered above. The log-likelihood distribution of a posteriori accepted models is in the interval -105 to -75. However, the blue line indicates the maximum log-likelihood of -824 obtained after generating 500000 independent realizations of the a priori pdf and evaluating the corresponding log-likelihood as part of running the rejection sampler. Thus, the 'best' model found after 500000 realizations is very far from leading to a data fit within data uncertainties. Even if $L_{max}$ could somehow be chosen around -68 (as indicated by the log-likelihood values of the accepted a posteriori models obtained from Metropolis sampling) the probability of locating just one realization from the a posteriori pdf using independent sampling of the a priori pdf, will be extremely low. The main problem with the rejection sampler is that it is computationally very inefficient for anything but very low dimensional problems. In general we suggest to make use of the extended Metropolis sampler to sample the a posteriori pdf of non-linear non-Gaussian inverse problems.

[Figure 7 about here.]

*Sampling the a posteriori pdf using least-squares.* As discussed in Hansen et al. (this issue), if the forward problem is linear, and a linear forward mapping operator given as `forward.G` is provided, then the a posteriori pdf can be sampled using least squares, kriging through error simulation or direct sequential simulation. Here we will consider using classical least squares type inversion, using `lsq_type='lsq'`. We will use exactly the same specification of the a priori model and the data model as used above.

To solve the linear Gaussian inverse problem using least squares type inversion, using the `ray`,`fat`, and `born` type forward model approximation, we use

```
forward.linear=1;
forward.type='ray';
forward.freq=10;
lsq_type='lsq';
nr=15;

% 'ray' type forward model
forward.type='ray';
[m_reals_ray,m_est_ray,Cm_est_ray] =
  sippi_least_squares(data,prior,forward,nr,lsq_type);

% 'fat' type forward model
forward.type='fat';
forward.freq=10;
[m_reals_fat,m_est_fat,Cm_est_fat] =
```

```
  sippi_least_squares(data,prior,forward,nr,lsq_type);

% 'born' type forward model
forward.type='born';
[m_reals_born,m_est_born,Cm_est_born] =
  sippi_least_squares(data,prior,forward,nr,lsq_type);
```

It is difficult to see any large difference between realizations from the a posteriori pdf using the three different types of forward models. Therefore Figure 8 shows the three a posteriori mean models, considering the a) `ray`, b) `fat`, c) and `born` type forward model, which demonstrates that on average there is a difference between the solutions obtain with these different forward choices.

[Figure 8 about here.]

*3.3.2. 2D non-linear inversion - AM24*

We now consider the 2D data recorded between borehole AM2 and AM4, perpendicular to the data set recorded between borehole AM1 and AM3. We make the same assumptions about the a priori and the forward model as considered in the application of the extended Metropolis sampler above

```
D=load('AM24_data.mat');
forward.sources=D.S;
forward.receivers=D.R;
forward.type='eikonal';
```

As above we make use of the extended Metropolis algorithm to sample the a posteriori pdf. Figure 9 shows 20 realizations of the 1D velocity from the a posteriori pdf considering the data sets AM13 and AM24, at location x=2.5m, where the two profiles cross each other. Also shown is the mean of 200 a posterior realization for both data sets.

[Figure 9 about here.]

Figure 9 reveals that where the two profiles intersect, the inferred velocity profile is quite similar even when the two data sets are inverted independently. In the top part of the model, where the consistency between realizations are weakest, the relative position of the relatively high velocity layers at depths of 2.8m and 5m is in agreement, while the velocity estimates of the more shallow parts differ only slightly. The reason for the observed inconsistencies can be related to the use of a 2D forward model describing data collected in a 3D world.

*3.3.3. 3D inversion using a Gaussian a priori model - AM1234*

Setting up an inversion using 3D data and a 3D parametrization of the a priori model is very similar to the 2D example above. Using the AM1234 data sets one can use

```
D=load('AM1234_data.mat');
forward.sources=D.S;
forward.receivers=D.R;
forward.type='eikonal';
```

The a priori model is identical to the one used above, except that a 3D parametrization needs to be specified. We also make use of a larger pixel size in order to keep the running time reasonable.

```
prior{im}.x=[-1:.5:6];
prior{im}.y=[-1:.5:6];
prior{im}.z=[0:.5:13];
```

Sampling the a priori and a posteriori pdf, can be performed in exactly the same manner as done for the 2D cases above. Figure 10 shows 5 independent realizations of the a posteriori pdf, obtained after the Metropolis algorithm has reached burn-in. Figure 11 compare the mean of an a posteriori sample obtained from inverting the AM13, AM24, AM1234 data sets, at the location where the two 2D profiles intersect. Also shown is realizations from the a posteriori pdf corresponding to the AM1234 data set. Above 8m depth the a posteriori mean is very similar for all cases. Below 8m depth, the inferred velocity is higher inverting the 3D data set compared to the 2D data set.

[Figure 10 about here.]

[Figure 11 about here.]

*3.3.4. 2D inversion with unknown covariance model properties*

Most all inversion methods relying on a Gaussian a priori model, require that the properties of the covariance model, such as the mean, range, anisotropy, and variance are known prior to inversion. The choice of a priori covariance model highly affect the inversion result and, therefore, some work has been done to estimate a (prior) covariance model consistent with observed data, Asli et al. (2000); Hansen et al. (2008a); Looms et al. (2010). As mentioned in Hansen et al. (this issue) the FFTMA method allows for separating such structural properties of the covariance model from the random component. SIPPI allows such properties to act as model parameters, that can be inferred as part of an inversion.

To demonstrate this we use the same data and setup as used previously from the 2D travel time data set obtained between borehole AM1 and AM3, i.e. data set AM13, but where the a priori model is changed to allow for inference of the horizontal and vertical range.

```
im=0;
% prior - HORIZONTAL RANGE
im=im+1;
prior{im}.type='gaussian';
prior{im}.m0=8;
prior{im}.std=6;
prior{im}.name='range_1';
prior{im}.prior_master=3;
prior{im}.norm=20;

% prior - VERTICAL RANGE
im=im+1;
prior{im}=prior{im-1};
prior{im}.name='range_2';

% prior - 2D VELOCITY FIELD
im=im+1;
prior{im}.type='FFTMA';
prior{im}.name='Velocity (m/ns)';
prior{im}.m0=0.145;
```

```
prior{im}.Va='.0003 Sph(6)';
prior{im}.x=[-1:.2:6];
prior{im}.y=[0:.2:13];
```

Note that the only difference to the first example of inverting the AM31 data set with a known a priori covariance model, is the definition of two a priori parameters, named `range_1` and `range_2`. Also, these two prior structures point to the third prior structure (the `FFTMA` type prior) as their 'master', indicating which prior structure it belongs to. This ensures that when the value of such a prior model is updated, so is the value of covariance model of the corresponding `prior_master` structure.

A sample of this a priori model is shown in Figure 12a. It is apparent that allowing variability in the ranges, determines an a priori model with much more a priori variability as compared to when the ranges is kept constant.

We now make use of the extended Metropolis sampler to sample the a posteriori pdf, in three cases where we use only 35 (every 20th observed data), 140 (every 5th observed data) and all 702 observed data, respectively. The subset of the data is chosen using the `data{id}.i_use=20:20:702` and `data{id}.i_use=5:5:702` respectively. The corresponding samples from the a posteriori pdf is shown in Figures 12b-d.

