Development and application of 2D and 3D transient electromagnetic inverse solutions based on adjoint Green functions: A feasibility study for the spatial reconstruction of conductivity distributions by means of sensitivities

Inaugural–Dissertation

zur
Erlangung des Doktorgrades
(Dr. rer. nat.)
der Mathematisch–Naturwissenschaftlichen Fakultät
der Universität zu Köln

vorgelegt von

Roland Martin

aus Hückeswagen

Köln 2009

Gutachter:	Prof. Dr. B. Tezkan
	Prof. Dr. A. Kemna

Tag der mündlichen Prüfung: 22. April 2009

Abstract

To enhance interpretation capabilities of transient electromagnetic (TEM) methods, a multi-dimensional inverse solution is introduced, which allows for a explicit sensitivity calculation with reduced computational effort. The main conservation of computational load is obtained by solving Maxwell's equations directly in time domain. This is achieved by means of a high efficient Krylov-subspace technique that is particularly developed for the fast computation of EM fields in the diffusive regime. Traditional modeling procedures for Maxwell's equations yields solutions independently for every frequency or, in the time domain, at a given time through explicit time stepping. Because of this, frequency domain methods are rendered extremely time consuming for multi-frequency simulations. Likewise the stability conditions required by explicit time stepping techniques often result in highly inefficient calculations for large diffusion times and conductivity contrasts.

The computation of sensitivities is carried out using the adjoint Green functions approach. For time domain applications, it is realized by convolution of the background electrical field information, originating from the primary signal, with the impulse response of the receiver acting as secondary source. In principle, the adjoint formulation may be extended allowing for a fast gradient calculation without calculating and storing the whole sensitivity matrix but just the gradient of the data residual. This technique, which is also known as migration, is widely used for seismic and, to some extend, for EM methods as well. However, the sensitivity matrix, which is not easily given by migration techniques, plays a central role in resolution analysis and would therefore be discarded. But, since it allows one to discriminate features in the a posteriori model which are data or regularization driven, it would therefore be very likely additional information to have. The additional cost of its storage and explicit computation is comparable low disbursement to the gain of a posteriori model resolution analysis.

Inversion of TEM data arising from various types of sources is approached by two different methods. Both methods reconstruct the subsurface electrical conductivity properties directly in the time domain. A principal difference is given by the space dimensions of the inversion problems to be solved and the type of the optimization procedure. For two-dimensional (2D) models, the ill-posed and non-linear inverse problem is solved by means of a regularized Gauss-Newton type of optimization. For three-dimensional (3D) problems, due to the increase of complexity, a simpler, gradient based minimization scheme is presented. The 2D inversion is successfully applied to a long offset (LO)TEM survey conducted in the Arava basin (Jordan), where the joint interpretation of 168 transient soundings support the same subsurface conductivity structure as the one derived by inversion of a Magnetotelluric (MT) experiment. The 3D application to synthetic data demonstrates, that the spatial conductivity distribution can be reconstructed either by deep or shallow TEM sounding methods.

Zusammenfassung

Zur Verbesserung der Interpretationsmöglichkeiten Transient elektromagnetischer (TEM) Methoden wird eine mehrdimensionale Inversionsmethode vorgestellt, die unter kleinstmöglichem Rechenzeitaufwand eine explizite Berechnung von Sensitivitäten erlaubt. Die hauptsächliche Zeitersparnis wird dadurch erreicht, die Maxwell'schen Gleichungen direkt im Zeitbereich zu lösen und ein sehr effizientes Krylov-Unterraum Verfahren einzusetzen, welches eigens zur schnellen Berechnung elektromagnetischer Diffusionsprozesse entwickelt wurde. Traditionelle Modellierungsverfahren zur Lösung der Maxwell'schen Gleichungen entwickeln ihre Lösungen in der Regel nur für einzelne Frequenzen oder bei Anwendung expliziter Zeitschrittverfahren zu einer bestimmten Zeit. Die Anwendung von Methoden die im Frequenzbereich arbeiten ist daher äußerst zeitraubend. Bei expliziten Zeitschrittverfahren führen die Stabilitätsbedingungen, die zur Berechnung von langen Diffusionszeiten und hohen Leitfähigkeitskontrasten eingehalten werden müssen, zu sehr langen Rechenzeiten.

Die Berechnung der Sensitivitäten wird mit der Methode der adjungierten Green-Funktionen durchgeführt. Für Zeitbereichsmethoden wird dies durch eine numerische Faltung der im Hintergrund enthaltenen elektrischen Felder des primären Stromsignals mit der Impulsantwort des Empfängers in Form einer sekundären Stromquelle realisiert. Prinzipiell könnte die Lösung des adjungierten Problems durch Miteinbeziehung des Daten-Residuums erweitert werden, um eine schnelle und direkte Berechnung von Gradienten zu ermöglichen, jedoch wird hierdurch auf die zusätzliche Information der Sensitivitäten verzichtet. Dieses Verfahren wird beispielsweise bei der Migration seismischer Daten und teilweise auch für elektromagnetische Daten verwendet. Die Sensitivitätsmatrix spielt jedoch in der Analyse der Auflösungseigenschaften postulierter Modelle eine zentrale Rolle. Bei eingehender Analyse der Auflösungseigenschaften kann beispielsweise unterschieden werden zwischen den regularisierungsbedingten Bereichen und denen die hauptsächlich durch die gemessenen Daten beeinflusst werden, was ausdrücklich erwünscht ist. Der zusätzliche Aufwand der bezüglich Speicherung und der expliziten Formulierung verloren geht ist eine vergleichsweise kleine Einbuße die aufgrund der zusätzlichen Möglichkeit, die Aussagekraft postulierter Modelle zu erhöhen, gerne in Kauf genommen wird.

Die Inversion von TEM Daten unterschiedlicher Quellsignale wird mit zwei verschiedenen Methoden realisiert. Beide Methoden haben zum Ziel die Leitfähigkeitsstruktur im Untergrund zu rekonstruieren, wobei eine prinzipielle Unterscheidung bezüglich der untersuchten Raumdimensionalität und der Art der Optimierungsmethode vorgenommen werden kann. Bei zweidimensionalen (2D) Modellen wird das nicht lineare und schlecht gestellte Inversionsproblem durch ein regularisiertes Gauß-Newton Verfahren gelöst. Für dreidimensionale (3D) Probleme wird aufgrund der großen Komplexität ein simpleres Gradienten basierendes Verfahren vorgestellt. Die 2D Inversion wird erfolgreich auf transiente Daten angewendet, die im Rahmen einer long offset (LO)TEM-Messkampagne, innerhalb des Arava Beckens (Jordanien), gewonnen wurden. Das Modell, welches bei der gemeinsamen Interpretation von 168 transienter Sondierungen postuliert wird, unterstützt die gleichen Leitfähigkeitsstrukturen wie das Inversionsmodell eines Magnetotellurik (MT) Experiments. Die 3D Anwendungen auf synthetische Daten zeigen, daß die räumliche Rekonstruktion der Leitfähigkeitsverteilung sowohl mit tiefen als auch mit oberflächennahen TEM Sondierungsmethoden möglich ist.

Contents

Table of contents **vii**

List of figures **x**

List of tables **xiv**

1 Introduction **1**

2 Inversion theory **9**

2.1 The non-linear optimization problem 9

2.1.1 Line search 11

2.1.2 Gradient based optimization 13

2.1.3 Newton's method 16

2.2 Least squares formulation 18

2.2.1 Data covariance 19

2.2.2 Gauss-Newton 19

2.3 Regularization 21

2.3.1 Smoothness constraints 22

2.4 Resolution analysis 26

2.4.1 Resolution of linear problems 26

2.4.2 Comparing resolution of local and global regularization 27

2.5 Summary 28

3 TEM forward and adjoint operators 29

3.1 EM theory 30

3.1.1 Complementary solution 33

3.1.2 Evaluating Bessel integrals numerically by a fast Hankel transform (FHT) 35

3.1.3 Step excitation and time transformation 36

3.1.4 Summary 37

3.2 Depth sounding with controlled source transient electromagnetics 38

3.2.1 Short offset transient electromagnetics (SHOTEM) 38

3.2.2 Long offset transient electromagnetics (LOTEM) 42

3.2.3 System response 45

3.2.4 Summary 48

3.3 3D forward solution with the *SLDMem3t* 49

3.3.1 Approximate solution in Krylov space 54

3.4 Grid design for *SLDMem3t* 57

3.4.1 Grid automate *make_sldm_grid* 58

3.4.2 Extended grid analysis 67

3.4.3 Summary/discussion 75

3.5 Sensitivity calculation and adjoint formulation for TEM 76

3.5.1 Finite difference sensitivity calculation (perturbation method) 77

3.5.2 Adjoint formulation 78

3.5.3 Implementation of TEM sensitivity calculation 82

3.5.4 Sensitivity comparison of long and short offset TEM 94

3.5.5 Summary 97

4 Multidimensional TEM inversion 99

4.0.6 Preliminary notes 99

4.1 General strategy for multidimensional TEM inversion 100

4.1.1 Lagrange parameter determination 101

4.1.2 Implementation for TEM inversion 102

4.1.3 χ^2 and RMS 104
4.1.4 Data transformations 105
4.2 Noise generation for synthetic data 108
4.3 2D TEM inversion 109
4.3.1 Implementations 110
4.3.2 2D inversion of synthetic LOTEM data 111
4.3.3 Comparison of 2D inversions using different receiver combinations . . 113
4.3.4 Discussion 119
4.4 2D Inversion results of the LOTEM DESERT survey 120
4.4.1 Target area: A part of the Dead Sea Transform 120
4.4.2 Projectives 121
4.4.3 Prior measurements and data 121
4.4.4 LOTEM measurements 121
4.4.5 Preliminary 2D inversion results of the LOTEM data 123
4.4.6 Selected LOTEM transients and model fit 125
4.4.7 Summary 127
4.5 3D TEM inversion 128
4.5.1 Optimization strategies for 3D TEM inversion 129
4.5.2 3D inversion of LOTEM data, synthetic example 131
4.5.3 3D inversion of SHOTEM data, synthetic example 136
4.5.4 Discussion 142

5 Conclusions and discussion 143

Bibliography 145

A 1D adjoint fields 157

A.0.5 TE- TM-mode calculation for a layered earth 157
A.0.6 Particular solutions for EM field potentials of inductive and galvanic sources over layered media 160
A.1 Adjoint field calculations for layered media 163
A.1.1 Adjoint fields for $\dot{\mathsf{H}}_z$ receivers 164
A.1.2 Adjoint fields for $\dot{\mathsf{H}}_x$ and $\dot{\mathsf{H}}_y$ receivers 164

A.1.3 Adjoint fields for the E_x and E_y receivers 165

B Direct and iterative solutions of linear systems 167

B.0.4 Matrix factorization 168

B.0.5 Conjugate gradient for solving linear systems (CG) 171

B.0.6 Preconditioning 174

C A posteriori model covariance 177

D Manual for *sinv* 179

D.1 MPI 179

D.2 Input files 181

D.2.1 *sinv.inp* 181

D.3 Inversion process 185

E DESERT data with 2D model fit 187

E.0.1 E_x-field data 188

E.0.2 $\dot{\mathsf{H}}_\mathrm{y}$ and $\dot{\mathsf{H}}_\mathrm{z}$ data 195

Acknowledgments 209

Used Symbols 213

List of Figures

2.1 Sectioning 12

2.2 Gradient descent sketch 14

2.3 Conjugated gradient vs. gradient descent 15

2.4 Newton vs. gradient descent 16

3.1 Resistivity ranges for earth materials (after Palacky [1988]). 31

3.2 Frequency and time range of some EM-techniques (after Tezkan [1999]). . . . 32

3.3 Common SHOTEM setups. 38

3.4 Sketch of TEM transmitter signal for 50% duty cycle and induced secondary fields. 39

3.5 Smoke ring concept after Nabighian [1979] 40

3.6 Two layer case model and SHOTEM in-loop transient response. 40

3.7 Basic LOTEM field setup. 42

3.8 Sketch of transmitter signal for 100% duty cycle and transient response. . . . 43

3.9 Two layer model and E_x-, $\dot{\mathsf{H}}_y$- and $\dot{\mathsf{H}}_z$-transient responses. 44

3.10 Sketch of the ramp time and system response. 47

3.11 Sketch of a Yee-cell 51

3.12 Common regular grids 52

3.13 Display of the Material Averaging scheme. 53

3.14 Comparison of two different logarithmic discretizations. 60

3.15 Dipole discretization. 61

3.16 Dipole discretization for a whole space model. 61

3.17 Bipole discretizations for LOTEM 62

3.18 Grid for square loop transmitters. 64

3.19 Comparison of fine and rough discretized square loop transmitters. 65

3.20 Fit of a in-loop $\dot{\mathsf{H}}_z$-transient with different discretizations for a coarse grid. . 65

3.21 Comparison of the transient responses for a medium sized discretization. . . 66

3.22 Comparison of the transient responses for a fine discretization. 66

3.23 Comparison of transient responses for the adjusted convergence criteria for *SLDMem3t*. 67

3.24 Comparison of transient responses for the adjusted convergence criteria for *SLDMem3t* with fine discretization. 67

3.25 Comparing M and $log_{10}\left(\|\mathbf{r}_e\|_1\right)$ as function of N_z and N_g with $\sigma_{min,max} = 1/100\,S/m$ 72

3.26 Comparing M and $log_{10}\left(\|\mathbf{r}_e\|_1\right)$ as function of N_z and N_g with $\sigma_{min,max} = (1/500, 1/50)\ S/m$. 73

3.27 Comparing M and $log_{10}\left(\|\mathbf{r}_e\|_1\right)$ as function of N_z and N_g with $\sigma_{min,max} = (1/1000, 1/10)\ S/m$. 73

3.28 Comparing M and $log_{10}\left(\|\mathbf{r}_e\|_1\right)$ as function of N_z and N_g with $\sigma_{min,max} = (1/1000, 1)\ S/m$. 74

3.29 Comparing M and $log_{10}\left(\|\mathbf{r}_e\|_1\right)$ as function of N_z and N_g with $\sigma_{min,max} = (1/5000, 1/5)\ S/m$. 74

3.30 Comparing M and $log_{10}\left(\|\mathbf{r}_e\|_1\right)$ as function of N_z and N_g with $\sigma_{min,max} = (1/10000, 1)\ S/m$. 75

3.31 Sensitivity calculation using FD. 78

3.32 Geometry for the sensitivity calculation using adjoint Green functions. . . . 79

3.33 Interpolation scheme for the background fields. 84

3.34 Comparison of cubic spline and linear interpolation at discontinuous fields. . 85

3.35 Parallelization scheme for distributed memory systems (Cluster) using MPI . 86

3.36 Re-weighting of the sensitivity density ($\mathbf{S}$) at the image point $\mathbf{r}'$ by volume integration. 87

3.37 Parameterization of the model for surface near regions. 88

3.38 Spatial 8-point interpolation operator. 89

3.39 Step and impulse response of electrical fields in the vicinity of a grounded dipole source. 91

3.40 Comparison of sensitivities for a surface near image point using different methods. 94

3.41 LOTEM setup, model and grid for a sensitivity comparison using adjoint and FD methods. 95

3.42 Comparison of LOTEM E_x-, $\dot{\mathsf{H}}_y$- and $\dot{\mathsf{H}}_z$-sensitivities using different calculation methods. 95

3.43 SHOTEM setup, model and grid for a sensitivity comparison using adjoint and FD methods. 96

3.44 Comparison of SHOTEM sensitivities using different calculation methods. . . 97

4.1 Comparison of the value range of four different data transformations over their domain. 106

4.2 $\dot{\mathsf{H}}_z$-field data noise. 110

4.3 Noise generation for synthetic data sets with *emuplus* 110

4.4 2D model for synthetic LOTEM data. 112

4.5 2D parameterization of the model for the inverse process. 113

4.6 Comparison of selected data for the starting model and the inverse solution with the original data. 115

4.7 Compilation of 2D inversion statistics and data fit for all receiver data. . . . 116

4.8 Comparison of 2D inversion results using first and second order smoothness. 117

4.9 Cross section of the 2D inversion results using different smoothing constraints and E_x-field transient data only. 117

4.10 Cross section of the 2D inversion results using different smoothing constraints and $\dot{\mathsf{H}}_y$-field transient data only. 118

4.11 Cross section of the 2D inversion results using different smoothing constraints and $\dot{\mathsf{H}}_z$-field transient data only. 118

4.12 Cross section of the 2D inversion results using different smoothing constraints and all $\dot{\mathsf{H}}$-field transient data. 118

4.13 Plate tectonic and rift system of the Dead Sea Transform 120

4.14 2D MT and seismic sections of the Arava valley. 121

4.15 The DESERT LOTEM setup on the Arava Fault 122

4.16 2D parameterization for the LOTEM DESERT model. 123

4.17 2D MT section of the measured area 124

4.18 Comparison of 2D DESERT LOTEM inversion results using all receiver data. 124

4.19 Normalized coverage of the preliminary 2D LOTEM inversion results 124

4.20 Selected DESERT E_x-field data and fit. 126

4.21 Selected DESERT $\dot{\mathsf{H}}_\mathsf{y}$-field data and fit 126

4.22 Selected DESERT $\dot{\mathsf{H}}_\mathsf{z}$-field data and fit 127

4.23 Plane view of the 3D model and LOTEM setup for synthetic data. 131

4.24 Simple conductivity model for the generation of synthetic LOTEM data. . . 131

4.25 Plane view and cross section of the parameterization for the 3D inversion. . . 132

4.26 Compilation of 3D LOTEM inversion statistics and data fit for all receiver data.133

4.27 Comparison of xy-slices through the model after the first and last iteration step134

4.28 Cross sections of the anomalous 3D body 135

4.29 Cross sections of the modeling domain showing the diagonal entries of the resolution matrix of the a posteriori 3D model after 10 iterations. 135

4.30 Plane view of the 3D model and SHOTEM setup for synthetic data. 137

4.31 Simple 3D conductivity model for the generation of measured SHOTEM data. 137

4.32 xz- and yz-cross section of the model parameterization for the 3D SHOTEM inversion. 138

4.33 Compilation of 3D SHOTEM inversion statistics and data fit for all receiver data. 139

4.34 Comparison of xy-slices of the model after the fourth and last iteration step. 140

4.35 Comparison of yz-cross sections of the model after the fourth and last iteration step. 140

4.36 Comparison of selected data for starting model and the 3D inverse solution with the original data. 141

A.1 Scheme of EM field calculation for layered conductivity profile 157

B.1 Singular value spectrum of a typical ill-posed problem and transform function using Tikhonov regularization 170

List of Tables

2.1 Comparison of update formulas of the Hessian matrix or its inverse. 18

3.1 Comparing the performance of grid1 and grid2 for grid layout of a homogeneous model. 69

3.2 Comparing the performance of grid1 and grid2 for grid layouts of a model with intermediate conductivity range. 70

3.3 Comparison of performance of symmetric grids with varying model layouts. . 71

3.4 Comparing the performance of symmetric grids with different model layouts. 71

4.1 Inversion statistics for 2D models using different receiver data sets 114

D.1 First part of *sinv.inp* which consists of the main switches to control the inversion process 182

D.2 regularization parameters for 2D-Occam or any 3D inversion process 182

D.3 regularization parameters for 2D-Marquardt 183

D.4 Grid section of the input file 183

D.5 Receiver section of the input file 184

D.6 Model section of the input file 184

CHAPTER 1

Introduction

To investigate the physical properties of the earth interior we use geophysical methods, since direct investigations are rarely possible. Geophysical depth soundings utilize different physical fields or processes interacting with earth material. The task is then to find a representative model for the earth which can explain the observed physical interactions (*interpretation*).

Since the physical processes are mainly well understood, they can be modeled by solving the appropriate partial differential equation (PDE). The solution represents the physical response of the model and is therefore mostly called *model response* or *forward solution.* Geophysical methods can be roughly separated into three different kind of PDEs, which are solved for their interpretation[1]:

- Wave equations, or *hyperbolic* PDEs, of the form $\mathbf{Af} + \ddot{\mathbf{f}} + \dot{\mathbf{f}} = 0$ arise for e.g. in Seismic, Radar or subsonic measurements. $\mathbf{A}$ refers to some kind of spatial operator (e.g. the Laplace operator), $\dot{\mathbf{f}}$ and $\ddot{\mathbf{f}}$ are the first and second derivative with respect to time and $\mathbf{f}$ is the field one measures or simulates using an appropriate model (e.g. sonic velocity of the earth material). Because of the second order derivation, those methods generally believed to deliver the highest spatial resolution.

- Diffusive equations, or *parabolic* PDEs are formulated through $\mathbf{Af} + \dot{\mathbf{f}} = 0$ and solved for the interpretation of electromagnetic investigations like Magnetotelluric (MT), Controlled Source ElectroMagnetic (CSEM) and or Transient ElectroMagnetic soundings (TEM). They deliver a moderate spatial resolution, but aim to investigate at a different parameter (e.g. conductivity).

- Last not least Potential methods, like Direct Current geoelectrics (DC), Magnetic, Gravity, or Self-potential (SP), etc., which are solved using the *elliptic* PDE: $\mathbf{Af} = 0$.

[1] For convenience, the physical nature of the earth is more complex and the underlying processes often obey mixed PDEs. But for a rough classification of the methods, they may be compiled in this manner.

> The elliptic PDEs are generally the most easiest class of equations to solve. As a drawback, potential methods are believed to have the poorest spatial resolution.

Presently, because of the tremendous growth of computational power during the last decades, the solution of the above stated PDEs can be solved for multidimensional (2D and 3D) models with very fine spatial discretization.

Transient electromagnetic methods have a well-established place in exploration geophysics, because they have the potential to provide very useful additional information for problems associated for example with mineral exploration [Sarma et al., 1976], oil exploration [Spies, 1983; Strack, 1985], volcano-logical hazards and hydrological investigations. An excellent review of the TEM method and its uses is given by Nabighian & Macnae [1991]. A collection of related publications can be found in the special TEM issue of Geophysics, Vol. 49, 1984.

As a key issue for the geophysical investigation of the subsurface within commercial applications, one uses more then one method aiming at different physical properties and combines their different advantages to deliver one representative model of the surrounding (*joint interpretation*). Since the connection between different physical parameters, like velocities and resistivity for e.g., is very difficult to establish, these integrated model interpretation are often carried out one after another.

The most common and easiest interpretation technique is to apply some sort of transformation to the measured signal or data (e.g. apparent resistivities or seismic velocities). This leads to an image of the parameter we like to know and is therefore often referred to as *direct imaging.* Direct imaging techniques can be applied in many fields of geophysics (e.g. MT, Seismic, etc.) by drawing a map of the transformed measurements with depth. While this transformation gives a rough image of the parameter distribution of the subsurface it is very incomplete. This is due to the inherent complexity of the parameter dependencies.

Interpretation by means of inverse solution(s)

To get a more complete image of the subsurface structure, the *inverse process* is applied. Herein, one wants to find a representative, or *a posteriori* model, which model response explains the measured data, within some specified manner.

The inverse process can be formulated as an *optimization* process of a predefined *cost functional* (e.g. least squares misfit), where the cost functional measures to what extend the model response fits the measured data. E.g., high cost corresponds to bad fit, whereas small cost denotes that the model response better agrees to the measurements, and hence the a posteriori model is more likely to describe the real model.

The simplest form of inverse process can be described with the trial and error procedure to seek the a posteriori model: (A) Choose a model (*prior*). (B) Calculate the model response. (C) Compute the cost and decide: *if* the cost is sufficiently small than terminate, *else* proceed with (A)-(C).

As a side effect, if the model is selected by somebody, it would be naturally biased towards individual preferences. Besides the problem, that the trial and error procedure may be very exhaustive as well, some questions would arise:

1. How do we choose the model?

2. Which parameters do influence the model response the most?

3. What are the uncertainties of the a posteriori model?

4. How can we incorporate uncertainties of the data into the model?

Therefore, and to give answers to the questions which arise, one may apply more sophisticated optimization techniques in terms of optimization algorithms. In general, the cost functional to minimize is of non linear nature and one can distinguish between three different approaches in non linear optimization: The statistical, the deterministic and hybrid approaches as a mixture of the previously mentioned. Although the term ”statistical” and ”deterministic” approach is not used in the literature, it is used here to distinguish between the two different optimization processes because of their underlying principle.

Deterministic approaches[2] try to find the minimum of the cost functional using standard mathematical models. They are applied most commonly and utilize the first (e.g. gradient methods) and second order derivatives (Newton or Quasi-Newton methods) of the cost functional to find an optimal parameter subset. Because of the non linear nature of the problem, the second order derivatives are rarely computable and one uses approximations to express them. However, the key feature within these approaches is to know, how changes in the model are mapped into the data space (*sensitivity*). Sensitivities can be imagined in the way, that they provide the inversion algorithm with the knowledge of which screw to drive in order to adjust the model in the way to minimize the misfit. Deterministic inversion schemes are generally the fastest methods in terms of computation time and widely applied in geophysics.

Statistical approaches, or global search methods, target to find not only a global minimum within the optimization process, but deliver further likelihood informations about the structure of the problem by forming a probability density of the cost functional (*pdf*). The pdf is widely believed to be the only real mapping for the non linear complexity of the underlying problem. Although, the probabilistic approach can be formulated in a least squares sense using the Gauss-Markov theorem [Tarantola, 1987; 2005], they require a Markov chain to get independent model sampling. This means, that the model should be (more or less) randomly selected, so that the model at any state of the inverse process is completely independent from the previous. The Markov chain can be applied in some cases, where the number of model unknowns is relatively small [Mosegaard & Tarantola, 1995; Mosegaard & Sambridge, 2005]. Therefore, this inverse process is regarded as a global search in the parameter space which treats the non linear nature of the physical model properly. But, due to the big amount of computational load, they are generally considered unfeasible for models with multidimensional character. Even for an one dimensional parameter distribution this can mean tremendous computation cost, because many forward calculations have to be carried out. Hence, most geophysicists favor the deterministic inverse mapping of the parameter, to get a (more or less) unbiased estimation of the subsurface parameter distribution. Yet, some applications of "intelligent" Markov chains, as the Neighboring Algorithm [Sambridge, 1999] or Genetic Algorithms [Stoffa & Sen, 1991] or the application of Neural Networks may yield interesting results for geophysical problems, as well. They are compiled under the term hybrid approach and try to bring the fast convergence of the deterministic approach together with the non linear treatment of the probabilistic approach.

[2] For brevity, the class of deterministic approaches is mostly called just inversion process.

Overview of present multidimensional TEM inversion schemes

Besides the physical background, which enables us to simulate the inherent processes, the most essential part for any data interpretation is the solution of the inverse problem. Although the forward and inverse problem for many applied geophysical methods are solved in 3D now, the interpretation of CSEM data sets with fine discretized spatial conductivity models is still a challenging task [Alumbaugh & Newman, 1997]. Especially the interpretation of TEM data sets are commonly still done with 1D models. This is mainly due to the relatively high computation cost which arises for the simulation of the diffusive EM processes which arise in 3D conductivity structures. Because of the additional inhomogeneous field starting values for the partial differential equations to be solved, the computational effort becomes even larger for transient electromagnetics. As opposite to the frequency domain, for time domain CSEM methods one has to take care of several decades of decay times, rendering the TEM methods a very hard forward problem for spatial conductivity distributions and are mostly solved within massive parallel computer environments (e.g. Newman & Commer [2005]).

Recently, there are some improved 1D inversion strategies successfully applied and further developed. The laterally constraint 1D inversion (LCI), as suggested by Auken & Christensen [2004], bonds vertical model sections of one direction (e.g. y-direction), leading to smoother lateral variations within the a posteriori model. Another approach, presented by Scholl [2005], may be called soft joint 1D inversion (SJI). It applies a common joint inversion of long offset (LO)TEM data not only for different components (e.g. E_x and $\dot{\mathsf{H}}_\mathrm{y}$) but for different receivers sites as well. It can be seen as a hybrid approach between the LCI and the joint inversion. Every transient sounding refers to it's own model and they where linked together through a smoothness matrix. In general this leads to more flexibility as common joint inversion strategies fail sometimes to find one representative model for every component at one receiver site. The spatially constraint inversion (SCI), as proposed by Viezzoli et al. [2008], represents the spatially extension of the LCI method. Herein, the concatenation of vertical model sections is not limited into one direction, but spatially neighboring receiver sites are linked together. This leads to much more complex structure within the a posteriori model which may be seen as quasi-3D inversion. However, the forward modeling is still done by means of 1D solutions which leads to approximation errors within the model response.

Published TEM inversion approaches, which attack the full 3D problem directly in the time domain are presented by Wang et al. [1994]; Zhdanov & Portniaguine [1997]; Commer [2003]; Newman & Commer [2005]; Commer & Newman [2008] and use gradient based inversion techniques. The main feature of the proposed solutions is to avoid a explicit sensitivity calculation, by using a migration scheme which is a standard interpretation technique in seismic (e.g. Loewenthal et al. [1976]). This may be advantageous for huge data sets with millions of unknowns, because the storage of the sensitivities can be prohibitive for this cases. However, the additional resolution information which inherits the sensitivity matrix can not be used without additional effort within common EM migration schemes [Zhdanov & Tolstaya, 2006]. Another approach, using the nearly quadratic convergence advantage of a Quasi-Newton scheme, was recently published by Haber et al. [2007]. Here, the authors use an interesting preconditioning approach to solve the adjoint problem, reaching nearly quadratic convergence for models with some hundred thousand parameters. Yet, even with a more "sophisticated" inversion techniques as described by Haber et al. [2007], the additional information of the sensitivity matrix is never stored and thus can not be accessed directly.

Thus, resolution analysis of the a posteriori model of 3D TEM inversion is very complicated and the current published inversion schemes do not deliver such information. Additionally, the inversion schemes presented by Commer & Newman [2008] are currently running on massive parallel workstations rendering the approach unfeasible for commercial applications.

Thesis

To enhance interpretation capabilities of TEM, a fast multidimensional inverse solution with reduced computational effort but explicit sensitivity calculation is established. The main conservation of computational load is achieved by solving the most relevant forward problems by means of the widely used SLDM, proposed by Druskin & Knizhnermann [1988]. The computation time for an inversion is further minimized by distributing the multiple forward simulations within a parallel computer environment. The optimization is based on gradient and Gauss-Newton methods, whereas the parallelization scheme is mainly designed for Linux-Clusters and compiled in a program called *sinv*.

In the current development state of *sinv*, the treatment of three different model parameterizations are implemented within different parts of the program:

1. (Low-parameterized 3D models (small-scale 3D inversion)).
2. Models following a 2D parameterization (2D inversion).
3. Arbitrary 3D models (large-scale 3D inversion).

The first approach represents a small-scale 3D inversion and is based upon the work of Commer [2003] and can be treated as starting point of my work. It was mainly been developed for the purpose of refining a priori known 3D underground structures. It uses an unconstrained least-squares inversion algorithm and is customized to invert for 3D earth models with a limited model complexity in a way that the problem is over determined by the number of data points. Therefore, a priori information is an important requirement to design a model such that its limited degrees of freedom describe the structures of interest. The inversion is successfully applied to data from a LOTEM survey at the active volcano Merapi in Central Java (Indonesia). The interpretation of the resistivity images obtained by the inversion have substantially advanced the structural knowledge about the volcano [Commer, 2003]. Because it is already described in detail in the work of Commer [2003] and no major adjustments to this part of the code are done, it is put in brackets to emphasize that it is not part of the investigations within this work.

With the low parameterized model assumption of the small scale 3D inversion scheme one has to rely on a very detailed first guess to the model parameterization. The most realistic image of the Earth can be obtained if a model variation in all three dimensions is allowed in an inversion. Therefore, to attack a multidimensional inversion with arbitrary model parametrization, two further inversion strategies, namely for 2D or 3D model discretizations are realized and described throughout this work. They may seen as a extension of the work of Newman & Commer [2005], though with two major differences:

1. All 3D forward modeling is carried out with the fast and efficient *SLDMem3t*. Thus, the inversion program may be used within low cost computer environments (e.g. small

Linux Clusters or modern PCs) and is not limited to expensive massive parallel workstations.

2. Resolution information can be accessed directly by investigating the sensitivity of single data points or the whole sensitivity distribution. Thus, one may carry out extensive resolution analysis for model and or survey designs.

The program is designed to compute the inverse solution as well as sensitivities for any source and receiver types commonly used within TEM investigations. It is tested with synthetic 2D and 3D models and successfully applied to field data from a 2D LOTEM investigation of a part of the Dead Sea Transform (DST) system. The LOTEM data originates from measurements that where taken between 2004 and 2005 within the joint venture project DESERT. There, a deep crustal study to gain more insight into the DST was carried out at the Arava Fault (Jordana), carrying out several MT/Seismic surveys as well [DESERT Working Group, 2004]. The inversion result shows very good agreement to the MT interpretation for lower crustal features, which where showed by Ritter et al. [2003], and enhances the interpretation for depths between $50\,m$ and $3\,km$.

This thesis is organized in three parts. The first part, containing chapter 2 and 3, is of theoretical nature. It gives a general insight into the formulation and solution of non linear inverse and the TEM forward problem in particular. In chapter 2, I derive the basic equations for the formulation of the *unconstraint* and the *constraint* or *regularized* non linear optimization problem, the *normal equations.* I explain the main strategy and present formal derivation of the algorithms which I use for gradient based and the more sophisticated Quasi-Newton methods. Further I describe the major differences in terms of resolution between the two major branches of non linear optimization. A key to the resolution analysis of the a posteriori model is the proper computation of sensitivities. For time domain methods, Hördt [1998] showed a convolution scheme based on the adjoint Green function approach [Roach, 1982; McGillivray & Oldenburg, 1990], which allows a fast computation of TEM sensitivities directly in the time domain. Therefore it is convenient to follow the approach of Hördt [1998] which implementation and further intricacies is given in chapter 3. Besides the computation of the sensitivity matrix, the solution to the forward problem for TEM in particular is focused in this chapter as well as the treatment of some peculiarities when dealing with TEM data, especially how the system response is treated. A key issue for the computation of transient fields by means of the *SLDMem3t* is the automated grid generation, which I discuss in detail in section 3.4.

The next part containing chapter 4, shows applications of the proposed inverse solution to synthetic and real field data, respectively. I focus on the implemented inversion strategies for 2D and 3D TEM inverse solutions and show proof of concept for both, 2D and 3D inversion with synthetic data. The application to a synthetic LOTEM survey reveals how the resolution capabilities of the inverse solution can be enhanced by combination of different receiver components. Afterwards, I show the previously mentioned application to field data and compare the 2D LOTEM inverse solution to the MT results of Ritter et al. [2003]. For 3D applications I show examples of an artificial 3D LOTEM campaign for deep crustal studies. To the opposite, I can show how the proposed inverse solution resolves a surface near conductivity anomaly, within a synthetic 3D SHOTEM campaign. In the last part, comprising chapter 5, I summarize the investigations of this work and give some conclusions.

Preliminary notes

Following Goldman et al. [1994], in all chapters the word voltage or induced voltage shall be used instead of both ”magnetic field time derivative” or ”magnetic induction time derivative”. Although it may be argued that electric field measurements also involve voltage measurements, I will call them just electric fields. Vectors and matrices will be represented by bold characters. Lower case characters are used for vectors, upper case letters are used for matrices.

In order to compare synthetic transient electromagnetic to measured data, which may be collected using different devices and receiver settings, I introduce the following normalizations to them:

- Transmitter current and
- Receiver moment

Thus, the displayed $\mathsf{E_x}$-field data are always given in V/m and the induced voltages are given in V/m^2. Please keep in mind, that the transmitter moment, which is either the length of a prolonged dipole, or the area of a (square) loop source is not included in those normalizations.

CHAPTER 2

Inversion theory

For the interpretation of measured data, we like to get an image of the reality through a model of more or less complexity. To get the model, the deterministic inverse process is applied to the prior model. It yields one or more a posteriori models which data describe the measurements within their uncertainty.

For a model, where the predicted data depends in a linear way on the model parameter and, if the model functional is twice differentiable, the inversion problem may be stated very simple: Take the first derivative, set it to zero, solve the equation for the model unknown and take a look at the second derivative at this point. If the second derivative is positive, you found the (global) minimum of the function, if it is negative, it is a maximum, etc.

If one concentrates on EM data, the model depends in a non-linear manner on the predicted data. Hence, we must apply non-linear optimization strategies which differ from the previous in the way that one has to linearize the whole problem and apply the same strategy, but iteratively.

2.1 The non-linear optimization problem

Assume a set of TEM measurements are taken at some points on the earth surface and the TEM data is influenced by the conduction properties in a certain region Ω in the subsurface. The measured data can be represented by the N-dimensional data vector $\mathbf{d} = (d_1, \ldots, d_N)^{\mathrm{T}}$ and Ω is some subset of the Euclidean space $\mathbb{R}^M$. To explain the measurements we synthesize a model for the parameter distribution. The synthetic model can be described through an M-dimensional vector $\mathbf{m} = (m_1, \ldots, m_M)^{\mathrm{T}}$ which is mostly just called *the model* or *parameterization*. The model is referring to the value of the parameter as well as it's discretization in the whole model domain. To reproduce or describe the real spatial conductivity distribution, or its reciprocal, $\mathbf{m}$ should be a continuous function in space like

$$\sigma(\mathbf{r}) \simeq \int_{\Omega} x(\mathbf{r}) \approx \sum_{i=1}^{M} m_i \theta_i(\mathbf{r}) \; . \tag{2.1}$$

In this equation, $\sigma(\mathbf{r})$ is the spatially and continuous conductivity distribution. The m_i's are serving as weighting coefficients for the discretizing basis functions θ_i. θ_i may be some sort of problem inherent function, like Chebychef or Legendre polynomials and may be defined as parallelepipeds as well. However, the basis functions $\theta_i(\mathbf{r})$ have to be selected such that the reality can be described with satisfying accuracy. Taking the basis functions as parallelepipeds they can form a grid in the Euclidean space. Therefore it seems convenient to introduce many basis functions to achieve a fine model discretization, for which the parameter distribution can be sought of a continuous function.

With the parametrized model we can formulate a direct problem to simulate the measurements taken. This can be described with the N-dimensional mapping $\mathbf{f}(\mathbf{m}) : \Omega \rightarrow \mathbb{R}^N$. If $\mathbf{y}$ is the residual vector between the measured data $\mathbf{d} \in \mathbb{R}^N$ and the model response, we can compare the model response with the measured data like:

$$y_i(\mathbf{m}) = d_i - f_i(\mathbf{m}) \; i = 1, \ldots, N \; . \tag{2.2}$$

For the interpretation of the data we are interested in the spatial distribution of the parameter $\mathbf{m}$ which model response fits to the measurements within the uncertainty of the data. Therefore we like to find a model which minimizes the difference between measured and predicted data.

For the general formulation of the minimization problem it is convenient to start with a functional which we like to minimize. This functional can be defined using an ℓ^n-norm of the residual vector $\mathbf{y}$ as:

$$\ell^n : \mathbb{R}^N \rightarrow \mathbb{R}, \; \ell^n(\mathbf{y}) = \sum_{i=0}^{\infty} |y_i|^n = \Phi(\mathbf{m}) = \min \; , \tag{2.3}$$

where the functional Φ is called an objective, or cost function.

The goal in a global optimization or minimization is, to find an element $\mathbf{m}_* \in \Omega$ such that

$$\forall \mathbf{m} \in \Omega : \quad \Phi(\mathbf{m}_*) \leq \Phi(\mathbf{m}) \, . \tag{2.4}$$

A feasible solution that minimizes the objective function, is called a stationary point of the cost function. But a stationary point may not be a global minimum. In fact, when the objective function does not present convexity, there may be several local minima and maxima. A local minimum $\mathbf{m}_*$ is defined as a point for which there exists some $\epsilon > 0$ so that for all $\mathbf{m} \in \Omega$ with

$$|\mathbf{m} - \mathbf{m}_*| \leq \epsilon \; , \tag{2.5}$$

the expression

$$\Phi(\mathbf{m}_*) \leq \Phi(\mathbf{m}) \tag{2.6}$$

holds. Whereas for a global minimum equation (2.4) is valid over the whole model domain.

If the optimization problem does not present convexity it may have several local minima and maxima and generally considered as *ill-posed problem.*

2.1.1 Line search

The line search is the simplest form of minimization technique. Nevertheless, it is a powerful tool to find the minimum of a function. It may be applied if the function is depending of one (1D line search) or just a few parameters.

Line search methods can be divided into exact and inexact line search methods. While inexact line searches deliver just the direction of the minimum by, generally three, function evaluations, exact line searches perform the minimum search within a given interval to find the minimum. Hence, the exact 1D line search can be treated as simple brute force minimization scheme. It can not fail if the problem is convex but diverges for non convex problems.

Algorithms performing an exact line search apply always the same policy: divide and conquer. Suppose, the initial interval $[a, c]$ includes the point where the function attains it's minimum at the value $\alpha \in [a, c]$. The initial interval then is reduced at each iteration to a smaller interval, already bracketing the minimum. We have so a series of encapsulated intervals (cf. fig. 2.1).

$$\alpha \in [a_i, c_i] \subset [a_{i-1}, c_{i-1}] \ldots \subset [a, c] \tag{2.7}$$

Suppose the minimum is bracketed by the starting interval, the exact line search will find it, regardless how long it will take. Overall, the performance of an exact line search is just linear, no matter if one takes bi-sectioning or the more sophisticated Fibonacci or golden section search, and thus often misjudged or not used at all. Another drawback is, that if the minimum is not included in the starting interval, the algorithm will fail after a long search.

The Wolfe conditions ([Wolfe, 1969; 1971] see below) for performing inexact line searches, can yield faster approximations to the minimum, but will not reach it in general. Moreover, if the problem is not convex, it may lead to wrong results. Formulas to perform an inexact line searches are given below.

Many people make a combination of sophisticated sectioning within an inexact line search: They choose a combination of the Fibonacci or the golden section search and terminate after one or both Wolfe conditions are satisfied.

Bisection method

Figure 2.1 describes how sectioning is applied while performing a line search. Here the graph of a simple quadratic function is plotted against the values of the data. Of course, the cost functional is generally a lot more complex and can bear many local minima. The mid point of every interval is denoted by $b_{i=0,1,\ldots}$. Note that the minimum of the function is, or should be, bracketed by the starting interval, which is not always the case. Thus, performing an exact line search, should be treated with special care.

The simplest form of an exact line search is the bisection method also known as dicotomic search. At the first iteration ($i = 0$) the interval $[a_i, c_i]$ is divided in two equal parts, $[a_i, b_i]$ and $[b_i, c_i]$, so that $b_i = \dfrac{a_i + c_i}{2}$. Then, after choosing $\epsilon > 0$, check for

$$\frac{\Phi(b_i + \epsilon)}{\Phi(b_i - \epsilon)} < 1 \; . \tag{2.8}$$

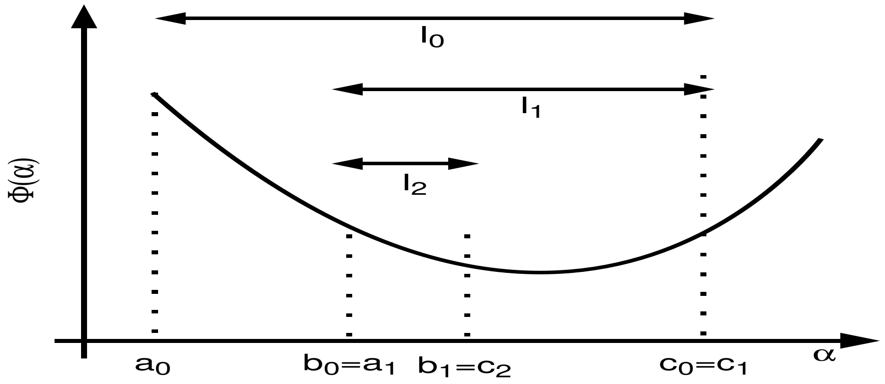


Figure 2.1: *Section search algorithm. The picture illustrates the sectioning of the interval which brackets the local minimum of the functional.*

If this equation is satisfied, repeat the whole process with the new interval $[a_i, b_i]$, otherwise repeat with $[b_i, c_i]$. Thus, the convergence to the approximation of the minimum is linear [Press et al., 1995].

Fibonacci search

A more sophisticated algorithm is the Fibonacci search, where at each iteration the length of the interval is chosen according to the Fibonacci rule:

$$l_{i-3} = l_{i-2} + l_{i-1} \tag{2.9}$$

Defining the initial intervals $l_0 = l_1 = (c_0 - a_0)$ follows

$$l_i = \frac{l_1 + l_{i-2}\epsilon}{l_i} \tag{2.10}$$

where l_i is the i-th number of the Fibonacci sequence. The number of function evaluations are again linear dependent on the number of iterations and the disadvantage of this method that ϵ must be chosen a priori.

Golden section search

The golden section search is one of the most famous sectioning methods. Given a triplet (a, b, c) that brackets the minimum, we choose a new point α_* that defines a new bracketing triplet (a, α_*, b) or (b, α_*, c) according to the rule:

$$\frac{\alpha_* - b}{c - a} = 1 - 2\frac{b - a}{c - a} \tag{2.11}$$

This implies that $|b - a| = |\alpha_* - c|$, and that at each iteration the interval is scaled of the same ratio ζ. Then we repeat the process with the new triplet. So the interval (a, c) is divided in two parts, a smaller and a larger, and the ratio between the whole interval and the larger is the same between the larger and the smaller, or in other words:

$$\frac{1}{\zeta} = \frac{\zeta}{1 - \zeta} \, , \tag{2.12}$$

giving for ζ the positive solution

$$\zeta = \frac{\sqrt{5}-1}{2} \ . \tag{2.13}$$

This fraction is known as the golden-mean or golden-section, whose aesthetic properties come from the ancient Pythagoreans.

Wolfe conditions

The Wolfe conditions [Wolfe, 1969; 1971], are a set of inequalities for performing inexact line-searches within iterative optimization techniques. Inexact line searches provide an efficient way of computing an acceptable step length $\delta\mathbf{m}_n = \mathbf{m}_{n+1} - \mathbf{m}_n$ that reduces the cost "sufficiently", rather than minimizing the cost over $\alpha_n \in \mathbb{R}$ exactly. The first inequality

$$\Phi(\mathbf{m}_n + \alpha_n \delta\mathbf{m}_n) \leq \Phi(\mathbf{m}_n) + c_1 \alpha_n \delta\mathbf{m}_n^{\mathrm{T}} \mathbf{g}_n \ , \tag{2.14}$$

is also known as Armijo rule [Gilbert & Nocedal, 1992] and the second

$$\mathbf{m}_n^{\mathrm{T}} \nabla\Phi(\mathbf{m}_n + \alpha_n \delta\mathbf{m}_n) \geq c_2 \delta\mathbf{m}_n^{\mathrm{T}} \mathbf{g}_n \ , \tag{2.15}$$

is also called the curvature condition, with $0 < c_1 < c_2 < 1$. For convenience, $\mathbf{g}_n = \nabla\Phi(\mathbf{m}_n)$ is the gradient of the cost functional evaluated for the n'th model iterate.

Equation (2.14) ensures that $\delta\mathbf{m}_n$ decreases the cost functional sufficiently, and equation (2.15) ensures that the slope of the function $\Phi(\mathbf{m}_n + \alpha_n \delta\mathbf{m}_n)$ at $\mathbf{m}_n$ is greater than c_2 times that at $\mathbf{m}_0$.

The Wolfe conditions are often used together with interval sectioning which arose following the Fibonacci rule or the golden section search to perform the inexact line search.

2.1.2 Gradient based optimization

If the parameter space has not the simple 1D dependency, iterative methods to find the (local or global) minimum should be applied. For the solution of the optimization problem of multidimensional parameter space, a large number of algorithms are proposed. Whereas the majority of commercially available solvers, are not capable of making a distinction between local optimal solutions and global optimal solutions, they will treat the former as actual solutions to the global problem. But the global optimization can be a very tedious task for non-linear $\mathbf{f}$ and is commonly avoided.

Consider the bisection method, which delivers a general solution of the one-dimensional problem. The application on non convex problems can be computationally prohibitive. Even for one dimensional problems it may diverge.

In the following I give a general introduction to solve the unconstrained minimization process by means of the gradient descent and the Newton method. Thereafter, the conjugated gradient descent method is described which applies the conjugate gradient (CG) approach of Hestenes & Stiefel [1952] to unconstrained optimization via a conjugation of steepest descent search directions. The Newton approach involves the computation of second derivatives of the cost functional. Because this can be computationally exhaustive, the linear equation which arises is often solved approximately. This is called quasi-Newton method which is briefly described.

Gradient descent (GD)

Gradient descent (GD) is a straight forward optimization algorithm which is also known as steepest descent, or the method of steepest descent. To find a local minimum of a functional it is convenient to take steps proportional to the negative of the gradient (or the approximate gradient) of the function at the current point.

It follows that, if

$$\mathbf{m}_1 = \mathbf{m}_0 - \alpha \nabla \Phi(\mathbf{m}_0) \tag{2.16}$$

for $\alpha > 0$ then $\Phi(\mathbf{m}_0) \geq \Phi(\mathbf{m}_1)$. Finding appropriate values for α is commonly done using an exact or inexact line search (e.g. the Armijo rule).

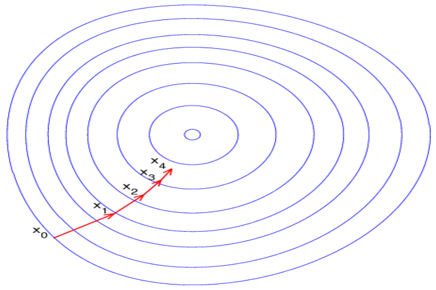


Figure 2.2: *Iterative process of gradient descent. The blue curves are the regions on which the value of Φ is constant.*

The iterative minimization process starts with a guess $\mathbf{m}_0$ for a local minimum of Φ, and, considering the sequence $\{\mathbf{m}_i : i = 0, 1, \ldots\}$ with $(\mathbf{g}_n = \Phi(\mathbf{m}_n))$

$$\mathbf{m}_{n+1} = \mathbf{m}_n - \alpha_n \mathbf{g}_n, \;\; n \geq 0 \tag{2.17}$$

we have

$$\Phi(\mathbf{m}_0) \geq \Phi(\mathbf{m}_1) \geq \Phi(\mathbf{m}_2) \geq \ldots. \tag{2.18}$$

Hence, the sequence $\Phi_n = \Phi(\mathbf{m}_n)$ converges to the desired (local) minimum. Note that the value of the step size α_n is allowed to change for every iteration.

To explain the iterative process of the optimization process using the gradient descent method, it is illustrated in picture 2.2. The blue lines indicating isolines of constant Φ. Here Φ is assumed to be defined on a simple 2D plane, and that its graph has a bowl shape. A red arrow originating at a point shows the direction of the negative gradient at that point. Note that the negative gradient at a point is orthogonal to the contour line which is going through that point. Following the red line from it's starting point at iteration 0, we see that GD leads to the bottom of the bowl. The bottom is the point where the value of the functional Φ attains it's minimum, which is a global minimum in this case.

Conjugate gradient descent and nonlinear conjugate gradient

The conjugate gradient descent (CGD) method generalizes the gradient descent method upon a modified Gram-Schmidt conjugation of the gradient search directions. According to Hestenes & Stiefel [1952], who called it conjugate directions, it is a general form of the conjugate gradient method. Following the formulation of Gilbert & Nocedal [1992] it extends for nonlinear optimization to the nonlinear conjugate gradient method (NLCG). Especially the nonlinear conjugate gradient methods are used for solving large scale optimization problems [Mackie & Madden, 1993; Rodi & Mackie, 2001; Commer, 2003; Commer & Newman, 2008], because they avoid the storage of matrices .

In the literature, the term conjugated gradient descent may not occur. It is chosen here to distinguish between the conjugate gradient method which is the method of choice for solving large scale linear systems and sometimes called conjugated gradient method Shewchuk [1994] which is used in unconstrained optimization. Although, the method is developed on the idea of Hestenes & Stiefel [1952] it is introduced to optimization problems for a different purpose and not used in the original form [Shewchuk, 1994].

Like steepest descent, the conjugate gradient descent method is an iterative methods of the form

$$\mathbf{m}_{n+1} = \mathbf{m}_n + \alpha_n \delta\mathbf{m}_n \tag{2.19}$$

where $\alpha_n > 0$ is a step size and $\delta\mathbf{m}_n$ is a search direction. The direction $\delta\mathbf{m}_n$ is recursively defined by

$$\delta\mathbf{m}_n = \begin{cases} -\mathbf{g}_n & n = 0 \\ -\mathbf{g}_n + \beta_n \delta\mathbf{m}_{n-1} \, , & n \geq 1 \end{cases} \tag{2.20}$$

where $\mathbf{g}_n$ is the gradient of Φ evaluated at $\mathbf{m}_n$ and β_n is a scalar. Note, how the gradient update is deviated from the gradient descent method which makes it a conjugated search direction. Additionally the step length can be different in every iteration without loosing conjugacy.

Three most prominent formulas for β are titled Fletcher-Reeves (FR), Polak-Ribiëre (PR), and Hestenes-Stiefel (HS) and given by

$$\beta_n^{\mathrm{FR}} = \frac{\delta\mathbf{m}_n^{\mathrm{T}} \delta\mathbf{m}_n}{\delta\mathbf{m}_{n-1}^{\mathrm{T}} \delta\mathbf{m}_{n-1}} \tag{2.21}$$

$$\beta_n^{\mathrm{PR}} = \frac{\delta\mathbf{m}_n^{\mathrm{T}} (\delta\mathbf{m}_n - \delta\mathbf{m}_{n-1})}{\delta\mathbf{m}_{n-1}^{\mathrm{T}} \delta\mathbf{m}_{n-1}} \tag{2.22}$$

$$\beta_n^{\mathrm{HS}} = \frac{\delta\mathbf{m}_n^{\mathrm{T}} (\delta\mathbf{m}_n - \delta\mathbf{m}_{n-1})}{\alpha_n \mathbf{m}_{n-1}^{\mathrm{T}} (\delta\mathbf{m}_n - \delta\mathbf{m}_{n-1})} \, . \tag{2.23}$$

They all deliver a modified Gram-Schmidt conjugacy scheme, which improves the convergence of the gradient descent search directions in a desired manner. Although, recent modifications of β investigated by Zhang et al. [2006], yield interesting convergence results as well.

If Φ is a strictly convex quadratic function and if α_n is the exact one-dimensional minimizer, equation (2.19) is called the linear conjugate gradient method [Gilbert & Nocedal, 1992]. On the other hand, equation (2.20) is called the nonlinear conjugate gradient method for general unconstrained optimization problems.

The main difference between CGD and gradient descent is illustrated in figure 2.3. Note, that the CGD search directions are always orthogonal on the previous and thus reducing iteration numbers. Subsequent search directions lose conjugacy requiring the search direction to be reset to the steepest descent direction at least every N iterations, or sooner if progress stops. However resetting every iteration turns the method into steepest descent.

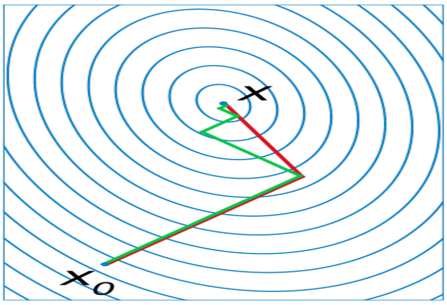


Figure 2.3: *CGD vs. gradient descent. The picture illustrates the difference in minimization between CGD (green) and gradient descent (red).*

If the line search is done exact, each iteration requires O(N) calculations [Gilbert & Nocedal, 1992] for convergence.

2.1.3 Newton's method

In mathematics, Newton's method is a well-known algorithm for finding roots of equations in one or more dimensions. It can also be used to find local maxima and local minima of functions by noticing that if a real number $\mathbf{m}_* \in \Omega$ is a stationary point of Φ, then $\mathbf{m}_*$ is a root of the derivative $\nabla\Phi(\mathbf{m}_*)$, and therefore one can solve for $\mathbf{m}_*$ by applying Newton's method to $\nabla\Phi(\mathbf{m}_0)$.
The Taylor expansion of $\Phi(\mathbf{m})$ at the point $\mathbf{m}_0$

$$\Phi(\mathbf{m}_0 + \delta\mathbf{m}_0) = \Phi(\mathbf{m}_0) + \nabla\Phi(\mathbf{m}_0)\delta\mathbf{m}_0 + \frac{1}{2}\nabla^2\Phi(\mathbf{m}_0)\delta\mathbf{m}_0^2 \tag{2.24}$$

attains its extrema when $\delta\mathbf{m}_0 = \mathbf{m} - \mathbf{m}_0$ solves the linear equation:

$$\boldsymbol{\mathcal{H}}_0\delta\mathbf{m}_0 = -\mathbf{g}_0 \ . \tag{2.25}$$

Where $\boldsymbol{\mathcal{H}}_0 = \nabla^2\Phi(\mathbf{m}_0)$ is called the Hessian matrix and $\mathbf{g}_0 = \nabla\Phi(\mathbf{m}_0)$ is the gradient. If $\boldsymbol{\mathcal{H}}_0$ is positive definite it is the (local) minimum.

Thus, provided that $\Phi(\mathbf{m})$ is a twice-differentiable function for $\mathbf{m} \in \Omega$ and the initial guess $\mathbf{m}_0$ is chosen close enough to $\mathbf{m}_*$, the sequence defined by

$$\mathbf{m}_{n+1} = \mathbf{m}_n - \boldsymbol{\mathcal{H}}_n^{-1}\mathbf{g}_n \ , \tag{2.26}$$

will converge towards $\mathbf{m}_*$.

Usually Newton's method is modified to include a small step size $\alpha > 0$ instead of $\alpha = 1$

$$\mathbf{m}_{n+1} = \mathbf{m}_n - \alpha\boldsymbol{\mathcal{H}}_n^{-1}\mathbf{g}_n \ . \tag{2.27}$$

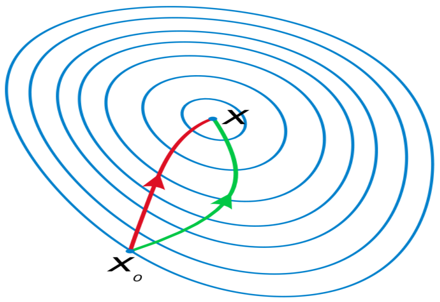


Figure 2.4: *The picture illustrates the difference in minimization between Newton's method (green) and gradient descent (red).*

This is often done to ensure that the Wolfe conditions are satisfied at each step $\mathbf{m}_n \rightarrow \mathbf{m}_{n+1}$ of the iteration.

The geometric interpretation of Newton's method is that at each iteration one approximates $\Phi(\mathbf{m})$ by a quadratic function around $\mathbf{m}_n$, and then takes a step towards the maximum/minimum of that quadratic function. If $\Phi(\mathbf{m})$ happens to be a quadratic function, then the exact extrema is found in one step.

Newton's method converges much faster towards a local maximum or minimum than gradient descent. This should be illustrated by picture 2.4. In fact, every local minimum has a neighborhood such that, if we start with $\mathbf{m}_0 \in \Omega$, Newton's method with step size $\alpha = 1$ converges quadratically (if the Hessian is invertible in that neighborhood). Because in GD the steps are always orthogonal to the isolines of the graph, it needs more iterations than Newton's method to converge.

Finding the inverse of the Hessian is an expensive operation, and can be prohibitive in most cases, so the linear equation

$$\boldsymbol{\mathcal{H}}\delta\mathbf{m} = -\mathbf{g} \ , \tag{2.28}$$

is often solved approximately using Quasi-Newton methods, where an approximation for the Hessian is used instead. The existing Quasi-Newton Methods are described in more detail below.

Note, that if the Hessian is close to a non-invertible matrix, the inversion of the Hessian can be numerically unstable and the solution may diverge.

Quasi-Newton methods

Quasi-Newton methods are a generalization of the secant method to find the root of the first derivative for multidimensional problems. The main attraction is that the Hessian matrix of second derivatives of the function to be minimized does not need to be computed. Therefore he QN method gains an increasing popularity among geophysics who aim for large scale inverse problems. The Hessian is updated by analyzing successive gradient vectors instead. The most common quasi-Newton algorithms are currently the SR1 formula (for symmetric rank one) and the widespread BFGS method, that was suggested independently by Broyden, Fletcher, Goldfarb, and Shanno, in 1970 [Dennis & Schnabel, 1996].

As in the Newton's method, one uses the second order approximation to find the minimum of a function Φ. From equation (2.24) one can derive the Taylor series of the gradient itself as

$$\nabla\Phi(\mathbf{m}_0 + \delta\mathbf{m}_0) = \mathbf{g}_0 + \boldsymbol{\mathcal{H}}_0\delta\mathbf{m}_0, \tag{2.29}$$

which is called secant equation.

Solving equation (2.29) for $\nabla\Phi(\mathbf{m}_0 + \delta\mathbf{m}_0) = 0$ provides the Newton step (eq. (2.26)). Various methods are used to find the solution to the secant equation which is symmetric ($\boldsymbol{\mathcal{H}}^{\mathrm{T}} = \boldsymbol{\mathcal{H}}$) and closest to the current approximate value $\boldsymbol{\mathcal{H}}_n$ according to some metric $\min_B \|\boldsymbol{\mathcal{H}} - \boldsymbol{\mathcal{H}}_n\|$. An approximate initial value of $\boldsymbol{\mathcal{H}}_0 = \mathbf{I}$ is sometimes sufficient to achieve convergence. The unknown $\mathbf{m}_n$ is updated applying the Newton's step which is calculated using the current approximate Hessian matrix $\boldsymbol{\mathcal{H}}_n$.

The whole process may be summarized by the following list:

- Compute $\delta\mathbf{m}_n = -\alpha_n\boldsymbol{\mathcal{H}}_n^{-1}\mathbf{g}_n$, with α chosen to satisfy the Wolfe conditions.
- Update the solution $\mathbf{m}_{n+1} = \mathbf{m}_n + \delta\mathbf{m}_n$.
- Compute the gradient at the new point $\mathbf{g}_{n+1}$, and
- $\mathbf{s}_n = \mathbf{g}_{n+1} - \mathbf{g}_n$, is used to update the Hessian $\boldsymbol{\mathcal{H}}_{n+1}$, or directly its inverse using the Sherman-Morrison formula (2.30)

Given an invertible square matrix $\mathbf{A}$ and the vectors $\mathbf{u}$ and $\mathbf{v}$, such that $1 + \mathbf{v}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{u} \neq 0$, the Sherman-Morrison formula states that [Dennis & Schnabel, 1996]:

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^{\mathrm{T}})^{-1} = \mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{u}\mathbf{v}^{\mathrm{T}}\mathbf{A}^{-1}}{1 + \mathbf{v}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{u}} \ . \tag{2.30}$$

Here, $\mathbf{u}\mathbf{v}^{\mathrm{T}}$ is the dyadic product of the two vectors $\mathbf{u}$ and $\mathbf{v}$.

The most popular update formulas for quasi-Newton methods are given in the table 2.1.

Table 2.1: *Popular update formulas of the Hessian matrix ($\boldsymbol{\mathcal{H}}_n$) or its inverse for quasi-Newton methods.*

Method	$\boldsymbol{\mathcal{H}}_n$	$\boldsymbol{\mathcal{H}}_n^{-1}$
DFP	$\left(\mathbf{I}-\frac{\mathbf{s}_n\delta\mathbf{m}_n^{\mathrm{T}}}{\mathbf{s}_n^{\mathrm{T}}\delta\mathbf{m}_n}\right)\boldsymbol{\mathcal{H}}_n\left(\mathbf{I}-\frac{\delta\mathbf{m}_n\mathbf{s}_n^{\mathrm{T}}}{\mathbf{s}_n^{\mathrm{T}}\delta\mathbf{m}_n}\right)+\frac{\mathbf{s}_n\mathbf{s}_n^{\mathrm{T}}}{\mathbf{s}_n^{\mathrm{T}}\delta\mathbf{m}_n}$	$\boldsymbol{\mathcal{H}}_n^{-1}+\frac{\delta\mathbf{m}_n\delta\mathbf{m}_n^{\mathrm{T}}}{\mathbf{s}_n^{\mathrm{T}}\delta\mathbf{m}_n}-\frac{\boldsymbol{\mathcal{H}}_n^{-1}\mathbf{s}_n\mathbf{s}_n^{\mathrm{T}}\boldsymbol{\mathcal{H}}_n^{-1\mathrm{T}}}{\mathbf{s}_n^{\mathrm{T}}\boldsymbol{\mathcal{H}}_n^{-1}\mathbf{s}_n}$
BFGS	$\boldsymbol{\mathcal{H}}_n+\frac{\mathbf{s}_n\mathbf{s}_n^{\mathrm{T}}}{\mathbf{s}_n^{\mathrm{T}}\delta\mathbf{m}_n}-\frac{\boldsymbol{\mathcal{H}}_n\delta\mathbf{m}_n(\boldsymbol{\mathcal{H}}_n\delta\mathbf{m}_n)^{\mathrm{T}}}{\delta\mathbf{m}_n^{\mathrm{T}}\boldsymbol{\mathcal{H}}_n\delta\mathbf{m}_n}\left(\mathbf{I}-\frac{\mathbf{s}_n\delta\mathbf{m}_n^{\mathrm{T}}}{\mathbf{s}_n^{\mathrm{T}}\delta\mathbf{m}_n}\right)^{\mathrm{T}}$	$\boldsymbol{\mathcal{H}}_n^{-1}\left(\mathbf{I}-\frac{\mathbf{s}_n\delta\mathbf{m}_n^{\mathrm{T}}}{\mathbf{s}_n^{\mathrm{T}}\delta\mathbf{m}_n}\right)+\frac{\delta\mathbf{m}_n\delta\mathbf{m}_n^{\mathrm{T}}}{\mathbf{s}_n^{\mathrm{T}}\delta\mathbf{m}_n}$
Broyden	$\boldsymbol{\mathcal{H}}_n+\frac{\mathbf{s}_n-\boldsymbol{\mathcal{H}}_n\delta\mathbf{m}_n}{\delta\mathbf{m}_n^{\mathrm{T}}\delta\mathbf{m}_n}\delta\mathbf{m}_n^{\mathrm{T}}$	
SR1	$\boldsymbol{\mathcal{H}}_n+\frac{(\mathbf{s}_n-\boldsymbol{\mathcal{H}}_n\delta\mathbf{m}_n)(\mathbf{s}_n-\boldsymbol{\mathcal{H}}_n\delta\mathbf{m}_n)^{\mathrm{T}}}{(\mathbf{s}_n-\boldsymbol{\mathcal{H}}_n\delta\mathbf{m}_n)^{\mathrm{T}}\delta\mathbf{m}_n}$	$\boldsymbol{\mathcal{H}}_n^{-1}+\frac{(\delta\mathbf{m}_n-\boldsymbol{\mathcal{H}}_n^{-1}\mathbf{s}_n)(\delta\mathbf{m}_n-\boldsymbol{\mathcal{H}}_n^{-1}\mathbf{s}_n)^{\mathrm{T}}}{(\delta\mathbf{m}_n-\boldsymbol{\mathcal{H}}_n^{-1}\mathbf{s}_n)^{\mathrm{T}}\mathbf{s}_n}$

Currently, the only available QN method which seems suitable for large scale inversion purposes is the L-BFGS method published by Liu & Nocedal [1989]. It maintains a comfortable method to store the updates for the Hessian matrix which can be challenging, because every one of the former updates are needed to perform the next one. As a drawback in terms of computational load, this method requires a cheap gradient calculation. This is due to the check of the curvature condition (eq. (2.15)) which ensures the positive definiteness of the updated Hessian at every iteration step.

The SR1 formula does not guarantee the update matrix to maintain positive definiteness and can be used for this problems. Yet, because of the great difficulties in the implementation of the SR1 formula, it was not tested within this work. The Broyden's method does not require the update matrix to be symmetric nor that it is positive definite. Moreover it is used to find the root of a general system of equations (rather than the gradient) by updating the Jacobian (rather than the Hessian) matrix. Yet, because there is no recursive inverse formula for the Broyden update available, it is discarded here as well so that the formal comparison above is given here for sake of completeness.

2.2 Least squares formulation

The above mentioned minimization techniques can be applied in general unconstrained optimization and to any ℓ-norm of the cost functional. For statistical reasons, which I describe briefly in the following, it makes sense to formulate the optimization problem in a least squares sense. The method of least squares or ordinary least squares is normally used to solve overdetermined systems.

Carl Friedrich Gauss is credited with developing the fundamentals of the basis for least-squares analysis in 1795. In 1829, Gauss was able to state that the least-squares approach to regression analysis is optimal in the sense that in a linear model where the errors have a mean of zero and are uncorrelated, the best linear unbiased estimators of the coefficients is the least-squares estimator. This result is known as the Gauss-Markov theorem. For more statistical background to this, the reader is referred to the book of Tarantola [2005].

2.2.1 Data covariance

Because we can only measure to a certain degree of exactness, the measured data itself is contaminated with noise and may be biased as well. The errors have not to be normal distributed nor to be linear independent, but only uncorrelated - a weaker condition, nor are they assumed to be identically distributed, but only having zero mean and equal variances. To take this error into account, it is common to introduce the inverse of a weighting matrix $\mathbf{C}_d^{-1}$, which is also known as the data-covariance-matrix. If the measurements are uncorrelated, $\mathbf{C}_d^{\frac{1}{2}}$ is a diagonal matrix with the standard deviation of each measured data point δd_i on the main diagonal

$$\begin{aligned} \mathbf{C}_d^{\frac{1}{2}} &= \operatorname{diag}\{\delta d_i\} && (2.31)\\ \mathbf{C}_d^{-1} &= \left(\operatorname{diag}\{\delta d_i\}\right)^{-2} \;. && (2.32) \end{aligned}$$

In general, if the assumptions of uncorrelated measurement error is not fulfilled, the data covariance matrix has not a diagonal shape. Taking the data covariance into account and using the ℓ^2-norm of the residual vector, equation (2.3) becomes

$$\Phi(\mathbf{m}) = \mathbf{y}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{y} = \sum_{i=1}^{N}\left(\frac{d_i - f_i(\mathbf{m})}{\delta d_i}\right)^2 \tag{2.33}$$

2.2.2 Gauss-Newton

The Gauss-Newton algorithm (GNA) is a method used to solve non-linear least squares problems and can be seen as a modification of Newton's method for finding a minimum of a function. Unlike Newton's method, the Gauss-Newton algorithm can only be used to minimize a sum of squared function values, but it has the advantage that second derivatives, which can be challenging to compute, are not required. Thus it may be seen as Quasi-Newton method, too.

The Gauss-Newton algorithm can be derived by the Taylor expansion of equation (2.33) for the current model vector $\mathbf{m}_n$. The stationary point of this expansion follows analogue to equation (2.25) with

$$\begin{aligned} \nabla\Phi(\mathbf{m}_n) &= \nabla\left(\left[\mathbf{d} - \mathbf{f}(\mathbf{m}_n)\right]^{\mathrm{T}}\mathbf{C}_d^{-1}\left[\mathbf{d} - \mathbf{f}(\mathbf{m}_n)\right]\right)\\ &= -\mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\left[\mathbf{d} - \mathbf{f}(\mathbf{m}_n)\right] - \mathbf{S}_n\mathbf{C}_d^{-1}\left[\mathbf{d} - \mathbf{f}(\mathbf{m}_n)\right]^{\mathrm{T}}\\ &= -2\mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\left[\mathbf{d} - \mathbf{f}(\mathbf{m}_n)\right]\\ &= -2\mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{y}_n \;, && (2.34) \end{aligned}$$

and the approximated second derivatives

$$\nabla^2\Phi(\mathbf{m}_n)\delta\mathbf{m}_n \approx 2\mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S}_n \;, \tag{2.35}$$

where $\mathbf{S}_n$ is the $N \times M$ Jacobian matrix, also called sensitivity matrix or just sensitivities, of partial derivatives of the model response evaluated at $\mathbf{m}_n$:

$$\mathbf{S}_n = \begin{pmatrix} \frac{\partial f_1(\mathbf{m}_n)}{\partial m_1} & \dots & \frac{\partial f_1(\mathbf{m}_n)}{\partial m_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_N(\mathbf{m}_n)}{\partial m_1} & \dots & \frac{\partial f_N(\mathbf{m}_n)}{\partial m_M} \end{pmatrix} . \tag{2.36}$$

Note, that $\mathbf{S}$ is full populated.

The task of finding proper $\delta\mathbf{m}_n$ minimizing the sum of squares of the right-hand side in every iteration is a linear least squares problem, which can be solved explicitly, yielding the normal equations in the algorithm.

Starting with an initial guess for the minimum, the method proceeds by the iterations

$$\mathbf{m}_{n+1} = \mathbf{m}_n + \delta\mathbf{m}_n, \tag{2.37}$$

with the increment $\delta\mathbf{m}_n$ satisfying the normal equation

$$\left(\mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S}_n\right)\delta\mathbf{m}_n = \mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{y}_n \tag{2.38}$$

Note, that the minus from the gradient of the cost functional vanishes because of the ℓ_2-norm of the residuum vector. The assumption $N \leq M$ in the algorithm statement is necessary, as otherwise the matrix $\mathbf{S}^{\mathrm{T}}\mathbf{S}$ is not invertible and the normal equations cannot be solved directly.

Levenberg-Marquardt algorithm

The Levenberg-Marquardt-algorithm (LMA) [Levenberg, 1944; Marquardt, 1963] interpolates between the GNA and the method of gradient descent. The LMA is more robust than the GNA, which means that in many cases it finds a solution even if it starts not close to the final minimum. On the other hand, for well-behaved functions and reasonable starting parameters, the LMA tends to be a bit slower than the GNA.

It is very popular among curve-fitting algorithm and the most software with generic curve-fitting capabilities provide an implementation of it. It was first used by Vozoff & Jupp [1975] for the inversion of geophysical data sets.

Levenberg [1944] contribution to the least squares optimization process is to replace equation (2.38) by a "damped version"

$$\left(\mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S} + \lambda\mathbf{I}\right)\delta\mathbf{m} = \mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{y}\,, \tag{2.39}$$

where the (non-negative) damping factor λ can be adjusted at each iteration. A similar damping factor appears in Tikhonov regularization [Tikhonov & Arsenin, 1977], which is used to solve linear ill-posed problems, as well as in ridge regression.

Like other numeric minimization algorithms, the LMA is an iterative procedure and one has to provide an initial guess for the parameter vector. In many cases, an uninformed standard guess like a homogeneous model parameterization will work fine; in other cases, the algorithm converges only if the initial guess is already somewhat close to the final solution [Meju, 1994]. If reduction of Φ is rapid, a smaller value can be used which brings smaller eigenvalues of the linear system into account. This also brings the algorithm closer to the GNA, whereas if an iteration gives insufficient reduction in the residual λ can be increased giving a step closer to the gradient descent direction.

Levenberg's algorithm has the disadvantage that if the value of damping factor λ is large compared to the singular values of $\mathbf{S}$, inverting $\mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S} + \lambda\mathbf{I}$ is not used at all.

Marquardt [1963] provided the insight that we can scale each component of the gradient according to the curvature so that there is larger movement along the directions where the gradient is smaller. This avoids slow convergence in the direction of small gradient. Therefore, Marquardt replaced the identity matrix $\mathbf{I}$ with the diagonal of the approximated Hessian matrix $\boldsymbol{\mathcal{H}} \approx \mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S}$, resulting in the original Levenberg-Marquardt algorithm

$$\left(\mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S} + \lambda\,\mathrm{diag}\left(\mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S}\right)\right)\delta\mathbf{m} = \mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{y}\ , \tag{2.40}$$

which is not commonly used in geophysical applications.

2.3 Regularization

Large scale inversion problems generally have more free model parameters than data. In many cases the model domain shows both well resolved and poorly resolved model parameters. Consequently, there exist a lot of models fitting the data in a specified manner. The use of the above described minimization scheme would probably lead to highly oscillating models with huge parameter contrasts. Therefore, it becomes necessary to eliminate the ambiguity of the problem. The main advantage of regularization is already introduced within the LMA where a factor was added to the main diagonal of $\mathbf{S}^{\mathrm{T}}\mathbf{S}$ which leads to a stabilized solution of the inverse process. Such techniques are denoted with the term regularization, or, more general, a Tikhonov regularization.

Overall the technique of the LMA, which is a special case of Tikhonov regularization, can be denoted as minimizing not only the fit, but the energy of the model or model update as well.

Consider the minimization of

$$\Phi = \Phi^d + \lambda\Phi^m \rightarrow \min\ , \tag{2.41}$$

where $\Phi^d = \mathbf{y}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{y}$ denotes the cost functional of the data and Φ^m is a cost functional of the model. The Lagrange parameter λ is a parameter who weights between the solely data based functional Φ^d and the fictitious model functional Φ^m. The introduction of Φ^m not just designs a class of preferred models, but stabilizes the inversion process. Because of its stabilizing nature to the inversion process, Zhdanov [2002] calls Φ^m just a stabilizer, which is a convenient term for it.

This should be illustrated by the following considerations. Suppose $\Phi_0^m = \|\delta\mathbf{m}_0\|_2^2$, the functional derivative of the Taylor expansion of Φ_0 around the starting model $\mathbf{m}_0$ is given by

$$\begin{aligned}\nabla\Phi_0 = 0 &= \nabla\Phi_0^d + \lambda\nabla\Phi_0^m \\ &= -2\mathbf{S}_0^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{y}_0 + \boldsymbol{\mathcal{H}}_0\delta\mathbf{m}_0 + 2\lambda\delta\mathbf{m}_0\ .\end{aligned} \tag{2.42}$$

Solving for the model update $\delta\mathbf{m}_0$ with $\boldsymbol{\mathcal{H}}_0 \approx 2\mathbf{S}_0^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S}_0$ leads to the normal equation

$$\left(\mathbf{S}_0^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S}_0 + \lambda\mathbf{I}\right)\delta\mathbf{m}_0 = \mathbf{S}_0^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{y}_0 \ , \tag{2.43}$$

which is exactly the equation given in (eq. (2.39)).

The corresponding functional denotes the quadratic length of the update vector, which is kept small to prevent oscillations in the model update. Thus, the LMA penalizes changes of the model update within the minimization process which are "too big". What "too big" means is controlled by the Lagrange parameter λ. From this idea the term damping originates, which is often used in connection with locally constrained inversion [Loke & Barker, 1996].

Because of the inherit properties and the notional nature of the stabilizing functional a lot of different stabilizers are used within different minimization schemes. Constraining the model update in every iteration step (e.g. LMA) is often referred to as *local regularization*, whereas the term *global regularization* is used if the model itself is constraint.

One of the most common stabilizers which gains a lot of attraction during the last decades of geophysical model appraisal is known as *Occam's razor*. It is used to generate smooth models by a global regularization of the model itself. The smooth model inversion approach has different advantages and is also used within this work, thus it is further introduced in the following. For a compilation of different stabilizers the reader is referred to the book of Zhdanov [2002].

The drawback of applying additional stabilizers may yield a biased inverse solution towards a model which is only driven by the stabilizer. Where the unregularized problem may not converge. For example, the application of smooth model inversion is known to generate inverse solutions with over accentuated model cells at the boundary of the model. The problem is now how the application of the stabilizer is presented in the inverse solution and how one should interpret the result. Therefore the resolution analysis of the regularized inverse problem has to be discussed.

2.3.1 Smoothness constraints

The principle of generating smooth models is based on the concept that the monk William of Occam stated in the 14th century:

> "*One should not increase, beyond what is necessary, the number of entities required to explain anything.*"

Thus, when comparing two models which describe the underlying mechanisms well, the simplest is to be preferred. It was introduced to the geophysical community by the smooth model inversion of Constable et al. [1987] and is mostly assumed with it. But what is simple is surely a matter of subjectivity. Besides the minimization of parameter gradients in the model, the number of occurring parameter jumps can be treated as function to be minimized. Such approaches are referred to as total variation [Vogel, 2002] or focused inversion [Portniaguine & Zhdanov, 1999] and introduce a non-linear model functional. This "simplicity before complexity" rule is not just applied in geophysics, but it is used already within different fitting algorithms, like cubic spline interpolation or is applied in image enhancement algorithms [Press et al., 1995].

Actually, every global regularization scheme is following Occam's principle defining its own interpretation of "simplicity". To the contrary Occam's principle is restricted to global regularizing methods, even though the desired model property (e.g. smoothness) can also be applied to the model update in a local regularization scheme. However, even if all model updates are relatively smooth, it is not evident that the final model has to be smooth as well.

From the information point of view, regularization supplements the information provided by the data. The additional information is either obtained by a-priori knowledge or generated by expectations to the investigated structures. Amongst all models fitting the data equivalently, the one with the smallest regularization is favored.

The simplest method to be implemented is the application of the identity matrix not on the model update, but on the model itself

$$\Phi^m = \|\mathbf{m} - \hat{\mathbf{m}}\|_2^2 \tag{2.44}$$

It is successfully used in cases, where some a-priori model $\hat{\mathbf{m}}$ is known from other investigations or geological concepts. However, using a more or less regular model parameterization, it turns out that well resolved model cells in the neighborhood of the sensors are over-accentuated while badly resolved regions, e.g. in deep layers, show much too less structure.

One way to circumvent this naturally occurring effect is to create a parameterization that takes into account the physical resolution properties. However, resolution is not known before inversion, so the realization becomes a trial-and-error procedure. In practice, an idea of geometrical resolution can often be derived by experience. Another disadvantage of using the identity matrix as stabilizer is the fact, that the model update vector is implicitly expected to possess a Gaussian distribution with zero mean. For many cases, for example large bodies of constant resistivity, this assumption leads to unreasonable artifacts. However, the method is suitable for small parameter inversions or in the absence of geometrical relations, e.g. for the subspace methods of Oldenburg et al. [1993] or Siripunvaraporn & Egbert [2000].

In terms of matrix inversion it is reasonable to write Φ^m as squared norm of a product of a constraint matrix. Consider the singular matrix of the form

$$\mathbf{C}_m \in \mathrm{I\!R}^{M\times M} = \begin{pmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -1 & 1 \\ 0 & \cdots & 0 & 0 & 0 \end{pmatrix} \tag{2.45}$$

which can be sought of a kind of operator matrix. Note, that the last entry is zero reflecting the one-dimensional boundary of our discretized model. This can be implemented in other ways instead, e.g. to satisfy Neumann or Dirichlet boundary conditions at the model boundary, or by a natural extension of the model to a homogeneous halfspace, where entries which are lying at the boundary are compared to the surrounding halfspace. For simplicity this notation is used.

Application of $\mathbf{C}$ to a discrete model yields a finite difference vector of the form

$$\mathbf{Cm} = \begin{pmatrix} m_2 - m_1 \\ \vdots \\ m_{i+1} - m_i \\ \vdots \\ m_M - m_{M-1} \\ 0 \end{pmatrix} , \tag{2.46}$$

which is the finite difference approximation to the first order directional derivative in one dimension. Thus, $\mathbf{Cm}$ can be treated as the gradient of the model structure or values, respectively.

If the matrix (eq. (2.45)) is applied to the model, or the difference between the model and a reference model $\hat{\mathbf{m}}$

$$\Delta\hat{\mathbf{m}} = \mathbf{m} - \hat{\mathbf{m}} \tag{2.47}$$

which may be known from a priori knowledge, a functional which evaluates the finite difference approximation of

$$\Phi^m := \int_\Omega \|\nabla\Delta\hat{\mathbf{m}}\|_2^2 \; \mathrm{d}V \approx \sum_{i=1}^{M} \left[(m_{i+1} - \hat{m}_{i+1}) - (m_i - \hat{m}_i)\right]^2 , \tag{2.48}$$

evaluates the *roughness* of the model.

Using the more compact matrix notation, equation (2.48) simplifies to

$$\Phi^m = \|\mathbf{C}(\Delta\hat{\mathbf{m}})\|_2^2 = (\mathbf{m} - \hat{\mathbf{m}})^{\mathrm{T}}\mathbf{C}^{\mathrm{T}}\mathbf{C}(\mathbf{m} - \hat{\mathbf{m}}) . \tag{2.49}$$

Note, that the matrix $\mathbf{C}$ as defined above just accounts for the neighboring model cells in one direction. If it is stored following a regular grid, this may be the x-direction, thus the matrix (eq. (2.45)) is denoted with $\mathbf{C}_x$. The subscript denotes the direction of the gradient. According to this, the gradient matrices for all directions can be assembled like equation (2.45) and are further referred to with $\mathbf{C}_y$ and $\mathbf{C}_z$. By introducing weights for the different orientations a_x, a_y and a_z, the total regularization matrix can be calculated:

$$\mathbf{C} = a_x\mathbf{C}_x + a_y\mathbf{C}_y + a_z\mathbf{C}_z . \tag{2.50}$$

The weights can be used to enforce lateral or vertical model changes, which is useful in layered media or to discriminate vertical boundaries of geological units. The model may be further divided through the distance of the model cells which would further weight the differences between the model parameters with respect to their spatial distribution. This can be sometimes advantageous if the discretization is not uniform.

From matrix (eq. (2.50)) the second order derivative matrix can be formed by quadratic application of the first order matrix

$$\mathbf{C}^{2nd} = \mathbf{C}^{1st^{\mathrm{T}}}\mathbf{C}^{1st} . \tag{2.51}$$

Note that, if $\mathbf{C}^{2nd}$ is weighted according to the distance of the cells, it is numerically equivalent to the Laplace operator of the model.

As stated previously, assumptions have to be made at the boundaries of the model. This corresponds to the boundary conditions for the discretization of partial differential equations. The use of homogeneous Neumann conditions, e.g., works if the model parameter were continued outside the model boundaries (e.g. homogeneous halfspace). As the roughness is required to be kept small, this can sometimes lead to artificial structures near the boundaries. One can avoid this by formulating derivatives at one boundary only with respect to the other directions. It has to be noted that the use of Neumann conditions at all boundaries leads to singular matrices, as known from the solution of elliptic boundary value problems [Günther, 2004]. Generally it has been observed, that models tend to show over-accentuated structures at the boundary cells. Since the smoothness is applied to the model changes, it seems reasonable to use homogeneous Dirichlet boundary conditions for the second order smoothness, which proves to yield acceptable results in most cases.

For the minimization of Φ^m one must calculate the gradient

$$\nabla\Phi^m = 2\mathbf{C}^{\mathrm{T}}\mathbf{C}\Delta\hat{\mathbf{m}} \tag{2.52}$$

and the second derivative

$$\nabla^2\Phi^m = 2\mathbf{C}^{\mathrm{T}}\mathbf{C} \tag{2.53}$$

of the model functional.

Applied to a least squares formulation this leads to the minimization of

$$\left(\mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S} + \lambda\mathbf{C}^{\mathrm{T}}\mathbf{C}\right)\delta\mathbf{m} = \mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{y} - \lambda\mathbf{C}^{\mathrm{T}}\mathbf{C}\Delta\hat{\mathbf{m}}\ . \tag{2.54}$$

The smoothness operator on the right hand side can be applied to the initial model or the one of the previous iteration.

$$\left(\mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S}_n + \lambda\mathbf{C}^{\mathrm{T}}\mathbf{C}\right)\delta\mathbf{m}_n = \mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{y}_n - \lambda\mathbf{C}^{\mathrm{T}}\mathbf{C}\Delta\hat{\mathbf{m}}_{n-1} \tag{2.55}$$

Note, that taking $\mathbf{C} = \mathbf{I}$ would lead to a similar model update like LMA but for a global regularization or damping.

Probabilistic formulation for regularized models

In terms of a probabilistic approach, the matrix $\mathbf{C}$ or $\mathbf{C}^{\mathrm{T}}\mathbf{C}$ it can be treated as prior model covariance matrix (the variance of the prior model)

$$\mathbf{C}^{\mathrm{T}}\mathbf{C} = \mathbf{C}_m^{-1}\ . \tag{2.56}$$

This seems a bit odd, because neither $\mathbf{C}$ nor $\mathbf{C}^{\mathrm{T}}\mathbf{C}$ are invertible matrices. But we never have to form $\left(\mathbf{C}^{\mathrm{T}}\mathbf{C}\right)^{-1}$ to get the inverse of the model covariance. Thus the cost functional (eq. (2.41)) can be written

$$\Phi = (\mathbf{d} - \mathbf{f}(\mathbf{m}))^{\mathrm{T}}\mathbf{C}_d^{-1}(\mathbf{d} - \mathbf{f}(\mathbf{m})) + \lambda(\mathbf{m} - \hat{\mathbf{m}})^{\mathrm{T}}\mathbf{C}_m^{-1}(\mathbf{m} - \hat{\mathbf{m}})\ . \tag{2.57}$$

The minimization of equation (2.57) leads to the formulation of the normal equations

$$\left(\mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S} + \lambda\mathbf{C}_m^{-1}\right)\delta\mathbf{m} = \mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{y} - \lambda\mathbf{C}_m^{-1}\Delta\hat{\mathbf{m}}\ . \tag{2.58}$$

which is similar to the one of Tarantola [2005]. This formulation is particularly useful for the formulation of the resolution capabilities of the inverse solution.

2.4 Resolution analysis

Practical decisions are often made based on the subsurface images obtained by inverting geophysical data. The solution to the invers problem is a model, a simplified concept of the reality, which is rarely true. It is one of many possible models fitting the data within their uncertainty. Therefore it is important to understand the resolution of the image, which is a function of several factors, including the underlying geophysical experiment, noise in the data, prior information and the ability to model the physics appropriately. An important step towards interpreting the image is to quantify how much of the solution is required to satisfy the data observations and how much exists solely due to the regularization used to stabilize the solution.

A procedure to identify the regions that are not constrained by the data would help when interpreting the image. For linear inverse problems this procedure is well established (e.g. [Backus & Gilbert, 1968; 1970]), but for nonlinear problems the procedure is more complicated. Oldenburg & Li [1999] supposed a practical and easy to use approach defining the depth of investigation index. Approaches with a theoretical formulation for resolution of nonlinear problems in geophysics were published in [Alumbaugh & Newman, 2000; Tarantola, 2005; Miller & Routh, 2007].

Insight into the resolution analysis can be gained by considering a linear inverse problem. In the quasi-linear problems where least squares approaches where applied, the result from linear resolution remains approximately true but has to be used with caution.

2.4.1 Resolution of linear problems

For linear inversion theory, resolution has been described in detail (e.g. Backus & Gilbert [1968]; Menke [1984]; Tarantola [2005]). Assuming a linear problem, where the observed data $\mathbf{d}$ is predicted by a linear mapping of the true model $\mathbf{m}_*$

$$\mathbf{d} = \mathbf{S}\mathbf{m}_* + \mathbf{n} \ , \tag{2.59}$$

where $\mathbf{n}$ denotes a noise error (e.g. the noise level of the measurement) and $\mathbf{S}$ denotes a linear mapping of the model into data space. For simplicity we can assume that $\mathbf{S}$ equals the sensitivity matrix. The least squares solution for equation (2.59) is then given in one single step by

$$\mathbf{m} = \boldsymbol{\mathcal{H}}^{-1}\mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S}\mathbf{m}_* - \boldsymbol{\mathcal{H}}_n^{-1}\mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{n} \ , \tag{2.60}$$

where the Hessian matrix is approximated through $\boldsymbol{\mathcal{H}} = \mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S} + \lambda\mathbf{C}_m^{-1}$.

Following Backus & Gilbert [1968] the equation (2.60) can be interpreted as a linear mapping

$$\mathbf{m} = \mathbf{R}\mathbf{m}_* - \boldsymbol{\mathcal{H}}_n^{-1}\mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{n} \ , \tag{2.61}$$

of the resolution matrix

$$\mathbf{R} = \boldsymbol{\mathcal{H}}^{-1}\mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S} \ , \tag{2.62}$$

which combines the forward and the inverse mapping. It serves as a filter function transferring the reality into our model estimate. Ideally, $\mathbf{R}$ would be the identity matrix, which corresponds to perfect resolution. Finding this, the solution is generally dominated by noise effects.

For gradient based unconstrained optimization schemes, the Hessian matrix is approximated by $\boldsymbol{\mathcal{H}} = \mathbf{I}$. Thus, the resolution matrix reduces to

$$\mathbf{R} = \mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S} \ . \tag{2.63}$$

The resolution matrix for unconstrained gradient based optimizations is easy to compute if the sensitivity matrix is available. For constrained gradient based optimization schemes, equation (2.63) changes to

$$\mathbf{R} = \mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S} - \lambda\mathbf{C}_m^{-1}(\mathbf{m}_n - \hat{\mathbf{m}}) \ , \tag{2.64}$$

which is slightly different to equation (2.63), depends on the Lagrange parameter λ and is thus regularization dependent.

In contrast, if one applies constrained optimization schemes using the general nonlinear approaches with $\boldsymbol{\mathcal{H}} \neq \mathbf{I}$, the resolution matrix of the a posteriori model is not as easy[1] to get as for gradient based optimizations in terms of GD or CGD/NLCG.

2.4.2 Comparing resolution of local and global regularization

In non-linear inversion the measured data is considered as the model response of the stationary point to the cost functional plus the noise term

$$\mathbf{d} = \mathbf{f}(\mathbf{m}_*) + \mathbf{n} \ , \tag{2.65}$$

which is sought by iterative optimization methods. A first order Taylor expansion of the model response at the model of the n'th iteration yields

$$\mathbf{d} = \mathbf{f}(\mathbf{m}_*) + \mathbf{n} = \mathbf{f}(\mathbf{m}_n) + \mathbf{S}_n(\mathbf{m}_* - \mathbf{m}_n) + \mathbf{n} \ . \tag{2.66}$$

For a local regularization method, the iterative procedure would produce the model update

$$\mathbf{m}_{n+1} = \mathbf{m}_n + \boldsymbol{\mathcal{H}}_n^{-1}\mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{y} \ . \tag{2.67}$$

Setting $\mathbf{m}_* = \mathbf{m}_{n+1}$ and using equations (2.62) and (2.66) it follows

$$\begin{aligned}
\mathbf{m}_* &= \mathbf{m}_n + \boldsymbol{\mathcal{H}}_n^{-1}\mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S}_n(\mathbf{m}_* - \mathbf{m}_n) && +\boldsymbol{\mathcal{H}}_n^{-1}\mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{n} \\
&= \boldsymbol{\mathcal{H}}_n^{-1}\mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S}_n\mathbf{m}_* + \left(\mathbf{I} - \boldsymbol{\mathcal{H}}_n^{-1}\mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S}_n\right)\mathbf{m}_n && +\boldsymbol{\mathcal{H}}_n^{-1}\mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{n} \\
&= \mathbf{R}\mathbf{m}_* + (\mathbf{I} - \mathbf{R})\,\mathbf{m}_n && +\mathbf{R}\mathbf{n}
\end{aligned} \tag{2.68}$$

In analogy to the linear problem the resolution matrix $\mathbf{R}$ serves as kernel function. The resolution is linear regarding the model update $\delta\mathbf{m}$. The term $\mathbf{I} - \mathbf{R}$ can be interpreted as a complementary resolution matrix. In regions of missing resolution the model stays unchanged from the previous iteration. But, as a known problem to local regularization schemes, the resolution of the starting model is not clear.

[1] easy in terms of that ones the sensitivities are computed there is no additional effort to get the resolution analysis of our posteriori model.

On the contrary, a global regularization scheme has the model update

$$\mathbf{m}_{n+1} = \mathbf{m}_n + \boldsymbol{\mathcal{H}}_n^{-1}\mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{y} - \boldsymbol{\mathcal{H}}_n^{-1}\mathbf{C}_m^{-1}(\mathbf{m}_n - \hat{\mathbf{m}}) \; . \tag{2.69}$$

Expressed in terms of the resolution matrix and setting $\mathbf{m}_* = \mathbf{m}_{n+1}$ this reads [Günther, 2004]

$$\mathbf{m}_* = \mathbf{R}\mathbf{m}_* + (\mathbf{I} - \mathbf{R})\,\hat{\mathbf{m}} + \mathbf{R}\mathbf{n} \; , \tag{2.70}$$

which is formal similar to equation (2.68) except that $\mathbf{m}_n$ is replaced by $\hat{\mathbf{m}}$. The difference to equation (2.68) is obvious. It shows, that the model estimate is filled up with the starting model (and not the model of the preceding iteration) at regions of missing resolution. The matrix $\mathbf{I} - \mathbf{R}$ can be interpreted as the complementary constraint resolution matrix and describes, how the prior model $\hat{\mathbf{m}}$ is mapped into reality.

The result of a global regularization scheme is considerably independent of the way the iteration took, whereas all iteration stages strongly affect the final model in local regularization. From the resolution point of inversion many authors favor the global minimization to keep control on how the model is determined. Particularly if prior information is available, the model is restricted to lie in the neighborhood of it.

