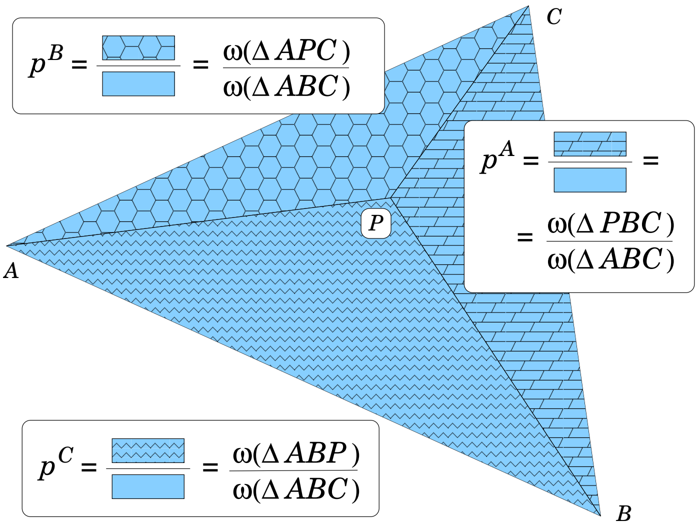
Statistical methods in geodesy
Maa-6.3282


C
$p^B = \frac{▨}{▭} = \frac{\omega(\Delta APC)}{\omega(\Delta ABC)}$
$p^A = \frac{▨}{▭} =$
$= \frac{\omega(\Delta PBC)}{\omega(\Delta ABC)}$
P
A
$p^C = \frac{▨}{▭} = \frac{\omega(\Delta ABP)}{\omega(\Delta ABC)}$
B

Martin Vermeer

April 7, 2011

Course Description

Workload 3 cr

Teaching Period III-IV, in springs of odd years

Learning Outcomes After completing the course the student understands the theoretical backgrounds of the various methods of adjustment calculus, optimization, estimation and approximation as well as their applications, and is able to apply them in a practical situation.

Content Free adjustment and constrainment to given points, the concepts of datum and datum transformation, similarity and affine transformations between datums, the use of *a priori* information, Helmert-Wolf blocking and stacking of normal equations, stochastic processes and Kalman filtering, measurement of shape, monitoring of deformations and 3D co-ordinate measurement; various approximation, interpolation and estimation methods and least squares collocation.

Foreknowledge Maa-6.203 or Maa-6.2203.

Equivalences Replaces course Maa-6.282.

Target Group

Completion Completion in full consists of the exam and the calculation exercises.

Workload by Component

- Lectures 13×2 h = 26 h
- Independent study 22 h
- Calculation exercises 4×8 h = 32 h(independent work)
- Total 80 h

Grading The grade of the exam becomes the grade of the course, 1-5

Study Materials Lecture notes.

Teaching Language English

Course Staff and Contact Info Martin Vermeer, room M309, name@tkk.fi

Reception times By agreement

CEFR-taso

Lisätietoja

Contents

Chapter 1

Free network and datum

Literature:

[Kal98b, s. 67-71, 141-150]

[Lan89]

[Lei95, s. 130-135]

[Coo87, s. 206-215, 311-321]

[SB97, s. 405-430]

[Lei95, s. 130-135]

[Baa73] partly.

1.1 Theory

A free network is a network, that is not in any way connected with external fixed points or a higher order (already measured) network.

Write the observation equations as follows

$$\underline{\ell} + \underline{\mathrm{v}} = A\widehat{\mathrm{x}}. \tag{1.1}$$

Here $\underline{\ell}$ is the vector of observations, $\underline{\mathrm{v}}$ that of residuals and $\widehat{\mathrm{x}}$ that of the unknowns, and A is the *design matrix*.

Let us assume, that for certain values of the vector $\underline{\mathrm{x}}$, i.e., c_i, $i = 1 \ldots r$:

$$A\mathrm{c}_i = 0.$$

Then we call the subspace of the space of observation vectors[1] $\underline{\mathrm{x}}$ which is spanned by the vectors c_i, having a dimensionality of r, the *null space* of A. The number r is called the *rank defect* of the matrix A[2]. Cf. [Kal98b, s. 68]. In this case the rank of the A matrixis less than its number of columns (or unknowns).

This produces the following situation:

> *If* $\widehat{\mathrm{x}}$ is the least squares solution of equation (1.1), then also every $\widehat{\mathrm{x}} + \sum_{i=1}^{r} \alpha^i \mathrm{c}_i$ is, with the same residuals. Here the coefficients α^i are arbitrary.

[1] A so-called abstract vector space

[2] The *rank* of a matrix is the number of its linearly independent rows or columns. Generally it is either the number of rows or the number of columns, whichever is smaller; for the design matrix it is thus the number of columns. If he rank is less than this, we speak of a *rank defect*.

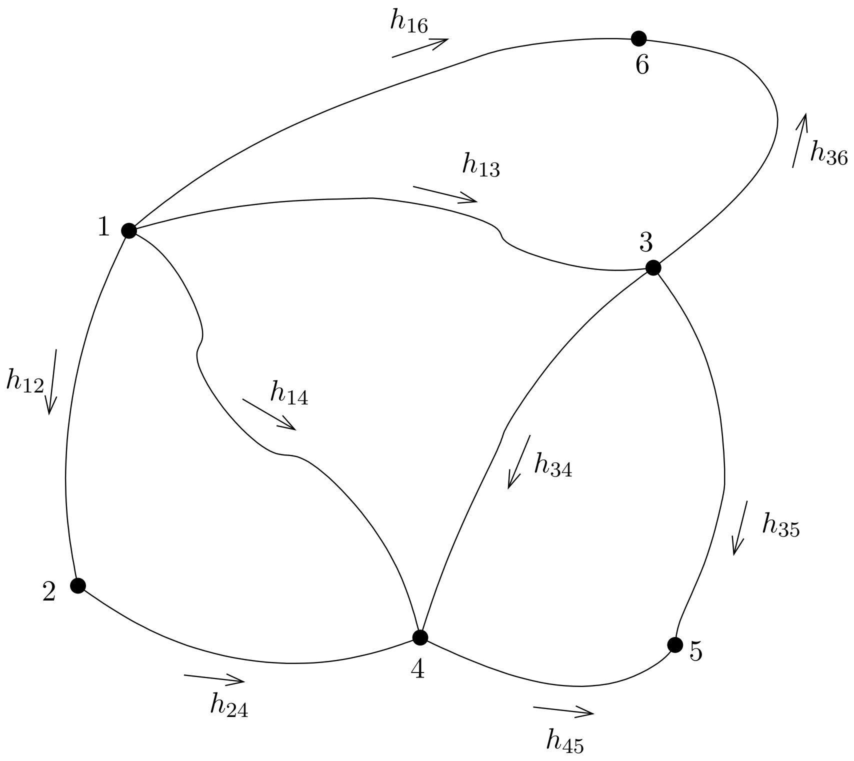


Figure 1.1: An example of a levelling network

1.2 Example: a levelling network

What do these null space vectors look like in realistic cases?

As an example, a *levelling network*. The network points i and j, heights H_i and H_j. As a measurement technique, levelling can only produce *differences* between heights, not absolute heights. Therefore the observation equations look as follows:

$$\ell_k + v_k = H_i - H_j.$$

Let the geometry of a levelling network be according to Figure 1.1. In this case the observation equations are

$$\begin{aligned}
\ell_1 + v_1 \equiv h_{12} + v_1 &= H_2 - H_1,\\
\ell_2 + v_2 \equiv h_{24} + v_2 &= H_4 - H_2,\\
\ell_3 + v_3 \equiv h_{14} + v_3 &= H_4 - H_1,\\
\ell_4 + v_4 \equiv h_{13} + v_4 &= H_3 - H_1,\\
\ell_5 + v_5 \equiv h_{16} + v_5 &= H_6 - H_1,\\
\ell_6 + v_6 \equiv h_{34} + v_6 &= H_4 - H_3,\\
\ell_7 + v_7 \equiv h_{35} + v_7 &= H_5 - H_3,\\
\ell_8 + v_8 \equiv h_{36} + v_6 &= H_6 - H_3,\\
\ell_9 + v_9 \equiv h_{45} + v_6 &= H_5 - H_4.
\end{aligned}$$

Written in matrix form:

$$\underline{\ell} + \underline{\mathrm{v}} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 & 0 \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \\ H_5 \\ H_6 \end{bmatrix}.$$

As can be easily verified, we obtain by summing together all columns of the matrix:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T.$$

Thus we have found one c vector: $\mathrm{c} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T$. Every element represents one column in the A matrix.

The rank defect of the above A matrix is 1 and its null space consists of all vectors $\alpha \mathrm{c} = \begin{bmatrix} \alpha & \alpha & \alpha & \alpha & \alpha & \alpha \end{bmatrix}^T$.

> In a levelling network, adding a constant to the height H_i of each point i does not change a single one of the levelling's observed quantities.

This is called the *datum defect.* Numerically the datum effect will cause the network adjustment's normal equations to be not solvable: the coefficient matrix is *singular* [3].

Every datum defect is at the same time an *invariant* of the observation equations, i.e., the left hand side does not change, even if we added to the vector of unknowns an element $\sum_{i=1}^{r} \alpha^i c_i$ of the null space of matrix A. In the example case, adding a constant to all heights is such an invariant.

1.3 Fixing the datum

We just saw, that if the design matrix has a rank defect r, then there exist $\mathbb{R}^r$ different but equivalent solutions $\underline{\mathrm{x}}$, that differ from each other only by an amount equal to the vector $\mathrm{c} \in \mathbb{R}^r$.

- Each such solution we call a *datum.*
- The transformation from such a datum to another one (example case: adding a constant to all heights in a network) we call a datum transformation or *S-transformation*[4].
- We can eliminate the datum defect by fixing r unknowns (arbitrarily and/or sensibly) to chosen values.

In our example case we can fix the datum, e.g., by fixing the height value in Helsinki harbour to mean sea level at the beginning of 1960, as we described earlier…

[3] Without special precautions, the program will probably crash on a zerodivide.

[4] S for *similarity.*

1.3.1 Constraints

Let us start from

$$A\mathrm{c}_i = 0,\; i = 1, \ldots, r$$

(i.e., the null space of the matrix A is r-dimensional). Let us form the matrix

$$C = \begin{bmatrix} \mathrm{c}_1 & \mathrm{c}_2 & \cdots & \mathrm{c}_i & \cdots & \mathrm{c}_{r-1} & \mathrm{c}_r \end{bmatrix}.$$

Now we may write the above condition as

$$AC = 0.$$

Now study the matrix[5]

$$\widetilde{A} = \begin{bmatrix} A \\ C^T \end{bmatrix}.$$

Calculate

$$\widetilde{A}C = \begin{bmatrix} A \\ C^T \end{bmatrix} C = \begin{bmatrix} AC \\ C^T C \end{bmatrix} = \begin{bmatrix} 0 \\ C^T C \end{bmatrix} \neq 0.$$

In other words: the adjustment problem described by $\widetilde{A}$ *has no rank defect.*

Such an adjustment problem is, e.g.:

$$\begin{bmatrix} \underline{\ell} \\ \mathrm{k} \end{bmatrix} + \begin{bmatrix} \underline{\mathrm{v}} \\ 0 \end{bmatrix} = \begin{bmatrix} A \\ C^T \end{bmatrix} \underline{\mathrm{x}}.$$

Forming the normal equations[6]:

$$\begin{bmatrix} A^T & C \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \underline{\ell} + \underline{\mathrm{v}} \\ \mathrm{k} \end{bmatrix} = \begin{bmatrix} A^T & C \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A \\ C^T \end{bmatrix} \underline{\mathrm{x}},$$

in which the normal matrix

$$\begin{aligned} \widetilde{N} &= \begin{bmatrix} A^T & C \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} A \\ C^T \end{bmatrix} = \\ &= A^T P A + C C^T. \end{aligned}$$

Here $N = A^T P A$ is the normal matrix of the original adjustment problem. The term CC^T *is new and represents the so-called* inner constraints (Cf. [Kal98b, ss. 69-71]). As the solution we obtain

$$\begin{aligned} \widehat{\mathrm{x}} &= \left[A^T P A + C C^T\right]^{-1} \begin{bmatrix} A^T & C \end{bmatrix} \begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \underline{\ell} + \underline{\mathrm{v}} \\ \mathrm{k} \end{bmatrix} = \\ &= \left[A^T P A + C C^T\right]^{-1} \left[A^T P \underline{\ell} + C\mathrm{k}\right]. \end{aligned}$$

[5] In the publication [Kal98b] the matrix C^T is called E.

[6] Observe from the form of the weight matrix $\widetilde{P} = \begin{bmatrix} P & 0 \\ 0 & I \end{bmatrix}$, that the formal additional observation vector khas been given the formal weight matrix I (unit matrix) and that we have assumed $\underline{\ell}$ and k to be statistically independent.

The most important change is the term added to the normal matrix, CC^T, which makes it invertable, i.e., $\widetilde{N}^{-1}$ exists even if N^{-1} would not exist. *This is why the literature also speaks about (Tikhonov-)*regularization[7]. The other change is the extra term $C\mathrm{k}$ added on the observation side.

Example: in the case of the above mentioned levelling network $\mathrm{c} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^{\mathrm{T}}$ and the observation equations extended with the inner constraint:

$$\begin{bmatrix} \underline{\ell} + \underline{\mathrm{v}} \\ k \end{bmatrix} = \left[\begin{array}{cccccc} -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ \hline 1 & 1 & 1 & 1 & 1 & 1 \end{array}\right] \begin{bmatrix} H_1 \\ H_2 \\ H_3 \\ H_4 \\ H_5 \\ H_6 \end{bmatrix}.$$

Here we can see that the added condition *fixes the sum* (or equivalently, the average) *of the heights of all the network's points* to the given value k, i.e.

$$\sum_{i=1}^{6} H_i = k.$$

This way of fixing yields the network solution in the "centre-of-gravity datum". The choice of the constant k (more generally: the vector of constants k) is arbitrary from the viewpoint of the "goodness" of $\widehat{\mathrm{x}}$ but it fixes it to certain numerical values.

1.3.2 Another approach: optimization

In the publication [Kal98b] on pages 69-71 as well as in publication [Coo87] the following approach is presented, however in an unclear fashion. Therefore here it is presented again.

The least squares solution of the adjustment problem

$$\underline{\ell} + \underline{\mathrm{v}} = A\underline{\mathrm{x}}$$

is obtained by minimizing literally the (weighted) square sum of residuals:

$$\varphi = \mathrm{v}^T Q^{-1} \mathrm{v} = (A\mathrm{x} - \ell)^T Q^{-1} (A\mathrm{x} - \ell) = \mathrm{x}^T A^T Q^{-1} A\mathrm{x} - \mathrm{x}^T A^T Q^{-1} \ell - \ell^T Q^{-1} A\mathrm{x} + \ell^T Q^{-1} \ell.$$

[7] . . . or "ridge regression". The terminology is somewhat confusing.

Differentiating with respect ot each x[8] yields

$$\frac{\partial\varphi}{\partial \mathrm{x}} = \mathrm{x}^T A^T Q^{-1} A + A^T Q^{-1} A\mathrm{x} - A^T Q^{-1}\ell - \ell^T Q^{-1} A + 0,$$

which must vanish (stationary point). This happens if

$$A^T Q^{-1} A\mathrm{x} - A^T Q^{-1}\ell = 0,$$

(because then also $\mathrm{x}^T A^T Q^{-1} A - \ell^T Q^{-1} A = 0$) which is precisely the system of normal equations.

Let us again study both the observation equations and the constraint equations:

$$\begin{bmatrix} \underline{\ell} \\ \mathrm{k} \end{bmatrix} + \begin{bmatrix} \underline{\mathrm{v}} \\ 0 \end{bmatrix} = \begin{bmatrix} A \\ C^T \end{bmatrix} \underline{\mathrm{x}}.$$

This can be interpreted as a minimization problem with "side conditions", i.e., a minimization problem with so-called LAGRANGE[9] *multipliers.* Let us write the quantity to be minimized as follows:

$$\varphi = (A\mathrm{x} - \ell)^T Q^{-1} (A\mathrm{x} - \ell) + \lambda^T \left(C^T\mathrm{x} - \mathrm{k}\right) + \left(C^T\mathrm{x} - \mathrm{k}\right)^T \lambda,$$

where λ is the (length r) vector of Lagrange multipliers. Minimizing the expression φ minimizes both the square sum of residuals and satisfies the additional conditions $C^T\mathrm{x} = \mathrm{k}$.

Differentiation with respect toxyields

$$\frac{\partial\varphi}{\partial \mathrm{x}} = \mathrm{x}^T A^T Q^{-1} A + A^T Q^{-1} A\mathrm{x} - A^T Q^{-1}\ell - \ell^T Q^{-1} A + \lambda^T C^T + C\lambda$$

which again must vanish. This is achieved by putting

$$A^T Q^{-1} A\mathrm{x} - A^T Q^{-1}\ell + C\lambda = 0$$

i.e., the normal equations

$$A^T Q^{-1} A\mathrm{x} + C\lambda = A^T Q^{-1}\ell.$$

Combining this with the constraint equation $C^T\mathrm{x} = \mathrm{k}$ yields

$$\begin{bmatrix} A^T Q^{-1} A & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} \mathrm{x} \\ \lambda \end{bmatrix} = \begin{bmatrix} A^T Q^{-1}\ell \\ \mathrm{k} \end{bmatrix}.$$

Here now the Lagrange multipliers are along as unknowns, and furthermore this set of equations looks deceptively like a set of normal equations... the matrix on the left hand side is invertible, albeit not particularly pretty.

The background for this acrobatics is the wish to find a form of the normal equations which allow the use of the generalized inverse or MOORE-PENROSE[10] inverse, also in the case that there is a rank defect. No more on this here.

[8] This is allowed, because

$$\frac{\partial x^i}{\partial x^j} = \delta^i_j = \begin{cases} 1 & i = j, \\ 0 & i \neq j, \end{cases}$$

(KRONECKER delta) where $\mathrm{x} = \begin{bmatrix} x^1 & \cdots & x^i & \cdots & x^m \end{bmatrix}^T$; or in "vector/matrix language"

$$\frac{\partial \mathrm{x}}{\partial \mathrm{x}} = I.$$

After this we apply the chain rule.

[9] Joseph-Louis (Giuseppe Lodovico) LAGRANGE (1736-1813), French (Italian) mathematician. `http://www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Lagrange.html`.

[10] Cf. `http://mathworld.wolfram.com/Moore-PenroseMatrixInverse.html`

1.3.3 Interpreting constraints as minimising a norm

The use of constraints as presented in part1.3.1 can be interpreted as minimizing the following expression:

$$\varphi = (A\mathrm{x} - \ell)^T Q^{-1} (A\mathrm{x} - \ell) + \left(C^T\mathrm{x} - \mathrm{k}\right)^T \left(C^T\mathrm{x} - \mathrm{k}\right).$$

On the right hand side of this expression we see what are mathematically two *norms*, and in the literature we speak of *minimum norm solution*. It is typical for using inner constraints, that the solution obtained is not "deformed" by the use of the constraints, e.g., when fixing a levelling network in this way, the height differences between points do not change. The only effect is to make the solution unique.

We can replace the above expression by:

$$\varphi = (A\mathrm{x} - \ell)^T Q^{-1} (A\mathrm{x} - \ell) + \lambda \left(C^T\mathrm{x} - \mathrm{k}\right)^T \left(C^T\mathrm{x} - \mathrm{k}\right),$$

where λ can be chosen arbitrarily, as long as $\lambda > 0$. The end result does not depend on λ, and we may even use $\varphi = \lim_{\lambda\downarrow 0} \varphi(\lambda)$, yielding still the same solution.

In fact, *any* choice that picks from all equivalent solutions xjust one, is a "legal" choice. E.g.

$$\varphi = (A\mathrm{x} - \ell)^T Q^{-1} (A\mathrm{x} - \ell) + \mathrm{x}^T\mathrm{x}$$

is just fine. Then we minimize the length of the x vector $\|\mathrm{x}\| = \sqrt{\mathrm{x}^T\mathrm{x}}$. A more general case is the form

$$(A\mathrm{x} - \ell)^T Q^{-1} (A\mathrm{x} - \ell) + \lambda\mathrm{x}^T G\mathrm{x},$$

in which G is a suitably positive (semi-)definite matrix.

If in the earlier equation we choose $\mathrm{k} = 0$, we obtain

$$\varphi = (A\mathrm{x} - \ell)^T Q^{-1} (A\mathrm{x} - \ell) + \lambda\mathrm{x}^T CC^T\mathrm{x},$$

which belongs to this group: $G = CC^T$.

1.3.4 More generally about regularization

Similar techniques are also used in cases, where A isn't strictly rank deficient, but just very poorly conditioned. In this case we speak of *regularization*. This situation can be studied in the following way. Let the normal matrix be

$$N = A^T Q^{-1} A.$$

saadaIf the matrix is regular, it will also be positive definite, ie., all its eigenvalues will be positive. Also, according to theory, will the corresponding eigenvectors be mutually orthogonal. Therefore, by a simple rotation in x space, we may get N "on principal axes"[11]:

$$N = R^T \Lambda R,$$

where Λ is a diagonal matrix having as elements the eigenvalues $\lambda_i, i = 1, m$ (m the number of unknowns, i.e. the length of the vector x.)

If the matrix N is not regular, then some of its eigenvalues are zero. Their number is precisely the rank defect of the matrix A. Adding a suitable term G to N will fix this singularity.

[11] You do remember, don't you, that a rotation matrix is orthogonal, i.e. $RR^T = R^T R = I$ or $R^{-1} = R^T$.

If some of the eigenvalues of N are instead of zero only very small, we speak of a *poorly conditioned* matrix[12]. Often it is numerically impossible to invert, or inversion succeeds only by used double or extended precision numbers. A good metric for the invertability of a matrix is its *condition number*

$$\kappa = \lambda_{max}/\lambda_{min},$$

The ratio between the largest and the smallest eigenvalue. Matlab offers the possibility to compute this number. The smaller, the better.

1.4 Other examples

1.4.1 Distance measurement

If we have a plane network, in which have been measured *only ranges* (distances), then the observation equations are of the form:

$$\underline{\ell}_k + \underline{\mathrm{v}}_k = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$

As we can easily see, increasing all x values — including both x_i and x_j — with a constant amount will not change the right hand side of this equation. The same with y. In other words:

> Shifting (translating) all points over a fixed vector $\begin{bmatrix} \Delta x & \Delta y \end{bmatrix}^T$ in the plane does not change the observation equations.

There is still a third invariant: the expression $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ is precisely the distance between points i and j, and it does not change even if the whole point field were to be rotated by an angle α, e.g., about the origin.

If we write the vector of unknowns in the form $\begin{bmatrix} \cdots & x_i & y_i & \cdots & x_j & y_j & \cdots \end{bmatrix}^T$, then the c vectors take on the form:

$$\mathrm{c}_1 = \begin{bmatrix} \vdots \\ 1 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \end{bmatrix}, \mathrm{c}_2 = \begin{bmatrix} \vdots \\ 0 \\ 1 \\ \vdots \\ 0 \\ 1 \\ \vdots \end{bmatrix}, \mathrm{c}_3 = \begin{bmatrix} \vdots \\ -y_i \\ +x_i \\ \vdots \\ -y_j \\ +x_j \\ \vdots \end{bmatrix}.$$

Here c_1 and c_2 represent the translations in the $x$and y directions and c_3 the rotation around the origin (let us assume for simplicity, that α is small).

The general datum transformation vector is now

$$\sum_{i=1}^{r} \alpha^i c_i = \Delta x \cdot \mathrm{c}_1 + \Delta y \cdot \mathrm{c}_2 + \alpha \cdot \mathrm{c}_3.$$

The rank defect r is 3.

[12] The whole adjustment problem is called *ill-posed*.

1.4.2 About the scale

If we measure, instead of distances, *ratios* between distances — which is what happens in reality if we use a poorly calibrated distance measurement instrument[13] — we will have, in addition to the already mentioned three datum defects, a fourth one: *the scale.* Its c vector is $\mathrm{c} = \begin{bmatrix} \cdots & x_i & y_i & \cdots & x_j & y_j & \cdots \end{bmatrix}^T$.

In this case the datum defect is *four.* It is eliminated by fixing *two points* or four co-ordinates.

The whole C matrix is now

$$C = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & -y_i & x_i \\ 0 & 1 & +x_i & y_i \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & -y_j & x_j \\ 0 & 1 & +x_j & y_j \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}.$$

Cf. [Kal98b, s. 70].

1.4.3 Angle measurement

If we have measured in the network also *angles*, the amount of datum defects does not change. Also angle measurements are invariant with respect to translation, rotation and (where appropriate) scaling.

1.4.4 Azimuth measurement

A rather rare situation. If we have measured absolute azimuths (e.g., with a gyrotheodolite), there will not be a datum defect associated with rotation. All the azimuths in the network will be obtained absolutely from the adjustment.

1.4.5 The case of space geodesy

In this case we measure, in three dimensions, (pseudo-)ranges. We may think that the datum defect would be six: three translations (components of the translation vector) and three rotation angles in space.

However,

1. if the measurements are done to satellites orbiting Earth, we obtain as the implicit origin of the equations of motion the centre of mass of the Earth. I.e., the three dimensional translation defect disappears.
2. if the measurements are done at different times of the day, then the Earth will have rotated about its axes between them. This *direction* of the Earth's rotation axis (two parameters) will then appear in the observation equations, and two of the three rotation

[13] Often the poorly known effect of the atmosphere (propagation medium) on signal propagation has a similar effect as poor calibration. Therefore it is a good idea to make the scale into an unknown in the network sides are long.

angles disappear, if the measurements are done between stations on the Earth's surface and satellites orbiting in space.

Only *one datum defect* is left: the rotation angle around the Earth's rotation axis.

1.4.6 Very long baseline interferometry (VLBI)

In this case the measurement targets in space are so far away, that the centre of mass of the Earth does not appear in the observation equations. There are four datum defects: a translation vector (i.e., the position vector of the origin of the co-ordinate system) of three components, and a rotation angle about the rotation axis of the Earth.

Chapter 2

Similarity transformations (S-transformations) and criterion matrices

Literature:

[Kal98b, s. 67-71, 141-150]

[Str82]

[Lei95, s. 130-135]

[Coo87, s. 206-215, 311-321]

[SB97, s. 405-430]

[Baa73] partially.

2.1 Complex co-ordinates and point variances

As we saw already earlier, we can advantageously express plane co-ordinates as complex numbers:

$$\mathbf{z} = x + iy,$$

where (x, y) are plane co-ordinates. Now also variances can be written complexly: if the real-valued variance and covariance definitions are

$$\begin{aligned} \mathrm{Var}(x) &\equiv E\left\{(x - E\{x\})^2\right\}, \\ \mathrm{Cov}(x, y) &\equiv E\{(x - E\{x\})(y - E\{y\})\}, \end{aligned}$$

we can make corresponding definitions also in the complex plane:

$$\begin{aligned} \mathrm{Var}(\mathbf{z}) &\equiv E\{(\mathbf{z} - E\{\mathbf{z}\})(\overline{\mathbf{z}} - E\{\overline{\mathbf{z}}\})\}, \\ \mathrm{Cov}(\mathbf{z}, \mathbf{w}) &\equiv E\{(\mathbf{z} - E\{\mathbf{z}\})(\overline{\mathbf{w}} - E\{\overline{\mathbf{w}}\})\}. \end{aligned}$$

Here, the overbar means complex conjugate, i.e., if $\mathbf{z} = x + iy$, then $\overline{\mathbf{z}} = x - iy$.

We can see by calculating (remember that $i^2 = -1$), that

$$\mathrm{Var}(\mathbf{z}) = \mathrm{Var}(x) + \mathrm{Var}(y).$$

In other words, the point variance $\sigma_P^2 \equiv \sigma_x^2 + \sigma_y^2 = \mathrm{Var}(x) + \mathrm{Var}(y)$ is the same as the complex variance $\mathrm{Var}(\mathbf{z})$(which thus is real valued), and the covariance between the co-ordinates x and y of the same point vanishes.

2.2 S-transformation in the complex plane

If given are the co-ordinates of the point field (x_i, y_i), we can transform them to the new co-ordinate system using a similarity transformation, by giving only the co-ordinates of two points in both the old and the new system. Let the points be A and B, and the co-ordinate differences between the old and the new system

$$\begin{aligned}\delta\mathbf{z}_A &= \mathbf{z}'_A - \mathbf{z}_A,\\ \delta\mathbf{z}_B &= \mathbf{z}'_B - \mathbf{z}_B.\end{aligned}$$

Here we assume $\mathbf{z}'_A, \mathbf{z}'_B$ to be *exact,* i.e., the points A and B act as *datum points*, the co-ordinates of which are a matter of definition and not the result measurement and calculation.

Then we can compute the *correction* to the co-ordinates of point $\mathbf{z}_i$ as the following linear combination of the corrections for points A and B:

$$\delta\mathbf{z}_i = \frac{\mathbf{z}_i - \mathbf{z}_A}{\mathbf{z}_B - \mathbf{z}_A}\delta\mathbf{z}_B + \frac{\mathbf{z}_i - \mathbf{z}_B}{\mathbf{z}_A - \mathbf{z}_B}\delta\mathbf{z}_A.$$

We define $\mathbf{z}_{AB} \equiv \mathbf{z}_B - \mathbf{z}_A$, $\mathbf{z}_{Ai} \equiv \mathbf{z}_i - \mathbf{z}_A$ etc. and write in matric form:

$$\begin{aligned}\mathbf{z}'_i &= \begin{bmatrix} 1 & \frac{\mathbf{z}_{iB}}{\mathbf{z}_{AB}} & \frac{\mathbf{z}_{Ai}}{\mathbf{z}_{AB}} \end{bmatrix} \begin{bmatrix} \mathbf{z}_i \\ \delta\mathbf{z}_A \\ \delta\mathbf{z}_B \end{bmatrix} = \\ &= \begin{bmatrix} 1 & -\frac{\mathbf{z}_{iB}}{\mathbf{z}_{AB}} & -\frac{\mathbf{z}_{Ai}}{\mathbf{z}_{AB}} \end{bmatrix} \begin{bmatrix} \mathbf{z}_i \\ \mathbf{z}_A - \mathbf{z}'_A \\ \mathbf{z}_B - \mathbf{z}'_B \end{bmatrix}. \end{aligned} \tag{2.1}$$

Note that the *sum of elements of the row matrix on the left hand side vanishes*:

$$1 - \frac{\mathbf{z}_{iB}}{\mathbf{z}_{AB}} - \frac{\mathbf{z}_{Ai}}{\mathbf{z}_{AB}} = 0.$$

2.3 Standard form of the criterion matrix

The precision effects of an S-transformation can be studied theoretically. Let us start by assuming, that the network has been measured by a method of which we know the precision behaviour. Instead of the true precision, we then often use a so-called criterion variance matrix [Baa73], which describes in a simple mathematical fashion the spatial behaviour of the point field precision.

The classification of geodetic networks into orders based upon precision may be considered a primitive form of the criterion variance idea.

A simple rule is, e.g., that the so-called relative point mean error between two points has to be a function of the distance separating the points, and that it does not depend upon the direction between them, and also not on the absolute location of the points. Suitable such so-called homogeneous and isotropic spatial variance structures can be found in the literature.

Often, following the Delft school, we use as the criterion matrix – some sort of idealized variance matrix, close to what we would get as the variance matrix in a regular, well designed network

– the following expression[1]:

$$\begin{aligned}\operatorname{Var}(\mathbf{z}) &= \alpha^2, && (2.2)\\ \operatorname{Cov}(\mathbf{z},\mathbf{w}) &= \alpha^2 - \frac{1}{2}\sigma^2(\mathbf{z}-\mathbf{w})(\overline{\mathbf{z}}-\overline{\mathbf{w}}) = \\ &= \alpha^2 - \frac{1}{2}\sigma^2\left\|\mathbf{z}-\mathbf{w}\right\|^2. && (2.3)\end{aligned}$$

Here, the value α^2 nis arbitrary; it is always positive. One can imagine, that it is very large, larger than $\frac{1}{2}\sigma^2\left\|\mathbf{z}-\mathbf{w}\right\|^2$ anywhere in the area of study, and represents the local (close to the origin) uncertainty in co-ordinates caused by the use of very remote fixed points.

> Intuitively, one can imagine a network made up of triangles, all of the same size, where all sides have been measured with equal precision, and the edge of the network is allowed to travel to infinity in all directions. The edge points are kept fixed. Then
>
> $$\alpha^2 \to \infty,$$
>
> but before that
>
> $$\operatorname{Cov}(\mathbf{z},\mathbf{w}) \;\to\; \operatorname{Var}(\mathbf{z}) - \frac{1}{2}\sigma^2\left\|\mathbf{z}-\mathbf{w}\right\|^2.$$
>
> See figure. 2.1)

After this definition, we can calculate the *relative variance matrix* between two points A and B:

$$\begin{aligned}\operatorname{Var}(\mathbf{z}_{AB}) &= \operatorname{Var}(\mathbf{z}_A) + \operatorname{Var}(\mathbf{z}_B) - 2\operatorname{Cov}(\mathbf{z}_A,\mathbf{z}_B) = \\ &= 2\alpha^2 - 2\alpha^2 + \sigma^2\mathbf{z}_{AB}\overline{\mathbf{z}_{AB}} = +\sigma^2\mathbf{z}_{AB}\overline{\mathbf{z}_{AB}}.\end{aligned}$$

We see that α^2 has vanished from this and the variance obtained is directly proportional to the second power of the inter-point distance:

$$\mathbf{z}_{AB}\overline{\mathbf{z}_{AB}} = (x_B - x_A)^2 + (y_B - y_A)^2.$$

This is also a real number, i.e., there is no correlation between the co-ordinates x and y and the error ellipses are *circles.*

2.3.1 A more general form

A more general form of the criterion function is an arbitrary function of the inter-point distance:

$$\begin{aligned}\operatorname{Var}(\mathbf{z}) &= \alpha^2,\\ \operatorname{Cov}(\mathbf{z},\mathbf{w}) &= \alpha^2 - \frac{1}{2}\sigma^2 f\left(\left\|\mathbf{z}-\mathbf{w}\right\|\right),\end{aligned}$$

e.g.,

$$\operatorname{Cov}(\mathbf{z},\mathbf{w}) = \alpha^2 - \frac{1}{2}\sigma^2\left\|\mathbf{z}-\mathbf{w}\right\|^{2\nu},$$

where ν is a constant to be chosen. In practice, values 0.5...1.0 are suitable.

[1] An alternative structure producing the same end results would be

$$\begin{aligned}\operatorname{Var}(\mathbf{z}) &= \sigma^2\mathbf{z}\overline{\mathbf{z}}\\ \operatorname{Cov}(\mathbf{z},\mathbf{w}) &= \frac{1}{2}\sigma^2(\mathbf{z}\overline{\mathbf{w}}+\overline{\mathbf{z}}\mathbf{w}) = \frac{1}{2}\sigma^2(\mathbf{z}\overline{\mathbf{z}}+\mathbf{w}\overline{\mathbf{w}}) - \frac{1}{2}\sigma^2\left[(\mathbf{z}-\mathbf{w})(\overline{\mathbf{z}}-\overline{\mathbf{w}})\right] = \\ &= \frac{1}{2}(\operatorname{Var}(\mathbf{z})+\operatorname{Var}(\mathbf{w})) - \frac{1}{2}\sigma^2\left[(\mathbf{z}-\mathbf{w})(\overline{\mathbf{z}}-\overline{\mathbf{w}})\right].\end{aligned}$$

The aesthetic advantage of this alternative is, that we have no need for an arbitrary α^2. However, the aestetic weakness is that it contains the *absolute point location* $\mathbf{z}$. The problem has only changed place.

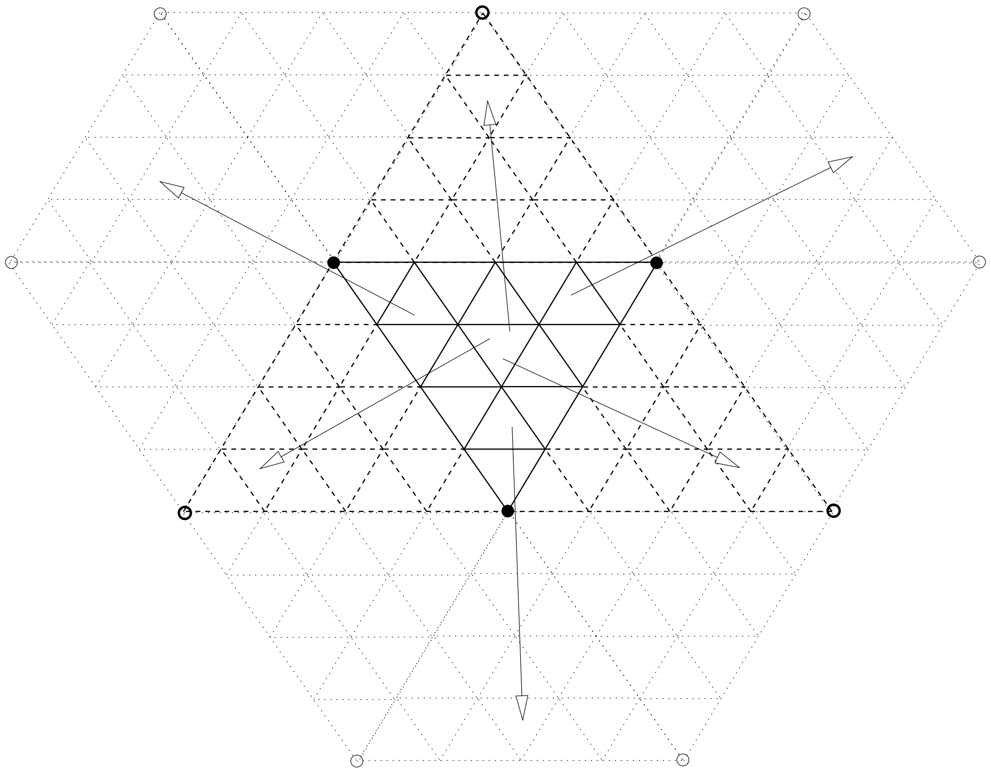


Figure 2.1: A regular triangle network extending in all directions to infinity

2.4 S-transformation of the criterion matrix

The criterion matrix of the point field $\mathbf{z}_i, \mathbf{z}_A, \mathbf{z}_B$ can be written as follows:

$$
\begin{aligned}
\mathrm{Var}\left(\begin{bmatrix} \mathbf{z}_i \\ \mathbf{z}_A \\ \mathbf{z}_B \end{bmatrix}\right) &= \begin{bmatrix} \mathrm{Var}\left(\mathbf{z}_i\right) & \mathrm{Cov}\left(\mathbf{z}_i, \mathbf{z}_A\right) & \mathrm{Cov}\left(\mathbf{z}_i, \mathbf{z}_B\right) \\ \mathrm{Cov}\left(\mathbf{z}_A, \mathbf{z}_i\right) & \mathrm{Var}\left(\mathbf{z}_A\right) & \mathrm{Cov}\left(\mathbf{z}_A, \mathbf{z}_B\right) \\ \mathrm{Cov}\left(\mathbf{z}_B, \mathbf{z}_i\right) & \mathrm{Cov}\left(\mathbf{z}_B, \mathbf{z}_A\right) & \mathrm{Var}\left(\mathbf{z}_B\right) \end{bmatrix} = \\
&= \begin{bmatrix} \alpha^2 & \alpha^2 - \frac{1}{2}\sigma^2 \mathbf{z}_{iA}\overline{\mathbf{z}_{iA}} & \alpha^2 - \frac{1}{2}\sigma^2 \mathbf{z}_{iB}\overline{\mathbf{z}_{iB}} \\ \alpha^2 - \frac{1}{2}\sigma^2 \mathbf{z}_{iA}\overline{\mathbf{z}_{iA}} & \alpha^2 & \alpha^2 - \frac{1}{2}\sigma^2 \mathbf{z}_{AB}\overline{\mathbf{z}_{AB}} \\ \alpha^2 - \frac{1}{2}\sigma^2 \mathbf{z}_{iB}\overline{\mathbf{z}_{iB}} & \alpha^2 - \frac{1}{2}\sigma^2 \mathbf{z}_{AB}\overline{\mathbf{z}_{AB}} & \alpha^2 \end{bmatrix}.
\end{aligned}
$$

Because in the formula 2.1 the co-ordinates $\mathbf{z}_A$ and $\mathbf{z}_B$ are exact, we may now write directly the *propagation law of variances*:

$$
\mathrm{Var}\left(\mathbf{z}_i'\right) = \begin{bmatrix} 1 & -\frac{\mathbf{z}_{iB}}{\mathbf{z}_{AB}} & -\frac{\mathbf{z}_{Ai}}{\mathbf{z}_{AB}} \end{bmatrix} \mathrm{Var}\left(\begin{bmatrix} \mathbf{z}_i \\ \mathbf{z}_A \\ \mathbf{z}_B \end{bmatrix}\right) \begin{bmatrix} 1 \\ -\frac{\overline{\mathbf{z}_{iB}}}{\overline{\mathbf{z}_{AB}}} \\ -\frac{\overline{\mathbf{z}_{Ai}}}{\overline{\mathbf{z}_{AB}}} \end{bmatrix}. \tag{2.4}
$$

Here the aforementioned variance matrix has been pre-multiplied by the coefficients of equation 2.1 as a row vector, and post-multiplied by the same coefficients *transposed* (i.e., as a column vector) *and complex conjugated.* This is the complex version of the propagation law of variances.

