Hochschule Neubrandenburg
University of Applied Sciences

GPS-based source inversion within the German Indonesian Early Warning System

by

SEBASTIAN TAUER

To graduate

as

MASTER OF ENGINEERING IN GEOINFORMATICS AND GEODESY

Mai 2013

urn: nbn:de:gbv:519-thesis2012-0694-3

Supervisor:
Prof. Dr. rer. nat. habil. Gerd Teschke

Assessor:
Dr. Natalja Rakowsky

Neubrandenburg, Mai 2013

Acknowledgement

It is a pleasure to thank the many people who made this thesis possible. This work would not have been possible without the support of Prof. Dr. Gerd Teschke and Dr. Natalja Rakowsky who has been very helpful and provided me in numerous ways.

My final words go to my family. I want to thank my girlfriend, parents and grandparents whose love and guidance are with me in whatever I pursue. Finally, I want to thank all people for their support and friendship.

Declaration of academic honesty

Herewith I declare that this thesis is the result of my independent work. All sources and auxiliary materials used by me in this thesis are cited completely.

Neubrandenburg,

Abstract

The aim of this master thesis is to work out algorithms to solve an underdetermined inverse problem in the context of the German Indonesian Early Warning System. The slip distribution, the localization of the solution and the magnitude of the earthquake will be calculated, depending on the shifting of the GPS sensors. Kaczmarz method, Landweber algorithm, Richardson iteration, Least Squares method, Tykonov Regularization via QR factorization and Least Norm via Singular Value Decomposition are examined and their stability, sensibility are tested. Furthermore, the magnitude is computed by using the Rupture Generator. The sensitivity of the GPS sensors is used to restrict the system of equations, to get a proper solution.

Zusammenfassung

Der Fokus der Masterarbeit liegt auf der Lösung eines unterbestimmten inversen Problems und wurde im Rahmen des GITEWS Projektes angefertigt. Berechnet wurden die Slipverteilung, die Lokalisation der Lösung und die Magnitude des Erdbebens in Abhängigkeit von der Verschiebung der GPS-Sensoren. Kaczmarz Methode, Landweber-Verfahren, Richardson-Verfahren, Methode der kleinsten Quadrate, Tykhonov-Regularisierung mit QR-Zerlegung und die direkte Singulärwertzerlegung wurden auf Stabilität und Fehleranfälligkeit getestet. Des Weiteren wurde aus der Slipverteilung die Magnitude mittels RuptGen berechnet. Darüber hinaus wurde die Sensibilität der GPS-Sensoren als Nebenbedingung implementiert.

Contents

1 Introduction

Indonesia is a region of high potential earthquakes because of the plate tectonics. It is located at the triple junction of the Indo-Australian Plate, Eurasian Plate and Pacific Plate. A tsunami hit the coast around the Indian Ocean on December 26, 2004 caused by a strong earthquake. The wave height was about 30 meters, 230000 people lost their lives, approximately 600000 houses were destroyed and 1.8 million people were left homeless.

Fig 1.1: devasting effects of a tsunami [16]

As a result, Germany and Indonesia have started to implement a tsunami early warning system. The German-Indonesian Tsunami Early Warning System (GITEWS) contains new scientific methods and novel technologies in comparison to previous Pacific tsunami warning systems. GITEWS is the first system which includes the distinctive geological situation [9].
If an earthquake originates along the Sunda Trench, which arches from the Northwestern tip of Sumatra to Flores in Eastern Indonesia (subducting zone) and is located in the Indian Ocean, it can have devastating effects.
An earthquake in this region can generate Tsunami waves which can reach the coast within 20 minutes. There is only a finite amount of time for early warning. This bounding factor was the core of the entire system. The geological situation and the short arrival times between an

earthquake and the following tsunami require a new approach in early warning. The new system relies on 300 land-based sensors throughout this region. The warning is based on quick response, precise earthquake capture and analysis, which is the core of the system (see figure 1.2).

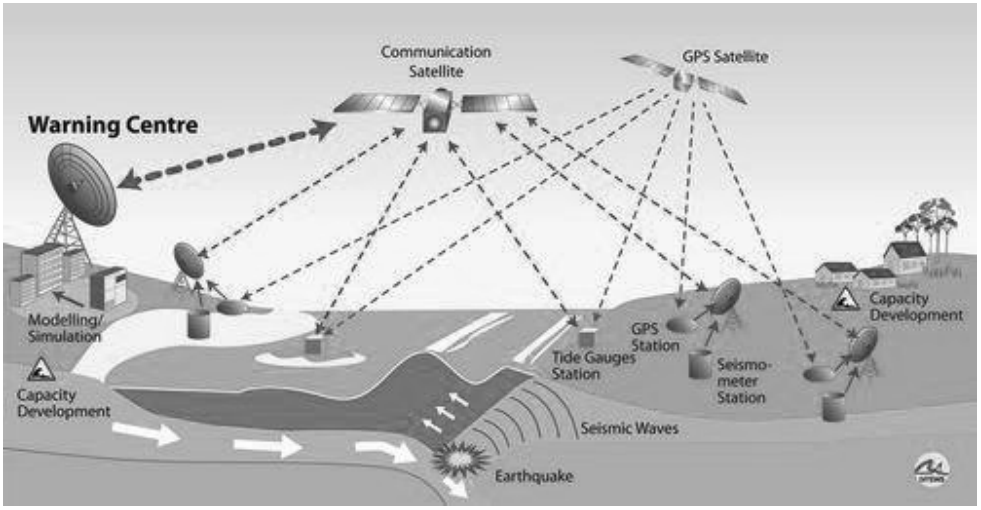

Fig 1.2: technical concept of GITEWS [2]

The fast determining of position, depth and magnitude is essential, for creating a prime model and the resulting warning. The data of GPS-stations and coast gauge are verifying the result more exactly. Indonesia is affected with very short early warning times of about 20 minutes. It is impossible to evacuate big cities (Padang) in such a short time frame, because of the dense population. The population has to be trained and emergency guides have to been realized [2].

2 Project tasks and scientific goals

The main task of the master thesis is to solve an underdetermined inverse problem. The master thesis can be divided into three parts. Firstly, the inversion algorithm of Manuela Schönrock has to be captured (QR decomposition) and improved (Tykhonov Regularization via QR decomposition). Furthermore, new methods can be implemented. Secondly “RuptGen” is used to simulate virtual earthquakes. The software tool needs the input of epicenter, magnitude and position and a few additional options can be set. It computes the slip distribution, the shift of the GPS sensors and the deformation of seabed. The Kaczmarz Method, Modified Richardson Iteration, Landweber Method, Tykhonov Regularization, Least Squares algorithm and Least Norm algorithm are chosen. The algorithms will be tested to examine their stability and sensibility in subjection to different noise levels. Furthermore, the magnitude will be computed and compared with the entered magnitude. The aim is to determine an algorithm which gives the best result.

3 Geophysical principles

3.1 Plate Tectonics

Plate tectonic assumes that the earth crust consists of seven major lithospheric plates and a few minor lithospheric plates. The lithosphere is the hard and rigid outer layer of the earth which includes the crust and the uppermost mantle. The movement of the plates is the consequence of the convection process. Indonesia is one of the most seismically active regions of the earth, because it is located at the triple junction of the Indo-Australian Plate, Eurasian Plate and Pacific Plate. The Indo-Australia plate moves around 6 centimeters per year.

There are three different types of plate boundaries. "Transform boundaries" happen where plates slide or pass each other, "Divergent boundaries" arise where plates slide apart from each other. In Indonesia mostly "convergent boundaries" exist. Two plates slide in the direction of each other and generally there accrues a subduction zone related with deep trenches [15].

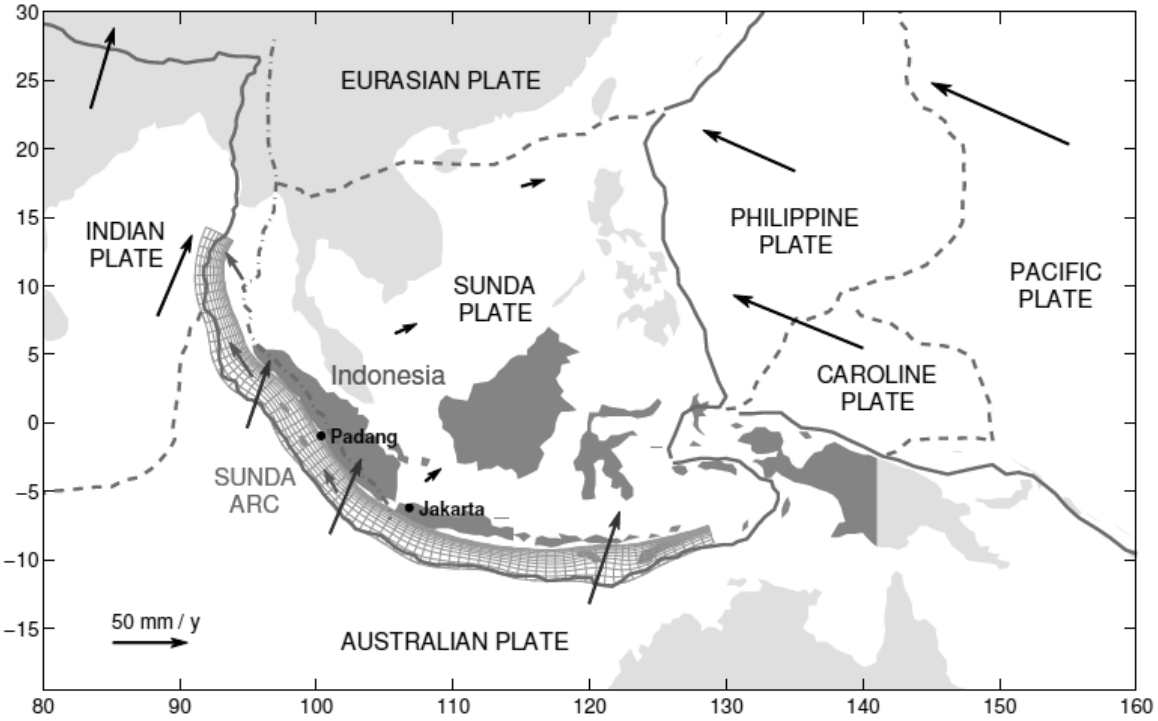

Fig 3.1: Plate tectonic of Indonesia [11]

3.2 Earthquake

Earthquakes are caused by the abrupt release of the collected energy in the earth crust where seismic waves are originating. Earthquakes with a magnitude of three or lower are imperceptible or faint and a magnitude higher than seven can cause high damage over large regions. The hypocenter of an earthquake is the location where the tense energy is first released. The epicenter which is above the hypocenter is the position on the earth where an earthquake originates. When the epicenter of a large earthquake is located offshore and the seabed is displaced sufficiently, a tsunami can be caused. There are three main types of faults which may cause an earthquake. Normal faults mainly happen in regions where the crust is being extended similar to a divergent boundary. Reverse faults arise in regions with a shortened crust such as a convergent boundary. Strike- slip faults are steep structures where the two sides of the fault slide horizontally. However, the possibility exist that earthquakes are released by movement on faults which have elements of both dip slip and strike slip, this is known as oblique slip [18].
The moment magnitude scale (MMS) is applied by seismologists to calculate the dimension of earthquakes in relation to the released energy. The Richter scale (developed 1930) was the predecessor of the moment magnitude scale, developed in 1970. An earthquake is denoted by the seismic moment and is given through

$$M_0 = \mu \cdot D \cdot S[N \cdot m] ,$$

where μ is a coefficient and describes the rigidity of the medium, D is the displacement deflection among the opposite edge of the fault, and S is the region of the rupture along the fault. The magnitude of an earthquake is devoted to the seismic moment and is given through [13].

$$M_w = \frac{\log_{10}(M_0)}{1.5} - 6.07$$

3.3 Tsunami

A tsunami (jap. Harbor wave) is a sequence of water waves that can reach speeds of up to 800 km/h and is usually caused by an earthquake, volcanic eruption or a costal landslide. Tsunamis have a very long wavelength (deep sea) often up to hundreds of kilometers. By comparison wind waves only have thirty to forty meters wavelength. In addition, the wave height offshore is very small. When a tsunami reaches the coast, the water becomes shallow. Wave shoaling compresses the wave and the speed becomes less than 80 Km/h. Thus the wavelength decreases below 20 kilometers and therefore the amplitude increases.

When the crest of the tsunami wave reaches the coast, the resulting height of the water onshore in comparing to a reference sea level is donated “run up”. Large tsunamis can feature multiple waves over a time frame of hours and it is unsure which wave has the highest run up [13].

3.4 Sunda Arc and Sunda Trench

The Sunda Arc equals to the Volcanic Arc that has created the islands of Sumatra and Java, the Sunda Strait and the Lesser Sunda Islands. A volcanic chain shapes the topography of these islands. The Arc tags an active destructive plate boundary among the East Eurasian Plate and the Indo-Australian Plate. The Indo- Australian Plate is subducting beneath the Eurasian Plate (consists of Sunda- and Burma Plate) and lies under Indonesia [21].

The Sunda Trench has a length of 2600 Kilometers, a maximum depth of 7725 meters and is the second deepest point in the Indian Ocean [22].

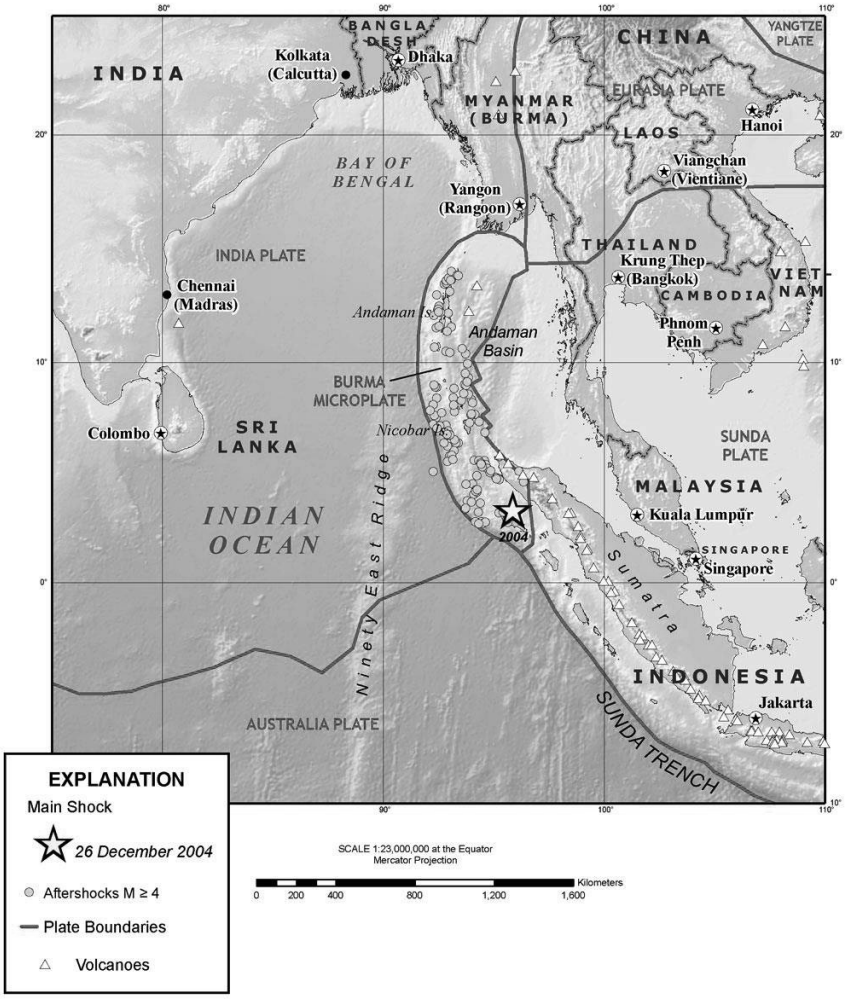

Fig 3.2: Sunda Trench [15]

4 Scenario modeling

4.1 Subfault array

The plate interface between the subducting Indian –Australian and the upper Sunda plate is discretized into a regular mesh of rectangular patches extending from 0 to 100km depth. The mesh is derived from the RUM model by Gudmundsson and Sambridge (1998) and follows the geometry of the plate interface. The mesh consists of $25 \cdot 150$ subfaults (altogether 3750 subfaults) with average dimensions of $40\text{km} \cdot 15\text{km}$ that simulate an area of $6000\text{km} \cdot 375\text{km}$. Every subfault with known geometry (width, height) and position (longitude, latitude, altitude) represents a rectangular fault plane [15].

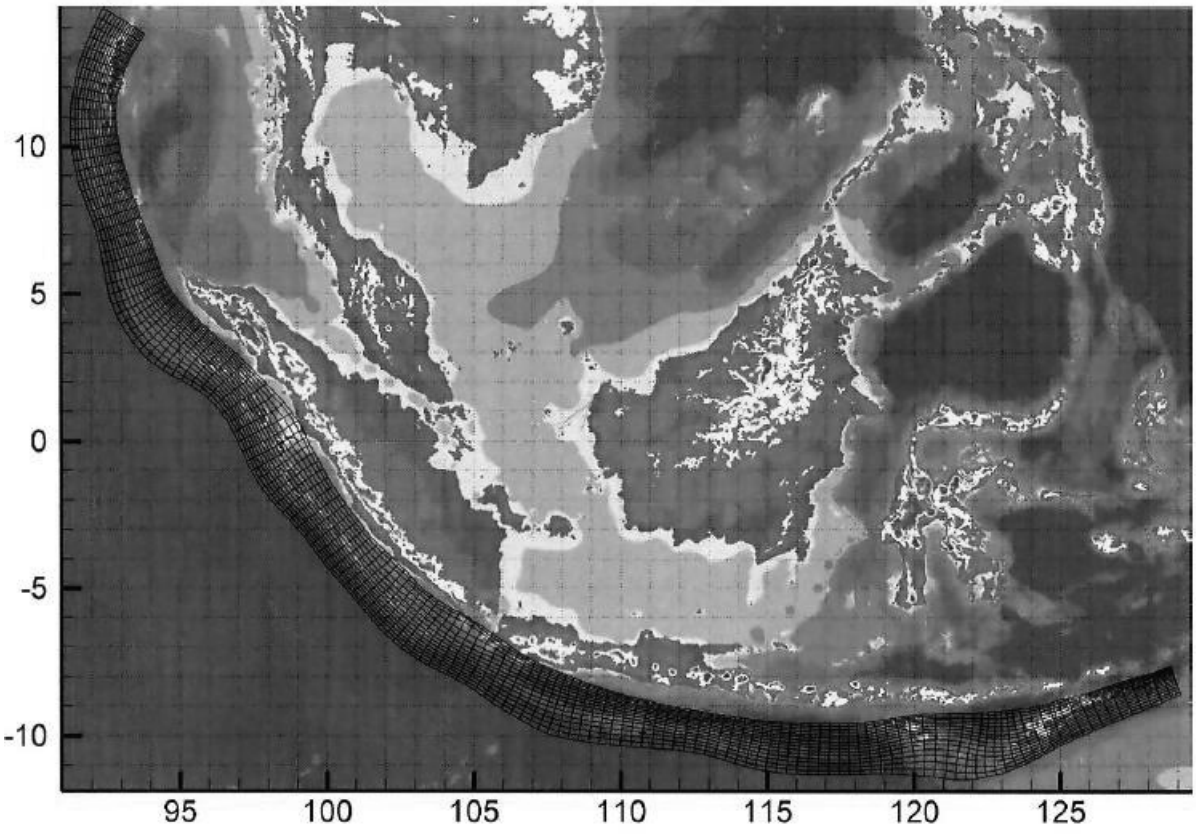

Fig 4.1: The old mesh (subfault array) consists of $15 \cdot 150$ and the new mesh consists of $25 \cdot 150$ patches [15]

Each subfault can be described by its orientation, particular through strike and dip angle. Strike is measured from north to south and represents the intersection of subfaults with the azimuth. Dip can be defined as the angle underneath the azimuth surface and describes their slope. Slip shows the motion of material and consists of dip slip and strike slip. Faults can be distinguished into three types related to the geometry of faulting. A dip slip fault is a fault where the main movement is vertical and can be classified into normal fault, reverse fault or thrust fault. Normal faults arise where the soil is stretching, one block of soil moves downwards relative to the footwall. In the case of the Sunda Trench, a thrust fault is the opposite of a normal fault with a dip angle of the fault less than 45°. A Reverse fault has the same sense of motion as a thrust fault and is relatively steep with an angle greater than 45°. The surface of faulting is typically near vertical and the footwall moves to the left or right. An oblique slip fault is a fault which has a substantial component of dip slip and strike slip [15].