For Gauss-Newton type of inversion schemes, the calculation of the resolution matrix can be prohibitive or impractical. Because of it's size, direct factorization methods like SVD or Cholesky decomposition can not be used. Therefore it may become reasonable to approximate $\mathbf{R}$ as proposed by Alumbaugh & Newman [2000]. For this equation (2.62) can be rearranged and applying iterative solvers to

$$\left(\mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S} + \lambda\mathbf{C}_m^{-1}\right)\mathbf{R} = \mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S} \tag{2.71}$$

The linear system represents an inverse subproblem for every column of the resolution matrix and is therefore very challenging to solve. It can be solved approximately, e.g. by the iterative CG routines as given in Appendix B.0.5.

2.5 Summary

The non-linear nature of the problem attacked by a linearization of the cost functional and applying iterative minimization techniques. Additional measurement errors and the limited number of observations leads to a non unique ill-posed inverse problem. To overcome ill-posedness, the problem is regularized in some specified manner leading to a (local) unique solution of the specific inverse problem but increases the non uniqueness of the original problem on the other hand. The regularization may be applied as local constraint on the model update, or like a global constraint on the model itself. Yet, the solution of the original problem is deviated in some specified manner as well as the resolution of the a posteriori model. Therefore one has to further address the a posteriori model resolution to distinguish which part in the a posteriori model is resolved by the data and which does depend on the artificial regularization.

However, the assumption for the validity of resolution matrices is that the model is "close to the true model". In best case scenario we know the estimated model's forward response to be close to the one of the true model. Thus, the resolution is linked to the estimated model and only plausible models have interpretable resolution properties.

CHAPTER 3

TEM forward and adjoint operators

Electromagnetic depth sounding targets to resolve the spatial resistivity distribution in the interior of the earth. This is done either with natural variations of the earths magnetic field or with artificial sources. While natural sources allows soundings for deeper structures in the upper mantle and in the earth crust, the smaller artificial sources aim for the resolution of the commercially usable range of several hecto to some kilometers.

EM methods all utilize the fact, that time varying electromagnetic fields induce secondary electromagnetic fields in the conducting subsurface, which superimpose the primary fields. Thus, the observed field at the surface of the earth depends on the time and spatial structure of the source field and the conductivity distribution in the earth interior.

The natural variations, which constitute of big current systems in the ionosphere and magnetosphere, and magnetic dipoles act as inductive primary fields whereas grounded electrical dipoles have additional galvanic coupling. The grounded wire contributes simply a galvanic distortion of the electrical field to the sounding and can be utilized to resolve the conductivity distribution in the static case (DC) as well.

Before the used transient electromagnetic methods are described I start with the basic equations and numerical considerations for EM to get insight into the underlying physics which occur in depth sounding with TEM.

This follows a brief description of far zone (LOTEM) and near zone (SHOTEM) methods. Their characteristic transient features are given in 3.2. An important interpretation step towards processing data from controlled source TEM methods is to properly account for the systems impulse response or system response. Therefore the numerical treatment of the system response for SHOTEM and LOTEM is given at the end of this section.

After the introduction of the SHOTEM and LOTEM methods I describe how a full three-dimensional modeling for TEM is achieved. This problem is far more complicated to solve than the 1D case. From the computation itself may occur potential (curl-free) components due to round-off errors (spurious-modes) belonging to the large null space of the Maxwell's' spatial operator that can completely destroy the numerical accuracy for large diffusion times.

But, in order to create an inversion code which should fit a 3D model response to gathered TEM data, it is necessary to solve Maxwell's equations for arbitrary spatial conductivity distribution. Despite the numerical effort of this task, it is necessary to solve it in finite computation time as well. Therefore the SLDM approach of Druskin & Knizhnermann [1988] is used for the 3D modeling carried out in this work. The SLDM belongs to the class of Krylov subspace techniques. They are known to be the most (time) efficient way to achieve the responses of TEM fields for arbitrary spatial conductivity distribution [Hördt et al., 1992; Weidelt, 2000]. To give some insight into and explain some peculiarities of the code, namely convergence estimates, computation cost and how the system matrix is set up, a brief introduction into the background of the SLDM is given in section 3.3.

The proper design of the grid is a crucial point when applying the SLDM [Hördt et al., 1992; Hördt, 1998; Weidelt, 2000]. Compared to the explicit time stepping scheme, where the grid design is based on the CFL criterion, the stability criteria for the SLDM is based on the number of involved Krylov-vectors. This peculiarity of the SLDM lets the grid design become the most time consuming part when applying the method. Therefore a model study with automated grid design is given in section 3.4.

After the forward solution and it's proper computation, given to some extend, the last but least step to set up a proper inversion scheme is treated. The computation of sensitivities is carried out using the adjoint Green functions approach [Roach, 1982; McGillivray & Oldenburg, 1990]. For the formulation of the adjoint problem, I follow the approach of Hördt [1998] allowing a fast computation of TEM sensitivities directly in the time domain. The underlying equations are derived and I give some numerical insights on their implementation in the last but one section 3.5.

3.1 EM theory

Since all EM phenomena obey the Maxwell's equations it is pointless to start with them. They are coupled with the constitutive relations forming a set of partial differential equations (PDEs) to describe the space time behavior of the electrical and magnetical fields. Applying the quasi-static approximation, the EM phenomena are formulated in the diffusion regime leading to a set of first kind parabolic PDEs of second order. To formulate 1D forward operators to calculate model responses for TEM transients, I give a brief introduction by describing the complementary solutions to a half-space for the occurring PDEs. They are solved following a separation approach of Weidelt [1986] for the transverse or tangential electric (TE) mode and the transverse or tangential magnetic (TM) mode using Debeye potentials. Because the 1D problem is already well discussed by authors like [Kaufman & Keller, 1983; Weidelt, 1986; Ward & Hohmann, 1988] the extension to the layered case as well as the particular solutions for the potentials that arose from sources which where used for TEM exploration are given in appendix A.

The basic equations to describe the EM phenomena are the Maxwell's equations and espe-

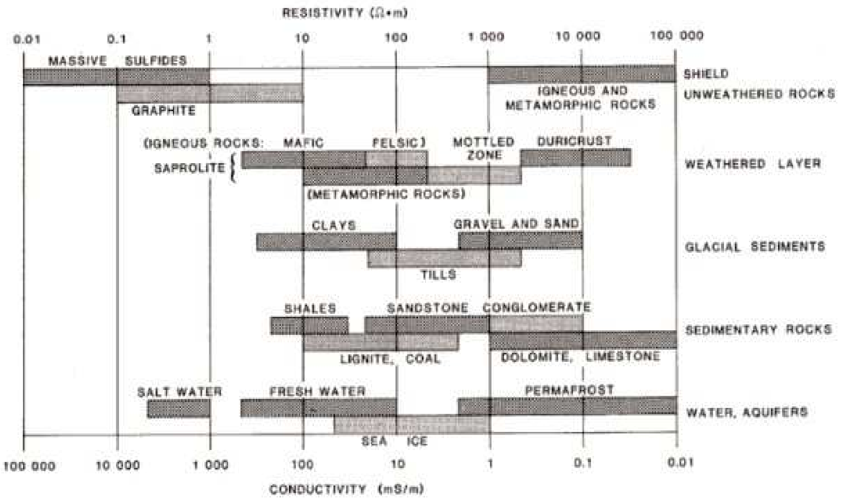


Figure 3.1: *Resistivity ranges for earth materials (after Palacky [1988]).*

cially Faraday's and Ampère's law. They are given below in differential and integral form:

$$\nabla \times \mathbf{E} = -\dot{\mathbf{B}} \qquad \oint_l \mathbf{E} \cdot \mathrm{d}\mathbf{l} = -\int_S \dot{\mathbf{B}} \cdot \mathrm{d}\mathbf{S} \tag{3.1}$$

$$\nabla \times \mathbf{H} = \mathbf{J} + \dot{\mathbf{D}} \qquad \oint_l \mathbf{H} \cdot \mathrm{d}\mathbf{l} = I_S + \int_S \dot{\mathbf{D}} \cdot \mathrm{d}\mathbf{S} \tag{3.2}$$

Here, $\mathbf{E}$ is the electrical field in $\frac{V}{m}$, $\mathbf{D}$ is the electric displacement field in $\frac{As}{m^2}$, $\mathbf{H}$ is the magnetic field in $\frac{A}{m}$, $\mathbf{B}$ is the magnetic induction or magnetic flux in $\frac{Vs}{m^2}$, $\mathbf{J}$ is the current density in $\frac{A}{m^2}$, I_S is the current in the plane S and the length l is the boundary of S.

The equations (3.1) and (3.2) can be coupled with the constitutive relations

$$\mathbf{B} = \mu \mathbf{H} \;, \qquad \mathbf{D} = \epsilon \mathbf{E} \tag{3.3}$$

and Ohm's law

$$\mathbf{J} = \sigma \mathbf{E} \;, \tag{3.4}$$

to form a set of second order PDEs. μ is the magnetic permeability ($\mu = \mu_r \mu_0$ with the free space permeability $\mu_0 = 4\pi 10^{-7}\,\frac{Vs}{Am}$ and the relative permeability μ_r), ϵ is the dielectric permittivity ($\epsilon = \epsilon_r \epsilon_0$ with $\epsilon_0 := \mu_0^{-1} c_0^{-2}\,\frac{As}{Vm}$ and the speed of light $c_0 = 299.792.458\,\frac{m}{s}$ for free space and its relative value ϵ_r) and σ is the conductivity. In general μ, ϵ and σ are tensors and depend on time, pressure, temperature and may be anisotropic as well. For most earth materials $\mu_r \approx 1$ but the conductivity can vary over a broad range.

According to the type of geological material, resistivity can vary from $1.6 \times 10^{-8} \Omega m$ for native silver to $10^{16} \Omega m$ for pure sulfur. Figure 3.1 shows ranges of resistivities of the earth's materials. Sedimentary rocks tend to be the most conductive due to the high pore fluid content, igneous rocks tend to have the highest resistivities and metamorphic rocks have intermediate overlapping resistivities.

In sedimentary rocks, resistivity depends on the clay content, porosity, dissolved mineral content and water saturation of the rocks. In these rocks the resistivity of the interstitial fluid is probably more important than that of the host rock. Archies formula, presented in equation (3.5) also known as Archies Law, is an empirical formula used to calculate the effective resistivity of a rock containing water:

$$\rho_a = a_f \phi_r^{-m_c} S_w^{-n_s} \rho_w \tag{3.5}$$

where ρ_a is the bulk or apparent resistivity of the rock, ρ_w the resistivity of the pore fluid (mainly water), ϕ_r is the rock porosity; S_w is the volume fraction of pores filled with water, a_f, m_c and n_s are constants where the formation factor $0.5 \leq a_f \leq 2.5$, the cementation exponent $1.3 \leq m_c \leq 2.5$ and the saturation exponent $n_s \approx 2$. This equation applies for clean sands or sandstones, where the pore space is filled ($a_f \approx 1$) or partially filled ($a_f < 1$) with fluid. However, it may be used as a first guess for other earth materials. It demonstrates a basic observation within EM studies of the subsurface, namely, that the measured conductivity of the host material is mostly influenced by the amount of water and mineralization type it bears.

Also the rocks behavior changes with temperature, depth, salinity and porosity and such in situ factors should be taken into account in the resistivities determination. Strack [1992] and Keller [1987] show these dependencies. The age of the rock can also influence its resistivity [Strack, 1992]. However, as we can see from figure 3.1, the conductivity does not drop below some 10^{-5} S/m for most earth materials.

According to Faraday's law, the time variation of the magnetic flux induces eddy currents in a conducting material (the earth). For artificial external sources, they are flowing in the same direction as the primary current system in order to establish the old primary magnetic field. When considering natural variations, the source system is further away and is described by a plane wave excitation. In contrast to any free space behavior, the retarded EM wave is critically dampened within conducting materials. Even if we consider very small sampling times of some μs the conductive currents $\mathbf{J} = \sigma\mathbf{E}$ are ten times stronger than the magnetic effect of the displacement currents $\dot{\mathbf{D}}$ [Ward & Hohmann, 1987].

Because of this fast decaying displacement currents and without taking any free charges into account, the physical properties of induced currents and magnetic fields inside a conducting body can be described with a diffusion equation. This *quasi static approximation* is generally valid if $\mu\epsilon\omega^2 << \mu\sigma\omega$. The magnetic induced displacement currents in the air are mainly generating a small phase shift to the EM fields.

Figure 3.2 displays the working frequencies of common EM techniques. As you one see

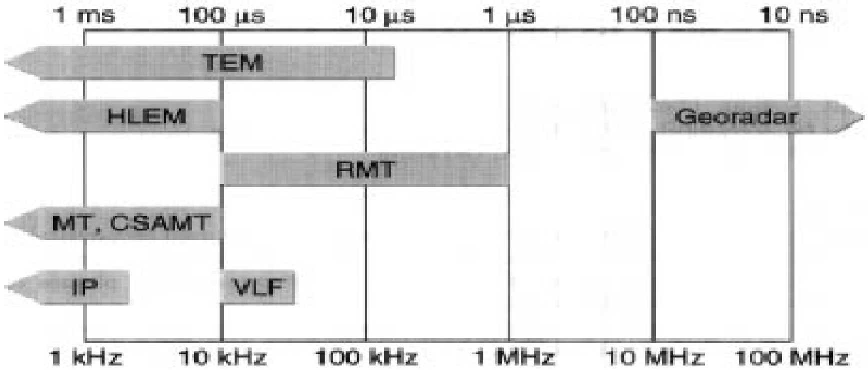


Figure 3.2: *Frequency and time range of some EM-techniques (after Tezkan [1999]).*

from figure 3.2, most EM techniques generally use a frequency/time range, where the displacement current for most earth materials can be neglected. A big exception is valid for Georadar, where the wave nature of the EM field is a substantial part of the technique. However, for TEM methods, the displacement currents can be neglected and the quasi-static approximation can be applied. Thus, the most relevant Maxwell equations read

$$\begin{aligned} \nabla \times \mathbf{E} &= -\dot{\mathbf{B}} && (3.6) \\ \nabla \times \mathbf{B} &= \mu\,(\sigma \mathbf{E} + \mathbf{J}) \ . && (3.7) \end{aligned}$$

Or after the elimination of **B**

$$\nabla \times \nabla \times \mathbf{E} + \mu\sigma \frac{\partial \mathbf{E}}{\partial t} = -\mu \dot{\mathbf{J}}_e \tag{3.8}$$

which is a parabolic PDE of second order. In equation (3.8), $\mathbf{J}_e$ accounts for the external source current, which is assumed to have a plane wave for natural external fields or which may be defined through the transmitter signal. This kind of PDE can also be found in heat transports for example.

The general solution of a boundary-value problem is the sum of the particular solution to the inhomogeneous PDE (source terms) and the complementary solution of the homogeneous (or source free) equation. To describe the forward problem, which is the key to the solution of the inverse process, it is convenient to start with the complementary solution of a homogeneous half-space.

The solution of the complementary problem for a layered medium involves the calculation of impedances accounting for different boundary consistencies of the modes. It contains more mathematical considerations and the basic algorithms are given in appendix A for their numerical calculation.

Because the underlying mathematical considerations for the generation of 1D model responses for arbitrary sources are already complicated, this theory section gives just a short introduction. A detailed description can be found in Weidelt [1986] and Ward & Hohmann [1988].

3.1.1 Complementary solution

The complementary solution is independent for each source receiver constellation and depends on the conductivity distribution only. For a homogeneous half-space it is the most easy way to compute the EM responses and is therefore very useful to get a first insight how an EM excitation is propagated through a conducting material. Since I used the approach of Weidelt [1986] they are given as the solution of scalar (Debeye) potentials. The complementary solution may be derived using vector (Schelkunoff) potentials as well. This approach is described in detail by Ward & Hohmann [1988].

For a homogeneous or a layered media with $\sigma = \sigma(z)$ and apart from the sources, the fields **E** and **B** are divergence free and can be split into toroidal and poloidal parts. With this separation one can describe the EM fields through two scalar potentials ϕ_e and ϕ_m. After

Weidelt [1986] the definitions

$$\mathbf{E} = \mathbf{E}_e + \mathbf{E}_m = -\nabla \times (\dot{\phi}_e \hat{\mathbf{e}}_z) + \frac{1}{\sigma} \nabla \times \nabla \times (\sigma \phi_m \hat{\mathbf{e}}_z) \tag{3.9}$$

$$\mathbf{H} = \mathbf{H}_e + \mathbf{H}_m = \frac{1}{\mu} \nabla \times \nabla \times (\phi_e \hat{\mathbf{e}}_z) + \nabla \times (\sigma \phi_m \hat{\mathbf{e}}_z) \ , \tag{3.10}$$

with unit vector in z-direction $\hat{\mathbf{e}}_z = (0, 0, 1)^{\mathrm{T}}$, are chosen to calculate the EM fields for homogeneous or layered media.

With the equations (3.9) and (3.10) one can derive the equations for every component of the EM fields as

$$\mathbf{E} = \begin{pmatrix} E_x \\ E_y \\ E_z \end{pmatrix} = \begin{pmatrix} -\partial_y \dot{\phi}_e + \frac{1}{\sigma} \partial^2_{xz} (\sigma \phi_m) \\ \partial_x \dot{\phi}_e + \frac{1}{\sigma} \partial^2_{yz} (\sigma \phi_m) \\ -\left(\partial^2_{xx} + \partial^2_{yy}\right) \phi_m \end{pmatrix} \tag{3.11}$$

for the electrical field components and

$$\mathbf{H} = \begin{pmatrix} H_x \\ H_y \\ H_z \end{pmatrix} = \begin{pmatrix} \frac{1}{\mu} \partial^2_{xz} \phi_e + \sigma \partial_y \phi_m \\ \frac{1}{\mu} \partial^2_{yz} \phi_e - \sigma \partial_x \phi_m \\ -\frac{1}{\mu} \left(\partial^2_{xx} + \partial^2_{yy}\right) \phi_e \end{pmatrix} \tag{3.12}$$

for the magnetic field components.

The definitions (3.9) and (3.10) lead to the PDEs

$$\nabla^2 \phi_e = \mu \sigma \dot{\phi}_e \ , \qquad \nabla \left[\frac{1}{\sigma} \nabla \left(\sigma \phi_m \right) \right] = \mu \sigma \dot{\phi}_m \ , \tag{3.13}$$

with $\sigma = \sigma(z)$ and $\mu = \mu_0$. This approach ensures also that

$$\nabla \times \mathbf{E}_e = -\mu \dot{\mathbf{H}}_e \ , \quad \nabla \times \mathbf{H}_m = \sigma \mathbf{E}_m \tag{3.14}$$

$$\nabla \times \mathbf{H}_e = \sigma \mathbf{E}_e \ , \quad \nabla \times \mathbf{E}_m = -\mu \dot{\mathbf{H}}_m \tag{3.15}$$

is automatically fulfilled.

ϕ_e does not generate electric field components in vertical direction and ϕ_m does not generate a magnetical component in vertical direction. Thus ϕ_e is also called the transversal or tangential electric (TE) mode and ϕ_m is called the transversal or tangential magnetic (TM) mode. Often the word polarization is used equivalent to mode. Note that ϕ_e and ϕ_m are both cylinder symmetric and dependent on z only. This occurs, if the exciting field is a multi-pole field (dipole, quadrupole, etc.), which is the only case considered here.

To evaluate ϕ_e and ϕ_m in the frequency domain ($\omega = 2\pi\mathsf{f} = 2\pi/t$), they are developed as synthesis of partial waves and inverse Fourier transformation

$$\phi_{e,m}(\mathbf{r}, \omega) = \frac{1}{4\pi^2} \iint\limits_{-\infty}^{\infty} f_{e,m}(z, \mathbf{k}, \omega) e^{i\mathbf{kr}} \, \mathrm{d}\mathbf{k} \ , \tag{3.16}$$

with $\mathbf{k} = k_x\hat{e}_x + k_y\hat{e}_y$, $k^2 = k_x^2 + k_y^2$.

Utilizing the rotational symmetry of $\phi_{e,m}$, a further simplification of the double integrals in the transformation (3.16) is possible. If the potential is rotational symmetric around the z-axis, which means it depends on $r^2 = x^2 + y^2$ only, (3.16) can be written as [Baños, 1966]

$$\phi_{e,m}(z,r,\omega) = \frac{1}{2\pi}\int_0^\infty f_{e,m}(z,k,\omega)J_0(kr)k \,\mathrm{d}k \;, \tag{3.17}$$

with J_0 the Bessel function of order zero and first kind.

The transformation (3.17) makes (3.13) independent from the $x-$ and $y-$directions and simplifies (3.13) to

$$\begin{aligned} \nabla^2 f_e(z,k,\omega) = \partial_z^2 f_e(z) &= u(z)^2 f_e(z) & (3.18)\\ \partial_z\left[\frac{1}{\sigma(z)}\partial_z\left(\sigma(z)f_m(z,k,\omega)\right)\right] &= u(z)^2 f_m(z)\;, & (3.19)\end{aligned}$$

with $u(z)^2 = k^2 + i\omega\mu\sigma(z)$.

For a homogeneous half-space, the amplitude of the transformed potential is therefore just an exponential damping of the initial amplitude at the surface:

$$f_{e,m}(z,k,\omega) = f_{e,m}(0,k,\omega)e^{-u(z)z}\;. \tag{3.20}$$

For the easiest case of a homogeneous half-space with the wavenumber ($k = 0$) which is called "‘quasi-homogeneous"’ field, **E** and **H** are decaying similarly like

$$\left.\begin{array}{c} f_{e,m} \\ \partial_z f_{e,m}\end{array}\right\} \sim e^{-\sqrt{i\omega\mu\sigma}z} = e^{\frac{-(1+i)z}{d^*}}\;. \tag{3.21}$$

Where ($\omega = 2\pi f = \frac{2\pi}{t}$)

$$d^* = \sqrt{\frac{2}{\omega\mu\sigma}} = \sqrt{\frac{t\rho}{\pi\mu}}\;, \tag{3.22}$$

is the electromagnetic *skin depth* or *diffusion depth*. The magnitude of d^* plays an important role in grid design, EM-case studies and planning the appropriate field setup For a layered case, the skin depth may be extended using the effective medial conductivity [Spies & Frischknecht, 1991]:

$$\sigma(z) = \frac{1}{z}\int_0^z \sigma(z')\,\mathrm{d}z'\;. \tag{3.23}$$

Note that **E** and **H** are decaying with different velocities in a layered or multidimensional media. A complete derivation of the EM field evaluation for layered media is given in appendix A.

3.1.2 Evaluating Bessel integrals numerically by a fast Hankel transform (FHT)

Because of the symmetry and through their transformations in the wavenumber domain, the PDE (3.13) becomes independent from $x-$ and $y-$ directions. Thus the eigenmodes of the two potentials and their z-derivatives can be calculated in the frequency-wavenumber domain by ascribing the equations through integration of Bessel functions of zero'th and first order. This is summarized by

$$h_{0,1}(r) = \int_0^\infty f(k) J_{0,1}(kr)\, \mathrm{d}k \tag{3.24}$$

$$dh_{0,1}(r) = \int_0^\infty \partial_z f(k) J_{0,1}(kr)\, \mathrm{d}k\ . \tag{3.25}$$

The numerical evaluation of the integrals can be done through constitution of appropriate filter coefficients $\tilde{H}_{0,1}$ for the Bessel functions $J_{0,1}$ (e.g. Anderson [1989]) and using the fast Hankel transform (FHT):

$$\hat{h}_{0,1}(r_m) = \frac{1}{r_m}\sum_{n=1}^{m+1} f(k_n)\tilde{H}_{0,1}(m-n) \tag{3.26}$$

$$d\hat{h}_{0,1}(r_m) = \frac{1}{r_m}\sum_{n=1}^{m+1} \partial_z f(k_n)\tilde{H}_{0,1}(m-n) \quad m \in \{0,1,\cdots,60\} \tag{3.27}$$

$r_m := \frac{1}{k_m} := r_{min}10^{\frac{m}{10}}$ expresses the discretized radial length. Keep in mind that this makes the integration independent from the real value of kr and thus the same coefficients can be used for nearly every discretization. For the set of filter coefficients that I use, the radial length was sampled with ten coefficients per decade. $r_{min} := r_{max}10^{-6}$ is the minimal r value which can be sampled for the evaluation of equation (3.26) and equation (3.27) at maximum distance from the source.

According to Weidelt [1986], the FHT reproduces the Bessel integration for a sufficient smooth argument $f(k)$ with the desired precision. This holds for nearly every 1D conductivity distribution, but it turns out that it can be violated for very extreme parameters. If, for e.g. $kr \ll 1$, the Bessel function can have strong oscillations which produces numerical errors. This can be avoided if one circumvents the evaluation of electrical fields within a critical radius to the source ($r < r_{min}$). Another criteria can be set upon physical thoughts as well. One can circumvent the problem if one restricts the evaluation of the EM fields to a radial distance bigger or equal the skin depth.

3.1.3 Step excitation and time transformation

The step-on excitation of a dipole source generating an EM field can be expressed as a heavy-side function in time

$$f(t)^{\text{step}} = \begin{cases} 0 & t < 0 \\ f_0 & t > 0 \end{cases} \tag{3.28}$$

Thus, the step response of TEM sources can be expressed as inverse Fourier transformations [Papoulis, 1962]

$$f(t)^{\text{step}} = \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} \frac{f(\omega)e^{i\omega t}}{i\omega}\, \mathrm{d}\omega\ . \tag{3.29}$$

Equation (3.29) can further be simplified as sinus or cosine transformation [Papoulis, 1962]:

$$f(t)^{\mathrm{step}} = \frac{2}{\sqrt{2\pi}} \int_0^\infty \Im\left(\frac{f(\omega)}{\omega}\right) cos(\omega t)\, \mathrm{d}\omega = \frac{2}{\sqrt{2\pi}} \int_0^\infty \Re\left(\frac{f(\omega)}{\omega}\right) sin(\omega t)\, \mathrm{d}\omega \ ,. \tag{3.30}$$

Here $\Im(f(\omega))$ denotes the imaginary part and $\Re(f(\omega))$ the real part of the complex valued $f(\omega)$. In practice, it seems numerically favorable to express the Fourier integrals through the fast Hankel transform [Johanson & Sørensen, 1979]:

$$f(t_n)^{\mathrm{step}} = \frac{1}{t_n}\sqrt{\frac{2}{\pi}} \sum_{j=1}^{N_t+N_f-1} \Im(\frac{f(\omega_j)}{i\omega_j})\tilde{H}_{-\frac{1}{2}}(n-j) \tag{3.31}$$

$$= \frac{1}{t_n}\sqrt{\frac{2}{\pi}} \sum_{j=1}^{N_t+N_f-1} \Re(\frac{f(\omega_j)}{i\omega_j})\tilde{H}_{\frac{1}{2}}(n-j) \quad n = 1, \ldots, N_t \ . \tag{3.32}$$

where N_t is the number of time points and N_f is the filter length. The definition of the sampling interval for t_n and ω_j is chosen according to the smallest time point t_{min}, the filter length and the sampling points per decade of $\tilde{H}_{\frac{1}{2}}$ $(N_f = 80)$

$$t_n = 10^{\frac{n-1}{10}} t_{min}, \qquad n \in \{1, 2, \ldots, N_t\} \tag{3.33}$$

$$\omega_j = 10^{\frac{41-j}{10}} \frac{1}{t_{min}} \qquad j \in \{1, 2, \ldots, N_t + 79\} \ , \tag{3.34}$$

For example, the calculation of $f(t)$ with $t_{min} = 1\,ms$ to $N_t = 41$ time points ($t_{N_t} = 10\,s$), one evaluates $f(\omega_j)$ at 120 frequencies ($\mathsf{f}_j = \frac{\omega_j}{2\pi}$) between $2\,10^{-6}\,Hz$ and $1.6\,10^{6}\,Hz$. This also exemplifies, that a relatively large frequency band is needed for the inverse Fourier transform.

3.1.4 Summary

The complementary problem is solved by partial wave synthesis in the frequency-wavenumber domain as products of exponential decay functions which can be converted to a convolution integral using a fast Hankel transform. They depend on the conductivity with depth structure only and are the same for different sources.

The electrical and the magnetical fields decay uniformly according to the skin depth $d^* = \sqrt{\frac{t}{\pi\mu\sigma}} \approx 503\sqrt{t\rho}$ in a homogeneous medium. However, **E** and **H** are decaying with different velocities in a layered or multidimensional media. In a layer containing the source, one must add the particular solution of the inhomogeneous problem to the complementary solution. Because the 1D problem is already well known, the derivation of the particular solutions to various sources as well as algorithms for a EM field calculation of layered models are given in appendix A.

The step excitation of a step on source field can be expressed as a fast Hankel transform, where the frequencies for the transformation are to be chosen appropriate to the modeled time range. This frequency range should be extended to higher frequencies for impulse responses in order to get numerically stable results for the sensitivity computation using the adjoint Green functions approach (see section 3.5).

3.2 Depth sounding with controlled source transient electromagnetics

Controlled source Transient Electromagnetic Methods, or shortly TEM, utilize imposed currents generated by a well known source, to investigate the subsurface conductivity structure. This is based on the principle, that a current flowing in a wire or loop source produces EM fields in the surrounding (primary field). After the current is switched off (or on), the primary field changes induces secondary EM fields in the surrounding and conducting subsurface (step response). With ongoing time, the step response is measured within electrical, magnetical, or inductive sensors. The time dependence of the measured EM fields is related to a specific depth correlated through the conductivity structure of the subsurface. The imaging process of the subsurface conductivity distribution is then achieved by the solution of the appropriate inverse problem.

A general distinction of the TEM method can be given by the source receiver separation. Near-zone, or short-offset soundings, refers to cases in which the loop size or source-receiver separation is less than the depth of investigation. The term far zone applies to soundings where the source-receiver separation or transmitter size is much greater than the depth of investigation. In practice a LOTEM sounding may involve aspects of both, near-zone and far-zone soundings, if measurements are made over a wide enough time range.

There are many different field setups and waveform characteristics for TEM measurements. Because the main focus in this work is to enhance the appraisal capabilities of short and long offset transient electromagnetic depth soundings, the discussed field setups are limited to the customary. Short offset transient soundings are traditionally done with rectangular or loop wire, whereas for long offset soundings a long grounded dipole or shortly bipole, is used. A detailed discussion on TEM in general can be found in Nabighian & Macnae [1991] and in the book of Strack [1992].

3.2.1 Short offset transient electromagnetics (SHOTEM)

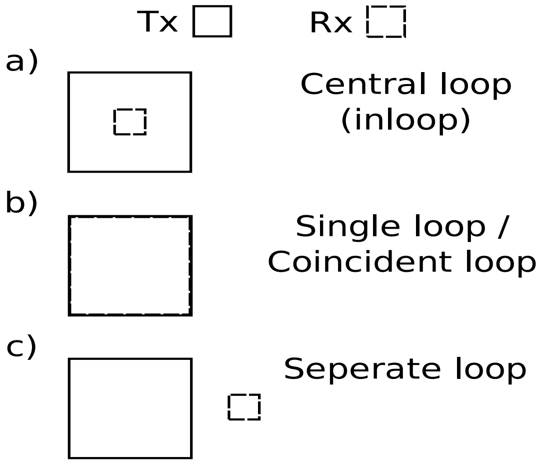


Figure 3.3: Three basic field setups for SHOTEM measurements: a) shows the most common Central-Loop arrangement. The Coincident-Loop setup is shown in b). Configurations for measurements outside the Loop are called Separate-Loop (c).

Figure 3.3 shows the basic three field configurations which are customary used for short offset transient electromagnetic soundings. The source generally consists of a loop wire (Tx) and the change of the magnetic induction or flux is measured in a coil (Rx). The most common field setup is realized within the central loop or in-loop configuration. For this setup, the receiver is placed at the center of the transmitter which can be seen under a). Coincident loop measurements are done, when the transient response is measured in a wire besides the transmitter (fig. 3.3 b)). Additionally, the transmitter coil can be utilized as receiver coil during switch off phase. This setup is called single coil measurement and is auxiliary imagined in setup b). Some measurements are done with the receiver site outside the transmitter loop, which is called separate loop configuration and shown in c). The receiver generally consists of a small wire which is laid out in a square upon the earth. To enhance the receiver moment, the coil can be build with many windings, have a core of magnetical material or both.

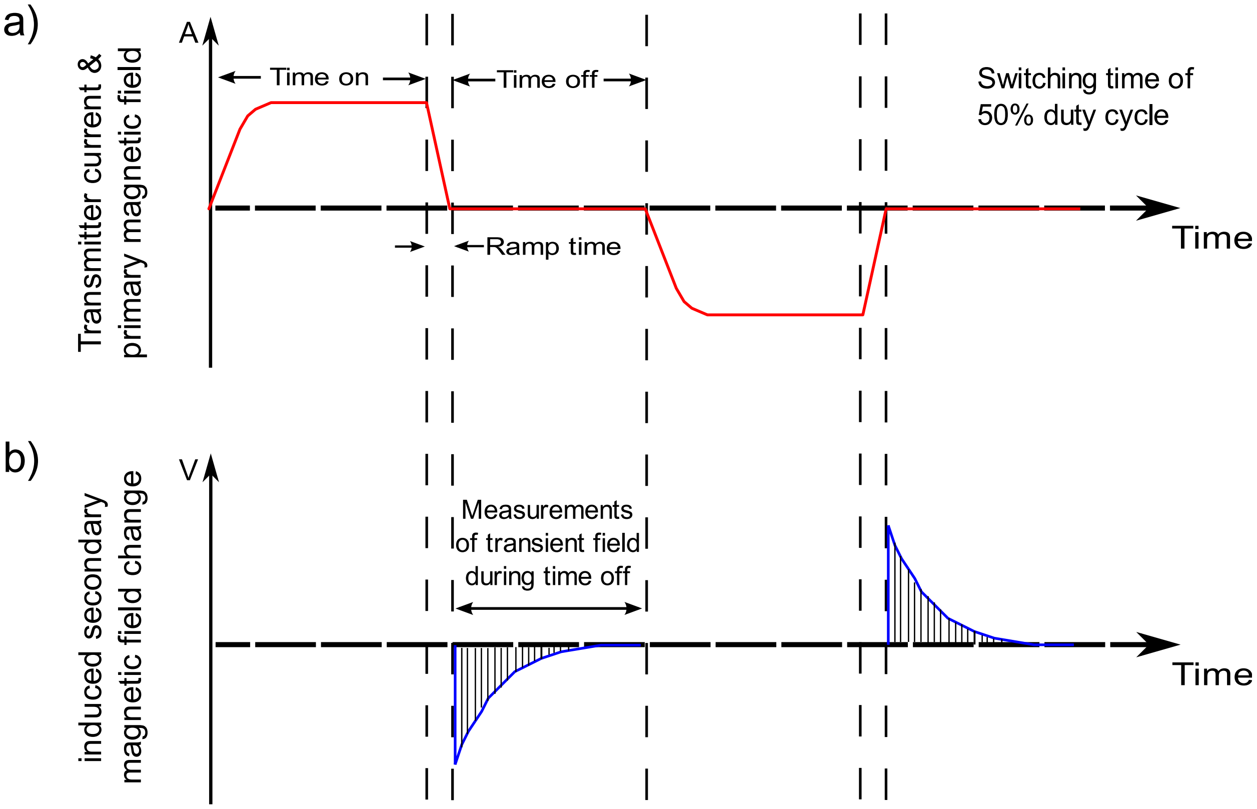


Figure 3.4: *TEM transmitting signal and induced currents for a common SHOTEM survey. The source current signal and primary magnetic field of the transmitting signal versus time is shown in a). b) shows the transient step response which would be measured as voltages in a coil. Further explanations are given in the text.*

To get insight into the used nomenclature which is used in the following, a typical transmitting waveform and measured earth responses are given in figure 3.4.

The source is assumed to have a square loop shape and the receiver is placed inside the loop (in-loop configuration). The transmitting signal (fig. 3.4 a)) consists of a square wave signal with 50% duty cycle. This means, that after a period of switched on current, the current is shut off (step off). Since the wire loop has its own inductive circuit, the current is not going to zero immediately. The time which is needed for the transmitter to reach the zero current is called *ramp time*. The current in the circuit is linear decreasing during the ramp time. It depends on the size of the transmitter and its dipole moment and the used electronic devices within the transmitter and depends nearly linear on the transmitter size.

The current stays at zero for the same duration as it was switched on, hence a 50% duty cycle is emitted. After the zero current time frame, the current is switched on with current flow in the opposite direction. Figure 3.4 b) shows the normalized transient step response which would be measured as voltages in a coil at the center of the loop. Note, that the time scale as well as the transient response shown here are not linear. Because the time decay is very fast, it is usually displayed using a logarithmic scale for both axis. The decay of the induced current systems in the earth conducting material can be impressively described by the ”smoke-ring” concept of Nabighian [1979] which is shown in figure 3.5. Here, the decay of the current system of a loop source is described through a current system which is going downward and gets broader at the same time inducing changes in the vertical magnetic fields corresponding to the depth of penetration.

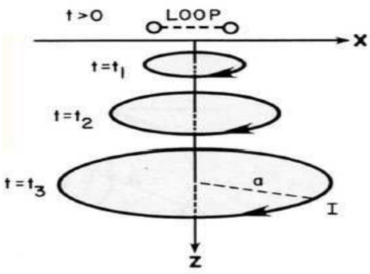


Figure 3.5: *Smoke ring concept after Nabighian [1979]. With ongoing delay time of the shut off signal ($t_1, \ldots, t_3$), the current system decays into the earth.*

Figure 3.6 shows the model (a) and three simulated $\dot{H}_z$-transients (b) using a in-loop configuration. In a), a square loop transmitter of 50 × 50 m^2 size is shown on a resistive medium. The resistivity model consists of a simple two layer case with varying resistivities ρ_1 and ρ_2. Further dimensions of the model can be taken from the picture.

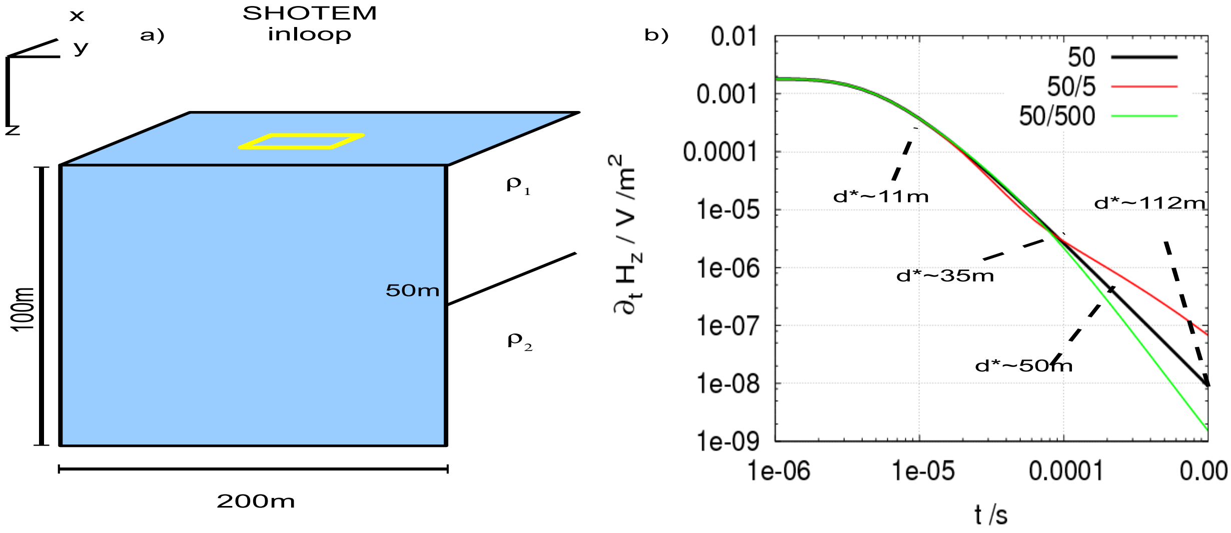


Figure 3.6: *a) Shows a simplified model of a typical SHOTEM in-loop configuration. A square loop transmitter is marked with a yellow rectangle. The transient responses of a central loop voltage in a log-log graph over time for three different conductivity profiles are given in b). Further explanations are given in the text.*

To gain insight into the sounding curves, I calculated the transient responses for three different two layer models. The SHOTEM $\dot{H}_z$-transients are displayed as voltages per m^2 and given in b) in a log-log graph over time t per seconds. The black curve shows the transient response of a 50 Ωm half-space ($\rho_1 = \rho_2$). For convenience, the approximated diffusion depth for the half-space model is calculated using (3.22) and marked at four different times.

Note that the skin depth according to (3.22) increases with $\sqrt{t}$ which can be seen from the picture, too. The red curve shows the transient response of a 10:1 and the green curve shows the response of a 1:10 resistivity contrast to the background. Asymptotic derivations for the late and early time range of the $\dot{\mathsf{H}}_\mathrm{z}$-response can be calculated to explain the time and conductivity dependence of the in-loop sounding curves. Following Kaufman & Keller [1983]

$$\dot{\mathsf{H}}_\mathrm{z}^{et} \sim \rho \qquad \dot{\mathsf{H}}_\mathrm{z}^{lt} \sim \rho^{-3/2} t^{-5/2} \tag{3.35}$$

The main characteristic time feature of the sounding curves is that they decay with $t^{-5/2}$ at late times whereas for early times there is no time dependence. For intermediate times, the connection to a simplified model is far more complicated. However, the graphs are displaying the characteristic features for early and late times nicely: For times $t \leq 2\,\mu s$, the early time range of the transient is showing a straight line, whereas for late times ($t \geq 0.1\,ms$), the form of the sounding curve is dominated by the underlying half-space ($\dot{\mathsf{H}}_\mathrm{z} \sim \rho^{-3/2} t^{-5/2}$).

The resistivity dependence for late times is given as $\rho^{-3/2}$. From fig. 3.6 b) one can see, that for the higher resistivity contrast, the green sounding curve is decaying steeper as the half-space solution, whereas the red curve, which corresponds to the higher conductivity decays slower. For early times, the transient is showing the resistivity of the half-space (or the first layer) which can be directly seen from Ohm's law and is also displayed in 3.6 b).

Note, that the green and the red line deviate from the half-space solution before the calculated skin depth reaches the conductivity contrast. This can be understood from the depth of investigation theory of Spies & Frischknecht [1991]. Since the skin depth gives the depth in which the signal amplitude is just reduced to $1/e$, the information content is traveling faster.

One method of estimating the contribution to the EM response from various depths is to calculate the Fréchet derivatives from the appropriate integral equation (e.g. Parker [1977b]; Chave [1984]; Spies & Frischknecht [1991]). The Fréchet derivatives give the sensitivity of the measurements to small local changes in conductivity, and the shape of the Fréchet kernel can be analyzed to obtain approximate penetration depths and the resolution intervals associated with any model and depth interval. As we will see later, the sensitivities computed with the adjoint method uses the same theoretical background and are mathematically equivalent to the Fréchet derivatives [McGillivray & Oldenburg, 1990].

A reasonable estimate for the depth of investigation for EM methods is therefore taken to be approximately equal to 1.5 skin depths [Spies & Frischknecht, 1991]. Thus, for the time of $t \approx 0.1\,ms$, the depth of investigation is roughly $50\,m$ which is in excellent agreement to the part graph where the sounding curves despair (see fig. 3.6 b).

In SHOTEM, the exploration depth is not affected by the transmitter/receiver separation, since the time derivative of the secondary field is measured in the absence of the primary field [Tezkan et al., 1996]. Yet, it is dependent on the the used current(I), the loop size (A) and the minimal signal which can be resolved by the acquisition system. After [Spies & Frischknecht, 1991], the maximum penetration depth which can be resolved, assuming a noise level of $10^{-3}\,nT$ is given by

$$z^*_{max} \approx 28\,(AI)^{\frac{1}{3}} \tag{3.36}$$

which may be used as a rule of thumb for most regions, except urban areas.

3.2.2 Long offset transient electromagnetics (LOTEM)

For the LOTEM method, the exploration depth is more affected by the transmitter/receiver separation but does not solely depend on it. In fact, the sensitivity of the method is depending on it, but the exploration depth generally depends on the signal strength and the measured time frame after the transmitting signal is switched on or off. With LOTEM, unlike the SHOTEM method, one aims at soundings for greater depth. Figure 3.7 shows the basic field configurations which are used for LOTEM soundings. Compared to the SHOTEM measurements, for LOTEM the source receiver separation is considered as intermediate or far zone. The field measurements are done at one or many different receiver sites within the offset r to the transmitter (Tx). The source generally consists of a grounded dipole (Fig. 3.7 a). The grounding is normally achieved by iron pipes or plane conducting material buried in the earth. Each source point is build up several hundred meters to several kilometers apart. To take this large separation into account, one often speaks of a *bipole* transmitter.

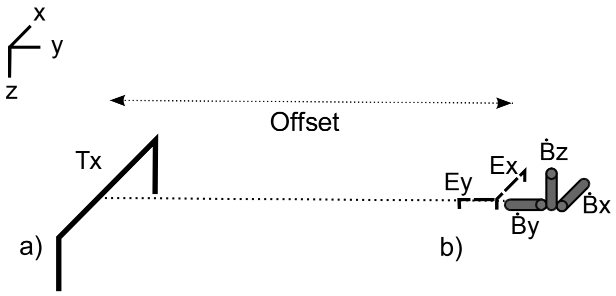


Figure 3.7: *Basic field setup for LOTEM measurements. The source consists of a grounded wire (Tx) with a big dipole length which can be seen in a) The receiver site (b) normally consists of electrical sensors (*E_x *and* E_y*) and induction coil sensors (*$\dot{H}_x$*,* $\dot{H}_y$ *and* $\dot{H}_z$*) which are placed at a collective offset.*

At the receiver sites (Fig. 3.7 b) the change of the magnetic induction or flux is measured by induction coils. Unlike the classical in-loop SHOTEM, horizontal magnetic components are measured, too. Additionally a large horizontal wire with a bigger dipole moment is sometimes used to measure $\dot{H}_z$. Generally there are two electric field sensors (E_x and E_y) measuring the potential difference between two adjacent points.

When measuring in the equatorial distance at $r = y$, with the center of the transmitter as origin and the current flowing in x-direction, one generally speaks of the *broadside* configuration. Contrariwise when speaking of the *inline* configuration one measures at $r = x$. The setup should, of course, be chosen appropriately to the problem. For borehole applications or marine TEM measurements additional informations may be caught while measuring the E_z component as well.

To get insight into the designations which are typical for LOTEM, a transmitting waveform and measured earth responses are given in figure 3.8. The source is assumed to be a grounded dipole. The transmitting signal (fig. 3.8 a)) consists of a square wave signal with 100% duty cycle. Note, that the current signal is alternating from minus to plus, which is commonly used for galvanic coupled transmitters. That way one can get probably twice the current amplitude as measuring with a 50% duty cycle. Figure 3.8 b) shows the normalized transient step response which would be measured as voltages in a coil at the distance r to the transmitter. A conventionalized E_x-field step response is shown in the last row (c). Note, that the electric field increases with time and becomes a asymptotic value for late times (DC-value) which is commonly observed in the *step on* response.

The conversion from step "off" to step "on" response is achieved through the DC-value

$$f^{\text{step off}} = f^{\text{DC}} - f^{\text{step on}} \tag{3.37}$$

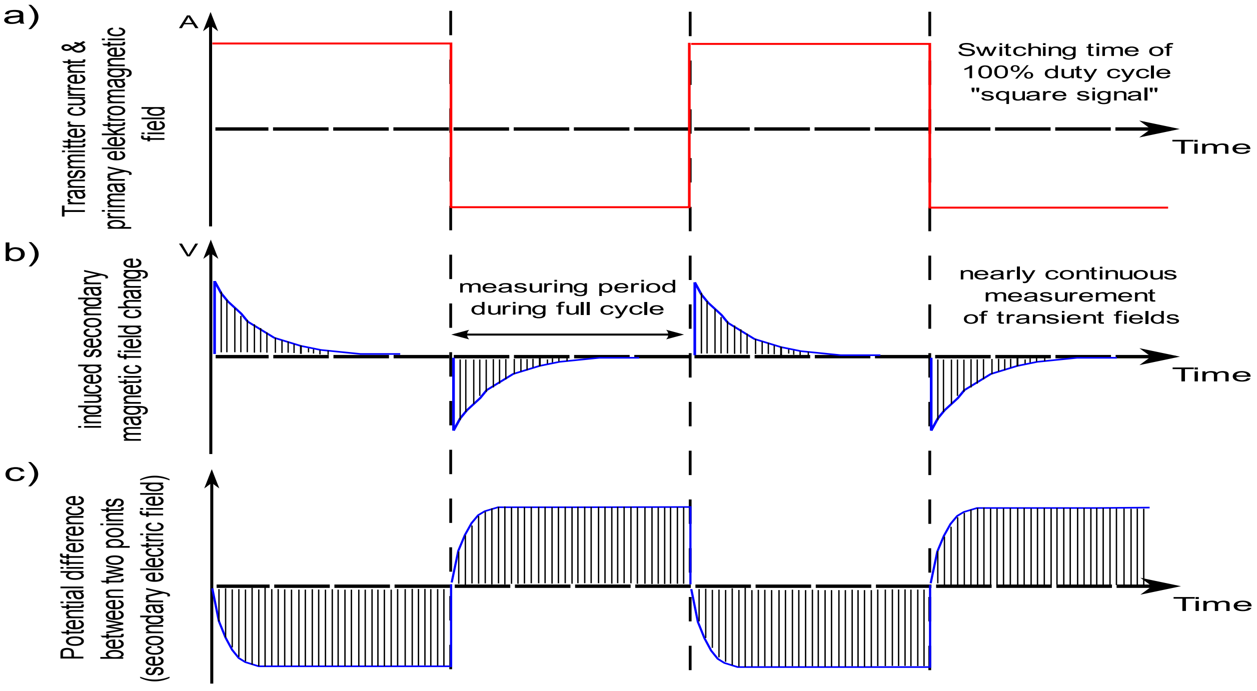

a)
Transmitter current & primary elektromagnetic field
A
Switching time of 100% duty cycle "square signal"
Time
b)
induced secondary magnetic field change
V
measuring period during full cycle
nearly continuous measurement of transient fields
Time
c)
Potential difference between two points (secondary electric field)
Time

Figure 3.8: *TEM transmitting signal and induced currents for a common LOTEM survey. The source current signal and primary electromagnetic field of the transmitting signal versus time is shown in a). b) the corresponding transient earth responses which would be measured in induction coils several dipole lengths away from the transmitter. c) shows the normalized potential difference between two adjacent points at some offset to the transmitter. Further explanations are given in the text.*

Moreover the step response of a 100% duty cycle transmitter signal would be as twice as large compared to the one from 50% duty cycle. However, due to coupling effects between transmitter grounding and surrounding (e.g. for high conducting media like seawater), LOTEM signals should be measured within a 50% duty cycle as well.

Figure 3.9 shows the model (a) and three times three simulated LOTEM transients (b-d). The fields are generated using a common offset of $r = y = 2\,km$ using a standard broadside configuration. In a), a bipole transmitter of $1\,km$ length is shown on a resistive medium. The resistivity model consists of a simple two layer case with varying resistivities ρ_1 and ρ_2. Further dimensions of the model can be taken from the picture.

To gain insight into the LOTEM sounding curves, I calculated the transient responses at each receiver site for three different two layer models. Magnetic field transients are displayed as voltages per m^2 and given in b) and d) in a log-log graph over time t per seconds. The electrical field change parallel to the transmitter (E_x) over time is displayed in V/m (c). The black curve shows the transient responses of a $50\,\Omega m$ half-space ($\rho_1 = \rho_2$). To give a diffusion depth for the half-space model the transients are marked at four different times.

The red curve shows the transient response of a $10/1$ and the green curve shows the response of a $1/10$ contrast. To explain the time and conductivity dependence of the LOTEM E_x, $\dot{\mathsf{H}}_z$ and $\dot{\mathsf{H}}_y$ sounding curves, asymptotic derivations for the late and early time range can be

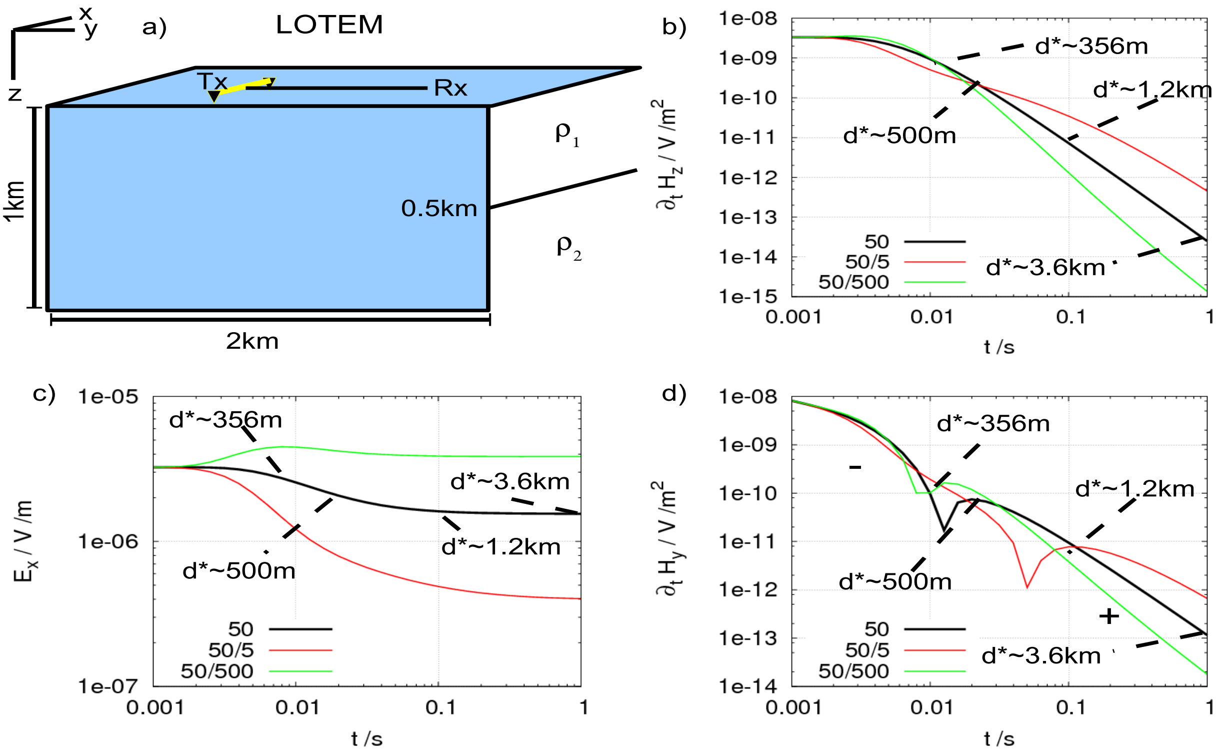


Figure 3.9: *a) Shows a simplified model of a typical LOTEM measurement. The grounded dipole transmitter is marked with a yellow line and the grounding points are indicated by triangles. The transient responses of a central loop* $\dot{\mathsf{H}}_\mathrm{z}$ *in a log-log graph over time for three different conductivity profiles are given in b). Further explanations are given in the text.*

calculated similar to the SHOTEM $\dot{\mathsf{H}}_\mathrm{z}$. After Kaufman & Keller [1983]:

$$\dot{\mathsf{H}}_\mathrm{z}^{et} \sim \rho r^{-4}, \qquad \dot{\mathsf{H}}_\mathrm{z}^{lt} \sim \rho^{-3/2} t^{-5/2} r \tag{3.38}$$

$$\dot{\mathsf{H}}_\mathrm{y}^{et} \sim -\rho^{1/2} t^{-1/2} r^{-3}, \qquad \dot{\mathsf{H}}_\mathrm{y}^{lt} \sim \rho^{-1} t^{-2} \tag{3.39}$$

$$\mathsf{E}_\mathrm{x}^{et,lt} \sim -\rho r^{-3} \tag{3.40}$$

Considering the $\dot{\mathsf{H}}_\mathrm{z}$-component first, it can be seen, that the transient decay is almost comparable to the in-loop transient as shown previously. Indeed, besides the r dependence, it can be shown that the LOTEM $\dot{\mathsf{H}}_\mathrm{z}$ and the SHOTEM $\dot{\mathsf{H}}_\mathrm{z}$ have comparable sounding curves. Although, the time range is different, they show the same time vs. conductivity with depth dependence: The early time range of the sounding curve can be identified for $t \leq 2\ ms$, whereas the transient is showing the late time dependence for $t \geq 0.2\ s$ (fig. 3.9 b).

In the $\dot{\mathsf{H}}_\mathrm{z}$ sounding curve of the $50/5\,\Omega m$ resistivity model shows another feature which can also be identified in the Central-Loop sounding curve: The voltage is decreased slightly before the diffusion depth of $500\,m$ is reached and crosses the half-space transient curve at the specified diffusion depth. Afterwards the voltage increases and reaches the late time decay behavior. Physically this can be understood, since the current flows into the conductor for early times and decreases the signal compared to a half-space transient. If the current system moves downward, the current resides in the top part of the conductor and increases the voltage for later times. To the opposite, the transient sounding over a resistive interface increases the voltage for early times due to the resistive interface whereas it decreases for late times as the current system travels faster in the resistor.

Only the horizontal magnetic component shows a time dependence at all times. Since the time behavior also depends on the resistivity it is not easy to distinguish the certain features. Remarkable is the sign change during intermediate times (fig. 3.9 d). It shows, that the $\dot{\mathsf{H}}_\mathrm{y}$ component bears a sign reversal, identified by the notch in the sounding curve, due to changes in the conductivity distribution. The fast decay of the signal in the magnetical receivers is due to the vanishing induced magnetic fields for the DC case. Note, that there is no r dependency for the horizontal component of the voltage specified in equation (3.39).

A remarkable feature of the E_x responses is the clear separation between the red and the green curve (fig. 3.9 c). The dynamic in time between $t \approx 2 - 100\, ms$ (intermediate times) can be described only qualitatively. The amplitude is exponentially muted from the doubled DC value to the DC level. Whereas there is no time dependence for early ($t \leq 2\, ms$) and late times ($t \geq 0.1\ s$). In the early and late time limits, it displays the resistivity of the underlying half-space. This feature can be clearly seen in the difference between the red and the green curve (fig. 3.9 c).

A comparison of the marked diffusion depths for each transient (b-d) reveals, that the depth of investigation derived by Spies & Frischknecht [1991] is valid for far zone soundings, too. It can be seen, that the underlying half-space is affecting each sounding curve clearly before the calculated diffusion depth.

Note, that the transients given above are calculated without consideration of the ramp time or systems impulse response.

3.2.3 System response

Any system, that in a large class is known as linear, is characterized by its impulse or system response: For any input function, the output function can be calculated in terms of the input and the systems impulse response.

The Laplace transform of the impulse response function is known as the transfer function. The Laplace transform of a system's output may be determined by the multiplication of the transfer function with the input function in the frequency domain. An inverse Laplace transform of this result will yield the output function in the time domain. To determine an output function ($\tilde{f}(t)$) directly in the time domain, it requires the convolution ($*$) of the input function ($f(t)$) with the impulse response function ($g(t)$). The convolution is defined as

$$\tilde{f}(t) = \quad (g * f)(t) \quad = (f * g)(t) \tag{3.41}$$

$$= \textstyle\int_{-\infty}^{\infty} g(t-t')f(t')\mathrm{d}t' \quad = \displaystyle\int_{-\infty}^{\infty} g(t')f(t-t')\mathrm{d}t' \ , \tag{3.42}$$

which is a commutative operator.

The numerical implementation of the convolution integral depends strongly on the data value definitions. A discrete convolution for data values which are equally linear sampled with Δ in N_t data points (e.g. $g_i = g(\Delta i)$, $f_i = f(\Delta i)$), can be carried out as a causal convolution like

$$\tilde{f}_i = \Delta \sum_{j=0}^{i} g_j f_{i-j} \quad \text{for } i = 0, \ldots, N_t \,, \tag{3.43}$$

or a a causal convolution like

$$\tilde{f}_i = \Delta \sum_{j=0}^{N_t} g_j f_{i-j} \quad \text{for } i = 0, \ldots, N_t \,. \tag{3.44}$$

The latter may be adjusted with a reflection around a symmetry axis to sum up like

$$\tilde{f}_i = \Delta \sum_{j=0}^{N_t} g_j f_{i+j} \quad \text{for } i = 0, \ldots, N_t \,. \tag{3.45}$$

As we can see, the numerical realization of the convolution has some peculiarities and might be difficult to implement.

A rough difference can be made between SHOTEM and LOTEM system responses. SHOTEM measurements are generally done with a square loop transmitter which has a step off characteristic like a linear ramp. The linear ramp can be treated analytically. LOTEM system responses can not be described with analytical functions. It has to be measured separately. Measuring the impulse response for the LOTEM system can be a challenging task which depends strongly on the configuration and the used receivers. For further information about the treatment of LOTEM system response measurements one may be referred to the works of Strack [1992]; Hördt [1992]; Hanstein [1992]; Helwig [2000]; Scholl et al. [2004].

For a proper treatment of the system response in the inversion process one has two possibilities.

- Either by convolution of the model response with the measured or calculated impulse response of the system, or
- by a deconvolution of the measured data.

Each of the mentioned treatments have advantages and disadvantages depending on the systems peculiarities.

Common TEM devices, limited by their digitizing units, do not allow to resolve the large dynamic of the transients over a big time scale. One therefore uses separate transient soundings with different time frames and adjusted transmitter sizes to increase resolution. Parameterized deconvolution (e.g. Hanstein [1992]) of the data may be the method of choice when one wants to treat the measured data for different transmitter settings as one data set. But, using a deconvolution means to alter the measured signal.

On the other hand one can treat every measured sounding separately in terms of system response, but make a joint inversion with the convoluted model responses. Since I am more used to the latter idea, to leave the measured signal as it is and to adjust the model response appropriately, I implemented the convolution of the model response with the systems (measured or calculated) impulse response in the inversion code. Because the numerical implementation of the convolution may not be a trivial task, the theoretical background for this treatment is given in the following.

Treatment of the systems impulse response by convolution of the model response

As already mentioned, the square loop wire acting as the SHOTEM transmitter has a self inductivity which can not be neglected. The inductivity of the wire causes a delay when switching off the current which is the systems step response. In the ideal case it can be described by a linear ramp $r(t)$ (see fig. 3.10)

$$r(t) = \begin{cases} I_0 & t < 0 \\ I_0 \left(1 - \frac{t}{t_r}\right) & 0 \leq t \leq t_r \\ 0 & t > t_r \end{cases} , \tag{3.46}$$

with the initial current amplitude I_0, the set off zero time t_0 and the ramp time t_r. Note, that the time derivative of the step response is equal to the impulse response. As is illustrated by figure 3.10, the derivative of the linear ramp $(r(t))$ is a rectangular function with time $(g(t))$.

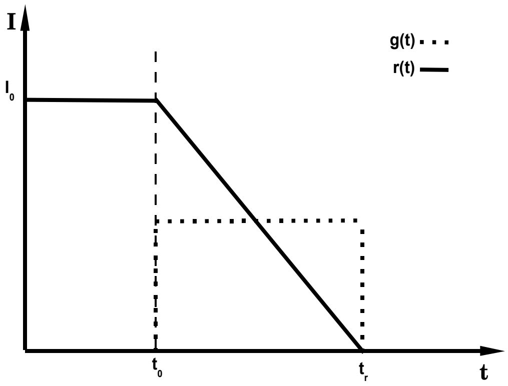


Figure 3.10: *Sketch of the step off ramp $r(t)$ and it's system response $g(t)$ over time. This type of system response is generally observed for rectangular transmitters.*

Note further, that the impulse response complies the systems response to a unit impulse $\delta(t)$ with $\int_{-\infty}^{\infty} \delta(t)\mathrm{d}t = 1$. Therefore, the impulse response which is calculated from the systems step response should be normalized to have unit area as well.

Taking the normalization into account, the impulse response of the ramp is then given by

$$\begin{aligned} g(t) &= \frac{\partial r(t)}{\partial t}\frac{1}{I_0} \\ &= \begin{cases} 0 & t < 0 \\ -\frac{1}{t_r} & t_0 \leq t \leq t_r \\ 0 & t > t_r \end{cases} \end{aligned} \tag{3.47}$$

which is a causal function for time t.

Consequently, the influence of the linear ramp on the SHOTEM data can be described by

$$\begin{aligned} \tilde{d}(t) &= \int_{-\infty}^{\infty} g(t')d^{\mathrm{step}}(t-t')\mathrm{d}t' \\ \text{(Eq. (3.47))} &= -\frac{1}{t_r}\int_{t_0}^{t_r} d^{\mathrm{step}}(t-t')\mathrm{d}t' . \end{aligned} \tag{3.48}$$

The implementation of equation (3.48) into the inversion code is done treating $g(t)$ as a rectangular function (see fig. 3.10). This is advantageous because treating it that way one must not distinguish between a measured or a calculated system response. Additionally, with this general treatment, the implemented algorithm can be applied for other measured system responses (e.g. LOTEM system response), too.

Because of its great dynamic, the model responses for TEM are generally calculated for logarithmic spaced time points. To achieve a fast but stable convolution scheme, several tests has been carried out (e.g. see section 3.5.3). The first idea was to make an interpolation of the logarithmic spaced data to linear spaced data and carry out the convolution according to equation (3.43) or (3.45). Dealing with huge data sets, this approach increased the overall calculation time to much and was therefore discarded. To reduce the numerical effort I implemented instead the convolution of logarithmic spaced data reading

$$\tilde{f}_i = \Delta \sum_{j=0}^{N_r} g_j Q(f(t_i - t_j)) \quad i = 1, \ldots, N_t \ , \tag{3.49}$$

where g_j are linear sampled data points of the system response to N_r times with the sampling interval length of Δ and N_t is the number of logarithmic spaced time points of the model response. $Q(f(t_i - t_j))$ denotes the model response interpolation to the time $t_i - j\Delta$. Q is an interpolation operator which may be carried out linear or via a cubic spline interpolation (e.g. Press et al. [1995]). Fortunately, the system response is commonly not very long ($50 < N_r < 500$) so that there are not so many additions have to be carried out.

Considering a ramp, e.g. to treat SHOTEM data, the g_j are given by $g_j = 1/N_r$ and the sampling interval length is the ramp time. Note, that the definition of the time point t_0 which accounts for the systems zero time depends on the used device. E.g. the GDP-32^{II} of Zonge Eng. Inc. defines the t_0 at the end of the ramp. Therefore a a causal integration has to be carried out which is achieved by changing the integration limits from $-t_r$ to t_0. To account for cases when $t_i - j\Delta < 0$ one may define ($t_0 = 0$) $Q(f(t_i - t_j)) = f(0)$ for $t_i - j\Delta < 0$.

As already mentioned, this general form of treating the system response by convolution of the model response can be carried out for measured data of other methods (e.g. LOTEM) in the same way. It does not rely on a linear ramp, which is a special case for rectangular loop transmitters, but may be applied to any methods where the system response can be measured or approximated.

3.2.4 Summary

Active transient electromagnetic methods utilize primary field changes to generate secondary fields in the subsurface. According to Faraday's law, the secondary fields try to perpetuate the primary fields and decay with ongoing time dissolving into the subsurface ("smoke rings").

TEM can be separated into two distinct classes, the far and the near zone applications. Among the various types of transmitter signals, the square wave form is the most common, and the secondary fields are the step response. The emitted signal may be further divided into the time where the recording system measures either electrical fields or induced voltages. Using a 50% duty cycle signal, the transient response is mostly recorded in the absence of the primary fields (step off response), but for the LOTEM method, the 100% duty cycle signal is normally used (step on response). However, the step on and step off response are connected through the DC value and may be transferred using equation 3.37.

For each method some characteristic sounding curves (transients) where given. With early and late time approximations, the transients show characteristic features which depend on the subsurface conductivity structure, decay time and, in case of LOTEM soundings, the offset.

If we are dealing with artificial sources, the impulse response or system response of the used devices and the source current term is affecting the measurement to no neglectible degree. To account this in the imaging process we have to treat either measured or calculated system responses. Therefore one can either try to remove the system response by deconvolution of the measured data, or convolve the model response with the system response. From numerical point of view, the latter seems to be more advantageous, because the data is left as it is. This approach is therefore preferred and promoted in this work.

3.3 3D forward solution with the *SLDMem3t*

In order to develop an inversion program which should fit a full 3D model response to gathered TEM data, it is necessary to solve Maxwell's equations for arbitrary spatial conductivity distributions. Despite the numerical effort of this task, it is necessary to solve it in finite computation time as well. This restricts the choice of eligible forward solution strategies to a very few.

Weidelt [2000] compared the most common approaches to model controlled source time domain electromagnetic data for 3D conductivity structures, namely

- Modeling in the frequency domain followed by a inverse Fourier transform into the time domain.
- Continuation of initial values by explicit time stepping (FDTD).
- Approximating the system matrix with the spectral Lanczos decomposition method (SLDM).

The application of the frequency domain CSEM approach was described first. Although excellent 3D frequency domain codes are freely available (e.g. from Randy Macky or Chester Weiss), the requirement of 3D frequency responses for a suitable number of logarithmically equidistant frequencies and their transformation to the time domain by a fast Hankel transform drastically speeds down the performance of this approach. Therefore the frequency domain modeling approach is very ineffective [Newman et al., 1986; Weidelt, 2000] and limits the feasible solvers to the ones which are operating directly in the time domain.

Integral equation (IE) approaches (e.g. Hohmann [1975]; Weidelt [1975]; Newman et al. [1986]) which are capable of computing high parameter contrasts are only feasible for a small number of anomal bodies. Weidelt [2000] favored either the explicit time-stepping scheme (presented by [Wang & Hohmann, 1993] for a inductive loop source or Commer & Newman [2004] for a galvanic coupled grounded wire) or the SLDM approach by Druskin & Knizhnermann [1988; 1994; 1999]. Recently, a direct approach using a finite volume technique combined with sophisticated preconditioning (iLU) approach was presented by Haber & Ascher [2001]; Haber et al. [2007].

Assuming a shut-off of sources at $t = 0$. Within the FDTD approach, the values for times $t > 0$ are achieved by finite difference continuation of the initial fields ("leap-frog"). For the solution of the diffusion equation (3.8) this method is stable only for very small time steps, which is due to the Courant-Friedrichs-Levy (CFL) criterion (e.g. [Richtmeyr & Morton, 1967])

$$c \leq \frac{\Delta}{\sqrt{3}\Delta t} \tag{3.50}$$

Equation (3.50) approximates a wave equation in which the wave speed c is automatically satisfying the CFL condition. This imposes an upper limit on the time step Δt, which can increase in time but slows down the performance of FDTD for late time responses. This approach can also be found in recently developed commercial solvers (e.g. COMSOL) where the method is expanded to a finite element formulation (FE). The advantage of this formulation is, that unstructured grids can be applied [Löhken, 2007].

The SLDM approach is assumed to be the most effective approach to solve the transient electromagnetic problem. Within the SLDM approach, the main conservation of computational load is achieved by solving the most relevant part of the underlying linear system by means of a spectral decomposition after Lanczos [1950]. Herein, the initial values are approximated by a system of orthogonal decay modes ("Ritz vectors") with decay constants ("Ritz values"), resulting as eigenvectors and eigenvalues from a low dimensional Krylov subspace approximation of the high dimensional system matrix. This approach depends strongly on the condition number and the Courant number of the system matrix. Within a finite difference formulation of the linear system, this can become a drawback for regions with high resistivity contrasts and big field gradients. As a consequence of the Lanczos phenomena, the decay constants can become slightly negative and destroy the orthogonality of the decay modes [Weidelt, 2000]. Thus, the dimension of the subspace, required for convergence, increases with time and complexity of the model.

Weidelt [2000] found, that both approaches produce acceptable results, whereas the latter is more efficient for modeling transient fields, due to the less amount of computation time. Other comparisons and reviews of different TEM modeling programs can be found in Hördt et al. [1992]; Hördt [1992]; Mitsuhata [2000]; Commer [2003]; Scholl et al. [2004]. However, in both cases, the computational load increases with $\sqrt{t}$, such that the late time transient response is more difficult to achieve. In FDTD this results in many time steps, where for the SLDM a greater dimension of the Krylov subspace is needed.

Because of the efficiency regarding computation time, all 3D model response in this work where computed using the SLDM code by Druskin & Knizhnermann [1988], which is called *SLDMem3t*. To explain how the program works and what difficulties are to overcome to produce acceptable results within an inverse process, a quick insight into the program are given.

To solve the 3D forward problem, the electric and magnetic fields are sampled on Yee-Lebedev staggered grid after Yee [1966] applying a finite difference formulation. The FD approach is widely used in EM methods due to the apparent simplicity of its numerical implementation. Figure 3.11 shows a sketch of the Yee-grid. Note, that the electrical field are sampled at the edge-centers and the magnetical fields are sampled on the plane-centers

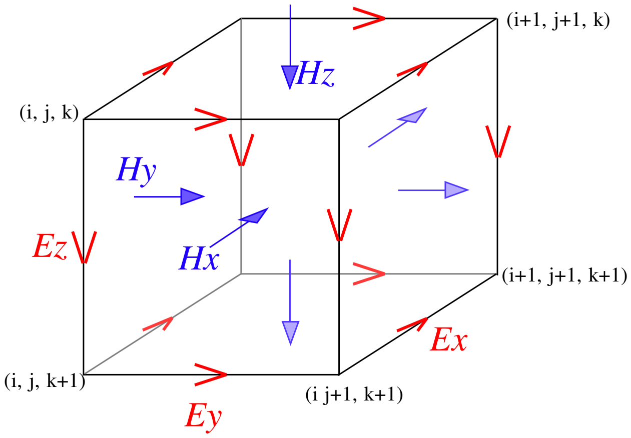


Figure 3.11: *Sketch of a Yee-grid cell. The electrical fields are sampled along the edges of the cell. The magnetic field values are calculated here at the center of the cell by forming the curl of the surrounding electrical fields.*

of the Yee-cell. The main attraction of the FE approach is that it is believed to be better able to accurately account for subsurface geometry, because of its unstructured grid. However, to discuss some numerical peculiarities of *SLDMem3t*, it is convenient to give a overview how the algorithm works.

The objective is to solve the Maxwell equations in the model domain Ω:

$$\Omega = \left\{ X : x_0^i < x^i < x_{N_i}^i \right\}, \quad i = x,\, y,\, z \tag{3.51}$$

with the boundary conditions given by

$$\hat{\mathbf{n}} \times \mathbf{E}|_{\partial\Omega} = 0\,; \qquad \hat{\mathbf{n}} \times \left.\frac{\partial \mathbf{H}}{\partial \hat{\mathbf{n}}}\right|_{\partial\Omega} = 0 \tag{3.52}$$

where $\hat{\mathbf{n}}$ is the normal vector to the surface boundary $\partial\Omega$ of Ω. The initial condition is given by

$$\mathbf{E}|_{t<0} = \mathbf{H}|_{t<0} = 0 \tag{3.53}$$

The domain Ω is further discretized into 3 N-tuples, forming a regular grid. The grid can be a Cartesian (fig. 3.12, left) or a rectilinear grid (fig. 3.12, right). The node points in the regular grid can be addressed via

$$x_{ijk}\,, \quad i = 1, \ldots, N_x,\; j = 1, \ldots, N_y,\; k = 1, \ldots, N_z \tag{3.54}$$

The edge centers are referred at

$$\begin{aligned} \hat{x}_{i+\frac{1}{2}jk} &= \left(x_{ijk} + x_{i+1jk}\right)/2 \\ \hat{x}_{ij+\frac{1}{2}k} &= \left(x_{ijk} + x_{ij+1k}\right)/2 \\ \hat{x}_{ijk+\frac{1}{2}} &= \left(x_{ijk} + x_{ijk+1}\right)/2, \qquad i = 1, \ldots, N_x,\; j = 1, \ldots, N_y,\; k = 1, \ldots, N_z \end{aligned} \tag{3.55}$$

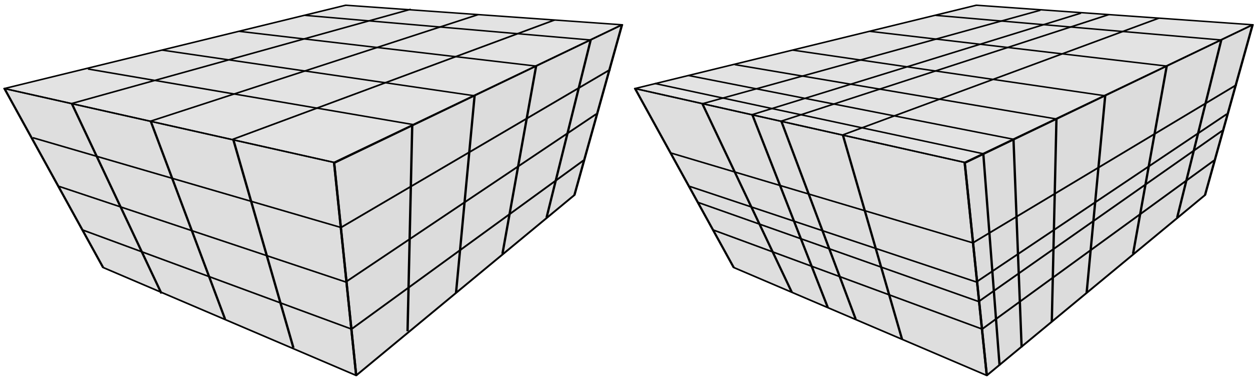


Figure 3.12: *Display of common regular grids. In the left row a Cartesian grid is shown. It constitutes of a linear discretization of the model domain. The right row exhibits a rectilinear grid. Note that the discretization here is not linear, but may be logarithmically.*

Which is numerically implemented using the fictitious nodes

$$
\begin{aligned}
\hat{x}_{1-\frac{1}{2}jk} &= x_{1jk} - \frac{x_{2jk}-x_{1jk}}{2} \\
\hat{x}_{N_x+\frac{1}{2}jk} &= x_{N_xjk} + \frac{x_{N_xjk}-x_{N_x-1jk}}{2}, \qquad j=1,\dots,N_y,\ k=1,\dots,N_z \qquad (3.56) \\
\hat{x}_{i1-\frac{1}{2}k} &= x_{j1k} - \frac{x_{i2k}-x_{i1k}}{2} \\
\hat{x}_{iN_y+\frac{1}{2}k} &= x_{N_xjk} + \frac{x_{iN_yk}-x_{iN_y-1k}}{2}, \qquad i=1,\dots,N_x,\ k=1,\dots,N_z \qquad (3.57) \\
\hat{x}_{ij1-\frac{1}{2}} &= x_{ij1} - \frac{x_{ij2}-x_{ij1}}{2} \\
\hat{x}_{ijN_z+\frac{1}{2}} &= x_{ijN_z} + \frac{x_{jkN_z}-x_{ijN_z-1}}{2}, \qquad i=1,\dots,N_x,\ j=1,\dots,N_y\ . \qquad (3.58)
\end{aligned}
$$

Note that, besides the practical advantage regarding it's implementation, the fictitious nodes are also approximating the Neumann boundary condition stated in (3.52) for $\mathbf{H}$ with a second-order error.

FD operators and system matrix

With the definitions given above, one can now introduce the finite difference operators working on the nodal field values

$$
\nabla_{ijk}\mathbf{f} = \begin{pmatrix} \frac{f_{i+1jk}-f_{ijk}}{x_{i+1jk}-x_{ijk}} \\ \frac{f_{ij+1k}-f_{ijk}}{x_{ij+1k}-x_{ijk}} \\ \frac{f_{ijk+1}-f_{ijk}}{x_{ijk+1}-x_{ijk}} \end{pmatrix}, \quad i=1,\dots,N_x\ j=1,\dots,N_y\ k=1,\dots,N_z \qquad (3.59)
$$

or on the edge field values

$$
\hat{\nabla}_{ijk}\mathbf{f} = \begin{pmatrix} \frac{f_{i+\frac{1}{2}jk}-f_{i-\frac{1}{2}jk}}{\hat{x}_{i+\frac{1}{2}jk}-\hat{x}_{i-\frac{1}{2}jk}} \\ \frac{f_{ij+\frac{1}{2}k}-f_{ij-\frac{1}{2}k}}{\hat{x}_{ij+\frac{1}{2}k}-\hat{x}_{ij-\frac{1}{2}k}} \\ \frac{f_{ijk+\frac{1}{2}}-f_{ijk-\frac{1}{2}}}{\hat{x}_{ijk+\frac{1}{2}}-\hat{x}_{ijk-\frac{1}{2}}} \end{pmatrix}, \quad i=1,\dots,N_x\ j=1,\dots,N_y\ k=1,\dots,N_z \qquad (3.60)
$$

are then introduced, to form the curl-curl operator in a finite difference scheme.

With the appropriate grid and node operators of equations (3.60) and (3.59), the grid analogue of equation

$$\nabla \times \nabla \times \mathbf{E} + \mu\sigma \frac{\partial \mathbf{E}}{\partial t} = -\mu \dot{\mathbf{J}}_e$$

can now be defined. For simplicity, we suppose we are interested in only $\mathbf{f} = \mathbf{f}_{e,m} = \{\mathbf{E}, \mathbf{H}\}$

$$\mathbf{A}\mathbf{f} + \dot{\mathbf{f}} = f_{0+} \ , \tag{3.61}$$

whereas the system or operator matrix $\mathbf{A}$ is given through

$$\mathbf{A}\mathbf{f} = \nabla \times (\hat{\nabla} \times \mathbf{f}). \tag{3.62}$$

The governing PDE (3.61) is solved with the initial condition

$$\mathbf{f}_{0+} = \delta(t)\mathbf{f}_0 \tag{3.63}$$

and the boundary condition given by equation (3.52).

Material averaging scheme

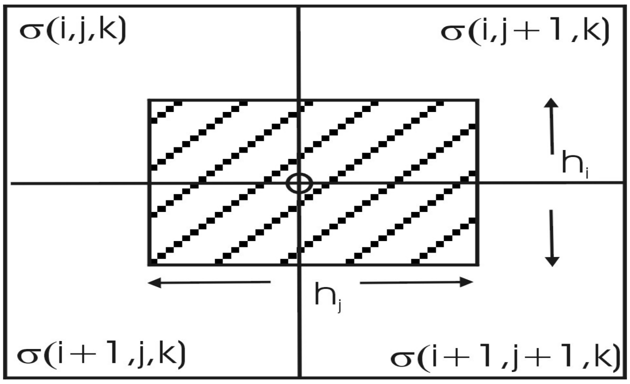


Figure 3.13: *Plane view of the modeling domain and the regular grid. The hatched rectangle shows the effective area of the field value at the grid node ijk. The conductivity is discretized around the node point.*

The linear system (eq. (3.61)) is still depending on the physical parameters. But it can be transformed to a system where the physical properties are already assembled within the field values. This is achieved by the material averaging (MA) scheme, first introduced by Yee [1966]. Figure 3.13 shows how the MA-scheme is applied within a finite difference formulation. The field values at any nodes are weighted by the conductivity value of each adjacent cell multiplied by it's effective area. By the throughout application of this at any node, the grid values become independent from the conductivity distribution and hence, the underlying model can be arbitrary discretized. For simplicity figure 3.13 shows a 2D-sketch of the model domain. The MA-scheme is extended for the 3D case by applying it in any direction.

With the MA-scheme, the transformed system ($\mathbf{f} \rightarrow \hat{\mathbf{f}}$)

$$\mathbf{A}\hat{\mathbf{f}} + \dot{\hat{\mathbf{f}}} = -\frac{1}{\sigma}\hat{f_{0+}}, \quad \mathbf{A}\hat{\mathbf{f}} = \frac{1}{\sigma}\nabla \times \left(\frac{1}{\mu}\hat{\nabla} \times \hat{\mathbf{f}}\right) \tag{3.64}$$

is formulated, which has the same solution as equation (3.61) [Druskin & Knizhnermann, 1994].

Summarizing some major attributes of $\mathbf{A}$:

- $\mathbf{A}$ is symmetric and
- positive definite.
- $\mathbf{A}$ is high dimensional ($\mathsf{dim}(\mathbf{A}) = N_{\mathbf{A}} \approx 3N_xN_yN_z \approx 10^5 - 10^6$), but
- sparse with 13 entries per row/column (apart from the boundary).

If $\mathbf{v}_n$ are the eigenvectors and λ_n are the eigenvalues of $\mathbf{A}$, then the exact solution to equation (3.64) is

$$\mathbf{f}(t) = \sum_{n=1}^{N} \left(\mathbf{v}_n^{\mathrm{T}}\mathbf{f}_0\right) \mathbf{v}_n \mathrm{e}^{-\lambda_n t} \ . \tag{3.65}$$

In practice, however, the eigenvectors and eigenvalues of this high dimensional matrix are difficult to obtain. Therefore a recursion to approximate the solution is required.

The aim of the SLDM process is to seek the approximate solution to equation (3.65) in the Krylov space.

3.3.1 Approximate solution in Krylov space

The approximate solution to equation (3.64) is sought in an M-dimensional subspace of the $N_{\mathbf{A}}$-dimensional space spanned by $\mathbf{A}$. This subspace is defined as

$$\mathbf{K}^{\mathsf{M}} = \left\{\mathbf{f}_0\mathbf{A}^0, \ldots, \mathbf{f}_0\mathbf{A}^{\mathsf{M}-1}\right\} \ , \tag{3.66}$$

as suggested by the exact solution of equation (3.64)

$$\mathbf{f}(t) = \mathsf{e}^{(-t\mathbf{A})}\mathbf{f}_0 = \sum_{m=0}^{\infty} \frac{1}{m!}(-t\mathbf{A})_m \mathbf{f}_0 \ . \tag{3.67}$$

Although this series is convergent for all $t \geq 0$, it is applicable only for moderate values of $t\lambda_1$. A more precise insight into the computation of matrix exponentials can be found in Moler & van Loan [1978].

Lanczos process

From the M Krylov vectors is formed an orthonormal basis (ONB) in the Krylov space

$$\mathbf{Q} := (\mathbf{q}_1, \ldots, \mathbf{q}_{\mathsf{M}})^{\mathrm{T}} \ , \tag{3.68}$$

which is formally achieved by the elegant Lanczos process [Parlett, 1980]. It is based on the observation that the sparse matrix $\mathbf{A}$ can be approximated by

$$\mathbf{A}\mathbf{q}_i = \mathbf{q}_{i-1}\beta_{i-1} + \mathbf{q}_i u_i + \mathbf{q}_{i+1}\beta_i \ . \tag{3.69}$$

The ONB can than be formed using a Gram-Schmidt scheme. A schematic implementation of the Lanczos process can be found in algorithm 1

Algorithm 1 Low dimensional approximation of $\mathbf{A}$

```
Require: $\mathbf{r}_0 := \mathbf{f}_0 \quad \beta_0 = |\mathbf{f}_0| \neq 0$
  for $i = 1$ to $\mathsf{M}$ do
    $\mathbf{q}_i = \mathbf{r}_{i-1}/\beta_{i-1}$
    $\mathbf{u}_i = \mathbf{A}\mathbf{q}_i$
    $\mathbf{r}_i = \mathbf{u}_i - \mathbf{q}_{i-1}\beta_{i-1}$      Residuum
    $\alpha_i = \mathbf{q}_i^{\mathrm{T}}\mathbf{r}_i$
    $\mathbf{r}_i = \mathbf{r}_i - \alpha_i\mathbf{q}_i$      updating the residuum
    $\beta_i = |\mathbf{r}_i|$
  end for
```

This approximation of $\mathbf{A}$ leads to

$$\mathbf{A}\mathbf{Q} = \mathbf{Q}\mathbf{T} + \widetilde{\mathbf{R}} \ . \tag{3.70}$$

With equation (3.70), equation (3.64) is reduced to a much simpler problem, because $\mathbf{T}$ is a tridiagonal matrix:

$$\mathbf{T} = \begin{pmatrix} u_1 & \beta_1 & & & & \\ \beta_1 & u_2 & \beta_2 & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & \beta_{\mathsf{M}-2} & u_{\mathsf{M}-1} & \beta_{\mathsf{M}-1} \\ & & & & \beta_{M-s1} & u_{\mathsf{M}} \end{pmatrix} \tag{3.71}$$

and $\widetilde{\mathbf{R}}$ is the reminder matrix which is neglected.

Summarizing the above this leads to the approximation

$$\mathbf{f}(t) \cong \sum_{j=1}^{\mathsf{M}} \mathbf{q}_j \sum_{m=1}^{\mathsf{M}} \gamma_m \hat{s}_{jm} \mathsf{e}^{-\theta_m t} \tag{3.72}$$

where $\hat{\mathbf{s}}_m$ and θ_m are the eigenvectors and eigenvalues of $\mathbf{T}$ and $\gamma_m = \mathbf{f}_0 s_{1m}$. The whole procedure is also known as the spectral Lanczos decomposition method, introduced by [Druskin & Knizhnermann, 1988].

A closer look at the solution reveals some important connections between the SLDM process and the CG. In exact arithmetic, the solution obtained after M steps with the bi-conjugate gradient method (BCG), which can be found e.g. in Saad [1996], is also the Galerkin solution in the same Krylov subspace. Actually, the SLDM solution coincides with that of BCG, except that the SLDM process allows one to obtain solutions for many frequencies or times using the same matrices $\mathbf{Q}$ and $\mathbf{T}$ computed in real arithmetic, as well as to take care of the spurious modes. The only extra price to be paid is the storage of the components $\mathbf{Q}$ for all the space points where the field is to be computed.

Convergence estimates

Following Druskin & Knizhnermann [1994], the convergence estimation of the SLDM algorithm depends on two major features of the linear system. They are given by the

- smallest eigenvalue: $\theta_1 \geq \frac{\pi^2}{d_{max}^2 \sigma_{max} \mu_{max}}$ and
- the largest eigenvalue: $\theta_{\mathsf{M}} \leq \frac{13}{d_{min}^2 \sigma_{min} \mu_{min}}$,

where d_{max} and d_{min} are the largest and smallest dimension of the discretization. This values influence the conditioning of the linear system, since the condition number κ of a *normal* matrix is defined as ratio of the biggest to the smallest eigenvalue:

$$\kappa(\mathbf{A}) = \left| \frac{\theta_{max}}{\theta_{min}} \right| = \left| \frac{\theta_{\mathsf{M}}}{\theta_1} \right| . \tag{3.73}$$

However, in real arithmetics it is necessary to drop all eigenvalues which are smaller than θ_1. This is due to the numerical approximation and the feature, that the decay constants θ_m can become even slightly negative. Negative decay constants are found, when the orthonormality is lost, which is generally observed when applying the Lanczos method. Especially, for large diffusion times, the application of a *shift and invert* technique [Druskin & Knizhnermann, 1999] reduces computational effort.

The computational expense depends on the dimension N of $\mathbf{A}$ and the dimension M of the approximated subspace. The arithmetical cost is mostly due to the M fold computation of the product $\mathbf{Aq}_j$ in the Lanczos process. For fixed N the efficiency therefore depends on the dimension of M. Druskin & Knizhnermann [1988; 1994] show that M can be evaluated by

$$\mathsf{M} \geq \mathsf{M}_{est} = \frac{4}{d_{min}} \sqrt{\frac{t_{max}}{\mu_0 \sigma_{min}}} , \tag{3.74}$$

in exact arithmetics.

Since the above given inequality is just a rough measure for the dimension of M, the convergence process is measured within the program in order to stop the Lanczos process if a desired $\epsilon > 0$ is reached. This is achieved by introducing a control receiver point *ncr*, a first step point *it*1 and a increment *it*2 into the *SLDMem3t* code [Druskin & Knizhnermann, 2000]: At the control receiver *ncr* the solution is assembled for the last time point t_{max} in *it*2 steps beginning at *it*1.

$$f_i =\cong \mathbf{q}_{j=ncr} \sum_{m=1}^{i} \gamma_m s_{(j=ncr)m} \mathsf{e}^{-\theta_m t_{max}} \quad i = it1 + k * it2,\ k = 0, \ldots . \tag{3.75}$$

The convergence criteria $(i > 3)$ is reached if

$$\hat{\epsilon}_i = \frac{|f_{i-1} - f_i|^2}{f_i \left(|f_{i-2} - f_{i-1}| - |f_{i-1} - f_i| \right)} \leq \epsilon . \tag{3.76}$$

However, since the convergence process could not be controlled at every point in the grid the choice of *ncr* seems erratic. In practice it has shown, that it sometimes makes sense to set $\epsilon = 0$ and set $\mathsf{M} \gg \mathsf{M}_{est}$ as a hard boarder of the Lanczos process. But, to set up a hard boarder and discard the convergence control given by equation (3.76) means, even if the solution can be achieved in less iterations, there is no way to reduce computational effort. Additionally, the estimate for M should be more precise.

3.4 Grid design for *SLDMem3t*

Convergence and stability problems are widely known for the computation of 3D EM processes. The application of the SLDM makes this sometimes worse. The solution to equation (3.64) is sought in a low dimensional subspace without a control mechanism for grid design, like the CFL criteria which is used by time stepping schemes. Therefore the grid design for the *SLDMem3t* is assumed to be a crucial task when applying it [Hördt et al., 1992; Druskin & Knizhnermann, 1994].

In the past, a lot of trial and error analysis to find appropriate grids for model studies with *SLDMem3t* was carried out [Hördt et al., 1992; Hördt, 1992; 1998; Schaumann, 2001]. In the mentioned studies, the authors used one or just a few different grids, which where designed mainly for one specific model or TEM setup. Hence, the grid was restricted to a specific kind of problem. Unfortunately, none of the authors proposed in detail how they designed the grid or proposed any algorithm for automated grid design.

Within an inverse process, one normally solves numerous forward problems for different models. Therefore it is likely to have some kind of automated discretization scheme, or, at least have already tested discretization techniques available.

Although, the first application of *SLDMem3t* within the inverse process was presented by Commer [2003], the author does not reported for any automated adjustments to the grid within the inverse process. Additionally, until now, there are nor detailed grid studies with *SLDMem3t* presented. Therefore, I present an algorithm, it's implementation and a detailed grid analysis in the following.

Automated grid design uses some kind of algorithm which should be capable of adjusting the grid as the model changes. But, as already mentioned, there is no exact criteria for every step size but just the first and the last and the way how to discretized when using the *SLDMem3t* is purely user dependent. The question is then, how coarse or fine a grid must be to get stable results. Some people use e.g. linear spacing when multiple receiver spreads are to be calculated in the region of the receiver stations. This results in many grid nodes and with a fine discretization within regions where no fine discretization is needed, e.g. at the bottom of the model. This grid design would artificially increase the condition number of **A**. Hence, the introduction of grid nodes, when there is no need for this, is considered prohibitive.

On the other hand, the field values are computed on the staggered Yee-Lebedev grid and thus *SLDMem3t* computes the needed field values either between grid node points or at grid centers. Additionally, for all 6 field values (e.g. three electrical and three magnetical components), for the node point ijk they are sampled at different positions, according to their orientation. Therefore, to compute grid independent field values, one has to set up a grid interpolation operator as well (see section 3.5.3).

However, the inversion process demands a user independent automatic grid design. The governing values for an automated grid design to deliver accurate forward solutions with *SLDMem3t* within the inversion process can be summarized like

1. The smallest and biggest conductivities of the model .

2. The latest and earliest times to compute.

3. The number of grid nodes.

The implementation of the governing values into the grid automate *make_ sldm_ grid* and how different discretization mechanisms for bipole (e.g. LOTEM) and rectangular loop transmitters (e.g. SHOTEM) are achieved, is given in 3.4.1. There I introduce and test two different discretization algorithms for regular grids. The test is done by comparing the two described grid algorithms by checking how they particularly perform in terms of iteration numbers of the Lanczos process and the accuracy of the solution. As a result, a modified log spaced discretization seems to perform substantively better than a common logarithmic spacing.

Having a stable discretization algorithm at hand one can now investigate more internal questions:

- How many grid values should be introduced to get a long term and overall stable grid for different models?
- Is there any major dependency if one uses finer discretization in a preferred direction, (e.g. vertical direction [Hördt et al., 1992])?

This questions demand an extended grid evaluation tests to get a sincere answer to the questions above. It is reported in section 3.4.2.

3.4.1 Grid automate *make_ sldm_ grid*

Since the *SLDMem3t* does not solve the PDE through time stepping, the Courant Friedrichs Levy criteria does not hold to get stability. To achieve proper convergence and accurate results, the discretization for *SLDMem3t* may be done in several ways. Commonly, in order to get a stable solution, one generally does a trial and error analysis which can not be carried out within an inverse process.

