3D affine coordinate transformations

Constantin-Octavian Andrei

Master's of Science Thesis in Geodesy No. 3091
TRITA-GIT EX 06-004

School of Architecture and the Built Environment
Royal Institute of Technology (KTH)
100 44 Stockholm, Sweden

March 2006

Abstract

This thesis investigates the three-dimensional (3D) coordinate transformation from a global geocentric coordinate system to a national terrestrial coordinate system. Numerical studies are carried out using the Swedish geodetic data SWEREF 93 and RT90/RH70. Based on the Helmert transformation model with 7-parameters, two new models have been studied: firstly a general 3D affine transformation model has been developed using 9-parameters (three translations, three rotations and three scale factors) and secondly the model with 8-parameters (three translations, three rotations and two scale factors) has been derived. To estimate the 3D transformation parameters from given coordinates in the two systems, the linearized observation equations were derived. Numerical tests were carried out using a local (North, East, Up) topocentric coordinate system derived from the given global geocentric system. The transformation parameters and the residuals of the coordinates of the common points were computed. The investigation shows the horizontal scale factor is significantly different by the vertical scale factor. The residuals of the control points were expressed in a separate (North, East, Up) coordinate system for each control point. Some investigations on the weighting process between horizontal and vertical components were also carried out, and an optimal weighting model was derived in order to reduce the residuals in horizontal components without changing the coordinates.

Keywords: Affine transformation ▪ scale factor ▪ translation ▪ rotation ▪ least squares adjustment ▪ weight ▪ residual.

Acknowledgements

I wish to express my gratitude to all who have contributed to the completion of this master thesis.

First and foremost, I like to express my sincerest gratitude to my supervisor and the director of the master's programme, Huaan Fan, for his guidance and encouragement during the thesis work. I am indebted to him for all the discussions and his valuable comments (not only regarding the thesis).

I am grateful to the *Swedish Institute* for the financial support they made available for my graduate studies. This degree could not have been initiated and completed without this support which played a crucial role by allowing me to concentrate my full effort on the studies.

I also would like to extend my gratitude to Professor Gheorghe Nistor, Technical University of Iasi, Romania, for all the encouraging guidance during both the student period and as his assistant and for initializing my first contact with Huaan Fan.

I would like to express my gratitude to Professor Lars E. Sjöberg, Division of Geodesy, KTH, my examinator of this Master of Science Thesis.

I finally wish to thank all my master's colleagues for their friendship and a pleasant time during my stay at KTH.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

A coordinate transformation is a mathematical operation which takes the coordinates of a point in one coordinate system into the coordinates of the same point in a second coordinate system. Hopefully, there should also exist, an inverse transformation to get back to the first coordinate system from the given coordinates in the second one. Many types of mathematical operations are used to accomplish this task.

Coordinate transformations are widely used in geodesy, surveying, photogrammetry and related professions. For instance, in geodesy three-dimensional (3D) transformations are used to convert coordinates related to the Swedish national reference frame RT90 to the new reference frame SWEREF [Jivall 2001]. In photogrammetry they are used in the interior and exterior orientation of aerial photographs [Mikhail et al. 2001], and in surveying engineering they form part of the monitoring and control systems used in large manufacturing projects, as the construction of the ANZAC frigates for the Australian and New Zealand Navies [Bellman and Anderson 1995]. In the two-dimensional (2D) form, transformations are used, for example, in the cadastral surveys to re-establishment [Leu et al 2003] or match cadastral maps [Chen et al. 2000].

In general, the effect of a transformation on a 2D or 3D object will vary from a simple change of location and orientation, with no change in shape or size, to a uniform change scale factor (no change in shape), to changes of the shape and size of different degrees of nonlinearity [Mikhail 1976].

1.1. Affine transformations

The most general transformation model is the *affine transformation*, where changes in position, size and shape of a network are allowed. The scale factor of such a transformation depends on the orientation but not on the position within the net. Hence the lengths of all lines in a certain direction are multiplied by the same scalar.

3D affine transformations have been widely used in computer vision and particularly, in the area of model-based object recognition, and they can have involved different number of parameters involved:

- 12-parameter affine transformation (3D translation, 3D rotation, different scale factor along each axis and 3D skew) used to define relationship between two 3D image volumes. For instance, in medical image computing, the transformation model is part of different software programs that compute fully automatically the spatial transformation that maps points in one 3D image volume into their geometrically corresponding points in another, related 3D image volume [Maes et al. 1997].

- 9-parameter affine transformation (three translations, three rotations, three scales), can be used in reconstructing the relief and evaluating the geometric features of the original documentation of the cultural heritage by 3D modelling [Niederöst 2001].

- 8-parameter affine transformation (two translations, three rotations, two scale factors and skew distortion within image space) to describe a model that transform 3D object space to 2D image space [Fraser 2003].

1.2. Similarity transformations

A transformation in which the scale factor is the same in all directions is called a similarity transformation. A similarity transformation preserves shape, so angles will not change, but the lengths of lines and the position of points may change. An orthogonal transformation is a similarity transformation in which the scale factor is unity. In this case the angles and distances within the network will not change, but the positions of points do change on transformation.

1.2.1 Bursa-Wolf model

One of the most commonly used transformation methods in the geodetic applications is the *3D conformal transformation* also known as *3D similarity transformation* or *Helmert* *transformation* or *7- parameter transformation*, which preserves shape, so angles are not changed, but lengths of lines and the position of points may be changed.

When this transformation is applied to a terrestrial reference frame, it has the effect of rotating and translating the network of points with respect to the Cartesian axes. Finally, by applying an overall scale factor, the transformed network is obtained, while the shape of the figure remains unchanged. Equivalently, one can think of a network of points remaining still while the Cartesian axes are rotated translated and rescaled.

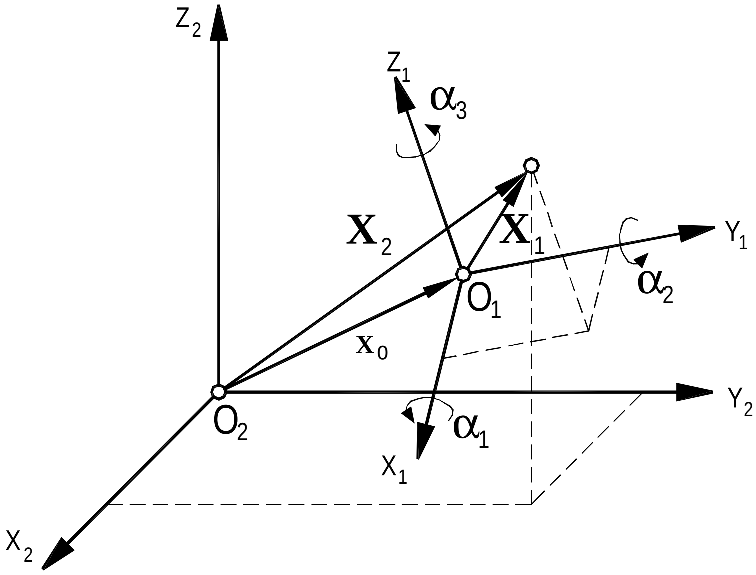

Figure 1.1: Three-dimensional transformation

Two data sets of three-dimensional rectangular coordinates defined in two different coordinate systems $\mathbf{X}_1$ respectively $\mathbf{X}_2$ (Figure 2.1), can be related to each other by the well known Bursa-Wolf formula for three-dimensional Helmert transformation:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{(2)} = \begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix} + \mu \cdot R(\alpha_1, \alpha_2, \alpha_3) \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{(1)} \tag{1.1}$$

where:

$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{(1)}$ denote the coordinates of point *i* in the first coordinate system;

$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{(2)}$ denote the coordinates of point *i* in the second coordinate system;

$\begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix}$ denote the three translations parameters;

$\alpha_1, \alpha_2, \alpha_3$ denote the three rotation angles around the *x*-, *y*- and *z*-axis, respectively;

μ denote the scale factor;

R denote the total rotation matrix which is the product of three individual rotation matrices:

$$\underset{3\times3}{\mathbf{R}} = \mathbf{R}(\alpha_1, \alpha_2, \alpha_3) = \mathbf{R}_3(\alpha_3) \cdot \mathbf{R}_2(\alpha_2) \cdot \mathbf{R}_1(\alpha_1)$$

$$= \begin{bmatrix} \cos\alpha_3 & \sin\alpha_3 & 0 \\ -\sin\alpha_3 & \cos\alpha_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\alpha_2 & 0 & -\sin\alpha_2 \\ 0 & 1 & 0 \\ \sin\alpha_2 & 0 & \cos\alpha_2 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha_1 & \sin\alpha_1 \\ 0 & -\sin\alpha_1 & \cos\alpha_1 \end{bmatrix}$$

(1.2)

$$= \begin{bmatrix} \cos\alpha_2\cos\alpha_3 & \cos\alpha_1\sin\alpha_3 + \sin\alpha_1\sin\alpha_2\cos\alpha_3 & \sin\alpha_1\sin\alpha_3 - \cos\alpha_1\sin\alpha_2\cos\alpha_3 \\ -\cos\alpha_2\sin\alpha_3 & \cos\alpha_1\cos\alpha_3 - \sin\alpha_1\sin\alpha_2\sin\alpha_3 & \sin\alpha_1\cos\alpha_3 + \cos\alpha_1\sin\alpha_2\sin\alpha_3 \\ \sin\alpha_2 & -\sin\alpha_1\cos\alpha_2 & \cos\alpha_1\cos\alpha_2 \end{bmatrix}$$

For the sake of simplicity, **R** can be written as

$$\underset{3\times3}{\mathbf{R}} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (1.3)$$

where all the elements r_{ij} $(i, j = 1,2,3)$ are functions of the rotation angles $\alpha_1, \alpha_2, \alpha_3$.

The usual mathematical form of the transformation is a linear formula which assumes that the rotation parameters are small. Rotations parameters between geodetic Cartesian systems are usually around 5-10 arc seconds, because the axes are conventionally aligned to the

Greenwich Meridian and the Pole. From these reasons Eq.(1.1) can be approximated by the following matrix equation:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{(2)} = \begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix} + \begin{bmatrix} 1+\delta\mu & \alpha_3 & -\alpha_2 \\ -\alpha_3 & 1+\delta\mu & \alpha_1 \\ \alpha_2 & -\alpha_1 & 1+\delta\mu \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{(1)} \tag{1.4}$$

where the translations along the x–, y– and z– axes, respectively are in *metres*; the rotations about the x–, y– and z– axes, respectively, are in *radians* and the scale factor change (unitless) is often stated in *parts per million* (ppm). Rotations are often given in arc seconds, which must be converted to radians.

The similarity transformation is popular due to:

- The small number of parameters involved
- The simplicity of the model, which is more easily implemented into software,
- The fact that it is adequate for relating two coordinates systems in the case when they are homogenous (no local distortion in scale or orientation).

In practice, the seven Helmert transformation parameters are not always known. Most often, they need to be estimated from some control points at which coordinates in the two coordinate systems are given. Theoretically, common coordinates at 3 points are sufficient for the solution of the 7-parameters transformation. If more points are known, a least squares adjustment can be performed to reduce the effect of errors in the given coordinates [Mikhail 1976, Fan 1997].

1.2.2 Molodensky-Badekas model

One problem with Bursa-Wolf model is that the adjusted parameters are highly correlated when the network of points used to determine the parameters covers only a small portion of the earth. The Molodensky-Badekas model [Badekas 1969] removes the high correlation between parameters by relating the parameters to the centroid of the network.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{(2)} = \begin{bmatrix} \overline{X} \\ \overline{Y} \\ \overline{Z} \end{bmatrix}_{(1)} + \begin{bmatrix} \delta x' \\ \delta y' \\ \delta z' \end{bmatrix} + \mu \cdot R \cdot \begin{bmatrix} X - \overline{X} \\ Y - \overline{Y} \\ Z - \overline{Z} \end{bmatrix}_{(1)} \tag{1.5}$$

where:

$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ centroid X coordinates for the points in the first coordinate system;

$\overline{Y} = \frac{1}{n}\sum_{i=1}^{n} Y_i$ centroid Y coordinates for the points in the first coordinate system;

$\overline{Z} = \frac{1}{n}\sum_{i=1}^{n} Z_i$ centroid Z coordinates for the points in the first coordinate system;

$\begin{bmatrix} \delta x' \\ \delta y' \\ \delta z' \end{bmatrix}$ Molodensky-Badekas translations;

remaining terms are as defined for the Bursa-Wolf model.

The adjusted coordinates, baseline lengths, scale factor, rotation angles, their variance-covariance matrices and the *a posteriori* variance factor computed by this model are the same as those from the corresponding Bursa-Wolf solution. However, the translations are different and their precisions are generally an order of magnitude smaller [Harvey, 1986]. The difference between the translation terms of Bursa-Wolf and Molodensky-Badekas models is due to the different scaling and rotating of the centroid of the network. This can be seen clearly by expanding Eq.(1.5) to give Eq.(1.6), where **k** is a constant term for all points and obviously affects the translation terms.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{(2)} = k + \begin{bmatrix} \delta x' \\ \delta y' \\ \delta z' \end{bmatrix} + \mu \cdot R \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{(1)} \quad \text{where} \quad k = \begin{bmatrix} \overline{X} \\ \overline{Y} \\ \overline{Z} \end{bmatrix}_{(1)} - \mu \cdot R \cdot \begin{bmatrix} \overline{X} \\ \overline{Y} \\ \overline{Z} \end{bmatrix}_{(1)} \tag{1.6}$$

When transformation parameters from the Molodensky-Badekas model are to be applied to transform coordinates of points, it is essential to know what values were used for the centroid ($\overline{X}_{(1)}$ $\overline{Y}_{(1)}$ $\overline{Z}_{(1)}$) when deriving the parameters. However, in the past they have not always been published with the transformation parameters [Mackie 1982].