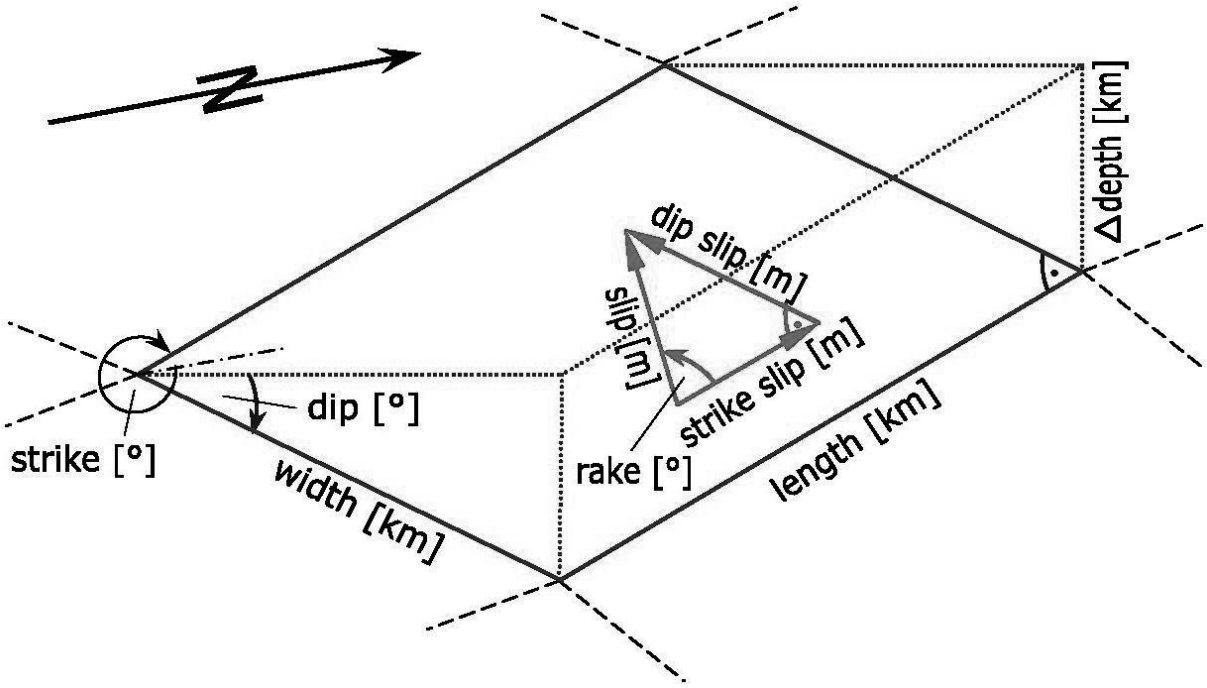

Fig 4.2: Definition of a single subfault [15]

4.2 Near field GPS arrays

The "GPS Shield" concept for Sumatra is based on near field GPS arrays which operate in real time. Two or more stations are located on the Mentawai islands among the Sumatra coast and the trench. They build the frontal part of the array. Stations in the center of the array are very close to each other (approximately 10 – 20 km) and justified perpendicular to the trench. One master station and various slave stations are mounted with accurate geodetic dual frequency receivers, a digital meteorological sensor, data processor unit, and radio modem. The master station steadily obtains high rate GPS – observations at 1Hz from its slave stations and computes the coordinate distinctions for every single measurement in real-time. The island hosted part of the arrays is directly located along the shore of Sumatra and supplemented by GPS – stations. The goal is to measure the surface deformation of every subfault, through GPS techniques. The real time GPS stations with a distance of 100 – 200 km are placed along the trench [3].

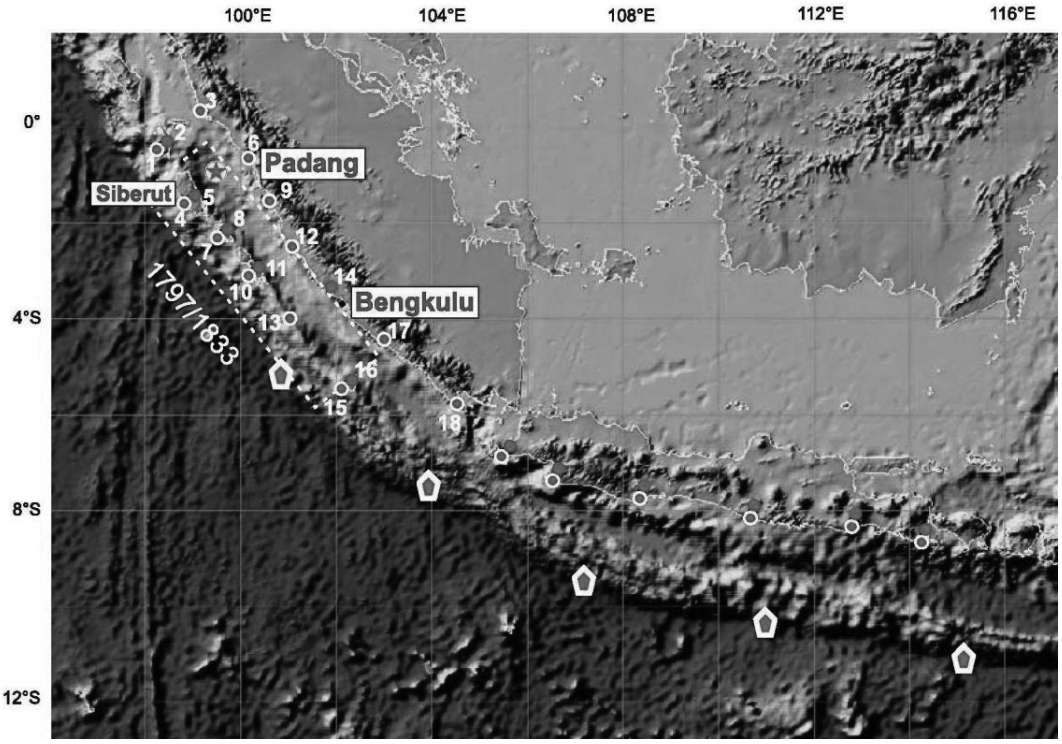

Fig 4.3: GPS-shield system in the region of Indonesia. The red circles and the red circles with white edging are the real-time GPS stations, but the red circles with the white edging are also equipped with a broadband seismometer and strong motion recorder. The red diamonds are buoys [11].

4.3 Rupture Generator

The Rupture Generator is a tool that was developed by A.Babeyko during the GITEWS (German Indonesian Tsunami Early Warning System) project. The tool is used to simulate virtual earthquakes. The program computes the surface deformation at the Sunda trench and the static displacement for a variable number of GPS receivers. The mathematical forward model of Green's functions is used and based on the concept of a subfault array along the subduction zone [1]. It defines the relationship between the earthquake source parameter and the geodetic measurement. In the mathematical case the linearity of these functions are essential [15]. The Green's function approach is based on the standard elastic half space model OKADA [14]. Three components relating to the surface deformation (longitude, latitude, altitude) are precomputed for every single patch as a result of dip- and strike slip. The output is stored in a database of Green's functions. It may be used to compute surface deformation for every single earthquake [1]. The 1D layered dislocation code and seismic velocities from the IASP91 model are used to compute the above mentioned database [15]. The plan is to implement a dislocation model which is based on 3D Finite Element Method (FEM). The Rupture Generator can be guided by the command line and by configuration files. The files of Green's functions are stored in the subdirectory pi and saved as a GRD format. Another setting file is PlateInterface.pi which includes the width and length of each patch and the coordinates of the corner points. The coordinates of the epicenter and the number of ruptured patches are saved in ruptgen.log [1]. The command in the command line has to be the following syntax:

$$\text{ruptgen.x}\{-\text{mw} < \text{mw} > | - \text{slip} < \text{slip} >\}\ \{-\text{xy} < \text{lon} >< \text{lat} > | - \text{ij} < \text{I} >< \text{J} >\}$$
$$\{-\text{output} < \text{string} >\}[-\text{rake} < 0 - 90 >][-\text{sclaw} < 1 \text{ or } 2 >][\text{distr} < 1 \text{ or } 2 >]$$
$$[\text{asymmetry} < 0 - 1 >][\text{taper} < 0 - 1 >]$$

It is possible to choose between a single slip value [m] (-slip) and the magnitude (-mw). Additionally, the epicenter can be defined by the longitude and latitude (-xy) or by a single patch (-ij). The i-index runs from 1 till 150 and the j-index from 1 to 25. The others must have option –output and can be specified in different ways. The most important ones are

- output gps < file_with_stations (output are the 3 displacements in x, y and z) >
- output slip (output slip dispersion [m])

Other output options are:

- output grid gridoption (output of the 3 displacements on a rectengular grid)
- output vgrid gridoption (output of the vertical displacements on a rectengular grid)

The option –rake (default value is 90°, pure dip slip) differentials between dip- and strike slip (strike slip is 0°). The additional option -rake can be used if slip or magnitude are chosen. The scaling laws can be specified with –sclaw.
The default option is

$$\text{lenght} = 2 \cdot \text{width}$$

and another variant is the scaling law by Wells and Coppersmith (1994). Other values can be defined with

- distr (distribution model).

The default value (distr=1) is the uniform model or the advanced model (distr=2) by Geist & Dmowska. The asymmetry option can vary between 0 and 1 and the default value 0.5 is symmetrical. The program processing depends on the entered input at the command line. If the magnitude (-mw) is given the rupture dimensions will be calculated by applying the scaling law. Hence the slip values will be placed around the epicenter and if the slip value is entered at the command line, only the epicenter patch is activated. Usually earthquakes follow scaling laws but the options have to be applied with care, because it is not usual that one patch is surrounded by patches with zero meters of slip [15].
A possible command with a given magnitude is:

- ruptgen. x − mw 8.4 − ij 100 10 − output slip

or a command with a single slip value and the output (3 displacements on every GPS receiver)

- ruptgen. x − slip 1 − xy 100 − 1 − output gps stations. dat

Another way to simulate an earthquake is to use multi single ruptures:

- ruptgen. x − slip 1 − ij 100 8

- $\text{ruptgen.x} - \text{slip } 1 - \text{ij } 100\ 9$
- $\text{ruptgen.x} - \text{slip } 1 - \text{ij } 100\ 10$

The three commands have to be written in myslip.dat and called with the following command:

- $\text{ruptgen.x} - \text{f myslip.dat} - \text{output gps stations.dat}$

Another way to simulate multi rupture earthquakes is:

- $\text{ruptgen.x} - \text{mw } 8.4 - \text{ij } 100\ 10 - \text{output gps stations.dat}$
- $\text{ruptgen.x} - \text{mw } 7.2 - \text{ij } 80\ 8 - \text{output slip}$

4.4 Confidence interval

A Confidence interval gives information about the interval estimation of a parameter, which is needed to display the reliability. It is an observed interval, basically different from sample to sample. Every GPS Sensor has a top interval boundary K_L and a bottom interval boundary K_R because of their noise. The noisy $\vec{b}^{\varepsilon}$ is a component of the following interval [17].

$$\vec{b}^{\varepsilon} \in \left[\vec{b} - K_R, \vec{b} + K_L\right]$$

Exemplary, there are two cases presented. They are both worst case estimations. The accuracy of the solution vector will be shown.

Case one

$$\|A \cdot \vec{x} - b^{\varepsilon}\|^2,$$

follows the error consideration of the least squares algorithm and describes the accuracy of the solution.

$$\vec{x}^{\varepsilon} = (A^T \cdot A)^{-1} \cdot A^T \cdot \vec{b}^{\varepsilon} = (A^T \cdot A)^{-1} \cdot A^T \cdot \left(\vec{b} + \varepsilon\right)$$

The noisy $\vec{b}^{\varepsilon}$ can be theoretical split into a part without noise and a noisy part.

$$(A^T \cdot A)^{-1} \cdot A^T \cdot \left(\vec{b} + \varepsilon\right) = (A^T \cdot A)^{-1} \cdot A^T \cdot \vec{b} + (A^T \cdot A)^{-1} \cdot A^T \cdot \varepsilon$$

Finally the result contains $\vec{x}$ and the attendant noise, but in the practical view it is not possible to split both.

$$(A^T \cdot A)^{-1} \cdot A^T \cdot \vec{b} + (A^T \cdot A)^{-1} \cdot A^T \cdot \varepsilon = \vec{x} + (A^T \cdot A)^{-1} \cdot A^T \cdot \varepsilon$$

Thus it follows

$$\|\vec{x}^{\varepsilon} - \vec{x}\| = \|(A^T \cdot A)^{-1} \cdot A^T \cdot \varepsilon\| \leq \frac{1}{\sigma_{min}} \cdot \|\varepsilon\| ,$$

Where σ_{min} is the smallest singular value of A, hence $\frac{1}{\sigma_{min}}$ goes against infinity. Small errors in b^{ε} induce large errors in $\vec{x}^{\varepsilon}$.

We consider case two

$$\|A \cdot \vec{x} - b^{\varepsilon}\| + \alpha \cdot \|\vec{x}\|^2$$

The Tykhonov Regularization is given through

$$\vec{x}^{\varepsilon} = (A^T \cdot A + \alpha \cdot I)^{-1} \cdot A^T \cdot \vec{b}^{\varepsilon}$$

and the Tykhonov Regularization via Singular value decomposition is hereby assigned by

$$A^T \cdot A + \alpha \cdot I = V \cdot \Sigma \cdot V^T + \alpha \cdot V \cdot V^T = V \cdot (\Sigma^2 + \alpha \cdot I) \cdot V^T$$

It is essential to define a rule when the algorithm does not work well, because of the confidence interval.

$$\|(A^T \cdot A + \alpha \cdot I)^{-1} \cdot A^T \cdot \varepsilon\| \leq \|V \cdot (\Sigma^2 + \alpha \cdot I)^{-1} \cdot V^T \cdot (V \cdot \Sigma \cdot U^T)\| \cdot \|\varepsilon\|$$

The simplified algorithm is given by

$$\|(A^T \cdot A + \alpha \cdot I)^{-1} \cdot A^T \cdot \varepsilon\| \leq \|(\Sigma^2 + \alpha \cdot I)^{-1}\| \cdot \|\Sigma\| \cdot \|\varepsilon\| ,$$

thus it follows

$$\|(A^T \cdot A + \alpha \cdot I)^{-1} \cdot A^T \cdot \varepsilon\| \leq \frac{\sigma_{max}}{(\sigma_{min}+\alpha)^2} \cdot \|\varepsilon\| .$$

where σ_{min} is the smallest singular value and σ_{max} the largest singular value of A. Hence $\frac{\sigma_{max}}{(\sigma_{min}+\alpha)^2}$ goes against a boundary value in addiction to the singular values and the chosen α.
Small errors in b^{ε} induce a stable $\vec{x}^{\varepsilon}$.

5 Mathematical Principles

5.1 Underdetermined inverse problem

In an underdetermined case, there are more equations than unknown and there are exists infinity solutions of the equation system. That means there are fewer sensors than subfaults, in this case six sensors and 3750 subfaults. Therefore the algorithms have to be stabilized and regularized. There are two kinds of problems, the forward problem (direct problem) and the inverse problem. The forward problem can be conceptually formulated as a transformation of data into model parameters. An inverse problem relates the model parameter to the observed data. Therefore, it contains a part of the solution of the direct problem, in this case the matrix $\vec{b}$. Finally, an inverse problem works back from the effect to the cause [20]. The Rupture Generator is used to simulate the displacements of the GPS - sensors ($\vec{b}$) for a given slip distribution ($\vec{x}$). It can be formulated as follows

$$A \cdot \vec{x} = \vec{b},$$

where

$\vec{b} = \begin{pmatrix} \vec{b}_x \\ \vec{b}_y \\ \vec{b}_z \end{pmatrix} \in \mathbb{R}$, $\vec{b}_i \sim$ displacement of a GPS-sensor,

$\vec{x} = \begin{pmatrix} x^1 \\ x^2 \end{pmatrix}$, $x^1 \sim$ dip slip, $x^2 \sim$ strike slip, $x^n \in \mathbb{R}^{3750}$.

$\vec{x}$ represents the slip distribution to be estimated and each element of $\vec{x}_i$ refers to one of 3750 patches [15].

A represents the propagation matrix (Appendix A.2) and the resulting equation system leads to

$$A \cdot \vec{x} = \begin{pmatrix} A_{dip,x} & A_{strike,x} \\ A_{dip,y} & A_{strike,y} \\ A_{dip,z} & A_{strike,z} \end{pmatrix} \cdot \begin{pmatrix} \vec{x}_{dip} \\ \vec{x}_{strike} \end{pmatrix} = \begin{pmatrix} \vec{b}_x \\ \vec{b}_y \\ \vec{b}_z \end{pmatrix} = \vec{b}\,.$$

In following we have a noisy $\vec{b}^{\epsilon}$, for reasons of clarity and comprehensibility we write $\vec{b}$. Depended on the algorithm, the notation may changed.

5.2 QR factorization

An underdetermined system of equations with $A \in \mathbb{R}^{m \times n}$ and $m < n$

can be solved by QR factorization of the transposed system

$$A^T = Q \cdot R ,$$

where

$Q \in \mathbb{R}^{n \times n}$ and orthogonal ($Q^T \cdot Q = I$) , $R \in \mathbb{R}^{n \times m}$ and a upper triangular matrix .

The QR factorization is usually calculated by the Gram Schmidt algorithm, because of the small rounding errors [5].

The reduced QR factorization is given through

$Q \cdot R = (Q_1 Q_2) \cdot \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$,$Q \in \mathbb{R}^{n \times m}$ and $R \in \mathbb{R}^{m \times m}$ thus it follows

$$A^T = Q_1 \cdot R_1 .$$

5.3 Singular Value decomposition

Assume that $A \in \mathbb{R}^{m \times n}$ with $m < n$, therefore the matrix A can be factored as

$$A = U \cdot \Sigma \cdot V^T ,$$

where

$U \in \mathbb{R}^{m \times m}$ and orthogonal ($U^T \cdot U = I$) and U_i are the left singular values of A,

$\Sigma \in \mathbb{R}^{m \times n}$ and diagonal, $\mathrm{diag}(\sigma_1 - \sigma_p)$ and Σ_i are the singular values of A,

$V \in \mathbb{R}^{n \times n}$ and orthogonal ($V^T \cdot V = I$) and V_i are the right singular values of A .

Based on the underdetermined case, $\Sigma \in \mathbb{R}^{m \times n}$ can be reduced to $\Sigma_1 \in \mathbb{R}^{m \times m}$

and $V \in \mathbb{R}^{n \times n}$ can be reduced to $V \in \mathbb{R}^{n \times m}$.

Thus the truncated SVD decomposition is given by

$$A^{m \times n} = U \cdot \Sigma_1^{m \times m} \cdot (V_1^{n \times m})^T .$$

Hence V_1 and S_1 has a smaller size, thus the computation is faster and less memory is required [6].

5.4 Direct solver

5.4.1 Least norm solution via Singular Value Decomposition

Assume the linear equation

$$A \cdot \vec{x} = \vec{b}, \tag{1.1}$$

where $\vec{x} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ and $\vec{b} \in \mathbb{R}^m$.