[Figure 12 about here.]

Because the horizontal and vertical ranges of the a priori covariance is also model parameters, the a posteriori distribution of these model parameters can also be quantified. Figure 13 shows the 1D marginal posterior distribution of the horizontal and vertical range respectively using every a) 20th, b) 5th

, and c) all available observed data. When few observed data are used only very little information can be inferred about the ranges (red lines). But, as the number of data increases, so does the resolution of the range parameters. When all 702 data are used the 1D marginal a posteriori distributions of the ranges reveal that the horizontal range is relative long, between 7m and 15m, while the vertical range is better resolved with values between 4.8m and 7m. These findings are consistent with the result reported by Looms et al. (2010). Looms et al. (2010) find the range estimates priori to inversion of the travel time data, while in the present approach information about the ranges is inferred as part of the inversion.

As the number of considered observed data increase so does the resolution, which is seen as the differences between the a posteriori realizations become smaller. Thus increasing the amount of data leads to a better constrained posterior sample. It is, however, important to notice that the posterior statistics inferred from an a posteriori sample using only a subset of the data is consistent with the full solution: Features that appear well resolved from a sample of the a posteriori pdf obtained using a subset of the data, will be consistent with the full inverse problem, unless some unaccounted for bias is present in data. There might be cases where the resolution provided by subset of the available data will be adequate. This will off course also result in an easier, more computationally efficient, sampling problem.

Traditional applications in inverse problems with Gaussian a priori models, rely on the existence of, or choice of, an a priori covariance model to describe spatial variability. The combination of the FFTMA prior model with the extended Metropolis sampler as implemented in SIPPI opens up new possibilities for solving non-linear inverse problems with unknown properties of the structural covariance model describing spatial variability.

[Figure 13 about here.]

*3.3.5. 2D inversion with training image based prior*

The a priori knowledge about the subsurface at Arrenæs does not readily call for a multiple point based a priori model, nor is such a model readily available. To demonstrate the use of a multiple point based a priori model, we generate a synthetic data set based on an a priori model defined by the training image in Figure 4 in Hansen et al. (this issue), and the SNESIM type a priori model, Strebelle (2002), defined using

```
im=1;
prior{im}.type='SNESIM';
prior{im}.ti='snesim_std.ti';
prior{im}.index_values=[0 1]; % optional
prior{im}.m_values=[.1 0.18]; % optional
prior{im}.scaling=.75; % optional
prior{im}.rotation=30; % optional
```

Figure 14a shows 5 realizations of this a priori model. The first model is chosen as the reference velocity model, from which synthetic data are computed by solving the forward problem. Finally some random Gaussian noise, according to the observed data uncertainties, are added to obtain an 'observed' data set.

```
id=1;
m_ref=sippi_prior(prior);
```

```
d_ref=sippi_forward(m_ref,forward,prior,data);
data{id}.d_obs=d_ref{1}+
 randn(size(d_ref{1})).*data{id}.d_std;
data{id}.Ct=0;
```

Then the Metropolis algorithm is run in the exact same manner as in the previous examples. Figure 14b shows 5 realizations from the a posteriori pdf obtained by running the extended Metropolis algorithm.

This small example demonstrates that the difficulty of using a more complex a priori model using SIPPI, lies mostly in the difficulty to locate or choose such a model. Implementation wise there is only very little difference in choosing a simple covariance based prior model as opposed to a more complex prior model based on multiple point statistics.

[Figure 14 about here.]

## 4. Conclusions

We have demonstrated the use of the SIPPI toolbox to sample the solution to cross hole travel time tomographic inverse problems. A number of different forward models ranging from simple ray theory, based on high frequency wave-theory, to fat ray forward models based on finite frequency theory are available. We have demonstrated how such a tomographic inverse problem can be solved by sampling the a posteriori pdf, for a non-linear formulation of the inverse problem using the extended Metropolis algorithm for both 2D and 3D cases. We have also shown how least squares based techniques can be used to directly generate samples of the a posteriori pdf in the case of linear inverse Gaussian problems. Examples are based on a cross hole georadar data set. We have demonstrated that SIPPI facilitates a novel approach, based on a combination of the FFTMA method and the extended Metropolis sampler, that allow sampling the a posteriori pdf of linear and non-linear inverse problem with a Gaussian a priori model, where the properties of the covariance can be treated as parameters, and thus inferred as part of the inversion. Thus, the structural parameters defining the Gaussian a priori model, need not be known prior to inversion. All code and data is available using open licenses.

**Acknowledgement**

We thank DONG for financial support. SIPPI source code and the data from Arrenæs can be downloaded from http://sippi.sourceforge.net/.

Asli, M., Marcotte, D., Chouteau, M., 2000. Direct inversion of gravity data by cokriging. In: Kleingeld, W., Krige, D. (Eds.), Proceedings of the 6th International Geostatistics Congress, Cape Town, South Africa, April 10-14, 2000. pp. 64–73.

Buursink, M., Johnson, T., Routh, P., Knoll, M., 2008. Crosshole radar velocity tomography with finite-frequency Fresnel volume sensitivities. Geophysical Journal International 172 (1), 1–17.

Cordua, K., Nielsen, L., Looms, M., Hansen, T., Binley, A., 2009. Quantifying the influence of static-like errors in least-squares-based inversion and sequential simulation of cross-borehole ground penetrating radar data. Journal of Applied Geophysics 68 (1), 71–84.

Cordua, K. S., Hansen, T. M., Mosegaard, K., 2012. Monte Carlo full waveform inversion of crosshole GPR data using multiple-point geostatistical a priori information. Geophysics 77, H19–H31.

Dafflon, B., Irving, J., Barrash, W., 2011. Inversion of multiple intersecting high-resolution crosshole gpr profiles for hydrological characterization at the boise hydrogeophysical research site. Journal of Applied Geophysics.

Dahlen, F., Hung, S., Nolet, G., 2000. Fréchet kernels for finite-frequency traveltimes-I. Theory. Geophysical Journal International 141 (1), 157–174.

Giroux, B., Gloaguen, E., 2012. Geostatistical traveltime tomography in elliptically anisotropic media. Geophysical Prospecting.

Gloaguen, E., Marcotte, D., Chouteau, M., 2005a. A non-linear tomographic inversion algorithm based on iterated cokrigriging and conditional simulations. In: Leuangthong, O., Deutsch, C. (Eds.), Geostatistics Banff 2004. Vol. 1. Springer, pp. 409–418.

Gloaguen, E., Marcotte, D., Chouteau, M., Perroud, H., 2005b. Borehole radar velocity inversion using cokriging and cosimulation. Journal of Applied Geophysics 57 (4), 242–259.

Hansen, T., Cordua, K., Looms, M., Mosegaard, K., 201x. SIPPI : A Matlab toolbox for Sampling the solution to Inverse Problems with complex Prior Information: Part 1, Methodology. Computers & Geosciences.

Hansen, T., Looms, M., Nielsen, L., 2008a. Inferring the subsurface structural covariance model using cross-borehole ground penetrating radar tomography. Vadose Zone Journal 7 (1), 249.

Hansen, T. M., Journel, A. G., Tarantola, A., Mosegaard, K., 2006. Linear inverse Gaussian theory and geostatistics. Geophysics 71 (6), R101–R111.