It should be noted that when working with global network of points the Molodensky-Badekas model has centroid coordinates equal the centre of the ellipsoid ($\overline{X}_{(1)} = \overline{Y}_{(1)} = \overline{Z}_{(1)} = 0$) and therefore reduces to the Bursa-Wolf model.

1.3. Thesis Objectives

Most often terrestrial national systems are a mixture of two separated coordinates systems: a two-dimensional triangulation network and a one-dimensional height system. Therefore, the horizontal and vertical components are very likely to have different scale factors.

Furthermore, today a geodetic reference frame of high accuracy can be established by using GPS technique. An internal accuracy (1σ) of better than 1 cm in the horizontal component and 1-3 cm in the vertical component is quite feasible, but generally, old national/local geodetic datum were determined to lower accuracy by a conventional terrestrial triangulation, measuring distances and angles, the local datum point being fixed on basis of astronomical observations. The measurements were reduced to the ellipsoid at best taking into account the separation between the geoid and the ellipsoid of the local datum. More than that, often the network has evolved over a time span of several decades. For these and other reasons the geometrical quality of the system might be impaired by considerable distortions, some of them are being quite local, others having a more systematic character (e.g. bias in the scale).

The main objective of the thesis is to carry out some investigations on a different model than the classical one, called *8-parameter transformation model,* using two different scale factors (one for the horizontal component and another for vertical component). In approaching of this goal, firstly a general model with three scale factors *(9-parameter transformation model)* is established and afterwards the 8-parameter transformation model is derived under the constraint that for the horizontal components one can use the same scale factor. This solution is chosen for the reason that it is very convenient and easy to implement in a computer programming language [see Appendix B].

At the end some numerical investigations are carried out, with the new-proposed model, considering the model influence in the residuals of the transformed coordinates.

Chapter 2

Affine transformation with 9-parameters

2.1 General Model

Consider two sets of three-dimensional Cartesian coordinates, forming the vectors $\mathbf{X}_1$ and $\mathbf{X}_2$ (Figure 1.1). The Helmert transformation between these two sets of data can be formulated according with the Eq.(1.1):

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{(2)} = \begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix} + \mu \cdot R(\alpha_1, \alpha_2, \alpha_3) \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{(1)} \tag{2.1}$$

For the sake of simplicity, the above relation can be written also [Moritz 2005]:

$$\mathbf{X}_2 = \Delta \mathbf{x} + \mu \cdot \mathbf{R} \cdot \mathbf{X}_1 \tag{2.2}$$

where:

$\mathbf{X}_1$, $\mathbf{X}_2$ the position vectors of the same point, both in fixed and transformed coordinate system,

$\Delta \mathbf{x}$ the translation (or shift) vector,

μ the scale factor

$\mathbf{R}$ the rotation matrix.

The components of the translation vector

$$\Delta \mathbf{x} = \begin{bmatrix} \delta x \\ \delta y \\ \delta z \end{bmatrix} \tag{2.3}$$

account for the translation of the origin of the C_1 system into the C_2 system.

The rotation matrix is an orthogonal matrix which is composed of three successive rotations and looks like:

$$\mathbf{R} = \mathbf{R}_3(\alpha_3) \cdot \mathbf{R}_2(\alpha_2) \cdot \mathbf{R}_1(\alpha_1) \tag{2.4}$$

A single scale factor is considered in Eq.(2.2), which does nor provide us with information about how big or how small are the scale changes along each axis. One way to solve this problem and gain this information is to assume that each of the axes has a different scale factor. Making this assumption the above model becomes:

$$\mathbf{X}_2 = \Delta\mathbf{x} + \mathbf{R} \cdot \mathbf{S} \cdot \mathbf{X}_1 \tag{2.5}$$

where: $\mathbf{S}$ denotes the total scale matrix (diagonal matrix) of the three scale factors:

$$\underset{3\times3}{\mathbf{S}} = \mathbf{S}(\mu_1, \mu_2, \mu_3) = \begin{bmatrix} \mu_1 & & \\ & \mu_2 & \\ & & \mu_3 \end{bmatrix}, \tag{2.6}$$

μ_1, μ_2, μ_3 being the three different scale factors for each *x*-, *y*- and *z*-axis and which can be written as a sum of unity and a scale change $\delta\mu_j$, often expressed as part per million (ppm):

$$\mu_j = 1 + \delta\mu_j, \quad j = 1, 2, 3 \tag{2.7}$$

Given coordinates contain errors, and in this case the general Eq.(2.5) is not exactly, and some residuals will appear:

$$\mathrm{C}_2 - \varepsilon = \Delta\mathbf{x} + \mathbf{R}(\alpha_1, \alpha_2, \alpha_3) \cdot \mathbf{S}(\mu_1, \mu_2, \mu_3) \cdot \mathrm{C}_1 \tag{2.8}$$

where: ε denotes the residual vector (of errors in the coordinates vector $\mathbf{X}_2$):

$$\varepsilon = \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \end{bmatrix}. \tag{2.9}$$

The task is to estimate the 9-parameter transformation from the two data sets of given coordinates, and Eq.(2.9) will serve as a matrix observation equation in a least squares adjustment. Theoretically, the Cartesian coordinates for three common (identical) points, also denoted as control points, are sufficient to solve for the nine unknown parameters. But usually, in practice, redundant common point information is used, and the unknown parameters are calculated by a least squares adjustment.

2.2 Least squares estimation of the transformation parameters

Three-dimensional coordinate transformation is an important subject for geodesy and also for photogrammetry where you have determined the coordinates of the terrestrial points from their perspective projections on the stereoscopic photographs coordinates in the image coordinate systems are to be transformed to coordinates in the terrestrial coordinates system.
Another example of the application field is laser scanning where the coordinates in the coordinate frames of camera are the output data and that one should be transformed to the terrestrial geodetic coordinate system.
In the both examples that have been described above the rotations and the scale factor changes are large. Furthermore, in the general model given by Eq.(2.8), the observation equations are not linear. For this reason we need to precede a rigorous linearization of Eq.(2.8) with respect to μ_1, μ_2, μ_3 and $\alpha_1, \alpha_2, \alpha_3$, around the approximate values of these six parameters.

2.2.1 Linearization of the rotation matrix

Let α_1^o, α_2^o and α_3^o be the approximate values of the rotation angles. These approximate values can be computed by applying a direct approach [Fan 2005]. If we denote the corresponding corrections by $\delta\alpha_1, \delta\alpha_2$ and $\delta\alpha_3$, respectively, the correct rotation angles will be:

$$\left.\begin{aligned} \alpha_1 &= \alpha_1^o + \delta\alpha_1 \\ \alpha_2 &= \alpha_2^o + \delta\alpha_2 \\ \alpha_3 &= \alpha_3^o + \delta\alpha_3 \end{aligned}\right\} \qquad (2.10)$$

Using the approximate rotation angles α_1^o, α_2^o and α_3^o, we can compute an approximate rotation matrix by Eq.(1.3):

$$\mathbf{R}^{o} = \begin{bmatrix} r_{11}^{o} & r_{12}^{o} & r_{13}^{o} \\ r_{21}^{o} & r_{22}^{o} & r_{23}^{o} \\ r_{31}^{o} & r_{32}^{o} & r_{33}^{o} \end{bmatrix} \tag{2.11}$$

Elements $r_{ij}\,(i, j = 1, 2, 3)$ in matrix **R** are functions of $\alpha_1, \alpha_2, \alpha_3$ and can be linearized at the approximate values α_1^o, α_2^o and α_3^o:

$$r_{ij} = r_{ij}(\alpha_1, \alpha_2, \alpha_3) = r_{ij}(\alpha_1^o + \delta\alpha_1, \alpha_2^o + \delta\alpha_2, \alpha_3^o + \delta\alpha_3) \approx$$

$$\approx r_{ij}(\alpha_1^o, \alpha_2^o, \alpha_3^o) + \frac{\partial r_{ij}}{\partial \alpha_1}\delta\alpha_1 + \frac{\partial r_{ij}}{\partial \alpha_2}\delta\alpha_2 + \frac{\partial r_{ij}}{\partial \alpha_3}\delta\alpha_3 = r_{ij}^o + e_{ij}\delta\alpha_1 + f_{ij}\delta\alpha_2 + g_{ij}\delta\alpha_3 \tag{2.12}$$

where:

$$\left.\begin{aligned} r_{ij}^o &= r_{ij}(\alpha_1^o, \alpha_2^o, \alpha_3^o) \\ e_{ij} &= \frac{\partial r_{ij}}{\partial \alpha_1} \\ f_{ij} &= \frac{\partial r_{ij}}{\partial \alpha_2} \\ g_{ij} &= \frac{\partial r_{ij}}{\partial \alpha_3} \end{aligned}\right\} \tag{2.13}$$

One observation should be done here, that all the derivatives from Eq.(2.13) have to be evaluated for the approximate rotation angles $\alpha_j^o\,(j = 1, 2, 3)$.

Finally, the linearized rotation matrix **R** becomes:

$$\underset{3\times3}{\mathbf{R}} = \mathbf{R}(\alpha_1, \alpha_2, \alpha_3) = \mathbf{R}^o + \mathbf{E}\,\delta\alpha_1 + \mathbf{F}\,\delta\alpha_2 + \mathbf{G}\,\delta\alpha_3 \tag{2.14}$$

where have been introduced the notations:

$$\underset{3\times3}{\mathbf{E}} = \begin{bmatrix} e_{11} & e_{12} & e_{13} \\ e_{21} & e_{22} & e_{23} \\ e_{31} & e_{32} & e_{33} \end{bmatrix}, \quad \underset{3\times3}{\mathbf{F}} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix}, \quad \underset{3\times3}{\mathbf{G}} = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & g_{33} \end{bmatrix} \tag{2.15}$$

All the components of **E** –, **F** – and **G** – matrices are given explicitly by the following formulas:

$$
\begin{array}{lll}
e_{11}=0 & e_{12}=-r^o_{13} & e_{13}=r^o_{12} \\
e_{21}=0 & e_{22}=-r^o_{23} & e_{23}=-r^o_{22} \\
e_{31}=0 & e_{32}=-r^o_{33} & e_{33}=r^o_{32}
\end{array}
\tag{2.16}
$$

$$
\begin{array}{lll}
f_{11}=-\sin\alpha^o_2\cos\alpha^o_3 & f_{12}=-r^o_{32}\cos\alpha^o_3 & f_{13}=-r^o_{33}\cos\alpha^o_3 \\
f_{21}=\sin\alpha^o_2\sin\alpha^o_3 & f_{22}=r^o_{32}\sin\alpha^o_3 & f_{23}=r^o_{33}\sin\alpha^o_3 \\
f_{31}=\cos\alpha^o_2 & f_{32}=\sin\alpha^o_1\sin\alpha^o_2 & f_{33}=-\cos\alpha^o_1\sin\alpha^o_2
\end{array}
\tag{2.17}
$$

$$
\begin{array}{lll}
g_{11}=r^o_{21} & g_{12}=r^o_{22} & g_{13}=r^o_{23} \\
g_{21}=-r^o_{11} & g_{22}=-r^o_{12} & g_{23}=-r^o_{13} \\
g_{31}=0 & g_{32}=0 & g_{33}=0
\end{array}
\tag{2.18}
$$

2.2.2 The scale factors matrix

Let μ^o_1, μ^o_2 and μ^o_3 be the approximate scale factors. Similar to previous section the approximate values can be computed by applying a direct approach [Fan 2005]. If we denote the corresponding corrections by $\delta\mu_1, \delta\mu_2$ and $\delta\mu_3$, respectively, the correct scale factors will be:

$$
\mu_j = \mu^o_j + \delta\mu_j, \quad j = 1, 2, 3 \tag{2.19}
$$

Using the above relation we can write the scale matrix as:

$$
\underset{3\times 3}{\mathbf{S}} = \mathbf{S}(\mu_1, \mu_2, \mu_3) = \begin{bmatrix} \mu^o_1 + \delta\mu_1 & & \\ & \mu^o_2 + \delta\mu_2 & \\ & & \mu^o_3 + \delta\mu_3 \end{bmatrix} = \mathbf{S}^o + \mathbf{M}\,\delta\mu_1 + \mathbf{N}\,\delta\mu_2 + \mathbf{P}\,\delta\mu_3 \tag{2.20}
$$

with the following notations:

$$
\mathbf{S}^o = \begin{bmatrix} \mu^o_1 & & \\ & \mu^o_2 & \\ & & \mu^o_3 \end{bmatrix} \tag{2.21}
$$

$$\underset{3\times3}{\mathbf{M}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \underset{3\times3}{\mathbf{N}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \underset{3\times3}{\mathbf{P}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.22}$$

2.2.3 Linearization of the observation equations

Taking advantage of Eqs. (2.14) and (2.20) and introducing them in Eq.(2.8), our model becomes:

$$\mathrm{C}_2 - \varepsilon = \Delta\mathbf{x} + (\mathbf{R}^o + \mathbf{E}\,\delta\alpha_1 + \mathbf{F}\,\delta\alpha_2 + \mathbf{G}\,\delta\alpha_3)\cdot(\mathbf{S}^o + \mathbf{M}\,\delta\mu_1 + \mathbf{N}\,\delta\mu_2 + \mathbf{P}\,\delta\mu_3)\cdot \mathrm{C}_1$$

$$\approx \Delta\mathbf{x} + (\mathbf{R}^o\mathbf{S}^o + \mathbf{R}^o\mathbf{M}\,\delta\mu_1 + \mathbf{R}^o\mathbf{N}\,\delta\mu_2 + \mathbf{R}^o\mathbf{P}\,\delta\mu_3 + \mathbf{E}\mathbf{S}^o\,\delta\alpha_1 + \mathbf{F}\mathbf{S}^o\,\delta\alpha_2 + \mathbf{G}\mathbf{S}^o\,\delta\alpha_3)\cdot \mathrm{C}_1 \tag{2.23}$$

where the second-order terms ($\delta\alpha_i \cdot \delta\mu_j \approx 0, \quad i,j = 1, 2, 3$) have been neglected.

