Computing Helmert transformations

G. A. Watson,
Department of Mathematics,
University of Dundee,
Dundee DD1 4HN, Scotland.

Abstract

The Helmert transformation is used in geodesy. It transforms a set of points into another by rotation, scaling and translation. When both sets of points are given, then least squares can be used to solve the inverse problem of determining the parameters. In particular, the parameters of the so-called 7 parameter transformation can be obtained by standard methods. In this note, it is shown how a Gauss-Newton method in the rotation parameters alone can easily be implemented to determine the parameters of the 9 parameter transformation (when different scale factors for the variables are needed).

Keywords Helmert transformation, least squares, Gauss-Newton method.

1 Introduction

The Helmert transformation is used in geodesy, which is the science of the measurement and mapping of the earth's surface (see, for example, [2]). It refers to the transformation involved in changing the coordinates of a point set with respect to one reference surface to make them refer to another reference surface, and involves rotation, scaling and translation. It is named after Professor Dr Friedrich Robert Helmert, who lived from 31 July 1843 to 15 June 1917. He was director of the geodetical institute in Potsdam, Germany, from 1886 to 1917, and a Professor at the University of Berlin. He

is described on his memorial stone as the founder of the mathematical and physical theories of modern geodesy [15].

In particular, the so-called 7-parameter Helmert transformation applies to point sets $\mathbf{p}_i, \mathbf{q}_i, i = 1, \ldots, m$, in R^3. So we can write

$$\mathbf{p}_i = dR(\phi)\mathbf{q}_i + \mathbf{t}, i = 1, \ldots, m, \tag{1}$$

where

$$R(\phi) = R_1(\alpha)R_2(\beta)R_3(\gamma),$$

with $\phi = (\alpha, \beta, \gamma)^T$, and where

$$R_1(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$R_2(\beta) = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix},$$

$$R_3(\gamma) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{bmatrix}.$$

In other words, R is a product of 3 elementary rotation matrices in the 3 co-ordinate planes. The remaining parameters are a scaling parameter d and a translation vector $\mathbf{t}$ so that the 7 parameters are $d \in R, \phi \in R^3, \mathbf{t} \in R^3$.

If the parameters are given, then it is possible to transform a given set of points $\mathbf{q}_i, i = 1, \ldots, m$ into another set $\mathbf{p}_i, i = 1, \ldots, m$. Of interest here is the inverse problem: given sets of points $\mathbf{p}_i, \mathbf{q}_i, i = 1, \ldots, m$, determine the parameters. Of course if $m > 7$ there will not normally be an exact fit. Therefore it may be appropriate to find a least squares solution, that is to minimize

$$F = \sum_{i=1}^{m} \|\mathbf{v}_i\|^2,$$

for example, where

$$\mathbf{v}_i = \mathbf{p}_i - dR\mathbf{q}_i - \mathbf{t}, i = 1, \ldots, m,$$

and the norm is the l_2 norm. The underlying problem can also be interpreted mathematically as the problem of fitting a $C_7(3)$ conformal group, that is a 7-parameter group in 3-dimensional space. For a study of this and other transformations in this context, see [6, 7, 8, 9, 10].

The key to efficient solution of this problem is to note that the solution ϕ is independent of d. Therefore the correct ϕ can be obtained by minimizing

$$\sum_{i=1}^{m} \|\mathbf{p}_i - R(\phi)\mathbf{q}_i - \mathbf{t}\|^2, \tag{2}$$

and this can be obtained by the method of Hansen and Norris [11], which uses the singular value decomposition of a 3×3 matrix. The basic requirement is just the solution of an orthogonal Procrustes problem, see for example [1], [4], [5]. It is easy to see by differentiation with respect to d and $\mathbf{t}$ that

$$d = \frac{\sum_{i=1}^{m} \mathbf{p}_i^T R\mathbf{q}_i - m\bar{\mathbf{p}}^T R\bar{\mathbf{q}}}{\sum_{i=1}^{m} \mathbf{q}_i^T \mathbf{q}_i - m\bar{\mathbf{q}}^T \bar{\mathbf{q}}}, \tag{3}$$

$$\mathbf{t} = \bar{\mathbf{p}} - dR\bar{\mathbf{q}}, \tag{4}$$

where

$$\bar{\mathbf{p}} = \frac{\sum_{i=1}^{m} \mathbf{p}_i}{m}, \quad \bar{\mathbf{q}} = \frac{\sum_{i=1}^{m} \mathbf{q}_i}{m}.$$

Thus we can use (3) and (4) to obtain the remaining parameters. So a 7-point Helmert transformation can readily be computed.