In practice, because of the structure of the coefficient matrix (the row sums vanish), the α^2 term may be left out from all elements, and we obtain

$$
\mathrm{Var}\left(\mathbf{z}_i'\right) = \sigma^2 \begin{bmatrix} 1 & -\frac{\mathbf{z}_{iB}}{\mathbf{z}_{AB}} & -\frac{\mathbf{z}_{Ai}}{\mathbf{z}_{AB}} \end{bmatrix} \begin{bmatrix} 0 & -\frac{1}{2}\mathbf{z}_{iA}\overline{\mathbf{z}_{iA}} & -\frac{1}{2}\mathbf{z}_{iB}\overline{\mathbf{z}_{iB}} \\ -\frac{1}{2}\mathbf{z}_{iA}\overline{\mathbf{z}_{iA}} & 0 & -\frac{1}{2}\mathbf{z}_{AB}\overline{\mathbf{z}_{AB}} \\ -\frac{1}{2}\mathbf{z}_{iB}\overline{\mathbf{z}_{iB}} & -\frac{1}{2}\mathbf{z}_{AB}\overline{\mathbf{z}_{AB}} & 0 \end{bmatrix} \begin{bmatrix} 1 \\ -\frac{\overline{\mathbf{z}_{iB}}}{\overline{\mathbf{z}_{AB}}} \\ -\frac{\overline{\mathbf{z}_{Ai}}}{\overline{\mathbf{z}_{AB}}} \end{bmatrix}.
$$

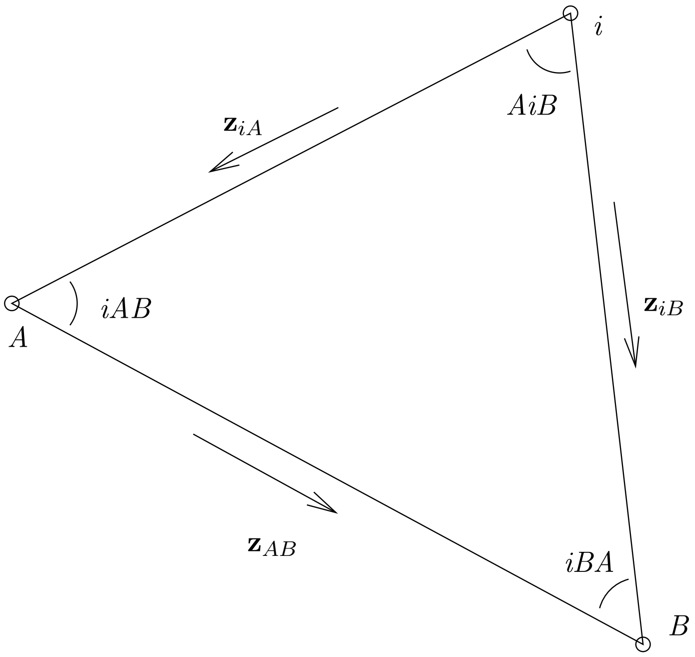


Figure 2.2: The quantities used in defining the criterion variance matrix

Careful calculation yields:

$$\operatorname{Var}\left(\mathbf{z}_i'\right) = \frac{1}{2}\sigma^2\left[\mathbf{z}_{iA}\overline{\mathbf{z}_{iA}}\left\{\frac{\mathbf{z}_{iB}}{\mathbf{z}_{AB}}+\frac{\overline{\mathbf{z}_{iB}}}{\overline{\mathbf{z}_{AB}}}\right\}+\mathbf{z}_{iB}\overline{\mathbf{z}_{iB}}\left\{\frac{\mathbf{z}_{iA}}{\mathbf{z}_{AB}}+\frac{\overline{\mathbf{z}_{iA}}}{\overline{\mathbf{z}_{AB}}}\right\}+\left(\mathbf{z}_{iA}\overline{\mathbf{z}_{iB}}+\overline{\mathbf{z}_{iA}\overline{\mathbf{z}_{iB}}}\right)\right].$$

Geometric interpretation: firstly we see, that this is real valued. Also:

$$\begin{aligned}
\mathbf{z}_{iA}\overline{\mathbf{z}_{iA}} &= \left\|\mathbf{z}_{iA}\right\|^2,\\
\mathbf{z}_{iB}\overline{\mathbf{z}_{iB}} &= \left\|\mathbf{z}_{iB}\right\|^2,\\
\frac{\mathbf{z}_{iA}}{\mathbf{z}_{AB}}+\frac{\overline{\mathbf{z}_{iA}}}{\overline{\mathbf{z}_{AB}}} &= 2\Re\left\{\frac{\mathbf{z}_{iA}}{\mathbf{z}_{AB}}\right\} = -2\frac{\left\|\mathbf{z}_{iA}\right\|}{\left\|\mathbf{z}_{AB}\right\|}\cos\angle iAB,\\
\frac{\mathbf{z}_{iB}}{\mathbf{z}_{AB}}+\frac{\overline{\mathbf{z}_{iB}}}{\overline{\mathbf{z}_{AB}}} &= 2\Re\left\{\frac{\mathbf{z}_{iB}}{\mathbf{z}_{AB}}\right\} = +2\frac{\left\|\mathbf{z}_{iB}\right\|}{\left\|\mathbf{z}_{AB}\right\|}\cos\angle iBA,\\
\mathbf{z}_{iA}\overline{\mathbf{z}_{iB}}+\overline{\mathbf{z}_{iA}\overline{\mathbf{z}_{iB}}} &= 2\Re\left\{\mathbf{z}_{iA}\overline{\mathbf{z}_{iB}}\right\} = 2\left\|\mathbf{z}_{iA}\right\|\left\|\mathbf{z}_{iB}\right\|\cos\angle AiB.
\end{aligned}$$

So:

$$\begin{aligned}
\operatorname{Var}\left(\mathbf{z}_i'\right) &= \sigma^2\left[\left\|\mathbf{z}_{iA}\right\|^2\Re\left\{\frac{\mathbf{z}_{iB}}{\mathbf{z}_{AB}}\right\}+\left\|\mathbf{z}_{iB}\right\|^2\Re\left\{\frac{\mathbf{z}_{iA}}{\mathbf{z}_{AB}}\right\}+\Re\left\{\mathbf{z}_{iA}\overline{\mathbf{z}_{iB}}\right\}\right] =\\
&= \sigma^2\left[\left\|\mathbf{z}_{iA}\right\|^2\frac{\left\|\mathbf{z}_{iB}\right\|}{\left\|\mathbf{z}_{AB}\right\|}\cos\angle iBA-\left\|\mathbf{z}_{iB}\right\|^2\frac{\left\|\mathbf{z}_{iA}\right\|}{\left\|\mathbf{z}_{AB}\right\|}\cos\angle iAB+\left\|\mathbf{z}_{iA}\right\|\left\|\mathbf{z}_{iB}\right\|\cos\angle AiB\right] =\\
&= \sigma^2\frac{\left\|\mathbf{z}_{iA}\right\|\left\|\mathbf{z}_{iB}\right\|}{\left\|\mathbf{z}_{AB}\right\|}\left[\left\|\mathbf{z}_{iA}\right\|\cos\angle iBA-\left\|\mathbf{z}_{iB}\right\|\cos\angle iAB+\left\|\mathbf{z}_{AB}\right\|\cos\angle AiB\right].
\end{aligned}$$

See figure 2.2.

2.5 S-transformations as members of a group

In mathematics a group G is defined (cf. `http://mathworld.wolfram.com/Group.html`) as a set with the following properties:

1. If A and B are two elements in G , then the product AB is also in G (closure)
2. *Multiplication* is *associative*, i.e., for all A, B, C in G, $(AB)\,C = A\,(BC)$
3. There is an *identity element* I so that $IA = AI = A\ \forall A \in G$
4. *There* must be an *inverse* of each element: for each element $A \in G$, the set contains also an element $B = A^{-1}$so that $AB = BA = I$.

The set of all invertible S-transformations (between two local datums) forms such a group.

2.5.1 The S-transformation from "infinity" to a local datum

The above described transformation of points from an "infinity datum" to a local datum can be generalized. Equation (2.4) for one point $\mathbf{z}_i$ can be written, e.g., for three different points $\mathbf{z}_i, \mathbf{z}_P, \mathbf{z}_Q$, so that we can compute, in addition to the variances, also the covariances between the points.

The formula looks then like this:

$$\mathrm{Var}\left(\begin{bmatrix} \mathbf{z}'_i \\ \mathbf{z}'_P \\ \mathbf{z}'_Q \end{bmatrix}\right) = \left[\begin{array}{ccc|cc} 1 & & & -\frac{\mathbf{z}_{iB}}{\mathbf{z}_{AB}} & -\frac{\mathbf{z}_{Ai}}{\mathbf{z}_{AB}} \\ & 1 & & -\frac{\mathbf{z}_{PB}}{\mathbf{z}_{AB}} & -\frac{\mathbf{z}_{AP}}{\mathbf{z}_{AB}} \\ & & 1 & -\frac{\mathbf{z}_{QB}}{\mathbf{z}_{AB}} & -\frac{\mathbf{z}_{AQ}}{\mathbf{z}_{AB}} \end{array}\right] \mathrm{Var}\left(\left[\begin{array}{c} \mathbf{z}_i \\ \mathbf{z}_P \\ \mathbf{z}_Q \\ \hline \mathbf{z}_A \\ \mathbf{z}_B \end{array}\right]\right) \left[\begin{array}{ccc} 1 & & \\ & 1 & \\ & & 1 \\ \hline -\frac{\overline{\mathbf{z}_{iB}}}{\overline{\mathbf{z}_{AB}}} & -\frac{\overline{\mathbf{z}_{PB}}}{\overline{\mathbf{z}_{AB}}} & -\frac{\overline{\mathbf{z}_{QB}}}{\overline{\mathbf{z}_{AB}}} \\ -\frac{\overline{\mathbf{z}_{Ai}}}{\overline{\mathbf{z}_{AB}}} & -\frac{\overline{\mathbf{z}_{AP}}}{\overline{\mathbf{z}_{AB}}} & -\frac{\overline{\mathbf{z}_{AQ}}}{\overline{\mathbf{z}_{AB}}} \end{array}\right].$$

The matrix featuring here may also be called

$$S_{(\infty)}^{(AB)} = \left[\begin{array}{ccc|cc} 1 & & & -\frac{\mathbf{z}_{iB}}{\mathbf{z}_{AB}} & -\frac{\mathbf{z}_{Ai}}{\mathbf{z}_{AB}} \\ & 1 & & -\frac{\mathbf{z}_{PB}}{\mathbf{z}_{AB}} & -\frac{\mathbf{z}_{AP}}{\mathbf{z}_{AB}} \\ & & 1 & -\frac{\mathbf{z}_{QB}}{\mathbf{z}_{AB}} & -\frac{\mathbf{z}_{AQ}}{\mathbf{z}_{AB}} \end{array}\right].$$

This matrix is rectangular and not invertible. This only illustrates the fact, that a datum once transformed from "infinity" to a local datum (AB) cannot be transformed back again. The above transformation equation is

$$\mathrm{Var}\left(\mathbf{z}^{(AB)}\right) = S_{(\infty)}^{(AB)} \mathrm{Var}\left(\mathbf{z}^{(\infty)}\right) \left[S_{(\infty)}^{(AB)}\right]^{\dagger},$$

where the symbol †, the so-called *hermitian*[2], designates the combination of transpose and complex conjugate.

2.5.2 The S-transformation between two local datums

Within a local network, there are always many alternatives for choosing the point pair A, B that act as datum points. It is even possible to transform a co-ordinate set that already refers to datum points A, B to some other datum point pair P, Q. That can be done as follows, starting from equation 2.1:

$$\begin{bmatrix} \mathbf{z}''_i \\ \mathbf{z}''_A - \mathbf{z}'_A \\ \mathbf{z}''_B - \mathbf{z}'_B \end{bmatrix} = \begin{bmatrix} 1 & -\frac{\mathbf{z}_{iQ}}{\mathbf{z}_{PQ}} & -\frac{\mathbf{z}_{Pi}}{\mathbf{z}_{PQ}} \\ 0 & -\frac{\mathbf{z}_{AQ}}{\mathbf{z}_{PQ}} & -\frac{\mathbf{z}_{PA}}{\mathbf{z}_{PQ}} \\ 0 & -\frac{\mathbf{z}_{BQ}}{\mathbf{z}_{PQ}} & -\frac{\mathbf{z}_{PB}}{\mathbf{z}_{PQ}} \end{bmatrix} \begin{bmatrix} \mathbf{z}'_i \\ \mathbf{z}'_P - \mathbf{z}''_P \\ \mathbf{z}'_Q - \mathbf{z}''_Q \end{bmatrix}.$$

[2] Charles Hermite, 1822-1901, French mathematician.
`http://www-history.mcs.st-andrews.ac.uk/Mathematicians/Hermite.html`.

Here we also obtain the corrections of the "old" datum points A and B, i.e., $\mathbf{z}''_A - \mathbf{z}'_A$ and $\mathbf{z}''_B - \mathbf{z}'_B$, to the new $''$ system, where we have as given co-ordinates $\mathbf{z}''_P$ and $\mathbf{z}''_Q$.

It is advantageous to use the following notation:

$$\begin{bmatrix} \mathbf{z}_i^{(PQ)} \\ \mathbf{z}_A^{(PQ)} - \mathbf{z}_A^{(AB)} \\ \mathbf{z}_B^{(PQ)} - \mathbf{z}_B^{(AB)} \end{bmatrix} = \begin{bmatrix} 1 & -\frac{\mathbf{z}_{iQ}}{\mathbf{z}_{PQ}} & -\frac{\mathbf{z}_{Pi}}{\mathbf{z}_{PQ}} \\ 0 & -\frac{\mathbf{z}_{AQ}}{\mathbf{z}_{PQ}} & -\frac{\mathbf{z}_{PA}}{\mathbf{z}_{PQ}} \\ 0 & -\frac{\mathbf{z}_{BQ}}{\mathbf{z}_{PQ}} & -\frac{\mathbf{z}_{PB}}{\mathbf{z}_{PQ}} \end{bmatrix} \begin{bmatrix} \mathbf{z}_i^{(AB)} \\ \mathbf{z}_P^{(AB)} - \mathbf{z}_P^{(PQ)} \\ \mathbf{z}_Q^{(AB)} - \mathbf{z}_Q^{(PQ)} \end{bmatrix}. \qquad (2.5)$$

Here we have as given as the *datum definition* in the (AB) system the co-ordinates $\mathbf{z}_A^{(AB)}, \mathbf{z}_B^{(AB)}$ (left hand side) and in the (PQ) system, $\mathbf{z}_P^{(PQ)}, \mathbf{z}_Q^{(PQ)}$. The matrix is often called

$$S_{(AB)}^{(PQ)} \equiv \begin{bmatrix} 1 & -\frac{\mathbf{z}_{iQ}}{\mathbf{z}_{PQ}} & -\frac{\mathbf{z}_{Pi}}{\mathbf{z}_{PQ}} \\ 0 & -\frac{\mathbf{z}_{AQ}}{\mathbf{z}_{PQ}} & -\frac{\mathbf{z}_{PA}}{\mathbf{z}_{PQ}} \\ 0 & -\frac{\mathbf{z}_{BQ}}{\mathbf{z}_{PQ}} & -\frac{\mathbf{z}_{PB}}{\mathbf{z}_{PQ}} \end{bmatrix}.$$

These transformation matrices form a mathematical *group*:

$$\begin{aligned} S_{(UV)}^{(PQ)} \cdot S_{(AB)}^{(UV)} &= S_{(AB)}^{(PQ)}, \\ \left(S_{(AB)}^{(PQ)}\right)^{-1} &= S_{(PQ)}^{(AB)}, \\ S_{(AB)}^{(AB)} &= I. \end{aligned}$$

i.e.,

1. transformations can be applied *successively* from the system (AB) through the system (UV) to the system (PQ);
2. the transformation $(AB) \rightarrow (PQ)$ has an *inverse transformation* $(PQ) \rightarrow (AB)$;
3. the trivial transformation $S_{(AB)}^{(AB)}$ also belongs to the group; it may be replaced by the unit matrix I because then on the right hand side, $\mathbf{z}_A^{(AB)} - \mathbf{z}_A^{(AB)} = \mathbf{z}_B^{(AB)} - \mathbf{z}_B^{(AB)} = 0$.[3]

Using this symbolism, we obtain

$$\mathbf{z}^{(PQ)} = S_{(AB)}^{(PQ)} \mathbf{z}^{(AB)},$$

where

$$\mathbf{z}^{(PQ)} \equiv \begin{bmatrix} \mathbf{z}_i^{(PQ)} \\ \mathbf{z}_A^{(PQ)} - \mathbf{z}_A^{(AB)} \\ \mathbf{z}_B^{(PQ)} - \mathbf{z}_B^{(AB)} \end{bmatrix} \equiv \begin{bmatrix} \mathbf{z}_i^{(PQ)} \\ \Delta\mathbf{z}_A^{(PQ)} \\ \Delta\mathbf{z}_B^{(PQ)} \end{bmatrix}, \quad \mathbf{z}^{(AB)} \equiv \begin{bmatrix} \mathbf{z}_i^{(AB)} \\ \mathbf{z}_P^{(AB)} - \mathbf{z}_P^{(PQ)} \\ \mathbf{z}_Q^{(AB)} - \mathbf{z}_Q^{(PQ)} \end{bmatrix} \equiv \begin{bmatrix} \mathbf{z}_i^{(AB)} \\ \Delta\mathbf{z}_P^{(AB)} \\ \Delta\mathbf{z}_Q^{(AB)} \end{bmatrix},$$

where the "delta quantities" $\Delta\mathbf{z}_P^{(AB)}$ etc. are defined according to the pattern !computed minus fixed by the datum definition".

All S-transformations are similarity transformations that preserve angles and ratios of lengths. Also the transformation formed by two successive S-transformations is again a similarity transformation, and so is the inverse of an S-transformation. This means that all operations defined for the group produce again a group member.

[3] More precisely, the matrix of the trivial transformation is $\begin{bmatrix} 1 & & \\ & -1 & \\ & & -1 \end{bmatrix}$; however, from the viewpoint of variance propagation this is equivalent with a unit matrix.

2.5.3 The case of several i-points

This formalism of S transformation matrices can easily be interpresteded – as it should be – more generally, if we let the point number i represent several points, $i = 1, \ldots, n$. Then

$$S^{(AB)}_{(\infty)} = \left[\begin{array}{ccc|cc} I_{n\times n} & & & \left[-\frac{\mathbf{z}_{iB}}{\mathbf{z}_{AB}}\right]_{i=1,\ldots,n} & \left[-\frac{\mathbf{z}_{Ai}}{\mathbf{z}_{AB}}\right]_{i=1,\ldots,n} \\ & 1 & & -\frac{\mathbf{z}_{PB}}{\mathbf{z}_{AB}} & -\frac{\mathbf{z}_{AP}}{\mathbf{z}_{AB}} \\ & & 1 & -\frac{\mathbf{z}_{QB}}{\mathbf{z}_{AB}} & -\frac{\mathbf{z}_{AQ}}{\mathbf{z}_{AB}} \end{array}\right],$$

where the square bracketed expressions ($[\cdot]_{i=1,\ldots,n}$) are column vectors of length nSimilarly

$$S^{(PQ)}_{(AB)} \equiv \begin{bmatrix} I_{n\times n} & \left[-\frac{\mathbf{z}_{iQ}}{\mathbf{z}_{PQ}}\right]_{i=1,\ldots,n} & \left[-\frac{\mathbf{z}_{Pi}}{\mathbf{z}_{PQ}}\right]_{i=1,\ldots,n} \\ O_{1\times n} & -\frac{\mathbf{z}_{AQ}}{\mathbf{z}_{PQ}} & -\frac{\mathbf{z}_{PA}}{\mathbf{z}_{PQ}} \\ O_{1\times n} & -\frac{\mathbf{z}_{BQ}}{\mathbf{z}_{PQ}} & -\frac{\mathbf{z}_{PB}}{\mathbf{z}_{PQ}} \end{bmatrix},$$

where $O_{1\times n}$ is a row vector of length n full of zeroes.

2.6 The S-transformation of variances

The variances are transformed in the following way:

$$\mathrm{Var}\left(\mathbf{z}^{(PQ)}\right) = S^{(PQ)}_{(AB)} \mathrm{Var}\left(\mathbf{z}^{(AB)}\right) \left[S^{(PQ)}_{(AB)}\right]^{\dagger},$$

where

$$\mathbf{z}^{(PQ)} = \begin{bmatrix} \mathbf{z}_i^{(PQ)} \\ \mathbf{\Delta z}_A^{(PQ)} \\ \mathbf{\Delta z}_B^{(PQ)} \end{bmatrix}, \ \mathbf{z}^{(AB)} = \begin{bmatrix} \mathbf{z}_i^{(AB)} \\ \Delta\mathbf{z}_P^{(AB)} \\ \Delta\mathbf{z}_Q^{(AB)} \end{bmatrix}, \text{ ja } \left[S^{(PQ)}_{(AB)}\right]^{\dagger} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{\overline{\mathbf{z}_{iQ}}}{\overline{\mathbf{z}_{PQ}}} & -\frac{\overline{\mathbf{z}_{AQ}}}{\overline{\mathbf{z}_{PQ}}} & -\frac{\overline{\mathbf{z}_{BQ}}}{\overline{\mathbf{z}_{PQ}}} \\ -\frac{\overline{\mathbf{z}_{Pi}}}{\overline{\mathbf{z}_{PQ}}} & -\frac{\overline{\mathbf{z}_{PA}}}{\overline{\mathbf{z}_{PQ}}} & -\frac{\overline{\mathbf{z}_{PB}}}{\overline{\mathbf{z}_{PQ}}} \end{bmatrix}.$$

Here, the delta quantities are $\Delta\mathbf{z}_A^{(PQ)} = \mathbf{z}_A^{(PQ)} - \mathbf{z}_A^{(AB)}$, $\Delta\mathbf{z}_P^{(AB)} = \mathbf{z}_P^{(AB)} - \mathbf{z}_P^{(PQ)}$, etc. As reference value we always use the location that was fixed for the datum point when defining the datum.

2.7 Harjoitukset

Chapter 3

The affine S-transformation

3.1 Triangulation and the finite elements method

The finite elements method is a way to discretize partial differential equations, such as are used, e.g., in statics, structural mechanics, geophysics, meteorology and astrophysics. The domain of computation is divided up into simple parts, finite elements, that have common border lines, surfaces and nodes. The we define *base functions* having value 1 in only one nodal point, and value 0 in all other nodal points. Inside the element, the base functions are simple, e.g., linear functions. Across border lines or surfaces, they are continuous.

The differential equations that are to be solved are now discretized, taking on a form reminiscent of normal equations (Ritz-Galerkin), making possible the solving for the unknowns, i.e., the function values at the nodes.

The most common element is the triangle, in which case we use as base functions linear functions of the co-ordinates. The surface of the Earth may be suitably divided into triangles using so-called *Delaunay triangulation*.

3.2 Bilinear affine transformation

In the publication [Ano03] it is proposed to use for the plane co-ordinate transformation between the Gauß-Krüger projection co-ordinates of ETRS-89 and the *ykj* co-ordinate system, a *triangle-wise affine transformation.*

Inside each triangle, the affine transformation can be written in the form

$$\begin{aligned} x^{(2)} &= \Delta x + a_1 x^{(1)} + a_2 y^{(1)} \\ y^{(2)} &= \Delta y + b_1 x^{(1)} + b_2 y^{(1)} \end{aligned}$$

where $\left(x^{(1)}, y^{(1)}\right)$ are the point co-ordinates in ETRS-GK27, and $\left(x^{(2)}, y^{(2)}\right)$ are the co-ordinates of the same point in *ykj*. This transformation formula has six parameters: Δx, Δy, a_1, a_2, b_1 ja b_2. If, in the three corners of the triangle, are given both $\left(x^{(1)}, y^{(1)}\right)$ and $\left(x^{(2)}, y^{(2)}\right)$, we can solve for these uniquely.

The transformation formula obtained is inside the triangles linear and continuous across the edges, but not differentiable: the scale is discontinuous across triangle edges. Because the mapping is not conformal either, the scale will also be dependent upon the direction considered.

A useful property of triangulation is, that it can be locally "patched": if better data is available in the local area – a denser point set, whose co-ordinate pairs $\left(x^{(i)}, y^{(i)}\right)$, $i = 1, 2$ are known – then we can take away only the triangles of that area and replace them by a larger number of smaller triangle, inside which the transformation will become more precise. This is precisely the procedure that local players, like municipalities, can use to advantage.

The equations above can also be written in vector form:

$$\begin{bmatrix} x^{(2)} \\ y^{(2)} \end{bmatrix} = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} x^{(1)} \\ y^{(1)} \end{bmatrix}.$$

Generally the co-ordinates in the (1)and (2) datums are close to each other, i.e., $\begin{bmatrix} \Delta x & \Delta y \end{bmatrix}^{\mathrm{T}}$ are small. In that case we may write the *shifts*

$$\begin{aligned} \delta x &\equiv x^{(2)} - x^{(1)} = \Delta x + (a_1 - 1)\, x^{(1)} + a_2 y^{(1)}, \\ \delta y &\equiv y^{(2)} - y^{(1)} = \Delta y + b_1 x^{(1)} + (b_2 - 1)\, y^{(1)}. \end{aligned}$$

If we now define

$$\Delta \mathbf{x} \equiv \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \; \mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \equiv \begin{bmatrix} a_1 - 1 & a_2 \\ b_1 & b_2 - 1 \end{bmatrix},$$

we obtain shortly

$$\delta \mathbf{x} = \Delta \mathbf{x} + \mathbf{A}\mathbf{x}^{(1)}.$$

Also in this generally, if the co-ordinates are close together, the elements of $\mathbf{A}$ will be "small". Let there be a triangle ABC. Then we have given the shift vectors of the corners

$$\begin{aligned} \delta \mathbf{x}_A &= \Delta \mathbf{x} + \mathbf{A}\mathbf{x}_A^{(1)}, \\ \delta \mathbf{x}_B &= \Delta \mathbf{x} + \mathbf{A}\mathbf{x}_B^{(1)}, \\ \delta \mathbf{x}_C &= \Delta \mathbf{x} + \mathbf{A}\mathbf{x}_C^{(1)}. \end{aligned}$$

Write this out in components, with $\Delta \mathbf{x}, \mathbf{A}$ on the right hand side:

$$\begin{aligned} \delta x_A &= \Delta x + a_{11} x_A^{(1)} + a_{12} y_A^{(1)} \\ \delta y_A &= \Delta y + a_{21} x_A^{(1)} + a_{22} y_A^{(1)} \\ \delta x_B &= \Delta x + a_{11} x_B^{(1)} + a_{12} y_B^{(1)} \\ \delta y_B &= \Delta y + a_{12} x_B^{(1)} + a_{22} y_B^{(1)} \\ \delta x_C &= \Delta x + a_{11} x_C^{(1)} + a_{12} y_C^{(1)} \\ \delta y_C &= \Delta y + a_{21} x_C^{(1)} + a_{22} y_C^{(1)} \end{aligned}$$

or in matrix form

$$\begin{bmatrix} \delta x_A \\ \delta y_A \\ \delta x_B \\ \delta y_B \\ \delta x_C \\ \delta y_C \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_A^{(1)} & 0 & y_A^{(1)} & 0 \\ 0 & 1 & 0 & x_A^{(1)} & 0 & y_A^{(1)} \\ 1 & 0 & x_B^{(1)} & 0 & y_B^{(1)} & 0 \\ 0 & 1 & 0 & x_B^{(1)} & 0 & y_B^{(1)} \\ 1 & 0 & x_C^{(1)} & 0 & y_C^{(1)} & 0 \\ 0 & 1 & 0 & x_C^{(1)} & 0 & y_C^{(1)} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ a_{11} \\ a_{21} \\ a_{12} \\ a_{22} \end{bmatrix},$$

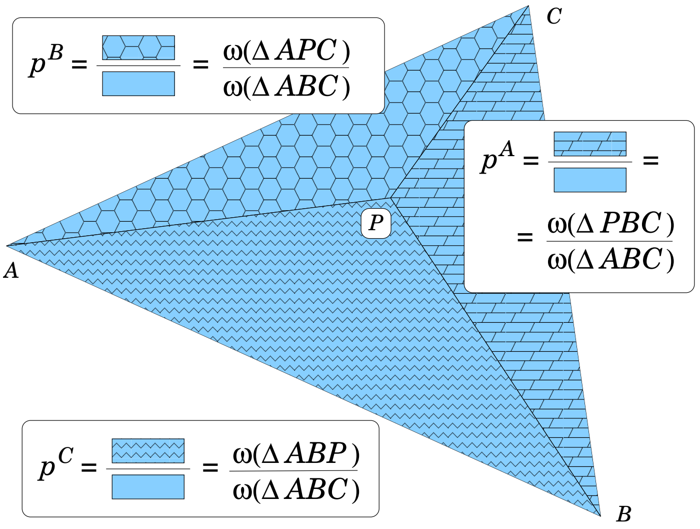

$p^B = \frac{▨}{▭} = \frac{\omega(\Delta APC)}{\omega(\Delta ABC)}$
$p^A = \frac{▨}{▭} = \frac{\omega(\Delta PBC)}{\omega(\Delta ABC)}$
$p^C = \frac{▨}{▭} = \frac{\omega(\Delta ABP)}{\omega(\Delta ABC)}$
A
B
C
P

Figure 3.1: Computing barycentric co-ordinates as the ratio of the surface areas of two triangles

from which they can all be solved.

Let us write the coordinates (x, y) as follows:

$$\begin{aligned} x &= p^A x_A + p^B x_B + p^C x_C, \\ y &= p^A y_A + p^B y_B + p^C y_{C,} \end{aligned}$$

with the additional condition $p^A + p^B + p^C = 1$. Then also

$$\begin{aligned} \delta x &= p^A \delta x_A + p^B \delta x_B + p^C \delta x_C, \quad (3.1) \\ \delta y &= p^A \delta y_A + p^B \delta y_B + p^C \delta y_{C.} \quad (3.2) \end{aligned}$$

The triplet $\left(p^A, p^B, p^C\right)$ is called the *barycentric co-ordinates* of point P See figure 3.1.

They can be found as follows (geometrically $p^A = \frac{\omega(\Delta BCP)}{\omega(\Delta ABC)}$ etc., where ω is the surface area of the triangle) using determinants:

$$p^A = \frac{\begin{vmatrix} x_B & x_C & x \\ y_B & y_C & y \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_A & x_B & x_C \\ y_A & y_B & y_C \\ 1 & 1 & 1 \end{vmatrix}}, \; p^B = \frac{\begin{vmatrix} x_C & x_A & x \\ y_C & y_A & y \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_A & x_B & x_C \\ y_A & y_B & y_C \\ 1 & 1 & 1 \end{vmatrix}}, \; p^C = \frac{\begin{vmatrix} x_A & x_B & x \\ y_A & y_B & y \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_A & x_B & x_C \\ y_A & y_B & y_C \\ 1 & 1 & 1 \end{vmatrix}}.$$

These equations can be directly implemented in software.

3.3 Applying the method of affine transformation in a local situation

If we wish to apply the method proposed in the JHS on the local level, we go through the following steps:

1. Construct a suitable triangulation for the area. Choose from the national triangulation a suitable set of triangles covering the area. Divide up the area in sufficiently small triangles, and formulate the equations for computing the co-ordinate shifts of the corner points of the triangles.
2. Study the error propagation in the chosen geometry and find it to be acceptable.
3. The transformation formulas, coefficients and all, are implemented in software.

The best would be an implementation in which the processing is distributed: the co-ordinates find a server and transformation software suitable for them. A denser and more precise solution is found for some municipalities, for other, the national solution will do. On the Internet, this would be implementable in the frame of an RPC based architecture (e.g., XML/SOAP).

3.4 A theoretical analysis of error propagation

The precision behaviour of the method can be studied by simulating the computation of co-ordinates with synthetic but realistic-looking errors. We can also use real observational material, from which we can leave out one point at a time, and investigate how well this approximation method succeeds in reproducing this point's co-ordinate shifts (*cross-validation*).

On the other hand we can also investigate the problem theoretically. We can start from the knowledge that the "old" network, from which the *ykj* co-ordinates originate, was measured by traditional triangulation and polygon measurements that have a certain known precision behaviour[1]. Todellisen Instead of the true precision, we often use a so-called *criterion variance matrix* [Baa73], which describes in a simple mathematical way the spatial behaviour of the precision of the point field.

3.4.1 Affine transformations

In the same way as for similarity transformations, we can treat the error propagation of affine transformations formally.

If we have three points A, B, C the co-ordinates of which are given, then the co-ordinate correction of an arbitrary point $\mathbf{z}_i$ can be written as follows (complexly):

$$\mathbf{z}'_i = \mathbf{z}_i + p_i^A \left(\mathbf{z}'_A - \mathbf{z}_A\right) + p_i^B \left(\mathbf{z}'_B - \mathbf{z}_B\right) + p_i^C \left(\mathbf{z}'_C - \mathbf{z}_C\right).$$

Again in matrix form:

$$\mathbf{z}'_i = \begin{bmatrix} 1 & -p_i^A & -p_i^B & -p_i^C \end{bmatrix} \begin{bmatrix} \mathbf{z}_i \\ \mathbf{z}_A - \mathbf{z}'_A \\ \mathbf{z}_B - \mathbf{z}'_B \\ \mathbf{z}_C - \mathbf{z}'_C \end{bmatrix}.$$

Here again $\mathbf{z}'_A, \mathbf{z}'_B, \mathbf{z}'_C$ are the fixed co-ordinates given as the (ABC) datum definition.

We write the affine datum transformations again in the familiar form[2] (equation 2.5):

[1] Compared to them, GPS measurements can be considered absolutely precise.

[2] Change of notation: $\mathbf{z} \to \mathbf{z}^{(ABC)}$ and $\mathbf{z}' \to \mathbf{z}^{(PQR)}$.

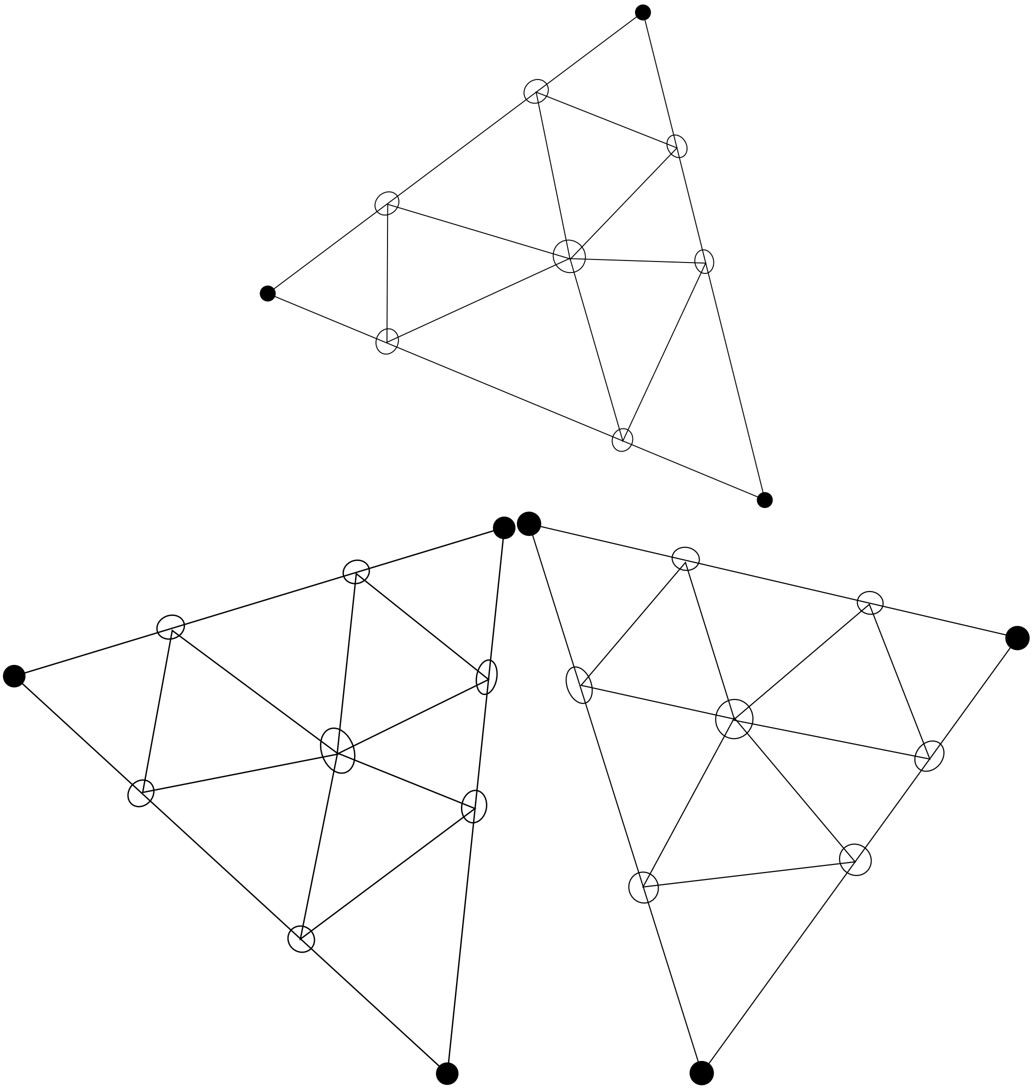


Figure 3.2: Error propagation in triangles of different sizes. Only qualitatively.

$$\begin{bmatrix} \mathbf{z}_i^{(PQR)} \\ \mathbf{z}_A^{(PQR)} - \mathbf{z}_A^{(ABC)} \\ \mathbf{z}_B^{(PQR)} - \mathbf{z}_B^{(ABC)} \\ \mathbf{z}_C^{(PQR)} - \mathbf{z}_C^{(ABC)} \end{bmatrix} = \begin{bmatrix} 1 & -p_i^P & -p_i^Q & -p_i^R \\ 0 & -p_A^P & -p_A^Q & -p_A^R \\ 0 & -p_B^P & -p_B^Q & -p_B^R \\ 0 & -p_C^P & -p_C^Q & -p_C^R \end{bmatrix} \begin{bmatrix} \mathbf{z}_i^{(ABC)} \\ \mathbf{z}_P^{(ABC)} - \mathbf{z}_P^{(PQR)} \\ \mathbf{z}_Q^{(ABC)} - \mathbf{z}_Q^{(PQR)} \\ \mathbf{z}_R^{(ABC)} - \mathbf{z}_R^{(PQR)} \end{bmatrix}.$$

Here all elements (p values) are, otherwise than in the case of a similarity transformation (S-transformation), all real valued.

Let us again write symbolically:

$$\mathbf{S}_{(ABC)}^{(PQR)} \equiv \begin{bmatrix} 1 & -p_i^P & -p_i^Q & -p_i^R \\ 0 & -p_A^P & -p_A^Q & -p_A^R \\ 0 & -p_B^P & -p_B^Q & -p_B^R \\ 0 & -p_C^P & -p_C^Q & -p_C^R \end{bmatrix},$$

where the p values are computed as explained before:

$$p_A^P = \frac{\omega\left(\Delta QRA\right)}{\omega\left(\Delta PQR\right)} = \frac{\begin{vmatrix} x_Q & x_R & x_A \\ y_Q & y_R & y_A \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_P & x_Q & x_R \\ y_P & y_Q & y_R \\ 1 & 1 & 1 \end{vmatrix}},$$

etc. For humans, this is hard, but not for computers.

Also affine transformations form a mathematical *group*. Two successive affine transformations $(ABC) \to (UVW) \to (PQR)$ produce again an affine transformation, the inverse transformation of $(ABC) \to (PQR)$, i.e., $(PQR) \to (ABC)$ does so as well, and the trivial tramnsformation $(ABC) \to (ABC)$ does also.

3.4.2 The affine transformation and the criterion matrix

We start again from the standard form of the criterion matrix 2.2, 2.3:

$$\begin{aligned} \mathrm{Var}\left(\mathbf{z}\right) &= \alpha^2, \\ \mathrm{Cov}\left(\mathbf{z},\mathbf{w}\right) &= \alpha^2 - \frac{1}{2}\sigma^2\left(\mathbf{z}-\mathbf{w}\right)\left(\overline{\mathbf{z}}-\overline{\mathbf{w}}\right). \end{aligned}$$

Propagation of variances yields

$$\begin{aligned} \mathrm{Var}\left(\mathbf{z}_i'\right) &= \begin{bmatrix} 1 & -p_i^A & -p_i^B & -p_i^C \end{bmatrix} \mathrm{Var}\left(\mathbf{z}_i,\mathbf{z}_A,\mathbf{z}_B,\mathbf{z}_C\right) \begin{bmatrix} 1 \\ -p_i^A \\ -p_i^B \\ -p_i^C \end{bmatrix} = \\ &= \begin{bmatrix} 1 & -p_i^A & -p_i^B & -p_i^C \end{bmatrix} \cdot \\ &\cdot \begin{bmatrix} \alpha^2 & \alpha^2 - \frac{1}{2}\sigma^2\mathbf{z}_{iA}\overline{\mathbf{z}_{iA}} & \alpha^2 - \frac{1}{2}\sigma^2\mathbf{z}_{iB}\overline{\mathbf{z}_{iB}} & \alpha^2 - \frac{1}{2}\sigma^2\mathbf{z}_{iC}\overline{\mathbf{z}_{iC}} \\ \alpha^2 - \frac{1}{2}\sigma^2\mathbf{z}_{iA}\overline{\mathbf{z}_{iA}} & \alpha^2 & \alpha^2 - \frac{1}{2}\sigma^2\mathbf{z}_{AB}\overline{\mathbf{z}_{AB}} & \alpha^2 - \frac{1}{2}\sigma^2\mathbf{z}_{AC}\overline{\mathbf{z}_{AC}} \\ \alpha^2 - \frac{1}{2}\sigma^2\mathbf{z}_{iB}\overline{\mathbf{z}_{iB}} & \alpha^2 - \frac{1}{2}\sigma^2\mathbf{z}_{AB}\overline{\mathbf{z}_{AB}} & \alpha^2 & \alpha^2 - \frac{1}{2}\sigma^2\mathbf{z}_{BC}\overline{\mathbf{z}_{BC}} \\ \alpha^2 - \frac{1}{2}\sigma^2\mathbf{z}_{iC}\overline{\mathbf{z}_{iC}} & \alpha^2 - \frac{1}{2}\sigma^2\mathbf{z}_{AC}\overline{\mathbf{z}_{AC}} & \alpha^2 - \frac{1}{2}\sigma^2\mathbf{z}_{BC}\overline{\mathbf{z}_{BC}} & \alpha^2 \end{bmatrix} \begin{bmatrix} 1 \\ -p_i^A \\ -p_i^B \\ -p_i^C \end{bmatrix} \end{aligned}$$

Note that again, the sum of elements of this row vector, $1 - p_i^A - p_i^B - p_i^C = 0$ and α^2 drops out of the equation. We obtain

$$\begin{aligned}
\mathrm{Var}\left(\mathbf{z}_i'\right) &= \sigma^2 \left[\begin{array}{cccc} 1 & -p_i^A & -p_i^B & -p_i^C \end{array}\right] \cdot \\
&\cdot \left[\begin{array}{cccc} 0 & -\frac{1}{2}\mathbf{z}_{iA}\overline{\mathbf{z}_{iA}} & -\frac{1}{2}\mathbf{z}_{iB}\overline{\mathbf{z}_{iB}} & -\frac{1}{2}\mathbf{z}_{iC}\overline{\mathbf{z}_{iC}} \\ -\frac{1}{2}\mathbf{z}_{iA}\overline{\mathbf{z}_{iA}} & 0 & -\frac{1}{2}\mathbf{z}_{AB}\overline{\mathbf{z}_{AB}} & -\frac{1}{2}\mathbf{z}_{AC}\overline{\mathbf{z}_{AC}} \\ -\frac{1}{2}\mathbf{z}_{iB}\overline{\mathbf{z}_{iB}} & -\frac{1}{2}\mathbf{z}_{AB}\overline{\mathbf{z}_{AB}} & 0 & -\frac{1}{2}\mathbf{z}_{BC}\overline{\mathbf{z}_{BC}} \\ -\frac{1}{2}\mathbf{z}_{iC}\overline{\mathbf{z}_{iC}} & -\frac{1}{2}\mathbf{z}_{AC}\overline{\mathbf{z}_{AC}} & -\frac{1}{2}\mathbf{z}_{BC}\overline{\mathbf{z}_{BC}} & 0 \end{array}\right] \left[\begin{array}{c} 1 \\ -p_i^A \\ -p_i^B \\ -p_i^C \end{array}\right] = \\
&= -\frac{1}{2}\sigma^2 \left[\begin{array}{cccc} 1 & -p_i^A & -p_i^B & -p_i^C \end{array}\right] \cdot \\
&\cdot \left[\begin{array}{cccc} 0 & \|\mathbf{z}_{iA}\|^2 & \|\mathbf{z}_{iB}\|^2 & \|\mathbf{z}_{iC}\|^2 \\ \|\mathbf{z}_{iA}\|^2 & 0 & \|\mathbf{z}_{AB}\|^2 & \|\mathbf{z}_{AC}\|^2 \\ \|\mathbf{z}_{iB}\|^2 & \|\mathbf{z}_{AB}\|^2 & 0 & \|\mathbf{z}_{BC}\|^2 \\ \|\mathbf{z}_{iC}\|^2 & \|\mathbf{z}_{AC}\|^2 & \|\mathbf{z}_{BC}\|^2 & 0 \end{array}\right] \left[\begin{array}{c} 1 \\ -p_i^A \\ -p_i^B \\ -p_i^C \end{array}\right].
\end{aligned}$$

Unfortunately we cannot readily make this formula neater. This is no problem, however, for the computer.

