TRMLOC
Reference manual
version 4.2

Wojciech Dębski

Institute of Geophysics Polish Academy of Sciences,
Department of Theoretical Geophysics
Warsaw, Poland

Warsaw, 11-04-2015

Preface

The spatial location of sources of seismic waves is one of the first tasks when transient waves from natural (uncontrolled) sources are analyzed in many branches of physics, including seismology, oceanology, to name a few. Source activity and its spatial variability in time, the geometry of recording network, the complexity and heterogeneity of wave velocity distribution are all factors influencing the performance of location algorithms and accuracy of the achieved results. While estimating of the earthquake foci location is relatively simple a quantitative estimation of the location accuracy is really a challenging task even if the probabilistic inverse method is used because it requires knowledge of statistics of observational, modelling, and *a priori* uncertainties. Unfortunatelly, this is usually not a case. In such a situation additional assumptions are necessary which additionally complicate the analysis of linversion uncertainties. It is thus of a greate importance to have a reliable and efficient software tools which could allowed an advanced error analysis possible. In this reference manual we describe such a software, called TRMLOC which fulfiels above requirements with respect to local sesimicity like , for example, in mines.

The TRMLOC location software (W.Debski and P.Klejment, 2015) performs efficiently the inversion of seismic (acoustic) first arrival time onsets for hypocenter location using the probabilistic inverse theory approach. It provides the maximum likelihood hypocenter location, enables other hypocenter location estimators to be calculated, and, what is most important, it allows an advanced analysis of location (inversion) uncertainties. This advanced performance is possible due to used numerical algorithm implemented in TRMLOC.

This reference provides the most comprehensive description of the TRMLOC software including algorithm description and technical description needed for using the software. Correspondingly, the manual consists of two parts. In the first part the basic concepts and numerical algorithms implemented in TRMLOCare described. The second part is a technical reference manual where all details concerning required parameters, data, file formats, etc. are described. For examples of using TRMLOC readers are refered to research papers (W.Debski and P.Klejment, 2015; Debski, 2015). Especially the former one demonstrates how TRMLOC can be used for the advanced error analysis of location tasks.

Acknowledgements

The development of the TRMLOC algorithm and its basic implementation in C language was financially supported by NCN as a part of the grant No. *2011/01/B/ST10/07305*. Creation of this reference manual and the port of TRMLOC to IS-EPOS platform was financially support by NCBiR under the contract *POIG.02.03.00-14-090/13*

Contents

List of Figures

List of Tables

Part I

Algorithms description

Chapter 1

Introduction - location algorithms

Let us assume that to locate a given source, the arrival times $\mathbf{t}_i^{obs}$ $i = 1 \cdots N_s$ are read from waveforms recorded by N_s sensors (geophones, seismometers, piezoceramic transducers, etc.). Let $\mathbf{t}_i^{th}(\mathbf{m})$ denote the theoretically predicted arrival time of waves originating at the point $\mathbf{m}$ and recorded by i^{th} sensor where the location parameters $\mathbf{m} = (\vec{r}, t_o)$ include three spatial coordinates ($\vec{r} = (x, y, z)$) and the origin time of the event t_o so

$$\mathbf{t}_i^{th}(\mathbf{m}) \;=\; t_o + \Delta_i(\vec{r}) \tag{1.1}$$

where $\Delta_i(\vec{r})$ is the propagation time from the source to a given sensor.

Finding the hypocenter location can now be formulated as the optimization task (Aki and Richards, 1985): searching for the model $\mathbf{m}^{ml}$ which minimizes the differences between observed ($\mathbf{t}_i^{obs}$) and predicted ($\mathbf{t}_i^{th}(\mathbf{m})$ travel times. The solution can be obtained by means of any convenient numerical optimization algorithm.

To complete the location task, an error analysis should be performed in order to evaluate the reliability of the solution. The most popular approach to this task is based on the linearizion of the optimized function $S(\mathbf{m})$ around the solution $\mathbf{m}^{ml}$ and calculation of the covariance matrix (Gibowicz and Kijko, 1994). However, this simple method is often not reliable. It fails if observational and/or modeling errors have nontrivial statistics, the recording network geometry is complicated or the velocity model has a complex structure (Lomax, 2005; Lomax et al., 2009; Rudzinski and Debski, 2012).

Another, probabilistic approach to the source location task relies on assigning to each model $\mathbf{m}$ (point in space and time) the *a posteriori* probability of $\mathbf{m}$ being the true source location (Debski, 2010; Lomax et al., 2000, 2009). The advantage of this approach is the possibility of full and exhaustive error and resolution analysis (Debski, 2010). In the simplest case, the *a posteriori* probability density $\sigma(\mathbf{m})$ assigned to model $\mathbf{m}$ reads (Mosegaard and Tarantola, 2002; Debski, 2010)

$$\sigma(\mathbf{m}) \;=\; \frac{1}{Z} f(\mathbf{m})\, L(\mathbf{m}) \tag{1.2}$$

where Z is the normalization factor called *evidence*, $f(\mathbf{m})$ is the probability density function describing the *a priori* estimation of the source location. The second term, traditionally called the likelihood function is defined as follow

$$L(\mathbf{m}) \;=\; \exp\left(-S(\mathbf{m})\right). \tag{1.3}$$

where

$$S(\mathbf{m}) = ||\mathbf{t}_i^{th}(\mathbf{m}) - \mathbf{t}_i^{obs}|| \tag{1.4}$$

is the so-called misfit function and $||\cdot||$ is a norm in the data space The choice of a given norm (l_1, l_2, Cauchy, etc.) reflects our expectations about errors statistics, existence of outliers, systematic bias, etc (Debski, 2010).

Various numerical estimators, like the maximum likelihood model ($\mathbf{m}^{ml}$) which maximizes $\sigma(\mathbf{m})$, the average model ($\mathbf{m}^{avr}$), the covariance matrix, etc. can easily be calculated from $\sigma(\mathbf{m})$. The technique is very general but it requires exhaustive sampling of the model space to determine the characteristics of $\sigma(\mathbf{m})$. Consequently, the approach is computationally demanding even if the very efficient Markov Chain Monte Carlo sampling technique (Gilks et al., 1995; Lomax et al., 2009; Debski, 2010) is used.

The new possibilities of the full probabilistic seismic data inversion for hypocenter coordinates opening when the time reversal mirroring technique is employed. This technique carefully analyzed in laboratory experiments (see, e.g., Fink, 1997; Ulrich et al., 2008), by numerical simulations (see, e.g., Kremers et al., 2011; Steiner and Saenger, 2012; Scalerandi et al., 2009), and theoretical investigations (see, e.g., Tromp et al., 2005; Ulrich et al., 2009; Masson et al., 2014) has already found out application in seismic prospecting (see, e.g., Witten and Artman, 2011; Gajewski and Tessmer, 2010) and location of seismic tremors (see, e.g., Larmat et al., 2008; Artman et al., 2010; O‘Brien et al., 2011). Combaining this technique with a modern eikonal solver has lead us to the proposition of the new TRMLOC location algorithm which enables very fast Bayesian inversion of travel time onset data for hypocenter location.

Chapter 2

Probabilistic Inverse Technique

2.1 Introduction

The hypocenter location inverse problems can be stated as a parameter estimation problem: having a given set of data (for example, a set of arrival times) and knowing how to model the seismic wave propagation times for given velocity model, what are the values of hypocener coordinates? The question "what are the values" should be understood not only in terms of obtaining the numerical values but also as the task of estimating the uncertainties of obtained results. The last, error analysis issue, is usually quite nontrivial because of three sources of uncertainties, namely the finite quality of measurements, an approximate, imprecise forward modelling, and vague, often difficult to quantify *a priori* expectations. An additional complication arrises due to the nonlinearity of the forward modeling.

To solve such inverse problem three different classes of numerical algorithms are commonly used (Menke, 1989; Tarantola, 1987; Parker, 1994). They are listed in Table 2.1.

Table 2.1: Comparison of three methods of solving inverse problems.

Method	Advantages	Limitations
Algebraic (*LSQR*) $\mathbf{m}^{ml} = (\mathbf{G}^T\mathbf{G} + \gamma\mathbf{I})^{-1}\mathbf{G}^T \cdot \mathbf{t}^{obs}$	- Simplicity - Very fast	- Only linear problems - Lack of robustness
Optimization $\|\mathbf{G}(\mathbf{m}) - \mathbf{t}^{obs}\| = \min$	- Simplicity - Fully nonlinear	- No error estimation
Probabilistic $p(\mathbf{m}) = p_A(\mathbf{m})\mathrm{L}(\mathbf{m}, \mathbf{t}^{obs})$	- Fully nonlinear - Full error handling	- More complex theory - Efficient sampler needed

In the TRMLOC algorithm we use only the probabilistic (hitorically called also Bayessian) approach. The basic principles of the method are following (Tarantola, 2005; Debski, 2010)

The probabilistic inverse theory can be regarded as a set of mathematical rules which extend classical experimental data analysis (Brandt, 1999) to the case where estimated parameters cannot be measured directly. It uses the statistical approach to quantify various types of uncertainties which appear when solving inverse problems. As a consequence of the unified treatment of all uncertainties, the probabilistic inverse theory provides an answer to the problem in hand logically expressed in a statistical way by means of the so-called *a posteriori* probability distribution It can be vougly interpreted as the statistics of inversion errors..

The probabilistic inverse theory is built around the notion of pieces of information which can be understood according to common sense as any information we have in hand when solving inverse problems. The solution of an inverse problem is regarded as a kind of extraction and joining of available information – an inference process. This process is quantified by means of the mathematical theory of probability (Debski, 2010). It should be kept in mind, though, that statistical methods used by the probabilistic inverse theory are only very convenient mathematical tools for reformulation of the inverse problem in a general way. They do not actually solve any inverse task but only provide a framework for it.

Actually, the most important operational difference between the probabilistic approach and the classical ones consists in the different form of the solutions. While the algebraic and optimization techniques provide a single, in a sense optimum estimation of the sought parameters, the probabilistic solution is a probability distribution over the model space which quantifies the “chance” that a given model is the true one. Thus, the probabilistic inverse theory reformulate the inverse problem from the optimization task - searching for the “best fitting” model to the problem of inspecting the *a posteriori* probability density.

The most comprehensive solution of inverse problems within the probabilistic approach relies thus on sampling of the *a posteriori* pdf. In most practical cases this can be done only by means of stochastic, Monte Carlo (MC) methods due to the large number of parameters (Robert and Casella, 1999; Fishman, 1996; Curtis and Lomax, 2001; Debski, 2004).

2.2 Building *a posteriori* probability density

It has been shown by (Tarantola and Vallete, 1982; Tarantola, 2005; Debski, 2010) that *a posteriori* probability density function $p(\mathbf{m})$ is the product of the distribution $f(\mathbf{m})$ describing *a priori* information by the *likelihood function* $L(\mathbf{m})$ which measures to what extent theoretical predictions fit the observed data:

$$p(\mathbf{m}) = const. f(\mathbf{m})\, L(\mathbf{m}) \tag{2.1}$$

where the constant represents a normalization of the probability density and the *likelihood function* $L(\mathbf{m})$ reads (see, e.g. Tarantola, 2005; Duijndam, 1988; Jackson and Matsu’ura, 1985)

$$L(\mathbf{m}) = \exp\left\{-\|\mathbf{t}^{obs} - \mathbf{t}^{th}(\mathbf{m})\|\right\}. \tag{2.2}$$

The symbol $\| \cdot \|$ stands for a norm which is employed to measure the “distance” between observed $\mathbf{t}$ and predicted $\mathbf{t}^{th}$ data. In the present studies the l_1 - norms with constant weightings (error variance) were assumed for which, for example

$$\|\mathbf{t}^{obs} - \mathbf{t}^{th}(\mathbf{m})\| = \frac{\left|\mathbf{t}^{obs} - \mathbf{t}^{th}(\mathbf{m})\right|}{\sigma_d} = \sum_i \frac{\left|t_i^{obs} - t_i^{th}(\mathbf{m})\right|}{\sigma_d} \tag{2.3}$$

Knowledge of the $p(\mathbf{m})$ distribution allows not only to find the most likelihood model $\mathbf{m}^{ml}$ for which $p(\mathbf{m}^{ml}) = \max$ (the equivalent of the optimization or algebraic solution) but also other characteristics like, for example, the average model and the *a posteriori* variance which provides a convenient measure of imaging accuracy. These two basic characteristics of the *a posteriori* distribution $p(\mathbf{m})$ are indeed very important.

More than just a technical problem emerges at this point, namely how to inspect the *a posteriori* distribution to extract the required information. The problem arises because in most practical cases $p(\mathbf{m})$ is a complicated, multi-parameter function. The most elementary approach relies on the calculation of the point estimators, among which the most useful are the lowest order central moments (Jeffreys, 1983):

1. the average model

$$\mathbf{m}^{avr} = \int_{\mathcal{M}} \mathbf{m}\, p(\mathbf{m})\mathrm{d}\mathbf{m}, \tag{2.4}$$

2. the covariance matrix

$$Cp_{ij} = \int_{\mathcal{M}} (\mathbf{m}_i - \mathbf{m}_i^{avr})(\mathbf{m}_j - \mathbf{m}_j^{avr})\, p(\mathbf{m})\mathrm{d}\mathbf{m}. \tag{2.5}$$

The importance of the average model $\mathbf{m}^{avr}$ comes from the fact that it provides not only information on the best fitting model but also includes information about other plausible models from the neighborhood of the “best” model $\mathbf{m}^{ml}$. If sub-optimum models defined as those for which $p(\mathbf{m}) \approx p(\mathbf{m}^{ml})$ are similar to $\mathbf{m}^{ml}$, then $\mathbf{m}^{avr} \sim \mathbf{m}^{ml}$. Thus, a simple comparison of $\mathbf{m}^{avr}$ and $\mathbf{m}^{ml}$ provides a qualitative evaluation of the reliability of the inversion procedure: the more $\mathbf{m}^{avr}$ differs from $\mathbf{m}^{ml}$, the more complex and non-bell-shaped is the form of the *a posteriori* pdf. It immediately implies that in such a case more care must be taken when interpreting the inversion results, especially inversion uncertainties. In such situations, using the confidence levels instead of the covariance matrix is highly recommended (Jeffreys, 1983).