The matrix A can be factored as

$$A = U \cdot \Sigma_1 \cdot V_1^T, \tag{1.2}$$

where $U \in \mathbb{R}^{m \times m}$, $\Sigma \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times m}$ and plugged into (6.11)

leads to

$$U \cdot \Sigma_1 \cdot V_1^T \cdot \vec{x} = \vec{b} . \tag{1.3}$$

After multiplying by U^T the rewritten equation is given by

$$\Sigma_1 \cdot (V_1^T \cdot x) = U^T \cdot \vec{b} . \tag{1.4}$$

Therefore, the forward substitution $y = V_1^T \cdot \vec{x}$ reduces system to

$$\Sigma_1 \cdot y = U^T \cdot \vec{b} , \tag{1.5}$$

Σ_1 has to be inverted and y leads to

$$y = \Sigma_1^{-1} \cdot U^T \cdot \vec{b}. \tag{1.6}$$

The backward substitution leads to

$$\vec{x} = V_1 \cdot y . \tag{1.7}$$

Therefore, it follows

$$\vec{x} = V_1 \cdot \Sigma_1^{-1} \cdot U^T \cdot \vec{b} . \tag{1.8}$$

The pseudo inverse or Moore-Penrose inverse is given by

$$A^{\dagger} = ((A^T \cdot A)^{-1} \cdot A^T) = V_1 \cdot \Sigma_1^{-1} \cdot U^T \text{ [8].} \tag{1.9}$$

5.4.2 Least norm solution via QR factorization

Consider the linear equation

$$A \cdot \vec{x} = \vec{b}, \tag{2.1}$$

where $\vec{x} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ and $\vec{b} \in \mathbb{R}^m$.

The transposition of the forward algorithm yields to

$$\vec{x}^T \cdot A^T = \vec{b}^T , \tag{2.2}$$

plugged $A^T = Q_1 \cdot R_1$ in the above equation it follows

$$\vec{x}^T \cdot Q_1 \cdot R_1 = \vec{b}^T . \tag{2.3}$$

Therefore, the reduced QR decomposition of the transposed system is given by

$$R_1^T \cdot Q_1^T \cdot \vec{x} = \vec{b} . \tag{2.4}$$

The forward substitution is given by

$$y = Q_1^T \cdot \vec{x} , \tag{2.5}$$

in order that

$$R_1^T \cdot y = \vec{b} . \tag{2.6}$$

Therefore, it follows

$$y = (R_1^T)^{-1} \cdot \vec{b} . \tag{2.7}$$

The backward substitution is leading to

$$\vec{x} = Q_1 \cdot y . \tag{2.8}$$

Hence the solution yields to

$$\vec{x} = Q_1 \cdot (R_1^T)^{-1} \cdot \vec{b} \text{ [8].} \tag{2.9}$$

5.4.3 Least squares

The aim is to find an approximate solution that minimizes

$$\left\|A \cdot \vec{x} - \vec{b}\right\|^2 = x^T \cdot A^T \cdot A \cdot \vec{x} - 2 \cdot \vec{x}^T \cdot A^T \cdot \vec{b} + \vec{b}^T \cdot \vec{b}\,, \tag{3.1}$$

where

$\vec{x} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $m < n$ and $\vec{b} \in \mathbb{R}^m$.

To minimize we take the first derivative respect to $\vec{x}$ and set

$$2 \cdot A^T \cdot A \cdot \vec{x} - 2 \cdot A^T \cdot \vec{b} = 0 \tag{3.2}$$

to zero [7].

Therefore, the normal equation is leading to

$$(A^T \cdot A) \cdot \vec{x} = A^T \cdot \vec{b}\,. \tag{3.3}$$

Hence the approximate solution is hereby assigned by

$$\vec{x} = (A^T \cdot A)^{-1} A^T \cdot \vec{b}\,. \tag{3.4}$$

The Least norm solution via QR factorization is given through

$$\vec{x} = Q_1 \cdot (R_1 \cdot R_1^T)^{-1} \cdot R_1 \cdot \vec{b}\,. \tag{3.5}$$

5.4.4 Tikhonov Regularization

Tikhonov regularization is the most commonly used algorithm of regularization of ill posed problems and is named for Andrey Nikolayevich Tikhonov, a Russian mathematician.

We have the following problem

$$A \cdot \vec{x} = \vec{b}\,. \tag{4.1}$$

The standard way to solve this problem is known as ordinary least squares and seeks to minimize

$$\left\|A \cdot \vec{x} - \vec{b}\right\|^2\,. \tag{4.2}$$

where $\|\cdot\|$ is the Euclidian norm. This can be done for overdetermined and underdetermined systems of equations. To get a better solution the regularization term

$$\frac{1}{2}\left\|\mathrm{A}\cdot\vec{\mathrm{x}}-\vec{\mathrm{b}}\right\|^{2}+\frac{1}{2}\|\Gamma\cdot\vec{\mathrm{x}}\|^{2} \tag{4.3}$$

has to be is included,

thus it follows

$$=\frac{1}{2}\cdot\left(\vec{\mathrm{x}}^{\mathrm{T}}\cdot\mathrm{A}^{\mathrm{T}}\cdot\mathrm{A}\cdot\vec{\mathrm{x}}+\vec{\mathrm{b}}^{\mathrm{T}}\cdot\vec{\mathrm{b}}-2\cdot\vec{\mathrm{x}}^{\mathrm{T}}\cdot\mathrm{A}^{\mathrm{T}}\cdot\vec{\mathrm{b}}\right)+\frac{1}{2}\cdot\Gamma\cdot\vec{\mathrm{x}}^{\mathrm{T}}\cdot\vec{\mathrm{x}}\,. \tag{4.4}$$

In many cases the matrix Γ is choosen as the identity matrix and has to

be larger than zero [23].

The first derivative is given of the form

$$\mathrm{A}^{\mathrm{T}}\cdot\mathrm{A}\cdot\vec{\mathrm{x}}-\mathrm{A}^{\mathrm{T}}\cdot\vec{\mathrm{b}}+\Gamma\cdot\vec{\mathrm{x}}=0\,, \tag{4.5}$$

in order that

$$(\mathrm{A}^{\mathrm{T}}\cdot\mathrm{A}\cdot+\Gamma\cdot\mathrm{I})\cdot\vec{\mathrm{x}}=\mathrm{A}^{\mathrm{T}}\cdot\vec{\mathrm{b}} \tag{4.6}$$

The approximate minimizer of the weighted sum objective is

$$\vec{\mathrm{x}}=(\mathrm{A}^{\mathrm{T}}\cdot\mathrm{A}+\Gamma^{\mathrm{T}}\cdot\Gamma)^{-1}\cdot\mathrm{A}^{\mathrm{T}}\cdot\vec{\mathrm{b}}\,. \tag{4.7}$$

A solution with a smaller norm will be expected and the regularization improves the condition of the problem. Related to the Bayes' theorem, the Tikhonov matrix is then $\Gamma=\alpha\cdot\mathrm{I}$.

An advanced method is the generalized Tikhonov regularization

$$\vec{\mathrm{x}}_{\mathrm{n+1}}=(\mathrm{A}^{\mathrm{T}}\cdot\mathrm{P}\cdot\mathrm{A}+\alpha\cdot\mathrm{I})^{-1}\cdot\left(\mathrm{A}^{\mathrm{T}}\cdot\mathrm{P}\cdot(\vec{\mathrm{b}}-\mathrm{A}^{\mathrm{T}}\cdot\vec{\mathrm{x}}_{\mathrm{n}})\right)\,, \tag{4.8}$$

where P is the inverse covariance matrix of $\vec{\mathrm{b}}$ [23].

5.4.4.1 Computing Tikhonov regularization using SVD decomposition

The Tikhonov regularization by using the SVD decomposition is given through

$$(V_1 \cdot \Sigma_1^2 \cdot V_1^T + \alpha \cdot I) \cdot \vec{x} = V_1 \cdot \Sigma_1^T \cdot U^T \cdot \vec{b}\,, \tag{4.9}$$

remark that

$$A^T \cdot A = V_1 \cdot \Sigma_1^2 \cdot V_1^T\,. \tag{4.10}$$

V_1 and V_1^T can be factorized to

$$V_1 \cdot (\Sigma_1^2 + \alpha \cdot I) \cdot V_1^T \cdot \vec{x} = V_1 \cdot \Sigma_1^T \cdot U^T \cdot \vec{b}\,, \tag{4.11}$$

multiplying with V_1leads to

$$(\Sigma_1^2 + \alpha \cdot I) \cdot V_1^T \cdot \vec{x} = \Sigma_1^T \cdot U^T \cdot \vec{b}\,. \tag{4.12}$$

Hence the solution formula is given through

$$\vec{x} = V_1 \cdot (\Sigma_1^2 + \alpha \cdot I)^{-1} \cdot \Sigma_1^T \cdot U^T \cdot \vec{b}\,. \tag{4.13}$$

5.4.4.2 Computing Tikhonov regularization using QR factorization

The Tykonov regularization by using the QR factorization is given through

$$\left((Q_1 \cdot R_1 \cdot (Q_1 \cdot R_1)^T) + \alpha \cdot I\right) \cdot \vec{x} = Q_1 \cdot R_1 \cdot \vec{b}\,, \tag{4.14}$$

Q_1 and Q_1^T can be factorized to

$$Q_1 \cdot (R_1 \cdot R_1^T + \alpha \cdot I) \cdot Q_1^T \cdot \vec{x} = Q_1 \cdot R_1 \cdot \vec{b}\,. \tag{4.15}$$

Multiplying with Q_1^T it yields to

$$(R_1 \cdot R_1^T + \alpha \cdot I) \cdot Q_1^T \cdot \vec{x} = R_1 \cdot \vec{b}\ . \tag{4.16}$$

Adapted from forward and backward substitution the solution formula is given by

$$\vec{x} = Q_1 \cdot (R_1 \cdot R_1^T + \alpha \cdot I)^{-1} \cdot R_1 \cdot \vec{b}\,. \tag{4.17}$$

5.4.4.3 L-curve criterion

The optimal regularization parameter α can be determined graphically. The basis is a 2d plot for a valid regularization parameter. This is the way to show the dependence between $\|\vec{x}\|$ and $\|A \cdot \vec{x} - \vec{b}\|$. The output of least squares is impaired by rounding errors and data errors. Hence, regularization is useful. Regularization is damping these errors but if the regularization is too high, they do not fit the given data, thus $\|A \cdot \vec{x} - \vec{b}\|$ is too large. Therefore, a small regularization involving a good fit of the data is useful, but in this case the data error is larger.

5.5 Iterative solver

5.5.1 Modified Richardson Iteration

The Richardson Iteration was suggested by Lewis Richardson in 1910 and is almost similar to Gaus-Seidel- and Jacobi method [19].

We have the following problem

$$A \cdot \vec{x} = \vec{b}\,. \tag{5.1}$$

The standard way to solve this is to minimize the following

$$\|A \cdot \vec{x} - \vec{b}\|^2 = x^T \cdot A^T \cdot A \cdot \vec{x} - 2 \cdot \vec{x}^T \cdot A^T \cdot \vec{b} + \vec{b}^T \cdot \vec{b}\,, \tag{5.2}$$

$$\text{where } \vec{x} \in \mathbb{R}^n\,, A \in \mathbb{R}^{m \cdot n},\ \vec{b} \in \mathbb{R}^m\,.$$

Therefore, the normal equation is given through

$$(A^T \cdot A) \cdot \vec{x} = A^T \cdot \vec{b}\,. \tag{5.3}$$

The fixed point iteration is

$$\vec{x}_n = (I - A^T \cdot A) \cdot \vec{x}_n + A^T \cdot \vec{b}\,, \tag{5.4}$$

in order that the Richardson iteration is defined by

$$\vec{x}_{n+1} = (I - A^T \cdot A) \cdot \vec{x}_n + A^T \cdot \vec{b}\,. \tag{5.5}$$

It is necessary to use the relaxation parameter α, which improves the convergence and weights the residuum [19]. The parameter α has to be between $0 < \alpha < \frac{2}{\lambda_{max}}$ and the optimal choice is

$$\alpha_{opt} = \frac{2}{\lambda_{max}+\lambda_{min}} .$$

Hence, the Modified Richardson Iteration is given through

$$\vec{x}_{n+1} = (I - \alpha \cdot A^T \cdot A) \cdot \vec{x}_n + \alpha \cdot A^T \cdot \vec{b} . \tag{5.6}$$

The method converges if

$$\|I - \alpha \cdot A\| < 1 .$$

5.5.1.1 Modified Richardson Iteration using Singular Value Decomposition

A has to be replaced by $A = U \cdot \Sigma \cdot V^T$ hence the iterative algorithm is given through

$$\vec{x}_{n+1} = \left(I - \alpha \cdot (U \cdot \Sigma_1 \cdot V_1^T)^T \cdot (U \cdot \Sigma_1 \cdot V_1^T)\right) \cdot \vec{x}_n + \alpha \cdot U \cdot \Sigma_1 \cdot V_1^T \cdot \vec{b} . \tag{5.7}$$

Therefore, the solution formula is given by

$$x_{n+1} = V_1 \cdot \left((I - \alpha \cdot \Sigma_1^2) \cdot V_1^T \cdot \vec{x}_n + \alpha \cdot \Sigma_1^T \cdot U^T \cdot \vec{b}\right) . \tag{5.8}$$

5.5.1.2 Modified Richardson Iteration using QR factorization

A has to be replaced by $A = Q_1 \cdot R_1$, hence the iterative algorithm is given through

$$x_{n+1} = \left(\left(I - \alpha \cdot \left((Q_1 \cdot R_1 \cdot (Q_1 \cdot R_1)^T)\right)\right) \cdot \vec{x}_n + \alpha \cdot Q_1 \cdot R_1 \cdot \vec{b}\right) . \tag{5.9}$$

The simplified algorithm is given by

$$x_{n+1} = Q_1 \cdot \left((I - R_1 \cdot R_1^T) \cdot Q_1^T \cdot \vec{x} + R_1 \cdot \vec{b}\right) \text{ [19].} \tag{5.10}$$

5.5.2 Kazcmarz's Method

The aim is to minimize

$$\min \left\| A \cdot \vec{x} - \vec{b} \right\|^2 \tag{6.1}$$

where $\vec{x} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $\vec{b} \in \mathbb{R}^m$.

The normal equation is given by

$$(A^T \cdot A) \cdot \vec{x} = A^T \cdot \vec{b} \,. \tag{6.2}$$

The equivalent linear fixed point iteration is hereby assigned

$$\vec{x} = \vec{x} - A^T \cdot \left(A \cdot \vec{x} - \vec{b} \right) . \tag{6.3}$$

We have only an approximation b^δ of the exact data if the condition

$$\left\| \vec{b}^\varepsilon - \vec{b} \right\| < \varepsilon$$

is complied.

Hence the Kaczmarz's sequence is

$$\vec{x}_{n+1} = \vec{x}_n - A^T \cdot \left(A \cdot \vec{x}_n - \vec{b}^{[\varepsilon]} \right) . \tag{6.4}$$

It is sensible to use a relaxation parameter α_n in combination with a new stopping rule [10]

$$\vec{x}_{n+1} = \vec{x}_n - \omega_n \cdot \alpha \cdot A^T \cdot \left(A \cdot \vec{x}_n - \vec{b}^{[\varepsilon]} \right), \tag{6.5}$$

where

$$\omega_n := \omega_n\left(\varepsilon, \vec{b}^\varepsilon\right) = \left\{ \begin{matrix} 1 & \left\| A \cdot x_n - b^\varepsilon \right\| > \tau \cdot C \\ 0 & \text{otherwise} \end{matrix} \right\} . \tag{6.6}$$

The choice of α is given by

$$0 < \alpha < \frac{2}{\|A\|^2} .$$

A modification of the Kaczmarz's algorithm is a blockwise Kaczmarz's method of the form

$$\vec{x}_{n+1} = \vec{x}_n - \omega_{[n]} \cdot \alpha_i \cdot A_{[i]}^T \cdot \left(A_{[i]} \cdot \vec{x}_n - \vec{b}^{[\varepsilon,i]} \right) . \tag{6.7}$$

The aim is to solve the equation separately for $i = \begin{bmatrix} w_{\vec{x}} \\ w_{\vec{y}} \\ w_{\vec{z}} \end{bmatrix}$.

The parameter $\tau > 2$ is an appropriate chosen constant.
Each sensor has a confidence interval δ (for $\vec{x}, \vec{y}, \vec{z}$) and C represents $\|\delta\|_2$. Therefore we can choose one confidence interval or we can choose three different confidence intervals for $\vec{x}, \vec{y}, \vec{z}$.
The iteration terminates if $w_{\vec{x}}$, $w_{\vec{y}}$ and $w_{\vec{z}}$ becomes zero.

5.5.2.1 Computing Kaczmarz's method using Singular value decomposition

After replacing $A = U \cdot \Sigma \cdot V^T$ the equation leads to

$$\vec{x}_{n+1} = \vec{x}_n - \left((U \cdot \Sigma_1 \cdot V_1^T)^T \cdot U \cdot \Sigma_1 \cdot V_1^T \cdot \vec{x}_n - (U \cdot \Sigma_1 \cdot V_1^T)^T \cdot \vec{b}^{[\varepsilon]}\right). \tag{6.8}$$

Hence the simplified algorithm is

$$\vec{x}_{n+1} = \vec{x}_n - V_1\left(\Sigma_1^2 \cdot V_1^T \cdot \vec{x}_n - \Sigma_1^T \cdot U^T \cdot \vec{b}^{[\varepsilon]}\right). \tag{6.9}$$

5.5.2.2 Computing Kaczmarz's method using QR factorization

By replacing $A^T = Q_1 \cdot R_1$ we get the reformulated algorithm

$$\vec{x}_{n+1} = \vec{x}_n - \left((Q_1 \cdot R_1 \cdot (Q_1 \cdot R_1)^T)\right) \cdot \vec{x}_n - Q_1 \cdot R_1 \cdot \vec{b}^{[\varepsilon]}, \tag{6.10}$$

therefore it follows

$$\vec{x}_{n+1} = \vec{x}_n - Q_1 \cdot R_1 \cdot (Q_1^T \cdot R_1^T \cdot \vec{x}_n - \vec{b}^{[\varepsilon]}). \tag{6.11}$$

6 Results

6.1 L_2 norm estimation

The Euclidian norm (L_2 norm) is given through

$$\|\vec{x}\|_2 = \sqrt{\sum_i x_i}\ .$$

In this case the Euclidian distance between the original slip distribution x_{opt} (given by RuptGen, see figure 6.1) and $\vec{x}$ (computed by an algorithm) is given by

$$\left\|\vec{x}_{opt} - \vec{x}\right\|_2 = \sqrt{\sum_i \left(x_{opt,i} - x_i\right)^2}$$

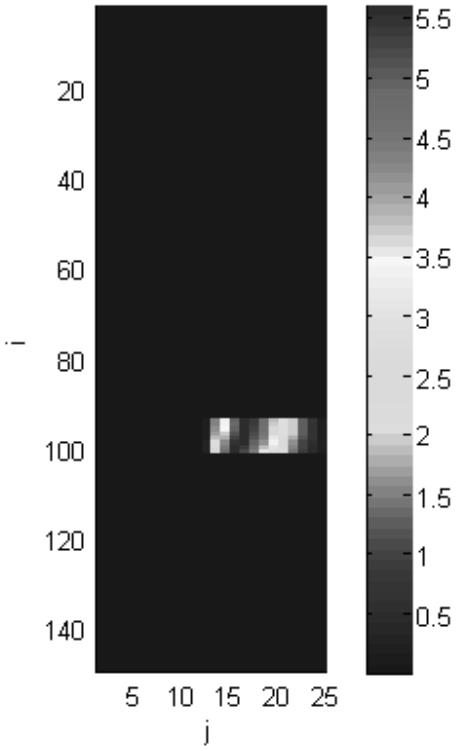

Fig 6.1: slip distribution created with RuptGen, MW=8.5, XY=100 -1

The slip distribution of every algorithm is computed 1000 times (with $\vec{b}$ + random noise ε) and averaged, because one computation is not representative.

The Tykhonov Regularization provides the best result, because of their smallL_2 norm (see figure 6.1). The Richardson Iteration and Landweber algorithm are decent, because their L_2 norm is larger. In the image are a sense of linearity between L_2 norm and the noise except the Kaczmarz method. There are a lot swings in the curve. All iterative algorithms have a new stopping rule (comparison with formula 6.6). The sense of the new stopping rule is that the algorithm interrupts before the solution becomes worse.