A more simple form to present the above relation is to write it in the matrix form:

$$\underset{3\times1}{\mathbf{L}_i} - \underset{3\times1}{\varepsilon_i} = \underset{3\times9}{\mathbf{A}_i}\cdot \underset{9\times1}{\delta\mathrm{x}}, \quad i = 1, \mathbf{K}, \mathrm{n} \tag{2.24}$$

where the observation vector $\mathbf{L}_i$, the residual vector ε_i, design matrix $\mathbf{A}_i$ and the vector $\delta\mathbf{x}_i$ containing the unknown parameters, are given by:

$$\underset{3\times1}{\mathbf{L}_i} = \begin{bmatrix} \mathbf{l}_1 \\ \mathbf{l}_2 \\ \mathbf{l}_3 \end{bmatrix}_{(i)} = \mathbf{X}_2 - \mathbf{R}^o\mathbf{S}^o \cdot \mathbf{X}_1, \tag{2.25}$$

$$\underset{3\times1}{\varepsilon_i} = \begin{bmatrix} \varepsilon_{\mathrm{x}} \\ \varepsilon_{\mathrm{y}} \\ \varepsilon_{\mathrm{z}} \end{bmatrix}_{(i)}, \tag{2.26}$$

$$\underset{3\times9}{\mathbf{A}_i} = \begin{bmatrix} 1 & 0 & 0 & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} \\ 0 & 1 & 0 & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} \\ 0 & 0 & 1 & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} \end{bmatrix} = \begin{bmatrix} a_{x_i}^T \\ a_{y_i}^T \\ a_{z_i}^T \end{bmatrix} \tag{2.27}$$

$$\underset{9\times1}{\delta\mathbf{x}} = [\delta x \quad \delta y \quad \delta z \quad \delta\mu_1 \quad \delta\mu_2 \quad \delta\mu_3 \quad \delta\alpha_1 \quad \delta\alpha_2 \quad \delta\alpha_3]^T \tag{2.28}$$

Elements in the reduced observation vector L_i can be computed from the explicit formulas:

$$\left.\begin{aligned}
\mathbf{l}_1 &= X_i^{(2)} - (\mu_1^o \cdot r_{11}^o \cdot X_i^{(1)} + \mu_2^o \cdot r_{12}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{13}^o \cdot Z_i^{(1)}) \\
\mathbf{l}_2 &= Y_i^{(2)} - (\mu_1^o \cdot r_{21}^o \cdot X_i^{(1)} + \mu_2^o \cdot r_{22}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{23}^o \cdot Z_i^{(1)}) \\
\mathbf{l}_3 &= Z_i^{(2)} - (\mu_1^o \cdot r_{31}^o \cdot X_i^{(1)} + \mu_2^o \cdot r_{32}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{33}^o \cdot Z_i^{(1)})
\end{aligned}\right\} \tag{2.29}$$

and the elements of the design matrix **A** are as follows:

$$\begin{aligned}
a_{14} &= r_{11}^o X_i^{(1)} & a_{15} &= r_{12}^o Y_i^{(1)} & a_{16} &= r_{13}^o Z_i^{(1)} \\
a_{24} &= r_{21}^o X_i^{(1)} & a_{25} &= r_{22}^o Y_i^{(1)} & a_{26} &= r_{23}^o Z_i^{(1)} \\
a_{34} &= r_{31}^o X_i^{(1)} & a_{35} &= r_{32}^o Y_i^{(1)} & a_{36} &= r_{33}^o Z_i^{(1)}
\end{aligned}$$

$$\begin{aligned}
a_{17} &= -\mu_2^o \cdot r_{13}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{12}^o \cdot Z_i^{(1)} \\
a_{27} &= -\mu_2^o \cdot r_{23}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{22}^o \cdot Z_i^{(1)} \\
a_{37} &= -\mu_2^o \cdot r_{33}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{32}^o \cdot Z_i^{(1)}
\end{aligned}$$

$$\begin{aligned}
a_{18} &= -\cos\alpha_3^o \cdot (\mu_1^o \cdot \sin\alpha_2^o \cdot X_i^{(1)} + \mu_2^o \cdot r_{32}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{33}^o \cdot Z_i^{(1)}) \\
a_{28} &= \sin\alpha_3^o \cdot (\mu_1^o \cdot \sin\alpha_2^o \cdot X_i^{(1)} + \mu_2^o \cdot r_{32}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{33}^o \cdot Z_i^{(1)}) \\
a_{38} &= \mu_1^o \cdot \cos\alpha_2^o \cdot X_i^{(1)} + \mu_2^o \cdot \sin\alpha_1^o \cdot \sin\alpha_2^o \cdot Y_i^{(1)} - \mu_3^o \cdot \cos\alpha_1^o \cdot \sin\alpha_2^o \cdot Z_i^{(1)}
\end{aligned}$$

$$\begin{aligned}
a_{19} &= \mu_1^o \cdot r_{21}^o \cdot X_i^{(1)} + \mu_2^o \cdot r_{22}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{23}^o \cdot Z_i^{(1)} \\
a_{29} &= -(\mu_1^o \cdot r_{11}^o \cdot X_i^{(1)} + \mu_2^o \cdot r_{12}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{13}^o \cdot Z_i^{(1)}) \\
a_{39} &= 0
\end{aligned} \tag{2.30}$$

Eq.(2.23) is the linearized observation equations for one single control point *i*. For all *n* points, the joint observation equation matrix form becomes:

$$\underset{3n\times1}{\mathbf{L}} - \varepsilon = \underset{3n\times9}{\mathbf{A}} \cdot \underset{9\times1}{\delta\mathbf{x}} \tag{2.31}$$

2.2.4 Least squares estimates of the transformation parameters

The least squares estimate of the corrections to the transformation parameters can be computed very easy using the well-known formula [Fan 1997]:

$$\delta\hat{\mathbf{x}} = (\mathbf{A}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{A})^{-1}\,\mathbf{A}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{L} \qquad (2.32)$$

where:

$\delta\hat{\mathbf{x}}$ is an estimation vector of $\delta\mathbf{x}$;

$\mathbf{C}$ denote the variance-covariance matrix of the rectangular coordinates in the first (transformed) coordinate system.

Finally, the nine transformation parameters are:

- the translations: $\delta x \quad \delta y \quad \delta z$;
- the corrected scale factors: $\hat{\mu}_j = \mu_j^o + \delta\hat{\mu}_j, \quad j = 1, 2, 3$;
- the corrected rotation angles: $\hat{\alpha}_j = \alpha_j^o + \delta\hat{\alpha}_j, \quad j = 1, 2, 3$.

Furthermore, the corrected values of the scale factors and rotations angles can be used as new approximated values in an iterative linearization and adjustment procedure. Normally the solution converges very quickly after two or three iterations.

2.2.5 Least squares estimates of the residuals

Taking advantage of Eqs. (2.14) and (2.20) and introducing them in Eq.(2.7), our model the least squares estimate of residuals are computed based on

$$\hat{\varepsilon} = \mathbf{L} - \mathbf{A} \cdot \delta\hat{\mathbf{x}} \qquad (2.33)$$

The variance-covariance matrix of the transformation parameters ($\mathbf{C}_{\hat{x}\hat{x}}$) can also be estimated based on the "*a posteriori*" estimated variance factor ($\hat{\sigma}_o^2$):

$$\hat{\sigma}_o^2 = \frac{\hat{\varepsilon}^{\mathrm{T}}\,\mathbf{C}^{-1}\,\hat{\varepsilon}}{3n - 9} \qquad (2.34)$$

It becomes:

$$\mathbf{C}_{\hat{x}\hat{x}} = \hat{s}_o^2 \cdot (\mathbf{A}^T \mathbf{C}^{-1} A)^{-1} \tag{2.35}$$

With this last step the least squares adjustment process is completed and the proposed new transformation model – *Affine transformation with 9-parameters* – has been established. The numerical investigations concerning this model are presented in the Chapter 4.

Chapter 3

Affine transformation with 8-parameters

So far we have established the model for a 3D coordinate transformation in the case when different scale factors are used for each of the three axes. Furthermore, it is generally the case that classical networks will differ from modern space-based networks due to the method of computation and establishment. Classical national networks were set up very early by classical terrestrial measurements (triangulation, triangulation-trilateration and precise levelling) and were consisted of different number of fix points with three dimensional coordinates. But this set of coordinates is in fact a mixture of two different independent networks: one horizontal network (set up by angle and/or distance measurements) and one levelling network. Taking into account this observation we can speculate that the coordinate components related with the different network type might have different scale factors. So, because the *x*- and *y*-components were simultaneously determined, they should have the same scale factor, while the *z*-component has a different scale factor as it was based on an independent adjustment.

To check this assumption and at the same time to take advantage of the general model with 9-parameter that has been described in the previous chapter and in order to establish the mathematical algorithm of the model with 8-parameters, we decided to apply the algorithm of least squares adjustment with constraints, described below.

3.1 Least squares adjustment with constraints

Assume the general observation equation model:

$$\underset{3n\times1}{\mathbf{L}} - \underset{3n\times1}{\boldsymbol{\varepsilon}} = \underset{3n\times9}{\mathbf{A}} \cdot \underset{9\times1}{\delta\mathbf{x}} \tag{3.1}$$

where $\mathbf{L}, \varepsilon$ and $\mathbf{A}$ are defined as in (2.25), (2.26) (2.27). The residual vector ε is assumed to have the following a priori statistical property:

$$E(\varepsilon) = 0, \qquad E(\varepsilon\varepsilon^T) = \sigma_o^2 P^{-1} \tag{3.2}$$

Having one scale factor for the horizontal component means the following equation of constraint:

$$\mu_1 = \mu_2 \tag{3.3}$$

or in a matrix form:

$$\underset{1\times 9}{\mathbf{A}_x} \cdot \underset{9\times 1}{\delta\mathbf{x}} = \underset{1\times 1}{\mathbf{d}} \tag{3.4}$$

where: $\mathbf{A}_x, \mathbf{d}$ are given vectors:

$$\mathbf{A}_x = \begin{bmatrix} 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.5}$$

$$\mathbf{d} = 0 \tag{3.6}$$

Now our task is to make the least squares procedure and find the estimation vector of the parameters ($\delta\hat{\mathbf{x}}$) and the least squares estimates of the residuals ($\hat{\varepsilon}$) such that the following three conditions are satisfied:

$$\hat{\varepsilon}^T \mathbf{P} \hat{\varepsilon} = \text{minimum} \tag{3.7}$$

$$\mathbf{L} - \varepsilon = \mathbf{A} \cdot \delta\hat{\mathbf{x}} \tag{3.8}$$

$$\mathbf{A}_x \cdot \delta\hat{\mathbf{x}} = \mathbf{d} \tag{3.9}$$

The least squares estimate of the corrections to the transformation parameters can be computed using the formula [Fan, 1997]:

$$\delta\hat{\mathbf{x}} = \mathbf{N}^{-1}(\mathbf{I} - \mathbf{A}_x^T \mathbf{N}_x^{-1} \mathbf{A}_x \mathbf{N}^{-1}) \mathbf{A}^T \mathbf{C}^{-1} \mathbf{L} + \mathbf{N}^{-1} \mathbf{A}_x^T \mathbf{N}_x^{-1} \mathbf{d} \tag{3.10}$$

with:

$$\mathbf{N} = \mathbf{A}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{A} \qquad \text{and} \qquad \mathbf{N}_{\mathrm{x}} = \mathbf{A}_{\mathrm{x}}\mathbf{N}^{-1}\mathbf{A}_{\mathrm{x}}^{\mathrm{T}} \tag{3.11}$$

The estimated residual $\hat{\varepsilon}$ is obtained in the usual way:

$$\hat{\varepsilon} = \mathbf{L} - \mathbf{A} \cdot \delta\hat{\mathrm{x}} \tag{3.12}$$

3.2 Derived formulas in 8-parameter model

Based on the above algorithm, further below are presented the explicitly formulas for all the quantities that are presented in the general linearized equation of *Affine transformation with 8-parameters*:

$$\underset{3\times1}{\mathbf{L}_i} - \underset{3\times1}{\varepsilon_i} = \underset{3\times8}{\mathbf{A}_i} \cdot \underset{8\times1}{\delta\mathrm{x}}, \quad i = 1,\ldots,\mathrm{n} \tag{3.13}$$

where the observation vector $\mathbf{L}_i$ is given by:

$$\underset{3\times1}{\mathbf{L}_i} = \begin{bmatrix} \mathbf{l}_1 \\ \mathbf{l}_2 \\ \mathbf{l}_3 \end{bmatrix} = \mathbf{X}_2 - \mathbf{R}^{\mathrm{o}}\mathbf{S}^{\mathrm{o}} \cdot \mathbf{X}_1, \tag{3.14}$$

with:

C_1 and C_2 stand for 3D coordinate vector of the same point, both in fixed coordinate system (1) and transformed coordinate system (2);

$\mathbf{R}^{\mathrm{o}}$ stand for approximate rotation matrix (squared matrix) computed based on approximate rotation angles $\alpha_1^{\mathrm{o}}, \alpha_2^{\mathrm{o}}$ and α_3^{o} and Eq.(2.11);

$\mathbf{S}^{\mathrm{o}}$ stand for approximate scale factor matrix (diagonal matrix) computed based on the approximate scale factors μ_1^{o} and μ_2^{o}

$$\mathbf{S}^{\mathrm{o}} = \begin{bmatrix} \mu_1^{\mathrm{o}} & & \\ & \mu_1^{\mathrm{o}} & \\ & & \mu_3^{\mathrm{o}} \end{bmatrix} \tag{3.15}$$