The diagonal elements of the *a posteriori* covariance matrix Cp are convenient estimators of the inversion uncertainties for each component of $\mathbf{m}$ while the non-diagonal elements measure the degree of correlation between pairs of parameters (Menke, 1989; Jeffreys, 1983). In fact, Cp given by Eq. 2.5 is a generalization of the least-squares covariance matrix to the case of an arbitrary statistics $p(\mathbf{m})$ including possibly nonlinear forward problems. As in the case of the average solution $\mathbf{m}^{avr}$, the posterior covariance matrix is meaningful only if the $p(\mathbf{m})$ distribution is unimodal. In cases of multi-modality, existence of non-resolved directions in the model space or other “pathologies”, a more exhaustive error analysis is necessary by a full inspection of the *a posteriori* distribution (Tarantola, 2005; Wiejacz and Debski, 2001; Rudzinski and Debski, 2011).

Two additional point estimators, namely the average model $\mathbf{m}^{avr}$ and the covariance matrix Cp, provide significantly more information than the classical $\mathbf{m}^{ml}$ solution but their evaluation requires full knowledge of $p(\mathbf{m})$ in order to calculate the appropriate integrals. The calculations are trivial only in the case when $p(\mathbf{m})$ is a Gaussian distribution (this requires a linear forward problem: $\mathbf{G}(\mathbf{m}) = \mathbf{G}\mathbf{m}$ and the Gaussian form of $f(\mathbf{m})$). In such a case $\mathbf{m}^{avr} = \mathbf{m}^{ml}$ is given by the classical least squares formula (see,e.g. Menke, 1989; Parker, 1994) and the covariance *a posteriori* matrix is given by Eq. 2.5. In any other case we need an efficient method of calculating the multi-dimensional integrals appearing in Eqs. 2.4, 2.5. At this point the Monte Carlo sampling technique proves to be very useful.

If a more comprehensive description of $p(\mathbf{m})$ is required, the marginal pdf's should be calculated because the *a posteriori* marginal distributions give a deeper insight into the form of $p(\mathbf{m})$ than the discussed point estimators. Their inspection is extremely important for the correct interpretation of the tomographic results. The one-dimensional (1D) marginal *a posteriori* pdf distribution is defined by integrating out all but one parameter from $p(\mathbf{m})$

$$\sigma_i(m_i) = \int p(\mathbf{m}) \prod_{j \neq i} \mathrm{d}m_j. \tag{2.6}$$

Multi-dimensional marginals are defined in a similar way. It is important to realize that marginal pdf distributions contain exactly the same information on thought parameters as $p(\mathbf{m})$ except for the correlation with other parameters (Jeffreys, 1983; Rudzinski and Debski, 2011). However, inspections of the full *a posteriori* pdf $p(\mathbf{m})$ and marginal pdf $\sigma_i(m_i)$ are not equivalent (Debski, 2010). To see the difference, let us consider two maximum likelihood solutions derived from the $p(\mathbf{m})$ and $\sigma_i(m_i)$ *a posteriori* distributions. In the first case, when seeking the $\mathbf{m}^{ml}$ model we consider the set of N_m parameters which simultaneously maximizes $p(\mathbf{m})$. In the second case we solve the 1D optimization problem seeking the optimum value of only one parameter, no matter what the values of the other components of $\mathbf{m}$ are. In other words, in the former case we obtain information about all the parameters simultaneously while the inspection of the marginal distributions provides information only about selected parameters, ignoring any relations with the remaining ones.

In the most frequently encountered situations, point estimators calculated from the marginal distributions and from $p(\mathbf{m})$ differ slightly (Wiejacz and Debski, 2001). A difference significantly larger than the corresponding elements of the *a posteriori* covariance matrix indicates a very strong correlation among parameters which, besides the case of intrinsically correlated parameters, may indicate that some parameters or their combinations are not resolved by the data.

Inspection of marginal pdf's is always recommended to verify if $p(\mathbf{m})$ is a multi-modal distribution or not. This is especially important if $\mathbf{m}^{avr}$ is used as the numerical estimator. Moreover, the inspection of the marginal pdf provides conclusive verification of whether estimating the inversion errors by *a posteriori* covariance is justified or not.

Before moving on to a description of the Monte Carlo technique used in the context of the Bayesian inversion, let us make two general comments on the use of the Bayesian inversion technique. The first one is the question why we should use this approach in the context of tomographic imaging. The answer is that the method allows us to estimate the reliability of the obtained solution. The point is that solving tomography problems we face various types of uncertainties (Tarantola, 2005). Besides observational uncertainties (accuracy of travel time estimation) we are dealing with theoretical errors (for example following the high frequency ray approximation) and numerical calculations which are also a source of additional errors (for example by using an approximate method of ray tracing). All these uncertainties contribute in a very complicated way to the final *a posteriori* imaging errors and only the Bayesian (probabilistic) method can treat them consistently.

The second comment concerns a popular use of the Bayesian inversion technique. Very often, instead of sampling of the *a posteriori* pdf only the global maximum of the $p(\mathbf{m})$ is sought (e.g. Debski and Young, 2002) and claimed to be the solution of the Bayesian inversion. This procedure is obviously correct, but it reduces the full Bayesian inversion to the optimization method, disregarding the potential ability of the Bayesian method to deal with error analysis. A similar situation occurs when the *a posteriori* pdf is a Gaussian distribution due to a specific assumption about the *a priori* pdf and the form of the likelihood function (linear or linearized forward problem and Gaussian error statistics). In such a case, the *a posteriori* pdf $p(\mathbf{m})$ is fully described by two parameters, namely the likelihood model $\mathbf{m}^{ml}$ and the covariance matrix Cp equivalent to the algebraic solution (Menke, 1989; Tarantola, 1987; Parker, 1994). Solving the inverse problem is then often understood as finding their numerical values, which is obviously correct but is more of an algebraic method than a general Bayesian technique.

Chapter 3

Time Reversal Mirroring

The wave equation (Aki and Richards, 1985) describing propagation of seismic waves exhibits among others two very important features. First of all, being second order partial differential equation with only add derivatives of time the equation is invariant under time reversal. Thus, if only boundary conditions does not depend on time the solution for forward and back propagating in time waves are identical. Secondly, the equation exhibits the spatial reciprocity invariance which means that wave propagation between two arbitrary points is invariant with respect to exchange the source and receiver points: the waves from source located at point A and recorded at point B are the same as waves recorded at point A if the same source is put at point B.

Combining both above properties of the wave equation allowed to construct a very simple and efficient numerical algorithm which can be used to study properties of seismic sources (Fink et al., 2000; Kremers et al., 2011). It relays on putting the virtual sources at the receiver locations (reciprocity principle) and simulating propagation of seismic waves from such virtual sources assuming as their temporal changes the recorded real signals with reversed time: the last arriving signal is "resend" as the first one. Resend signals due to a complicated interferences not only focus in the point where the real source ruptured and at the origin time of the rupture but also provide very important information on kinematics and dynamics of the rupture process. It is worth to mentioned at this point that this approach is an example of the back projection inversion technique as it directly "projects" the observed data onto the values of parameters used to describe the seismic source. One of the most important feature of the back projection inversion technique is its very high computational efficiency as it avoids any direct sampling of the model space. Besides that, using the full waveform as input data allows to proceed cases with very low Signal to Noise ratio (Steiner and Saenger, 2011) and/or achieved the very high "super-resolution" much beyond theclassical diffraction limit (Hanafy et al., 2009).

Currently, the practical applications of the time reversal algorithm for seismic source location consist of two elements namely, a direct nonlinear back-projection of the recorded signals onto the location space (re-sending recorded seismograms) followed by searching for the loci at which the positive waves interference lead to the maximum of the appropriately chosen imaging field (Larmat et al., 2009).

Unfortunately, this procedure is subjected to many disturbing factors which cause that the information about seismic source retrieved this way can be quite errorneous. The two most important factors introducing uncertainties into this algorithm are noisy data and limited knowledge of the velocity and attenuation properties of the medium through which waves are supposed to be back-in-time propagated. Recently, it has also been pointed out (Kremers et al., 2011) that some properties of the source are not, or at least are very poorly resolved by this approach. In such a case an analysis of the uncertainties of the imaging procedure is of the highest importance (Tarantola, 2005).

The analysis of the resolution and/or accuracy of inversion procedures is usually the very complicated and time consuming task because it requires an exhaustive exploring of the model space. Thus, the necessity of performing such analysis using even the modern, very efficient Monte Carlo sampling algorithms like Markov Chain Monte Carlo (Robert and Casella, 1999) essentially destroys the most important feature of the time-reversal imaging approach - its efficiency. Fortunately, this can be avoided when seismic waveforms are inverted for the most basic seismic source parameters, namely source location. The point is that when re-sending numerically the recorded seismograms the displacement field is calculated for all points in the descritized location space during the single forward modelling. Since the displacement field is used to construct the misfit function defining the *a posteriori* probability density simultaneously with the "re-sending" (back modelling) of recorded data the implicit sampling of the model space is performed at no additional numerical cost . This is the corner stone of the proposed algorithm.

Chapter 4

Eikonal Solver: Fast Sweeping Method

Under the high frequency approximation the full wave equation can be split into the eikonal equation describing spatial propagation of wavefronts and transport equation describing changes of wave amplitudes. As we are interested here in travel times modeling, we consider only the eikonal equation which together with the boundary condition at source location Γ reads

$$\begin{array}{l} \nabla T \cdot \nabla T = \frac{1}{v^2} \\ T|_\Gamma = 0 \end{array} \tag{4.1}$$

where T describes wavefront position in space originating from the source at Γ and v denotes velocity. This is a special case of the Hamiltonian-Jacobi, hyperbolic type nonlinear equation for which the term on right-hand side is always positive. For numerical purpose, such equation can be discretized by using the first-order Godunov upwind type discretization (Zhao, 2005; Sethian, 1999) For internal grid points this finite difference scheme leads to the following discrete approximation of Eq. 4.1

$$\left[(T_{i,j} - T_{xmin})^+\right]^2 + \left[(T_{i,j} - T_{ymin})^+\right]^2 = h^2 s_{i,j}^2 \tag{4.2}$$

where i, j are indexes of the grid point $\mathbf{x}_{i,j} = (x_i, y_j)$, h is the grid size (for simplicity the quadratic grid is assumed), $s_{i,j}$ is the value of slowness at grid point $(\mathbf{x}_{i,j})$ $(s_{i,j} = 1/v_{i,j})$ and the following shorthand notation is used:

$$T_{xmin} = \min(T_{i-1,j}, T_{i+1,j}), \qquad T_{ymin} = \min(T_{i,j-1}, T_{i,j+1}) \tag{4.3}$$

and

$$(x)^+ = \begin{cases} x, x > 0 \\ 0, x \leq 0 \end{cases} \tag{4.4}$$

The Fast Sweeping Algorithm proposed by Zhao (2005) is using the above discretization and solves the resulting system of nonlinear equation iteratively as follows.

- Initialization: a large positive value is assigned to all $T_{i,j}$. Then, for all grid points $(\mathbf{x}_{i,j}^s)$ within the source of waves (it can be a single grid node for a point-like seismic source model or an extended area Γ for the finite source model) the boundary condition $T(\mathbf{x}^s) = 0$ is set.

- Iterations with alternating sweeping: the following procedure is repeated until the conversion to stable solution is reached.

  - At each grid point $(\mathbf{x}_{i,j})$ not set during the initializations the solution $\tilde{T}$ is computed using current values of T at neighborhood points and then $T_{i,j}$ is updated us follows

    $$T_{i,j}^{new} = \min\left(T_{i,j}^{cur}, \tilde{T}\right) \tag{4.5}$$

    where the updating solution $\tilde{T}$ is the solution of Eq. 4.2 and reads

    $$\tilde{T}_{i,j} = \begin{cases} \min(T_{xmin}, T_{ymin}) + s_{ij}h & |T_{xmin} - T_{ymin}| \geq s_{ij}h, \\ \frac{T_{xmin}+T_{ymin}+\sqrt{2s_{ij}^2h^2-(T_{xmin}-T_{ymin})^2}}{2} & |T_{xmin} - T_{ymin}| < s_{ij}h, \end{cases} \tag{4.6}$$

  - During one iteration the value of $\tilde{T}$ is recalculated four times with different alternating orderings of grid sweeping:

    $$\begin{array}{l} a)\ i = 1 : N_x,\ \ j = 1 : N_y \\ b)\ i = 1 : N_x,\ \ j = N_y : 1 \\ c)\ i = N_x : 1,\ \ j = 1 : N_y \\ d)\ i = N_x : 1,\ \ j = N_y : 1 \end{array} . \tag{4.7}$$

As follows from the above descriptio the proposed scheme shows the numerical complexity of order $O(kN)$ where N is the total number of the grid nodes and k is a constant depending on the number of iterations.