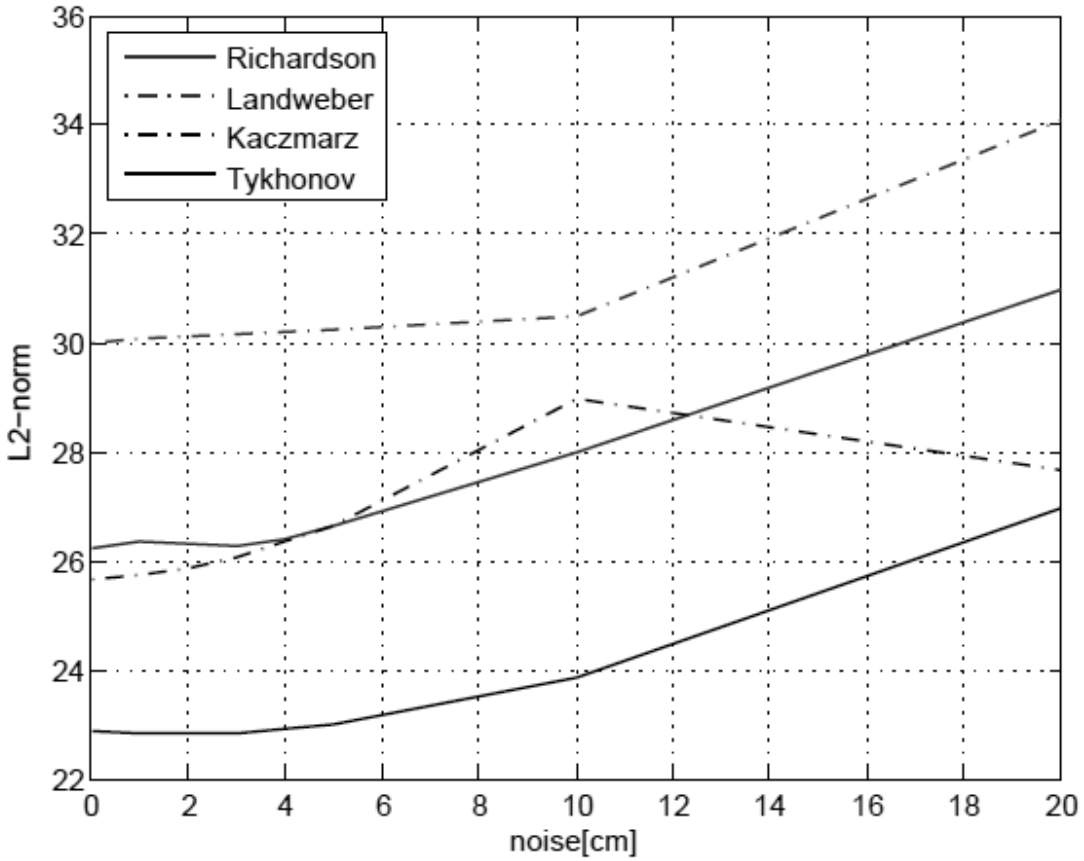

Fig 6.2: L_2 norm in addiction to the noise level

The Tykhonov Regularization is the most stable algorithm (in comparison to 6.1). It is a direct method, therefore no stopping rule is required. The regularization parameter Γ makes the algorithm stable and less sensitive. Figure 6.2 shows differences between a noiseless slip distribution and a slip distribution with noise=5cm. The right image in Figure 6.2 shows a bit sprays above and on the left side of the earthquake in comparison with the left image. The maximal amplitude of both objects is nearly the same.

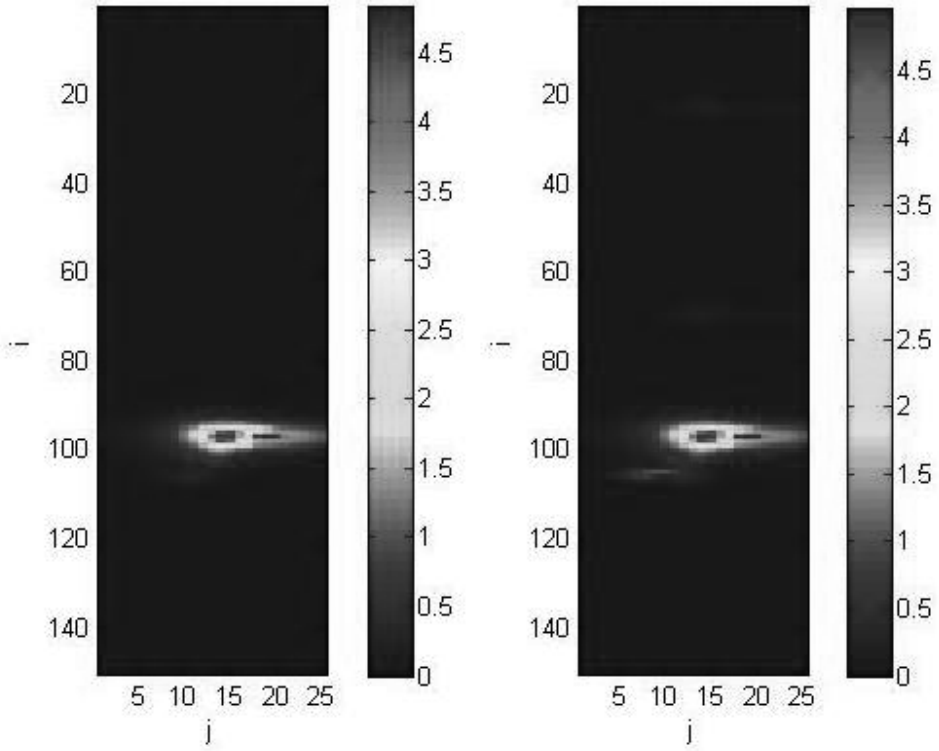

Fig 6.3: slip distribution, Tykhonov Regularization, noise level = 0cm (left), noise level = 5cm (right)

The Kaczmarz Method (see figure 6.3) is visual mostly the same in comparison to Tykhonov Regularization. The maximal amplitude is lower by comparison with $\vec{x}_{opt}$ (see figure 6.1). Furthermore, the maximal amplitude decreases appreciable, however, the Tykhonov Regularization (see figure 6.2) is stable and the maximal amplitude only decreases a little.

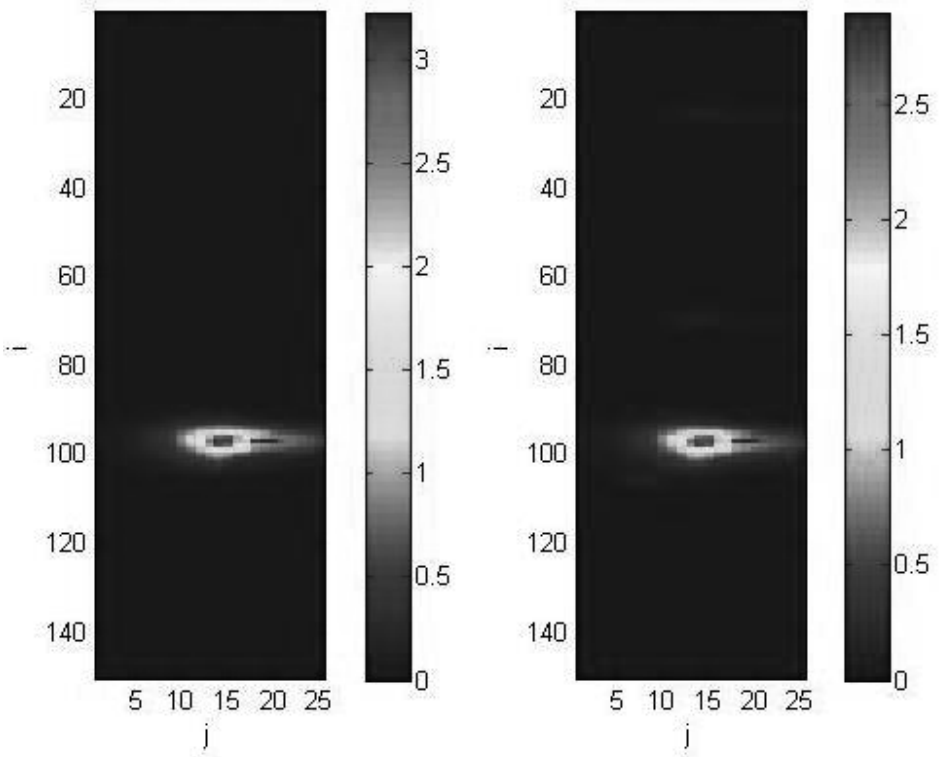

Fig 6.4: slip distribution, Karzmarz Method, noise level = 0cm (left), noise level = 5cm (right)

The Modified Richardson Iteration is a stable algorithm with the fewest spray of all algorithms. The left part of figure 6.4 shows the noiseless solution. The maximal amplitude is very close to $\vec{x}_{opt}$ (see figure 6.1) but the maximal amplitude decreases too fast (right part of Figure 6.4). The algorithm is less stable than the Tykhonov Regularization (see figure 6.2).

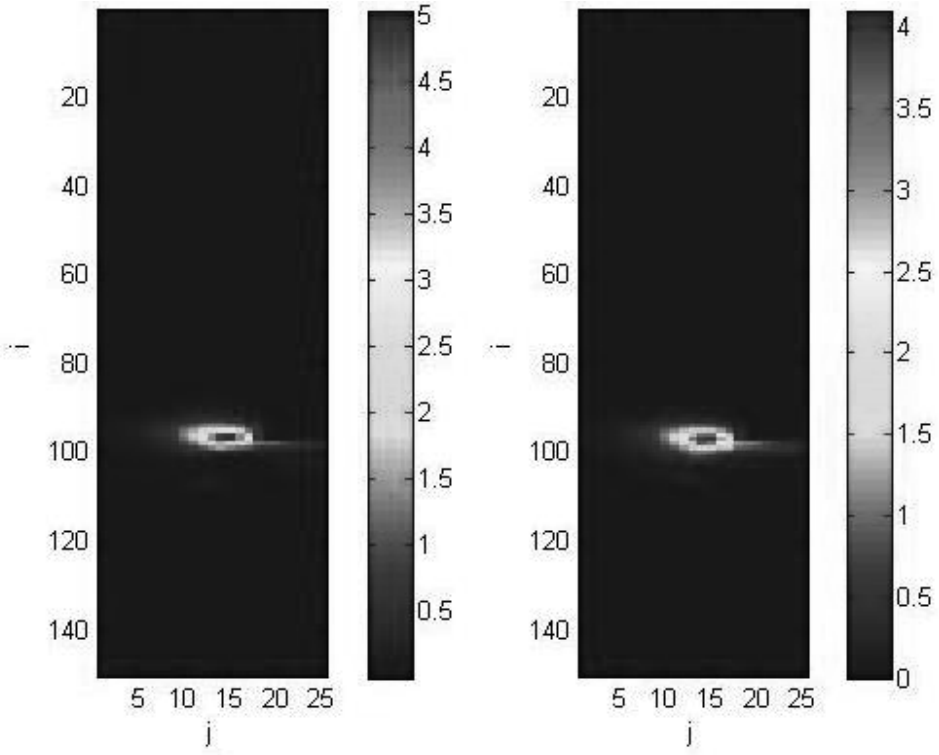

Fig. 6.5: slip distribution, Modified Richardson Iteration, noise level = 0cm (left), noise level = 5cm (right)

The Landweber Method (see figure 6.5) is an algorithm with a lesser maximal amplitude but the method is stable, because the maximal amplitude changes less. The effect of spray is similar to Tykhonov and Kaczmarz.

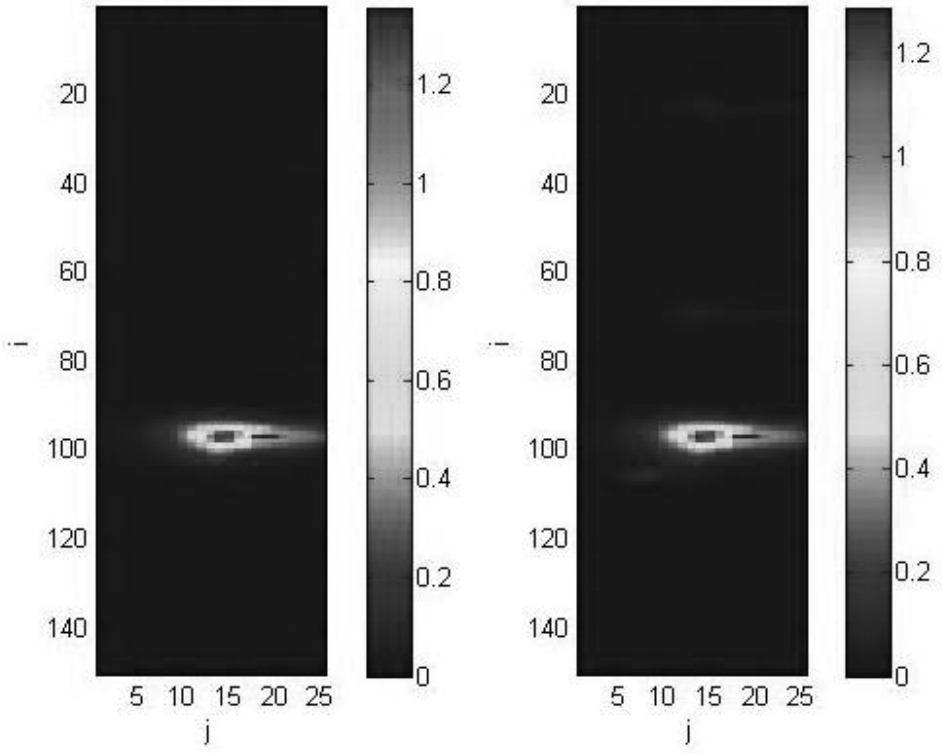

Fig. 6.6: slip distribution, Landweber Method, noise level = 0cm (left), noise level = 5cm (right)

The SVD method and LSQ Method give much worse results. The effect of noise makes both algorithms very sensitive. Therefore, the slip values are increasing very fast and so the error vector increases too. There is no sense to use the algorithms to solve such problems, because they are not stable (image).

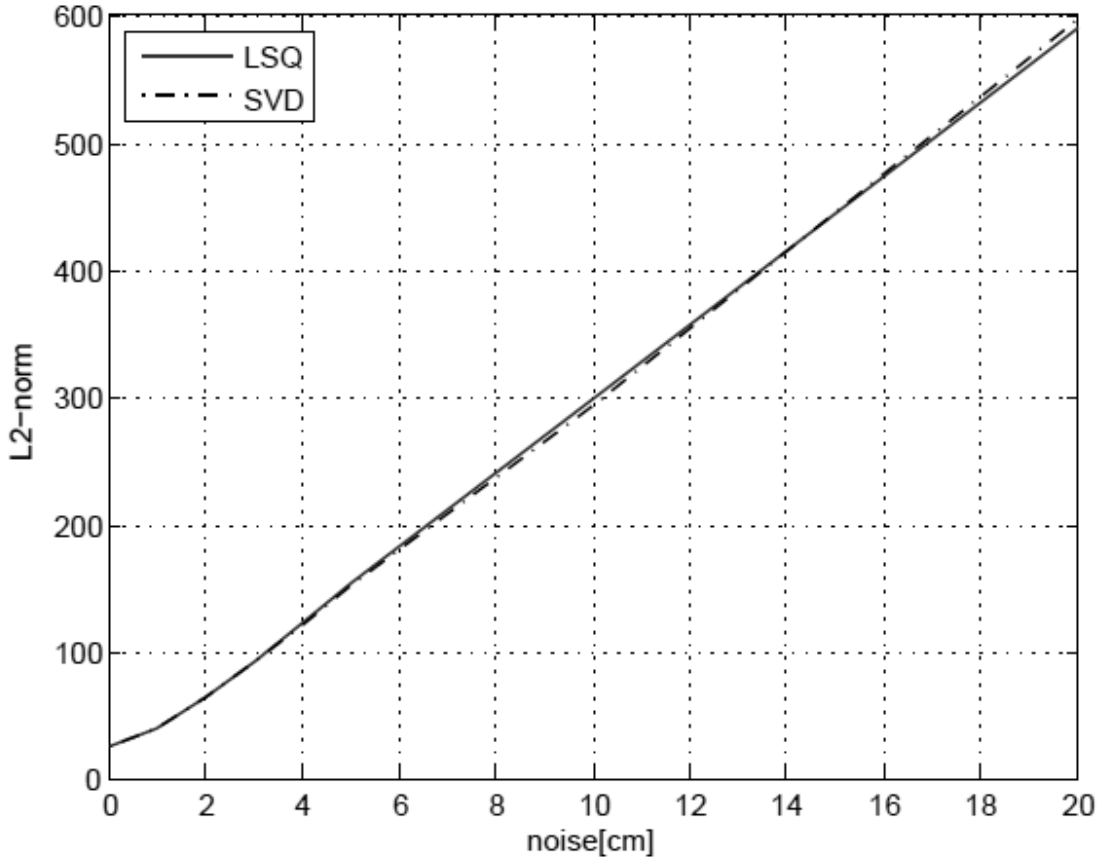

Fig. 6.7: L_2 norm in addiction to the noise level, LSQ, SVD

The least norm solution via SVD decomposition (see figure 6.6) is very unstable, because the influence of noise changes this result dramtically. It is not possible to locate the earthquake. The earthquake displaces to the north of the right part of the image (see figure 6.6). In addition, the maximal amplitude increases. Therefore, it is not possible to say something true about the earthquake.

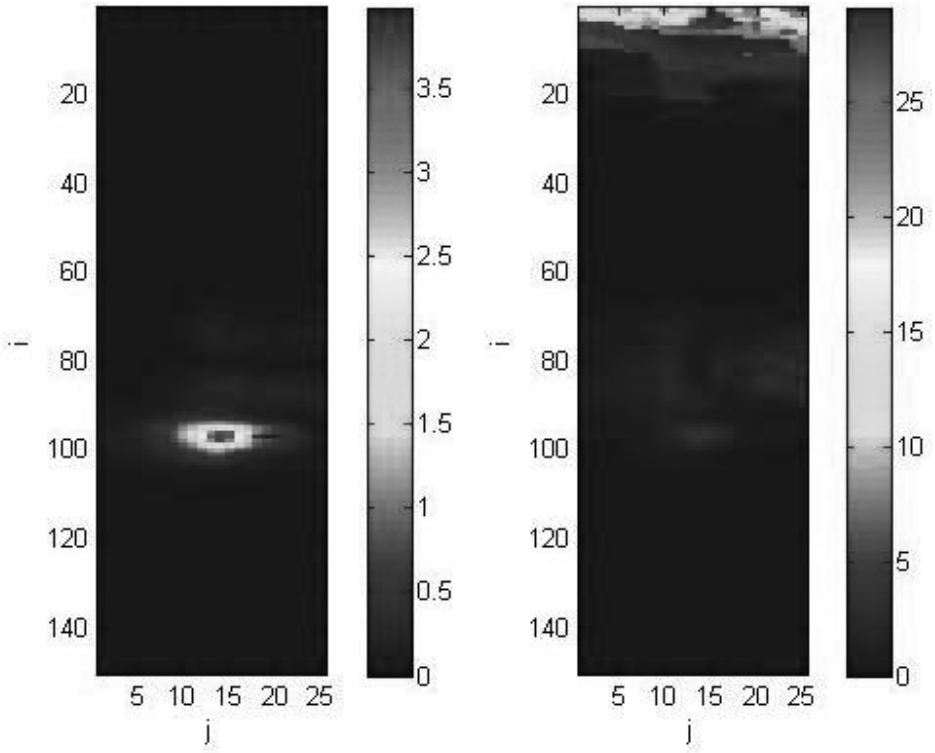

Fig. 6.8: slip distribution, SVD Method, noise level = 0cm (left), noise level = 5cm (right)

The result which is given trough LSQ is the same as SVD, because the worse slip values und maximal amplitude are mostly the same. The algorithm is also not stable and to sensitive. It makes no sense to use the method in that case.