Hansen, T. M., Mosegaard, K., 2008. VISIM: Sequential simulation for linear inverse problems. Computers and Geosciences 34 (1), 53–76.

Hansen, T. M., Mosegaard, K., Cordua, K. C., 2008b. Using geostatistics to describe complex a priori information for inverse problems. In: Ortiz, J. M., Emery, X. (Eds.), VIII International Geostatistics Congress. Vol. 1. Mining Engineering Department, University of Chile, pp. 329–338.

Hassouna, M., Farag, A., 2007. Multistencils fast marching methods: A highly accurate solution to the eikonal equation on cartesian domains. Pattern Analysis and Machine Intelligence, IEEE Transactions on 29 (9), 1563–1574.

Husen, S., Kissling, E., 2001. Local earthquake tomography between rays and waves: fat ray tomography. Physics of the earth and Planetary Interiors 123 (2), 127–147.

Irving, J., Knight, R., Holliger, K., et al., 2009. Estimation of the lateral correlation structure of subsurface water content from surface-based ground-penetrating radar reflection images. Water Resour. Res 45 (12), W12404.

Irving, J., Knoll, M., Knight, R., 2007. Improving crosshole radar velocity tomograms: A new approach to incorporating high-angle traveltime data. Geophysics 72, J31.

Irving, J., Singha, K., 2010. Stochastic inversion of tracer test and electrical geophysical data to estimate hydraulic conductivities. Water Resour. Res 46.

Jensen, J., Jacobsen, B., Christensen-Dalsgaard, J., 2000. Sensitivity kernels for time-distance inversion. Solar Phys.: SOHO9 topical issue 192 (1-2), 231–239.

Journel, A. G., Huijbregts, C. J., 1978. Mining Geostatistics. Academic Press.

Liu, Y., Dong, L., Wang, Y., Zhu, J., Ma, Z., 2009. Sensitivity kernels for seismic Fresnel volume tomography. Geophysics 74, U35–U46.

Looms, M., Hansen, T., Cordua, K., Nielsen, L., Jensen, K., Binley, A., 2010. Geostatistical inference using crosshole ground-penetrating radar. Geophysics 75 (6), J29.

Marquering, H., Dahlen, F., Nolet, G., 1999. Three-dimensional sensitivity kernels for finite-frequency traveltimes: the banana–doughnut paradox. Geophysical Journal International 137 (3), 805–815.

Menke, W., 1989. Geophysical Data Analysis: Discrete Inverse Theory. Vol. 45 of International Geophysics Series. Academic Press, revised Edition.

Mosegaard, K., 1998. Resolution analysis of general inverse problems through inverse Monte Carlo sampling. Inverse problems 14, 405.

Mosegaard, K., Tarantola, A., 1995. Monte Carlo sampling of solutions to inverse problems. Journal of Geophysical Research 100 (B7), 12431–12447.

Nielsen, L., Looms, M., T.M., H., Cordua, K., Stemmerik, L., 2010. Estimation of Chalk Heterogeneity from Stochastic Modeling Conditioned by Crosshole GPR Traveltimes and Log Data. No. 15 in Geophysical Developments Series. Society of Exploration Geophysicists, Ch. 23, pp. 379–396.

Scholer, M., Irving, J., Looms, M., Nielsen, L., Holliger, K., 2012. Bayesian markov-chain-monte-carlo inversion of time-lapse crosshole ground-penetrating radar to characterize the vadose zone at the arrenaes field site, denmark. Vadose Zone Journal xx (xx), xxx–xxx.

Sethian, J., Popovici, A., 1999. 3-d traveltime computation using the fast marching method. Geophysics 64 (2), 516–523.

Spetzler, J., Snieder, R., 2004. The Fresnel volume and transmitted waves. Geophysics 69, 653.

Strebelle, S., 2002. Conditional simulation of complex geological structures using multiple-point statistics. Math. Geol 34 (1), 1–20.

Tarantola, A., 2005. Inverse Problem Theory and Methods for Model Parameter Estimation. SIAM.

Tarantola, A., Valette, B., 1982. Generalized nonlinear inverse problems solved using the least squares criterion. Rev. Geophys. Space Phys 20 (2), 219–232.

Topp, G., Davis, J., Annan, A., et al., 1980. Electromagnetic determination of soil water content: Measurements in coaxial transmission lines. Water Resour. Res 16 (3), 574–582.

## List of Figures

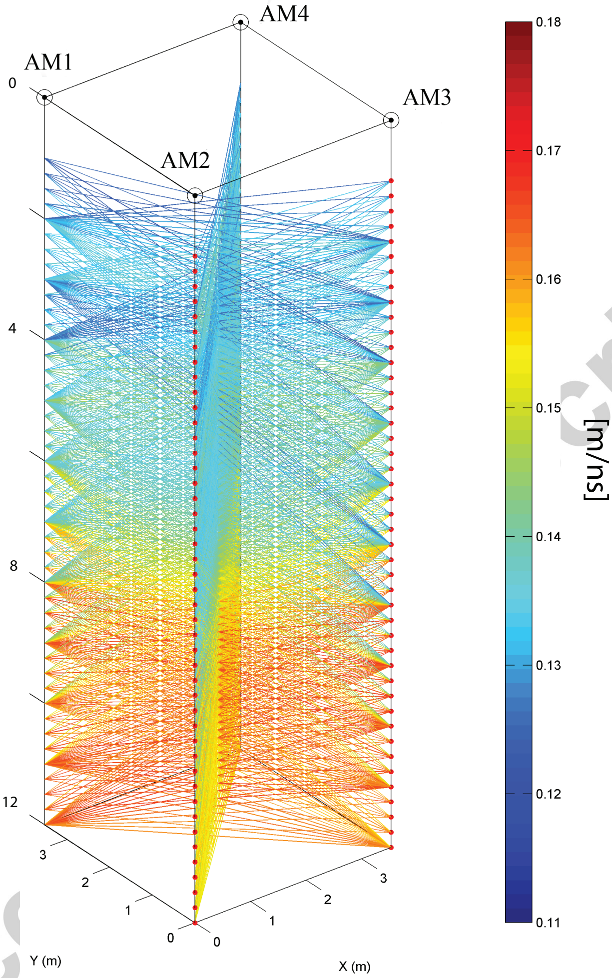



Figure 1: Apparent ray coverage (using the linear high frequency approximation). The color of each ray reflects the apparent average velocity along the ray path.