The number of iterations to be performed depends on the complexity of the velocity model. In many cases if the velocity model is reasonably smooth and without large velocity contrasts only a few (usually 2-3) iterations are sufficient for convergence of the algorithm. The reason is that each sweep (Eq. 4.7) provides the exact solution in one iteration for one spatial quarter, provided the characteristics of the eikonal equation do not intersect (Zhao, 2005). This is the case of smooth velocity models. Moreover, the upwind Godunov difference scheme enforces the causality of the solution (Sethian, 1999), because the solution at a given grid point is determined by only those neighborhood points for which T is smaller. This is exactly what happens (Heughen's principle) during an advancing of the wavefront. In consequence, the iteration procedure converges very quickly and the solution is optimally accurate. However, we have to keep in mind that the used upwind scheme is based on the first order difference stencil. This implies the first order accuracy of the method.

In figure 4.1 an example of a simulation of wavefront positions for the velocity model of Rudna mine (vertical section) is shown. The ability of the algorithm to model the complex

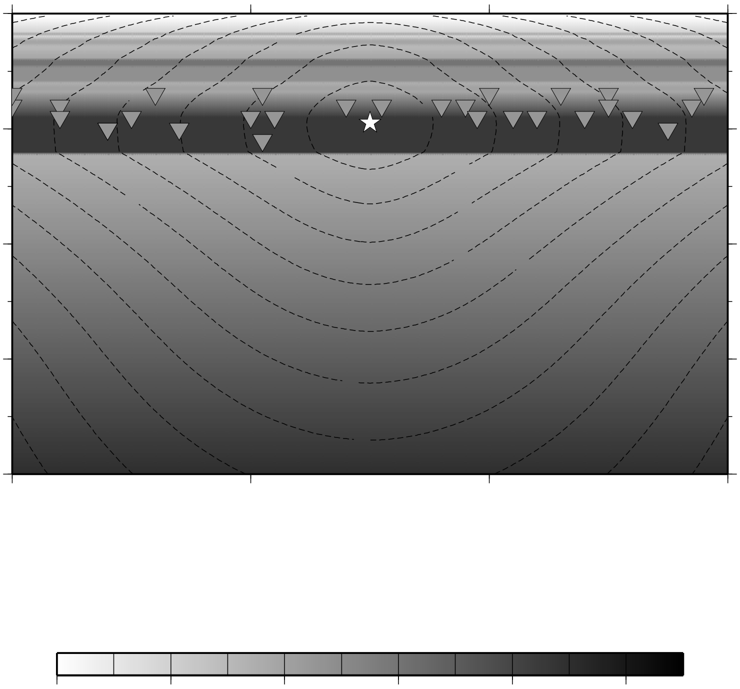


Figure 4.1: The vertical section of the synthetic velocity model and wavefront positions simulated by FSM technique from the hypothetic rockburst (star). Open triangles denote seismometers of the underground seismic network operating in the mine.

wavefront structure, including reflection and refraction effects, is clearly visible. The computation time for this simulation (2D grid with $N = 4\,10^5$ grid points) on 4 cores 2.4GHz clocked Intel processor was about 90 milliseconds.

Chapter 5

TRMLOC algorithm

The existing application of the above-described time reversal technique for hypocenter location consists of two steps, namely back propagation ("re-sending") of recorded signals and scanning of the model (location) space in order to find the optimum location where all the back-propagated signals positively interfere (O'Brien et al., 2011). The algorithm used in this paper extends the above idea in two aspects. First of all, instead of searching for a point at which the largest positive interference occurs, we propose to construct the *a posteriori* probability density based on the differences of the back-propagated observational data. Thus a full probabilistic (Bayesian) technique is implemented in the algorithm. Secondly, for the location of sources with well determined time onsets on a given set of sensors, we do not need to perform a full waveform back propagation. Instead, we consider only the wavefronts whose propagation in time is described by the much simpler to solve eikonal equation. The very important consequence of such formulation of the location task is that the most numerically demanding part of any probabilistic inversion, namely sampling of the *a posteriori* pdf can be performed implicit.

We propose to construct the *a posteriori* probability density using the differential misfit function $\bar{S}(\mathbf{m})$

$$\bar{S}(\mathbf{m}) = \frac{1}{2N_s} \sum_{i,k;i\neq k} ||\mathbf{t}_m^i - \mathbf{t}_m^k|| \tag{5.1}$$

based on the difference of the back propagated wavefronts from all considered sensors. In this equation, $\mathbf{t}_m^i$ and $\mathbf{t}_m^k$ stand for waveform onsets recorded by i-th and k-th receivers, respectively, and back-propagated to the point $\mathbf{m}$. N_s is the number of the receivers used (number of available observational data) and 1/2 takes into account the symmetry of the sum. Following this assumption, we postulate the *a posteriori* probability distribution as

$$\sigma(\mathbf{m}) \ = \ const.f(\mathbf{m}) \exp\left(-\bar{S}(\mathbf{m})\right) \tag{5.2}$$

The physical intuition behind the above definition of $S(\mathbf{m})$ is quite clear. In an ideal case (no noise, exact forward modelling) all back-propagated travel times should be equal to the source origin time (t_o) at the true hypocenter location point. Thus, the condition for the hypocenter location is the equality of all back-propagated arrival times. Due to the presences of observational and modeling errors, this condition cannot in general be directly fulfilled and thus a reasonable solution is to look for the point in space where $S(\mathbf{m})$ gets minimum. Let us note, that as follows from Eq. 1.1 the origin time t_o does not enter $S(\mathbf{m})$ and thus an original 4D inversion is reduced to 3D problem: a search for the hypocenter's spatial coordinates only.

The idea of using the differential-time form of the misfit function is by no means new and can be traced back in time to Zhou (1994) and Matsu'ura (1984). In various forms it has already been implemented in different optimization-based location algorithms under various names among which the equal difference time (EDT) is the most popular (see, e.g., Lomax et al., 2009; Font et al., 2004; Zhou, 1994). The EDT formulation relies on searching the point $\mathbf{m}$ for which hyperbolic surfaces defined by the condition

$$\Delta_i(\mathbf{m}) - \Delta_j(\mathbf{m}) \ = \ \mathbf{t}_i^{obs} - \mathbf{t}_j^{obs} \tag{5.3}$$

intersect for all pairs of stations (i, j). This condition can be rewritten as $\mathbf{t}_i^{obs} - \Delta_i(\mathbf{m}) = \mathbf{t}_j^{obs} - \Delta_j(\mathbf{m})$ for all (i, j), which is actually the condition of equality of all back-propagated observational time onsets at the hypocenter location. The advantage of using the EDT-type differential misfit function relies in removing of origin time from inversion procedure (Matsu'ura, 1984) and also lower sensitivity of location results to velocity model (Font et al., 2004; Waldhauser and Ellsworth, 2000; Rudzinski and Debski, 2011; Zhou, 1994). Additionally, in the developed algorithm, it has also allowed to perform an implicit sampling of the *a posteriori* distribution, as discussed latter on. The EDT type misfit function is the cornerstone of the modern relative location methods, namely the double differences and extended double differences techniques (Waldhauser and Ellsworth, 2000; Rudzinski and Debski, 2012).

Finding the minimum of $S(\mathbf{m})$ will provide the hypocenter location, so $S(\mathbf{m})$ can serve as the cost function for any optimization-based location algorithm. Much less obvious is whether this misfit function can also be used within the probabilistic inversion framework for generating the likelihood function according to Eq. 5.1. The problem is that the "true" likelihood function $L(\mathbf{m})$ defined by the probabilistic inverse theory is actually a convolution of probability distributions of observational and modelling errors (Tarantola, 2005; Debski, 2010). Thus, from the statistical point of view, it describes the statistic of sum of the observed and modeled errors. Apparently the function $L(\mathbf{m})$ defined by $\bar{S}(\mathbf{m})$ is not such the statistic. It is rather the statistics of sum of differential errors so the question is if the errors estimated by using this proxy of the likelihood function are not systematically biased. Although this point has not been clarified yet, the differential misfit function has already been implemented in some probabilistic location algorithms (Lomax et al., 2000; Rudzinski and Debski, 2011) and we use it also in the TRMLOC algorithm.

Having defined the *a posteriori* distribution $\sigma(\mathbf{m})$ we have to explore the space of model parameters in order to obtain various characteristics of $\sigma(\mathbf{m})$ including the position of the global maximum, checking an existence of secondary maxima, etc. This is the most demanding numerical part of any probabilistic inversion. However, in case of the location task the model space which has to be sampled is exactly the same space (3D configuration space) as that over which the forward modelling operator acts. This opens a possibility of performing an implicit sampling of the *a posteriori* distribution simultaneously with solving the forward problem. The idea is as follows. Assume that the forward modeling method used to calculate $\Delta_i(\mathbf{m})$ is able to provide the back-in-time propagated observed time onsets for a set (for example, regular grided) of spatial points. Then, according to Eq. 5.1, the *a posteriori* distribution $\sigma(\mathbf{m})$ can be immediately calculated with minimum numerical computations for all grid points. This way we have sampled $\sigma(\mathbf{m})$ at all these points. If the points form a dense enough, regular set, we end up with the well sampled $\sigma(\mathbf{m})$ so we do not need any additional sampling indispensable in the classical probabilistic inversion. We call this mechanism the implicit sampling. The forward modeling techniques fulfilling the above requirement are the all wave equation or eikonal solvers based on the finite difference, finite element, spectral elements, or similar numerical methods (Virieux et al., 2009; Sethian, 1999). Thus, summarizing the above consideration we propose the algorithm whose flowchart is shown in Fig. 5.1.

- Discretize space $\mathbf{m} = (X_i, Y_j, Z_k), \quad i, j, k = 1, 2, \ldots$
- Set the *a priori* density function $f(\mathbf{m})$
- Repeat for each receiver (in parallel)
  - back propagate observed time onsets $\mathbf{t}_i^{obs}$ using eikonal (FSM) solver on defined spatial grid
- calculate $S(\mathbf{m}) = \frac{1}{2N_s} \sum\limits_{i,k;i \neq k} (\mathbf{t}_m^i - \mathbf{t}_m^k)^2 / Cp^2$
- determine $\sigma(\mathbf{m}) = const. f(\mathbf{m}) \exp\left(-S(\mathbf{m})\right)$
- calculate statistical estimators $\mathbf{m}^{ml}$, $\mathbf{m}^{avr}$, $\Delta\mathbf{m}$, evidence, entropy, etc.
- if needed perform inspection of the full $\sigma(\mathbf{m})$ or marginal *a posteriori* distributions

Figure 5.1: The basic steps of the TRMLOC algorithm

One very important feature of the algorithm is its high speed, as will be demonstrated later on, which follows from:

- reducing inverse problem from 4D to 3D by eliminating event's origin time from inversion
- employing the modern finite-difference very fast eikonal solver.
- avoiding explicit sampling of the model space: $\sigma(\mathbf{m})$ is evaluated at each grid nodes simultaneously with forward modelings
- parallelization of the algorithm

Let us also note that the backward propagation of the observed time onsets through the back-in-time forward modeling has to be performed only N_s times - as many as the number of sensors is used. This is a direct advantage of using the time-reversal and reciprocity invariance principle.

Actually, the TRMLOC algorithm is very similar to the algorithm NLloc developed by Lomax et al. (2000). Both approaches use the probabilistic inverse approach, eikonal solvers for forward modelling and similar EDT-based likelihood function. The main differences arise from using different eikonal solvers (NLloc uses the method of Podvin and Lacomte (1991) while TRMLOC the Fast Sweeping Method) and from different implementation of the *a posteriori* pdf sampler.

The very important element of the TRMLOC algorithm is the eikonal solver which enables very efficient calculation of the wavefront positions in the entire 3D domain for a general velocity model. Constructing the TRMLOC algorithm, two finite-difference type eikonal solvers were considered, namely the Fast Marching Method (FMA) developed by Sethian (1999) and the Fast Sweeping Method (FSM) developed by Zhao (2005). The FMA algorithm exhibits numerical complexity[1] of the order of $N\log(N)$, where N is the number of all grid nodes and is optimal for complex velocity models (Sethian, 1999). The FSM method is faster for smooth velocity models with numerical complexity proportional to N but it is over-performed by FMA in cases of complex velocity models. TRMLOC has been designed for a local/regional analysis when velocity models are relatively smooth so the FSM technique has been selected.

[1] By numerical complexity we understand here number of floating point operations performed by an algorithm

Chapter 6

Algorithms summary

Being based on general concepts of the modern probabilistic inverse theory, the TRMLOC algorithm exhibits the same level of generality as any other, more traditional Bayesian location algorithms, However, unlike the classical probabilistic approaches, it performs implicit sampling simultaneously with the forward modeling due to the use of the finite difference based eikonal solver and EDT type of the misfit function.

The algorithm has also some limitations. First of all, the eikonal solver provides solutions only for the first arriving seismic phases (or the first arriving P or S waves in case of elastic waves). Including other phases within TRMLOC is possible, but it requires using the full waveform modeling algorithms , or multi-phases extensions to eikonal solver (Hauser et al., 2008; Rawlinson and Sambridge, 2004). In both cases, however, the numerical efficiency of the algorithm is lost.