From the convergence estimations it is likely to have a minimum and maximum grid spacing.

$$d_{min} = a_g \sqrt{\frac{t_{min}}{\pi \mu_0 \sigma_{max}}} \tag{3.77}$$

and

$$d_{max} = b_g \sqrt{\frac{t_{max}}{\pi \mu_0 \sigma_{min}}} \tag{3.78}$$

Each one is directly related to the diffusion depth and depend on the minimum/maximum conductivities and times used in the model and setup. To increase the variability the factors a_g and b_g are introduced Hördt [1992]. In a heuristic investigation Hördt [1992] proposed $a_g = b_g = 3$ for which he gets results within good agreement of other (analytical and 3D) solutions.

The factors just account for a bigger variation of the model which shows good results, even if the model is not known before (e.g. during the inverse process). During some initial testing of the grid automate with *SLDMem3t*, I discovered that taking $a - g \approx 1$ and $3 \leq b_g \leq 9$ produce excellent results as well. Generally it is observed, that applying a_g and b_g increases the corresponding conductivity range which increases capabilities of the code. The dynamic of the model during the inverse process demands stability to the solution, which means, that the response should be computed correctly for arbitrary changed models. If the solution of a grid becomes inaccurate, e.g. during a line search, the influence on the inverse process is quite remarkable.

For the minimum and maximum conductivity values of the model I selected them in the way, that they should be calculated (more or less) independent of their distribution. To implement values for the smallest and biggest model parameter into the inversion code, I normally give hard boarders (sigmin, sigmax) which should not be exceeded by the grid generator to avoid unnecessary complicated grids. This is achieved by using the minimum and maximum array value functions MIN(**arg**), MAX(**arg**).

$$\sigma_{min} = \mathsf{MAX}\left(\mathsf{sigmin}, \mathsf{MIN}(\mathbf{m})\right), \quad \sigma_{max} = \mathsf{MIN}\left(\mathsf{sigmin}, \mathsf{MAX}(\mathbf{m})\right) \ , \tag{3.79}$$

which are intrinsic *Fortran 90* functions.

Algorithm 2 Log spaced rectilinear grid

```
Require: $0 < d_{min} < d_{max}$
Require: $N_g > 0$
  $q = \left(\frac{d_{max}}{d_{min}}\right)^{\frac{1}{N_g - 1}}$
  for $i = 1$ to $N_g$ do
    $g_i = d_{min} q^{i-1}$
  end for
```

To account for the exponential behavior of the underlying PDE, best results are obtained for rectilinear grids with logarithmic spacings as it is also reported by Hördt [1992]. For my implementation into *make_sldm_grid*, the interval between d_{min} and d_{max} is divided logarithmically equidistant for N_g values. This can be achieved with a simple log-spaced pseudo code which is given in algorithm 2.

A closer look to algorithm 2 reveals a problematic case for the first values after d_{min}. Since, the stepsize is suited to divide the interval into N_g frames, the first steps after d_{min} can be smaller than the initial value. This leads to discretization steps $\Delta < d_{min}$ which should be omitted. Therefore, one wants to achieve a discretization increasing in d_{min} steps. A pseudo code for a proper treatment of this is given in algorithm 3. The increment in algorithm 3 is now treated as incremental sum of d_{min} with additional pieces of the interval $[d_{min}, d_{max}]$. A similar discretization technique is also described roughly, but not in detail by Hördt [1992].

Algorithm 3 Modified log spaced rectilinear grid

```
Require: $0 < d_{min} < d_{max}$
Require: $N_g > 0$
  $q = \left(\frac{d_{max}}{d_{min}} - (N_g - 1)\right)^{\frac{1}{N_g - 1}}$
  for $i = 1$ to $N_g$ do
    $g_i = d_{min}\left((i - 1) + q^{i-1}\right)$
  end for
```

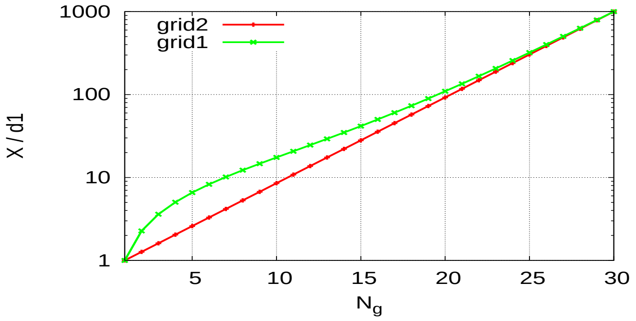


Figure 3.14: *Comparison of two different logarithmic discretization strategies. The normalized distance of the grid value $x/d_1 = g_i/d_{min}$ is plotted against the number of grid node $i = [1, \ldots, N_g]$. grid1 follows algorithm 3 whereas grid2 uses algorithm 2.*

A comparison of the discretization following algorithm 2 and 3, are displayed in Figure 3.14. As one can see from the illustrated graphs, the interval $[d_{min}, d_{max}]$ is always divided into N_g pieces, but with different spacings. The spacing for grid1 is substantially larger for the first grid values, whereas for the end points, the discretizations lead to almost the same spacings. As shown in the following, the second discretization algorithm 3 seems more appropriate for our aims. Therefore it is referred to as grid1 whereas the simple log spaced discretization following the first algorithm, is denoted grid2.

Given a discretization scheme that allows for a log spaced division of the interval between the interval $[d_{min}, d_{max}]$, the total number of grid points N_g becomes a leading value for grid design. For a 3D grid I refer to the horizontal number of grid nodes as N_x and N_y, whereas for the vertical direction I speak of N_z in the following. If just N_g is given, than all values are taken the same ($N_x = N_y = N_z = N_g$) which is referred to as a *symmetric grid* in terms of total number of grid points.

Further, the modeling domain to compute ground based TEM transient sounding curves is substantially different discretized than for models where the transmitter is found in a conducting surrounding (e.g. marine TEM setup). The question of how to discretized for different source terms (e.g. bipole or rectangular loop transmitters) and how to implement it becomes a necessity to answer. Since a prolonged electrical dipole (bipole) is constructed as a summation of single dipole sources it is convenient to start with a discretization example for the single dipole.

For simplicity I assume the mid point of the transmitter as origin, which makes the definition of the receivers more easy. Of course, any grid can be shifted towards every point in another model by a simple translation.

Dipole discretization

For ground based TEM setups, the horizontal directions would be discretized as illustrated in figure 3.15 (left). It displays a x-oriented dipole source in the origin which is marked with a red double arrow. Note, that the discretization is tight around the introduced current source, becoming wider at more distance to the source. This should account for the biggest EM field gradients in the model.

As traditionally found in Yee-Lebedev FD schemes, the dipole is discretized as current line

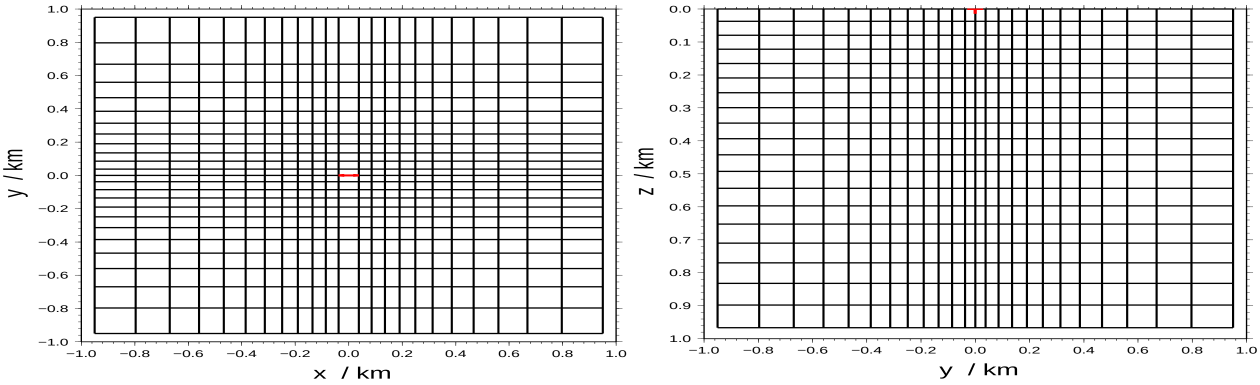


Figure 3.15: Two frames of a dipole discretization for ground based TEM. A xy-section of the modeling domain is displayed to the left, whereas the yz-plane is illustrated to the right. The source point of a single x-oriented dipole is denoted with a red double arrow (left) and a red cross indicating current flow into the plane. Further explanations are given in the text.

between two adjacent cells. The current is flowing according to the orientation of the dipole. Hence, the source can not be set up as a real point, but always as a vector with the dipole length $\widetilde{\mathbf{D}}$. For e.g., the dipole length of the x-oriented dipole would be $\widetilde{\mathbf{D}} = D_x = I\Delta x$. For this example I took $\Delta x = 2d_{min}$.

Note, that there is a gap in the x-discretization at the center of the dipole. It is spared out e.g. in order to calculate the E_x-component in the equatorial plane of the dipole. The horizontal discretization is done as reflection of a underlying first grid (e.g. using algorithm 3) at the source origin yielding the negative part of the model domain (cf. fig. 3.15 (left)).

The vertical direction is discretized just by taking the underlying grid1. Illustrated in figure 3.15 (right) for ground based TEM, I normally discretized just the conducting subsurface. As inherent feature of the *SLDMem3t*, the air does not need to be discretized for this case, because the magnetical field components can be calculated at the air interface by upwards continuation [Druskin & Knizhnermann, 1994].

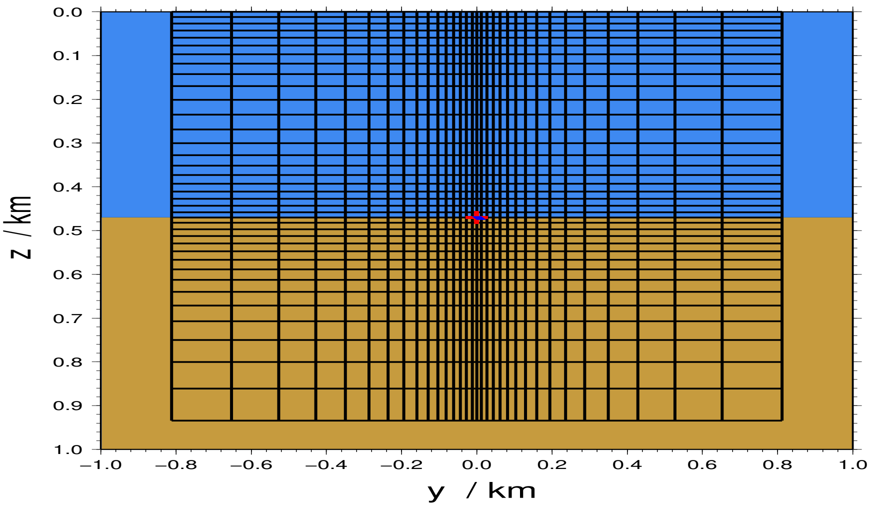


Figure 3.16: Example of a dipole transmitter discretization within a conducting medium. The source point of a single x-oriented dipole is denoted with a red cross. Further explanations are given in the text.

However, for models where the transmitter is found within a conducting medium, or if there are topographical effects needed to take into account, the discretization will become different. In order to compare a vertical discretization scheme which accounts for a different surrounding, a typical marine TEM setup is displayed in figure 3.16. Here, the sea is discretized from the seabottom to the air given in a blue background color. Lightbrown color is referring to the sea bottom which extends from $z = 480\,m$ to infinity. Compared to a ground based z-discretization, displayed in figure 3.15 (right), the spacings between source and air are increasing with distance from the source and again decreasing to the air/sea interface. This is done to account for increasing EM field gradients at the air boundary. The shown model is given as an example using *make_ sldm_ grid* for a marine setup.

Another case would be to take any topographical effects into account. E.g. Commer [2003] successfully used *SLDMem3t* to interpret LOTEM measurements within the volcanic terrain of mount Merapi (Indonesia). Herein, the air is discretized taking a second grid based on a very high resistivity value for the air (e.g. $\sigma_{min} = 10^{-6}\ S/m$) leading to very large discretization steps for the vertical air discretization.

LOTEM bipole discretization

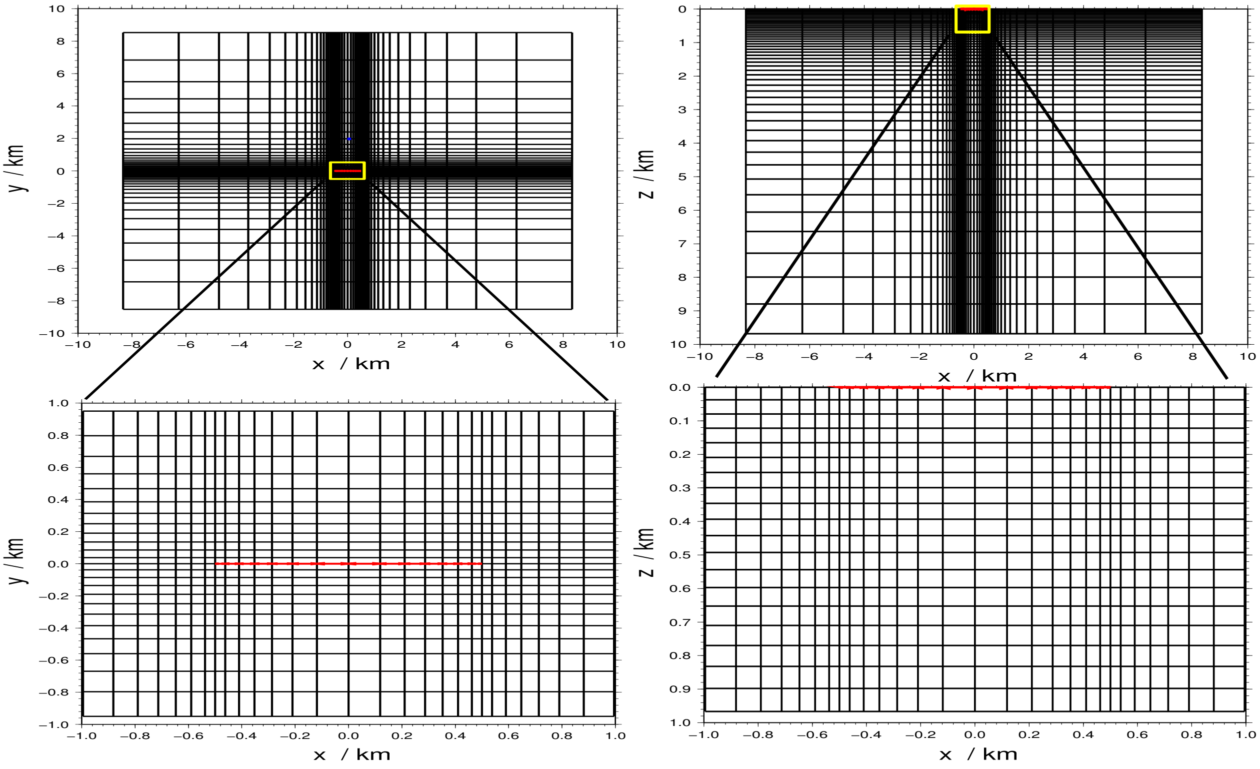


Figure 3.17: *Typical bipole discretization of a LOTEM setup. xy-plane for two different scales are displayed in the left row. The upper part shows the whole modeling domain and the yellow rectangle marks the frame which is zoomed in the lower part. The right row shows the xz plane with two different scales. Again, the yellow rectangle marks the frame which is zoomed into in the lower part. The source points of the bipole are denoted with red arrows. Lengths are given in km. Further explanations are given in the text.*

The discretization of the elongated dipole source, which is mostly just called bipole, is carried out as summation of many single dipole sources which follow the same orientation.

Illustrated in figure 3.17 (left), the horizontal discretization can be further described like two distinct point sources. It is fine around the first source point (e.g. the negative x-axis), according to the highest gradients of the EM fields. With increasing distance towards the center of the x-oriented bipole, the step sizes increase as the gradient in field change would decrease. From the center on, this scheme is just reflected taking the mid point of the bipole as symmetry axis. Applying this scheme leads to a logarithmic discretization in the horizontal plane as displayed in figure 3.17 (left). The discretization is done basically by using algorithm 3. Note, that the vertical discretization is carried out in the same way as for a ground based dipole source.

Grid for rectangular loop transmitters and comparison of discretization schemes

Illustrated in figure 3.18 is the horizontal discretization of a rectangular loop transmitter. The transmitter is discretized by summation of galvanic dipoles forming a closed circuit (cf. fig. 3.18). In general it is constructed of four adjacent bi-poles by rotation, translation and reflection. The current flows, alas not marked, from bottom left to the top right in positive and from top right to bottom left in negative direction. Note, the fine discretization around the transmitter corners which should account for the biggest EM field gradients.

In the following I made a further comparison of the two different grid discretizations by taking a closer look into the convergence during the Lanczos process for grid1 and grid2. For this, the evolution of equation (3.75) for the different grids is checked at every iteration of the Lanczos process. Additionally, the convergence criteria given by equation (3.76), is compared for the two discretizations and last not least the model response of grid1 and grid2 is compared to the analytical solution.

In order to reduce the erratic choice of the control receiver, I selected a SHOTEM setup to compute the model responses. Here, the most meaningful decision would be the central loop receiver as control receiver. Therefore no interpolation has to be carried out for any grid that would be designed with *make_ sldm_ grid* in the following.

The grid design is first based on a homogeneous half-space of $\sigma_{min} = \sigma_{max} = 1/100\ S/m$ and the time range is $t_{min} = 1\ \mu s\ t_{max} = 0.1\ ms$. This is extended further to compute transients for a bigger time range of $t_{max} = 1\ ms$ and $\sigma_{min} = 1/50\ S/m\ \sigma_{max} = 1/500\ S/m$ where the latter values are assumed for the grid discretization. The real model is kept fix. For convenience, ϵ was set to a very low value of $1/10^5$ in order to get realistically accurate results.

A variable parameter for the first comparison would be to increase the maximum allowed grid nodes from a coarse grid ($N_g = 30$) to a very fine grid ($N_g = 90$), which implies the grid is always symmetric.

The following calculations of transient responses are carried out for 2 grids with increasing N_g of 10 per simulation. Figure 3.18 shows xy-planes of the coarse grid ($N_g = 30$) and figure 3.19 shows an intermediate grid ($N_g = 60$) to get an impression of the model dimensions, the grids and the SHOTEM setup. The red rectangle shows the transmitter, whereas the blue rectangle corresponds to the area in which center the $\dot{\mathsf{H}}_\mathsf{z}$ response is calculated.

A comparison between fig. 3.18 and fig. 3.19 reveals, that applying algorithm 2 with the already mentioned problematic discretization length, lead to over discretized grids in the

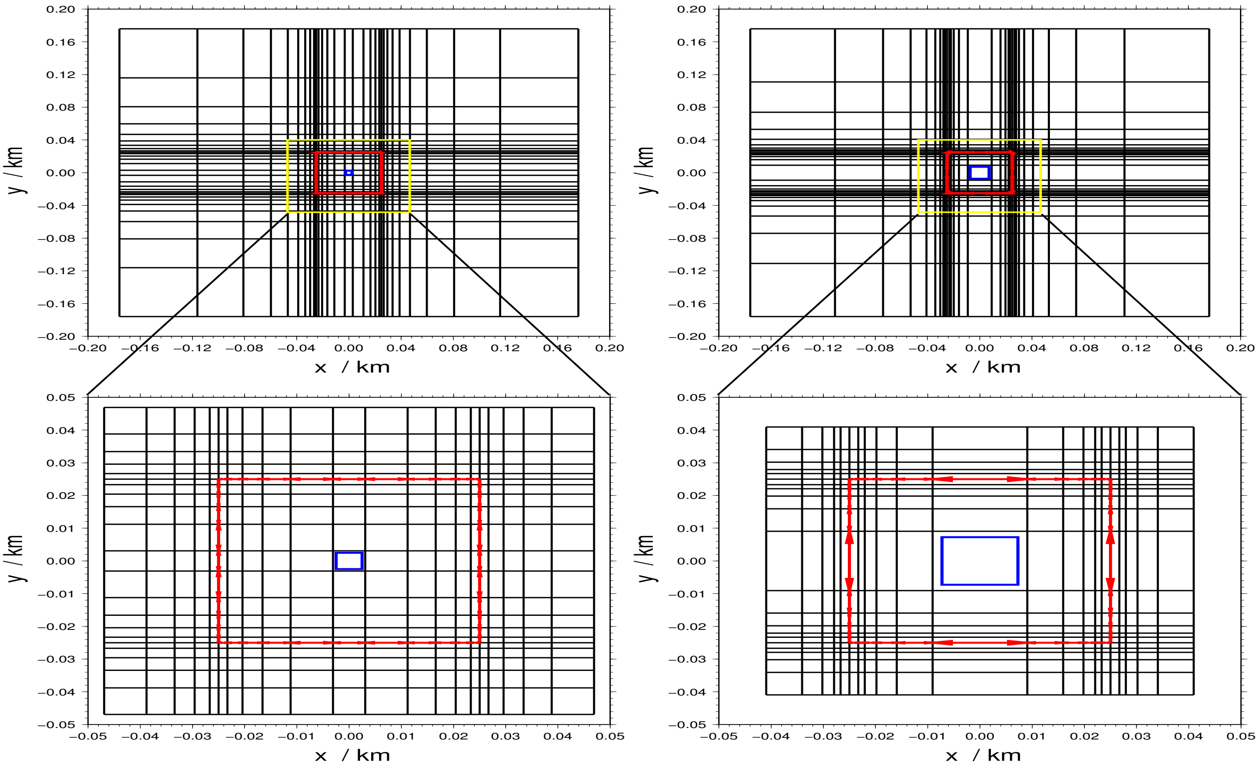


Figure 3.18: *Comparison of grids of a rectangular loop source with coarse spacings (N_g = 30). This type of transmitter discretization is commonly used to model SHOTEM surveys. Note the different source discretizations compared to the LOTEM bipole discretization (cf. Fig. 3.17). The current flow of the transmitter (red arrows) is approximated with line currents forming a rectangular loop. The left row shows a xy-plane discretized with grid1 whereas grid2 is shown in the right row. The blue rectangle indicates the receiver coil measuring $\dot{\mathsf{H}}_\mathsf{z}$-voltages. Further explanations are given in the text.*

vicinity of the source.

For each grid I computed transient responses and compared it with the analytical solution (ana) (cf. Fig. 3.20). Further I compare the evolution of the Lanczos process evaluating equation (3.75) at every iteration ($it1 = 1$ and $it2 = 1$) and give an estimation of the subspace dimension as given by equation (3.74). Additionally in every grid label a number is shown, telling how many iterations where made after reaching the convergence estimate, given by equation (3.76).

The data displayed in figure 3.20 (left) is showing that for the coarse discretization (N_g = 30) both grids achieve acceptable results compared to the analytical solution. The overall difference is slightly below 5% at early times and around 1% for late times. It seems that, except for late times, grid2 achieves a slightly better result. A look at the evolution of the latest time point (cf. fig. 3.20, right) reveals, that grid1 performs much better than grid2. The convergence process seems more stable, because it reaches the desired voltage in far less iterations, though the convergence criteria is achieved in comparable iteration numbers.

The situation dramatically changes if N_g is increased. For N_g = 60, which results are displayed in figure 3.21, the overall error for grid2 is leaving the acceptable level after the first decade. Even grid1 performs not with the desired accuracy. A closer look to the evolution of the last time point (cf. fig. 3.21 right) reveals, that grid1 does not converge

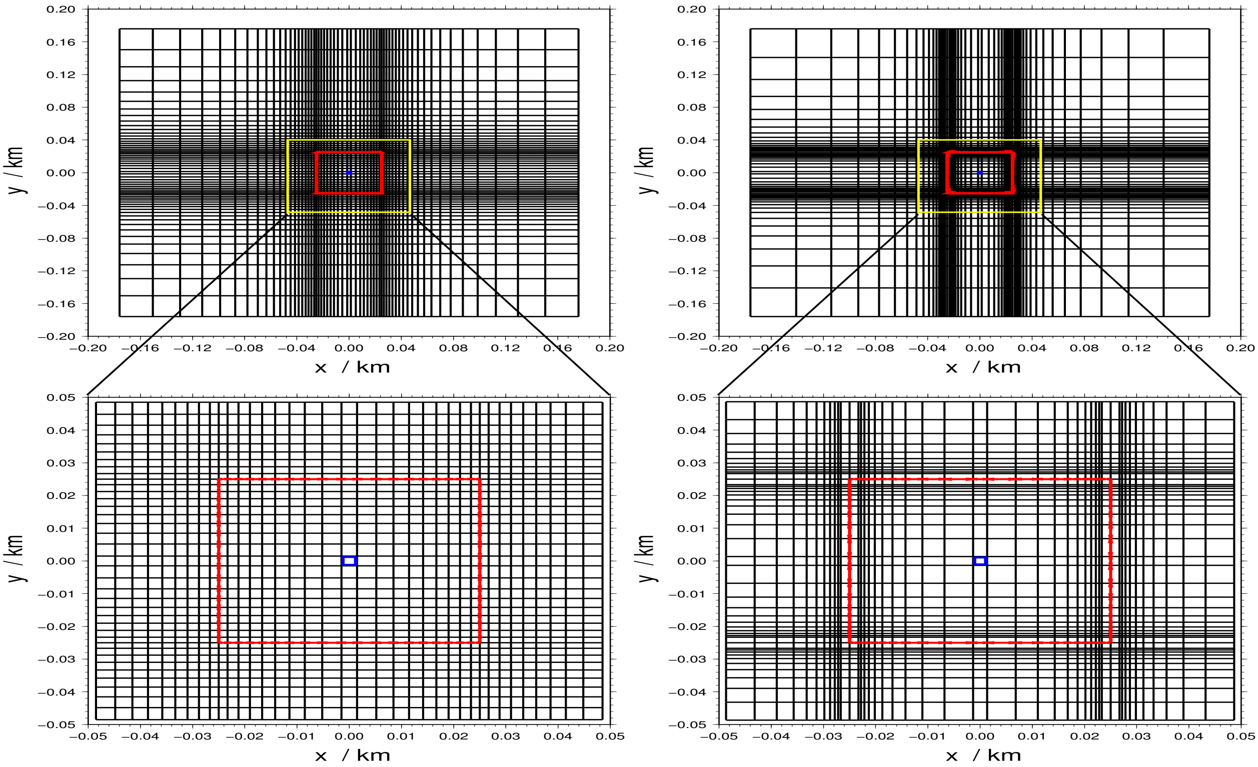


Figure 3.19: *xy-plane of grid1 (left row) and grid2 (right row) with fine grid ($N_g = 90$). Further explanations are given in the text.*

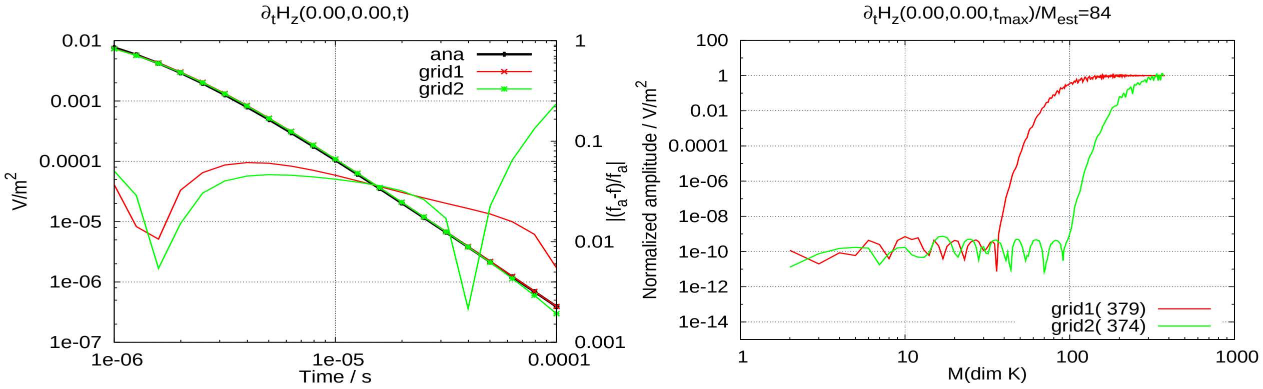


Figure 3.20: *Comparison of the transient responses of two different grid discretizations (grid1 and grid2) to the analytical solution (ana) over time (left). For convenience, $|(f^a - f)/f|$ shows the relative difference to the analytical solution. The receiver type and its relative position to the center of the transmitter loop is given in the title of the plots (here in-loop configuration). The evolution of the normalized last time point, given by equation (3.75), is displayed on the right. Here M_{est} denotes the estimation of the Krylov subspace calculated by equation (3.74). The number in brackets behind grid1 and grid2 shows the number of iterations after reaching the convergence criteria (3.76). The total number of grid nodes is limited to $N_g = 30$ in this comparison. Further explanations are given in the text.*

properly and grid2 not at all. Looking at the data for $N_g = 90$ (cf. fig. 3.22) the situation of grid2 becomes even worse. Here, the transient o grid2 is showing some numerical noise which is nicely explained through the evolution of the last time point (cf. fig. 3.22 right). For grid2 it shows, that the convergence criteria is stopping the Lanczos process before the solution was acceptable. To account for this, one may increase *it*2 to some erratic value, or

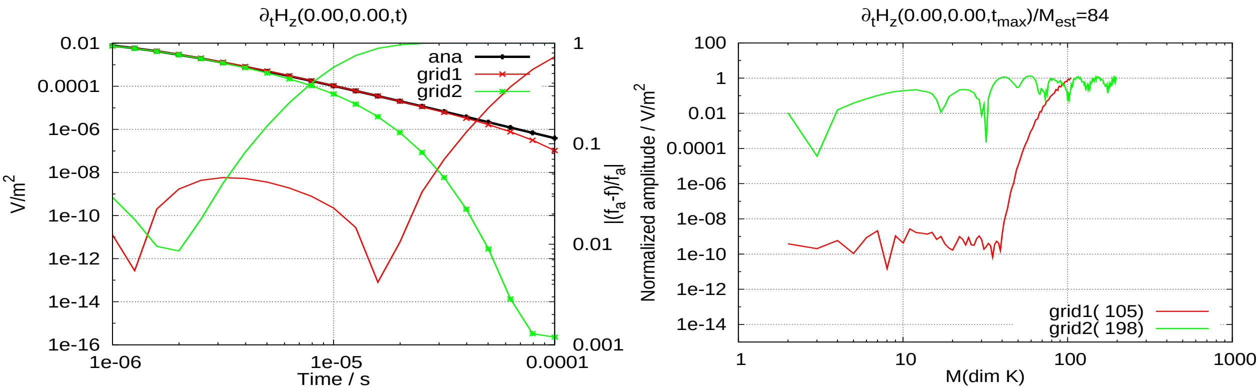


Figure 3.21: *Comparison of the transient response (left) and evolution of equation (3.75) (right) for $N_g = 60$. $|(f^a - f)/f|$ gives the relative difference to the analytical solution. For further explanations refer to figure 3.20 and the text.*

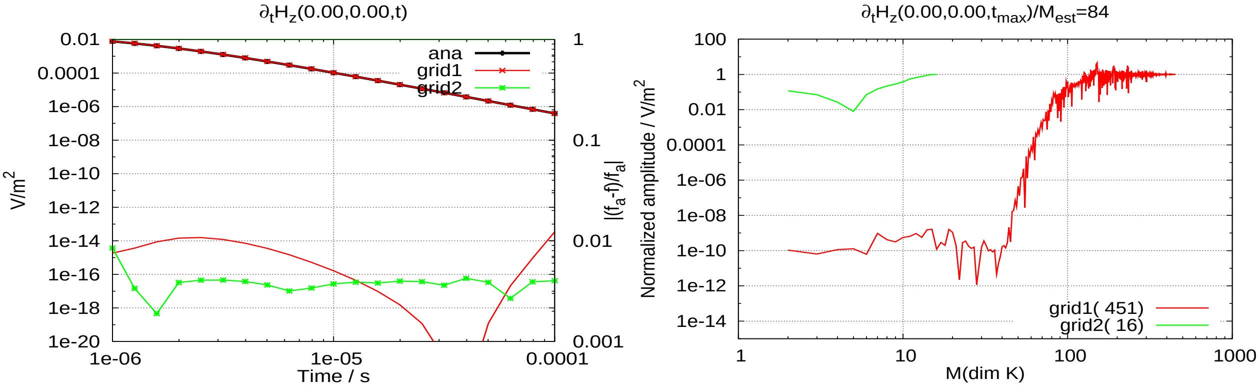


Figure 3.22: *Comparison of the transient response (left) and evolution of equation (3.75) (right) for $N_g = 90$. $|(f^a - f)/f|$ gives the relative difference to the analytical solution. For further explanations refer to figure 3.20 and the text.*

check the ϵ criteria twice. As another fixed parameter, I set that the convergence can only stop after at least M_{est} are carried out. This leads to a simple adjustment to the Lanczos convergence criteria which reads

$$\hat{\epsilon}_i \leq \hat{\epsilon}_{i+1} \leq \epsilon \quad \mathsf{M}_{est} < i = it1 + k * it2 \tag{3.80}$$

The estimated value after (3.74) should be $\mathsf{M}_{est} = 84$ in this example. In case, that convergence will never be reached, I set a fixed upper limit $\mathsf{M}_{max} = \mathsf{M}_{est} * 20 = 1680$ for the first comparisons.

The influence on the convergence process due to the adjustment is quite remarkable and shown in fig. 3.23 and fig. 3.24. Note, the transient responses now do reach an overall acceptable result compared to the analytical solution. This is the case, except for the late time responses of the finest grid (cf. fig. 3.24 left) where the relative difference of grid2 is increased over 10%. Further note, that the finest discretization for grid1 seems more accurate, because of its low difference to the analytical solution. Yet, the evolution of (cf. fig. 3.24 right) showing strong oscillations compared to the coarser grid (cf. fig. 3.23 right) which is also a rough indicator for a bad conditioned system matrix **A**.

As consequence of the stronger convergence criteria, the overall required number of iterations to satisfy equation 3.80, is dramatically increased. This is observed now for every simulation.

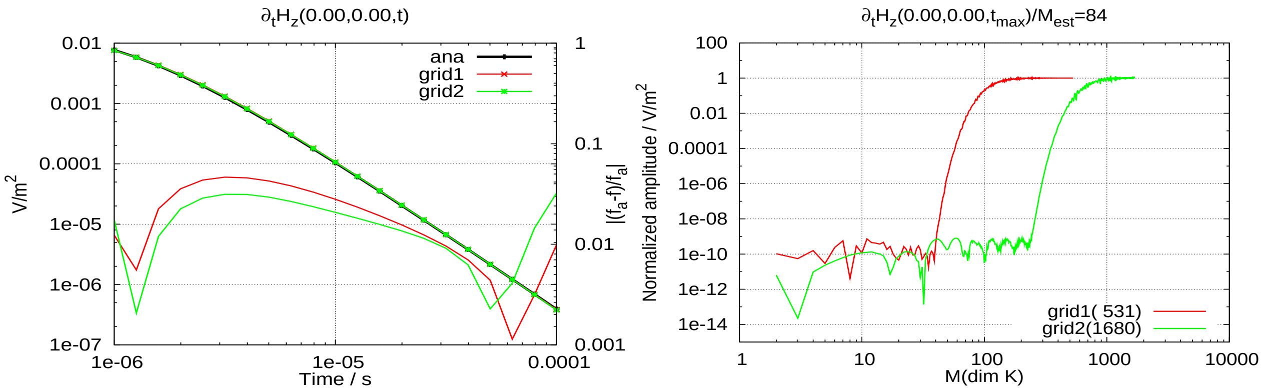


Figure 3.23: Comparison of the transient response (left) and evolution of equation (3.75) (right) for $N_g = 60$ and the adjusted convergence criteria. $|(f^a - f)/f|$ gives the relative difference to the analytical solution. For further explanations refer to figure 3.20 and the text.

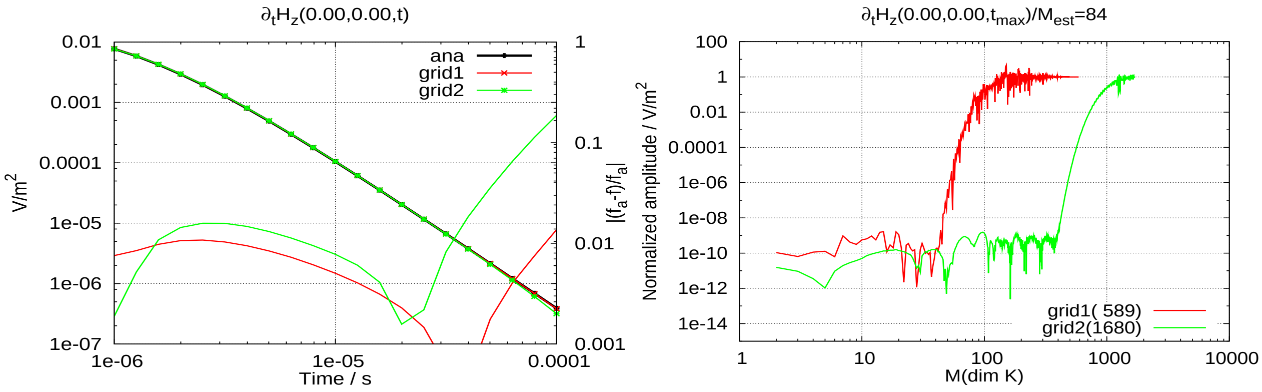


Figure 3.24: Comparison of the transient response (left) and evolution of equation (3.75) (right) for $N_g = 90$ and adjusted convergence criteria. $|(f^a - f)/f|$ gives the relative difference to the analytical solution. For further explanations refer to figure 3.20 and the text.

This also means, that the convergence of the Lanczos process needs far more iterations than suggested by equation (3.74).

The question now is, whether there exists any chance of getting another equation or a rough estimate to which extend the Lanczos process should be carried out, in order to get stable results, and which number of predefined grid nodes give the best results for predefined model borders.

3.4.2 Extended grid analysis

To introduce the grid study that was carried out, just a few remarks about the used values for the performance analysis are given. First I compare the two described grid algorithms by checking how they particularly perform in terms of iteration numbers of the Lanczos process and the accuracy of the solution. Therefore M shows the total number of iterations that where used within the Lanczos process to reach convergence. For the comparison to the analytical solution I preferred to take the ℓ^1-norm measure of the relative differences. It is more robust to outliers than the ℓ^2-norm if there is no variance available for the data points. For convenience, I normalized the ℓ^1-norm measure to the number of data points and display them as

$$\|\mathbf{r}_e\|_1 = \frac{100}{N_t} \sum_{i=1}^{N_t} \left| \frac{f_i^a - f_i}{f_i^a} \right| , \tag{3.81}$$

given in %, where f^a is the reference solution of the homogeneous half-space. M_{max} is kept fixed at $20\mathsf{M}_{est}$ for every grid test. ϵ was again set to $1/10^5$.

As a result of the first comparisons, the modified log spaced discretization following algorithm 3, turned out to perform far better than the simple log spaced grid. Hence, the performance of this grid scheme is tested for different model setups. At this, the grid with varying leading values of σ_{min} and σ_{max} are compared for symmetric grids with varying N_g. To conclude the extended grid analysis, the dependence of the performance for varying vertical grid points N_z with different horizontal grid points $N_x = N_y = N_g$ is evaluated.

Performance of two grid algorithms

The leading value for the next comparison is t_{max}. Because M_{est} changes for that, the maximum value of allowed iterations is increased. The other leading values, as $\sigma_{min} = 1/100\ S/m$, $\sigma_{max} = \sigma_{min}$ and $t_{min} = 1\ \mu s$ are kept fixed during the computations. N_g is the total number of grid values used for discretization of the symmetric grids. Since it is likely to know how many grid values are needed to get good results, N_g is varied throughout the calculations. The number of grid values N_g changes from $N_g = 30$ to $N_g = 90$ with increment $\Delta N_g = 10$.

To compare the results for response times of 2 and 3 decades, they are compiled in a overview within tabular 3.1.

Comparing the two response times, the maximum number of allowed iterations for grid2 is reached for $N_g \geq 50$ (cf. Tabular 3.1). Grid1 does never exceed the maximum value for any N_g. The overall error norm for grid1 is likely to achieve the best fit to the analytical solution for $N_g = 70$ (2 decades) and $N_g = 90$ (3 decades). Taking the number of iterations M into account these values look appropriate. An interesting feature appears to be, that increasing the number of grid nodes does not increase numerical accuracy in any case. This feature of the *SLDMem3t* is also observed by other authors (e.g. Hördt et al. [1992]; Weidelt [2000]). For grid2 the values of $N_g = 70$ and $N_g = 50$ seems to have the best agreement to the analytical solution.

For grid design one normally assumes to cover a bigger range of conductivities. In the last test, they were fixed to the conductivity of the underlying half-space. In order to get more realistic values for the grid layout, the leading values of $\sigma_{min} = 1/50\ S/m$ and $\sigma_{max} = 1/500\ S/m$ where changed. Yet, the conductivity of the real model stays at $100\,\Omega m$.

The compilation of the second grid test simulations are given in Table 3.2. Comparing the error values $\|\mathbf{r}_e\|_1$ of the two grids, the results appear slightly different to the first test. For grid1 $N_g = 70$ for 2 decades seems to achieve the best fit to the analytical solution, as well as for 3 decades the value of $N_g = 80$. Similar results for N_g of grid1 can be observed in the first test (cf. Tabular 3.1). Remarkable is, that grid2 with $N_g = 80$ and $\|\mathbf{r}_e\|_1 = 0.35\,\%$ performs slightly better than the best fit of grid1 ($\|\mathbf{r}_e\|_1 = 0.52\,\%$) for 2 decades of response time. Yet, comparing the computational effort, M is five times less for grid1, which seems more appropriate.

Table 3.1: *Comparison of required iterations* (M) *and agreement to the analytical solution* ($\|\mathbf{r}_e\|_1$) *for varying time ranges and the corresponding changes in grid layout of grid1 and grid2.*

$t_{min} = 1\,\mu s$ $t_{max} = 0.1\,ms$ $\mathsf{M}_{est} = 84$ $\mathsf{M}_{max} = 1680$				$t_{min} = 1\,\mu s$ $t_{max} = 1\,ms$ $\mathsf{M}_{est} = 268$ $\mathsf{M}_{max} = 5360$	
N_g	Grid	M	$\|\mathbf{r}_e\|_1$ [%]	M	$\|\mathbf{r}_e\|_1$ [%]
30	grid1	506	3.36564	1535	2.204
	grid2	1180	3.44535	2925	7.04593
40	grid1	589	0.49065	1575	3.313
	grid2	1644	2.95603	4508	1.79032
50	grid1	546	1.54069	1819	0.645663
	grid2	1680	0.598745	5360	0.989691
60	grid1	531	2.18707	1690	1.12864
	grid2	1680	1.77814	5360	1.19026
70	grid1	825	0.34461	1726	1.48309
	grid2	1680	0.310091	5360	1.28805
80	grid1	656	0.503617	1700	1.68831
	grid2	1680	0.774629	5360	2.72417
90	grid1	589	0.631818	2557	0.287265
	grid2	1680	2.66789	5360	1.20748

The overall performance of grid1 for 3 decades response times, is clearly better than grid2 in terms of accuracy and iteration numbers. Therefore, grid2 is discarded in the following and further grid analysis are done with grid1. However, it is observed, that the solution for response times of 3 decades perform little less accurate for every grid.

Performance of symmetric grids

In order to check how the grid performs for different model boundaries, the leading values σ_{min} and σ_{max} are now varied. The values for the grid performance, M and $\|\mathbf{r}_e\|_1$, are then compared to each other. Accounting for a general exponential behavior of the model dependencies, the conductivity interval length increases logarithmically around the initial value of $\sigma_{min} = 1/100\,S/m$ from zero to four decades. Thus, σ_{min} takes values of $\{1/100, 1/500, 1/1000, 1/5000, 1/10000\}$ and σ_{max} is varied like $\{1/100, 1/50, 1/10, 1/5, 1\}$, where $\sigma_{min} = 1/1000\,S/m$ is used twice for $\sigma_{max} = \{1/10, 1\}\ S/m$, giving 6 models to compare. The model intervals are chosen to cover up the expected resistivity values for most earth materials which are studied within this work.

Because d_{min} and d_{max} changes for this conductivity values, M_{est} is increased the like as well as the maximum value of allowed iterations. The other leading values, as $t_{min} = 1\,\mu s$ and $t_{max} = 0.1\,ms$ are kept fixed during the computations. Again, N_g is the total number of grid values used for a symmetric grid. They are varied from $N_g = 30$ to $N_g = 90$ with increment $\Delta N_g = 5$ to give finer discretization steps.

To compare the results they are compiled within Table 3.3 for 3 different model boundaries.

For the homogeneous model boundaries ($\sigma_{min} = \sigma_{max} = 1/100\,S/m$), the best fit is reached

Table 3.2: *Comparison of required iterations (M) and agreement to the analytical solution ($\|\mathbf{r}_e\|_1$) for varying time ranges ($[t_{min}, t_{max}]$) and the corresponding changes in grid layout of grid1 and grid2. Conductivity values for the grid layout are set to a intermediate conductivity range of $[\sigma_{min} = 1/500\,S/m, \sigma_{max} = 1/50\,S/m]$.*

		$t_{min} = 1\,\mu s$ $t_{max} = 0.1\,ms$ $\mathsf{M}_{est} = 268$ $\mathsf{M}_{max} = 5360$		$t_{min} = 1\,\mu s$ $t_{max} = 1\,ms$ $\mathsf{M}_{est} = 848$ $\mathsf{M}_{max} = 16960$	
N_g	Grid	M	$\|\mathbf{r}_e\|_1$ [%]	M	$\|\mathbf{r}_e\|_1$ [%]
30	grid1	570	6.06088	3262	5.23899
	grid2	2219	3.75323	5539	9.54688
40	grid1	1042	3.14964	3740	1.78826
	grid2	3069	5.04615	7731	2.81064
50	grid1	1398	0.924415	3720	2.93895
	grid2	3952	1.26031	7687	1.51482
60	grid1	894	2.00047	3825	1.06952
	grid2	3354	1.19622	12515	1.58766
70	grid1	1397	0.515619	4145	1.57351
	grid2	5331	1.31097	14559	1.73784
80	grid1	1456	0.980274	4474	0.398925
	grid2	5360	0.352253	11662	1.30429
90	grid1	997	1.33022	2070	0.667414
	grid2	5360	0.818171	16000	0.491219

for $N_g = 70$ nodes, which agrees well to the previous study of grid comparison. According to Tabular 3.3, the grid with model boundaries of $(\sigma_{min} = 1/1000, \sigma_{max} = 1/10)\ S/m$ reaches a comparable fit for the same number of grid nodes, but with $N_g = 90$ it reaches the smallest value of ≈ 0.25 % for the compared model layout grids. This result slightly changes for the results of the grid layout with $\sigma_{max} = 1\ S/m$, where the minimum fit of ≈ 0.5 % is reached for $N_g = 75$.

From Table 3.3 it can be observed, that the number of iterations to reach convergence is increasing dramatically from the homogeneous to the more extreme model values for the grid layout. For the intermediate grid layout with $(\sigma_{min} = 1/1000, \sigma_{max} = 1/10)\ S/m$ we find values of $1500 < \mathsf{M} < 4000$ with tendency to increase as N_g increases. For the grid layout with $(\sigma_{min} = 1/1000, \sigma_{max} = 1)\ S/m$, this seems more erratic and in general one can not observe, that M increases as the fit does.

Taking more model values for the symmetric grid layout into account, the situation changes as expected. From the values given in Tabular 3.4 one can see, that the number of iterations and the misfit increases the most for the grid layout of highest conductivity contrast of $\sigma_{min} = 1/10000\ S/m$ and $\sigma_{max} = 1\ S/m$ for any number of grid nodes. The best results of the second comparisons are given for the smaller conductivity variation of $\sigma_{min} = 1/500\,S/m$ and $\sigma_{max} = 1/50\ S/m$ where $\|\mathbf{r}_e\|_1$ attains the minimum value of ≈ 0.24 % for $N_g = 65$. Furthermore, this value seems the best result at all, compared to the best fit of the first results (cf. Tabular 3.3).

Comparing the number of iterations of Tables 3.3 and 3.4, it can be seen that the grid layout of highest conductivity contrast uses the most iterations in the Lanczos process. This result

Table 3.3: *First comparison of required iterations (*M*) and agreement to the analytical solution* $\|\mathbf{r}_e\|_1$ *for varying conductivity ranges (*$[\sigma_{max}, \sigma_{min}]$*) and the corresponding changes for symmetric grid layouts.*

$\sigma_{min} = 1/100\ S/m$ $\sigma_{max} = 1/100\ S/m$ $\mathsf{M}_{est} = 84$			$\sigma_{min} = 1/1000\ S/m$ $\sigma_{max} = 1/10\ S/m$ $\mathsf{M}_{est} = 848$		$\sigma_{min} = 1/1000\ S/m$ $\sigma_{max} = 1\ S/m$ $\mathsf{M}_{est} = 2683$	
N_g	M	$\|\mathbf{r}_e\|_1$ [%]	M	$\|\mathbf{r}_e\|_1$ [%]	M	$\|\mathbf{r}_e\|_1$ [%]
30	508	3.36564	1573	6.40728	5137	7.75109
35	409	1.30482	1968	8.56122	4428	8.3253
40	589	0.49065	2680	4.25773	5603	5.59941
45	655	0.67628	3008	1.52145	4610	3.93575
50	546	1.54069	3995	1.1948	4983	2.70538
55	590	1.50574	2650	1.7665	4953	5.40666
60	531	2.18707	3472	0.530166	5240	4.17776
65	701	1.84131	3123	1.62218	6526	3.22332
70	825	0.34461	2950	0.471722	4583	2.40105
75	785	0.41741	3323	1.28283	4765	0.554561
80	656	0.50617	2990	0.358958	6960	0.584448
85	614	0.66131	2736	0.899561	5288	1.23514
90	589	0.631818	3733	0.249885	5871	1.91515

Table 3.4: *Second comparison of required iterations (*M*) and agreement to the analytical solution (*$\|\mathbf{r}_e\|_1$*) for different conductivity ranges (*$[\sigma_{min}, \sigma_{max}]$*) and the corresponding changes for symmetric grid layouts.*

$\sigma_{min} = 1/10000\ S/m$ $\sigma_{max} = 1\ S/m$ $\mathsf{M}_{est} = 8485$			$\sigma_{min} = 1/500\ S/m$ $\sigma_{max} = 1/50\ S/m$ $\mathsf{M}_{est} = 268$		$\sigma_{min} = 1/5000\ S/m$ $\sigma_{max} = 1/5\ S/m$ $\mathsf{M}_{est} = 2683$	
N_g	M	$\|\mathbf{r}_e\|_1$ [%]	M	$\|\mathbf{r}_e\|_1$ [%]	M	$\|\mathbf{r}_e\|_1$ [%]
30	8494	16.4397	1177	6.08711	3144	8.12846
35	8539	6.85483	1157	1.30758	2758	10.6108
40	8493	3.68227	1102	3.15057	2879	5.7164
45	8500	7.01053	1094	3.6767	3163	2.34616
50	8499	7.00863	1410	0.924348	3750	1.65686
55	8555	2.29251	1342	1.50606	3890	3.184
60	8487	1.46872	1255	1.99758	3088	1.26599
65	8649	2.94979	1685	0.241104	3495	2.33906
70	8651	3.35496	1504	0.515265	3302	1.65242
75	8490	0.886627	1268	0.715311	7872	0.503316
80	10004	0.831135	1375	0.980156	2706	1.70779
85	8696	2.40805	1305	1.13257	3823	0.339606
90	8794	1.49552	1344	1.3297	3657	1.15512

seems very reasonable, because it is indicating the most ill posed system matrix $\mathbf{A}$ due to the large values of d_{min} and d_{max}. For the small and intermediate conductivity variations of $\sigma_{min} = \{1/100, 1/500, 1/1000\}\ S/m$ and $\sigma_{max} = \{1/100, 1/50, 1/10\}\ S/m$ the dimension of the Krylov subspace is substantially smaller. Overall it can be observed, that the iterations needed in the Lanczos process decrease as the conductivity contrast decreases.

Since, the conductivity interval length is more or less taken logarithmically symmetric to the underlying model, the systematical increase of the iteration numbers may be expressed in a formula. This would give a rough factor of 2.5 increase in number of required iterations per decade in model variation for the grid layout.

Grid dependence of horizontal and vertical grid nodes

Up to now, the performance analysis was given for symmetric grids where the number of grid nodes of all directions N_x, N_y and N_z was varied equally. The question now arises, if the performance of the grid can be enhanced if one introduces a finer grid, e.g. in vertical direction. Therefore, I summarize a further study of the grid performance by varying both, the vertical N_z and the horizontal discretization. Because of symmetry, the horizontal directions are chosen equally in the following. Since, the tabular overview would be far less comparative, I selected to compile the results for M and $\|\mathbf{r}_e\|_1$ as a function of N_z and N_g within 2D graphs.

Each graph corresponds to a grid layout for different $\sigma_{min,max}$ values which where chosen exactly like the previous study. Thus, σ_{min} takes values of $\{1/100, 1/500, 1/1000, 1/5000, 1/10000\}$ and σ_{max} is varied like $\{1/100, 1/50, 1/10, 1/5, 1\}$, where $\sigma_{min} = 1/1000\, S/m$ is used twice for $\sigma_{max} = \{1/10, 1\}\ S/m$.

In the following graphs, the number of iterations is shown to the left whereas the misfit is displayed to the right. The grid node numbers N_z and $N_g = N_{x,y}$ are varied between 30 and 90 with increment of $\Delta N = 5$. High values of the misfit and M are indicated by orange and yellow, whereas a good fit (mostly below 1 %) and low M values are denoted by dark red and blue colors. Note, that M is illustrated linear but the misfit is not. I selected to display $\|\mathbf{r}_e\|_1$ as log_{10} values to enhance differentiation of the single values.

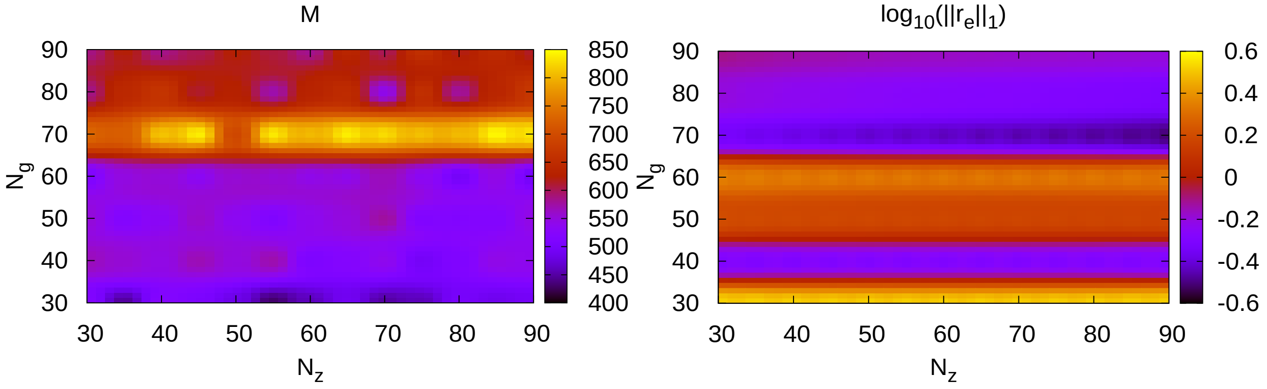


Figure 3.25: *Comparison of number of iterations* M *(left) and the misfit* $log_{10}\left(\|\mathbf{r}_e\|_1\right)$ *(right) as function of* N_z *and* N_g*. High values of the misfit and* M *are indicated by orange and yellow whereas the fit below* $1\,\%$ *and low* M *values are denoted by dark red and blue colors. The leading values for the grid design are* $\sigma_{min,max} = 1/100\, S/m$*. Further explanations are given in the text.*

Figure 3.25 shows the number of iterations and the misfit as function of N_z and N_g for the first grid layout with $\sigma_{min,max} = 100\, S/m$. As perceived from figure 3.25 (right), there seems to be no N_z dependence of the misfit. Further, the dependence seems governed by the horizontal discretization, which can be recognized by the horizontal stripes in the graph. The yellow horizontal stripes for $N_g = 30$ and $N_g = 50-60$ seem to be a systematic pattern of the grid leading to higher errors in the transient responses. The highest fit is reached for

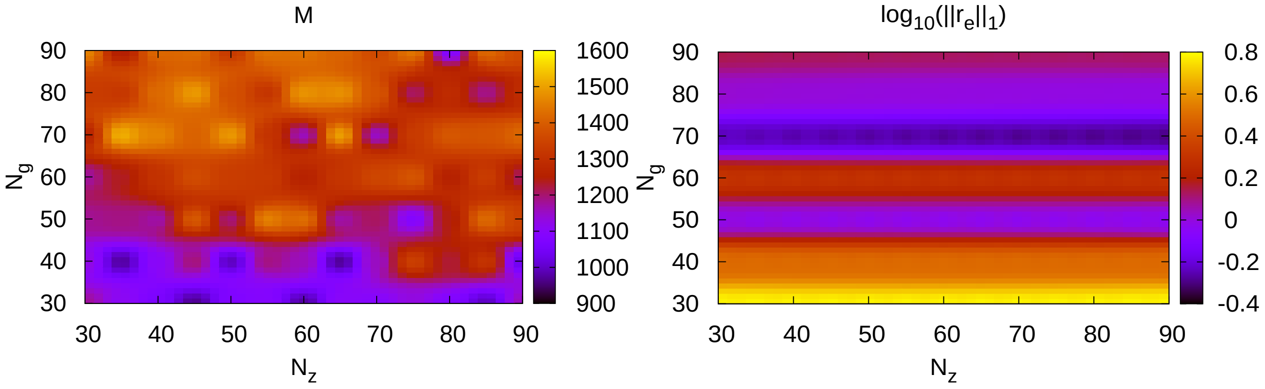


Figure 3.26: *Number of iterations* M *(left) and the misfit* $log_{10}\left(\|\mathbf{r}_e\|_1\right)$ *(right) as function of* N_z *and* N_g *for* $\sigma_{min,max} = (1/500, 1/50)$ S/m*. Further explanations are given in figure 3.25 and the text.*

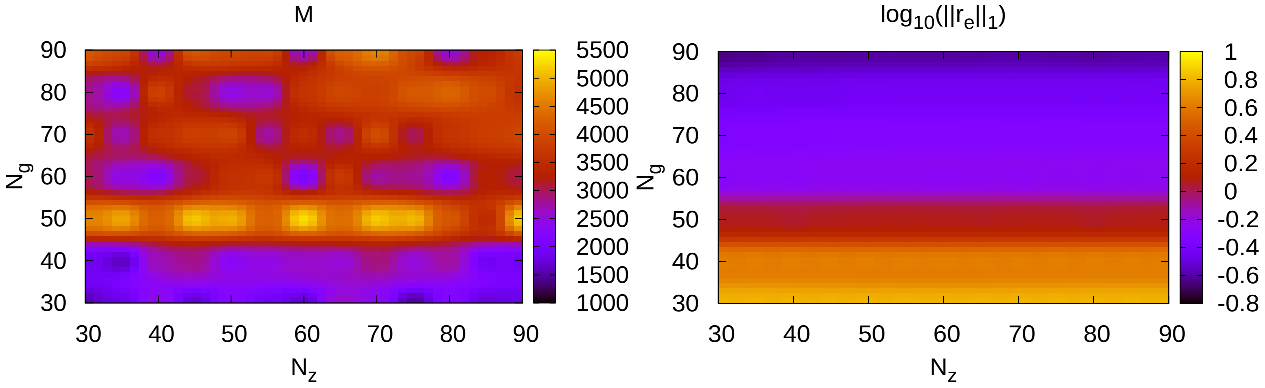


Figure 3.27: *Number of iterations* M *(left) and the misfit* $log_{10}\left(\|\mathbf{r}_e\|_1\right)$ *(right) as function of* N_z *and* N_g *with* $\sigma_{min,max} = (1/1000, 1/10)$ S/m*. Further explanations are given in figure 3.25 and the text.*

$N_g = 70,\ N_z = 90$ ($\|\mathbf{r}_e\|_1 \approx 0.25\,\%$) which is in excellent agreement to the further study of symmetric grids. However, as indicated from figure 3.25 (left), M attains the highest values for this grid combination.

The performance comparison for the grid layout with the leading values of $\sigma_{min,max} = (1/500, 1/50)$ S/m is displayed in figure 3.26. Comparing the evolution of M (cf. fig. 3.26 (left)) reveals that it covers a broader range as displayed in figure 3.25 (left). M attains the highest values for grid combinations with $N_g \leq 70$. Note, the appearance of some blue holes in the yellow stripes which can be connected to grids which are more stable. The development of the misfit in figure 3.26 (right) shows a similar pattern to figure 3.25 (right): A N_g dependence of the misfit seems quite obvious whereas the N_z dependence can not be recognized. The highest fit is reached for $N_g = 70$ ($\|\mathbf{r}_e\|_1 \approx 0.3\,\%$) which seems a little bit higher as for the study of symmetric grids.

Illustrated in figure 3.27 is the performance comparison for the grid layout with $\sigma_{min,max} = (1/1000, 1/10)$ S/m. The evolution of M shows a pronounced maximum for $N_g = 50$ (cf. fig. 3.27 (left)). Comparing the misfit in figure 3.27 (right), a flatter pattern for all grid node values is conspicuous. It reveals a smoother misfit performance for the grid values from lower to higher numbers. The highest fit is reached for $N_g = 90$ ($\|\mathbf{r}_e\|_1 \leq 0.2\,\%$) which seems a little less than the values which where observed in the study of symmetric grids.

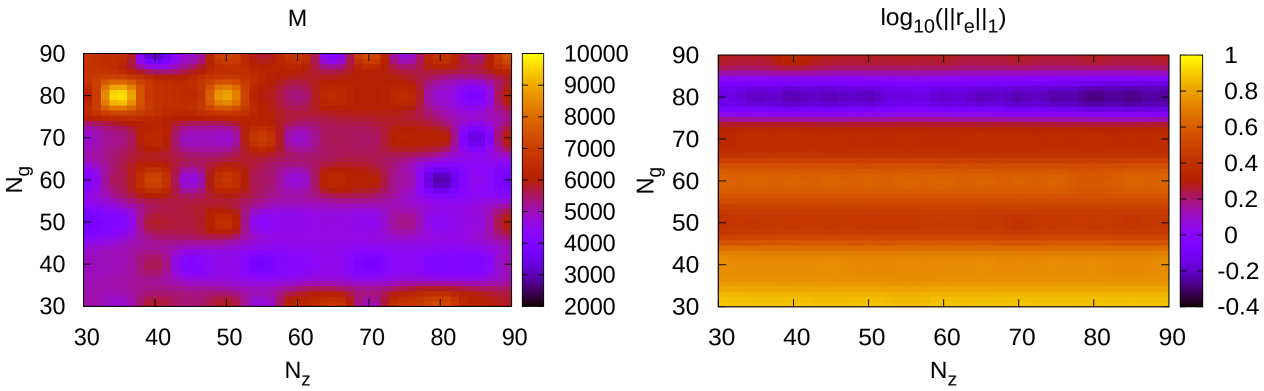


Figure 3.28: *Number of iterations* M *(left) and the misfit* $log_{10}(\|\mathbf{r}_e\|_1)$ *(right) as function of* N_z *and* N_g *with* $\sigma_{min,max} = (1/1000, 1)$ S/m. *Further explanations are given in figure 3.25 and the text.*

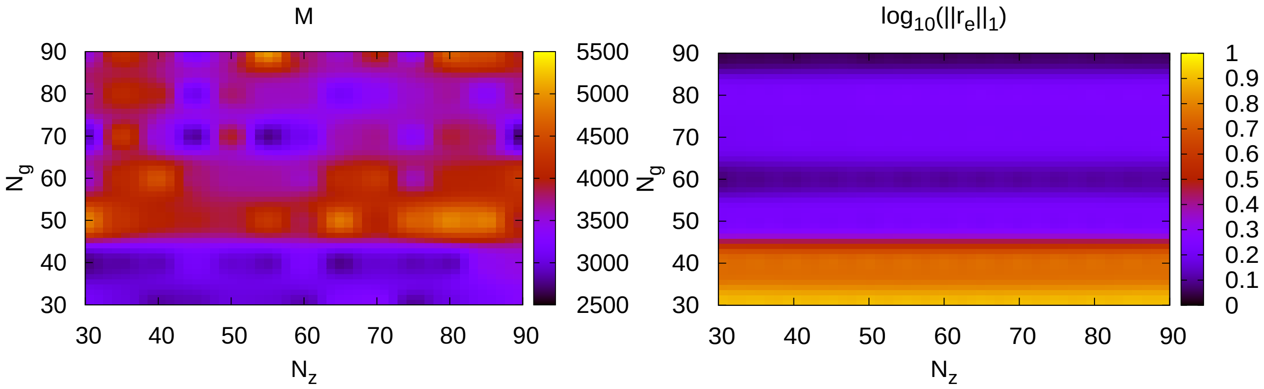


Figure 3.29: *Number of iterations* M *(left) and the misfit* $log_{10}(\|\mathbf{r}_e\|_1)$ *(right) as function of* N_z *and* N_g *with* $\sigma_{min,max} = (1/5000, 1/5)$ S/m. *Further explanations are given in figure 3.25 and the text.*

The comparison of the performance for the grid with $\sigma_{min,max} = (1/1000, 1)$ S/m is displayed in figure 3.28. Herein, the values for M (cf. fig. 3.28 (left)) are showing some relatively high values for the grids with high fit. A fit below 1 % is attained just for N_g around 80 which is a protruding feature of the misfit graph (cf. fig. 3.28 (right)). Yet, this is in quite good agreement to the previous study for this grid layout.

Figure 3.29 shows the performance for the grid with $\sigma_{min,max} = (1/5000, 1/5)$ S/m . A comparison of the values of M with the previous grid layout (cf. fig. 3.28 (left)) reveals, that the values given in figure 3.29 (left) are substantially smaller. The misfit of the grids with the logarithmic equally spaced interval around $\sigma = 1/100$ S/m do attain a fit below 1 % for $N_g = 60$ and $N_g = 90$.

Last not least, the performance for the grid with $\sigma_{min,max} = (1/10000, 1)$ S/m can be found in figure 3.30. As expected from the previous study for symmetric grids, this values produce grids with the worst performance. Compared to the other grid layouts, the misfit drops below the 1 % mark for grids with $75 \leq N_g \geq 85$, but the overall M reaches very high values. The high computational cost would render the grid layouts designed with the interval $\sigma_{min,max} = (1/10000, 1)$ S/m unfeasible for the scope of inverse solution. Nevertheless, it may be used for simple forward modeling.

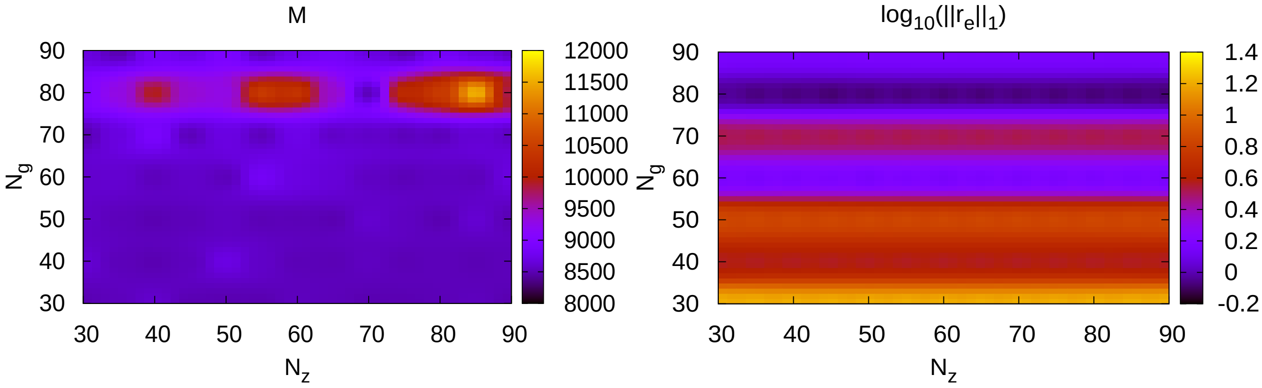


Figure 3.30: *Number of iterations* M *(left) and the misfit* $log_{10}\left(\|\mathbf{r}_e\|_1\right)$ *(right) as function of* N_z *and* N_g *for* $\sigma_{min,max} = (1/10000, 1)$ *S/m. Further explanations are given in figure 3.25 and the text.*

3.4.3 Summary/discussion

Summarizing the results of the grid study it turned out, that a grid with $70 \leq N_g \leq 80$ grid points leads to satisfying agreements to the analytical solution. Since there is no obvious dependence if one chooses the vertical discretization independent from the horizontal, I suppose a symmetric grid with $N_x = N_y = N_z$ would go well. To save computational load, three decades of conductivity variations for the grid layout would be recommended.

For the conductivity interval of $\sigma_{min,max} = (1/1000, 1)$ S/m, where the center of the log-spaced interval is slightly deviated in relation to the 100 Ωm half-space, the overall performance seemed slightly inferior to the grid layouts with the interval of $\sigma_{min,max} = (1/5000, 1/5)$ S/m which is, symmetrically log-spaced to 100 Ωm. Hence, the conductivity interval should be chosen log-spaced symmetrically centered around the mean, or expected value of the model. The model intervals are chosen to cover up the expected resistivity values for most earth materials which are studied within this work.

During the numerous tests and applications of *SLDMem3t* for realistic 3D models, an increase of M can generally be observed. Therefore, the estimates for M, as result of the previous study can be seen as lower boundary and the value of M may be increased by an order of magnitude. Of course, if one investigates or expects other resistivity values, e.g. sulfides, ores or one carries out model studies for TEM within a marine environment, one has to adjust the intervals and the expected M_{max} appropriate. A special case would be to take topographic effects into account, because of the high resistive air. For higher resistivity contrasts it is generally observed, that the performance of a grid is less effective.

The numerical examples shown here exemplify that the choice of a finer grid discretization does not in any case lead to better convergence nor quality of the solution but the number of grid nodes should be choosed appropriate to the time window and model of investigation.

3.5 Sensitivity calculation and adjoint formulation for TEM

Within every optimization process the sensitivity or Jacobian matrix is, besides the accurate forward solution, considered as the key issue to solve the problem. A single sensitivity value denotes the change of the forward response as a function of the model $f_i(\mathbf{m})$ with respect to a finite or infinite small change of the model parameter m_j

$$S_{ij}(\mathbf{m}) = \frac{\partial f_i(\mathbf{m})}{\partial m_j} \tag{3.82}$$

Since it is a function of the model, we can never speak about ”the sensitivity” but only of the sensitivity of a particular model. Even though the term is often used for the one of the homogeneous half-space or a-priori model. As conductivity changes within the model, the EM fields changes and thus the sensitivity.

Arranging the sensitivities of all forward responses with respect to all model parameters in a matrix yields the Jacobian (or sensitivity) matrix

$$\mathbf{S}_n = \begin{pmatrix} \frac{\partial f_1(\mathbf{m}_n)}{\partial m_1} & \cdots & \frac{\partial f_1(\mathbf{m}_n)}{\partial m_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_N(\mathbf{m}_n)}{\partial m_1} & \cdots & \frac{\partial f_N(\mathbf{m}_n)}{\partial m_M} \end{pmatrix} , \tag{3.83}$$

which is used in non-linear inversion for updating the model vector. The sensitivities of all existing model parameters plotted in model space is referred to as sensitivity distribution. It provides a clear view into the physical processes and thus represents a powerful tool for the comprehension and the interpretation of measurements. In the last decades, numerous papers on sensitivities of electromagnetic measurements were published (e.g. Gómez-Treviño [1987]; McGillivray & Oldenburg [1990]; Spies & Habashy [1995]).

The sensitivity distribution of one single data primarily denotes regions, which are sensitive to the measurement. Following travel-time tomography, the sum of the absolute sensitivities values of all measurements yields the cumulative sensitivity or coverage of a model

$$\mathrm{cov}_j(\mathbf{m}) = \sum_{i=1}^{N} |S_{ij}(\mathbf{m})| \tag{3.84}$$

Note, that the sum of the square values of each line of the sensitivity matrix

$$\sum_{i=1}^{N} S_{ij}^2(\mathbf{m}) \approx R_{jj} , \tag{3.85}$$

is equal to the corresponding diagonal entry of the resolution matrix if one applies an unconstrained gradient based inversion schemes (see section 2.4 equation (2.63). The coverage is therefore a very important tool to investigate the a posteriori model. Even if a constrained inversion is applied, the coverage yields important information into the resolution capabilities of the inverse solution.

The comparison of sensitivity distributions for the individual data point provides insight into how the model parameters can be distinguished from each other. When designing data sets, a trade-off between the abilities of detecting and distinguishing has to be sought. Thus, the coverage or the sensitivity distribution plays an important rule by designing field setups and during feasibility studies.

For homogeneous models, the sensitivity for TEM methods can be calculated directly from analytical equations (e.g. Kaufman & Keller [1983]; Ward & Hohmann [1988]; Spies & Frischknecht [1991]). Although some complicated terms evaluating the error function may arise, they can be expressed in closed form. However, since the response of non homogeneous models can not be expressed in a closed form, the sensitivity of inhomogeneous models can not be calculated directly. In general there are three different methods to get them.

The first and most easiest way would be the finite difference or perturbation method, described in 3.5.1. The second two methods are referred as adjoint methods. Adjoint methods all utilize additional forward calculations. They are explained in section 3.5.2. A special adjoint method is called the adjoint Green function approach or adjoint equation method [McGillivray et al., 1994]. This approach utilizes reciprocity and the adjoint Green function to calculate the influence of the model response at the receiver site due to changes in the model. Since, the sensitivity calculation currently implemented in the 3D TEM inversion code use this method, it is explained more detail than the other methods.

After the theoretical introduction, the implementation of the adjoint Green function approach for TEM is focused and given at the end of this section.

3.5.1 Finite difference sensitivity calculation (perturbation method)

The finite difference (FD) calculation, also known as perturbation method is a straight forward technique to deliver sensitivities for inhomogeneous models. The calculation of a single element of **S** using the perturbation method is achieved building the first order finite difference approximation.

$$S_{ij}(\mathbf{m}) = \frac{\partial f_i}{\partial m_j} = \frac{f_i(m_1, \ldots, m_j(1+\Delta), \ldots, m_M) - f_i(\mathbf{m})}{\Delta} \tag{3.86}$$

It displays finite difference of the model response due to the change in the model. A simple model for the calculation of sensitivities via the FD approach is shown in figure 3.31. Here a simplified setup with a source (T) and a receiver (R) within an offset **r** is displayed. Within the volume (V), the conductivity of the model parameter is changed like $\sigma' = \sigma(1+\Delta)$. Δ should take small values (i.e. $0.01 \leq \Delta \leq 0.1$) to conserve the linear behavior of the first order difference.

Note, that a forward operator generally involves the calculation of model responses for all receiver stations at once or at least in a very few forward runs. However, in order to get the full sensitivity matrix, one has to carry out equation (3.86) for every model parameter. Therefore this method is sometimes called *brute force*. For 2 and 3D inversions the number of parameters usually exceeds several thousands, making the approach unfeasible. Additionally, it can be shown, that the finite volume must exceed a distinct size to achieve a change in the model response. This depends, of course on accuracy of the used forward operator as well as the sensitivity of the setup. E.g. if the sensitivity is already very low due to changes far from the receiver or source. Therefore, the computation via FD must not yield numerically stable results in any case.

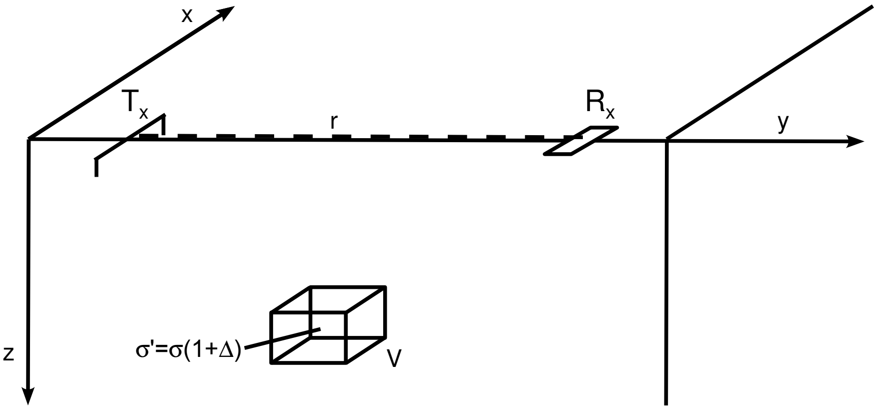


Figure 3.31: *Scheme of the sensitivity calculation using FD displaying a transmitter (T), a receiver (R) at offset (r) and a finite volume (V). Within the volume, the parameter m_j resides and is perturbed ($\sigma' = \sigma(1 + \Delta)$) in order to calculate the finite difference of the model response due to the parameter change. .*

Nevertheless it is often used to verify sensitivities calculated with other methods, which involve more complex numerical issues, or if the number of parameters is quite small.

3.5.2 Adjoint formulation

A often used alternative to the FD method is the computation of sensitivities via the adjoint formulation of the forward problem. If the forward problem can be stated like

$$\mathbf{A}\mathbf{f} = \mathbf{s}_a \ , \tag{3.87}$$

with $\mathbf{A}$ referring to the system matrix, $\mathbf{f}$ is the model response to be calculated and $\mathbf{s}_a$ accounts for a inhomogeneous source term. Since the source term is not depending on the model ($\frac{\partial \mathbf{s}_a}{\partial \mathbf{m}} = 0$), the partial derivatives of $\mathbf{f}$ with respect to the model $\mathbf{m}$ can be formulated like

$$\mathbf{A}\frac{\partial \mathbf{f}}{\partial \mathbf{m}} = -\frac{\partial \mathbf{A}}{\partial \mathbf{m}}\mathbf{f} \tag{3.88}$$

Which is a particular forward problem to be solved for each source term on the right hand side. Since each element of the matrix $\mathbf{A}$ contains the neighboring conductivities of the according node, sensitivity sources occur only at the nodes surrounding the particular element m_j.

However, for each model cell all sources have to be considered, resulting again in solving M single forward equations. If the reciprocity relation is used one can achieve this in less effort reducing the number to just one forward calculation (e.g. Haber et al. [2000; 2007]). However, since I use the multi frequency forward operator as given in section 3.3, the model response is not as easy to evaluate as given in equation 3.87. Additionally, if the system matrix of the forward operator, or it's approximation in Krylov subspace, is not available one has no other choice as to use another adjoint formulation in order to calculate the sensitivity directly.

Adjoint Green function approach

The adjoint Green function approach, or adjoint equation method [McGillivray et al., 1994] utilizes reciprocity and uses the receiver adjoint Green function to calculate the influence of the model response at the receiver site due to changes in the model. The sensitivities computed with this approach is general considered as to be equivalent to the Fréchet derivatives and the formal treatment can be found in Lanczos [1960]; Morse & Feshbach [1963]; Roach [1982]. It is widely used for sensitivity computation of EM methods working in the frequency domain (e.g. Madden & Mackie [1989]; McGillivray & Oldenburg [1990]; Madden [1990]; Oldenburg [1990]; Ellis & Oldenburg [1993]). Hördt [1998] showed a nice exploitation of the integral equation and derived a set of equations directly in time domain using a convolution approach. Since the computation in the time domain and it's derivation is not used that often, I describe it in the following.

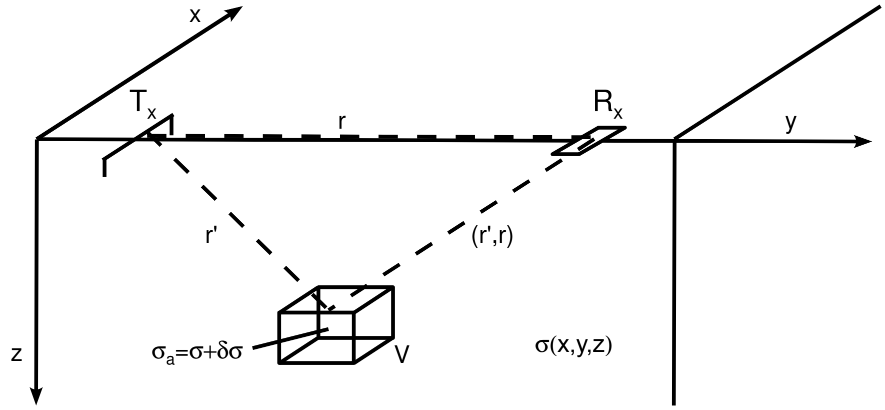


Figure 3.32: *Schematic description of the influence on the signal measured at* $\mathbf{r}$ *due to the anomalous conductivity* σ_a *at the image point* $\mathbf{r}'$ *within volume (V). The dyadic Green function* $\boldsymbol{\mathcal{G}}(\mathbf{r}|\mathbf{r}')$ *relates the anomalous current* $\mathbf{J}_a$ *at* $\mathbf{r}'$ *to the receiver at* $\mathbf{r}$.

The integral equation of the electrical field is given through [Hohmann, 1988]:

$$\mathbf{E}(\mathbf{r},t) = \mathbf{E}^p(\mathbf{r},t) + \int_V \int_0^t \boldsymbol{\mathcal{G}}(\mathbf{r},t|\mathbf{r}',t')\mathbf{J}_a(\mathbf{r}',t')\,\mathrm{d}t'\,\mathrm{d}\mathbf{r}' \,. \tag{3.89}$$

Illustrated in figure 3.32, the total electrical field at the observation point $\mathbf{r}$ can be described by a known primary field $\mathbf{E}^p(\mathbf{r})$ (background field) plus a scattered field generated by the anomalous current $\mathbf{J}_a$ in the volume V at $\mathbf{r}'$. Boerner & West [1989] indicate that the anomalous conductivity change $\delta\sigma_a = \sigma + \sigma_a$ is proportional to a somewhat equivalent electric dipole source distribution. It is parallel to and proportional with the current density $\mathbf{J}_a$ created in the earth at $\mathbf{r}'$ by the source field, thus

$$\mathbf{J}_a(\mathbf{r}',t') = \delta\sigma_a(\mathbf{r}')\mathbf{E}(\mathbf{r}',t') \,. \tag{3.90}$$

Describing the scattered electric field at $\mathbf{r}$ due to a conductivity perturbation by means of a "point-source excitation" at $\mathbf{r}'$ introduces $\boldsymbol{\mathcal{G}}(\mathbf{r},t|\mathbf{r}',t')$ as the tensor or dyadic Green's function [Felsen & Marcuvitz, 1973]. In the case of electric fields (magnetic fields will be considered further below), it represents the electric field impulse response at $\mathbf{r}$ arising from an unit vector force density acting in the direction of the unit vector at $\mathbf{r}'$ [Felsen & Marcuvitz, 1973]. Note, that $\boldsymbol{\mathcal{G}}$ satisfies the solution of the underlying PDE to a unit source [Ward & Hohmann, 1988]

$$\mathbf{A}\boldsymbol{\mathcal{G}}(\mathbf{r},t) = -\delta(x)\delta(y)\delta(z) \,, \tag{3.91}$$

with the differential operator matrix $\mathbf{A} = \nabla^2 + k^2$ and $k^2 = \mu\epsilon\partial_t^2 - \mu\epsilon\sigma\partial_t$. In practice, the elements of $\boldsymbol{\mathcal{G}}$ are obtained by calculating all three Cartesian components of the electric field's impulse response for each Cartesian orientation of the unit dipole source at $\mathbf{r}'$ [Weidelt, 1975]. For example $E_y\boldsymbol{\mathcal{G}}(\mathbf{r},t|\mathbf{r}',t')E_x$ represents the electric field impulse response in the y direction at $\mathbf{r}$ produced by an x directed unit electric dipole at $\mathbf{r}'$. Here, the reciprocity theorem can be applied [Ward & Hohmann, 1988; Hohmann, 1988]:

$$\boldsymbol{\mathcal{G}}(\mathbf{r}'|\mathbf{r}) = \boldsymbol{\mathcal{G}}(\mathbf{r}|\mathbf{r}') \ . \tag{3.92}$$

It says that if both the source/receiver positions and the source/receiver components are interchanged, the calculated fields will be the same. Since $\boldsymbol{\mathcal{G}}(\mathbf{r}'|\mathbf{r})$ is also known as the adjoint dyadic Green function of $\boldsymbol{\mathcal{G}}(\mathbf{r}|\mathbf{r}')$, the reciprocity relation states the adjungation of the Green function to the receiver. Expressed by the example, instead of calculating the y directed electric field impulse response at $\mathbf{r}$ due to a x directed electric dipole, one can reverse the process such that the unit transmitter at $\mathbf{r}'$ operates as a receiver. Thus, the x directed electric field at $\mathbf{r}'$ caused by an impulsive y directed electric dipole source at $\mathbf{r}$ is identical to the example above.

For small conductivity changes $\sigma_a \rightarrow 0$ within the finite volume V, the total electrical field equals the background electric field $\mathbf{E}(\mathbf{r}') = \mathbf{E}^p(\mathbf{r}')$, which can already be seen from equation (3.89) (Born approximation). This includes the assumption that $\mathbf{E}(\mathbf{r}')$ is Fréchet differentiable, that is the neglected remainder term of the approximation is of second order in $\delta\sigma_a$ [Chave, 1984; Boerner & West, 1989]. This leads to the sensitivity equation for electrical receivers [Hördt, 1998]:

$$\frac{\delta\mathbf{E}(\mathbf{r},t)}{\delta\sigma_a} = \int_V \int_0^t \boldsymbol{\mathcal{G}}(\mathbf{r}',t|\mathbf{r},t')\mathbf{E}^p(\mathbf{r}',t')\,\mathrm{d}t'\,\mathrm{d}\mathbf{r}' \ , \tag{3.93}$$

with $\delta\mathbf{E}(\mathbf{r},t) = \mathbf{E}(\mathbf{r},t) - \mathbf{E}^p(\mathbf{r},t)$. From a classical point of view, the left-hand side represents a measure for the field variation at $\mathbf{r}$ produced by a change in the model parameter at $\mathbf{r}'$, thus it is the parameter sensitivity. The integral on the right-hand side represents an efficient way of calculating the sensitivities for the whole model domain. It is realized by carrying out a convolution of the background electric field $\mathbf{E}^p(\mathbf{r}')$ with the electric field impulse response at $\mathbf{r}'$ sourced at $\mathbf{r}$ by an unit step electric dipole oriented like the original receiver. The method is efficient, because reciprocity implies that the sensitivity at $\mathbf{r}$ for each image point $\mathbf{r}'$ over the model is obtained by a single convolution [Hördt, 1998].

Adjoint Green function for magnetic field sensors

Analogue to the derivation of the electrical field perturbation due to the conductivity change in the image point $\mathbf{r}'$, the equation describing the change of the magnetical flux due to the parameter changes in the image point can be derived. The integral equation for the magnetical field is given through

$$\mathbf{H}(\mathbf{r},t) = \mathbf{H}^p(\mathbf{r},t) + \int_V \int_0^t \boldsymbol{\mathcal{G}}^m(\mathbf{r},t|\mathbf{r}',t')\mathbf{J}_a(\mathbf{r}',t')\,\mathrm{d}t'\,\mathrm{d}\mathbf{r}' \ . \tag{3.94}$$

With $\boldsymbol{\mathcal{G}}_m(\mathbf{r},t|\mathbf{r}',t')$ the dyadic Green function of an electrical point source at $\mathbf{r}'$ generating a magnetical field impulse response at $\mathbf{r}$. Here, special care must be taken to relate the adjoint Green function of a magnetical receiver to the electrical source at $\mathbf{r}'$. Interchanging the source and receiver function can be carried out using the reciprocity relation between the electrical receiver moment $\widetilde{\mathbf{D}}$ and magnetical receiver moment $\widetilde{\mathbf{M}}$

$$\widetilde{\mathbf{D}} = -\mu \partial_t \widetilde{\mathbf{M}} \tag{3.95}$$

It expresses, that the magnetic field impulse response of an electric source is equivalent to the electric field step response of a magnetic dipole source [Hördt, 1998]. Also, the dyadic Green function is a causal function

$$\boldsymbol{\mathcal{G}}(\mathbf{r}, t|\mathbf{r}', t') = 0, \quad t \leq t' \ . \tag{3.96}$$

Therefore the adjoint dyadic Green function for the magnetic receiver step response reads

$$\begin{aligned}
\boldsymbol{\mathcal{G}}^{m,\text{step}}(\mathbf{r}', t|\mathbf{r}, t') &= \int_t^{t'} \boldsymbol{\mathcal{G}}^m(\mathbf{r}', t|\mathbf{r}, t') \, \mathrm{d}t' \\
\text{(Eq. (3.95))} \quad &= -\mu \int_t^{t'} \partial_t \boldsymbol{\mathcal{G}}^m(\mathbf{r}, t|\mathbf{r}', t') \, \mathrm{d}t' \\
&= -\mu \boldsymbol{\mathcal{G}}^m(\mathbf{r}, t'|\mathbf{r}', t') + \mu \boldsymbol{\mathcal{G}}^m(\mathbf{r}, t|\mathbf{r}', t') \\
\text{(Eq. (3.96))} \quad &= \mu \boldsymbol{\mathcal{G}}^m(\mathbf{r}, t|\mathbf{r}', t') \ .
\end{aligned} \tag{3.97}$$

Thus, $\boldsymbol{\mathcal{G}}^{m,\text{step}}(\mathbf{r}', t|\mathbf{r}, t')$ is equivalent to the magnetic field impulse response at $\mathbf{r}$, due to a unit-impulse electrical source at $\mathbf{r}'$.

Following the deliberations for the sensitivities of the electrical field receiver, one finds a similar notation for the sensitivity of the magnetical field receiver

$$\frac{\delta \mathbf{H}(\mathbf{r}, t)}{\delta \sigma_a} = \int_V \int_0^t \boldsymbol{\mathcal{G}}^{m,\text{step}}(\mathbf{r}', t|\mathbf{r}, t') \mathbf{E}^p(\mathbf{r}', t') \, \mathrm{d}t' \, \mathrm{d}\mathbf{r}' \ , \tag{3.98}$$

where $\delta \mathbf{H}(\mathbf{r}, t) = \mathbf{H}(\mathbf{r}, t) - \mathbf{H}^p(\mathbf{r}, t)$ and $\mathbf{E}^p(\mathbf{r}', t')$ is the same background field as in equation (3.93). The adjoint Green function $\boldsymbol{\mathcal{G}}^{m,\text{step}}$ is the electrical field impulse response of an unit step magnetic field sourced at $\mathbf{r}$.

Since the magnetic field is not always measured directly, but the change of the magnetic flux over time, the derivation of the sensitivity for the induced voltage in a receiver coil is of interest as well. The derivation of the adjoint Green function for the voltage is given by the relation

$$\boldsymbol{\mathcal{G}}^m(\mathbf{r}', t|\mathbf{r}, t') = -\mu \frac{\partial \boldsymbol{\mathcal{G}}^m(\mathbf{r}, t|\mathbf{r}', t')}{\partial t} \ . \tag{3.99}$$

The adjoint Green function of the induced magnetic flux change measured in a coil ($\boldsymbol{\mathcal{G}}^m(\mathbf{r}', t|\mathbf{r}, t')$) is equivalent to the induced voltage impulse response (which is indeed the second derivative of magnetic field step response with respect to time), of an electric field unit-impulse sourced at $\mathbf{r}'$.

In agreement to Hördt [1998], the sensitivity equation for the voltage measured at $\mathbf{r}$ due to a conductivity change in the image point at $\mathbf{r}'$ is given by

$$\frac{\delta \dot{\mathbf{H}}(\mathbf{r}, t)}{\delta \sigma} = \int_V \int_0^t \boldsymbol{\mathcal{G}}^m(\mathbf{r}', t|\mathbf{r}, t') \mathbf{E}^p(\mathbf{r}', t') \, \mathrm{d}t' \, \mathrm{d}\mathbf{r}' \ , \tag{3.100}$$

where $\delta \dot{\mathbf{H}}(\mathbf{r}, t) = \dot{\mathbf{H}}(\mathbf{r}, t) - \dot{\mathbf{H}}^p(\mathbf{r}, t)$ and $\mathbf{E}^p(\mathbf{r}', t')$ is again the same background field as in equation (3.93). The adjoint Green function $\boldsymbol{\mathcal{G}}^m$ is the electrical field impulse response of an unit impulse magnetic coil sourced at $\mathbf{r}$.

3.5.3 Implementation of TEM sensitivity calculation

The Born approximation implies, that the local change of the conductivity does not influence the (background) electrical field in the volume V at $\mathbf{r}'$, if the conductivity change and the volume is small enough. For our purpose, where the conductivity change should indeed vanish, one can assume that the assumption is valid. With a proper discretized parameterization of the volume V where the conductivity is assumed to be constant, the 3D integration of $\mathbf{r}'$ becomes independent from the integrand. This means $\boldsymbol{\mathcal{G}}(\mathbf{r}', t|\mathbf{r}, t')$ and $\mathbf{E}^p(\mathbf{r}', t')$ are constant within V, and the neglected remainder term of the approximation is of second order in $\delta\sigma_a$. Thus, one can now derive a volume based fractional sensitivity or sensitivity density

$$\mathbf{S}^{e,m}(\mathbf{r}', t) = \int_0^t \boldsymbol{\mathcal{G}}^{e,m}(\mathbf{r}', t|\mathbf{r}, t')\mathbf{E}^p(\mathbf{r}', t')\,\mathrm{d}t' \;, \tag{3.101}$$

which is assumed to be constant over V. The assumption $\mathbf{S}^{e,m}(\mathbf{r}')|_V =$ const holds, if the changes in $\mathbf{S}$ are of second order in $\delta\sigma_a$ or the volume is sufficiently small.

$\mathbf{S}^{e,m}$ accounts for a measured field change at $\mathbf{r}$ due to a small conductivity change at $\mathbf{r}'$ per volume $[\Delta V] = m^3$. The superscript e denotes the adjoint Green function of an electrical field receiver and m refers to the adjoint Green function of the voltage measured in a receiver coil at $\mathbf{r}$. Units of the sensitivity density for the electrical receiver are given in $[\mathbf{S}^e] = V^2/A/m^3$, whereas the sensitivity density of the the voltages is denoted by $[\mathbf{S}^m] = V^2/A/m^4$. For simplicity, the abbreviation “adjoint field” is used in the following. It is a simplification for the adjoint Green function which is always the electrical impulse response, either due to an electrical or a magnetical receiver acting as the source.

In the current development state of the 3D TEM inversion program, three different approaches for a sensitivity density calculation are implemented. Their designation follows the combination of the method which is used to calculate the primary and the adjoint fields:

1. 1D/1D (1D sensitivities), where the primary and the adjoint fields are calculated using 1D solutions.

2. 3D/1D (hybrid sensitivities), where the primary field calculation is carried out using the 3D code *SLDMem3t* and the adjoint fields are calculated as 1D responses.

3. 3D/3D (3D sensitivities), where all fields are calculated with the 3D forward operator *SLDMem3t*.

The first approach considered to be the most easiest way to implement the sensitivity calculation. If the primary or adjoint fields are computed as 1D solutions, the Bessel integration from the wavenumber domain to the Euclidean space delivers all field values within one single forward run for each frequency. This can be carried out in no time compared to a 3D field computation. Since the field values are computed within the whole modeling domain, it is independent from a grid and the computed sensitivity density can be integrated without taking any complicated interpolation schemes into account (see below). Additionally, the convolution of the background and the adjoint fields can be carried out directly in the frequency domain which is just a multiplication. Followed by a inverse Fourier transform this yields the 1D sensitivity. Nevertheless, it is computed to account for a 3D parameterization of the model.

To increase the 3D content of the sensitivities, the background fields and the adjoint fields are computed with *SLDMem3t*. Though, applying the SLDM for the adjoint field computation means, that the field values are restricted to the grid which is used for their computation. In addition to more complex discretization structure of the field values, the numerical treatment of the convolution has to be taken care of. Thus, three major tasks do arise now

1. Interpolation of the adjoint fields and the background response to a spatial coincident discretization.
2. Carry out the discrete convolution of $\boldsymbol{\mathcal{G}}^{e,m}$ and $\mathbf{E}^{p}$ yielding a spatial discretized sensitivity density.
3. The discretization of the model to invert (parameterization) may be different to that of the forward grid. The integration of the sensitivity density to calculate the influence of each parameter onto the measurement may therefore be considered as re-weighting the sensitivity density to the parameterization of the inverse problem.

The first two tasks come along with inherent features of the spatial discretization belonging to the forward operator. For inhomogeneous conductivity models the *SLDMem3t* is used, whereas for homogeneous models, a fast 1D solution would be preferred. Since, the latter case has some additional benefits to speed up the sensitivity calculation (e.g. multiplication in the frequency domain), the integration gets more attention.