3.5 The case of height measurement

In height measurement, the quantity being studied is a scalar, h, which nevertheless is a function of location $\mathbf{z}$. Therefore we may write $h(\mathbf{z})$.

In the case of height measurement we know, that the *relative* or inter-point errors grow with the *square root* of the distance between the points (because the measurement method is levelling). For this reason it is wise to study the more general case where the error is proportional to some power ν of the distance, which thus generally is not $\nu = 1$ but $\nu = 0.5$.

Then we can (still in the case of location co-ordinates) define the standard form of the criterion matrix as follows:

$$\begin{aligned}
\mathrm{Var}(\mathbf{z}) &= \alpha^2, \\
\mathrm{Cov}(\mathbf{z}, \mathbf{w}) &= \alpha^2 - \frac{1}{2}\sigma^2 (\mathbf{z} - \mathbf{w})^\nu (\overline{\mathbf{z}} - \overline{\mathbf{w}})^\nu = \\
&= \alpha^2 - \frac{1}{2}\sigma^2 \|\mathbf{z} - \mathbf{w}\|^{2\nu}.
\end{aligned}$$

We again obtain for the relative variance

$$\begin{aligned}
\mathrm{Var}(\mathbf{z}_{AB}) &= \mathrm{Var}(\mathbf{z}_A) + \mathrm{Var}(\mathbf{z}_B) - 2\mathrm{Cov}(\mathbf{z}_A, \mathbf{z}_B) = \\
&= 2\alpha^2 - 2\alpha^2 + \sigma^2 (\mathbf{z}_{AB}\overline{\mathbf{z}_{AB}})^\nu = +\sigma^2 (\mathbf{z}_{AB}\overline{\mathbf{z}_{AB}})^\nu.
\end{aligned}$$

Let us apply this now to height measurement. We obtain ($\nu = 0.5$)

$$\mathrm{Var}(\Delta h_{AB}) = \sigma^2 \|\mathbf{z}_{AB}\|$$

and

$$\sigma_{\Delta h_{AB}} = \sigma\sqrt{\|\mathbf{z}_{AB}\|},$$

as is well known.

In realistic networks, however, due to the increased strength brought by the network adjustment, also in the case of location networks $\nu < 1$, and for levelling networks we may have $v < 0.5$. The values given here are however a decent first approximation.

In the case of *GPS measurements* we know, that the relative precision of point locations can be well described by a power law of the distance with an exponent of $\nu \approx 0.5$ (the so-called Bernese rule-of-thumb).

Chapter 4

Determining the shape of an object (circle, sphere, straight line)

(More generally: building and parametrizing models in preparation for adjustment)

Literature:

[Kal98b, s. 140-143]

[Kal98a]

[Kra83]

[Nor99a]

[SB97, s. 441-444]

[Lei95, s. 123-130]

4.1 The general case

Let be given a figure in the plane, on the edge of which

$$f(x, y) = 0.$$

Edge points of this figure have been measured n times:

$$(x_i, y_i), i = 1, \ldots, n$$

Let us assume, that the shape of the figure depends on exterior parameters a_j, $j = 1, \ldots, m$. I.e.,

$$f(x, y; a_j) = 0.$$

Let us call the observations $(x_i, y_i), i = 1, \ldots, n$. We construct *approximate values* that are sufficiently close to the observations, and for which holds

$$f\left(x_i^{(0)}, y_i^{(0)}; a_j^{(0)}\right) = 0.$$

Now we can write the Taylor expansion:

$$f(x_i, y_i; a_j) = f\left(x_i^{(0)}, y_i^{(0)}; a_j^{(0)}\right) + \left.\frac{\partial f}{\partial x}\right|_{x=x_i^0} \Delta x_i + \left.\frac{\partial f}{\partial y}\right|_{y=y_i^0} \Delta y_i + \sum_{j=1}^{m} \frac{\partial f}{\partial a_j} \Delta a_j,$$

where

$$\Delta x_i = x_i - x_i^{(0)}, \Delta y_i = y_i - y_i^{(0)}, \text{ ja } \Delta a_j = a_j - a_j^{(0)}.$$

The expression $f(x_i, y_i; a_j) - f\left(x_i^{(0)}, y_i^{(0)}; a_j^{(0)}\right)$ *must vanish.*

This is how we obtain our final observation equation

$$\frac{\partial f}{\partial x_i}\Delta x_i + \frac{\partial f}{\partial y_i}\Delta y_i + \sum_{j=1}^{m} \frac{\partial f}{\partial a_j}\Delta a_j = 0.$$

Here, the two left hand side terms constitute a linear combination of the edge point observations (x_i, y_i) which is computable if the partial derivatives of the edge function $f(x, y; a_j)$ with respect to x and y can be computed. The same for the elements of the design matrix $\frac{df}{da_j}$.

More generally, if we had, instead of a curve, a surface in three-dimensional space, we would obtain as observation equations:

$$\frac{\partial f}{\partial x_i}\Delta x_i + \frac{\partial f}{\partial y_i}\Delta y_i + \frac{\partial f}{\partial z_i}\Delta z_i + \sum_{j=1}^{m} \frac{\partial f}{\partial a_j}\left(a_j - a_j^0\right) = 0.$$

If the observations (x_i, y_i) have the same weight (and are equally precise in the x and y directions), we must still require, that

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x_i}\right)^2 + \left(\frac{\partial f}{\partial y_i}\right)^2 + \left(\frac{\partial f}{\partial z_i}\right)^2}$$

is a constant, in other words, does not depend on the values of x_i and y_iOnly then are the variances of the “replacement observable” $\ell_i \equiv \frac{\partial f}{\partial x_i}\Delta x_i + \frac{\partial f}{\partial y_i}\Delta y_i + \frac{\partial f}{\partial z_i}\Delta z_i$ the same, and one may use a unit matrix as the weight coefficient matrix.

4.2 Example: circle

The equation for the circle is

$$x^2 + y^2 = r^2,$$

where r is the circle's radius. The equation for a freely positioned circle is

$$(x - X)^2 + (y - Y)^2 = r^2,$$

where (X, Y) are the co-ordinates of the circle's centre.

The function f is

$$f(x, y; a_j) = (x - X)^2 + (y - Y)^2 - r^2$$

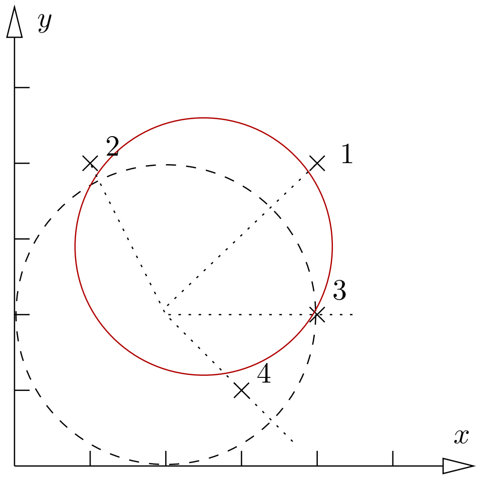
and the vector a_j:

$$\mathrm{a} = \begin{bmatrix} X \\ Y \\ r \end{bmatrix}.$$

Partial derivatives:

$$\frac{\partial f}{\partial x} = 2(x - X), \frac{\partial f}{\partial y} = 2(y - Y), \frac{\partial f}{\partial a_1} = -2(x - X), \frac{\partial f}{\partial a_2} = -2(y - Y), \frac{\partial f}{\partial a_3} = -2r.$$

These partial derivatives are evaluated at suitable approximate values $X^{(0)}, Y^{(0)}, r^{(0)}$.



We get as *observation equations*

$$\left(x_i^{(0)} - X^{(0)}\right)\Delta x_i + \left(y_i^{(0)} - Y^{(0)}\right)\Delta y_i - \left(x_i^{(0)} - X^{(0)}\right)\Delta X - \left(y_i^{(0)} - Y^{(0)}\right)\Delta Y - r^{(0)}\Delta r = 0,$$

from which the linearized unknowns $\Delta X, \Delta Y$ and Δr (corrections to the assumed approximate values) can be solved if the number of equations exceeds three.

Let the following observation points be given: $(4,4), (1,4), (4,2)$ and $(3,1)$. Let the starting or approximate values be $X^0 = Y^0 = 2$,$r^0 = 2$. We obtain approximate values for the observations as follows. From the figure we see, that

$$\begin{aligned} x_i^{(0)} &= X^{(0)} + r^{(0)}\cos\varphi_i^{(0)}, \\ y_i^{(0)} &= Y^{(0)} + r^{(0)}\sin\varphi_i^{(0)}, \end{aligned}$$

where φ is the direction angle. Graphically we obtain suitable values for φ, and thus

i	$\varphi^{(0)}$	x_i	y_i	$x_i^{(0)}$	$y_i^{(0)}$	Δx_i	Δy_i	$x_i^{(0)} - X^{(0)}$	$y_i^{(0)} - Y^{(0)}$
1	45°	4	4	3.414	3.414	0.586	0.586	1.414	1.414
2	120°	1	4	1.000	3.732	0.000	0.268	-1.000	1.732
3	0°	4	2	4.000	2.000	0.000	0.000	2.000	0.000
4	−45°	3	1	3.414	0.586	-0.414	0.414	1.414	-1.414

Thus we obtain

$$\left(x_i^{(0)} - X^{(0)}\right)\Delta x_i + \left(y_i^{(0)} - Y^{(0)}\right)\Delta y_i = \begin{bmatrix} 1.657 \\ 0.464 \\ 0.000 \\ -1.171 \end{bmatrix},$$

and we get for our observation equation

$$\begin{bmatrix} 1.657 \\ 0.464 \\ 0.000 \\ -1.171 \end{bmatrix} = \begin{bmatrix} 1.414 & 1.414 & 2 \\ -1.000 & 1.732 & 2 \\ 2.000 & 0.000 & 2 \\ 1.414 & -1.414 & 2 \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta r \end{bmatrix}.$$

Solving this by means of Matlabin/Octave yields $\Delta X = 0.485, \Delta Y = 0.966, \Delta r = -0.322$, and thus $X = 2.485, Y = 2.966, r = 1.678$. This solution was drawn into the graphic. As can be seen is the solution for r rather poor, which undoubtedly is due to nonlinearity together with poor starting values. Iteration would improve the solution.

4.3 Exercises

If we wish to determine a straight line going through a point cloud, we have the following regression alternatives:

- *Traditional linear regression*:
$$y_i + v_i = a + bx_i,$$
minimizes $s \equiv \sum_i v_i^2$ by means of the variables a, b.

  Esim.E.g., calculating a: demand
$$\frac{\partial s}{\partial a} = 0 = \frac{\partial}{\partial a}\sum_i (a + bx_i - y_i)^2 = 2n(a + b\overline{x} - \overline{y}),$$
where $\overline{x}, \overline{y}$ are the co-ordinates of the *centre of mass* of the n measured points (x_i, y_i) If we choose the origin or our co-ordinates such that
$$\overline{x} = \overline{y} = 0$$
(so-called barycentric co-ordinates), it follows that $2na = 0 \Rightarrow a = 0$. *In other words, the regression line goes through the centre of mass of the measured points.*
- Interchange the roles of x and y *and perform a* regression of x with respect to y:
$$y_i = a + b(x_i + w_i),$$
minimize $\sum w_i^2$.
- Orthogonal regression:
$$y_i - u_i\sqrt{1-b^2} = a + b(x_i + bu_i),$$
minimize $\sum u_i^2$.

1. *Prove* that also when regressing x with respect to y, the regression line goes through the centre of mass of the measured points.

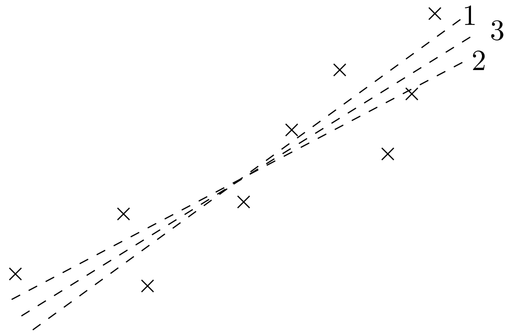
   Answer: $s = \sum w_i^2$ is to be minimized. Calculate in the same way
$$\begin{aligned}\frac{ds}{da} &= \frac{d}{da}\sum w_i^2 = \sum \frac{d}{da}\left(\frac{y_i - a - bx_i}{b}\right)^2 = \\ &= 2 \cdot -\frac{1}{b} \cdot \sum_i \left(\frac{y_i - a - bx_i}{b}\right) = -\frac{2n}{b^2}(\overline{y} - a - b\overline{x});\end{aligned}$$
in barycentric co-ordinates $\overline{x} = \overline{y} = 0$ this becomes zero **iff** (if and only if):
$$\frac{2na}{b^2} = 0 \Rightarrow a = 0.$$
(A requirement in this case is, that $b \neq 0$.)

2. The same proof in the case of orthogonal regression.

   Answer: Minimize $s = \sum u_i^2$.
$$\begin{aligned}\frac{ds}{da} &= \frac{d}{da}\sum_i \left(\frac{a + bx_i - y_i}{b^2 + \sqrt{1-b^2}}\right)^2 = \\ &= 2 \cdot \frac{1}{b^2 + \sqrt{1-b^2}} \cdot \sum_i \frac{a + bx_i - y_i}{b^2 + \sqrt{1-b^2}} = \\ &= \frac{2n}{\left(b^2 + \sqrt{1-b^2}\right)^2}(a + b\overline{x} - \overline{y}),\end{aligned}$$

which again vanishes in the case of $\overline{x} = \overline{y} = 0$ **iff** $a = 0$. In this case there are *no* restrictions on b, i.e., this will work even if $b = 0$ or $b = 1$ or in-between those values.

3. In the figure are drawn the lines found by the three different regression methods. *Identify* them according to the above classification.



Answer: The least tilted line is the traditional regression. The middle one is the orthogonal regression.The steepest line is the regression of x with respect to y.

Chapter 5

3D network, industrial measurements with a system of several theodolites

Literature:

[Nor99b]

[Nor99a]

[Kär93]

[SB97, s. 363-368]

[Sal95, ss. 17-31]

5.1 Three dimensional theodolite measurement (EPLA)

We describe here the method according to the model of Cooper and Allan. In this method, we observe with theodolites that are oriented with respect ot each other, the point P.

In this method, the distance in space between the lines AP and BP is minimized. The directions of these lines are represented by the unit vectors $\mathbf{p}$ and $\mathbf{q}$, respectively; the inter-theodolite vector is $\mathbf{b}$. These vectors are now, with the origin in point A and the X axis along AB:

$$\mathbf{b} = \begin{bmatrix} x_B \\ 0 \\ z_B \end{bmatrix},$$

$$\mathbf{p} = \begin{bmatrix} \sin\eta_A \cos\angle A \\ \sin\eta_A \sin\angle A \\ \cos\eta_A \end{bmatrix}, \mathbf{q} = \begin{bmatrix} -\sin\eta_B \cos\angle B \\ \sin\eta_B \sin\angle B \\ \cos\eta_B \end{bmatrix}.$$

Now the closing vector is

$$\mathbf{e} = -\lambda\mathbf{p} + \mathbf{b} + \mu\mathbf{q},$$

where we choose the multipliersλ and μ so as to minimise the length or norm of $\mathbf{e}$. Then, the best estimate of point P will be the centre point of $\mathbf{e}$, i.e.

$$\mathbf{x}_P = \frac{1}{2}\left(\lambda\mathbf{p} + \mathbf{b} + \mu\mathbf{q}\right).$$

Below we provide an example of how to do the computation in an approximate fashion.

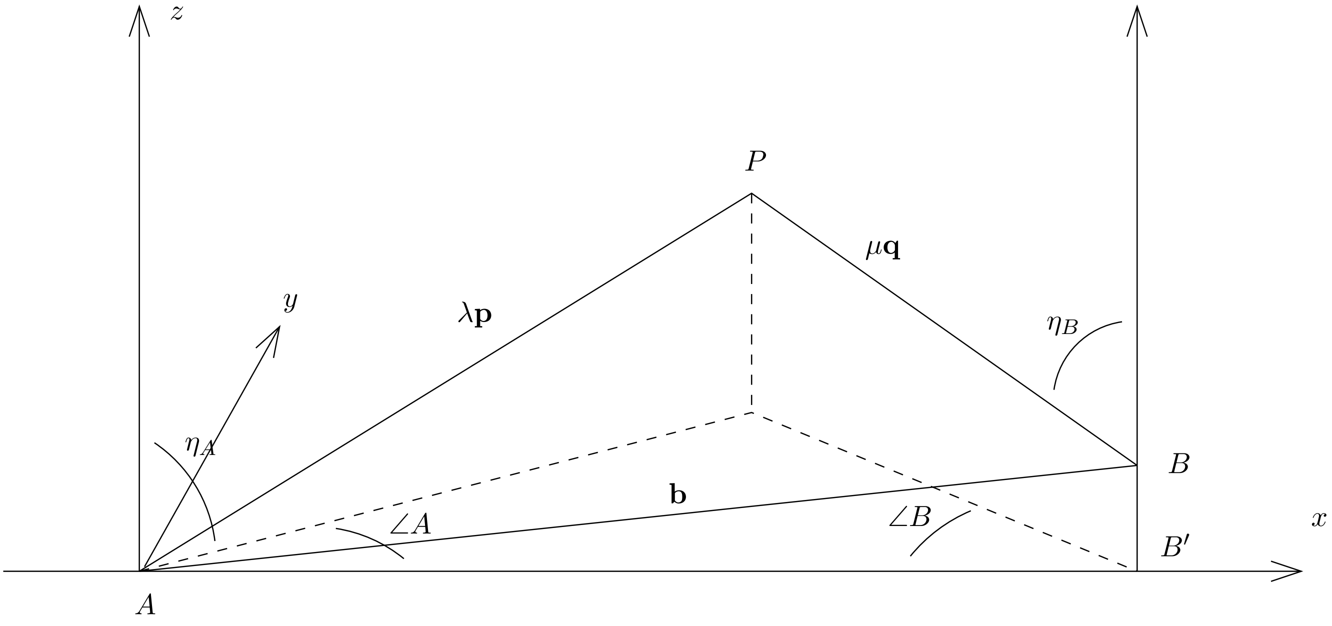


Figure 5.1: Cooper & Allan method

5.2 Example

The point P is observed using two theodolites from two standpoints. The co-ordinates of the standpoints A, B in the co-ordinate frame of the factory hall:

	x (m)	y (m)	z (m)
A	9.00	12.40	2.55
B	15.45	16.66	2.95

The angle measurements are the following:

	Horizontal (gon) $\angle$	Vertical η (gon)
A	61.166	14.042
B	345.995	9.081

The measurements are done in a local co-ordinate system defined by the theodolites. See the figure. We minimize the length of the vector

$$\mathbf{e} \equiv -\lambda\mathbf{p} + \mathbf{b} + \mu\mathbf{q}$$

. $\mathbf{p}$ and $\mathbf{q}$ are unit vectors.

Approximative method:

1. Compute *at first only* the *horizontal* co-ordinates x, y of point P in the local co-ordinate frame defined by the theodolites.
2. Compute the parameters λ ja μ.

3. Compute the length $\|\mathbf{e}\|$of the vertical difference vector $\mathbf{e}$.

   Answer: let us first compute the projection of $\mathbf{b} = AB$:n upon the horizontal plane:

$$\|\mathbf{b}_\perp\| = \sqrt{(15.45-9.00)^2 + (16.66-12.40)^2} = \sqrt{41.6025+18.1476} = 7.7298\,\text{m}.$$

   In the horizontal plane triangle ABP is

$$\angle P = 200 - 61.166 - (400 - 345.995) = 84.829\,\text{gon}.$$

   The sine rule:

$$AP_\perp = AB_\perp \frac{\sin\angle B}{\sin\angle P} = 7.7298 \frac{0.75016}{0.97174} = 5.9672\,\text{m}.$$

   This distance is in the horizontal plane. *In space* $AP = AP_\perp / \cos\eta_A = 6.1510\,\text{m} = \lambda$.

   Now, using the vertical angle

$$z_P = z_A + AP_\perp \tan\eta_A = 2.55 + 5.9672 \tan 14.042 \text{ gon} = 3.8880\,\text{m}.$$

$$BP_\perp = AB_\perp \frac{\sin\angle A}{\sin\angle P} = 7.7298 \frac{0.81965}{0.97174} = 6.5200\,\text{m}.$$

   Again, this distance is in the horizontal plane. *In space* $BP = BP_\perp / \cos\eta_B = 6.6028\,\text{m} = \mu$.

$$z_P = z_B + BP_\perp \tan\eta_B = 2.95 + 6.5200 \tan 9.081 \text{ gon} = 3.8864\,\text{m}.$$

   So

$$\|\mathbf{e}\| = z_{P,B} - z_{P,A} = 3.8864 - 3.8880 = -1.6\,\text{mm}.$$

   Co-ordinates in the system defined by the theodolite (origin point A, x axis direction AB, z axis up):

$$\begin{aligned} x_P &= AP_\perp \cos\angle A = 5.9672 \cdot 0.5729\,\text{m} = 3.4184\,\text{m} \\ y_P &= AP_\perp \sin\angle A = 5.9672 \cdot 0.81965\,\text{m} = 4.8910\,\text{m} \\ z_P &= \frac{1}{2}(z_{P,1} + z_{P,2}) - z_A = 3.8872\,\text{m} - 2.55\,\text{m} = 1.3372\,\text{m} \end{aligned}$$

We must note that this is an *approximate* method, which is acceptable only if the vertical angles η are close to 100^{g}. Of course also an exact least squares solution is possible.

5.3 Different ways to create the scale

- By knowing the distance between the theodolite points A,B
- By using a known scale rod or staff
- By including at least two points the distance between which is known, into the measurement set-up.

Chapter 6

Deformation analysis

Literature:

[Aho01]

[FIG98, s. 191-256]

[Kal98b, s. 95-101]

[Coo87, s. 331-352]

[VK86, s. 611-659]

6.1 One dimensional deformation analysis

The simplest case is that, where the same levelling line or network has been measured twice:

$$\begin{aligned} &h_i(t_1), \qquad i = 1, \ldots, n \\ &h_i(t_2), \qquad i = 1, \ldots, n \end{aligned}$$

and the corresponding variance matrices of the heights are available: $Q(t_1)$ and $Q(t_2)$.

Obviously the comparison is possible only, if both measurements are first reduced to the same reference point of *datum*. E.g., choose the first point as the datum point:

$$h_1^{(1)}(t_1) = h_1^{(1)}(t_2) \ \ (= \text{some known value, e.g., } 0)$$

After this the variance matrices for both measurement times or *epochs* are only of size $(n-1) \times (n-1)$, because now point 1 is *known* and no longer has (co-)variances.

$$Q^{(1)}(t_1) = \begin{bmatrix} q_{22}^{(1)} & q_{23}^{(1)} & \cdots & q_{2n}^{(1)} \\ q_{32}^{(1)} & q_{33}^{(1)} & \cdots & q_{3n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ q_{n2}^{(1)} & q_{n3}^{(1)} & \cdots & q_{nn}^{(1)} \end{bmatrix},$$

and the same for $Q^{(1)}(t_2)$. Here

$$\begin{aligned} q_{ii}^{(k)} &= \operatorname{Var}\left(h_i^{(k)}\right), \\ q_{ij}^{(k)} &= \operatorname{Cov}\left(h_i^{(k)}, h_j^{(k)}\right). \end{aligned}$$

Now, calculate the height differences between the two epochs and their variances, assuming that the masurements made at times t_1 and t_2 are statistically independent of each other:

$$\Delta h_i^{(1)} = h_i^{(1)}(t_2) - h_i^{(1)}(t_1), \ i = 2, \ldots, n;$$

$$Q^{(1)}_{\Delta h \Delta h} = Q^{(1)}(t_1) + Q^{(1)}(\mathrm{t}_2).$$

After this it is intuitively clear that the following quantity has the χ^2_{n-1} distribution:

$$\mathcal{E} = \left[\underline{\Delta \mathrm{h}}^{(1)}\right]^{\mathrm{T}} \left[Q^{(1)}_{\Delta h \Delta h}\right]^{-1} \underline{\Delta \mathrm{h}}^{(1)}.$$

Statistical testing uses this quantity. Here

$$\underline{\Delta \mathrm{h}}^{(1)} = \begin{bmatrix} \Delta h_2^{(1)} \\ \Delta h_3^{(1)} \\ \vdots \\ \Delta h_n^{(1)} \end{bmatrix}$$

is the vector of height differences.

6.2 Two dimensional deformation analysis

This goes in the same way as in the one dimensional case, except that

1. the co-ordinates are treated as complex numbers, and
2. there are two datum points, the co-ordinates of which are considered known.

So, if there are n points, then the size of the variance matrix is now $(n-2) \times (n-2)$. Also the variance matrix is complex valued.

The testing variate is

$$\mathcal{E} = \left[\underline{\mathrm{d}}^{(AB)}\right]^{\dagger} \left[\mathrm{Q}^{(AB)}_{dd}\right]^{-1} \underline{\mathrm{d}}^{(AB)},$$

where $\underline{\mathrm{d}}$ is the complex vector of all co-ordinate differences:

$$\underline{\mathrm{d}}^{(AB)} = \begin{bmatrix} x_3^{(AB)}(t_2) - x_3^{(AB)}(t_1) + i\left[y_3^{(AB)}(t_2) - y_3^{(AB)}(t_1)\right] \\ x_4^{(AB)}(t_2) - x_4^{(AB)}(t_1) + i\left[y_4^{(AB)}(t_2) - y_4^{(AB)}(t_1)\right] \\ \\ x_n^{(AB)}(t_2) - x_n^{(AB)}(t_1) + i\left[y_n^{(AB)}(t_2) - y_n^{(AB)}(t_1)\right] \end{bmatrix}.$$

AB is the chosen datum or starting point for both epochs t_1 and t_2. The other points are numbered $3, 4, \ldots, n$. The symbol † signifies both transposition and complex conjugate, the so called *hermitian*:

$$A^{\dagger} \equiv \overline{A^T} = \overline{A}^T.$$

Warning

In Cooper's book [Coo87, s. 335] there is an *error* under equation (9.52), the right equation is (inverse, not transpose):

$$\Omega = \widehat{\mathbf{d}}^{\mathrm{t}} Q_{\mathrm{d}}^{-1} \widehat{\mathbf{d}}.$$

6.3 Example

Let the adjusted co-ordinates $\mathbf{x}_i(t_1)$, $i = 1, \ldots, 4$ of the deformation network from the first measurement epoch be the following:

Point	x (m)	y (m)
1	1234.123	2134.453
2	2224.045	2034.487
3	2232.495	975.456
4	1148.865	879.775

and the co-ordinates of the second measurement epoch $\mathbf{x}_i(t_2)$, $i = 1, \ldots, 4$ be the following:

Point	x (m)	y (m)
1	1234.189	2134.485
2	2224.004	2034.433
3	2232.451	975.497
4	1148.929	879.766

Intermezzo: so we are computing:

d	x	y
1	-0.066	-0.032
2	+0.041	+0.054
3	+0.044	-0.041
4	-0.064	+0.009

$$\underline{\mathrm{d}}^T\underline{\mathrm{d}} = \sum_{i=1}^{4}\left[\{x_i(t_2) - x_i(t_1)\}^2 + \{y_i(t_2) - y_i(t_1)\}^2\right] = 0.017771\mathrm{m}^2$$

(Similarly with complex numbers:

$$\underline{\mathrm{d}}^\dagger\underline{\mathrm{d}} = \sum_{i=1}^{4}\{\overline{\mathbf{z}_i}(t_2) - \overline{\mathbf{z}_i}(t_1)\}\{\mathbf{z}_i(t_2) - \mathbf{z}_i(t_1)\} = 0.017771\mathrm{m}^2,$$

as can be verified by computation. Here $\mathbf{z}_i \equiv x_i + iy_i$ and $\overline{\mathbf{z}_i} = x_i - iy_i$.)

Let the precisions (mean co-ordinate errors) of the co-ordinates of the first epoch be $x_i(t_1)$ ja $y_i(t_1)$ $m_{0,1} = \pm 5\,\mathrm{cm}$, and the precisions of the co-ordinates of the second measurement $x_i(t_2), y_i(t_2)$ tarkkuudet $m_{0,2} = \pm 1\,\mathrm{cm}$. The variance matrices of the co-ordinate vectors are thus $Q_1 = m_{0,1}^2 I$ and $Q_2 = m_{0,2}^2 I$.

1. Compute the mean value m_0 of a single co-ordinate difference $\Delta x = x(t_2) - x(t_1)$. Propagation of variances yields

   $$m_0^2 = m_{0,1}^2 + m_{0,2}^2 = (25 + 1)\ \mathrm{cm}^2 = 26\,\mathrm{cm}^2.$$

   Now the variance matrix of co-ordinate *differences* is

   $$Q = Q_1 + Q_2 = m_0^2 I.$$

   From this still $m_0 = \sqrt{26}\mathrm{cm} = 5.1\mathrm{cm} = 0.051\mathrm{m}$.

2. Compute the deformation's testing variate

$$\underline{\mathcal{E}} = \underline{d}^T Q^{-1} \underline{d} = \frac{\underline{d}^T \underline{d}}{m_0^2}.$$

Here $\underline{\mathrm{d}} = \underline{\mathrm{x}}_2 - \underline{\mathrm{x}}_1$ is the shift vector, i.e., the vector of co-ordinate differences between the epochs. Because we assume that both co-ordinate sets are given in the same, common datum, the starting points of which nevertheless *don't* belong to the set $1-4$, we may assume that all co-ordinates are free. In that case the number of degrees of freedom is $h = 2n = 8$, where n is the number of points. The variance matrix of the co-ordinates of the shift vector $\underline{\mathrm{d}}$ is $m_0^2 I$.

Answer:

$\underline{\mathcal{E}} = \frac{1}{0.0026\,\mathrm{m}^2}\left(\underline{\mathrm{d}}^T \underline{\mathrm{d}}\right) = \frac{0.017771\,\mathrm{m}^2}{0.0026\,\mathrm{m}^2} = 6.835.$

3. The quantity $\underline{\mathcal{E}}$ is distributed according to the χ_8^2 distribution. If the limit value of this distribution for a significance level of 95% is 15.51 (cf. [Coo87] page 355), has in this case a deformation probably taken place?

Answer: No, it has not. $6.835 < 15.51$.

4. If, however, the assumed precisions were $m_{0,1} = m_{0,2} = \pm 1\mathrm{cm}$, would then, at a significance level of 95%, probably a deformation have taken place?

Answer: Yes, it would. $m_0^2 = (1+1)\ \mathrm{cm}^2 = 0.0002\,\mathrm{m}^2$ ja $\mathcal{E} = \frac{1}{0.0002\,\mathrm{m}^2}\left(\underline{\mathrm{d}}^T \underline{\mathrm{d}}\right) = \frac{0.017771\,\mathrm{m}^2}{0.0002\,\mathrm{m}^2} = 88.9 > 15.51$.

6.4 Strain tensor and affine deformation

We start from the known formula for the affine transformation:

$$\begin{aligned} x^{(2)} &= \Delta x + a_1 x^{(1)} + a_2 y^{(1)}, \\ y^{(2)} &= \Delta y + b_1 x^{(1)} + b_2 y^{(1)}. \end{aligned}$$

Now we apply this, instead of to the relationship between two different *datums*, to the relationship between a given point field seen at two different *epochs* $t_1$and t_2:

$$\begin{aligned} x(t_2) &= \Delta x + a_1 x(t_1) + a_2 y(t_1), \\ y(t_2) &= \Delta y + b_1 x(t_1) + b_2 y(t_1), \end{aligned}$$

or in matrix form:

$$\begin{bmatrix} x \\ y \end{bmatrix}(t_2) = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} + \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}(t_1).$$

Now we define the scaling and rotation parameters:

$$\begin{aligned} m &= \frac{1}{2}(a_1 + b_2), \\ \theta &= \frac{1}{2}(b_1 - a_2), \end{aligned}$$

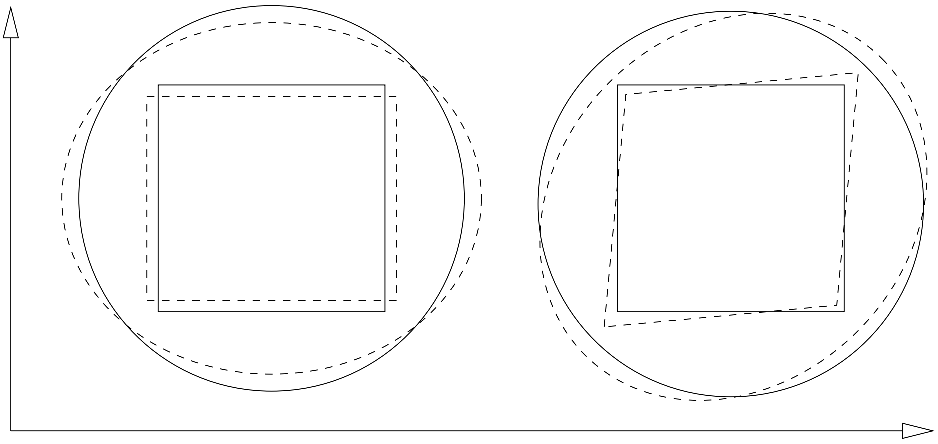


Figure 6.1: The strain tensor's two main components

allowing us to write

$$\begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} = \begin{bmatrix} m & 0 \\ 0 & m \end{bmatrix} + \begin{bmatrix} 0 & -\theta \\ \theta & 0 \end{bmatrix} + \begin{bmatrix} \frac{1}{2}(a_1 - b_2) & \frac{1}{2}(b_1 + a_2) \\ \frac{1}{2}(b_1 + a_2) & -\frac{1}{2}(a_1 - b_2) \end{bmatrix}.$$

The rightmost matrix is symmetric and we may write it as

$$S = \begin{bmatrix} \frac{1}{2}(a_1 - b_2) & \frac{1}{2}(b_1 + a_2) \\ \frac{1}{2}(b_1 + a_2) & -\frac{1}{2}(a_1 - b_2) \end{bmatrix} = \begin{bmatrix} s_{xx} & s_{xy} \\ s_{xy} & s_{yy} \end{bmatrix}.$$

This matrix is called the *strain tensor*. It describes how the shape of a little square of Earth surface *deforms*, and is thus the only "intrinsic" descriptor of deformation (The parameter m describes the change in surface area; θ describes rotational motion which neither changes surface area nor shape.).

The tensor has two main components which are at an angle of 45° to each other: $\frac{1}{2}(s_{xx} - s_{yy}) = \frac{1}{2}(a_1 - b_2)$ ("") describes elongation in the x direction together with compression in the y direction – or the opposite if negative –, while $s_{xy} = \frac{1}{2}(b_1 + a_2)$ ("") describes extension in the 45° direction together with compression in the orthogonal $-45°$ direction.

This is easily generalized to three dimensions, where there are three principal strains and three shear strains. Also, the analysis can be done over a whole area, using a DELAUNAY triangulation of the set of measured points. Sometimes this has been done to study crustal deformations after a seismic event.

Chapter 7

Stochastic processes and time series

Kirjallisuutta:

[San81, s. 31-46]

[Pap65]

7.1 Definitions

A *stochastic process* is a stochastic variable, the domain of which is a function space.

A *stochastic variable* $\underline{x}$ *is a recipe for producing* realizations $x_1, x_2, x_3, \ldots x_i, \ldots$ Every realization value has a certain *probability* of happening. If we repeat the realizations or "throws" often enough, the long-term percentage of that value happening tends towards the probability value.

First example E.g., throwing a die: the domain is $\{1, 2, 3, 4, 5, 6\}$ and the probability of realization is $p_1 = p_2 = \cdots = p_6 = \frac{1}{6}$ eli n. 16%.

Second example angle measurement with a theodolite: the domain is $\mathbb{R}$, all real values[1] and the probability distribution is

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2},$$

where σ is the mean error (standard deviation) of the distribution, and μ its expectancy. Here is is assumed, that the distribution is *normal*, i.e., the Gaussian bell curve.

In this case we speak of probability *density* and not the probability of a certain realization value x. The probability of a realization falling within a certain interval $[x_1, x_2\}$ is computed as the integral

$$p = \int_{x_1}^{x_2} p(x)\, dx.$$

The stochastic process $\underline{x}(t)$ is now a stochastic variable, the realizations of which are *functions* $x_1(t), x_2(t), x_3(t), \ldots, x_i(t), \ldots$

The argument t is usually time, but can also be, e.g., place (φ, λ) on the Earth's surface.

A *time series* is a series obtained from a stochastic process by specializing the argument t to more or less regularly spaced, chosen values t_j, $j = 1, 2, \ldots$ In other words, a stochastic process that is regularly measured.

A stochastic process — or a time series – is called *stationary* if its statistical properties do not change, when the argument t is replaced by the argument $t + \Delta t$.

[1] More precisely: all rational values $\mathbb{Q}$.

7.2 Variances and covariances of stochastic variables

Let us remember the definitions of the variance and covariance of stochastic variables:

$$\begin{aligned}\operatorname{Var}(\underline{x}) &\equiv E\left\{(\underline{x}-E\{\underline{x}\})^2\right\}\\ \operatorname{Cov}(\underline{x},\underline{y}) &\equiv E\left\{(\underline{x}-E\{\underline{x}\})\left(\underline{y}-E\{\underline{y}\}\right)\right\}\end{aligned}$$

and *correlation*:

$$\operatorname{Corr}(\underline{x},\underline{y}) \equiv \frac{\operatorname{Cov}(\underline{x},\underline{y})}{\sqrt{\operatorname{Var}(\underline{x})}\sqrt{\operatorname{Var}(\underline{y})}}.$$

The correlation is always between -100% and 100%.

7.3 Auto- and cross-covariance and -correlation

Studying stochastic processes is based on studying the dependencies between them. The *auto-covariance function* of a stochastic process describes the internal statistical dependency of one process:

$$\begin{aligned}A_x(t_1,t_2) &= \operatorname{Cov}\{\underline{x}(t_1),\underline{x}(t_2)\} =\\ &= E\{(\underline{x}(t_1)-E\{\underline{x}(t_1))\})(\underline{x}(t_2)-E\{\underline{x}(t_2))\})\}.\end{aligned}$$

Similarly we can define the *cross covariance function* between two different processes:

$$\begin{aligned}C_{xy}(t_1,t_2) &= \operatorname{Cov}\left\{\underline{x}(t_1),\underline{y}(t_2)\right\} =\\ &= E\left\{(\underline{x}(t_1)-E\{\underline{x}(t_1))\})\left(\underline{y}(t_2)-E\left\{\underline{y}(t_2))\right\}\right)\right\}.\end{aligned}$$

In case the processes in question are stationary, we obtain

$$\begin{aligned}A_x(t_1,t_2) &= A_x(\Delta t),\\ C_{xy}(t_1,t_2) &= C_{xy}(\Delta t),\end{aligned}$$

where $\Delta t = t_2 - t_1$.

Autocorrelation is defined in the following way[2]:

$$\operatorname{Corr}_x(\Delta t) \equiv \frac{A_x(t_1,t_2)}{\sqrt{A_x(t_1,t_1)\,A_x(t_2,t_2)}} = \frac{A_x(\Delta t)}{A_x(0)}.$$

Here one sees, that if $\Delta t = 0$, the autocorrelation is 1, and otherwise it is always between -1 and $+1$.

Cross-correlation is defined in the following non-trivial way:

$$\operatorname{Corr}_{xy}(\Delta t) \equiv C_{xy}(\Delta t)\,/\sqrt{A_x(0)\,A_y(0)}$$

all the time assuming stationarity.

Remember that

$$A_x(0) = \operatorname{Var}(\underline{x}(t)) = E\left\{(\underline{x}(t)-E\{\underline{x}(t)\})^2\right\}.$$

[2] Note that in the book [Pap65] an entirely different definition od auto- and cross-correlation is used!

Because[3]

$$\left|E\left\{\left(\underline{x}\left(t_1\right)-E\left\{\underline{x}\left(t_1\right)\right\}\right)\left(\underline{y}\left(t_2\right)-E\left\{\underline{y}\left(t_2\right)\right\}\right)\right\}\right| \leq$$
$$\leq \sqrt{E\left\{\left(\underline{x}\left(t_1\right)-E\left\{\underline{x}\left(t_1\right)\right\}\right)^2\right\} E\left\{\left(\underline{y}\left(t_2\right)-E\left\{\underline{y}\left(t_2\right)\right\}\right)^2\right\}},$$

the cross correlation is always between -1 and $+1$, and is 1 if *both* $\Delta t = 0$ *and* $\underline{x} = \underline{y}$.

7.4 Estimating autocovariances

This is done in roughly the same way as variance estimation in general. If we have, of the stochastic process $\underline{x}\left(t\right)$, realizations $x_i\left(t\right), i = 1, \ldots, n$, then an unbiased estimator (stationarity assumed) is:

$$\widehat{A_x\left(\Delta t\right)} \equiv \frac{1}{n-1}\sum_{i=1}^{n}\lim_{T\to\infty}\frac{1}{T}\int_0^T\left[\left(x_i\left(t\right)-\overline{x}\left(t\right)\right)\left(x_i\left(t+\Delta t\right)-\overline{x}\left(t+\Delta t\right)\right)\right]dt,$$

where

$$\overline{x}\left(t\right) \equiv \frac{1}{n}\sum_{i=1}^{n} x_i\left(t\right).$$

Again assuming the process $\underline{x}\left(t\right)$ to be stationary[4], and that we have available n process realizations $x_i\left(t_j\right), i = 1, \ldots n;\ j = 1, \ldots, m$ or *time series*, we may construct also from those an estimator. Nevertheless, either

[3] Why? Eric Weisstein xgives the following proof (`http://mathworld.wolfram.com/StatisticalCorrelation.html`):

Let be given the stochastic quantities $\underline{x}$ ja $\underline{y}$ — which may be values of the stochastic process $\underline{x}\left(t_1\right)$ ja $\underline{x}\left(t_2\right)$, or values from two different processes $\underline{x}\left(t_1\right)$ ja $\underline{y}\left(t_2\right)$ — define normalized quantities:

$$\underline{\xi} = \frac{\underline{x}}{\sqrt{\operatorname{Var}\left(\underline{x}\right)}}, \eta = \frac{\underline{y}}{\sqrt{\operatorname{Var}\left(\underline{y}\right)}}.$$

Silloin

$$\operatorname{Cov}\left(\underline{\xi},\underline{\eta}\right) = \frac{\operatorname{Cov}\left(\underline{x},\underline{y}\right)}{\sqrt{\operatorname{Var}\left(\underline{x}\right)\operatorname{Var}\left(\underline{y}\right)}} = \operatorname{Corr}\left(\underline{x},\underline{y}\right).$$

The following variances must be positive:

$$\begin{aligned}0 \leq \operatorname{Var}\left(\underline{\xi}+\underline{\eta}\right) &= \operatorname{Var}\left(\underline{\xi}\right)+\operatorname{Var}\left(\underline{\eta}\right)+2\cdot\operatorname{Cov}\left(\underline{\xi},\underline{\eta}\right),\\ 0 \leq \operatorname{Var}\left(\underline{\xi}-\underline{\eta}\right) &= \operatorname{Var}\left(\underline{\xi}\right)+\operatorname{Var}\left(\underline{\eta}\right)-2\cdot\operatorname{Cov}\left(\underline{\xi},\underline{\eta}\right).\end{aligned}$$

Also

$$\operatorname{Var}\left(\underline{\xi}\right)\frac{\operatorname{Var}\left(\underline{x}\right)}{\sqrt{\left(\operatorname{Var}\left(\underline{x}\right)\right)^2}} = 1,\ \operatorname{Var}\left(\underline{\eta}\right) = 1.$$

It follows that

$$\begin{aligned}0 &\leq 2+2\cdot\operatorname{Cov}\left(\xi,\eta\right),\\ 0 &\leq 2-2\cdot\operatorname{Cov}\left(\xi,\eta\right),\end{aligned}$$

i.e.,

$$-1 \leq \operatorname{Cov}\left(\underline{\xi},\underline{\eta}\right) = \operatorname{Corr}\left(\underline{x},\underline{y}\right) \leq 1.$$

[4] . . . and so-called ergodic.