The next limitation of the algorithm is related to the fact that the spatial resolution achieved by the algorithm is limited by the grid size used by the forward modeling algorithm. Achieving higher resolution requires a finer spatial grid but this increases the computation time linearly. Another problem connected to the spatial grid is that the eikonal solver used by TRMLOC is the first-order differential solver requiring quite fine grid for high numerical accuracy. Using higher-order solvers or more advanced front propagation techniques (Zhang et al., 2005) may thus be advisable.

An accumulated experience gathered when using the TRMLOC algorithm shows that the most time-critical part of the algorithm is calculation of integrated statistical characteristics of the *a posteriori* distribution like evidence, entropy, average model, etc. For large grid, with the number of nodes of order 10^8 it takes about 70-80% of the whole calculation time. A remedy to this bottle-necked part of the algorithm is its redesigning using GPGPU technology which is extremely efficient in this type calculations (Kloc and Danek, 2012; Danek and Debski, 2014). Further efficiency improvement is expected by porting the algorithm, especially the forward modeling part, to the paralel distributed computational platform, using, for example MPI paralelization schemata (Quinn, 2008)

Part II

Technical reference manual

In this part of the manual technical aspects of the TRMLOC software are provided. We begin we a short compilation and installing instruction for Linux and others Unix-type platforms. Next, we explain the floachart of the code, describe in details all parameters controling its behaviour, and formats of input and output files. Although a lot of work was done to make this description clear and self-explanatory some points of runing of TRMLOC may still be doubtful so some examples of input and output files are provided.

Chapter 7

Building and Structure of TRMLOC

7.1 Building TRMLOC

The described in the previous part of the manual TRMLOC algorithm was implemented in the software application of the same name (*trmloc*). The application is designed for running under the Unix/Linux based computers with the full support of paralell computations under the shared-memory paralelization model and an afford to port *trmloc* to a standard cluster MPI-based architectures is undertaken. Newertheless, its use under different operating systems is possible. The only a problematic part of porting *trmloc* to non-unix operating system can be a support of multicore, shared memory parallelization. Actually, *trmloc* uses the OpenMP (ver 3.*) standard to support the paralelization (Quinn, 2008) which can be very easily switched-off in case of troubles with instaling it on given platform.

The *trmloc* application is coded in standard C language and besides OpenMP extension and corresponding libraries relias only on standard C and Math libreries. Originally developed under Linux system is compiled by default using *GCC* compiler throught the standard `Make` utility. Compilation under different platform depends on used compiler and development environment.

The standard instalation should run as follow:

1. unpack the source code in selected directory

2. go to the created CURRENT subdirectory

3. edit the `Makefile.rul` end change the `BINDIR` to point the directory where the executable is expected to be copied. If the OpenMP support has to be switched off (not recommended) clean `OPEN_MP` variable by removing `-fopenmp` from the definition of `OPEN_MP` and removing `-fopenmp` from `LD_FLAG`.

4. go to the CURRENT/TRMLOC subdirectory. and type sequence:
   > make,
   >make install,
   > make clean

5. to verify if compilation and instalation was sucessful type
   > trmloc *-v*

Inspecting the `Makefile` in the CURRENT/TRMLOC subdirectory shows which source files are necessary for sucessfull compilation of the code. In case or necessity of re-arrangement of source code files appropriate changes has to be also in the Makefile.

7.2 Structure of TRMLOC

The *trmloc* application uses the standard data processing schemata shown in Fig. 7.1

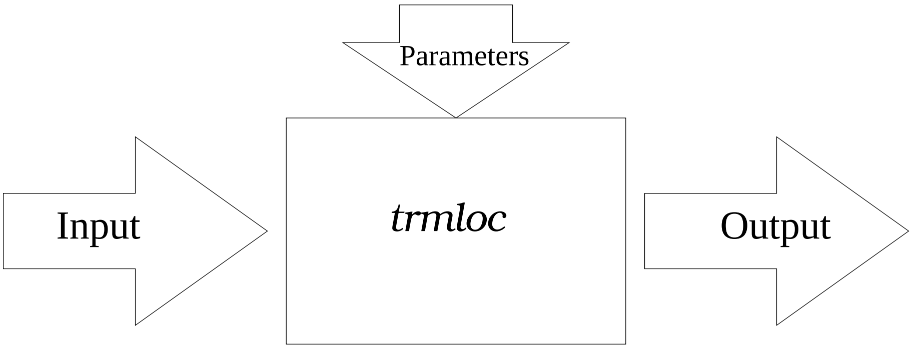


Figure 7.1: Data processing schemata used by *trmloc*.

The inputp data comprise a velocity model and time onset readings read from two different files. The output is a set of ASCII files and eventuall messages. Messages are by default send to the the standard output STD_OUT (under Unix/Linux) while error messages are send to standard error STD_ERR channel and can easily be redirected in a standard way. The parameters controlling action of TRMLOC are read either from standard input STD_INP or from an external file. In the last case, recommended for the *IS-EPOS* platform calling *trmloc* should looks like:

$$\textit{trmloc} \quad -P : file\_name$$

where $file\_name$ is the name of the file with controlling parameters and **no space** between “-P:” and *file_name is allowed.*

Chapter 8

Setting the parameters

The *trmloc* subroutine requires some external parameters to control its action, to define input/output files, etc. To make the task of setting the parameters as flexible as possible, we follow the idea of parameter parsing developed for the TOM-3D software package (Debski, 2002).

The parameters may be provided generally in two different ways. First of all they may be specified on-line as arguments following the called subroutine:

$$\textit{trmloc} \quad arg_1 \quad arg_2 \cdots$$

Secondly, they may be read in from an external file. In this case the parameters are sought in a user specified file whose name is provided as the on-line argument. Finally, it is possible to mix both approaches and specify some parameters as on-line arguments and leaving the definition of the remaining ones to the external file. For the purpose of the *IS-EPOS* platform it is suggested to declare all necessary parameters throught an external file prepared and verified through an external platform-dependent software. Thus, if the parameters are defined in the $parameter\_file$ file then the *trmloc* should be called in the following way

$$\textit{trmloc} \; -\mathrm{P} \; :parameter\_file$$

.

The on-line arguments and the contents of the read files, form an input list which is scanned for the parameters recognized by the TRMLOC subroutine. The parameters are identified by keywords which are several-character names (often being an abbreviation of corresponding parameter names) which either begin with the minus (**-**) sign or end with the equal sign (**=**). The keywords may be generally classified into three categories depending on the logical meaning of the parameter they represent.

The first category consists of the keywords that have the form **-c** where **c** is a single small alphanumeric character. These keywords identify parameters (called switches) that get no values but are used to enable/disable some actions like disable writing progress reports to the console (**-q**), displaying the default file names (**-f**), listing the available keywords (**-h**), to name a few of the most common ones. The attached parameters are set on/off depending **on the presence** of the corresponding keyword in the input parameter list. Thus, for example calling TRMLOC with the (**-h**) on-line argument will cause that all other arguments are disregarded and the subroutine will only list all recognized keywords.

The second category consists of keywords which are supposed to identify numerical values to be assigned to the attached parameters. They have the form keyword=*value* where keyword is a nickname of the keyword (usually the abbreviated parameter name for easier identification) and *value* is the value which will be assigned to the corresponding parameter. No blank space is allowed either before or after the assignment (=) mark. Using such keywords without an assigned value will usually terminate the action of the subroutine with an error message.

The next category consists of those keywords whose goal is to provide names of input and output data files. They have a generic form -C:*name* where C is a capital letter and *name* is a single word (string of characters). The *name* field is sometimes optional and may be omitted, which results in the use of predefined, default file names. However, we do not recomend using this mechanism on *IS-EPOS* platform as potantialy prone to errors.

The precedence of parsing of the input parameter list for a given keyword is as follows. First the on-line arguments are parsed. Then the file specified by the user with -P:*name* is scanned. When a given keyword is encountered then an associated (if any) numerical value, a file name or a list of numbers is read, and no further search for an input for the same keyword is performed. Thus, if a given keyword appears, for example, twice as an on-line argument and in an input file, only the value specified by the keyword encountered first (here, the on-line argument) will be taken into account.

The keywords, no matter whether used on-line or in a parameter file may appear in an arbitrary order. Any other text may also be added to the input parameter file as a comment and will be ignored provided it includes no keyword pattern as a separate word or a substring. Since all the keywords include either ”**-**” or ”**=**” characters it is advisable not to use these characters in comments. The parsing mechanism used allows to include almost arbitrary comments into the file for documentation purposes or to explain the meaning of the parameter setting. Currently the total size of the parameter file plus (if used) on-line options is limited to 4kB.

Incorrect setting of input parameters may not be easy to detect. If a setting apparently breaks some implicit rules, for example, a negative value is provided for a parameter that may only be positive, *trmloc* terminates with an appropriate error message. However, no exhaustive and complete check of the consistency of the input data specified through the keyword assignment is possible. Thus, it is always advisable to check the job output log file to verify how the parameters were parsed and set.

Table 8.2 provides the complete list of all keywords recognized by the *trmloc* subroutine. In this list the we use the keyword syntax convention summarized in table 8.1.

Table 8.1: The convention used to describe the keywords syntax. **-X** stands for any keyword that is used for setting a string/numerical parameter `param`.

Column: Requested	
I	`[Inactive]` Active only when set. If omitted no action is undertaken
O	`[Optional]` Active even if not set - default value is then used
R	`[Required]` Must be set
S	`[Select]` One of the possibilities must be selected
Column: Value	
string	a string of characters with no blank spaces (a single word)
real	a real number
list	a list of real numbers enclosed between **” ”**
integer	an integer number
g/l/· · ·	a list of possible choices
(a· · · b)	allowed range. Missing **b** means ∞
Column: Description	
`STDERR`	Unix standard error device (usually console)
`STDOUT`	Unix standard output device (usually console)
[·]	Default value

8.1 List of parameters

8.1.1 Parameters controling file names

The first group of keywords (**-C:** . . . **-X:**) is used to define name of files with input and output data. Format of these files is described in details in the next section. The requsted file names has to form a single computer word consisnting of a string of characters with no blank characters (like space, tabular, new line character, etc). Using characters with codes above 254 (e.g. polish diacrytic characters) is allowed, but not recommended as can potentially create problems with opening/accessing files, depending on setting operating system under which *trmloc* is currently run. The file names can consists of up to 127 characters. Most of file names is initilized to default names (see **-f**) and thus the *fname in the setting **-C:**fnamecan be omitted, However we do not recomend using this feature during rutine runs of* trmloc *since it leads to loosing information about used input data at the post-calculation stage.

Table 8.2: List of parameters used by trmloc.

Keyword	Req.	Value	Description
-C:fname	I,O	string	Cp listfile (active only with --e)
-D:fname	R	string	input data file
-V:fname	R	string	velocity model
-P:fname	R	string	input parameter file
-L:fname	O	string	output log file
-O:fname	O	string	output pdf
-R:fname	O	string	output residua
-X:fname	I,O	string	Cp curves (active only with --e)
-a	O		authors info
-e	O		extended mode
-f	O		prints default file names
-h	O		prints parameters/option syntax
-m	O		enables marginal pdf calculations
-o	O		prints set values of parameters to STDOUT
-q	O		silent mode: no messages to STDERR
Fvi=	-	0	Input velocity file type (0-TNF)
Fdi=	S	0/1	Input data type (ASCII 5/6 column)
Fpo=	O,S	0...4	Output PDF format (1D,2D,3D, all marginal PDF
Cme=	R		a priori variance for X/Y coordinates
Cmz=	R		a priori variance for Z (depth)
Ia=	R,S	g/l/c/m	norm used for prior distribution
Xap=	R	real	A priori X location
Yap=	R	real	A priori Y location
Zap=	R	real	A priori Z location
Ip=	R,S	g/l/c/m	A posteriori pdf norm [g/l/m/c]
Cp=	R	real	Likelihood variance
Fi=	O	0/1/2	Fisher Norm derivative schemata
Fh=	O	real	Fisher Norm FD-derivative step
Es=	O	string	Event description
Cpm=	O	real	Minimum Cp value (active only with --e)
CpM=	O	integer	Maximum Cp value (active only with --e)
Cpi=	O	integer	Cp increment ratio (active only with --e)
Cpf=	O	0/1	Cp increment mode 0-arithmetic, 1-geometric
Ftm=	O	real	TT initialization
Fit=	O	integer	No. of G-S itterations
Fsc=	O	real	Gradient scaling
Fec=	O	real	Converence criterion

8.1.2 Switching parameters

*The switching parameters (**-a:** ... **-q:**) hove double meaning. Some of them are for information purpose only and their use results in printing appropriate messages to standard output and exiting. To this group belong the following parameters*

- ***-a:** - prints author and copyright information*
- ***-f:** - prints info about file names and their default values*
- ***-h:** - prints a list of all recognized keywords*
- ***-o:** - prints a list of all sett parameters*

*Among these parameters the most often used is **-h:**, which provides a kind of "online help" and allows a quick checking of syntax of kewords. On the other hand **-o:** is very usefull for a quick verification of values of set parameters and **-f:** lists the types and suggested and acctual assigned default file names. Prowided information describes file types (R - for reading, W - for writing) and if given file name can be automatically initialized (+) or must be exlipcit provided (-).*

The second group of switching parameters is used to control the scope of undertaken calculations as follows:

- ***-e:** - Enables so called extended mode. In this mode after completing calculations for given setup of input parameters* trmloc*performs additional relocations with different settings of the **Cp** parameters. The range of examinated **Cp** is either read-in from an input file (**-C**) or generated by* trmloc *according to settings of (**Cpm=**,... **Cpf=**). Selected characteristics of relocation results are saved in the **-X** file.*
- ***-m:** - Enables calculation of characteristics of all marginal distributions. Since it takes additional computation time can be avoid if no advanced error analysis is required.*