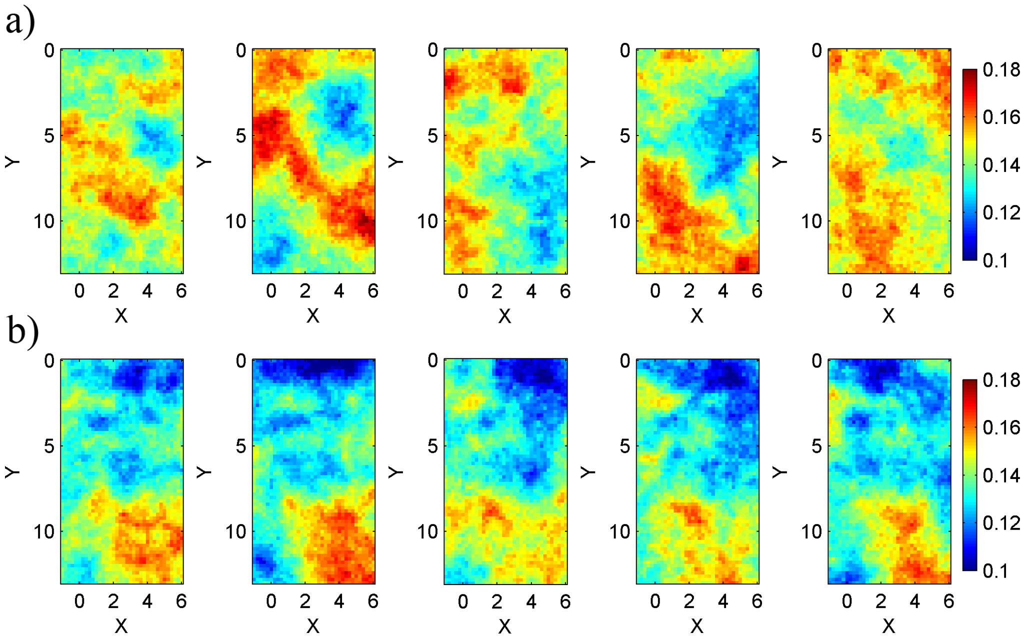



Figure 2: 5 realizations from the a) a priori model, and b) a posteriori pdf considering dataset AM13.

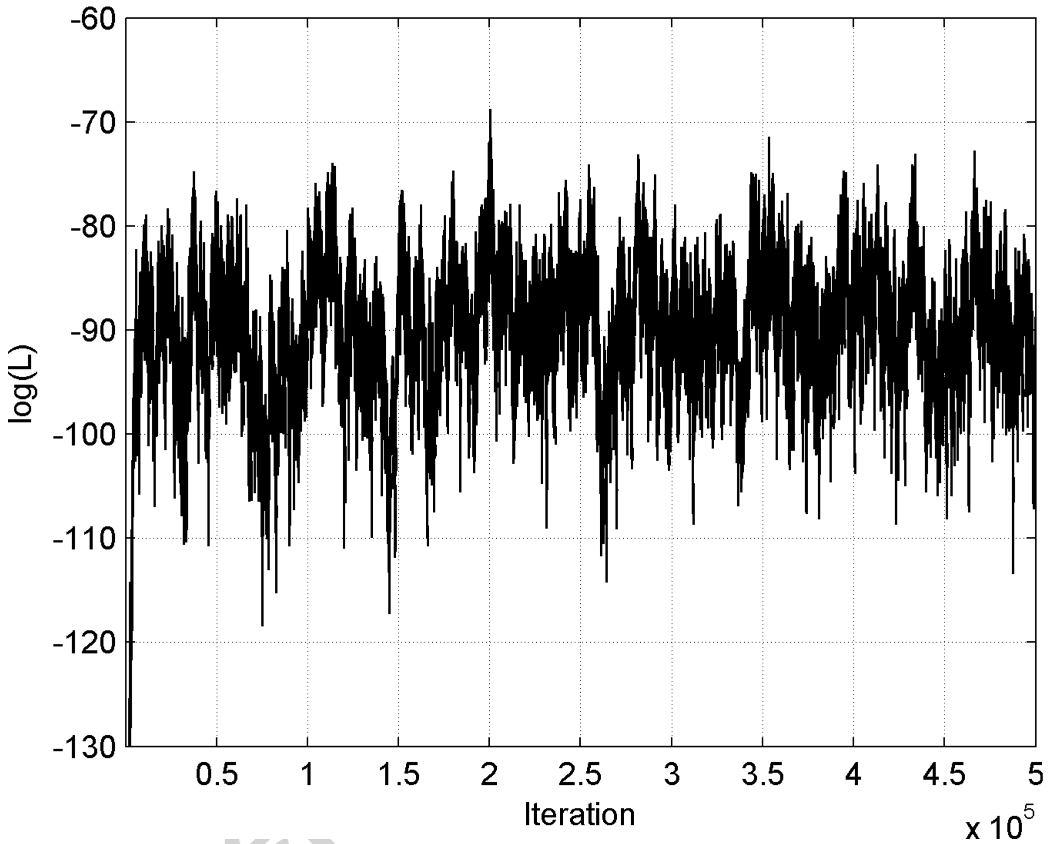



Figure 3: Likelihood as a function of iteration number.

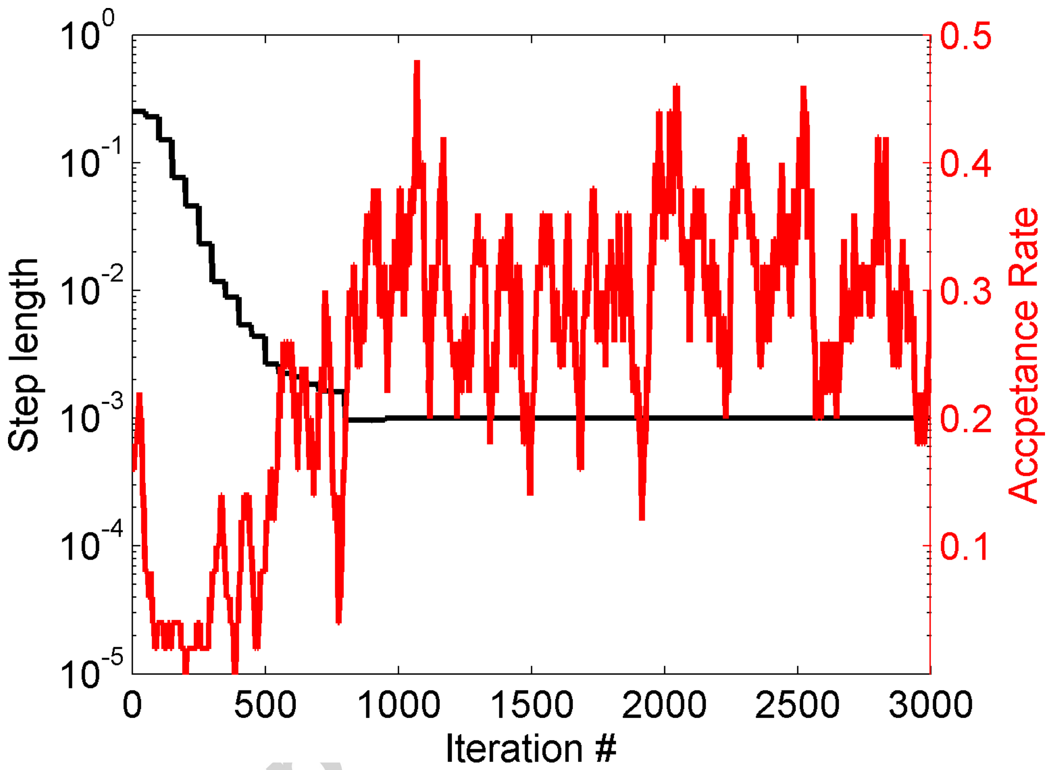



Figure 4: Step length and acceptance rate of the Metropolis algorithm during the first 3000 iterations.