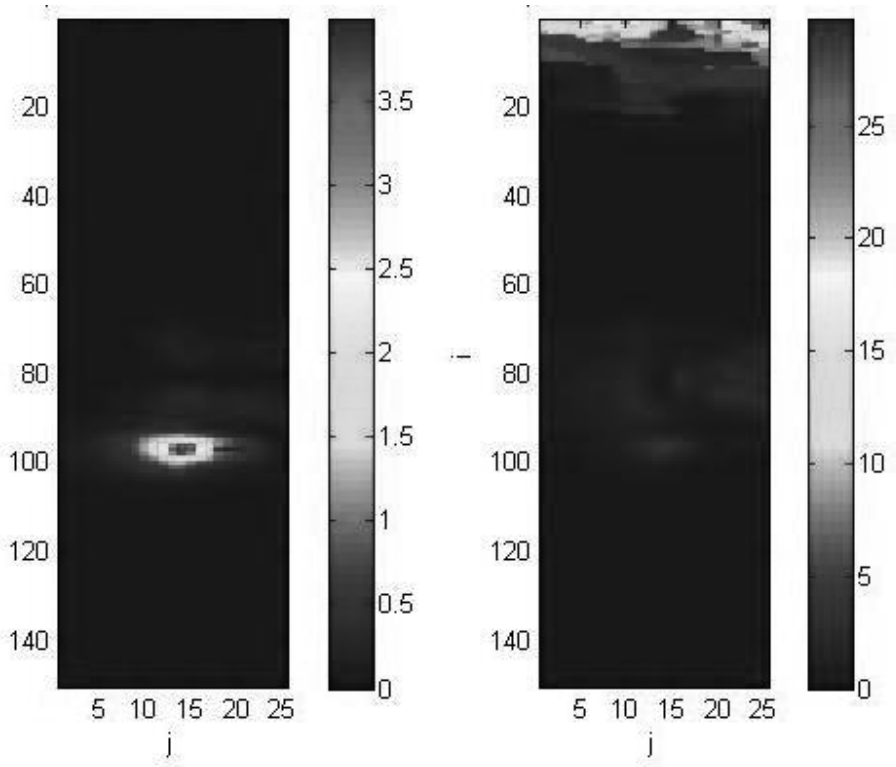

Fig. 6.9: slip distribution, LSQ Method, Left: noise level 0cm, right: noise level 5cm

6.2 Amplitude comparison

The largest slip value is an indicator how powerful an earthquake can be. It is essential to calculate the amplitude correctly. The Tykhonov Regularization has the best result, because differences between $\max(\vec{x}) - \max(\vec{x}_{opt})$ are very small. The algorithm is stable in comparison with the other three algorithms. The SVD and LSQ method are not part of the comparison, because their result is too bad. If there is noise till 1cm, the Richardson Iteration supplies a better result, but the algorithm is not stable and too sensitive (see figure 6.8).

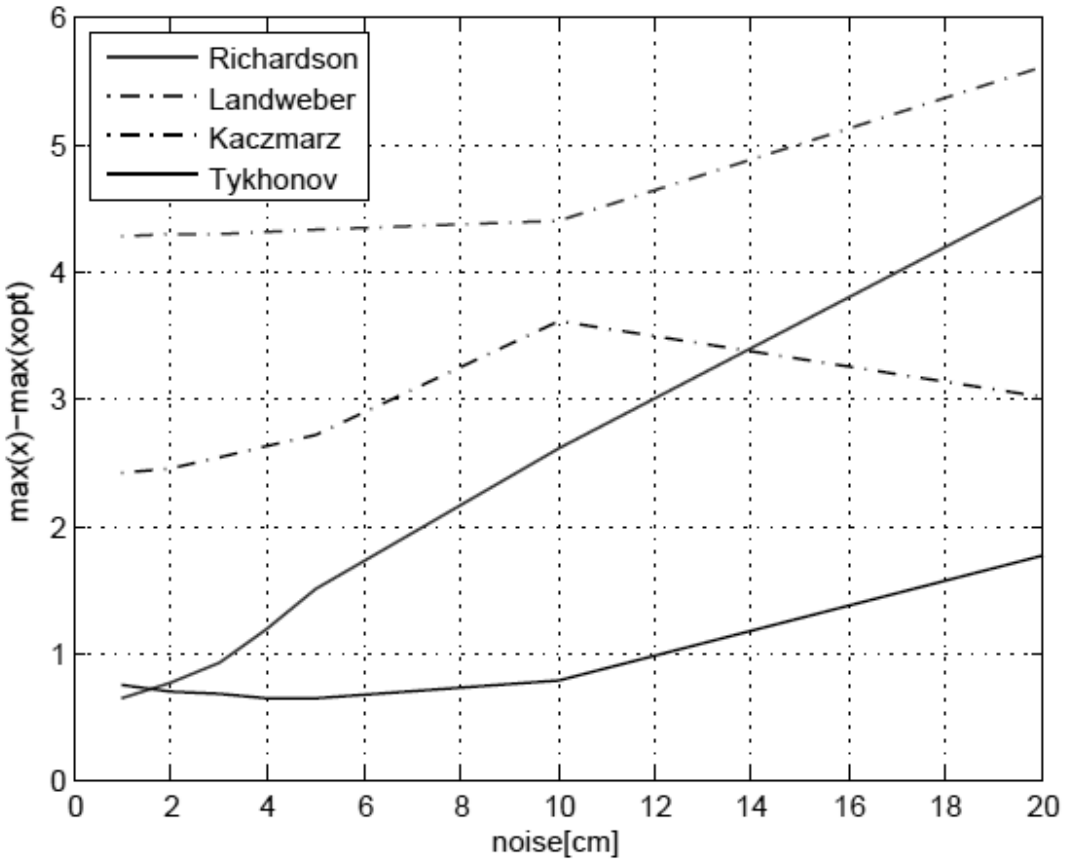

Fig. 6.10: amplitude comparison

6.3 Accuracy of the earthquake localization

The localization of an earthquake is another important factor to determine the accuracy of an algorithm. The distance between the largest value of the calculated slip distribution and given slip distribution (RuptGen) can be computed with the Haversine formula

$$d = 2 \cdot r \cdot \arcsin\left(\sqrt{\left(\sin^2\left(\frac{\Phi_2-\Phi_1}{2}\right) + \cos(\Phi_1)\cdot\cos(\Phi_2)\cdot\sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)\right)}\right)$$

The image (see figure 6.9) shows that the Richardson Iteration yields to the best result, because the algorithm has the closest distance to $\vec{x}_{org}$. If there is noise included the Richardson iteration, Landweber method, Kaczmarz Method and Richardson Regularization supply the same result in addition to different noise levels.

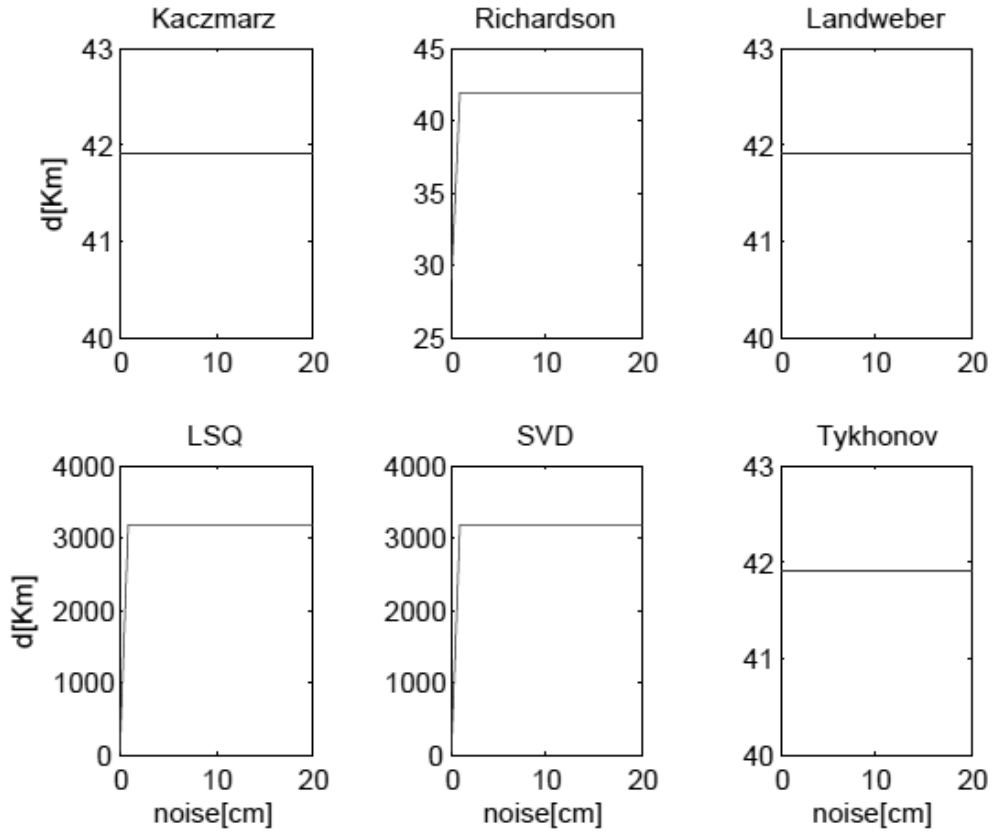

Fig 6.11: distance between the maximal slip values from the original slip distribution and computed slip distribution,x-axis shows the noise[cm] and y-axis shows the distance

6.4 Magnitude determination with RuptGen

Firstly, the slip distribution has to be computed (for example: Tykhonov Regularization). Afterwards a command has to be executed for every single subfault and stored in a file

$$\text{ruptgen.x} - \text{slip } 4.23 - \text{ij } 100\ 8\ .$$

So we have 3750 commands as subfaults and RuptGen can be started.

$$\text{ruptgen.x} - \text{f myslip.dat} - \text{output gps stations.dat}$$

The computed magnitude is stored in ruptgen.log and can be compared with the entered magnitude ($MW_o = 8.5$) MW_o . MW_c represents the computed magnitude by RuptGen with the slip distribution of the used algorithms.

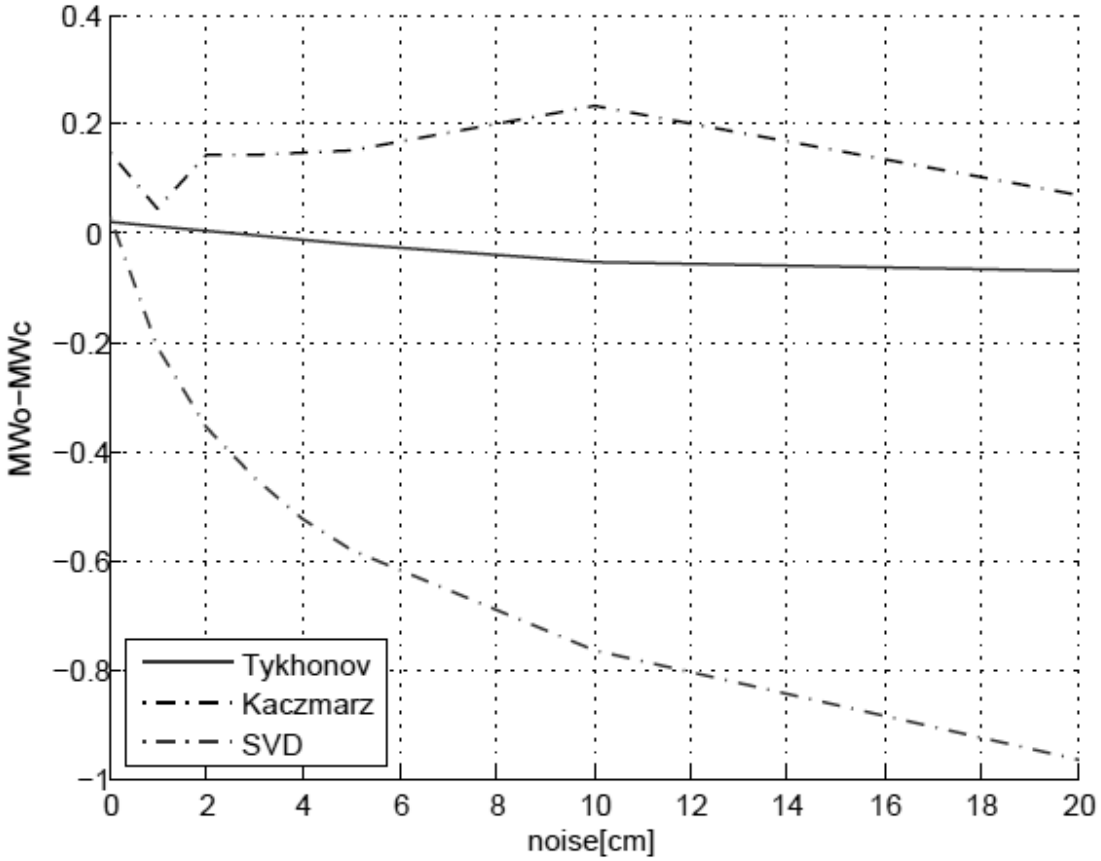

Figure 6.12: magnitude comparison: Tykhonov Regularization, Kaczmarz Method, SVD

The Tykhonov Regularization (see figure 7.1) provides the best result, because

$$-0.1 < (\mathrm{MW_o} - \mathrm{MW_c}) < 0.1 \ .$$

However, the SVD Method (see figure 7.1) and LSQ Method (see figure 7.2) give the worst result, because

$$-0.9 < (\mathrm{MW_o} - \mathrm{MW_c}) < 0.1 \ .$$

The Landweber Method is not suitable from a noise larger than 10cm, because the magnitude increases too fast. The scope is too large and the earthquake becomes underestimated, hence the algorithm is not suitable. The Kaczmarz Method and Modified Richardson Iteration are both decent algorithms.

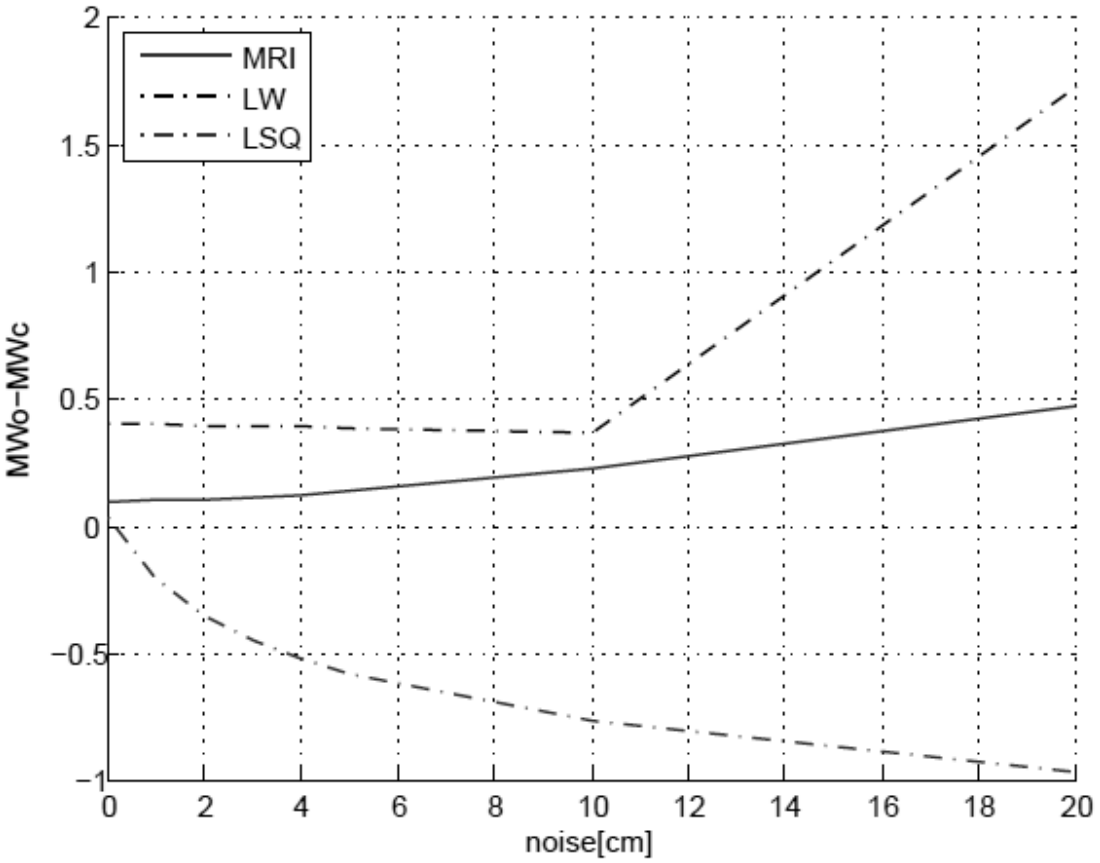

Fig. 6.13: magnitude comparison, Modified Richardson Iteration, Landweber, Least Squares

6.5 Sensor sensitivity as an auxiliary condition by the example of Tykhonov Regularization

It is important to know in which area a sensor can reconstruct (see figure 8.1). This information gives the opportunity to restrict the patches, which could save calculation time. The regions where the reconstruction is possible can be visualized. The six sensors overlay the most parts of the area. In figure 6.12 are shown six sensors and their displacement in x, y, z. The displacements are calculated with RuptGen. The following command

$$\text{ruptgen. x} - \text{slip 1} - \text{ij 1 1}$$

has to be executed on each subfault, but the i-index and j-index changes after every execution. The resultant displacements of each sensor are stored in a file and visualized in figure 6.12.

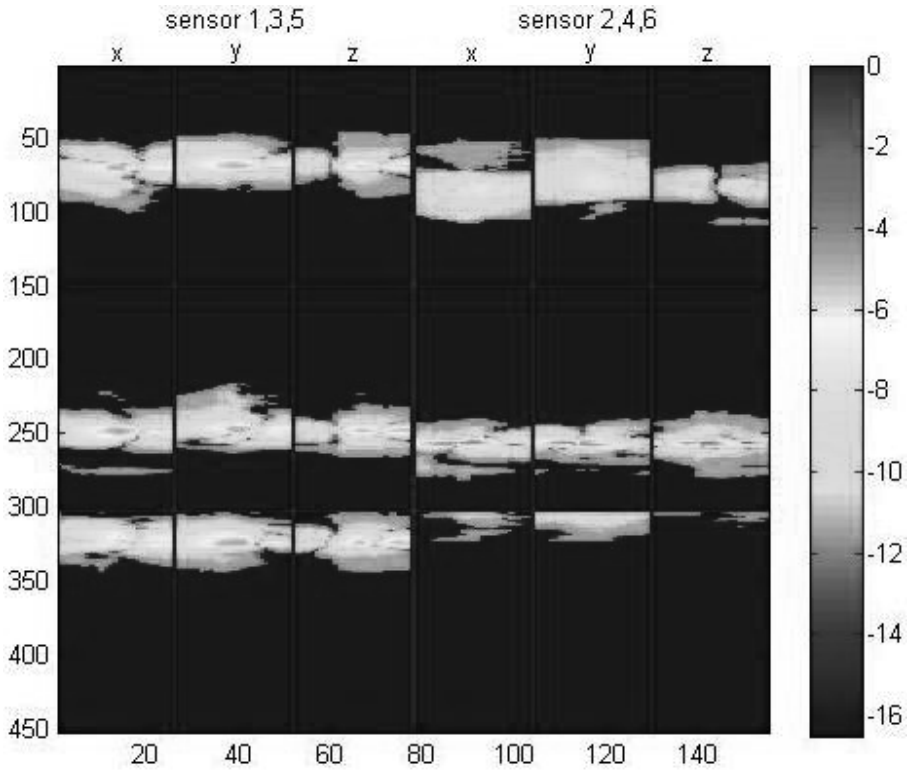

Fig. 6.14: sensitivity of 6 GPS sensors with x y and z, left sensors 1 3 5, right sensors 2 4 6, logarithmic scale, slip=1

The six sensors overlay a special region (see figure 8.2). The red regions are areas where the sensors are able to reconstruct the slip distribution. The blue regions are areas where none reconstruction is possible because of fewer sensors. One way is to add new sensors to overlay the whole area. The red color becomes murkier, because of the overlapping sensors. In such areas the reconstruction is more stable than in areas with a brighter red color.