Sensitivity density of homogeneous or 1D models

For sensitivities of the starting model, which may be homogeneous, the sensitivity calculation may be carried out using 1D solutions only. Of course, if one uses prolonged sources (e.g. bipole), the background field would be given by superposition of fields originating of translated dipole source currents. Thus, the 1D background fields would be better referred to as 1.5D fields. However, if only 1D solutions are used, the fields would be calculated in the frequency domain. Therefore, the convolution in time domain would be carried out as multiplication in the frequency domain, which would spare a lot of numerical effort.

If the step on excitation of the used dipole sources can be expressed as heavy-side function in time

$$f(t)^{\mathrm{step}} = \begin{cases} 0 & t < 0 \\ f_0 & t > 0 \end{cases} ; ,$$

the step on response of the frequency dependent field $f(\omega)$, expressed by the fast Hankel transform (see section 3.1.3), reads

$$f(t_n)^{\mathrm{step}} = \frac{1}{t_n}\sqrt{\frac{2}{\pi}} \sum_{j=1}^{N_t+N_f-1} \Im\left(\frac{f(\omega_j)}{i\omega_j}\right) \tilde{H}_{-\frac{1}{2}}(n-j) \ .$$

Thus, the impulse response is evaluated by

$$f(t_n)^{\mathrm{impulse}} = \frac{1}{t_n}\sqrt{\frac{2}{\pi}} \sum_{j=1}^{N_t+N_f-1} \Im\left(f(\omega_j)\right) \tilde{H}_{-\frac{1}{2}}(n-j) \ , \qquad (3.102)$$

where N_t is a given number of time points and N_f a fixed number of filter coefficients. For convenience, I use the filter with ten sampling points per decade, as already explained in section 3.1.3. Therefore, $t_n := 10^{\frac{n-1}{10}} t_{min}, \quad n \in \{1, \ldots, N_t\}$ and $\omega_j := 10^{\frac{41-j}{10}} \frac{1}{t_{min}}, \quad j \in \{1, \ldots, N_t + 79\}$.

With the background $\mathbf{E}^p(\omega)$ and adjoint fields $\mathbf{E}^{e,m}(\omega)$, the inverse Fourier transform of the TEM sensitivity density of 1D models may be expressed through a Hankel transform as well:

$$\mathbf{S}^{e,m}(\mathbf{r}', t_n) = \frac{1}{t_n}\sqrt{\frac{2}{\pi}} \sum_{j=1}^{N_t+N_f-1} \Im\left(\frac{\mathbf{E}^{e,m}(\mathbf{r}', \omega_j)\mathbf{E}^p(\mathbf{r}', \omega_j)}{i\omega_j}\right) \tilde{H}_{-\frac{1}{2}}(n-j) \,. \tag{3.103}$$

Note, that for two complex functions f, g, the imaginary part of their product over $i\omega$ reads

$$\Im\left(\frac{fg}{i\omega}\right) = \frac{\Re(f)\Re(g)}{\omega} - \frac{\Im(f))\Im(g))}{\omega} \,. \tag{3.104}$$

Thus, the sensitivity density is calculated using every part of the complex valued and frequency depending background and adjoint fields. For convenience, the equations for the adjoint fields using various sources are given in appendix A.

A big advantage of this formulation is, that equation 3.103 can be easily evaluated at any $\mathbf{r}'$ within the model. Thus, to calculate the influence of any model parameter on the data, the spatial integration of 3.103 may be carried out for any desired discretization. Yet, to achieve comparable results and for simplicity, the discretization may be coincident with the grid of the 3D forward operator.

Parallelization, interpolation and integration of spatial discrete field values

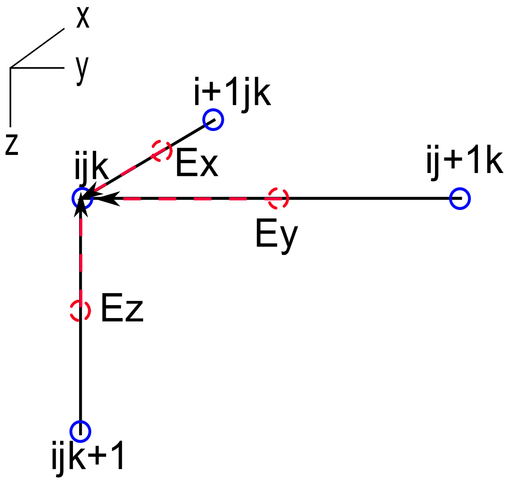


Figure 3.33: *Simple interpolation scheme for the background field ($\mathbf{E}^p$). It relates the node centered grid values (red dashed circles) back to the node points (blue solid circles). The original field values of the background field according to the orientation within a Yee-Lebedev grid is displayed along the edges (cf. Fig. 3.11). Each grid node point is referred to with the index set ijk, where i is denoting the x, j the y and k the z grid node number.*

If the *SLDMem3t* is used to calculate the background field, the inherent Yee-Lebedev discretization demands an interpolation of the calculated fields to coincident grid nodes. As already displayed in figure 3.11, the electrical field values resides not on each grid node, but are calculated on the edge centers. To take this into account a simple interpolation scheme is set up for the background electrical fields. Illustrated in figure 3.33, the interpolation scheme relates the edge centered original field values (red dashed circles) to the grid nodes (blue solid circles). As displayed in the figure, the original fields are present at different positions within the staggered grid for every orientation. E.g. the background field ($f \equiv \mathsf{E}_\mathrm{x}^p$) residing at $i + 1/2jk$ is related to the node point ijk by the linear operator Q.

$$Q(f_{ijk}) = f_{i+\frac{1}{2}jk} + \left(\frac{f_{i+\frac{1}{2}jk} - f_{i-\frac{1}{2}jk}}{x_{i+\frac{1}{2}jk} - x_{i-\frac{1}{2}jk}} \right) \left(x_{ijk} - x_{i+\frac{1}{2}jk} \right) \tag{3.105}$$

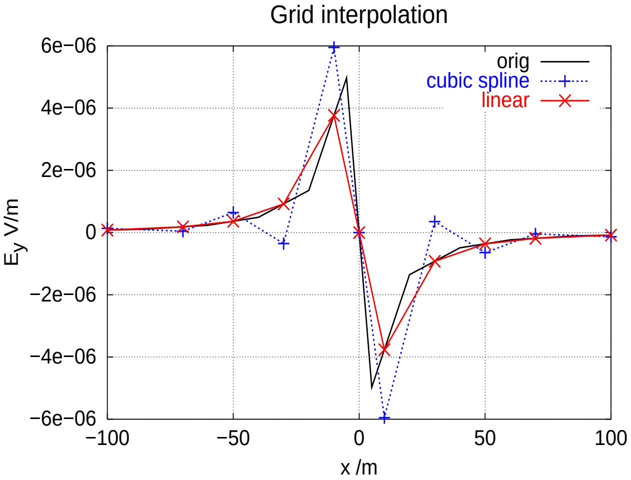


Figure 3.34: *Comparison of cubic spline and linear interpolation of discontinuous field values from Martin [2003]. The original field values of the y-component of the electrical field is displayed in black solid, the cubic spline interpolated field values are given as dotted blue graph and the linear interpolation is shown as red solid line.*

Because the field values are discontinuous in space, a cubic spline interpolation should be treated carefully. A comparison of the grid interpolation results of a cubic spline (cubic spline) and a linear interpolation (linear) is illustrated in figure 3.34. As expected for a third order interpolation, the cubic spline interpolation does not reproduce the original field values around the discontinuity properly. Before and after the sign reversal, one will notice strong oscillations of the interpolated field values (cf. fig. 3.34 dotted blue). To treat the discontinuous field values better, I implemented a linear interpolation scheme for this task, which is quite easy to implement within a regular grid. As anticipated, the results of the linear interpolation are more stable to the outliers of the discontinuity (cf. fig. 3.34 solid red). Therefore, the grid interpolation is and should be carried out with a linear operator.

As suggested by equation 3.101, the background field should be calculated with the most spatial resolution, in order to ensure that $\mathbf{S}^{e,m}(\mathbf{r}')|_V =$ const holds for each volume V. As turned out by the grid study, the most stable regular discretization (e.g. for a rectangular loop transmitter) can be found with $70 \leq N_g \leq 80$ grid nodes. For all three electrical field components of $\mathbf{E}^p$ this gives roughly $1 - 1.5 \times 10^6$ field values to compute for the grid of the background field (background grid).

Remember, that the memory usage of the *SLDMem3t* is roughly given by $N_n \times \mathsf{M}$, with N_n equals the number of field values to compute and M denoting the dimension of the

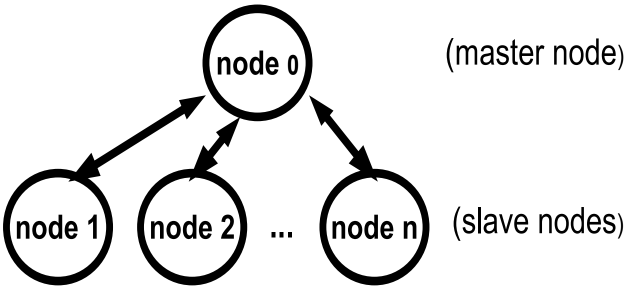


Figure 3.35: *Parallelization scheme for distributed memory systems (Cluster) using MPI. The master node (node 0) distributes the workload to the slave nodes (node 1 ... node n) within the cluster. The arrows denoting network communication between the slave and master layer. More details are given in the text.*

Krylov subspace. If we assume a realistic value for M like $1-2\times10^4$ the used memory by *SLDMem3t* would be given by $1-3\times10^{10}\times8$ Bytes. One can imagine, that a factor of two, let's say by reducing the number of vertical grid nodes, would spare a lot of memory usage. However, even with coarse grids and less resolution of the sensitivity density one can not store all field values within conventional PC's. Hence, the whole problem demands effective parallelization which is carried out using the Message Passing Interface (MPI) on Linux Clusters (more details on MPI can be found in appendix D.1).

The underlying principle of the parallelization scheme is rather simple:
Divide and conquer.

Illustrated in figure 3.35, the cluster is working as a system of single nodes each having a CPU with own memory. Therefore, a Cluster is normally considered as a *distributed memory* system. Since the CPU's in a Linux Cluster are considered as relatively fast (in terms of floating point operations) compared to massive parallel workstations, where the single CPU's are relatively slow but have (more or less) access to the same memory, the problem may be equally divided onto the slave nodes. The disadvantage using a Cluster may be the relatively high data transfer between the master and the slave nodes, denoted by the black arrows in figure 3.35, which may be significantly reduced using hard-disk data transfer for big arrays. However, as the underlying principle, divide and conquer, demands, the parallelization scheme is not trivial. Within the program code one has to take care of every distribution step, demanding highly dynamic memory allocation and the implementation may be more costly.

As already mentioned, the parallelization for the computation of the background field is achieved by dividing the background grid of the whole model domain into sub-domains. Each sub-domain consists of a region and number of grid points, where the field values should be computed. This partitioning of the grid is done by the master node and distributed to free nodes (slave nodes) within the cluster framework. Each slave node carries out the computation of the field values and writes it's success to the master node via network, waiting for more work. For convenience, the master node can be utilized as slave node, too. If everything is done, the master node collects all the field values from hard-disk and carries out the interpolation of the background fields to the grid nodes of the background grid.

Now, that the background field is calculated, the computation of the adjoint fields must be taken care of. Assume our TEM measurements are done with N_{R} receiver sites, thus there are N_{R} adjoint fields to compute. Since a parallelization scheme is already used, it would be sensible to let each slave node carry out the computation of it. This scales the whole parallelization scheme to the number of receivers, which enhances the overall performance drastically. Further assume, the adjoint field is computed with fast 1D solutions, as given in appendix A.1. After distribution of the background field to each node, the convolution and the integration of the sensitivity density can thus be carried out parallel as well. For convenience, the 1D solution is far less expensive in terms of computation time. Additionally, there is no need for a special interpolation scheme for this case, because 1D solutions can be calculated for any **r** within the whole modeling domain in a little while using the fast Hankel transform (see appendix 3.1.2).

Since the sensitivity density is considered to be constant over the volume V, it can be put outside the integral. Than, the volume is simply

$$\int_V \mathrm{d}\mathbf{r}' = \Delta x'\Delta y'\Delta z' = \Delta V(\mathbf{r}') \; . \tag{3.106}$$

In order to calculate the influence of the change of conductivity within the volume associated image point j on the measured data point of the receiver (t_i), the $\mathbf{r}'$ integration of the sensitivity density may be carried out like

$$S_{ij} = \Delta V_j(\mathbf{r}')\mathbf{S}_j(\mathbf{r}', t_i) \; . \tag{3.107}$$

For simplicity, the superscript of the sensitivity density is dropped. Illustrated by figure 3.36,

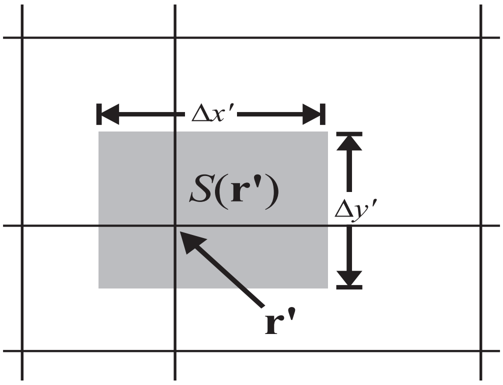


Figure 3.36: *Integration of the sensitivity density (*$\mathbf{S}$*) at the image point* $\mathbf{r}'$ *over the area spanned by* $\Delta x'\Delta y'$*. The gray shaded face denotes the area of constant* $\mathbf{S}$*.*

where the sensitivity density is assumed to be constant over the shaded area, the sensitivity of the volume associated image point may be calculated by equation (3.107).

With this simple integration scheme of the sensitivity density, one would achieve the sensitivity of a model parameterization which would depend on the grid itself. It can be seen as a background sensitivity density, since it depends on the background grid. Yet, within a general 3D inversion program, one would like to parameterize the model independent from the grid of the forward operator. Indeed, it turned out, that the stability of the inverse problem can be drastically enhanced if one parameterizes the model independent from the grid.

To implement this, a weighting scheme of the background sensitivity, to account for the parameterization of the model, is carried out. This scheme is applied analogue to the material averaging scheme, but with the opposite direction. Within the MA scheme, already discussed in section 3.3, the conductivity of the model cell is split onto the grid realizing the model parameterization independent of the forward grid. Now, the grid based sensitivity must be weighted according to the parameterization of the model. Thus, it can be described as inverse material averaging or re-weighting of the sensitivities.

The parameterization of the conductivity model is given through

$$\sigma(\mathbf{r}) \simeq \int_{\Omega} x(\mathbf{r}) \approx \sum_{j=1}^{M} m_j \theta_j(\mathbf{r}) \ . \tag{3.108}$$

Thus, for the j'th parameter, the re-weighting of the background sensitivity may be done by using fractioned parts of each volume within θ_j

$$S_{ij} = \sum_{k=1}^{M_j} \delta V_k(\mathbf{r}') \mathbf{S}^{e,m}(\mathbf{r}', t_i) \ , \tag{3.109}$$

where $\delta V_k(\mathbf{r}')$ is the part of $\Delta V(\mathbf{r}')$ which contributes to θ_j.

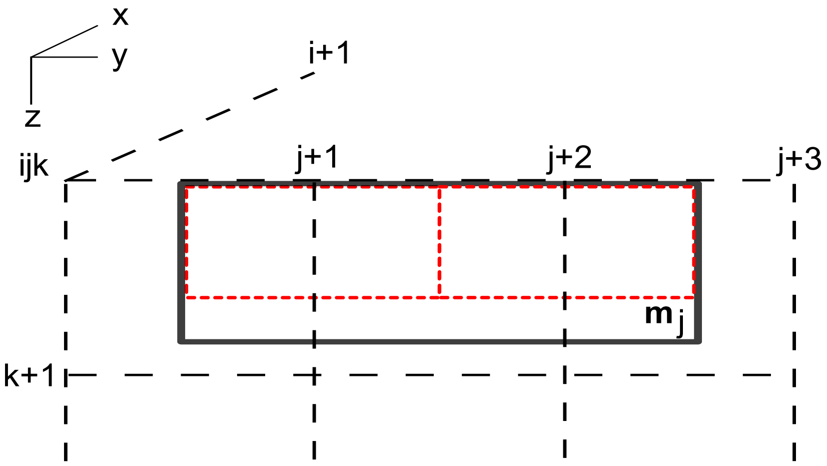


Figure 3.37: *Parameterization of the model for surface near regions. The model parameter m_j (solid rectangle) is defined within a volume between some grid node points (dashed black lines). Red dashed lines mark the areas influenced by the background sensitivity from the surface.*

During some initial testing of the re-weighting scheme it turned out, that surface near parameters are sometimes over-accentuated. Illustrated in figure 3.37, where the parameter m_j is residing between the grid node-points in vertical direction, the sensitivity calculation of the surface near parameter is mainly influenced by the background sensitivity of the surface.

To treat the high surface values of the sensitivities appropriate a second interpolation scheme within the re-weighting process of the background sensitivity is set up. Within this, the sensitivity density values corresponding to a image point, are interpolated between adjacent grid nodes to the center of the fractional volume $\delta V(\mathbf{r}')$. Thus, for the j'th parameter, the interpolated re-weighting scheme of the background sensitivity reads

$$S_{ij} = \sum_{k=1}^{M_j} \delta V_k(\mathbf{r}') Q(\mathbf{S}^{e,m}(\mathbf{r}', t_i)) \ , \tag{3.110}$$

where Q is a linear interpolation operator.

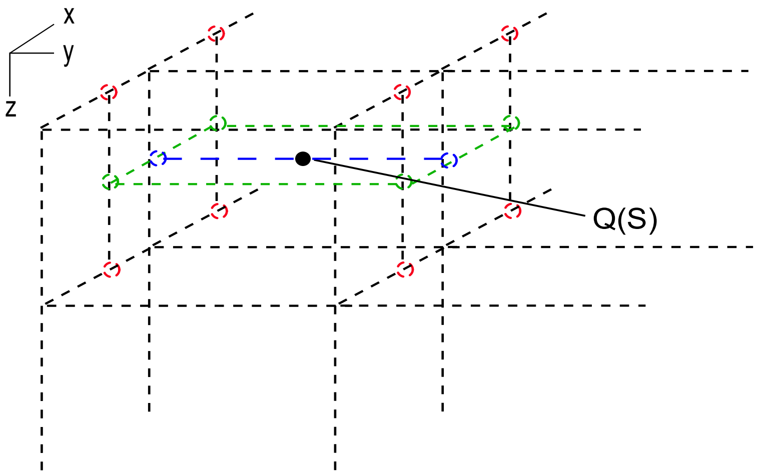


Figure 3.38: *Spatial interpolation scheme for the calculation of re-weighted sensitivities ($Q(S)$). Dashed black lines show the background grid where open red circles refer to the sensitivity density. The open circles and dashed area which is colored green denotes an intermediate interpolation stage. Interpolation stage three is denoted with the single dashed blue line.*

Figure 3.38 shows a sketch of the linear interpolation scheme implemented for a 3D grid. Compared to the single 1D interpolation as given in equation (3.105), the spatial interpolation scheme is more complex. Generally, the spatial interpolation operator (Q) works in three steps. From 8 adjacent grid nodes, two intermediate interpolations are carried out: The first step interpolates from the 8 point volume to a 4 point plane (cf. fig. 3.38 green dashed area). In the next step, the 4 point plane is interpolated to a two point line, denoted by the dashed blue line in figure 3.38. In the last step, the point associated value $Q(S)$ is computed out of the interpolated values of the line.

Of course, the three steps can be compiled into one single equation, using the slopes between the 8 adjacent points. For simplicity it may be explained in the three step operations given above. Because, the linear 3D interpolation operator calculates the interpolated value out of 8 adjacent grid nodes, it is often referred to as linear 8 point operator.

Fortunately, the spatial interpolated re-weighting procedure does no only deliver a low pass version of the background sensitivity density, but it ensures moreover, that the sensitivity density can be treated as piecewise continuous function in space. Thus, as a side effect, the spatial re-weighted sensitivities reduce the error that was made utilizing the Born approximation.

Calculation of adjoint fields with *SLDMem3t*

Compared to the relatively small effort implementing the sensitivity computation with 1D adjoint fields, some more care has to be taken if the adjoint fields are calculated with *SLDMem3t*. Taking the *SLDMem3t* as forward operator for the adjoint fields may be implemented utilizing the background grid with different source position. As demanded by the adjoint Green function, a different source would be used as well. The benefit of this approach is easily perceived: One does not need to invoke further interpolation schemes to bring the adjoint fields to the background grid.

But, as the background grid is not principally designed for the particular source, this may lead to faulty adjoint fields itself. Yet, during some initial testing it turned out, that the use of another grid, the adjoint or secondary grid, would be more effective for this purpose. By using a secondary grid, this grid can be adjusted according to the source. Thus, the overall performance of the grid and the computation as well as the quality of the adjoint fields are drastically enhanced. On the other hand, the adjoint fields has to be interpolated (e.g. by the linear 8 point operator Q) to the background grid which would indeed lead to some numerical errors.

However, if a full 3D computation of the adjoint fields would be carried out, the same problem arises as for the computation of the background field does: The high memory usage of the *SLDMem3t*.
To take this into account, a similar divide and conquer scheme, as already explained for the computation of the background field, is set up. But now, with the limitation, that this time a single node has to take care of the whole grid domain.

For realistic values of the grid and needed field values this leads to a huge computational effort, compared to a 1D adjoint field computation, if a large number of receiver sites is used. Therefore one has to keep in mind what would be more effective: The full 3D sensitivity computation with more or less exact sensitivity values but much more computational effort, or a hybrid scheme with full 3D background fields combined with fast 1D adjoint fields in far less (some orders of magnitude!) time. However, as we will see in chapter 4, and as already investigated by Farquharson & Oldenburg [1996], the hybrid scheme would go well within a 3D inversion and the additional effort may be wasted.

Discrete convolution of the adjoint Green function and the background field

Now, by investigating the sensitivity density for the single image point $\mathbf{r}'$, and since the convolution operator is commutative, the equations

$$\mathbf{S}^{e,m}(t) = \int_0^t \boldsymbol{\mathcal{G}}^{e,m}(t-t')\mathbf{E}^p(t')\,\mathrm{d}t' \tag{3.111}$$

$$= \int_0^t \mathbf{E}^p(t-t')\boldsymbol{\mathcal{G}}^{e,m}(t')\,\mathrm{d}t' \ , \tag{3.112}$$

apply. Or, using a more concise notation with the convolution operator (3.41)

$$\mathbf{S}^{e,m}(t) = \boldsymbol{\mathcal{G}}(t) * \mathbf{E}^p(t) = \mathbf{E}^p(t) * \boldsymbol{\mathcal{G}}(t) \ . \tag{3.113}$$

This expression may be further simplified by taking just a single receiver component into account. For e.g. if one calculates the sensitivity of an E_x-receiver one would like to write

$$S(t) = \frac{\partial \mathbf{E}(t)}{\partial t} * \mathbf{E}^p(t) \ , \tag{3.114}$$

instead, where $\frac{\partial \mathbf{E}(t)}{\partial t} = \boldsymbol{\mathcal{G}}^{\mathsf{E}_\mathsf{x}}(t)$ denotes the electric field impulse response of the electrical E_x-receiver at $\mathbf{r}$ acting as source.

As already described by Hördt [1998], there are several things to care for, if one has to carry out the convolution of the adjoint fields with the background fields. Especially if the fields

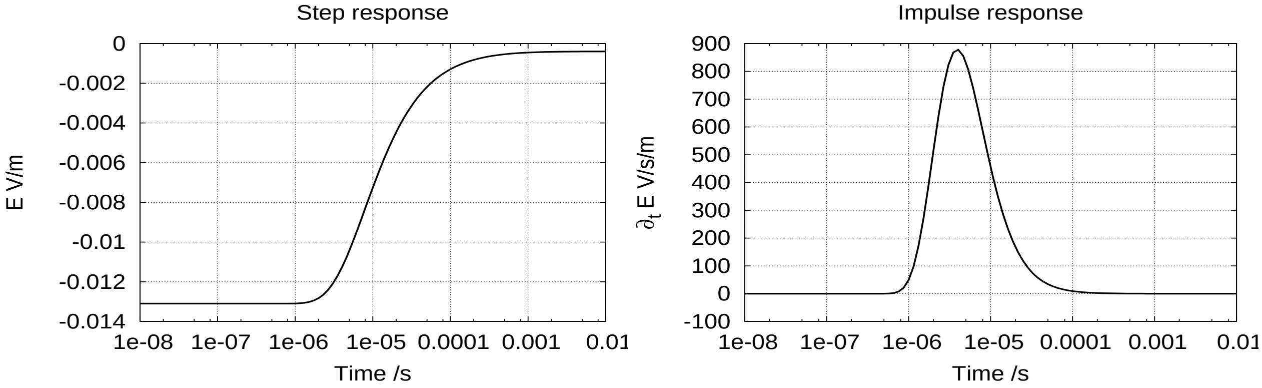


Figure 3.39: *Step and impulse response of the electrical field parallel to a grounded wire over log time. The responses are calculated close to the source expressing the rapid change in the field step response (left) and impulse response (right).*

are non causal (e.g. using a galvanic bipole), or the sensitivity should be given to image points very close to the transmitter or receiver.

This is illustrated in figure 3.39, where the step response of a point in the vicinity of a LOTEM transmitter is given to the left and the corresponding impulse response is given to the right. The simulated data of figure 3.39 is calculated for a relatively large time window ($10\ ns \leq t \leq 10\ ms$) in order to cover up the time change of the field values appropriate. As one can see from the figure, the background field of a grounded dipole is not causal at the surface of a conducting half-space (left). Additionally, the impulse response is decaying rapidly and drops to zero for very early times ($\partial_t \mathsf{E}_\mathrm{x}|_{t \geq 0.1\,ms} = 0$). This may become crucial, because the time window for the calculation of the adjoint fields is commonly not designed to cover these early times. Therefore it may become interesting to achieve some workaround which would take care for this. To overcome the problematic behavior of the impulse response, Hördt [1998] investigates some corrections for the adjoint fields. With the integral condition

$$\begin{aligned} \mathbf{E}^{DC} &= \int_0^\infty \frac{\partial \mathbf{E}}{\partial t}\, dt \\ &= \lim_{\epsilon \to 0} \underbrace{\int_0^\epsilon \frac{\partial \mathbf{E}}{\partial t}\, dt}_{\text{Singularity}} + \underbrace{\int_\epsilon^\infty \frac{\partial \mathbf{E}}{\partial t}\, dt}_{\text{Regular part}} \quad , \end{aligned} \tag{3.115}$$

a separation of the singular part from the regular part of the impulse response is achieved. Now, the correction

$$\frac{\partial \mathbf{E}}{\partial t}|_{t \leq t_{min}} \approx \frac{\mathbf{E}^{DC}}{\Delta t} - \sum_{i=1}^{N_t} \frac{\partial \mathbf{E}}{\partial t}(t_i) \tag{3.116}$$

may be applied to stabilize the numerical treatment of the impulse response for surface near image points.

If the impulse response is not calculated directly by the program but the step response, one may implement a numerical differentiation of the step response using central differences

$$\frac{\partial f}{\partial t}(t_i) \approx \frac{\Delta f}{\Delta t}(t_i) = \begin{cases} \frac{(f(t_2)-f(t_1))}{t_2-t_1} & i=1 \\ \frac{(f(t_{i+1})-f(t_{i-1}))}{2(t_{i+1}-t_{i-1})} & 1<i<N_t \\ \frac{\left(f(t_{N_t})-f(t_{N_t-1})\right)}{t_{N_t}-t_{N_t-1}} & i=N_t \end{cases} , \tag{3.117}$$

which turns out to be advantageous especially for treating singular points of the impulse response. With the proper treatment of the particular singular points ($i = 1$), the convolution integral may be carried out like treating causal functions.

The causal convolution of linear equidistant data (e.g. $g_i = g(\Delta i)$, $f_i = f(\Delta i)$), can be carried out like

$$\tilde{f}_i = \Delta \sum_{j=0}^{i} g_j f_{i-j}, \quad \text{for } i = 0, \ldots, N_t \, . \tag{3.118}$$

Because of the large time frame which is normally used for transient responses, and in order to reduce memory usage, the time points of the field values are generally sampled and stored logarithmic equidistant. Suppose the transient response of three decades in time ($N_t = 1000$), the logarithmic sampling of $N_d = 10$ samplings per decade would reduce the storage about $\frac{N_t}{N_d log_{10}(N_t)} \approx 33$. Further note that, if the log-spaced functions are interpolated to linear samplings and the convolution is carried out using above equation, the number of multiplications would roughly be N_t^2, which turns out to be very time consuming. Imagine, that the convolution has to be done for three orientations separately and for every image point on the background grid.

The idea is now, to use a convolution scheme which suites for logarithmic sampled data with $N_l = N_d log(N_t)$. To implement this, special care has to be taken, because the convolution argument f may not be given at intermediate times $t_i - t_j$. The modified convolution for logarithmic data (logconv) may therefore be expressed through

$$\tilde{f}_i = \sum_{j=0}^{i} g_j Q(f(t_i - t_j)) \Delta t_j, \quad i = 1, \ldots, N_l \, , \tag{3.119}$$

where g_j are log-spaced data points and $Q(f(t_i - t_j))$ denotes the interpolation of f to the time $t_i - t_j$. Q may be given by equation (3.105), or, if the function f is piecewise continuous in time, one may apply Q as a cubic spline interpolation (e.g. Press et al. [1995]), too. Note, that the number of multiplications are now significantly reduced by $\left(\frac{N_t}{N_d log(N_t)}\right)^2 \approx 1000$.

Before one applies the logconv to the adjoint and background fields, some further remarks on the peculiarities of the convolution and the adjoint fields. For the hybrid sensitivity calculation, one may like to interpolate the 1D adjoint field instead of the 3D background field

$$S(t_i) = \sum_{j=0}^{i} Q\left(\frac{\partial \mathbf{E}}{\partial t}(t_i - t_j)\right) \mathbf{E}^p(t_j))\Delta t_j, \quad i = 1, \ldots, N_l \tag{3.120}$$

which suited very well because of two reasons: The 1D adjoint fields can be calculated directly using equation (3.102) and they may be evaluated for very early times. Feasibility checks showed, that it is useful to calculate the 1D adjoint fields at least two decades earlier than it is necessary to cover the needed bandwidth of the actual model background response. This artificial extension of the time range does not increase the number of operations needed to carry out the numerical convolution. It just increases the causal precision of the adjoint fields which has to be interpolated.

Assuming, that the adjoint field is not calculated direct but according to a finite difference in time one may write

$$\begin{aligned} S(t_i) &= \sum_{j=0}^{i} \frac{\partial \mathbf{E}}{\partial t}(t_j) Q\left(\mathbf{E}^p(t_i - t_j)\right) \Delta t_j \\ &= \frac{\partial}{\partial t} \sum_{j=0}^{i} \mathbf{E}(t_j) Q\left(\mathbf{E}^p(t_i - t_j)\right) \Delta t_j, \quad i = 1, \ldots, N_l \end{aligned} \tag{3.121}$$

and carry out the time derivative after the convolution, or one may take

$$\begin{aligned} S(t_i) &= \sum_{j=0}^{i} Q\left(\mathbf{E}^p(t_i - t_j)\right) \frac{\Delta \mathbf{E}(t_j)}{\Delta t_j} \Delta t_j \\ &= \sum_{j=0}^{i} Q\left(\mathbf{E}^p(t_i - t_j)\right) \Delta \mathbf{E}(t_j), \quad i = 1, \ldots, N_l \ , \end{aligned} \tag{3.122}$$

which is a particular simplification of the 3D adjoint fields reducing the sensitivity calculation to adjoint field step responses only. However, it can be shown, that equation (3.121) and equation (3.122) are numerically equivalent, but the number of operations needed to carry out the sensitivity calculation are a little bit reduced. Note, that the finite difference operator $\Delta \mathbf{E}(t_j)$ is still to be applied on the adjoint field step response.

Illustrated in figure 3.40 is a qualitative comparison of different sensitivity calculation methods, for a surface near image point. The sensitivity designation is FD: perturbation method, S_1: 1D sensitivities, S_2: hybrid sensitivities (using equation (3.120)), S_3: 3D sensitivities (using equation (3.122)). They are calculated for a single E_x-receiver, thus the units are displayed in V^2/A. The background model consists of a homogeneous half-space ($50\,\Omega m$). The spatial dimension of the parameter is denoted as $40 \times 40 \times 20\,m^3$, where the sensitivities of the adjoint methods are spatial integrated. As you can see from the figure 3.40, the FD method is numerically not very stable, which can be seen by the small oscillations. In fact, the relative difference of the model parameter is already increased to $\Delta = 15\%$.

Focusing on the first time points of the sensitivities in figure 3.40 one can see the effect of the singularity treatment of the different adjoint methods: The 1D sensitivities seem to be continuous for times $t \leq t_{min}$, treating the singular point very well. Comparing the hybrid sensitivities with the full 3D sensitivities reveal the impact of the causal convolution and the treatment of the singular point. Where the hybrid sensitivities are stabilized (in terms of time continuity) by the early times of the 1D adjoint fields, the 3D sensitivities do not attain a continuous level in the limit $t \leq t_{min}$. Yet, the first time point of the 3D sensitivities is still within reasonable relative amplitude.

Overall, one will notice, that the different methods produce sensitivities within good agreement. The amplitudes of the E_x-sensitivities seem to be relatively small, but keep in mind,

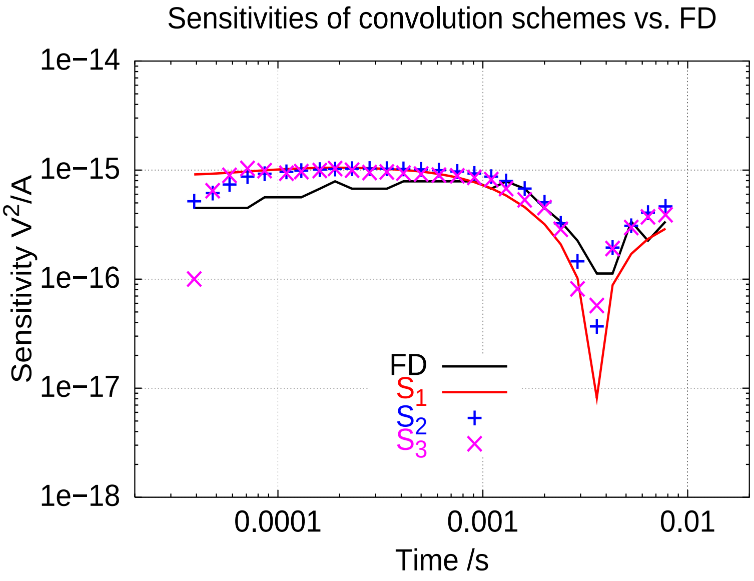


Figure 3.40: *Qualitative comparison of different convolution schemes using the adjoint equation method ($S_{1,2,3}$ with the perturbation method (FD) for a surface near image point (from Martin [2003]). Displayed are sensitivities of an E_x-receiver signal at $\mathbf{r} = (0, 700, 0)m$ due to a parameter change at $\mathbf{r}' = (50, 0, 10)m$ over time in a log-log graph. Further explanations are given in the text.*

that the transient responses are small as well.

3.5.4 Sensitivity comparison of long and short offset TEM

Within a multidimensional TEM inversion program suited for the LOTEM and SHOTEM methods, different kind of receiver/transmitter combinations are possible. For LOTEM, these are mainly horizontal electrical and induced magnetical fields in all 3 space dimensions. Applying the SHOTEM method one would normally measure the vertical induced voltages only. Yet, in section 4.5.3, I demonstrate how the horizontal voltage receivers deliver useful additional information to resolve surface near structures. Therefore, I do now compare different calculation methods of all the mentioned receiver types for image points that are commonly within the depth of investigation of these methods.

Figure 3.41 shows the model and the forward grid for the sensitivity calculation of a classical LOTEM setup in equatorial configuration. It consists of a galvanic coupled dipole transmitter (Tx) of 800 m dipole length into the plane and one receiver location (Rx). Here the E_x-, $\dot{\mathsf{H}}_y$- and $\dot{\mathsf{H}}_z$-transients are calculated for 25 delay times after the current was switched on (step on response) between 5 ms and 0.83 s in log spacing. Exemplary, three voxels with 100 m edge length are embedded in a homogeneous subsurface of 100 Ωm. The mid points of the left and the right voxels are located at depth 750 m which is the appropriate diffusion depth for this setup ($z^* \approx 503\sqrt{0.02\,s \times 100\,\Omega m} \approx 700\,m$). The first is placed under the transmitter, the second in the middle between Tx and Rx, where the last is placed below the receiver site. For the $\dot{\mathsf{H}}_z$-receiver one expects that the sensitivities are very low for this image points. The azimuth dependency of the $\dot{\mathsf{H}}_z$-field of a grounded dipole is proportional $\sin(\phi)$, where ϕ is the azimuth of the dipole axes. Thus, it is discontinuous for the yz-plane along the dipole axes. The voxel in the middle is placed further more to the surface, because the sensitivity in the middle between Tx and Rx for this LOTEM setup has weaker sensitivities [Hördt, 1998]. The response of a small resistivity change within the voxels are calculated with the

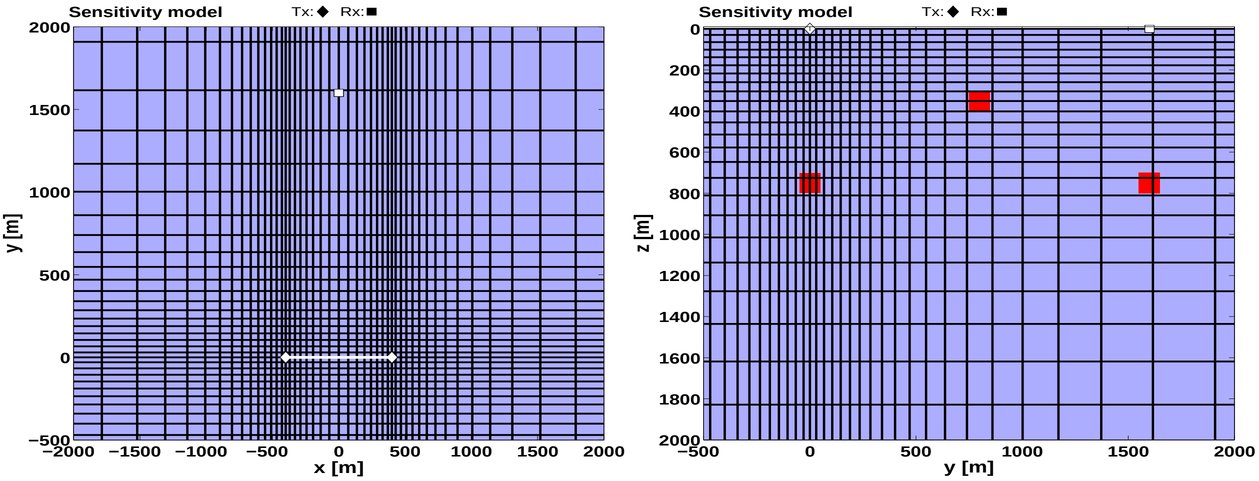


Figure 3.41: *xy-plane (left) and yz slice of the sensitivity model and 3D forward grid for the LOTEM setup. The transmitter is located at the origin (Tx) and has a dipole moment of* 1×800 *Am. The receiver site (Rx) is placed at an offset of* 1.6 *km. In the yz-plane one can see the voxels (red rectangles) embedded in a homogeneous subsurface (light blue color) at three voxels with each (red rectangles).*

FD method (pert) and the 1D, respectively the hybrid adjoint approach. For the FD method I took a moderate relative resistivity change of $\Delta\rho = 0.1\rho$. The results are compiled in figure 3.42.

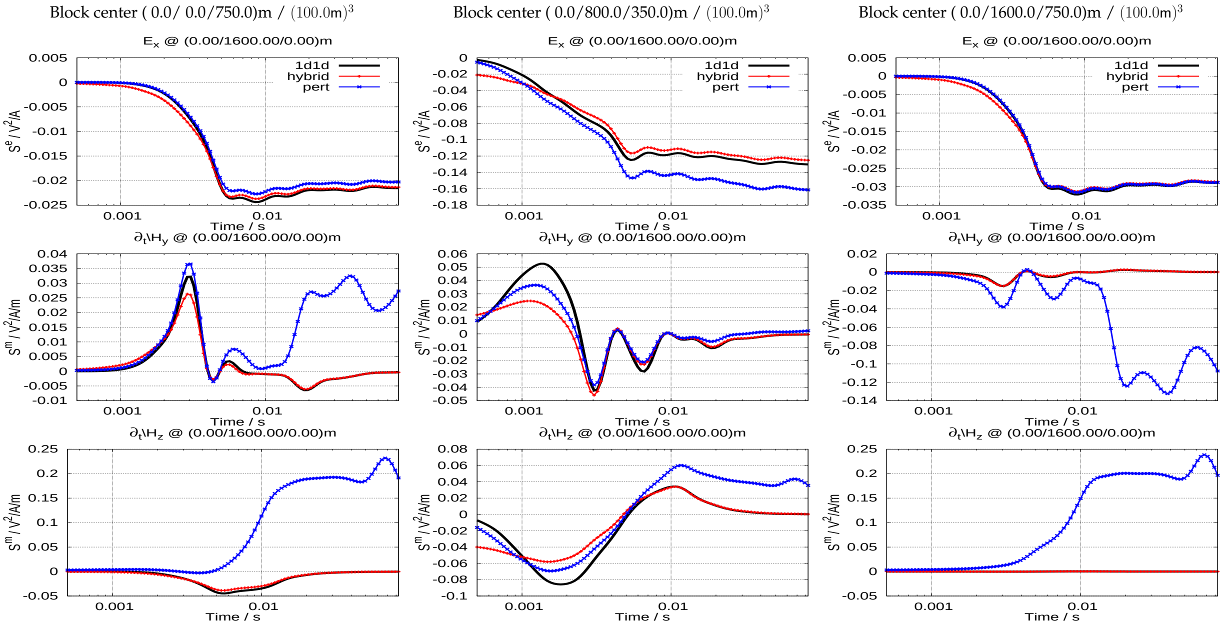


Figure 3.42: *Comparison of LOTEM* $\mathsf{E_x}$*-,* $\dot{\mathsf{H}}_\mathsf{y}$*- and* $\dot{\mathsf{H}}_\mathsf{z}$*-sensitivities using different calculation methods. The sensitivities of* $\mathsf{E_x}$*- (top),* $\dot{\mathsf{H}}_\mathsf{y}$*- (middle) and* $\dot{\mathsf{H}}_\mathsf{z}$*-transient responses (bottom) due to resistivity changes of three different voxels positions are displayed. The geometry of the setup and the spatial dimension of the model cells are given in the title of each plot and in figure 3.41.*

The sensitivities for the non causal $\mathsf{E_x}$-fields are matching very good for every voxel position (top row) and every method. For the shallow voxel we can see some more differences in the amplitude between the adjoint calculated sensitivities and that of the FD method. However, they show the same time behavior. The $\dot{\mathsf{H}}_y$-sensitivities comprise at least of one or two sign reversals (see fig. 3.42). The overall best agreement can be depicted for the shallow image point where the sensitivities show a huge amplitude for early times and two smaller minima with opposite sign for later times. This can be understood directly from the half-space behavior of the $\dot{\mathsf{H}}_y$-response since it shows relocation of the sign reversal over time due to resistivity changes in the subsurface. Negative sensitivities mean, that if the resistivity is increased, the voltage would be amplified and vice versa. It shows, that the sign reversal of the $\dot{\mathsf{H}}_y$-field, which can be observed for the homogeneous half-space, is sensitive to both, conductivity as well as resistivity contrasts. This can not directly be seen from the asymptotic formulas, because it's valid for intermediate times.

The huge discrepancy between the pert-sensitivities and the adjoint sensitivities for the deep image points at later times are due to numerical errors which can occur within the FD method sensitivity calculation. Comparable numerical errors can be seen for the $\dot{\mathsf{H}}_z$-sensitivities as well (see fig. 3.42, bottom row). Since the magnetical voltages do vanish for late times, the sensitivities should be zero, which is shown nicely by the adjoint sensitivities.

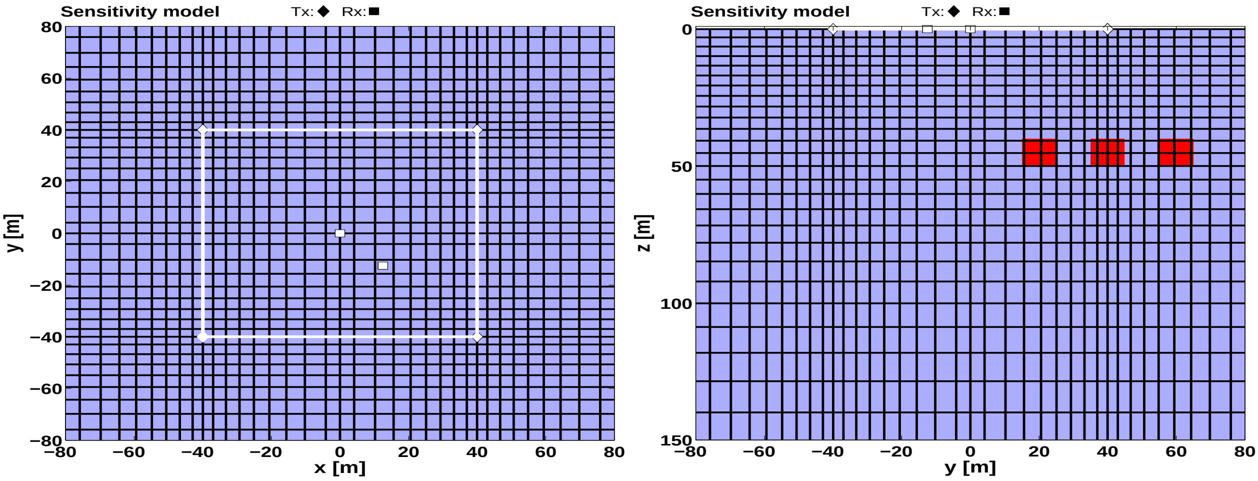


Figure 3.43: *xy-plane (left) and yz slice sensitivity model and 3D forward grid for the SHOTEM setup. The transmitter consists of a square loop centered around the origin (Tx) with* $80\,m$ *edge length. The receiver sites (Rx) are placed at two different positions within the transmitter. Embedded in a homogeneous subsurface (light blue color) are three voxels with* $10\,m$ *edge length each (red rectangles).*

Figure 3.43 shows the conductivity model and the forward grid for the sensitivity calculation of a SHOTEM setup. The setup seems traditional except, that the horizontal magnetical voltages are measured as well. The transmitter (Tx) is a square loop wire matching $80\,m$ from edge to edge. After switching off the current of $1\,A$, the delay times between $1\,\mu s$ to $0.1\,ms$ are calculated at two different receiver locations (Rx). The $\dot{\mathsf{H}}_z$ is located in the center (in loop), whereas a $\dot{\mathsf{H}}_x$- and a $\dot{\mathsf{H}}_y$-receiver are placed offside the center at $x = 12.5\,m = -y$. Exemplary, three voxels are again introduced in a homogeneous subsurface of $100\,\Omega m$. The mid points of the voxels are located at depth $45\,m$ which is roughly the depth of investigation for $30\,\mu s$ delay time.

The response of a small resistivity change within the voxels are calculated with the same methods and conditions as described above. The results are compiled in figure 3.44. The sensitivities show the best agreement for the image point which is located directly below the transmitter edge (see fig. 3.43 and fig. 3.44). Here, according to the "smoke-ring" concept of

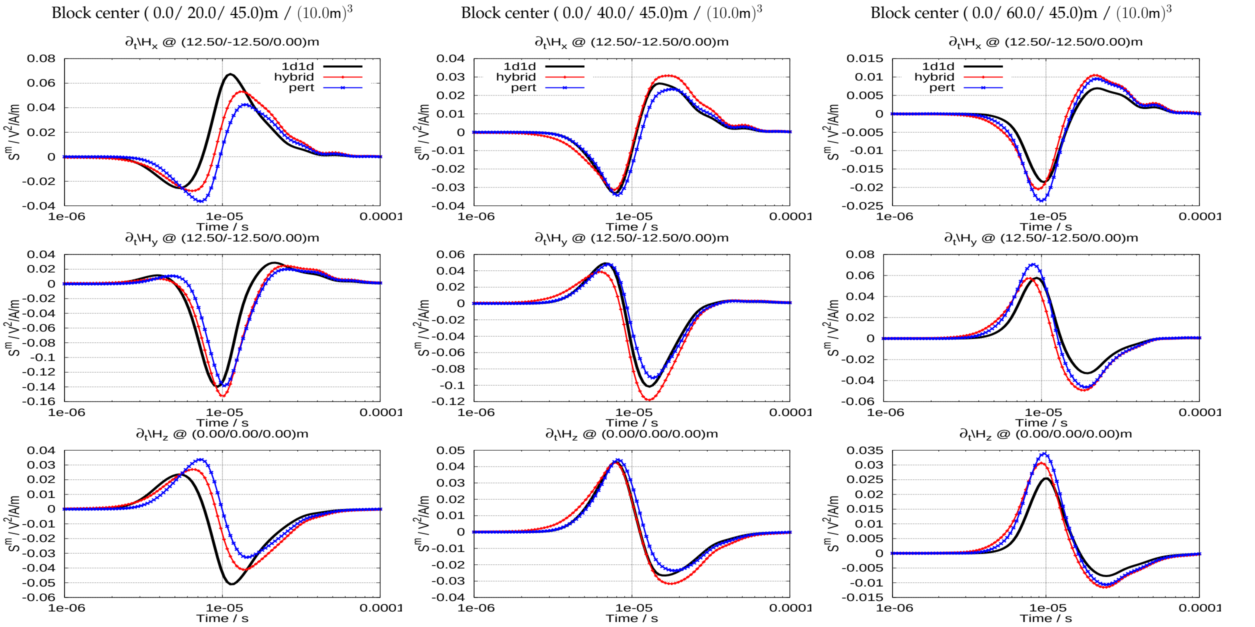


Figure 3.44: SHOTEM sensitivities of $\dot{\mathsf{H}}_{\mathrm{x}}$*-,* $\dot{\mathsf{H}}_{\mathrm{y}}$*- and* $\dot{\mathsf{H}}_{\mathrm{z}}$*-transient soundings over time due to resistivity changes of three different voxels positions. The geometry of the setup and the spatial dimension of the model cells are given in the title of each plot and in figure 3.43.*

Nabighian [1979], the sensitivity values should attain their maximum which should give the best responses for the forward calculation. Another explanation can be found if the forward grid is further investigated. Below the transmitter edges the grid is much finer than for the surrounding points.

This increases the number of sensitivity density points and thus delivers more points to the sensitivity integration. Compared to the other image points left and right, the 1D sensitivities show the biggest differences. Here, the grid is a little coarser compared to the image point in the middle which seems another indicator for the above statement. Another reason may be, that the source approximation is not good enough, since it consists of several superimposed dipoles. Yet, comparing the SHOTEM sensitivity values and the computed sensitivities of the magnetical LOTEM responses (see fig. 3.42), show that the different adjoint approaches produces sensitivities within very good agreement. The numerical instability of the FD method, which was observed for the LOTEM magnetical voltages may be removed by adjusting the grid appropriately or taking higher values for $\Delta\rho$. For a further investigation of TEM sensitivity calculation, the reader is referred to the works of Petrat [1996]; Hördt [1998]; Schneider [2000]; Martin & Hördt [2001]; Martin [2003]; Martin et al. [2003].

3.5.5 Summary

With sensitivities we can form a good approximation of the Hessian matrix to enhance convergence during the inverse process. Further, the sensitivity matrix can be utilized to discuss model ambiguities and resolution properties of our inverse solutions.

A fast convolution scheme, after Hördt [1998], is established for the calculation of sensitivities direct in time domain. The adjoint problem as well as the background electrical fields can either be calculated with fast 1D approximations or by means of the *SLDMem3t.* For homogeneous models, the 1D adjoint approach yields a convenient estimation of the sensitivity distribution within the model space. It may therefore be used either for a homogeneous starting model, or if the parameterization of the model is to be checked. If the model gets more structured, the hybrid adjoint approach is promoted because the 1D adjoint fields can be calculated in no time compared to a full 3D simulation. Especially with the proposed parallelization of the problem, a significant speedup is achieved by utilizing a simple divide and conquer technique. Comparisons of the adjoint sensitivities with the FD solution show quite good agreement, especially for surface near regions of the model. Although not shown here, the sensitivities calculated with full 3D solutions provide the same good agreement to the FD solutions [Hördt, 1998; Martin & Hördt, 2001; Martin, 2003].

CHAPTER 4

Multidimensional TEM inversion

In the current development state of the inversion program, the treatment of three different model parameterizations are implemented. The small-scale 3D inversion, which is optimized for low parameterized models, is already discussed in detail by Commer [2003] and therefore not treated here. In this section I focus on the parts of the program which are developed throughout this work and suited for large scale 2D and 3D model parameterizations. Therefore, I give the formulation of the implemented equations and show applications of 2D and 3D TEM inversions which where done for synthetic data. Although special treatment to the synthetic data is carried out by adding suitably noise to the data, the application to synthetic data may just be seen as a "proof of concept" for the proposed TEM inversion scheme. The application to real field data from a 2D LOTEM survey demonstrates how the proposed inverse solution performs for a real world application, which may be regarded as a first "hard test". Moreover, it can be shown that the 2D a posteriori model derived by the inversion of LOTEM data can constrain the predictions of a MT survey.

4.0.6 Preliminary notes

In order to compare synthetic transient electromagnetic fields to measured data, which may be collected using different devices and receiver settings, I introduce the following normalizations to them:

- Transmitter current and
- receiver moment.

Thus, the displayed electrical field data are always given in V/m and the induced voltages are given in V/m^2. Although one may argue that they should be given in $V/A/m^2$, I do not want to cause to much confusion about the units. Please keep in mind, that the transmitter moment, which is either the length of a prolonged dipole, or the area of a (square) loop source is not included in those normalizations.

4.1 General strategy for multidimensional TEM inversion

To formulate the implemented inversion scheme, I use the common vector notation of the data $\mathbf{d} \in \mathrm{I\!R}^N$, the model $\mathbf{m} \in \mathrm{I\!R}^M_{>0}$ and the model response $\mathbf{f}(\mathbf{m}) : \mathrm{I\!R}^M_{>0} \rightarrow \mathrm{I\!R}^N$, as already introduced in chapter 2. Within the current development state of the inversion program, $\mathbf{f}(\mathbf{m})$ is always solved by means of the SLDM, as described in detail in the previous chapter. Because of this, the field values are calculated on a staggered grid (see section 3.3). Thus, to account for realistic receiver positions within the inversion program, the field values have to be interpolated. This is achieved by using the linear eight-point interpolation operator Q as given in detail in section 3.5.3.

The data error is assumed to be uncorrelated and thus, the least squares criterion delivers the best linear unbiased estimator of the a posteriori model. Because of the non-linearity of the problem, the minimization of the data misfit alone leads to ill-posedness of the inverse solution if the normal equations are solved. Thus, I follow a Tikhonov regularization for the optimization approach to achieve a better conditioning of the normal equations and inverse solutions.

According to section 2.2 (eq. (2.31)) and 2.3 (eq. (2.49)), the regularized cost functional is given by

$$\Phi = \left\| \mathbf{C}_d^{-\frac{1}{2}} \left(\mathbf{d} - \mathbf{f} \right) \right\|_2^2 + \lambda \left\| \mathbf{C}(\mathbf{m} - \hat{\mathbf{m}}) \right\|_2^2 \rightarrow \min \ . \tag{4.1}$$

which minimizes the misfit and reduces the structure between the current model $\mathbf{m}$ and the prior model $\hat{\mathbf{m}}$. This is minimized iteratively by solving the normal equations

$$\left(\mathbf{S}_n^{\mathrm{T}} \mathbf{C}_d^{-1} \mathbf{S}_n + \lambda \mathbf{C}^{\mathrm{T}} \mathbf{C} \right) \delta \mathbf{m}_n = \mathbf{S}_n^{\mathrm{T}} \mathbf{C}_d^{-1} \mathbf{y}_n - \lambda_n \mathbf{C}^{\mathrm{T}} \mathbf{C} \Delta \hat{\mathbf{m}}_n \ , \tag{4.2}$$

for a global regularization, or,

$$\left(\mathbf{S}_n^{\mathrm{T}} \mathbf{C}_d^{-1} \mathbf{S}_n + \lambda \mathbf{C}^{\mathrm{T}} \mathbf{C} \right) \delta \mathbf{m}_n = \mathbf{S}_n^{\mathrm{T}} \mathbf{C}_d^{-1} \mathbf{y}_n \ , \tag{4.3}$$

for a local regularization. For convenience, $\mathbf{C}^{\mathrm{T}} \mathbf{C} = \mathbf{I}$ in the latter case.

By using a more simple notation, the above equations are compiled by solving

$$\boldsymbol{\mathcal{H}}_n \delta \mathbf{m}_n = -\mathbf{g}_n \tag{4.4}$$

for every iteration step n to get the model update $\mathbf{m}_{n+1} = \mathbf{m}_n + \delta \mathbf{m}_n$. Thus, the normal equations are M linear simultaneous equations in the unknown increments, $\delta \mathbf{m}_n$.

The regularization term can be defined by the user in the input file of the program (see appendix D). It may be choosed to consist of the first or second order smoothness matrix $\mathbf{C} = \left\{ \mathbf{C}^{1st}, \mathbf{C}^{2nd} \right\}$ leading to a global regularization, or, if the identity matrix is selected, as local regularization.

In the examples given below, $\hat{\mathbf{m}}$ is selected to be non existent. Nevertheless, it can be defined by the user to have a fixed prior value. This may be updated in the inversion as well, leading to a local regularization, or not.

Non uniqueness

Because the data we collect is contaminated with noise, the solution of the inverse process is not unique and the model parameters are not independent of each other. Speaking of (T)EM inversion, the parameter we like to know, either resistivity or conductivity, is only resolved as product of the parameter with its volume. Due to the volumetric parameterization of the subsurface, the 3D inversion suffers the less from this problem and the 1D inversion the most. As shown by Weidelt [1972] and Parker [1977a] for 2D parameterizations, a unlimited number of measurements at the earth surface, with unlimited precision would be needed to resolve every parameter uniquely and independent from each other. Additionally, the non uniqueness is artificially increased by regularization, leading to regularization dependent model structures which can only be resolved by using different types of regularization methods.

Oldenburg & Li [1999] defining the depth-of-investigation index as

$$\mathrm{DOI} = \frac{\mathbf{m}_*^a - \mathbf{m}_*^b}{\hat{\mathbf{m}}^b - \hat{\mathbf{m}}^a} \,, \tag{4.5}$$

with the posteriori models $\mathbf{m}_*^{a,b}$ from two different starting models $\hat{\mathbf{m}}^{a,b}$. The DOI index reveals, to what degree the model parameters are determined by the different starting models.

Similar to this I propose that if one compares inversion results yielded with different regularizations, e.g. $\mathbf{C}^{1st}$ with $\mathbf{C}^{2nd}$, regularization invariant features in the inverse solution can be detected. This can be done either by visual comparison, or via a simple difference.

4.1.1 Lagrange parameter determination

The regularization or Lagrangian parameter λ arises for explicit regularization schemes. It weights the model constraints against the data misfit.

Small values of λ will produce structured models with huge parameter contrasts but a small data misfit. In contrast large λ values result in smoother models but poor data fit.

Generally, a trade-off between data fit and model constraints has to be sought. For a resolution analysis it is necessary to think about how the Lagrange parameter can be optimized in a way not to loose valuable information, because the regularization parameter significantly influences the inverse solution.

I give some general considerations on how the Lagrange parameter can be chosen during the inverse process followed by the implemented method for the TEM inversion. For an overview the reader is referred to Vogel [2002] and Kilmer & O'Leary [2001].

Discrepancy principle

This approach, as proposed by Constable et al. [1987], is simply to choose the value of λ which minimizes the data misfit. Thus, the smoothness is not explicitly minimized but an auxiliary regularization. The approach can be implemented by univariate or bivariate line-search algorithms as already discussed in section 2.1.1. Constable et al. [1987] promote the use of the golden section search which is the most convenient and therefore chosen in the implementation as well.

L-curve criterion

A little more computational effort is involved if the L-curve criterion is used. It is achieved by plotting the data functional Φ^d against the model functional Φ^m for a varying λ on log-log axis. The graph tends to have a characteristic "L" shape and is therefore often just called the L-curve [Hansen, 1992]. One very simple suggestion for the optimum λ is the point, where the curvature of the L-curve attains it's maximum. The curvature c of $\Phi^d(\lambda) = \psi_d$ and $\Phi^m(\lambda) = \psi_m$ can be calculated by [Hansen, 1992]

$$c(\psi_d, \psi_m) = \frac{\ddot{\psi}_m \dot{\psi}_d - \ddot{\psi}_d \dot{\psi}_m}{\left(\dot{\psi}_m^2 + \dot{\psi}_d^2\right)^{\frac{3}{2}}} \,. \tag{4.6}$$

Here the single and double dots represent the first and second order derivatives of the parametric functions with respect to λ.

Generalized cross validation (GCV)

The idea of the generalized cross validation is based on the "leaving out one" lemma [Whaba, 1990]. It is observed, that if any row of the unregularized normal equations is left out, the regularized solution will fit the data as well, independent of λ. After Whaba [1990], the GCV function reads:

$$c(\lambda) = \frac{\|\boldsymbol{\mathcal{H}}_n \delta \mathbf{m}_n(\lambda) + \mathbf{g}_n\|_2^2}{\mathrm{trace}\left(\mathbf{I} - \boldsymbol{\mathcal{H}}_n^{\mathrm{T}} \boldsymbol{\mathcal{H}}_n^{-1}(\lambda)\right)} \,. \tag{4.7}$$

Where the minimum point of the GCV function determines the optimal regularization.

4.1.2 Implementation for TEM inversion

For the computation of the main diagonal of $\boldsymbol{\mathcal{H}}_n \boldsymbol{\mathcal{H}}_n^{-1}(\lambda)$, a proper matrix factorization is required, which is indeed computational demanding within a large scale inversion scheme. Besides, as was investigated by Farquharson & Oldenburg [2004], the GCV yield similar results to the L-curve criterion. Thus it is not considered any further within this work. For the L-curve criteria many additional evaluations of $\mathbf{f}(\mathbf{m}_*)$ are needed to calculate ψ_d. In order to economize the time consuming forward calculations for all models $\mathbf{m}_n + \delta\mathbf{m}_n(\lambda)$, the linearization

$$\mathbf{f}(\mathbf{m}_n + \delta\mathbf{m}_n(\lambda)) \approx \mathbf{f}(\mathbf{m}_n) + \mathbf{S}_n \delta\mathbf{m}_n(\lambda) \tag{4.8}$$

is used in order to give an approximation to the L-curve.

Similar to Mitsuhata et al. [2002] it was found by Scholl [2005], that for TEM the linearization errors are too big. This could be verified during a simple 1D inversion of synthetic central loop data. The approximated L-curve never showed the "optimal" λ value properly and the curvature lead to more structured models as desired. If the "optimal" value for the Lagrange parameter would be determined from the linearized curve, ψ_d in fact would be close to the maximal value.

During some initial testing minimizing the whole cost functional ($\Phi^d + \lambda\Phi^m$), it turned out, that the models tend to present far less structure than if the minimization is following the discrepancy principle. Therefore neither the L-curve nor the GCV was considered any further during this work, but the discrepancy principle seems most suitable.

Changing the regularization parameter during inversion

A well-discussed question in non-linear problems is that of changing the regularization parameter during inversion. Different from global schemes, the use of local constraints leads to an independent linear sub problem in every iteration. For each of them an appropriate λ can be determined by the methods described above. Many authors using local regularization schemes as Marquardt [1963]; Loke & Barker [1996]; Kemna [2000] discuss the use of decreasing λ beginning from a large starting value down to a minimum value. Also, Farquharson & Oldenburg [2004] applying a global regularization scheme using a cooling type schedule of decreasing λ.

To prevent overshooting in the early iterations, a line search parameter can be applied to ensure convergence. Generally, the use of larger λ yields similar, but smoothed, structures with less magnitude, which represents an easy-to-control alternative to the line search procedure. Yet, the resolution analysis (cf. section 2.4) shows that the a posteriori model is strongly influenced by the final λ. Therefore it has to be chosen properly.

If the regularization is incorporated into a non-linear conjugated gradient method the regularization parameter should be fixed during inversion. This is due to to keep the gradients properly conjugated. Any change of λ during the process would reset the conjugated search directions setting it to steepest descent. However, Commer & Newman [2008] use a NLCG scheme with changing the Lagrangian during inversion at intermediate iteration steps leading to a restarted NLCG scheme. They report quite good convergence of the method.

Thus, I followed this approach within the large scale 3D inversion.It is implemented like a additional line search for λ leading to a moderate version of a bivariate line search. If the decrease in the misfit is not fulfilled, the line search is revoked with another λ restarting the conjugation scheme. Since the computation of many λ's is prohibitive, a small fixed number as dropout criterion is implemented. This turned out to give good results, but not in any case.

Generally, the underlying minimization procedure treats λ as parameter, which depends on the number of data and their errors, the model discretization and the used constraints. All of these aspects are determined before the inversion, which pleads for constant regularization even for global regularization techniques.

4.1.3 χ^2 and RMS

For uncorrelated data error, the data covariance matrix is a diagonal matrix ($\mathbf{C}_d = \text{diag}\{\delta\mathbf{d}\}$) and the data cost functional is further simplified to

$$\Phi^d = \sum_{i=1}^{N} \left(\frac{d_i - f_i(\mathbf{m})}{\delta d_i} \right)^2 . \tag{4.9}$$

Because Φ^d is quadratic and the target value should be $\Phi^d_* = N$, it is more likely to introduce a normalized misfit functional

$$\chi^2 = \frac{\Phi^d}{N} \tag{4.10}$$

or its square root.

Stacking

A common technique within TEM measurements is to collect several single measurements at a single receiver site to enhance the signal-to-noise ratio (stacking). The number of stacks vary between some hundreds to some thousands of stacks, depending on the used devices.

The additional positive side-effect of stacking is, that the error of the measurement can be calculated via the standard deviation. Likewise, for a stack of K measurements, the error of the single data point is given through

$$\delta d_n = \sqrt{\frac{1}{K} \sum_{i=1}^{K} \left(d_i - \overline{d}_n \right)^2} \tag{4.11}$$

with the mean value

$$\overline{d}_n = \frac{1}{K} \sum_{i=1}^{K} d_i . \tag{4.12}$$

For simplicity, one takes $\overline{d}$ for the true data point d, but keep in mind, that it presents the average over K single measurements.

If every data point has a constant relative error and follows a normal distribution, say every data point has a unique $p\%$ standart deviation, it becomes a norm which is generally considered as the root-mean-square (RMS) misfit. Although it is a χ^2-error in this sense, it is used here to distinguish the χ^2 of the real error analysis with the RMS error which may be used for synthetic data. It represents just the relative standard deviation of two data sets scaled by $\frac{100}{p}$.

$$\begin{aligned} \chi(p) &= \sqrt{\frac{1}{N} \sum_{N}^{i=1} \left(\frac{d_i - f_i(\mathbf{m})}{p\, d_i} 100 \right)^2} \\ &= \frac{100}{p} \sqrt{\frac{1}{N} \sum_{N}^{i=1} \left(\frac{d_i - f_i(\mathbf{m})}{d_i} \right)^2} \end{aligned} \tag{4.13}$$

The latter is commonly assumed for synthetic data sets or for real data if there is no "real" standard deviation available. In fact, it can be shown [Press et al., 1995] that synthetic data is contaminated with systematical errors which are not normal distributed, but they are generally omitted or assumed by a simple standard deviation $1 < p < 10$.

However, if

$$\chi^2 = 1 \Leftrightarrow \delta d_i = d_i - f_i(\mathbf{m}_*) \ , \tag{4.14}$$

which means, that every data point is fitted by the model response of $\mathbf{m}_*$ within the uncertainty of the measurement. From this follows that $\mathbf{m}_*$ is a stationary point of the cost functional which is the ultimate criterion for the optimization process.

4.1.4 Data transformations

Data transformations are often made for better treatment of non-linear problems. It is generally considered as to make the problem more suited for the linearized inversion techniques. Thus, data transformations may be seen as preconditioning.

A often used transformation is the apparent resistivity transform. It is widely used for EM and DC data and may be used for TEM data as well. The use of early and late time apparent resistivity transformation may yield a quick overview for real data. Yet, the lack of the ”all time” apparent resistivity transform [Strack, 1992; Karlik, 1995] limits the use of this transformation. Therefore the developed inversion scheme is always applied to the transient voltages/electric fields and not to apparent resistivities.

Transformation of the model response

Diffusive EM fields have widely different amplitudes at different times and receiver locations. To reduce the dynamic range of the data, transformations are preferable in order to equalize the influence of each datum. Otherwise, the error functional of equation (4.4) may be dominated by high amplitude data points, thus deteriorating the convergence in an inversion [Meju, 1994]. A simple logarithmic transformation [Vozoff & Jupp, 1975] can be used if all data points possess the same sign. For details on logarithmic parameters see Tarantola [2005].

Depending on the type of the field component, sign reversals within TEM data can occur over a 1D earth, as for e.g. the $\dot{\mathsf{H}}_\mathrm{y}$ component. However, TEM measurements over 3D structures often involve sign reversals over the measurement time range, thus requiring to distinguish between positive and negative data. Different methods exist to take both large amplitude variations and different signs into account. For example, Ward & Hohmann [1988] suppose a logarithmic transformation with a linear scale straddling amplitudes near zero and a discrimination between positive and negative logarithms of data values. This transformation is also called stitched logarithm and given by [Ward & Hohmann, 1988]:

$$\text{s-log}(x,a) = \begin{cases} \ln(x) & x > a \\ x & |x| \le a \\ -\ln(-x) & x < -a \end{cases} \tag{4.15}$$

Here, $a > 0$ is a scaling factor which should correspond to a assumed noise level. Ward & Hohmann [1988] suppose to take a like 2 times the noise value of the receiver which seems reasonable. Another transformation scheme realized in *sinv* is described by Hördt [1992] and based upon the Area-Sinus-Hyperbolicus function:

$$\operatorname{asinh}(x, a) = \ln\left(\frac{x}{a} + \sqrt{\left(\frac{x}{a}\right)^2 + 1}\right) . \tag{4.16}$$

The function has a logarithmic behavior for arguments $|x| > a$ and a nearly linear for arguments with $|x| \leq a$. This transformation has proven to be suitable for the 1D inversion of LOTEM data containing sign reversals. Because the normalized data values are generally below 1, the scaling factor $a > 0$, corresponding to the previous deliberations, should be applied as well.

To compare the behavior of the two transformation function, they are displayed in figure 4.1. The advantage of the asinh-transformation over the stitched logarithm is that it is not only a bijective function, but also continuous and differentiable over the whole domain, which can be seen clearly from the figure (black curve). The drawback of the stitched logarithm on the other hand is very notably, too. For very small values of the noise factor ($a \approx 0$), the s-log$(x, 0)$ reveals a huge discontinuity leading to a very complicated shape of the transformed data (magenta curve). Even if one takes higher values of the scaling factor (e.g. $a = 1$), the s-log$(x, 1)$ curve is showing discontinuous behavior for $a < |x - \epsilon|$. Therefore it is more likely to implement a modified form of stitched logarithm

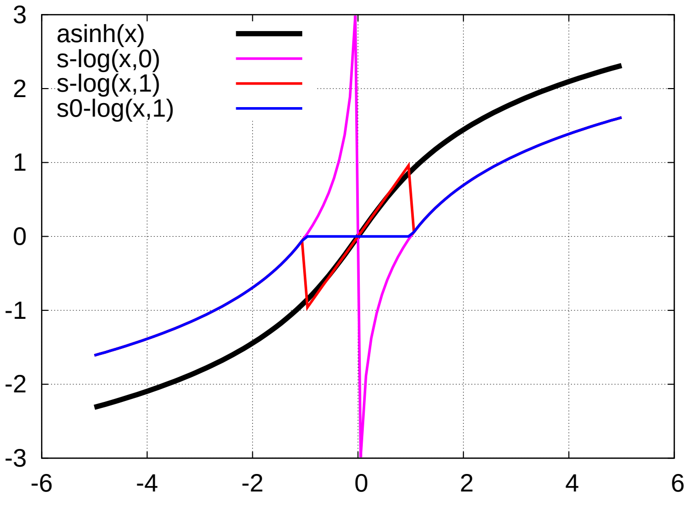


Figure 4.1: *Comparison of the value range of four different data transformations over their domain.*

$$\text{s0-log}(x, a) = \begin{cases} \ln(x) & x > a \\ 0 & |x| \leq a \\ -\ln(-x) & x < -a \end{cases} , \tag{4.17}$$

into the inversion code, if one uses the stitched logarithm at all. Applying the modified form of the stitched logarithm transform has the desired impact on the data values (see fig. 4.1, blue curve). It treats the data values below the threshold in a manner, that it gets piecewise continuous at least.

After various tests I run with the different types of data transformations, I figured out, that the inversion yield nearly the same results with all kind of transformations. Yet, the convergence process is the fastest if one applies the asinh- or s0-log transformation. Therefore I suggest to use the asinh-transform for the model responses within any inversion process. It is also used for the synthetic examples that are given later.

Note, that the data transformation affects the data errors and thus the data covariance matrix as well.

Model parameter transformation

The model parameter m_j may represent the resistivity ρ_j or the conductivity σ_j . By the inversion of resistivity or conductivity, resistive or conductive bodies are enhanced, respectively. This can be of advantage, but also leads to artifacts. To enforce positiveness and in order to get rid of the choice, often m_j is chosen to represent the logarithm of the resistivity or conductivity.

$$\mathrm{tr}(m_j) = \ln(\rho_j) = \ln\left(\frac{1}{\sigma_j}\right) = -\ln(\sigma_j) \tag{4.18}$$

If a priori information $\hat{m}_j$ is incorporated into the inversion process, the model update selection can be further constraint by using a logarithmic barrier technique, as described by Newman & Alumbaugh [1997]; Li & Oldenburgh [2003]:

$$\mathrm{tr}(m_j) = \ln(m_j - \hat{m}_j) \; . \tag{4.19}$$

The incorporation of the a priori information has the positive side effect to ensure that $m_j > \hat{m}_j > 0$. This technique can also be used to limit m_j to an upper bound m_j^u by setting

$$\mathrm{tr}(m_j) = \ln(m_j^u - m_j) \; . \tag{4.20}$$

Both limits can be combined using the model parameter

$$\mathrm{tr}(m_j) = \ln\left(\frac{m_j - \hat{m}_j}{m_j^u - m_j}\right) \; . \tag{4.21}$$

The accompanying update step is formulated as

$$m_j^{n+1} = \frac{m_j^u(m_j^n - \hat{m}_j)\mathrm{exp}(\delta m_j^n) + \hat{m}_j(m_j^u - m_j^n)}{(m_j^n - \hat{m}_j)\mathrm{exp}(\delta m_j^n) + m_j^u - \hat{m}_j} \; . \tag{4.22}$$

Because, the model parameter transformation turned out to produce the best results, it is used consequently, which is in agreement to many other authors using logarithmic model parameter transformations with upper and lower barriers like Li & Oldenburgh [2003]; Günther [2004]; Commer & Newman [2008]. For the synthetic examples, the lower and upper boundaries are not used, but they proofed useful for the inversion of the DESERT field data shown in the next chapter.

Sensitivity transformation

It needs to be taken into account that both data and model parameter transformations carry over to the calculation of the sensitivity matrix as well. Applying the chain rule to equation (2.36) with the transformed data leads to

$$S_{ij}^{tr} = \frac{\partial(\mathrm{tr}(f_i))}{\partial(\mathrm{tr}(m_j))} = \frac{\partial f_i}{\partial m_j}\frac{\mathrm{tr}'(f_i)}{\mathrm{tr}'(m_j)} = S_{ij}^{lin}\frac{\mathrm{tr}'(f_i)}{\mathrm{tr}'(m_j)} \ , \tag{4.23}$$

where tr' denotes the derivative of the transformations. For e.g. with $\mathrm{tr}(\mathbf{f}) = \mathrm{asinh}(\mathbf{f})$ and $\mathrm{tr}(\mathbf{m}) = \ln(\mathbf{m})$, the derivatives are given by

$$\begin{aligned} \mathrm{tr}'(f_i) &= \frac{1}{\sqrt{f_i}} \\ \mathrm{tr}'(m_j) &= \frac{1}{\rho_j} = -\frac{1}{\sigma_j} \ . \end{aligned} \tag{4.24}$$

The transformed sensitivities are the linear sensitivities scaled by the resistivity value or the negative reciprocal conductivity on the model cell. For logarithmic transformed model response data, the derivative would lead to a further division by the forward data, the Area-Sinus-Hyperbolicus transform scales the sensitivities by the square root of the model response. Thus, the transformation leads to relative changes and makes the sensitivities unitless.

If the sensitivities are calculated via the adjoint Green functions approach and if the model vector consists of resistivities, one has to apply further transformations. The adjoint sensitivity equations for the electrical field (eq. (3.93)), the magnetic field (eq. (3.98)) and the voltages (eq. (3.100)) can be subsumed into

$$S_{ij}^{adj} = \frac{\partial \mathbf{f}^{\mathsf{R}}}{\partial \sigma_j} = \int_{V_j}\int_0^t \boldsymbol{\mathcal{G}}^{\mathsf{R}}(\mathbf{r}', t|\mathbf{r}, t')\mathbf{e}^b(\mathbf{r}', t')\, \mathrm{d}t'\, \mathrm{d}\mathbf{r}' \ , \tag{4.25}$$

where $\boldsymbol{\mathcal{G}}^{\mathsf{R}}$ denotes the adjoint Green functions of the of the electrical, the magnetical or the voltage receiver placed at $\mathbf{r}$, respectively. V_j denotes the space volume of the parameter m_j. Taking the adjoint sensitivity calculation into account, equation (4.23) may be expressed by

$$S_{ij}^{tr} = -S_{ij}^{adj}\sigma_j \mathrm{tr}'(f_i) \ . \tag{4.26}$$

If further constraints on the model, like a priori information or a upper boundary is used, this affects the sensitivities as well. Taking the a priori model as lower boundary and the upper boundaries into account, the sensitivity is further scaled by the reciprocal ratio of the model transforms:

$$\hat{S}_{ij}^{tr} = \frac{m_j^u - m_j}{m_j - \hat{m}_j} S_{ij}^{lin}\mathrm{tr}'(f_i) \tag{4.27}$$

4.2 Noise generation for synthetic data

A common procedure for testing an inversion program with synthetic data sets is, to add noise to the data of the forward model. The noise commonly consists of random data, with zero mean and the standard deviation of p % means that the data point is deviated like

$$d_i = d_i \left(1 + \hat{r}_n \frac{p}{100}\right) \ , \tag{4.28}$$

where r_n is a normal distributed random number.

Generally, a N_c-Bit A/D converter would roughly have resolution from $\frac{A_0}{2^{N_c-1}}$ to $a\frac{A_0}{2^{N_c-2}}$, where a would be the amplification and one Bit would be left for the sign. Of course, the given estimations are very ideal and the true resolution depends on the used device and the ambient noise. Most devices use a 24-16-Bit A/D converter. Sometimes oversampling is used to enhance resolution of high frequencies leading to a true resolution of 24-Bits, but for low frequencies there are still 16-Bits.

For data which comprise of a huge dynamic range, e.g. $\dot{\mathsf{H}}_z$-fields, it can be observed that the relative error do increase as the amplitude decrease. This is because the huge dynamic is very challenging for the A/D converter of the measuring device and depends on the relation of amplitude between the first and the last time point.

To illustrate this, figure 4.2 shows a measured $\dot{\mathsf{H}}_z$-transient which comprises of a sign reversals at late time due to a high noise level. Note, that the receiver data is normalized to the receiver moment, thus the amplitudes are quite low. Figure 4.3 shows the comparison of two different types of noise generation modes on a central loop $\dot{\mathsf{H}}_z$-response of a homogeneous halfspace. For the generation of the synthetic data, the 1D inversion program *emuplus* is used. The left figure displays the treatment of the common procedure of noise adding. The right figure shows if one takes the dynamic range of the transient into account.

The idea is now, that the measurements do not have a relative variance, but a constant absolute noise floor which is depending on the relation to the maximum amplitude and has to be simulated appropriately. This is achieved by applying the "noia" (noise absolute) function of *emuplus*, which assumes a constant noise level for the whole measurement. The artificial noia function treats 30% of the assumed noise level (d_{nlev}) as absolute standart deviation ($p = 0.3d_{nlev}$). The noise generation with the fixed value is then applied to all data points of one receiver like

$$d_i^{noia} = d_i + 0.3\hat{r}_n d_{nlev} \ . \tag{4.29}$$

In fact, the whole idea is not new and similar considerations are given by Hördt [1992] for e.g., where the noise level is assumed to have 0.1% of the maximal amplitude. Since this seems to be a convenient value it is taken as default setting for "noia" as well, but it can be chosen otherwise.

Comparing figure 4.3 (right) with the measured data (fig. 4.2) shows that the noise of the synthetic data suited much more to the real noise. In some cases, the "noia"-noise treats the transients a little to heavy, especially if the assumed noise level is overestimated. Note, that the χ-misfit of the "noia"-transient is increased to a higher level (4.3, right), which means that the expected value for synthetic $\dot{\mathsf{H}}$-data must not be 1, but may be sometimes larger.

4.3 2D TEM inversion

If the model parameterization is infinite in one dimension (e.g. the x-direction), it is referred to as 2D model. For an inversion scheme this may be advantageous, because it limits the

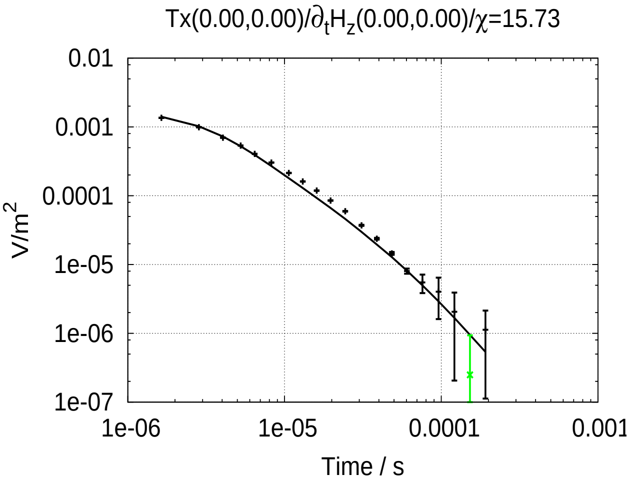


Figure 4.2: *Example of a measured $\dot{\mathsf{H}}_z$-transient (dotted) and model fit (solid) over time. The Tx coordinates refer to the central point of a $80x80m^2$ square loop whereas the coordinates of the receiver are relative to the Tx central point. The last but one data point displayed in green is indicating a sign reversal.*

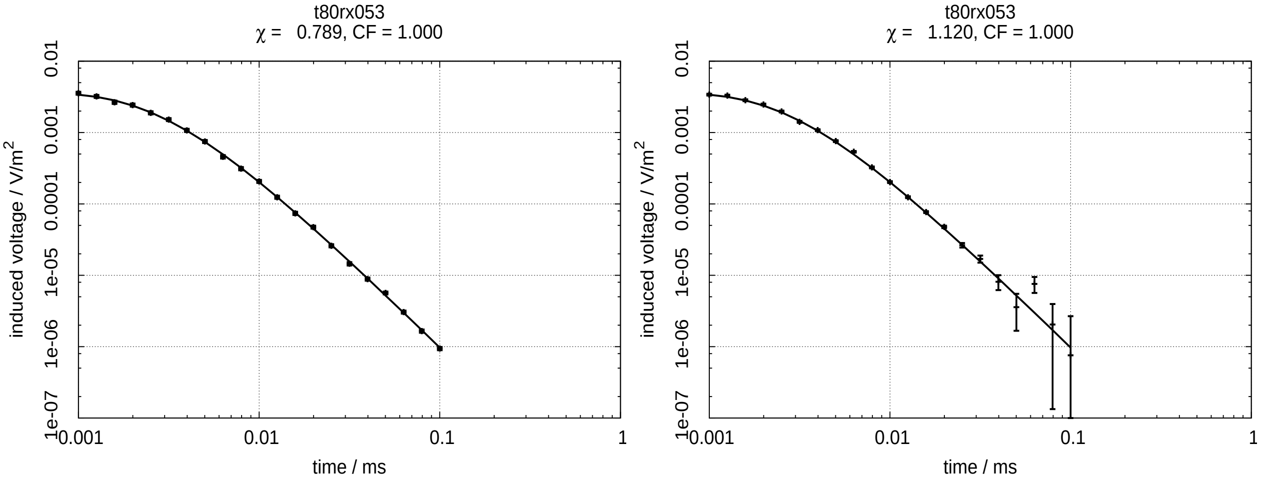


Figure 4.3: *Comparison of two different noise generation switches of emuplus. Left side shows a classical 5% random noise with zero mean added to the data. To the right one can see the application of the "noia" function, as described below, which suitably simulates the noise of a real transient, as shown in fig. 4.2.*

degree of freedom and thus reduces the number of model unknowns significantly. The most essential advantages are then:

1. The solution of the normal equations (4.4) can be calculated using a direct solver.
2. Enough data can be provided to solve an overdetermined system.

4.3.1 Implementations

Within the current version of the program, the solution of the normal equation is achieved by means of the Cholesky factorization. To increase stability a further damping factor was introduced leading to a mixture of local and a global regularization. The damping factor affects the left hand side of the normal equations only, adjusting $\boldsymbol{\mathcal{H}}$ to become

$$\boldsymbol{\mathcal{H}}_n = \mathbf{S}_n^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S}_n + \gamma_n\mathbf{I} + \lambda_n\mathbf{C}^{\mathrm{T}}\mathbf{C}\ . \tag{4.30}$$

The damping factor γ and the constraint parameter λ can be kept fixed during inversion or may be chosen by the program using the discrepancy principle. The additional damping term is rarely used, and in the examples given below it is always set to zero.

However, in some early stages of the program $\gamma > 0$ was sometimes chosen to dampen over-accentuated model cells which may occur for e.g. below the transmitter or receiver, leading to shallow artifacts in the a posteriori model. Those overvalued image points may be flattened either by a singularity removal technique which can be applied within the sensitivity integration, or by increasing the integration space of the sensitivities. The integration space of the sensitivities may be a crucial point, because the model extents infinite in one dimension and to reproduce the sensitivities with the adjoint method well, one has to integrate over a long volume. Another way of removing those overweighted points may be achieved by adjusting the model parameterization and dividing the over-accentuated model cells into smaller pieces. Since the sensitivities can be accessed directly, the appropriate parameterization for a model may be found by looking at the coverages.This feature is the most apparent advantage of the inversion code because it reduces the effort to find a good parameterization significantly.

4.3.2 2D inversion of synthetic LOTEM data

In the following, I show results of 2D inversions that where carried out with synthetic LOTEM data. The results obtained demonstrate how the proposed inversion algorithm works for 2D parameterizations and may be regarded as proof of concept. Moreover, I show how different receiver combinations allow for different model resolutions.

2D model and parameterization

With regard to the 2D MT inversion results of Ritter et al. [2003] (see section 4.4, fig. 4.14 - 4.18), a simplified 2D model of the subsurface conductivity distribution of the Arava Fault was created. Figure 4.4 (left) shows the LOTEM setup and a cross section of the model (right). The model extent is bigger to each direction to account for the lateral borders (see fig. 4.5) and only the part of interest of $4\,km$ depth and $12\,km$ horizontal length is shown here. I emphasize on the term ”simplified” because the true conductivity structure is far more complicated than reflected in this model.

The main features of the model are a high conductive zone to the left and resistive parts to the right. The high conductive zone consists of two bodies. The first reaches from $y = -\infty$ to $y = 0\,m$ in lateral directions and from $z = 1\,km$ to ∞ in depth and is the dominating conductive feature to the left. To the right ($y > 0\,m$), you can see two separated resistive blocks from $z = 500\,m$ to $z = 1.5\,km$ and a third resistive body at the bottom of the cross section. Also you can see a fourth resistive body with $2\,km$ vertical and $1.5\,km$ lateral extent. Surface near, you can see two good conducting features. The first is a weaker conductive zone with a lateral extend of $2.5\,km$ and $300\,m$ thickness. The second is a small

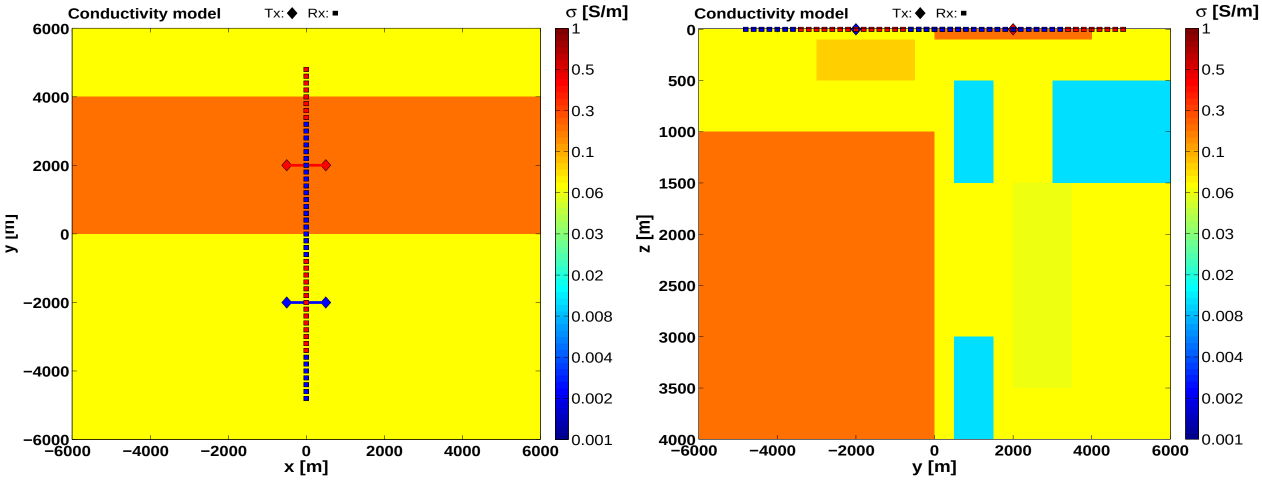


Figure 4.4: *Plane view (left) and cross section (right) of the 2D model with LOTEM setup. The LOTEM setup consists of two galvanic coupled dipole transmitters (Tx) with corresponding receiver sites (Rx). The color scale is chosen in the casual way of EM results. Red and orange part of the model show conducting regions, whereas blue and light blue/green parts show more resistive zones.*

conductive overburden, overlaying the resistive part of the model. It reaches from the surface to $z = 100\,m$ and has a limited lateral expansion.

The LOTEM setup is leaned on to the DESERT field campaign. In opposite to the real field data set, the setup is chosen a little smaller to reduce numerical effort. Therefore, the setup consists of only two transmitters (Tx), which are shown in the plane view of figure 4.4. Each transmitter comprises of 28 receiver sites (Rx) placed at each side of the x-oriented galvanic dipole of $1\,km$ length. At the receiver sites, which are modeled along the equatorial axis of the transmitter, the E_x-, $\dot{\mathsf{H}}_y$- and $\dot{\mathsf{H}}_z$- fields are calculated for delay times of $1\,ms$ to $1\,s$ as step on responses. Thus, the total number of receiver sites is $2 \times 3 \times 28 = 168$. The synthetic data are generated with the program *FDTD3D*, presented by Commer & Newman [2004], to deliver forward data independent from the used *SLDMem3t*. Figure 4.5 displays the parameterization of the model that was used for the inverse solution. Here a bigger part of the model is shown in order to exemplify some inherent features of the inverse solution. The total number of model cells are given as $M = 70 \times 35 = 2450$ in y and z direction. Note, that the vertical discretization of the model cells is increasing with depth, which is very important for the inverse solution. The lateral extent of the cells is $\approx 200\,m$ throughout the main parts of the cross section and increases at the borders. Thus, the model is discretized relatively fine, in terms of the scale, for the parts of main interest, and is broadened for parts of less importance. This kind of parameterization is only possible because of the already mentioned material averaging and the re-weighting scheme for the sensitivity density and allows for a grid independent model parameterization.

Treatment of the noise

Since a spread of receiver data is measured, it is convenient to use a constant noise floor for the whole data domain. Since the E_x-fields generally comprise a smaller dynamic than the $\dot{\mathsf{H}}$-fields, I decided to take two different noise level for the voltage and the electrical receiver components. Also, it could be observed from the DESERT field data, that the electric-field step-on responses comprise of higher errors for early times (see fig. 4.20). To take this into

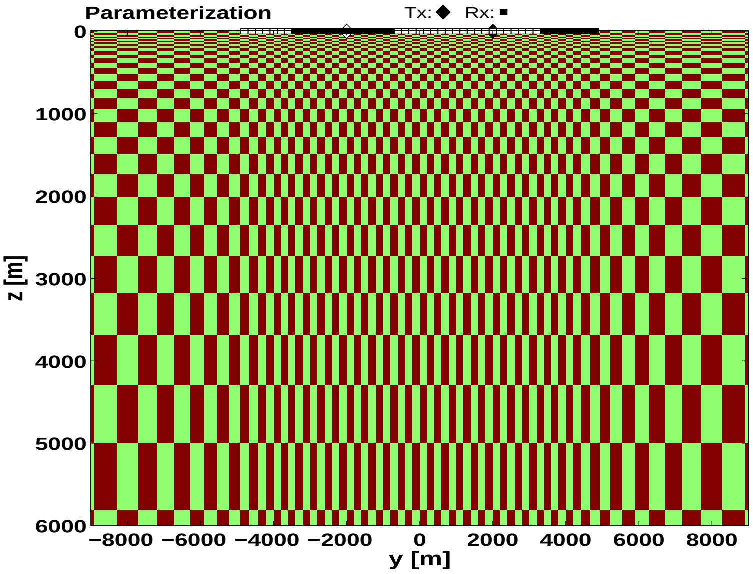


Figure 4.5: *Cross section of the model parameterization for the inversion of synthetic LOTEM data. The colors show the discretization of the model into 2D cells. The thickness of the cells increases with depth.*

account, I assumed that the error for the first time points of the E_x-field is 5 times larger than the actual, gradually decreasing with time.

4.3.3 Comparison of 2D inversions using different receiver combinations

Compared to the MT, the transient EM fields of a grounded dipole inhere more complicated structure, especially for lateral conductivity variations. Thus, they can not be decomposed into TE- and TM-mode, as commonly done for MT measurements, but have both modes. In order to test the resolution capabilities, the data of the synthetic model (fig. 4.4) is inverted using different receiver components, (e.g. only E_x-receivers) and two different smoothness constraints. After Ward & Hohmann [1988], one would expect from the theory, that every receiver component should be sensitive to both, resistive and conductive parts of the model. In particular, they may be sensitive to different features within the 2D conductivity variations and reveal different results, but a joint interpretation of all receiver information should reduce the non uniqueness to a great extend.

Table 4.1 shows a compilation of the statistics obtained by the 2D inversions. In the left row, the used receiver components are listed (Data set). The second row shows what type of smoothness constraint (**C**) was used throughout the inversion. In the third row, the expected misfit values are given (Expect.). They are obtained by running a forward solution of the true model with *SLDMem3t*. As you can see from table 4.1, the expected values for χ are substantially larger than 1. This is caused by the ”noia” function of *emuplus*, which was used for the noise generation. Yet, there are some systematical deviations between the forward solutions of *SLDMem3t* and *FDTD3D* carried out with the original model, too, but they are neglected here.

Summarizing the listed results, it seems that the 2D inversion of the magnetical receivers,

Table 4.1: *Inversion statistics for 2D models using different receiver data sets.*

Data set	**C**	Expect. χ	Start χ	End χ	Iterations
All	$1st$	2.16	30.7	3.65	4
	$2nd$			3.00	4
E_x	$1st$	2.47	37.8	3.80	2
	$2nd$			3.90	2
$\dot{\mathsf{H}}_\mathrm{y}$	$1st$	1.37	10.9	0.88	7
	$2nd$			1.45	4
$\dot{\mathsf{H}}_\mathrm{z}$	$1st$	2.49	35.9	2.83	4
	$2nd$			1.73	7
$\dot{\mathsf{H}}_\mathrm{y}$ & $\dot{\mathsf{H}}_\mathrm{z}$	$1st$	1.99	26.4	2.12	4
	$2nd$			1.55	5

either with the components alone or all together, reach a very satisfying fit of approximately the expected value. The inverse solutions, which where obtained by using the whole data set, fits the data not ideal in terms of χ, but satisfactory. The cause is found by looking at the inversion results of the E_x-fields and the corresponding data fit (see table 4.1). The fit of the inversion using the E_x-fields alone are not bad, but the worst compared to the other results. For your information, a homogeneous halfspace of $\sigma = 0.4\,S/m$ was taken as starting model for the inverse solutions.