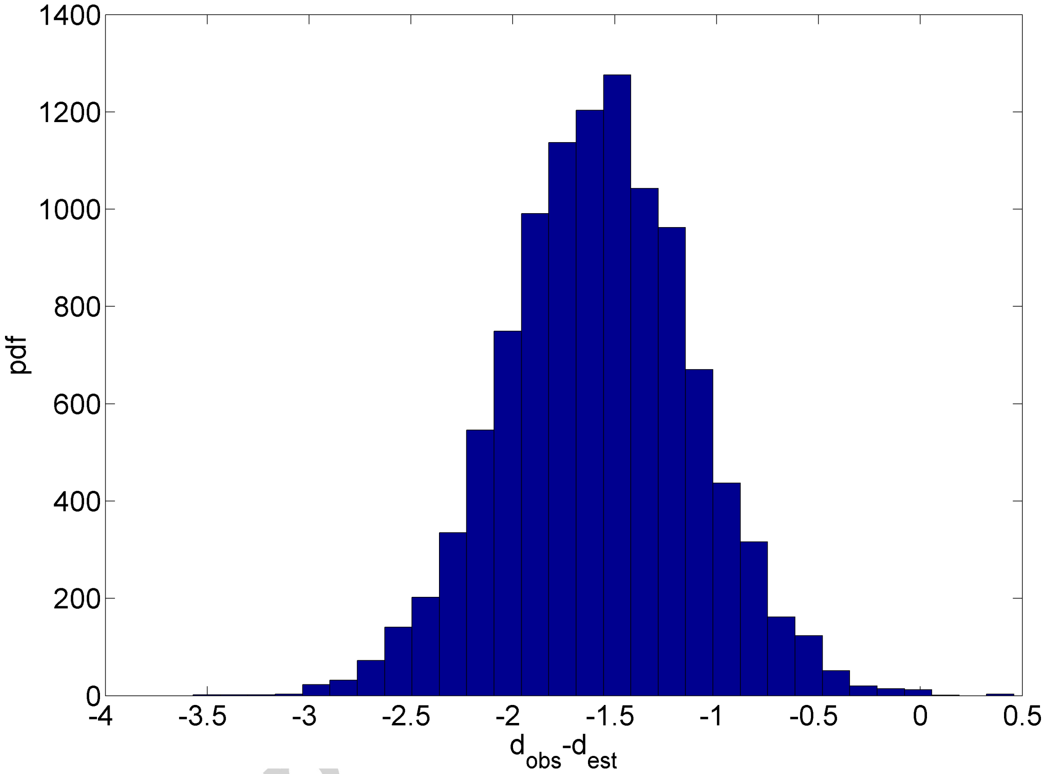





Figure 5: Distribution of the difference between observed traveltime data and the traveltime data associated to 10 realizations of the a posteriori pdf.

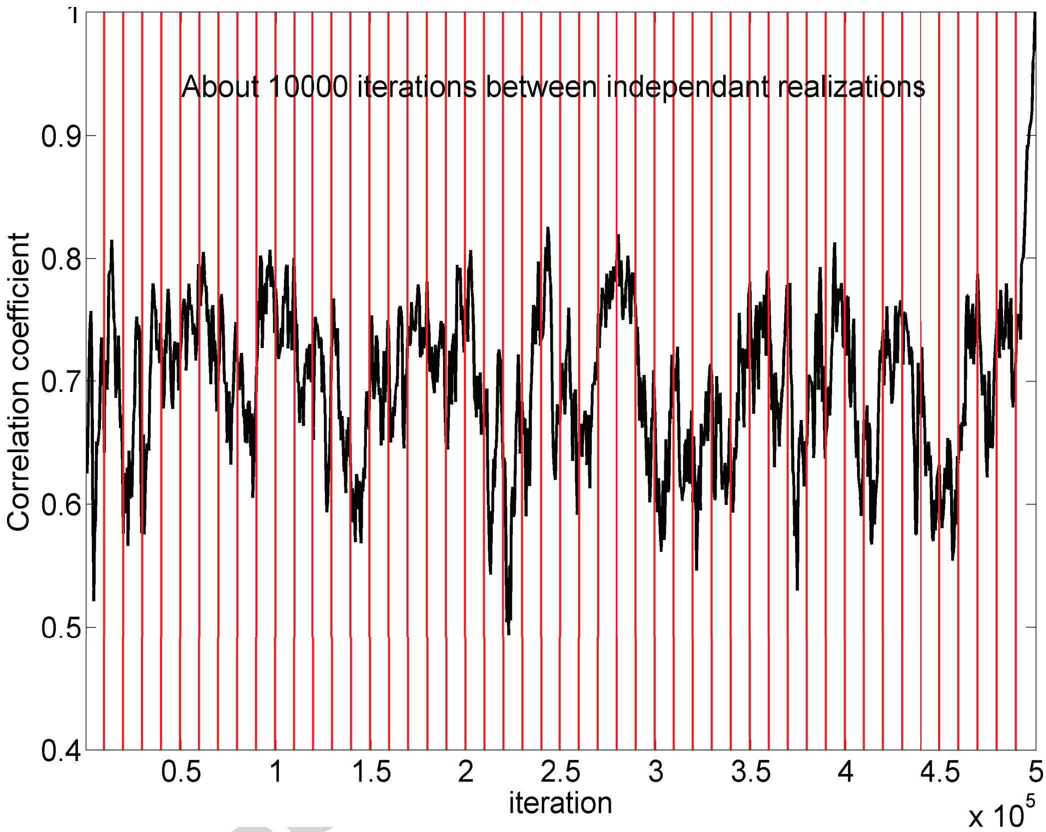



Figure 6: Correlation coefficient between the last accepted model from the a posteriori pdf, and all other realizations of the a posteriori pdf.