*Finally, the last switching parameter (**-q**) controles showing information about calculation progress to console. If set,* trmloc *is running in the "silent" mode with no information (except errors) about calculation progress at all, what slightly speeds up calculations. It is recomended to use it only during a rutine data processing.*

8.1.3 Numerical parameters

This group of parameters provides necessary information for running trmloc *and consists of subgroups defining the* a priori *setttings, controlling the forward modelling and inversion process, and finally indetification of input/output data formats. Their meening is following.*

*First, the keywords **Fvi=**, **Fdi=**, and **Fpo=** are used to identify formats (types) of input/output data files. In the current releas only a few types are fully supported, as described in the next* *chapter. For example for the velocity model data* ***Fvi*** *is acually inactive and set* ***Fvi=0*** *since only the native for the TOM-3D package formatTNF is currently fully supported.*

Another parameter ***Fdi****, defines a type of data file format. Currently the only ASCII column-orientated format is supported (see next chapter). In the current version* ***Fdi*** *can takes values 0 and 1 only.*

The ***Fpo=*** *parameter controls which* a posteriori *PDF distributions are stored in the output (****-O****) file. It can the following values:*

- *0 - (default) 1D and 2D marginals are stored. In most cases this is the most versatile output format.*
- *1 - only 1D marginals are stored what results in a small files but with limited information contents*
- *2 - 2D marginals are stored. This is the appropriate format if epicenter errors are to be analysed (like classical "error eplipse")*
- *3 - the full 3D* a posteriori *pdf is stored what usually results in huge file size.*
- *4 - all marginals and* a posteriori *pdf are stored. Again a size of file is usually huge.*

Parameters controlling setting of the a priori *solution are* ***Cme=****,****Cmz=****,* ***Ia=****,* ***Xap=****,* ***Yap=****, and* ***Zap=*** *and their meaning is following*

- ***Cme*** *- postulated uncertainty for the horizontal coordinates X and Y of the* a priori *model,*
- ***Cmz*** *- postulated uncertainty for the depth coordinate Z of the* a priori *model,*
- ***Ia*** *- norm used to generate the* a priori *pdf function. Recognized values are:* ***g*** *-* l_2 *norm (Gaussian likelihood) - appropriate for a high confidence* a priori *location,* ***l*** *-* l_1 *norm. More robust norm enable handling less accurate* a priori *locations.* ***c*** *-* l_c *Cauchy norm, and* ***m*** *-* l_m *modified Cauchy norm. The most robust norms which can be used to handle vogue* a priori *locations,* ***n*** *- no* a priori *solution at all.*
- ***Xap=****,* ***Yap=****, and* ***Zap=*** *- hypocenter coordinates of the* a priori *location.*

The inversion process is directly controlled by two parameters, namely ***Ip*** *and* ***Cp****. The* ***Ip*** *parameter defines the norm used to generate likelihood function. Recognized values are:*

- ***g*** *-* l_2 *norm (Gaussian likelihood) - appropriate for high quality data and accurate forward modelling (fine grid),*
- ***l*** *-* l_1 *norm. More robust norm assuring a more reliable solution if outliers are present in data or corse modelling grid is used,*
- ***c*** *-* l_c *Cauchy norm,*

- ***m*** - l_m *modified Cauchy norm. Both**c** and **m** are the most robust norms which can be used to invert low quality data.They use often lead to an overestimation of inversion errors,*

The trmloc *allows to calculate the Fisher matrix which can be used in advanced analysis of inversion uncertainties. Two parameters, namely **Fi** and **Fh** controls this part of calculations. **Fi** can takes value 0,1,2 which corresponds to skipping calculation (0), using first order (3-points) central finite difference schemata (**Fi=1**) or 5-points respectivally (**Fi=2**). The second parameter **Fh** defines a finite difference step.*

Finaly, trmloc *provides a mechanism of inversion/location annotation by adding an arbitrary description of given run by a single word defined by the **Es** keyword* **Es=description**.

In the extended mode trmloc *performs relocations of event at hand for various settings of Cp. The list of used Cp values can either be read-in from an external file (**-C**) or defined by providing minimum (**Cpm** parameter) and maximum (**CpM** parameter) values of used Cp together with an increment step defined by **Cpi**. The generated list of Cp can be either algebraic serious or geometric one defined by **Cpf=0/1** respectivally.*

*Finally, there is a group of parameters which allow to controle the forward modelling (FSM) procedure. Those parameters are **Ftm**, **Fit**, **Fsc**, and **Fec**. Thy control initilalization (Ftm), maximum number of Gauss-Lobbato iterations (Fit - default is 20), gradient scaling (Fsc, default no scaling), and convergion accuracy (Fec - default is 0.01). The meaning of the last parameter is that if RMS differences between two subsquent full sweeps solutions differe by less than Fec the itteration procedure of FSM terminates. The default values are robust enough to handle most of practical 3D velocity models so those parameters typically do not need to be changed.*

Chapter 9

File formats

9.1 Input files

TRMLOC requires following input data:

- *control parameters*
- *observational travel time data,*
- *station coordinates,*
- *velocity model,*
- *computational grid specifications*

.

*They are provided in three different input files: input velocity model (**-V**), data file (**-D**) and parameters file (**-P**). All input/output files used by the current version of TRMLOC procedure are ASCII files. Their formats are described in the following subsections.*

9.1.1 Controll parameters file (-P)

*This is a free-style ASCII file which contains settings of all control parameters in form **Key=**value, **-key**, **-Key:**file_name separated by spaces, empty lines or any comments (strings). The only restriction imposed on the format of the file is that the maximum number of read characters (size of file in bytes) have to be less then 4040. In addition the length of each file name is restricted to 127 characters. A care must be taken to avoid using keywords in comments. If it hppens, TRMLOC parser can make a wrong value assigment to given parameters. **It is recomended always to verify settings of parameters by inspecting the output log file**. An example of the input parameter file is shown in Fig. 10.1.*

By default TRMLOC expects that all dimensional parameters and input data are express in SI system (meters, seconds, etc.) Thus, the arrival times defined in data file are in seconds,

Table 9.1: Description of the input data file format.

Column	Format	Description
1	integer	station/sensor Id number
2	real	station X coordinate
3	real	station Y coordinate
4	real	station Z coordinate
5	real	wave arrival time

*station coordinates, are in meters with respect to a local coordinate system and used velocity is expressed in meters per second. Similarly, **Xap, Yap, Zap** and **Cm** and **Cz** are also in meters and **Cp** in seconds. Change to other length/time units (for example, centimeters, kilometers, miliseconds, etc.) is possible but all above parameters must be adjusted accordingly*

9.1.2 Input data file (-D)

*The input data file provides the observational travel time data and station location information This ASCII file consists of a number of lines composed of 5 (**Fdi=0**) or 6 (**Fdi=1**) columns: one integer (first column) and 4 (5) real numbers. Each line corresponds to an arrival time of a seismic phase (the same for all entries) recorded by a given station. The meaning of each of the columns is explained in table 9.1. The selection between 5 and 6 columns format is done on the base of **Fdi** keyword. The optional sixth column allows to assign a different data variance to each of the stations/sensors separately which offers the possibility of an independent evaluation of data quality for each station. The floating point precision used to describe station locations, arrival times and data variances, if used, is arbitrary.*

In current implementations trmloc *reads the entire data file and evaluates the number of data entries by a number of read lines. It **DOES NOT PERFORM** any data verification for data coherence, data duplications, etc. If given line consists of less then 5 (6) entries TRM-LOCreports error and exits. An example of the input data file for the default setting **Fdi=0** is shown in Fig. 10.2.*

9.1.3 Velocity data file (-V)

This file has by now the most complex structure in order to accomodate in the simplest way all practically important velocity models. In the current release only the native TNFdata format is supported which correspond to setting by default Fvi=0. [1]

In the TNF format the file consists of a header and a data block as shawn in Fig. 9.1. The header has a fixed format and consists of 9 lines terminated by end-of-line (
n) character and begining with # character in the first column. In all header lines after the #

[1] The key **Fvi=** is inactive in the current release.

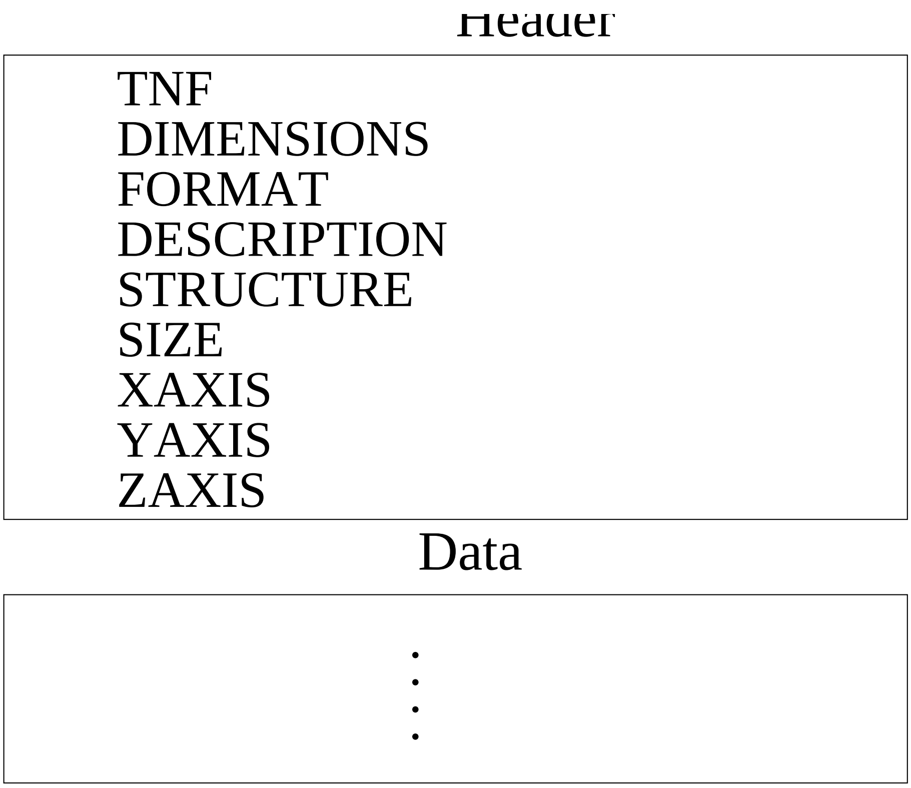


Figure 9.1: General structure of the TNF velocity model format.

mark the identification string appears and is followed by user set appropriate numerical values. The meaning of the identification strings listed in order of their apearence in the header and expected associated values are shown in table 9.2.

Most of parameters defined in the header part of the file are self-explanatory. The comment is only required for the ***STRUCTURE*** *parameter. This parameter controls type of used grid. If the set value is 0 the structred (cubic) grid is assumed with grid node distances dx, dy dz. Setting the value 1 assumes potentially arbitrary grid nodes distribution. This type of unstructered grids (time scales) are not fully supported yet.*

The heder is followed by the data block. Structure of the the data block depends on values of ***DIMENSION*** *and* ***STRUCTURE*** *defined in the header.*

- ***DIMENSION = 0***

  This is a case of constant velocity model. The block data consists of a single real value $\boxed{V}$ *which is assigned to each grid node.*

- ***DIMENSION = 1***

  This is a case of 1D velocity model. The model consists of Nz-1 homogeneous horizontal layers. The block data contains now Nz real values $\boxed{V_1\ V_2 \cdots V_{Nz-1}}$ *in arbitrary arrangement (line, column, mixed).*

- ***DIMENSION = 2***

Line	String	Description	Values	Format
1	**TNF**	TNF format	1.0	fixed
2	**DIMENSION**	dimension of the model	0/1/2/3	integer
3	**FORMAT**	type of Block DATA	ASCII	fixed
3	**DESCRIPTION**	model description		up to 255 characters
5	**STRUCTURE**	type of grid	0/1	integer
6	**SIZE**	No of nodes	Nx Ny Nz Ng	4 × integer(long)
7	**XAXIS**	X-axis description	xmin xmax dx	3 × real
8	**YAXIS**	Y-axis description	ymin ymax dy	3 × real
9	**ZAXIS**	Z-axis description	zmin zmax dz	3 × real

Table 9.2: .

This is a case of 2D velocity model. In this class of models the vellocity is assumed to change along x and z coordinates (one horizontal and depth) dimensions and remains constant along y. All y=const sections are identical. The block data consists now of Nx × Nz *real values* $V_{ix,iz}$ *asigned to nodes (ix, iy, iy=any) arrange in the order of increasing Nx and Nz :* $\boxed{V_{0,0}\ V_{1,0}\ldots V_{Nx-1,0}\ V_{0,1}\ldots V_{Nx-1,1}\ V_{0,2}\ldots V_{Nx-1,Nz-1}}$ *The data arrangement (line, column, mixed) is arbitrary.*

- ***DIMENSION = 3***

  This is a case of the full 3D velocity model. The block data consists now of Nx × Ny× *real values* $V_{ix,iz}$ *asigned to nodes (ix, iy, iy) arrange in the order of increasing Nx, Ny, and Nz :as shown in Fig. 9.2. The data arrangement (line, column, mixed) is arbitrary.*

$$
\begin{array}{llll}
V_{0,0,0} & V_{1,0,0} & \cdots & V_{Nx-1,0,0} \\
V_{0,1,0} & V_{1,1,0} & \cdots & V_{Nx-1,1,0} \\
\vdots & & & \\
V_{0,Ny-1,0} & V_{1,Ny-1,0} & \cdots & V_{Nx-1,Ny,0} \\
V_{0,0,1} & V_{1,0,1} & \cdots & V_{Nx-1,0,1} \\
V_{0,1,1} & V_{1,1,1} & \cdots & V_{Nx-1,1,1} \\
\vdots & & & \\
V_{0,Ny-1,1} & V_{1,Ny-1,1} & \cdots & V_{Nx-1,Ny-1,1} \\
\vdots & & & \\
V_{0,Ny-1,Nz-1} & V_{1,Ny-1,Nz-1} & \cdots & V_{Nx-1,Ny-1,Nz-1}
\end{array}
$$

Figure 9.2: Arrangment of values in the block data for the 3D velocity model.