The residual vector ε_i is given by:

$$\underset{3\times1}{\varepsilon_i} = \begin{bmatrix} \varepsilon_{x_i} \\ \varepsilon_{y_i} \\ \varepsilon_{z_i} \end{bmatrix}, \tag{3.16}$$

The design matrix $\mathbf{A}_i$ is given by:

$$\underset{3\times8}{\mathbf{A}_i} = \begin{bmatrix} 1 & 0 & 0 & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} \\ 0 & 1 & 0 & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} \\ 0 & 0 & 1 & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} \end{bmatrix} = \begin{bmatrix} a_{x_i}^T \\ a_{y_i}^T \\ a_{z_i}^T \end{bmatrix} \tag{3.17}$$

The vector $\delta\mathbf{x}$ containing the unknown parameters is given by:

$$\underset{8\times1}{\delta\mathbf{x}} = \begin{bmatrix} \delta x \\ \delta y \\ \delta z \\ \delta\mu_1 \\ \delta\mu_2 \\ \delta\alpha_1 \\ \delta\alpha_2 \\ \delta\alpha_3 \end{bmatrix} \tag{3.18}$$

Elements in the reduced observation vector $\mathbf{L}_i$ can be computed from the explicit formulas:

$$\left.\begin{aligned} \mathbf{l}_1 &= X_i^{(2)} - (\mu_1^o \cdot r_{11}^o \cdot X_i^{(1)} + \mu_1^o \cdot r_{12}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{13}^o \cdot Z_i^{(1)}) \\ \mathbf{l}_2 &= Y_i^{(2)} - (\mu_1^o \cdot r_{21}^o \cdot X_i^{(1)} + \mu_1^o \cdot r_{22}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{23}^o \cdot Z_i^{(1)}) \\ \mathbf{l}_3 &= Z_i^{(2)} - (\mu_1^o \cdot r_{31}^o \cdot X_i^{(1)} + \mu_1^o \cdot r_{32}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{33}^o \cdot Z_i^{(1)}) \end{aligned}\right\} \tag{3.19}$$

and the elements of the design matrix **A** are as follows:

$$a_{14} = r_{11}^o X_i^{(1)} + r_{12}^o Y_i^{(1)}$$
$$a_{24} = r_{21}^o X_i^{(1)} + r_{22}^o Y_i^{(1)}$$
$$a_{34} = r_{31}^o X_i^{(1)} + r_{32}^o Y_i^{(1)}$$

$$a_{15} = r_{13}^o Z_i^{(1)}$$
$$a_{25} = r_{23}^o Z_i^{(1)}$$
$$a_{35} = r_{33}^o Z_i^{(1)}$$

$$a_{16} = -\mu_1^o \cdot r_{13}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{12}^o \cdot Z_i^{(1)}$$
$$a_{26} = -\mu_1^o \cdot r_{23}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{22}^o \cdot Z_i^{(1)}$$
$$a_{36} = -\mu_1^o \cdot r_{33}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{32}^o \cdot Z_i^{(1)}$$

$$a_{17} = -\cos\alpha_3^o \cdot (\mu_1^o \cdot \sin\alpha_2^o \cdot X_i^{(1)} + \mu_1^o \cdot r_{32}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{33}^o \cdot Z_i^{(1)})$$
$$a_{27} = \sin\alpha_3^o \cdot (\mu_1^o \cdot \sin\alpha_2^o \cdot X_i^{(1)} + \mu_1^o \cdot r_{32}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{33}^o \cdot Z_i^{(1)})$$
$$a_{37} = \mu_1^o \cdot \cos\alpha_2^o \cdot X_i^{(1)} + \mu_1^o \cdot \sin\alpha_1^o \cdot \sin\alpha_2^o \cdot Y_i^{(1)} - \mu_3^o \cdot \cos\alpha_1^o \cdot \sin\alpha_2^o \cdot Z_i^{(1)}$$

$$a_{18} = \mu_1^o \cdot r_{21}^o \cdot X_i^{(1)} + \mu_1^o \cdot r_{22}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{23}^o \cdot Z_i^{(1)}$$
$$a_{28} = -(\mu_1^o \cdot r_{11}^o \cdot X_i^{(1)} + \mu_1^o \cdot r_{12}^o \cdot Y_i^{(1)} + \mu_3^o \cdot r_{13}^o \cdot Z_i^{(1)}) \tag{3.20}$$
$$a_{38} = 0$$

The above Eq.(3.13) is for one single control point *i*. For all *n* points, the joint observation equation matrix form becomes:

$$\underset{3n\times1}{\mathbf{L}} - \varepsilon = \underset{3n\times8}{\mathbf{A}} \cdot \underset{8\times1}{\delta\mathbf{x}} \tag{3.21}$$

The "*a posteriori*" estimate of unit-weight standard error ($\hat{\sigma}$) can be obtained based on the estimated residual vector ($\hat{\varepsilon}$):

$$\hat{\sigma}_o = \sqrt{\frac{\hat{\varepsilon}^T \mathbf{C}^{-1} \hat{\varepsilon}}{3n - 8}} \tag{3.22}$$

Using error propagation law, we can derive the variances of estimated unknown parameters vector ($\delta\hat{\mathbf{x}}$):

$$\mathbf{C}_{\hat{x}\hat{x}} = \hat{\sigma}_{\mathbf{o}}^2(\mathbf{N}^{-1} - \mathbf{N}^{-1}\mathbf{A}_x^{\mathbf{T}}\mathbf{N}_x^{-1}\mathbf{A}_x\mathbf{N}^{-1}) \tag{3.23}$$

and for the estimated residual vector ($\hat{\varepsilon}$):

$$\mathbf{C}_{\hat{\varepsilon}\hat{\varepsilon}} = \hat{\sigma}_0^2[\mathbf{C} - \mathbf{A}(\mathbf{N}^{-1} - \mathbf{N}^{-1}\mathbf{A}_{\mathbf{x}}^{\mathrm{T}}\mathbf{N}_{\mathbf{x}}^{-1}\mathbf{A}_{\mathbf{x}}\mathbf{N}^{-1})\,\mathbf{A}^{\mathrm{T}}] \tag{3.24}$$

Eq.(3.23) and (3.24) are convenient for a computer implementation because in the same file we can have all the models and a variable which allows as to chose how many parameters the model should have is required only [Appendix B].

In this way the transformation model with 8-parameters has been developed together with all the formulas and equations. With this model the numerical investigations and the tests for the new proposed model are conducted in the next chapter.

Chapter 4

Numerical tests

4.1 Preparation of numerical tests

The most common situation of coordinate transformation is the transformation between a global reference coordinate frame (WGS 84, ETRS89, $ITRF_{xx}$ or local / national realisations of ETRS89) and some national or local horizontal datum.

For our research work we use a set of 20 points with geocentric rectangular coordinates known in a geocentric coordinate system SWEREF 93, the Swedish realization of ETRS89 (EUREF89) and local reference coordinate system, which is a mixture of the Swedish triangulation network RT90 and the 2^{nd} Swedish precise levelling network RH70 (Table 4.1). It is assumed that the coordinates of the global system has a high internal accuracy and this is superior to the local one.

When it comes to the problem of computing transformation parameters between a globally adjusted reference frame and a local geodetic datum Eq.(1.1) is not so well suited for the following reasons [Reit 1998]:

- The rotation matrix should be linearized;
- Most software implementations of Eq.(1.1) presuppose two geocentric systems (although the equation is not restricted to that case);
- The known coordinates ought to be assigned appropriate weights in the fitting process.

The Cartesian coordinates of the Swedish triangulation network are in fact a mixture of (ϕ, λ, h). Furthermore, working with geocentric coordinates makes it hard to evaluate the result of the adjustment process. Residuals and rotations are much easier to interpret when they are expressed in a topocentric (local) coordinate system (*e, n, u*), where the origin of this coordinate system is considered each point *i (X, Y, Z).*

Table 4.1: Geocentric rectangular coordinates of the common points

No. Pnt.	System 1 (SWEREF 93)			System 2 (RT90/RH70)		
	X	Y	Z	X	Y	Z
1	2441775.419	2441276.712	799286.666	5818162.025	799268.100	5818729.162
2	3464655.838	3464161.275	845805.461	5269712.429	845749.989	5270271.528
3	3309991.828	3309496.800	828981.942	5370322.060	828932.118	5370882.280
4	3160763.338	3160269.913	759204.574	5468784.081	759160.187	5469345.504
5	2248123.493	2247621.426	865698.413	5885856.498	865686.595	5886425.596
6	3022573.157	3022077.340	802985.055	5540121.276	802945.690	5540683.951
7	3104219.427	3103716.966	998426.412	5462727.814	998384.028	5463290.505
8	2998189.685	2997689.029	931490.201	5532835.154	931451.634	5533398.462
9	3199093.294	3198593.776	932277.179	5419760.966	932231.327	5420322.483
10	3370658.823	3370168.626	711928.884	5349227.574	711876.990	5349786.786
11	3341340.173	3340840.578	957963.383	5329442.724	957912.343	5330003.236
12	2534031.166	2533526.497	975196.347	5751510.935	975174.455	5752078.309
13	2838909.903	2838409.359	903854.897	5620095.593	903822.098	5620660.184
14	2902495.079	2902000.172	761490.908	5609296.343	761455.843	5609859.672
15	2682407.890	2681904.794	950423.098	5688426.909	950395.934	5688993.082
16	2620258.868	2619761.810	779162.964	5743233.630	779138.041	5743799.267
17	3246470.535	3245966.134	1077947.976	5364716.214	1077900.355	5365277.896
18	3249408.275	3248918.041	692805.543	5425836.841	692757.965	5426396.948
19	2763885.496	2763390.878	733277.458	5682089.111	733247.387	5682653.347
20	2368885.005	2368378.937	994508.273	5817909.286	994492.233	5818478.154

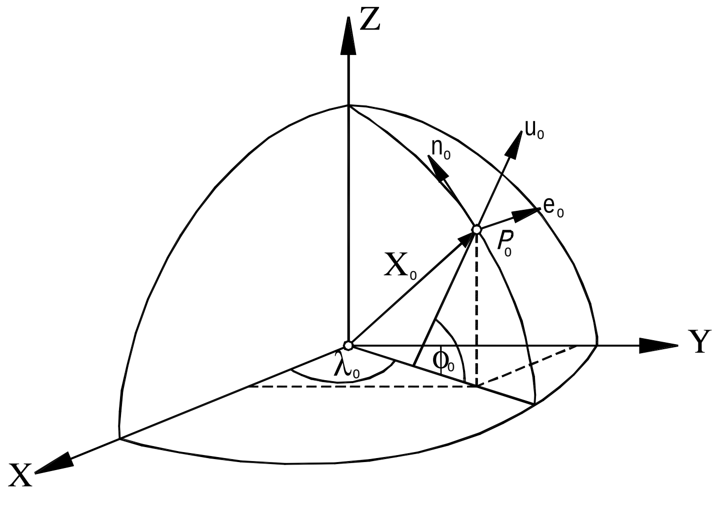

Figure 4.1: Global geocentric and local level coordinate systems

Therefore, one introduce a “local level system” [Moritz, 2005] referred to a tangential plane to the level surface at point P_o with known geocentric coordinates (X_o, Y_o, Z_o) and to the local vertical. This point may be chosen arbitrarily, but it might be convenient to choose the barycentre of the transformed network.

$$X_o = \frac{1}{n}\sum_{i=1}^{n} X_i, \qquad Y_o = \frac{1}{n}\sum_{i=1}^{n} Y_i, \qquad Z_o = \frac{1}{n}\sum_{i=1}^{n} Z_i \tag{4.1}$$

The axes $\mathbf{n}_o, \mathbf{e}_o, \mathbf{u}_o$ of this local coordinate system at point P_o, corresponding to the north, east and up directions are thus represented in the global system by:

$$\mathbf{n}_o = \begin{bmatrix} -\sin\phi_o \cos\lambda_o \\ -\sin\phi_o \sin\lambda_o \\ \cos\phi_o \end{bmatrix}, \quad \mathbf{e}_o = \begin{bmatrix} -\sin\lambda_o \\ \cos\lambda_o \\ 0 \end{bmatrix}, \quad \mathbf{u}_o = \begin{bmatrix} \cos\phi_o \cos\lambda_o \\ \cos\phi_o \sin\lambda_o \\ \sin\phi_o \end{bmatrix}, \tag{4.2}$$

where (ϕ_o, λ_o) stand for geodetic coordinates of the barycentre of the reference system and the vectors $\mathbf{n}_o$ and $\mathbf{e}_o$ span the tangent plane in point P_o (Figure 4.1). The third coordinate axis of the local level system (the vector $\mathbf{u}_o$) is orthogonal to the tangential plane and has the direction of the plumb line.

Furthermore, similar to geocentric coordinates, we might assign different scale factors for the new topocentric coordinate sets. According to the proposed new model the following remarks are derived:

- $\mathbf{e}_o$ and $\mathbf{n}_o$ might be regarded as the horizontal components, and for both of them we assume only one common scale factor;
- $\mathbf{u}_o$ might be regarded as a height component, and we assign another scale factor, different from the one of the horizontal components.