A collection of some selected data of each transmitter location is shown in figure 4.6. From the figure it can be observed, that the fit to the original data is already reached after 4 iterations, which is very satisfying. Especially the data of the second transmitter $(\mathrm{Tx}(0, 2000)\,m)$, which are displayed to the right side of the figure ,show some very distinct features between the starting model and the inversion result. This shows, that the inversion process is independent from the starting model. The fit of the sign reversal, which can be identified by the ”notch” within the $\dot{\mathsf{H}}_\mathrm{z}$-response (bottom right), is quite excellent. The heavy noise, that was generated with the noia-function, can also be identified by the strong oscillations in the horizontal and vertical induced voltages at late times, respectively.

Figure 4.7 exemplarily displays the evolution of χ, the stepsize $||\delta\mathbf{m}||$ (top) and a comparison for the error weighted data residual

$$\hat{y}(\mathbf{m})_i = \frac{d_i - f(\mathbf{m})_i}{\delta d_i}, \quad i = 1, \ldots, N \tag{4.31}$$

of the starting model and the one obtained after 4 iterations (bottom). The plots show the convergence process and the misfit decrease for the inverse solution using all receiver data and first order smoothness constraints. The inversion process of the other inversions show nearly the same features and is therefore omitted.

As can be seen from the top figure 4.7, the inverse process shows strong convergence at the first steps and reaches the final misfit relatively fast ($\approx 3 - 4$ iterations). The misfit of the last iteration is acceptable. From the evolution of $||\delta\mathbf{m}||$ it can be observed, that the most adjustments to the homogeneous starting model where done through the first 2 iterations. For the last two iterations, although the misfit decreases, the model alteration decreases, too, which is a confident result. As you can see from figure 4.7 (bottom), the error weighted residual is decreased successful to a level of satisfying acceptance. Although,

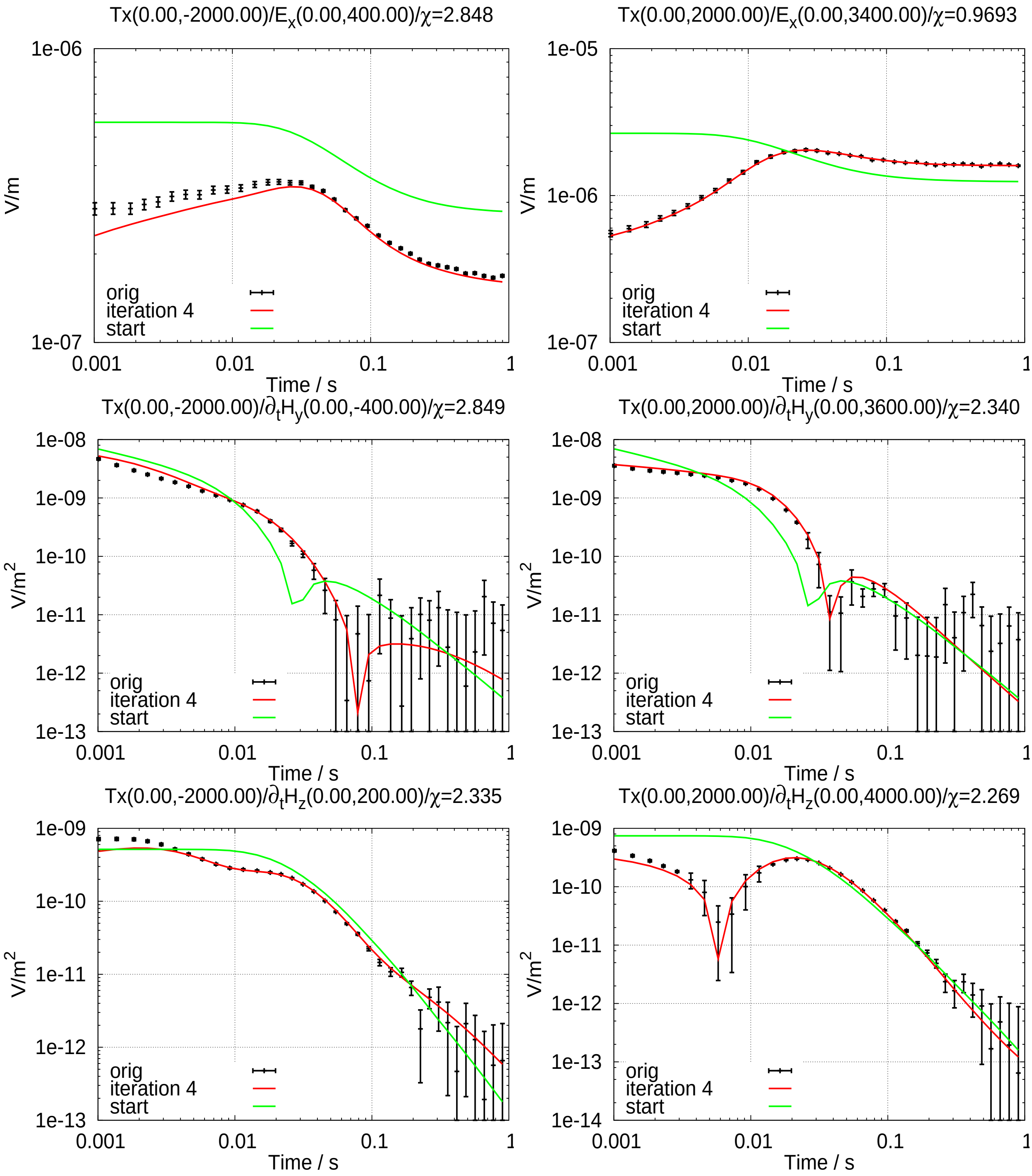


Figure 4.6: *Comparison of selected data for the starting model and the inverse solution with the original data at different receiver stations. The displayed transients for* E_x- *(top)* $\dot{\mathsf{H}}_y$- *(middle) and* $\dot{\mathsf{H}}_z$*-responses (bottom) are selected for each transmitter. The precises coordinates and receiver type can be depicted from the title of each figure as well as the fit of the a posteriori model.*

there are still some features which can be observed in both plots (the blue and orange/yellow regions), the whole residual level is decreased. The ordering of the receiver number is, that the electrical receivers are listed first (from 1 to 56) and the magnetical receivers are listed last. The blue and orange/yellow spots are thus identified as belonging to a bad fit of some magnetical receivers. It is caused by the "heavy" noia-noise as could already observed within the magnetical field data, displayed in figure 4.6.

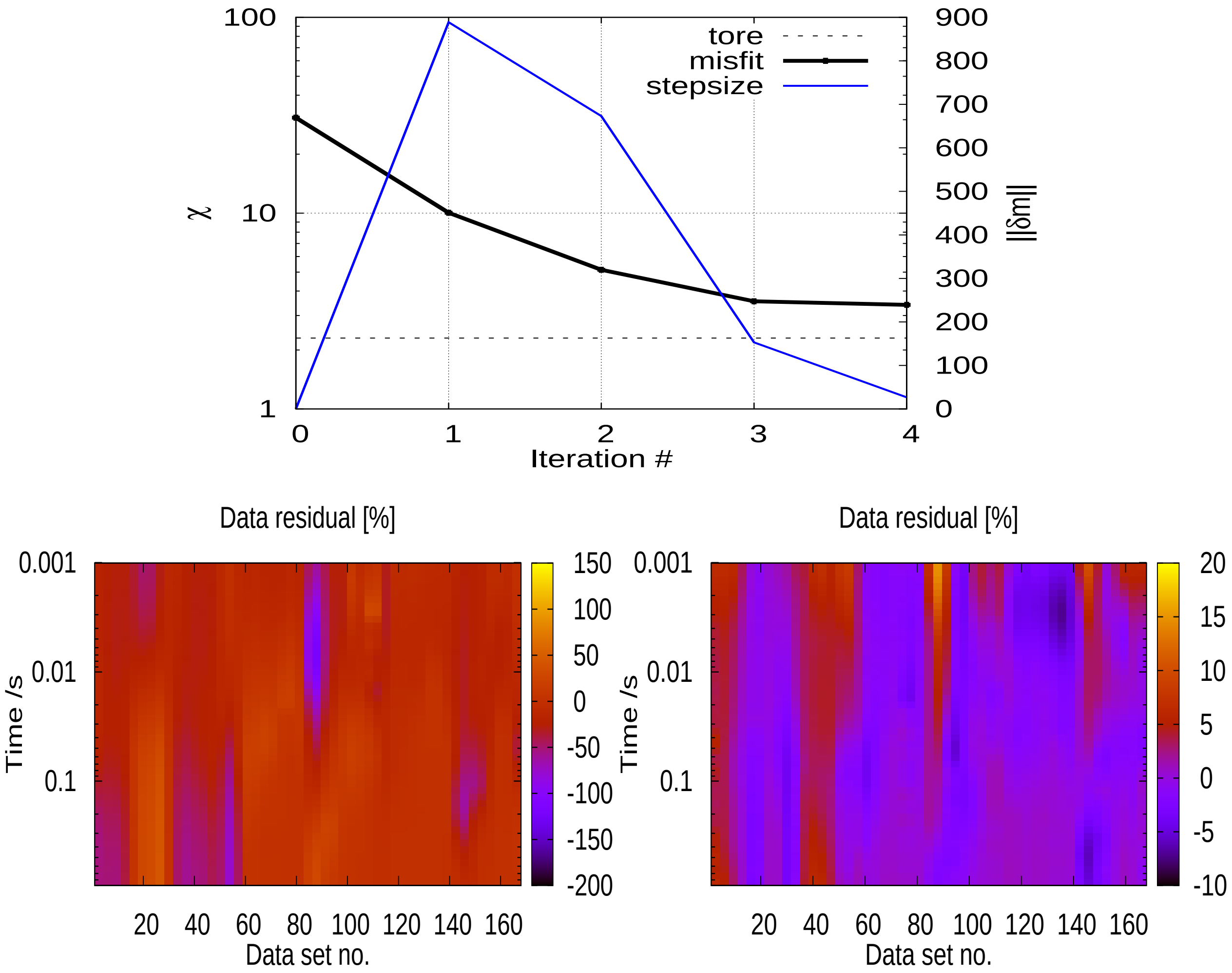


Figure 4.7: *Compilation of 2D inversion statistics and data fit for all receiver data. The evolution of χ (misfit) for each iteration is given at the top. Here, the value of tore ("to reach") gives the expected χ for the data (see table 4.1). The step size $\|\delta\mathbf{m}\|$ specifies how much the model is adjusted in each iteration step. The data fit of each data point of the corresponding receiver number is shown as contour plots at the bottom row. It gives a comparison of the error weighted residual $\hat{y}_i$ for the homogeneous starting model (bottom left) and the a posteriori model (bottom right). Please note, that the linear scale of the two plots is different.*

Comparison of the 2D LOTEM inversion results

A comparison of the 2D a posteriori models[1] resp. the inversion results are now given. Inversion solutions, obtained by using first and second order smoothness constrains are shown in figures 4.8-4.12.

As one can see from figure 4.8, the conductivity structures of the original model are well resolved up to the depth of 2.5 until 3 km. Especially the good conducting features, namely the big conductive zone to the bottom left as well as the two conductive regions near the surface are well resolved. The high resistive parts in the model, and especially the spatial separation are relatively good resolved. The upper edge of the big conductive anomaly, shows different depths in both inversion results. I would estimate, that the upper border can be

[1] In fact "a posteriori" model is commonly used only if the real model is not known and therefore may be unusual in this context.

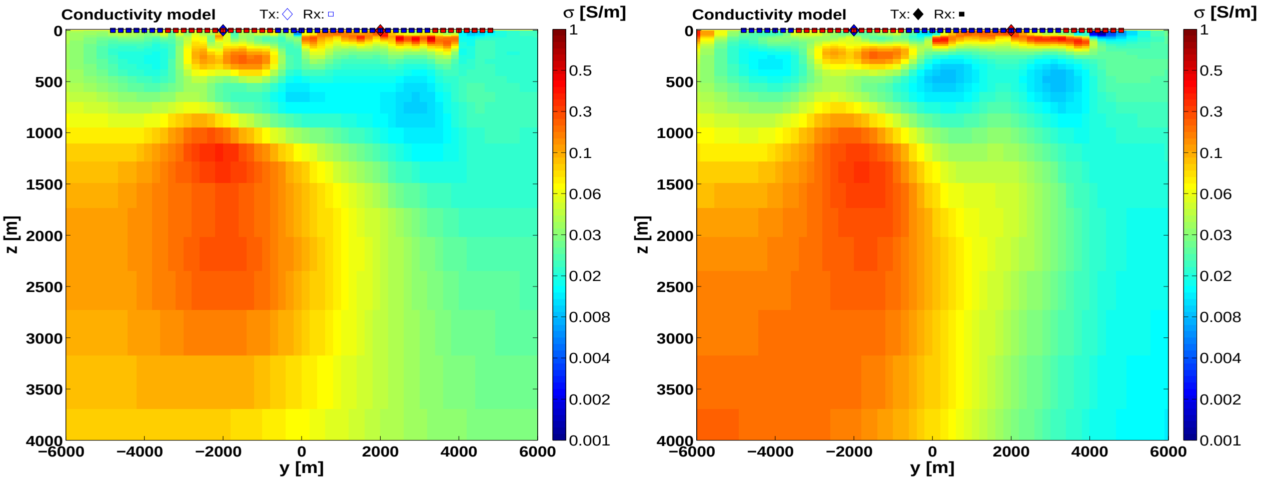


Figure 4.8: *Comparison of 2D inversion models after 4 iterations using every data set and different smoothing constraints. The inversion result with* $\mathbf{C} = \mathbf{C}^{1st}$ *is shown to the left and the result obtained by using* $\mathbf{C}^{2nd}$ *is shown to the right.*

resolved to a relative precision of $\approx \pm 100\,m$, which is not bad in terms of the scale. On the opposite, the vertical edge of the good conductive anomaly overshoots the true location by about $1.6\,km$ in both results, which means that the true lateral border is not well reproduced by the soundings. The resistive parts beneath $2\,km$ can not be reproduced at all, which is not surprisingly. For $t = 1\,s$, the voltages are already very weak, and the integrated volume which is penetrated, very huge. However, the depth of $3\,km$ may be regarded as very good in terms of resolution.

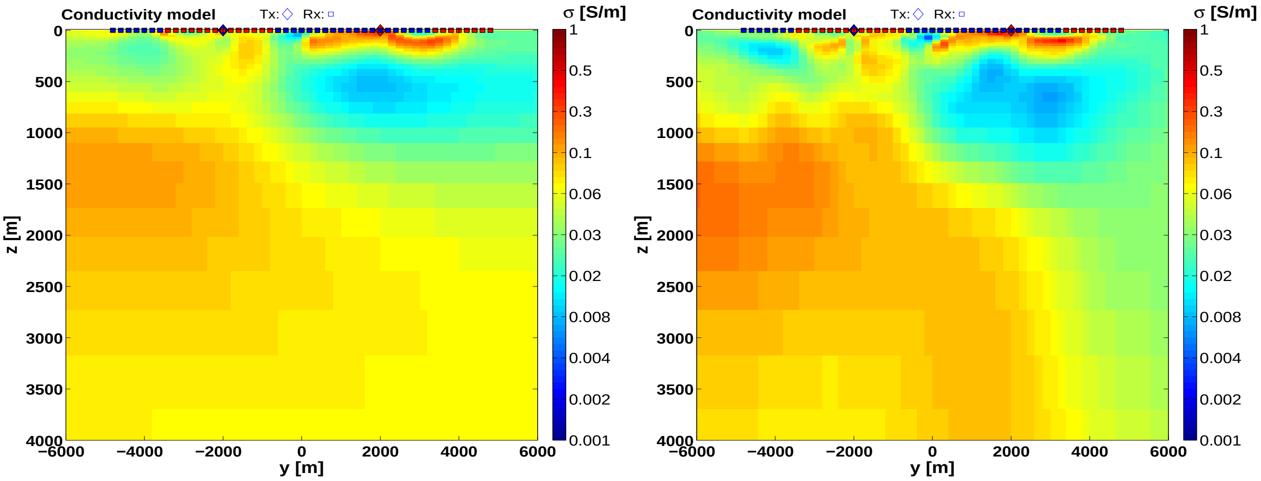


Figure 4.9: *Cross section of the 2D inversion results using different smoothing constraints and* E_x*-field transient data only. After three iterations, the inversion for both regularization types showed no more progress. The inversion result with* $\mathbf{C} = \mathbf{C}^{1st}$ *is shown to the left and the result obtained by using* $\mathbf{C}^{2nd}$ *is shown to the right.*

From the inversion results, obtained by using the E_x-receivers only (fig. 4.9), one can identify roughly the same features as the one found in the inversion results for all receivers (fig. 4.8). Yet, it shows a weakness regarding the resolution of the lateral conductivity distribution. Such, the border of the good conducting anomaly is smeared out to a great extend of 3 to $5\,km$. The conducting overburden is captured well, whereas the resistive zones in the upper part are just signified. Moreover, the separation of the two resistive zones is not as obvious as

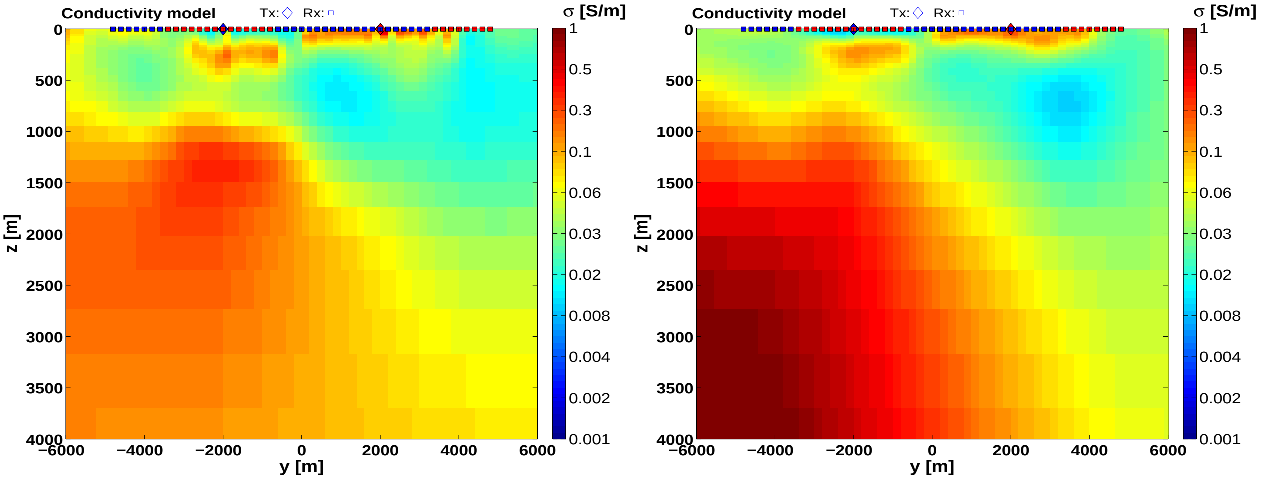


Figure 4.10: *Cross section of the 2D inversion results using different smoothing constraints and only $\dot{H}_y$-field transient data.*

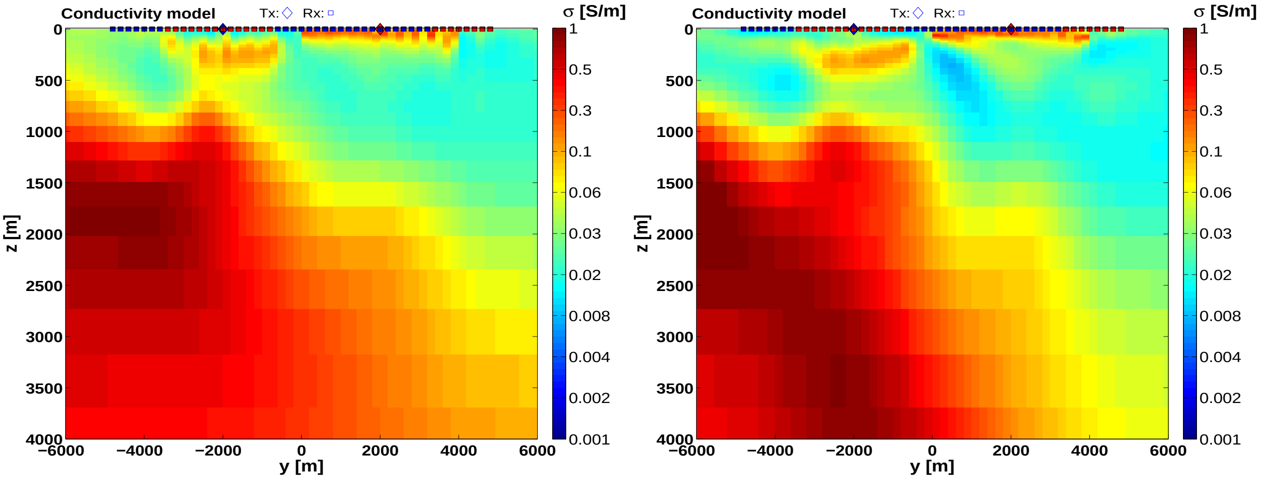


Figure 4.11: *Cross section of the 2D inversion results using different smoothing constraints and $\dot{H}_z$-field transient data.*

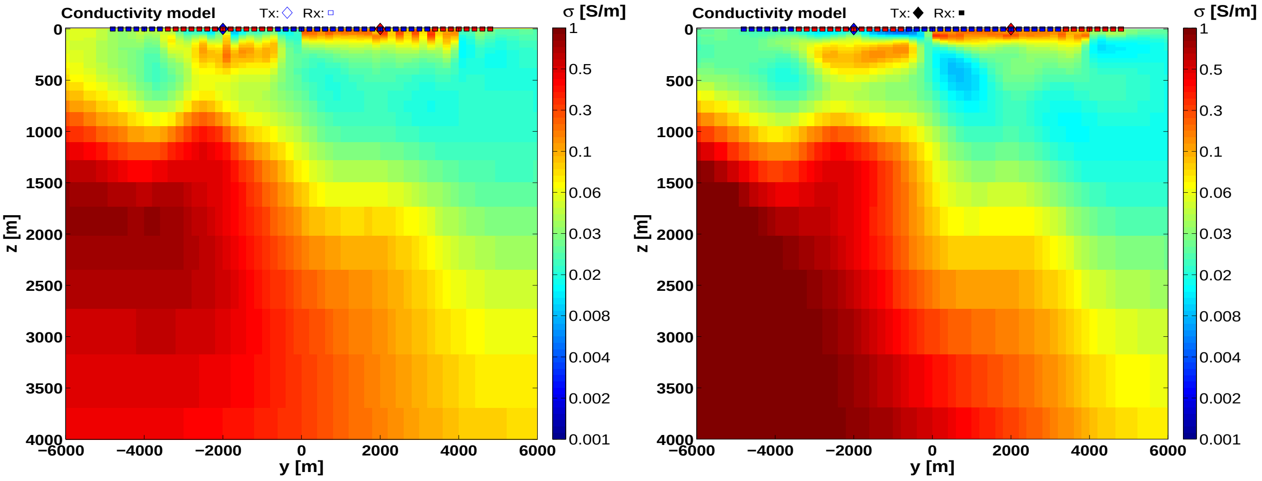


Figure 4.12: *Cross section of the 2D inversion results using different smoothing constraints and all $\dot{H}$-field transient data.*

in the inversion results obtained using all receiver information. The surface near conductive part, overlaying the big conductive anomaly, is mistreated as a vertical conducting feature.

Comparing the two inversion results (fig. 4.9 left and right), it can be stated, that the resolution beneath 1.5 km is lost if one restricts the interpretation to the E_x-receivers only.

The $\dot{\mathsf{H}}_y$-inversion result obtained using the first, as well as the a posteriori model obtained using the second order smoothness constraints, reveal the same shape of the big conductive block on the left side of the model (figure 4.10). Both regularization types tend to draw the rectangular shape of the body over the $y = 0\,m$ marker to the right. This is similar to the E_x-inversion results (see figure 4.9). Yet, the two conductive regions near the surface, can be resolved very well. As expected, one can say, that the resolution of the conductive structures is better than for the resistive parts, if one inverts with the voltage information only. Nevertheless, it seems that the resistive parts are detected, although, relatively weak.

Comparing the $\dot{\mathsf{H}}_y$-inversion results with the one obtained by using only the vertical voltage information (see figure 4.11) reveals, that the $\dot{\mathsf{H}}_y$-fields may do slightly better with the resistive zones in the model. The inversion result obtained by using every voltage information seems to be the same as for the vertical voltage, except for very few differences. Yet, as can be depicted from figure 4.10 to 4.12, the lateral extend of the conductive anomaly is not good resolved if one uses only voltage information.

4.3.4 Discussion

Although the fit is not ideal, in the sense of the expected χ-value, it is confident that the developed inverse solution decreases the overall misfit in a desired manner. Additionally, it has to be mentioned, that the decrease in misfit in terms of number of iterations, is very fast. The final fit of the predicted data is reached within 4 iterations. As shown in figure 4.6, the fit of the predicted data to the synthetic forward data seems excellent. Especially the sign reversal, which occurred in the $\dot{\mathsf{H}}_z$-field data, could be fitted very good. Comparing the predicted data with the response of the starting model demonstrates, that the proposed inverse solution is relatively independent from the starting model, which is very satisfying.

The features which occur in the contour plots of the data residuals 4.7, are strikingly. The cause could be identified as belonging to the random noise with constant noise level, which was used in order to simulate real world data. Yet, the absolute noise level may lead to biased information in the data, but seems to yield realistic distortions of the signals.

The comparison of inverse solutions, yielded with two different smoothness constraints, reveal regularization invariant features in the a posteriori solution. Therefore this simple comparison may be treated as similar approach to the DOI-index proposed by Oldenburg & Li [1999]. Yet, the visual comparison has not the same validity, but may be used as good guess to the resolution capabilities of the inverse solution.

The expected resolution capabilities of the different receiver components, namely that they are all sensitive to different parts of the model, could be validated within the 2D inversion results given above. A simple decomposition of the receiver data into TE- and TM- mode is thus not possible, which agrees very well to the theory. 2D inversions that where done using single receiver components only, lead to strong non uniqueness within the inverse solution. On the opposite, the joint inversion of all receiver components reduces the non uniqueness and improves the a posteriori solution drastically.

4.4 2D Inversion results of the LOTEM DESERT survey

The inversion results of the synthetic LOTEM data demonstrate, that the proposed 2D inverse solution performs well in terms of data fit. Although some effort was spend to simulate realistic measurement data, they where still simulated. The next step in a "proof of concept" would naturally be the application of the inverse solution to real field data. The data of interest was collected during a field campaign, within the DESERT project [DESERT Working Group, 2004].

4.4.1 Target area: A part of the Dead Sea Transform

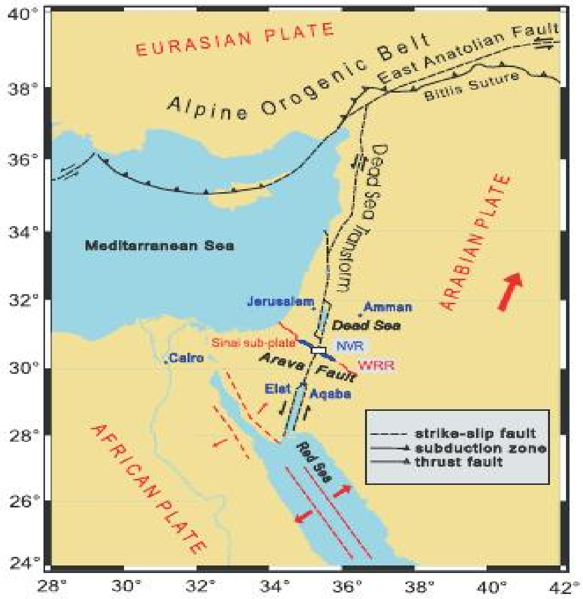


Figure 4.13: Plate tectonic setting of the Dead Sea Rift. The measuring area (white rectangle) is located in the Arava valley, which extends for 200 km between the Dead Sea and the Red Sea (Gulf of Aqaba/Elat). Also shown are locations of the seismic wide angle reflection/refraction (WRR) and near vertical reflection (NVR) experiments (after Ritter et al. [2003]).

The Dead Sea Transform (DST) fault is a major strike slip fault in the continental lithosphere. It forms the boundary between the African and Arabian plates (see figure 4.13) and joins the divergent plate boundaries along the Red Sea rift in the south with the Alpine orogenic belt to the north along a length of more than 1000 km. The regional style of tectonism has been described as rifting, (leaky) transform, or some combination of the two. The GFZ-Potsdam carried out MT measurements, as part of the multi-disciplinary DESERT (DEad SEa Rift Transect) project, a 300 km long transect traversing Israel, Jordan and the Palestine territories. The location in the Arava valley was chosen in order to study the DST (locally the Arava Fault, AF - the dominant fault of the DST between the Dead Sea and the Gulf of Elat-Aqaba) and the tectonic processes controlling it in their simplest expression [Ritter et al., 2003].

4.4.2 Projectives

The aim of a LOTEM survey, within the DESERT project was to resolve the conductivity distribution between 50 m and 3 km to further constrain the MT interpretation. Especially high resistive areas are of interest.

The LOTEM measurements, which where done by the IGM-Cologne between year 2003 and 2004 where conducted by Olaf Koch and described within Koch et al. [2004].

4.4.3 Prior measurements and data

The 2D-MT inversion results as well as seismic velocity profiles are displayed in figure 4.14. According to Ritter et al. [2003], the resolution at depth is good down to 3 km for the seismic profile. As due to the limited length of the MT profile (10 km), the resolution for the MT data degrades at larger depths. Thus, the interpretation was limited to $3-4\,km$. Both the MT and seismic data indicate significant lateral changes in physical properties across the AF.

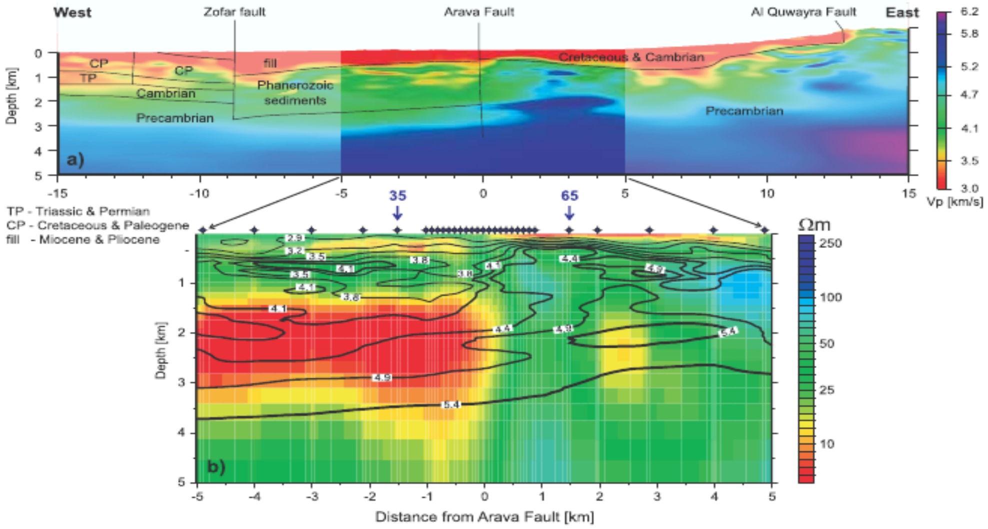


Figure 4.14: *a) Part of the tomographic P wave velocity model, centered on the Arava Fault (AF) zone. Main geological units are also indicated. b) Blowup of superimposed magnetotelluric (color-coded) and seismic P wave velocity results (contour lines derived from a) in the vicinity of the AF. The 30 MT recording sites are shown as diamonds at the top. Red and yellow colors indicate high conductivity. (after Ritter et al. [2003])*

4.4.4 LOTEM measurements

Figure 4.15 displays the setup of the LOTEM measurements that where carried out. The setup is already transformed to the model domain of interest. The LOTEM setup is centered perpendicular to the Fault-strike. Since it is also visible at the surface, it is approximated through a straight yellow stripe in the xy-overview. The whole spread has a total length of $\approx 12\,km$ with 4 different transmitter/receiver setups. Transmitters are located at positions

-4.6, -2, 1.8 and 5 km. Outer transmitters have receivers only in one direction. The corresponding receiver spread is pointing from the transmitter mid-point to the fault (broadside). The inside transmitters have receivers to both sides of the the x-oriented bipole. The whole data set consists of 164 receiver sites.

It has to be mentioned, that this is one of the first large scale 2D LOTEM setups which was carried out by the Institute of Geophysics and Meteorology, University of Cologne, so far. Thus, it was very challenging from the logistical point of view. Fortunately, the surrounding was a desert, making some things easier. However, the transmitter operation is challenging for a hot dry environment, since the current supply during the measurement has to kept quite constant.

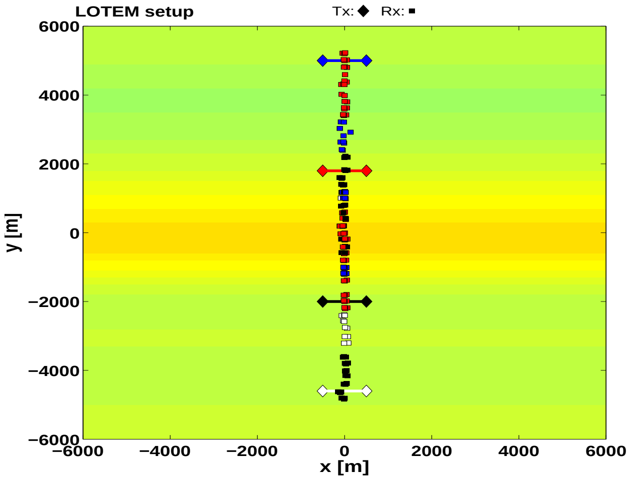


Figure 4.15: *LOTEM setup across the fault region. It consists of 4 different Tx locations (diamonds). The corresponding receiver locations are color coded respectively. The visible part of the fault system at the surface is approximated through yellow stripes at $y \approx 0\,m$.*

Each receiver site consists of SUMMIT/TEAMEX receivers (see for example Helwig [2000]), measuring the E_x-, $\dot{\mathsf{H}}_y$- and $\dot{\mathsf{H}}_z$-decay fields from 1 ms to 1 s in 16000/4800 data points, linear sampled in time. After the processing was done, they where logarithmically sampled with twelve points per decade. Thus, the time series consists of 42 time points per receiver site. After data preselection, the whole data domain consists of 6835 data points.

Before a 2D inversion was carried out, many of the receiver data was inverted with 1D models in order to find a set of appropriate starting models. They where stitched together to form a preliminary 2D model, which was tried to resolve with the large scale 2D inversion. Also various test runs with different forward grids for *SLDMem3t* where carried out in order to achieve a stable discretization. During this stage of the inversion program *sinv*, no automatic grid generator was available.

After a set of suitable grids was found, the next phase of the preparation for the 2D inversion was carried out: The model parameterization. Figure 4.16 displays the parameterization of the model that was found most suitable after various tests. The total number of model cells are given as $142 \times 31 = 4458$ in y and z direction. The lateral extent of the cells is $\approx 100\,m$ throughout the main parts of the cross section and increases at the borders. The number of lateral cells was approximately doubled compared to the 2D inversion of the synthetic LOTEM data. This was due to achieve a fine resolution of the lateral extension below the Fault region.

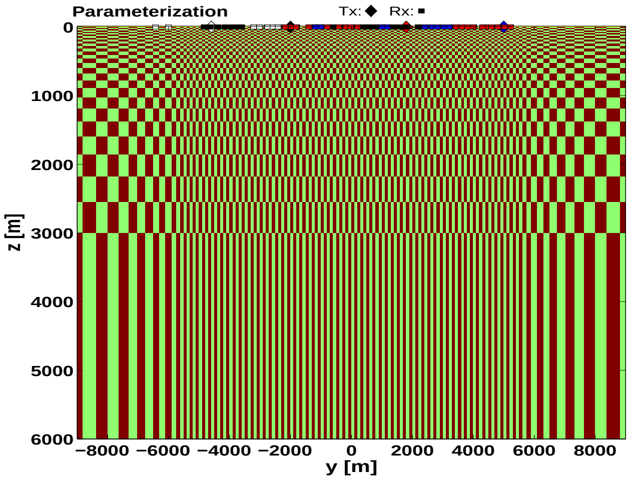


Figure 4.16: *2D parameterization for the LOTEM DESERT model.*

4.4.5 Preliminary 2D inversion results of the LOTEM data

The final parameterization was then used for the inverse solution of the DESERT data (fig. 4.18). For the inverse solutions presented here, a homogeneous halfspace of 22 Ωm was used as initial model. Consistent with the synthetic 2D inversions, the inversion of the field data was carried out with different regularizations. After eight iterations, a comparable misfit of $\chi \approx 8$ was reached with both regularization types.

Although the a posteriori LOTEM model (fig. 4.18) shows some different features, the overall acceptance to the MT inversion result of Ritter et al. [2003] (cf. fig 4.4.5) is quite excellent. As you can see from the inversion results (fig. 4.18), the conducting anomaly to the N/W, which was also studied within the synthetic examples (see figures 4.4 and 4.8), could be resolved very well. The lateral extend of the conductive anomaly has a very distinct border which coincides with the Fault region. The resistive zones, are clearly visible in both inverse solutions. Although, the spatial separation is not significant, it can be detected, that the resistive parts, S/E of the Fault are present.

Striking features in the models are shown for model cells directly below the receiver/transmitter sites, although the sites are not shown in this plot. Here, the model consists of conductive zones which where prolonged into depth, forming vertical dike structures. This originates from the fact, that the inverse solutions are obtained by using a vertical to horizontal smoothing ratio of 10:1. This means, that horizontal structures are far less regularized than vertical structures. The unequal ratio was used, in order to enhance the lateral resolution of the conductive zones with depth. With increasing depth, as could be observed within the synthetic study (see section 4.3), the inverse solution tends to smear out lateral conductivity contrast, leading to washed out and blurry images of the depth structures. As side effect, the unequal smoothing for deep parts leads to more lateral inhomogeneities for surface near regions. This can be identified easily by the rough changes in the lateral conductivity distribution below the transmitters/receivers. In contrast, the surface near conductive regions form smooth lateral conductivity zones in the inversion results of the MT. This seems much more likely than a sharp vertical dike structure.

Comparing the two a posteriori models, obtained by applying the first and the second order derivative smoothing, show, that the resolution of the LOTEM data is good until 2.5 *km*

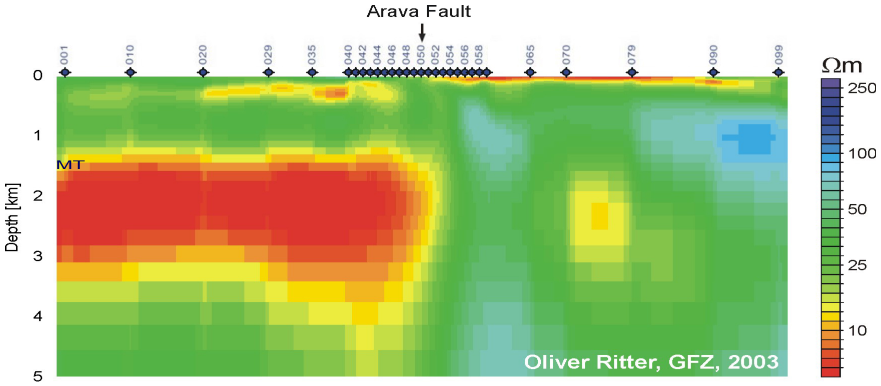


Figure 4.17: *Separate 2D MT result of the Arava basin after Ritter et al. [2003].*

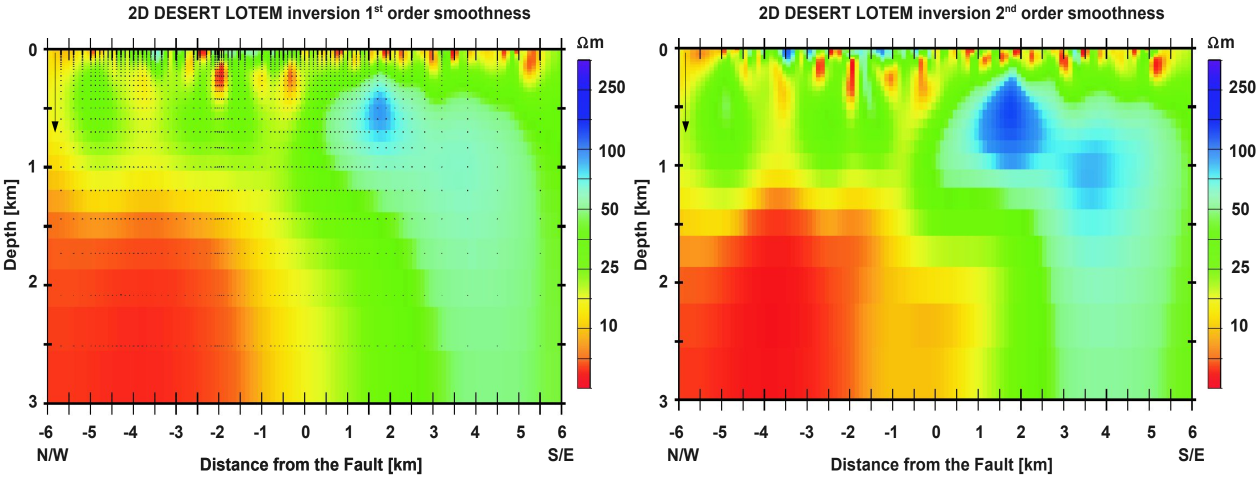


Figure 4.18: *Comparison of 2D DESERT LOTEM a posteriori models applying first (left) and second order smoothness constraints (right).*

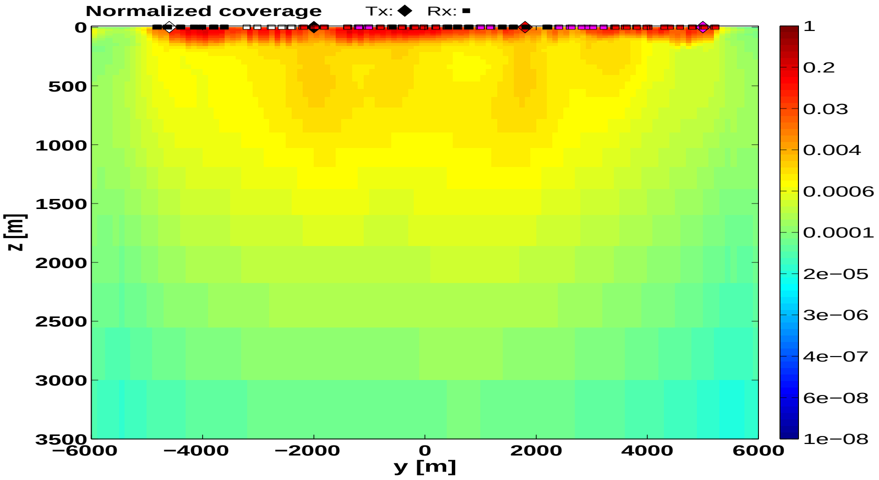


Figure 4.19: *Normalized coverage of the 2D models derived by the inversion of LOTEM data.*

depth. Yet, the bottom of the good conductive anomaly, or any vertical limit of this structure can not be detected within the LOTEM inversion results.

The coverage, displayed in figure 4.19, was weighted by the data covariance

$$\widehat{\mathrm{cov}}_j(\mathbf{m}) = \sum_{i=1}^{N} |\mathbf{C}_{d_i}^{-\frac{1}{2}} S_{ij}(\mathbf{m})| \ . \tag{4.32}$$

The obtained values where than normalized in order to distinguish different features, like

$$\widehat{\mathrm{cov}}_j^{norm} = \frac{\widehat{\mathrm{cov}}_j(\mathbf{m})}{\max\{\mathrm{cov}_j(\mathbf{m})\}} \ . \tag{4.33}$$

Good resolved model structures are displayed by values of $1 > \widehat{\mathrm{cov}} > 10^{-2}$, whereas poor resolved structures have values of $10^{-2} > \widehat{\mathrm{cov}} > 10^{-4}$. Model cells with values below 10^{-4} are not resolved.

As you can see from the figure, the model resolution is good down to depth of $2\,km$ for a central section of -5 to $5\,km$ ($\mathrm{cov}^{norm} \geq 10^{-2}$). To the borders, the resolution is decreasing fast. It can be assumed, that regions with (normalized) coverages below 10^{-2} are just poor resolved, which can be identified for structures below $2\,km$. This stands in very good agreement to the resolution estimates that where given by comparing different features of the two models obtained with different regularization. Here the inverse solutions show the most differences as well.

4.4.6 Selected LOTEM transients and model fit

Because the primary focus within this work is a feasibility study of the proposed multidimensional inverse solution, the fit of the predicted data to the measured signals is of main interest, too. Displayed in the figures 4.20 to 4.22, a compilation of selected LOTEM data are given. The complete LOTEM DESERT data you will find in appendix E.

As you can see from the figures, the fit of the predictions to the field data is quite good. Especially the $\dot{\mathsf{H}}_\mathrm{z}$-field transient sounding of the transmitter located at $1.8\,km$ S/E of the the Fault, which is showing a sign reversal can be reproduced excellent. However, the fit for the $\dot{\mathsf{H}}_\mathrm{y}$-transient of the last transmitter, is showing a double sign reversal which can not be reproduced.

A very obvious feature is present in all the $\dot{\mathsf{H}}_\mathrm{z}$-transients, namely the deviation to the halfspace solution of the signal for early times, which can be identified as "hump". For early times one would expect a constant signal from the theory, because for the early delay times the vertical voltage is mainly influenced by the surface conductance and not by lateral inhomogeneities. This "hump" is caused by the system response, which produce such a typical shape. This "feature" can be generally observed within the early time limit of the $\dot{\mathsf{H}}_\mathrm{z}$-transient data (see for e.g. [Hördt, 1992; Strack, 1992; Commer, 2003; Scholl, 2005]). Since the $\dot{\mathsf{H}}_\mathrm{z}$-transient is already an impulse response to the step-on current, the system response acts like a delay of the typical pulse form of the original signal. Thus, the $\dot{\mathsf{H}}_\mathrm{z}$-transient increases from nearly zero to the early time limit with a substantial delay caused by the system. Since the $\dot{\mathsf{H}}_\mathrm{y}$-voltages do decay over the whole time range, the affect of the system response is not that obvious in the early times. Yet, the $\dot{\mathsf{H}}_\mathrm{y}$-fields are affected from the system response the like.

Also, it can be observed from the figures, that the noise level for the $\dot{\mathsf{H}}_\mathrm{z}$-transients, is at least one decade below the $\dot{\mathsf{H}}_\mathrm{y}$-responses. Therefore, it will be more likely to assume a different

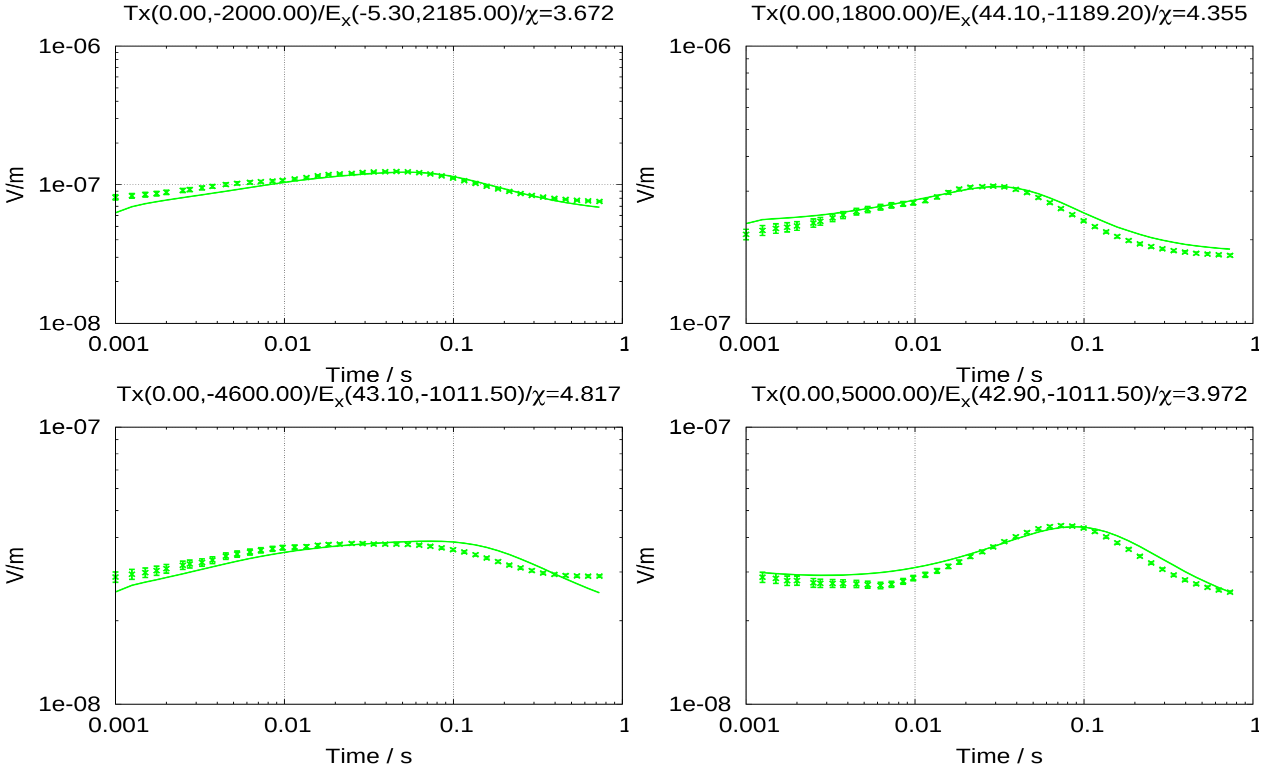


Figure 4.20: *Compilation of selected* $\mathsf{E_x}$*-field DESERT data (dotted) and fit (solid) for 4 different receiver sites. The coordinates of the transmitter and receiver locations are given in the title of each sub-plot, corresponding to fig. 4.15.*

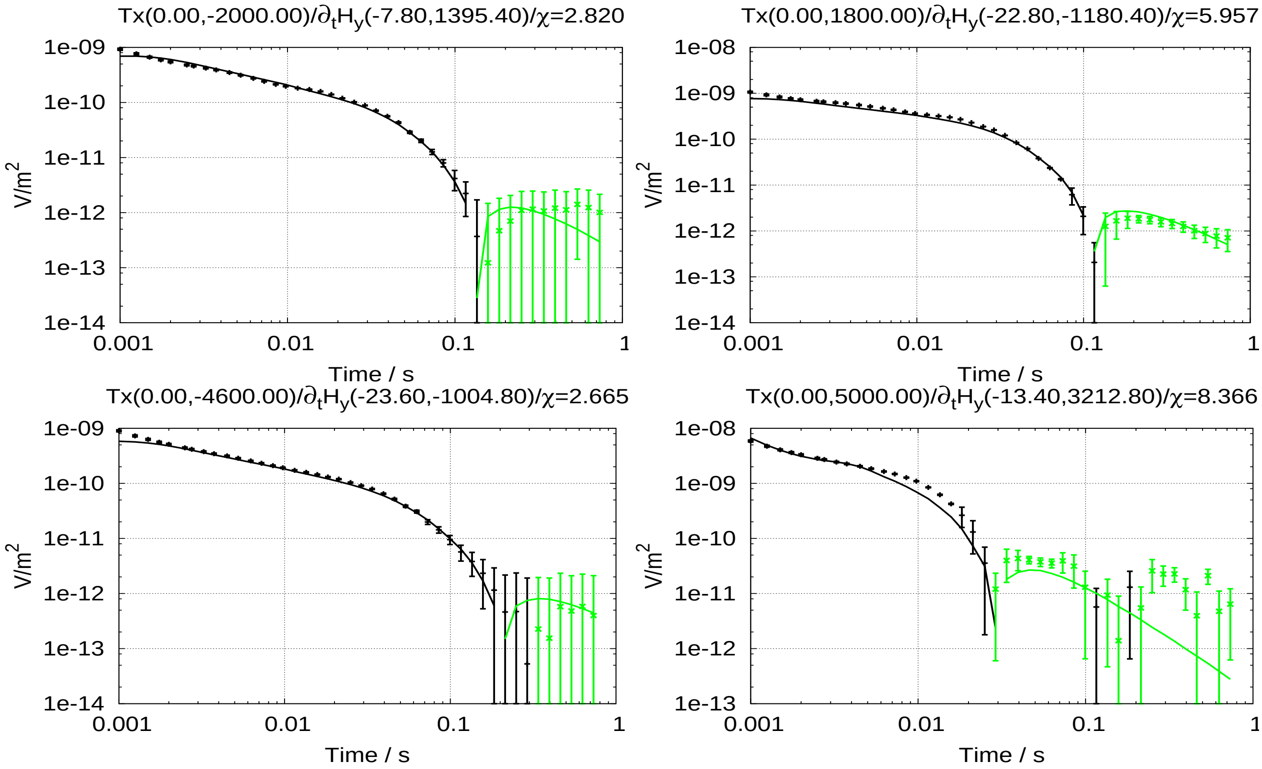


Figure 4.21: *Compilation of selected* $\dot{\mathsf{H}}_\mathsf{y}$*-field DESERT data (dotted) and fit (solid) for 4 different receiver sites. The coordinates of the transmitter and receiver locations are given the title of each sub-plot, corresponding to fig. 4.15.*

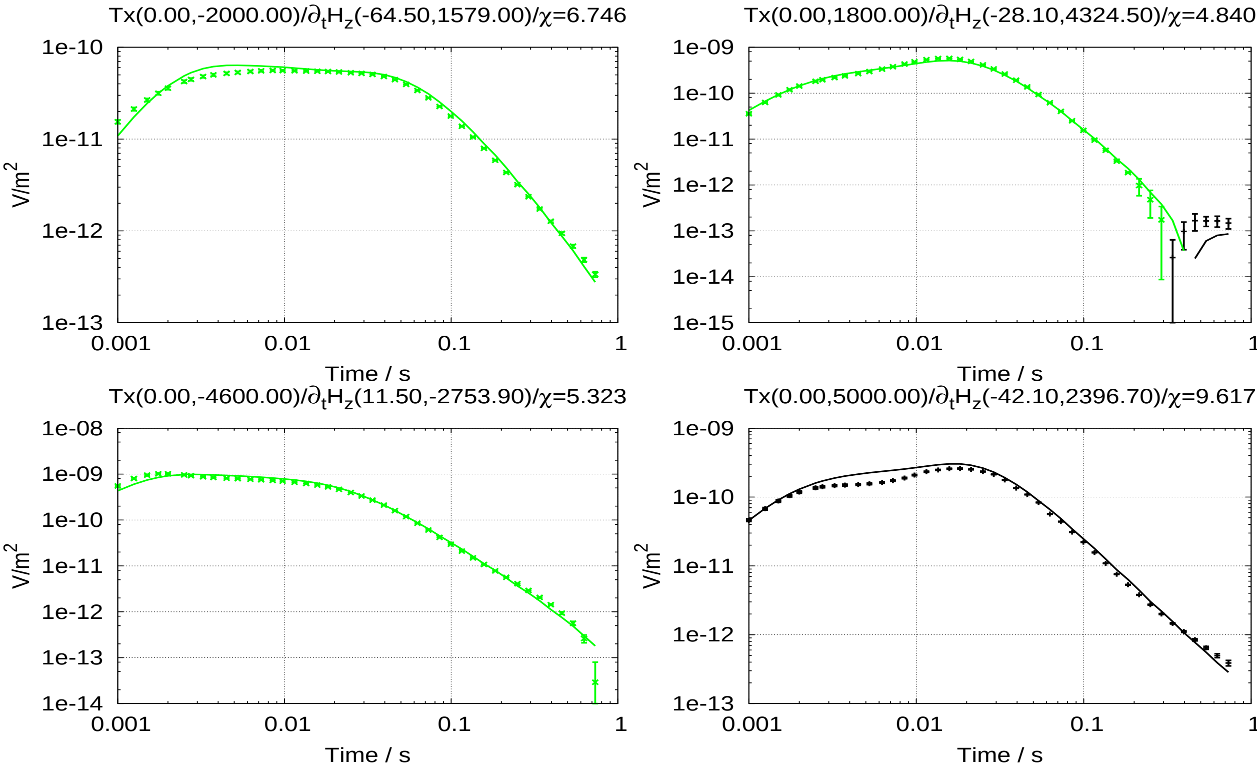


Figure 4.22: Desert $\dot{\mathrm{H}}_z$*-field DESERT data (dotted) and fit (solid) for 4 different receiver sites. The coordinates of the transmitter and receiver locations are given in the title of each sub-plot, corresponding to fig. 4.15.*

noise level for the horizontal and vertical voltages of synthetic data. However, since the measurements where done in a very unpopulated area it can be assumed that the noise level is substantially smaller than in more populated or urban areas.

4.4.7 Summary

For the joint interpretation of LOTEM and MT data, a coincident earth model could be found. Although, the different EM methods show particular variations, the main conductivity and resistivity features of the subsurface below the Arava Fault can be detected by means of 2D inversions of LOTEM and MT data. Comparing the 2D LOTEM inversion results, obtained by different smoothness constraints, increase resolution estimations of the a posteriori model. Investigation of the normalized coverage, which is calculated for the 2D a posteriori model, show nearly the same estimations. Also, predicted LOTEM data fit the measurements to a satisfying degree. Thus, the proposed inverse solution can be assumed to be valid for the main features of the a posteriori model.

According to Ritter et al. [2003], a geologic interpretation of the conductive anomaly is attributed to saline fluids within the sedimentary filling of the Arava basin. In this sense, the DST appears to act as an impermeable barrier between two different rock formations. Such a localized fluid barrier is consistent with models of fault zone evolution but has so far not been imaged by geophysical methods. The situation at the DST is remarkably different from active segments of the San Andreas Fault, which typically show a conductive fault core acting as a fluid conduit [Unsworth et al., 1999].

4.5 3D TEM inversion

If the model unknowns can vary in every three Cartesian directions, the inversion process becomes more complex. If the same spatial resolution should be delivered within a large scale 3D inversion, the number of unknowns increases by at least one to sometimes three orders of magnitude, compared to a 2D inversion. This leads to some ten to millions of model parameter to determine, rendering the normal equations unsuitable to solve with common direct solvers. As complexity increases, the non uniqueness based on the equivalence principle, increases as well. Additionally, it is merely possible to deliver a sufficient amount of data to establish a overdetermined system. Therefore the normal equations within large scale 3D inversion is mostly a under-determined system to be solved.

Preliminary notes

If M gets large, the only suitable way to solve the normal equations is to apply iterative solvers, like CG (see appendix B.0.5). Therefore, the iterative CG technique, with regard to global and local regularizations, was also realized within the inversion program. Yet, during some initial tests of the inverse process with CG, I figured out that it requires much iterations to solve the linear system. Connected with strong regularization, which was required to keep the model alterations within reasonable bounds, this leads to a very low convergence process. For e.g., it took more then twenty iterations for the CG scheme to yield a model update which was attained within the first step of an NLCG scheme. This may be significantly improved by proper preconditioning (see appendix B.0.6), but it was not tested throughout this work.

Newton based methods or Quasi-Newton methods converge in fewer iterations, although each iteration requires more computation than a conjugated gradient iteration. Quasi-Newton methods also require more memory to operate, because it stores all previous information of $\boldsymbol{\mathcal{H}}$. Although, this can be reduced using the limited memory L-BFGS method, described by Liu & Nocedal [1989]. During the development of the inversion program, I tested the L-BFGS minimization for large scale TEM optimization. After some initial tests of the implemented method it turned out, that the curvature condition (eq. (2.15)), which requires additional gradient calculation of the model to be updated, renders this approach unfeasible for the purpose studied in this work. The gradient calculation involves the main computational effort within the inversion scheme presented here.

If the initial guess is far away from the solution, the linear term on the left hand side of equation 4.4 dominates the event within the initial stage of the process. Thus, the steepest descent or NLCG method seems more convenient for the first few iteration steps, as for e.g. the L-BFGS method. Besides, the NLCG proofed already its usefulness during the inversion of 3D CSEM data sets, as presented by Newman & Commer [2005], and can be seen as a feasible way to attack large scale inverse problems which occurs for the inversion of 3D TEM data sets. As demonstrated by Commer & Newman [2008], a preconditioned variant of the NLCG showed additional improvements on the overall convergence of NLCG . As consequence, the results presented here, are achieved with a regularized NLCG-minimization scheme.

4.5.1 Optimization strategies for 3D TEM inversion

For gradient based minimization, the normal equations are stated using $\boldsymbol{\mathcal{H}} = \mathbf{I}$. Therefore, the normal equation is not solved, but the first order gradient information is used to reduce the cost.

This is achieved by means of a NLCG scheme (see section 2.1.2) for the model update and carrying out an (exact) Armijo-type line search with the condition

$$\frac{\Phi(\mathbf{m}_n + \alpha_n \delta\mathbf{m}_n)}{\Phi(\mathbf{m}_n)} < c_1 < 1 \ , \tag{4.34}$$

to minimize the cost. Here, the direction $\delta\mathbf{m}_n$ is recursively defined by

$$\delta\mathbf{m}_n = \begin{cases} -\mathbf{g}_n & n = 0 \\ -\mathbf{g}_n + \beta_n \delta\mathbf{m}_{n-1} & n \geq 1 \end{cases} . \tag{4.35}$$

For global regularization, the gradient is given by

$$\mathbf{g}_n = -\mathbf{S}_n^{\mathrm{T}} \mathbf{C}_d^{-1} \mathbf{y}_n + \lambda_n \mathbf{C}^{\mathrm{T}} \mathbf{C} \Delta \hat{\mathbf{m}}_n \ , \tag{4.36}$$

and for unconstraint optimization, the sensitivity information delivers the steepest descent direction:

$$\mathbf{g}_n = -\mathbf{S}_n^{\mathrm{T}} \mathbf{C}_d^{-1} \mathbf{y}_n \tag{4.37}$$

After some experiments with the formulas (2.21) to (2.23) for the conjugation scalar β, a modificated form of equation (2.22), reading

$$\beta_n = \max\left\{ \frac{\delta\mathbf{m}_n^{\mathrm{T}}(\delta\mathbf{m}_n - \delta\mathbf{m}_{n-1})}{\delta\mathbf{m}_{n-1}^{\mathrm{T}} \delta\mathbf{m}_{n-1}}, 0 \right\} \ , \tag{4.38}$$

was implemented into the code. The advantage of this form over the original β of Polyak-Ribière is, that it gives a automatic reset of the conjugation scheme without taking care of exceptional values for β.

The line search for the step length α_n is implemeted as follows:

1. In the first iteration, an exact line search is carried out, using a bisection/golden section search. After the appropriate α_n is found, the information is stored.
2. In the following iteration, an inexact line search is carried out based on the previous α_n, using a golden section into two directions.
3. If one of the new α values are a descent direction, a parabola is fitted through the α_*-value providing the best descent direction under the Armijo-condition (4.34).
4. If no proper descent direction is found with the first two guesses, return to 1.

After some tests it turned out that the exact line search tends to diverge sometimes, especially if the minimum can not be bracketed within the initial guess for α. This is a serious problem of an exact line search, which caused very much trouble during early development stages of the program. In order to prevent an exact line search to diverge, a secondary dropout criteria was installed. For example, if the model adjustments with a given α, namely the step size norm $\|\delta\mathbf{m}\|$, is to low, the search for a smaller α is not useful. It is implemeted to be the square of c_1 (see equation (4.34)) by default, but can be defined by the user. Therefore the secondary dropout criterion for an exact line search reads

$$\alpha_n \|\delta\mathbf{m}_n\| > c_2 \tag{4.39}$$

For the regularization parameter λ, I follow the deliberations given in section 4.1.2.

Resolution of the a posteriori model

Despite the apparent simplicity of the NLCG method, compared to the more elaborate CG or Quasi-Newton methods, the advantage of the NLCG minimization strategy is, that the resolution matrix is quite easy to guess. Because the assumption $\boldsymbol{\mathcal{H}} = \mathbf{I}$ is valid, the resolution matrix of the problem is of linear nature.

According to section 2.4.1, the resolution of the NLCG minimization process is given through

$$\mathbf{R} = \mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S} - \lambda\mathbf{C}_m^{-1}(\mathbf{m}_n - \hat{\mathbf{m}}) \ , \tag{4.40}$$

for the (global) regularized problem, and

$$\mathbf{R} = \mathbf{S}^{\mathrm{T}}\mathbf{C}_d^{-1}\mathbf{S} \ , \tag{4.41}$$

for the unconstrained optimization problem.

Therefore, the look into the diagonal elements of the resolution matrix reveals the resolution capabilities of the a posteriori model. Because, the sensitivity information is computed explicitly within the proposed inversion algorithm, this information can be accessed quickly.

The results presented in the following, show two extreme applications of TEM methods. The first, given in section 4.5.2, applies to an artificial crustal study, applying the LOTEM method. The second example, given in section 4.5.3, applies to surface near TEM soundings. Because the 3D part of the inversion scheme is just in a premature state and in order to keep it simple, both examples are chosen to resolve one single anomalous conductivity structure. In both examples, I suppose the target has sharp borders. Therefore, the second order smoothness constraint is used for a global regularization.

4.5.2 3D inversion of LOTEM data, synthetic example

To test the 3D capabilities of the proposed inverse solution, an artificial deep crustal study is assumed. Within this, the lateral and vertical extents of a conductive anomaly should be detected by means of the LOTEM method.

3D model and parameterization

The model that is used to simulate the measured data is quite simple (figure 4.24). In a homogeneous $10\,\Omega m$ halfspace, a conductive cube is embedded. The cube has a conductivity of $1\,S/m$ and it's top is situated at $300\,m$ depth. To avoid confusion, the word "measured data" is used synonymous for the synthetic data of the original 3D model.

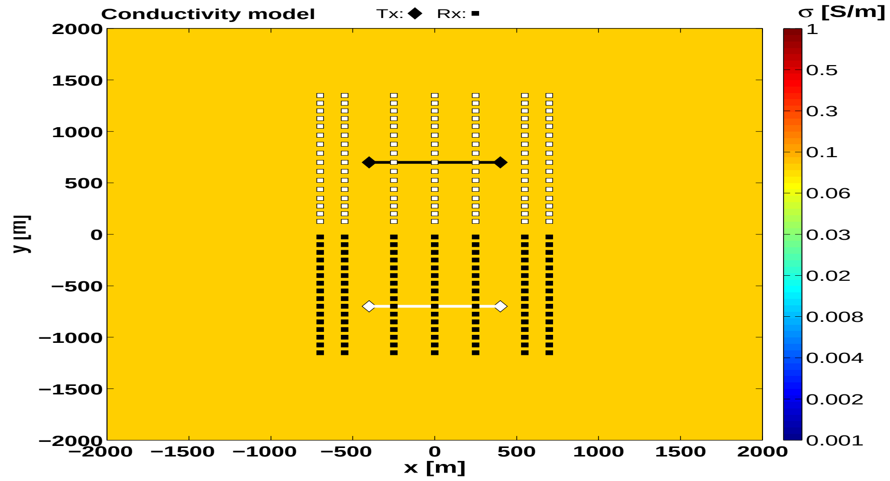


Figure 4.23: *Plane view ($z = 0\,m$) of the 3D model and LOTEM setup for synthetic data. The LOTEM setup consists of two galvanic coupled dipole transmitters (Tx) with corresponding receiver sites (Rx). The receiver and transmitters are color coded the same, to enhance distinction.*

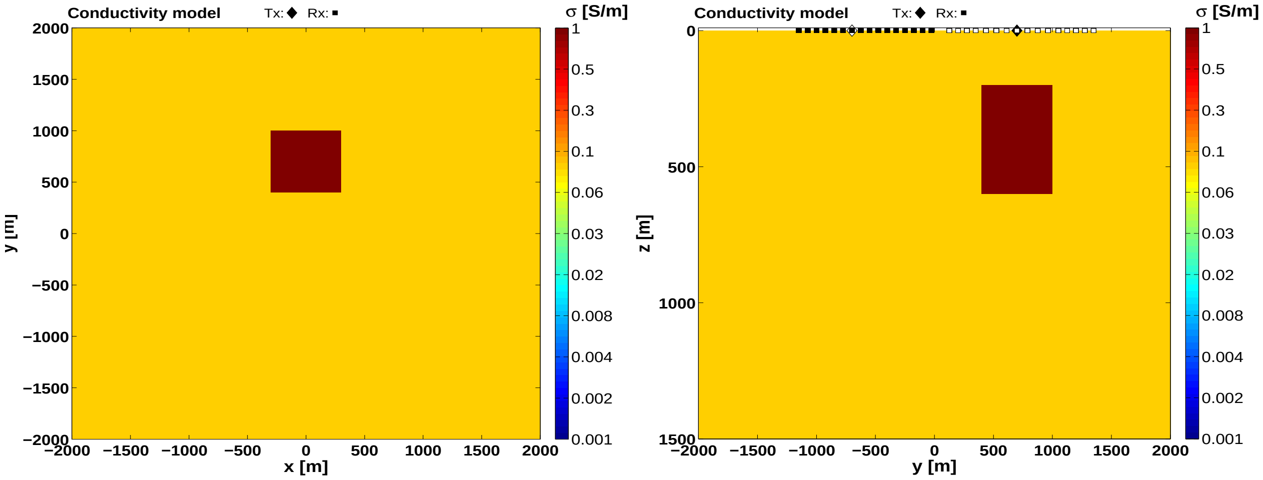


Figure 4.24: *Plane view ($z = 300\,m$) and cross section ($x = 0\,m$) of the synthetic forward model. A conductive cube of $500 \times 500 \times 400\,m^3$ edge length is embedded in a homogeneous-$10\,\Omega m$ halfspace. The conductivity contrast between the anomalous body and the surrounding is $1 : 10$.*

To generate the measured data, a simple symmetric LOTEM setup is designed. In figure

4.23, you can see a plane view of the LOTEM setup at the air earth interface. It mainly consists of two transmitter positions. Each transmitter has a x-oriented dipole length of $1km$ and comprises of 168 receiver sites. The sites of the transmitter at $(y = 0.7km)$ spread across the conductive anomaly to be detected. At the sites, three receiver components, measuring the E_x-, the $\dot{\mathsf{H}}_y$- and the $\dot{\mathsf{H}}_z$-response are assumed. The receiver sites are spatially distinct, which is due to the staggered Yee-grid of the 3D forward solver. It was chosen in that way in order to reduce interpolation errors that may be occur for the measured and predicted data. Therefore, no spatial interpolation scheme for $\mathbf{f}(\mathbf{m})$ is used in this example. The transient response of the anomalous conductivity is calculated between $0.5ms$ and $0.83s$ to 21 different delay times (log spaced). Thus, the whole data set consists of $N = 2 \times 168 \times 21 = 7056$ data points and 336 receiver components.

After the deliberations given in section 4.2 and 4.3, I decided to set up a constant noise floor for the whole data set, but to take care of the different receiver components. The assumption that where made for the errors of the electric fields, namely that the error of the first few time points is higher seemed convenient with regard to the DESERT field data. After adding noise to the data, I checked for the expected χ-value with another forward calculation. A value of $\chi = 1.4$ would fit the data within the error-bars of the noise level. It is substantially smaller than the values observed in 4.3, which is due to the assumption of a constant noise floor for the whole data domain.

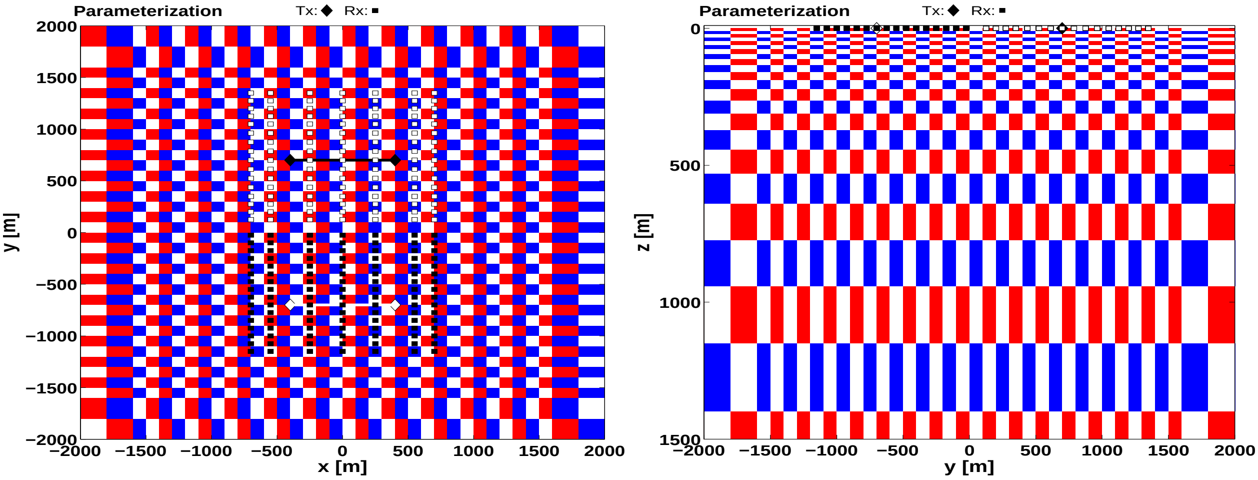


Figure 4.25: *Plane view and cross section of the parameterization for the 3D inversion.*

In order to reduce the numerical effort in terms of RAM-usage, and after a few test runs, the parameterization of the model (figure 4.25) seemed to yield quite good results. The model consists of $M = 42 \times 42 \times 23 = 40572$ model unknowns in x, y and z direction, which is very coarse, compared to the 2D inversions shown in section 4.3 and 4.4. The main part of the model is discretized with $100 \times 100\,m^2$ in the horizontal directions. One will notice, that the vertical extend is far less ($\approx 10\,m$) for the first $200\,m$ of depths. However, due to the fact that the parameterization can be chosen independently from the grid of the 3D forward solver, the vertical extensions increase logarithmically with depth. As can be observed from figure 4.25, for $z \geq 1\,km$, the vertical extent already reaches several hundred meters.

The statistics of the 3D inversion are given in figure 4.26. As you can see from the figure (top picture), the misfit during the 3D inverse process is decreasing in two steps. The first floor is reached after 4 iterations, where the misfit was already decreased to 40% of the starting value. At this time, the inversion code decided to search for a new regularization parameter,

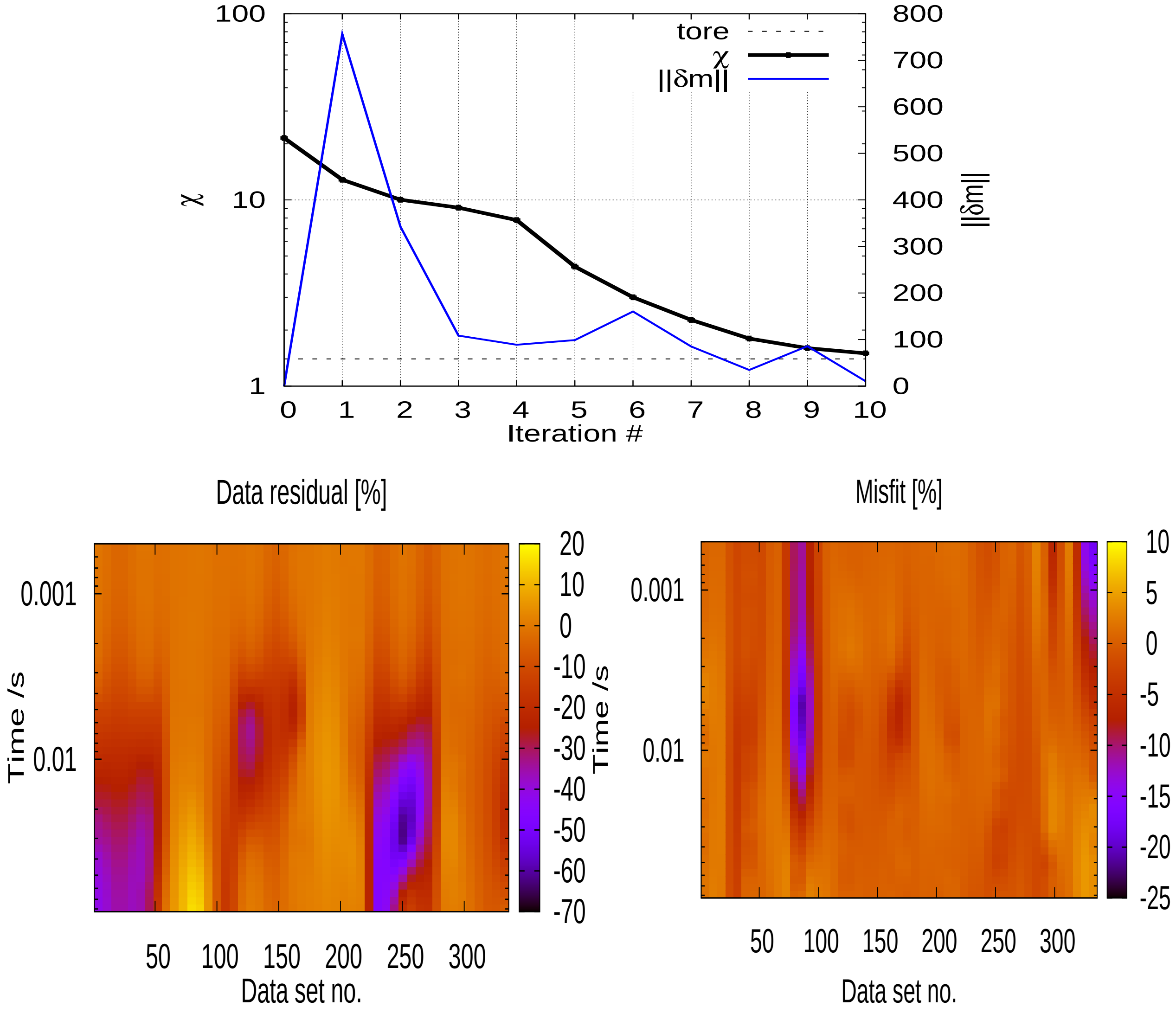


Figure 4.26: *Compilation of 3D LOTEM inversion statistics and data fit for all receiver data. The evolution of χ (misfit) for each iteration is given at the top. tore gives the expected χ for the data. The step size $||\delta\mathbf{m}||$ specifies how much the model is adjusted in each iteration step. The data fit of each data point of the corresponding receiver number is shown as contour plots at the bottom row. It gives a comparison of the error weighted residual $\hat{y}_i$ for the homogeneous starting model (bottom left) and the a posteriori model (bottom right). Please note, that the linear scale of the two plots is different.*

which was initially chosen as $\lambda = 10$ and thus decreased by 50%. After the restart of the conjugation scheme, a new α was searched and found using an exact line search. With the adjusted parameters, the inversion algorithm continued to work until iteration 10. After the $10th$ iteration, the secondary dropout criteria (eq. (4.39)) was reached.

As you can see from the comparison between the initial data residual (fig. 4.26 bottom left) and the one from the a posteriori model (fig. 4.26 bottom right), the data was fitted relatively good. Except for some regions of substantially larger residuals, the overall level of the fit was increased by at least one order of magnitude. The numbering of the data sets is like that the electrical fields are stored first, and the voltages last. Thus the pronounced blue stripe is due to five electrical field receivers, which responses are not fitted to the desired degree. Yet, the blue and yellow spots, which can be identified in the upper right and bottom right corner of fig. 4.26 (bottom right), are connected to three magnetical field responses, which where also fitted not sufficiently.

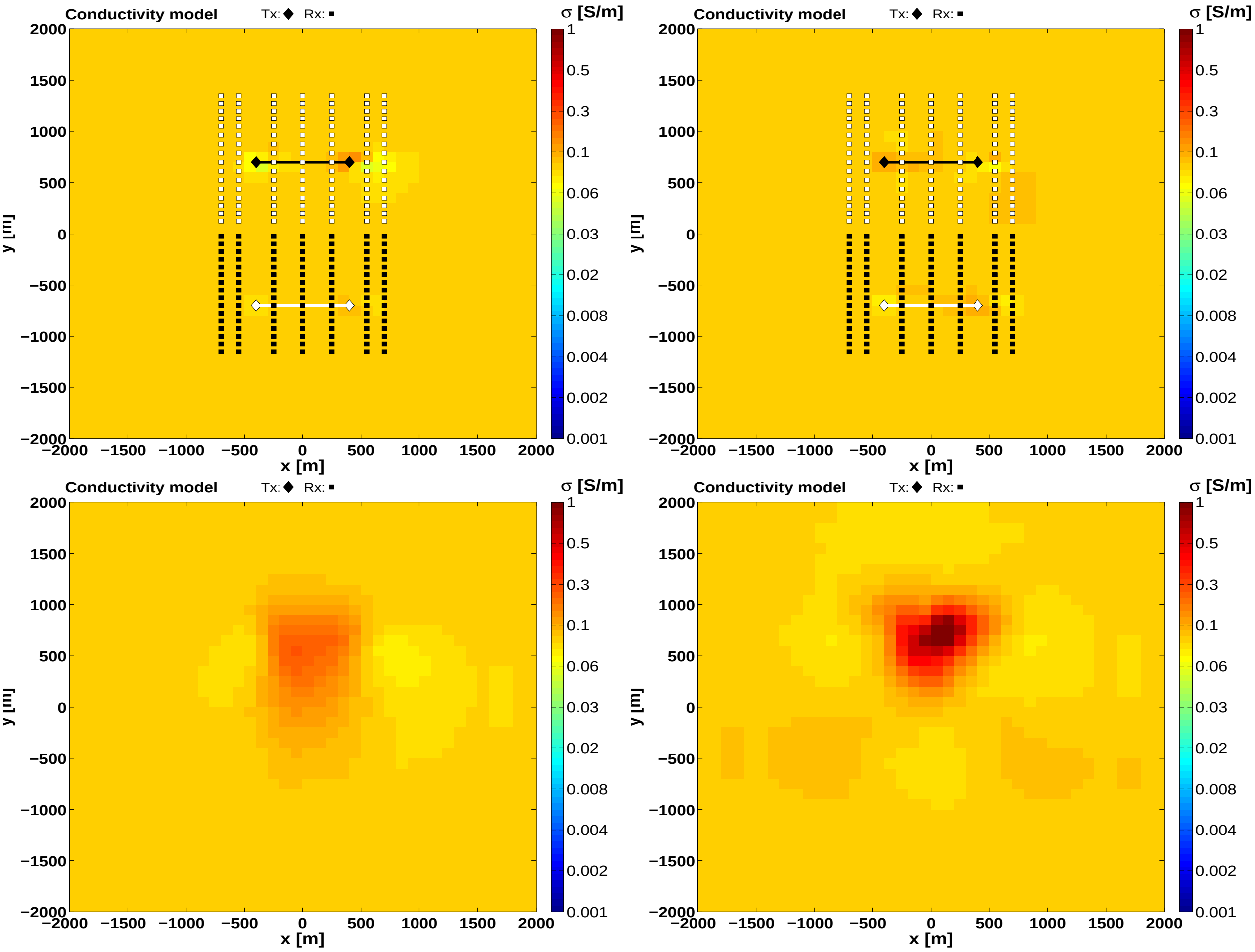


Figure 4.27: Comparison of xy-plane views through the model after the first (left) and last iteration step (right). At the top line you can see a xy-section at the air/earth interface. The bottom line shows a cut through the model across the anomalous body at $z = 300\,m$.

The figures 4.27, show a comparison of the 3D inverse solutions that are obtained after the first (left) and after the final iteration (right). The various slices through the model domain reveal, that the lateral extent of the anomalous body can be detected well. Especially the exact conductivity value of the body can be revealed by the proposed inverse solution. In particular, the figures displayed in 4.27 (top), where you can see a plane view of the air/earth interface, show only a few small-scale artifacts around the galvanic coupled dipoles, which is very confident. As you can see in figure 4.28 (left), where a xz-plane is shown through the axis of the transmitter at $y = 700\,m$, some surface near artifacts do occur. Some authors like Wang et al. [1994]; Newman & Commer [2005] reported problems the like for small surface near artifacts as well. Therefore it can be assumed, that this kind of surface near artifacts, which are due to over-accentuated model cells below the transmitter, are of minor relevance. However, they do influence the receiver stations which are directly above these features. After looking through the field data, which are not shown here, these receiver stations could be identified as the one with the high data residuals (see fig. 4.26, bottom right), which I already mentioned.

As described above (see section 4.5.1), the resolution information of the a posteriori model,

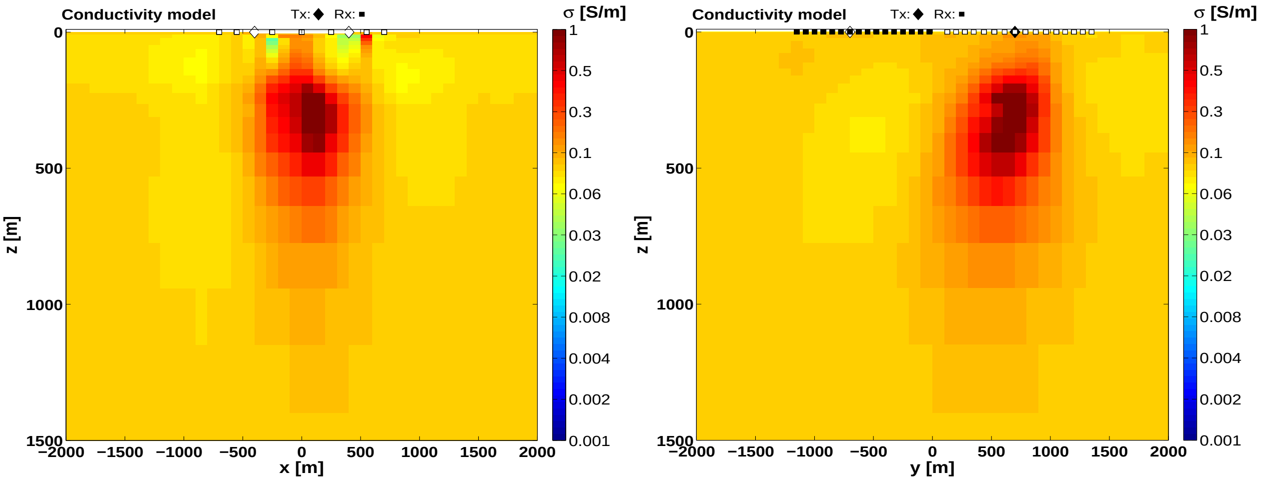


Figure 4.28: Cross sections through the a posteriori model after 10 iterations. The picture to the left shows the xz-slice across the anomalous conductive body at $y = 700\,m$. To the right you can see a yz-section in the equatorial plane of the transmitters ($x = 0\,m$).

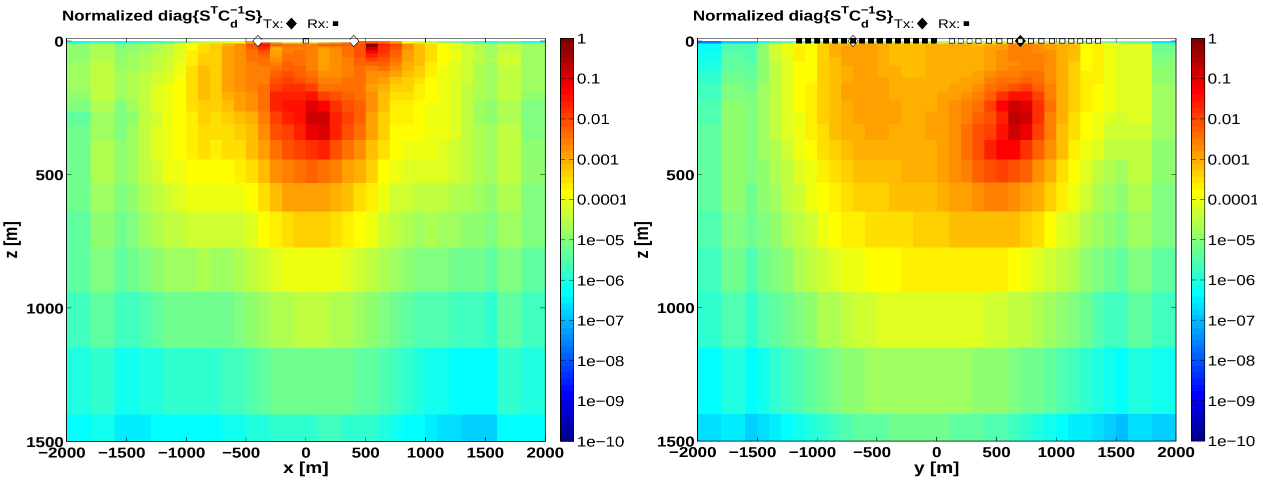


Figure 4.29: Cross sections of the modeling domain showing the diagonal entries of the resolution matrix of the a posteriori 3D model after 10 iterations. The figure to the left shows xz-slice through the anomalous conductive body at $y = 700\,m$. To the right you can see a yz-section for $x = 0\,m$.

can be accessed by looking at the diagonal entries of the resolution matrix. In order to compare the resolution of different features, I introduce a similar normalization to the resolution matrix, as was done for the normalized coverage (eq. (4.33)):

$$\hat{R}_{jj} = \frac{R_{jj}}{\max\{R_{jj}\}} \, . \tag{4.42}$$

Because of the inherent similarity to the normalized coverage, I assume that good resolved features attain values of 1 to 10^{-6}, whereas poor resolved features have lower values. Figure 4.29 displays the normalized diagonal entries of the resolution matrix for the same model slices as shown in figure 4.28. As you can see from figure 4.29, the anomalous structure and its surrounding is very well resolved ($\hat{R}_{jj} \approx 0.1$). Especially the surrounding of the anomalous structure is of particular interest. Here, the values of $\hat{R}$ attain values of $0.1 > \hat{R} > 0.01$, which is very good resolved.

Computational expenses

With the model parameterization described above, the sensitivity matrix, stored as 8-Byte single precision matrix, took roughly 2GB of RAM. For the homogeneous starting model, the calculation of the sensitivity matrix was done following the 1D adjoint approach. It was used to deliver a fast insight into the model resolution of the initial model parameterization and took about one hour on 4 nodes of a small Linux-Cluster. Each node within the Cluster is equipped with an AMD Athlon™XP 2200+ CPU, with 4GB of RAM. Therefore, the 3D inversion did not scratch at the limit of each single nodes.

For the sensitivity calculation using the hybrid adjoint approach, the calculation of the background fields consumes the most computing time. If it is run parallel, and for intermediate grid sizes of about $N_n = 70 \times 70 \times 60$ grid nodes, the total number of background fields to be calculated is roughly given by $3 \times 60 \times 60 \times 55 = 6 \times 10^5$. The reduced size is due to the fact, that the grid dimensions are generally larger than the modeling domain (see section 3.4). Therefore, the part of the background fields for the domain of interest is much smaller than for the whole background grid domain.

However, even with moderate dimensions for the Krylov subspace (e.g. $\mathsf{M} \approx 5 \times 10^3$), the whole problem already leads to tremendous computational effort in terms of storage ($16 \times 3 \times 10^9 \approx 44$ GBytes). A fast hard-disk access is therefore a very helpful solution, because the results can be stored partially on disk. Thus the above mentioned number of storage is just flowing through the CPU's of the nodes and never stored as whole (which is of course impossible on small systems). Yet, compared to explicit time stepping scheme, this is still relatively moderate in terms of computational effort.

4.5.3 3D inversion of SHOTEM data, synthetic example

Here, the SHOTEM method is applied, to reveal the lateral borders of a thin conductive anomaly between $3\,m$ and $12\,m$. To increase spatial resolution, a very uncommon variation of an in-loop setup was chosen (fig. 4.30). The main difference to a conventional in-loop configuration are listed as follows

1. Besides the conventional vertical magnetic voltages, the horizontal voltage information is measured (multi component setup).
2. The three components of voltage information are collected within different positions inside the loop (multi site/multi component).
3. The whole setup consists of several overlapping transmitters covering the area of interest. Together with the corresponding multi-side receivers a dense grid of measurements can be performed to enhance lateral resolution.

3D model and parameterization

The model that is used to simulate the measured data is quite simple (figure 4.31). In a homogeneous $100\,\Omega m$ halfspace, a conductive sheet is embedded. The sheet has a resistivity of $2\,\Omega m$ and it's top is situated at $2\,m$, which is apparently very shallow.

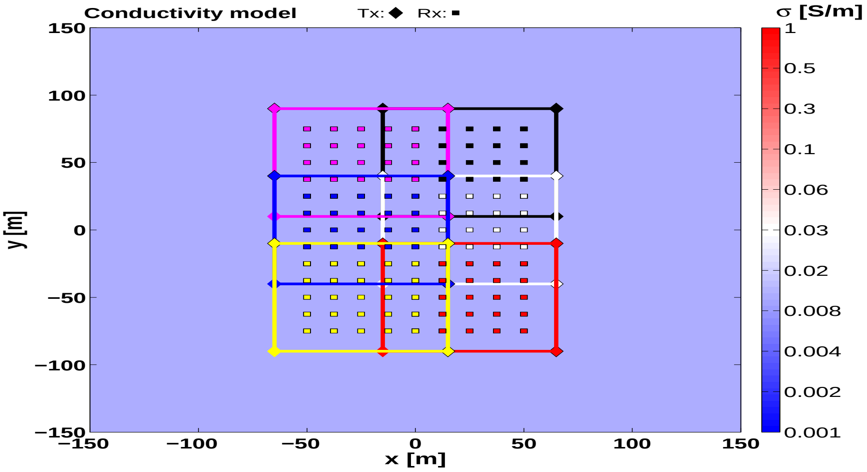


Figure 4.30: *Plane view of the unconventional SHOTEM setup. The setup consists of six loop wires with overlapping areas. The color of each loop wire corresponds to the multicomponent receiver sites which are color coded as the transmitter, resp.*

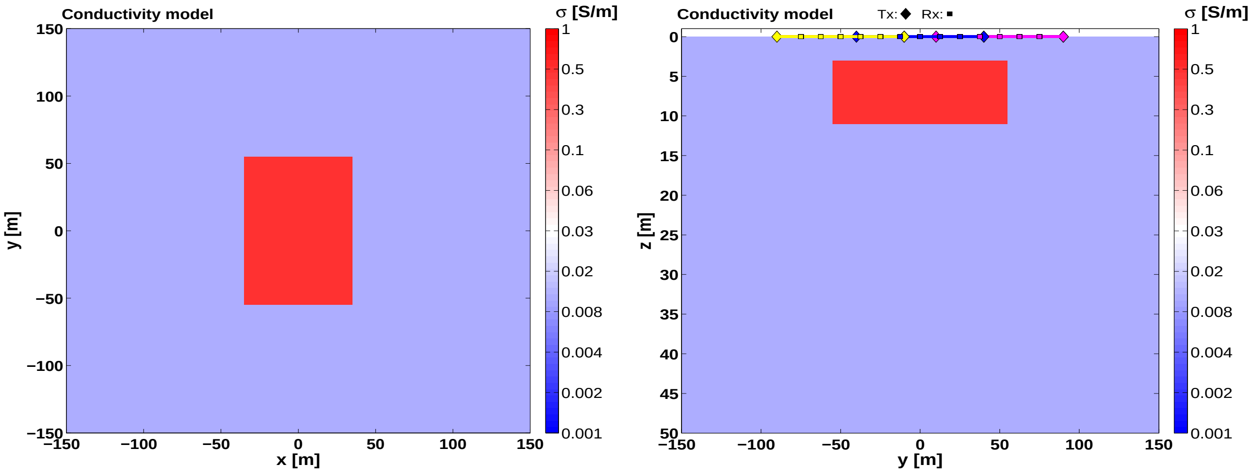


Figure 4.31: *Simple 3D conductivity model for the generation of measured SHOTEM data. A thin conductive anomaly of* $110x70x10\,m$ *extent is embedded in a homogeneous-*$100\ \Omega m$ *halfspace. The conductivity contrast between the anomalous body and the surrounding is* $2:100$.