- ***STRUCTURE = 1***

  In this case of the full 3D velocity model is assumed and data block consists of lines with grid node coordinates and assigned velocity values. This format is shown in Fig. 9.3.

In the current version of TRMLOC only the regular grids are supported, thus this format reduces efficiently to the 3D model (DIMENSION = 3) but with an explicite given coordinates of each grid node.

$$\begin{array}{llll}
x_0 & y_0 & z_0 & V \\
x_1 & y_0 & z_0 & V \\
\vdots & & & \\
x_{Nx-1} & y_0 & z_0 & V \\
x_0 & y_1 & z_0 & V \\
\vdots & & & \\
x_{Nx-1} & y_{Ny-1} & z_0 & V \\
x_0 & y_0 & z_1 & V \\
\vdots & & & \\
x_{Nx-1} & y_{Ny-1} & z_{Nz-1} & V
\end{array}$$

Figure 9.3: Arrangment of values in the block data for the 3D velocity model defined over an unstructured grid (STRUCTURE = 1)

If number of data read-out from the data-block does not fit the number of grid nodes define in the header (SIZE line) TRMLOC reports error and exits. Also, if any read-out velocity is non-positive all error is reported and program exits.

9.1.4 Cp list file (-C)

*If TRMLOCis run in the exended mode (**-e** is set) a range of Cp values for which* a posteriori *corresponding solutions are calculated can be read out from a file identified by the **`-C:fname`** keyword. This file contains in the first line a number of used (to be read-out) Cp followed by given Cp in subsquent lines. An example of such file is shawn in Fig. 10.3.*

9.2 Output files

9.2.1 Output log file (-L)

The output log file is a formatted ASCII file which provides the exhaustive information on given run of TRMLOC including numerical values of obtained point-like solutions like maximum likelihood solution, average solution, and error estimates. It also reports set values of all significant parameters controling given calculationa. It consists of a "heder" followed by 14 named sections as it is shown in fig. 10.4. The begining part of the file consists of 5 lines "header" providing information about the TRMLOC version, date of run, computational time (CPU line) used memory (does not work on all platforms) and an identification of processed event (EVENT *line) define by **Es=** keyword. The following sections are line-formated with an unique identification string commencing with # character at the beginig of the line. Important numerical values including obtained solutions appear mostly as the last entry in a given line and thus are easily retrivable by external software including standard unix shell tools.*

Some information provided in the LOG file are self-explanatory like, for example, Data *entry at the begining of the file, names of different segments, file names, etc. The meaning of other important parameters is explained in table 9.3 where parameters are listed in order they apper in the file. The parameters listed in the LOG file which are not implemented yet in the current version of TRMLOCȧre skipped.*

We also skip description of two sections of the LOG file, namely IO FORMATS *(which describes features not fully implemented yet) and* PARALELIZATIONS *which is purelly informative.*

Table 9.3: Selected parameters reported in the LOG file.

Line	*Format*
FSM SETTING	
#H1	*type (2D or 3D) used eikonal solver*
#H2	*initialization value for eikonal solver (should be large positive value)*
#H3	*maximum number of iterations*
#H4	*scaling (internal option for debuging only)*
#H5	*accuracy of itteration procedure*
DATA	
#D0	*Number of data (stations)*
#D1 - #D3	*span of station network coordinates*
#D-0,#D-1 ...	*slongtabletations ID_number and coordinates*
VELOCITY MODEL	
#V0	*Model description (read-out from the **-V** file*
#V1 -#V4	*simple statistics of the model*
FSM TRM MODELLING	
#M-0 -#M-1, ...	*description of the convergense of FSM solver for each data (station) modelling*
INVERSION SETUP	
#I1	*used Cme*
#I2	*used Cmz*
#I3	*used Cp* *In case of extended mode the values of calculations for different Cp are reported only in the outpu **-X** file. The solutions reported in the log file as well as provided by the residua **-R** and output pdf file **-O** refers to this value of Cp.*
#I4	*used norm for* a priori *pdf*
#I5	*used norm for* a priori *pdf*
#I6 -#I8	a priori *location*
#I9	*order of FD approximation for Fisher matrix calculation*

Table 9.3: Selected parameters reported in the LOG file.

#Ia	step for FD Fisher matrix calculation
EXTENDED MODE	
#X1 - #2	minimu and maximum values of Cp
#X3	constant ratio/distnce used to generate Cp list as geometric/arithmetic series
#X4	geometriclist (1), arithmetic (0)o type of Cp list
#X5	length of the Cp list
INVERSION MEASURES	
#O1	log of evidence
#O2	a posteriori entropy
#O3	Fisher measure
#O4	Information about Fisher measure calculations
POINT SOLUTIONS	
#C1	minimum of the missfit function
#C8 - #Cd	maximum likelihood and average solutions
#Ce - #Cf	event origint time (with respect to time onsets of data)
COVARIANCE	
#E1 - #E6	elements of the a posteriori covariance matrix
#E7	variance of origin times for MLL solution
#E8	variance of origin times for AVR solution
MARGINAL SOLUTIONS	
#S1 - #Sc	Hypocenter coordinates estimators for MLL and AVR solutions calculated from different marginal PDF distributions. Disp stands for an inversion error for given coordinate and Rml_rms and Rml_avr are root mean squares residua for MLL and AVR solutions.

9.2.2 Output *a posteriori* pdf file (-O)

The file containing the output a posteriori probability density distributions has a context-dependent format depending on the **Fpo** parameter. It consists of segments corresponding to one dimensional (1D) , two dimensional (2D) marginal distributions and the full three dimensional (3D) a posteriori distribution each with its own format. Each segment is uniquelly describe by an identification string corresponding to a given a posteriori distribution. They are listed in table 9.4. Depending on setting **Fpo** parameter only 1D (Fpo=1), 2D (Fpo=2), 3D (Fpo=3) or all (Fpo=4) segments appear in the output file. The format for which segment is shown in table 9.5 All but first columns (segment identifications) are real numbers of arbitrary floating point precision.

An example of fragments of the a posteriori pdf output file for setting Fpo=4 Fig. 10.5.

Table 9.4: Abreviations of *a posteriori* pdf names used to identify different segments in the output (**-O**) file

Abbreviation	Dimension	Description
1D-X	1	$\sigma_X(x)$
1D-Y	1	$\sigma_Y(y)$
1D-Z	1	$\sigma_Z(z)$
2D-XY	2	$\sigma_{XY}(x,y)$
2D-XZ	2	$\sigma_{XZ}(x,z)$
2D-YZ	2	$\sigma_{YZ}(y,z)$
3D-XYZ	3	$\sigma(x,y,z)$

Table 9.5: Format of the PDF segments in the output pdf (**-O**) file.

Segment ID	Columns			
	2	3	4	5
1D-X	x	$\sigma_x(x)$		
1D-Y	y	$\sigma_y(y)$		
1D-Z	z	$\sigma_z(z)$		
2D-XY	x	y	$\sigma_{xy}(x,y)$	
2D-XZ	x	z	$\sigma_{xz}(x,z)$	
2D-YZ	y	z	$\sigma_{yz}(y,z)$	
3D-XYZ	x	y	x	$\sigma(x,y,z)$

9.2.3 Output residua file (-R)

The last output file generated by TRMLOC is the file containing residua data for the maximum likelihood and average solutions for the full and (if ***-m*** *is set) all marginal* a posteriori *distributions. The file begin with a single header line marked by the # followed by segments corresponding to solutions for the full (XYZ), 1D marginals (X, Y, Z) and 2D marginal (XY, XZ, YZ) pdf distributions. The meaning of each of the columns in the file is explained in table 9.6. Finally, an example of the* a posteriori *residua file is shawn in Fig. 10.6.*

9.2.4 Extended mode output file (-X)

*If TRMLOC is run in the so called extended mode (****-e*** *is set) then the additional output file identified by* ***-X*** *keyword is creating. It contains selected characteristics of* a posteriori *solutions calculated for different Cp parameter. The file begins with two header lines marked by the # character at the begining of each line with a simpliefied description following dataorganized in columns. The meaning of each data column in the file is explained in table 9.7 and an example*

Table 9.6: Description of the output residua file format.

Column	Format	Description
1	string	marginal pdf identification
2	integer	station ID number
3	real	measured time onset (data)
4	real	time onset calculated for MLL solution
5	real	time onset calculated for AVR solution
6	real	propagation time from the MLL locationn
7	real	propagation time from the AVR location
8	real	RMS residua for MLL solution
9	real	RMS residua for AVR solution
10	real	data variance

of such a file is shawn in Fig. 10.7.

Table 9.7: Extended mode output file format. The file begins with two header lines marked by the # character at the begining of each line followed by lines consisting of 13 columns.

Column	Format	Description
1	real	Cp
2	real	Entropy
3	real	Evidence
4	real	Fisher info.
5	real	RMS residua
6	real	X MLL solution
7	real	Y MLL solution
8	real	Z MLL solutio
9	real	Origin time
10	real	Ex errors
11	real	Ey errors
12	real	Ez errors

Chapter 10

Examples

10.1 File formats

In this section examples of formats of input/output files used by TRMLOC are gathered.

```
Event description: Es=9992

Files:
Log file         -L:LOG/R-9992_h10.log
Output PDF       -O:PDF/R-9992_h10.apo
Parameter file   -P:trmloc.rcp
Data             -D:TMP/D9992.dat
Velocity model   -V:TMP/V9992.vel
Residua          -R:RES/R-9992_h10.res

Formats:
 Fdi=0    (the same data quality)
 Err=0    error message format
 Fpo=1    (save only 1D PDF)
 Fvi=0    (dafult TNF native format - V=const)

Calculate all marginals:
  -m
  -q   (no messages to std_out)

 Cm=2000  a priori (horizontal uncertainties)
 Cz=500   a priori (vertical uncertainties)
 Cp=0.02  data variance

Norms:
 Ia=g
 Ip=g

A_priori solution:
  Xap=12963
  Yap=7108
  Zap=350
```

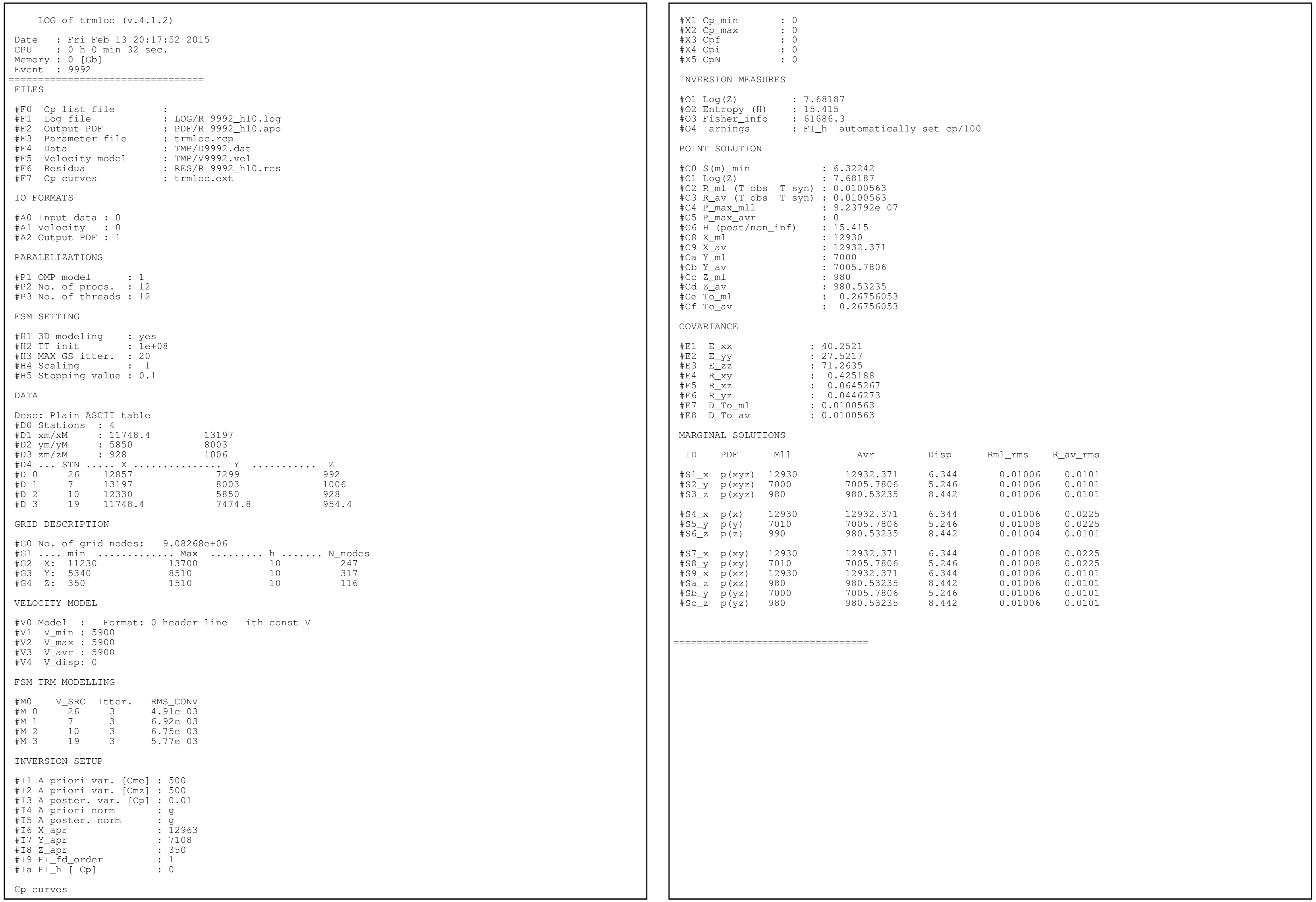
Figure 10.1: An example of the TRMLOC control parameters file (see text for description).

```
26   12857      7299      992       0.05273
10   12330      5850      928       0.17673
 7   13197      8003      1006      0.19473
19   11748.4    7474.8    954.4     0.23073
```