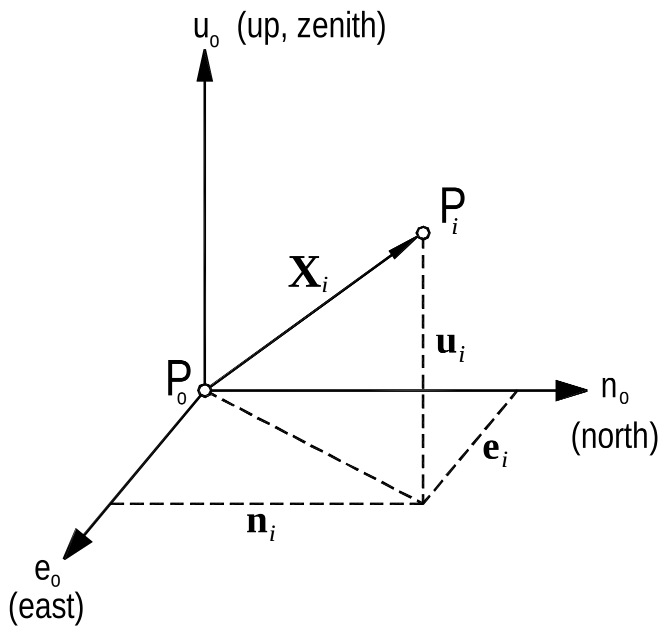

Figure 4.2: Components of the position vector in the local level system

Now the components n_i, e_i, u_i of the vector $\mathbf{x}_i$ in the local level system are introduced. Considering Figure 4.2, these components are obtained by a projection of the vector $\mathbf{X}_i$ onto the local axes $\mathbf{n}_i, \mathbf{e}_i, \mathbf{u}_i$. Analytically, this is achieved by scalar products. Therefore,

$$\mathbf{x}_i = \begin{bmatrix} n_i \\ e_i \\ u_i \end{bmatrix} = \begin{bmatrix} \mathbf{n}_i \cdot \mathbf{X}_i \\ \mathbf{e}_i \cdot \mathbf{X}_i \\ \mathbf{u}_i \cdot \mathbf{X}_i \end{bmatrix} \tag{4.3}$$

is obtained. Assembling the vectors $\mathbf{n}_o, \mathbf{e}_o, \mathbf{u}_o$ of the local level system as columns in an orthogonal matrix $\mathbf{D}_o$,

$$\mathbf{D}_o = \mathbf{D}_o(\phi_o, \lambda_o) = \begin{bmatrix} -\sin\phi_o \cos\lambda_o & -\sin\lambda_o & \cos\phi_o \cos\lambda_o \\ -\sin\phi_o \sin\lambda_o & \cos\lambda_o & \cos\phi_o \sin\lambda_o \\ \cos\phi_o & 0 & \sin\phi_o \end{bmatrix}, \tag{4.4}$$

Eq.(4.3) may be written concisely as:

$$\mathbf{x}_i = \mathbf{D}_o^T \mathbf{X}_i \tag{4.5}$$

We should mention that in Eq.(4.5) $\mathbf{X}_i$ is defined as the vector between the barycentre and the ***i-th*** point in the global coordinate system (as can be easier seen from Figure 4.2 as well):

$$\mathbf{X}_i = \begin{bmatrix} X_i - X_o \\ Y_i - Y_o \\ Z_i - Z_o \end{bmatrix} \tag{4.6}$$

The numerical values for the topocentric coordinates are presented in Table 4.2.

As has been said above we prefer to express the residuals of the least squares adjustment and the rotations as local north-, east- and up-components. To be able to do so, we introduce two local systems, one with the origin at the barycentre of the control points expressed in the first system and other with the origin at the barycentre of the control points expressed in the second system. The transformation model between the two local systems using Eq.(2.8) looks like:

$$\mathbf{x}_2 - \bar{\varepsilon} = \Delta\bar{\mathbf{x}} + \bar{\mathbf{R}} \cdot \bar{\mathbf{S}} \cdot \mathbf{x}_1 \tag{4.7}$$

where:

$\mathbf{x}_1$, $\mathbf{x}_2$ stand for topocentric coordinates of the fixed coordinates system (1) and the transformed coordinate system (2) respectively,

$\Delta\bar{\mathbf{x}}$ stands for translation between the local origins;

$\bar{\mathbf{R}}$ stands for rotation matrix expressing the rotations between the two local systems, and it can be written according to Eq.(1.2);

$\bar{\mathbf{S}}$ stands for scale factors matrix;

$\bar{\varepsilon}$ stands for residual vector with respect to the local topocentric reference system having origin in the barycentre of the control points.

To express the vector of residuals ($\bar{\varepsilon}$) in the North, East and Up components (ε_{NEU}) with respect to a coordinate system having its origin at the point ***i-th***, first one multiplies it with the matrix $\mathbf{D}_o$ and then with the inverse of the matrix $\mathbf{D}_i$ where the matrices are defined according to Eq.(4.4) and with the respect to the transformed coordinate system:

$$\varepsilon_{NEU} = \mathbf{D}_{i,(2)}\,\mathbf{D}_{o,(2)}^{-1}\,\bar{\varepsilon} \tag{4.8}$$

where: (4.9)

$$\mathbf{D}_i = \mathbf{D}_i(\phi_i, \lambda_i) = \begin{bmatrix} -\sin\phi_i \cos\lambda_i & -\sin\lambda_i & \cos\phi_i \cos\lambda_i \\ -\sin\phi_i \sin\lambda_i & \cos\lambda_i & \cos\phi_i \sin\lambda_i \\ \cos\phi_i & 0 & \sin\phi_i \end{bmatrix} \tag{4.10}$$

Eq.(4.8) is convenient because it offers us the possibility to investigate the horizontal residuals versus vertical residuals and if we can reduce them without changing the coordinates.

4.2 Tests for 7 –, 8 – and 9 – parameter model

Numerical investigations are carried out using the commercial package of software MATLAB and it involves three major steps as has been stated before:

1. Computation of the local topocentric coordinates (Table 4.2);

2. Numerical computation of the transformation parameters for a transformation model with 7 – 8 – and 9 – parameters both for the global geodetic coordinates (Table 4.3) and local topocentric coordinates (Table 4.4);
3. Investigation concerning the least squares estimates of the residuals in the coordinates of the common points used in the transformation.

Table 4.2: Local topocentric coordinates of the common points. The origin of the topocentric coordinate reference frame is assumed to be the barycentre of the network

No. Pnt.	System 1			System 2		
	N (north)	E (east)	U (up)	N (north)	E (east)	U (up)
1	563600.255	78292.294	-11736.010	563599.438	78303.693	-11732.331
2	-572086.679	-165547.693	-14554.597	-572083.961	-165559.007	-14558.496
3	-389444.783	-138070.352	48.624	-389442.388	-138078.183	45.902
4	-199314.707	-162930.880	8097.170	-199311.750	-162935.054	8095.194
5	742636.961	196622.159	-32692.936	742634.029	196637.599	-32687.846
6	-59589.289	-81954.850	12846.961	-59587.808	-81956.282	12846.157
7	-213807.100	82529.526	9137.567	-213808.996	82525.222	9136.609
8	-74352.930	48211.961	12632.778	-74354.020	48210.393	12632.469
9	-297922.030	-7691.529	6252.912	-297922.126	-7697.480	6251.229
10	-421681.043	-267482.392	-6335.302	-421676.157	-267491.043	-6338.615
11	-467361.549	-23163.971	-3851.951	-467361.472	-23173.153	-3854.723
12	410450.991	221045.611	-3779.987	410446.981	221053.973	-3776.884
13	108427.764	66617.956	11936.146	108426.475	66620.020	11936.876
14	84944.278	-87900.945	12494.129	84946.045	-87899.491	12494.311
15	261424.043	155432.805	5982.270	261421.151	155438.029	5984.114
16	382401.617	8649.706	2156.920	382401.878	8657.257	2159.254
17	-400260.055	118706.523	-413.138	-400262.820	118698.691	-414.835
18	-278118.357	-251634.874	2322.696	-278113.712	-251640.685	2320.308
19	243523.285	-75879.036	8566.650	243525.055	-75874.384	8567.895
20	576529.330	286147.983	-19110.901	576524.157	286159.887	-19106.586

To compute the numerical values of the transformation parameters a new function called "*GeneralHelmert*" has been implemented. This function requires six input variables, as it is described below and in the Appendix B:

- ***Var1***, ***Var2*** two files which contains the Cartesian coordinates of the common points that are used to compute the transformation parameters, in the first (fix) coordinate system and respectively the second (transformed) coordinate system;
- ***Var3***, a column vector with the approximate values of the scale factors;
- ***Var4***, a column vector with the approximate values of the rotations angles corresponding for each axis in part;

- ***Var5***, variance-covariance matrix related to the data set that is concerning in the transformation;
- ***Var6***, number of the parameters that the model should involve.

A remark should be made here that an iterative method is used in the computation of the transformation parameters.

Table 4.3: Transformation parameters and their standard errors when geocentric coordinates are assumed to be used in the transformations

Parameter	**7-parameters**		**8-parameters**		**9-parameters**	
	Value	Stand. dev	Value	Stand. dev	Value	Stand. dev
δx [m]	-419.568	0.39	-421.199	2.69	-422.604	4.32
δy [m]	-99.246	1.44	-99.753	1.67	-99.903	1.72
δz [m]	-591.456	0.43	-588.071	5.55	-585.318	8.65
$\delta\mu_1$ [ppm]	1.0237	0.06	1.1370	0.19	1.2425	0.32
$\delta\mu_2$ [ppm]	1.0237	0.06	1.1370	0.19	1.0807	0.24
$\delta\mu_3$ [ppm]	1.0237	0.06	0.5497	0.78	0.1642	1.21
$\delta\alpha_1$ [arsec]	0.850189	0.04	0.862322	0.05	0.868641	0.05
$\delta\alpha_2$ [arsec]	1.814145	0.01	1.765104	0.08	1.724197	0.13
$\delta\alpha_3$ [arsec]	-7.853479	0.02	-7.859223	0.03	-7.861238	0.03
$\hat{\sigma}_o$	0.110		0.111		0.112	

Table 4.4: Transformation parameters and their standard errors in the local coordinate system with the origin in the barycentre of the network

Parameter	**7-parameters**		**8-parameters**		**9-parameters**	
	Value	Stand. dev	Value	Stand. dev	Value	Stand. dev
$\delta\bar{x}$ [m]	0.000	0.00	0.000	0.00	0.000	0.00
$\delta\bar{y}$ [m]	0.000	0.00	0.000	0.00	0.000	0.00
$\delta\bar{z}$ [m]	0.000	0.00	0.000	0.00	0.000	0.00
$\delta\mu_1$ [ppm]	1.0237	0.06	1.0281	0.06	1.0200	0.06
$\delta\mu_2$ [ppm]	1.0237	0.06	1.0281	0.06	**1.0804**	**0.21**
$\delta\mu_3$ [ppm]	1.0237	0.06	**-4.3883**	**2.14**	**-4.3886**	**2.16**
$\delta\bar{\alpha}_1$ [arsec]	-0.739390	0.05	-0.726803	0.04	-0.726660	0.04
$\delta\bar{\alpha}_2$ [arsec]	1.192284	0.02	1.183746	0.02	1.183791	0.02
$\delta\bar{\alpha}_3$ [arsec]	-4.109449	0.01	-4.109537	0.01	-4.106671	0.02
$\hat{\sigma}_o$	0.110		0.105		0.106	

Please notice that the results presented in the above tables are obtained by using an adjustment model where all the coordinates have the *same weights* or the *same variances*. Corresponding to this case, the expression "*conventional adjustment model*" is introduced according on [Reit 1999].

The numerical values of the transformation parameters as well as their standard errors are given in the Table 4.4. To have a clearer idea about how the model is influencing the numerical computations, the results are also presented for *7-parameter* and *9-parameter* transformation models. Based on these numerical results some remarks might be concluded:

1. As was expected, the translations are zero due to fact that a system of coordinates whose origin is at the barycentre of the common transformation points has been adopted and furthermore the model from Eq.(4.7) is reduced to a combination of scale and rotation only (and the size of numbers involved in the computations is reduced);
2. The 9-parameter transformation model, in which we assumed different scale factors for all three axes, shows for horizontal components very close results: ($\mu_1 = 1.0200\text{ppm}$ and $\mu_2 = 1.0806\text{ppm}$), which reinforces the assumption (used in establishing the 8-parameter model) that these two components might have similar scale factors;
3. The 8-parameter transformation model shows that scale factor for horizontal components ($\mu_1 = \mu_2 = \mu_H = 1.0281\text{ppm}$) is different from the vertical scale factor ($\mu_3 = \mu_V = -4.3883\text{ppm}$)

Last remark requires a statistical investigation and from this reason, the difference of the mean values of these estimated two scale factors is computed:

$$\hat{y} = \mu_V - \mu_H = \mu_3 - \mu_1 \tag{4.11}$$

The new function ($\hat{y}$) is a linear combination of μ_1 and μ_3, two correlated variables which are part from the parameter vector ($\delta\hat{\mathbf{x}}$) with full variance-covariance matrix ($\mathbf{C}_{\hat{x}\hat{x}}$).

Now, re-writing the above Eq.(4.10) in a matrix form

$$\hat{y} = \underset{1\times9}{\mathbf{F}} \cdot \underset{9\times1}{\delta\hat{\mathbf{x}}} \tag{4.12}$$

with **F** – matrix being known

$$\underset{1\times 9}{\mathbf{F}} = [0 \quad 0 \quad 0 \quad -1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0], \tag{4.13}$$

the variance of this function can be then computed by applying the general error propagation law in a matrix form as follows:

$$\hat{\sigma}_y^2 = \mathrm{F} \cdot \mathrm{C}_{\hat{x}\hat{x}} \cdot \mathrm{F}^{\mathrm{T}} . \tag{4.14}$$

If the confidence interval of is given by:

$$\left[\hat{y} - \mathrm{t}_{\alpha/2}(\mathrm{n}-\mathrm{m}) \cdot \sigma_y, \quad \hat{y} + \mathrm{t}_{\alpha/2}(\mathrm{n}-\mathrm{m}) \cdot \sigma_y\right], \tag{4.15}$$

for a risk level α, the probability that the (y) falls inside the confidence interval is:

$$\mathrm{P}\left\{\hat{y} - \mathrm{t}_{\alpha/2}(\mathrm{n}-\mathrm{m}) \cdot \sigma_y \le y \le \hat{y} + \mathrm{t}_{\alpha/2}(\mathrm{n}-\mathrm{m}) \cdot \sigma_y\right\} = 1-\alpha, \tag{4.16}$$

where $\mathrm{t}_{\alpha/2}(\mathrm{n}-\mathrm{m})$ is the critical value of *t*-distribution with (***n-m***) degrees of freedom at risk level $\alpha/2$

Choosing a risk level 5%, we have:

$$\mathrm{t}_{\alpha/2}(60-8) = 2.008, \quad \hat{\sigma}_{\hat{y}} = 2.138, \quad \hat{y} = 5.416$$
$$\hat{y} - \mathrm{t}_{\alpha/2}(\mathrm{n}-\mathrm{m}) \cdot \hat{\sigma}_{\hat{y}} = 1.124$$
$$\hat{y} + \mathrm{t}_{\alpha/2}(\mathrm{n}-\mathrm{m}) \cdot \hat{\sigma}_{\hat{y}} = 9.709$$

Thus, we can say with 95% confidence that (y) lies in the interval $[1.124 \ldots 9.709]$ and the vertical scale factor is ***significantly different*** from the scale factor of the horizontal components, and *the 8-parameter model corresponds to our expectation*.