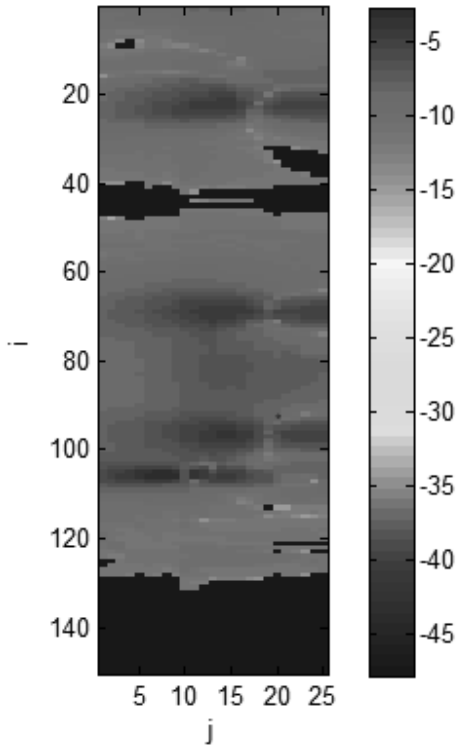

Fig. 6.15: overlaid area by six GPS sensors, logarithmic scale, represents matrix B

The matrix B represents the area which is visualized in figure 6.13. The aim is to eliminate the regions where no reconstruction is possible (blue regions). On the basis of equation 4.6 the matrix A can be restricted as follows and the equation leads to

$$B\cdot (A^T \cdot A \cdot + \Gamma \cdot I) \cdot \vec{x} = A^T \cdot \vec{b}\,.$$

Hence the above system of equations becomes smaller, less time and equations are required. The resulting slip distribution (see figure 6.14) is equal to figure 6.2. Thus it makes sense to restrict the matrix A.

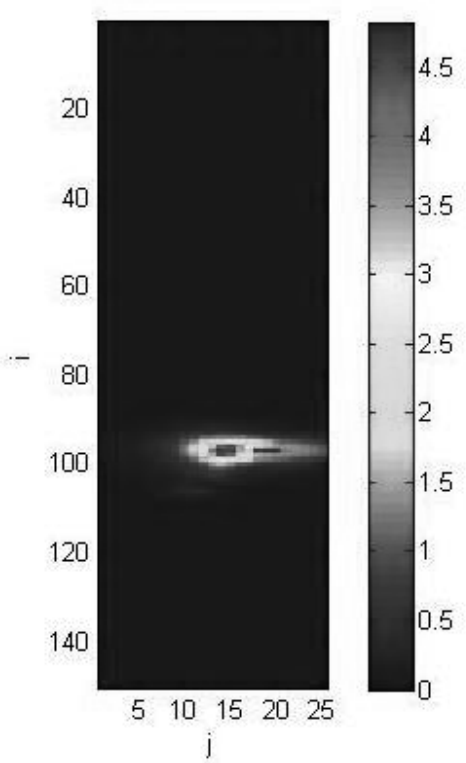

Fig. 6.16: slip distribution, Tykhonov Regularization with auxiliary condition (matrix B), noise level 0cm,

6.6 Auxiliary ellipse around the epicenter by the example of Tykhonov Regularization

The ellipse around the epicenter represents the area in which the tsunami genic fault can be expected for an equation of the given epicenter and magnitude. The patches outside the ellipse become penalized. Therefore the spray around the main faults can be reduced [1]. The improved Tykhonov Regularization (similar to 5.4.4, only B added) is given by

$$\frac{1}{2}\left\|A\cdot\vec{x}-\vec{b}\right\|^2+\frac{1}{2}\alpha\cdot\|B\cdot\vec{x}\|^2 .$$

The solution formula is given through

$$\vec{x}=(A^T\cdot A+\alpha\cdot B^T\cdot B)^{-1}\cdot A^T\cdot\vec{b} .$$

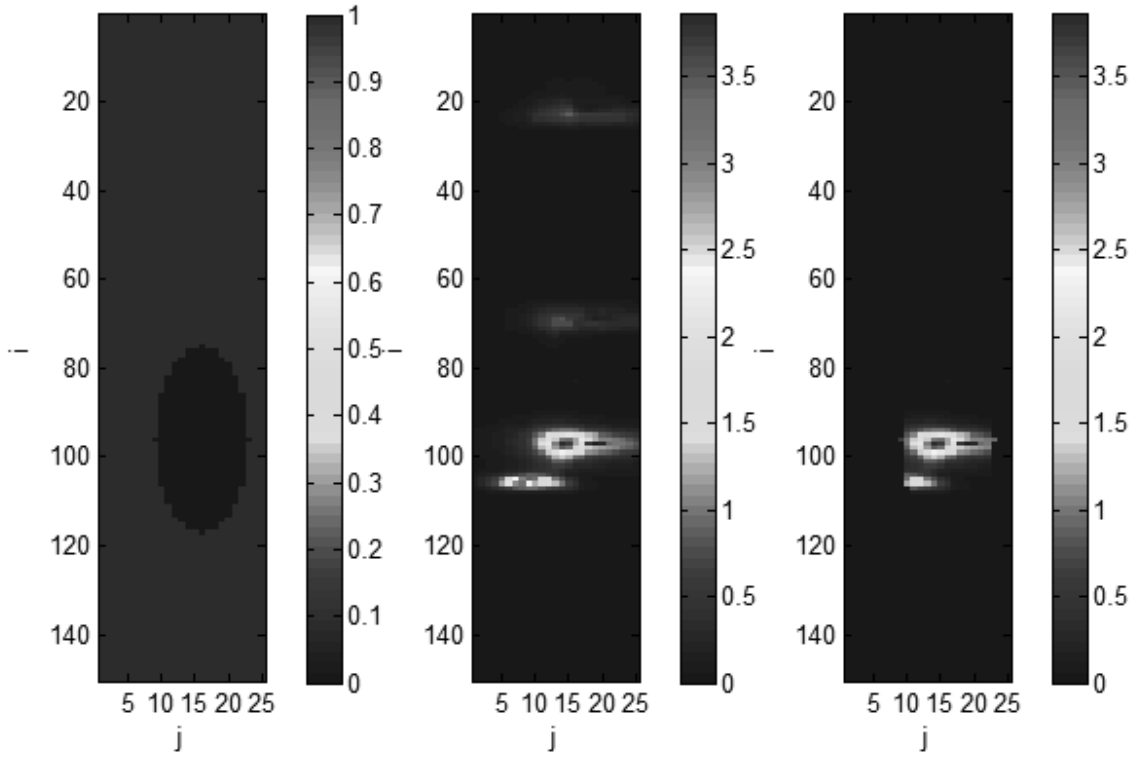

Fig. 6.17: left ellipse, centre: Tykhonov Regularization with noise level 20cm, right: Tykhonov Regularization in combination with ellipse

7 Conclusion and Outlook

There are many algorithms which could be used to solve the problem of earthquake parameter reconstruction, but only the Tykhonov Regularization is stable and gives acceptable results. These kinds of algorithms depend on the situation, if the system is underdetermined, over determined or the variables and equations are equal. In another situation the same method can give a better result or a worse result. The algorithm has to be chosen with care and must depend on the issue.

This thesis differs from Manuella Schönrocks thesis, because only six sensors are available as opposed to 61 sensors. The aim was to implement new algorithms and auxiliary conditions. The improvements in comparing with Manuelle Schönrocks thesis is that new auxiliary conditions are created (see chapter 6.5). Furthermore this thesis examined the accuracy of localization of an earthquake and determines the magnitude.

In the future it could be possible to combine seismic und GPS-inversion to locate the earthquake more exactly and get the slip distribution with more accuracy. In addition, adding more sensors can overlay the whole area and earthquakes could be reconstructed at each position.

In the field of early warning there are pioneering methods under development. The Scatterometry and Reflectometry allow the detection of reflected signals (electromagnetic waves), because of the high reflectivity signals on water, ice and snow. The past experience with special GPS receivers in planes and balloons told us that those measurements of the sea level can obtain an accuracy of up to 5cm.

Another possibility is the tsunami detection from space using GNSS Reflectometry. The GNSS-R uses given signals from navigation satellites (Global Navigation Satellite System), just like Radar. The Radar signals are reflecting from the sea surface and can be used to calculate the height of the reflecting surface [4].

A Appendix

A.1 Comparison of slip distribution

A.1.1 Tykhonov Regularization

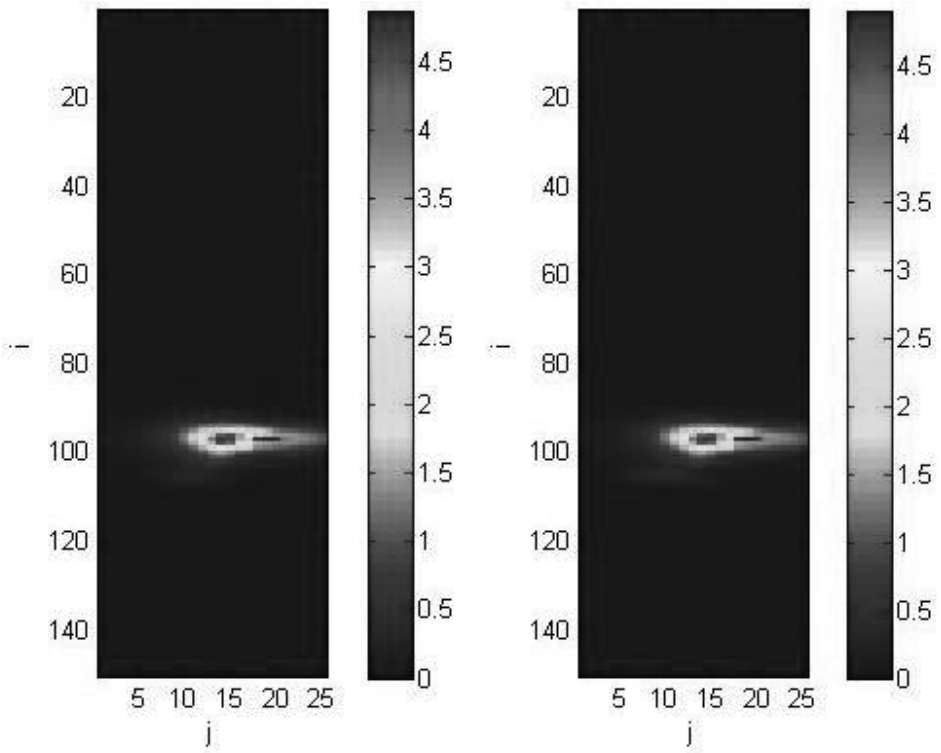

Fig. A.1: slip distribution, Tykhonov Regularization, noise level = 1cm (left), noise level = 2cm (right)

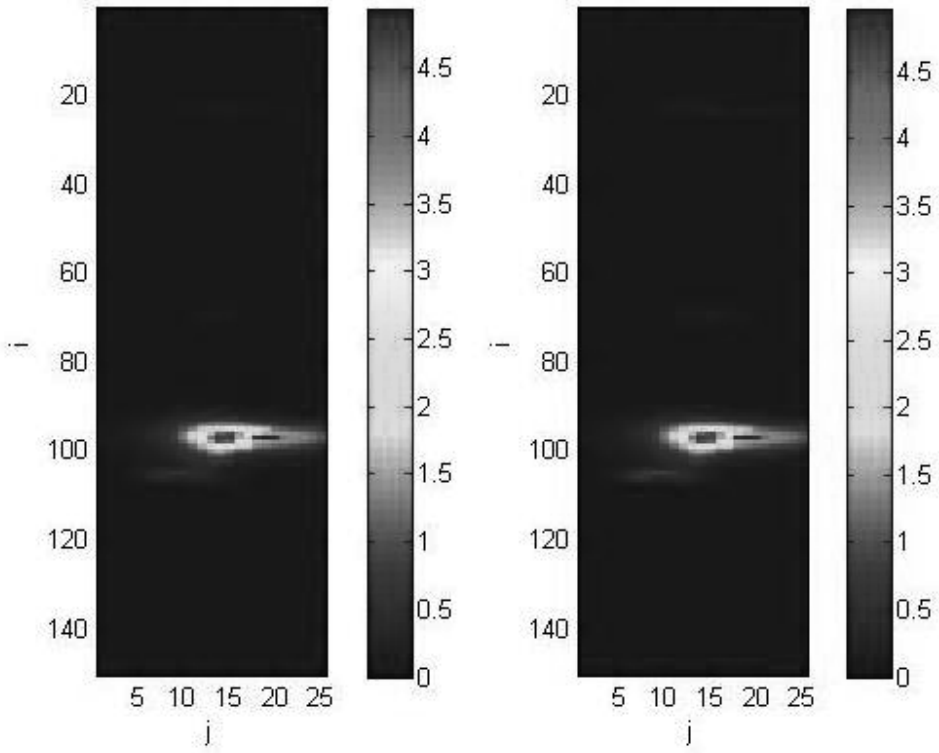

Fig A.2: slip distribution, Tykhonov Regularization, noise level = 3cm (left), noise level = 4cm (right)

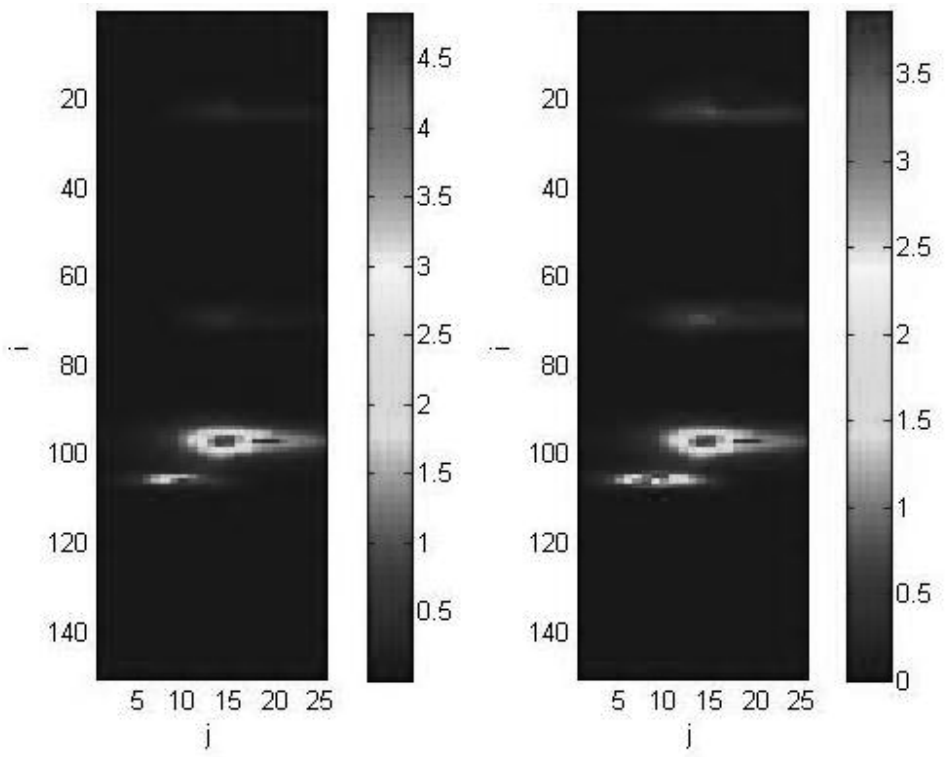

Fig. A.3: slip distribution, Tykhonov Regularization, noise level = 10cm (left), noise level = 20cm (right)

A.1.2 Kaczmarz Method

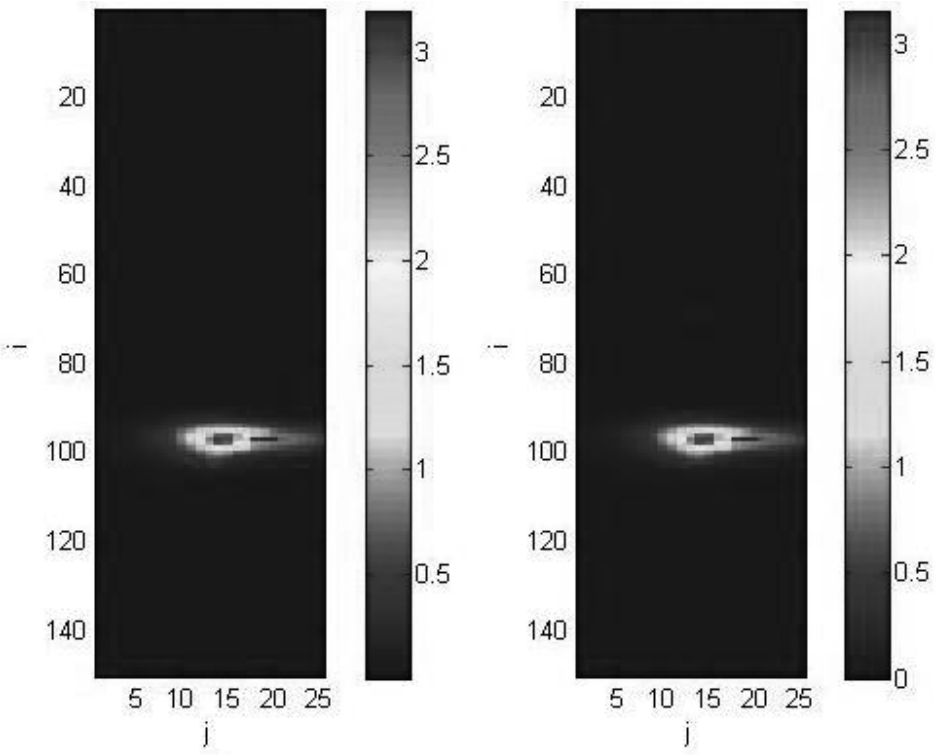

Fig. A.4: slip distribution, Kaczmarz Method, noise level = 1cm (left), noise level = 2cm (right)

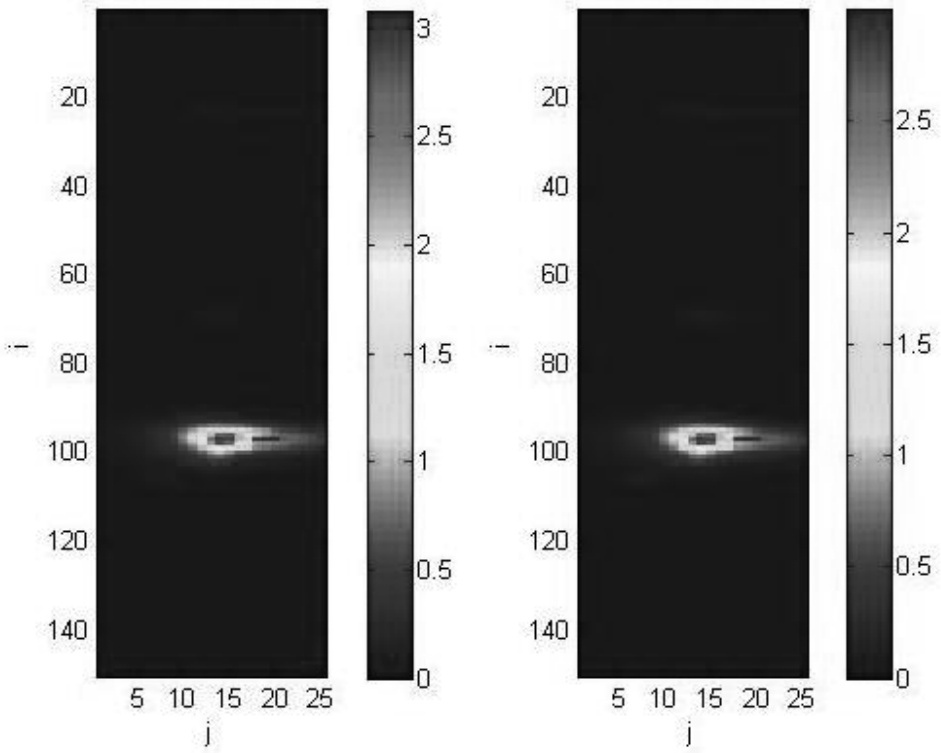

Fig. A.5: slip distribution, Kaczmarz Method, noise level = 3cm (left), noise level = 4cm (right)