- it must be assumed that the arguments t_j are equi-spaced, i.e., $t_{j+1} - t_j = \delta t$, or
- all possible values for Δt must be classified into classes or "bins" $[0, \delta t)$, $[\delta t, 2\delta t)$, $[2\delta t, 3\delta t)$, ...

Let us choose the first alternative. Then we can compute

$$\widehat{A_x\left(k\delta t\right)} \equiv \frac{1}{n-1}\sum_{i=1}^{n}\frac{1}{m-k-1}\sum_{j=1}^{m-k}\left[\left(x_i\left(t_j\right)-\overline{x}\left(t_j\right)\right)\left(x_i\left(t_{j+k}\right)-\overline{x}\left(t_{j+k}\right)\right)\right].$$

So we use for computing every autocovariance value only those argument values of the process, that have a suitable distance between them. The problem is, that we can compute these autocovariances only for those discrete values $0, \delta t, 2\delta t, \ldots$.

The other mentioned approach has the same problem through the back door: the size of the classes is finite.

Remark. In case we have a stochastic process of place (φ, λ) and not of time, we use bins according to the spherical distance ψ. This is common practice in gravity field theory. The often made assumption corresponding to stationarity is *homogeneity* (the statistical properties of the process do not depend on place) and *isotropy* (don't depend either on the compass heading between two points P, Q, but only on the angular distance ψ_{PQ}.

7.5 Autocovariance and spectrum

Let us assume a *stationary* stochastic process $\underline{x}\left(t\right)$,which at the same time is *noise*, i.e., its expectancy vanishes:

$$E\left\{x\left(t\right)\right\} = 0.$$

The FOURIER transform of the autocovariance of this process is (we use here the notation t for the earlier Δt):

$$S\left(\omega\right) = \int_{-\infty}^{+\infty} A_x\left(t\right)e^{-i\omega t}dt. \tag{7.1}$$

This function is called PSD or Power Spectral Density. Another formula for computing it is

$$\underline{S}\left(\omega\right) = \lim_{T\to\infty}\frac{1}{2T}\left|\int_{-T}^{T}\underline{x}\left(t\right)e^{-i\omega t}dt\right|^2, \tag{7.2}$$

i.e., the long term "average power" on different frequencies ω. This formula can be used for empirically computing$S$, if from the process itself $\underline{x}\left(t\right)$ we have available a realization $x_i\left(t\right)$. The former formula is used if we have rather a closed expression for the auto covariance.

The inverse FOURIER equation for computing $A_x$from $S\left(\omega\right)$ is

$$A_x\left(t\right) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} S\left(\omega\right)e^{i\omega t}d\omega.$$

Proofs can be found in the book [Pap65].

7.6 AR(1), lineaariregressio ja varianssi

7.6.1 Pienimmän neliösumman regressio ilman autocorrelaatiota

Linear regression starts from the well known equation

$$y = a + bx$$

where we have given many point pairs $(x_i, y_i), i = 1, \ldots n$. This is more precisely an *observation equation*

$$\underline{y}_i = a + bx_i + \underline{n}_i,$$

where stochastic process $\underline{n}_i$ models the *stochastic uncertainty of the measurement process,* i.e., the *noise.*

We assume the noise to behave so, that the variance is a constant independent of i ("homoskedasticity"), and the covariance vanishes identically ("white noise")[5]:

$$\begin{aligned} \mathrm{Var}\left(\underline{n}_i\right) &= \sigma^2, \\ \mathrm{Cov}\left(\underline{n}_i, \underline{n}_j\right) &= 0,\ i \neq j. \end{aligned}$$

This is called the *statistical model.*

We may write the observation equations into the form

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & \vdots \\ 1 & x_{n-1} \\ 1 & x_n \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix},$$

where now $\mathrm{y} = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}^T$ is the vector of observations (in an n-dimensional abstract vector space), $\mathrm{x} = \begin{bmatrix} a & b \end{bmatrix}^T$ is the vector of unknowns (parameters), and

$$A = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_{n-1} \\ 1 & x_n \end{bmatrix}$$

is the (second order) *design matrix.* This way of presentation is referred to as the *functional model.*

Based on the assumed statistical model we may compute the least squares solution with the help of the *normal equations*:

$$\left(A^T A\right) \widehat{\mathrm{x}} = A^T \underline{\mathrm{y}}.$$

More concretely:

$$A^T A = \begin{bmatrix} n & \sum x \\ \sum x & \sum x^2 \end{bmatrix},$$

[5] This set of assumptions is often called i.i.d., "independent, identically distributed"

or (Cramer's rule):

$$\left(A^TA\right)^{-1} = \frac{1}{n\sum x^2 - \left(\sum x\right)^2}\left[\begin{array}{cc} \sum x^2 & -\sum x \\ -\sum x & n \end{array}\right],$$

from which

$$\begin{aligned} \hat{a} &= \frac{\sum x^2 \sum \underline{y} - \sum x \sum x\underline{y}}{n\sum x^2 - \left(\sum x\right)^2}, \\ \hat{b} &= \frac{-\sum x \sum \underline{y} + n\sum x\underline{y}}{n\sum x^2 - \left(\sum x\right)^2} \end{aligned}$$

are the least squares *estimators* of the unknowns. Their precision (uncertainty, mean error) is given (formal error propagation) by the diagonal elements of the inverted normal matrix $\left(A^TA\right)^{-1}$, scaled by σ:

$$\sigma_a = \sigma\sqrt{\frac{\sum x^2}{n\sum x^2 - \left(\sum x\right)^2}},\ \sigma_b = \sigma\sqrt{\frac{n}{n\sum x^2 - \left(\sum x\right)^2}}.$$

Most often we are particularly interested in the trend b, meaning that we should compare the value $\hat{b}$ obtained with its own mean error σ_b. If σ is not known *a priori*, it should be evaluated from the *residuals*: the square sum of residuals

$$\sum \underline{v}^2 = \sum\left(\underline{y} - \hat{a} - \hat{b}x\right)^2$$

has the expected value of $(n-2)\,\sigma^2$, where $n-2$ is the number of *degrees of freedom* (overdetermination), 2 being the number of unknowns estimated. Or

$$\widehat{\sigma^2} = \frac{\sum v^2}{n-2}.$$

7.6.2 AR(1) process

The assumption made above, that the observational errors $\underline{n}_i$ are uncorrelated between themselves, is often *wrong*. Nevertheless least squares regression is such a simple method – available, e.g., in popular spreadsheets and pocket calculators – that it is often used even though the zero correlation requirement is not fulfilled.

If the autocorrelation of the noise process $\underline{n}_i$ does not vanish, we can often model it as a so-called AR(1) (auto-regressive first-order) or Gauß-Markov process. Such a process is described as a *Markov chain*:

$$\underline{n}_{i+1} = \rho\underline{n}_i + \widetilde{n}_i, \tag{7.3}$$

where ρ is a suitable parameter, $0 < \rho < 1$, and $\tilde{n}$ is a truly non-correlating "white noise" process:

$$\begin{aligned} \mathrm{Var}\left(\widetilde{\underline{n}}_i\right) &= \widetilde{\sigma}^2, \\ \mathrm{Cov}\left(\widetilde{\underline{n}}_i, \widetilde{\underline{n}}_j\right) &= 0,\ i \neq j. \end{aligned}$$

Write now the observation equation two times, multiplied the second time around by $-\rho$:

$$\begin{aligned} y_{i+1} &= a + bx_{i+1} + n_{i+1}, \\ -\rho y_i &= -\rho a - \rho b x_i - \rho n_i\,, \end{aligned}$$

…and sum together:

$$y_{i+1} - \rho y_i = (a - \rho a) + b\,(x_{i+1} - \rho x_i) + (n_{i+1} - \rho n_i)\,.$$

This equation is of the form

$$Y_i = A + bX_i + \tilde{n}_i,$$

where

$$\begin{aligned} \widetilde{n}_i & \quad \text{as described above,} \\ A &= (1-\rho)\,a, \\ X_i &= x_{i+1} - \rho x_i, \\ Y_i &= y_{i+1} - \rho y_i. \end{aligned}$$

i.e., the formula for the non-correlated linear regression.

The recipe now is:

1. Compute X_i and Y_i according to above formulae;
2. Solve $\widehat{A}$ and $\widehat{b}$ according to non-correlated linear regression;
3. Compute $\hat{a} = (1-\rho)^{-1}\hat{A}$;
4. The ratio between $\widetilde{\sigma}^2$ and σ^2: from equation 7.3 it follows, that based on stationarity

   $$\sigma^2 = \rho^2\sigma^2 + \widetilde{\sigma}^2,$$

   in other words,

   $$\left(1-\rho^2\right)\sigma^2 = \widetilde{\sigma}^2.$$

   So, one either computes an empirical $\widetilde{\sigma}^2$ and transforms it into a σ^2 of the original observations, or the given σ^2 of the original observations is transformed to $\widetilde{\sigma}^2$ in order to evaluate the precisions of the estimators of A and b.
5. From point 4 we may also conclude that

   $$\sigma^2_{b,\mathrm{AR}(1)} = \frac{\sigma^2_{b,\mathrm{nocorr}}}{1-\rho^2},$$

   where $\sigma^2_{b,\mathrm{nocorr}}$ is the "naively" calculated variance of the trend parameter.

Conclusion: if there is autocorrelation in the data, a simple linear regression will give a *much too optimistic picture* of the trend parameter b's mean error, and thus also of its significance (difference from zero).

If the data is given as an equi-spaced function of time, i.e., $x_i = x_0 + (i-1)\,\Delta t$, we may connect the parameter ρ of the AR(1) process in a simple way to its *correlation length*: the solution of equation 7.3 (without noise) is

$$n_j = \rho^{j-i} n_i = e^{(j-i)\ln\rho} = \exp\left(-\frac{(j-i)\,\Delta t}{\tau}\right),$$

where τ is the correlation length in units of time.

For consideration of non-strictly-AR(1) processes, see [?].

Dimensional analysis

One can say that the *dimension* of process $\underline{x}(t)$ is, e.g., [length]. In equation 7.1 we then have for the dimension of $A_x(t)$ [length]^2, and for the dimension of $S(\omega)$[length]^2 [time]. Also according to equation 7.2 the dimension is ([length] × [time])^2 / [time].

Chapter 8

Variants of adjustment theory

8.1 Adjustment in two phases

Often we run into the situation, where a local network must be connected to "given" fixed points. This is a special case of the previous section, where $\underline{\ell}_1$ are the observations in the local network and $\underline{\ell}_2$ are the co-ordinates of the points given by the higher order network. Let x_2 be the co-ordinate unknowns of which there also exist "given" co-ordinates.

The observation equations are now

$$\begin{aligned}\underline{\ell}_1 + \underline{\mathrm{v}}_1 &= A_1\widehat{\mathrm{x}}_1 + A_2\widehat{\mathrm{x}}_2 \\ \underline{\ell}_2 + \underline{\mathrm{v}}_2 &= I\widehat{\mathrm{x}}_2\end{aligned}$$

– i.e., the A matrix is

$$A = \begin{bmatrix} A_1 & A_2 \\ 0 & I \end{bmatrix}.$$

The variance matrix is

$$Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix},$$

where the observation, residuals and unknowns vectors are

$$\underline{\ell} = \begin{bmatrix} \underline{\ell}_1 \\ \underline{\ell}_2 \end{bmatrix}, \underline{\mathrm{v}} = \begin{bmatrix} \underline{\mathrm{v}}_1 \\ \underline{\mathrm{v}}_2 \end{bmatrix}, \underline{\mathrm{x}} = \begin{bmatrix} \widehat{\mathrm{x}}_1 \\ \widehat{\mathrm{x}}_2 \end{bmatrix}.$$

Here Q_2 is the variance matrix of the given points. We get as the solution

$$\widehat{\mathrm{x}} = \left(A^T Q^{-1} A\right)^{-1} A^T Q^{-1} \underline{\ell},$$

where

$$A^T Q^{-1} A = \begin{bmatrix} A_1^T Q_1^{-1} A_1 & A_1^T Q_1^{-1} A_2 \\ A_2^T Q_1^{-1} A_1 & A_2^T Q_1^{-1} A_2 + Q_2^{-1} \end{bmatrix}$$

and

$$A^T Q^{-1} \underline{\ell} = \begin{bmatrix} A_1^T Q_1^{-1} \underline{\ell}_1 \\ A_2^T Q_1^{-1} \underline{\ell}_1 + Q_2^{-1} \underline{\ell}_2 \end{bmatrix},$$

and thus

$$\begin{bmatrix} A_1^T Q_1^{-1} A_1 & A_1^T Q_1^{-1} A_2 \\ A_2^T Q_1^{-1} A_1 & A_2^T Q_1^{-1} A_2 + Q_2^{-1} \end{bmatrix} \begin{bmatrix} \widehat{\mathrm{x}}_1 \\ \widehat{\mathrm{x}}_2 \end{bmatrix} = \begin{bmatrix} A_1^T Q_1^{-1} \underline{\ell}_1 \\ A_2^T Q_1^{-1} \underline{\ell}_1 + Q_2^{-1} \underline{\ell}_2 \end{bmatrix}.$$

From this we see that generally the adjusted $\widehat{\ell_2} = \widehat{\mathrm{x}}_2$ differs from the original value $\underline{\ell}_2$!

Generally this is not acceptable.

> *The co-ordinates of a higher-order network may not change as the result of the adjustment of a lower-order network!*

How to solve this quandary?

One proposed solution is the so-called pseudo-least squares method (Baarda).

Put in front of the matrix Q_2 a coefficient α, so

$$\begin{bmatrix} A_1^TQ_1^{-1}A_1 & A_1^TQ_1^{-1}A_2 \\ A_2^TQ_1^{-1}A_1 & A_2^TQ_1^{-1}A_2+\alpha^{-1}Q_2^{-1} \end{bmatrix}\begin{bmatrix} \widehat{\mathrm{x}}_1 \\ \widehat{\mathrm{x}}_2 \end{bmatrix}=\begin{bmatrix} A_1^TQ_1^{-1}\underline{\ell}_1 \\ A_2^TQ_1^{-1}\underline{\ell}_1+\alpha^{-1}Q_2^{-1}\underline{\ell}_2 \end{bmatrix}.$$

Now, let $\alpha \to 0$, i.e, assume the given points to be infinitely precise.

Multiply the last row with α:

$$\begin{bmatrix} A_1^TQ_1^{-1}A_1 & A_1^TQ_1^{-1}A_2 \\ \alpha A_2^TQ_1^{-1}A_1 & \alpha A_2^TQ_1^{-1}A_2+Q_2^{-1} \end{bmatrix}\begin{bmatrix} \widehat{\mathrm{x}}_1 \\ \widehat{\mathrm{x}}_2 \end{bmatrix}=\begin{bmatrix} A_1^TQ_1^{-1}\underline{\ell}_1 \\ \alpha A_2^TQ_1^{-1}\underline{\ell}_1+Q_2^{-1}\underline{\ell}_2 \end{bmatrix}.$$

Now let $\alpha \to 0$:

$$\begin{bmatrix} A_1^TQ_1^{-1}A_1 & A_1^TQ_1^{-1}A_2 \\ 0 & Q_2^{-1} \end{bmatrix}\begin{bmatrix} \widehat{\mathrm{x}}_1 \\ \widehat{\mathrm{x}}_2 \end{bmatrix}=\begin{bmatrix} A_1^TQ_1^{-1}\underline{\ell}_1 \\ Q_2^{-1}\underline{\ell}_2 \end{bmatrix}$$

or (multiply the last row with Q_2):

$$\begin{bmatrix} A_1^TQ_1^{-1}A_1 & A_1^TQ_1^{-1}A_2 \\ 0 & I \end{bmatrix}\begin{bmatrix} \widehat{\mathrm{x}}_1 \\ \widehat{\mathrm{x}}_2 \end{bmatrix}=\begin{bmatrix} A_1^TQ_1^{-1}\underline{\ell}_1 \\ \underline{\ell}_2 \end{bmatrix}.$$

As can be seen do the now given co-ordinates no longer change: $\widehat{\mathrm{x}}_2=\underline{\ell}_2$.

The solution $\widehat{\mathrm{x}}_1$ is obtained from the following normal equations:

$$\left(A_1^TQ_1^{-1}A_1\right)\widehat{\mathrm{x}}_1=A_1^TQ_1^{-1}\underline{\ell}_1-A_1^TQ_1^{-1}A_2\underline{\ell}_2=A_1^TQ_1^{-1}\left(\underline{\ell}_1-A_2\underline{\ell}_2\right).$$

If we look closer at this equation, we see that it tells that $\widehat{\mathrm{x}}_1$ is a linear combination of the observations $\begin{bmatrix} \underline{\ell}_1 & \underline{\ell}_2 \end{bmatrix}^T$: thus we may write

$$\widehat{\mathrm{x}}_1=L_1\underline{\ell}_1+L_2\underline{\ell}_2,$$

where

$$\begin{aligned} L_1 &= \left(A_1^TQ_1^{-1}A_1\right)^{-1}A_1^TQ_1^{-1}, \\ L_2 &= \left(A_1^TQ_1^{-1}A_1\right)^{-1}A_1^TQ_1^{-1}\left(-A_2\right). \end{aligned}$$

The *true* variance of $\widehat{\mathrm{x}}_1$ will now be, according to the propagation law:

$$Q_{\widehat{\mathrm{x}}_1\widehat{\mathrm{x}}_1}=L_1Q_1L_1^T+L_2Q_2L_2^T=\left(A_1^TQ_1^{-1}A_1\right)^{-1}\left(A_1^TQ_1^{-1}A_1+A_1^TA_2Q_2^{-1}A_2^TA_1\right)\left(A_1^TQ_1^{-1}A_1\right)^{-1},$$

after some simplifications.

Note that this is *bigger* than the “naively” computed variance

$$Q^*_{\widehat{x}_1\widehat{x}_1}=\left(A_1^TQ_1^{-1}A_1\right)^{-1},$$

which does not consider that the given co-ordinates, even though these were assumed “errorless” for the sake of computing the co-ordinates, nevertheless contain error which will propagate into the local solution $\widehat{\mathrm{x}}_1$.

So:

- We compute the estimates of the unknowns using the “wrong” variance matrix;
- and then we compute the variances of those estimators “right”, using the propagation law of variances.

8.2 Using a priori knowledge in adjustment

Sometimes we estimate unknowns from observations, even though we already know “something” about those unknowns. E.g., we estimate the co-ordinates of points from geodetic observations, but those co-ordinates are already approximately known, e.g., read from a map (we have found the points to be measured, haven’t we!) The measurement was planned using approximate co-ordinates, so apparently they already exist.

Let there be for the unknowns *a priori* values and variance matrix

$$\underline{\mathrm{x}}, Q_{xx}$$

and let the observation equations be

$$\underline{\ell} + \underline{\mathrm{v}} = A\widehat{\mathrm{x}}.$$

Let the variance matrix of the vector of observations be

$$Q_{\ell\ell}$$

and let the values of observations and *a priori* unknowns be statistically independent from each other. Then we may *extend* the observation equations:

$$\begin{bmatrix} \underline{\ell} \\ \underline{\mathrm{x}} \end{bmatrix} + \begin{bmatrix} \underline{\mathrm{v}} \\ \underline{\mathrm{v}}_x \end{bmatrix} = \begin{bmatrix} A \\ I \end{bmatrix} \widehat{\mathrm{x}}.$$

If we define formally

$$\widetilde{A} \equiv \begin{bmatrix} A \\ I \end{bmatrix}, \widetilde{Q} \equiv \begin{bmatrix} Q_{\ell\ell} & 0 \\ 0 & Q_{xx} \end{bmatrix}, \underline{\widetilde{\ell}} \equiv \begin{bmatrix} \underline{\ell} \\ \underline{\mathrm{x}} \end{bmatrix}, \widetilde{\underline{\mathrm{v}}} = \begin{bmatrix} \underline{\mathrm{v}} \\ \underline{\mathrm{v}}_x \end{bmatrix}$$

the solution is

$$\begin{aligned} \widehat{\mathrm{x}} &= \left[\widetilde{A}^T \widetilde{Q}^{-1} \widetilde{A}\right]^{-1} \widetilde{A}^T \widetilde{Q}^{-1} \underline{\widetilde{\ell}} = \\ &= \left[A^T Q_{\ell\ell}^{-1} A + Q_{xx}^{-1}\right]^{-1} \left[A^T Q_{\ell\ell}^{-1} \underline{\ell} + Q_{xx}^{-1} \underline{\mathrm{x}}\right]. \end{aligned}$$

If we are talking about *linearized* observation equations, and in the linearization the same approximate values have been used as are being used now as *a priori* values (i.e., $\mathrm{x}_0 = \underline{\mathrm{x}}$, in other words, $\underline{\Delta \mathrm{x}} = \underline{x} - \mathrm{x}_0 = \underline{0}$), we obtain

$$\widehat{\Delta \mathrm{x}} = \left[A^T Q_{\ell\ell}^{-1} A + Q_{xx}^{-1}\right]^{-1} \left[A^T Q_{\ell\ell}^{-1} \underline{\Delta \ell} + Q_{xx}^{-1} \underline{\Delta \mathrm{x}}\right] = \left[A^T Q_{\ell\ell}^{-1} A + Q_{xx}^{-1}\right]^{-1} A^T Q_{\ell\ell}^{-1} \underline{\Delta \ell}.$$

The variance matrix of the unknowns after adjustment (*a posteriori*) is, based on the propagation law

$$\begin{aligned} Q_{\widehat{\mathrm{x}}\widehat{\mathrm{x}}} &= \left[A^T Q_{\ell\ell}^{-1} A + Q_{xx}^{-1}\right]^{-1} \left[A^T Q_{\ell\ell}^{-1} \left(Q_{\Delta\ell\Delta\ell}\right) Q_{\ell\ell}^{-1} A + Q_{xx}^{-1} \left(Q_{\Delta x \Delta x}\right) Q_{xx}^{-1}\right] \left[A^T Q_{\ell\ell}^{-1} A + Q_{xx}^{-1}\right]^{-1} = \\ &= \left[A^T Q_{\ell\ell}^{-1} A + Q_{xx}^{-1}\right]^{-1}, \end{aligned}$$

because $\mathrm{Var}\left(\underline{\Delta \ell}\right) = Q_{\Delta\ell\Delta\ell} = Q_{\ell\ell}$ and $\mathrm{Var}\left(\underline{\Delta x}\right) = Q_{\Delta x \Delta x} = Q_{xx}$.

Sometimes this method is used in order to *stabilize* unstable observation or normal equations. The name for this is Tikhonov-regularization or ridge regression. Inverting the above matrix may be impossible, e.g., if the matrix A has a rank defect. Then, adding the matrix Q_{xx}^{-1} makes its inversion possible.

Often it is assumed that

$$Q_{xx}^{-1} = \alpha I,$$

where α is called a *regularization parameter.* When this is done just in order to stabilize the equations, it should be remembered, that it means *adding information to the solution.* If this information does not really exist, the result of the adjustment will be too optimistic about the precision. In this case *statistical testing* is a good way of finding out, as the residuals will be overly large.

8.3 Stacking of normal equations

Let us assume, that we have available the mutually independent observations $\underline{\ell}_1$ ja $\underline{\ell}_2$ which depend on the same unknowns x. The observation equations are

$$\begin{aligned} \underline{\ell}_1 + \underline{\mathrm{v}}_1 &= A_1 \widehat{\mathrm{x}} \\ \underline{\ell}_2 + \underline{\mathrm{v}}_2 &= A_2 \widehat{\mathrm{x}} \end{aligned}$$

and the variance matrices of the observations are Q_1 and Q_2. Then the joint variance matrix is

$$Q = \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix}$$

and the joint system of equations

$$\underline{\ell} + \underline{v} = A\widehat{\mathrm{x}}\,,$$

where $\ell = \begin{bmatrix} \underline{\ell}_1 \\ \underline{\ell}_2 \end{bmatrix}$, $\underline{\mathrm{v}} = \begin{bmatrix} \underline{v}_1 \\ \underline{v}_2 \end{bmatrix}$ and $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$. The system of normal equations reads

$$\left(A^T Q^{-1} A\right) \widehat{\mathrm{x}} = A^T Q^{-1} \underline{\ell}$$

i.e.,

$$\left(A_1^T Q_1^{-1} A_1 + A_2^T Q_2^{-1} A_2\right) \widehat{\mathrm{x}} = \left(A_1^T Q_1^{-1} \underline{\ell}_1 + A_2^T Q_2^{-1} \underline{\ell}_2\right)\ , \tag{8.1}$$

from which we see, that the total solution is obtained by summing up both the normal matrices $A^T QA$ *and the normal vectors* $A^T Q^{-1} \underline{\ell}$. The procedure is called *Normals stacking.*

In GPS computations this principle is exploited; the results of the adjustment of GPS networks are often distributed in a compact "normals" form. Those can then be easily combined. SINEX = Software Independent EXchange format.

If, of the elements of the vector of unknowns x , only a small part depends on both observation vectors ℓ_1 and ℓ_2, we can exploit this to solve the system of equations 8.1 efficiently in phases (Helmert-Wolf blocking). More generally there exist so-called *sparse matrix algorithms* that exploit the special situation where the elements of the matrices A and Q vanish for the most part.

8.4 Helmert-Wolf blocking

8.4.1 Principle

Often, a large adjustment problem can be naturally divided into small parts. E.g., the adjustment of a large international triangulation network can be executed in this way, that firstly we adjust every country separately; then the border point co-ordinates obtained are fed into a continental adjustment; and the corrections at the border points obtained from that are again propagated back into every country's internal points.

This procedure was used, e.g., in the European ED50 triangulation adjustment, as well as in the NAD (North American Datum) adjustment in North America.

The theory is the following: let there be two classes of unknowns,

1. "global" unknowns $\widehat{\mathrm{x}}_e$ on which observations $\underline{\ell}_1, \underline{\ell}_2, \ldots, \underline{\ell}_n$ in *all* countries $1, 2, \ldots, n$ depend, i.e., the "top level", "European" unknowns;
2. "local" unknowns $\widehat{\mathrm{x}}_i$, $i = 1, 2, \ldots n$ on which only depend observations $\underline{\ell}_i$ in a single country i.

Then the observation equations are[1]

$$\begin{bmatrix} \underline{\ell}_1 \\ \underline{\ell}_2 \\ \vdots \\ \underline{\ell}_n \end{bmatrix} + \begin{bmatrix} \underline{\mathrm{v}}_1 \\ \underline{\mathrm{v}}_2 \\ \vdots \\ \underline{\mathrm{v}}_n \end{bmatrix} = \left[\begin{array}{cccc|c} A_1 & & & & B_1 \\ & A_2 & & & B_2 \\ & & \ddots & & \vdots \\ & & & A_n & B_n \end{array}\right] \left[\begin{array}{c} \widehat{\mathrm{x}}_1 \\ \widehat{\mathrm{x}}_2 \\ \vdots \\ \widehat{\mathrm{x}}_n \\ \hline \widehat{\mathrm{x}}_e \end{array}\right].$$

From this we obtain the normal equation system:

$$\left[\begin{array}{cccc|c} A_1^T Q_1^{-1} A_1 & & & & A_1^T Q_1^{-1} B_1 \\ & A_2^T Q_2^{-1} A_2 & & & A_2^T Q_2^{-1} B_2 \\ & & \ddots & & \vdots \\ & & & A_n^T Q_n^{-1} A_n & A_n^T Q_n^{-1} B_n \\ \hline B_1^T Q_1^{-1} A_1 & B_2^T Q_2^{-1} A_2 & \cdots & B_n^T Q_n^{-1} A_n & \sum_{i=1}^n B_i^T Q_i^{-1} B_i \end{array}\right] \left[\begin{array}{c} \widehat{\mathrm{x}}_1 \\ \widehat{\mathrm{x}}_2 \\ \vdots \\ \widehat{\mathrm{x}}_n \\ \hline \widehat{\mathrm{x}}_e \end{array}\right] = \left[\begin{array}{c} A_1^T Q_1^{-1} \underline{\ell}_1 \\ A_2^T Q_2^{-1} \underline{\ell}_2 \\ \vdots \\ A_n^T Q_n^{-1} \underline{\ell}_n \\ \hline \sum_{i=1}^n B_i^T Q_i^{-1} \underline{\ell}_i \end{array}\right]$$

where $Q_i \equiv Q_{\ell_i \ell_i}$ is the variance matrix of the observations $\underline{\ell}_i$ — we assume that the observations of different "countries" ℓ_i, ℓ_j don't correlate.

Note that the above normal matrix is "arrow shaped" ($\searrow$), i.e., a "bordered main diagonal matrix". Such matrices occur often in adjustment theory and make possible a simplified treatment.

In more symbolic form:

$$\begin{bmatrix} N_{II} & N_{IE} \\ N_{EI} & N_{EE} \end{bmatrix} \begin{bmatrix} \mathrm{x}_I \\ \mathrm{x}_E \end{bmatrix} = \begin{bmatrix} \mathrm{b}_I \\ \mathrm{b}_E \end{bmatrix},$$

where the definitions of the various sub-matrices and -vectors are clear.

This matric equation represents two equations:

$$\begin{aligned} N_{II}\mathrm{x}_I + N_{IE}\mathrm{x}_E &= \mathrm{b}_I, \\ N_{EI}\mathrm{x}_I + N_{EE}\mathrm{x}_E &= \mathrm{b}_E. \end{aligned}$$

Multiply the first equation with the matrix $N_{EI} N_{II}^{-1}$ and *subtract* it from the second, in order to eliminate the "local" unknowns x_i, $i = 1, \ldots, n$:

$$\left(N_{EE} - N_{EI} N_{II}^{-1} N_{IE}\right) \mathrm{x}_E = \mathrm{b}_E - N_{EI} N_{II}^{-1} \mathrm{b}_I, \tag{8.2}$$

[1] Note the "normals stacking" is a special case of this: $\mathrm{A}_i = 0$, $i = 1, \ldots, n$ and the x_i do not exist.

the so-called reduced system of normal equations, from which we obtain $\widehat{\mathrm{x}}_E$. Written open:

$$\left(\sum_{i=1}^n B_i^T\left(I_i - Q_i^{-1}A_i\left(A_i^TQ_i^{-1}A_i\right)^{-1}A_i^T\right)Q_i^{-1}B_i\right)\mathrm{x}_E = \\ = \left(\sum_{i=1}^n B_i^T\left(I_i - Q_i^{-1}A_i\left(A_i^TQ_i^{-1}A_i\right)^{-1}A_i^T\right)Q_i^{-1}\ell_i\right).$$

After that, by substituting into the first equation:

$$N_{II}\mathrm{x}_I = \mathrm{b}_I - N_{IE}\widehat{\mathrm{x}}_E,$$

i.e., the local solutions. Written out:

$$\begin{bmatrix} A_1^{\mathrm{T}}Q_1^{-1}A_1 & & & \\ & A_2^{\mathrm{T}}Q_2^{-1}A_2 & & \\ & & \ddots & \\ & & & A_n^TQ_n^{-1}A_n \end{bmatrix}\begin{bmatrix}\mathrm{x}_1\\ \mathrm{x}_2\\ \vdots\\ \mathrm{x}_n\end{bmatrix} = \begin{bmatrix} A_1^TQ_1^{-1}\left(\ell_1 - B_1\widehat{\mathrm{x}}_E\right)\\ A_2^TQ_2^{-1}\left(\ell_2 - B_2\widehat{\mathrm{x}}_E\right)\\ \vdots\\ A_n^TQ_n^{-1}\left(\ell_n - B_n\widehat{\mathrm{x}}_E\right)\end{bmatrix},$$

so, the individual solution of a country is obtained nicely separated:

$$\left(A_i^TQ_i^{-1}A_i\right)\mathrm{x}_i = A_i^TQ_i^{-1}\left(\ell_i - B_i\widehat{\mathrm{x}}_E\right), \tag{8.3}$$

from which we obtain the estimators $\widehat{\mathrm{x}}_i$, $i = 1, \ldots, n$.

The major advantage of Helmert-Wolf is, that the new normal equations (8.2, 8.3) have only the size of the vectors x_E or x_i, $i = 1, \ldots n$, and not of the whole system, $\left[\begin{array}{cccc|c} \mathrm{x}_1 & \mathrm{x}_2 & \cdots & \mathrm{x}_n & \mathrm{x}_E\end{array}\right]^T$. This matters because the computational work required for solving the whole csystem of equations (like also the work of inverting a matrix) is proportional to the *third power* of its size!

8.4.2 Variances

The variance matrix of the whole vector of unknowns $\begin{bmatrix}\mathrm{x}_I & \mathrm{x}_E\end{bmatrix}^{\mathrm{T}}$ is the inverse N^{-1} of the normal matrix

$$N \equiv \begin{bmatrix} N_{II} & N_{IE}\\ N_{EI} & N_{EE}\end{bmatrix}$$

This inverse matrix is

$$\begin{aligned} N^{-1} &= \begin{bmatrix} N_{II}^{-1} + N_{II}^{-1}N_{IE}Q_EN_{EI}N_{II}^{-1} & -N_{II}^{-1}N_{IE}Q_E\\ -Q_EN_{EI}N_{II}^{-1} & Q_E\end{bmatrix},\\ Q_E &\equiv \left(N_{EE} - N_{EI}N_{II}^{-1}N_{IE}\right)^{-1},\end{aligned}$$

which we can easily verify by multiplication $N \cdot N^{-1} = I$.

It follows that the variances of the least-squares estimators are:

$$\mathrm{Var}\left(\widehat{\mathrm{x}}_E\right) = Q_E;$$

$$\mathrm{Var}\left(\widehat{\mathrm{x}}_I\right) = N_{II}^{-1} + N_{II}^{-1}N_{IE}Q_EN_{EI}N_{II}^{-1}.$$

Because N_{II} (and thus also N_{II}^{-1}) is a block diagonal matrix

$$N_{II} = \begin{bmatrix} A_1^TQ_1^{-1}A_1 & & & \\ & A_2^TQ_2^{-1}A_2 & & \\ & & \ddots & \\ & & & A_n^TQ_n^{-1}A_n\end{bmatrix},$$

we may write separately for each local block $i = 1, \ldots, n$:

$$\begin{aligned}\operatorname{Var}(\widehat{\mathrm{x}}_i) &= \left(A_i^T Q_i^{-1} A_i\right)^{-1} + \left(A_i^T Q_i^{-1} A_i\right)^{-1} N_{iE} Q_E N_{Ei} \left(A_i^T Q_i^{-1} A_i\right)^{-1} = \\ &= \left(A_i^T Q_i^{-1} A_i\right)^{-1} + \left(A_i^T Q_i^{-1} A_i\right)^{-1} A_i^T Q_i^{-1} B_i Q_E B_i^T Q_i^{-1} A_i \left(A_i^T Q_i^{-1} A_i\right)^{-1}.\end{aligned}$$

With the global adjustment, *correlation* is introduced between the internal unknowns of different country blocks $i \neq j$ However, computing all covariances $\operatorname{Cov}(\widehat{\mathrm{x}}_i, \widehat{\mathrm{x}}_j)$, even if possible, is rarely sensible or useful. The savings of Helmert-Wolf are mostly achieved by not computing them.

8.4.3 Practical application

In international co-operation it has been the habit of sending *only* the "buffer matrices" of country i,

$$\left(B_i^T \left(I_i - Q_i^{-1} A_i \left(A_i^T Q_i^{-1} A_i\right)^{-1} A_i^{\mathrm{T}}\right) Q_i^{-1} B_i\right)$$

and

$$\left(B_i^T \left(I_i - Q_i^{-1} A_i \left(A_i^T Q_i^{-1} A_i\right)^{-1} A_i^T\right) Q_i^{-1} \ell_i\right),$$

to the international computing centre, which in turn sends the vector $\widehat{\mathrm{x}}_E$ and variance matrix Q_E computed by it, back to the individual countries.

Among the major advantages of theHelmert-Wolf method are still, that

1. the amounts of data to be processed at one time remain small
2. Also local observational material can be tested separately before its use in a "jumbo adjustment", and remove from them possible gross errors
3. The greater part of the computation, consisting of national/provincial/regional/per-different-epoch partial adjustments, can be executed independently from each other, e.g., *parallelly.* Helmert-Wolf is the perfect example of parallel processing!

8.5 Intersection in the plane

8.5.1 Precision in the plane

Intersection from two known points is a good example for error propagation, It is also a good example for *optimization,* because optimization of the result can be done in at least three different ways, which all three in different ways make sense.

We start from the error ellipse (figure 8.1), which describes the precision of determination of the co-ordinates (x, y) in the plane. If the variance matrix of the vector $\widehat{\mathbf{x}} \equiv \begin{bmatrix} x & y \end{bmatrix}^T$ is

$$\Sigma_{\widehat{x}\widehat{x}} = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix},$$

the formula for the error ellipse is

$$\widehat{\mathbf{x}}^T \Sigma_{\widehat{x}\widehat{x}}^{-1} \widehat{\mathbf{x}} = 1.$$

The matrix $\Sigma_{\widehat{x}\widehat{x}}$ has two *eigenvalues*, λ_1 and λ_2, the square roots of which are the major and minor semi-axes of the ellipse, see figure.

The size of the error ellipse can now be determined in the following different ways:

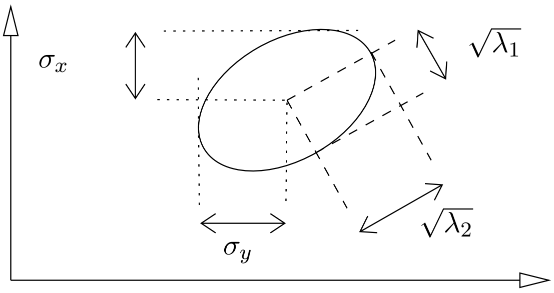


Figure 8.1: The parameters of the error ellipse.

1. The *radius* of the circle inside which the ellipse fits. This is the minimization of the expression $Max\,(\lambda_1, \lambda_2)$, i.e., the largest eigenvalue of the matrix must be made as small as possible. This so-called minimax optimization corresponds in practice to so-called *tolerance*, i.e., the error may not exceed a pre-set value.
2. The mean point error, $\sigma_P \equiv \sqrt{\sigma_x^2 + \sigma_y^2} = \sqrt{\lambda_1 + \lambda_2}$. The *trace* of the variance matrix $\Sigma_{\widehat{x}\widehat{x}}$, i.e., the sum of the main diagonal elements. This corresponds to minimizing the diagonal of the rectangle within which the error ellipse fits ("television screen size in inches").
3. Minimize the matrix *determinant* $\det\left(\Sigma_{\widehat{x}\widehat{x}}\right)$. This corresponds to minimizing the quantity $\sqrt{\lambda_1\lambda_2}$,i.e., the *surface area* of the ellipse.

8.5.2 The geometry of intersection

Let, for simplicity, the co-ordinates of the points A, B be $(-1, 0)$ and $(+1, 0)$. From these points have been measured *directions* to the point P with a constant precision. If the co-ordinates of point P are (x, y), the observation equations will be:

$$\begin{aligned} \alpha &= \arctan\left(\frac{y}{x+1}\right), \\ \beta &= \arctan\left(\frac{y}{x-1}\right). \end{aligned}$$

We linearize:

$$\begin{aligned} d\alpha &= \frac{1}{1+\left(\frac{y}{x+1}\right)^2}\cdot\left(\frac{1}{x+1}dy - \frac{y}{(x+1)^2}dx\right), \\ d\beta &= \frac{1}{1+\left(\frac{y}{x-1}\right)^2}\cdot\left(\frac{1}{x-1}dy - \frac{y}{(x-1)^2}dx\right). \end{aligned}$$

After simplifying:

$$\begin{aligned} d\alpha &= \frac{1}{(x+1)^2 + y^2}\left((x+1)\,dy - ydx\right), \\ d\beta &= \frac{1}{(x-1)^2 + y^2}\left((x-1)\,dy - ydx\right). \end{aligned}$$

The A matrix is

$$A = \begin{bmatrix} \frac{x+1}{(x+1)^2+y^2} & \frac{-y}{(x+1)^2+y^2} \\ \frac{x-1}{(x-1)^2+y^2} & \frac{-y}{(x-1)^2+y^2} \end{bmatrix} = \begin{bmatrix} \frac{1}{a}\cos\alpha & -\frac{1}{a}\sin\alpha \\ \frac{1}{b}\cos\beta & -\frac{1}{b}\sin\beta \end{bmatrix},$$

where $a = \sqrt{(x+1)^2 + y^2}$ ja $b = \sqrt{(x-1)^2 + y^2}$.

The normal matrix:

$$\begin{aligned} N &= A^{\mathrm{T}} A = \\ &= \begin{bmatrix} \left(\frac{\cos\alpha}{a}\right)^2 + \left(\frac{\cos\beta}{b}\right)^2 & -\frac{\sin\alpha\cos\alpha}{a^2} - \frac{\sin\beta\cos\beta}{b^2} \\ -\frac{\sin\alpha\cos\alpha}{a^2} - \frac{\sin\beta\cos\beta}{b^2} & \left(\frac{\sin\alpha}{a}\right)^2 + \left(\frac{\sin\beta}{b}\right)^2 \end{bmatrix}. \end{aligned}$$

The determinant:

$$\begin{aligned} \det N &= \left(\frac{\cos^2\alpha\sin^2\alpha}{a^4} + \frac{\cos^2\alpha\sin^2\beta}{a^2b^2} + \frac{\cos^2\beta\sin^2\alpha}{a^2b^2} + \frac{\cos^2\beta\sin^2\beta}{b^4}\right) - \\ &- \left(\frac{\cos^2\alpha\sin^2\alpha}{a^4} + \frac{\cos^2\beta\sin^2\beta}{b^4} + 2\frac{\sin\alpha\sin\beta\cos\alpha\cos\beta}{a^2b^2}\right) = \\ &= \frac{\cos^2\beta\sin^2\alpha + \cos^2\alpha\sin^2\beta - 2\sin\alpha\sin\beta\cos\alpha\cos\beta}{a^2b^2} = \\ &= \left(\frac{\sin\alpha\cos\beta - \cos\alpha\sin\beta}{ab}\right)^2 = \left(\frac{\sin(\alpha-\beta)}{ab}\right)^2. \end{aligned}$$

Compute the inverse matrix using Cramèr's rule:

$$\Sigma_{\widehat{x}\widehat{x}} = N^{-1} = \frac{1}{\det N}\begin{bmatrix} \frac{\sin^2\alpha}{a^2} + \frac{\sin^2\beta}{b^2} & \frac{\sin\alpha\cos\alpha}{a^2} + \frac{\sin\beta\cos\beta}{b^2} \\ \frac{\sin\alpha\cos\alpha}{a^2} + \frac{\sin\beta\cos\beta}{b^2} & \frac{\cos^2\alpha}{a^2} + \frac{\cos^2\beta}{b^2} \end{bmatrix}.$$

The *trace* of this matrix is

$$\begin{aligned} \left(\Sigma_{\widehat{x}\widehat{x}}\right)_{11} + \left(\Sigma_{\widehat{x}\widehat{x}}\right)_{22} &= \frac{a^2b^2}{\sin^2(\alpha-\beta)}\left(\frac{\sin^2\alpha}{a^2} + \frac{\sin^2\beta}{b^2} + \frac{\cos^2\alpha}{a^2} + \frac{\cos^2\beta}{b^2}\right) = \\ &= \frac{a^2b^2}{\sin^2(\alpha-\beta)}\left(\frac{1}{a^2} + \frac{1}{b^2}\right) = \frac{(a^2+b^2)}{\sin^2(\alpha-\beta)}. \end{aligned}$$

Here still $a^2 + b^2 = (x-1)^2 + (x+1)^2 + 2y^2 = 2(x^2 + y^2 + 1)$.