To ensure a high lateral resolution, the uncommon SHOTEM setup was used to generate the measured data. In figure 4.30, you can see a plane view of the SHOTEM setup at the air/earth interface. It consists of 6 overlapping transmitters. Each transmitter comprises of 25 receiver sites following a grid like structure. The horizontal distance between two receiver sites is $12.5\ m$, in each direction. For the transmitter size of an $80 \times 80\ m^2$ loop wire, the effective area for the receiver sites covers an area of $50 \times 50\,m^2$. For convenience, the effective area of the loop transmitter is the area, in which the induced voltage is not deviated through the primary signal. Therefore, you will find five by five receiver stations forming a regular grid within the $80 \times 80\ m^2$ loop transmitter. The setup ensures a dense lateral coverage for the area of investigation. Since the main effort within TEM measurements is to lay out the transmitter properly, a fast progress of the measurements can be obtained.

The transmitters and receiver sites are spread across the conductive anomaly to be detected. The whole setup is ideally centered around the mid point of the conductive region (see fig. 4.31). At the sites, the previous mentioned receiver components, measuring the $\dot{\mathsf{H}}_x$-, $\dot{\mathsf{H}}_y$- and

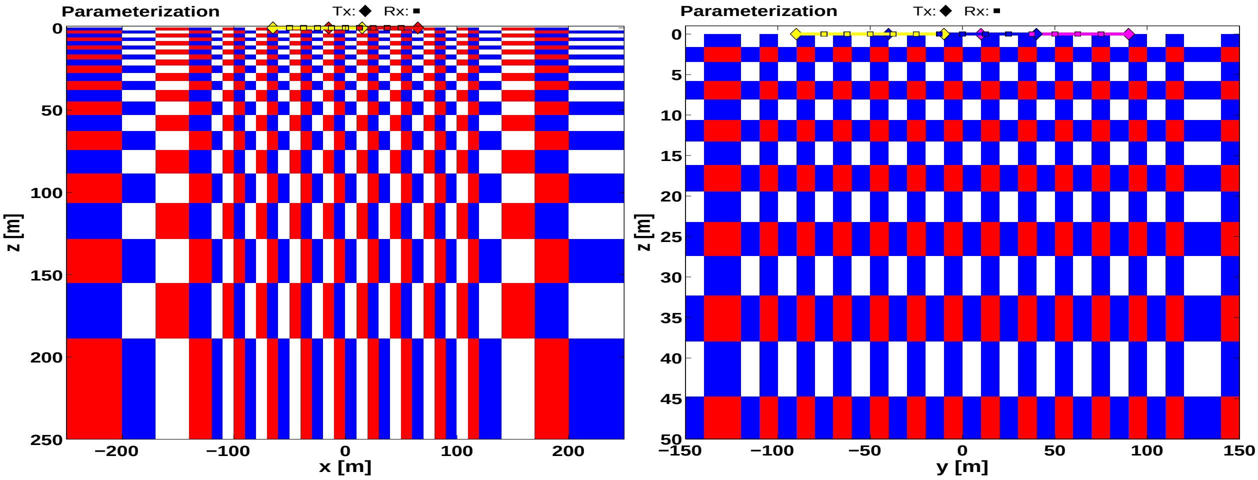


Figure 4.32: *xz- (left) and yz-cross section (right) of the model parameterization for the 3D SHOTEM inversion.*

the $\dot{\mathsf{H}}_z$-response, are assumed. The receiver sites are not chosen to be coincident with the underlying forward grid of the *SLDMem3t*.

Before I compile the number of total receiver sites, I have to give some remarks on ”spurious modes” of the induced horizontal voltages. From the theory it can be observed, that the horizontal voltage components comprise of some areas, where the measured voltages are very weak. For e.g., the induced horizontal fields vanish for a homogeneous halfspace, if they are measured within the center of the loop. Moreover, the $\dot{\mathsf{H}}_x$-response within the loop and over a homogeneous halfspace vanishes at every $y = 0\,m$ along the whole profile. Similar results can be obtained for the $\dot{\mathsf{H}}_y$-response, which vanishes for every $x = 0\,m$ along the y-axis. In order reduce numerical problems with the forward solver in advance, I excluded these ”spurious modes” from the profile. Thus, the total number of transient response measured above the anomalous conductive sheet is given by $6 \times 25 + 12 \times 50 = 390$ measured voltages. They are calculated between $1\,\mu s$ and $0.1\,ms$ for 21 different delay times (log spaced). Therefore, the whole data set consists of $N = 390 \times 21 = 8190$ data points in total.

I assumed a constant noise level for the whole data domain. After adding the noise, I checked for the expected χ-value with another forward calculation. Therefore, a value of $\chi = 3.6$ would fit the data within the error-bars of the noise level. As you will see later, the noise treatment caused exceptionally high oscillations onto some horizontal transient data. But, as the transient data seems more likely to simulate for a worse (but not worst) case and may be ”real world” scenario, the noise treatment seemed appropriate. Yet, as is demonstrated in the following, the proposed inverse solution can manage those difficulties to a confident degree.

For the findings of the right parameterization of the starting model, it took several test runs with different horizontal and vertical discretizations of the model. Figure 4.32 displays the parameterization of the model domain to carry out the 3D inversion. As can be seen from the figure, the xz-section shows a bigger part of the whole modeling domain, whereas the yz-section displays the parameterization of the main region of interest. Here, the horizontal extensions of the model are 10 m, laterally symmetric. Since the xz-section (fig. 4.32, left) shows a bigger part of the model, it can be observed, that the model cells increase horizontally and vertically logarithmically with increasing distance from the central region

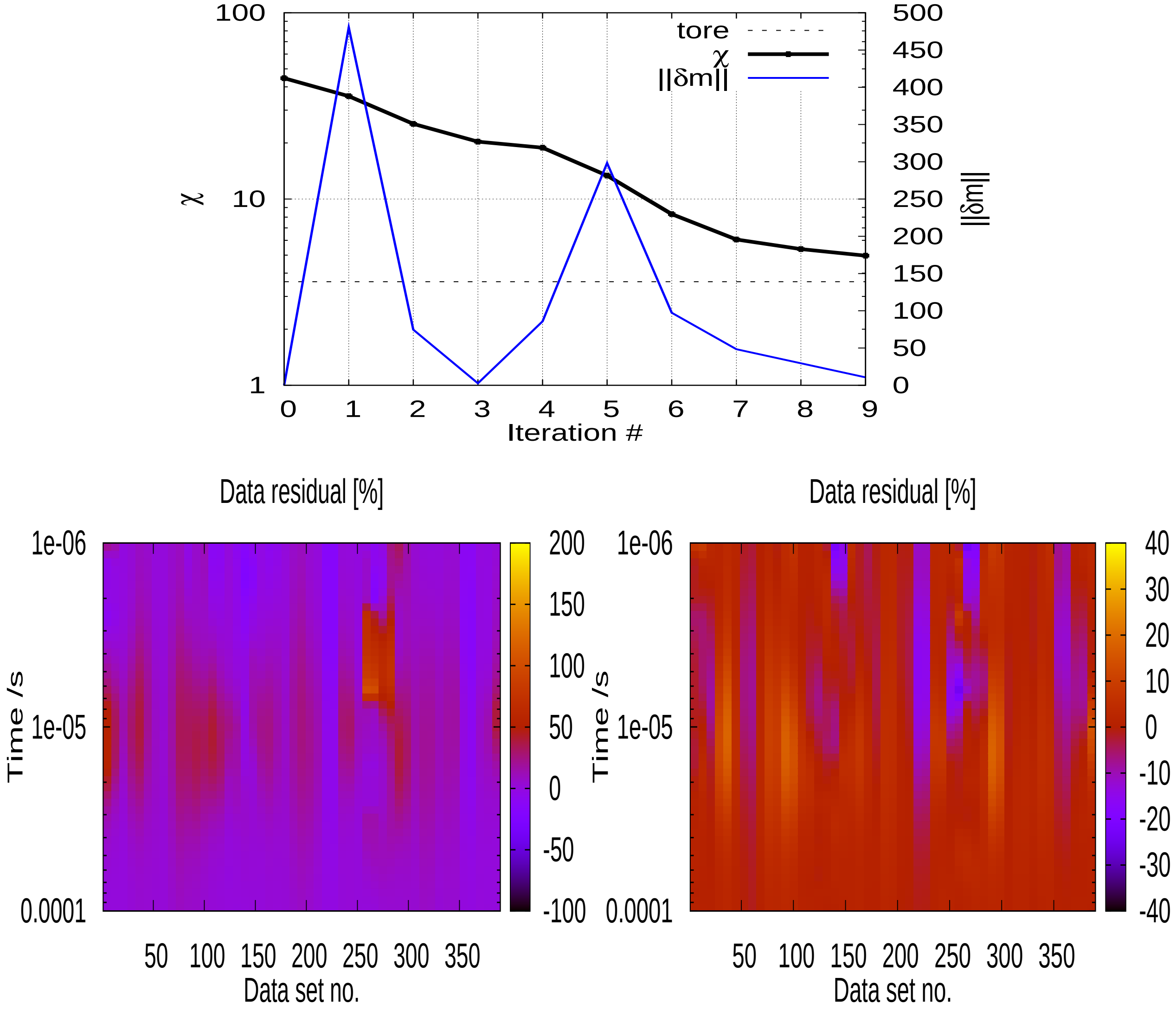


Figure 4.33: *Compilation of 3D SHOTEM inversion statistics and data fit for all receiver data. The evolution of χ (misfit) for each iteration is given at the top. tore gives the expected χ for the data (tore $= 3.6$). The step size $\|\delta\mathbf{m}\|$ specifies how much the model is adjusted in each iteration step. The data fit of each data point of the corresponding receiver number is shown as contour plots at the bottom row. It gives a comparison of the error weighted data residual $\hat{y}_i$ for the homogeneous starting model (bottom left) and the a posteriori model (bottom right). Please note, that the linear scale of the two plots is different.*

(space dimensional). For this surface near application, the vertical extent of the model cells are taken to be $1.6\,m$ for the first four vertical cells. The vertical extent of the cells increase from $z = 10\,m$ following a logarithmic increase of the thickness. Altogether, the modeling domain consists of $M = 40 \times 40 \times 22 = 35200$ model unknowns in x-, y- and z direction.

As given in the statistical analysis, which are displayed in figure 4.33, a similar "two step" misfit decrease, can be observed. The same particular feature was already described within the 3D LOTEM inversion statistics (see fig. 4.26), and is due to the reset of the conjugation scheme. This time, another "two step" progression in the the model adjustments can be analyzed, which coincides with the reset of the conjugation scheme. It was due to the fact, that the regularization parameter was decreased from $\lambda = 10$ as initial value to $\lambda = 0.01$ after the fourth iteration. Therefore one would expect the inverse solution to have much

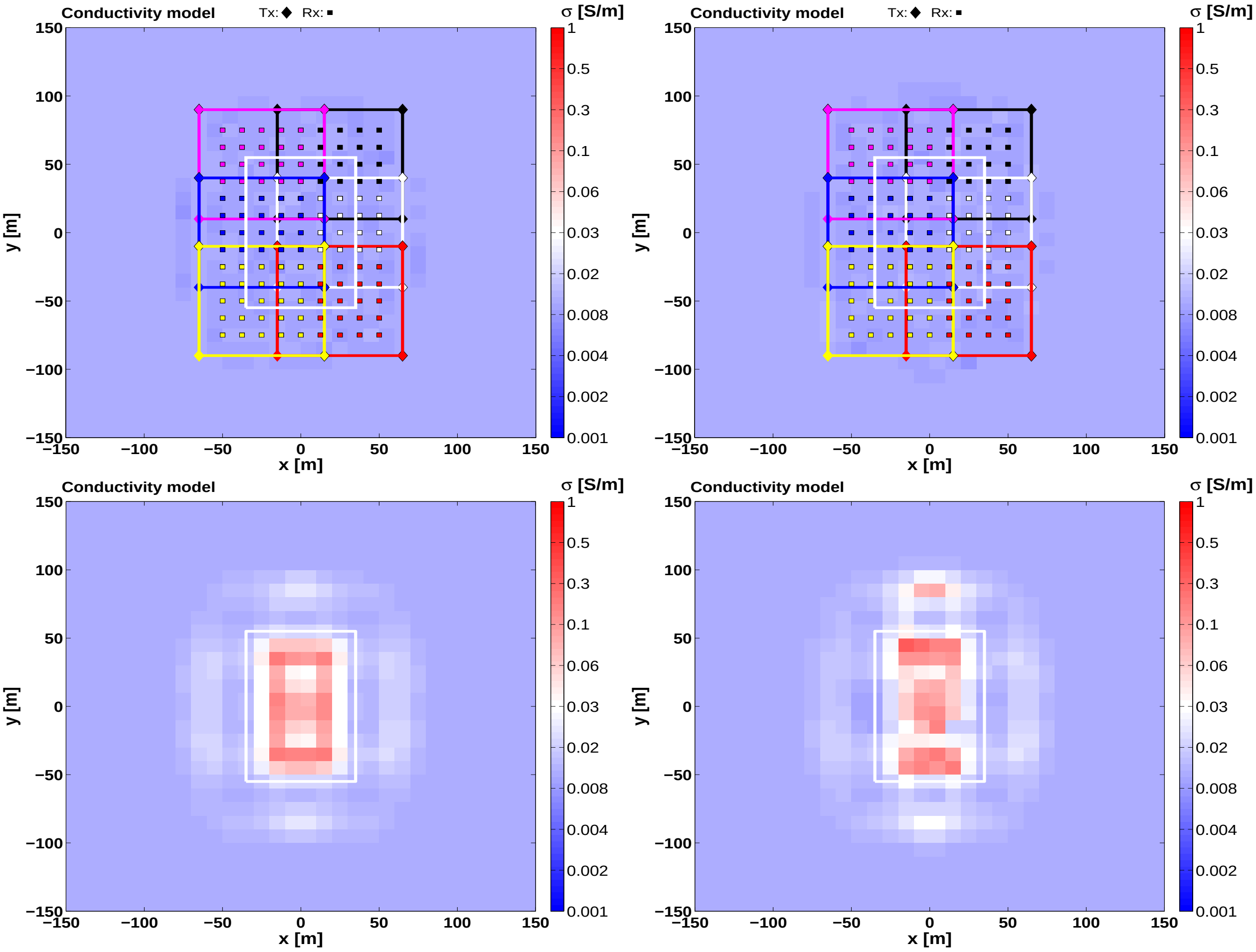


Figure 4.34: *Comparison of xy-plane views through the model after the fourth (left) and last iteration step (right). At the top line you can see a xy-section at the air/earth interface showing the 6 overlapping transmitters. The white rectangle marks the shape of the original 3D body. The bottom line shows a cut through the anomalous sheet at* $z = 6\,m$.

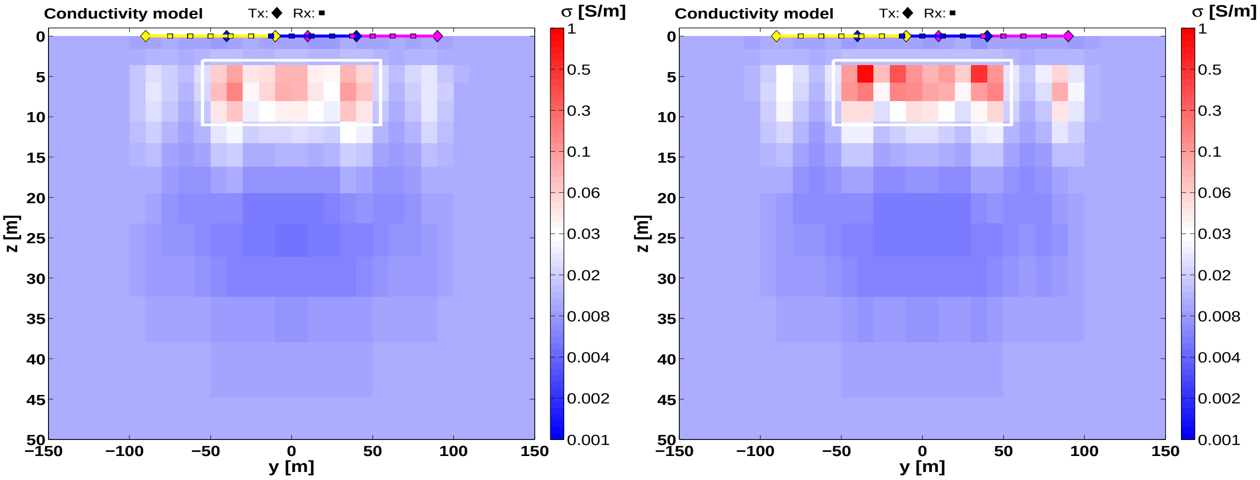


Figure 4.35: *Comparison of yz-cross sections of the model after the fourth (left) and last iteration step (right). The white rectangle marks the spatial extent of the original 3D body.*

more structure, than the model of the fourth iteration.

To proof the extend of convergence progress, that where made during the first four iteration

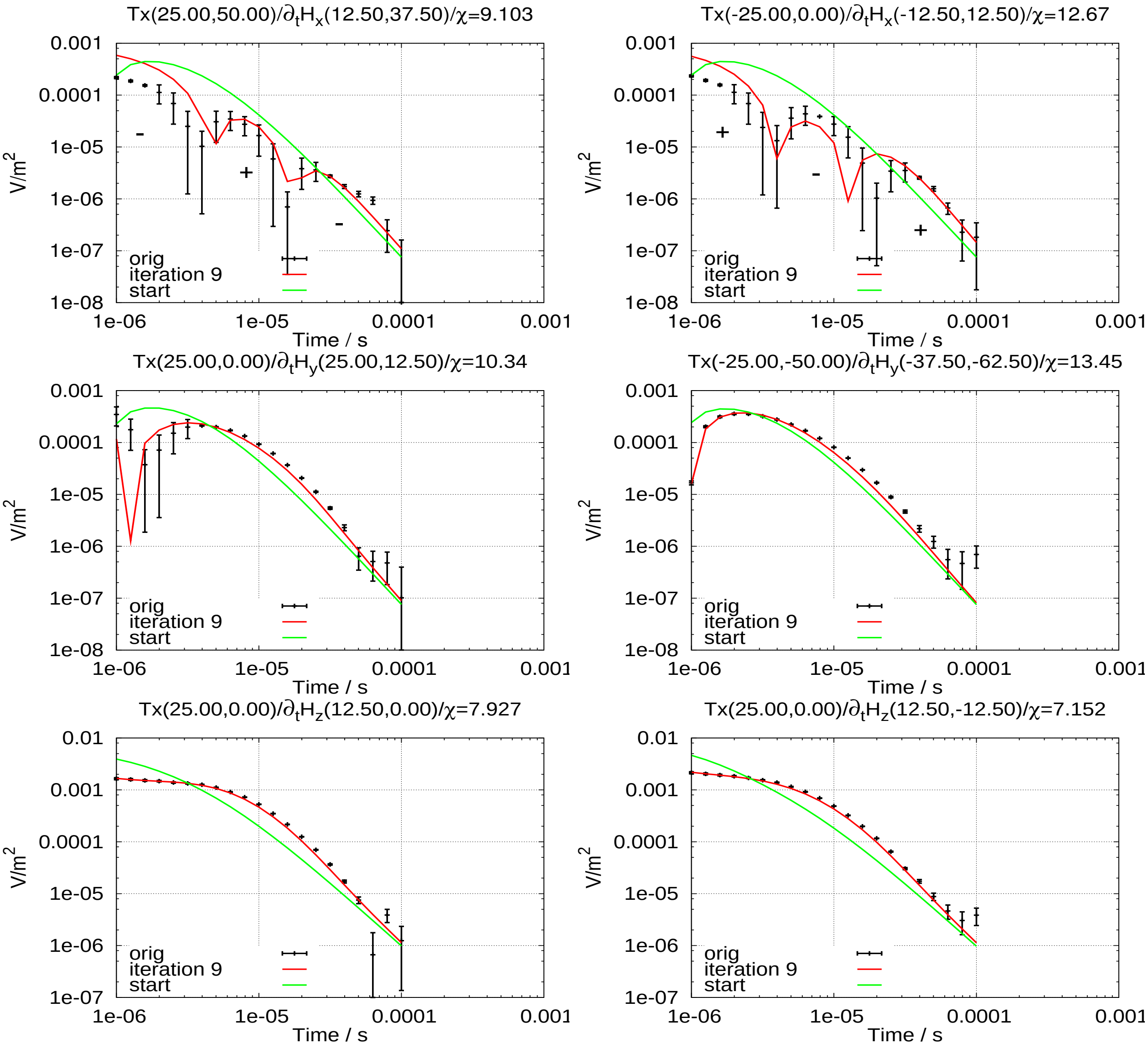


Figure 4.36: Comparison of selected data for the starting model and the inverse solution with the original data for different receiver stations. The displayed transient responses for $\dot{H}_x$*- (top)* $\dot{H}_y$*- (middle) and* $\dot{H}_z$*-responses (bottom) are selected for distinct receiver/transmitter combinations. The precises coordinates of the receiver, as well as the transmitter can be depicted from the title of each figure. There you will find the fit of the a posteriori model, too.*

steps, please compare the model sections displayed in figures 4.34 and 4.35. Here you can see, that the a posteriori model, that was obtained after the first four iteration steps, already reveals the lateral extent of the anomalous sheet quite satisfying. Although the conductivity values, that where obtained by the inverse solution are not recovered, the particular shape of the anomalous body is clearly visible. The image which is obtained after the early inversion stages yield some major artifacts. Yet, they are enhanced after the last iteration was done (see fig. 4.34 and 4.35, right). During the last iteration steps, the true conductivity of the anomalous region can be resolved very well. Besides the anomalous domain, a resistive region below the body can be identified as another artificial structure within the a posteriori model.

The comparison between the predictions, the data of the starting model and the measurements is compiled in figure 4.36. As you can see from the top row, the $\dot{\mathsf{H}}_{\mathsf{x}}$-response yield two separate sign reversals, which could be fitted at least well. If one take a close look to the χ-error of the predictions and the measurements, one can see that the overall error level is very high, although the predictions fit the data to a obvious degree. The cause for this strong irregularity is of course due to the high rate of noise, which is added to the measurements. However, even with this high noise level, the proposed inverse solution fits the data quite satisfying.

4.5.4 Discussion

Comparing the inverse solutions which are proposed for 2D and 3D TEM data, it could be observed, that the 2D inversion which is established by an regularized Gauss-Newton type inverse solution converges much faster, especially with regard to the complexity of the problems which where solved. However, regarding the premature state of the 3D TEM inversion scheme, the results which where investigated with synthetic data show very promising results. Although, simplified synthetic 3D experiments lack complexity, namely to detect conductive bodies within a homogeneous halfspace, the spatial dimensions as well as the resistivity values are well resolved within both, the deep crustal and the surface near application.

CHAPTER 5

Conclusions and discussion

This work has presented two approaches for the inversion of TEM field data. Each one uses the sensitivity information as main driver for the model adjustments within the optimization procedure. With the implemented method and according to Hördt [1998], the sensitivity information of a single receiver within the whole model domain can be computed by convolving the background electrical fields with the adjoint Green functions of the particular receiver acting as source. The problem of the involved numerical implementations, namely the convolution of the time series and the re-weighting scheme to deliver the sensitivity information independent of the spatial discretization from the forward calculation, could be solved to a satisfying degree. The latter ensures, together with the material-averaging-scheme of the 3D forward solver *SLDMem3t*, that the parameterization of the model domain can be chosen independently of any predefined grid. It may therefore be chosen to fit best to the properties of the inverse problem, which is reported as necessary for the stability of the inverse solution. In order to reduce storage space and time, the 1D sensitivities, as given in section 3.5, would produce satisfying results for the starting model.

Since the SLDM demands a stable grid to produce accurate results, I strongly encourage a detailed grid analysis as exemplified in section 3.4, before any inverse process is started. Further I would suggest some extensive sensitivity analysis for the model parameterization in order to check how the model to data interaction is mapped. The sensitivity information proofed to be very useful in terms of finding a confident parameterization for a stable inverse process. Additionally, the sensitivity information can be used to yield insight of the resolution capabilities of the inverse solution. Especially for gradient based minimization techniques, where the linear assumption of the resolution matrix is valid.

The 2D inversion, as presented in section 4.3, is suited for TEM data sets collected along profiles aiming to resolve the subsurface conductivity distribution by means of 2D models. It was successfully applied to synthetic LOTEM data, leaned on a real world application. The proposed inversion scheme showed very quick convergence and the fit of the predictions to the data was excellent. Especially sign reversals, which could be observed for the induced voltages could be reproduced to a satisfying degree.

Hördt et al. [2000] for e.g. used 2D forward modeling in order to explain sign reversals in the voltage transient data of a LOTEM survey in the Odenwald area (Germany). Among several attempts, one model includes a conductive dyke embedded in a homogeneous half-space placed under the receiver spread to explain the measured data. The data fit can probably be improved by the proposed inversion method. Besides, it could be elaborated, that the joint inversion of different LOTEM transient fields reduce the inherent ambiguity of the problem to a high extend.

As demonstrated by the 2D inversion of the field data from the LOTEM survey of the Arava basin, which was given in section 4.4, the proposed 2D inversion method yields very satisfying results. For the joint interpretation of LOTEM and MT data, a coincident earth model could be found which is also in good agreement to results from seismic investigations. As I am not aware of successful 2D inversion attempts with real LOTEM field data in the geophysical literature, it shall be emphasized that for the first time a combined LOTEM data set using the complete information of 168 receiver sites could be inverted for a 2D model.

To gain insight into the resolution capabilities of the a posteriori model I showed two different methods yielding approximately the same results: Firstly, by comparing the results which are obtained applying different kind of global regularizations, and secondly by investigating the coverage of the model.

Suppose the amount and coverage of the transient data is sufficient, the most striking advantage of the 2D inversion scheme is, that the non linear optimization problem be set up as overdetermined system. The number of model unknowns is generally in orders of some thousands, even for models with a very fine discretization. For common TEM setups along a profile, where the number of receiver sites may be given as some tens to hundreds of measured transients, the overdetermined system may therefore be not easy but manageable to establish.

In contrast, the supply of a data sets with the coverage as demanded by a 3D TEM inversion is not easy to deliver. Because of the logistical effort which comes along with common LOTEM surveys, the capabilities of the inversion code for reconstructing a full 3D conductivity distribution is tested just with synthetic LOTEM and SHOTEM data. The 3D inversion of the artificial LOTEM survey, designed to reveal a conductive anomaly within a deep crustal study, showed that the space-dimensional borders as well as the true resistivity values could be resolved well. For the surface near application, a very shallow TEM sounding was conducted, which depth validity is commonly neglected. It could be demonstrated, how horizontal voltage information can improve a surface near SHOTEM sounding significantly. Moreover, the proposed unconventional field setup, namely the multicomponent and multi-receiver site setup, as given in the last section demands a small logistical effort, compared to a 3D LOTEM setup. Thus, this type of field setup may be used within future applications, to reveal lateral changes of surface near conductivity variations with SHOTEM. From the results that where obtained by the 3D inversion of the synthetic SHOTEM data, it could be observed that the lateral extensions of an anomalous conductive sheet can be revealed.

The convergence progress, investigated for the proposed 3D TEM inverse solution, shows typical features for gradient based minimization schemes. Compared to the 2D inverse solutions, where the second order information is included by solving the normal equations, the gradient based minimization scheme takes roughly double the number of iterations to converge. Especially if one takes the complexity of the inherent conductivity structure into account, the Gauss-Newton based inversion scheme seems far superior. Yet, compared to the large scale TEM inversion schemes presented by Wang et al. [1994]; Newman & Commer [2005]; Commer & Newman [2008], far less iterations where used to decrease the cost. However, the use of more powerful optimization strategies are likely to be beneficial in the large scale 3D TEM inverse problem. While Newton or Quasi-Newton methods will converge in far fewer iterations, which is also reported by Haber et al. [2007], the time required per inversion iteration can be prohibitively expensive. Thus, I would strongly recommend to implement a preconditioned-CG based iterative solver into the 3D inversion scheme for the next years.

Bibliography

Alumbaugh, D. L. & Newman, G. A., 1997. Three-dimensional massively parallel electromagnetic inversion - ii. analysis of a crosswell electromagnetic experiment, *Geophys. J. Int.*, **128**, 355–363.

Alumbaugh, D. L. & Newman, G. A., 2000. Image appraisal for 2-D and 3-D electromagnetic inversion, *Geophysics*, **65**, 1455–1467.

Anderson, W. L., 1989. A hybrid fast Hankel transform algorithm for electromagnetic modeling, *Geophysics*, **54**(2), 263–266.

Auken, E. & Christensen, A. V., 2004. Layered and laterally constrained 2D inversion of resistivity data, *Geophysics*, **69**, 752–761.

Backus, G. & Gilbert, F., 1968. The resolving power of gross earth data, *Geophys. Journ. of the Royal Astron. Soc.*, **16**, 169–205.

Backus, G. & Gilbert, F., 1970. Uniqueness in the inversion of gross earth data, *Phil. Trans. of the Royal Soc., Ser. A, Math. and Phys. Sciences*, **266**, 123–192.

Baños, A., 1966. *Dipole radiation in the presence of a conducting half-space*, Pergamon Press, Inc.

Boerner, D. E. & West, G. F., 1989. A generalized representation of the electromagnetic fields in a layered earth, *Geophys. J.*, **97**, 529–548.

Chave, A. D., 1984. The fréchet derivatives of electromagnetic induction, *J. Geophys. Res.*, **89**, 3373–3380.

Commer, M., 2003. *Three-dimensional inversion of transient electromagnetic data: A comparative study*, Ph.D. thesis, Universität zu Köln, Institut für Geophysik und Meteorologie.

Commer, M. & Newman, G. A., 2004. A parallel finite-difference approach for 3d transient electromagnetic modeling with galvanic sources, *Geophysics*, **69**(5), 1192–1202.

Commer, M. & Newman, G. A., 2008. New advances in three-dimensional controlled-source electromagnetic inversion, *Geophys. J. Int.*, **172**(2), 513–535.

Constable, S. C., Parker, R. L., & Constable, C. G., 1987. Occam's inversion: a practical algorithm for generating smooth models from em sounding data, *Geophysics*, **52**, 289–300.

Dennis, J. & Schnabel, R., 1996. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia.

DESERT Working Group, 2004. The crustal structure of the Dead Sea Transform, *Geophys. J. Int.*, **156**, 655–681.

Druskin, V. L. & Knizhnermann, L. A., 1988. A spektral semi-discrete method for the numerical solution of 3d nonstationary problems in electrical prospecting, *Phys. Solid Earth*, **24**, 641–648.

Druskin, V. L. & Knizhnermann, L. A., 1994. Spectral approach to solving three-dimensional maxwell's diffusion in the time and frequency domains, *Radio Science*, **29**, 937–953.

Druskin, V. L. & Knizhnermann, L. A., 1999. New spectral lanczos decomposition method for induction modeling in arbitrary 3-d geometry, *Geophysics*, **64**, 701–706.

Druskin, V. L. & Knizhnermann, L. A., 2000. *User's guide for the program complex to compute 3D nonstationary electromagnetic fields in inhomogenous conductive media*.

Ellis, R. & Oldenburg, D., 1993. Magnetotelluric inversion using green's function and conjugate gradients, *Geophys. J. Int.*.

Farquharson, C. G. & Oldenburg, D. W., 1996. Approximate sensitivities for the electromagnetic inverse problem, *Geophys. J. Int.*, **126**, 235–253.

Farquharson, C. G. & Oldenburg, D. W., 2004. A comparison of automatic techniques for estimating the regularization parameter in non-linear inverse problems, *Geophys. J. Int.*, **156**, 411–425.

Felsen, L. B. & Marcuvitz, N., 1973. *Radiation and Scattering of Waves*, Prentice-Hall, Englewood Cliffs, N. Y.

Gilbert, J. C. & Nocedal, J., 1992. Global convergence properties of conjugate gradient methods for optimization, *SIAM J. Optimization*, **2**(1), 21–42.

Goldman, M., Tabarovsky, L., & Rabinovich, M., 1994. On the influence of 3-D structures in the interpretation of transient electromagnetic sounding data, *Geophysics*, **59**(889-901).

Golub, G. H. & Loan, C. F. V., 1996. *Matrix Computations*, Johns Hopkins, 3rd edn.

Gómez-Treviño, E., 1987. A simple sensitivity analysis of time-domain and frequecy-domain electromagnetic measurements, *Geophysics*, **52**, 1418–1423.

Günther, T., 2004. *Inversion Methods and Resolution Analysis for the 2D/3D Reconstruction of Resistivity Structures from DC Measurements*, Ph.D. thesis, Technischen Universität Bergakademie Freiberg.

Haber, E. & Ascher, U., 2001. Fast finite volume simulation of 3d electromagnetic problems with highly doscontinous coefficients, *SIAM J. Sci. Comput.*, **22**, 1943–1961.

Haber, E., Ascher, U., & Oldenburg, D. W., 2000. On optimization techniques for solving nonlinear inverse problems, *Inverse Problems*, **16**, 1263–1280.

Haber, E., Oldenburg, D. W., & Shekhtman, R., 2007. Inversion of time domain three-dimensional electromagnetic data, *Geophys. J. Int.*, **171**, 550–564.

Hansen, P. C., 1992. Analysis of discrete ill-posed problems by means of l-curve, *SIAM Reviews*, **34**, 561–580.

Hanstein, T., 1992. Iterative und parametrisierte Dekonvolution für LOTEM Daten, in *Protokoll über das 14. Kolloquium Elektromagnetische Tiefenforschung*, pp. 163–172, Dt. Geophys. Gesellschaft.

Helwig, S. L., 2000. *VIBROTEM Ein Vergleich zu Long Offset Transient Electromagnetics in Theorie und Praxis*, Ph.D. thesis, Universität zu Köln, Institut für Geophysik und Meteorologie.

Hestenes, M. R. & Stiefel, E., 1952. Methods of Conjugate Gradients for Solving Linear Systems, *Jrn. Res. Nat. Bur. Stand.*, **49**.

Hohmann, G. W., 1975. Three-dimensional induced polarization and electromagnetic modeling, *Geophysics*, **40**, 309–324.

Hohmann, G. W., 1988. Electromagnetic methods in applied geophysics, chap. Numerical Modeling for Electromagnetic Methods of Geophysics, pp. 314–363, ed. Nabighian, M. N., Soc. Expl. Geophys.

Hördt, A., 1992. *Interpretation transient elektromagnetischer Tiefensondierungen für anisotrop horizontal geschichtete und für dreidimensionale Leitfähigkeitsstrukturen*, Ph.D. thesis, Universität zu Köln, Institut für Geophysik und Meteorologie.

Hördt, A., 1998. Calculation of electromagnetic sensitivities in the time domain, *Geophys. J. Int.*, **133**, 713–720.

Hördt, A., Druskin, V. L., & Knizhnerman, L. A., 1992. Interpretation of 3-D effects on long-offset transient electromagnetic (LOTEM) soundings in the Münsterland area/Germany, *Geophysics*, **57**, 1127–1137.

Hördt, A., Dautel, S., Tezkan, B., & Thern, H., 2000. Interpretation of long-offset transient electromagnetic data from the Odenwald area, Germany, using two-dimensional modelling, *Geophys. J. Int.*.

Johanson, H. K. & Sørensen, K., 1979. Fast hankel transforms, *Geophys. Prospect.*, **27**, 876–901.

Kalscheuer, T. & Pedersen, L. B., 2007. A non-linear truncated SVD variance and resolution analysis of two-dimensional magnetotelluric models, *Geophys. J. Int.*, **169**(2), 435–447.

Karlik, G., 1995. *Eine schnelle und direkte Inversionsmethode für transient elektromagnetische Daten*, Ph.D. thesis, Universität zu Köln, Institut für Geophysik und Meteorologie.

Kaufman, A. A. & Keller, G. V., 1983. *Frequency and transient soundings*, Methods in Geochemistry and Geophysics, Bd. 16, Elsevier.

Keller, G. V., 1987. Rock and Mineral Properties, in *Electromagnetic Methods in Applied Geophysics*, vol. 1, chap. 2, ed. Nabighian, M. N., Soc. Expl. Geophys.

Kemna, A., 2000. *Tomographic inversion of complex resistivity*, Ph.D. thesis, Ruhr-Universität Bochum.

Kilmer, M. E. & O'Leary, D. P., 2001. Choosing regularization parameters in iterative methods for ill-posed problems, *SIAM Journal on matrix analysis and applications*, **22**, 1204–1221.

Koch, O., Helwig, S. L., & Meqbel, N., 2004. Vertical near surface conductivity-anomaly detected at the Dead-Sea-Transform, in *17th international workshop on electromagnetic industion in the Earth, Poster*.

Lanczos, C., 1950. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators, *J.Res.Nat.Bur.Standards*, **45**, 225–280.

Lanczos, C., 1960. *Linear Differential Operators*, D. van Nostrand.

Levenberg, K., 1944. A method for the solution of certain nonlinear porblems in least squares, *Quart. Appl. Math.*, **2**, 164–168.

Li, Y. & Oldenburgh, D. W., 2003. Fast inversion of large-scale magnetic data using wavelet transforms and a logarithmic barrier method, *Geophys. J. Int.*, **152**, 251–265.

Liu, D. C. & Nocedal, J., 1989. On the limited memory bfgs method for large scale optimization, *Mathematical Programming*, **45**, 503–528.

Loewenthal, D., Lu, L., Robertson, R., & Sherwood, J. W. C., 1976. The wave equation applied to migration, *Geophys. Propect.*, **24**, 380–399.

Löhken, J., 2007. *Analytische Berechnung, Finite Elemente Simulation und Inversion von Metalldetektorsignalen im Zeit- und Frequenzbereich. Untersuchungen zur Reduktion der Fehlalarmrate bei der Landminensuche.*, Ph.D. thesis, Universität zu Köln, Institut für Geophysik und Meteorologie.

Loke, M. H. & Barker, R. D., 1996. Rapid least-squares inversion of apparent resistivity pseudosections by a quasi-newton method, *Geophys. Prospect.*, **44**, 131–152.

Mackie, R. & Madden, T., 1993. Three dimensional magnetotelluric inversion using conjugate gradients, *Geophys. J. Int.*, **115**, 215–229.

Madden, T., 1990. *Oceanographic and Geopghysical Tomography*, chap. Inversion of low frequency electromagnetic data, pp. 379–408, North Holland, New Amsterdam.

Madden, T. & Mackie, R., 1989. Three dimensional magnetotelluric modeling and inversion, *Proc. Inst. Electron. Electric. Eng.*, **77**, 318–333.

Marquardt, D., 1963. An algorithm for least squares estimation of non-linear parameters, *SIAM J. Sci. Stat. Comput.*, **11**, 431–441.

Martin, R., 2003. *Realisierung von laufzeitoptimierten Methoden zur Bestimmung der 3D-TEM-Jacobimatrix.*, Master's thesis, Universität zu Köln, Institut für Geophysik und Meteorologie.

Martin, R. & Hördt, A., 2001. Approximierte Sensitivität für TEM, in *Elektromagnetische Tiefenforschung, 19. Kolloquium*, Dt. Geophys. Gesellschaft.

Martin, R., Scholl, C., Helwig, S. L., & Hördt, A., 2003. Sensitivitätsberechung mit adjungierten Green'schen Funktionen für eine mehrdimensionale TEM-Inversion auf Linux-Clustern., in *Elektromagnetische Tiefenforschung, 20. Kolloquium*, Dt. Geophys. Gesellschaft.

McGillivray, P. & Oldenburg, D., 1990. Methods for calculating fréchet derivatives and sensitvities for the non-linear inverse problem: a comparative study, *Geophys. Prospect.*, **38**, 499–524.

McGillivray, P., Oldenburg, D., Ellis, R., & Habashy, T., 1994. Calculation of sensitivities for the frequency-domain electromagnetic problem, *Geophys. J. Int.*, **116**, 1–4.

Meju, M. A., 1994. *Geophysical Data Analysis: Understanding Inverse Problem in Theory and Practice*, Soc. Expl. Geophys.

Menke, W., 1984. *Geophysical data analysis: discrete inverse theory*, Academic Press inc.

Miller, C. R. & Routh, P. S., 2007. Resolution analysis of geophysical images: Comparison between point spread function and region of data influence measures, *Geophys. Prospect.*, **55**, 835–852.

Mitsuhata, Y., 2000. 2-d electromagnetic modeling by finite element method with a dipole source and topography, *Geophysics*, **65**(2), 465–475.

Mitsuhata, Y., Uchida, T., & Amano, H., 2002. 2.5-d inversion of frequency-domain electromagnetic data generated by a grounded-wire source, *Geophysics*, **67**, 1753–17687.

Moler, C. & van Loan, C., 1978. Nineteen dubious ways to compute the exponentials of a matrix, *SIAM Review*, **20**, 801–835.

Morse, P. & Feshbach, H., 1963. *Methods of Theoretical Physics*, McGraw-Hill, New York, NY.

Mosegaard, K. & Sambridge, M., 2005. Monte Carlo analysis of inverse problems, *Inverse Problems*, **18**(B7), R29–R54.

Mosegaard, K. & Tarantola, A., 1995. Monte Carlo sampling of solutions to inverse problems, *J. Geophys. Res.*, **100**(B7), 12431–12447.

Nabighian, M. N., 1979. Quasi-static transient response of a conducting half-space — an approximate representation, *Geophysics*, **44**(10), 1700–1705.

Nabighian, M. N. & Macnae, J. C., 1991. Time Domain Electromagnetic Prospecting Methods, in *Electromagnetic Methods in Applied Geophysics*, vol. 2, chap. 6, ed. Nabighian, M. N., Soc. Expl. Geophys.

Newman, G. A. & Alumbaugh, D. L., 1997. Three-dimensional massivel parallel electromagnetic inversion–i. theory, *Geophys. J. Int.*, **128**, 345–354.

Newman, G. A. & Commer, M., 2005. New advances in three dimensional transient electromagnetic inversion, *Geophys. J. Int.*, **160**, 5–32.

Newman, G. A., Hohmann, G. W., & Anderson, W. L., 1986. Transient electromagnetic response of a three-dimensional body in a layered earth, *Geophysics*, **51**(8), 1608–1627.

Oldenburg, D. W., 1990. Inversion of electromagnetic data: an overview of new tchniques, *Surv. Geophys.*, **11**, 231–270.

Oldenburg, D. W. & Li, Y., 1999. Estimating the depth of investigation in DC resistivity and IP surveys, *Geophysics*, **64**, 403–416.

Oldenburg, D. W., McGillivray, P. R., & Ellis, R. G., 1993. Generalized subspace methods for large-scale inverse problems, *Geophys. J. Int.*, **114**, 12–20.

Paige, C. C. & Saunders, M. A., 1982. LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares, *ACM Transactions on Mathematical Software*, **8**(1), 43–71.

Palacky, G. J., 1988. Resistivity characteristics of geologic targets, in *Electromagnetic Methods in Applied Geophysics*, vol. 1, chap. 3, pp. 106–121, Soc. Expl. Geophys.

Papoulis, A., 1962. *The fourier integral and its applications*, McGraw-Hill.

Parker, R. L., 1977a. Understanding inverse theory, *Ann. Rev. Earth. Plan. Sci.*, **5**, 35–64.

Parker, R. L., 1977b. The Fréchet derivative for the one-dimensional electromagnetic induction problem, *Geophys. J. Roy. Astr. Sot.*, **49**, 543–547.

Parlett, B. N., 1980. *The symmetric eigenvalue problem*, Prentice-Hall,Inc.,Eaglewood Cliffs.

Petrat, L., 1996. *Zweidimensionale Inversion von Long Offset Transient Electromagnetics-Daten*, Master's thesis, Universität zu Köln, Institut für Geophysik und Meteorologie.

Portniaguine, O. & Zhdanov, M. S., 1999. Focusing geophysical inversion images, *Geophysics*, **64**(3), 874–887.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P., 1995. *Numerical Recipies*, Cambridge University Press, 2nd edn.

Richtmeyr, R. D. & Morton, K. W., 1967. *Difference methods for initial value problems*, no. 30 in Methods in Geochemistry and Geophysics, John Wiley and Sons, New York.

Ritter, O., Ryberg, T., Weckmann, U., Hoffmann-Rothe, A., Abueladas, A., & Garfunkel, Z., 2003. Geophysical images of the Dead Sea Transform in Jordan reveal an impermeable barrier for fluid flow, *Geophys. Res. Lett.*, **30**(14), 1741.

Roach, G., 1982. *Green's Functions*, Cambridge University Press.

Rodi, W. & Mackie, R. L., 2001. Nonlinear conjugate gradients algorithm for 2D magnetotelluric inversion, *Geophysics*, **66**(1), 174–187.

Saad, Y., 1996. *Iterative Methods for Sparse Linear Systems*, PWS, Boston.

Sambridge, M., 1999. Geophysical inversion with a neighbourhood algorithm - i. Searching a parameter space, *Geophys. J. Int.*, **138**, 479–494.

Sarma, D. G., Maru, V. M., & Varadarajan, G., 1976. An improved pulse transient airborne electromagnetic system for locating good conductors, *Geophysics*, **41**, 287–299.

Schaumann, G., 2001. *Transientelektromagnetische Messungen auf Mülldeponien - Untersuchung des Einflusses von 3D-Leitfähigkeitsvarationen und 1D-frequenzabhängiger Polarisierbarkeit*, Ph.D. thesis, TU-Braunschweig.

Schneider, T., 2000. *Zeitlich optimierte Sensitivitätsberechnung für "Long-Offset Transient Electromagnetics" LOTEM unter Verwendung des Reziprozitätsgesetz*, Master's thesis, Universität zu Köln, Institut für Geophysik und Meteorologie.

Scholl, C., 2005. *The influence of multidimensional structures on the interpretation of LOTEM data with one-dimensional models and the application to data from Israel*, Ph.D. thesis, Universität zu Köln, Institut für Geophysik und Meteorologie.

Scholl, C., Martin, R., Commer, M., Helwig, S. L., & Tezkan, B., 2004. 2D-Inversion von LOTEM-Daten, in *Protokoll über das 20. Kolloquium Elektromagnetische Tiefenforschung*, Dt. Geophys. Gesellschaft.

Shewchuk, J. R., 1994. An introduction to the conjugate gradient method without the agonizing pain, ftp://warp.ce.cme.edu.

Siripunvaraporn, W. & Egbert, G., 2000. An effecient data-subspace inversion method for 2-d magnetotelluric data, *Geophysics*, **65**(3), 791–803.

Spies, B. R., 1983. Recent developments in the use of surface electical methods for oil and gas exploration in the Soviet Union, *Geophysics*, **48**, 1102–1112.

Spies, B. R. & Frischknecht, F. C., 1991. Electromagnetic Sounding, in *Electromagnetic methods in applied geophysics*, vol. 2, chap. 5, ed. Nabighian, M. N., Soc. Expl. Geophys.

Spies, B. R. & Habashy, T. M., 1995. Sensitivity analysis of crosswell electromagnetics, *Geophysics*, **60**(3), 834–845.

Stoffa, P. L. & Sen, M. K., 1991. Nonlinear multiparameter optimization using genetic algorithms: Inversion of plane-wave seismograms, *Geophysics*, **56**(11), 1794–1810.

Strack, K. M., 1985. *Das Transient-Elektromagnetische Tiefensondierungsverfahren angewandt auf die Kohlenwasserstoff- und Geothermie-Exploration*, Ph.D. thesis, Universität zu Köln, Institut für Geophysik und Meteorologie.

Strack, K. M., 1992. *Exploration with deep transient electromagnetics*, Methods in Geochemistry and Geophysics, Bd. 30, Elsevier, Amsterdam.

Tarantola, A., 1987. *Inverse problem theory, methods for data fitting and model parameter estimation*, Elsevier, Amsterdam.

Tarantola, A., 2005. *Inverse problem theory and methods for model parameter estimation*, SIAM, Philadelphia.

Tezkan, B., 1999. A review of environmental applications of quasi-stationary electromagnetic techniques, *Surveys in Geophysics*, **20**, 279–308.

Tezkan, B., Goldman, M., Greinwald, S., Hördt, A., Müller, I., Neubauer, F., & Zacher, G., 1996. A joint application of radiomagentotellurics and transient electromagnetics to the investigation of waste deposit in cologne (germany), *J. Appl. Geophys.*, **34**, 199–212.

Tikhonov, A. N. & Arsenin, V. A., 1977. *Solution of Ill-posed Problems*, Winston & Sons, Washington.

Unsworth, M. J., Egbert, G., & Booker, J., 1999. High-resolution electromagnetic imaging of the San Andreas fault in Central California, *J. Geophys. Res.*, **105**, 1131–1150.

Viezzoli, A., Christensen, A., Auken, E., & Sørensen, K., 2008. Quasi-3D modeling of airborne TEM data by spatially constrained inversion, *Geophysics*, **73**, 105–113.

Vogel, C. R., 2002. *Computational Methods for Inverse Problems*, Society for Industrial and Applied Mathematics.

Vozoff, K. & Jupp, D. L. B., 1975. Joint inversion of geophysical data, *J. R. astr. Soc.*, **42**, 977–991.

Wang, T. & Hohmann, G. W., 1993. A finite-difference, time-domain solution for three-dimensional electromagnetic modeling, *Geophysics*, **58**, 797–809.

Wang, T., Oristaglio, M., Tripp, A., & Hohmann, G. W., 1994. Inversion of diffusive transient electromagnetic data using conjugate-gradient method, *Radio Science*, **29**, 1143–1156.

Ward, S. H. & Hohmann, G. W., 1987. Electromagnetic Theory for Geophysical Exploration, in *Electromagnetic Methods in Applied Geophysics*, vol. 1, chap. 4, ed. Nabighian, M. N., Soc. Expl. Geophys.

Ward, S. H. & Hohmann, G. W., 1988. in *Electromagnetic methods in applied geophysics*, vol. 1, chap. 4, pp. 131–311, ed. Nabighian, M. N., Soc. Expl. Geophys.

Weidelt, P., 1972. The inverse problem of geomagnetic induction, *Geophysics*, **38**, 257–289.

Weidelt, P., 1975. Electromagnetic Induction in Three-Dimensional Structures, *J. Geophys.*, **41**, 85–109.

Weidelt, P., 1986. *Einführung in die elektromagnetische Tiefenforschung*, Lecture Technische Universät Braunschweig.

Weidelt, P., 2000. Numerical modelling of transient-electromagentic fields in three-dimensional conductors: A comparative study, in *Elektromagnetische Tiefenforschung, 18. Kolloquium*, pp. 216–231, Dt. Geophys. Gesellschaft.

Westlake, J. R., 1968. *A Handbook of Numerical Matrix Inversion and Solution of Linear Equations*, Wiley, New York.

Whaba, G., 1990. *Spline Models for Observational Data*, SIAM, Philadelphia.

Wolfe, P., 1969. Convergence conditions for ascent methods, *SIAM Rev.*, **11**(11), 226–235.

Wolfe, P., 1971. Convergence conditions for ascent methods ii: some corrections, *SIAM Rev.*, **13**(13), 185–188.

Yee, K. S., 1966. Numerical solutions of initial boundary problems involving maxwell's equations in isotropis media, *IEEE Trans. Antennas Propag.*, **AP-14**, 302–309.

Zhang, L., Zhou, W., & Li, D. H., 2006. Global convergence of a modified Fletcher-Reeves conjugate gradient method with Armijo-type line search, *Numerische Mathematik*, **10**, 561–572.

Zhdanov, M., 2002. *Geophysical Inverse Theory and Regularization Problems*, vol. 36 of **Methods in Geochemistry and Geophysics**, Elsevier.

Zhdanov, M. S. & Portniaguine, O., 1997. Time-domain electromagnetic migration in the solution of inverse problems, *Geophys. J. Int.*, **131**(4), 293–309.

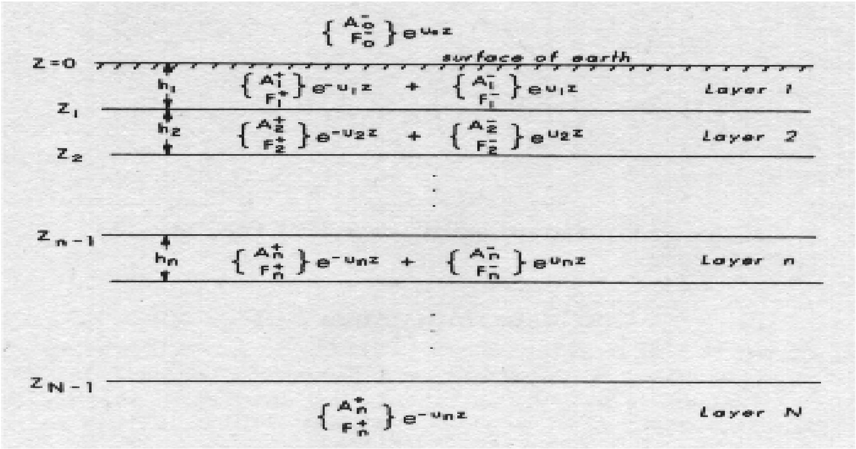
Zhdanov, M. S. & Tolstaya, E., 2006. A novel approach to the model appraisal and resolution analysis of regularized geophysical inversion, *Geophysics*, **71**(6), R79–R90.

APPENDIX A

1D adjoint fields

A.0.5 TE- TM-mode calculation for a layered earth

For a layered conductivity model of N-layers we can formulate $f_{e,m}$ in the wavenumber-frequency-domain as sums of exponential factors (see Fig. A.1). The conductivities are given as $\sigma_1, \ldots, \sigma_N$, the thickness of each layer is $h_1, \ldots, h_{N-1}$ and the layer boundary's denoted like $Z_1, \ldots, Z_{N-1}$. In this case f_e and f_m comply the same PDEs but have different continuity relations. f_e and $\partial_z f_e$ resp. σf_m and $\partial_z f_m$ are continuous at layer boundaries.



$\left\{ \begin{matrix} A_0^- \\ F_0^- \end{matrix} \right\} e^{u_0 z}$
z=0
surface of earth
h_1
$\left\{ \begin{matrix} A_1^+ \\ F_1^+ \end{matrix} \right\} e^{-u_1 z} + \left\{ \begin{matrix} A_1^- \\ F_1^- \end{matrix} \right\} e^{u_1 z}$
Layer 1
Z_1
h_2
$\left\{ \begin{matrix} A_2^+ \\ F_2^+ \end{matrix} \right\} e^{-u_2 z} + \left\{ \begin{matrix} A_2^- \\ F_2^- \end{matrix} \right\} e^{u_2 z}$
Layer 2
Z_2
Z_{n-1}
h_n
$\left\{ \begin{matrix} A_n^+ \\ F_n^+ \end{matrix} \right\} e^{-u_n z} + \left\{ \begin{matrix} A_n^- \\ F_n^- \end{matrix} \right\} e^{u_n z}$
Layer n
Z_{N-1}
$\left\{ \begin{matrix} A_n^+ \\ F_n^+ \end{matrix} \right\} e^{-u_n z}$
Layer N

Figure A.1: *Schematic view of the recursive calculation of EM fields in a layered subsurface. The excited EM field at the surface is damped through the conductive earth. The attenuation is described as outgoing and incoming partial waves with different impedance's for every layer (from Ward & Hohmann [1988]).*

For the recursive calculation of $f_{e,m}(z)$ we also need to know about the impedances for the

TE- and TM-mode

$$\begin{aligned} A(z,k,\omega) = & \tfrac{E_{e,x}}{H_{e,y}} = & i\omega\mu\left(-\frac{f_e(z,k,\omega)}{\partial_z f_e(z,k,\omega)}\right) \end{aligned} \tag{A.1}$$

$$\begin{aligned} F(z,k,\omega) = & \tfrac{E_{m,x}}{H_{m,y}} = & \left[\sigma\left(-\frac{f_m(z,k,\omega)}{\partial_z f_m(z,k,\omega)}\right)\right] . \end{aligned} \tag{A.2}$$

In the homogeneous n'th layer $(Z_n < z < Z_{n+1})$ f_e and f_m fulfill the same PDE

$$\partial_z^2 f_{e,m}(z) = u_n^2 f_{e,m}(z) \tag{A.3}$$

with $u_n^2 := k^2 + i\omega\mu\sigma_n$.
They are solved via

$$\left\{ \begin{array}{c} f_e \\ f_m \end{array} \right\}(z) = \left\{ \begin{array}{c} A_n^+ \\ F_n^+ \end{array} \right\} e^{-u_n(z-h_n)} + \left\{ \begin{array}{c} A_n^- \\ F_n^- \end{array} \right\} e^{u_n(z-h_n)} , \tag{A.4}$$

which is describing an outgoing $(e^{-u_n(z-h_n)})$ and incoming $(e^{u_n(z-h_n)})$ wave in every layer. Beginning at the bottom layer, with the boundary condition that the potentials should vanish at infinity ($f_{e,m} \to 0$ for $z \to \infty$) , we find the initial values for our calculations:

$$\left\{ \begin{array}{c} f_e \\ f_m \end{array} \right\}(z) = \left\{ \begin{array}{c} A_M^+ \\ F_M^+ \end{array} \right\} e^{-u_n(z-h_n)} , \quad z \geq h_M . \tag{A.5}$$

As is shown by Weidelt [1986], we can use the modified impedance A_n for the computation of the TE- and F_n for the TM-mode, respectively. They are defined by the ratios

$$A_n := -\frac{\partial_z f_e(h_n)}{f_e(h_n)} , \qquad F_n := -\frac{\partial_z f_m(h_n)}{f_m(h_n)} . \tag{A.6}$$

With (A.6) and an initial value for the underlying halfspace one can give recursive formulas for the modified impedances of every layer from the bottom to the top (see algorithm 4). Note that in algorithm 4 the variable s_n accounts for the different continuities of the TM-mode which is necessary to include due to the fact, that the vertical component of the magnetic field is continous at the layer boundary

Algorithm 4 Recursion algorithm for modified impedance's A_n, F_n

[ht!]

```
Require: $\omega > 0$, $k \in \mathbb{R}$, $\sigma_n \in \mathbb{R} > 0$ , $n = 1, \ldots, N$
  $A_N = F_N = u_N$
  for $n = N - 1$ to 1 do
    $u_n = \sqrt{k^2 + i\omega\mu\sigma_n}$
    $s_n = \frac{\sigma_{n+1}}{\sigma_n}$
    $A_n = u_n \frac{A_{n+1} + u_n \tanh(u_n h_n)}{u_n + A_{n+1} \tanh(u_n h_n)}$
    $F_n = u_n \frac{F_{n+1} + u_n s_n \tanh(u_n h_n)}{u_n s_n + F_{n+1} \tanh(u_n h_n)}$
  end for
```

$$\frac{H_z(h_{n+1})}{H_z(h_n)} = \frac{f_e(h_{n+1})}{f_e(h_n)} = \frac{\partial_z f_m(h_{n+1})}{\partial_z f_m(h_n)} , \tag{A.7}$$

as well as the vertical component of the current density jumps about the ratio $\frac{\sigma_{n+1}}{\sigma_n}$

$$\frac{J_z(h_{n+1})}{J_z(h_n)} = \frac{\sigma_{n+1} f_m(h_{n+1})}{\sigma_n f_m(h_n)} = \frac{\partial_z f_e(h_{n+1})}{\partial_z f_e(h_n)} \; . \tag{A.8}$$

With (A.7) and (A.8) one can formulate continuity relations for $f_{e,m}$ at the layer boundaries $n \rightarrow n+1$

$$f_e(h_{n+1}) = \quad f_e(h_n) \tfrac{u_n + A_n}{u_n + A_{n+1}} e^{-u_n h_n} \tag{A.9}$$

$$f_m(h_{n+1}) = \quad f_m(h_n) \tfrac{u_n + F_n}{u_n s_n + F_{n+1}} e^{-u_n h_n} \tag{A.10}$$

and the corresponding relations for the partial derivatives with respect to z

$$\partial_z f_e(h_{n+1}) = \quad \partial_z f_e(h_n) \tfrac{1 + u_n A_n^{-1}}{1 + u_n A_{n+1}^{-1}} e^{-u_n h_n} \tag{A.11}$$

$$\partial_z f_m(h_{n+1}) = \quad \partial_z f_m(h_n) \tfrac{1 + u_n F_n^{-1}}{1 + u_n s_n F_{n+1}^{-1}} e^{-u_n h_n} \tag{A.12}$$

as well. Because the exponents with positive realparts are numerically unfavorable one may express them in terms of negative realparts. This is achieved by the symmetrical relation

$$\frac{A_{n+1} + u_n}{A_n + u_n} e^{u_n h_n} = \frac{A_{n+1} - u_n}{A_n - u_n} e^{-u_n h_n} \; . \tag{A.13}$$

With the symmetric relation, the attenuation for the partial waves in the n'th layer ($Z_n \leq z \leq Z_{n+1}$, $1 \leq n \leq M-1$) is given by

$$f_e(z) = \quad f_e(h_n) \tfrac{1}{2} \left(1 + \tfrac{A_n}{u_n}\right) \left[a^+ - \tfrac{A_{n+1} - u_n}{A_{n+1} + u_n} a^-\right] \tag{A.14}$$

$$f_m(z) = \quad f_m(h_n) \tfrac{1}{2} \left(1 + \tfrac{F_n}{u_n}\right) \left[a^+ - \tfrac{F_{n+1} - u_n s_n}{A_{n+1} + u_n s_n} a^-\right] \tag{A.15}$$

$$\partial_z f_e(z) = \quad \partial_z f_e(h_n) \tfrac{1}{2} \left(1 + \tfrac{u_n}{A_n}\right) \left[a^+ + \tfrac{1 - u_n A_{n+1}^{-1}}{1 + u_n A_{n+1}^{-1}} a^-\right] \tag{A.16}$$

$$\partial_z f_m(z) = \quad \partial_z f_m(h_n) \tfrac{1}{2} \left(1 + \tfrac{u_n}{F_n}\right) \left[a^+ + \tfrac{1 - u_n s_n F_{n+1}^{-1}}{1 + u_n s_n F_{n+1}^{-1}} a^-\right] \tag{A.17}$$

with $a^+ = e^{-u_n(z - Z_n)}$ and $a^- = e^{-u_n(Z_{n+1} - z)}$.

Summarizing this into a pseudo code, we arrive at algorithm 5, which allows us to compute the relative damping of the initial amplitudes at the depth of interest.

For the underlying halfspace applies

$$f_{e,m}(z) = \quad f_{e,m}(h_M) e^{-u_n(z - h_M)} \tag{A.18}$$

$$\partial_z f_{e,m}(z) = \quad \partial_z f_{e,m}(h_M) e^{-u_n(z - h_M)} \qquad , z \geq h_M \; , \tag{A.19}$$

which can already be seen from equation (A.5).

Algorithm 5 $f_{e,m}$ and $\partial_z f_{e,m} = df_{e,m}$ as product of partial waves

```
[H]
Require: $0 \leq Z_{m-1} < z \leq Z_m$ and $m > 1$
Require: $f_e = A_0 \frac{1}{(k+A_1)}$; $f_m = -M_0 \frac{1}{\sigma_1}$
Require: $df_e = -f_e A_1$; $df_m = -f_m F_1$
  if $m > 1$ then
    for $n = 1$ to $m - 1$ do
      $a^+ = e^{-u_n h_n}$
      $f_e = f_e \frac{u_n + A_n}{u_n + A_{n+1}} a^+$
      $f_m = f_m \frac{u_n + F_n}{u_n s_n + F_{n+1}} a^+$
      $df_e = df_e \frac{1 + u_n A_n^{-1}}{1 + u_n A_{n+1}^{-1}} a^+$
      $df_m = df_m \frac{1 + u_n F_n^{-1}}{1 + u_n s_n F_{n+1}^{-1}} a^+$
    end for
  end if
  $a^+ = e^{-u_n(z - Z_m)}$
  if $m < N$ then
    $a^- = e^{-u_n(Z_{m+1} - z)}$
    $f_e = f_e \frac{1}{2} \left(1 + \frac{A_n}{u_n}\right) \left[a^+ - \frac{A_{n+1} - u_n}{A_{n+1} + u_n} a^-\right]$
    $f_m = f_m \frac{1}{2} \left(1 + \frac{F_n}{u_n}\right) \left[a^+ - \frac{F_{n+1} - u_n s_n}{A_{n+1} + u_n s_n} a^-\right]$
    $df_e = df_e \frac{1}{2} \left(1 + \frac{u_n}{A_n}\right) \left[a^+ + \frac{1 - u_n A_{n+1}^{-1}}{1 + u_n A_{n+1}^{-1}} a^-\right]$
    $df_m = df_m \frac{1}{2} \left(1 + \frac{u_n}{F_n}\right) \left[a^+ + \frac{1 - u_n s_n F_{n+1}^{-1}}{1 + u_n s_n F_{n+1}^{-1}} a^-\right]$
  end if
  $f_e = f_e a^+$
  $df_e = df_e a^+$
  $f_m = f_m a^+$
  $df_m = df_m a^+$
```

A.0.6 Particular solutions for EM field potentials of inductive and galvanic sources over layered media

Up to now, the field components are normalized onto their surface values. This is probably useful for the modeling of plane wave incident fields. In this section, the source field is generated by external inductive or galvanic potential fields.

To take the external source into account, one adds the particular solution of the inhomogeneous PDE ($f^p_{e,m}$) to the complementary solution of the homogeneous problem, in the layer containing the source [Ward & Hohmann, 1988]:

$$\phi_{e,m}(\mathbf{r}, \omega) = \frac{1}{2\pi} \int_0^\infty f^p_{e,m} e^{-k(z+h)} f_{e,m}(z, k, \omega) J_0(kr) k \, \mathrm{d}k \, , \tag{A.20}$$

with $h > 0$ given as the height of the source above the surface.

Note that, if the potentials are formulated, one can compute the EM field components in the frequency domain from equations (3.9) and (3.10). This can be achieved either by calculating the potentials first, using a fast Hankel transform to evaluate the Bessel integrals and make the derivation after equation (3.9) and (3.10) thereafter, or one differentiates first and makes the Hankel transform later.

The particular solution for the potentials of the inhomogeneous PDE to various sources, as derived by [Weidelt, 1986], are given in the following.

For simplification I use the definitions

$$\begin{aligned} B_{e,m} &= -\frac{\partial_z f_{e,m}(0)}{f_{e,m}(0)} \\ \gamma(k,h) &:= \left\{1 - \frac{B_e - k}{B_e + k}\right\} e^{-k(z-h)} \, . \end{aligned} \tag{A.21}$$

Vertical magnetic dipole (VMD)

Due to the symmetry of the problem, currents flow only horizontally. Hence, there is no TM-mode at any depth and the electric field of a vertical magnetic dipole has a simple r^{-1}-TE-potential. With magnetic moment $\widetilde{\mathbf{M}}(\omega)$ and $R^2 = (z+h)^2 + r^2$, the relation for the magnetic field is given by

$$\mathbf{h}(\mathbf{r},\omega) = \frac{1}{\mu}\nabla\partial_z\phi_e(\mathbf{r},\omega) = \frac{\widetilde{\mathbf{M}}(\omega)}{4\pi}\nabla\partial_z\frac{1}{R} \, . \tag{A.22}$$

Applying the the Weber-integral

$$\int_0^\infty e^{-ak} J_0(bk) \, \mathrm{d}k = \frac{1}{\sqrt{a^2+b^2}} \quad a \geq 0 \tag{A.23}$$

one can express the particular solution of the VMD-TE-potential between the dipole and the earth according to

$$\phi_e^{\mathsf{VMD}}(\mathbf{r},z,\omega) = \frac{\mu\widetilde{\mathbf{M}}(\omega)}{4\pi}\int_0^\infty e^{-k(z+h)} J_0(kr) \, \mathrm{d}k \, . \tag{A.24}$$

The relation given above is valid within the space above the surface ($z \leq 0$) for the source located at height $h > 0$.

If the source is located at the surface, the total potential within the medium is given by

$$\phi_e^{\mathsf{VMD}}(\mathbf{r},z,\omega) = \frac{\mu\widetilde{\mathbf{M}}(\omega)}{4\pi}\int_0^\infty \left\{e^{-k(z+h)} - \gamma(k,h)\right\} f_e(z,k,\omega) J_0(kr) \, \mathrm{d}k \, , \tag{A.25}$$

Large horizontal loop

The derivation of the potential which arise from a large horizontal loop with radius A, is similar to the one from the VMD. It can be derived with the substitution

$$\widetilde{\mathbf{M}}(\omega) \rightarrow \frac{2AI(\omega)}{k} J_1(kA) \, . \tag{A.26}$$

The potential of the loop than reads

$$\phi_e^{\mathrm{Loop}}(\mathbf{r},z,\omega) = \frac{\mu I(\omega)}{2\pi}\int_0^\infty \left\{e^{-k(z+h)} - \gamma(k,h)\right\} f_e(z,k,\omega) J_1(ka) J_0(kr) \, \mathrm{d}k \, , \tag{A.27}$$

With $A \rightarrow 0$ one can notice that $J_1(kA) \approx \frac{1}{2}kA$ which is the same potential as for the VMD.

Horizontal magnetic dipole (HMD)

The EM field of a x-oriented horizontal magnetic dipole (HMD_x) with moment $\widetilde{\mathbf{M}}(\omega)$ located at $h > 0$ has both, a TE- and a TM-mode.

The TE-potential is derived through the relation from the VMD, but with rotated coordinates:

$$\mathbf{h}_e(\mathbf{r},\omega) = \frac{1}{\mu}\nabla\partial_z\phi_e(\mathbf{r},\omega) = \frac{\widetilde{\mathbf{M}}(\omega)}{4\pi}\nabla\partial_x\frac{1}{r} \; . \tag{A.28}$$

Comparing either side of (A.28) we find

$$\begin{aligned} \partial_z\phi_e(\mathbf{r},\omega) &= \frac{\mu\widetilde{\mathbf{M}}(\omega)}{4\pi}\partial_x\int_0^\infty \left\{e^{-k(z+h)} - \gamma(k,h)\right\} J_0(kr)\,\mathrm{d}k \\ &= -\frac{\mu\widetilde{\mathbf{M}}(\omega)x}{4\pi r}\int_0^\infty \left\{e^{-k(z+h)} - \gamma(k,h)\right\} J_1(kr)k\,\mathrm{d}k \; . \end{aligned} \tag{A.29}$$

Since the integrand has a simple z-dependence in the layer containing the source, the primitive of (A.29) is found via the deliberation

$$\begin{aligned} \tilde{f}(z) &:= \tilde{f}_0 e^{-k(z+h)} \\ \tilde{F}(z) &= \int_0^z \tilde{f}(z) = -\frac{1}{k}\tilde{f}(z) \; . \end{aligned} \tag{A.30}$$

Thus

$$\phi_e^{\mathsf{HMD}_x}(\mathbf{r},z,\omega) = \frac{\mu\widetilde{\mathbf{M}}(\omega)x}{4\pi r}\int_0^\infty \left\{e^{-k(z+h)} - \gamma(k,h)\right\} f_e(z,k,\omega)J_1(kr)\,\mathrm{d}k \; . \tag{A.31}$$

The TM mode can be derived by rotating the vertical electric field of a VMD and the deliberations

$$\begin{aligned} E_z &= -\frac{i\omega\mu_0\widetilde{\mathbf{M}}(\omega)}{4\pi R^2} \\ (3.9) &= \partial_z^2\phi_m(\mathbf{r},\omega) \\ &\Leftrightarrow \\ \partial_z^2\phi_m^{\mathsf{HMD}_x}(\mathbf{r},z,\omega) &= \frac{i\omega\mu_0\widetilde{\mathbf{M}}(\omega)}{4\pi}\partial_y\int_0^\infty e^{-k(z+h)}J_0(kr)\,\mathrm{d}k \\ &= -\frac{i\omega\mu_0\widetilde{\mathbf{M}}(\omega)y}{4\pi r}\int_0^\infty e^{-k(z+h)}J_1(kr)\frac{1}{k}\,\mathrm{d}k \; . \end{aligned} \tag{A.32}$$

In closed formulation one may choose

$$\phi_m^{\mathsf{HMD}_x}(\mathbf{r},z,\omega) = -\frac{i\omega\mu_0\widetilde{\mathbf{M}}(\omega)y}{4\pi R} = -\frac{i\omega\mu_0\widetilde{\mathbf{M}}(\omega)y}{4\pi\sqrt{r^2+(z+h)^2}} \tag{A.33}$$

The complete potential in $z < 0$ is, because of the boundary condition $\partial_z\phi_m(z = 0) = 0$, achieved by reflection at $z = 0$. However, the TM-mode is only present in the air ($\phi_m = 0$ for $z > 0$).

Horizontal electric dipole (HED)

The HED implies the most complex particular solution. Because it has a non zero vertical magnetic and a non zero vertical electric field, both, the TE- and the TM-mode have contributions to the electrical field within the medium and must be considered.

Because only the TE-mode has a vertical magnetic field we find the TE-potential by applying Biot-Savart to a x-oriented current filament ($\widetilde{\mathbf{D}}(\omega) = I(\omega)\Delta x$) at the surface. From

$$H_z(x, y, 0) = \frac{\widetilde{\mathbf{D}}(\omega)y}{4\pi r^3} = \frac{1}{\mu}\partial^2_{zz}\phi_e(\mathbf{r}, \omega) \tag{A.34}$$

the potential is derived as

$$\phi_e^{\mathsf{HED}_x}(\mathbf{r}, z, \omega) = \frac{\mu\widetilde{\mathbf{D}}(\omega)y}{4\pi r}\int_0^\infty \left\{e^{-k(z+h)} - \gamma(k, h)\right\} J_1(kr)\frac{1}{u_0}\,\mathrm{d}k\,. \tag{A.35}$$

Because only the TM-mode has a vertical electric field, the vertical current component is used for it's derivation. The vertical component of the current density is continuous at the surface boundary and is given from (3.11) as

$$J_z(x, y, 0) = -\sigma_1(\partial^2_{xx} + \partial^2_{yy})\phi_m(r)\,. \tag{A.36}$$

Considering the $x-$oriented current filament at the surface ($h = 0$) with moment $\widetilde{\mathbf{D}}$ one will notice

$$J_z(x, y, 0) = -\widetilde{\mathbf{D}}(\omega)\delta(x)'\delta(y) \Rightarrow J_z(0, k, \omega) = -ik_x\frac{\widetilde{\mathbf{D}}(\omega)}{4\pi^2} \tag{A.37}$$

as well.

Combining (A.36) and (A.37), the relation for the TM-potential reads

$$\begin{aligned}\phi_m^{\mathsf{HED}_x}(\mathbf{r}, z, \omega) &= \frac{\widetilde{\mathbf{D}}(\omega)x}{2\pi\sigma_1 r^2} \\ &= \frac{\widetilde{\mathbf{D}}(\omega)x}{2\pi\sigma_1 r}\int_0^\infty B_h e^{-k(z+h)}J_1(kr)\frac{1}{k}\,\mathrm{d}k\,.\end{aligned} \tag{A.38}$$

A.1 Adjoint field calculations for layered media

The sensitivity calculation of 3D model parameters is done via the adjoint field approach. To compute the adjoint fields, the receiver is acting as a source. To cover the whole range of possibilities of excitation, which are needed for the computation of the adjoint green's functions, we need formulations for the inhomogeneous PDEs of all kind of possible source terms. This is

- vertical magnetic dipoles (VMD) in z-direction, corresponding to the measured vertical magnetic field change $\dot{\mathsf{H}}_\mathrm{z}$.
- horizontal magnetic dipoles (HMD) in x- and y-orientation, according to magnetic induction coils measuring $\dot{\mathsf{H}}_\mathrm{x}$ and $\dot{\mathsf{H}}_\mathrm{y}$.
- horizontal electrical dipoles (HED) in x- and y-direction, corresponding to the electric field sensors E_x and E_y.

A.1.1 Adjoint fields for $\dot{\mathsf{H}}_\mathrm{z}$ receivers

From (3.9) one can derive the equations for the electric field components in the frequency domain:

$$\begin{aligned} E_x(\mathbf{r},\omega) &= -i\omega\partial_y\phi_e(z,r,\omega) \\ E_y(\mathbf{r},\omega) &= i\omega\partial_x\phi_e(z,r,\omega)\ . \end{aligned} \tag{A.39}$$

The x-derivative of the Bessel function $J_0(kr)$ is given by

$$\partial_x J_0(kr) = \partial_x r \partial_r(kr)\partial_v J_0(v) = -\frac{x}{r}kJ_1(kr)\ . \tag{A.40}$$

Thus, the electric field from the VMD reads

$$\begin{aligned} E_x^{\mathsf{VMD}}(z,r,\omega) &= \frac{i\omega\mu\widetilde{\mathbf{M}}(\omega)y}{4\pi r}\int_0^\infty f_e(z,k,\omega)J_1(kr)k\ \mathrm{d}k \\ E_y^{\mathsf{VMD}}(z,r,\omega) &= -\frac{i\omega\mu\widetilde{\mathbf{M}}(\omega)x}{4\pi r}\int_0^\infty f_e(z,k,\omega)J_1(kr)k\ \mathrm{d}k\ . \end{aligned} \tag{A.41}$$

Note that from a VMD excitation, the currents flowing only horizontal within the conductive layers.

Taking the relation of the different receiver moments into account to compute the proper adjoint fields, (3.95) is used to alter (A.41) into

$$\begin{aligned} E_x^{\mathsf{VMD}}(z,r,\omega) &= -\frac{\widetilde{\mathbf{D}}(\omega)y}{4\pi r}\int_0^\infty f_e(z,k,\omega)J_1(kr)k\ \mathrm{d}k \\ E_y^{\mathsf{VMD}}(z,r,\omega) &= \frac{\widetilde{\mathbf{D}}(\omega)x}{4\pi r}\int_0^\infty f_e(z,k,\omega)J_1(kr)k\ \mathrm{d}k\ , \end{aligned} \tag{A.42}$$

which was the used expression for the adjoint electric field calculation of a VMD in the frequency domain.

The proper impulse response of the fields are evaluated during the inverse Fourier transform to the time domain, which is especially addressed in section 3.1.2.

A.1.2 Adjoint fields for $\dot{\mathsf{H}}_\mathrm{x}$ and $\dot{\mathsf{H}}_\mathrm{y}$ receivers

Notifying that

$$\partial_x J_n(x) = -\frac{n}{x}J_n(x) + J_{n-1}(x) \tag{A.43}$$

$$\partial_x\left(\frac{x}{r}kJ_1(kr)\right) = \left(\frac{1}{r}-\frac{x^2}{r^3}\right)kJ_1(kr) + \frac{x^2}{r^2}k\left(J_1(kr)+kJ_0(kr)\right) \tag{A.44}$$

$$\partial_y\left(\frac{x}{r}kJ_1(kr)\right) = -\frac{xy}{r^3}kJ_1(kr) + \frac{xy}{r^2}k\left(J_1(kr)+kJ_0(kr)\right) \tag{A.45}$$

and by applying (A.39) one can calculate the electric fields which arise from a x-oriented HMD as

$$\begin{aligned} E_x^{\mathsf{HMD}_x}(z,r,\omega) &= - \frac{i\omega\mu\widetilde{\mathbf{M}}(\omega)}{4\pi}\left[\left(\frac{1}{r}-\frac{x^2}{r^3}\right)\int_0^\infty f_e(z,k,\omega)J_1(kr)\frac{k}{u_0}\,\mathrm{d}k \right.\\ &\quad \left. +\frac{x^2}{r^2}\int_0^\infty f_e(z,k,\omega)\frac{k}{u_0}\left(\frac{1}{r}J_1(kr)+kJ_0(kr)\right)\,\mathrm{d}k\right] \qquad (A.46)\\ E_y^{\mathsf{HMD}_x}(z,r,\omega) &= \frac{i\omega\mu\widetilde{\mathbf{M}}(\omega)}{4\pi}\left[\frac{xy}{r^3}\int_0^\infty f_e(z,k,\omega)J_1(kr)\frac{k}{u_0}\,\mathrm{d}k \right.\\ &\quad \left. -\frac{xy}{r^2}\int_0^\infty f_e(z,k,\omega)\frac{k}{u_0}\left(\frac{1}{r}J_1(kr)+kJ_0(kr)\right)\,\mathrm{d}k\right] \qquad (A.47)\end{aligned}$$

By applying the reciprocal relations for the transmitting moments, the adjoint electrical field components of a HMD read

$$\begin{aligned} E_x^{\mathsf{HMD}_x}(z,r,\omega) &= \frac{\widetilde{\mathbf{D}}(\omega)}{4\pi}\left[\left(\frac{1}{r}-\frac{x^2}{r^3}\right)\int_0^\infty f_e(z,k,\omega)J_1(kr)\frac{k}{u_0}\,\mathrm{d}k \right.\\ &\quad \left. +\frac{x^2}{r^2}\int_0^\infty f_e(z,k,\omega)\frac{k}{u_0}\left(\frac{1}{r}J_1(kr)+kJ_0(kr)\right)\,\mathrm{d}k\right] \qquad (A.48)\\ E_y^{\mathsf{HMD}_x}(z,r,\omega) &= \frac{\widetilde{\mathbf{D}}(\omega)}{4\pi}\left[\frac{xy}{r^3}\int_0^\infty f_e(z,k,\omega)J_1(kr)\frac{k}{u_0}\,\mathrm{d}k \right.\\ &\quad \left. -\frac{xy}{r^2}\int_0^\infty f_e(z,k,\omega)\frac{k}{u_0}\left(\frac{1}{r}J_1(kr)+kJ_0(kr)\right)\,\mathrm{d}k\right] \qquad (A.49)\end{aligned}$$

A.1.3 Adjoint fields for the E_x and E_y receivers

With (A.35) and (A.39), the TE-mode of the electrical field components can be calculated from

$$\begin{aligned} E_{x,e}^{\mathsf{HED}_x}(\mathbf{r},\omega) &= \frac{\widetilde{\mathbf{D}}(\omega)}{4\pi\mu}\left[\frac{xy}{r^3}\int_0^\infty f_e(z,k,\omega)J_1(kr)\frac{1}{u_0}\,\mathrm{d}k \right.\\ &\quad \left. -\frac{xy}{r^2}\int_0^\infty f_e(z,k,\omega)\frac{1}{u_0}\left(\frac{1}{r}J_1(kr)+kJ_0(kr)\right)\,\mathrm{d}k\right] \qquad (A.50)\\ E_{y,e}^{\mathsf{HED}_x}(\mathbf{r},\omega) &= \frac{\widetilde{\mathbf{D}}(\omega)}{4\pi\mu}\left[\left(\frac{1}{r}-\frac{y^2}{r^3}\right)\int_0^\infty f_e(z,k,\omega)J_1(kr)\frac{1}{u_0}\,\mathrm{d}k \right.\\ &\quad \left. +\frac{y^2}{r^2}\int_0^\infty f_e(z,k,\omega)\frac{1}{u_0}\left(\frac{1}{r}J_1(kr)+kJ_0(kr)\right)\,\mathrm{d}k\right] \qquad (A.51)\end{aligned}$$

With (A.38) and (3.11), the TM-mode of the electrical field components of the HMD can be computed from

$$\begin{aligned} E^{\mathsf{HED}_x}_{x,m}(\mathbf{r},\omega) &= \frac{1}{\sigma}\partial^2_{xz}\left(\sigma\phi_m\right) \\ &= \frac{\widetilde{\mathbf{D}}(\omega)}{4\pi\mu}\left[\left(\frac{1}{r}-\frac{x^2}{r^3}\right)\int_0^\infty f_m(z,k,\omega)J_1(kr)\frac{1}{u_0^2}\,\mathrm{d}k\right. \\ &\quad\left.+\frac{x^2}{r^2}\int_0^\infty f_m(z,k,\omega)\frac{1}{u_0^2}\left(\frac{1}{r}J_1(kr)+kJ_0(kr)\right)\,\mathrm{d}k\right] \qquad \text{(A.52)} \\ E^{\mathsf{HED}_x}_{y,m}(\mathbf{r},\omega) &= \frac{1}{\sigma}\partial^2_{yz}\left(\sigma\phi_m\right) \\ &= -\frac{\widetilde{\mathbf{D}}(\omega)}{4\pi\mu}\left[\frac{xy}{r^3}\int_0^\infty f_m(z,k,\omega)J_1(kr)\frac{1}{u_0^2}\,\mathrm{d}k\right. \\ &\quad\left.+\frac{xy}{r^2}\int_0^\infty f_m(z,k,\omega)\frac{1}{u_0^2}\left(\frac{1}{r}J_1(kr)+kJ_0(kr)\right)\,\mathrm{d}k\right] \qquad \text{(A.53)} \end{aligned}$$

Or, taking the relation of the moments into account in order to calculate the adjoint fields

$$\begin{aligned} E^{\mathsf{HED}_x}_{x,m}(\mathbf{r},\omega) &= \frac{1}{\sigma}\partial^2_{xz}\left(\sigma\phi_m\right) \\ &= \frac{\widetilde{\mathbf{D}}(\omega)}{4\pi\mu}\left[\left(\frac{1}{r}-\frac{x^2}{r^3}\right)\int_0^\infty f_m(z,k,\omega)J_1(kr)\frac{1}{u_0^2}\,\mathrm{d}k\right. \\ &\quad\left.+\frac{x^2}{r^2}\int_0^\infty f_m(z,k,\omega)\frac{1}{u_0^2}\left(\frac{1}{r}J_1(kr)+kJ_0(kr)\right)\,\mathrm{d}k\right] \qquad \text{(A.54)} \\ E^{\mathsf{HED}_x}_{y,m}(\mathbf{r},\omega) &= \frac{1}{\sigma}\partial^2_{yz}\left(\sigma\phi_m\right) \\ &= -\frac{\widetilde{\mathbf{D}}(\omega)}{4\pi\mu}\left[\frac{xy}{r^3}\int_0^\infty f_m(z,k,\omega)J_1(kr)\frac{1}{u_0^2}\,\mathrm{d}k\right. \\ &\quad\left.+\frac{xy}{r^2}\int_0^\infty f_m(z,k,\omega)\frac{1}{u_0^2}\left(\frac{1}{r}J_1(kr)+kJ_0(kr)\right)\,\mathrm{d}k\right] \qquad \text{(A.55)} \end{aligned}$$

In particular, the adjoint fields from every receiver-source considered in this work, can be calculated as a linear combination of the translated sources given above.

APPENDIX B

Direct and iterative solutions of linear systems

The solution to the linear system

$$\mathbf{A}\mathbf{x} = \mathbf{b} \,, \tag{B.1}$$

with $\mathbf{A} \in \mathbb{R}^{N\times N}$, $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{b} \in \mathbb{R}^N$ is one of the most basically problems we encounter in computational sciences. Nevertheless it is one of the most challenging, too.

A lot of different applications of linear systems can be found and sometimes the underlying algorithm benefit from the special nature of equation (B.1) which results in sparse and banded linear systems. Because of the wide range of applications for large linear system solution and their sometimes very specific solution strategies, it is not possible to cover them all in a few pages. Here you will find just a small compilation of solution strategies which where applied in this work.

For geophysical applications, equation (B.1) occurs two times:

1. if we want to solve the forward problem where (B.1) is sparse, the solution unique and normally well conditioned[1], or

2. in the inverse problems where the system matrix is not sparse, the solution not unique and the problem ill posed.

The solution of (B.1) can be made iteratively or by using a factorization of the matrix $\mathbf{A}$ as expressed in B.0.4. Matrix factorization using the Cholesky decomposition is numerically favorable because it is very fast and stable. As a drawback it can suffer from round off problems and it does not deliver any singular values which can be advantageous for resolution analysis of the inverse solution. The decomposition based on eigenvalues is commonly done within a singular value decomposition (SVD). The latter has a special interest in the solution of the normal equations, because of it's inherit possibility to resolve which parameters do

[1] As we will see, the well posedness of the forward problem is not always the case

influence the inverse process the most and how strong they are represented during the inverse process. But they all suffer from the same problem regarding memory usage computation effort and can not generally be applied in large scale inversions.

If the linear system (B.1) gets to big, iterative methods to it's solution are applied. These are subsumed under the term conjugated gradients (CG). Their basic equations and algorithms are given in B.0.5.

B.0.4 Matrix factorization

The most prominent matrix factorization is called the LU decomposition. It is applicable to any square matrix $\mathbf{A}$. The decomposition

$$\mathbf{A} = \mathbf{LU} \; , \tag{B.2}$$

where $\mathbf{L}$ is lower triangular and $\mathbf{U}$ is a upper triangular matrix. With the appropriate factorization, the systems

$$\mathbf{L}(\mathbf{Ux}) = \mathbf{b} \tag{B.3}$$

and

$$\mathbf{Ux} = \mathbf{L} - \mathbf{Ib} \tag{B.4}$$

are much easier to solve than the original linear system.

The most prominent way to achieve a LU-factorization is the Gaussian elimination. For a square matrix of dimension N the Gaussian elimination would take roughly $\frac{2}{3}N^3$ arithmetical operations [Westlake, 1968; Press et al., 1995]. But, due to numerical stability this should only be applied to regular matrices. In fact a stable solution demands a invertible matrix on the left hand side. As this is generally not the case if we consider ill posed problems, the matrix $\mathbf{A}$ is somehow regularized. However, in the recent future, the incomplete LU decomposition has gained interest for preconditioning issues.

Cholesky decomposition

This decomposition can be applied to a square, symmetric and positive definite matrix $\mathbf{A}$. The Cholesky decomposition is sometimes called a incomplete LU factorization, because

$$\mathbf{A} = \mathbf{U}^{\mathrm{T}}\mathbf{U} \; , \tag{B.5}$$

where $\mathbf{U}$ is upper triangular with positive diagonal entries, is a special case of the symmetric LU decomposition, with $\mathbf{L} = \mathbf{U}^{\mathrm{T}}$. Most implementations of equation (B.5) (e.g. within the free linear algebra package LAPACK) allow that the decomposition can be done with the lower triangular matrix, too. With the decomposition at hand, the solution to equation (B.1) is achieved via forward substitution of

$$\mathbf{Uy} = \mathbf{b} \tag{B.6}$$

and back substitution of

$$\mathbf{U}^{\mathrm{T}}\mathbf{x} = \mathbf{y} \; . \tag{B.7}$$

Anyways, the application of the equations (B.6) and (B.7) is normally already done within the implemented algorithm (e.g. LAPACK), so that the user has not to concern. For a square matrix of dimension N the Cholesky decomposition would take roughly $\frac{1}{3}N^3$ arithmetical operations, which is half the cost of a full LU decomposition.

SVD

Suppose $\mathbf{A}$ is an $n \times m$ real valued or complex matrix, then there exists a factorization of the form

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^{\mathrm{T}} \tag{B.8}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal and unitary matrices and $\boldsymbol{\Sigma}$ is an $n \times m$ diagonal matrix. Such a factorization is called a singular-value decomposition of $\mathbf{A}$.

- The $m \times m$ matrix $\mathbf{V} = \mathbf{V}^{\mathrm{T}}$ contains a set of orthonormal "‘input"’ or "‘analyzing"’ basis vector directions for $\mathbf{A}$
- The $n \times n$ matrix $\mathbf{U} = \mathbf{U}^{\mathrm{T}}$ contains a set of orthonormal "‘output"‘ basis vector directions for $\mathbf{A}$
- The matrix $\boldsymbol{\Sigma}$ contains the singular values, which can be thought of as scalar "’gain controls"‘ by which each corresponding input is multiplied to give a corresponding output.

A common convention is to order the singular values s_i, $i = 1, \ldots, n$ in decreasing order. However, the diagonal matrix $\boldsymbol{\Sigma}$ is uniquely determined by $\mathbf{A}$.

The main advantage in the application of the SVD to solve the normal equation is, that $\mathbf{V}$ and $\mathbf{U}$ are easily to invert because of their orthogonality ($\mathbf{U}^{\mathrm{T}}\mathbf{U} = \mathbf{V}^{\mathrm{T}}\mathbf{V} = \mathbf{I}$). Applying the SVD, a natural inverse of $\mathbf{A}$ can be found

$$\mathbf{A}^{\dagger} = \mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{U}^{\mathrm{T}}\ , \tag{B.9}$$

which is also know as pseudo inverse of $\mathbf{A}$.

Suppose we can find a inverse solution, the formulation of the Gauss-Newton least squares problem, with $\mathbf{A} = \mathbf{S}$, $\mathbf{x} = \delta\mathbf{m}$, $\mathbf{b} = \mathbf{y}$, reads

$$(\mathbf{A}^{\mathrm{T}}\mathbf{A})\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}\ , \tag{B.10}$$

which is solved by

$$\mathbf{x} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{\dagger}\mathbf{A}^{\mathrm{T}}\mathbf{b} = \mathbf{A}^{\dagger}\mathbf{b} = \sum_{N}^{i=1} \frac{\mathbf{u}_i^{\mathrm{T}}\mathbf{b}}{s_i}\mathbf{v}_i \tag{B.11}$$

The weighting coefficients for the individual model vectors are determined dividing the dot product of $\mathbf{b}$ and the corresponding data vector $\mathbf{u}_i$ by the singular value s_i. Figure B.1 (left) shows the singular value spectrum which typically arises for ill-posed problems. The spread of several decades between the largest and smallest singular value is an indication for the ill-posedness. It is generally observed that $\mathbf{v}_i$ for small i show relatively simple

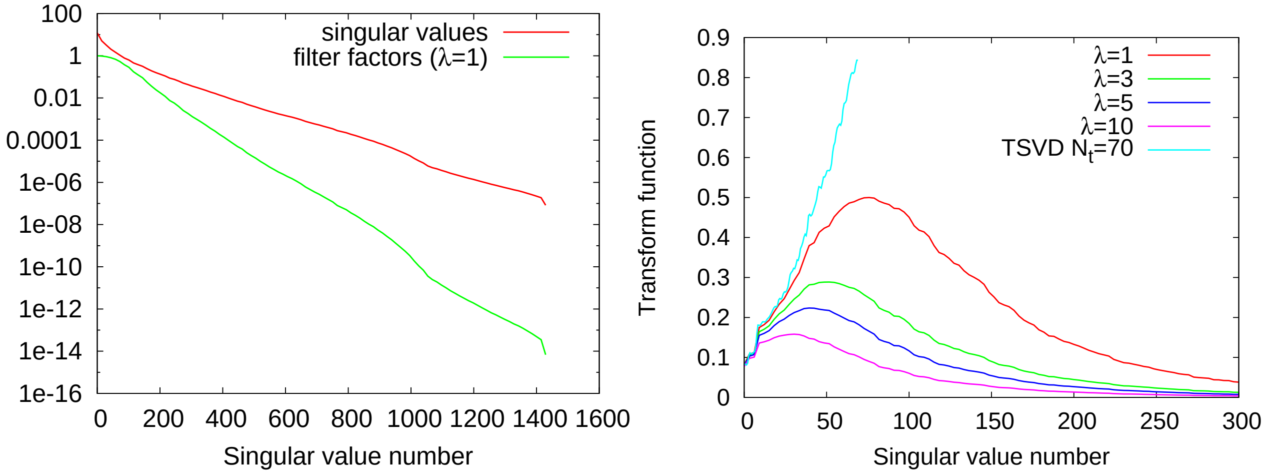


Figure B.1: *Spectrum of singular values and filter factors for a typical ill posed problem (left). Transform functions within a Tikhonov regularization for different values of* λ *(right). For comparison, the TSVD spectrum is shown for a threshold of* $N_t = 70$ *singular values.*

characteristics, whereas model vectors connected to small singular values tend to show high-frequency oscillations. Due to the division by s_i the model vectors for small singular values are amplified resulting in a strongly structured model. Although, this model fits the data well, it lacks plausibility. Hence, damping or truncation methods are introduced to stabilize this affects. The truncated SVD (TSVD) limits the summation up to an certain singular value $N_t < N$, from that on the singular values are treated as zero. A small value corresponds to large regularization and vice versa. For more insight on how this can be implemented into inversion and more likely a variance analysis the reader is referred to Kalscheuer & Pedersen [2007].

Another way to reduce the effect of near-zero singular values from exploding is to introduce a damping. A dampened version of the SVD can be used with the definition of filter factors [Tikhonov & Arsenin, 1977]

$$f_i = \frac{s_i^2}{s_i^2 + \lambda} \; . \tag{B.12}$$

Applying them changes equation (B.11) to

$$\mathbf{x} = \mathbf{V}\mathrm{diag}\left\{\frac{f_i}{s_i}\right\}\mathbf{U}^{\mathrm{T}} = \sum_{N}^{i=1} \frac{s_i \mathbf{u}_i^{\mathrm{T}}\mathbf{b}}{s_i^2 + \lambda}\mathbf{v}_i \tag{B.13}$$

This regularization strategies is also known as Tikhonov regularization. Note that the filter factors decay much faster than the singular values (cf. fig. B.1, left), which is implied by the quadratic definition of the function.