An iterative method for (2) which uses (4) (with $d = 1$) but which does not require the use of the singular value decomposition is given by Späth [13]. For fixed ϕ, it obtains $\mathbf{t}$, and for $\mathbf{t}$ fixed, a new value of ϕ is calculated, and so on. This is a descent process which can be interpreted as the alternating algorithm.

A modification of the problem where there are different scale factors on each variable is also considered by Späth in [14], where a method is given analogous to that which he uses for minimizing (2). The definition of $\mathbf{v}_i$ is altered to

$$\mathbf{v}_i = \mathbf{p}_i - DR\mathbf{q}_i - \mathbf{t}, i = 1, \ldots, m, \tag{5}$$

where $D = \text{diag}\{d_1, d_2, d_3\}$. This is now a 9-parameter Helmert transformation problem. Another interpretation of this is as a 9-parameter linear group

general transformation, as it appears in continuum mechanics and navigation, for example. The Hanson-Norris approach no longer applies to this problem.

The main purpose of this note is to point out that the Gauss-Newton method (or variants) can be easily implemented for the 9 parameter problem using separation of variables, and iteration with respect to the rotation parameters alone. Apart from a linear least squares solution, only simple calculations are involved. The method is analogous to other methods for separated least squares problems, which go back at least to Golub and Pereyra [3].

It is of interest to note that sometimes a simpler version of (1) is used where it is assumed that the rotation angles are small so that we can replace R by the simpler matrix

$$S(\phi) = \begin{bmatrix} 1 & -\alpha & -\beta \\ \alpha & 1 & -\gamma \\ \beta & \gamma & 1 \end{bmatrix},$$

(see, for example [12]). For the 6 (or 7) parameter (inverse) problem, this offers no advantage, as S is no longer an orthogonal matrix, for example. However, the method developed in the next section could readily be modified to treat this variant.

2 A numerical method for the 9-parameter problem

For the least squares problem

$$\text{minimize} \quad \sum_{i=1}^{m} \|\mathbf{v}_i\|^2$$

where $\mathbf{v}_i$ is given by (5), then differentiating with respect to $\mathbf{t}$ and $\mathbf{d} = [d_1, d_2, d_3]^T$ is easily shown to lead to systems of linear equations which express the components of $\mathbf{t}$ and $\mathbf{d}$ in terms of ϕ:

$$\begin{bmatrix} \sum_{i=1}^m (\mathbf{e}_j^T R\mathbf{q}_i)^2 & \sum_{i=1}^m \mathbf{e}_j^T R\mathbf{q}_i \\ \mathbf{e}_j^T R\bar{\mathbf{q}} & 1 \end{bmatrix} \begin{bmatrix} d_j \\ t_j \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^m (\mathbf{e}_j^T \mathbf{p}_i)(\mathbf{e}_j^T R\mathbf{q}_i) \\ \mathbf{e}_j^T \bar{\mathbf{p}} \end{bmatrix}, j = 1, 2, 3.$$

Alternatively these can be written as the symmetric systems

$$M_j(\phi) \begin{bmatrix} d_j \\ t_j \end{bmatrix} = \mathbf{r}_j(\phi), j = 1, 2, 3, \tag{6}$$

where for $j = 1, 2, 3$,

$$M_j(\phi) = \begin{bmatrix} \sum_{i=1}^m (\mathbf{e}_j^T R \mathbf{q}_i)^2 & m \mathbf{e}_j^T R \bar{\mathbf{q}} \\ m \mathbf{e}_j^T R \bar{\mathbf{q}} & m \end{bmatrix}, \ \mathbf{r}_j(\phi) = \begin{bmatrix} \sum_{i=1}^m (\mathbf{e}_j^T \mathbf{p}_i)(\mathbf{e}_j^T R \mathbf{q}_i) \\ m \mathbf{e}_j^T \bar{\mathbf{p}} \end{bmatrix}.$$

Then for any given ϕ we can use (6) to give the vectors $\mathbf{t}$ and $\mathbf{d}$ in terms of ϕ. The objective function can then be thought of as a function of ϕ alone, say

$$F(\phi) = \|\mathbf{v}_i\|^2,$$

and we will show how to implement a Gauss-Newton type method in ϕ. The basic Gauss-Newton subproblem requires the solution to the linear least squares problem

$$\text{minimize} \sum_{i=1}^m \|\mathbf{v}_i + J_i \mathbf{z}\|^2, \tag{7}$$

where

$$J_i = \nabla_\phi \mathbf{v}_i, i = 1, \ldots, m.$$

This may be used in conjunction with a line search to force convergence, or it may be modified in a variety of ways (Levenberg-Marquardt, trust region) to give a more robust method (for example [16]), but a crucial requirement is always going to be the efficient calculation of J_i.

