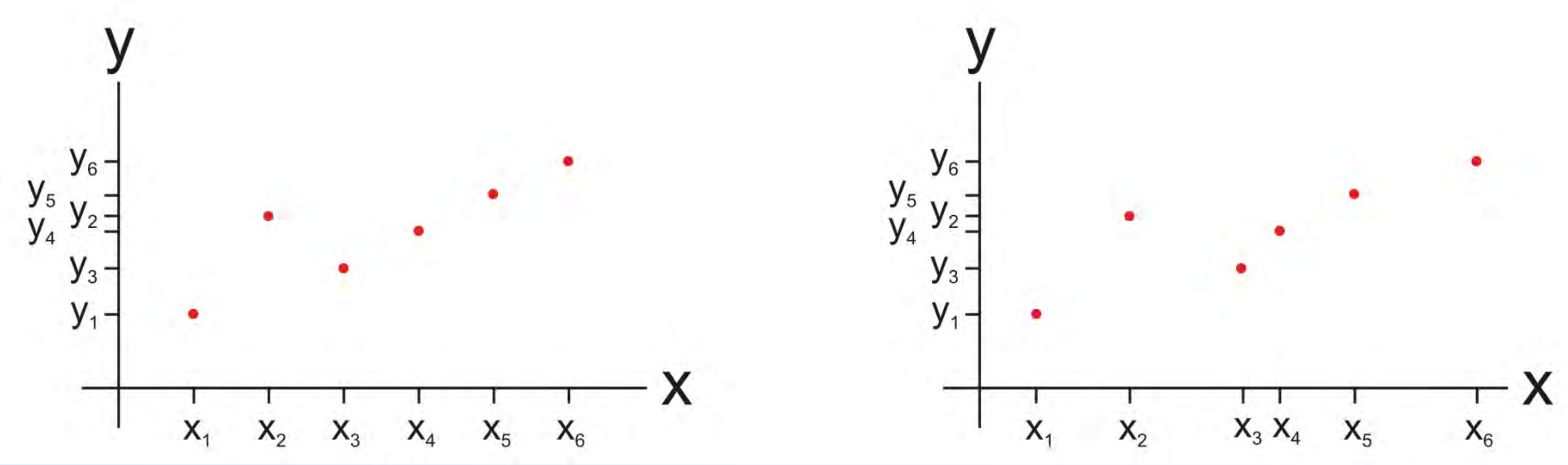
A-model: Linear Regression

The subject of linear regression is to find, e.g., constants $\hat{a}$ (axis intercept) and $\hat{b}$ (slope) as estimates for the model parameter a and b, given supporting points x_i on the abscissa (input signal) and measured observations y_i on the ordinate (output signal).

In other words: How can we fit a line to the set of values x_i and y_i ?



regular x spacing

irregular x spacing

NE

→ A-model: Linear Regression

A-model: Linear Regression

How do we funnel linear regression model into A-model of adjustment theory, $y = Ax + e$? What are y, A and x (in the A-model) as compared to x_i, y_i, a and b in the regression model ?

Answer: output signal = observations $y_i \rightarrow$ observation vector y
unknown parameters a, b $\rightarrow$ vector of unknowns x
input signal $x_i \rightarrow$ design matrix A

Example: straight line $y_i = a + bx_i = 1a + x_i b$

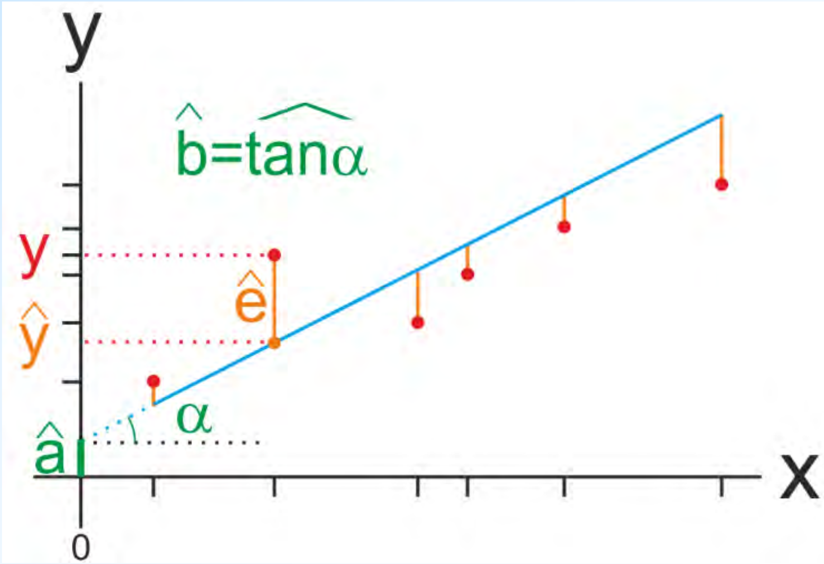
$$\underset{m\times 1}{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}, \quad \underset{n\times 1}{x} = \begin{bmatrix} a \\ b \end{bmatrix}, \quad \underset{m\times n}{A} = \begin{bmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_m \end{bmatrix}$$