Geometrically:

1. The curves $\sin(\alpha-\beta)$ =constant are *circles* going through A and B.
2. The curves $a^2 + b^2$ =constant, i.e., $x^2 + y^2$ =constant, are circles also, but around the *origin* $(0,0)$.
3. Because of this, the value $a^2+b^2 = 2(x^2+y^2+1)$ is minimized[2] on the curve $\sin(\alpha-\beta)$ =constant, when $x = 0$.

Conclusion:

The optimal point is located on the y axis or symmetry axis, i.e., $x = 0$.

[2] Assumption: the angle $\gamma > 90°$. We shall see that this is the case.

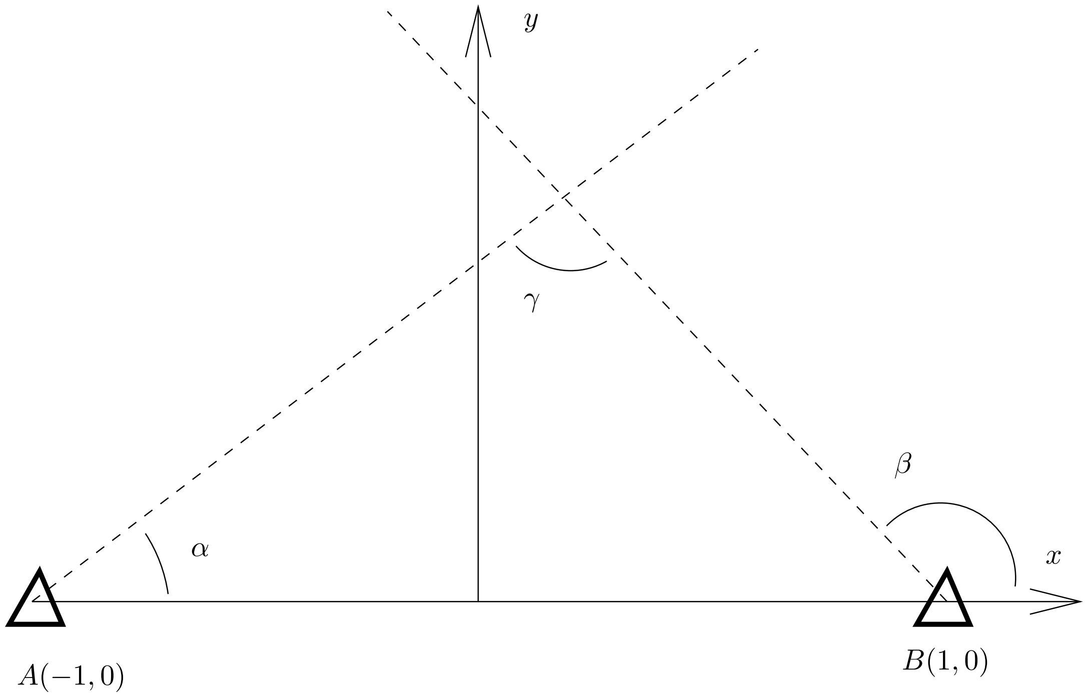


Figure 8.2: Intersection in the plane

8.5.3 Minimizing the point mean error

Let us still compute where. For a point on the y axis it holds that

$$\alpha = \arctan y = 180^\circ - \beta,$$

i.e.,

$$\sin(\alpha - \beta) = \sin(2\alpha - 180^\circ) = -\sin 2\alpha.$$

Now

$$\begin{aligned}(\Sigma_{\widehat{x}\widehat{x}})_{11} + (\Sigma_{\widehat{x}\widehat{x}})_{22} &= \tfrac{\left(a^2+b^2\right)}{\sin^2(\alpha-\beta)} = \frac{2\left(y^2+1\right)}{\sin^2 2\alpha} = \frac{2\left(y^2+1\right)}{\left(2\sin\alpha\cos\alpha\right)^2} = \\ &= \frac{2\left(y^2+1\right)}{\left(2\frac{y}{\sqrt{y^2+1}}\cdot\frac{1}{\sqrt{y^2+1}}\right)^2} = \frac{\left(y^2+1\right)^3}{2y^2}.\end{aligned}$$

Require now that this derivative vanish, i.e., a *stationary point*:

$$\begin{aligned}\frac{d}{dy}\frac{\left(y^2+1\right)^3}{2y^2} &= \frac{1}{2}\frac{d}{dy}\left(y^4+3y^2+3+y^{-2}\right) = \\ &= 2y^3+3y-1y^{-3} = \frac{2y^6+3y^4-1}{y^3} = 0.\end{aligned}$$

This can be done as follows:

$$2y^6+3y^4-1=0$$

Matlab yields (*verify* by substitution!):

$$\begin{aligned}y_{1,2} &= \pm\frac{1}{2}\sqrt{2}\\ y_{3,4} &= \pm i\\ y_{5,6} &= \pm i\end{aligned}$$

Of these, only the real values are of interest:

$$y = \frac{1}{2}\sqrt{2} \Rightarrow \alpha = \mathrm{arctan} y = 35^\circ.2644.$$

Then $\beta = 180^\circ - 35^\circ.2644 = 144^\circ.7356$ and $\gamma = 109^\circ.4712$.

This result, which required so much work, has further interest: it is precisely the angle between chemical bonds between atoms in, e.g., a diamond crystal or a methane molecule...

8.5.4 Minimizing the determinant

Alternatively we may minimize $\det\left(\Sigma_{\widehat{x}\widehat{x}}\right)$ (or its square root), i.e., the geometrical mean of $\Sigma_{\widehat{x}\widehat{x}}$'s eigenvalues, $\sqrt{\lambda_1\lambda_2}$ (when minimizing the trace amounts to minimizing there arithmetical mean, $\frac{1}{2}\left(\lambda_1+\lambda_2\right)$). Then we compute first

$$a^2b^2 = \left[(x+1)^2+y^2\right]\left[(x-1)^2+y^2\right].$$

When $x = 0$, this is

$$a^2b^2 = \left(y^2+1\right)^2.$$

Then also, just like before

$$\sin^2(\alpha-\beta) = \sin^2 2\alpha = \left(2\frac{y}{\sqrt{y^2+1}}\cdot\frac{1}{\sqrt{y^2+1}}\right)^2$$

and we obtain as the final result

$$\det\left(\Sigma_{\widehat{x}\widehat{x}}\right) = \left(\frac{\sin(\alpha-\beta)}{ab}\right)^{-2} = \frac{\left(y^2+1\right)^4}{4y^2},$$

the stationary points of which we seek.

MATLAB[3] yields the solution

$$y_{1,2} = \pm\frac{1}{3}\sqrt{3},$$

the other solutions are imaginary ±i. From this $\alpha = \arctan y = 30^\circ$.

8.5.5 "Minimax" optimization

As the third alternative we minimize the biggest eigenvalue, i.e., we minimize $\max\left(\lambda_1,\lambda_2\right)$. On the axis $x = 0$ we have $a = b$ and $\sin\alpha\cos\alpha = -\sin\beta\cos\beta$, i.e., the form of the matrix N is:

$$N = \frac{2}{a^2}\begin{bmatrix}\sin^2\alpha & 0\\ 0 & \cos^2\alpha\end{bmatrix} = \frac{2}{y^2+1}\begin{bmatrix}\frac{y^2}{y^2+1} & 0\\ 0 & \frac{1}{y^2+1}\end{bmatrix}.$$

Because $\alpha = \arctan y$, it follows that $\sin\alpha = \frac{y}{\sqrt{y^2+1}}$ and $\cos\alpha = \frac{1}{\sqrt{y^2+1}}$, and $a^2 = y^2+1$.

The eigenvalues of this matrix are thus

$$\mu_1 = \frac{2y^2}{\left(y^2+1\right)^2}, \quad \mu_2 = \frac{2}{\left(y^2+1\right)^2}$$

[3] Use *symbolic computation.* First define the function $f(y)$, then its derivative (`diff` function), and finally, using the `solve` function, its zero points.

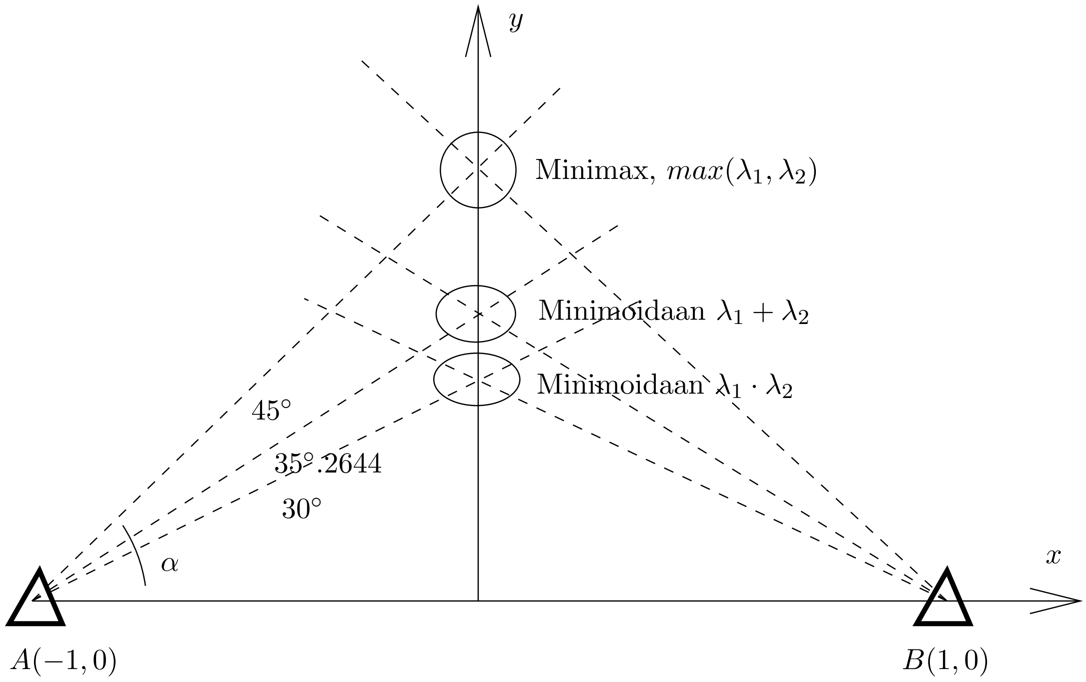

y
Minimax, $max(\lambda_1, \lambda_2)$
Minimoidaan $\lambda_1 + \lambda_2$
Minimoidaan $\lambda_1 \cdot \lambda_2$
45°
35°.2644
30°
α
x
$A(-1,0)$
$B(1,0)$

Figure 8.3: Three different optimal solutions for intersection

and the eigenvalues of the inverse matrix

$$\lambda_1 = \frac{1}{\mu_1} = \frac{1}{2y^2}\left(y^2+1\right)^2, \quad \lambda_2 = \frac{1}{\mu_2} = \frac{1}{2}\left(y^2+1\right)^2.$$

When $y = 1$, these are the same; when $y > 1$, λ_2 is the bigger one and grows with y.

When y lies in the interval $(0,1)$, then λ_1 is the bigger one, $\lambda_1 = \frac{1}{2}\left(y^2+2+y^{-2}\right) \Rightarrow \frac{d}{dy}\lambda_1 = y - y^{-3} < 0$, i.e., λ_1 descends monotonously.

End result:

the optimal value is $y = 1$ and $\alpha = \arctan 1 = 45°$.

8.6 Exercises

1. Derive the corresponding equations as in section 8.5 for the case where we make *distance measurements* from points A and B, the precision of which does *not* depend on distance.
2. Show, that if the eigenvalues of matrix N are close to each other,

$$\begin{aligned}\lambda_1 &= \lambda_0 + \Delta\lambda_1,\\ \lambda_2 &= \lambda_0 + \Delta\lambda_2,\\ &\cdots\\ \lambda_n &= \lambda_0 + \Delta\lambda_n,\end{aligned}$$

where the $\Delta\lambda_i$ are small compared to λ_0, that then

$$(\det N)^{\frac{1}{n}} = \left(\prod_{i=1}^{n}\lambda_i\right)^{\frac{1}{n}} = \frac{1}{n}\sum_{i=1}^{n}\lambda_i = \frac{1}{n}\mathrm{Tr}\,(N).$$

[Hint: use the binomial expansion $(1+x)^y \approx 1+yx+\ldots$]

So, in this case *minimizing the determinant is equivalent to minimizing the trace.*

3. [*Challenging.*] Show that if, in three dimensional space, we measure the *distance* of point P,

$$s=\sqrt{\left(x_P-x_A\right)^2+\left(y_P-y_A\right)^2+\left(z_P-z_A\right)^2}$$

from three known points A, B and C, the optimal geometry is that in which the three directions PA, PB ja PC are *mutually orthogonal*, i.e., $PA \perp PB \perp PC$. The assumption is that the measurements from all three points are equally precise and independent of distance.

[Hint: write the 3×3 design matrix which consists of the three unit vectors, and maximize its determinant, or (geometrically intuitively) the volume spanned by these vectors. $\det\left(N^{-1}\right)=\left(\det A\right)^{-2}$, so this minimizes the determinant of $Q_{\widehat{x}\widehat{x}}$]

4. [*Challenging.*] Show, that if we measure, in the plane, the *pseudo-range* to a vessel A (DECCA system!)

$$\rho=\sqrt{\left(x_A-x_M\right)^2+\left(y_A-y_M\right)^2}+c\Delta T_A,$$

from three points M, R, G (Master, Red Slave, Green Slave), the optimal geometry is that, where the angles between the directions AM, AR, AG are 120°.

In the equation, ΔT_A is the clock unknown of vessel A.

[Hint: write the 3×3 design matrix; remember that also ΔT is an unknown. After that, as in the previous case.]

Chapter 9

Kalman filter

Literature:

[Kal98b, s. 62-66, 154-155]

[SB97, s. 543-584]

[Lei95, s. 115-130]

[Coo87, s. 215-223]

[MA76, s. 333-392]

The Kalman filter is a linear predictive filter. Like a coffee filter which filters coffee from drags, the Kalman filter filters signal (the so-called *state vector*) from the noise of the measurement process.

The inventors of theKalman filter were Rudolf KALMAN and Richard BUCY in the years 1960-1961 ([Kal60]; [KB61]). The invention was widely used in the space programme and in connection with missile guidance systems. However, the Kalman filter is generally applicable and has been used except in navigation, also in economic science, meteorology etc.

A Kalman filter consists of two parts:

1. The dynamic model; it describes the motion process according to which the *state vector* evolves over time.
2. The observation model; it describes the process by which observables are obtained, that tell us something about the state vector at the moment of observation.

Special for the Kalman filter is, that the state vector propagates in time one step at a time; also the observations are used for correcting the state vector only at the moment of observation. For this reason the Kalman filter does not require much processing power and doesn't handle large matrices. It can be used inside a vehicle in real time.

9.1 Dynamic model

In the linear case, the synamic model looks like this:

$$\frac{d}{dt}\underline{\mathrm{x}} = F\underline{\mathrm{x}} + \underline{\mathrm{n}}, \tag{9.1}$$

where $\underline{\mathrm{x}}$ is the state vector, $\underline{\mathrm{n}}$ is the dynamic noise (i.e., how imprecisely the above equations of motion are valid) and F is a coefficient matrix.

The *variance matrix* of the state vector's $\underline{\mathrm{x}}$ estimator $\widehat{\mathrm{x}}$ which is available at a certain point in time may be called Q or $Q_{\widehat{x}\widehat{x}}$. It describes the probable deviation of the true state $\underline{\mathrm{x}}$ from the estimated state $\widehat{\mathrm{x}}$. The noise vector $\underline{\mathrm{n}}$ in the above equation describes, how imprecisely the equations of motion, i.e., the dynamic model, *in reality* are valid, e.g., in the case of satellite motion, the varying influence of atmospheric drag. A large dynamical noise $\underline{\mathrm{n}}$ means that $Q_{\widehat{x}\widehat{x}}$ will inflate quickly with time. This can then again be reduced with the help of *observations* to be made and the *state updates* to be performed using these.

9.2 State propagation in time

The computational propagation in time of the state vector itself is simple:

$$\frac{d}{dt}\widehat{\mathrm{x}} = F\widehat{\mathrm{x}}.$$

In the corresponding discrete case:

$$\widehat{\mathrm{x}}\left(t_1\right) = \Phi_0^1 \widehat{\mathrm{x}}\left(t_0\right),$$

where

$$\Phi_0^1 = e^{F(t_1 - t_0)},$$

a discrete version of the the coefficient matrix integrated over time $[t_0, t_1)$.

If we call the variance matrix of $\underline{\mathrm{n}}$ (more precisely: the autocovariance function of $\underline{\mathrm{n}}\,(\mathrm{t})$) N, we may also write the discrete proagation formula for the variance matrix:

$$Q(t_1) = \left(\Phi_0^1\right) Q(t_0) \left(\Phi_0^1\right)^T + \int_{t_0}^{t_1} N\left(t\right) dt.$$

Here we have assumed, that $\underline{\mathrm{n}}\,(t)$ is by its nature *white noise.* The proof of this equation is difficult.

9.3 Observational model

The development of the state vector in time would not be very interesting, if we could not also somehow *observe* this vector. The observational model is as follows:

$$\underline{\ell} = H\underline{\mathrm{x}} + \underline{\mathrm{m}},$$

where $\underline{\ell}$ is an observable (vector), $\underline{\mathrm{x}}$ is a state vector ("true value"), and $\underline{\mathrm{m}}$ is the observation process's noise. H is the *observation matrix.* As the variance of the noise we have given the variance matrix R; $E\left\{\underline{\mathrm{m}}\right\} = 0$ and $E\left\{\underline{\mathrm{m}}\ \underline{\mathrm{m}}^{\mathrm{T}}\right\} = R$.

9.4 The update step

Updating is the optimal use of new observation data. It is done in such a way, that the difference between the observable's value $\widehat{\ell}_i = H\widehat{\mathrm{x}}_i$ computed from the *a priori* state vector $\widehat{\mathrm{x}}_i$, and the truly observed observable $\underline{\ell}_i$, is used as a "closing error", which we try to "adjust" away in an optimal fashion, according to the principle of least squares.

Let us construct an improved estimator

$$\widehat{\mathrm{x}}_{i+1} = \widehat{\mathrm{x}}_i + K\left(\underline{\ell}_i - H_i\widehat{\mathrm{x}}_i\right).$$

Note that $\widehat{\underline{x}}_{\mathrm{i+1}}$ is the estimator of the state vector after observation i, i.e., *a posteriori*. However, relative to the index $i+1$ is is again *a priori*.

Here the matrix K is called the Kalman "gain matrix".

The "optimal" solution is obtained by choosing

$$K = QH^T \left(HQH^T + R\right)^{-1},$$

which gives as solution

$$\widehat{\underline{x}}_{i+1} = \widehat{\underline{x}}_i + Q_i H_i^T \left(H_i Q_i H_i^T + R\right)^{-1} \left(\underline{\ell}_i - H_i \widehat{\underline{x}}_i\right).$$

Updating the state variances as follows:

$$Q_{i+1} = Q_i - Q_i H_i^T \left(H_i Q_i H_i^T + R_i\right)^{-1} H_i Q_i = \left(I - K_i H_i\right) Q_i,$$

without proof.

9.5 Sequential adjustment

Sequential adjustment is the Kalman filter applied to the case where the state vector to be estimated (i.e., the vector of unknowns) does not depend on time. In this case the formulas become simpler, but using the Kalman formulation may nevertheless be advantageous, because it allows the addition of new information to the solution immediately when it becomes available. Also in network adjustment one sometimes processes different groups of observations sequentially, which facilitates finding possible errors. The co-ordinates are in this case state vector elements independent of time.

The dynamical model is in this case

$$\frac{d}{dt}\underline{x} = 0,$$

i.e., $F = 0$ and $\underline{n} = 0$. There is no dynamical noise.

Often one encounters also applications in which a part of the state vector's elements are constants, and another part time dependent. E.g., in satellite geodesy, earth station co-ordinates as opposed to satellite orbital elements.

9.5.1 Sequential adjustment and stacking of normal equations

We may write the update step of the Kalman filter also as a parametric adjustment problem.

The "observations" are the real observation vector $\underline{\ell}_i$ and the *a priori* estimated state vector $\widehat{\underline{x}}_i$. Observation equations:

$$\begin{bmatrix} \underline{\ell}_i \\ \widehat{\underline{x}}_i \end{bmatrix} + \begin{bmatrix} \underline{v}_i \\ \underline{w}_i \end{bmatrix} = \begin{bmatrix} H_i \\ I \end{bmatrix} \left[\widehat{\underline{x}}_{i+1}\right].$$

Here, the design matrix is

$$\widetilde{A} \equiv \begin{bmatrix} H_i \\ I \end{bmatrix}.$$

The variance matrix of the "observations" is

$$\widetilde{Q} \equiv \mathrm{Var}\left(\left[\begin{array}{cc} \underline{\ell}_i & \widehat{\mathrm{x}}_i \end{array}\right]^T\right) = \left[\begin{array}{cc} R_i & 0 \\ 0 & Q_i \end{array}\right],$$

and we obtain as the solution

$$\begin{aligned} \widehat{\mathrm{x}}_{i+1} &= \left[\widetilde{A}^T \widetilde{Q}^{-1} \widetilde{A}\right]^{-1} \widetilde{A}^T \widetilde{Q}^{-1} \left[\begin{array}{c} \underline{\ell}_i \\ \widehat{\mathrm{x}}_i \end{array}\right] = \\ &= \left[H_i^T R_i^{-1} H_i + Q_i^{-1}\right]^{-1} \left[H_i^T R_i^{-1} \underline{\ell}_i + Q_i^{-1} \widehat{\mathrm{x}}_i\right]. \qquad (9.2) \end{aligned}$$

As the variance we obtain

$$Q_{i+1} = \left[H_i^T R_i^{-1} H_i + Q_i^{-1}\right]^{-1}. \qquad (9.3)$$

Ks. [Kal98b, ss. 63-64].

Now we exploit the second formula derived in appendix A:

$$(A + UCV)^{-1} = A^{-1} - A^{-1} U \left(C^{-1} + V A^{-1} U\right)^{-1} V A^{-1}.$$

In this way:

$$\left[H_i^T R_i^{-1} H_i + Q_i^{-1}\right]^{-1} = Q_i - Q_i H^T \left(R_i + H_i Q_i H_i^T\right)^{-1} H_i Q_i.$$

Substitution yields

$$\begin{aligned} \widehat{\mathrm{x}}_{i+1} &= \left[Q_i - Q_i H^T \left(R_i + H_i Q_i H_i^T\right)^{-1} H_i Q_i\right] \left[H_i^T R_i^{-1} \underline{\ell}_i + Q_i^{-1} \widehat{\mathrm{x}}_i\right] = \\ &= \left[I - Q_i H^T \left(R_i + H_i Q_i H_i^T\right)^{-1} H_i\right] \left[Q_i H_i^T R_i^{-1} \underline{\ell}_i + \widehat{\mathrm{x}}_i\right] = \\ &= \left[Q_i H_i^T R_i^{-1} \underline{\ell}_i + \widehat{\mathrm{x}}_i\right] - Q_i H^T \left(R_i + H_i Q_i H_i^T\right)^{-1} \left[H_i Q_i H_i^T R_i^{-1} \underline{\ell}_i + H_i \widehat{\mathrm{x}}_i\right] \\ &= \widehat{\mathrm{x}}_i + Q_i H_i^T R_i^{-1} \underline{\ell}_i - Q_i H^T \left(R_i + H_i Q_i H_i^T\right)^{-1} \left(H Q_i H_i^T + R_i\right) R_i^{-1} \underline{\ell}_i + \\ &\quad + Q_i H^T \left(R_i + H_i Q_i H_i^T\right)^{-1} R_i R_i^{-1} \underline{\ell}_i - Q_i H^T \left(R_i + H_i Q_i H_i^T\right)^{-1} H_i \widehat{\mathrm{x}}_i = \\ &= \widehat{\mathrm{x}}_i + Q_i H^T \left(R_i + H_i Q_i H_i^T\right)^{-1} \left[\underline{\ell}_i - H_i \widehat{\mathrm{x}}_i\right], \qquad (9.4) \end{aligned}$$

and

$$Q_{i+1} = Q_i - Q_i H^T \left(R_i + H_i Q_i H_i^T\right)^{-1} H_i Q_i. \qquad (9.5)$$

The equations 9.4 and 9.5 are precisely the update equations of the Kalman filter. Compared to the equations 9.2 and 9.3, the matrix to be inverted has the size of the vector of observables $\underline{\ell}$ and not that of the state vector $\underline{\mathrm{x}}$. Often the matrix size is even 1×1 , i.e, a simple number[1]. Being able to compute inverse matrices more quickly makes real-time applications easier.

From the preceding we see, that sequential adjustment is the same as Kalman filtering in the case that the state vector is constant. Although the computation procedure in adjustment generally is parametric adjustment (observation equations), when in the Kalman case, condition equations adjustment is used.

[1] ...or may be reduced to such, if the observations made at one epoch are statistically independent of each other. Then they may be formally processed sequentially, i.e., separately.

9.6 Kalman “from both ends”

If we have available the observations $\underline{\ell}_i$, $i = 1, \ldots, n$ and the functional model is the system of differential equations

$$\frac{d}{dt}\mathrm{x} = F\mathrm{x}$$

(without dynamic noise $\underline{\mathrm{n}}$), we may write

$$\mathrm{x}\left(\mathrm{t}_i\right) = \Phi_0^i \mathrm{x}\left(\mathrm{t}_0\right),$$

where Φ_0^i is the state transition matrix to be computed. Thus, the observation equations may be written

$$\underline{\ell}_i + \underline{\mathrm{v}}_i = H_i \mathrm{x}\left(\mathrm{t}_i\right) = H_i \Phi_0^i \mathrm{x}\left(\mathrm{t}_0\right),$$

a traditional system of observation equations, where the desgin matrix is

$$A = \begin{bmatrix} H_0 \\ \vdots \\ H_i \Phi_0^i \\ \vdots \\ H_n \Phi_0^n \end{bmatrix}$$

and the unknowns $\mathrm{x}\left(\mathrm{t}_0\right)$.

From this we see, that the least-squares solution can be obtained by solving an adjustment problem.

As we saw in section 8.3,we may divide the observations into several, e.g., two, parts:

$$\underline{\ell} = \begin{bmatrix} \underline{\ell}_e \\ \underline{\ell}_j \end{bmatrix}, \, A = \begin{bmatrix} A_e \\ A_j \end{bmatrix}$$

and form separate normal equations:

$$\begin{aligned} \left[A_e^T Q_e^{-1} A_e\right] \widehat{\mathrm{x}}_e &= A_e^T Q_e^{-1} \underline{\ell}_e, \\ Q_{xx,e} &= \left[A_e^T Q_e^{-1} A_e\right]^{-1}, \end{aligned}$$

and

$$\begin{aligned} \left[A_j^T Q_j^{-1} A_j\right] \widehat{\mathrm{x}}_j &= A_j^T Q_j^{-1} \underline{\ell}_j, \\ Q_{xx,j} &= \left[A_j^T Q_j^{-1} A_j\right]^{-1}. \end{aligned}$$

These separate solutions (e = before, j = after) can now be “stacked”, i.e., combined:

$$\left[A_e^T Q_e^{-1} A_e + A_j^T Q_j^{-1} A_j\right] \widehat{\mathrm{x}} = \left[A_e^T Q_e^{-1} \underline{\ell}_e + A_j^T Q_j^{-1} \underline{\ell}_j\right],$$

the original full equations system, and

$$Q_{xx} = \left[Q_{xx,e}^{-1} + Q_{xx,j}^{-1}\right]^{-1} = \left[A_e^T Q_e^{-1} A_e + A_j^T Q_j^{-1} A_j\right]^{-1},$$

the variance matrix of the solution from the full adjustment.

An important remark is, that the partial tasks — “before” and ”after” — can be solved also with the help of the Kalman filter! In other words, we may, for an arbitrary observation epoch t_i, compute separately

1. The solution of the Kalman filter from the starting epoch t_0 forward, by integrating the dynamical model and updating the state vector and its variance matrix for the observations $0, \ldots, i$, and
2. The Kalman filter solution from the final moment t_n *backward in time* integrating the dynamic modal, updating the state vector and the variance matrix using the observations $n, \downarrow, i+1$ (in reverse order).
3. *Combining* the partial solutions obtained into a total solution using the above formulas.

In this way, the advantages of the KALMANmethod may be exploited also in a post-processing situation.

9.7 Exercises

Let x be an unknown constant which we are trying to estimate. x has been observed at epoch 1, observation value 7, mean error ±2, and at epoch 2, observation value 5, mean error ±1.

1. Formulate the observation equations or an ordinary adjustment problem and the variance matrix of the observation vector. Compute $\widehat{x}$.

$$\underline{\ell} + \underline{\mathrm{v}} = A\widehat{\mathrm{x}}$$

where $\ell = \begin{bmatrix} 7 \\ 5 \end{bmatrix}$, $\mathrm{Q}_{\ell\ell} = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$, $A = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Then

$$\begin{aligned} \widehat{\mathrm{x}} &= \left[A^T Q_{\ell\ell}^{-1} A\right]^{-1} A^T Q_{\ell\ell}^{-1} \underline{\ell} = \\ &= \frac{4}{5} \cdot \begin{bmatrix} \frac{1}{4} & 1 \end{bmatrix} \begin{bmatrix} 7 \\ 5 \end{bmatrix} = \frac{27}{5} = 5.4. \end{aligned}$$

The variance matrix:

$$Q_{\widehat{x}\widehat{x}} = \left[A^T Q_{\ell\ell}^{-1} A\right]^{-1} = \frac{4}{5} = 0.8.$$

2. Write the dynamical equations for the Kalman filter of this example. Remember that x *is a constant.*

Answer

: The general dynamical equation may be written in the discrete case

$$\underline{x}_{i+1} = \Phi \underline{x}_i + \underline{w}$$

where $\Phi = I$ (unit matrix) and $\underline{w} = 0$ (deterministic motion, dynamic noise absent). Thus we obtain

$$x_{i+1} = x_i.$$

Alternatively we write the differential equation:

$$\frac{d\underline{x}}{dt} = F\underline{x} + \underline{n}$$

Again in our example case:

$$\frac{dx}{dt} = 0,$$

no dynamic noise: $\underline{n} = 0$.

3. Write the update equations for the Kalman filter of this example:

$$\widehat{x}_i = \widehat{x}_{i-1} + K_i \left(\underline{\ell}_i - H_i \widehat{x}_{i-1} \right)$$

and

$$Q_{\widehat{x}\widehat{x},i} = \left[I - K_i H_i \right] Q_{\widehat{x}\widehat{x},i-1},$$

where the gain matrix

$$K_i = Q_{\widehat{x}\widehat{x},i-1} H_i^T \left(Q_{\ell\ell,i} + H_i^T Q_{\widehat{x}\widehat{x},i-1} H_i \right)^{-1}.$$

(so, how do in this case look the H- and K matrices?)

Answer: Because in his case the observation $\ell_i = x_i$ (i.e., we observe directly the state) we have $H_i = [1]$, i.e., a 1×1 matrix, the only element of which is 1.

$$K = \frac{Q_{\widehat{x}\widehat{x},i-1}}{Q_{\ell\ell} + Q_{\widehat{x}\widehat{x},i-1}}.$$

If the original $Q_{\widehat{x}\widehat{x},i-1}$ is large, then $K \sim 1$.

$$\begin{aligned}\widehat{x}_i &= \widehat{x}_{i-1} + \frac{Q_{\widehat{x}\widehat{x}}}{Q_{\ell\ell} + Q_{\widehat{x}\widehat{x}}} \left(\underline{\ell}_i - \widehat{x}_{i-1} \right) = \\ &= \frac{Q_{\widehat{x}\widehat{x},i-1}}{Q_{\ell\ell,i} + Q_{\widehat{x}\widehat{x},i-1}} \underline{\ell}_i + \frac{Q_{\ell\ell,i}}{Q_{\ell\ell,i} + Q_{\widehat{x}\widehat{x},i-1}} \widehat{x}_{i-1} = \frac{Q_{\widehat{x}\widehat{x},i-1} \underline{\ell}_i + Q_{\ell\ell,i} \widehat{x}_{i-1}}{Q_{\ell\ell,i} + Q_{\widehat{x}\widehat{x},i-1}}.\end{aligned}$$

In other words: *thea posteriori* state $\widehat{x}_i$ is the weighted mean of the *a priori* state $\widehat{x}_{i-1}$ and the observation ℓ_i.

$$Q_{\widehat{x}\widehat{x},i} = \left[1 - K \right] Q_{\widehat{x}\widehat{x},i-1} = \frac{Q_{\ell\ell,i}}{Q_{\ell\ell,i} + Q_{\widehat{x}\widehat{x},i-1}} Q_{\widehat{x}\widehat{x},i-1}.$$

In other words: the poorer the *a priori* state variance $Q_{\widehat{x}\widehat{x},i-1}$ compared to the observation precision $Q_{\ell\ell,i}$, the more the updated state variance $Q_{\widehat{x}\widehat{x},i}$ will improve.

4. Laske manuaalisesti läpi molemmat Kalman-havaintotapahtumat ja anna sen jälkeinen tila-arvio $\widehat{x}_1$ ja sen varianssimatriisi. Tilasuureen x alkuarvioksi saa ottaa 0 ja sen varianssimatriisin alkuarvoksi “numeerisesti ääretön”:

$$Q_0 = [100].$$

Vastaus:

Ensimmäinen askel:

$$K_1 = 100 \left(4 + 100 \right)^{-1} = \frac{100}{104}.$$

siis

$$\widehat{x}_1 = 0 + \frac{100}{104} \left(7 - 0 \right) = 6.73$$

$$Q_1 = \left[1 - \frac{100}{104} \right] 100 = \frac{400}{104} = 3.85.$$

Toinen askel:

$$K_2 = 3.85 \left(1 + 3.85 \right)^{-1} = 0.79.$$

$$\begin{aligned}\widehat{x}_2 &= 6.73 + 0.79 \left(5 - 6.73 \right) = \\ &= 6.73 - 0.79 \cdot 1.73 = \\ &= 5.36.\end{aligned}$$

$$Q_2 = \left[1 - 0.79 \right] \cdot 3.85 = 0.81.$$

Chapter 10

Approximation, interpolation, estimation

10.1 Concepts

Approximation means trying to find a function that, in a certain sense, is "as close as possible" to the given function. E.g., a reference ellipsoid, which is as close as possible to the geoid or mean sea surface

An often used rule is the *square integral rule*: if the argument of the function is $x \in D$, we minimize the integral

$$\int_D (\Delta f(x))^2 \, dx,$$

where

$$\Delta f(x) = \overline{f}(x) - f(x),$$

the difference between the function $f(x)$ and its approximation $\overline{f}(x)$. Here, D is the function's *domain*.

Interpolation means trying to find a function that describes the given data points in such a way, that the function values *reproduce* the given data points. This means, that the number of parameters describing the function must be the same as the number of given points.

Estimation is trying to find a function, that is as close as possible to the given data points. The number of parameters describing the function is less than the number of data points, e.g., in linear regression, the number of parameters is two whereas the number of data points may be very large indeed.

"as close as possible" is generally — but not always! — understood in the least squares sense.

- Minimax rule: the greatest occurring residual is minimized
- $L1$ rule: the sum of absolute values of the residuals is minimized.

10.2 Spline interpolation

Traditionally a "spline" has been a flexible strip of wood or ruler, used by shipbuilders to create a smooth curve.

Nowadays a spline is a mathematical function having the same properties. The smoothness minimizes the energy contained in the bending. The function is used for interpolating between given points. Between every pair of neighbouring points there is one polynomial, the value of which (and possibly the values of derivatives) are the same as that of the polynomial in the

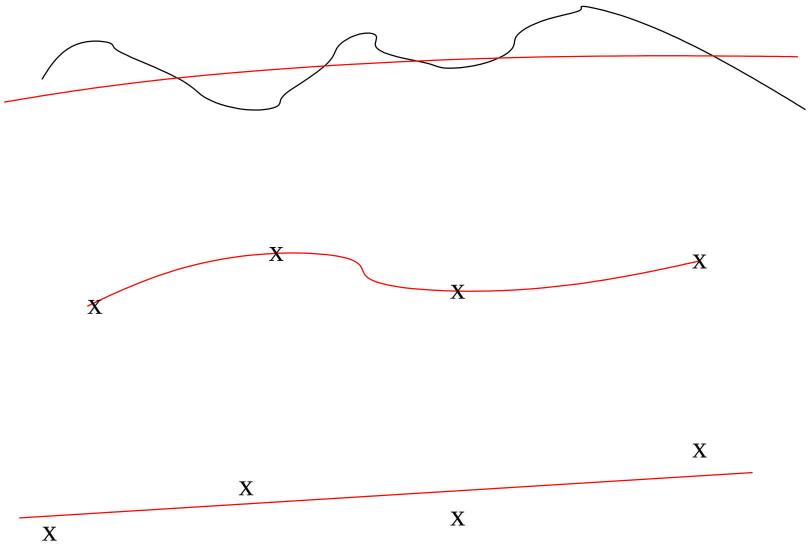


Figure 10.1: Approximation (top), interpolation (middle) and estimation (bottom)

adjoining interval. So, we speak of *piecewise polynomial interpolation*. If the support points are $(x_i, t_i), i = 1, \ldots, n$, the property holds for the spline function f, that $f(t_i) = x(t_i)$, the *reproducing* property.

There exist the following types of splines:

- Linear: the points are connected by straight lines. Piecewise linear interpolation. The function is continuous but not differentiable
- Quadratic: between the points we place *parabolas*. Both the function itself and its first derivative are continuous in the support points
- Cubic. These are the most common[1]. Itse Both function and first and second derivatives are continuous in the support points
- Higher-degree splines.

10.2.1 Linear splines

Ks. `http://mathworld.wolfram.com/CubicSpline.html`.

Linear splines are defined in the following way: let a function be given in the form

$$f_i = f(t_i), i = 1, \ldots, N,$$

where N is the number of support points. Now in the interval $[t_i, t_{i+1}]$, the function $f(t)$ can be approximated by linear interpolation

$$\overline{f}(t) = A_i f_i + B_i f_{i+1},$$

where

$$A_i = \frac{t_{i+1} - t}{t_{i+1} - t_i} \quad B_i = \frac{t - t_i}{t_{i+1} - t_i}.$$

The function A_i is a linear function of t, the value of which is 1 in the point t_i and 0 in the point t_{i+1}. The function $B_i = 1 - A_i$ again is 0 in point t_i and 1 in point t_{i+1}.

[1] Cubic splines are also used in computer typography to describe the shapes of characters, so-called Bézier curves.

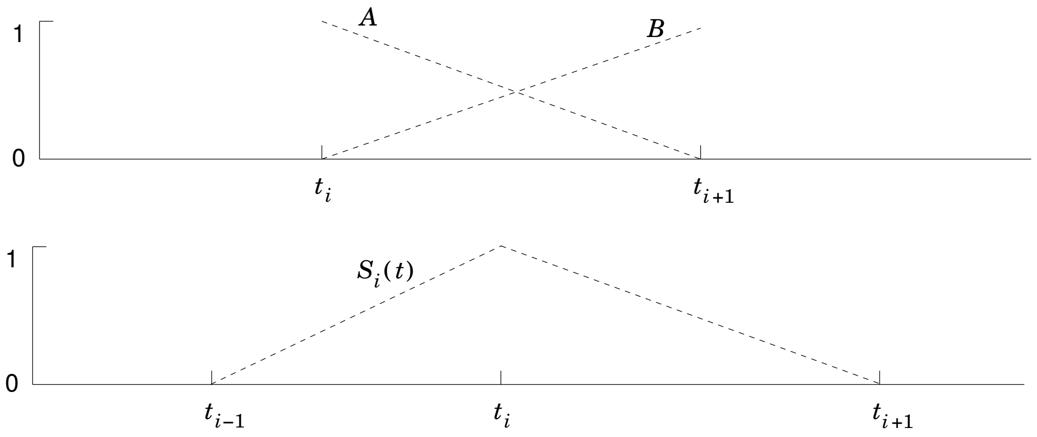


Figure 10.2: Linear spline

Cf. figure 10.2. If we now define for the whole interval $[t_1, t_N]$ the functions

$$S_i(t) = \begin{cases} 0 & \text{jos} \quad t < t_{i-1} \\ B_{i-1} = \frac{t-t_{i-1}}{t_i-t_{i-1}} & \text{jos} \quad t_{i-1} < t < t_i \\ A_i = \frac{t_{i+1}-t}{t_{i+1}-t_i} & \text{jos} \quad t_i < t < t_{i+1} \\ 0 & \text{jos} \quad t > t_{i+1} \end{cases},$$

the graph of which is also drawn (figure 10.2 below). Of course, if i is a border point, half of this “pyramid function” falls away.

Now we may write the function $f(t)$ as the approximation:

$$\overline{f}(t) = \sum_{i=1}^{N} f_i S_i(t),$$

a piecewise linear function.

10.2.2 Cubic splines

Assume given again the values

$$f_i = f(t_i).$$

In the interval $[t_i, t_{i+1}]$ we again approximate the function $f(t)$ by the function

$$\overline{f}(t) = A_i f_i + B_i f_{i+1} + C_i g_i + D_i g_{i+1}, \tag{10.1}$$

in which g_i will still be discussed, and

$$C_i = \frac{1}{6}\left(A_i^3 - A_i\right)(t_{i+1} - t_i)^2 \quad D_i = \frac{1}{6}\left(B_i^3 - B_i\right)(t_{i+1} - t_i)^2.$$

We see immediately, that $A_i^3 - A_i = B_i^3 - B_i = 0$ both in point t_i and in point t_{i+1} (because both A_i and B_i are either 0 or 1 in both points). So, still

$$\overline{f}(t_i) = f(t_i)$$

in the support points.