Considering $\mathbf{v}_i(\phi, \mathbf{d}, \mathbf{t})$ to show dependence on the 3 sets of parameters, let $\nabla_1, \nabla_2, \nabla_3$ denote the operation of partial differentiation with respect to $\phi, \mathbf{d}, \mathbf{t}$, respectively. Then we can write

$$\nabla_\phi \mathbf{v}_i = \nabla_1 \mathbf{v}_i + \nabla_2 \mathbf{v}_i \nabla_\phi \mathbf{d} + \nabla_3 \mathbf{v}_i \nabla_\phi \mathbf{t}, i = 1, \ldots, m.$$

First consider $\nabla_j \mathbf{v}_i, j = 1, 2, 3$. Define vectors

$$\mathbf{g}_j^{(1)} = R_3^T R_2^T {R_1'}^T \mathbf{e}_j, j = 1, 2, 3,$$

$$\mathbf{g}_j^{(2)} = R_3^T {R_2'}^T R_1^T \mathbf{e}_j, j = 1, 2, 3,$$

$$\mathbf{g}_j^{(3)} = {R_3'}^T R_2^T R_1^T \mathbf{e}_j, j = 1, 2, 3,$$

where the dash denotes differentiation with respect to the single variable on which the matrix depends. Let

$$G_j = [\mathbf{g}_j^{(1)} : \mathbf{g}_j^{(2)} : \mathbf{g}_j^{(3)}], j = 1, 2, 3.$$

Then we have for each $i, i = 1, \ldots, m$,

$$\mathbf{e}_j^T \nabla_1 \mathbf{v}_i = -d_j \mathbf{q}_i^T G_j, j = 1, 2, 3,$$

$$\mathbf{e}_j^T \nabla_2 \mathbf{v}_i = -(\mathbf{e}_j^T R \mathbf{q}_i)\mathbf{e}_j^T, j = 1, 2, 3,$$

$$\nabla_3 \mathbf{v}_i = -I.$$

Next we need to calculate $\nabla_\phi \mathbf{d}$ and $\nabla_\phi \mathbf{t}$. But from (6), regarded as an identity in ϕ, we have the total derivative with respect to ϕ is zero and so

$$M_j(\phi)\nabla_\phi \begin{bmatrix} d_j \\ t_j \end{bmatrix} = \nabla_\phi \mathbf{r}_j(\phi) - \nabla_\phi(M_j(\phi)) \begin{bmatrix} d_j \\ t_j \end{bmatrix}, j = 1, 2, 3.$$

Therefore we can easily compute $\nabla_\phi \mathbf{d}$ and $\nabla_\phi \mathbf{t}$ by calculating the right hand side and solving these systems. So consider the right hand side.

Define the matrices

$$B = \sum_{i=1}^{m} \mathbf{q}_i \mathbf{q}_i^T,$$

$$C = \sum_{i=1}^{m} \mathbf{q}_i \mathbf{p}_i^T.$$

Then we can write (we now show again explicitly dependence on ϕ)

$$\mathbf{r}_j(\phi) = \begin{bmatrix} \mathbf{e}_j^T R(\phi) C \mathbf{e}_j \\ m \mathbf{e}_j^T \bar{\mathbf{p}} \end{bmatrix}, j = 1, 2, 3,$$

and it is readily seen that

$$\nabla_\phi(r_j(\phi)) = \begin{bmatrix} \mathbf{e}_j^T C^T \\ 0^T \end{bmatrix} G_j(\phi), j = 1, 2, 3.$$