Table 4.5: Residuals in the local topocentric coordinate system when its origin is considered the point itself after a conventional transformation model

Model / Point	7-parameters			8-parameters			9-parameters		
	North [m]	East [m]	Up [m]	North [m]	East [m]	Up [m]	North [m]	East [m]	Up [m]
1	0.084	0.049	0.161	0.085	0.049	0.126	0.088	0.053	0.126
2	-0.070	0.205	0.018	-0.077	0.204	-0.095	-0.079	0.205	-0.094
3	0.048	0.068	0.021	0.048	0.068	-0.004	0.047	0.070	-0.004
4	-0.047	-0.011	-0.246	-0.046	-0.010	-0.221	-0.045	-0.005	-0.221
5	-0.003	0.322	0.139	0.010	0.328	0.006	0.013	0.328	0.006
6	-0.021	-0.117	-0.177	-0.021	-0.116	-0.115	-0.020	-0.113	-0.115
7	0.021	-0.095	-0.030	0.024	-0.096	0.016	0.021	-0.104	0.016
8	0.021	-0.090	-0.065	0.022	-0.092	0.003	0.020	-0.095	0.003
9	0.094	0.015	0.055	0.096	0.015	0.076	0.094	0.011	0.077
10	-0.041	0.008	0.093	-0.044	0.007	0.025	-0.043	0.015	0.025
11	0.074	0.139	0.010	0.073	0.139	-0.032	0.070	0.134	-0.031
12	-0.040	-0.056	-0.063	-0.042	-0.058	-0.053	-0.041	-0.064	-0.053
13	0.001	-0.120	-0.150	-0.001	-0.122	-0.077	-0.001	-0.124	-0.077
14	-0.002	-0.104	-0.005	-0.003	-0.104	0.061	-0.001	-0.098	0.061
15	-0.013	-0.117	-0.234	-0.016	-0.120	-0.182	-0.016	-0.124	-0.182
16	0.050	-0.068	0.093	0.046	-0.069	0.121	0.049	-0.064	0.121
17	0.018	0.016	0.191	0.020	0.016	0.180	0.015	0.004	0.180
18	-0.064	0.005	0.122	-0.064	0.006	0.107	-0.062	0.015	0.107
19	0.053	-0.091	0.104	0.049	-0.091	0.155	0.053	-0.084	0.155
20	-0.174	0.040	-0.037	-0.171	0.043	-0.099	-0.169	0.036	-0.099
RMS	0.063	0.117	0.127	**0.063**	**0.118**	**0.111**	0.063	0.118	0.111
MAX	0.094	0.322	0.191	0.096	0.328	0.180	0.094	0.328	0.180
MIN	-0.174	-0.120	-0.246	-0.171	-0.122	-0.221	-0.169	-0.124	-0.221

The above table indicates that for a conventional 8-parameter and 9-parameter transformation model the residuals have a maximum distortion in East component of **0.328** metres which is slightly different from the maximum distortion in the 7-parameter model, but not very significant. Furthermore, all three models considered in the analysis show more or less the same values for the RMS in all three components, and the new models does not bring any improvements as far as the residuals are concerned.

4.3 Optimal weighting

In practice, the accuracy of the local vertical component is low. Not taking this fact into account might be hazardous when using Eq.(4.7) in compute the parameters of the transformation. Therefore the approach is to estimate the parameters of the tree-dimensional transformation without letting the vertical positions of the local datum influence the fitting process. To achieve this we shall assign appropriate weights to the coordinates of all common points used in the least square adjustment model.

The weighting approach is also favourable in another way as it makes it possible to take into consideration the fact that triangulation points often have heights of poor accuracy, while the benchmarks might have horizontal coordinates from the digitizing with accuracy not better than 5-10 metres [Reit 1998].

For the investigations and analysis of the latter approach the residuals vector given in Eq.(4.8) has been computed for seven different situations of weights for vertical component: 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} and 10^{-6}.

For this investigation a residual horizontal vector ($\varepsilon_{\mathrm{Horiz}}$) is also computed and taken into account in the analysis process, based one the following equation:

$$\varepsilon_{\mathrm{Horiz}} = \sqrt{\varepsilon_{\mathrm{N}}^{2} + \varepsilon_{\mathrm{E}}^{2}}\ , \tag{4.17}$$

and its orientation in the local levelled plane for each point:

$$\alpha_{\varepsilon_{\mathrm{Horiz}}} = \arctan \frac{\varepsilon_{\mathrm{E}}}{\varepsilon_{\mathrm{N}}} \tag{4.18}$$

One can see (from Table 4.6) that the distribution of the values of the horizontal vector with respect to variance of the vertical datum has a decreasing tendency and the minimum RMS of the horizontal vector is somewhere between the interval values [400 … 1300] of variance of the vertical datum.

Table 4.6: RMS of the residuals for each component with respect to different values of the variance of the vertical datum

Variance \ RMS	North [m]	East [m]	Horiz [m]	Up [m]
1	0.063	0.118	0.068	0.111
10	0.059	0.110	0.065	0.153
100	0.044	0.068	0.042	0.564
200	0.043	0.054	0.035	0.739
300	0.044	0.047	0.034	0.826
400	0.045	0.044	0.033	0.880
500	0.046	0.042	0.033	0.917
600	0.047	0.040	0.033	0.944
700	0.047	0.039	0.033	0.965
800	0.048	0.038	0.033	0.982
900	0.048	0.038	0.033	0.997
1000	0.049	0.037	0.033	1.009
1100	0.049	0.037	0.033	1.019
1200	0.049	0.036	0.033	1.028
1300	0.050	0.036	0.033	1.036
1400	0.050	0.036	0.034	1.043
2000	0.051	0.035	0.034	1.074
5000	0.053	0.033	0.036	1.131
10000	0.053	0.033	0.036	1.157
100000	0.054	0.033	0.037	1.184
1000000	0.054	0.033	0.037	1.187

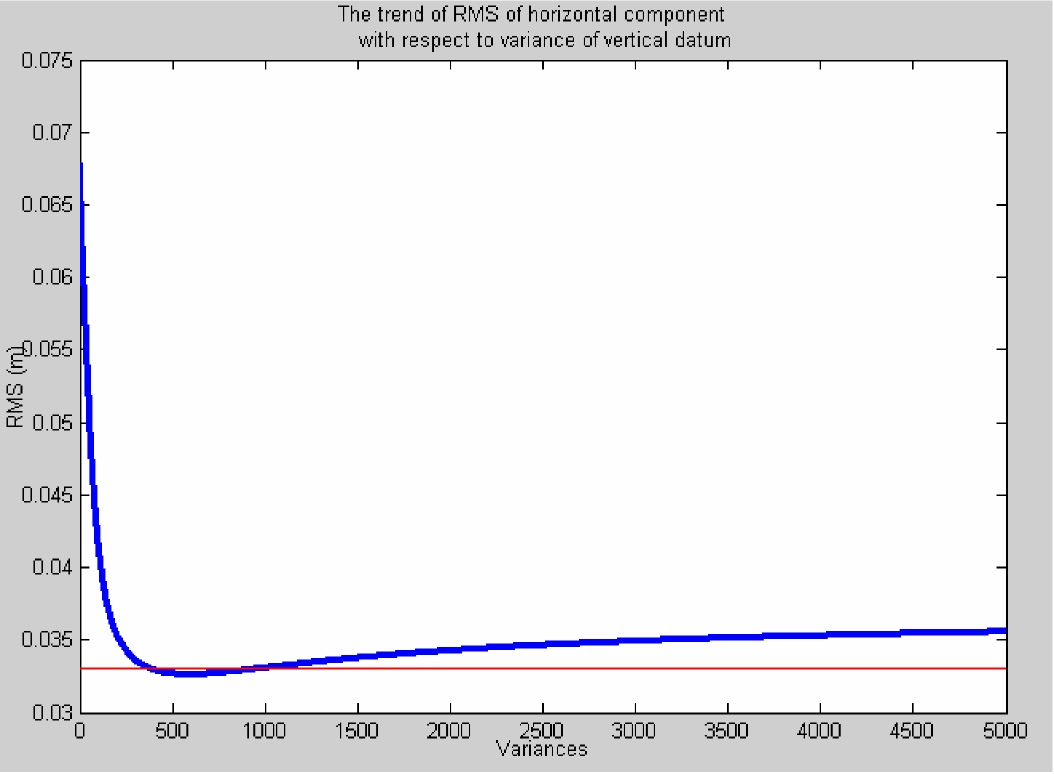

Figure 4.3: The trend of RMS of the horizontal vector residual when different weights are used for the vertical datum in the least squares adjustment

The numerical computation shows a minimum RMS of the horizontal vector of *0,033m* and can be obtained when the variance of vertical datum is between interval [400 … 1300] times bigger than the one for the horizontal datum. Speaking in another way if the a priori standard deviation for each of the two horizontal components is 1cm a minimum RMS can be obtained when the standard deviation of the vertical component is between 18 and 36cm.

According to this, an optimal horizontal fitting model has been computed for the same stations. In this computation the weights used have been based on a priori standard deviation of 0.01metres for each of the two horizontal components and 0.25metres for the vertical component.

Table 4.7: Residual after an optimal 8-parameters transformation model has been carried out

Point	North [m]	East [m]	Horiz [m]	Up [m]
1	0.062	-0.007	0.062	0.628
2	-0.003	0.049	0.049	0.433
3	0.081	0.044	0.092	0.434
4	-0.019	0.059	0.062	0.650
5	0.060	0.123	0.137	0.127
6	-0.032	-0.012	0.034	0.282
7	-0.042	-0.025	0.049	-1.146
8	-0.015	0.010	0.018	-0.681
9	0.058	0.041	0.070	-0.463
10	0.083	-0.026	0.087	1.540
11	0.034	0.068	0.076	-0.614
12	0.015	-0.013	0.020	-1.081
13	-0.020	-0.011	0.023	-0.580
14	-0.024	0.014	0.028	0.765
15	-0.005	-0.026	0.027	-1.062
16	0.011	-0.021	0.024	0.694
17	-0.098	0.009	0.098	-1.482
18	0.019	0.055	0.058	1.635
19	0.010	0.010	0.014	1.090
20	-0.052	-0.003	0.052	-1.169
RMS	**0.047**	**0.040**	**0.033**	0.950
MAX	0.083	0.123	0.137	1.635
MIN	-0.098	-0.026	0.014	-1.482

Compared the above results with results of Table 4.5 for "*conventional*" 8-parameters approach, the horizontal residuals have been significantly reduced at the cost of increasing vertical residuals.

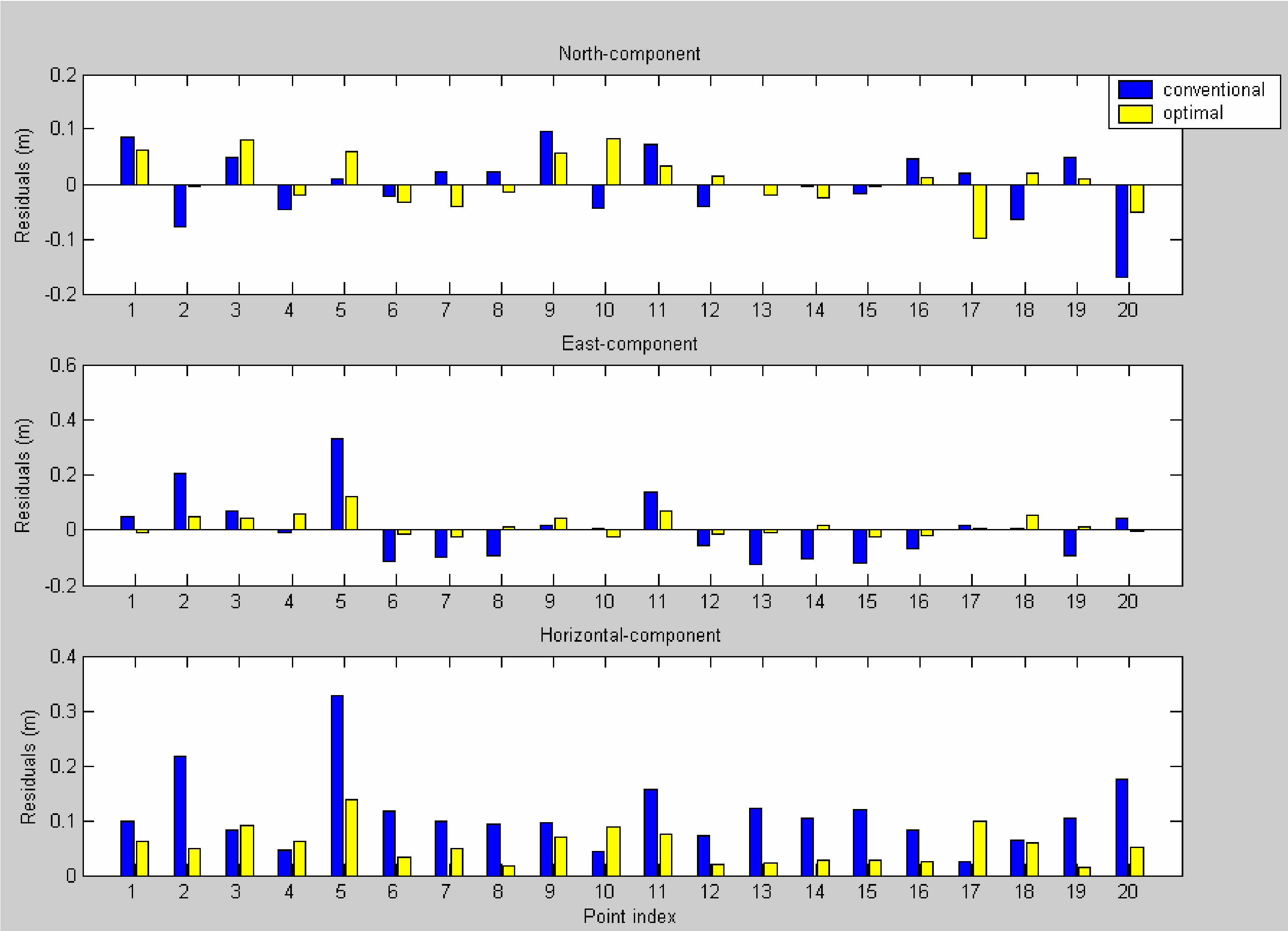

Figure 4.4: North, East and Horizontal components of the residuals of the common transformation points for a *"conventional"* 8-parameter model and an *"optimal"* 8-parameter model, respectively