The values g_i, $i = 1, \ldots, N$ are fixed by requiring the second derivative of the function $\overline{f}(t)$ to be continuous in all support points, and zero[2] in the terminal points 1 and N. Let us derivate equation (10.1):

$$\overline{f''}(t) = f_i \frac{d^2 A_i(t)}{dt^2} + f_{i+1} \frac{d^2 B_i(t)}{dt^2} + g_i \frac{d^2 C_i(t)}{dt^2} + g_{i+1} \frac{d^2 D_i(t)}{dt^2}.$$

Here apparently the first two terms on the right hand side valish, because both A_i and B_i are linear functions in t. We obtain

$$\begin{aligned} \frac{d^2 C_i(t)}{dt^2} &= \frac{d}{dt}\left[\frac{1}{2} A_i^2(t) \frac{dA_i}{dt} - \frac{1}{6}\frac{dA_i}{dt}\right](t_{i+1} - t_i)^2 = \\ &= -\frac{d}{dt}\left[\frac{1}{2} A_i^2(t) - \frac{1}{6}\right](t_{i+1} - t_i) = \\ &= +A_i(t). \end{aligned}$$

Similarly

$$\frac{d^2 D_i(t)}{dt^2} = B_i(t),$$

and we obtain

$$\overline{f''}(t) = A_i g_i + B_i g_{i+1}.$$

So, the parameters g_i are the second derivatives in the support points!

$$g_i = \overline{f''}(t_i).$$

Now, the continuity conditions. The first derivative is

$$\begin{aligned} \overline{f'}(t) &= f_i \frac{dA_i}{dt} + f_{i+1}\frac{dB_i}{dt} + g_i \frac{dC_i}{dt} + g_{i+1}\frac{dD_i}{dt} = \\ &= f_i \frac{-1}{t_{i+1} - t_i} + f_{i+1}\frac{+1}{t_{i+1} - t_i} \\ &\quad -g_i\left[\frac{1}{2}A_i^2 - \frac{1}{6}\right](t_{i+1} - t_i) + \\ &\quad +g_{i+1}\left[\frac{1}{2}B_i^2 - \frac{1}{6}\right](t_{i+1} - t_i) = \\ &= \frac{f_{i+1} - f_i}{t_{i+1} - t_i} + (t_{i+1} - t_i)\left(-g_i\left[\frac{1}{2}A_i^2 - \frac{1}{6}\right] + g_{i+1}\left[\frac{1}{2}B_i^2 - \frac{1}{6}\right]\right). \end{aligned}$$

Let us specialize this to the point $t = t_i$, in the interval $[t_i, t_{i+1}]$:

$$\overline{f'}(t_i) = \frac{f_{i+1} - f_i}{t_{i+1} - t_i} - \left(\frac{1}{3}g_i + \frac{1}{6}g_{i+1}\right)(t_{i+1} - t_i) \tag{10.2}$$

and in the interval $[t_{i-1}, t_i]$:

$$\overline{f'}(t_i) = \frac{f_i - f_{i-1}}{t_i - t_{i-1}} + \left(\frac{1}{6}g_{i-1} + \frac{1}{3}g_i\right)(t_i - t_{i-1}). \tag{10.3}$$

By assuming these to be equal in size, and subtracting them from each other, we obtain

$$\frac{1}{6}(t_i - t_{i-1})\, g_{i-1} + \frac{1}{3}(t_{i+1} - t_{i-1})\, g_i + \frac{1}{6}(t_{i+1} - t_i)\, g_{i+1} = \frac{f_{i+1} - f_i}{t_{i+1} - t_i} - \frac{f_i - f_{i-1}}{t_i - t_{i-1}}.$$

[2] Alternatives: given (fixed)values, continuity condition $f''(t_N) = f''(t_1)$, ...

Here the number of unknowns is N: $g_i, i = 1, \ldots, N$. The number of equations is $N - 2$. Additional equations are obtained from the edges, e.g., $g_1 = g_N = 0$. Then, all g_i can be solved for:

$$\frac{1}{6}\begin{bmatrix} 2(t_2-t_1) & t_2-t_1 & & & & & \\ t_2-t_1 & 2(t_3-t_1) & t_3-t_2 & & & & \\ & t_3-t_2 & 2(t_4-t_2) & t_4-t_3 & & & \\ & & t_4-t_3 & 2(t_5-t_3) & t_5-t_4 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & t_{N-1}-t_{N-2} & 2(t_N-t_{N-2}) & t_N-t_{N-1} \\ & & & & & t_N-t_{N-1} & 2(t_N-t_{N-1}) \end{bmatrix}\cdot$$

$$\cdot\begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ \vdots \\ g_{N-1} \\ g_N \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ \vdots \\ b_{N-1} \\ b_N \end{bmatrix},$$

where

$$b_i = \frac{f_{i+1}-f_i}{t_{i+1}-t_i} - \frac{f_i - f_{i-1}}{t_i - t_{i-1}},\ i = 2, \ldots, N-1;\ b_1 = \frac{f_2 - f_1}{t_2 - t_1}, b_N = -\frac{f_N - f_{N-1}}{t_N - t_{N-1}}$$

This is a so-called *tridiagonal* matrix, for the solution of the associated system of equations of which exist efficient special algorithms.

In case the support points are *equidistant*, i.e., $t_{i+1} - t_i = \Delta t$, we obtain[3]

$$\frac{\Delta t^2}{6}\begin{bmatrix} 2 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & 1 & 4 & 1 & & & \\ & & 1 & 4 & 1 & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & 1 & 4 & 1 \\ & & & & & 1 & 2 \end{bmatrix}\begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ \vdots \\ g_{N-1} \\ g_N \end{bmatrix} = \begin{bmatrix} f_2 - f_1 \\ f_3 - 2f_2 + f_1 \\ f_4 - 2f_3 + f_2 \\ \vdots \\ f_{N-1} - 2f_{N-2} + f_{N-3} \\ f_N - 2f_{N-1} + f_{N-2} \\ -f_N + f_{N-1} \end{bmatrix}.$$

10.3 Finite element method

The finite element method is used to solve multidimensional field problems, so-called *boundary value problems*, that can be described by partial differential equations. In geodesy, this means mostly the gravity field.

10.3.1 Example

Let us first study a simple example. The problem domain is

$$D : [0,1] \times [0,1] = \{(x,y), 0 \leq x < 1, 0 \leq y < 1\}.$$

[3] In case of a circular boundary condition, the 2 in the corners of the matrix change into 4, and b_1 and b_N are modified correspondingly.

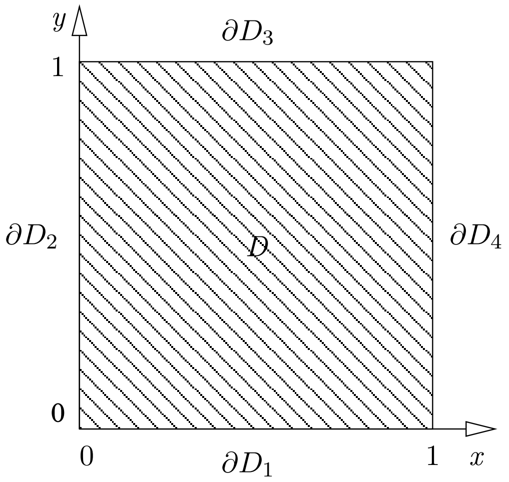


Figure 10.3: A simple domain

I.e., a square of size unity in the plane. The boundary of the domain may be called ∂D and it consists of four parts $\partial D_1 \ldots \partial D_4$, see figure.

Let g now be a real-valued function on D Our problem is finding a function $u(x, y)$, i.e., a *solution*, with the following properties:

1. Twice differentiable on D. Let us call the set of all such functions by the name V.
2. $u_{xx} + u_{yy} = g$ on the domain D
3. *Periodical* boundary conditions, i.e.,

   a) $u(x, y) = u(x + 1, y)$ and
   b) $u(x, y) = u(x, y + 1)$.

   We may visualize this by rolling up D into a *torus*, i.e., the topology of a torus.

The expression

$$u_{xx} + u_{yy} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial y^2}$$

is often called Δu where the delta operator

$$\Delta \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$$

is referred to as the Laplace operator in two dimensions. E.g., the gravitational field in vacuum or the flow of an incompressible fluid can be described by

$$\Delta u = 0.$$

In the case of our example

$$\Delta u = g,$$

and g is called the *source function*, e.g., in the case of the gravitational field $4\pi G\rho$, where G is Newton's gravitational constant and ρ the density of matter.

10.3.2 The "weak" formulation of the problem

The problem $\Delta u = g$ can also be formulated in the following form. Let ϕ be a *functional* in V — i.e., a map producing for every *function* $v \in V$ a real value $\phi(v)$ —, so, that

$$\phi(tu + v) = t\phi(u) + \phi(v),$$

nimelläi.e., a *linear* functional. Let us call the set of all such linear functionals V^*.

Then, the following statements are equivalent:

$$\Delta u = g$$

and

$$\forall \phi \in V^* : \phi(\Delta u) = \phi(g).$$

This is called the *weak formulation* of the problem $\Delta u = g$.

10.3.3 The bilinear form of the delta operator

In fact we don't have to investigate the whole set V^*, it suffices to look at all functionals of form

$$\phi_v(f) \equiv \int_0^1 \int_0^1 v(x,y) f(x,y)\, dxdy,$$

where $v(x,y)$ satisfies the (periodical) boundary conditions that were already presented.

So now the problem is formulated as that of finding a $u \in V$ so that

$$\phi_v(\Delta u) = \phi_v(g) \tag{10.4}$$

for all $v \in V$.

Using integration by parts we may write

$$\begin{aligned} \int_0^1 \int_0^1 v u_{xx} dxdy &= \int_0^1 [v u_x]_0^1 \, dy - \int_0^1 \int_0^1 v_x u_x dxdy, \\ \int_0^1 \int_0^1 v u_{yy} dxdy &= \int_0^1 [v u_y]_0^1 \, dx - \int_0^1 \int_0^1 v_y u_y dxdy. \end{aligned}$$

Because of the periodical boundary condition, the first terms on the right hand side vanish, and by summation we obtain

$$\int_0^1 \int_0^1 v(u_{xx} + u_{yy})\, dxdy = -\int_0^1 \int_0^1 (v_x u_x + v_y u_y)\, dxdy.$$

Thus we find, that

$$\phi_v(\Delta u) = -\int_0^1 \int_0^1 (v_x u_x + v_y u_y)\, dxdy.$$

Let us call this[4]

$$\psi(u,v) \equiv \phi_v(\Delta u) = \phi_v(g) = \int_0^1 \int_0^1 v(x,y)\, g(x,y)\, dxdy.$$

[4] This is the bilinear form of the operator Δ.

Now we obtain the weak formulation (10.4) of the problem as the integral equation

$$-\int_0^1\int_0^1 (v_x u_x + v_y u_y)\, dxdy = \int_0^1\int_0^1 vgdxdy.$$

In this equation appear only the first derivatives with respect to place of the functions u, v: if we write

$$\nabla v \equiv \begin{bmatrix} \frac{\partial v}{\partial x} \\ \frac{\partial v}{\partial y} \end{bmatrix}, \nabla u \equiv \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix},$$

(where ∇, or *nabla*, is the *gradient* operator) we can write

$$-\int_0^1\int_0^1 \langle \nabla v \cdot \nabla u\rangle\, dxdy = \int_0^1\int_0^1 vgdxdy.$$

10.3.4 Test functions

Next, we specialize the function ν as a series of *test functions.* Let the set of suitable test functions (countably infinite) be

$$E \equiv \{e_1, e_2, e_3, \ldots\}.$$

Let us demand that for all e_i

$$\psi(u, e_i) = \int_0^1\int_0^1 ge_i dxdy. \tag{10.5}$$

In order to solve this problem we write

$$u = u_1e_1 + u_2e_2 + \ldots = \sum_{i=1}^{\infty} u_ie_i.$$

In practice we use from the infinite set E only a finite subset $E_n = \{e_1, e_2, \ldots, e_n\} \subset E$, and also the expansion of u is truncated. Now the problem has been reduced to the determination of n coefficients $u_1, u_2, \ldots u_n$ from n equations:

$$u = \sum u_ie_i, \tag{10.6}$$

$$g = \sum g_ie_i. \tag{10.7}$$

Now we *discretise* the domain D in the following way: we divide it into triangles having common borders and corner points, see figure.

To every nodal point i we attach one test function e_i, which looks as follows:

1. Inside every triangle it is linear
2. It is 1 in node i and 0 in all other nodes
3. It is continuous and “piecewise” differentiable.

See figure.

Now the above set of equations (10.5) after the substitutions (10.6, 10.7) has the following form:

$$\sum_{j=1}^{n} \psi(e_j, e_i)\, u_j = \sum_{j=1}^{n} g_j \int_0^1\int_0^1 e_je_i dxdy,\ i = 1, \ldots, n,$$

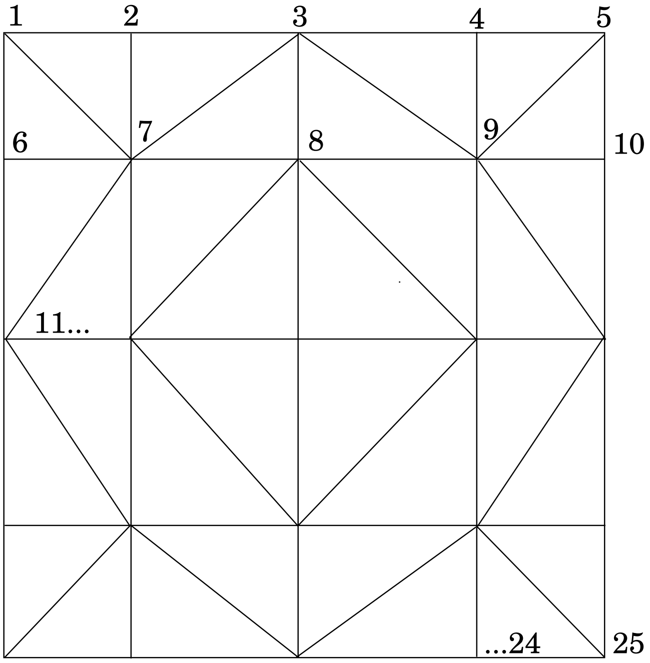


Figure 10.4: Triangulation of the domain and numbers of nodes

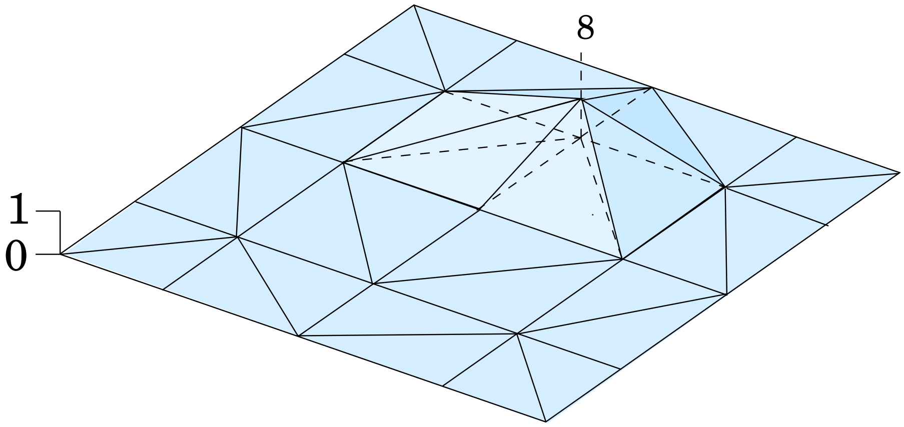


Figure 10.5: Test function e_8

or as a matric equation:

$$P\mathrm{u} = Q\mathrm{g},$$

where

$$\mathrm{u} = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} \text{ ja g} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{bmatrix}.$$

The matrices are

$$\begin{aligned} P_{ji} &= \psi(e_j, e_i) = -\int_0^1 \int_0^1 \langle \nabla e_j \cdot \nabla e_i \rangle \, dxdy, \\ Q_{ji} &= \int_0^1 \int_0^1 e_j e_i dxdy. \end{aligned}$$

The P matrix is called the *stiffness matrix* and the Q matrix the *mass matrix.*

10.3.5 Computing the matrices

In order to calculate the elements of the matrix P, we look at the triangle ABC. The test functions are in this case the, already earlier presented, *barycentric co-ordinates*:

$$e_A = \frac{\begin{vmatrix} x_B & x_C & x \\ y_B & y_C & y \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_A & x_B & x_C \\ y_A & y_B & y_C \\ 1 & 1 & 1 \end{vmatrix}}, \; e_B = \frac{\begin{vmatrix} x_C & x_A & x \\ y_C & y_A & y \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_A & x_B & x_C \\ y_A & y_B & y_C \\ 1 & 1 & 1 \end{vmatrix}}, \; e_C = \frac{\begin{vmatrix} x_A & x_B & x \\ y_A & y_B & y \\ 1 & 1 & 1 \end{vmatrix}}{\begin{vmatrix} x_A & x_B & x_C \\ y_A & y_B & y_C \\ 1 & 1 & 1 \end{vmatrix}}.$$

These can be computed straightforwardly. The gradients again are

$$\begin{aligned} \nabla e_A = \begin{bmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \end{bmatrix} e_A &= \left(\begin{vmatrix} x_A & x_B & x_C \\ y_A & y_B & y_C \\ 1 & 1 & 1 \end{vmatrix} \right)^{-1} \begin{bmatrix} \begin{vmatrix} y_B & y_C \\ 1 & 1 \end{vmatrix} \\ -\begin{vmatrix} x_B & x_C \\ 1 & 1 \end{vmatrix} \end{bmatrix} = \\ &= \left(\begin{vmatrix} x_A & x_B & x_C \\ y_A & y_B & y_C \\ 1 & 1 & 1 \end{vmatrix} \right)^{-1} \begin{bmatrix} y_B - y_C \\ x_C - x_B \end{bmatrix}, \end{aligned}$$

and so on for the gradients ∇e_B and ∇e_C, cyclically changing the names A, B, C. We obtain

$$\langle \nabla e_A \cdot \nabla e_A \rangle = \left(\begin{vmatrix} x_A & x_B & x_C \\ y_A & y_B & y_C \\ 1 & 1 & 1 \end{vmatrix} \right)^{-2} \left\| \overrightarrow{BC} \right\|^2$$

and

$$\langle \nabla e_A \cdot \nabla e_B \rangle = \left(\begin{vmatrix} x_A & x_B & x_C \\ y_A & y_B & y_C \\ 1 & 1 & 1 \end{vmatrix} \right)^{-2} \left\langle \overrightarrow{BC} \cdot \overrightarrow{CA} \right\rangle,$$

and so forth.

The gradients are constants, so we can compute the integral over the whole triangle by multiplying it by the surface are, which happens to be $\frac{1}{2} \begin{vmatrix} x_A & x_B & x_C \\ y_A & y_B & y_C \\ 1 & 1 & 1 \end{vmatrix}$.When we have computed

$$\iint_{\Delta} \langle \nabla e_j \cdot \nabla e_i \rangle \, dxdy$$

over all triangles — six values for every triangle —, the elements of P are easily computed by summing over all the triangles belonging to the test function. Because these triangles are only small in number, is the matrix P in practice *sparse*, which is a substantial numerical advantage.

Computing, and integrating over the triangle, the terms $e_A e_B$ etc. for the computation of the Q matrix is left as an exercise.

10.3.6 Solving the problem

As follows:

1. Compute (generate) the matrices P and Q. Matlab offers ready tools for this
2. Compute (solve) from the function $g(x, y)$ the coefficients g_i, i.e., the elements of the vector g, from the equations
$$\int_0^1 \int_0^1 g(x, y) \, e_j(x, y) \, dxdy = \sum_i g_i \int_0^1 \int_0^1 e_i(x, y) \, e_j(x, y) \, dxdy, \; j = 1, \ldots, n.$$
3. Solve the matric equation $P\mathrm{u} = Q\mathrm{g}$ for the unknown u and its elements u_i
4. Compute $u(x, y) = \sum u_i e_i$. Draw on paper or plot on screen.

10.3.7 Different boundary conditions

If the boundary conditions are such, that in the key integration by parts

$$\begin{aligned} \int_0^1 [vu_x]_0^1 \, dy + \int_0^1 [vu_y]_0^1 \, dx &= \\ &= \int_0^1 (v(1, y) \, u_x(1, y) - v(0, y) \, u_x(0, y)) \, dy + \\ &+ \int_0^1 (v(x, 1) \, u_y(x, 1) - v(x, 0) \, u_y(x, 0)) \, dx \end{aligned}$$

do not vanish, then those integrals too must be evaluated over boundary elements: we obtain integrals shaped like

$$\begin{gathered} \int_0^1 e_j(0, y) \frac{\partial}{\partial x} e_i(0, y) \, dy, \int_0^1 e_j(1, y) \frac{\partial}{\partial x} e_i(1, y) \, dy, \\ \int_0^1 e_j(x, 0) \frac{\partial}{\partial y} e_i(x, 0) \, dx, \int_0^1 e_j(x, 1) \frac{\partial}{\partial y} e_i(x, 1) \, dx \end{gathered} \tag{10.8}$$

i.e., one-dimensional integrals along the edge of the domain. In this case we must distinguish *internal* nodes and elements from *boundary* nodes and elements. The above integrals differ from zero only if e_i and e_j are both boundary elements. The boundary condition is often given in the following form:

$$u(x,y) = h(x,y) \text{ at the domain edge} \partial D.$$

This is a so-called DIRICHLET boundary value problem. Write

$$h(x,y) = \sum h_i e_i(x,y)$$

like earlier for the u and g functions.

Alternatively, the NEUMANN- problem, where given is the *normal derivative* of the solution function on the boundary:

$$\frac{\partial}{\partial n} u(x,y) = h(x,y) \text{ at the domain edge} \partial D.$$

In case the edge is not a nice square, we can use the GREEN *theorem in order to do integration by parts. Then we will again find integrals on the boundary that contain both the test functions e_i themselves and their first derivatives* in the normal direction $\frac{\partial}{\partial n} e_j$. Just like we already saw above (equation 10.8).

Also the generalization to three dimensional problems and problems developing in time, where we have the additional dimension of time t, must be obvious. In that case we have, instead of, or in addition to, boundary conditions, *initial conditions.*

10.4 Function spaces and Fourier theory

In an *abstract vector space* we may create a *base*, with the help of which any vector can be written as a linear combination of the base vectors: if the base is $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$, we may write an arbitrary vector $\mathbf{r}$ in the form:

$$\mathbf{r} = r_1\mathbf{e}_1 + r_2\mathbf{e}_2 + r_3\mathbf{e}_3 = \sum_{i=1}^{3} r_i \mathbf{e}_i.$$

Because three base vectors are always enough, we call ordinary space *three-dimensional*.

We can define to a vector space a *scalar product*, which is a linear map from two vectors to one number ("bilinear form"):

$$\langle \mathbf{r} \cdot \mathbf{s} \rangle .$$

Linearity means, that

$$\langle \alpha \mathbf{r}_1 + \beta \mathbf{r}_2 \cdot \mathbf{s} \rangle = \alpha \langle \mathbf{r}_1 \cdot \mathbf{s} \rangle + \beta \langle \mathbf{r}_2 \cdot \mathbf{s} \rangle ,$$

and symmetry means, that

$$\langle \mathbf{r} \cdot \mathbf{s} \rangle = \langle \mathbf{s} \cdot \mathbf{r} \rangle$$

If the base vectors are mutually *orthogonal*, i.e., $\langle \mathbf{e}_i \cdot \mathbf{e}_j \rangle = 0$ if $i \neq j$, we can simply calculate the coefficients r_i:

$$\mathbf{r} = \sum_{i=1}^{3} \frac{\langle \mathbf{r} \cdot \mathbf{e}_i \rangle}{\langle \mathbf{e}_i \cdot \mathbf{e}_i \rangle} \mathbf{e}_i = \sum_{i=1}^{3} r_i \mathbf{e}_i \qquad (10.9)$$

If additionally still $\langle \mathbf{e}_i \cdot \mathbf{e}_i \rangle = \|\mathbf{e}_i\|^2 = 1 \ \forall i \in \{1,2,3\}$, in other words, the base vectors are *orthonormal* – the quantity $\|\mathbf{r}\|$ is called the *norm* of the vector $\mathbf{r}$ – then equation 10.9 simplifies even further:

$$r_i = \sum_{i=1}^{3} \langle \mathbf{r} \cdot \mathbf{e}_i \rangle \, \mathbf{e}_i . \tag{10.10}$$

Here, the coefficients $r_i = \langle \mathbf{r} \cdot \mathbf{e}_i \rangle$.

Also *functions* can be considered as elements of a vector space. If we define the scalar product of two functions f, g as the following integral:

$$\left\langle \overrightarrow{f} \cdot \overrightarrow{g} \right\rangle \equiv \frac{1}{\pi} \int_0^{2\pi} f(x)\, g(x)\, dx,$$

it is easy to show that the above requirements for a scalar product are met.

One particular base of this vector space (a *function space*) is formed by the so-called Fourier functions,

$$\begin{aligned} \overrightarrow{e_0} &= \frac{1}{2}\sqrt{2} \quad (k=0) \\ \overrightarrow{e_k} &= \cos kx, \ k = 1,2,3,... \\ \overrightarrow{e_{-k}} &= \sin kx, \ k = 1,2,3,... \end{aligned}$$

This base is *orthonormal* (proof: exercise). It is also a *complete* basis, which we shall not prove. Now every function $f(x)$ that satifies certain conditions, can be expanded according to equation (10.10) , i.e.,

$$f(x) = a_0 \frac{1}{2}\sqrt{2} + \sum_{k=1}^{\infty} \left(a_k \cos kx + b_k \sin kx \right),$$

– the familiar Fourier expansion – where the coefficients are

$$\begin{aligned} a_0 &= \left\langle \overrightarrow{f} \cdot \overrightarrow{e_0} \right\rangle = \frac{1}{\pi} \int_0^{2\pi} f(x) \frac{1}{2}\sqrt{2} dx = \sqrt{2} \cdot \overline{f(x)} \\ a_k &= \left\langle \overrightarrow{f} \cdot \overrightarrow{e_k} \right\rangle = \frac{1}{\pi} \int_0^{\pi} f(x) \cos kx dx \\ b_k &= \left\langle \overrightarrow{f} \cdot \overrightarrow{e_{-k}} \right\rangle = \frac{1}{\pi} \int_0^{2\pi} f(x) \sin kx dx \end{aligned}$$

This is the familiar way in which the coefficients of a Fourier series are computed.

10.5 Wavelets

The disadvantage of the base functions, i.e., sines and cosines, used in Fourier analysis, is that they extend over the whole domain of study, as function values differing from zero.

Often we would wish to use base functions, that are different from zero only on a bounded area. Of course the base functions of finite elements presented above are of such nature. They are however not "wave-like" by their nature.

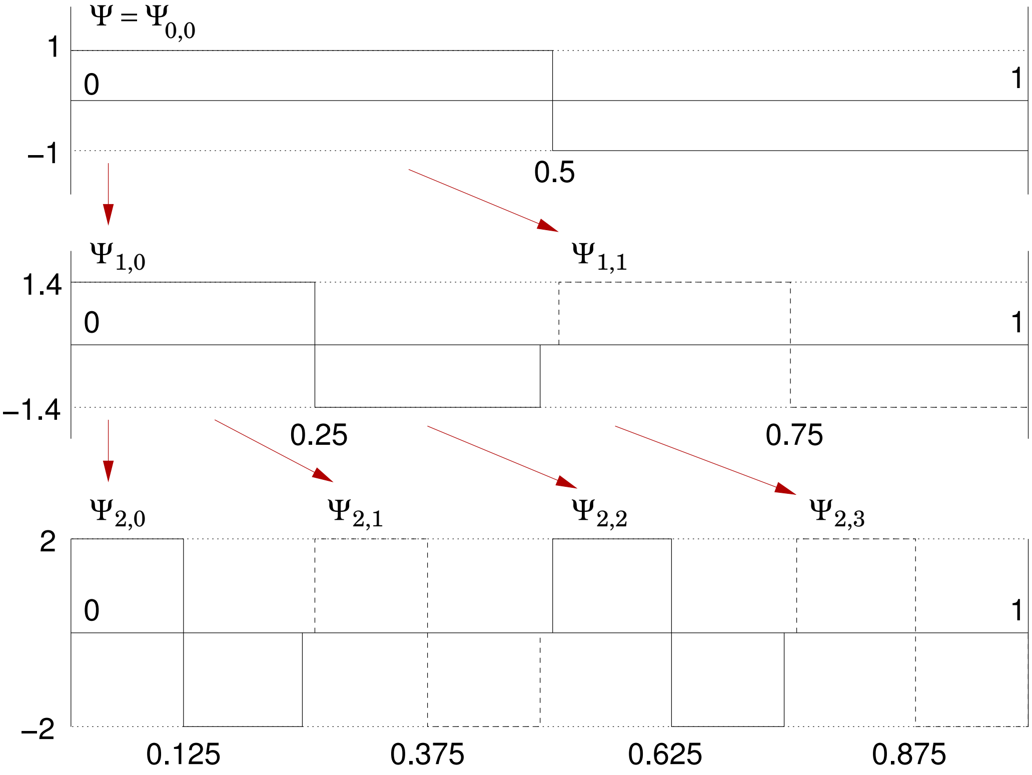


Figure 10.6: Haar wavelets

The solution is given by *wavelets*. A wavelet is a wave-like function that is of bounded support. There are wavelets of different levels; higher level wavelets have a smaller area of support, but within it, offers a higher resolution. A higher level wavelet is obtained from a lower level one by *scaling and shifting*.

The simplest of all “mother wavelets”, and a good example, is the so-called HAAR wavelet. Its form is

$$\psi(x) = \begin{cases} 1 \text{ if } 0 < x < \frac{1}{2} \\ -1 \text{ if } \frac{1}{2} < x < 1 \\ 0 \text{ elsewhere} . \end{cases}$$

From this, we can then obtain all other necessary wavelets as follows:

$$\psi_{j,k}(x) = 2^{j/2}\psi\left(2^j x - k\right) .$$

So, the wavelet $\psi_{j,k}$ is 2^j times narrower than ψ, $\sqrt{2^j}$ times taller, and shifted along the horizontal axis by an amount k to the right.

The number j is called the *resolution level*, the number k the *location number*.

From one mother wavelet we obtain 2 first level, 4 second level, 8 third level daughters, etc.

It can be easily verified that wavelets are *orthonormal*: the “dot product” of the function space is

$$\langle \psi_{j,k} \cdot \psi_{j'.k'} \rangle = \int_0^1 \psi_{j,k}(x)\,\psi_{j',k'}(x)\,dx = \begin{cases} 1 \text{ if } j = j' \text{ and } k = k' \\ 0 \text{ otherwise} . \end{cases}$$

For this reason we may write an arbitrary function as the following series expansion:

$$f(x) = \sum_{j=0}^{\infty} \sum_{k=0}^{2^j-1} f_{j,k}\psi_{j,k}(x) ,$$

where the coefficients are

$$f_{j,k} = \int_0^1 f(x)\,\psi_{j,k}(x)\,dx$$

i.e, again, the dot product in function space $\langle f \cdot \psi_{j,k} \rangle$, the projection of f on the axis $\psi_{j,k}$.

Try the function

$$f(x) = \sin 2\pi x.$$

Compute

$$\begin{aligned}
\langle f \cdot \psi_{0,0} \rangle &= \int_0^{0.5} \sin 2\pi x - \int_{0.5}^1 \sin 2\pi x = \\
&= 2\int_0^{0.5} \sin 2\pi x = -\frac{1}{\pi}\left[\cos 2\pi x\right]_0^{0.5} = \\
&= \frac{2}{\pi} \approx 0.6366.
\end{aligned}$$

Also (symmetry)

$$\langle f \cdot \psi_{1,0} \rangle = \langle f \cdot \psi_{1,1} \rangle = 0$$

and

$$\begin{aligned}
\langle f \cdot \psi_{2,0} \rangle &= 2\int_0^{0.125} \sin 2\pi x \cdot 2dx - 2\int_{0.125}^{0.25} \sin 2\pi x \cdot 2dx = \\
&= -\frac{1}{\pi}\left[\cos 2\pi x\right]_0^{0.125} + \frac{1}{\pi}\left[\cos 2\pi x\right]_{0.125}^{0.25} = \\
&= -\frac{1}{\pi}\left(\cos\frac{\pi}{4} - \cos 0\right) + \frac{1}{\pi}\left(\cos\frac{\pi}{2} - \cos\frac{\pi}{4}\right) = \\
&= \frac{1}{\pi}\left(1 - \frac{1}{2}\sqrt{2} + 0 - \frac{1}{2}\sqrt{2}\right) = \\
&= \frac{1}{\pi}\left(1 - \sqrt{2}\right) \approx -0.1318.
\end{aligned}$$

Using the symmetry argument, we now obtain

$$\begin{aligned}
\langle f \cdot \psi_{2,1} \rangle &= \frac{1}{\pi}\left(\sqrt{2} - 1\right) \approx +0.1318, \\
\langle f \cdot \psi_{2,2} \rangle &= \frac{1}{\pi}\left(\sqrt{2} - 1\right) \approx +0.1318, \\
\langle f \cdot \psi_{2,3} \rangle &= \frac{1}{\pi}\left(1 - \sqrt{2}\right) \approx -0.1318.
\end{aligned}$$

With the aid of this we may perform a synthesis: the result is a stepping function, as drawn in the figure.

The result is somewhat blocky; in the literature, much more “wave-like” wavelets can be found.The stepwise, hierarchically bisecting improvement of the Haar wavelets resembles, how a GIF image is built up stepwise on a computer screen, if it arrives over a slow line. In fact, one large application area for wavelets is precisely the compression of imagery.

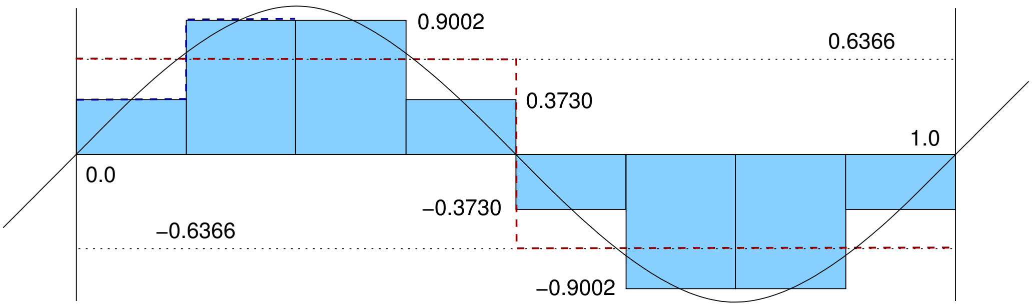


Figure 10.7: A sine function expanded into Haar wavelets

10.6 Legendre and Chebyshev approximation

10.6.1 Polynomial fit

If we are asked to approximate a function given by measurement values on the interval $[-1,1]$, a logical approach is to try and use polynomial fit. We describe the function byon

$$f(x)=\sum_{i=0}^{\infty}a_ix^i,$$

and estimate the coefficients $a_i$from the data. In practice, the series is truncated at $i=I$: the approximation obtained is then

$$\widetilde{f}(x)=\sum_{i=0}^{I}a_ix^i.$$

This can be written as an observation equation as

$$\widetilde{f}(x)=\left[\begin{array}{ccccc}1 & x & x^2 & \cdots & x^I\end{array}\right]\left[\begin{array}{c}a_0\\a_1\\a_2\\\vdots\\a_I\end{array}\right].$$

Now, let us have *observations* regularly spread out over the interval $[-1,1]$, e.g., at the points $-1,-0.5,0,+0.5$ and $+1$. Let us also assume, for the sake of example, that $I=3$. Then the set of observation equations becomes

$$\left[\begin{array}{c}\widetilde{f}(-1)\\\widetilde{f}(-0.5)\\\widetilde{f}(0)\\\widetilde{f}(0.5)\\\widetilde{f}(1)\end{array}\right]=\left[\begin{array}{cccc}1 & -1 & 1 & -1\\1 & -0.5 & 0.25 & -0.125\\1 & 0 & 0 & 0\\1 & 0.5 & 0.25 & 0.125\\1 & 1 & 1 & 1\end{array}\right]\left[\begin{array}{c}a_0\\a_1\\a_2\\a_3\end{array}\right].$$

The matrix in the above equation is A, the design matrix. From it, the normal matrix is calculated as

$$N=A^TA=\left[\begin{array}{cccc}5 & 0 & 2.5 & 0\\0 & 2.5 & 0 & 2.125\\2.5 & 0 & 2.125 & 0\\0 & 2.125 & 0 & 2.03125\end{array}\right].$$

The condition number of this matrix is $\lambda_{\max}/\lambda_{\min} \approx 50$. It is clearly non-diagonal. On the basis of experience we may say, that polynomial fit in these cases is a suitable method only for low polynomial degree numbers I. Already for values $I > 12$ the solution begins to be so poorly conditioned, that numerical precision begins to suffer.

10.6.2 Legendre interpolation

See `http://en.wikipedia.org/wiki/Legendre_polynomials`.

We can choose as base functions, instead of simple polynomials $1, x, x^2, x^3, \ldots$, Legendre[5] *polynomials*, which have the useful property of *orthogonality* on the interval $[-1, 1]$: if we formally define the inner product of two functions $f(x)$ and $g(x)$ as the integral

$$\left\langle \overrightarrow{f} \cdot \overrightarrow{g} \right\rangle = \int_{-1}^{+1} f(x)\, g(x)\, dx,$$

then we can say for the Legendre polynomials $P_n(x)$,that

$$\langle P_n \cdot P_m \rangle = \int_{-1}^{+1} P_n(x)\, P_m(x)\, dx = \left\{ \begin{array}{cc} 0 & m \neq n \\ \frac{2}{2n+1} & m = n \end{array} \right. .$$

The Legendre polynomials are most easily generated by the following *recursive relationship*:

$$nP_n(x) = -(n-1)\, P_{n-2}(x) + (2n-1)\, xP_{n-1}(x)\,.$$

In this way we find

$$\begin{array}{rcl} P_0(x) & = & 1, \\ P_1(x) & = & x, \\ P_2(x) & = & \dfrac{3}{2}x^2 - \dfrac{1}{2}, \\ P_3(x) & = & \dfrac{5}{2}x^3 - \dfrac{3}{2}x, \end{array}$$

etcetera.

Now if we write our approximation of function $f(x)$ as follows:

$$\widetilde{f}(x) = \sum_{i=0}^{I} a_i P_i(x)\,,$$

we obtain again for a row of our observation equation:

$$\widetilde{f}(x) = \left[\begin{array}{ccccc} P_0(x) & P_1(x) & P_2(x) & \cdots & P_I(x) \end{array} \right] \left[\begin{array}{c} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_I \end{array} \right].$$

[5] Adrien-Marie Legendre, 1752 - 1833, French mathematician.

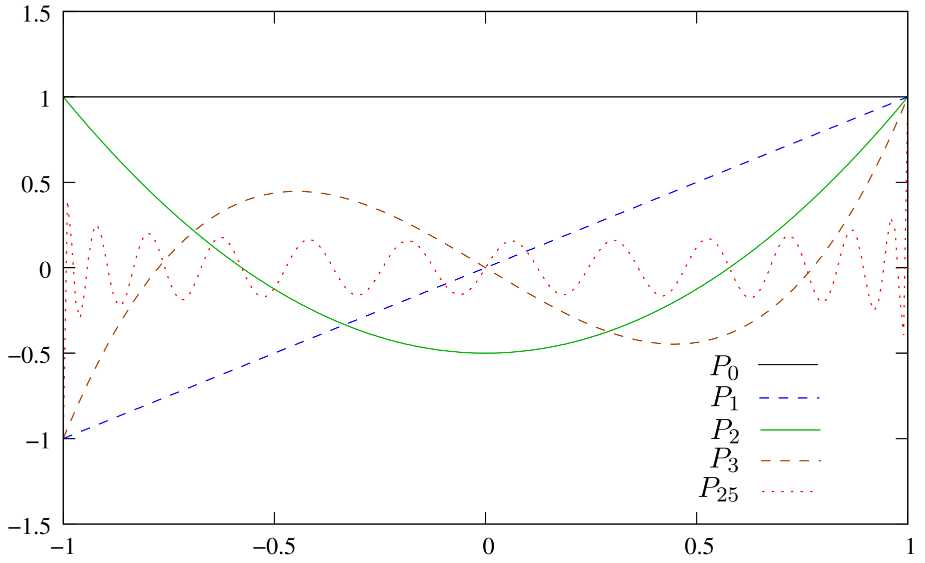


Figure 10.8: Examples of LEGENDRE polynomials

Again choosing the values $-1, -0.5, 0, 0.5$ and 1 yields:

$$\begin{bmatrix} \widetilde{f}(-1.0) \\ \widetilde{f}(-0.5) \\ \widetilde{f}(0.0) \\ \widetilde{f}(0.5) \\ \widetilde{f}(1.0) \end{bmatrix} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & -0.5 & -0.125 & 0.4375 \\ 1 & 0 & -0.5 & 0 \\ 1 & 0.5 & -0.125 & -0.4375 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix},$$

and the corresponding normal matrix is

$$N = A^T A = \begin{bmatrix} 5 & 0 & 1.25 & 0 \\ 0 & 2.5 & 0 & 1.5625 \\ 1.25 & 0 & 2.28125 & 0 \\ 0 & 1.5625 & 0 & 2.38281 \end{bmatrix}.$$

Now, the condition number $\lambda_{\text{max}}/\lambda_{\text{min}}$ is 6.25, a lot better than for simple polynomials!

The normal matrix looks approximately, but not precisely, diagonal. If we had a larger number of support points, all spread uniformly over the interval $[-1, 1]$, we would see the N matrix become very nearly a diagonal matrix. (And even if some of the support points would be missing, the matrix would still be close to diagonal.)

What this means is that the polynomial approximation done this way is *more stable* even for very high polynomial degree numbers. Evaluating each polynomial $P_n(x)$ for a given support point argument x can be done very efficiently using the above given recurrence relationship.

10.6.3 Chebyshev interpolation

See `http://en.wikipedia.org/wiki/Chebyshev_approximation#Chebyshev_approximation`, `http://en.wikipedia.org/wiki/Chebyshev_polynomials`.

Another kind of polynomials often used for interpolation are CHEBYSHEV[6] *polynomials* of the

[6] Pafnuty Lvovich CHEBYSHEV, 1821 - 1894, Russian mathematician.

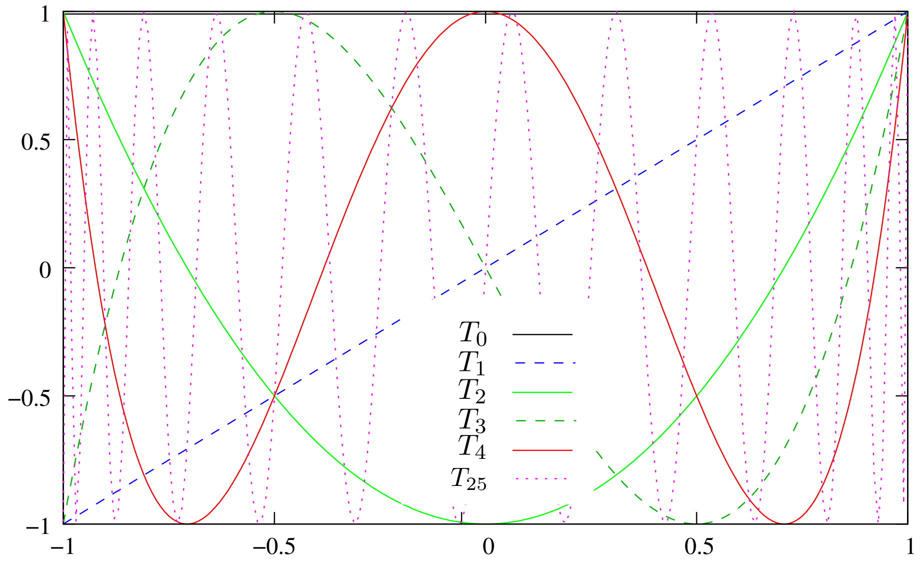


Figure 10.9: Examples of Chebyshev polynomials.

first kind. They can be formally defined as[7]

$$T_n(x) = \cos(n \arccos x). \tag{10.11}$$

Like Legendre's polynomials, they are easily computed recursively:

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x),$$

starting from $T_0(x) = 1$ and $T_1(x) = x$. The first few polynomials are:

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x, \end{aligned}$$

and so on.