Further,

$$M_j(\phi) = \begin{bmatrix} \mathbf{e}_j^T R(\phi) B R(\phi)^T \mathbf{e}_j & m \mathbf{e}_j^T R(\phi) \bar{\mathbf{q}} \\ m \mathbf{e}_j^T R(\phi) \bar{\mathbf{q}} & m \end{bmatrix}, j = 1, 2, 3,$$

and so

$$M_j(\phi)\begin{bmatrix} d_j \\ t_j \end{bmatrix} = \begin{bmatrix} d_j \mathbf{e}_j^T R(\phi) B R^T(\phi)\mathbf{e}_j + m t_j \mathbf{e}_j^T R(\phi)\bar{\mathbf{q}} \\ m d_j \mathbf{e}_j^T R(\phi)\bar{\mathbf{q}} + m t_j \end{bmatrix}, j = 1, 2, 3.$$

It follows that

$$\nabla_\phi(M_j(\phi))\begin{bmatrix} d_j \\ t_j \end{bmatrix} = \begin{bmatrix} 2 d_j \mathbf{e}_j^T R(\phi) B + m t_j \bar{\mathbf{q}}^T \\ m d_j \bar{\mathbf{q}}^T \end{bmatrix} G_j(\phi), j = 1, 2, 3.$$

Thus to obtain $\nabla_\phi \mathbf{d}$ and $\nabla_\phi \mathbf{t}$, we need to solve the three 2×2 systems of equations

$$M_j(\phi)\nabla_\phi \begin{bmatrix} d_j \\ t_j \end{bmatrix} = \begin{bmatrix} \mathbf{e}_j^T (C^T - 2 d_j R(\phi) B) - m t_j \bar{\mathbf{q}}^T \\ -m d_j \bar{\mathbf{q}}^T \end{bmatrix} G_j(\phi), j = 1, 2, 3.$$

We will summarise the calculation of the data for one iteration. It is assumed that B and C are available, and ϕ is given.

1. Calculate for $j = 1, 2, 3$

$$M_j(\phi) = \begin{bmatrix} \mathbf{e}_j^T R(\phi) B R(\phi)^T \mathbf{e}_j & m \mathbf{e}_j^T R(\phi)\bar{\mathbf{q}} \\ m \mathbf{e}_j^T R(\phi)\bar{\mathbf{q}} & m \end{bmatrix},$$

$$\mathbf{r}_j(\phi) = \begin{bmatrix} \mathbf{e}_j^T R(\phi) C \mathbf{e}_j \\ m \mathbf{e}_J^T \bar{\mathbf{p}} \end{bmatrix}.$$

2. Solve for $j = 1, 2, 3$

$$M_j(\phi)\begin{bmatrix} d_j \\ t_j \end{bmatrix} = \mathbf{r}_j(\phi),$$

   to obtain $\mathbf{d}$ and $\mathbf{t}$.

3. Calculate for $j = 1, 2, 3$, $G_j(\phi)$ where

$$G_j^T(\phi) = \begin{bmatrix} \mathbf{e}_j^T R_1'(\phi) R_2(\phi) R_3(\phi) \\ \mathbf{e}_j^T R_1(\phi) R_2'(\phi) R_3(\phi) \\ \mathbf{e}_j^T R_1(\phi) R_2(\phi) R_3'(\phi) \end{bmatrix}.$$

4. For each j=1,2,3, solve

$$M_j(\phi)\begin{bmatrix}\mathbf{x}^T\\ \mathbf{y}^T\end{bmatrix} = \begin{bmatrix}\mathbf{e}_j^T C^T - 2d_j\mathbf{e}_j^T R(\phi)B - mt_j\bar{\mathbf{q}}^T\\ -md_j\bar{\mathbf{q}}^T\end{bmatrix} G_j(\phi)$$

and calculate

$$\mathbf{e}_j^T\nabla_\phi\mathbf{v}_i = -d_j\mathbf{q}_i^T G_j(\phi) - (\mathbf{e}_j^T R(\phi)\mathbf{q}_i)\mathbf{x}^T - \mathbf{y}^T,\ i = 1,\ldots,m.$$

In Step 4, $\mathbf{x}^T = \nabla_\phi d_j$, $\mathbf{y}^T = \nabla_\phi t_j$ for each value of j. The Jacobian can be built up in 3 steps with m rows calculated at each step. So rows $1, 4, 7, .., 3m-2$ are calculated for $j = 1$, rows $2, 5, 8, .., 3m-1$ for $j = 2$ and rows $3, 6, 9, .., 3m$ for $j = 3$.