Conclusions

A three-dimensional (3D) coordinate transformation, combining axes rotations, scale factors change, and origin shift is a practical mathematical model of the relationships between different 3D coordinate systems. Widely used in surveying and related professions, 3D coordinate transformations are usually conformal, that means the transformations preserve angles and the shape of the objects and supposes a simple change of location and orientation and a uniform change in scale.

A new method has been proposed using two different scale factors: one for horizontal component and another one for the vertical component.

A general mathematically algorithm has been developed concerning 3D transformation coordinates on the assumption that the transformation on a 3D object will consist of changes in location (three translations), orientation (three rotations) and shape and size (one scale factor along each axis). The 9-parameter transformation has been computed by a least square adjustment method. All the formula and the equations have been derived. Based on general model under constrain that the scale factor for ***x-*** and *y-* axis are equal, the *Affine model with 8-parameters* has been derived.

The numerical investigation has revealed that the scale factor for horizontal vector differs significantly from by the one for vertical component.

Furthermore the general transformation model with 9-parameter has confirmed our assumption that the horizontal components have similar scale factor.

Using the direct results of the adjustment in geocentric coordinates does not offer any clear meaning of all these results. To achieve this, the expressions in a topocentric reference system (north, east, up) with the origin of the reference system in each point itself of the residuals have been determined. Some numerical investigations have been carried out assigning different weights in the least square adjustment for vertical component. The analyses on all these alternatives reveal that for a conventional 8-parameter transformation model (the weight has been considered equal to the unit for both horizontal and vertical components) the residuals indicate a maximum distortion in East component of 0.328 metres that is slightly different by the classical model with 7-parameters for which is 0.322 metres.

Another remark is that by increasing the variance for the vertical (up) component (σ_u^2), one can decrease the residual for the horizontal component (ε_h), but at the cost of increasing vertical residuals.

Then the distribution of the values of RMS of the residuals of the horizontal vector with respect to the assigned weight of the vertical datum has a minimum RMS of 3.3cm, when the ratio between the assigned weight of vertical datum and horizontal datum is between 400 and 1300. According to this, an optimal horizontal model has been computed for a priori standard deviation of 1 cm for each of the two horizontal components of the horizontal vector and 25cm for the vertical component. The results show a considerable reduction of the horizontal residuals compared to the conventional 8-parameters approach from 11.1cm ("*conventional*" approach) to 3.3cm ("*optimal*" approach).

When computing the transformation parameters between two three-dimensional coordinates systems it is important to only use points with known accuracy that is to be sure that no gross errors are presented in the coordinates. The number of the control points used in the adjustment must be large enough to ensure that local distortions stay local and do not have a bad influence on the estimated transformation parameters.

References

[1] **Andrei C.O.** (2005). Analysis of geodetic networks using simulated observations. Project work. Department of Geodesy, Royal Institute of Technology, Stockholm.

[2] **Badekas J.** (1969). Investigations Related to the Establishment of a World Geodetic System. Report No.124, Department of Geodetic Science and Surveying, Ohio State University, USA.

[3] **Bellman C., Anderson L.** (1995). Close range photogrammetry for the dimensional control in shipbuilding. In: Shortis M.R., and C.L. Ogleby (eds), SAM'95: Proceedings of the 3rd Symposium on Surveillance and Monitoring Surveys, Melbourne, Australia, November 1-2, 1995. Department of Geomatics, University of Melbourne.pp.1-8.

[4] **Chen X., Du J., Doihara T.** (2000). Generalization of cadastral map based on graphics matching. Geoinformatics 2000, USA, pp. 40-45.

[5] **Deakin R.E.** (1998). 3D coordinates transformations. Surveying and Land Information Systems, Vol.58, No.4, 1998, pp.223-234.

[6] **Fan H.** (1997). Theory of errors and least squares adjustment. TRITA GEOFOTO 1997:21. Department of Geodesy and Photogrammetry, Royal Institute of Technology, Stockholm, October 1997.

[7] **Fan H.** (2000). Theoretical Geodesy. TRITA GEOFOTO 2000:23. Department of Geodesy and Photogrammetry, Royal Institute of Technology, Stockholm, October 2000.

[8] **Fan H.** (2005). Three-Dimensional Coordinate Transformation with large rotations and scale change. International Workshop on Education in Geospatial Information Technology, 27-28 October, Technical University of Moldova, Chisinau, Moldova.

[9] **Fraser C.S., Yamakawa T.** (2003). Applicability of the affine model for Ikonos image orientation over mountainous terrain. Workshop on HRM from Space, 6-8 October, Hanover, 6p.

[10] **Harvey B.R**. (1986). Transformation of 3D co-ordinates. The Australian Surveyor, June, Vol.33, No.2, pp.105-125.

[11] **Hugli H., Schutz C.** (1997). Geometric Matching of 3D Objects: Assessing the Range of Successful Initial Configurations. 3dim, p.101, First International Conference on Recent Advances in 3D Digital Imaging and Modelling (3DIM'97).

[12] **Leu., I. N., Budiu, V., Moca, V., Ritt, C., Ciolac, Valeria, Ciotlaus, Ana, Negoescu, I.,** (2003), Topografie generala si aplicata – Cadastru (in romanian), Editura Universul, Bucuresti, 600 pag., ISBN 973-9027-64-4.

[13] **Jivall L.,** (2001). SWEREF 99 – New ETRS 89 Coordinates in the Sweden. LMV Reports in Geodesy och Geographical Information Systems 2001:6, Gävle.

[14] **Jivall L., Lidberg M., Lilje M., Reit B.G.** (2001). Transformationssamband mellan SWEREF 99 och RT 90/RH 70. LMV Reports in Geodesy och Geographical Information Systems 2001:7, Gävle.

[15] **Mackie J.B.** (1982). The relationship between the WGS72 Doppler satellite datum and the New Zealand Geodetic Datum 1949. Report No.178, Geophysics Division, Department of Scientific and Industrial Research, New Zealand.

[16] **Maes F., Collignon A., Vandermeulen D., Marchal G., Suetens P.** (1997). Multimodality image registration by maximization of mutual information, IEEE transactions on Medical Imaging, vol. 16, no. 2, pp. 187-198, April 1997.

[17] **Mikhail E.M.** (1976). Observations and least squares. New York, New York: IEP/A. Dun-Donelley.

[18] **Mikhail E.M., Bethel J.S., McGlone J.C.** (2001). Introduction to Modern Photogrammetry, John Wiley and Sons, Inc., New York.

[19] **Moritz H., Hofmann-Wellenhof B.** (2005). Physical Geodesy. Springer, WienNewYork.

[20] **Niederöst J.** (2001). 3D reconstruction and accuracy analysis of historical relief models from the 18th century. 3rd International Image Sensing Seminar on New Development in Digital Photogrammetry, Gifu, Japan, 24+27 September 2001.

[21] **Reit B.G.** (1998). The 7-parameter transformation to a horizontal geodetic datum. Survey Review, 34(268): 400-404.

[22] **Reit B.G.** (1999). Improving a horizontal datum without changing the coordinates. Survey Review, 35(272): 400-404.

[23] **Yang Y.** (1999). Robust estimation of geodetic datum transformation. Journal of Geodesy, 73(1999): 268-274.

[24] http://www.lm.se

[25] http://www.linz.govt.nz/rcs/linz/6081/recommended_transformation_parameters.pdf

Appendices

Appendice A – North, East and Up components of the residuals of the common transformation points, using different weighting approach for the vertical datum in least square adjustment model

P = 1

Point	Nord[m]	East[m]	Up[m]
1	0.085	0.049	0.126
2	-0.077	0.204	-0.095
3	0.048	0.068	-0.004
4	-0.046	-0.010	-0.221
5	0.010	0.328	0.006
6	-0.021	-0.116	-0.115
7	0.024	-0.096	0.016
8	0.022	-0.092	0.003
9	0.096	0.015	0.076
10	-0.044	0.007	0.025
11	0.073	0.139	-0.032
12	-0.042	-0.058	-0.053
13	-0.001	-0.122	-0.077
14	-0.003	-0.104	0.061
15	-0.016	-0.120	-0.182
16	0.046	-0.069	0.121
17	0.020	0.016	0.180
18	-0.064	0.006	0.107
19	0.049	-0.091	0.155
20	-0.171	0.043	-0.099
RMS	0.063	0.118	0.068
MAX	0.096	0.328	0.180
MIN	-0.171	-0.122	-0.221

P = 10^{-1}

Point	Nord[m]	East[m]	Up[m]
1	0.085	0.049	0.126
2	-0.077	0.204	-0.095
3	0.048	0.068	-0.004
4	-0.046	-0.010	-0.221
5	0.010	0.328	0.006
6	-0.021	-0.116	-0.115
7	0.024	-0.096	0.016
8	0.022	-0.092	0.003
9	0.096	0.015	0.076
10	-0.044	0.007	0.025
11	0.073	0.139	-0.032
12	-0.042	-0.058	-0.053
13	-0.001	-0.122	-0.077
14	-0.003	-0.104	0.061
15	-0.016	-0.120	-0.182
16	0.046	-0.069	0.121
17	0.020	0.016	0.180
18	-0.064	0.006	0.107
19	0.049	-0.091	0.155
20	-0.171	0.043	-0.099
RMS	0.059	0.110	0.065
MAX	0.092	0.307	0.261
MIN	-0.160	-0.111	-0.256

$P = 10^{-2}$

Point	Nord[m]	East[m]	Up[m]
1	0.085	0.049	0.126
2	-0.077	0.204	-0.095
3	0.048	0.068	-0.004
4	-0.046	-0.010	-0.221
5	0.010	0.328	0.006
6	-0.021	-0.116	-0.115
7	0.024	-0.096	0.016
8	0.022	-0.092	0.003
9	0.096	0.015	0.076
10	-0.044	0.007	0.025
11	0.073	0.139	-0.032
12	-0.042	-0.058	-0.053
13	-0.001	-0.122	-0.077
14	-0.003	-0.104	0.061
15	-0.016	-0.120	-0.182
16	0.046	-0.069	0.121
17	0.020	0.016	0.180
18	-0.064	0.006	0.107
19	0.049	-0.091	0.155
20	-0.171	0.043	-0.099
RMS	0.044	0.068	0.042
MAX	0.071	0.206	0.954
MIN	-0.105	-0.067	-0.863

$P = 10^{-3}$

Point	Nord[m]	East[m]	Up[m]
1	0.085	0.049	0.126
2	-0.077	0.204	-0.095
3	0.048	0.068	-0.004
4	-0.046	-0.010	-0.221
5	0.010	0.328	0.006
6	-0.021	-0.116	-0.115
7	0.024	-0.096	0.016
8	0.022	-0.092	0.003
9	0.096	0.015	0.076
10	-0.044	0.007	0.025
11	0.073	0.139	-0.032
12	-0.042	-0.058	-0.053
13	-0.001	-0.122	-0.077
14	-0.003	-0.104	0.061
15	-0.016	-0.120	-0.182
16	0.046	-0.069	0.121
17	0.020	0.016	0.180
18	-0.064	0.006	0.107
19	0.049	-0.091	0.155
20	-0.171	0.043	-0.099
RMS	0.049	0.037	0.033
MAX	0.093	0.114	1.753
MIN	-0.101	-0.026	-1.522

$P = 10^{-4}$

Point	Nord[m]	East[m]	Up[m]
1	0.085	0.049	0.126
2	-0.077	0.204	-0.095
3	0.048	0.068	-0.004
4	-0.046	-0.010	-0.221
5	0.010	0.328	0.006
6	-0.021	-0.116	-0.115
7	0.024	-0.096	0.016
8	0.022	-0.092	0.003
9	0.096	0.015	0.076
10	-0.044	0.007	0.025
11	0.073	0.139	-0.032
12	-0.042	-0.058	-0.053
13	-0.001	-0.122	-0.077
14	-0.003	-0.104	0.061
15	-0.016	-0.120	-0.182

16	0.046	-0.069	0.121
17	0.020	0.016	0.180
18	-0.064	0.006	0.107
19	0.049	-0.091	0.155
20	-0.171	0.043	-0.099
RMS	0.053	0.033	0.036
MAX	0.119	0.099	2.121
MIN	-0.102	-0.019	-1.587