Like Legendre's polynomials, also Chevyshev's polynomials satisfy an orthogonality relationship, but for a different inner product: if we define

$$\left\langle \overrightarrow{f} \cdot \overrightarrow{g} \right\rangle = \int_{-1}^{+1} \frac{f(x)\, g(x)}{\sqrt{1-x^2}} dx, \tag{10.12}$$

where we call $(1-x^2)^{-1/2}$ the *weighting factor*, we have

$$\langle T_n \cdot T_m \rangle = \int_{-1}^{+1} \frac{T_n(x)\, T_m(x)}{\sqrt{1-x^2}} dx = \begin{cases} 0 & n \neq m \\ \pi & n = m = 0 \\ \pi/2 & n = m \neq 0 \end{cases}.$$

[7] T like in the French transliteration Tshebyshev.

Again, we may approximate a function f as follows:

$$\widetilde{f}(x)=\sum_{i=0}^{I}a_iT_i(x), \tag{10.13}$$

from which the observation equation for the coefficients a_i becomes

$$\widetilde{f}(x)=\begin{bmatrix} T_0(x) & T_1(x) & T_2(x) & \cdots & T_I(x)\end{bmatrix}\begin{bmatrix} a_0\\ a_1\\ a_2\\ \vdots\\ a_I\end{bmatrix}.$$

For the same case of observed function values in support points $-1, -0.5, 0, 0.5$ and 1 we get:

$$\begin{bmatrix}\widetilde{f}(-1.0)\\ \widetilde{f}(-0.5)\\ \widetilde{f}(0.0)\\ \widetilde{f}(0.5)\\ \widetilde{f}(1.0)\end{bmatrix}=\begin{bmatrix}1 & -1 & 1 & -1\\ 1 & -0.5 & -0.5 & 1\\ 1 & 0 & -1 & 0\\ 1 & 0.5 & -0.5 & -1\\ 1 & 1 & 1 & 1\end{bmatrix}\begin{bmatrix}a_0\\ a_1\\ a_2\\ a_3\end{bmatrix}.$$

The normal matrix is

$$N=A^TA=\begin{bmatrix}5 & 0 & 0 & 0\\ 0 & 2.5 & 0 & 1\\ 0 & 0 & 3.5 & 0\\ 0 & 1 & 0 & 4\end{bmatrix},$$

with a condition number of $\lambda_{\max}/\lambda_{\min}=2.5$, which is pretty good!

This all looks very interesting... but what is the advantage of using Chebyshev approximation? To understand that, look at the figure 10.9. Or look at equation (10.11). Each polynomial oscillates between the extremal values +1 and −1. Compare this to Legendre polynomials, which also oscillate, and at the ends of the interval ±1 assume values ±1 as well... but *in-between they oscillate a lot less.*

If we assume for a moment that the Chebyshev expansion (10.13) converges rapidly, then we may say approximately, that the error is equal to the first neglected term:

$$f(x)-\widetilde{f}(x)=\sum_{i=I+1}^{\infty}a_iT_i(x)\approx a_{I+1}T_{I+1}(x).$$

Where a_{I+1} is a constant, and $T_{I+1}(x)$ a function that is *uniformly bounded* from above by $+1$ and from below by -1 on the domain $[-1,1]$.

This demonstrates what Chebyshev approximation is useful for: it constitutes *uniform* approximation, where the error is absolutely bounded to the *same* value $|a_{I+1}|$ *all over* the domain $[-1,1]$. For this reason it is used, e.g., in pocket calculators, or numerical libraries, for evaluating standard functions like sines and cosines and logarithms. It is a way to guarantee that always the same number of computed decimals is correct, irrespective of the argument value of the function chosen.

For comparison: if you look at the Legendre polynomials drawn in Fig. 10.8, they are oscillating much less in the middle than towards the end points ±1. This means by the same argument, that the error of approximation will also be larger towards the end points when using Legendre approximation. The weight function $(1 - x^2)^{-1/2}$ which is present in the Chebychev inner product definition (10.12) serves just to "force" the approximation to become more precise there. The "floppy loose ends" of the approximation are suppressed.

10.7 "Inversion-free" interpolation

Inversion-free interpolation works generally in this way, that from the neightbourhood of the prediction point we pick a suitable set of data points and calculate from them a weighted average. The weighting is generally done according to some power of the distance of the data points.

A robust method is formed by taking from the neighbourhood of the prediction point one data point – the nearest point –from every *quadrant*.

10.8 Regridding

If some geophysical field has been given on a regular grid with point spacing Δx, than in the signal contained in this grid are found only the "frequencies" under a certain limit. The shortest possible wavelength, that the grid still is able to represent somewhat reliably, is $2\Delta x$. This is called the Nyquist limit.

10.9 Spatial interpolation, spectral statistics

Literature:

[BG95, s. 141-203]

[MA76, s. 393-426]

[Shi00]

Chapter 11

Least squares collocation

Literature:

[HM67] ss. 251-286.

11.1 Least squares collocation

11.1.1 Stochastic processes

Collocation is a statistical estimation technique which is used to predict a *stochastic process*, of which we have available certain realization values.

Let $\underline{s}(t)$ be a stochastic process having an autocovariance function $C(t_1, t_2)$. Let this process also be *stationary*, i.e., $C(t_1, t_2) = C(t_2 - t_1)$. The argument t generally is time, but it can be almost any parameter, e.g., travelled distance.

Let now be given n observations of this process, $\underline{s}(t_1), \underline{s}(t_2), \ldots, \underline{s}(t_n)$; then the variance matrix of these realizations, or *stochastic quantities*, may be written as follows:

$$\operatorname{Var}(\underline{s}_i) = \begin{bmatrix} C(t_1, t_1) & C(t_2, t_1) & \cdots & C(t_1, t_n) \\ C(t_1, t_2) & C(t_2, t_2) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ C(t_1, t_n) & C(t_2, t_n) & \cdots & C(t_n, t_n) \end{bmatrix}.$$

Let us use for this the symbol C_{ij}. Both for a single element of the matrix, $C_{ij} = C(t_i, t_j)$, and for the whole matrix, $C_{ij} = [C(t_i, t_j), i, j = 1, \ldots, n]$. The symbol $\underline{s}_i$ again means a vector composed of the observations $\underline{s}(t_i), i = 1, \ldots, n$ – or its element $\underline{s}(t_i)$.

Note that, if the function $C(t_2 - t_1)$ is known, we can compute the whole matrix and all of its elements as long as all t_i are known.

Let the problem now be formulated as that of *estimating* the value of the process $\underline{s}$ at the moment (epoch) T. We have available *observations* of the process at times t_i, i.e., $\underline{s}(t_i), i = 1, \ldots, n$.

In the same way as we earlier computed the covariances between $\underline{s}(t_i)$:n and $\underline{s}(t_j)$ (the elements of the variance matrix C_{ij}), we can also compute the covariances between $\underline{s}(T)$and all the $\underline{s}(t_i), i = 1, \ldots, n$. We obtain

$$\operatorname{Cov}(\underline{s}(T), \underline{s}(t_i)) = \begin{bmatrix} C(T, t_1) \\ C(T, t_2) \\ \vdots \\ C(T, t_n) \end{bmatrix}.$$

For this we may again use the notation C_{Tj}.

11.1.2 Signal and noise

It is good to remember here, that the process $\underline{s}(t)$ is a *physical phenomenon* in which we are *interested.* Such a stochastic process is called a *signal.* There exist also stochastic processes that behave in the same way, but in which we are *not* interested. Such stochastic processes we call *noise.*

When we make an observation, the goal of which it is to obtain a value for the quantity $\underline{s}(t_i)$, we obtain in reality a value which is not absolutely exact. We thus obtain

$$\underline{\ell}_i = \underline{s}(t_i) + \underline{n}_i.$$

Here, $\underline{n}_i$ is a stochastic quantity called *observational error* or *noise.* Let its variance be D_{ij}; this is quite a similar matrix as the above C_{ij}. The only difference is, that D describes a phenomenon *in which we have no interest.* Generally it is safe to assume, that the errors in two different observations ℓ_i, ℓ_j do not correlate, in which case D_{ij} is a diagonal matrix.

11.1.3 An estimator and its error variance

Now we construct an estimator

$$\widehat{s}(T) \equiv \sum_j \Lambda_{Tj}\underline{\ell}_j,$$

a linear combination of the available observations $\underline{\ell}_i$. The mission in life of this estimator is to get as close as possible to $\underline{s}(T)$. Siis minimoitava suure on erotus

$$\widehat{s}(T) - \underline{s}(T) = \Lambda_{Tj}\underline{\ell}_j - \underline{s}(T) = \Lambda_{tj}\left(\underline{s}(t_j) + \underline{n}_j\right) - \underline{s}(T).$$

Here we left, for the sake of writing convenience, the summation symbol $\sum$ off (Einstein summation convention).

Let us study the *variance* of this difference, i.e.,

$$\Sigma_{TT} \equiv Var\left(\widehat{s}(T) - \underline{s}(T)\right).$$

We use the *law of proagation of variances*, the notations given above, and our knowledge, that it is highly unlikely that between the observation process $\underline{n}_i$ and the signal $\underline{s}$ there would be any kind of physical connection or *correlation.* So:

$$\Sigma_{TT} = \Lambda_{Tj}\left(C_{jk} + D_{jk}\right)\Lambda_{kT}^T + C_{TT} - \Lambda_{Tj}C_{jT}^T - C_{Ti}\Lambda_{iT}^T. \tag{11.1}$$

11.1.4 The optimal and an alternative estimator

Now choose

$$\Lambda_{Tj} \equiv C_{Ti}\left(C_{ij} + D_{ij}\right)^{-1}.$$

Then, from equation (11.1):

$$\begin{aligned} \Sigma_{TT} &= C_{Ti}\left(C_{ij} + D_{ij}\right)^{-1} C_{jT}^T + C_{TT} - \\ &- C_{Ti}\left(C_{ij} + D_{ij}\right)^{-1} C_{jT}^T - C_{Ti}\left(C_{ij} + D_{ij}\right)^{-1} C_{jT}^T = \\ &= C_{TT} - C_{Ti}\left(C_{ij} + D_{ij}\right)^{-1} C_{jT}^T. \end{aligned} \tag{11.2}$$

Next, we investigate the *alternative choice*

$$\Lambda_{Tj} = C_{Ti}\left(C_{ij} + D_{ij}\right)^{-1} + \delta\Lambda_{Tj}.$$

In this case we obtain

$$\begin{aligned}
\Sigma'_{TT} &= C_{TT} - C_{Ti}\left(C_{ij} + D_{ij}\right)^{-1} C_{jT}^{\mathrm{T}} + \\
&+ \delta\Lambda_{ij} C_{jT}^{T} + C_{Ti}\delta\Lambda_{iT}^{T} - \delta\Lambda_{Tj} C_{jT}^{T} - C_{Ti}\delta\Lambda_{iT}^{T} + \\
&+ \delta\Lambda_{Tj}\left(C_{ij} + D_{ij}\right)\delta\Lambda_{jT}^{T} = \\
&= C_{TT} - C_{Ti}\left(C_{ij} + D_{ij}\right)^{-1} C_{jT}^{T} + \delta\Lambda_{Tj}\left(C_{ij} + D_{ij}\right)\delta\Lambda_{jT}^{T}.
\end{aligned}$$

Here the last term is positive, because the matrices C_{ij} ja D_{ij} are positive definite. In other words, $\Sigma'_{TT} > \Sigma_{TT}$, except if $\delta\Lambda_{Tj} = 0$.

In other words, the already given solution

$$\Lambda_{Tj} = C_{Ti}\left(C_{ij} + D_{ij}\right)^{-1} \Rightarrow \widehat{\mathrm{s}}\left(T\right) = C_{Ti}\left(C_{ij} + D_{ij}\right)^{-1} \underline{\ell}_{j}$$

is truly *optimal* in the sense of least squares (more precisely, in the sense of minimising Σ_{TT}).

11.1.5 Stochastic processes on the Earth's surface

Least squares collocation is much used on the Earth surface for optimally estimating gravity values and values of other functionals of the gravity field.

If the gravity anomaly in the point P_i – location (φ_i, λ_i) – is written as Δg_i, then the covariance between two gravity anomalies is

$$Cov\left(\underline{\Delta g}_i, \underline{\Delta g}_j\right) = C_{ij}.$$

Generally C_{ij} depends *only* on the distance ψ between points P_i, P_j; if this is the case, we speak of an *isotropic process* $\underline{\Delta g}\left(\varphi, \lambda\right)$.

A popular covariance function that is used for gravity anomalies, is Hirvonen's formula:

$$C\left(\psi\right) = \frac{C_0}{1 + \psi^2/\psi_0^2} = \frac{C_0}{1 + s^2/d^2}, \tag{11.3}$$

where $C_0 = C\left(0\right)$ and d are descriptive parameters for the behaviour of the gravity field. C_0 is called the *signal variance*, d the *correlation length.* d is the typical distance over which there is still significant correlation between the gravity anomalies in different points. The metric distance $s \approx R\psi$ and $d \approx R\psi_0$.

If now we have given n points $P_i, i = 1, \ldots, n$, where have been measured gravity values (anomalies) Δg_i, we may, like above, construct a *variance matrix*

$$\begin{aligned}
\mathrm{Var}\left(\underline{\Delta g}_i\right) &= \begin{bmatrix} C_0 & C\left(\psi_{21}\right) & \cdots & C\left(\psi_{n1}\right) \\ C\left(\psi_{12}\right) & C_0 & \cdots & C\left(\psi_{n2}\right) \\ \vdots & \vdots & \ddots & \vdots \\ C\left(\psi_{1n}\right) & C\left(\psi_{2n}\right) & \cdots & C_0 \end{bmatrix} = \\
&= \begin{bmatrix} C_0 & C_{21} & \cdots & C_{n1} \\ C_{12} & C_0 & \cdots & C_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ C_{1n} & C_{2n} & \cdots & C_0 \end{bmatrix} \equiv C_{ij},
\end{aligned}$$

where all $C(\psi_{ij})$ are computed with the aid of the above given formula (11.3).

If we still also compute for the point Q the gravity of which is unknown:

$$\mathrm{Cov}\left(\Delta g_Q, \Delta g_i\right) = \begin{bmatrix} C(\psi_{Q1}) \\ C(\psi_{Q2}) \\ \vdots \\ C(\psi_{Qn}) \end{bmatrix} \equiv C_{Qj},$$

we obtain, in precisely the same way as before, as the *least squares collocation* solution:

$$\widehat{\Delta g_Q} = C_{Qj}\left(C_{jk} + D_{jk}\right)^{-1} \underline{\Delta g}_k,$$

where the $\underline{\Delta g}_k$ are the results of gravity anomaly observations made in the points P_k, $k = 1, \ldots, n$. The matrix D_{jk} again describes the random observation error (imprecision) occurring when making these observations. Generally on may assume, that D_{jk} is a diagonal matrix (the observations are uncorrelated) and furthermore, that $D_{jk} \ll C_{jk}$: The precision of gravity observations is nowadays better than 0.1 mGal, whereas the variability of the gravity field itself is of order 50-100 mGal (i.e., $C_0 \sim 2500 - 10\,000\,\mathrm{mGal}^2$).

11.1.6 The gravity field and applications of collocation

The method of least squares collocation as presented above is applied, e.g., for computing gravity anomalies in a point where no measurements have been made, but where there are measurement points in the vicinity. E.g., if a processing method requires, that gravity anomalies must be available at the nodes of a regular grid, but the really available measurement values refer to freely chosen points – then one ends up having to use the collocation technique.

Collocation may also be used to estimate quantities of different types: e.g., geoid undulations or deflections of the vertical from gravity anomalies. This requires a much more developed theory than the one that was presented here. See, e.g., `http://www.uni-stuttgart.de/gi/research/schriftenreihe/kotsakis.pdf`.

11.2 Kriging

Kriging is a form of least squares collocation, an interpolation technique.

Starting from the above Hirvonen covariance function (11.3), we can compute the variance of the *difference* between two gravity anomalies in points $P$and Q, as follows:

$$\begin{aligned} Var\left\{\underline{\Delta g_P} - \underline{\Delta g_Q}\right\} &= Var\left\{\underline{\Delta g_P}\right\} + Var\left\{\underline{\Delta g_Q}\right\} - 2Cov\left\{\underline{\Delta g_P}, \underline{\Delta g_Q}\right\} = \\ &= 2C_0 - 2\frac{C_0}{1 + (\psi/\psi_0)^2} = \frac{2C_0\,(\psi/\psi_0)^2}{1 + (\psi/\psi_0)^2} = \\ &= \frac{2C_0\psi^2}{\psi^2 + \psi_0^2}. \end{aligned}$$

In the situation where $\psi \ll \psi_0$, we get

$$Var\left\{\underline{\Delta g_P} - \underline{\Delta g_Q}\right\} \approx 2C_0\psi^2/\psi_0^2.$$

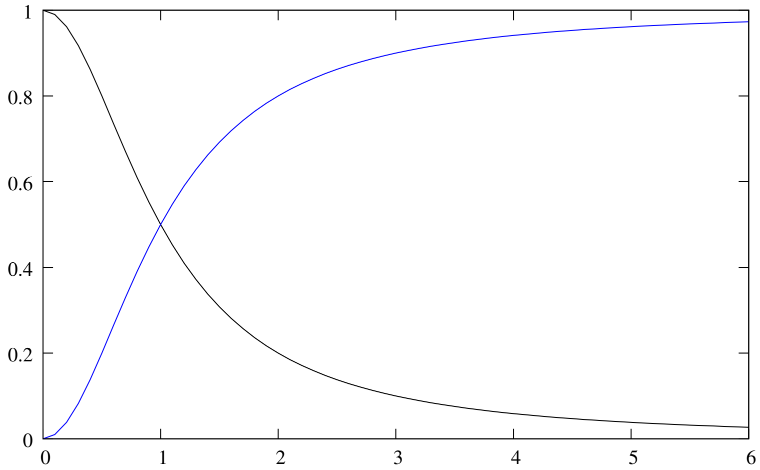


Figure 11.1: Hirvonen's covariance function (for parameter values $C_0 = \psi_0 = 1$) and the associated semi-variance functionja vastaava semivarianssifunktio.

On the other hand, for $\psi \gg \psi_0$, we get

$$Var\left\{\underline{\Delta g_P} - \underline{\Delta g_Q}\right\} \approx 2C_0.$$

We can identify *half* of this expression, $\frac{1}{2}Var\left\{\underline{\Delta g_P} - \underline{\Delta g_Q}\right\}$, with the *semi-variance* of $\underline{\Delta g}$. We also recognize ψ_0, or perhaps a few times ψ_0, as the "sill" at which the semi-variance levels off to the constant value C_0.

For the alternative Markov covariance function, defined as:

$$C(\psi) = C_0 e^{-\psi/\psi_0},$$

we get

$$\begin{aligned}\frac{1}{2}Var\left\{\underline{\Delta g_P} - \underline{\Delta g_Q}\right\} &= C_0 - C_0 e^{-\psi/\psi_0} = \\ &= C_0\left(1 - e^{-\psi/\psi_0}\right).\end{aligned}$$

Now, for $\psi \ll \psi_0$ this becomes $C_0\psi/\psi_0$, while for $\psi \gg \psi_0$ we obtain again C_0. Note the linear behaviour for small ψ, which differs from the quadratic behaviour of the Hirvonen function and is typical for a "random walk" type process .

Kriging is a form of least-squares collocation described within this semi-variance formalism.

11.3 Exercises

11.3.1 Hirvonen's covariance formula

Hirvonen's covariance formula is

$$C(s_{PQ}) = \frac{C_0}{1 + (s_{PQ}/d)^2},$$

where (in the case of Ohio) $C_0 = 337\,\mathrm{mGal}^2$ and $d = 40\,\mathrm{km}$. The formula gives the covariance between the gravity anomalies in two points P and Q:

$$C\left(s_{PQ}\right) = Cov\left(\underline{\Delta g_P}, \underline{\Delta g_Q}\right).$$

s_{PQ} is the inter-point distance.

1. Compute $Var\left(\underline{\Delta g_P}\right)$ and $Var\left(\underline{\Delta g_Q}\right)$ [Hint: remember that according to the definition, $Var\left(x\right) = Cov\left(x, x\right)$].
2. Compute $Cov\left(\underline{\Delta g_P}, \underline{\Delta g_Q}\right)$ if $s_{PQ} = 10$ km.
3. Compute the *correlation*

$$Corr\left(\underline{\Delta g_P}, \underline{\Delta g_Q}\right) \equiv \frac{Cov\left(\underline{\Delta g_P}, \underline{\Delta g_Q}\right)}{\sqrt{Var\left(\underline{\Delta g_p}\right) Var\left(\underline{\Delta g_Q}\right)}}.$$

4. Repeat the computations (sub-problems) 2 and 3 if $s_{PQ} = 80\,\mathrm{km}$.

11.3.2 Prediction of gravity anomalies

Let us have given the measured gravity anomalies for two points 1 and 2, $\underline{\Delta g_1}$ and $\underline{\Delta g_2}$. The distance between the points is 80 km and between them, at the same distance of 40 km from both, is located point P. Compute the gravity anomaly $\underline{\Delta g_P}$ of point P using the *prediction method.* The prediction formula is

$$\widehat{\Delta g_P} = C_{Pi}\left(C_{ij} + D_{ij}\right)^{-1} \underline{\Delta \mathrm{g}_j},$$

where $\underline{\Delta \mathrm{g}_j} = \left[\begin{array}{cc} \underline{\Delta g_1} & \underline{\Delta g_2} \end{array}\right]^T$ is the vector of observed anomalies,

$$C_{ij} = \left[\begin{array}{cc} Var\left(\underline{\Delta g_i}\right) & Cov\left(\underline{\Delta g_i}, \underline{\Delta g_j}\right) \\ Cov\left(\underline{\Delta g_i}, \underline{\Delta g_j}\right) & Var\left(\underline{\Delta g_j}\right) \end{array}\right]$$

is its variance matrix, and $C_{Pi} = \left[\begin{array}{cc} Cov\left(\underline{\Delta g_p}, \underline{\Delta g_1}\right) & Cov\left(\underline{\Delta g_P}, \underline{\Delta g_2}\right) \end{array}\right]$ the covariance matrix between it and $\underline{\Delta g_P}$. D_{ij} is the variance matrix of the *observation process* of $\underline{\Delta g_1}, \underline{\Delta g_2}$.

1. Compute (as a formula) the matrix C_{ij}, assuming HIRVONEN's covariance formula (previous problem) and parameter values.
2. Compute (as a formula) C_{Pi}.
3. Compute (as a formula, but fully written out) $\widehat{\Delta g_P}$. Assume that $D_{ij} = 0$. (
   a) Inverting the

   C_{ij} matrix is possible on paper, but rather use Matlab or similar.)
4. Compute (as a formula) *the variance of prediction* (Note $C_{jP} = C_{Pi}^{\mathrm{T}}$):

$$m_{PP}^2 = C_{PP} - C_{Pi} C_{ij}^{-1} C_{jP}$$

11.3.3 Prediction of gravity (2)

Let us again have the gravity anomalies $\underline{\Delta g_1}$ and $\underline{\Delta g_2}$ measured at points 1 and 2. This time, however, the points 1, 2 and P are lying on a rectangular triangle, so, that the right angle is at point P, and the distances of point P from the points 1 and 2 are, just like before, 40 km. The distance between points 1 and 2 is now only $40\sqrt{2}$ km.

1. Compute C_{ij}, C_{Pi}, $\widehat{\Delta g_P}$ and m^2_{PP}.
2. Compare with the earlier result. Conclusion?

11.3.4 An example of collocation on the time axis

Given is, that the covariance function of the signal function $s(t)$ between two moments t_1 and t_2 is

$$C(t_2 - t_1) = \frac{C_0}{\left(1 + \frac{\|t_2 - t_1\|}{\Delta t}\right)},$$

where the constants are $C_0 = 100\,\mathrm{mGal}$ and $\Delta t = 10\,\mathrm{s}$. Also given are the values of the observation quantity

$$\underline{\ell}_i = \underline{s}(t_i) + \underline{n}_i$$

: $t_1 = 25\,\mathrm{s}$, $\ell_1 = 25\,\mathrm{mGal}$ and $t_2 = 35\,\mathrm{s}$, $\ell_2 = 12\,\mathrm{mGal}$. Compute by least-squares collocation $\widehat{s}(t_3)$, if $t_3 = 50\,\mathrm{s}$. You may assume that $\underline{n}_i = 0$, i.e., the observations are exact.

Answer:

$$\widehat{s}_3 = \begin{bmatrix} C_{31} & C_{32} \end{bmatrix} \begin{bmatrix} C_{11} & C_{21} \\ C_{12} & C_{22} \end{bmatrix}^{-1} \begin{bmatrix} \ell_1 \\ \ell_2 \end{bmatrix},$$

where

$$C_{11} = C_{22} = 100\,\mathrm{mGal},$$

$$C_{12} = C_{21} = \frac{100}{1 + \frac{10}{10}}\mathrm{mGal} = 50\,\mathrm{mGal},$$

$$C_{31} = \frac{100}{1 + \frac{25}{10}}\,\mathrm{mGal} = \frac{1000}{35}\mathrm{mGal} = 28.57\,\mathrm{mGal}$$

$$C_{32} = \frac{100}{1 + \frac{15}{10}}\,\mathrm{mGal} = 40\,\mathrm{mGal},$$

i.e.:

$$\begin{aligned} \widehat{s}_3 &= \begin{bmatrix} 28.57 & 40 \end{bmatrix} \begin{bmatrix} 100 & 50 \\ 50 & 100 \end{bmatrix}^{-1} \begin{bmatrix} 25 \\ 12 \end{bmatrix} = \\ &= \frac{1}{150} \begin{bmatrix} 28.57 & 40 \end{bmatrix} \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 25 \\ 12 \end{bmatrix} = \\ &= \frac{1045.86}{150} = 6.97\,\mathrm{mGal}. \end{aligned}$$

Chapter 12

Various useful analysis techniques

In many practical adjustment and related problems, especially large ones, one is faced with the need to manipulate large systems of equations. Also in geophysical applications, one needs adjustment or analysis techniques to extract information that is significant, while leaving out unimportant information. Also, the methods used should be numerically stable.

In these cases, the following techniques may be useful. We describe them here from a practical viewpoint; they are typically available in rapid prototyping languages such as Matlab or Octave. It is always advisable to use these for proof of principle before coding a production-quality application.

12.1 Computing eigenvalues and eigenvectors

An eigenvalue problem is formulated as

$$[A - \lambda I]\,\mathbf{x} = 0, \tag{12.1}$$

to be determined all values λ_i – eigenvalues – and associated vectors $\mathbf{x}_i$ – eigenvectors – for which this holds. The matrix A is $n \times n$ and the vector $\mathbf{x}$, dimension n.

Determining the λ_i is done formally by taking the *determinant*:

$$\det[A - \lambda I] = 0.$$

This is an n-th degree equation that has n roots – which may well be complex. After they are solved for, they are back substituted, each producing a linear system of equations

$$[A - \lambda_i I]\,\mathbf{x}_i = 0$$

to be solved for $\mathbf{x}_i$.

12.1.1 The symmetric (self-adjoint) case

If A is symmetric, i.e., $A = A^T$, or *self-adjoint*, i.e., $\langle A\mathbf{x} \cdot \mathbf{y}\rangle = \langle \mathbf{x} \cdot A\mathbf{y}\rangle$, or equivalently, in matrix language, $\mathbf{x}^T A^T \mathbf{y} = \mathbf{x}^T A \mathbf{y}$ for all $\mathbf{x}, \mathbf{y}$, we can show that the eigenvalues are real and the corresponding eigenvectors mutually orthogonal. As follows: Say we have λ_i with $\mathbf{x}_i$ and λ_j with $\mathbf{x}_j$. Then from the above

$$\begin{aligned} \lambda_i \mathbf{x}_i &= A\mathbf{x}_i, \\ \lambda_j \mathbf{x}_j &= A\mathbf{x}_j. \end{aligned}$$

Multiply the first equation from the left with $\mathbf{x}_j$,and the second from the right with $\mathbf{x}_i$ The result:

$$\begin{aligned} \lambda_i \langle \mathbf{x}_j \cdot \mathbf{x}_i \rangle &= \langle \mathbf{x}_j \cdot A\mathbf{x}_i \rangle , \\ \lambda_j \langle \mathbf{x}_j \cdot \mathbf{x}_i \rangle &= \langle A\mathbf{x}_j \cdot \mathbf{x}_i \rangle . \end{aligned}$$

Subtract these:

$$(\lambda_i - \lambda_j) \langle \mathbf{x}_j \cdot \mathbf{x}_i \rangle = 0$$

using the self-adjoint nature of A. Now if we have two different eigenvalues $\lambda_i \neq \lambda_j$, we must have

$$\langle \mathbf{x}_j \cdot \mathbf{x}_i \rangle = 0,$$

in other words, $\mathbf{x}_i \perp \mathbf{x}_j$. If $\lambda_i = \lambda_j$, we have a *degeneracy*, but still we will be able to find two vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ spanning the two-dimensional subspace of vectors satrisfying this eigenvalue. The same is several eigenvalues are identical.

We can put all these eigenvectors $\mathbf{x}_i, i = 1, \ldots, n$ into a matrix R as columns:

$$R = \left[\begin{array}{ccccc} \mathbf{x}_1 & \cdots & \mathbf{x}_i & \cdots & \mathbf{x}_n \end{array} \right].$$

Because then

$$\langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle = \mathbf{x}_i^T \mathbf{x}_j = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

we also have

$$R^T R = I,$$

i.e., R is an *orthogonal matrix.*

Because all columns of R satisfy Eq. (12.1), albeit for different values of λ_i, we may write

$$[A - \Lambda] R = 0,$$

where now Λ is the diagonal matrix made up of the eigenvalues: $\Lambda = \operatorname{diag}(\lambda_1, \lambda_2, \cdots, \lambda_n)$. Multiply from the left with R^T:

$$R^T A R = R^T \Lambda R = \Lambda,$$

because of the orthogonality property of R. So now we have found the *rotation matrix that brings A on principal axes*:

$$A = R \Lambda R^T,$$

readily obtained by multiplying from the left with R and from the right with R^T, and observing $RR^T = I$.

12.1.2 The power method

Often we are interested in computing only the biggest eigenvalue of a matrix. In this case a recommendable method is the "power method".

If our matrix is A, we choose a starting vector $\mathbf{x}$ and multiply it repeatedly by A, obtaining

$$A^n \mathbf{x},\ n \to \infty.$$

This will converge to the eigenvector $\mathbf{x}_1$ associated with the largest eigenvalue λ_1, which is then obtained by

$$\lambda_1 = \lim_{n\to\infty} \frac{\|A^{n+1}\mathbf{x}\|}{\|A^n\mathbf{x}\|}.$$

The smallest eigenvalue can be similarly obtained by applying the process to A^{-1} instead.

Note that A^n may become numerically uncomputable for large n: it may overflow (leading to a crash) or underflow (leading to loss of precision). Therefore one should *re-scale* the product $A^n\mathbf{x}$ for every step.

12.2 Singular value decomposition (SVD)

12.2.1 Principle

Singular value decomposition writes an arbitrary matrix A as the product of three matrices:

$$A = USV^T.$$

Here, the matrix S is a matrix that represents "scaling", when the matrices U ja V represent rotations – they are both orthogonal[1]. S on diagonaalimatriisi (More precisely, a diagonal matrix to which have been edded zero columns or rows.).

Note that this works for an arbitrary matrix. A may be rectangular, i.e., the numbers of columns and rows may differ, and there may be a rank defect. If the dimensions of A are $n \times m$, then U is of size $n \times n$ and V of size $m \times m$. The matrix S has the same size as A.

12.2.2 Square matrix

If A is square, then we can compute its deteminant:

$$\det A = \det U \det S \det V,$$

and because the determinant of an orthogonal matrix is always ± 1, we obtain

$$\det A = \pm \det S.$$

The elements of S on the main diagonal are at the same time its eigenvalues:

$$\det S = \prod_{i=1}^{n} \lambda_i,$$

where n is the amount of columns or rows of S, whichever is larger (so: the dimension of the greatest possible square submatrix contained in it).

From this we see, that if some eigenvalue of S vanishes, then we have $\det S = 0$ and therefore necessarily also $\det A = 0$, i.e., A is singular.

Geometrically we can say, that the rotation matrices U and V turn the matrix A "upon principal axes", after which every axis is independent of the others: every axis has its own factor λ_i. More

[1] I.e., $U^TU = UU^T = I$ and the same for V.

precisely, the columns of U are the eigenvectors of AA^T, while those of V are the eigenvectors of A^TA. Both latter matrices are square and symmetric.

We can write $A^TA = \left(USV^T\right)^T USV^T = VS^TU^TUSV^T = VS^TSV^T$, showing the eigenvectors of this matrix to be the columns of V and the eigenvalues, those of the matrix S^TS, which has $m \times m$ elements. They are the squares of the eigenvalues of S itself. Proving that the eigenvectors of AA^T are the columns of U is done similarly. Note that if $m > n$, then AA^T will have $|m - n|$ vanishing eigenvalues, and the same for A^TA if $m < n$.

Calculating the inverse matrix if A is square:

$$\begin{aligned} A^{-1} &= \left(USV^T\right)^{-1} = \left(V^T\right)^{-1} S^{-1}U^{-1} = \\ &= VS^{-1}U^T. \end{aligned}$$

In other words: The SVD of the inverse matrix has the same U and V *as the SVD of the original matrix* – only their roles have been interchanged. Computing the matrix S^{-1} is trivial: every diagonal element is the inverse number of the corresponding diagonal element of S. Of course this presupposes, that all are $\neq 0$!

12.2.3 General matrix

In the general case the form of the matrix S will be one of the following, depending on whether $n < m$ or $n > m$:

$$\begin{bmatrix} \Lambda \\ \emptyset \end{bmatrix} = \left[\begin{array}{ccc} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \\ \hline 0 & \cdots & 0 \\ \vdots & & \vdots \\ 0 & \cdots & 0 \end{array}\right], \quad \begin{bmatrix} \Lambda & \emptyset \end{bmatrix} = \left[\begin{array}{ccc|ccc} \lambda_1 & & & 0 & \cdots & 0 \\ & \ddots & & \vdots & & \vdots \\ & & \lambda_n & 0 & \cdots & 0 \end{array}\right].$$

Then, when we write also U and V out in column vectors:

$$U = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \cdots & \mathbf{u}_n \end{bmatrix}, \quad V = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \cdots & \mathbf{v}_m \end{bmatrix},$$

we may write

$$A = \sum_{i=1}^{\min(m,n)} \lambda_i \mathbf{u}_i \mathbf{v}_i.$$

This expansion explains the name "singular value decomposition": λ_i are the eigenvalues or singular values, organised in descending order of absolute size.

12.2.4 Applications

In applications often the A matrix contains observation values: the element $A_{ij} = A\left(x_i, t_j\right)$,where x_i is the place and t_j the time. In this case the columns of V, $V_{ki} = V_k\left(x_i\right)$ are different patterns of place, and $U_{kj} = U_k\left(t_j\right)$ are correspondingly different time series or spectral components. Every pattern has its own time series, having the same k value and amplitude S_{kk}.

Thus SVD is useful for analysing geophysical phenomena, which depend both on place and on time. The corresponding element of the S matrix, $S_{kk} = \lambda_k$ osana.describes the *strength* of the pattern in question as a part of the total phenomenon.

Example: the ocean tide. A_{ij} is the total value of the tide at place x_i (which thus is two-dimensional, $x_i = (\varphi, \lambda)_i \in \mathbb{R}^2$), at the moment t_j. The solutions are the various tidal constituents, e.g., $k = 1$ semidiurnal lunar tide, $k = 2$ semidiurnal solar tide, etc.

12.2.5 SVD as a compression technique

Only components making a significant contribution to the total signal have $S_{kk} \not\cong 0$. Those elements of S that are in practice zero, can be removed from the S matrix, and correspondingly the meaningless columns in the rotation matrices U and V. In fact, this is an often used *compression technique*: $A_{ij} = A(x_i, t_j)$can be a video clip, and U, V and S may together be considerably smaller than the original A!

We can also look at the expansion

$$A = \sum_{i=1}^{\min(n,m)} \lambda_i \mathbf{u}_i \mathbf{v}_i.$$

In a realistic case, many of the eigenvalues are very close to zero, if not precisely zero. If they are not zero, this will generally be due to the data being used containig noise. By removing all eigenvalues that are absolutely smaller than some suitable limit, we also *filter* the matrix A for this noise. Thus, SVD is also a *data cleansing method.*

12.2.6 Example (1)

We use here the example of a one-dimensional oscillating mirrored cavity. The spatial dimension in $x \in [0, 2\pi)$. The temporal dimension t could extend to infinity, but we limit it here also to $t \in [0, 2\pi)$. We assume the wave funtion to be the sum of two oscillations:

$$f(x,t) = \sin\frac{1}{2}x \sin\frac{1}{2}t + \sin x \sin t.$$

The matrix A describing the wave motion now becomes, choosing 5×5 support points on the square domain:

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -0.500 & 0.707 & 1.500 & 0 \\ 0 & 0.707 & 1.000 & 0.707 & 0 \\ 0 & 1.500 & 0.707 & -0.500 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Here we have retained three decimals (note that the value 0.707 is a truncation of $\frac{1}{2}\sqrt{2}$).

Doing an SVD on this matrix produces

$$U = \begin{bmatrix} 0 & 0 & 0 & 1.00000 & 0 \\ -0.70711 & -0.50000 & 0.50000 & 0 & 0 \\ 0 & -0.70711 & -0.70711 & 0 & 0 \\ 0.70711 & -0.50000 & 0.99992 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.00000 \end{bmatrix},$$

$$S = \begin{bmatrix} 2.00000 & 0 & 0 & 0 & 0 \\ 0 & 1.99985 & 0 & 0 & 0 \\ 0 & 0 & 0.00015 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

and

$$V = \begin{bmatrix} 0 & 0 & 0 & 1.00000 & 0 \\ 0.70711 & -0.50000 & 0.50000 & 0 & 0 \\ 0 & -0.70711 & -0.70711 & 0 & 0 \\ -0.70711 & -0.50000 & 0.99992 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.00000 \end{bmatrix}$$

Inspecting the S matrix, we see two large eigenvalues 2 and 1.99985, followed by the much smaller 0.00015. This smaller value is due to numerical rounding error, as can be readily verified by repeating the process with a larger number of decimals in A.

Retaining only the first two terms, we can compute

$$A = U\widetilde{S}V^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -0.50004 & 0.70705 & 1.49996 & 0 \\ 0 & 0.70705 & 0.99992 & 0.70705 & 0 \\ 0 & 1.49996 & 0.70705 & -0.50004 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

close to the original matrix.

12.2.7 Example (2)

We start from the tide gauge data of the Finnish Institute of Marine Research 1943-1968, see table 12.2.

On this data, written as an array A, we have performed a sigular value decomposition. The singular values, the diagonal elements of the S matrix, are given in table 12.3 in descending order of magnitude.

It can be seen that there are some three dominant values, the rest being uniformly much smaller.

In order to identify what patterns in the data these singular values represent, we have plotted the corresponding *columns of V,* representing spatial patterns over all 13 tide gauges. The length of each of these columns is 13. The plots are in figure 12.1.

We can see that the first singular value, a horizontal line, represents the *common mode* of all the tide gauges: the waters of the Baltic Sea moving up and down together, in almost the same way at each location. This mode represents the *total water volume* of the Baltic, influenced mostly by in- and outflow through the Danish straits, inflow of river water, and precipitation (evaporation being negligible).

Table 12.1: Finnish tide gauges.

	Tide gauge	Latitude	Longitude
1	Hamina	60.56	27.17
2	Helsinki	60.15	24.97
3	Hanko	59.83	22.97
4	Degerby	60.03	20.39
5	Turku	60.41	22.10
6	Rauma	61.13	21.48
7	Mäntyluoto	61.60	21.48
8	Kaskinen	62.39	21.22
9	Vaasa	63.10	21.57
10	Pietarsaari	63.72	22.70
11	Raahe	64.70	24.50
12	Oulu	65.03	25.43
13	Kemi	65.75	24.55

Table 12.2: Tide gauge data from the Finnish coast, years 1943-86. Yearly averages. Years 1949 and 1953 are missing, being incomplete.

```
A = [2083,2060,2035,1994,2030,1972,1972,1970,1964,1938,1969,1996,2011; %1943
     1998,1987,1973,1933,1964,1896,1899,1894,1885,1856,1880,1906,1936; %1944
     1986,1978,1971,1928,1933,1880,1877,1858,1849,1810,1827,1850,1850; %1945
     1952,1935,1922,1882,1893,1849,1848,1839,1819,1799,1827,1869,1867; %1946
     1832,1827,1807,1763,1767,1725,1718,1701,1700,1656,1686,1720,1722; %1947
     2042,2006,1992,1942,1955,1908,1902,1885,1869,1849,1885,1906,1929; %1948
     1977,1972,1955,1914,1920,1872,1866,1854,1820,1810,1829,1862,1862; %1950
     1847,1830,1812,1782,1786,1742,1737,1732,1701,1699,1730,1769,1769; %1951
     1997,1963,1959,1912,1919,1870,1850,1831,1801,1781,1808,1845,1848; %1952
     1933,1912,1888,1835,1847,1795,1784,1779,1742,1712,1759,1801,1794; %1954
     1996,1975,1945,1883,1896,1830,1814,1786,1764,1726,1765,1807,1786; %1955
     1966,1951,1923,1871,1876,1811,1793,1768,1747,1697,1740,1762,1753; %1956
     2008,1985,1953,1887,1900,1840,1822,1795,1768,1725,1777,1812,1799; %1957
     1914,1900,1881,1824,1832,1769,1745,1717,1690,1647,1689,1741,1721; %1958
     1853,1842,1824,1767,1768,1711,1688,1663,1644,1603,1656,1692,1683; %1959
     1772,1778,1770,1721,1723,1669,1635,1608,1573,1530,1572,1605,1590; %1960
     2036,2004,1977,1922,1943,1873,1851,1824,1799,1764,1817,1852,1843; %1961
     2004,1980,1951,1882,1891,1825,1802,1772,1750,1708,1786,1819,1786; %1962
     1860,1829,1804,1738,1750,1683,1661,1626,1603,1569,1610,1662,1637; %1963
     1964,1930,1894,1824,1843,1762,1747,1720,1696,1675,1719,1766,1759; %1964
     1895,1891,1865,1798,1804,1733,1702,1670,1638,1607,1637,1693,1657; %1965
     1857,1847,1825,1761,1778,1709,1684,1655,1627,1597,1639,1712,1670; %1966
     2024,2012,1980,1916,1927,1860,1841,1806,1782,1748,1796,1850,1834; %1967
     1886,1868,1840,1768,1776,1700,1648,1642,1615,1578,1616,1658,1645]; %1968
```

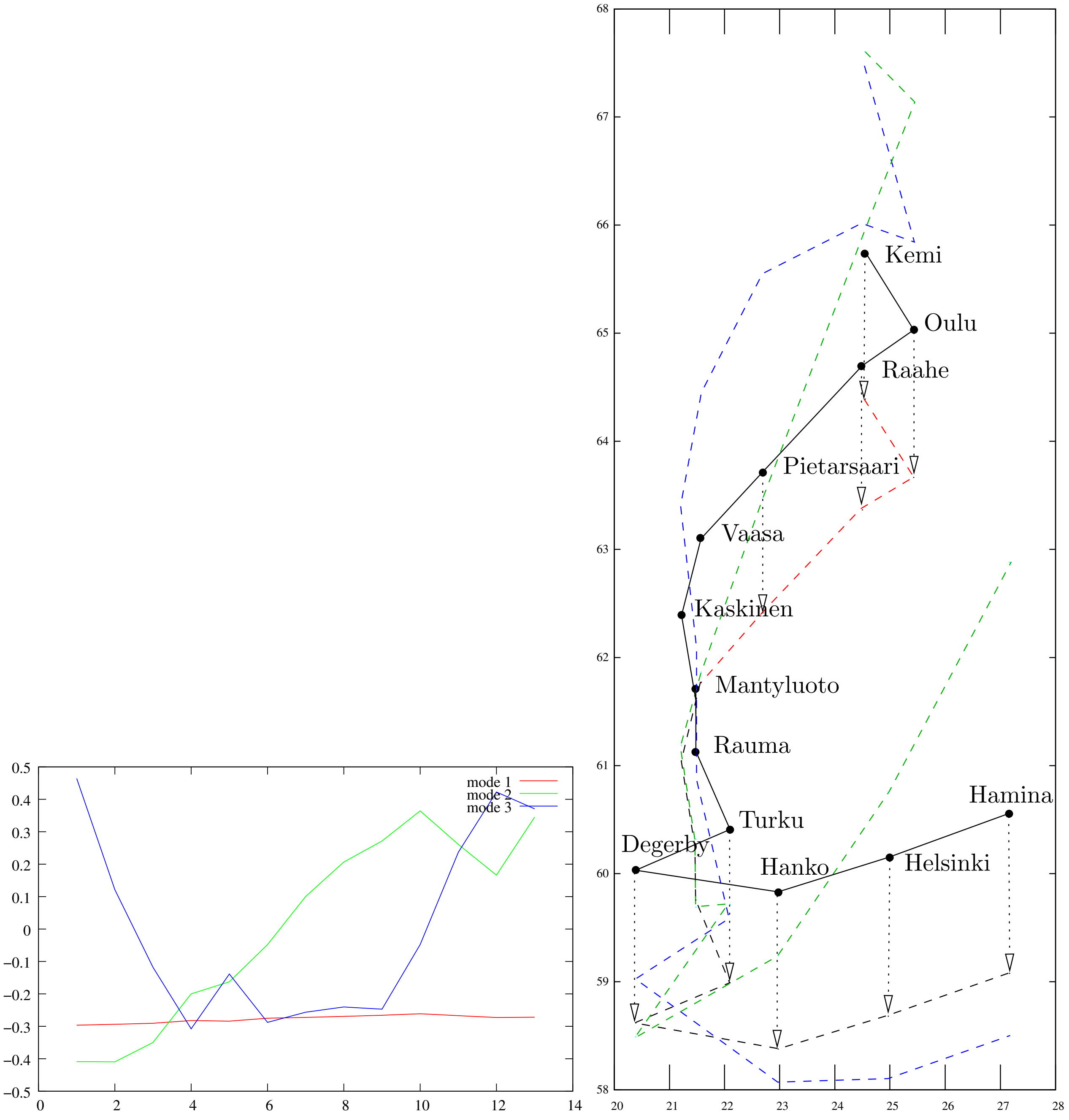


Figure 12.1: The spatial patterns of the first three singular values found by SVD. Left, horizontal scale is tide gauge number; right, geographic plot.