Figure 10.2: An example of the TRMLOC input data file (see text for description).

```
8
 0.2
 0.1
 0.05
 0.02
 0.01
 0.005
 0.002
 0.001
```

Figure 10.3: An example of the input “CPO” file defining the values of Cp for which TRMLOC calculates solutions when runing in the extended mode.



```
    LOG of trmloc (v.4.1.2)

 Date   : Fri Feb 13 20:17:52 2015
 CPU    : 0 h 0 min 32 sec.
 Memory : 0 [Gb]
 Event  : 9992
==================================
 FILES

 #F0  Cp list file        :
 #F1  Log file            : LOG/R 9992_h10.log
 #F2  Output PDF          : PDF/R 9992_h10.apo
 #F3  Parameter file      : trmloc.rcp
 #F4  Data                : TMP/D9992.dat
 #F5  Velocity model      : TMP/V9992.vel
 #F6  Residua             : RES/R 9992_h10.res
 #F7  Cp curves           : trmloc.ext

 IO FORMATS

 #A0 Input data : 0
 #A1 Velocity   : 0
 #A2 Output PDF : 1

 PARALELIZATIONS

 #P1 OMP model      : 1
 #P2 No. of procs.  : 12
 #P3 No. of threads : 12

 FSM SETTING

 #H1 3D modeling    : yes
 #H2 TT init        : 1e+08
 #H3 MAX GS itter.  : 20
 #H4 Scaling        :  1
 #H5 Stopping value : 0.1

 DATA

 Desc: Plain ASCII table
 #D0 Stations  : 4
 #D1 xm/xM     : 11748.4         13197
 #D2 ym/yM     : 5850            8003
 #D3 zm/zM     : 928             1006
 #D4 ... STN ..... X ..............  Y  ..........  Z
 #D 0     26    12857              7299              992
 #D 1     7     13197              8003              1006
 #D 2     10    12330              5850              928
 #D 3     19    11748.4            7474.8            954.4

 GRID DESCRIPTION

 #G0 No. of grid nodes:   9.08268e+06
 #G1 .... min  ............ Max  ......... h ....... N_nodes
 #G2  X:  11230            13700            10          247
 #G3  Y:  5340             8510             10          317
 #G4  Z:  350              1510             10          116

 VELOCITY MODEL

 #V0 Model  :   Format: 0 header line   ith const V
 #V1  V_min : 5900
 #V2  V_max : 5900
 #V3  V_avr : 5900
 #V4  V_disp: 0

 FSM TRM MODELLING

 #M0    V_SRC  Itter.   RMS_CONV
 #M 0     26     3      4.91e 03
 #M 1     7      3      6.92e 03
 #M 2     10     3      6.75e 03
 #M 3     19     3      5.77e 03

 INVERSION SETUP

 #I1 A priori var. [Cme] : 500
 #I2 A priori var. [Cmz] : 500
 #I3 A poster. var. [Cp] : 0.01
 #I4 A priori norm       : g
 #I5 A poster. norm      : g
 #I6 X_apr               : 12963
 #I7 Y_apr               : 7108
 #I8 Z_apr               : 350
 #I9 FI_fd_order         : 1
 #Ia FI_h [ Cp]          : 0

 Cp curves
```

```
 #X1 Cp_min       : 0
 #X2 Cp_max       : 0
 #X3 Cpf          : 0
 #X4 Cpi          : 0
 #X5 CpN          : 0

 INVERSION MEASURES

 #O1 Log(Z)         : 7.68187
 #O2 Entropy (H)    : 15.415
 #O3 Fisher_info    : 61686.3
 #O4  arnings       : FI_h  automatically set cp/100

 POINT SOLUTION

 #C0 S(m)_min            : 6.32242
 #C1 Log(Z)              : 7.68187
 #C2 R_ml (T obs  T syn) : 0.0100563
 #C3 R_av (T obs  T syn) : 0.0100563
 #C4 P_max_mll           : 9.23792e 07
 #C5 P_max_avr           : 0
 #C6 H (post/non_inf)    : 15.415
 #C8 X_ml                : 12930
 #C9 X_av                : 12932.371
 #Ca Y_ml                : 7000
 #Cb Y_av                : 7005.7806
 #Cc Z_ml                : 980
 #Cd Z_av                : 980.53235
 #Ce To_ml               :  0.26756053
 #Cf To_av               :  0.26756053

 COVARIANCE

 #E1  E_xx             : 40.2521
 #E2  E_yy             : 27.5217
 #E3  E_zz             : 71.2635
 #E4  R_xy             :  0.425188
 #E5  R_xz             :  0.0645267
 #E6  R_yz             :  0.0446273
 #E7  D_To_ml          : 0.0100563
 #E8  D_To_av          : 0.0100563

 MARGINAL SOLUTIONS

  ID    PDF      Mll           Avr         Disp      Rml_rms    R_av_rms

 #S1_x  p(xyz)  12930        12932.371     6.344       0.01006    0.0101
 #S2_y  p(xyz)  7000         7005.7806     5.246       0.01006    0.0101
 #S3_z  p(xyz)  980          980.53235     8.442       0.01006    0.0101

 #S4_x  p(x)    12930        12932.371     6.344       0.01006    0.0225
 #S5_y  p(y)    7010         7005.7806     5.246       0.01008    0.0225
 #S6_z  p(z)    990          980.53235     8.442       0.01004    0.0101

 #S7_x  p(xy)   12930        12932.371     6.344       0.01008    0.0225
 #S8_y  p(xy)   7010         7005.7806     5.246       0.01008    0.0225
 #S9_x  p(xz)   12930        12932.371     6.344       0.01006    0.0101
 #Sa_z  p(xz)   980          980.53235     8.442       0.01006    0.0101
 #Sb_y  p(yz)   7000         7005.7806     5.246       0.01006    0.0101
 #Sc_z  p(yz)   980          980.53235     8.442       0.01006    0.0101



==================================
```

Figure 10.4: An example of the TRMLOC log file

```
1D-X    12450      0
1D-X    12460      0
1D-X    12500      6.89336e-313
1D-X    12510      2.66177e-307
      .
      .
      .
1D-Y    5730       1.43752e-306
1D-Y    5740       1.96329e-300
1D-Y    5750       2.48219e-294
      .
      .
      .
1D-Y    8490       0
1D-Y    8500       0
1D-Z    420        0.00315202
1D-Z    430        0.00314822
      .
      .
      .
1D-Z    1520       0.00536521
2D-XY   9260       5840   0
2D-XY   9280       5840   0
2D-XY   9300       5840   0
2D-XY   9320       5840   0
2D-XY   9340       5840   0
      .
      .
      .
2D-XZ   9260       20  0
2D-XZ   9280       20  0
2D-XZ   9300       20  0
      .
      .
      .
2D-YZ   8380       1160   4.5722e-08
2D-YZ   8400       1160   8.38709e-09
      .
      .
      .
2D-YZ   11080      1520   0
3D-XYZ  9260       5840   20       0
3D-XYZ  9280       5840   20       0
3D-XYZ  9300       5840   20       0
       .
       .
       .
```

Figure 10.5: An example of parts of the TRMLOC output PDF file. Each marginal PDF is identify by an appropriate identification string in the first column (see text for description).

```
# PDF STC    Tobs    Tsyn_ml  Tsyn_av    TT_ml   TT_avr    R_ml     R_av   Cd

  XYZ 26     0.182    0.182    0.176    0.177    0.160    0.000    0.006  1
  XYZ 10     0.256    0.258    0.255    0.253    0.240   -0.001    0.000  1
  XYZ 7      0.256    0.255    0.256    0.251    0.241    0.001   -0.000  1
  XYZ 23     0.290    0.290    0.288    0.286    0.272    0.000    0.002  1
  XYZ 43     0.310    0.313    0.315    0.308    0.299   -0.002   -0.004  1
  XYZ 19     0.338    0.340    0.343    0.335    0.327   -0.001   -0.004  1

  X   26     0.182    0.182    0.180    0.177    0.177    0.000    0.002  1
  X   10     0.256    0.258    0.261    0.253    0.258   -0.001   -0.004  1
  X   7      0.256    0.255    0.254    0.251    0.251    0.001    0.002  1
  X   23     0.290    0.290    0.293    0.286    0.291    0.000   -0.003  1
  X   43     0.310    0.313    0.309    0.308    0.307   -0.002    0.001  1
  X   19     0.338    0.340    0.342    0.335    0.339   -0.001   -0.003  1

  Y   26     0.182    0.183    0.180    0.178    0.177   -0.000    0.002  1
  Y   10     0.256    0.257    0.261    0.252    0.258   -0.000   -0.004  1
  Y   7      0.256    0.257    0.254    0.252    0.251   -0.000    0.002  1
  Y   23     0.290    0.289    0.293    0.284    0.291    0.001   -0.003  1
  Y   43     0.310    0.313    0.309    0.309    0.307   -0.002    0.001  1
  Y   19     0.338    0.341    0.342    0.336    0.339   -0.002   -0.003  1

  Z   26     0.182    0.183    0.176    0.180    0.160   -0.000    0.006  1
  Z   10     0.256    0.258    0.255    0.255    0.240   -0.001    0.000  1
  Z   7      0.256    0.255    0.256    0.253    0.241    0.001   -0.000  1
  Z   23     0.290    0.290    0.288    0.287    0.272    0.000    0.002  1
  Z   43     0.310    0.313    0.315    0.310    0.299   -0.002   -0.004  1
  Z   19     0.338    0.340    0.343    0.337    0.327   -0.001   -0.004  1

  XY  26     0.182    0.183    0.180    0.178    0.177   -0.000    0.002  1
  XY  10     0.256    0.257    0.261    0.252    0.258   -0.000   -0.004  1
  XY  7      0.256    0.257    0.254    0.252    0.251   -0.000    0.002  1
  XY  23     0.290    0.289    0.293    0.284    0.291    0.001   -0.003  1
  XY  43     0.310    0.313    0.309    0.309    0.307   -0.002    0.001  1
  XY  19     0.338    0.341    0.342    0.336    0.339   -0.002   -0.003  1

  YZ  26     0.182    0.182    0.176    0.179    0.160    0.000    0.006  1
  YZ  10     0.256    0.258    0.255    0.254    0.240   -0.001    0.000  1
  YZ  7      0.256    0.255    0.256    0.252    0.241    0.001   -0.000  1
  YZ  23     0.290    0.290    0.288    0.287    0.272    0.000    0.002  1
  YZ  43     0.310    0.313    0.315    0.309    0.299   -0.002   -0.004  1
  YZ  19     0.338    0.340    0.343    0.336    0.327   -0.001   -0.004  1

  XZ  26     0.182    0.182    0.176    0.179    0.160    0.000    0.006  1
  XZ  10     0.256    0.258    0.255    0.254    0.240   -0.001    0.000  1
  XZ  7      0.256    0.255    0.256    0.252    0.241    0.001   -0.000  1
  XZ  23     0.290    0.290    0.288    0.287    0.272    0.000    0.002  1
  XZ  43     0.310    0.313    0.315    0.309    0.299   -0.002   -0.004  1
  XZ  19     0.338    0.340    0.343    0.336    0.327   -0.001   -0.004  1
```