$\Rightarrow \hat{x} = [\hat{a} \quad \hat{b}]' = (A'A)^{-1}A'y$

$\Rightarrow \hat{y} = A\hat{x}$ "adjusted/estimated observations"

$\Rightarrow \hat{e} = y - \hat{y}$ "estimated residuals"

$\Rightarrow \hat{e}'\hat{e}$ "square sum of residuals" ('quality' measure)



→ A-model: Linear Regression

A-model: Linear Regression

Example:

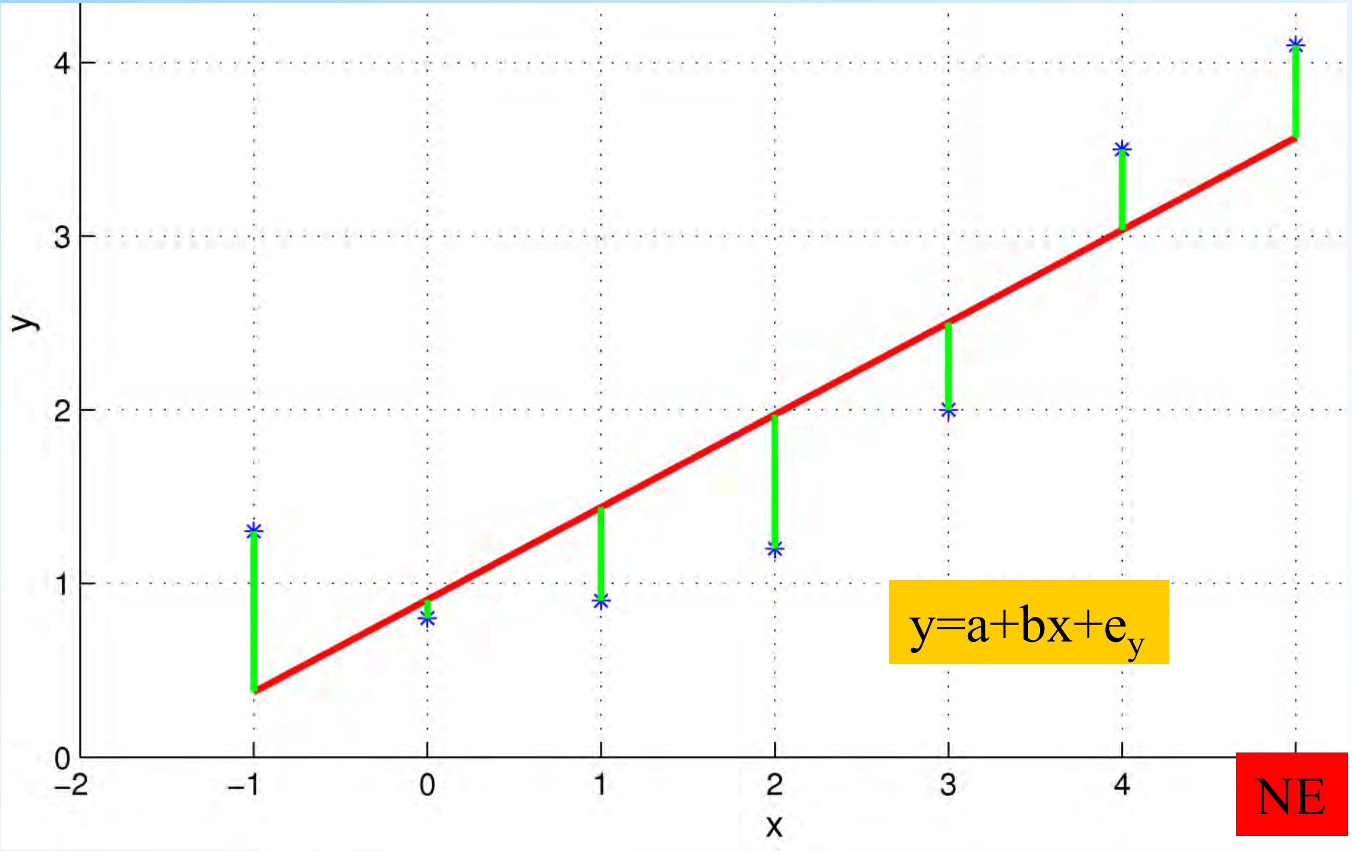
i	1	2	3	4	5	6	7
x_i [m]	−1	0	1	2	3	4	5
y_i [m]	1.3	0.8	0.9	1.2	2.0	3.5	4.1