P = 10^{-5}

Point	Nord[m]	East[m]	Up[m]
1	0.085	0.049	0.126
2	-0.077	0.204	-0.095
3	0.048	0.068	-0.004
4	-0.046	-0.010	-0.221
5	0.010	0.328	0.006
6	-0.021	-0.116	-0.115
7	0.024	-0.096	0.016
8	0.022	-0.092	0.003
9	0.096	0.015	0.076
10	-0.044	0.007	0.025
11	0.073	0.139	-0.032
12	-0.042	-0.058	-0.053
13	-0.001	-0.122	-0.077
14	-0.003	-0.104	0.061
15	-0.016	-0.120	-0.182
16	0.046	-0.069	0.121
17	0.020	0.016	0.180
18	-0.064	0.006	0.107
19	0.049	-0.091	0.155
20	-0.171	0.043	-0.099
RMS	0.054	0.033	0.037
MAX	0.123	0.097	2.201
MIN	-0.101	-0.018	-1.645

P = 10^{-6}

Point	Nord[m]	East[m]	Up[m]
1	0.085	0.049	0.126
2	-0.077	0.204	-0.095
3	0.048	0.068	-0.004
4	-0.046	-0.010	-0.221
5	0.010	0.328	0.006
6	-0.021	-0.116	-0.115
7	0.024	-0.096	0.016
8	0.022	-0.092	0.003
9	0.096	0.015	0.076
10	-0.044	0.007	0.025
11	0.073	0.139	-0.032
12	-0.042	-0.058	-0.053
13	-0.001	-0.122	-0.077
14	-0.003	-0.104	0.061
15	-0.016	-0.120	-0.182
16	0.046	-0.069	0.121
17	0.020	0.016	0.180
18	-0.064	0.006	0.107
19	0.049	-0.091	0.155
20	-0.171	0.043	-0.099
RMS	0.054	0.033	0.037
MAX	0.124	0.097	2.210
MIN	-0.101	-0.017	-1.652

Appendice B – General algorithm of 3D coordinate transformation with all the subroutines

```
function [L, A, parameters, SE_cap, epsilon_cap, CXX, Cee, CLL] =
GeneralHelmert(Sist1, Sist2, scale, rotation, C, no_param)

% compute the transformation parameters of a general 3D transformation
% model
% Sist1 = rectangular coordinates of the first system (fix)
% Sist2 = rectangular coordinates of the second system (transformed)
% both should have the same number of points and the following header
%               |Index      X       Y       Z|
% scale = scale factors vector (column vector)
% rotation = rotations angles around each axis (column vector)
% C = the apriori information concerning the coordinates of the points
% no_param = number of the parameters model (can be 7,8 or 9)

[L, A, dX_cap, parameters] = ParamIter(Sist1, Sist2, scale, rotation, C,
no_param);

% compute the a-posteriori information
[SE_cap, epsilon_cap, CXX, Cee, CLL] = VarCovMatrices(L, A, dX_cap, C,
no_param);

function [L, A, dX_cap, parameters] = ParamIter(Sist1, Sist2, scale,
rotation, C, no_param)

% compute the transformation parameters, design matrix and the observation
% vector for a Affine transformation using THREE different scale factors
% and a REITERATIVE process


% compute the ro-number
ro = 180 * 60 * 60 / pi;

% generate the design matrix and observation vector
[L, A] = DesignMatrix(Sist1, Sist2, scale, rotation);

%compute the approximated parameters
[dX_cap] = Parameters(L, A, C, no_param);

% compute the update values of the scale factors and rotation angles
scale_new = scale + dX_cap(4:6) .* 1e-6;
rotation_new = rotation + dX_cap(7:9) ./ ro;

% count the reiterations
count = 0;

% reiteration process
while count <= 5 %( dX_cap(4) >= 1e-20 & dX_cap(5) >= 1e-20 & dX_cap(6) >=
1e-20)
    count = count+1;

    % asign the new values
    scale = scale_new;
    rotation = rotation_new;

    % run the algotithm for the update values
    [L, A] = DesignMatrix(Sist1, Sist2, scale, rotation);
    [dX_cap] = Parameters(L, A, C, no_param);

    % compute the new values for the scale factors and rotation angles
```

```
    scale_new = scale + dX_cap(4:6) .* 1e-6;
    rotation_new = rotation + dX_cap(7:9) ./ ro;
end

% compute the total correction of the scale factors
parameters = [dX_cap(1:3); (scale_new - ones(3,1))*1e+6; rotation_new*ro];


function [L, A] = DesignMatrix(Sist1, Sist2, scale, rotation)

% compute the design matrix and the observation vector for an Affine
% transformation using THREE different factor scale


if size(Sist1,1) ~= size(Sist2,1)
    disp(sprintf('YOU   DO   NOT   HAVE   THE   SAME   NUMBER   OF   COMMON
POINTS'))
    %break
else
    no_p = size(Sist1,1);
end

% computation of ro-number
ro = 180 * 60 * 60 / pi;

% the starting values for scale factors
s1_0 = scale(1); s2_0 = scale(2); s3_0 = scale(3);

% the starting values for rotation angles in RADIANS
alfa1 = rotation(1); alfa2 = rotation(2); alfa3 = rotation(3);


for i = 1 : no_p
    X1(i,1) = Sist1(i,2);
    Y1(i,1) = Sist1(i,3);
    Z1(i,1) = Sist1(i,4);

    X2(i,1) = Sist2(i,2);
    Y2(i,1) = Sist2(i,3);
    Z2(i,1) = Sist2(i,4);

    % compute the rotation matrix
    R = RotMatrix(alfa1, alfa2, alfa3);

    % the reduced observations
    L(3*(i-1)+1,1) = X2(i,1) - (s1_0 * R(1,1) * X1(i,1) + s2_0 * R(1,2) *
Y1(i,1) + s3_0 * R(1,3) * Z1(i,1));
    L(3*(i-1)+2,1) = Y2(i,1) - (s1_0 * R(2,1) * X1(i,1) + s2_0 * R(2,2) *
Y1(i,1) + s3_0 * R(2,3) * Z1(i,1));
    L(3*(i-1)+3,1) = Z2(i,1) - (s1_0 * R(3,1) * X1(i,1) + s2_0 * R(3,2) *
Y1(i,1) + s3_0 * R(3,3) * Z1(i,1));

    % unknown dx
    A(3*(i-1)+1,1) = 1;
    A(3*(i-1)+2,1) = 0;
    A(3*(i-1)+3,1) = 0;

    % unknown dy
    A(3*(i-1)+1,2) = 0;
    A(3*(i-1)+2,2) = 1;
    A(3*(i-1)+3,2) = 0;

    % unknown dz
    A(3*(i-1)+1,3) = 0;
```

```
    A(3*(i-1)+2,3) = 0;
    A(3*(i-1)+3,3) = 1;

    % unknown ds1
    A(3*(i-1)+1,4) = R(1,1) * X1(i,1) / (1e+6);
    A(3*(i-1)+2,4) = R(2,1) * X1(i,1) / (1e+6);
    A(3*(i-1)+3,4) = R(3,1) * X1(i,1) / (1e+6);

    % unknown ds2
    A(3*(i-1)+1,5) = R(1,2) * Y1(i,1) / (1e+6);
    A(3*(i-1)+2,5) = R(2,2) * Y1(i,1) / (1e+6);
    A(3*(i-1)+3,5) = R(3,2) * Y1(i,1) / (1e+6);

    % unknown ds3
    A(3*(i-1)+1,6) = R(1,3) * Z1(i,1) / (1e+6);
    A(3*(i-1)+2,6) = R(2,3) * Z1(i,1) / (1e+6);
    A(3*(i-1)+3,6) = R(3,3) * Z1(i,1) / (1e+6);

    % unknown dalfa1
    A(3*(i-1)+1,7) = (-s2_0 * R(1,3) * Y1(i,1) + s3_0 * R(1,2) * Z1(i,1)) /
ro;
    A(3*(i-1)+2,7) = (-s2_0 * R(2,3) * Y1(i,1) + s3_0 * R(2,2) * Z1(i,1)) /
ro;
    A(3*(i-1)+3,7) = (-s2_0 * R(3,3) * Y1(i,1) + s3_0 * R(3,2) * Z1(i,1)) /
ro;

    % unknown alfa2
    A(3*(i-1)+1,8) = -cos(alfa3) * (s1_0 * sin(alfa2) * X1(i,1) + s2_0 *
R(3,2) * Y1(i,1) + s3_0 * R(3,3) * Z1(i,1)) / ro;
    A(3*(i-1)+2,8) = sin(alfa3) * (s1_0 * sin(alfa2) * X1(i,1) + s2_0 *
R(3,2) * Y1(i,1) + s3_0 * R(3,3) * Z1(i,1)) / ro;
    A(3*(i-1)+3,8) = (s1_0 * cos(alfa2) * X1(i,1) + s2_0 * sin(alfa1) *
sin(alfa2) * Y1(i,1) - s3_0 * cos(alfa1) * sin(alfa2) * Z1(i,1)) / ro;

    % unknown alfa3
    A(3*(i-1)+1,9) = (s1_0 * R(2,1) * X1(i,1) + s2_0 * R(2,2) * Y1(i,1) +
s3_0 * R(2,3) * Z1(i,1)) / ro;
    A(3*(i-1)+2,9) = -(s1_0 * R(1,1) * X1(i,1) + s2_0 * R(1,2) * Y1(i,1) +
s3_0 * R(1,3) * Z1(i,1)) / ro;
    A(3*(i-1)+3,9) = 0 / ro;

end

function [dX_cap] = Parameters(L, A, C, no_param)

% compute the normal matrix
N = A' * inv(C) * A;

% the constraints
if (no_param == 7) | (no_param == 8)
    if no_param == 7
        Ax = [0 0 0 1 -1 0 0 0 0;
            0 0 0 1 0 -1 0 0 0];
        d = [0; 0];
    elseif no_param == 8
        Ax = [0 0 0 1 -1 0 0 0 0];
        d = [0];
    end

    % compute the Lagrangian multipliers
    Nx = Ax * inv(N) * Ax';
    lamda = inv(Nx) * (Ax * inv(N) * A' * inv(C) * L - d);

    % compute the unknown parameters
```

```
    dX_cap = inv(N) * A' * inv(C) * L - inv(N) * Ax' * lamda;

elseif no_param == 9

    % compute the unknown parameters
    dX_cap = inv(N) * A' * inv(C) * L;

else
    disp(sprintf('YOU   DO   NOT   HAVE   THE   SAME   NUMBER   OF   COMMON
POINTS'))
end

function [SE_cap, epsilon_cap, CXX, Cee, CLL] = VarCovMatrices(L, A, dX_cap,
C, no_param)

% compute the RESIDUALS for the coordinates and the VARIANCE-COVARIANCE
% matrices of the least squares estimate (unknowns,residuals and
% observations) for a Helmert transformation using THREE different factor
% scales


% compute the ro-number
ro = 180 * 60 * 60 / pi;

% compute the normal matrix
N = A' * inv(C) * A;

% the constraints
if (no_param == 7) | (no_param == 8)
    if no_param == 7
        Ax = [0 0 0 1 -1 0 0 0 0;
            0 0 0 1 0 -1 0 0 0];
        d = [0; 0];
    elseif no_param == 8
        Ax = [0 0 0 1 -1 0 0 0 0];
        d = [0];
    end

    % compute the Lagrangian multipliers
    Nx = Ax * inv(N) * Ax';
end

% compute the estimated residuals
epsilon_cap = L - A * dX_cap;

% number of observations, unknowns and number of over-determinations
n = size(L,1);
m = size(dX_cap,1);
if (no_param == 7) | (no_param == 8)
    r = length(d);
    t = n - m  + r;
elseif no_param == 9
    t = n - m;
end

% the a posteriori unit-weigth standard error
SE_cap = sqrt( epsilon_cap' * inv(C) * epsilon_cap / t );

% the variance-covariance of the unknowns dX_cap
if (no_param == 7) | (no_param == 8)
    QXX = inv(N) - inv(N) * Ax' * inv(Nx) * Ax * inv(N);
elseif no_param == 9
    QXX = inv(N);
else
```

```
    disp(sprintf('YOU   DO   NOT   HAVE   THE   SAME   NUMBER   OF   COMMON
POINTS'))
end

CXX = SE_cap^2 * QXX;

% the variance-covariance matrix of the adjusted observations L_cap
CLL = A * CXX * A';

% the variance-covariance of the estimated residuals epsilon_cap
Cee = SE_cap^2 * ( inv(C) - A * QXX * A' );
```

Reports in Geographic Information Technology

The TRITA-GIT Series, ISSN 1653-5227

06-001 **Uliana Danila**. Corrective *surface for GPS-levelling in Moldova.* Master of Science thesis in geodesy No. 3089. Supervisor: Lars Sjöberg. TRITA-GIT EX 06-001. January 2006.

06-002 **Ingemar Lewén**. *Use of gyrotheodolite in underground control network*. Master of Science thesis in geodesy No. 3090. Master of Science thesis in geodesy. Supervisor: Erick Asenjo. TRITA-GIT EX 06-002. January 2006.

06-003 **Johan Tornberg**. *Felfortplantningsanalys i GIS-projekt genom Monte Carlo-simulering*. Master of Science thesis in geoinformatics. Supervisor: Mats Dunkars. TRITA-GIT EX 06-003. February 2006.

06-004 **Constantin-Octavian Andrei**. 3D affine coordinate transformations. Master of Science thesis in geodesy No. 3091. Master of Science thesis in geodesy. Supervisor: Huaan Fan. TRITA-GIT EX 06-004. March 2006.

TRITA-GIT EX 06-004
ISSN 1653-5227
ISRN KTH/GIT/EX--06/004-SE