The Hanson and Norris method can be used to provide a starting value of ϕ. It is necessary to compute the singular value decomposition $U\Sigma V^T$, say, of the matrix $C^T - m\bar{\mathbf{p}}\bar{\mathbf{q}}^T$, and then $R(\phi) = UV^T$ and the individual angles may be computed using the formulae

$$\tan(\alpha) = \frac{R(2,1)}{R(1,1)},$$

$$\tan(\beta) = \frac{R(3,1)}{\cos(\alpha)R(1,1) + \sin(\alpha)R(2,1)},$$

$$\tan(\gamma) = \frac{\cos(\beta)R(3,2) - \sin(\beta)(\cos(\alpha)R(1,2) + \sin(\alpha)R(2,2))}{\cos(\alpha)R(2,2) - \sin(\alpha)R(1,2)}.$$

3 Numerical results

A Matlab program was written to test an implementation of a Gauss-Newton method using the formulae of the previous section. Starting from a unit step length, a simple halving strategy was used (if necessary) to force descent at each iteration, and the iteration process was terminated when the increment vector $\mathbf{z}$ solving (7) satisfied

$$\|\mathbf{z}\|_\infty < 10^{-5}.$$

We illustrate by two examples.

$\mathbf{q}_i$	1	0	2	0	-1	-3	4	2	5	0	-2	3	0	5	-4	1
	0	2	2	0	1	3	3	8	-1	0	-2	0	0	6	2	1
	2	3	0	1	-4	4	1	3	-1	5	-3	0	4	7	-6	5

Table 1: $\mathbf{q}_i$ for Example 1

k	F	$\|\mathbf{z}\|_\infty$	γ
1	341.6849	0.8076	1
2	29.6457	0.1464	1
3	0.3099	0.0220	1
4	7.2×10^{-6}	0.0002	1
5	3.2×10^{-8}	3×10^{-8}	

Table 2: Performance of method for Example 1(a)

Example 1 Consider the example with $m = 16$ used by Späth in [14]. We consider two cases, both with $\mathbf{q}_i$ given by Table 1.
(a) First define $\mathbf{p}_i$ by

$$\mathbf{p}_i = DR(\phi)\mathbf{q}_i - \mathbf{t}, i = 1, \ldots, m,$$

where $D = \text{diag}\{2, 6, 0.5\}$, $\phi = [0.5, 2, 4.5]^T$ and $\mathbf{t} = [1, -3, 2]^T$.

The 6 parameter solution gives $\phi = [2.8374, 1.1514, 2.2012]^T$ with $F(\phi) = 341.6849$, and the algorithm performs as in Table 2, where γ is the step length, and k is the iteration number. The final ϕ is $\phi = [3.6416, 1.1416, 1.3584]^T$.
(b) Next define $\mathbf{p}_i$ as in Table 3, which shows perturbed values of the $\mathbf{p}_i$ in (a). Then again starting from the 6 parameter solution, which gives $\phi = [2.7213, 1.2658, 2.318]^T, F = 377.4950$, the algorithm performs as in Table 4, reaching $\phi = [3.7661, 1.2114, 1.2314]^T$. From the starting point used by Späth, namely $\phi = [2.5, 1, 5.5]^T$ the algorithm takes 9 iterations. The method of Späth takes 137 iterations from this point to the same objective function value, but because of the stopping criteria, the parameter values obtained here are likely to be more accurate.
Example 2 Let $\mathbf{q}_i, i = 1, \ldots, 100$ be generated by the Matlab rand command, and define $\mathbf{p}_i, i = 1, \ldots, 100$ by (5) with $\phi = [0.5, 1.5, 2.5]^T, \mathbf{t} =$

	1	1	4	0	4	5	3	10	-4	-3	-1	0	-2	4	10	1
$\mathbf{p}_i$	6	17	-3	1	-24	26	1	24	-15	26	-19	-7	20	38	-28	24
	3	1	2	3	2	0	5	3	3	1	-1	4	1	6	-1	3

Table 3: $\mathbf{p}_i$ for Example 1(b)

k	F	$\|\mathbf{z}\|_\infty$	γ
1	377.4950	1.1014	1
2	94.4261	0.1928	1
3	45.7000	0.0116	1
4	45.5718	8.0×10^{-5}	1
5	45.5718	3.4×10^{-7}	

Table 4: Performance of method for Example 1(b)

$[-1, 5, 4]^T$ and $D = \text{diag}\{1, 10, 0.1\}$. The elements of the vectors $\mathbf{p}_i, i = 1, \ldots, 100$ are now randomly perturbed. The number of iterations depends on the size of the perturbations, as this impacts on the size of F at a solution. For one level of perturbation, the 6 parameter solution gives $\phi = [0.0504, 1.3602, 2.6416]^T$ and the iteration proceeds as in Table 5.