Figure 10.6: An example of the TRMLOC output residua file. The identification key in the first column indicates for which (marginal) PDF distribution given residua were calculated (see text for description). The first, header line is identify by the # symbol and segments are separated by empty line.

```
#  Cp          H           Z           F           R           Xml        Yml         Zml        T           Ex         Ey          Ez
#
 1.00e-03   7.30e+00  -4.47e+02   9.06e+06   9.21e-03   1.29e+04   6.79e+03   9.85e+02  -2.10e-02  4.15e+00   2.50e+00   2.67e+00
 5.00e-03   1.61e+01   1.06e+01   4.97e+04   1.47e-01   1.12e+04   5.34e+03   4.25e+02  -2.41e-01  5.18e+01   3.01e+01   1.05e+02
 1.00e-02   2.01e+01   1.85e+01   8.09e+02   1.47e-01   1.12e+04   5.34e+03   4.25e+02  -2.41e-01  2.32e+02   1.51e+02   2.70e+02
 2.00e-02   2.02e+01   1.87e+01   6.24e+02   1.47e-01   1.12e+04   5.34e+03   4.25e+02  -2.41e-01  2.44e+02   1.64e+02   2.72e+02
 4.00e-02   2.05e+01   1.91e+01   3.77e+02   1.47e-01   1.12e+04   5.34e+03   4.25e+02  -2.41e-01  2.66e+02   1.92e+02   2.76e+02
 6.00e-02   2.06e+01   1.92e+01   2.95e+02   1.47e-01   1.12e+04   5.34e+03   4.25e+02  -2.41e-01  2.77e+02   2.08e+02   2.77e+02
 8.00e-02   2.09e+01   1.95e+01   1.82e+02   1.47e-01   1.12e+04   5.34e+03   4.25e+02  -2.41e-01  2.99e+02   2.43e+02   2.79e+02
 1.00e-01   2.10e+01   1.96e+01   1.43e+02   1.47e-01   1.12e+04   5.34e+03   4.25e+02  -2.41e-01  3.10e+02   2.62e+02   2.80e+02
 2.00e-01   2.17e+01   2.03e+01   1.83e+01   1.47e-01   1.12e+04   5.34e+03   4.25e+02  -2.41e-01  4.14e+02   4.32e+02   2.81e+02
 3.00e-01   2.20e+01   2.05e+01   3.10e+00   1.47e-01   1.12e+04   5.34e+03   4.25e+02  -2.41e-01  4.64e+02   5.14e+02   2.81e+02
 4.00e-01   2.21e+01   2.07e+01   6.71e-01   1.47e-01   1.12e+04   5.34e+03   4.25e+02  -2.41e-01  4.87e+02   5.54e+02   2.80e+02
 5.00e-01   2.22e+01   2.07e+01   3.95e-01   1.47e-01   1.12e+04   5.34e+03   4.25e+02  -2.41e-01  4.92e+02   5.63e+02   2.80e+02
```

Figure 10.7: An example of the TRMLOC output CPO file (extended mode) gathering metacharacteristics of the *a posteriori* PDF and basic maximum lilelihood solutions calculated for a range of Cp values (see text for description). The first two header lines begin with the # symbol as the first character in a line.

Bibliography

Aki, K. and P. G. Richards (1985). Quantitative Seismology. *San Francisco: Freeman and Co.*

Artman, B., I. Podladtchikov, and B. Witten (2010). Source location using time-revers imaging. Geophys. Prosp. 58*, 861–873. doi: 10.1111/j.1365-2478.2010.00911.x.*

Brandt, S. (1999). Data Analysis. Statistical and Computational Methods for Scientists. *Springer-Verlag.*

Curtis, A. and A. Lomax (2001). Prior information sampling distributions and the curse of dimensionality. Geophysics 66*(2), 372–378.*

Danek, T. and W. Debski (2014). GPU Accelerated Wave Form Inversion Using Monte Carlo Sampling . Acta Geophys..

Debski, W. (2002). Seismic Tomography Software Package. Publs. Inst. Geophys. Pol. Acad. Sc. B-30*(353), 1–105.*

Debski, W. (2004). Application of Monte Carlo techniques for solving selected seismological inverse problems. Publs. Inst. Geophys. Pol. Acad. Sc. B-34*(367), 1–207.*

Debski, W. (2010). Probabilistic Inverse Theory. Adv. Geophys. 52*, 1–102. doi: 10.1016/S0065-2687(10)52001-6.*

Debski, W. (2015). Using meta-information of a posteriori Bayesian solutions of the hypocenter location task for improving accuracy of location error estimation. Geophys. J. Int. 201*(3), 1399–1408. doi: 10.1093/gji/ggv083.*

Debski, W. and R. Young (2002). Tomographic imaging of thermally induced fractures in granite using Bayesian inversion. Pure Appl. Geophys. 159*(1-3), 277–307. doi: 10.1007/978-3-0348-8179-1_13.*

Duijndam, A. W. (1988). Bayesian Estimation in Seismic Inversion. Part I: Principles. Geophys. Prosp. 36*, 878–898.*

Fink, M. (1997). Time reversed acoustic. Physics Today 50*, 34–40.*

Fink, M., D. Cassereau, A. Derode, C. Prada, P. Roux, M. Tanter, J.-L. Thomas, and F. Wu (2000). Time-reversed acoustics. Reports on Progress in Physics 63*(12), 1933–1994.*

Fishman, G. S. (1996). Monte Carlo Concepts Algorithms and Applications. *Springer Series in Operations Research. Springer.*

Font, Y., H. Kao, S. Lallemand, C.-S. Liu, and L.-Y. Chiao (2004). Hypocentre determination offshore of eastern Taiwan using the Maximum Intersection method. Geophys. J. Int. 158*, 655–675. doi: 10.1111/j.1365-246X.2004.02317.x.*

Gajewski, D. and E. Tessmer (2010). Reverse modelling for seismic event characterization. Geophys. J. Int. 163*, 276–284. doi: 0.1111/j.1365-246X.2005.02732.x.*

Gibowicz, S. J. and A. Kijko (1994). An Introduction to Mining Seismology. *San Diego: Academic Press.*

Gilks, W., S. Richardson, and D. Spiegelhalter (1995). Markov Chain Monte Carlo in Practice. *Chapman& Hall/CRC Press.*

Hanafy, S., W. Cao, K. McCarter, and G. Schuster (2009). Using super-stacking and super-resolution properties of time-reversal mirors to locate trapped miners. The Leading Eage 28*(3), 302–307. doi: 10.1190/1.3104074.*

Hauser, J., M. Sambridge, and N. Rawlinson (2008). Multiarrival wavefront tracking and its applications. Geochemistry, Geophysics, Geosystems 9*(11). doi: 10.1029/2008GC002069.*

Jackson, D. D. and M. Matsu'ura (1985). A Bayesian Approach to Nonlinear Inversion. J. Geophys. Res. 90*(B1), 581–591.*

Jeffreys, H. (1983). Theory of Probability. *Oxford: Clarendon Press.*

Kloc, M. and T. Danek (2012). The Multi GPU Accelerated Waveform Inversion in Distributed OpenCL Environment*, Volume 151 of* Lecture Notes in Electrical Engineering. *Springer New York. doi: 10.1007/978-1-4614-3558-7_19.*

Kremers, S., A. Fichtner, G. Brietzke, H. Igel, C. Larmat, L. Huang, and M. Kaser (2011). Exploring the potentials and limitations of the time-reversal imaging of finite seismic sources. Solid Earth 2*, 95–105. doi: 10.5194/se-2-95-2011.*

Larmat, C., R. Guyer, and P. A. Johnson1 (2009). Tremor source location using time reversal: Selecting the appropriate imaging field. Geophys. Res. Lett. 36*(L22304), L22304. doi: 10.1029/2009GL040099.*

Larmat, C., J. Tromp, Q. Liu, and J.-P. Montagner (2008). Time reversal location of glacial earthquakes. J. Geophys. Res. 113*(B09314), 1–9. doi: 10.1029/2008JB005607.*

Lomax, A. (2005). A Reanalysis of the Hypocentral Location and Related Observations for the Great 1906 California Earthquake. Bull. Seismol. Soc. Am. 95*(3), 861–877. doi: 10.1785/0120040141.*

Lomax, A., A. Michelini, and A. Curtis (2009). Earthquake Location, Direct, Global-Search Methods*, Volume 5. New York: Springer. doi: 10.1007/978-0-387-30440-3.*

Lomax, A., J. Virieux, P. Volant, and C. Berge (2000). Probabilistic earthquake location in 3D and layered models: Introduction of a Metropolis-Gibbs method and comparison with linear locations. *Amsterdam: Kluver. doi: 10.1007/978-94-015-9536-0_5.*

Masson, Y., P. Cupillard, Y. Capdeville, and B. Romanowicz (2014). On the numerical implementation of time-reversal mirrors for tomographic imaging. Geophys. J. Int.*, 1–11. doi: 10.1093/gji/ggt459.*

Matsu'ura, M. (1984). Bayesian estimation of hypocenter with origin time eliminated. J. Phys. Earth. 32*(6), 469–83.*

Menke, W. (1989). Geophysical Data Analysis: Discrete Inverse Theory. *International Geophysics Series. San Diego: Academic Press.*

Mosegaard, K. and A. Tarantola (2002). International Handbook of Earthquake & Engineering Seismology*, Volume 81 of* International Geophysics Series*. Academic Press.*

O'Brien, G., J. Lokmer, L. D. Barros, C. Bean, G. Saccorotti, J.-P. Metaxian, and D. Patane (2011). Time reverse location of seismic long-period events recorded on Mt Etna. Geophys. J. Int. 184*, 452–462. doi: 10.1111/j.1365-246X.2010.04851.x.*

Parker, R. L. (1994). Geophysical Inverse Theory. *New Jersey: Princeton University Press.*

Podvin, P. and I. Lacomte (1991). Finite-difference computtion of traveltimes in very contrasted velocity models: a massively paarallel approach and its associated tools. Geophys. J. Int. 105*, 271–284.*

Quinn, M. (2008). Parallel Programming in C with MPI and OpenMP. *McGraw-Hill Education.*

Rawlinson, N. and M. Sambridge (2004). Multiple reflection and transmission phases in complex layered media using a multistage fast marching method. Geophys. 69*(5), 2178–2193. doi: 10.1190/1.1801950.*

Robert, C. P. and G. Casella (1999). Monte Carlo Statistical Methods. *Springer Verlag.*

Rudzinski, L. and W. Debski (2011). Extending the Double Difference location technique for mining applications - part I: Numerical study. Acta Geophys. 59*(4), 785–814. doi: 10.2478/s11600-011-0021-5.*

Rudzinski, L. and W. Debski (2012). Extending the Double Difference location technique - improving hypocenter depth determination. J. Seismol. 17*(1), 83–94. doi: 10.1007/s10950-012-9322-7.*

Scalerandi, M., M. Griffa, and P. Johnson (2009). Robustness of computational time reversal imaging in media with elastic constant uncertainties. JOURNAL OF APPLIED PHYSICS 106*, 114911. doi: 10.1063/1.3269718.*

Sethian, J. A. (1999). Level set methods and fast marching methods: evolving interfaces in computational geometry fluid mechanics computer vision and materials science. *Cambridge Monographs on Applied and Computational Mathematics. New York: Cambridge University Press.*

Steiner, B. and E. Saenger (2012). Comparison of 2D and 3D time-reverse imaging - A numerical case study. Computers & Geosciences 46*, 174–182. doi: 10.1016/j.cageo.2011.12.005.*

Steiner, B. and H. Saenger (2011). Comparison of 2D and 3D time-reverse imaging - A numerical case study. Computers & Geosciences*. doi: 10.1016/j.cageo.2011.12.005.*

Tarantola, A. (1987). Inverse Problem Theory: Methods for Data Fitting and Model Parameter Estimation. *Amsterdam: Elsevier.*

Tarantola, A. (2005). Inverse Problem Theory and Methods for Model Parameter Estimation. *Philadelphia: SIAM.*

Tarantola, A. and B. Vallete (1982). Inverse Problems = Quest for Information. J. Geophys. 50*, 159–170.*

Tromp, J., C. Tape, and Q. Liu (2005). Seismic tomography, adjoints methods, time reversal and banana-doughnut kernels. Geophys. J. Int. 160*, 195–216. doi: 10.1111/j.1365-246X.2004.02453.x.*

Ulrich, T., K. V. D. Abeele, P.-Y. L. Bas, M. Griffa, B. Anderson, and R. Guyer (2009). Three component time reversal: Focusing vector components using a scalar source. J. Appl. Physics 106*, 113504. doi: 10.1063/1.3259371.*

Ulrich, T., A. Sutin, R. Guyer, and P. Johnson (2008). Time reversal and non-linear elastic wave spectroscopy (TR NEWS) techniques. Int. J. of Non-Lin. Mechanics 43*(3), 209–216. doi: 10.1016/j.ijnonlinmec.2007.12.017.*

Virieux, J., S. Operto, H. Ben-Hadj-Ali, R. Brossier, V. Etienne, and F. Sourbier (2009). Seismic wave modeling for seismic imaging. Leading Eadge*, 538–544. doi: 10.1190/1.3124928.*

Waldhauser, F. and W. Ellsworth (2000). A double-difference earthquake location algorithm: method and application. Bull. Seismol. Soc. Am. 90*, 1353–1368.*

W.Debski and P.Klejment (2015). The new algorithm for fast probabilistic hypocenter locations. Acta Geophys..

Wiejacz, P. and W. Debski (2001). New Observation of Gulf of Gdansk Seismic Events. Phys. Earth Planet. Int. 123*(2-4), 233–245. doi: 10.1016/S0031-9201(00)00212-0.*

Witten, B. and B. Artman (2011). Signal-to-noise estimates of time-reverse images. Geophysics 76*(2), MA1–MA10. doi: 10.1190/1.3543570.*

Zhang, L., J. Rector, and G. Hoversten (2005). Eikonal solver in the celerity domain. Geophys. J. Int. 162*, 1–8. doi: 10.1111/j.1365-246X.2005.02626.x.*

Zhao, H. (2005). Fast Sweeping Method for Eikonal equations. Math. of Comput. 74*, 603–627. doi: 10.1090/S0025-5718-04-01678-3.*

Zhou, H. (1994). Rapid three-dimensional hypocentral determination using a master station method. J. Geophys. Res. 99*(B8), 715439–15455.*