Table 12.3: Singular values from SVD in descending order

1	32163.6050
2	377.3280
3	99.2063
4	52.3408
5	37.3715
6	32.1409
7	29.5212
8	26.3864
9	22.2418
10	19.6933
11	17.5263
12	10.6901
13	9.1831

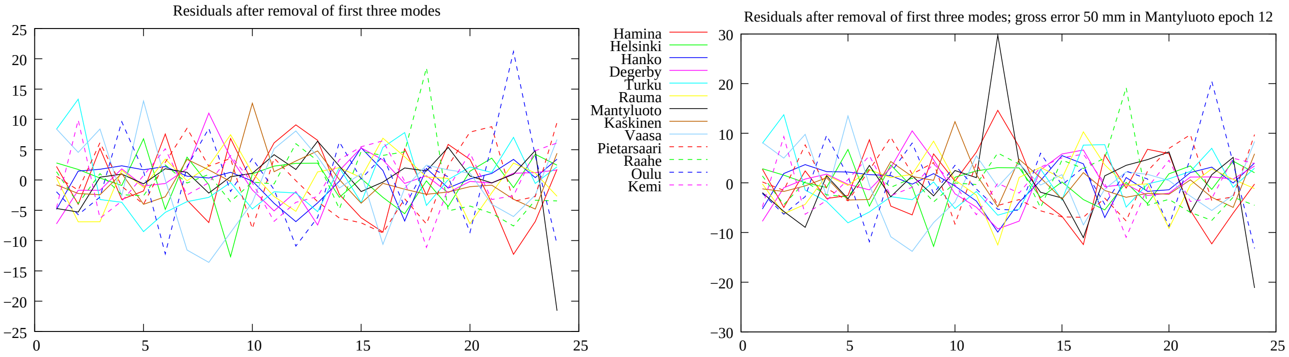


Figure 12.2: Residuals after removal of the first three singular values. Unit: mm.
Right: a gross error of 50 mm was introduced into Mäntyluoto epoch 12.

The second and third modes are very similar, though different-looking. They represent (plane) *tilts* of the water surface, the so-called “bathtub modes”. As was argued in [VKM+88], these three modes describe pretty much all the vertical motion of the water surface of the Baltic.

This hypothesis can be tested. We can retain, in the S matrix, only the first three singular values (diagonal elements), setting the remainder to 0. Then, we compute $A - U\widetilde{S}V^T$, where $\widetilde{S}$ is the thus truncated S matrix. These residuals, which represent all signal unexplained by the three first modes, are given in table 12.2.

The unit in this figure is millimetres. We see that the largest residuals are ±21 mm. Most residuals are within ±10 mm. For comparison, removal of only the first singular value (common mode) leaves many residuals of over ±50 mm, especially in the ends of the Gulfs of Bothnia and Finland.

The right hand side of the figure shows an experiment where a *gross error* was added to one observation. It shows up as an outlier. Nevertheless, one should be careful when using SVD as an outlier detection technique: if there are many singular values included in the reconstruction, one of them may “absorb” the gross error and the whole solution will be deformed by it.

We still plot the common mode as a function of time, computed as

$$\overline{h}\left(t_i\right) U_{i1} S_{11} \frac{1}{m} \sum_{k=1}^{m} V_{k1}.$$

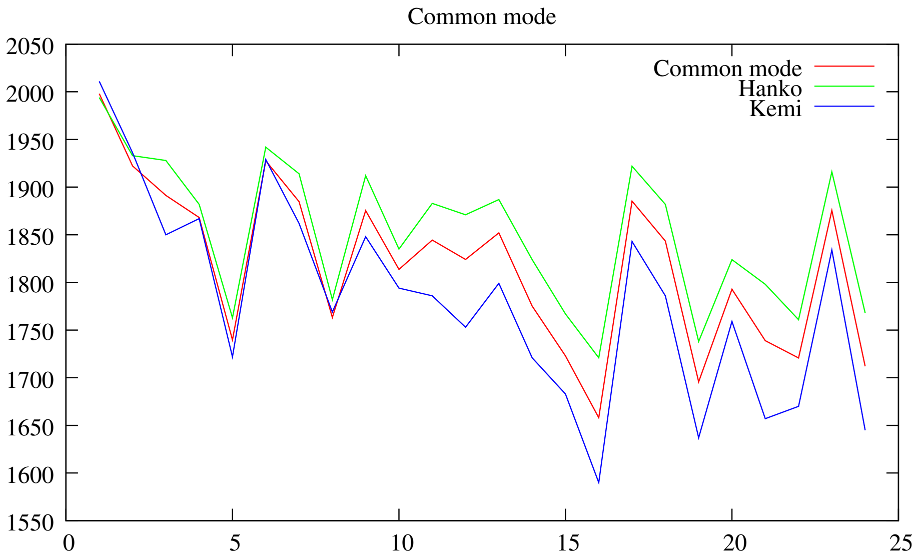


Figure 12.3: Common mode, Hanko and Kemi tide gauge time series

This is plotted in figure 12.3, with, for comparison, the time series of Hanko and Kemi. We see that the general behaviour of the Baltic water masses is captured well.

12.2.8 Additional reading

A good explanation with calculated example:
`http://geosci.uchicago.edu/~gidon/geosci236/mathSvd/svd.html`.

Here the whole course: `http://geosci.uchicago.edu/~gidon/geosci236.html`

12.3 Principal Component Analysis (PCA) or Empirical Orthogonal Functions (EOF)

These methods are closely related to SVD. If we look at an observation matrix $A_{ij} = A(x_i, t_j)$, and consider that it contains, instead of raw observations, deviations from some average value or simple model, then we can generate an (empirical) *variance-covariance matrix* by the following simple operation:

$$Q = A^T A$$

or

$$Q_{ik} = \sum_{j=1}^{n} A(x_i, t_j) A(x_k, t_j).$$

Here we have averaged over time to get spatial covariances. Q will be a square, positive-definite matrix on which we can do SVD, or more simply, we can bring it on principal axes:

$$Q = R \Lambda R^T,$$

where R is an orthogonal rotation matrix and Λ the diagonal matrix of eigenvalues. Every column of R now represents a *spatial pattern*; the corresponding eigenvalue λ_i from the Λ

Table 12.4: Eigenvalues of the PCA variance-covariance matrix Q, descending order

1	98762.0346
2	3398.0554
3	419.6834
4	92.7742
5	57.5507
6	43.4691
7	37.7828
8	27.1429
9	18.6642
10	16.3040
11	11.9502
12	4.7022
13	3.2739

Table 12.5: Residual covariance matrix after removal of the first three singular values. Unit: mm^2

23	-6	-6	-6	-9	-3	0	4	12	8	0	-21	0
-6	16	0	-4	-3	-9	-3	1	11	1	0	0	-3
-6	0	8	4	-1	0	-3	-3	-3	1	-1	3	3
-6	-4	4	11	0	4	-1	-2	-14	0	2	4	3
-9	-3	-1	0	27	0	-8	-4	-4	-1	-8	5	9
-3	-9	0	4	0	15	5	0	-15	-9	2	9	0
0	-3	-3	-1	-8	5	27	-2	-9	-10	1	13	-8
4	1	-3	-2	-4	0	-2	14	-4	0	0	-2	-2
12	11	-3	-14	-4	-15	-9	-4	43	0	2	-18	-2
8	1	1	0	-1	-9	-10	0	0	30	-12	-10	0
0	0	-1	2	-8	2	1	0	2	-12	27	-6	-8
-21	0	3	4	5	9	13	-2	-18	-10	-6	42	-15
0	-3	3	3	9	0	-8	-2	-2	0	-8	-15	23

matrix represents its *strength*. As the spatial patterns are uncorrelated (i.e., the columns of R are orthogonal) the name "empirical orthogonal functions" becomes obvious.

In a practical situation, often only those few eigenvalues significantly different from zero are retained; the others are thrown away, reducing the dimensions of Λ and R (which now becomes rectangular). This will represent the original data to good accuracy, but retaining only a fraction of the original data. So again, we have a *compression method*.

We analyze the same tide gauge data as above for SVD. This time we give the eigenvalues of the variance-covariance matrix, and the residuals of this matrix with respect to the reconstruction using only the three dominant eigenvalues. This is the "energy" that is "unexplained" by these principal components.

Choosing how many principal components to retain is somewhat arbitrary and depends on the intended use.

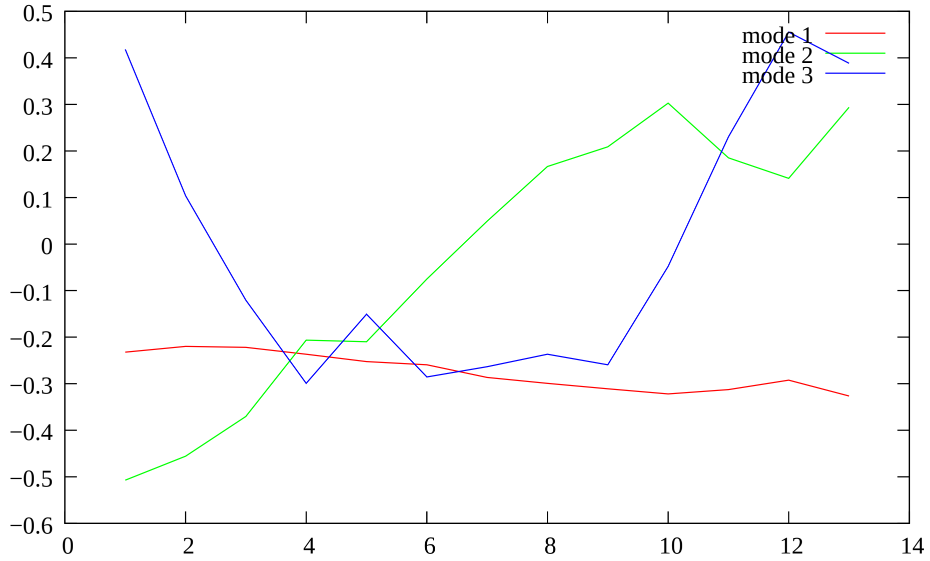


Figure 12.4: The spatial patterns of the first three singular values found by PCA. Horizontal scale is tide gauge number. It is seen that the result is very similar as for SVD.

12.4 The RegEM method

RegEM (Regularized Expectation-Maximation) A method similar to Principal Component Analysis, where data may be arbitrarily missing, and is filled in in an optimal way using the variance-covariance matrix from the non-missing data. Then, the variance-covariance matrix is recomputed with the filled-in data included, carefully taking into account the uncertainty of filling in, and the procedure repeated iteratively until convergence. Cf. `http://web.gps.caltech.edu/~tapio/imputation/`.

12.5 Matrix methods

12.5.1 Cholesky decomposition

This is for a symmetric matrix: it means, for a given matrix A, to compute Γ where

$$A = \Gamma\Gamma^T.$$

12.5.2 LU-decomposition

This means decomposing a given matrix A into an upper and lower triangle matrix:

$$A = LU,$$

or

$$A = LDU,$$

where D is a diagonal matrix.

We can study what happens in the simple case of a 3×3 matrix:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}.$$

We reduce this matrix by subtracting from the second row, a_{21}/a_{11} times the first row, and from the third row, a_{31}/a_{11} times the first row. This is equivalent to multiplying by

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & 0 \\ -\frac{a_{31}}{a_{11}} & 0 & 1 \end{bmatrix}.$$

Next, we reduce by subtracting from the third row, the second row multiplied by $\widetilde{a_{32}}/\widetilde{a_{22}}$, where the tilde indicates that these elements are from the first reduction step: $\widetilde{a_{32}} = a_{32} - \frac{a_{31}}{a_{11}}a_{12}$ and $\widetilde{a_{22}} = a_{22} - \frac{a_{21}}{a_{11}}a_{12}$. This is equivalent to a multiplication by

$$L_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{\widetilde{a_{32}}}{\widetilde{a_{22}}} & 1 \end{bmatrix}.$$

It can be seen that $L = L_2 L_1$ is again a lower triangular matrix:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{a_{21}}{a_{11}} & 1 & 0 \\ -\frac{a_{31}}{a_{11}} + \frac{a_{21}}{a_{11}}\frac{\widetilde{a_{32}}}{\widetilde{a_{22}}} & -\frac{\widetilde{a_{32}}}{\widetilde{a_{22}}} & 1 \end{bmatrix}.$$

The reduced A matrix will look like

$$U = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22} - \frac{a_{21}}{a_{11}}a_{12} & a_{23} - \frac{a_{21}}{a_{11}}a_{13} \\ 0 & 0 & a_{33} - \frac{a_{31}}{a_{11}}a_{13} - \frac{a_{32}-a_{31}a_{13}/a_{11}}{a_{22}-a_{21}a_{12}/a_{11}}\left(a_{23} - \frac{a_{21}}{a_{11}}a_{13}\right) \end{bmatrix}.$$

This looks very complicated, but numerically it is straightforward. It works just as well for larger dimensions than 3[2]. So we now have

$$A = LU.$$

Note that the diagonal elements of the L matrix are 1, while those of the U matrix are not, we can still write

$$U = D\overline{U},$$

where D is a diagonal matrix, and $\overline{U}$ has ones on the main diagonal. Now we have

$$A = LD\overline{U}.$$

LU-decomposition is a way to solve the system of equations

$$A\mathbf{x} = LU\mathbf{x} = \mathbf{b} :$$

one first solves

$$L\mathbf{y} = \mathbf{b}$$

for $\mathbf{y}$, which can be done one element at a time by back substitution; then, we solve

$$U\mathbf{x} = \mathbf{y}$$

similarly for $\mathbf{x}$, but starting from the last element backwards.

For a symmetrix matrix A, we have

$$A = LDU = \Gamma\Gamma^T,$$

where $L = U^T$ and $\Gamma = L\sqrt{D} = \left(\sqrt{D}U\right)^T$. The square root of a diagonal matrix is trivially defined.

[2] Although for a stable reduction one should re-arrange the row and columns in descending order of magnitude – "pivoting".

12.6 Information criteria

Often we wish to know what is the “best” way to model a given body of observational data. What often complicates the answer to that question is, that strongly autocorrelated time series contain significantly less *independent* information than they on the surface, based on the number of data points, appear to contain. This begs the question, how one can judge the amount of meaningful information a data set really contains, i.e., especially in the context of statistical inference, how well one can *model*, i.e., describe, the data using a minimum of free parameters.

This contains two issues:

1. goodness of fit of the model to the data, typically using the metric of sum of squares of residuals of fit;
2. number of free parameters needed.

There are different ways of combining these two aspects in building criteria for *model selection.*

12.6.1 Akaike

The Akaike information criterion is [?]

$$AIC = 2k - 2\ln L,$$

where k is the number of model parameters or unknowns, and L is the value of the likelihood function for the model to be maximized.

In the common case of normally and independently distributed observations, this becomes

$$AIC = 2k + n\left[\ln\frac{2\pi\sum_{i=1}^{n} v_i^2}{n} + 1\right],$$

where v_i are the residuals, and n the number of observations.

Typically the information criterion is used to *intercompare* alternative models for the same data, i.e., the same n. Then, we may drop any constants and write

$$AIC = 2k + n\ln\frac{\sum_{i=1}^{n} v_i^2}{n}.$$

In the more general case of possibly interdependent data, we may write

$$AIC = 2k + n\ln\frac{\mathcal{E}}{n},$$

where we define

$$\mathcal{E} = \mathrm{v}^T Q^{-1}\mathrm{v},$$

the *weighted sum of squared residuals.* Here, Q is the observational variance-covariance matrix, which is χ^2_{n-k} distributed.

12.6.2 Akaike for small samples

In case n is small, or k is not negligibly small compared to n, a small-sample correction is often used

$$AIC_c = AIC + \frac{2k\,(k+1)}{n-k-1}.$$

12.6.3 Bayesian

The alternative SCHWARZ or *Bayesian* information criterion [**?**] is

$$BIC = k \ln n - 2 \ln L,$$

and again in the normally and independently distributed case

$$BIC = k \ln n + n \ln \frac{\sum_{i=1}^{n} v_i^2}{n}.$$

The idea with all of this is, that the parameter should be minimized, leading to as small as possible residuals, but not at the expense of using a large number of free parameters.

12.7 Statistical tricks

12.7.1 Monte Carlo, Resampling, Jackknife, Bootstrap

Monte Carlo simulation techniques have become popular. The idea is the generate a large number of realizations of a physical process studied, by adding synthetic noise of the right properties to a single solution for the process.

Statistical properties of the generated set of solutions or *ensemble* are then studied empirically, as one would with real data. The number of ensemble members can be very large, many tens of thousands; in fact often much larger than the empiricalla available data on the process studied.

A variant of the method generates realizations by, at random, picking elements from a pre-existing, observed set of realizations. The picking process has to be properly random; one technique, *sampling with placeback*, allows the generation of very large ensembles nevertheless sharing the statistical properties of the original set of observatios. This technique is referred to as *bootstrapping*.

A primitive, early version of this technique was, if not invented, made useful by John TUKEY, also known for his role in inventing FFT. It is called the *jackknife*, for its simplicity and wide applicability. It works as follows: if you have a method M to compute useful parameters from observational data, you apply the method *leaving out one observation*, in turn for each of your observations.

Then, you compute the mean and variance for any estimated parameter from these n results: $M_{-1}, M_{-2}, \ldots, M_{-(n-1)}, M_{-n}$. It turns out that the mean is an unbiased estimator of the parameter mean, and the jackknife variance, after scaling, a pretty good estimator of its variance.

In Table 12.6 you find a Matlab script doing linear regression the traditional, ordinary least quares, way, but also computing Jackknife estimates for both intercept and trend parameters and their mean errors. The advantage of the jackknife is that we do not have to have an insight in the mechanism used for estimating our parameters. In this case we have, so we can compare.

12.7.2 Parzen windowing

This is a technique to create a continuous function from discrete values given at realizations, e.g., by a Monte Carlo simulation. It amounts to multiplying every discrete probability value

```
%
% This script computes a linear regression for 30 annual mean global
% temperatures from the GIStemp data set. Units: C, relative 1951-1980.
%
giss = [1978 0.055;
1979 4.04769e-18;
1980 0.19;
1981 0.225833;
1982 0.119167;
1983 0.200833;
1984 0.166667;
1985 0.0575;
1986 0.121667
1987 0.1475;
1988 0.349167;
1989 0.179167;
1990 0.328333;
1991 0.368333;
1992 0.29;
1993 0.0991667;
1994 0.129167;
1995 0.336667;
1996 0.32;
1997 0.339167;
1998 0.528333;
1999 0.434167;
2000 0.343333;
2001 0.365;
2002 0.575833;
2003 0.491667;
2004 0.544167;
2005 0.545;
2006 0.564167;
2007 0.610833];
dataspan = [1:30];
n = size(dataspan,2)
% Loop through the data items, leaving one out in turn
% NOTE: the last looping (n+1) produces the LSQ solution for ALL data
for i = 1:n+1
  dataspan = [1:i-1 i+1:n];
  times = giss(dataspan,1) - 1978;
  ell = giss(dataspan,2);
  o = ones(size(dataspan))';
  % Design matrix:
  A = [o times];
  % least squares:
  N = inv(A'*A);
  RHS = A'*ell;
  sol = N*RHS;
  v = ell-A*sol;
  % Variance of unit weight, posterior est.
  sigma2 = (v'*v)/(n-1);
  N = sigma2 * N;
  % Store jackknife solutions
  if i < n+1
    J1(i) = sol(1);
    J2(i) = sol(2);
  end
end
fprintf(1,'\nLeast Squares solution:\n');
fprintf(1,'----------\n');
fprintf(1,'Intercept:  %10.5f +/- %10.5f\n', sol(1), sqrt(N(1,1)));
fprintf(1,'Trend:      %10.5f +/- %10.5f\n', sol(2), sqrt(N(2,2)));
Jmean1 = mean(J1);
dJ = J1 - Jmean1;
Jvar1 = ((n-1)/n) * (dJ*dJ');
Jmean2 = mean(J2);
dJ = J2 - Jmean2;
Jvar2 = ((n-1)/n) * (dJ*dJ');
fprintf(1,'\nJackknife solution:\n');
fprintf(1,'Intercept:  %10.5f +/- %10.5f\n', Jmean1, sqrt(Jvar1));
fprintf(1,'Trend:      %10.5f +/- %10.5f\n', Jmean2, sqrt(Jvar2));
fprintf(1,'\nCompare this with the above.\n');
```

Table 12.6: Jackknife vs. traditional least squares

with a bell-shaped distribution function centred at its location, and then summing these up. If the values given are $(x_i, y(x_i)), i = 1, \ldots, n$, the constructed function may look like

$$\widetilde{y}(x) = \sum_{i=1}^{n} y(x_i) \left(\Delta\sqrt{2\pi}\right)^{-1} \exp\left(-\frac{1}{2}\frac{(x - x_i)^2}{\Delta^2}\right),$$

for Gaussian base functions. The width Δ of the base function must be chosen judiciously, which is a problem all of its own.

12.7.3 Bayesian inference

This is a very broad subject. To get started, an example from the Internet.

This example is from [Yud03]:

- 1.0% of women contract breast cancer.
- 80% of women with breast cancer test positive.
- 9.6% of women without breast cancer also test positive.

What is the probability that a woman who tested positive, has breast cancer?

In this case, Bayesian analysis looks at *frequencies*[3]. Say, we have 1000 women. Let the parameter be P, having two possible values, $P = 0$ no cancer, $P = 1$ cancer. Let the observation be the test Q, 0 meaning testing negative, 1 testing positive. Then we can draw the following PQ diagram of frequencies:

	$Q = 0$	$Q = 1$
$P = 0$	895	95
$P = 1$	2	8

From this we see that of the 95+8 women who test positive, 8, or slightly under 8%, actually have breast cancer.

We can abstract this from the size of the population by dividing by it, yielding percentages:

	$Q = 0$	$Q = 1$
$P = 0$	89.5	9.5
$P = 1$	0.2	0.8

We can now define the following probabilities: $p(P)$ the probability of having $(p(P = 1) = 1\%)$ or not having $(p(P = 0) = 99\%)$ cancer $p(Q)$ the probability of testing positive $(p(Q = 1) = 10.3\%)$ or negative $(p(Q = 0) = 89.7\%)$. $p(Q|P)$ conditional probability of Q given P: e.g., 9.5%/(89.5%+9.5%) = 9.6% for getting $Q = 1$ if $P = 0$, i.e., getting a false positive. $p(P|Q)$ conditional probability of P given Q: e.g. 0.8%/(0.8%+9.5%) = 7.7% for getting $P = 1$ when $Q = 1$, i.e. having cancer if testing positive.

Now, Bayes' theorem says (and this is easy to prove in this case where we have complete frequency population data):

$p(P|Q) = {}^{p(Q|P)p(P)}/_{p(Q)}$.

[3] In the literature, you will often see Bayesian opposed to "frequentist" approaches. There is a substantial body of underlying philosophy connected with this apparent contradiction.

The interesting case arises where we don't have access to such complete data. E.g., we have observations Q and knowledge of which distribution of observations will be produced by any given parameter value P; and we want to know, or infer, what the probability distribution is of P given our observations Q. This is called *reverse inference*, and the above theorem allows us to do that... provided we have access to the distribution $p(P)$, the so-called *prior* distribution.

Bibliography

[Aho01] Joel Ahola. Nuottavaaran siirroksen deformaatiotutkimus GPS-mittausten avulla. Master's thesis, Teknillinen korkeakoulu, maanmittausosasto, Espoo, helmikuu 2001.

[Ano03] Anon. JHS154. ETRS89 -järjestelmään liittyvät karttaprojektiot, tasokoordinaatistot ja karttalehtijako. website, Maanmittauslaitos, 2003. URL:

`http://www.intermin.fi/intermin/hankkeet/juhta/home.nsf/pages/2A84782CF8BE2527C2256D580047282A/$file/JHS154.pdf.`

[Baa73] Willem Baarda. *S-transformations and criterion matrices*, volume 5 no. 1 of *New*. Netherlands Geodetic Commission, Delft, 1973.

[BG95] T.C. Bailey and A.C. Gatrell. *Interactive Spatial Data Analysis*. Longman Scientific & Technical, England, 1995.

[Coo87] M. A. R. Cooper. *Control surveys in civil engineering*. Collins, Department of Civil Engineering, The City University, London, 1987.

[FIG98] FIG Commission 6. Engineering Surveys. Technical report, FIG, 1998.

[HM67] Weikko A. Heiskanen and Helmut Moritz. *Physical Geodesy*. W.H. Freeman and Company, San Francisco, 1967.

[Kal60] R.E. Kalman. A new approach to linear filtering and prediction problems. *Trans. ASME, J. Basic Eng., Series 82D*, pages 35–45, 1960.

[Kal98a] Ulla Kallio. Suora, ympyrä, pallo ja taso. In Jaakko Santala, editor, *Moniulotteinen mallintaminen ja visualisointi*, pages 372–399. Teknillinen korkeakoulu, maanmittausosasto, Otaniemi, 1998. Mittaus- ja kartoitustekniikan tutkijankoulutuskurssi (1. jakso), 2-6 marraskuuta 1998.

[Kal98b] Ulla Kallio. *Tasoituslasku*. Number 587. Otatieto, 1998.

[KB61] R.E. Kalman and R.S. Bucy. New results in linear filtering and prediction theory. *Trans. ASME, J. Basic Eng., Series 83D*, pages 95–108, 1961.

[Kra83] J. Edward Krakiwsky, editor. *Papers for the CIS adjustment and Analysis seminar*. The Canadian Institute of Surveying, 1983.

[Kär93] Esa Kärkäs. 3d approach in theodolite observation analysis. In Jaakko Santala, editor, *Esitelmien tekstit*, pages 66–75. Teknillinen korkeakoulu, maanmittausosasto, Otaniemi, 1993. Mittaus- ja kartoitustekniikan tutkijankoulutusseminaari, 6-10 syyskuuta 1993.

[Lan89] Ulla Lankinen. Vapaan pisteverkon tasoituksesta ja analysoinnista. Master's thesis, Teknillinen korkeakoulu, rakennus- ja maanmittaustekniikan osasto, Espoo, joulukuu 1989.

[Lei95] Alfred Leick. *GPS Satellite Surveying*. Wiley, 1995.

[MA76] Edward M. Mikhail and F. Ackermann. *Observations and Least Squares*. Harper & Row, 1976.

[Nor99a] Tommi Norri. Optinen kolmiulotteinen koordinaattimittaus. In Jaakko Santala and Katri Koistinen, editors, *Mittausteknologiat ja mittausprosessit*, pages 66–75. Teknillinen korkeakoulu, maanmittausosasto, Otaniemi, 1999. Mittaus- ja kartoitustekniikan tutkijankoulutuskurssi, 25-26 marraskuuta 1999.

[Nor99b] Tommi Norri. Tarkka kolmiulotteinen teodoliittimittausjärjestelmä ja sen tarkkuustutkimus. Master's thesis, Teknillinen korkeakoulu, maanmittausosasto, Espoo, toukokuu 1999.

[Pap65] Athanasios Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill Book Company, New York, 1965.

[Sal95] Hannu Salmenperä. Valvonta- ja muodonmuutosmittaukset. Technical report, Tampereen teknillinen korkeakoulu, Rakennustekniikan osasto, Geoinformatiikka, Tampere, 1995.

[San81] Jaakko Santala. *Teodoliitin jaotusvirheiden määrittäminen multikollimaattorihavainnoista.* PhD thesis, Teknillinen korkeakoulu, maanmittausosasto, 1981. lisensiaattityö.

[SB97] Gilbert Strang and Kai Borre. *Linear Algebra, Geodesy, and GPS.* Wellesley - Cambridge Press, 1997.

[Shi00] Syed Abdul Rahman Shibili. Afaq on geostatistics. http://www.ai-geostatis.org/, 2000.

[Str82] Govert Strang van Hees. Variance- Covariance Transformation of Geodetic Networks. *manuscripta geodaetica*, 7:1–18, 1982.

[VK86] Peter Vaníček and Edward Krakiwsky. *Geodesy – The Concepts.* Elsevier Science Publishers, Amsterdam, 1986.

[VKM$^+$88] Martin Vermeer, Juhani Kakkuri, Pentti Mälkki, Kimmo Kahma, Matti Leppäranta, and Hanna Boman. Land uplift and sea level variability spectrum using fully measured monthly means of tide gauge readings. In *Finnish Marine Research*, pages 3–75. Finnish Institute of Marine Research, Helsinki, 1988.

[Yud03] Eliezer Yudkowsky. An Intuitive Explanation of Bayesian Reasoning, 2003. URL: `http://yudkowsky.net/bayes/bayes.html`. Accessed July 2, 2008.

Appendix A

Useful matric equations

Ensimmäinen kaava:

$$\begin{aligned}(A+B)^{-1} &= \left[A\left(I+A^{-1}B\right)\right]^{-1} = \left[A\left(B^{-1}+A^{-1}\right)B\right]^{-1} = \\ &= B^{-1}\left[A^{-1}+B^{-1}\right]^{-1}A^{-1}.\end{aligned}$$

Sijoitetaan

$$B^{-1} = \left(A^{-1}+B^{-1}\right) - A^{-1}$$

ja saadaan

$$\begin{aligned}(A+B)^{-1} &= \left[\left(A^{-1}+B^{-1}\right) - A^{-1}\right]\left[A^{-1}+B^{-1}\right]^{-1}A^{-1} = \\ &= A^{-1} - A^{-1}\left[A^{-1}+B^{-1}\right]^{-1}A^{-1}.\end{aligned}$$

Toinen kaava:

Kirjoitetaan

$$B = UCV.$$

Tutkitaan seuraava partitioitu yhtälö:

$$\begin{bmatrix} A & U \\ V & -C^{-1}\end{bmatrix}\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22}\end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I\end{bmatrix}.$$

Tätä voidaan kirjoittaa neljäksi matriisiyhtälöryhmäksi:

$$\begin{aligned} AD_{11}+UD_{21} &= I, & \text{(A.1)}\\ AD_{12}+UD_{22} &= 0, \\ VD_{11}-C^{-1}D_{21} &= 0, & \text{(A.2)}\\ VD_{12}-C^{-1}D_{22} &= \mathrm{I}. \end{aligned}$$

Näistä neljästä yhtälöstä vain *ensimmäistä* ja *kolmatta* tarvitaan jarkossa.

Ynnätään yhtälö A.2 kerrottuna UC:n kanssa yhtälöön A.1:

$$(A+UCV)\,D_{11} = I \Rightarrow D_{11} = (A+UCV)^{-1}. \tag{A.3}$$

Vähennetään yhtälö A.1 kerrottuna VA^{-1}:n kanssa yhtälöstä A.2:

$$\left(C^{-1}-VA^{-1}U\right)D_{21} = -VA^{-1} \Rightarrow D_{21} = -\left(C^{-1}-VA^{-1}U\right)^{-1}VA^{-1}.$$

Sijoitetaan takaisin yhtälöön A.1:

$$AD_{11} - U\left(C^{-1} - VA^{-1}U\right)^{-1} VA^{-1} = I \Rightarrow D_{11} = A^{-1} + A^{-1}U\left(C^{-1} - VA^{-1}U\right)^{-1} VA^{-1}. \tag{A.4}$$

Nyt meillä on kaksi eri ilmaisua alamatriisille D_{11}, jotka ovat oltavia *identtisiä*. Näin saadaan:

$$(A + UCV)^{-1} = A^{-1} + A^{-1}U\left(C^{-1} - VA^{-1}U\right)^{-1} VA^{-1}, \tag{A.5}$$

Woodburyn matriisikaava (K. Inkilä, henk. tied.).

Appendix B

The Gauss reduction scheme

Jo K.F. Gaußin ajoista on peräisin perin yksinkertainen ja kätevä reduktiomenetelmä lineaarisen yhtälöryhmän ratkaisun laskemiseksi.

Olkoon ratkaistava yhtälöryhmä seuraava:

$$AX = B.$$

Sen ratkaisu on ilmeisesti

$$X = A^{-1}B.$$

Kirjoitetaan auki:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1k} \\ x_{21} & x_{22} & \cdots & x_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mk} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1k} \\ b_{21} & x_{22} & \cdots & b_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \cdots & b_{nk} \end{bmatrix}.$$

Tämän yhtälöryhmän ratkaisumatriisi X ei muutu, vaikka

1. Sekä A:n että B:n tietty rivi kerrotaan vakion c kanssa, tai
2. Sekä A.n että B:n tietty rivi ynnätään vastaavaan toiseen sekä A:n että B:n riviin.

Jätetään nyt matriisi pois ja käytetään *notaatio*:

$$\left[\begin{array}{cccc|cccc} a_{11} & a_{12} & \cdots & a_{1m} & b_{11} & b_{12} & \cdots & b_{1k} \\ a_{21} & a_{22} & \cdots & a_{2m} & b_{21} & b_{22} & \cdots & b_{2k} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} & b_{n1} & b_{n2} & \cdots & b_{nk} \end{array}\right]$$

Tässä notaatiossa voidaan nyt, samalla tavalla kuin yllä luetteloitiin, kertoa rivejä vakion kanssa tai ynnätä rivi toiseen riviin, elementti kerrallaan.

Menetellään seuraavalla tavalla:

1. Kerro ensimmäinen rivi kertoimella a_{11}^{-1}.
2. Vähennä se kaikilta muilta riviltä i kertoimella a_{i1}:lla kerrottuna (*Gauß-reduktio*).

   Lopputulos:

$$\left[\begin{array}{cccc|cccc} 1 & a_{11}^{-1}a_{12} & \cdots & a_{11}^{-1}a_{1m} & a_{11}^{-1}b_{11} & a_{11}^{-1}b_{12} & \cdots & a_{11}^{-1}b_{1k} \\ 0 & a_{22} - a_{21}a_{11}^{-1}a_{12} & \cdots & a_{2m} - a_{21}a_{11}^{-1}a_{1m} & b_{21} - a_{21}a_{11}^{-1}b_{11} & b_{22} - a_{21}a_{11}^{-1}b_{12} & \cdots & b_{2k} - a_{21}a_{11}^{-1}b_{12} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2} - a_{n1}a_{11}^{-1}a_{12} & \cdots & a_{nm} - a_{n1}a_{11}^{-1}a_{1m} & b_{n1} - a_{n1}a_{11}^{-1}b_{11} & b_{n2} - a_{n1}a_{11}^{-1}b_{12} & \cdots & b_{nk} - a_{n1}a_{11}^{-1}b_{1k} \end{array}\right]$$

Kirjoitetaan symbolisesti

$$\left[\begin{array}{cccc|cccc} 1 & a_{12}^{(1)} & \cdots & a_{1m}^{(1)} & b_{11}^{(1)} & b_{12}^{(1)} & \cdots & b_{1k}^{(1)} \\ 0 & a_{22}^{(1)} & \cdots & a_{2m}^{(1)} & b_{21}^{(1)} & b_{22}^{(1)} & \cdots & b_{2k}^{(1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nm}^{(1)} & b_{n1}^{(1)} & b_{n2}^{(2)} & \cdots & b_{nk}^{(1)} \end{array}\right]$$

Elementti (1) kutsutaan tämän operaation *pivotiksi.*

3. Toista operaatiot 1,2 elementillä $a_{22}^{(1)}$. Lopputulos on tämän näköinen:

$$\left[\begin{array}{ccccc|ccccc} 1 & 0 & a_{13}^{(2)} & \cdots & a_{1m}^{(2)} & b_{11}^{(2)} & b_{12}^{(2)} & b_{13}^{(2)} & \cdots & b_{1k}^{(2)} \\ 0 & 1 & a_{23}^{(2)} & \cdots & a_{2m}^{(2)} & b_{21}^{(2)} & b_{22}^{(2)} & b_{23}^{(2)} & \cdots & b_{2k}^{(2)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3m}^{(2)} & b_{31}^{(2)} & b_{32}^{(2)} & b_{33}^{(2)} & \cdots & b_{3k}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nm}^{(2)} & b_{n1}^{(1)} & b_{n2}^{(2)} & b_{n3}^{(2)} & \cdots & b_{nk}^{(2)} \end{array}\right]$$

   Huomaa yksikkömatriisin ilmaantuminen vasemmalle ylänurkalle.

4. Yllä olevaa reduktiomenetelmää voidaan suorittaa, paitsi rivi kerrallaan, myös *riviblokki* kerrallaan. Partitioidaan yhtälö:

$$\left[\begin{array}{cc} A_{11} & A_{12} \\ A_{21} & A_{22} \end{array}\right]\left[\begin{array}{cc} X_{11} & X_{12} \\ X_{21} & X_{22} \end{array}\right]=\left[\begin{array}{cc} B_{11} & B_{12} \\ B_{21} & B_{22} \end{array}\right].$$

   Osittainen reduktio tässä tapauksessa antaa

$$\left[\begin{array}{cc|cc} A_{11} & A_{12} & B_{11} & B_{12} \\ A_{21} & A_{22} & B_{21} & B_{22} \end{array}\right] \Rightarrow \left[\begin{array}{cc|cc} I & A_{11}^{-1}A_{12} & A_{11}^{-1}B_{11} & A_{11}^{-1}B_{12} \\ 0 & A_{22}-A_{21}A_{11}^{-1}A_{12} & B_{21}-A_{21}A_{11}^{-1}B_{11} & B_{22}-A_{21}A_{11}^{-1}B_{12} \end{array}\right]$$

   Tästä näkyy, että, jos haluat laskea matriisi-ilmaisua $P-UQ^{-1}V$ — usein esiintyvä tarve — voit laittaa ne neljä osa-matriisiä vain laskentataulukkoon seuraavalla tavalla:

$$\begin{array}{c|c} Q & V \\ \hline U & P \end{array}$$

   ...ja *redukoida* tämä taulukko rivi kerrallaan, kunnes osamatriisin Q paikkaan ilmaantuu yksikkömatriisi:

$$\begin{array}{c|c} I & Q^{-1}V \\ \hline 0 & P-UQ^{-1}V \end{array}$$

   Nyt voidaan “poimia” osamatriisin P paikasta ilmaisu $P-UQ^{-1}V$.

5. *Sovellusesimerkki:* parametrinen tasoitustehtävän ratkaisu on

$$\begin{aligned} \widehat{\mathrm{x}} &= \left[A^TQ_{\ell\ell}^{-1}A\right]^{-1}A^TQ_{\ell\ell}^{-1}\underline{\ell}, \\ Q_{xx} &= \left[A^TQ_{\ell\ell}^{-1}A\right]^{-1}. \end{aligned}$$

   Muodostetaan seuraava taulukkko:

$$\begin{array}{c|c|c} Q_{\ell\ell} & A & \underline{\ell} \\ \hline A^T & 0 & 0 \end{array}$$

reduktio antaa:

$$\begin{array}{c|c|c} I & Q_{\ell\ell}^{-1}A & Q_{\ell\ell}^{-1}\underline{\ell} \\ \hline 0 & -A^TQ_{\ell\ell}^{-1}A & -A^TQ_{\ell\ell}^{-1}\underline{\ell} \end{array}$$

Poistetaan tästä diagrammasta ensimmäiset rivi ja sarake ja lisää sarake oikealle:

$$-A^TQ_{\ell\ell}^{-1}A \;\Big|\; -A^TQ_{\ell\ell}^{-1}\underline{\ell} \;\Big|\; -I$$

Jatketaan reduktio:

$$I \;\Big|\; \left[A^TQ_{\ell\ell}^{-1}A\right]^{-1} A^TQ_{\ell\ell}^{-1}\underline{\ell} \;\Big|\; \left[A^TQ_{\ell\ell}^{-1}A\right]^{-1}$$

Kuten näkyy, on sekä *ratkaisu* $\widehat{\mathrm{x}}$ että sen *varianssimatriisi* Q_{xx} valmiit poimittaviksi!

Tätä lähestymistapaa voidaan helposti laajentaa esim. jäännösvirheiden ja painoyksikön keskivirheen laskemiseen. Myös pienimmän neliösumman kollokaation ja Kalman-suodattimen kaavat voidaan laskea tällä tavoin, joka on helposti implementoitavissa tietokoneellekin.