Doubling the size of the perturbations results in a requirement for 9 iterations, with convergence to a value $F = 101.0338$.

In general the method appears to work well, and the 6 parameter solution can give a good starting point even when the scaling factors differ significantly for each variable. A small number of Gauss-Newton steps is usually required.

4 Concluding Remarks

The purpose of this note has been to point out that a method of Gauss-Newton type in the rotation parameters alone can readily be developed for the 9-parameter Helmert transformation problem. By taking as initial approximation the solution to the 6 (or 7) parameter problem, which can be obtained by the Hanson-Norris method, convergence to a limit point of the

k	F	$\|\mathbf{z}\|_\infty$	γ
1	103.7140	0.1416	1
2	26.0219	0.1720	1
3	25.2628	0.0228	1
4	25.2591	1.2×10^{-4}	1
5	25.2591	9.3×10^{-6}	

Table 5: Performance of method for Example 2

iteration is generally obtained in relatively few iterations. Of course other starting points may be available in practice. The main computational cost at each iteration is in the solution of a linear least squares problem with $3m$ equations and 3 unknowns. The calculation of the Jacobian matrix requires the solution of sets of linear equations with three 2×2 matrices. The limit point is not, of course, guaranteed to be a minimum of F, and this will depend on the quality of the initial approximation.

References

[1] Awange, J. L., Fukuda, Y. and E. Grafarend. Exact solution of the nonlinear 7-parameter datum transformation by Groebner basis, Bollettino di Geodesia e Scienze Affini 63, 117–127 (2004).

[2] Geodesy, www.absoluteastronomy.com/encyclopedia/G/Ge/Geodesy.htm

[3] Golub, G. H. and V. Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate, SIAM J. Num. Anal. 10, 413–432 (1973).

[4] Golub, G. H. and C. F. Van Loan. Matrix Computations, Johns Hopkins University Press, Second Edition (1989).

[5] Grafarend, E. and J. L. Awange. Nonlinear analysis of the three-dimensional datum transformation [conformal group $C_7(3)$], J. of Geodesy 77, 66–76 (2003).

[6] Grafarend, E., Hendricks, A. and A. Gilbert. Transformation of conformal coordinates of type Gauss-Krüger or UTM from a local datum (regional, national, European) to a global datum (WGS84, ITRF 96), Part II, AVN 6, 218–222 (2000).

[7] Grafarend, E. W. and Kampmann, G. $C_{10}(3)$: The ten parameter conformal group as a datum transformation in threedimensional Euclidean space, Zeitschrift für Vermessungswesen 2, 68–77 (1996).

[8] Grafarend, E. W., Knickmeyer, E. H. and B. Schaffrin. Geodätische Datumtransformationen, Zeitschrift für Vermessungswesen 1, 15–25 (1982).

[9] Grafarend, E. W., Krumm, F. and Okeke, F. Curvilinear geodetic datum transformations, Zeitschrift für Vermessungswesen 7, 334–350 (1995).

[10] Grafarend, E. and Syffus, R. Transformation of conformal coordinates of type Gauss-Krüger or UTM from a local datum (regional, national, European) to a global datum (WGS84), Part I, The transformation equations, AVN 4, 134–141 (1998).

[11] Hanson, R. J. and M. J. Norris. Analysis of measurements based on the singular value decomposition, SIAM J. Sci. and Stat. Comp. 2, 363–374 (1981).

[12] NATO Geodesy and Geophysics Working Group, http://earth-info.nga.mil/GandG/datums/natodt.html.

[13] Späth, H. Identifying spatial point sets, Math. Comm. 8, 69–75 (2003).

[14] Späth, H. A numerical method for determining the spatial HELMERT transformation in the case of different scale factors, Zeitschrift für Vermessungswesen 129, 255–259 (2004).

[15] Volk, W. Memorial stone for F. R. Helmert, www.w-volk.de/museum/memori02.htm

[16] Watson, G. A. Approximation Theory and Numerical Methods, Wiley, Chichester (1980).