The ratio $\frac{f_i}{s_i} = \frac{s_i}{s_i^2+\lambda}$ works as transform function for the individual model vectors. Note that the TSVD inversion can also be formulated in this manner by using

$$f_i = \begin{cases} 1 & i \le N_t \\ 0 & \text{else} \end{cases} \tag{B.14}$$

Figure B.1 (right) shows the transform functions for TSVD and Tikhonov regularization. The λ are selected for typical values of 1 to 10 and the truncation value for TSVD is chosen such that the results are comparable. For the Tikhonov regularization the transform function is maximized at singular values between 30 (for $\lambda = 10$) and 70 (for $\lambda = 1$). It can be seen, that larger or smaller values for λ shift the maximum towards lower or higher singular value numbers, respectively. In contrast, the largest contributions of the TSVD are provided by the model vectors directly below the threshold. Applying TSVD would therefore result in more structured models.

Given this example the most powerful application of the SVD becomes clear. Once the factorization has been carried out, solutions for many regularization parameters can be obtained in no time.

Additionally, the SVD provides a powerful tool for resolution analysis as well. Once the factorization is done, the resolution matrix can be easily obtained by the generalized inverse:

$$\begin{aligned}\mathbf{R} &= \mathbf{V}\mathrm{diag}\left\{\frac{f_i}{s_i}\right\}\mathbf{U}^{\mathrm{T}}\mathbf{U}\mathrm{diag}\left\{s_i\right\}\mathbf{V}^{\mathrm{T}} \\ &= \mathbf{V}\mathrm{diag}\left\{f_i\right\}\mathbf{V}^{\mathrm{T}} \,. \end{aligned} \tag{B.15}$$

It can be seen, that the resolution is dominated by the Tikhonov regularization, thus implicitly depends on λ.

If the model constraints are formulated using $\mathbf{C} \neq \mathbf{I}$ the generalized inverse can not be represented by the formulation given above. For global regularization schemes a more general approach is needed which can be found in Golub & Loan [1996].

B.0.5 Conjugate gradient for solving linear systems (CG)

The conjugate gradient method as an iterative method for the solution of (B.1) is developed by Hestenes & Stiefel [1952]. It utilizes the residual formulation of the linear system to update its solution iteratively. This also allows for the solution of systems where N is so large that the direct method would take too much time.

If the conjugate gradient method is applied naively to the normal equations

$$\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b} \,, \tag{B.16}$$

the method does not perform well because of the ill conditioned left hand side. To a large extent this is due to the explicit use of vectors of the form $\mathbf{A}^{\mathrm{T}}\mathbf{A}\mathbf{p}_i$.

An algorithm with better numerical properties is easily derived by a slight algebraic rearrangement, making use of the intermediate vector $\mathbf{q}_i = \mathbf{A}\mathbf{p}_i$ [Hestenes & Stiefel, 1952]. However, applying the conjugate gradient method to the normal equations can be solved using different approaches. Using a formulation after Paige & Saunders [1982], the norm of the residual vector

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\mathbf{x} \tag{B.17}$$

can be minimized using the CGLS[2] algorithm. A pseudo-code of the CGLS is given in algorithm 6.

[2] CGLS stands for Conjugate Gradient Linear System

Algorithm 6 Conjugate gradient for minimization of the residual norm (CGLS)

```
Require: $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$; $\mathbf{z}_0 = \mathbf{p}_0 = \mathbf{A}^\mathrm{T}\mathbf{r}_0$; $0 < \epsilon < 1$ ; $N_{max} \leq N$
  $i = 0$
  while $\frac{\|\mathbf{r}_i\|}{\|\mathbf{r}_0\|} > \epsilon$ and $i \leq N_{max}$ do
    $\mathbf{q}_i = \mathbf{A}\mathbf{p}_i$ {intermediate vector}
    $\alpha_i = \frac{\|\mathbf{z}_i\|^2}{\|\mathbf{q}_i\|^2}$ {projection of the error norm onto $\mathbf{A}$}
    $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i$ {conjugation of the solution}
    $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{q}_i$ {conjugation of the residual}
    $\mathbf{z}_{i+1} = \mathbf{A}^\mathrm{T}\mathbf{r}_{i+1}$ {updating the error}
    $\beta_i = \frac{\|\mathbf{z}_{i+1}\|^2}{\|\mathbf{z}_i\|^2}$ {projection of the error}
    $\mathbf{p}_{i+1} = \mathbf{z}_{i+1} + \beta_i \mathbf{p}_i$ {conjugation of the error }
    $i = i + 1$
  end while
```

On the other hand, one can minimize the error norm $\|\mathbf{x} - \mathbf{x}_i\|$ of the normal equation as well. The algorithm is called CGNE[3] and also known as Craig's method [Paige & Saunders, 1982; Saad, 1996]. Further convergence analysis of the CGLS and the CGNE can be found in the book of Saad [1996]. For this work, I used a slightly modificated version with appropriate stopping criteria. It's pseudo-code is given in algorithm 7.

Algorithm 7 Conjugate gradient for minimization of the error norm (CGNE)

```
Require: $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$; $\mathbf{p}_0 = \mathbf{A}^\mathrm{T}\mathbf{r}_0$; $0 < \epsilon < 1$; $N_{max} \leq N$
  $i = 0$
  while $\frac{\|\mathbf{p}_i\|}{\|\mathbf{p}_0\|} > \epsilon$ and $i \leq N_{max}$ do
    $\alpha_i = \frac{\|\mathbf{r}_i\|^2}{\|\mathbf{p}_i\|^2}$ {projection of the residual onto the error}
    $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i \mathbf{p}_i$ {conjugation of the initial solution}
    $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i \mathbf{A}\mathbf{p}_i$ {conjugation of the residual}
    $\beta_i = \frac{\|\mathbf{r}_{i+1}\|^2}{\|\mathbf{r}_i\|^2}$ {projection of the residual}
    $\mathbf{p}_{i+1} = \mathbf{A}^T\mathbf{r}_{i+1} + \beta_i \mathbf{p}_i$ {conjugation of the error}
    $i = i + 1$
  end while
```

Note, that the most numerical effort in the presented algorithms 6 and 7 is due to the two matrix vector products which can not be circumvented. Overall, the performance of the products can be accelerated significantly by using inline formulations of the matrix vector products. This can be achieved by using the MATMUL function which is included in the native FORTRAN90 library. A parallelized scheme can be helpful as well, but it is only useful on systems where the processors have access to the same memory (shared memory). To make further enhancements of the code, the vectors given in algorithm 6 and 7 can be overwritten in every iteration. However, the residual norm of the previous iteration must be stored to compute β.

Note further, that the convergence criteria for both algorithms is different. As suggested by the underlying minimization scope of the CGLS, the algorithm breaks,if the ratio of the residual norm at the beginning compared to the one yielded in every iteration is below a

[3] CGNE stands for Conjugate gradient normal equation

certain threshold $0 < \epsilon < 1$. On the other hand, the CGNE algorithm drops out if the norm of the error is smaller than the desired accuracy level. During application of both algorithms it has been found, that for moderate values of $0.01 < \epsilon < 0.001$ the iteration numbers are mostly below the maximum value N of iterations.

As expected for iterative solvers and ill posed systems, the error norm $\frac{\|\mathbf{r}_i\|}{\|\mathbf{r}_0\|}$ can oscillate strongly between the iterations. As stated before, low-frequency components of the solution tend to converge faster than high-frequency parts. This renders the approach very unstable if one uses small values of ϵ.

Taking this into account, a further dropout criteria $N_{max} \leq N$ is introduced into the algorithms. The resulting algorithm can thus be seen as truncated least squares method and the stopping criterion N_{max} can be treated as regularization parameter. If one wants to apply the TLS only, the stopping criteria should be altered properly (e.g. setting $\epsilon = 0$).

During some tests of the stated algorithms I discovered that the CGNE performs a little better than the CGLS. This may be due to the different projection of the residual vector and the drop out criterion, which is the only real difference between the two algorithms. Additionally a preconditioning scheme, as given more detailed below, and the incorporation of regularization is easier to implement into the CGNE. However, they yield comparable results, the CGNE seemed more appropriate for the scope of this work. Although, the following formulations can be incorporated in the CGLS in a manner the like, I just proceed with the CGNE. A formulation of the CGLS which accounts for regularization can be found in [Günther, 2004].

Damped normal equations

To incorporate the regularization schemes (see section 2.3) into the CGNE algorithm, it has to be adjusted appropriately. Assume a Marquardt type of regularization resulting in the damped normal equations

$$\left(\mathbf{A}^{\mathrm{T}}\mathbf{A} + \lambda\mathbf{I}\right)\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b}\ . \tag{B.18}$$

The residual of the basis equation is the same, but the gradient $\mathbf{p}$

$$\mathbf{p} = \mathbf{A}^{\mathrm{T}}(\mathbf{b} - \mathbf{A}\mathbf{x}) - \lambda\mathbf{I}\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{r} - \lambda\mathbf{x} \tag{B.19}$$

is altered. The additional term affects not only the gradient directions but implicitly also the coefficients from the projection of the residual on the error norm. Result of the changes yields algorithm 8, called CGDNE. Note, that the additional stopping criterion N_{max} is still delivered. Through application of the CGDNE it turns out that it is sometimes needed, but not in any case.

Constraint normal equation

To account for a smooth model inversion, one has to perform some more changes to the algorithm.

Assume the constraint matrix $\mathbf{C}$ as given in section 2.3, the normal equation read

$$\left(\mathbf{A}^{\mathrm{T}}\mathbf{A} + \lambda\mathbf{C}^{\mathrm{T}}\mathbf{C}\right)\mathbf{x} = \mathbf{A}^{\mathrm{T}}\mathbf{b} - \mathbf{C}^{\mathrm{T}}\mathbf{C}\Delta\hat{\mathbf{x}}\ . \tag{B.20}$$

Algorithm 8 Conjugate gradient applied to damped normal equations (CGDNE)

```
Require: $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$; $\mathbf{p}_0 = \mathbf{A}^{\mathrm{T}}\mathbf{r}_0 - \lambda\mathbf{x}_0$; $0 < \epsilon < 1$; $N_{max} \leq N$
  $i = 0$
  while $\|\mathbf{p}_i\| > \epsilon\,\|\mathbf{p}_0\|$ and $i \leq N_{max}$ do
    $\alpha_i = \frac{\|\mathbf{r}_i\|^2}{\|\mathbf{p}_i\|^2}$
    $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i\mathbf{p}_i$
    $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i\mathbf{A}\mathbf{p}_i$
    $\beta_i = \frac{\|\mathbf{r}_{i+1}\|^2}{\|\mathbf{r}_i\|^2}$
    $\mathbf{p}_{i+1} = \mathbf{A}^T\mathbf{r}_{i+1} + \beta_i\mathbf{p}_i - \lambda\mathbf{x}_{i+1}$
    $i = i + 1$
  end while
```

Algorithm 9 Conjugate gradient applied to constraint normal equations (CGCNE)

```
Require: $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$; $\mathbf{p}_0 = \mathbf{A}^{\mathrm{T}}\mathbf{r}_0 - \lambda\mathbf{C}^{\mathrm{T}}\mathbf{C}\mathbf{x}_0$; $0 < \epsilon < 1$; $N_{max} \leq N$
  $i = 0$
  while $\|\mathbf{p}_i\| > \epsilon\,\|\mathbf{p}_0\|$ and $i \leq N_{max}$ do
    $\alpha_i = \frac{\|\mathbf{r}_i\|^2}{\|\mathbf{p}_i\|^2}$
    $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i\mathbf{p}_i$
    $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i\mathbf{A}\mathbf{p}_i$
    $\beta_i = \frac{\|\mathbf{r}_{i+1}\|^2}{\|\mathbf{r}_i\|^2}$
    $\mathbf{p}_{i+1} = \mathbf{A}^T\mathbf{r}_{i+1} + \beta_i\mathbf{p}_i - \lambda\mathbf{C}^{\mathrm{T}}\mathbf{C}\mathbf{x}_{i+1}$
    $i = i + 1$
  end while
```

The additional term affects again the gradient directions $\mathbf{p}_i$ and the coefficients α_i. Which seems odd because intuitively, the term on the right hand side should affect the residual as well. But if the gradient is given by

$$\mathbf{p} = \mathbf{A}^{\mathrm{T}}\mathbf{r} - \lambda\mathbf{C}^{\mathrm{T}}\mathbf{C}\mathbf{x}\ , \tag{B.21}$$

the additional term is already reflected in the residual by forming

$$\mathbf{r}_{i+1} = \mathbf{r}_i - \mathbf{A}\mathbf{p}_i \tag{B.22}$$

The solution to (B.20) is displayed in algorithm 9.

B.0.6 Preconditioning

In linear algebra and numerical analysis, a preconditioner $\mathbf{P}$ of a matrix $\mathbf{A}$ is a matrix such that $\mathbf{P}^{-1}\mathbf{A}$ has a smaller condition number than $\mathbf{A}$ [Saad, 1996]. The condition number is given by the ratio

$$\kappa(\mathbf{A}) = \left\|\mathbf{A}^{-1}\right\| \|\mathbf{A}\| = \frac{\sigma_{\mathsf{max}}(\mathbf{A})}{\sigma_{\mathsf{min}}(\mathbf{A})} \tag{B.23}$$

where $\sigma_{\mathsf{max}}(\mathbf{A})$ is the largest and $\sigma_{\mathsf{min}}(\mathbf{A})$ is the smallest Eigenvalue of $\mathbf{A}$. Preconditioners are reported very useful when using an iterative method to solve a large linear system, since the rate of convergence for most iterative linear solvers degrades as the condition number of a matrix increases. Anyway, the preconditioning can be used for direct solvers as well.

Instead of solving the original linear system above, one may solve either the left preconditioned system

$$\mathbf{P}^{-1}\mathbf{A}\mathbf{x} = \mathbf{P}^{-1}\mathbf{b}, \tag{B.24}$$

via the two solves

$$\mathbf{c} = \mathbf{P}^{-1}\mathbf{b}, \qquad \left(\mathbf{P}^{-1}\mathbf{A}\right)\mathbf{x} = \mathbf{c}, \tag{B.25}$$

or the right preconditioned system

$$\mathbf{A}\mathbf{P}^{-1}\mathbf{P}\mathbf{x} = \mathbf{b}, \tag{B.26}$$

via

$$\left(\mathbf{A}\mathbf{P}^{-1}\right)\mathbf{y} = \mathbf{b}, \qquad \mathbf{x} = \mathbf{P}^{-1}\mathbf{y}, \tag{B.27}$$

which are both equivalent to solving the original system as long as the preconditioner matrix $\mathbf{P}$ is nonsingular.

Incorporating the left preconditioning (B.24) into the CGCNE algorithm is easy to implement. The result is called LPCGCNE which pseudo code is given in algorithm 10.

Algorithm 10 Left-Preconditioned conjugate gradient applied to constraint normal equations (LPCGCNE)

```
Require: $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$; $\mathbf{z}_0 = \mathbf{P}^{-1}\mathbf{r}_0$, $\mathbf{p}_0 = \mathbf{A}^{\mathrm{T}}\mathbf{z}_0 - \lambda\mathbf{C}^{\mathrm{T}}\mathbf{C}\mathbf{x}_0$; $0 < \epsilon < 1$; $N_{max} \leq N$
  $i = 0$
  while $\|\mathbf{p}_i\| > \epsilon \|\mathbf{p}_0\|$ and $i \leq N_{max}$ do
    $\alpha_i = \frac{\mathbf{z}_i^{\mathrm{T}}\mathbf{r}_i}{\|\mathbf{p}_i\|^2}$
    $\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha_i\mathbf{p}_i$
    $\mathbf{r}_{i+1} = \mathbf{r}_i - \alpha_i\mathbf{A}\mathbf{p}_i$
    $\mathbf{z}_{i+1} = \mathbf{P}^{-1}\mathbf{r}_{i+1}$
    $\beta_i = \frac{\mathbf{z}_{i+1}^{\mathrm{T}}\mathbf{r}_{i+1}}{\mathbf{z}_i^{\mathrm{T}}\mathbf{r}_i}$
    $\mathbf{p}_{i+1} = \mathbf{A}^T\mathbf{z}_{i+1} + \beta_i\mathbf{p}_i - \lambda\mathbf{C}^{\mathrm{T}}\mathbf{C}\mathbf{x}_{i+1}$
    $i = i + 1$
  end while
```

Typically there is a trade-off in the choice of $\mathbf{P}$. Since the operator $\mathbf{P}^{-1}$ must be applied at each step of the iterative linear solver, it should be very efficient in order to reduce the computing time of its application. Summarizing, the desired effect in applying a preconditioner is to make the quadratic form of the preconditioned operator $\mathbf{P}^{-1}\mathbf{A}$ to be nearly spherical [Paige & Saunders, 1982].

Jacobi preconditioner

The Jacobi preconditioner is one of the simplest forms of preconditioning. The preconditioner is chosen to be the diagonal of the matrix

$$\mathbf{P} = \mathbf{D} \tag{B.28}$$

That is

$$P_{ij} = A_{ii}\delta_{ij} = \begin{cases} A_{ii} & i = j \\ 0 & \text{otherwise} \end{cases} \tag{B.29}$$

and so

$$P_{ij}^{-1} = \frac{\delta_{ij}}{A_{ii}} \; . \tag{B.30}$$

Thus, suppose the sensitivity matrix is available, this preconditioner is very easy to compute.

APPENDIX C

A posteriori model covariance

Besides the resolution matrix, the investigation of the a-posteriori model covariance matrix MCM is of keen interest.

Based on the derivations given in chapter 2 and using the rule of variance propagation for the linear system $\mathbf{Ax} = \mathbf{b}$

$$\mathrm{Cov}(\mathbf{Ax} + \mathbf{b}) = \mathbf{A}\mathrm{Cov}(\mathbf{x})\mathbf{A}^{\mathrm{T}} \ , \tag{C.1}$$

the MCM can be derived directly from equation (2.70). Since the true model has no variance and the noise has the variance of the data ($\mathrm{Cov}(\mathbf{d}) = \mathbf{C}_d$), the model covariance matrix reads

$$\mathbf{C}_M = (\mathbf{I} - \mathbf{R})\mathbf{C}_m^{-1}\hat{\mathbf{m}}(\mathbf{I} - \mathbf{R})^{\mathrm{T}} + \boldsymbol{\mathcal{H}}^{-1}\boldsymbol{\mathcal{H}}^{-1^{\mathrm{T}}} \tag{C.2}$$

It consists of two parts: The first one results from the variance of the starting model. The second part originates from the error propagation throughout the inversion process. Since the true variance of the starting model is generally not known, the first term is often omitted.

The model covariance matrix has the same dimensions as $\mathbf{R}$. Its columns can be displayed as co-variations of the model with the corresponding cell. It reveals the statistical nature of the inversion process. The essential information of $\mathbf{C}_M$ is contained in the main diagonal elements, whose square-roots can be interpreted as uncertainties of the corresponding model parameters.

This is valuable if the model parameter values are to be used for petrophysical purposes such as the calculation of porosity or water saturation. Then, the uncertainty of those parameters is known by error propagation as well. In summary, both resolution matrices give information about the geometrical reliability of the model. The MCM is used for parameter uncertainties, whereas $\mathbf{R}$ provides information about the universe of the inverse process with respect to model parameters.

APPENDIX D

Manual for *sinv*

The following chapter gives a detailed description of the 3D TEM inversion program *sinv*which was developed within this thesis.

*sinv*is written in the F90 language and designed to run under Unix like operating systems as for example Linux. Since it should attack the full 3D TEM inversion problem it was parallelized using the Massage Passing Interface (MPI) environment (http://www.open-mpi.org).

D.1 MPI

The Message Passing Interface is a language-independent computer communications descriptive application programmer interface (API) with defined semantics and flexible interpretations; it does not define the protocol by which these operations are to be performed in the sense of sockets for TCP/IP or other layer-4 and below models in the ISO/OSI Reference Model. It is consequently a layer-5+ type set of interfaces, although implementations can cover most layers of the reference model, with sockets+TCP/IP as a common transport used inside the implementation. MPI's goals are high performance, scalability, and portability. Productivity of the interface for programmers is not one of the key goals of MPI, and MPI is generally considered to be low-level. It expresses parallelism explicitly rather than implicitly. MPI is considered successful in achieving high performance and high portability, but is often criticized for its low-level qualities. There is, at present, no effective replacement to MPI, so it remains a crucial part of parallel programming. MPI is not sanctioned by any major standards body, but nonetheless has worldwide practical acceptance.

MPI is a de facto standard for communication among the processes modeling a parallel program on a distributed memory system. Often these programs are mapped to clusters, actual distributed memory supercomputers, and to other environments. However, the principal MPI-1 model has no shared memory concept, and MPI-2 has only a limited distributed shared memory concept used in one portion of that set of extensions.

Most MPI implementations consist of a specific set of routines (API) callable from Fortran, C, or C++ and from any language capable of interfacing with such routine libraries. The advantages of MPI over older message passing libraries are portability (because MPI has been implemented for almost every distributed memory architecture) and speed (because each implementation is in principle optimized for the hardware on which it runs). Interestingly, MPI is supported on shared-memory and NUMA (Non-Uniform Memory Access) architectures as well, where it often serves both as an important portability architecture, but also helps achieve high performance in applications that are naturally owner-computes oriented.

MPI is a specification, not an implementation. MPI has Language Independent Specifications (LIS) for the function calls and language bindings. The first MPI standard specified ANSI C and Fortran-77 language bindings together with the LIS. The draft of this standard was presented at Supercomputing 1994 (November 1994) and finalized soon thereafter. About 128 functions comprise the MPI-1.2 standard as it is now defined.

There are two versions of the standard that are currently popular: version 1.2, which emphasizes message passing and has a static runtime environment (fixed size of world), and, MPI-2.1, which includes new features such as scalable file I/O, dynamic process management, collective communication with two groups of processes, and C++ language bindings. MPI-2's LIS specifies over 500 functions and provides language bindings for ANSI C, ANSI Fortran (Fortran90), and ANSI C++. Interoperability of objects defined in MPI was also added to allow for easier mixed-language message passing programming. A side effect of MPI-2 standardization (completed in 1996) was clarification of the MPI-1 standard, creating the MPI-1.2 level.

It is important to note that MPI-1.2 programs, now deemed "legacy MPI-1 programs," still work under the MPI-2 standard although some functions have been deprecated. This is important since many older programs use only the MPI-1 subset.

MPI is often compared with PVM, which was a popular distributed environment and message passing system developed in 1989, and which was one of the systems that motivated the need for standard parallel message passing systems. Most computer science students who study parallel programming are taught both Pthreads and MPI programming as complementary programming models. he MPI interface is meant to provide essential virtual topology, synchronization and communication functionality between a set of processes (that have been mapped to nodes/servers/ computer instances) in a language independent way, with language specific syntax (bindings), plus a few features that are language specific. MPI programs always work with processes, although commonly people talk about processors. When one tries to get maximum performance, one process per processor (or more recently core) is selected, as part of the mapping activity; this mapping activity happens at runtime, through the agent that starts the MPI program, normally called mpirun or mpiexec.

Such functions include, but are not limited to, point-to-point rendezvous-type send/receive operations, choosing between a Cartesian or graph-like logical process topology, exchanging data between process pairs (send/receive operations), combining partial results of computations (gathering and reduction operations), synchronizing nodes (barrier operation) as well as obtaining network-related information such as the number of processes in the computing session, current processor identity that a process is mapped to, neighboring processes accessible in a logical topology, and so on. Point-to-point operations come in synchronous, asynchronous, buffered, and ready forms, to allow both relatively stronger and weaker semantics for the synchronization aspects of a rendezvous-send. Many outstanding operations are possible in asynchronous mode, in most implementations.

MPI guarantees that there be progress of asynchronous messages independent of the subsequent calls to MPI made by user processes (threads). This rule is often neglected in practical implementations, but is an important underlying principle when one thinks of using asynchronous operations. The relative value of overlapping communication and computation, asynchronous vs. synchronous transfers, and low latency vs. low overhead communication remain important controversies in the MPI user and implementer communities, although recent advances in multi-core architecture are likely to re-enliven such debate. As such, it is relatively easy to write multi-threaded point-to-point MPI codes, which has the key advantage for us to use it in our program design.

D.2 Input files

The program is invoked from the shell command line and is navigated via some input files.

The main input file is named *sinv.inp.*

D.2.1 *sinv.inp*

Within this file there are several sections which control the inversion process. Since *sinv*is designed for 2D and 3D inversion and the input file should run for both types, there are some parts which belong to both and some parts which do not.

Inversion parameter section

The first section controls the inverse process. Here you have to state several switches like the type of regularization, stopping criteria and the type of inverse process. Additionally you have to specify the kind of sensitivity calculation and transformation type for the forward data.

Regularization parameter section

If the Occam type for a 2D or any 3D inversion is chosen you have to quantify the following parameters: The regularization parameters have different senses for each inversion type. For 3D inversion it is

- The Regularization parameter 1 (λ) quantifies the weighting between ϕ_m and ϕ_d.
- The Regularization parameter 2 (α_x) quantifies horizontal smoothing in x-direction.
- The Regularization parameter 4 (α_y) quantifies horizontal smoothing in y-direction.
- The Regularization parameter 6 (α) quantifies the stepsize.

Value	Type	Description
1	Integer	Generate some output during inversion process
1.0	Float	Stopping criterion no. 1: The inverse process is stopped when the functional is reduced to this level
0.5	Float	Stopping criterion for each iteration (Wolfe condition)
10	Integer	Stopping criterion no. 2: The inverse process is stopped when the number of iterations exceeds this value
0	Integer	Number of (parallel) bracket iterations (2D only)
0	Integer	Number of advance Jacobian calculations parallel to bracket iterations (2D only)
285	Integer	Senstype (tested bitwise, see below)
0	Integer	Transformation type for the forward data 0: linear 1: stitched-log 2: asinh
0.12	Float	Perturbation factor for the model. This is only necessary for perturbation approach (2D and 3D)
3	Integer	Inversion type 1:Occam (2D) 2: Marquardt (2D) 3: 3D Occam Inversion

Table D.1: *First part of sinv.inp which consists of the main switches to control the inversion process*

Value	Type	description
0	Integer	Steering of inversion process (tested bitwise, see below)
1.0	Float	regularization parameter 1
1.0	Float	regularization parameter 2
1.0	Float	regularization parameter 4
1.0	Float	regularization parameter 6
0	Integer	Number of timepoints to cut from every transient

Table D.2: *regularization parameters for 2D-Occam or any 3D inversion process*

For 2D-Occam inversion it is

- The Regularization parameter 1 (λ) quantifies the weighting between ϕ_m and ϕ_d.
- The Regularization parameter 2 (α_x) quantifies the relative horizontal to vertical smoothing (h/z).
- The Regularization parameter 4 (α_y) quantifies horizontal smoothing.
- The Regularization parameter 6 (α) quantifies the stepsize.

If you have chosen a 2D Marquardt inversion you just have to quantify only one regularization parameter. The stated regularization parameter is the value which is added to the main diagonal of the unity matrix

Value	Type	Description
1.0	Float	Regularization parameter for truncation of small eigenvalues

Table D.3: *regularization parameters for 2D-Marquardt*

Grid files section

Within this section you have to particularize the grid files which contain the transmitter grid and the appropriate times.

Value	Type	description
1	Integer	Number of grid-files
t_80e.max	Character(30)	Files describing grid specifications
0.0	Float	Shift in x-direction for the whole grid [km]
0.0	Float	Shift in y-direction for the whole grid [km]
10.0 10.0	2xFloat	Only if Senstype +8!! Integration space in horizontal and z direction [m]
1.5	Float	Only if Senstype +4!! Factor for the radius used in singularity removal

Table D.4: *Grid section of the input file*

Receiver files section

This section gives the receiver specifications and forward data self explaining

Model section

The first section controls

Value	Type	description
0	Integer	turn on periodic: 2, turn on 1, turn off 0
1	Integer	Number of receivers (*ndat*)
The next subsection is repeated *ndat* times		
t80rx001.dat	Character(12)	Name of receiver file containing field data
0 0 0. 1 1 1 1 1.	2xInteger Float 4xInteger Float	Cut early and late data points, delay, fix calibration factor (1: fix; 0: free), grid file number, interpolate receiver (0: no, 1: yes), receiver type, factor
The next 3 entries are optional depending on the receiver type		
0	Integer	Only for E-field receivers (receiver type $<$ 5)!! Changes the level for periodic data acquisition
0.0 0.0	2xFloat	Only if receiver type is 8!! Azimuth and inclination angles (α)
0.0	Float	Only if receiver type is 4!! Azimuth angle (°)
0 hzsum.syt	Integer Character(30)	System response (0: no, 1: yes), filename of the system response

Table D.5: *Receiver section of the input file*

Value	Type	description
0	Integer	Topography switch (0: no topography; 1: topography)
topo.dat	Character(30)	Name of topography file
3 2	2xInteger	Number of sections (*nos*), section input type (0: odd sections; 1: one example for all; 2: horizontal symmetric (3D only) 3: odd sections (3D)
-20.0 20.0	(*nos*-1)xFloat	Section borders [m]
The next subsection is repeated *nos* times if the section input type is 0 or 3		
2	Integer	Number of layers in each section (*nols*)
10.0 10.0	(*nols*)xFloat	Resistivity values for each layer
100.0	(*nols*-1)xFloat	Thickness of each layer. The last layer is extended to infinity
1	Integer	Number of additionally 3D cubes (*nob*)
The next subsection is repeated *nob* times		
-1.0 -1.0 0.0	3xFloat	Upper left corner of the body [m]
1.0 1.0 2.0	3xFloat	Lower right corner of the body [m]
1.0 1	Float Integer	Resistivity of the body, alteration switch (1: inverted; 0: do not alter)

Table D.6: *Model section of the input file*

D.3 Inversion process

Before we can start the inversion program we have to clarify the number of nodes on which to run the code. For small Linux clusters we use the LAM environment which invokes a *shell* on each node where we can use the MPI communication. However, in the future this will not necessary anymore since the LAM/MPI project merges with the Open-MPI project now and thus the lamboot sequence is not necessary in future.

However, we assume you have the Open-MPI implementation or lamboot executed, than *sinv*is started via the command line

```
shell$ mpirun -np <number of nodes> sinv
```

Now *sinv*is running on each node of the Cluster. The number of nodes is the number of participants in the Cluster where you want to start *sinv*.

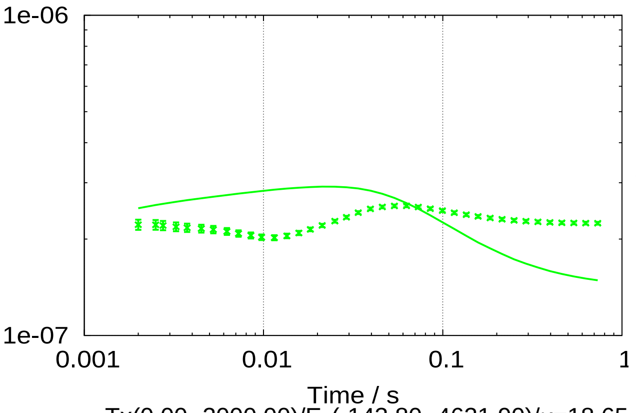
APPENDIX E

DESERT data with 2D model fit

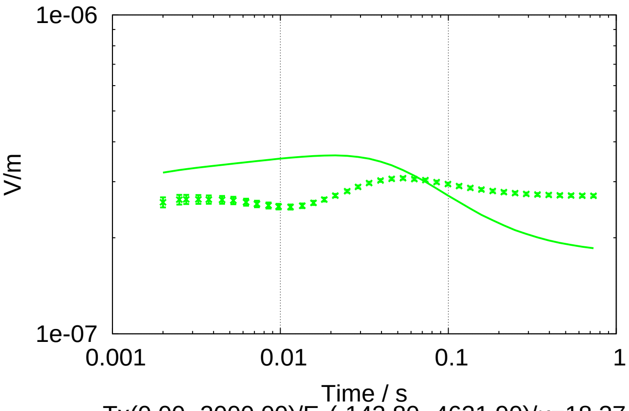
Figures give a compilation of the DESERT data and the fit of the preliminary 2D model shown in section 4.4.5. For convenience, the most important information on the receiver type, the coordinates of the transmitter and the receiver station as well as the misfit, are displayed in the title of each sub-plot.

E.0.1 E_x-field data

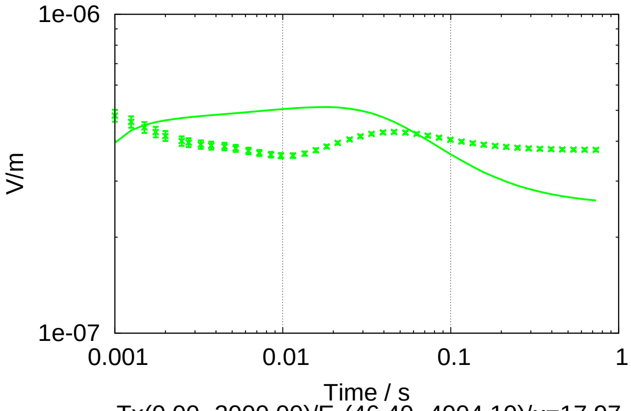
Tx(0.00,-2000.00)//χ=20.03



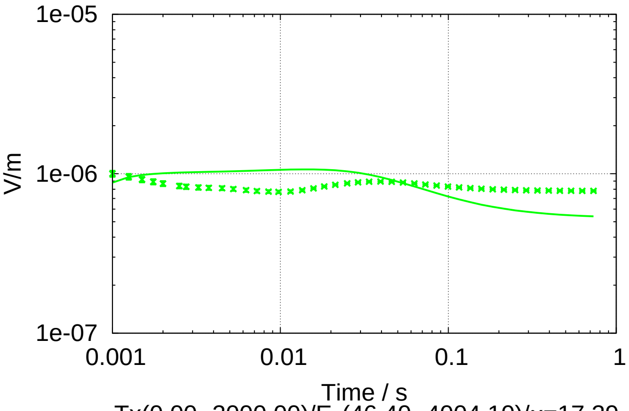
Tx(0.00,-2000.00)/E_x(-143.80,-4621.90)/χ=19.75



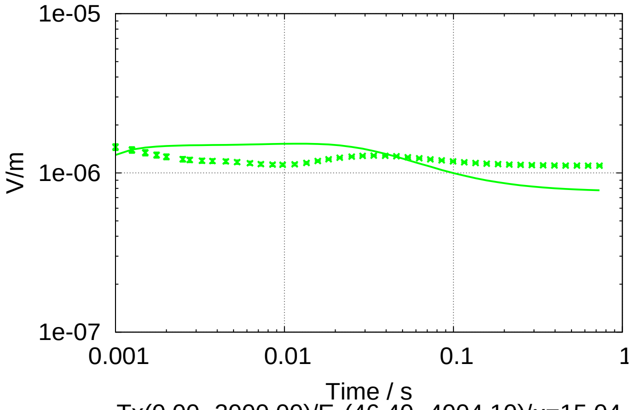
Tx(0.00,-2000.00)/E_x(-143.80,-4621.90)/χ=18.65



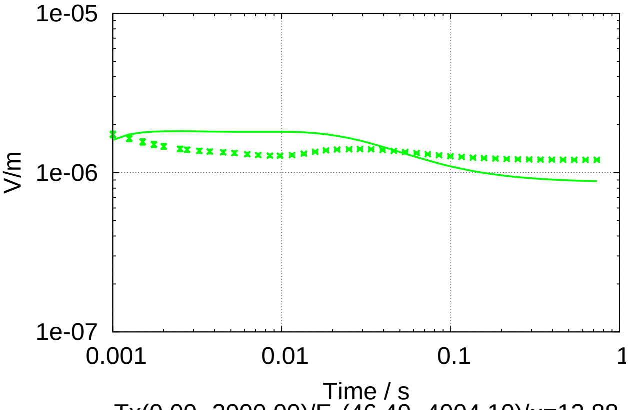
Tx(0.00,-2000.00)/E_x(-143.80,-4621.90)/χ=18.37



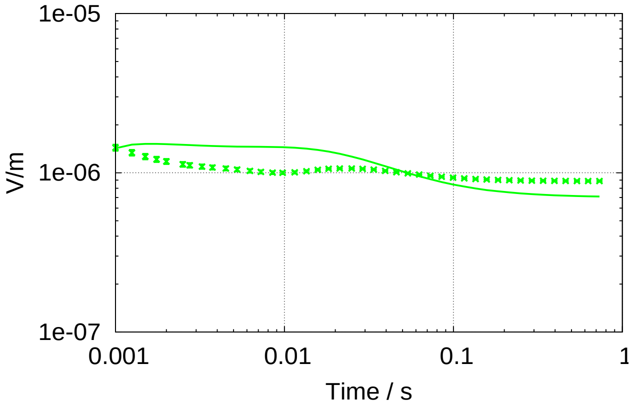
Tx(0.00,-2000.00)/E_x(46.40,-4004.10)/χ=17.97



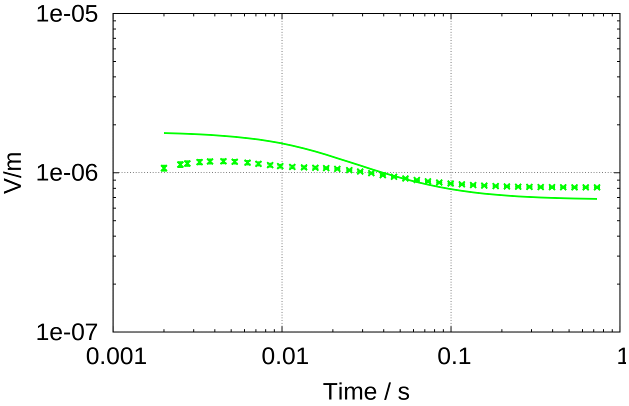
Tx(0.00,-2000.00)/E_x(46.40,-4004.10)/χ=17.29



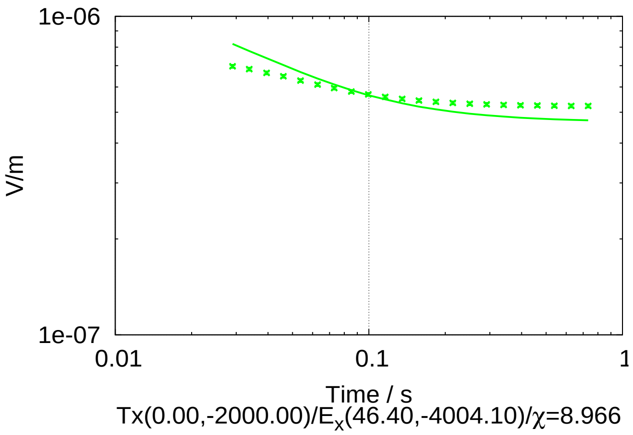
Tx(0.00,-2000.00)/E_x(46.40,-4004.10)/χ=15.04



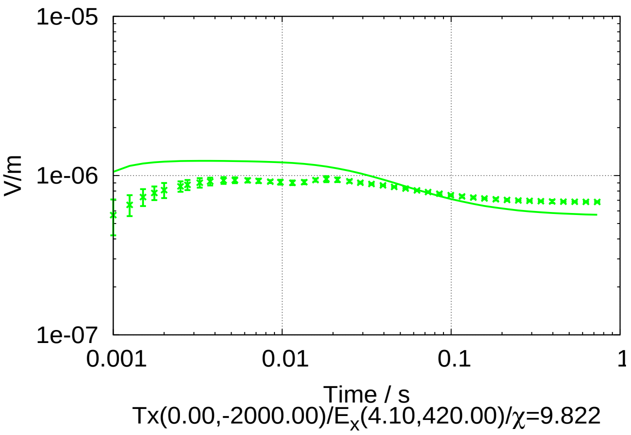
Tx(0.00,-2000.00)/E_x(46.40,-4004.10)/χ=12.88



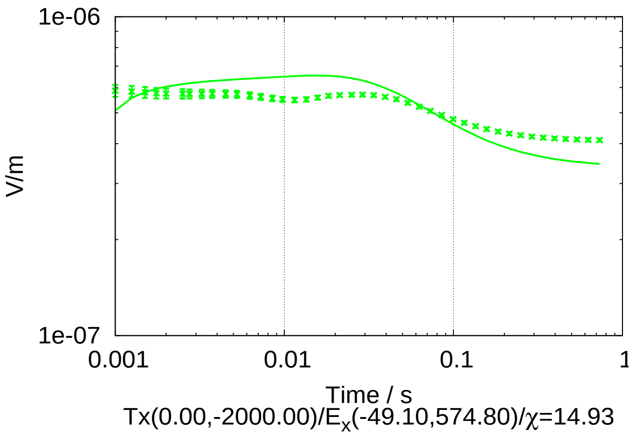
Tx(0.00,-2000.00)/E_x(46.40,-4004.10)/χ=8.097



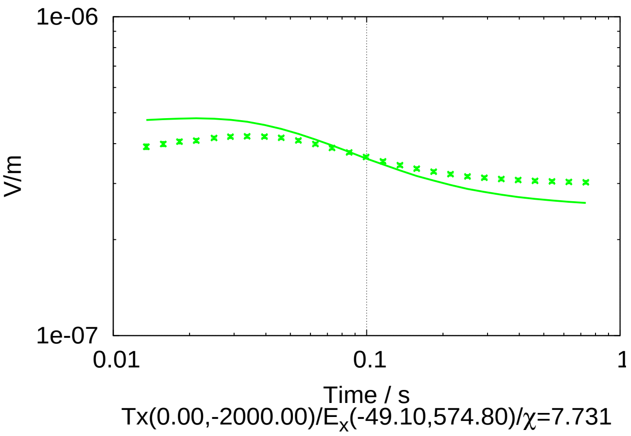
Tx(0.00,-2000.00)/E_x(46.40,-4004.10)/χ=9.086



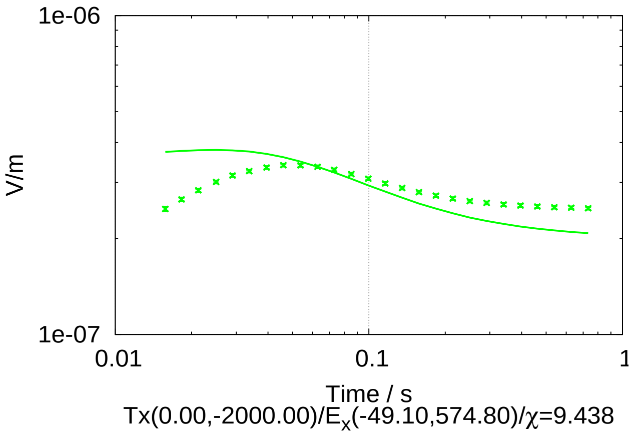
Tx(0.00,-2000.00)/E_x(46.40,-4004.10)/χ=8.966



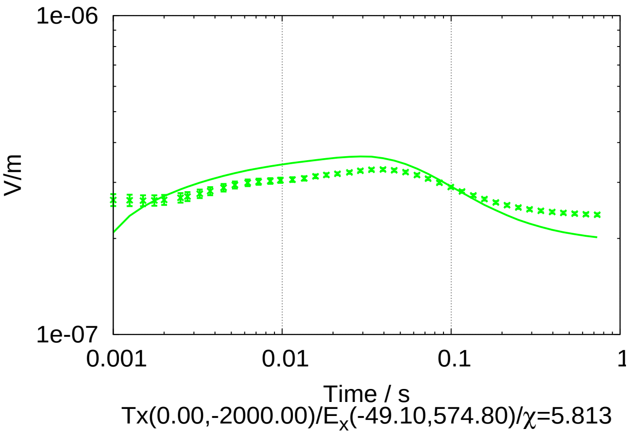
Tx(0.00,-2000.00)/E_x(4.10,420.00)/χ=9.822



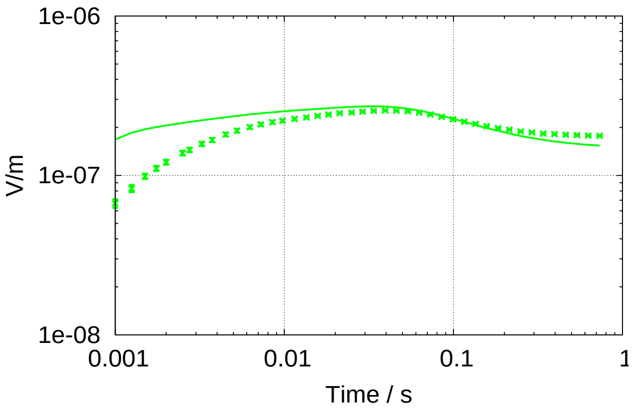
Tx(0.00,-2000.00)/E_x(-49.10,574.80)/χ=14.93



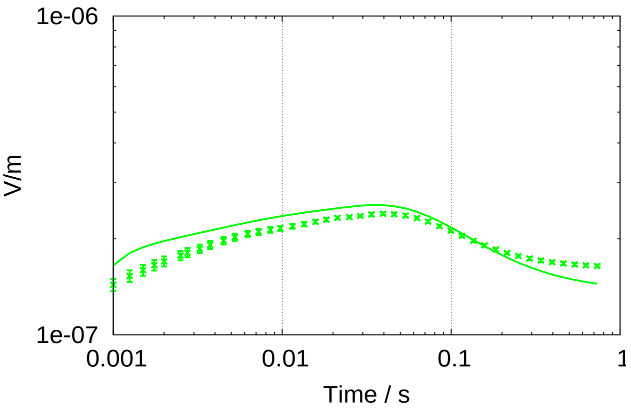
Tx(0.00,-2000.00)/E_x(-49.10,574.80)/χ=7.731



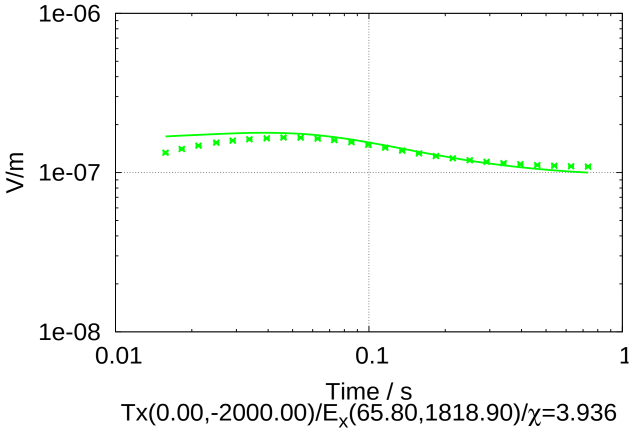
Tx(0.00,-2000.00)/E_x(-49.10,574.80)/χ=9.438



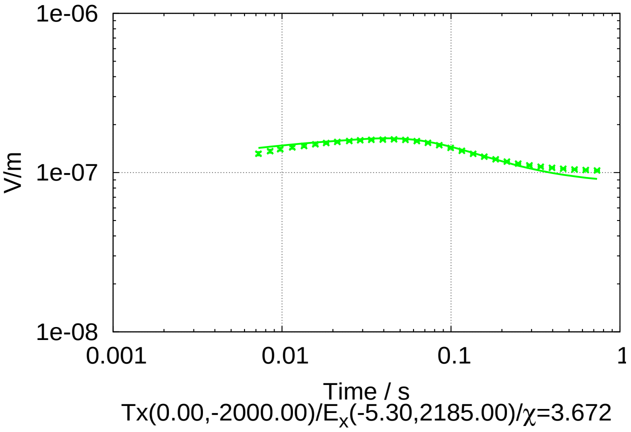
Tx(0.00,-2000.00)/E_x(-49.10,574.80)/χ=5.813



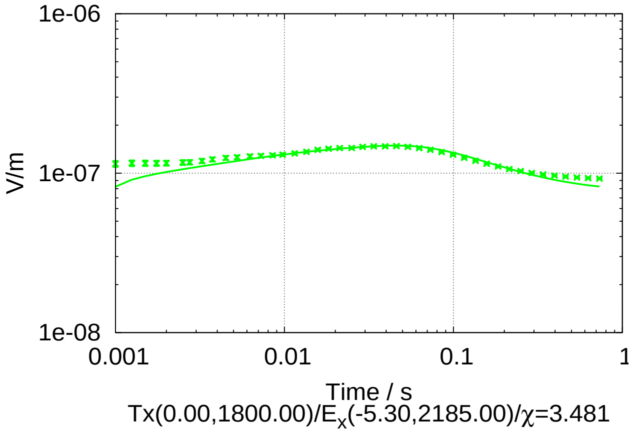
Tx(0.00,-2000.00)/E_x(-49.10,574.80)/χ=7.630



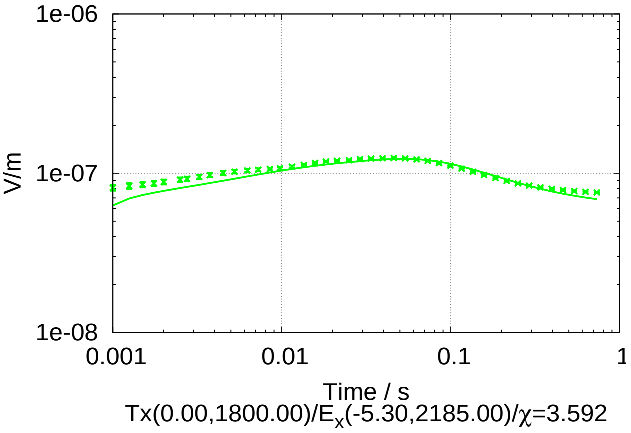
Tx(0.00,-2000.00)/E_x(-113.00,1603.70)/χ=4.480



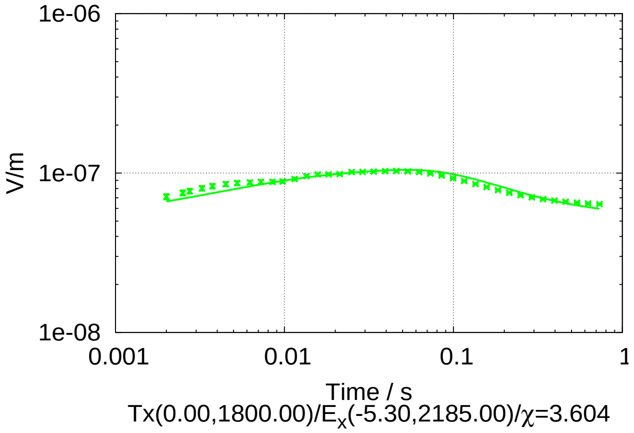
Tx(0.00,-2000.00)/E_x(65.80,1818.90)/χ=3.936



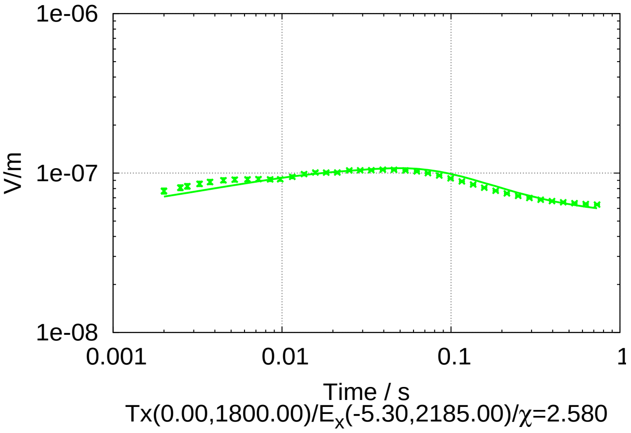
Tx(0.00,-2000.00)/E_x(-5.30,2185.00)/χ=3.672



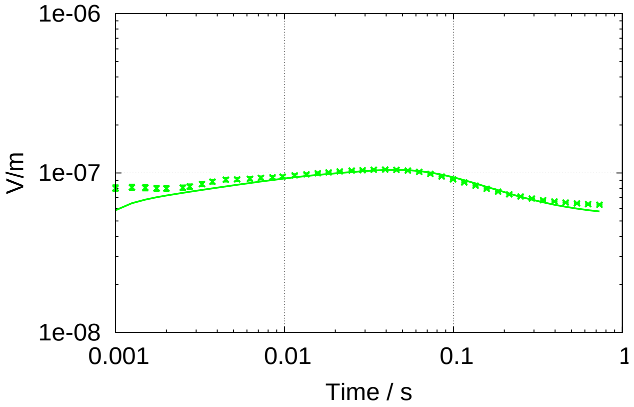
Tx(0.00,1800.00)/E_x(-5.30,2185.00)/χ=3.481



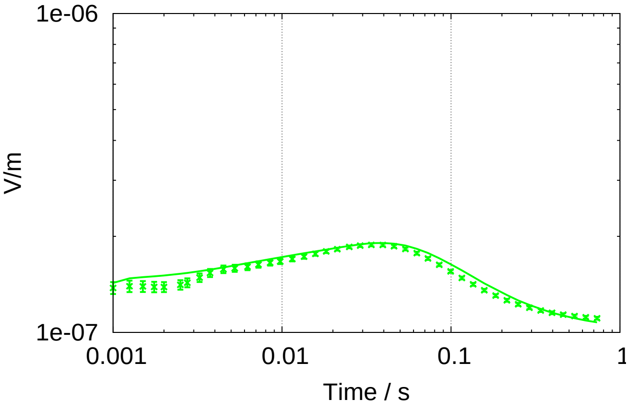
Tx(0.00,1800.00)/E_x(-5.30,2185.00)/χ=3.592



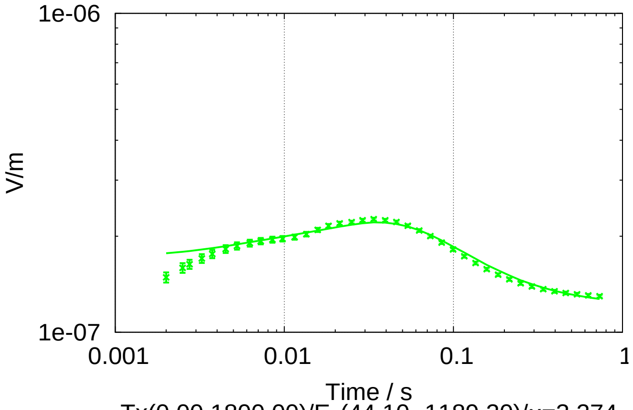
Tx(0.00,1800.00)/E_x(-5.30,2185.00)/χ=3.604



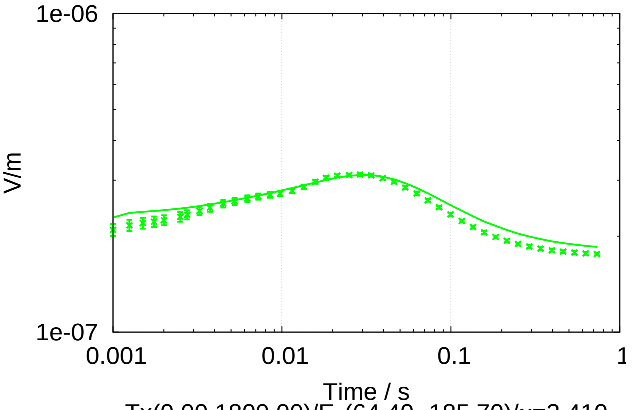
Tx(0.00,1800.00)/E_x(-5.30,2185.00)/χ=2.580



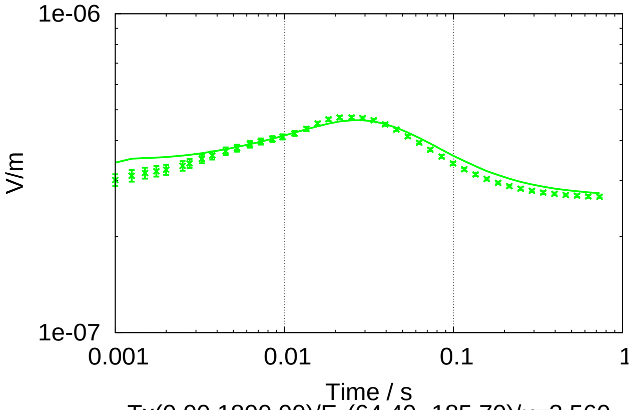
Tx(0.00,1800.00)/E_x(44.10,-1189.20)/χ=1.940



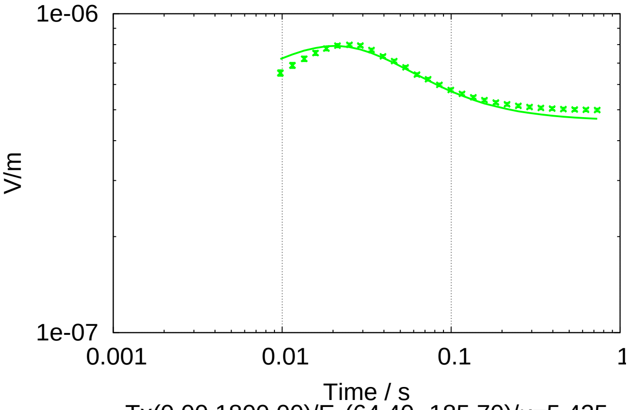
Tx(0.00,1800.00)/E_x(44.10,-1189.20)/χ=4.355



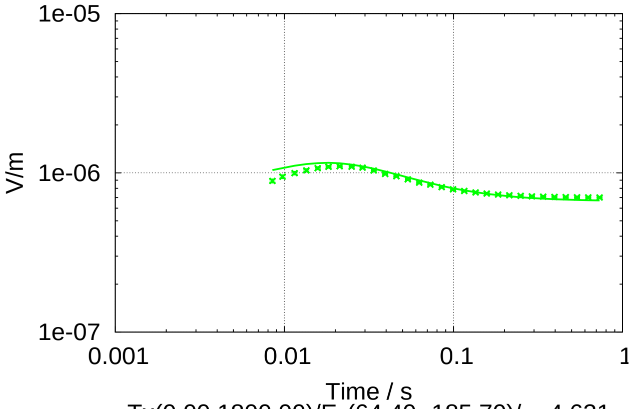
Tx(0.00,1800.00)/E_x(44.10,-1189.20)/χ=3.274



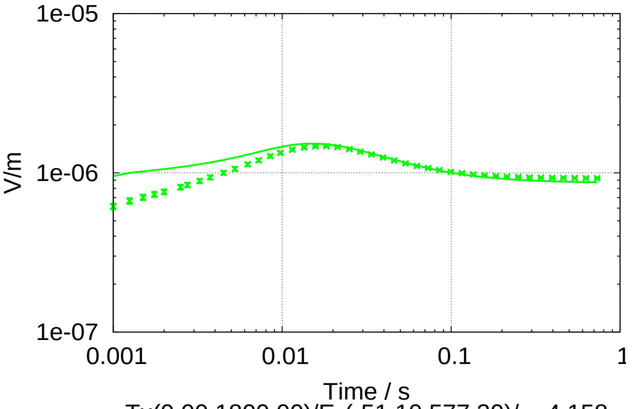
Tx(0.00,1800.00)/E_x(64.40,-185.70)/χ=3.410



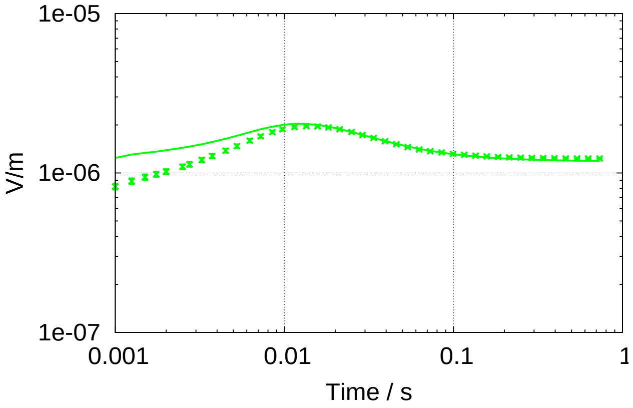
Tx(0.00,1800.00)/E_x(64.40,-185.70)/χ=3.560



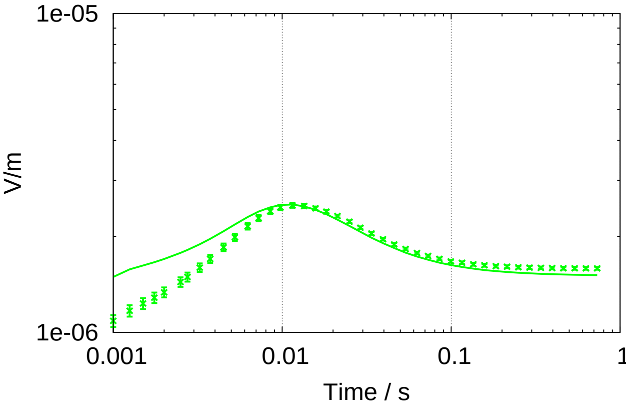
Tx(0.00,1800.00)/E_x(64.40,-185.70)/χ=5.425

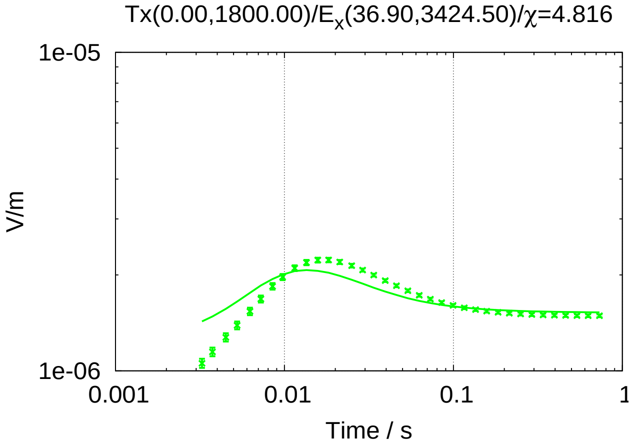


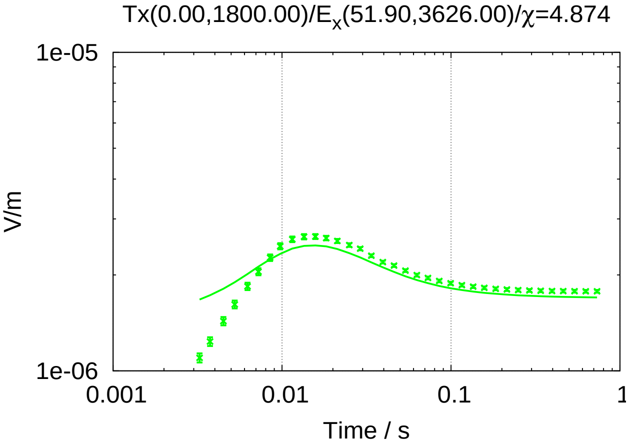
Tx(0.00,1800.00)/E_x(64.40,-185.70)/χ=4.631

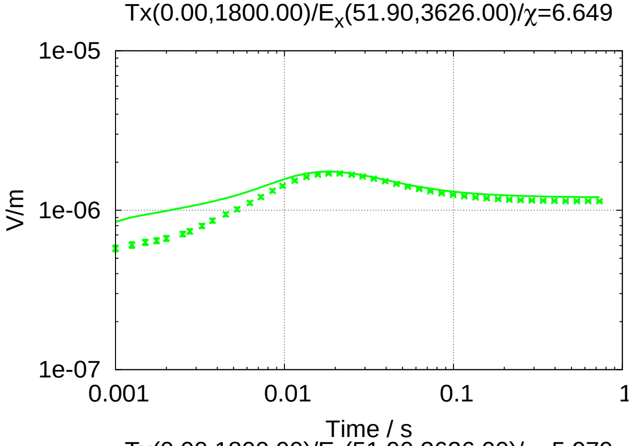


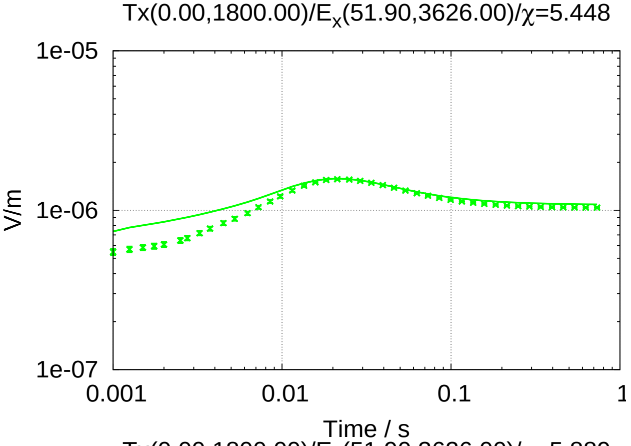
Tx(0.00,1800.00)/E_x(-51.10,577.30)/χ=4.158

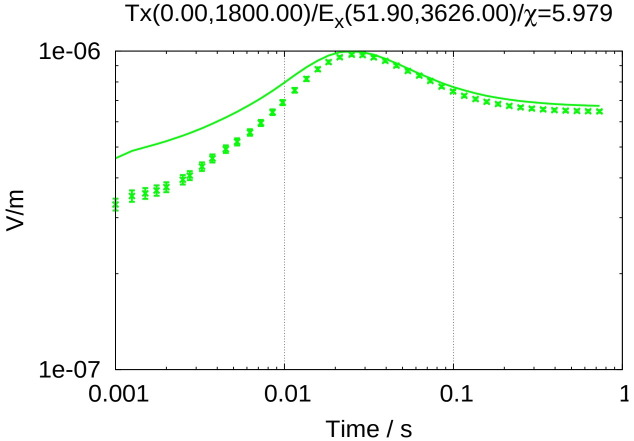


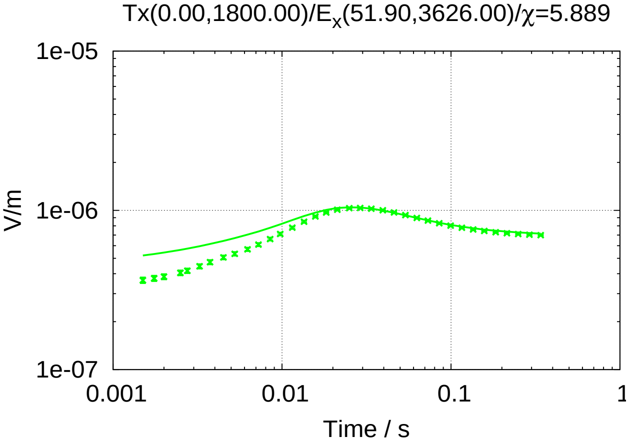

Tx(0.00,1800.00)/E_x(36.90,3424.50)/χ=4.816
1e-05
1e-06
V/m
0.001
0.01
0.1
1
Time / s

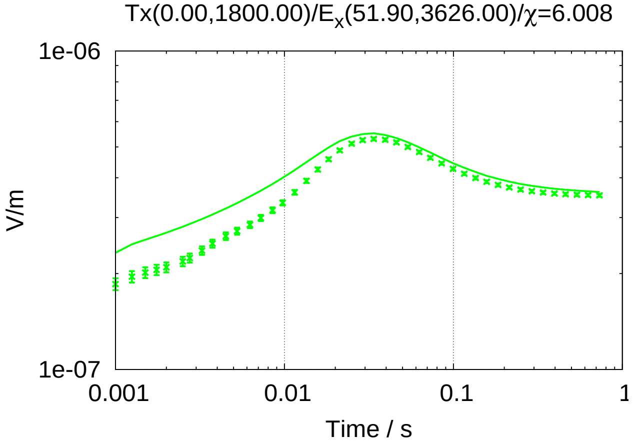

Tx(0.00,1800.00)/E_x(51.90,3626.00)/χ=4.874
1e-05
1e-06
V/m
0.001
0.01
0.1
1
Time / s

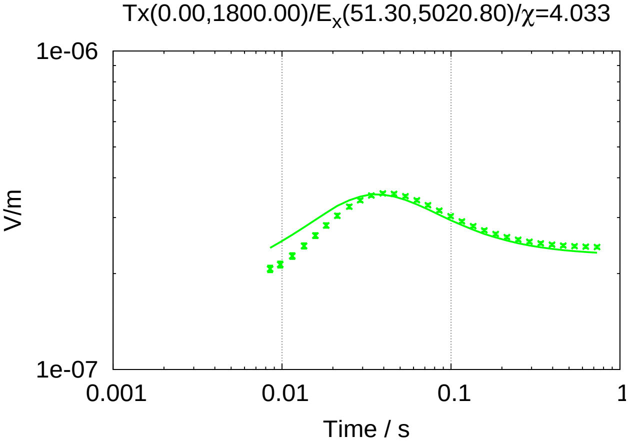

Tx(0.00,1800.00)/E_x(51.90,3626.00)/χ=6.649
1e-05
1e-06
1e-07
V/m
0.001
0.01
0.1
1
Time / s

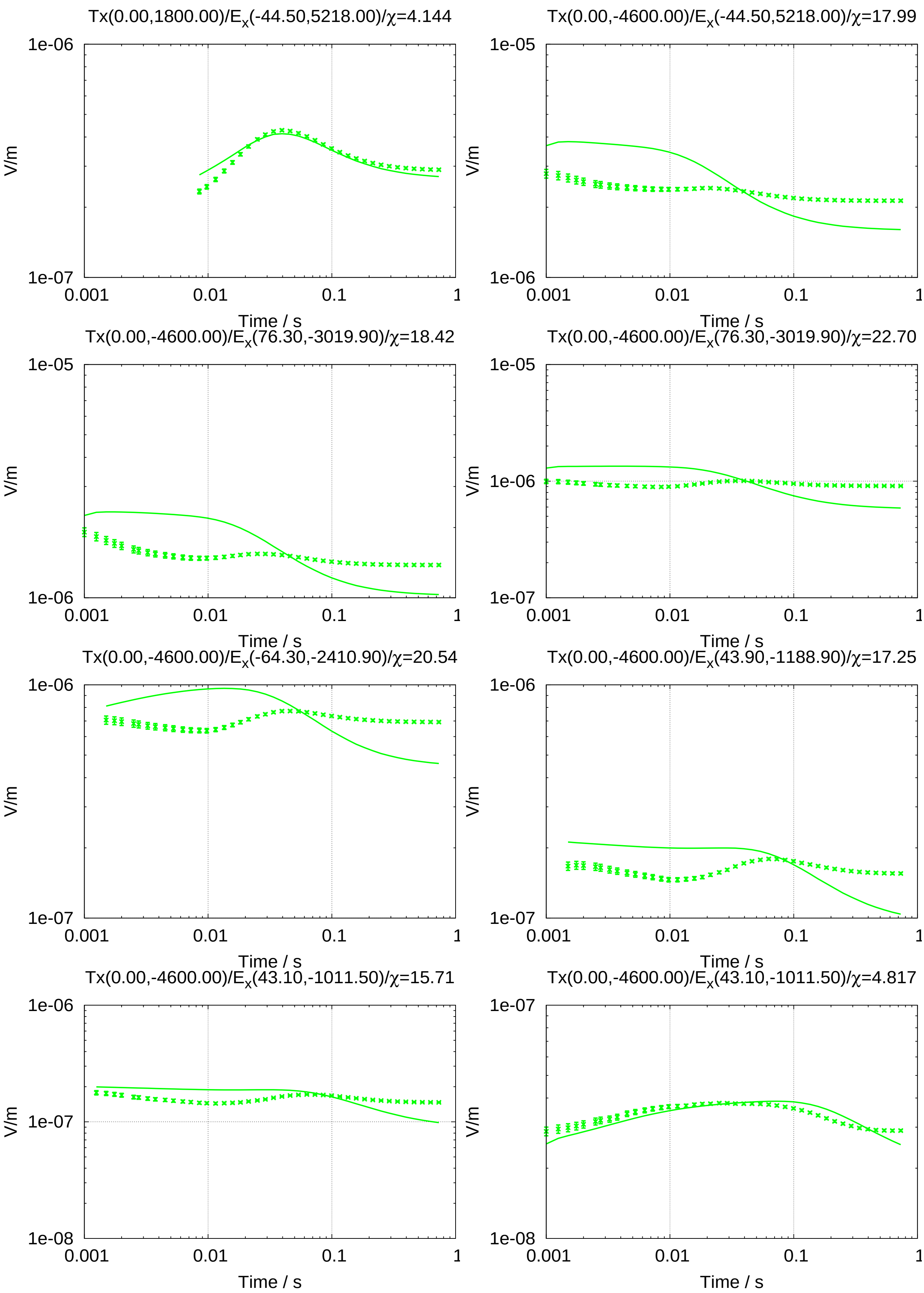

Tx(0.00,1800.00)/E_x(51.90,3626.00)/χ=5.448
1e-05
1e-06
1e-07
V/m
0.001
0.01
0.1
1
Time / s


Tx(0.00,1800.00)/E_x(51.90,3626.00)/χ=5.979
1e-06
1e-07
V/m
0.001
0.01
0.1
1
Time / s

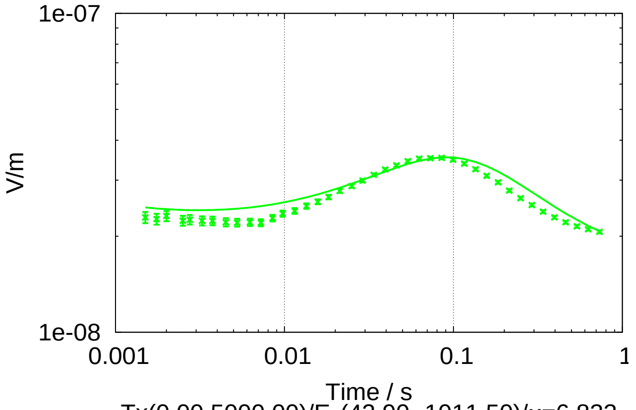

Tx(0.00,1800.00)/E_x(51.90,3626.00)/χ=5.889
1e-05
1e-06
1e-07
V/m
0.001
0.01
0.1
1
Time / s


Tx(0.00,1800.00)/E_x(51.90,3626.00)/χ=6.008
1e-06
1e-07
V/m
0.001
0.01
0.1
1
Time / s

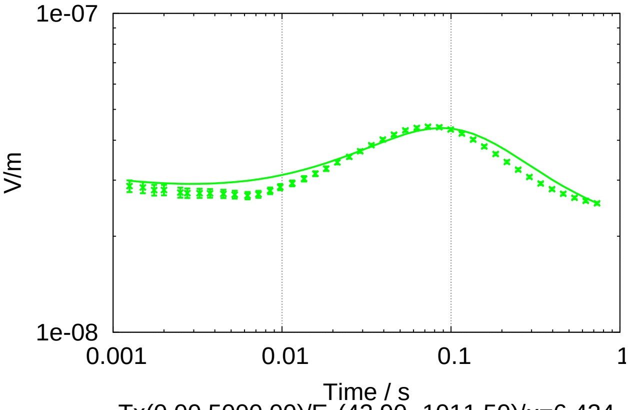

Tx(0.00,1800.00)/E_x(51.30,5020.80)/χ=4.033
1e-06
1e-07
V/m
0.001
0.01
0.1
1
Time / s


Tx(0.00,1800.00)/E_x(-44.50,5218.00)/χ=4.144
1e-06
1e-07
V/m
0.001
0.01
0.1
1
Time / s
Tx(0.00,-4600.00)/E_x(-44.50,5218.00)/χ=17.99
1e-05
1e-06
V/m
0.001
0.01
0.1
1
Time / s
Tx(0.00,-4600.00)/E_x(76.30,-3019.90)/χ=18.42
1e-05
1e-06
V/m
0.001
0.01
0.1
1
Time / s
Tx(0.00,-4600.00)/E_x(76.30,-3019.90)/χ=22.70
1e-05
1e-06
1e-07
V/m
0.001
0.01
0.1
1
Time / s
Tx(0.00,-4600.00)/E_x(-64.30,-2410.90)/χ=20.54
1e-06
1e-07
V/m
0.001
0.01
0.1
1
Time / s
Tx(0.00,-4600.00)/E_x(43.90,-1188.90)/χ=17.25
1e-06
1e-07
V/m
0.001
0.01
0.1
1
Time / s
Tx(0.00,-4600.00)/E_x(43.10,-1011.50)/χ=15.71
1e-06
1e-07
1e-08
V/m
0.001
0.01
0.1
1
Time / s
Tx(0.00,-4600.00)/E_x(43.10,-1011.50)/χ=4.817
1e-07
1e-08
V/m
0.001
0.01
0.1
1
Time / s

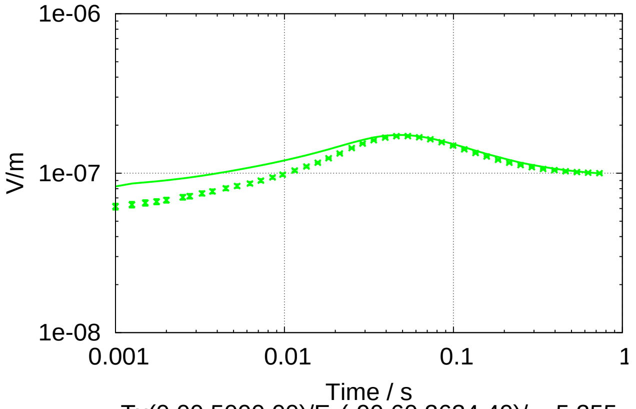
Tx(0.00,5000.00)/E_x(43.80,-1189.00)/χ=4.454



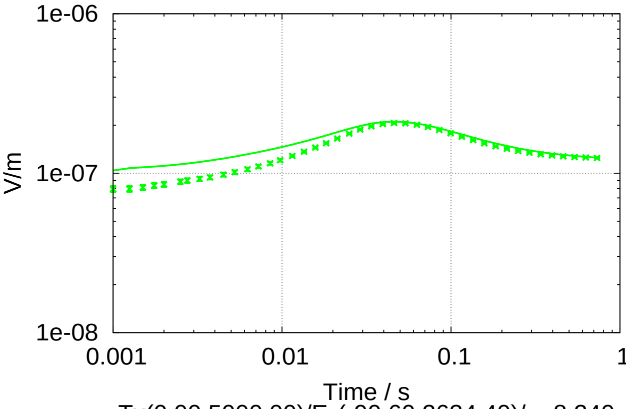
Tx(0.00,5000.00)/E_x(42.90,-1011.50)/χ=3.972



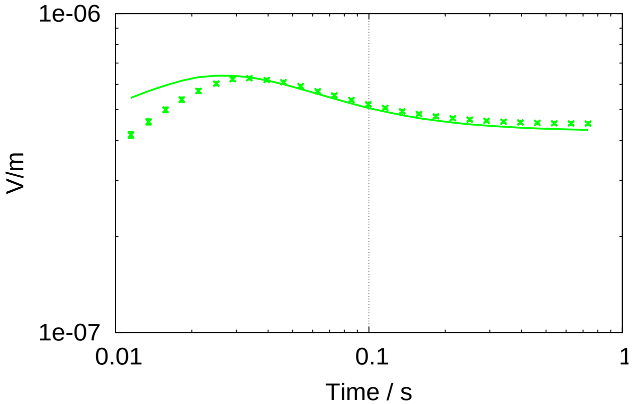
Tx(0.00,5000.00)/E_x(42.90,-1011.50)/χ=6.833



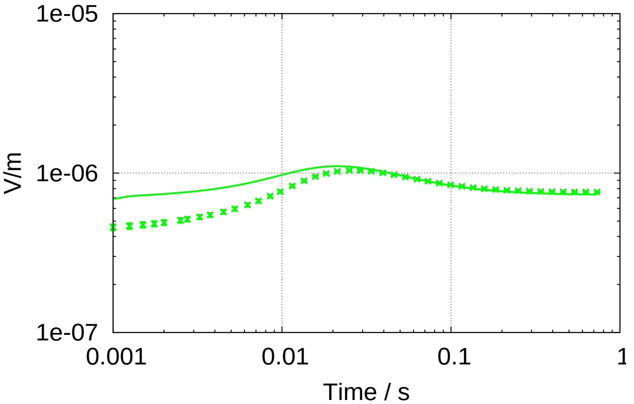
Tx(0.00,5000.00)/E_x(42.90,-1011.50)/χ=6.424



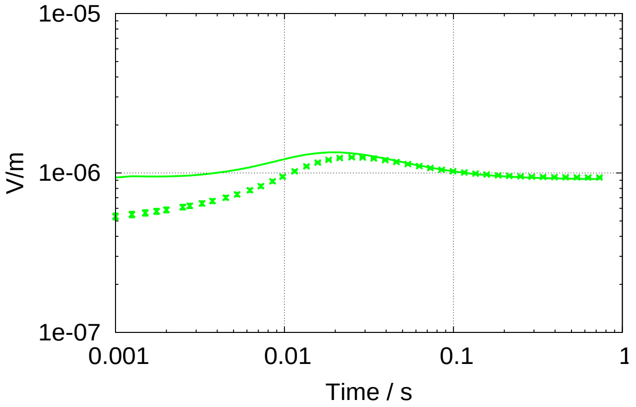
Tx(0.00,5000.00)/E_x(-90.60,2634.40)/χ=5.255



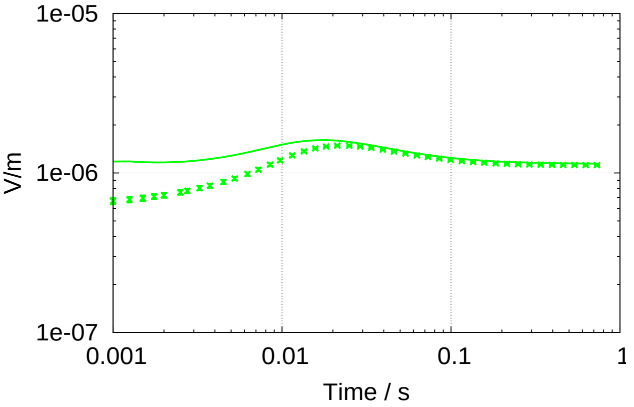
Tx(0.00,5000.00)/E_x(-90.60,2634.40)/χ=8.240



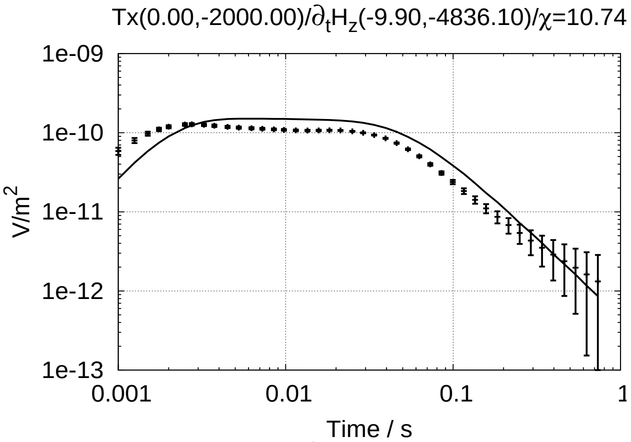
Tx(0.00,5000.00)/E_x(-90.60,2634.40)/χ=9.132

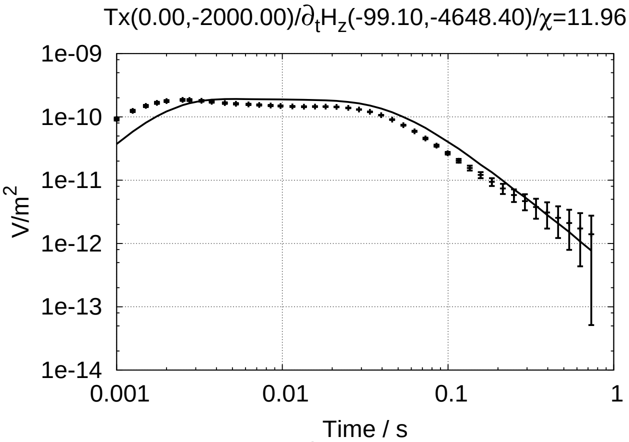


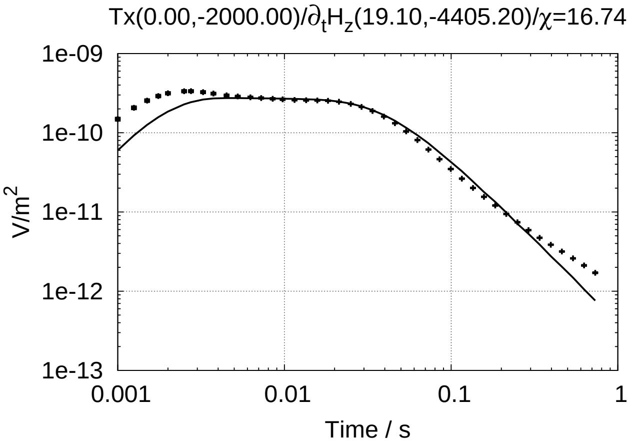
Tx(0.00,5000.00)/E_x(-90.60,2634.40)/χ=8.819

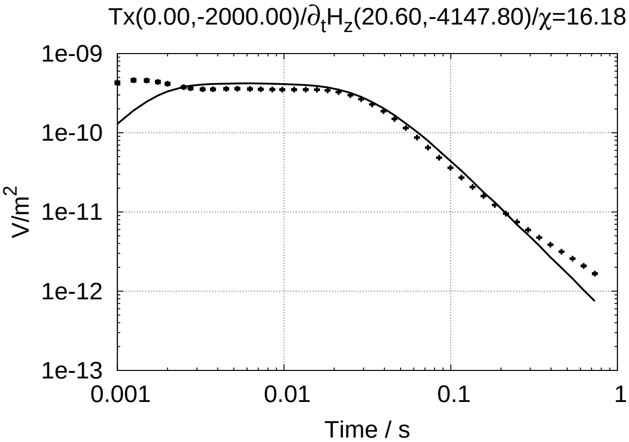


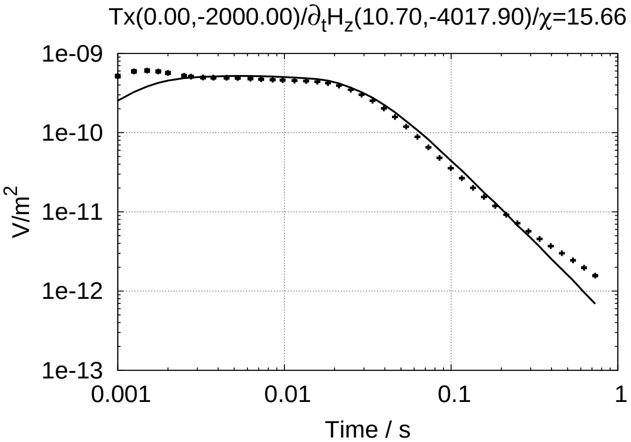
E.0.2 $\dot{\mathrm{H}}_\mathrm{y}$ and $\dot{\mathrm{H}}_\mathrm{z}$ data

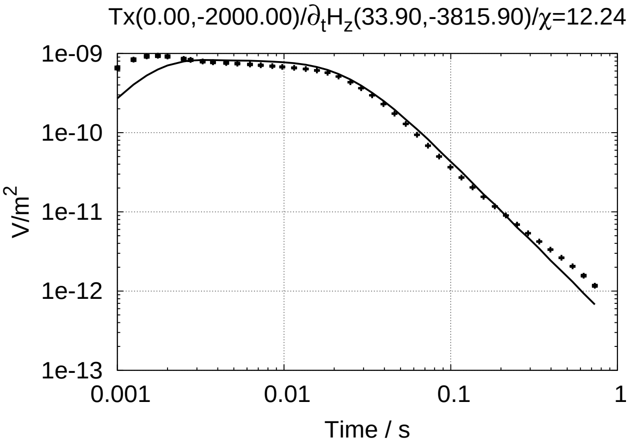


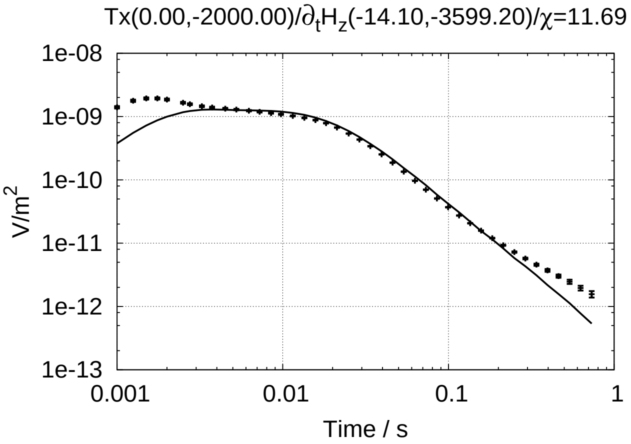


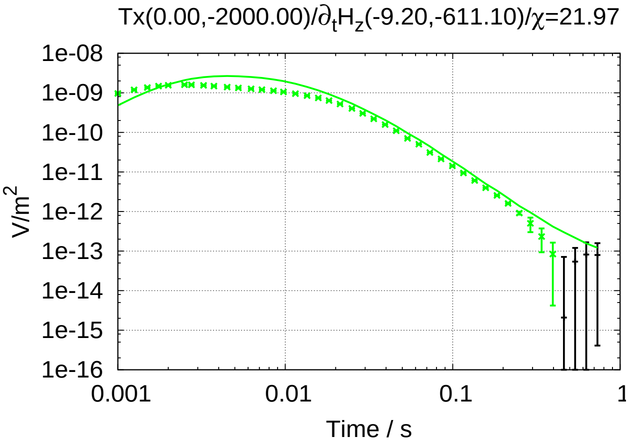




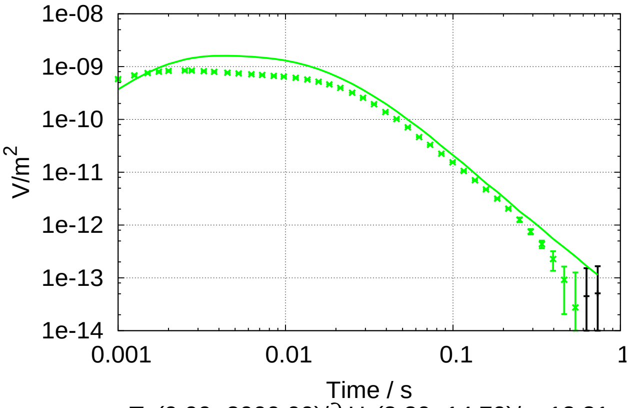




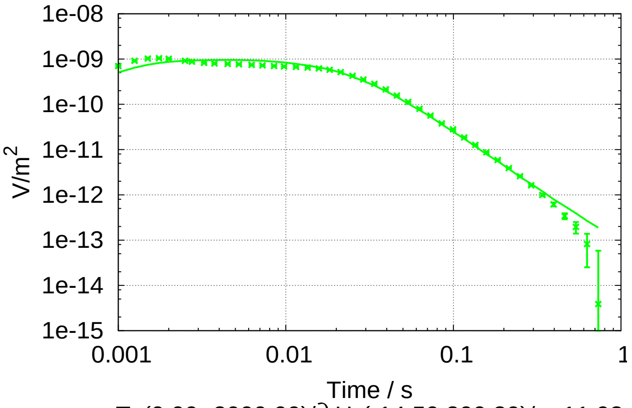




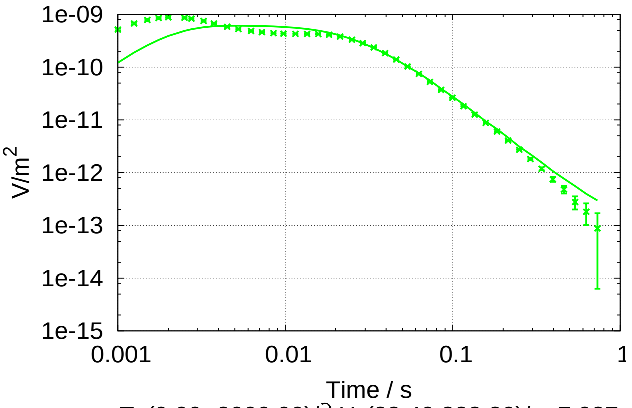
Tx(0.00,-2000.00)/$\partial_t H_z$(10.90,-396.50)/χ=26.42



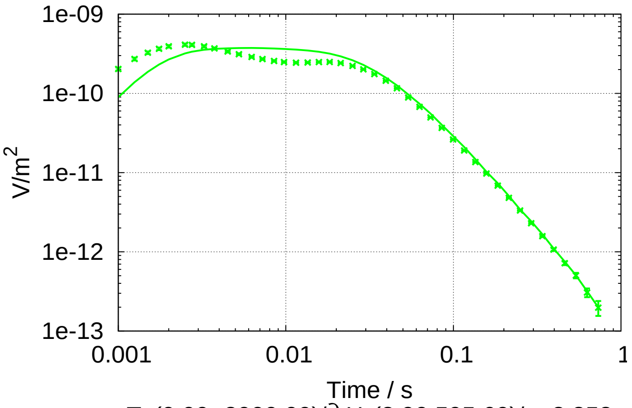
Tx(0.00,-2000.00)/$\partial_t H_z$(2.60,-209.80)/χ=6.773



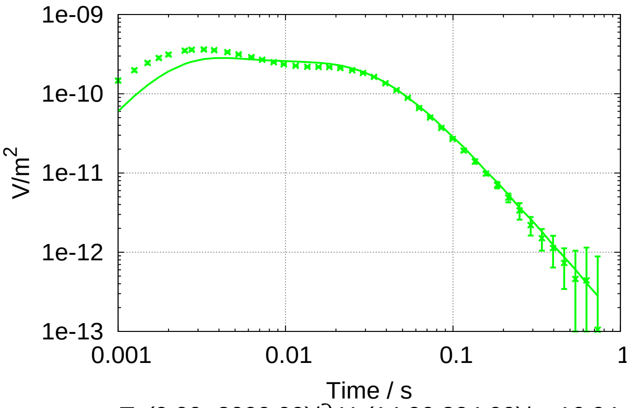
Tx(0.00,-2000.00)/$\partial_t H_z$(3.20,-14.70)/χ=12.31



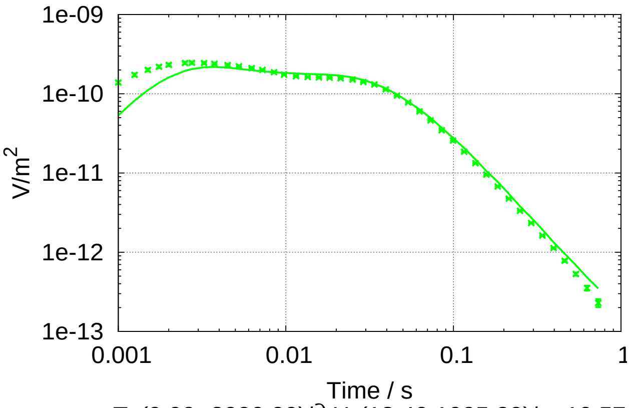
Tx(0.00,-2000.00)/$\partial_t H_z$(-14.50,200.80)/χ=11.93



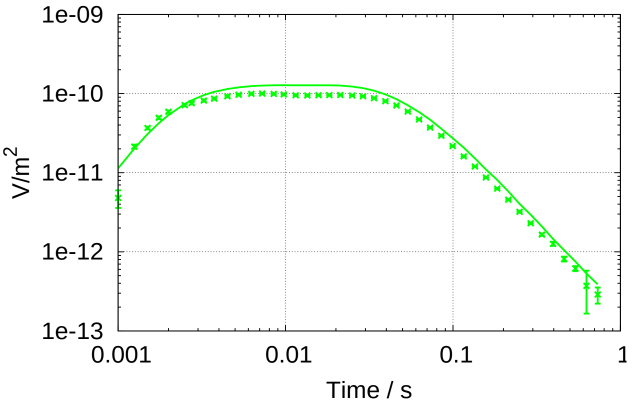
Tx(0.00,-2000.00)/$\partial_t H_z$(28.40,383.80)/χ=7.087



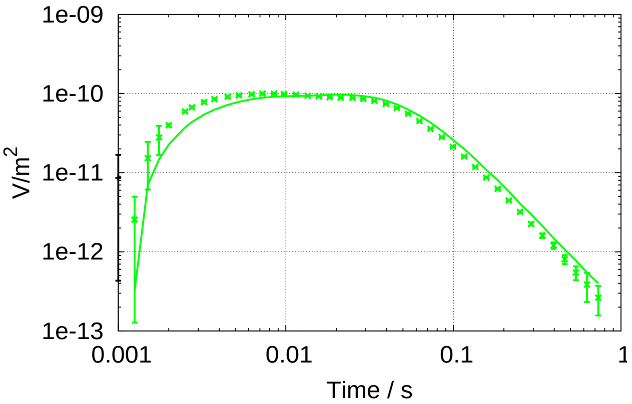
Tx(0.00,-2000.00)/$\partial_t H_z$(2.90,595.60)/χ=8.858



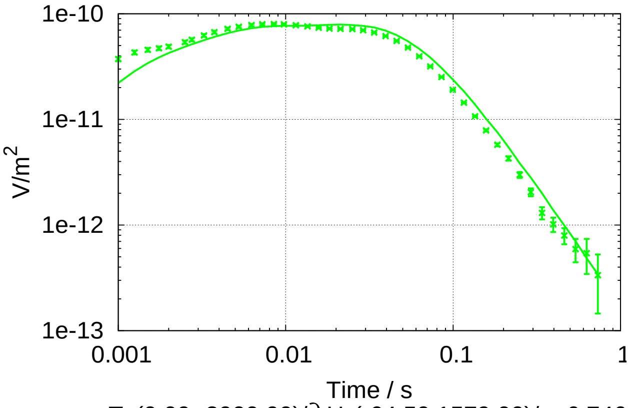
Tx(0.00,-2000.00)/$\partial_t H_z$(14.90,804.00)/χ=16.04