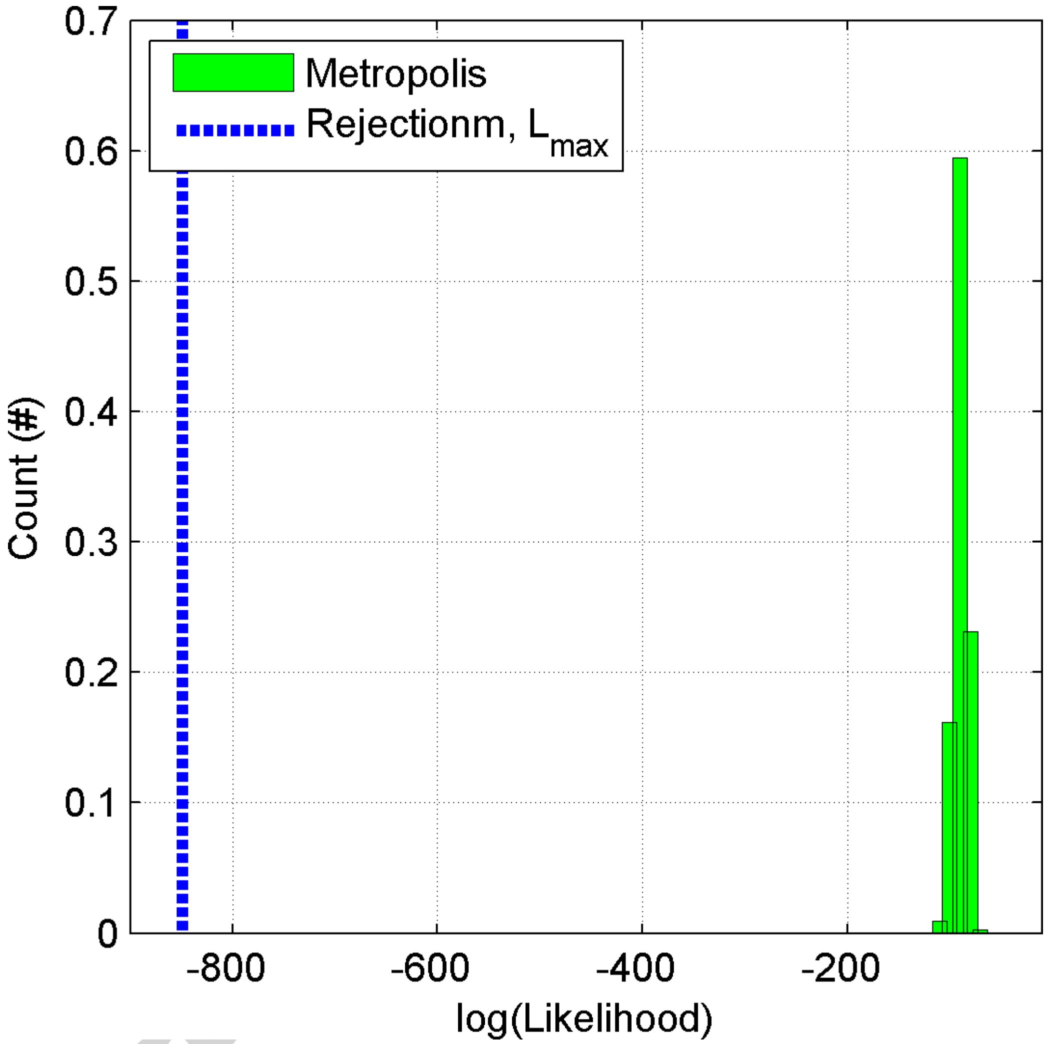



Figure 7: Distribution of log-likelihood of the models considered in 500000 iterations of the Metropolis sampler (green), and the one model of 500000 considered model using rejection sampling with maximum-likelihood (blue dashed line).

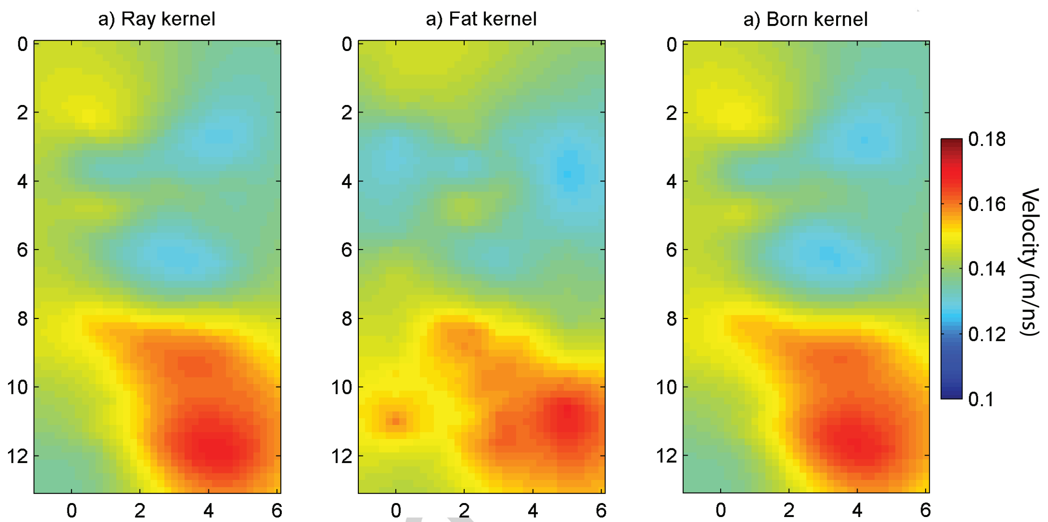



Figure 8: 5 realizations of the a posteriori pdf, using the a) ray, b) fat, and c) Born type linear forward models.

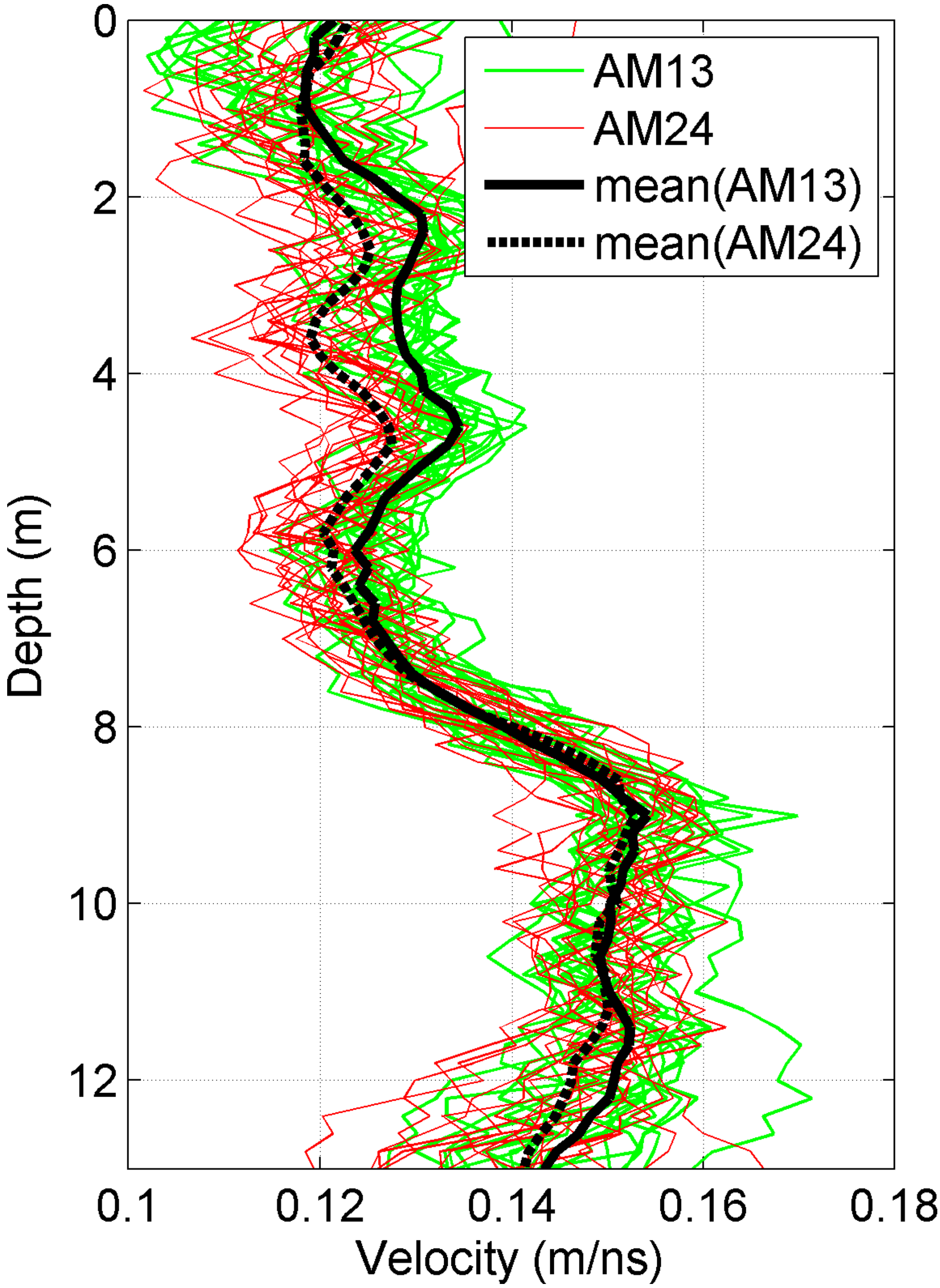



Figure 9: 20 realizations at x=2.5 considering data sets AM13 (green lines) and AM24 (red lines). The solid black line and dashed line show the corresponding average 1D velocity profile of 200 realizations of the a posteriori pdf.