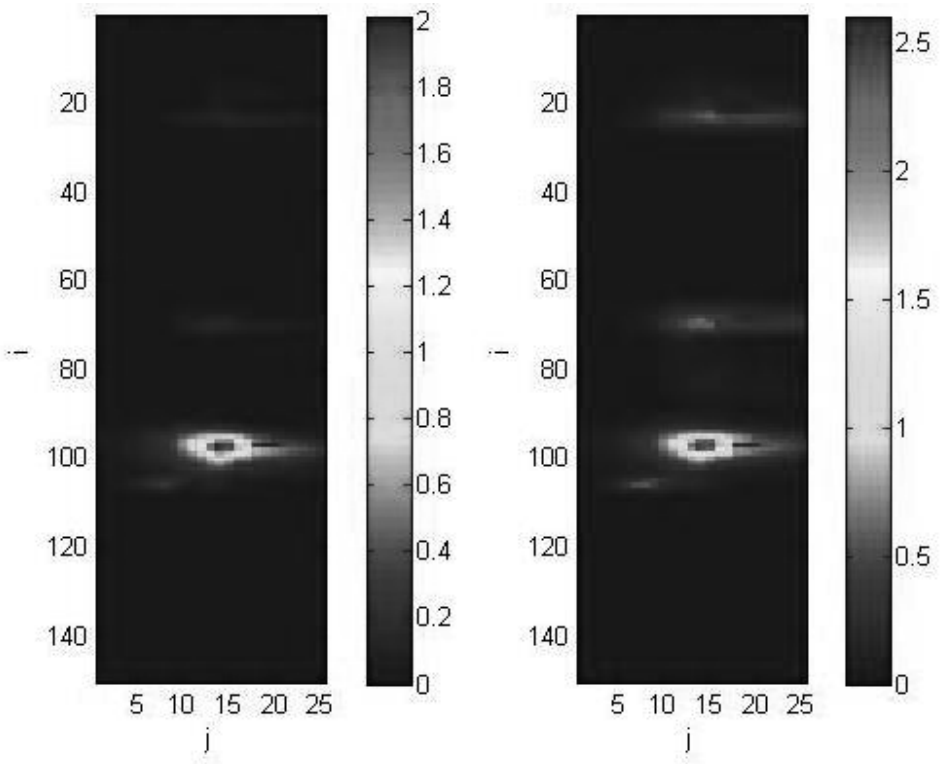

Fig. A.6: slip distribution, Kaczmarz Method, noise level = 10cm (left), noise level = 20cm (right)

A.1.3 Least norm solution via Singular Value Decomposition

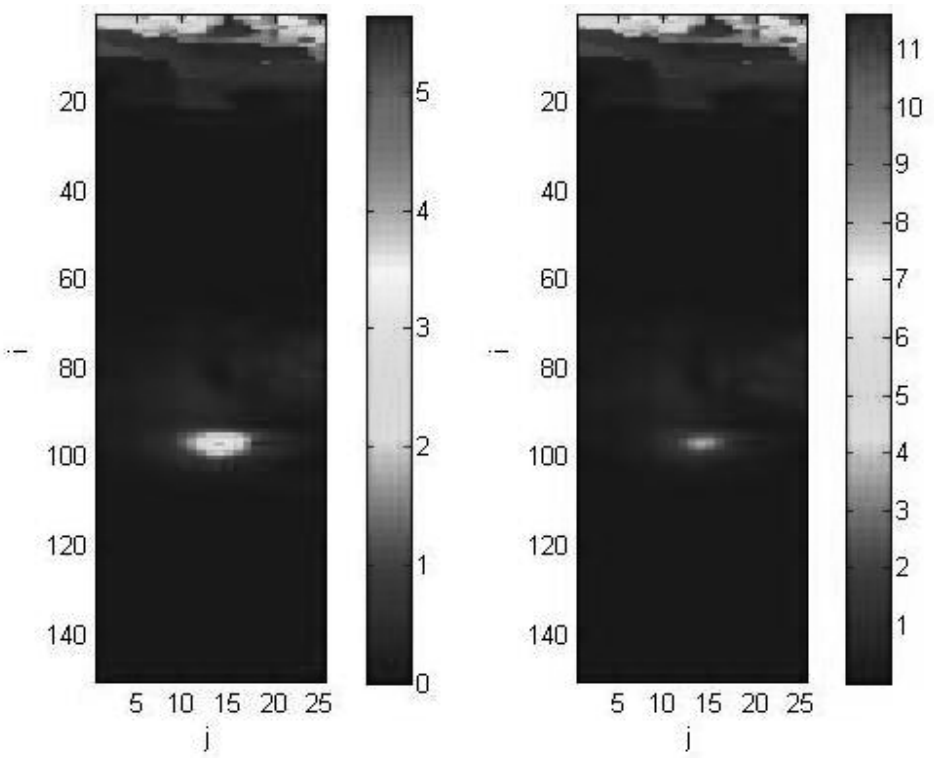

Fig. A.7: slip distribution, Least Norm Solution via Singular Value Decomposition, noise level = 1cm (left), noise level = 2cm (right)

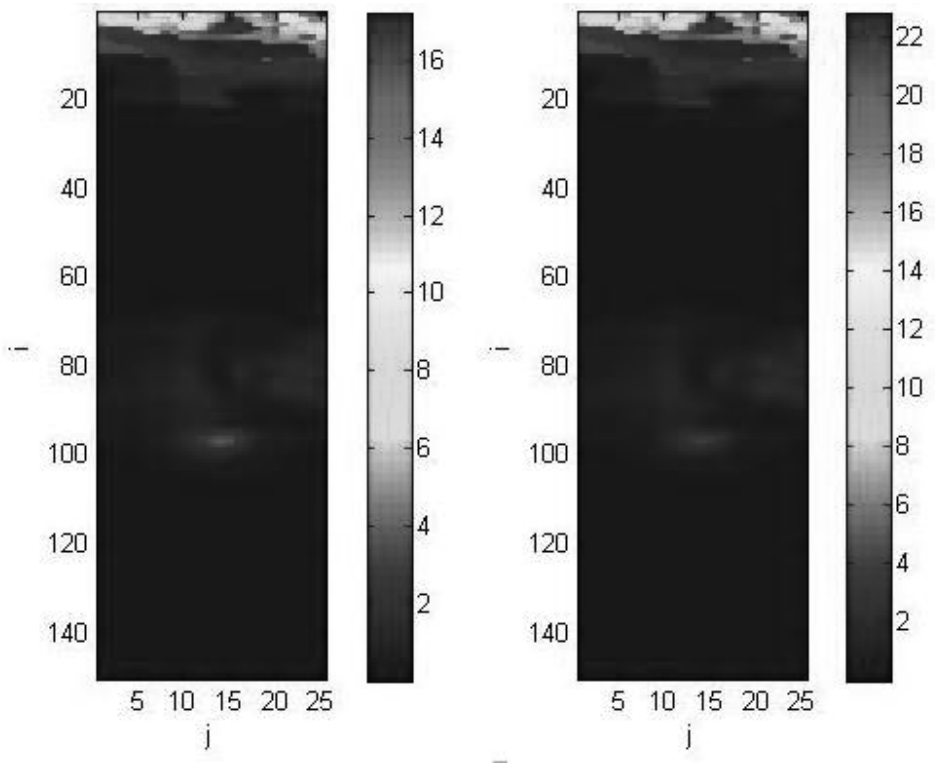

Fig. A.8: slip distribution, Least Norm Solution via Singular Value Decomposition, noise level = 3cm (left), noise level = 4cm (right)

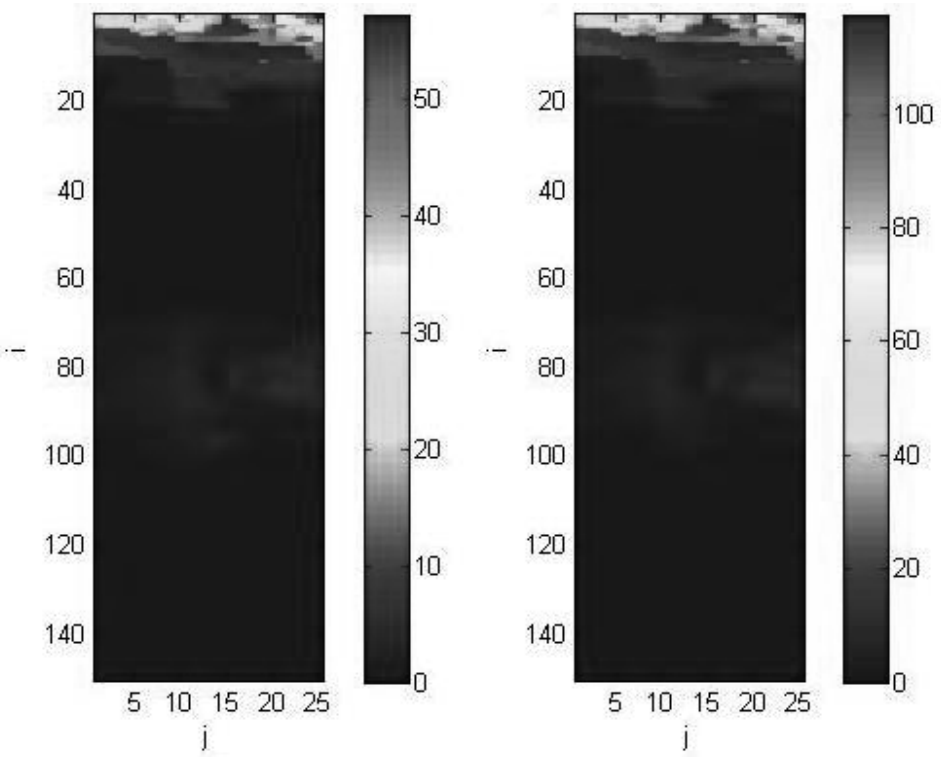

Fig. A.9: slip distribution, Least Norm Solution via Singular Value Decomposition, noise level = 10cm (left), noise level = 20cm (right)

A.1.4 Least squares via QR factorization

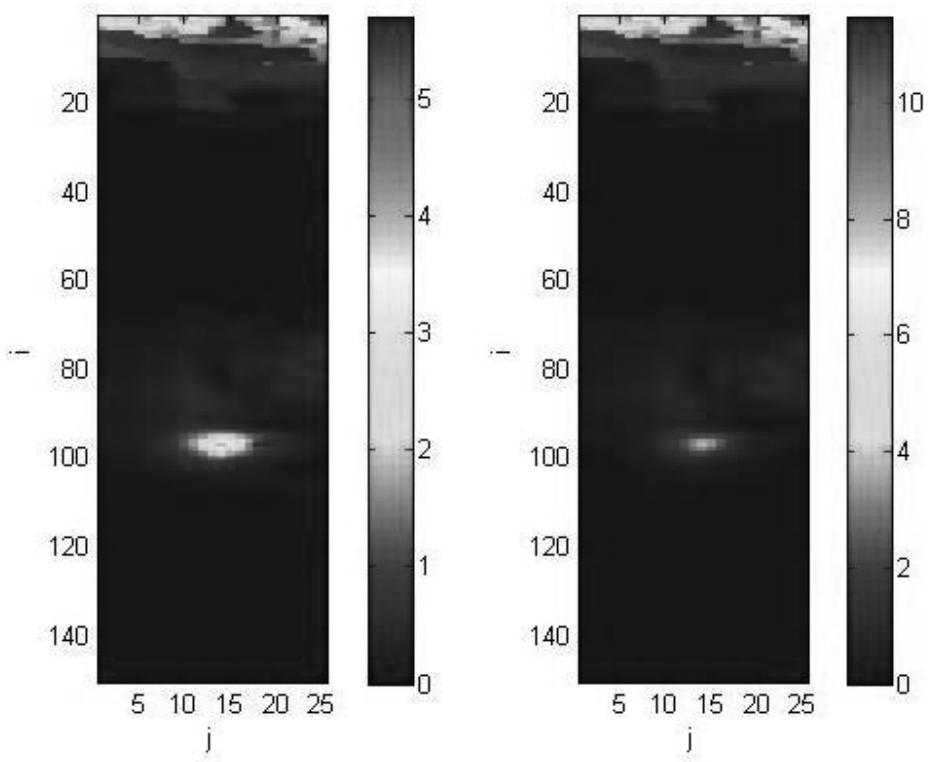

Fig. A.10: slip distribution, Least Squares via QR factorization, noise level = 1cm (left), noise level = 2cm (right)

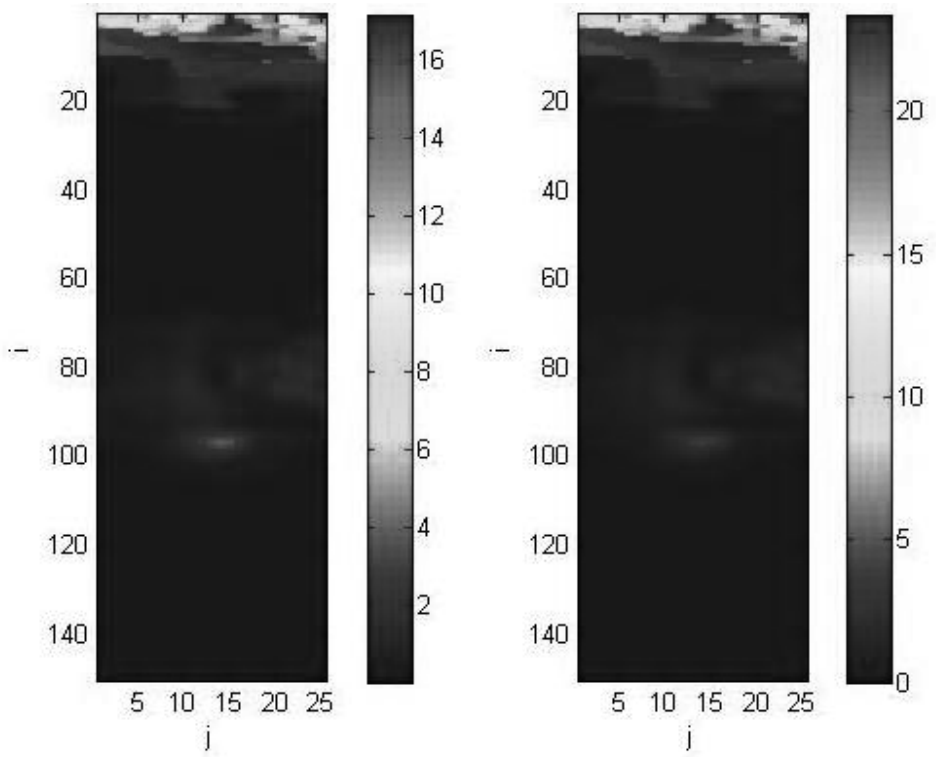

Fig. A.11: slip distribution, Least Squares via QR factorization, noise level = 3cm (left), noise level = 4cm (right)

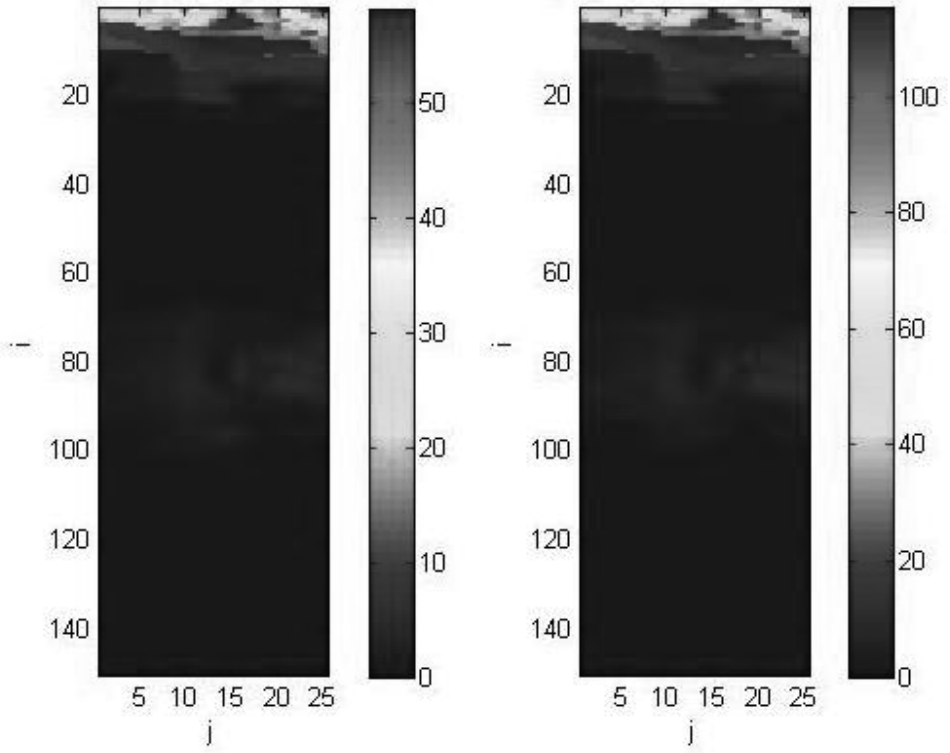

Fig. A.12: slip distribution, Least Squares via QR factorization, noise level = 10cm (left), noise level = 20cm (right)

A.1.5 Modified Richardson Iteration

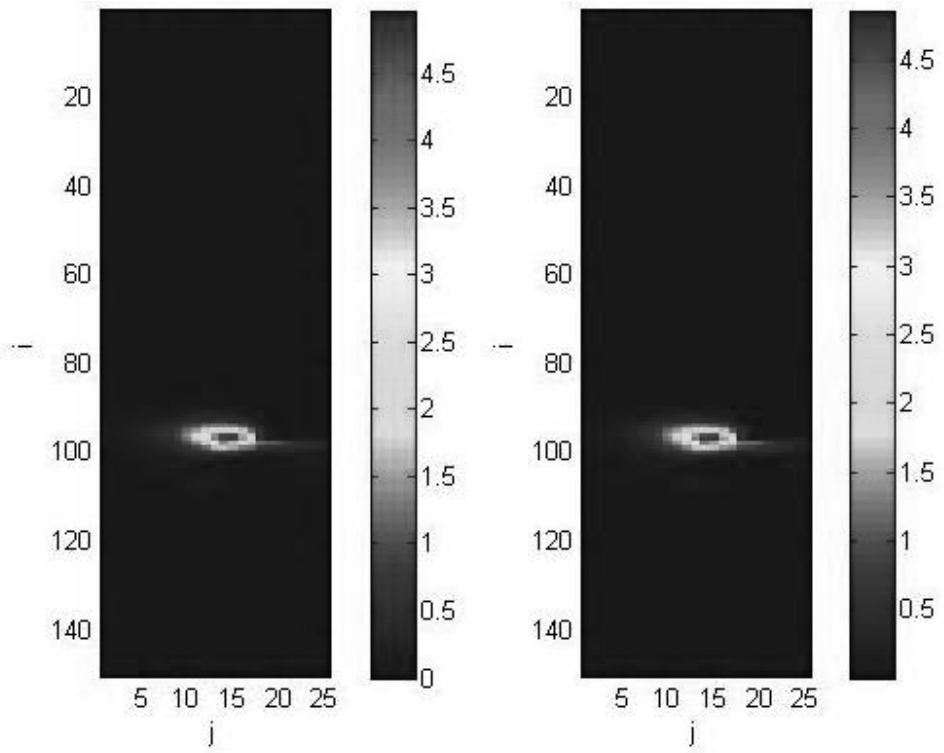

Fig. A.13: slip distribution, Modified Richardson Iteration, noise level = 1cm (left), noise level = 2cm (right)