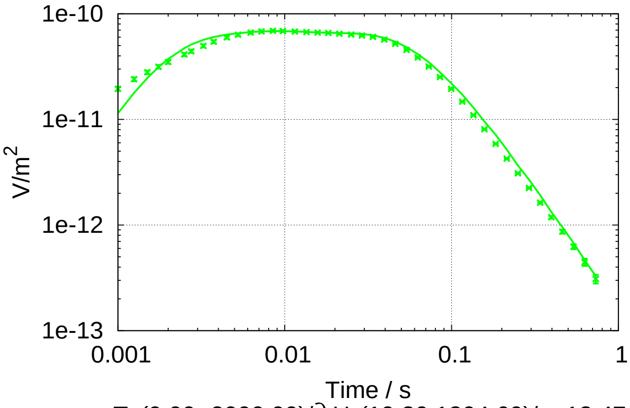
Tx(0.00,-2000.00)/$\partial_t H_z$(13.40,1005.20)/χ=10.57



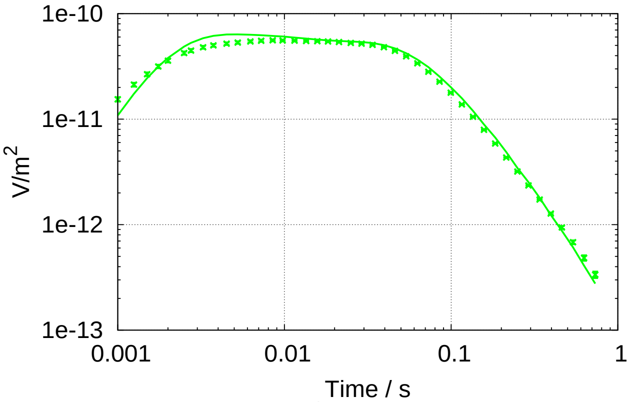
Tx(0.00,-2000.00)/$\partial_t H_z$(5.00,1192.90)/χ=9.493



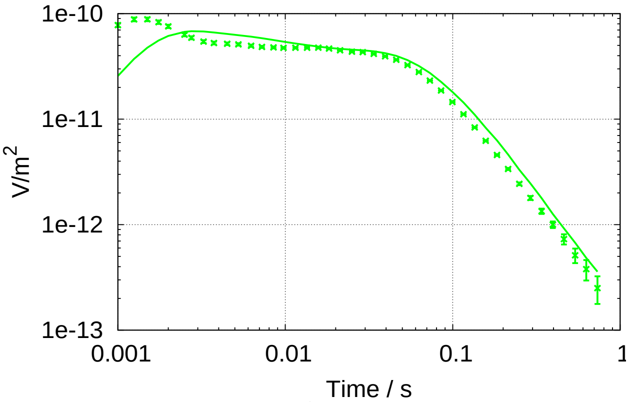
Tx(0.00,-2000.00)/$\partial_t H_z$(-31.10,1383.50)/χ=8.168



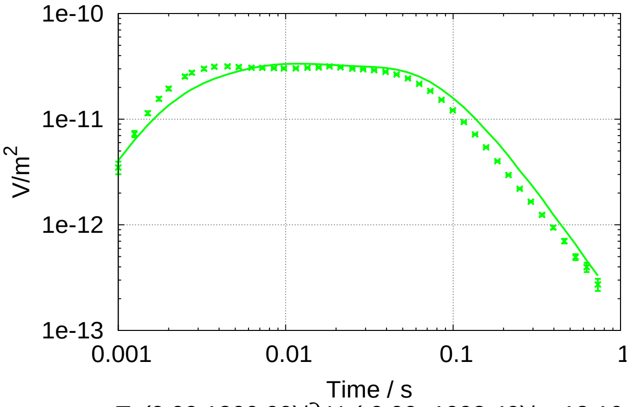
Tx(0.00,-2000.00)/$\partial_t H_z$(-64.50,1579.00)/χ=6.746



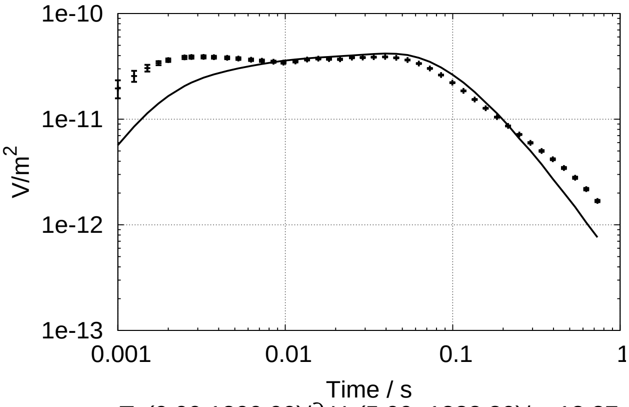
Tx(0.00,-2000.00)/$\partial_t H_z$(19.30,1804.60)/χ=12.47



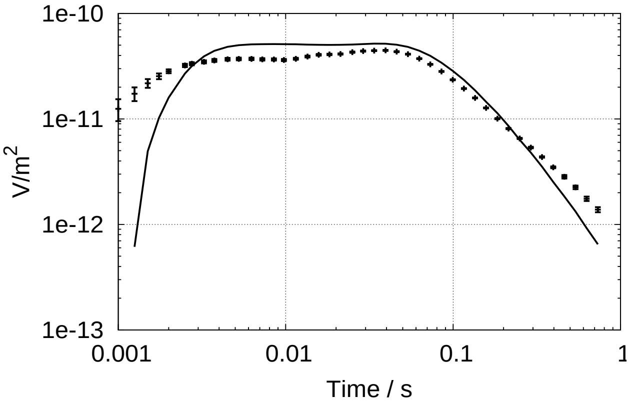
Tx(0.00,-2000.00)/$\partial_t H_z$(16.30,2221.90)/χ=16.56



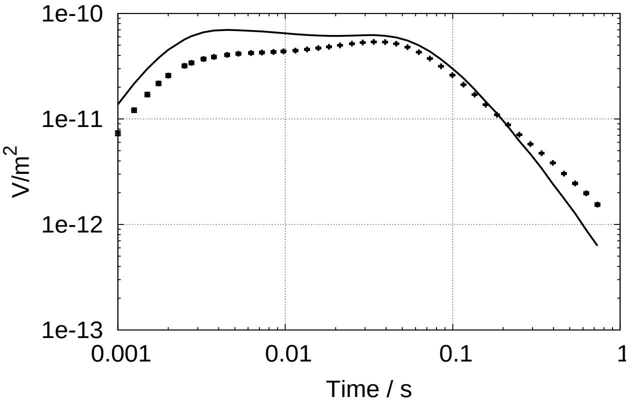
Tx(0.00,1800.00)/$\partial_t H_z$(12.80,-2205.80)/χ=12.90



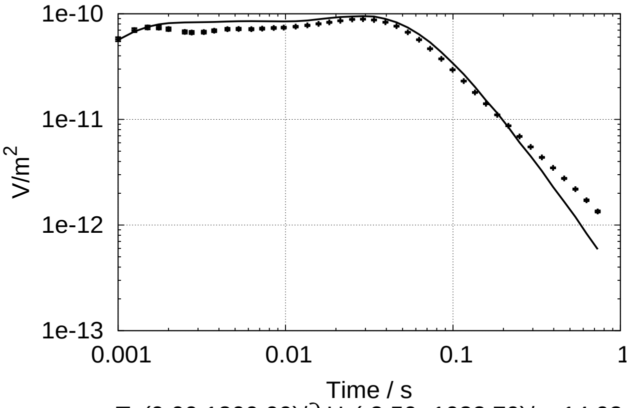
Tx(0.00,1800.00)/$\partial_t H_z$(-0.90,-1998.40)/χ=13.16



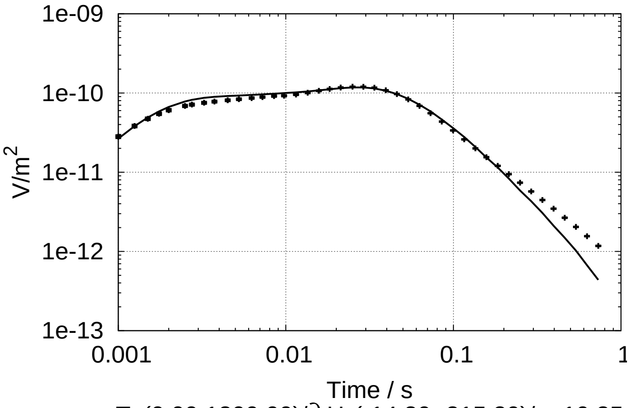
Tx(0.00,1800.00)/$\partial_t H_z$(5.00,-1822.30)/χ=18.87



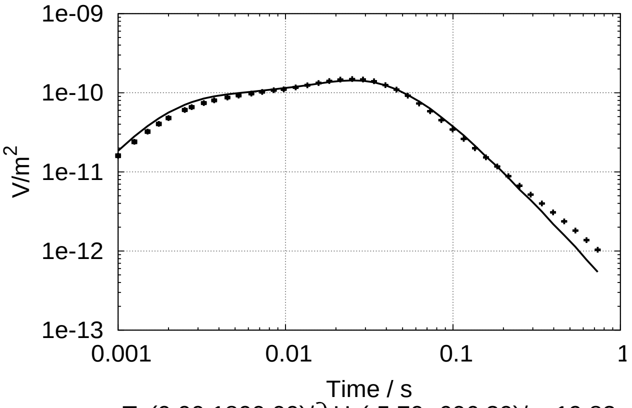
Tx(0.00,1800.00)/$\partial_t H_z$(6.20,-1401.70)/χ=14.64



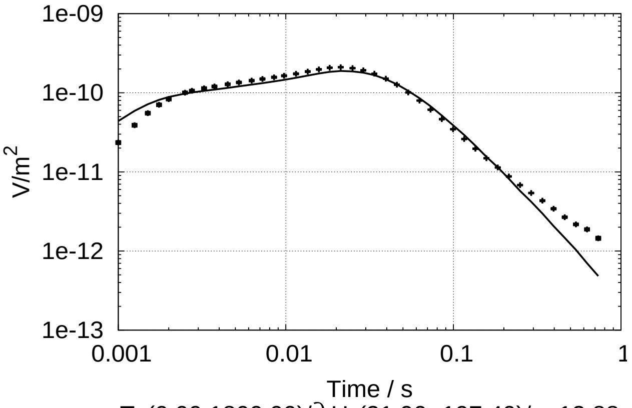
Tx(0.00,1800.00)/$\partial_t H_z$(-3.60,-1204.10)/χ=19.62



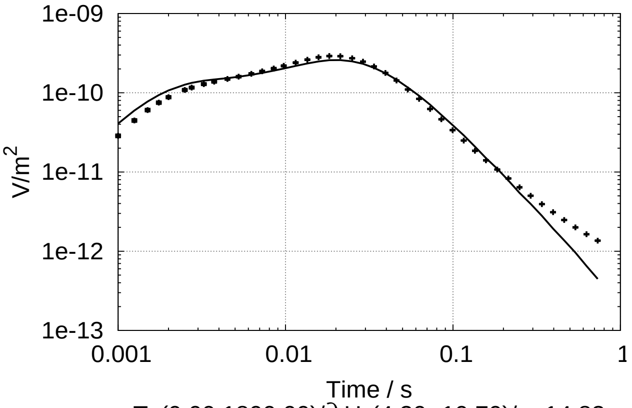
Tx(0.00,1800.00)/$\partial_t H_z$(-2.50,-1032.70)/χ=14.02



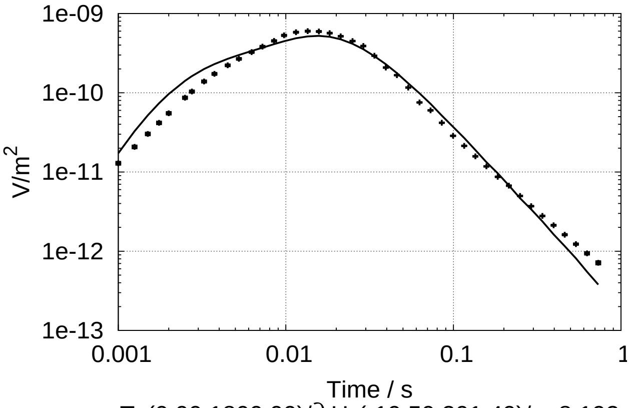
Tx(0.00,1800.00)/$\partial_t H_z$(-14.80,-815.20)/χ=10.35



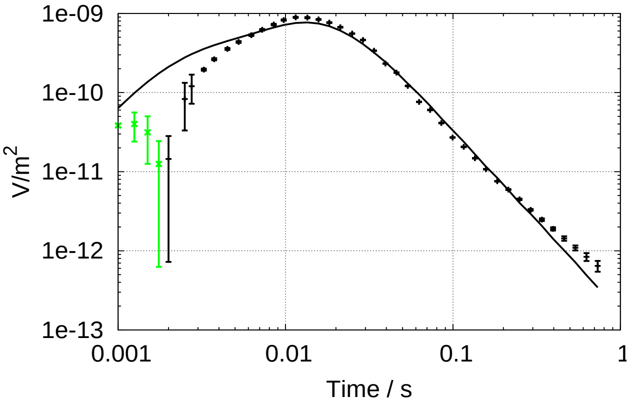
Tx(0.00,1800.00)/$\partial_t H_z$(-5.70,-606.30)/χ=19.82



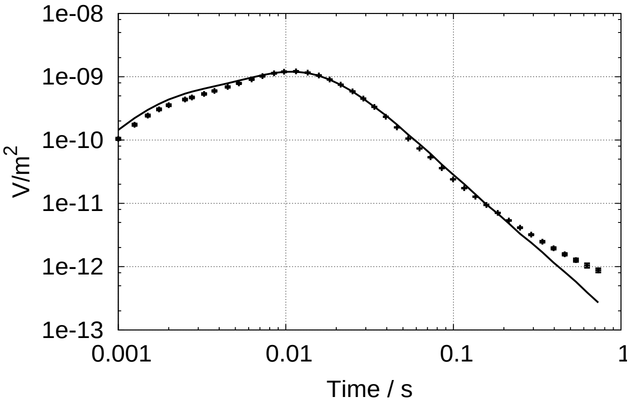
Tx(0.00,1800.00)/$\partial_t H_z$(21.90,-197.40)/χ=12.88

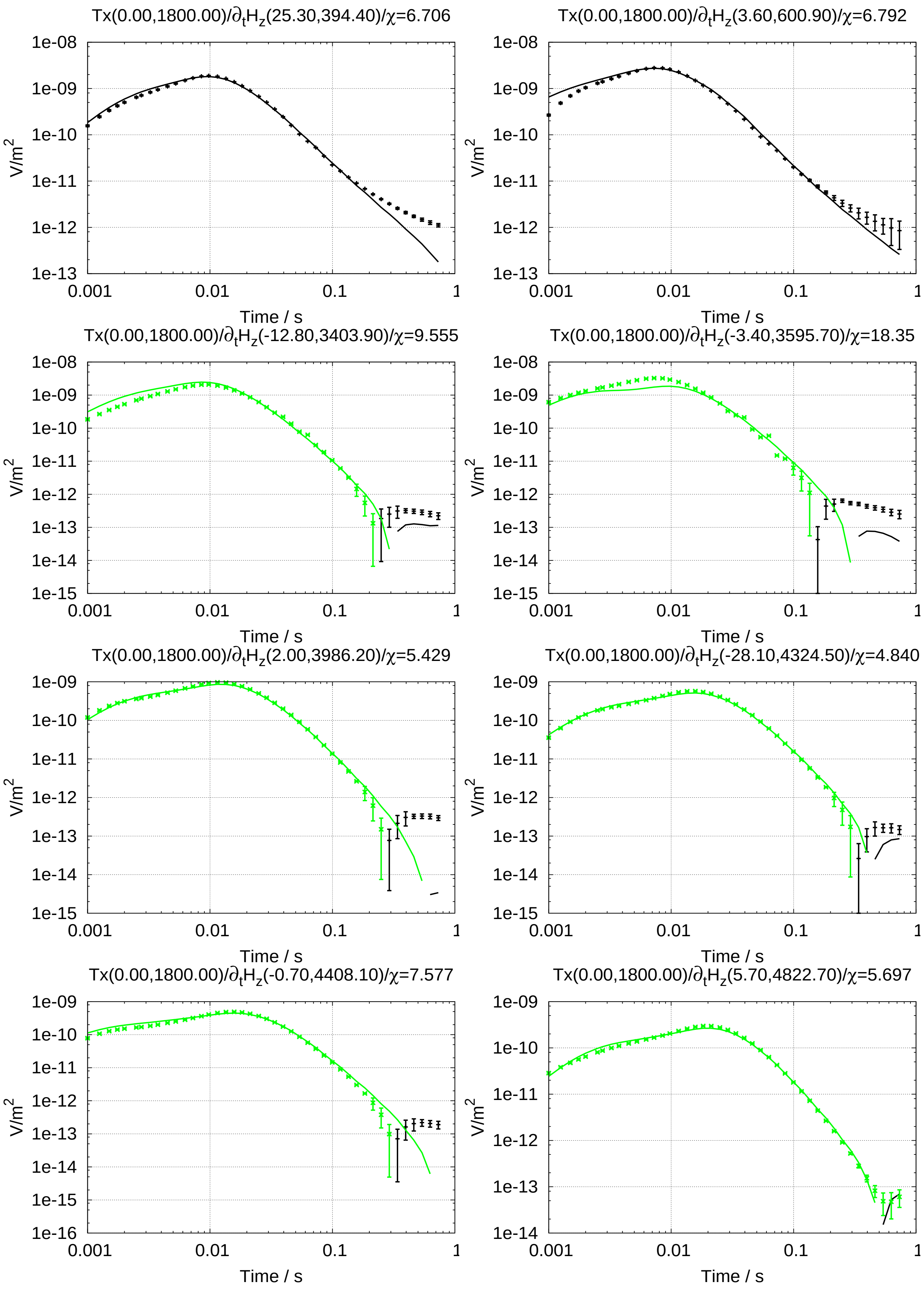


Tx(0.00,1800.00)/$\partial_t H_z$(4.20,-10.70)/χ=14.83



Tx(0.00,1800.00)/$\partial_t H_z$(-19.50,201.40)/χ=8.192





Tx(0.00,1800.00)/$\partial_t H_z$(25.30,394.40)/χ=6.706

Tx(0.00,1800.00)/$\partial_t H_z$(3.60,600.90)/χ=6.792

Tx(0.00,1800.00)/$\partial_t H_z$(-12.80,3403.90)/χ=9.555

Tx(0.00,1800.00)/$\partial_t H_z$(-3.40,3595.70)/χ=18.35

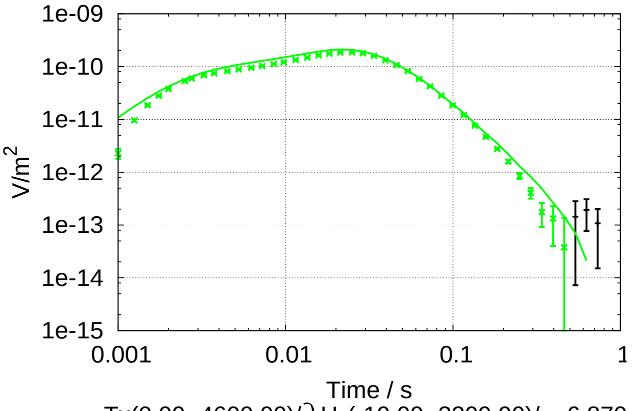
Tx(0.00,1800.00)/$\partial_t H_z$(2.00,3986.20)/χ=5.429

Tx(0.00,1800.00)/$\partial_t H_z$(-28.10,4324.50)/χ=4.840

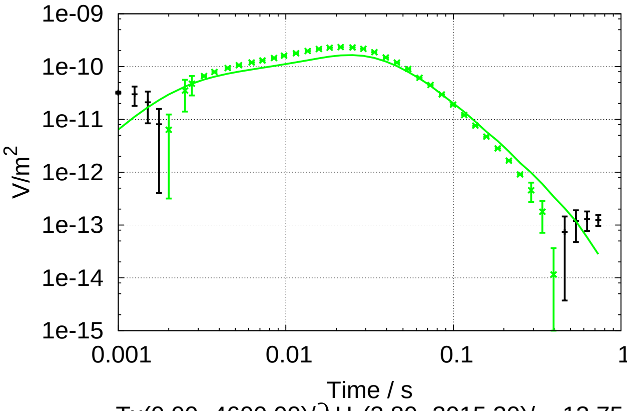
Tx(0.00,1800.00)/$\partial_t H_z$(-0.70,4408.10)/χ=7.577

Tx(0.00,1800.00)/$\partial_t H_z$(5.70,4822.70)/χ=5.697

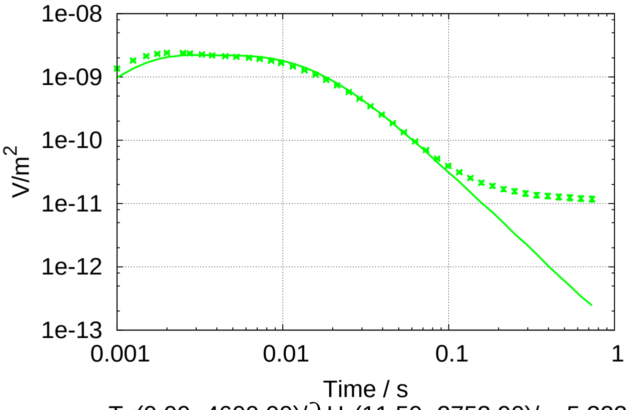
Tx(0.00,1800.00)/$\partial_t H_z$(-1.00,4999.00)/χ=6.705



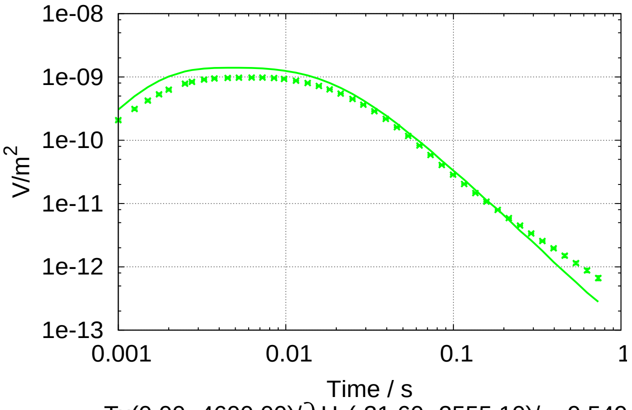
Tx(0.00,1800.00)/$\partial_t H_z$(-0.30,5199.20)/χ=19.77



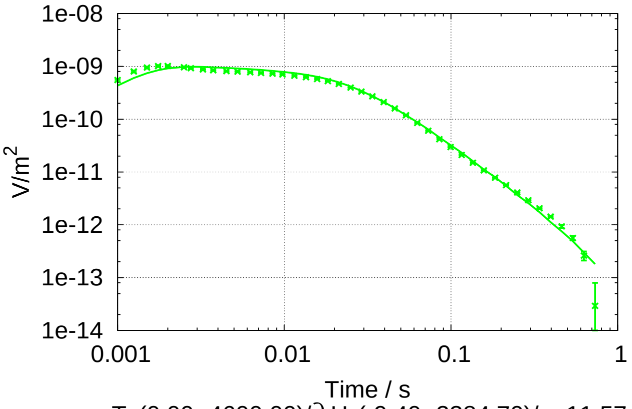
Tx(0.00,-4600.00)/$\partial_t H_z$(-10.00,-3209.90)/χ=6.870



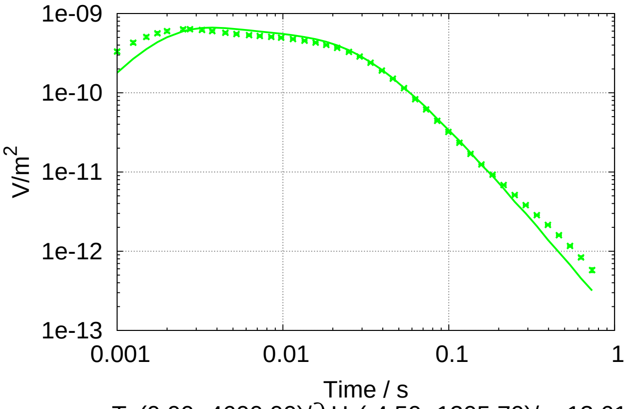
Tx(0.00,-4600.00)/$\partial_t H_z$(3.80,-3015.30)/χ=13.75



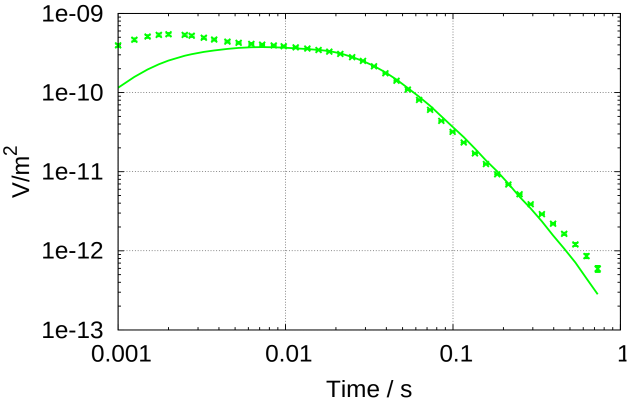
Tx(0.00,-4600.00)/$\partial_t H_z$(11.50,-2753.90)/χ=5.323



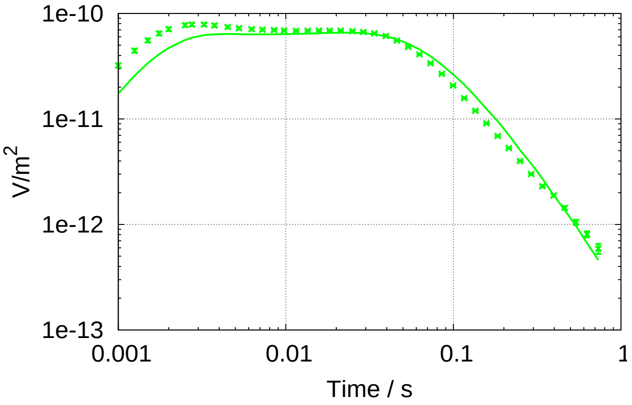
Tx(0.00,-4600.00)/$\partial_t H_z$(-31.60,-2555.10)/χ=9.549

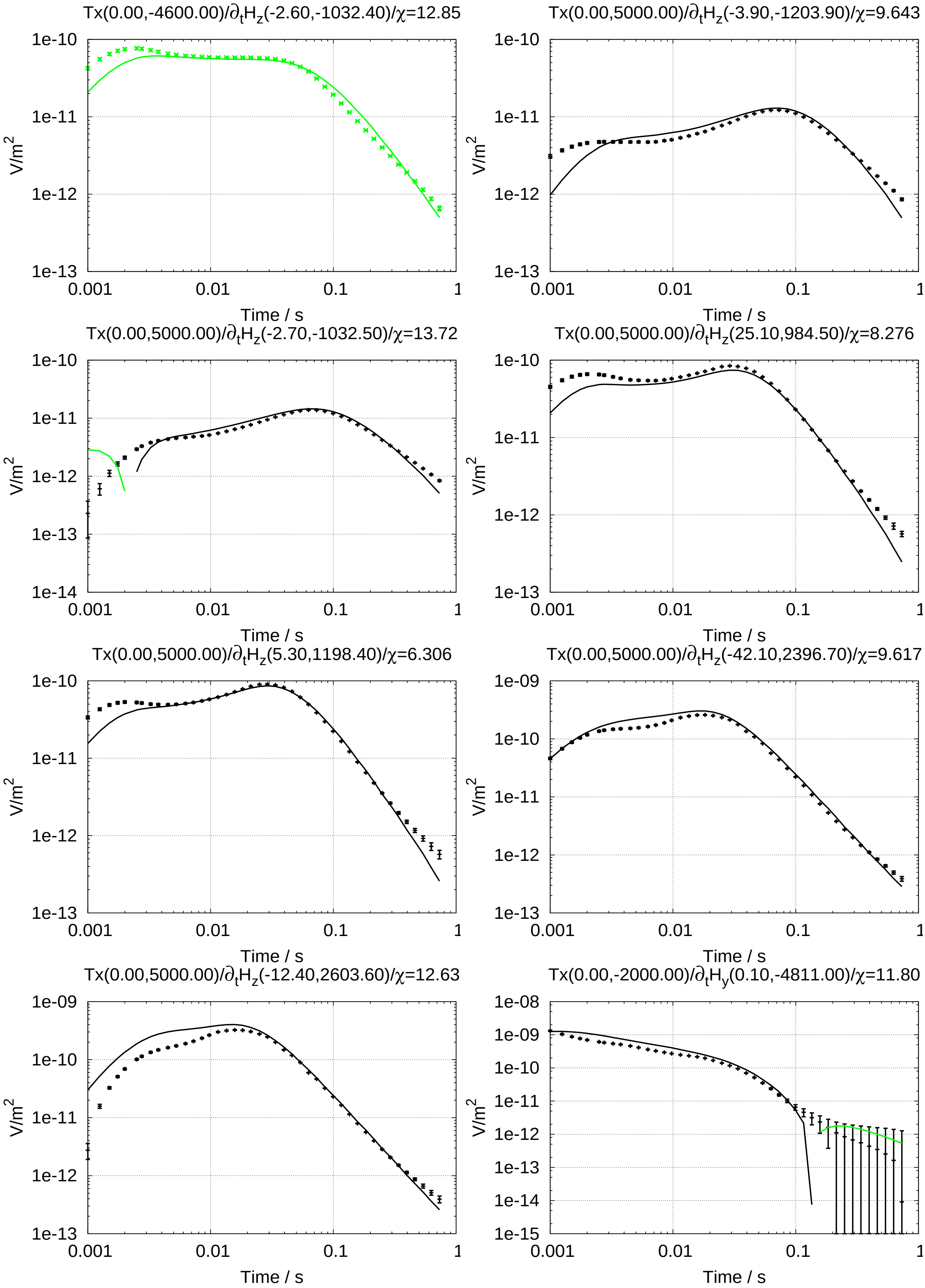


Tx(0.00,-4600.00)/$\partial_t H_z$(-9.40,-2384.70)/χ=11.57



Tx(0.00,-4600.00)/$\partial_t H_z$(-4.50,-1205.70)/χ=13.61





Tx(0.00,-4600.00)/$\partial_t H_z$(-2.60,-1032.40)/χ=12.85

Tx(0.00,5000.00)/$\partial_t H_z$(-3.90,-1203.90)/χ=9.643

Tx(0.00,5000.00)/$\partial_t H_z$(-2.70,-1032.50)/χ=13.72

Tx(0.00,5000.00)/$\partial_t H_z$(25.10,984.50)/χ=8.276

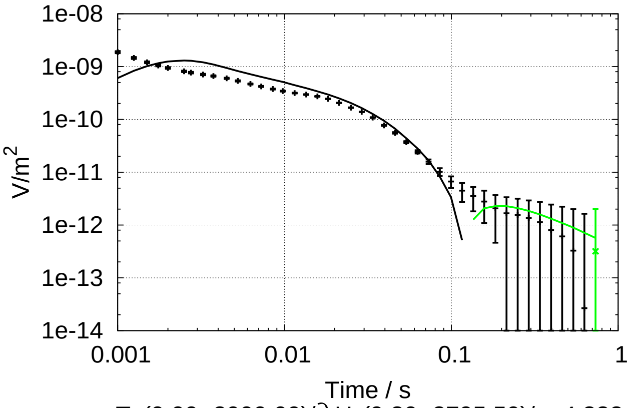
Tx(0.00,5000.00)/$\partial_t H_z$(5.30,1198.40)/χ=6.306

Tx(0.00,5000.00)/$\partial_t H_z$(-42.10,2396.70)/χ=9.617

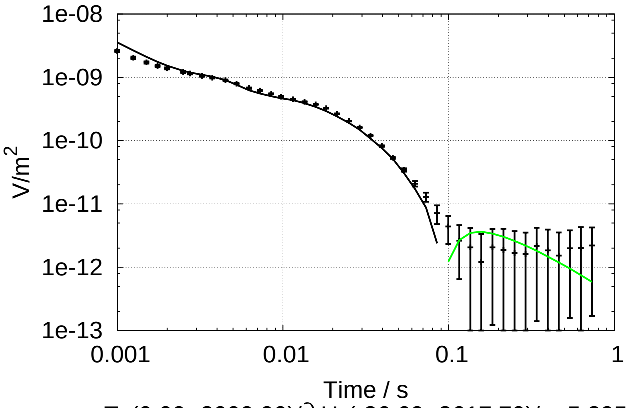
Tx(0.00,5000.00)/$\partial_t H_z$(-12.40,2603.60)/χ=12.63

Tx(0.00,-2000.00)/$\partial_t H_y$(0.10,-4811.00)/χ=11.80

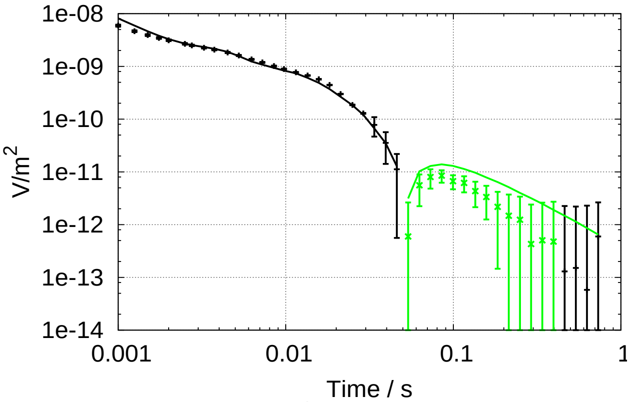
$Tx(0.00,-2000.00)/\partial_t H_y(-77.20,-4628.60)/\chi=11.81$



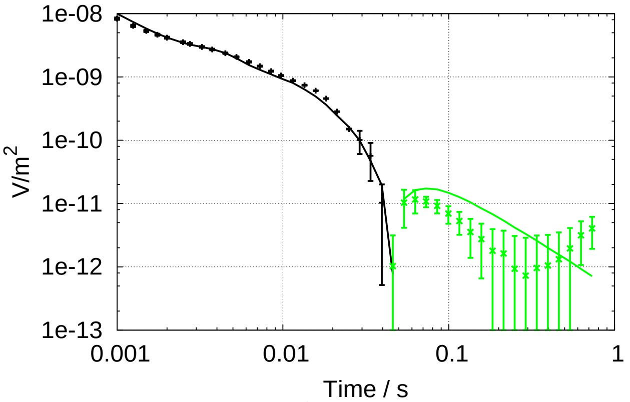
$Tx(0.00,-2000.00)/\partial_t H_y(-16.60,-4401.00)/\chi=3.684$



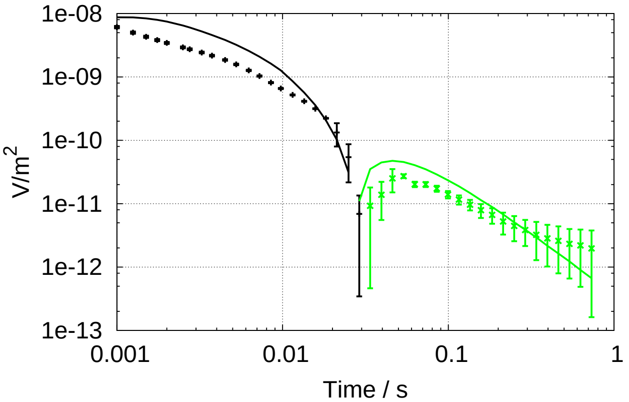
$Tx(0.00,-2000.00)/\partial_t H_y(9.30,-3795.50)/\chi=4.332$



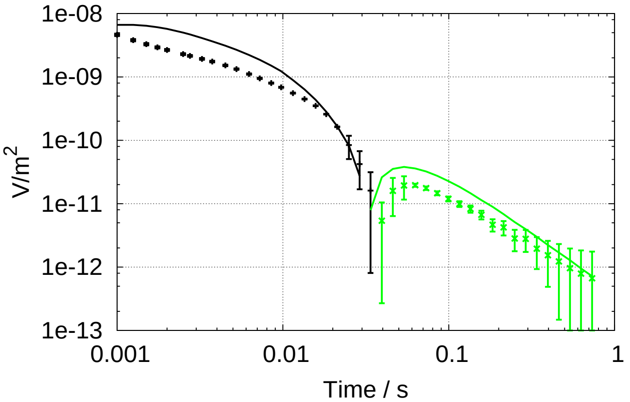
$Tx(0.00,-2000.00)/\partial_t H_y(-36.60,-3617.70)/\chi=5.295$



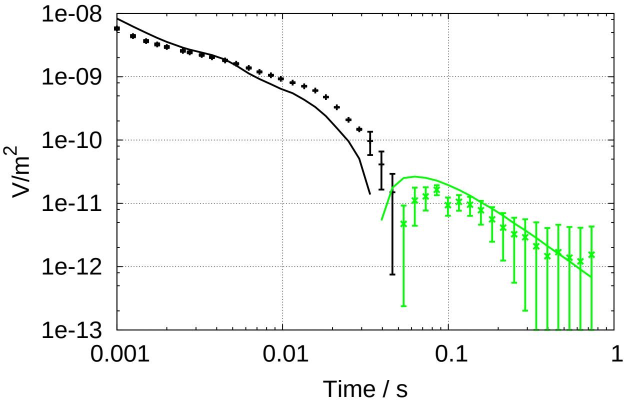
$Tx(0.00,-2000.00)/\partial_t H_y(-36.60,-3617.70)/\chi=15.64$



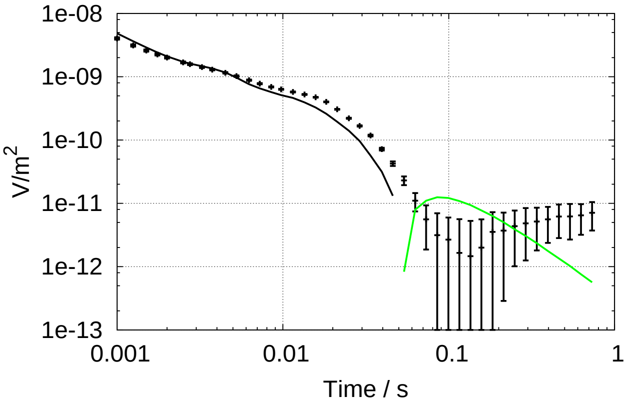
$Tx(0.00,-2000.00)/\partial_t H_y(-36.60,-3617.70)/\chi=15.59$



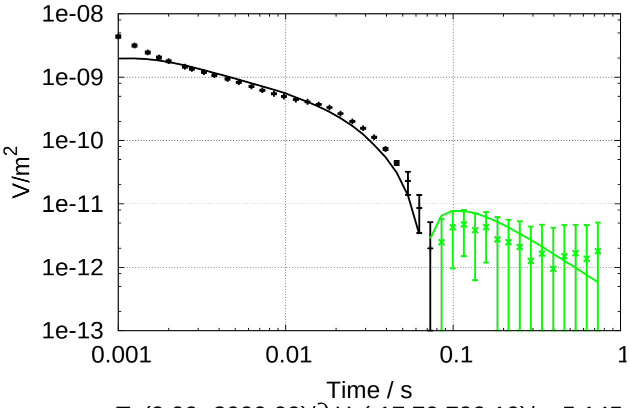
$Tx(0.00,-2000.00)/\partial_t H_y(-36.60,-3617.70)/\chi=18.50$



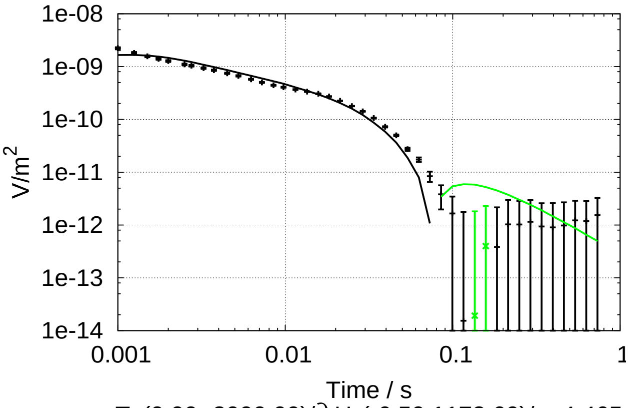
$Tx(0.00,-2000.00)/\partial_t H_y(-36.60,-3617.70)/\chi=10.88$



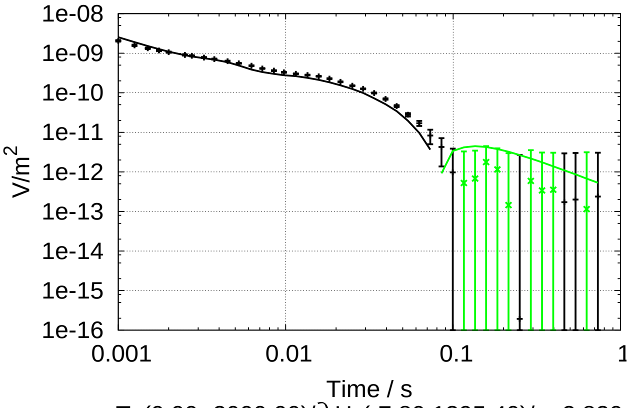
Tx(0.00,-2000.00)/$\partial_t H_y$(27.40,413.90)/χ=5.525



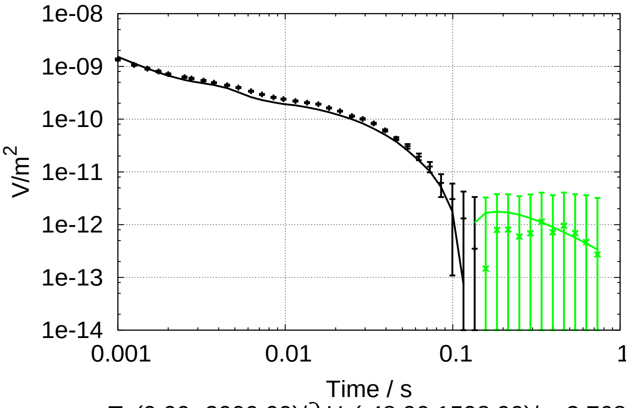
Tx(0.00,-2000.00)/$\partial_t H_y$(-23.00,582.50)/χ=5.358



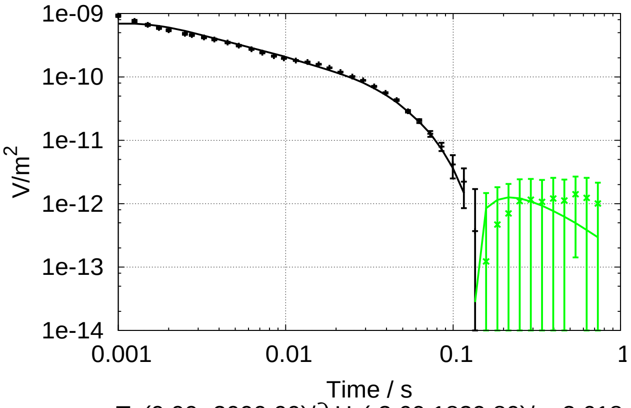
Tx(0.00,-2000.00)/$\partial_t H_y$(-17.70,790.10)/χ=5.145



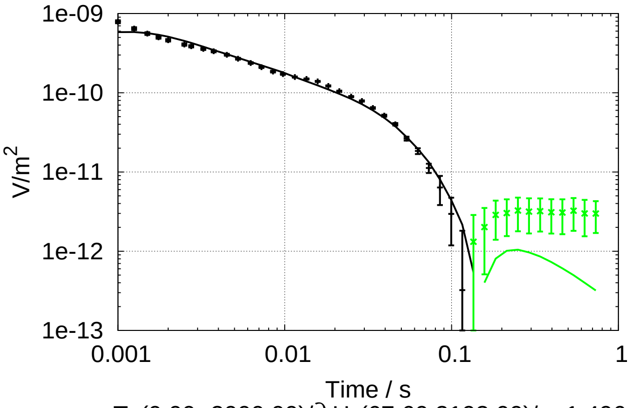
Tx(0.00,-2000.00)/$\partial_t H_y$(-6.50,1172.60)/χ=4.465



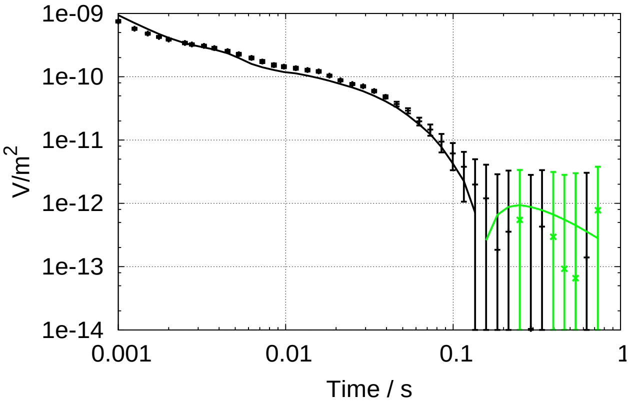
Tx(0.00,-2000.00)/$\partial_t H_y$(-7.80,1395.40)/χ=2.820



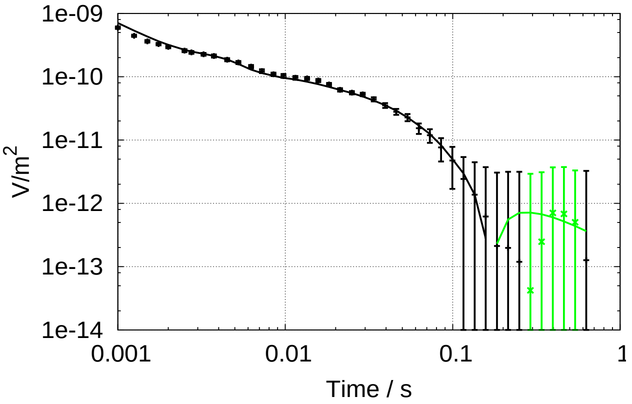
Tx(0.00,-2000.00)/$\partial_t H_y$(-48.90,1598.90)/χ=2.768



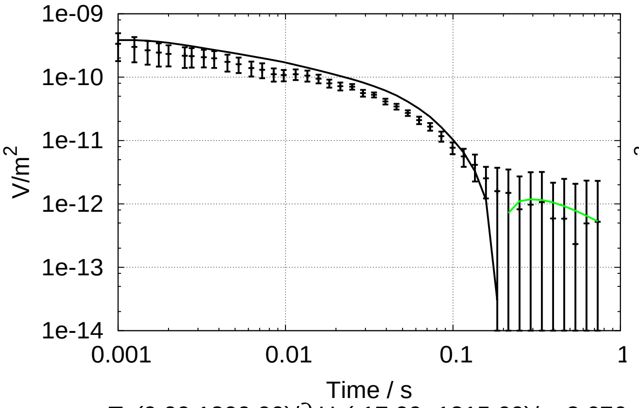
Tx(0.00,-2000.00)/$\partial_t H_y$(-3.60,1829.80)/χ=2.618



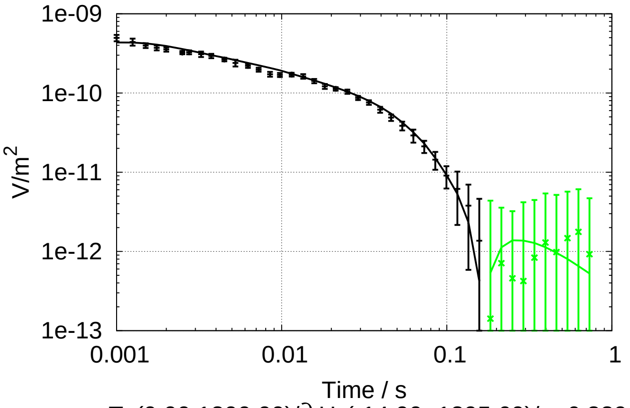
Tx(0.00,-2000.00)/$\partial_t H_y$(67.60,2193.00)/χ=1.406



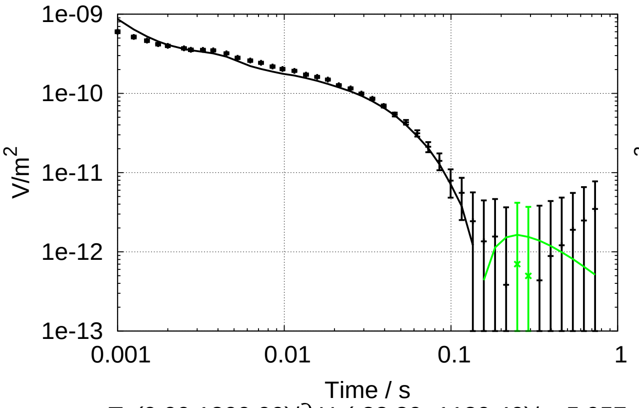
Tx(0.00,1800.00)/$\partial_t H_y$(-0.60,-2174.50)/χ=1.919



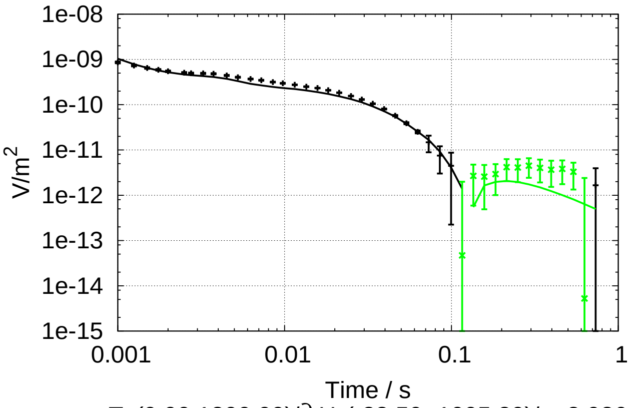
Tx(0.00,1800.00)/$\partial_t H_y$(-17.20,-1969.30)/χ=0.9929



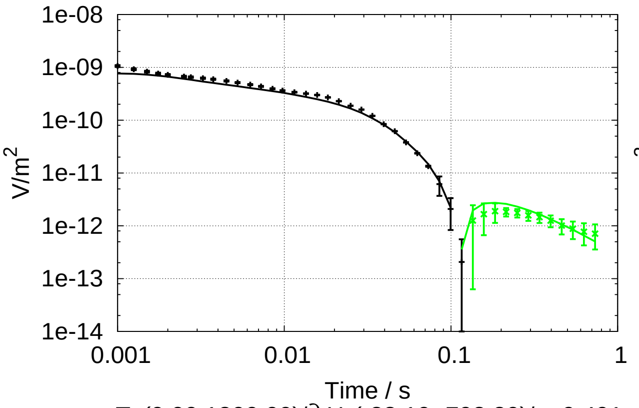
Tx(0.00,1800.00)/$\partial_t H_y$(-17.00,-1815.60)/χ=2.676



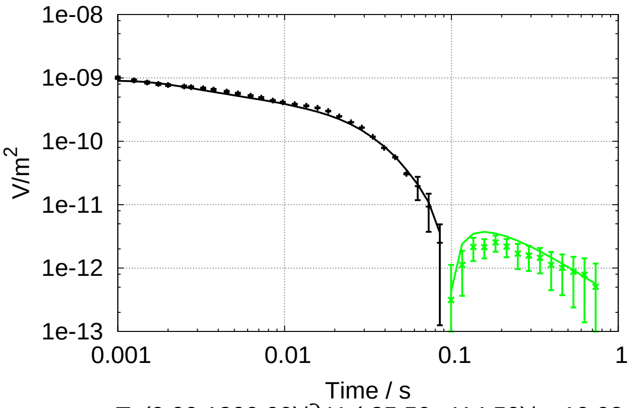
Tx(0.00,1800.00)/$\partial_t H_y$(-14.00,-1395.60)/χ=6.339



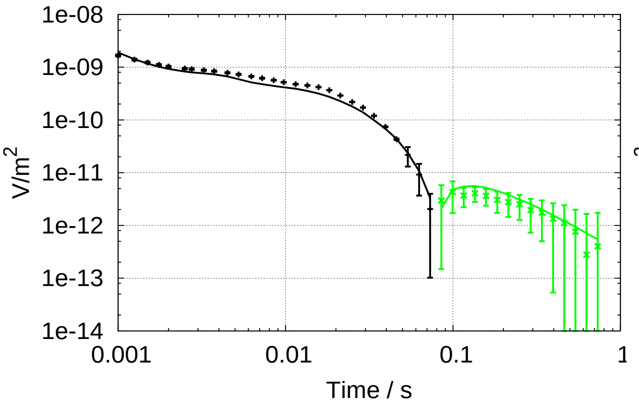
Tx(0.00,1800.00)/$\partial_t H_y$(-22.80,-1180.40)/χ=5.957



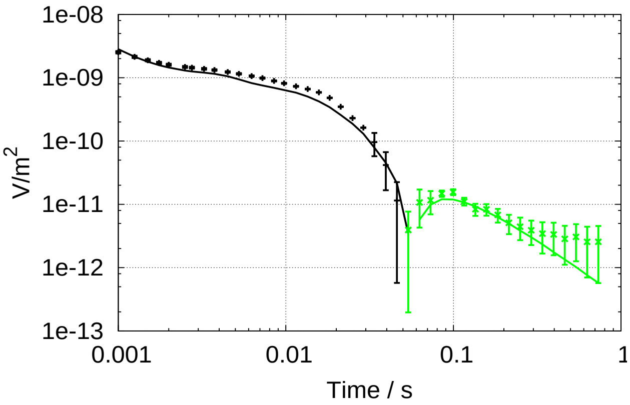
Tx(0.00,1800.00)/$\partial_t H_y$(-23.50,-1005.20)/χ=3.936

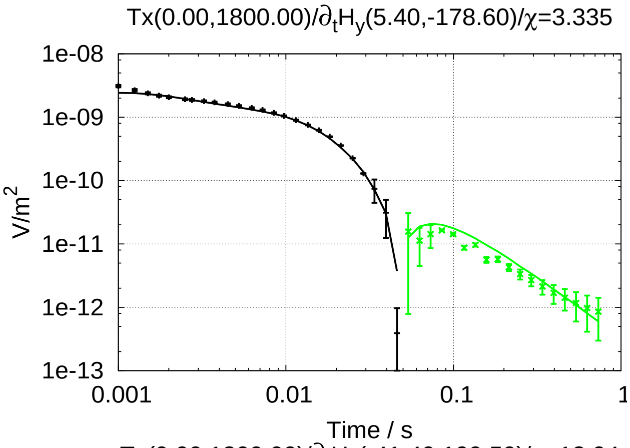


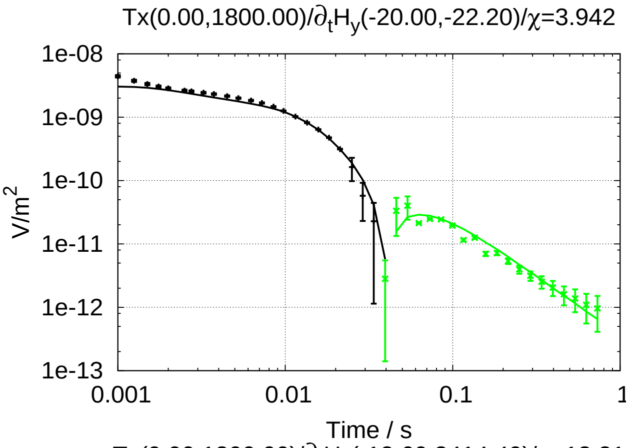
Tx(0.00,1800.00)/$\partial_t H_y$(-33.10,-798.20)/χ=9.491

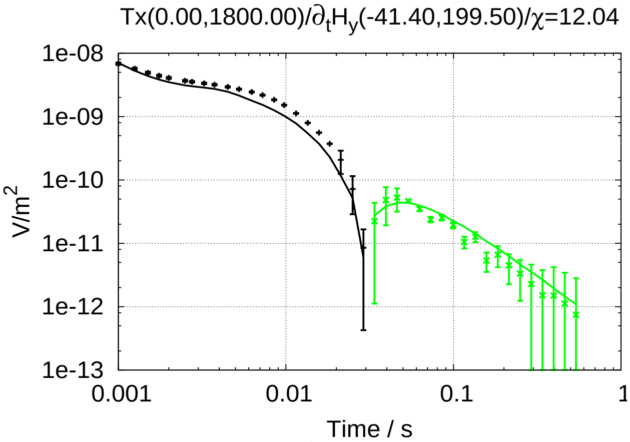


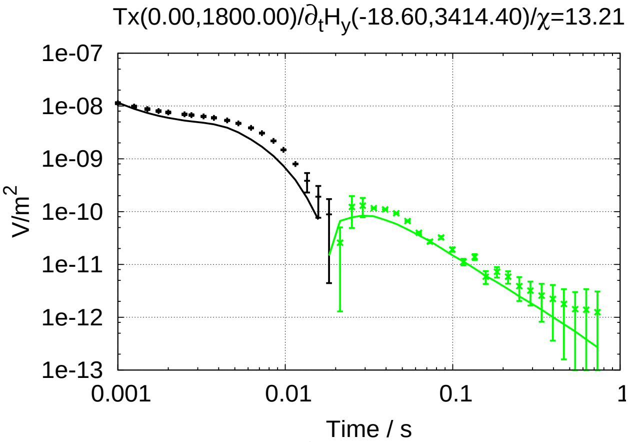
Tx(0.00,1800.00)/$\partial_t H_y$(-35.50,-414.50)/χ=10.08

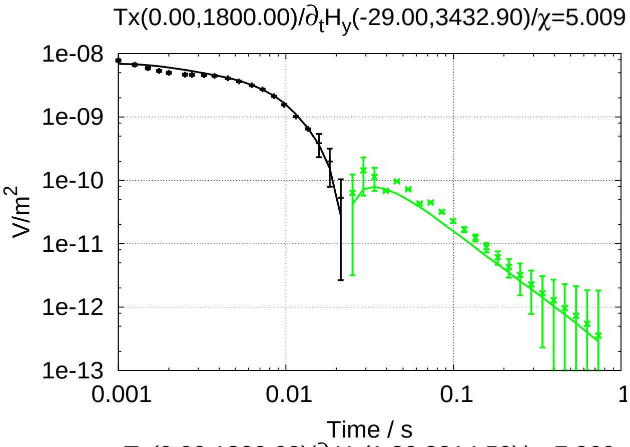


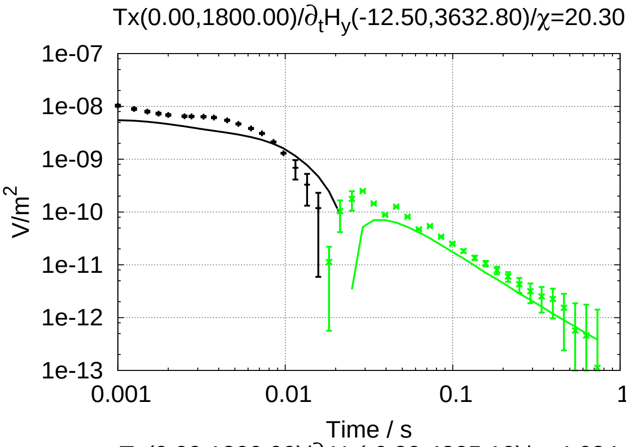

Tx(0.00,1800.00)/$\partial_t H_y$(5.40,-178.60)/χ=3.335

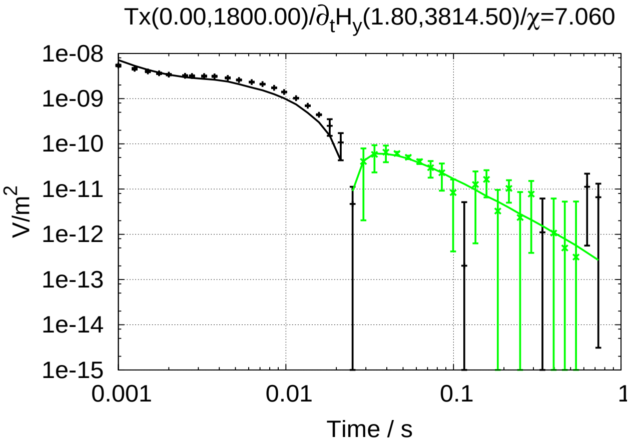

Tx(0.00,1800.00)/$\partial_t H_y$(-20.00,-22.20)/χ=3.942

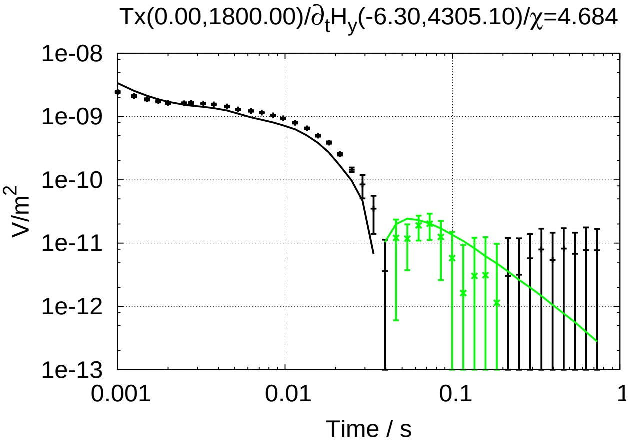

Tx(0.00,1800.00)/$\partial_t H_y$(-41.40,199.50)/χ=12.04

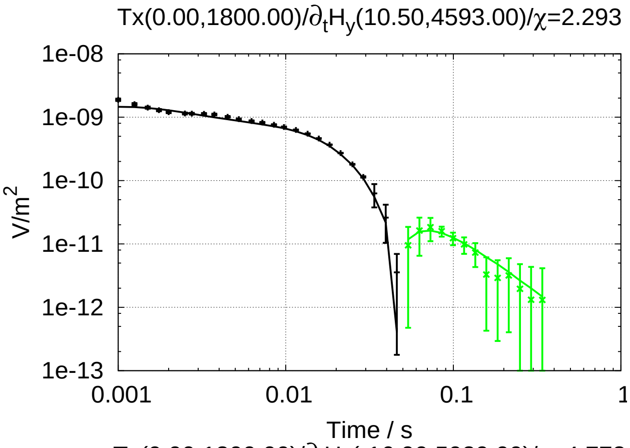

Tx(0.00,1800.00)/$\partial_t H_y$(-18.60,3414.40)/χ=13.21


Tx(0.00,1800.00)/$\partial_t H_y$(-29.00,3432.90)/χ=5.009

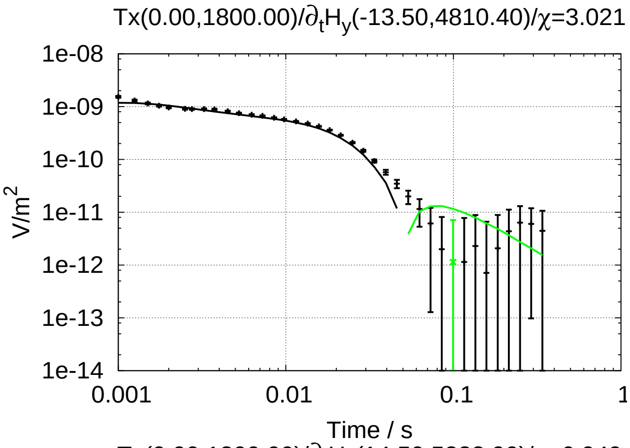

Tx(0.00,1800.00)/$\partial_t H_y$(-12.50,3632.80)/χ=20.30


Tx(0.00,1800.00)/$\partial_t H_y$(1.80,3814.50)/χ=7.060

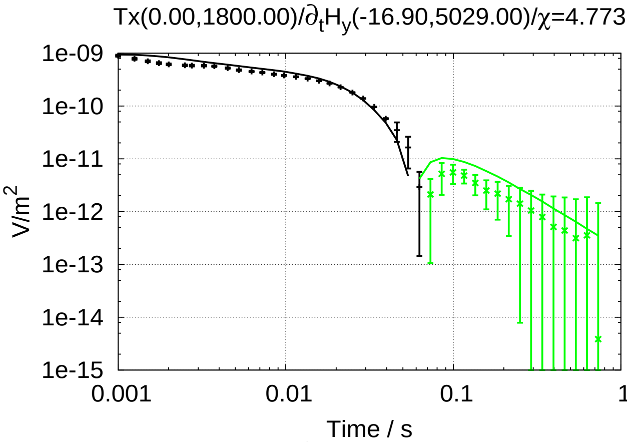

Tx(0.00,1800.00)/$\partial_t H_y$(-6.30,4305.10)/χ=4.684



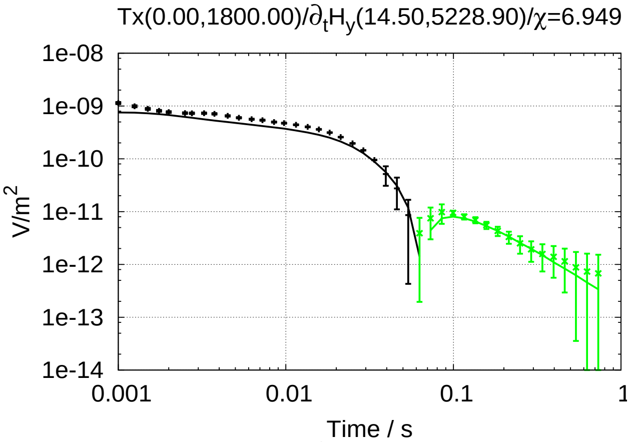
Tx(0.00,1800.00)/$\partial_t H_y$(10.50,4593.00)/χ=2.293



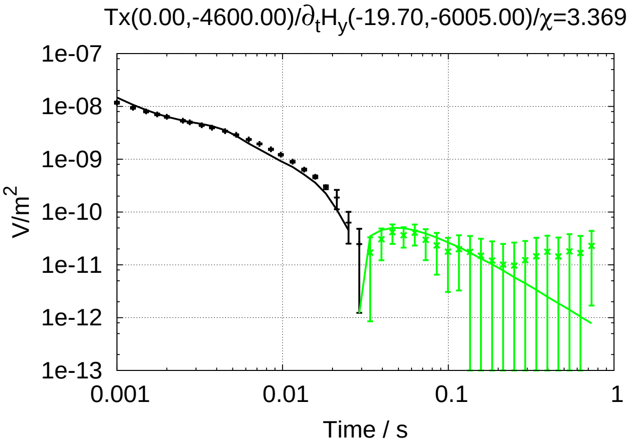
Tx(0.00,1800.00)/$\partial_t H_y$(-13.50,4810.40)/χ=3.021



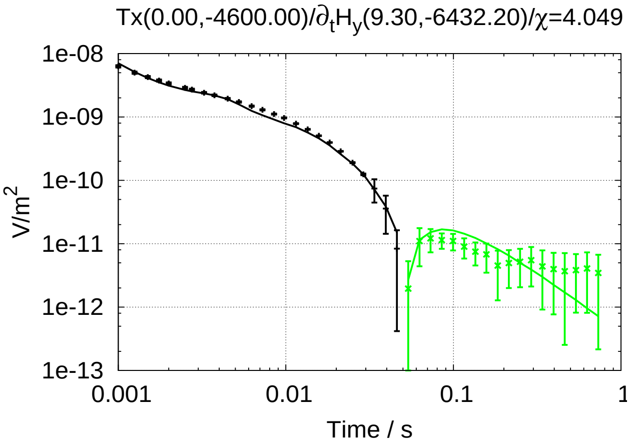
Tx(0.00,1800.00)/$\partial_t H_y$(-16.90,5029.00)/χ=4.773



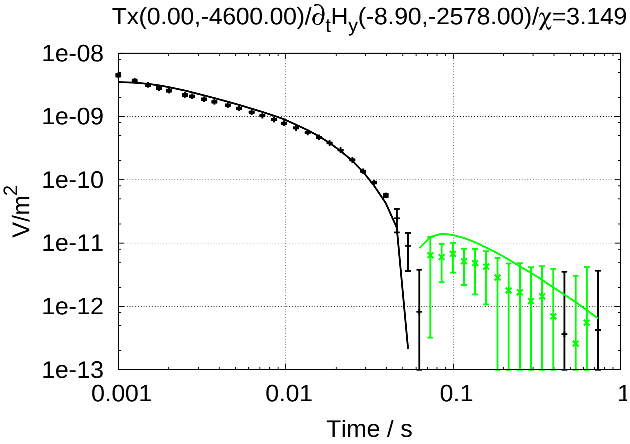
Tx(0.00,1800.00)/$\partial_t H_y$(14.50,5228.90)/χ=6.949



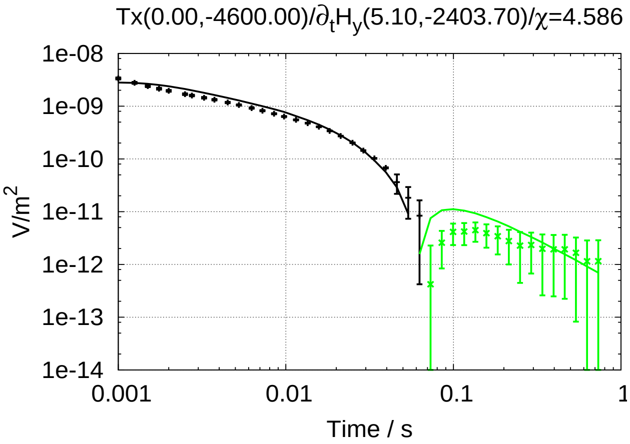
Tx(0.00,-4600.00)/$\partial_t H_y$(-19.70,-6005.00)/χ=3.369



Tx(0.00,-4600.00)/$\partial_t H_y$(9.30,-6432.20)/χ=4.049

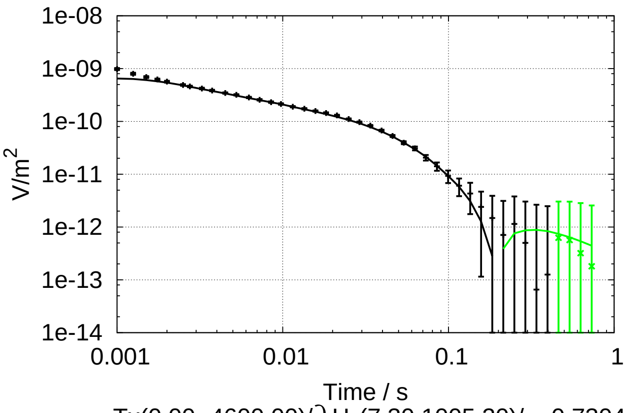


Tx(0.00,-4600.00)/$\partial_t H_y$(-8.90,-2578.00)/χ=3.149

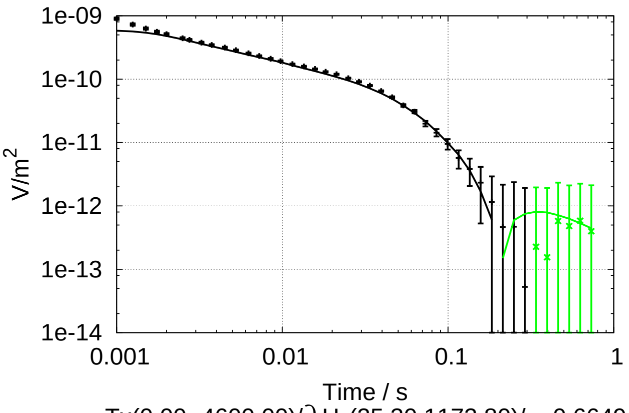


Tx(0.00,-4600.00)/$\partial_t H_y$(5.10,-2403.70)/χ=4.586

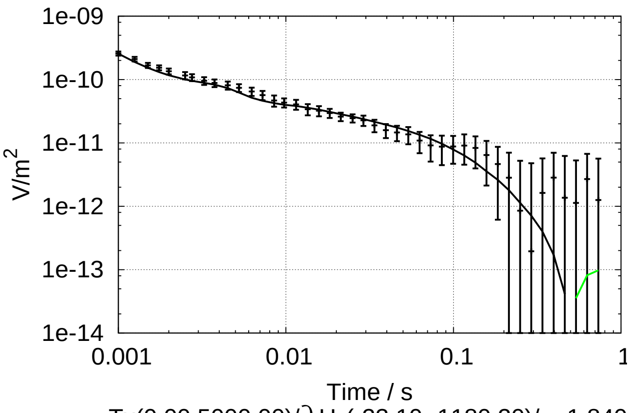
Tx(0.00,-4600.00)/$\partial_t H_y$(-23.00,-1180.10)/χ=1.946



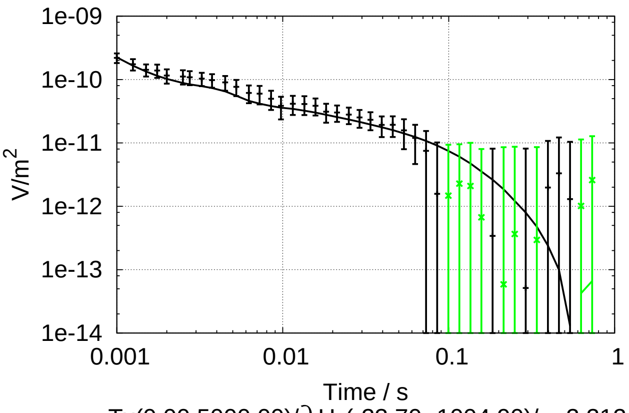
Tx(0.00,-4600.00)/$\partial_t H_y$(-23.60,-1004.80)/χ=2.665



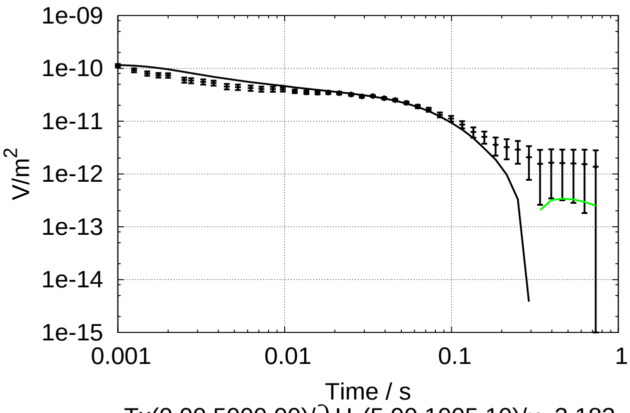
Tx(0.00,-4600.00)/$\partial_t H_y$(7.30,1005.20)/χ=0.7304



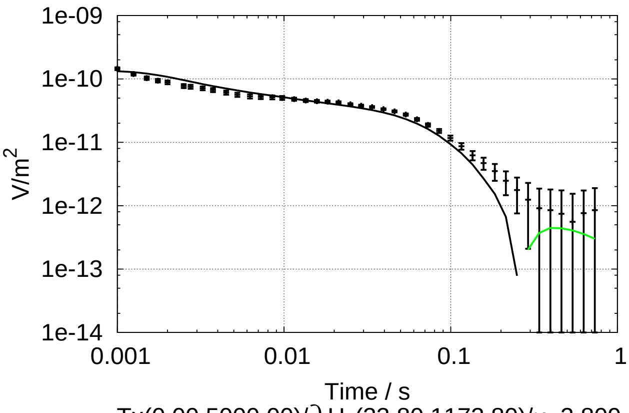
Tx(0.00,-4600.00)/$\partial_t H_y$(25.30,1172.80)/χ=0.6649



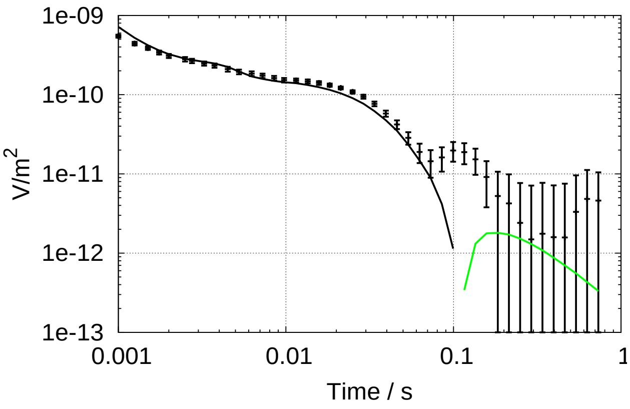
Tx(0.00,5000.00)/$\partial_t H_y$(-23.10,-1180.20)/χ=1.840



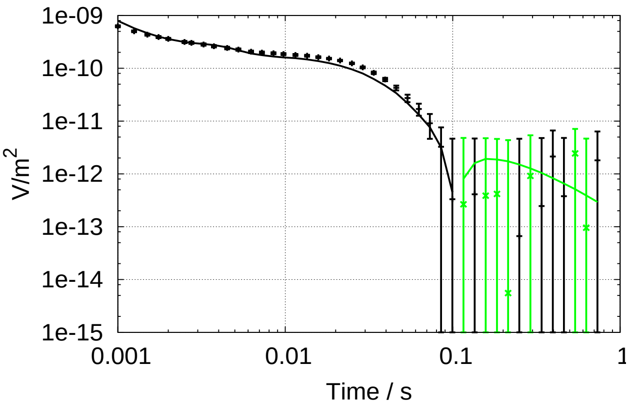
Tx(0.00,5000.00)/$\partial_t H_y$(-23.70,-1004.90)/χ=2.212

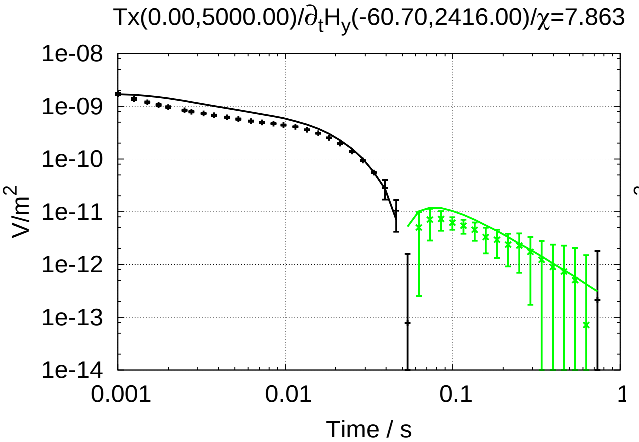


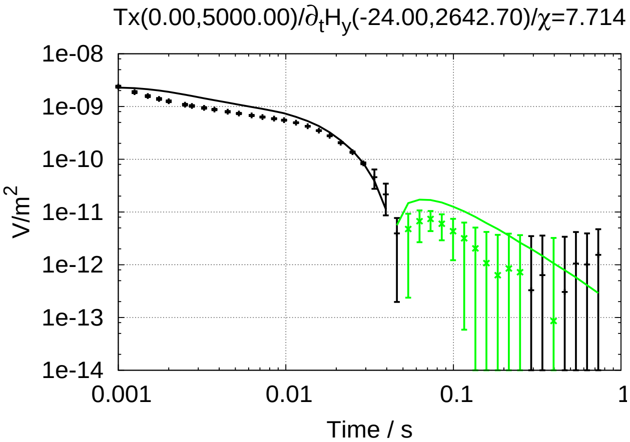
Tx(0.00,5000.00)/$\partial_t H_y$(5.90,1005.10)/χ=2.183

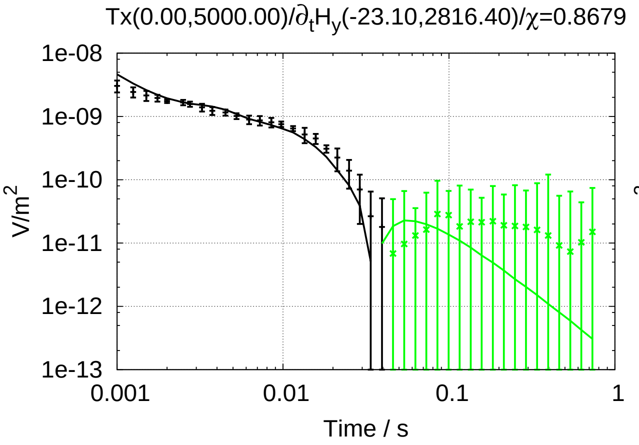


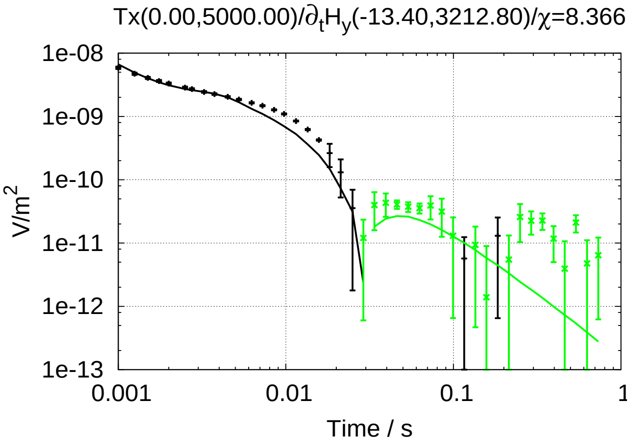
Tx(0.00,5000.00)/$\partial_t H_y$(23.80,1172.80)/χ=2.809




Tx(0.00,5000.00)/$\partial_t H_y$(-60.70,2416.00)/χ=7.863
1e-08
1e-09
1e-10
1e-11
1e-12
1e-13
1e-14
V/m^2
0.001
0.01
0.1
1
Time / s


Tx(0.00,5000.00)/$\partial_t H_y$(-24.00,2642.70)/χ=7.714
1e-08
1e-09
1e-10
1e-11
1e-12
1e-13
1e-14
V/m^2
0.001
0.01
0.1
1
Time / s


Tx(0.00,5000.00)/$\partial_t H_y$(-23.10,2816.40)/χ=0.8679
1e-08
1e-09
1e-10
1e-11
1e-12
1e-13
V/m^2
0.001
0.01
0.1
1
Time / s


Tx(0.00,5000.00)/$\partial_t H_y$(-13.40,3212.80)/χ=8.366
1e-08
1e-09
1e-10
1e-11
1e-12
1e-13
V/m^2
0.001
0.01
0.1
1
Time / s

Acknowledgments

At the end I like to express my thanks to the many people who helped and supported me throughout this work:

Prof. Dr. Bülent Tezkan for being the doctoral thesis supervisor, for providing advice during this work and for the support throughout the last years..

Prof. Dr. Andreas Kemna for appraising this thesis and for providing extraordinary support for the soccer team of the ”Ageos”..

Prof. Dr. Peter Weidelt for the many helpful advices and insights into EM methods..

Vladimir Druskin and *Leonid Knizhnerman* for providing the *SLDMem3t*..

The people who worked with me at the former *Aussenstelle Explorationsgeophysik*, above all *Dr. Carsten Scholl*, *Olaf Koch*, *Dr. Stefan L. Helwig* and *Prof. Dr. Andreas Hördt* for their friendship, the many fruitful discussions about TEM and EM in general, implementation strategies and much more..

Dr. Gregory A. Newman and *Dr. Michael Commer* for providing the synthetical 2D data and many insight into the “leap-frog” time-stepping..

all the ”*Eties*” and *Ageos*” at the IGM..

the *DFG* for funding and the local *RRZK* for *cliot*..

my family and especially my mother *Marita*, my father *Franz-Josef* and my sister *Sandra* for all the love and support..

family Audenrieth and especially..

Iris Audenrieth for her love and our son *Kilian*.

This work was partially funded by the DFG under contract TE 170/11-1

Erklärung

Ich versichere, dass ich die von mir vorgelegte Dissertation selbständig angefertigt, die benutzten Quellen und Hilfsmittel vollständig angegeben und die Stellen der Arbeit –einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, in jedem Einzelfall als Entlehnung kenntlich gemacht habe; dass diese Dissertation noch keiner anderen Fakultät oder Universität zur Prüfung vorgelegen hat; dass sie –abgesehen von unten angegebenen Teilpublikationen –noch nicht veröffentlicht worden ist sowie, dass ich eine solche Veröffentlichung vor Abschluss des Promotionsverfahrens nicht vornehmen werde. Die Bestimmungen dieser Promotionsordnung sind mir bekannt. Die von mir vorgelegte Dissertation ist von Prof. Dr. Bülent Tezkan betreut worden.

Köln, Mai 2009

Teilpublikationen

Martin, R., Hördt, A., 2001. *Approximierte Sensitivität für TEM. in* Hördt, A., und Stoll, J., Hrsg., Elektromagentische Tiefenforschung, 19. Kolloqium (ISSN 0946-7467).

Martin, R., 2003. *Realisierung von Laufzeitoptimierten Methoden zur Bestimmung der 3D-TEM-Jacobimatrix. Diplomarbeit*, Universität zu Köln, Institut für Geophysik und Meteorologie.

Martin, R., Scholl, C., Helwig, S. L., Hördt, A. und Tezkan, B., 2003. *Sensitivitätsberechung mit adjungierten Green'schen Funktionen für eine mehrdimensionale TEM-Inversion auf Linux-Clustern. in* Hördt, A., und Stoll, J., Hrsg., Elektromagentische Tiefenforschung, 20. Kolloqium (ISSN 0946-7467).

Scholl, C., **Martin, R.**, Commer, M., Helwig, S. L., Tezkan, B., 2003. *2D-Inversion von LOTEM-Daten. in* Hördt, A., und Stoll, J., Hrsg., Elektromagentische Tiefenforschung, 20. Kolloqium (ISSN 0946-7467).

Used Symbols

General notations	
a	A scalar value
$\mathbf{a}$	Vector
a_i	i'th component of vector $\mathbf{a}$
$\mathbf{A}$	Matrix
$\mathbf{A}^{\mathrm{T}}$	Transposed matrix
A_{ij}	i'th line and j'th column of matrix $\mathbf{A}$
$i,\ j,\ k$	Integer
ℓ^n	n-norm
∇	Nabla differential operator
δ	Small finite deviation
Δ	Finite difference
$\epsilon > 0$	Very small real valued constant
$\Re,\ \Im$	Real and imaginary part of a complex number
$\mathbf{d} \in \mathbb{R}^N$	Vector of measured data
$\mathbf{m} \in \mathbb{R}^M$	Vector of model data
$\mathbf{f},\ \mathbf{f}(\mathbf{m})$	Forward solution or model response
N	Number of data points
M	Number of model unknowns
Inversion symbols of chapter 2	
$\alpha,\ \alpha_*$	Step length, optimal step length
a_i	Interval start point used within line search
$a_{x,y,z}$	Directional weighting coefficients
$\mathbf{A}$	Square matrix
β_n^{xx}	Conjugated gradient parameter
$\mathbf{b}$	Ride hand side vector
b_i	Interval mid point
$c_{1,2}$	Real valued constants to satisfy the Wolfe conditions
c_i	Interval end point
$\mathbf{C}$	Constraint or smoothness matrix
$\mathbf{C}^{1st,2nd}$	Smoothness matrix of first and second order finite difference
$\mathbf{C}_{\mathbf{d}}$	Data covariance matrix
$\mathbf{C}_{\mathbf{m}}$	The prior model covariance matrix
$\mathbf{C}_{x,y,z}$	Space-directional finite difference matrices
$\mathrm{Cov}(X)$	Covariance of X
$\delta\mathbf{d}_i$	Error of i'th data point
ζ	Value of golden-mean
$\mathbf{g}$	Gradient of the cost functional
θ_i	Basis functions of the model
$\boldsymbol{\mathcal{H}}$	Hessian matrix of the cost functional
$\mathbf{I}$	Identity matrix
λ	Lagrange, regularization or damping factor
$\mathbf{m}_*$	Stationary point
$\hat{\mathbf{m}}$	A priori model
$\delta\mathbf{m}$	Model update vector
n	Iteration number
Φ	Cost functional

$\Phi^{d,m}$	Data and model cost functional
Ω	Model domain
$\mathbf{R}$	Resolution matrix
$\mathbf{s}$	Secand vector
$\mathbf{S}$	Sensitivity matrix
$\mathbf{v},\ \mathbf{u}$	Vectors
$\mathbf{y}$	Data residual vector
(T)EM symbols of chapter 3	
a_f	Linear factor within Archie's law
a_g	Scalar value for grid design (e.g. $a_g \approx 1$)
A	Area of a square loop transmitter
$\mathbf{A}$	Differential operator matrix
b_g	Scalar value for grid design (e.g. $3 \leq b_g \leq 9$)
$\mathbf{B}$	Magnetic induction
$\dot{\mathbf{B}}$	Time change of magnetic induction
γ_m	Ritz vector
$c,\ c_0$	Speed of light (vacuum)
cov	Coverage
$\delta(t)$	Dirac delta function
d^*	Diffusion depth
$d_{min,max}$	Minimum/maximum grid cell length
$d^{\mathsf{step}}(t)$	Step response data
$\tilde{d}(t)$	Measured data with system response
$\mathbf{D}$	Displacement field
$\widetilde{\mathbf{D}}$	Dipole moment of a electrical transmitter
$\epsilon_r,\ \epsilon_0$	Dielectric permittivity, relative and free space
$\hat{\epsilon}$	Convergence criteria parameter for *SLDMem3t*
$\hat{\mathbf{e}}$	Unit vector
$\mathbf{E}$	Electric field
$\mathbf{E}_{e,m}$	TE- and TM-mode of electric field
$\mathbf{E}^p$	Background or primary electric field
$\mathsf{E}_\mathrm{x},\ \mathsf{E}_\mathrm{y},\ \mathsf{E}_\mathrm{z}$	Electric field sensor components
f	Frequency
$f(t)$	Function in time
f^a	Analytical solution
$f^{\text{step}}(t)$	Step response
$f^{\text{step (on,off)}}$	Step (on/off) response
$f_{e,m}$	Transformed TE- and TM-mode in wavenumber domain
$\tilde{f}$	Convolution of two functions
$\mathbf{f}_{e,m},\ \mathbf{f}_0$	Model response for electric and magnetic field, initial value
$g(t)$	Function in time
$g_{i=1,\dots,N_t}$	Discrete function in time
$\boldsymbol{\mathcal{G}},\ \boldsymbol{\mathcal{G}}^{\mathsf{R},e,m}$	Dyadic green function, adjoint dyadic Green function
$\theta_{1,\dots,m,\dots\mathsf{M}}$	Smallest/largest eigenvalue of Krylov subspace
$\mathbf{H}$	Magnetic field
$\tilde{H}_{0,1,-\frac{1}{2},\frac{1}{2}}$	Filter coefficients for fast Hankel transform
$\mathbf{H}_e,\ \mathbf{H}_m$	TE- and TM- mode of magnetic field
$\dot{\mathbf{H}}$	Time change of magnetic field

$\dot{\mathsf{H}}_{\mathrm{x}}$, $\dot{\mathsf{H}}_{\mathrm{y}}$, $\dot{\mathsf{H}}_{\mathrm{z}}$	Voltage receiver components
I, I_0	Current/initial current
$\mathbf{J}$, $\mathbf{J}_e$, $\mathbf{J}_a$	Current density, external and anomalous current density
$J_{0,1}$	Bessel function of first kind of order zero and one
$\mathbf{K}^{\mathsf{M}}$	Krylov space
$\mathbf{k}$	Wavenumber vector
k	Wavenumber
λ_m	Eigenvalues of $\mathbf{A}$
μ_r, μ_0	Magnetic permeability, relative and free space
m_c	Cementation index
M, $\mathsf{M}_{est,max}$	Dimension of Krylov space, estimated and maximum Krylov iterations
$\widetilde{\mathbf{M}}$	Transmitter moment of magnetic dipole
n_s	Saturation exponent
$N_{\mathbf{A}}$	Dimension of the system matrix $\mathbf{A}$
N_d	Points per decade
N_f	Number of frequencies
N_g	Number of grid points in horizontal direction
N_n	Number of total grid node points within a grid
N_l	Number of logspaced data points
N_{R}	Number of receiver sites
N_t	Number of time points
$N_{x,y,z}$	Number of grid points in x-, y- and z-direction
ϕ_r	Porosity of rock
ϕ_e, ϕ_m	Scalar (Debeye) potentials
$\mathbf{q}$	Krylov space basis vector
Q	Interpolation operator
$\mathbf{Q}$	Orthonormal basis in Krylov space
ρ	Resistivity
ρ_a	Apparent resistivity
$\mathbf{r}$, $\mathbf{r}'$	Position vectors
$\mathbf{r}_e$	Residual vector
$r_{min.max}$	Smallest/largest distance
σ, $\sigma_{min,max}$	Conductivity, minimum/maximum conductivity
$\mathbf{s}_a$	Inhomogeneous source term
$\hat{\mathbf{s}}_m$	m'th eigenvector of the Krylov subspace
S_w	Volume fraction of rock filled with water
$S(t)$	Sensitivity density as continuous time function
$\mathbf{S}_j(t_i)$	Sensitivity density of a volume fraction at timepoint t_i
S_{ij}	Discrete sensitivity value
t	Time
t_r	Last timepoint of the system response
$t_{min,max}$	Smallest/largest timepoint
$\mathbf{v}_m$	Eigenvectors of $\mathbf{A}$
V_k	Volume of model cells within finite volume
ω	Angular frequency
x, y, z	Coordinates in space
$*$	Convolution operator

	Multidimensional TEM inversion (chapter 4)
α_n	Step length of n'th iteration
β_n	Conjugation scalar
$c(\lambda),\ c(\psi_d, \psi_m)$	Generalized cross validation and L-curve
$c_2 > 0$	Constant real value of Armijo type line search
$\mathbf{C}^{1st,2nd}$	First and second order smoothness operators
$\widehat{\mathsf{cov}},\ \widehat{\mathsf{cov}}^{norm}$	Error weighted and normalized coverage
d_{nlev}	Noise level
d^{noia}	Artificial noise
DOI	Depth-of-investigation index
γ	Damping factor
λ	Regularization parameter
$\mathbf{m}_*^{a,b}$	Posterior models
$\hat{\mathbf{m}}^{a,b}$	Starting (prior) models
m_j^u	Upper boundary of j'th model parameter
n	Iteration number
N_c	Number of Bits of a A/D-converter
$\psi_d,\ \psi_m$	Data misfit norm and roughness
p	Real valued factor for the standart deviation
$\hat{r}_n$	Normal distributed random number
$\widehat{R}_{jj}$	Normalized main diagonal of the resolution matrix
S_{ij}^{adj}	Adjoint sensitivities
S_{ij}^{lin}	Linear sensitivities
S_{ij}^{tr}	Transformed sensitivities
tr	Transformation function
χ^2	Normal distribution
x, y, z	Coordinates in space

	Appendix A
$a^{+/-}$	Damping coefficients within a layer
a	Real valued variable
A	Loop radius
$A(z, k, \omega)$	TE-impedance
$A_n^{+/-}$	TE-impedance for in-/out-going wave within layer n
b	Real valued variable
$F(z, k, \omega)$	TM-impedance
$F_n^{+/-}$	TE-impedance for in-/out-going wave within layer n
$\gamma(k, h)$	Reflection coefficient for wavenumber k and depth h
h_n	Hight of the n'th layer
HED_x	Horizontal electric dipole in x-direction
HMD_x	Horizontal magnetic dipole in x-direction
R	Cylinder radius
u_n	Discrete wavenumber of layer n
VMD	Vertical magnetic dipole
x, y, z	Coordinates in space

	Appendix B
α	Projection value
$\mathbf{A}$	Ride hand side matrix

β	Projection value
$\mathbf{b}$	Left hand side vector
$\mathbf{c}$	Intermediate solution vector
$\mathbf{D}$	Diagonal matrix
f_i	Filter factors
λ	Damping factor
$\mathbf{L}$	Lower triangular matrix
N_t	Threshold of singular values
$\mathbf{p}$	Intermediate vector storing $\mathbf{A}^{\mathrm{T}}\mathbf{r}$
$\mathbf{P}$	Preconditioner
$\mathbf{q}$	Intermediate vector storing $\mathbf{Ap}$
$\mathbf{r}$	Residual vector
$\boldsymbol{\Sigma}$	Singular value matrix
s_i	Singular value
$\mathbf{u}_i$	Row vector of the upper triangular matrix
$\mathbf{U}$	Upper triangular matrix
$\mathbf{x}$	Solution vector
$\mathbf{y}$	Intermediate solution
Appendix C	
$\mathbf{A}$	Ride hand side matrix
$\mathbf{b}$	Left hand side vector
$\mathbf{C}_d$	The data covariance matrix
$\mathbf{C}_m$	The prior model covariance matrix
$\mathbf{C}_M$	The posterior model covariance matrix (MCM)
$\mathrm{Cov}(X)$	Covariance of X
$\boldsymbol{\mathcal{H}}$	Hessian matrix of second derivatives
$\hat{\mathbf{m}}$	Prior model
$\mathbf{R}$	Resolution matrix
$\mathbf{S}$	Sensitivity matrix
$\mathbf{x}$	Solution vector