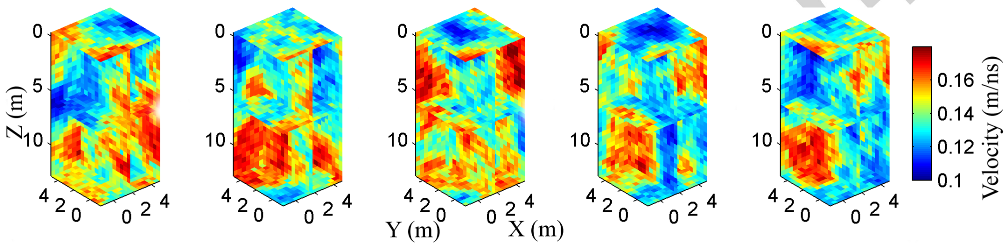



Figure 10: 5 realizations from the a posteriori pdf using the AM1234 3D data set

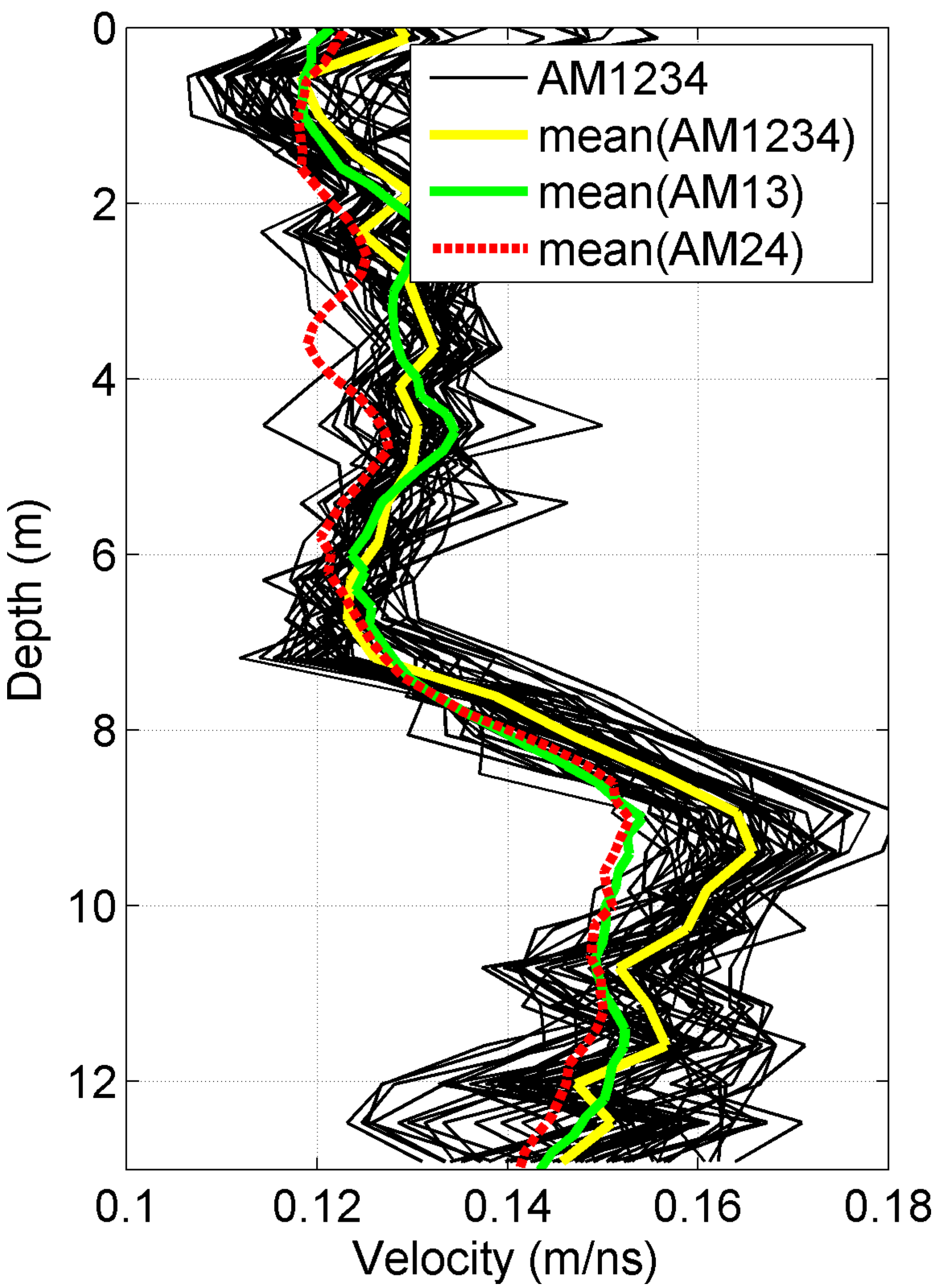



Figure 11: 20 relizations of the a posterior pdf considering the 3D AM1234 data set of the center of the 3D grid where the two 2D profiles intersect (thin black) lines. Also shown is the mean of all s posteriori realizations considering the AM13 (green), AM24 (red), and AM1234 data sets (yellow).