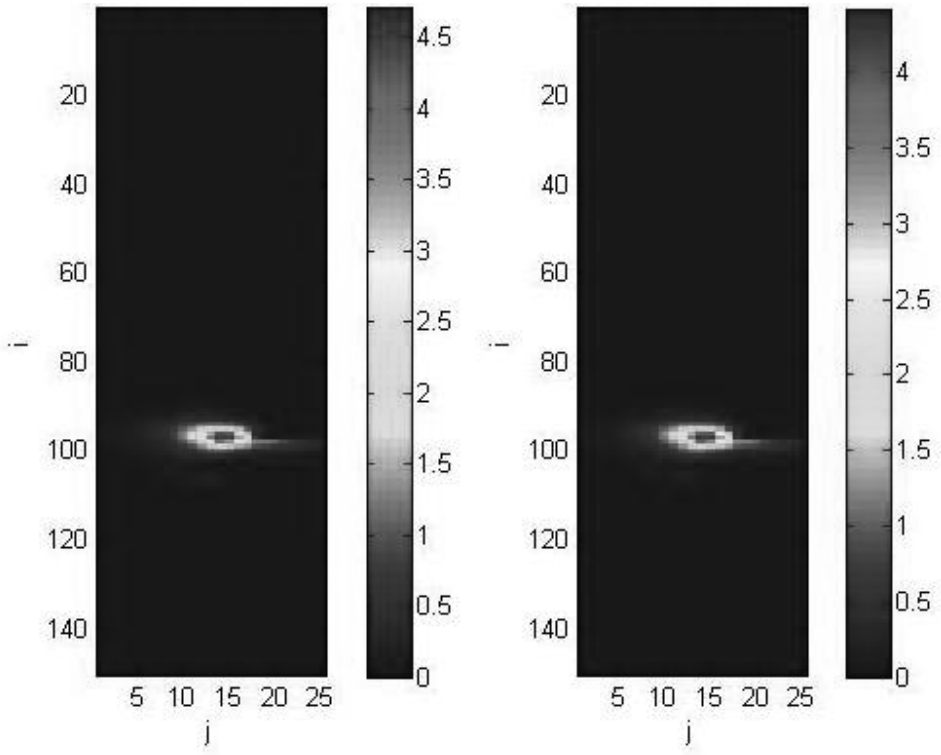

Fig. A.14: slip distribution, Modified Richardson Iteration, noise level = 3cm (left), noise level = 4cm (right)

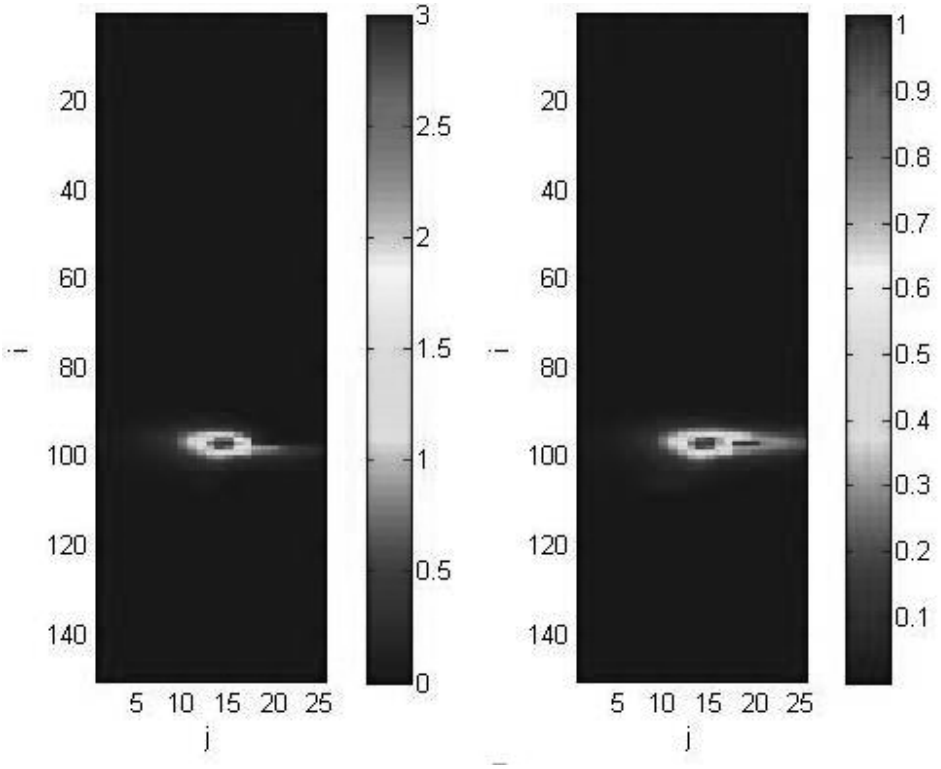

Fig. A.15: slip distribution, Modified Richardson Iteration, noise level = 10cm (left), noise level = 20cm (right)

A.1.6 Landweber Method

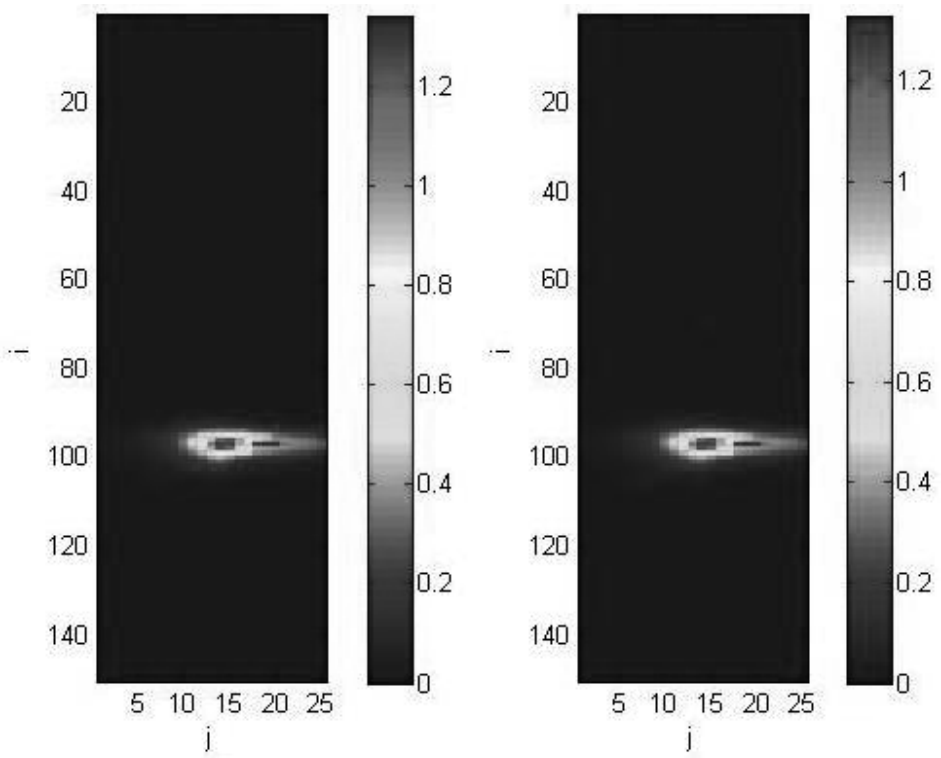

Fig. A.16: slip distribution, Landweber Method, noise level = 1cm (left), noise level = 2cm (right)

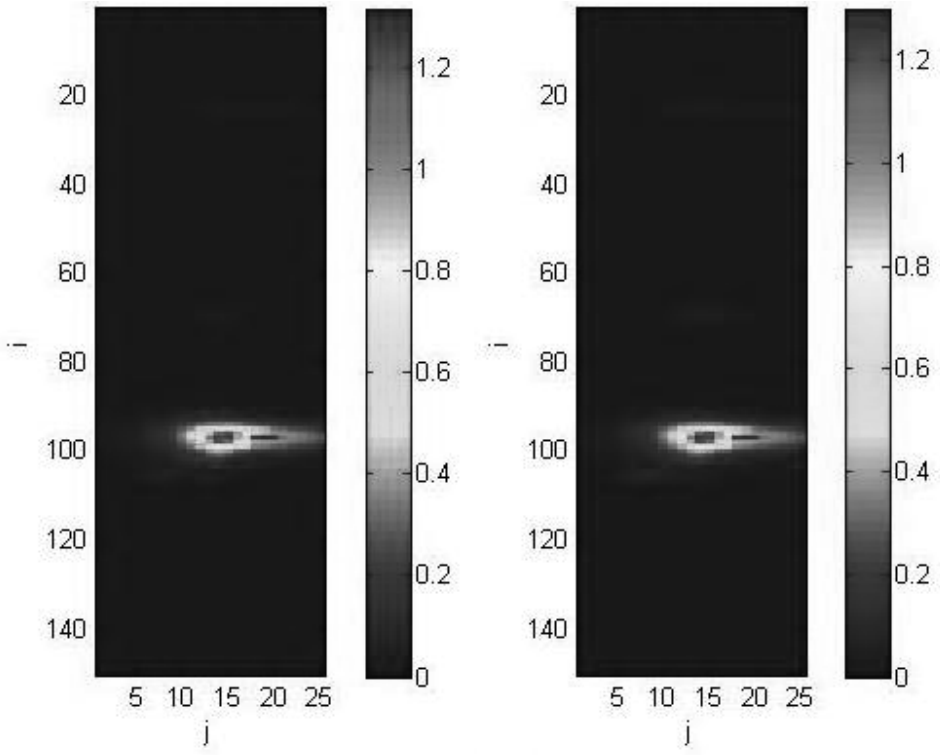

Fig. A.17: slip distribution, Landweber Method, noise level = 3cm (left), noise level = 4cm (right)

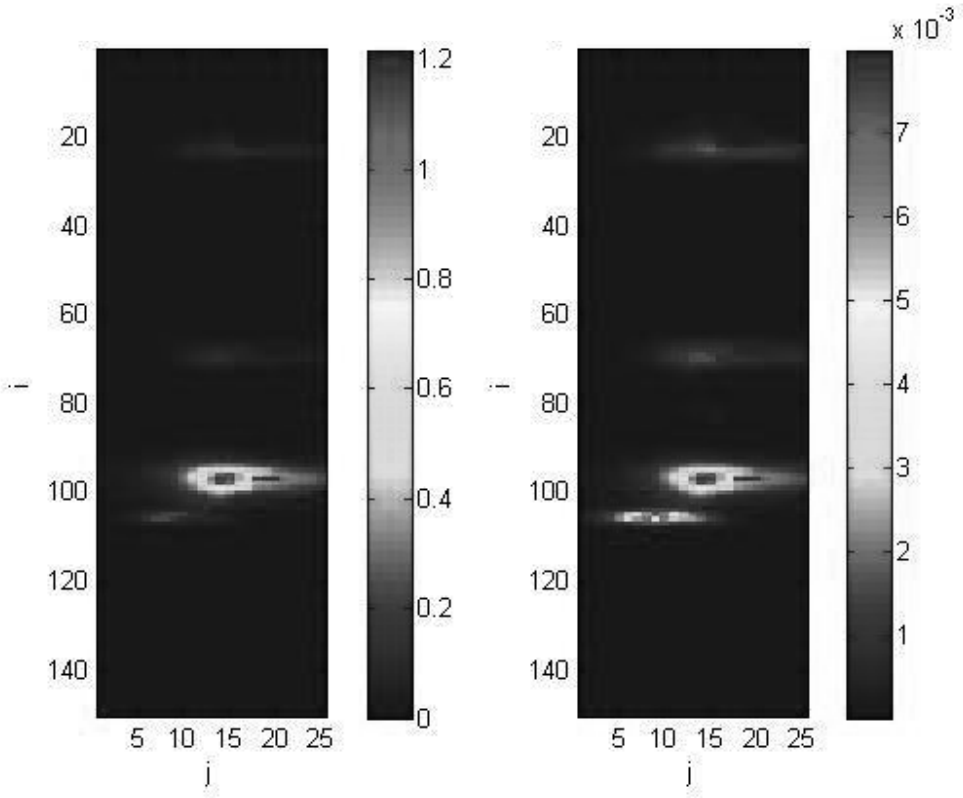

Fig. A.18: slip distribution, Landweber Method, noise level = 10cm (left), noise level = 20cm (right)

A.2 Source code (creation of matrix Astrike and Adip)

```
clear all;
clc;
Adx = zeros(6,25*150);
Ady=Adx;Adz=Adx;
slip1=1;

 for j=1:1:150

     for i=1:1:25

         unix(['ruptgen.x -slip ',int2str(slip1),'  -ij ',int2str(j),'
      ',int2str(i),' -rake 0 -output gps stations.dat']);
        F=load('stations.dat.out');
        n = (j-1)*25+i;
        [j i n]
        Adx(:,n,slip1)=F(:,3);
        Ady(:,n,slip1)=F(:,4);
        Adz(:,n,slip1)=F(:,5);

     end

 end

save Astrike.mat Adx Ady Adz
```

Figure A.21 creation of matrix A (strike slip)

```
clear all;
clc;
Adx = zeros(6,25*150);
Ady=Adx;Adz=Adx;
slip1=1;

 for j=1:1:150

     for i=1:1:25

         unix(['ruptgen.x -slip ',int2str(slip1),'  -ij ',int2str(j),'
      ',int2str(i),' -rake 90 -output gps stations.dat']);
        F=load('stations.dat.out');
        n = (j-1)*25+i;
        [j i n]
        Adx(:,n,slip1)=F(:,3);
        Ady(:,n,slip1)=F(:,4);
        Adz(:,n,slip1)=F(:,5);

     end
 end

save Adip.mat Adx Ady Adz
load Adip.mat
```

Figure A.22 creation of matrix A (dip slip)

A.3 Source code algorithms

A.3.1 Least Norm via Singular Value Decomposition

```
[U S V]=svd(A(1:18,:)); SVD decomposition
S1(1:18,1:18)=S(1:18,1:18);
V1(1:7500,1:Z)=V(1:7500,1:18);

SVD=abs(V1*inv(S1)*U(1:Z,1:Z)'*Y(1,1:Z)');

x=sqrt(SVD(1:3750,1).^2+SVD(3751:7500).^2); solution
```

A.3.2 Least squares via QR factorization

```
Q R]=qr(A(1:18,:)'); %QR factorization
QQ=Q(:,1:18);
RR=R(1:18,1:18);

LSQ=abs(QQ*inv(RR*RR')*(RR*Y'));

x=sqrt(LSQ(1:3750,1).^2+LSQ(3751:7500).^2); solution
```

A.3.3 Tykhonov Regularization via QR factorization

```
Q R]=qr(A(1:18,:)'); %QR factorization
QQ=Q(:,1:18);
RR=R(1:18,1:18);

I=eye(18);
L=0.84;

TRQR=abs(((RR*Y(1,1:18)')\(RR*RR'+L*I))*QQ');
TRQR=TRQR';

x=sqrt(TRQR(1:3750,1).^2+TRQR(3751:7500).^2);
```

A.3.4 Kaczmarz Method

```
L=[1 1 1]*24;
k=0;
x0=zeros(7500,1);
x=x0;
tt=1;

 for j=1:1:33
    for i=1:1:3
        if norm(A(k+1:6*i,:)*x0-Y(1,k+1:6*i)')>((tt*EF1(i,1)))
   Kms=(x0-(L(i)*A(k+1:6*i,:)'*(A(k+1:6*i,:)*x0-    Y(1,k+1:6*i)')));
          x0=Kms;
        end
     k=6*i;
    end
  slip(:,f)=sqrt(Kms(1:3750,1).^2+Kms(3751:7500).^2);
  k=0;
 end
```

A.3.5 Modified Richardson Iteration

```
AA=A(1:Z,:)'*A(1:Z,:);
YY=A(1:Z,:)'*Y(1,1:Z)';
I=eye(size(AA));
ew=(eig(A'*A));
gamma=2/((max((ew)))+min((ew)));
m=I-gamma*AA;
n=gamma*YY;

   for i=1:1:100
        if norm(A(1:Z,:)*x0-Y(1,1:Z)')>EFtest
           MRI=abs((m*x0+n));
        end

     x0=MRI;
   end

x=sqrt(MRI(1:3750,1).^2+MRI(3751:7500).^2);
```

A.3.6 Landweber Iteration

```
I=eye(7500);
alpha=2/(max(sum(abs(A'))));

LW=zeros(7500,1);
x0=zeros(7500,1);

 for i=1:1:100

    if norm(A(1:Z,:)*x0-Y(1,1:Z)')>EFtest
       LW=x0+alpha*A(1:18,:)'*(Y(1,1:18)'-A(1:18,:)*x0);
       x0=LW;
    end

 end

x=sqrt(LW(1:3750,1).^2+LW(3751:7500).^2);
```

B References

1. Babeyko, A, Y. Hoechner S. V. Sobolev (2010): Source modeling and inversion with near real-time GPS. a GITEWS perspective for Indonesia. Hg. v. GFZ. Available from http://www.nat-hazards-earth-syst-sci.net/10/1617/2010/nhess-10-1617-2010.pdf, [Accessed 3 April 2013] 1

2. Baier, M. GITEWS. Concept. Available from http://www.gitews.org/index.php?id=22&L=1, [Accessed 20 January 2013] 2

3. Baier, M. GITEWS. GPS Real-Time Reference Stations. Available from http://www.gitews.org/index.php?id=80&L=1, [Accessed 7 January 2013] 3

4. Baier, M. GITEWS. GPS-Reflektometrie. Available from http://www.gitews.de/index.php?id=21&L=0 [Accessed April 2013] 4

5. Boyd, S. (2007). QR Factorization. Available from http://see.stanford.edu/materials/lsoeldsee263/04-qr.pdf [Accessed 1 December 2012] 5

6. Boyd, S. (2007). SVD Decomposition. Available from http://see.stanford.edu/materials/lsoeldsee263/16-svd.pdf [Accessed 1 December 2012] 6

7. Boyd, Stephen. Least squares. Available from http://see.stanford.edu/materials/lsoeldsee263/05-ls.pdf [Accessed 10 November 2012] 7

8. Boyd, S. Least Norm Solution via QR factorization. Available from http://see.stanford.edu/materials/lsoeldsee263/08-min-norm.pdf [Accessed 12 November 2012] 8

9. German Research Centre for Geosciences (GFZ). 2007. The German Contribution to the Tsunami Early Warning System for the Indian Ocean. Potzdam. German Research Centre for Geosiences 9

10. Haltmeier, M., Leitao, A., Scherzer, S., 2007. Kaczmarz Methods for regularizing nonlinear ill-posed equations. Innsbruck Available from: http://www.mtm.ufsc.br/~aleitao/public/reprints/pap2007-hls-IPI.pdf [Accessed 10 November 2012] 10

11. Höchner, A. (2010): GPS based analysis of earthquake induced phenomena at the Sunda Arc - hoechner_diss.pdf. Available from http://opus.kobv.de/ubp/volltexte/2011/5316/pdf/hoechner_diss.pdf [Accessed 10 April 2013] 22

12. Louis, A.,K. (1989). Inverse und schlecht gestellte Probleme. Stuttgart: B.G. Teubner 11

13. Levin, B., Nosov, M. (2009): Physics of tsunamis. Dordrecht, London: Springer 12

14. Okada Y. (1992): Okada92.pdf. Available from http://www.geophysik.uni-muenchen.de/~malservisi/TECTOGPS/papers/Okada92.pdf [Accessed 15 April 2012] 21

15. Schönrock, M. (2009). GPS-based earthquake parameter reconstruction within the German Indonesian Tsunami early Warning System. Neubrandenburg. Available from http://digibib.hs-nb.de/file/dbhsnb_derivate_0000000204/Diplomarbeit-Schoenrock-2009.pdf, [Accessed 15 December 2013] 13

16. NOAA Tsunami Website (2013). Available from http://www.tsunami.noaa.gov/ [Accessed 10 April 2013] 23

17. Wikimedia foundation Inc. Confidence interval. Available from: http://en.wikipedia.org/w/index.php?oldid=548738635 [Accessed 4 April 2013] 14

18. Wikimedia foundation Inc. Earthquake. Available from: http://en.wikipedia.org/wiki/Earthquake [Accessed 8 January 2013] 15

19. Wikimedia foundation Inc. Richardson Iteration. Available from: http://de.wikipedia.org/wiki/Richardson-Verfahren [Accessed 20 November 2012] 16

20. Wikimedia foundation Inc. inverse problem. Available from: http://de.wikipedia.org/wiki/Inverses_Problem [Accessed 11 November 2012] 17

21. Wikimedia foundation Inc. Sunda arc. Available from: http://de.wikipedia.org/wiki/Sundabogen [Accessed 8 January 2013] 18

22. Wikimedia foundation Inc. Sunda trench. Available from: http://en.wikipedia.org/wiki/Sunda_Trench [Accessed 23 November 2012] 19

23. Wikimedia foundation Inc: Tikhonov regularization. Available from http://en.wikipedia.org/w/index.php?oldid=545391055 [Accessed 12 November 2012] 20

C List of Figures