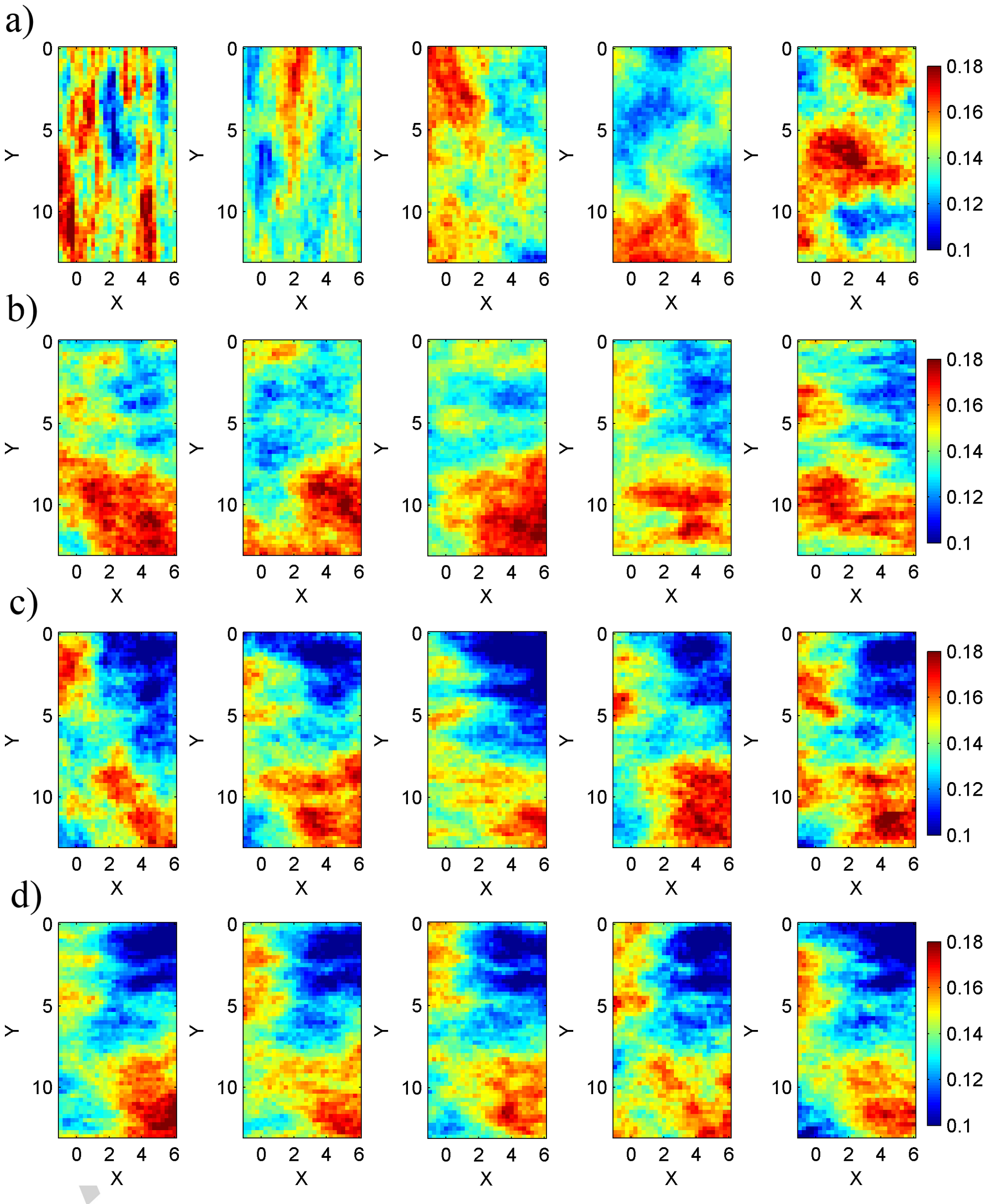



Figure 12: 5 realizations from the a) a priori distribution and a posteriori distribution of the velocity field, using b) 35, c) 140, and d) 702 observed data respectively.

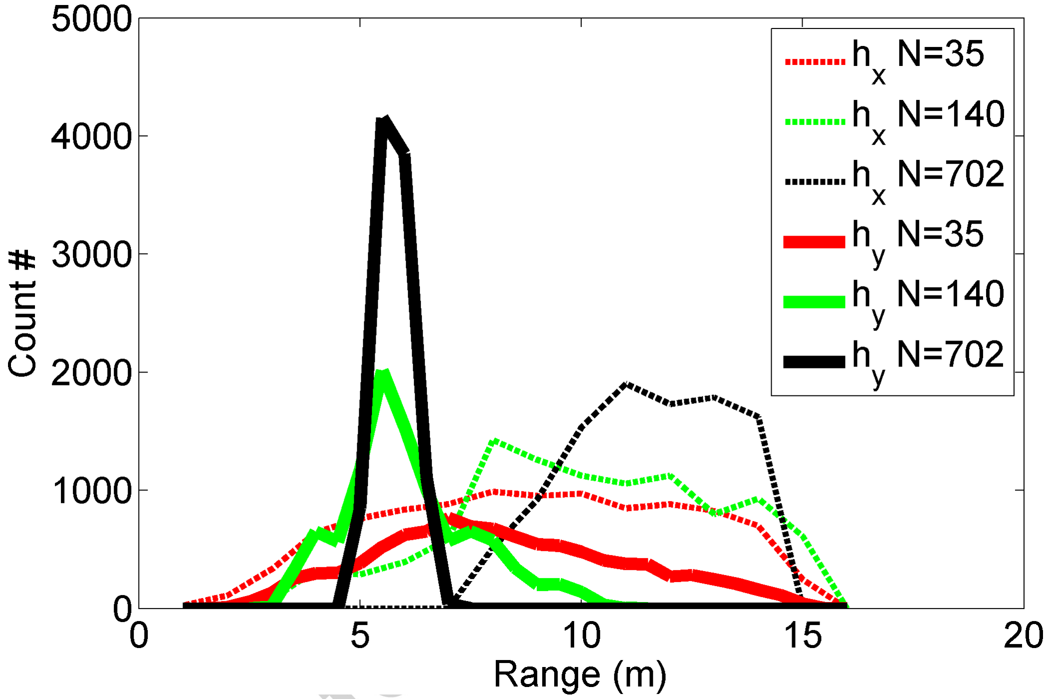



Figure 13: 1D marginal a posteriori distribution of the horizontal ($h_x$) and vertical ($h_x$) range, using 35, 140 and 702 data observations respectively.

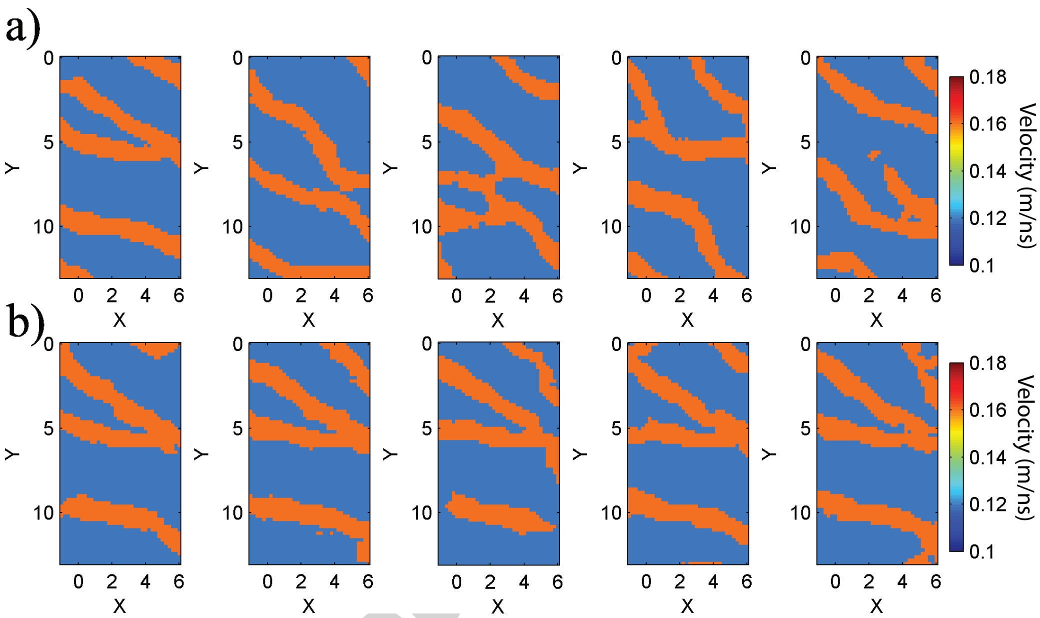



Figure 14: Sample from the a) a priori and b) a posteriori distribution, considering the SNESIM type prior model, and synthetic data. The reference true model is the first of the 5 a priori realizations.