$$\Rightarrow \quad \hat{a} = 0.907\,[\mathrm{m}] \qquad \hat{b} = 0.532\,[-]$$

$\hat{y}_i$ [m]	$\hat{e}_{y_i}$ [m]
0.375	0.925
0.907	−0.107
1.439	−0.539
1.971	−0.771
2.504	−0.504
3.036	0.464
3.568	0.532

$$\Downarrow$$

$$\hat{e}'\hat{e} = 2.505\,[\mathrm{m}^2]$$



→ A-model: Linear Regression

A-model: Linear Regression

Example:

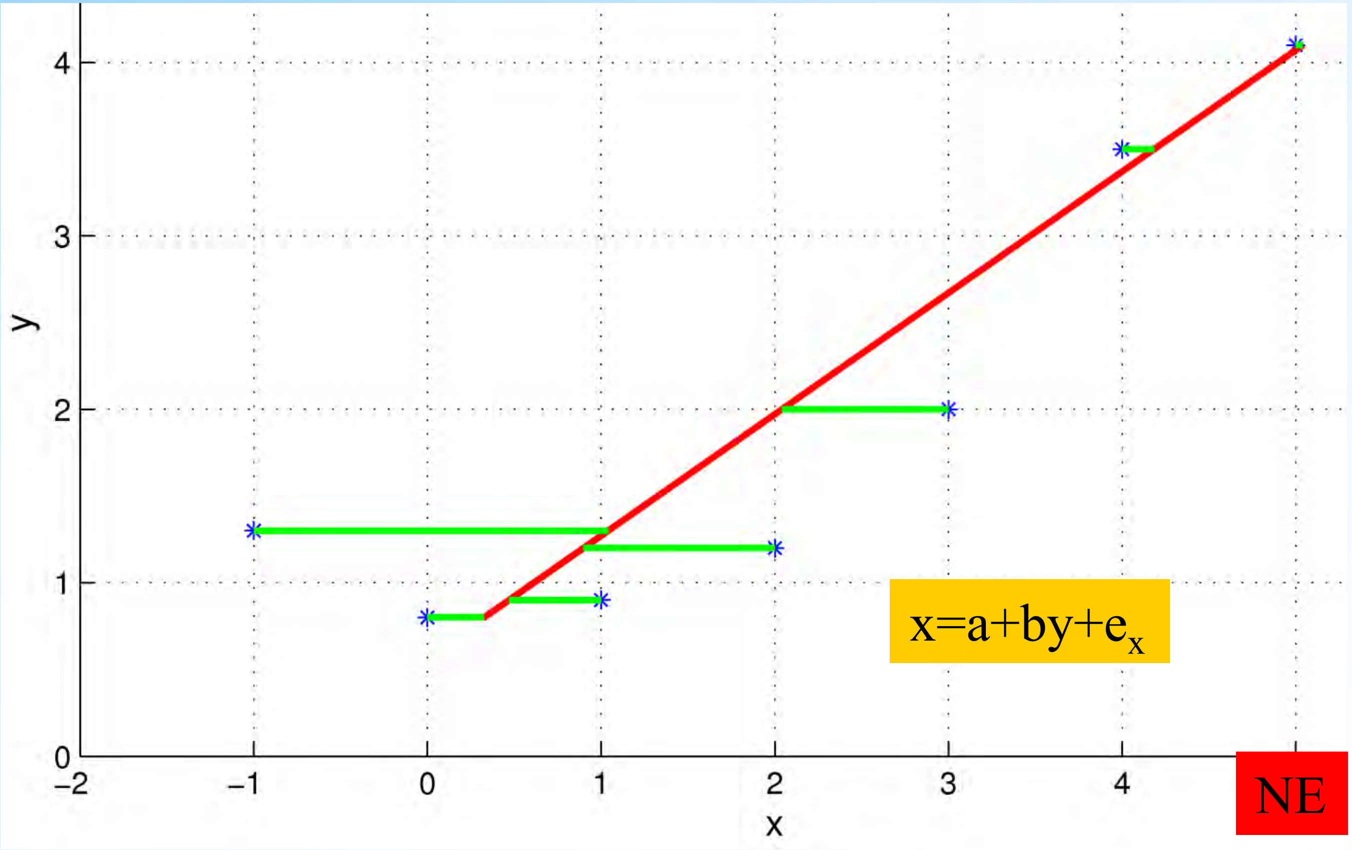
i	1	2	3	4	5	6	7
x_i [m]	–1	0	1	2	3	4	5
y_i [m]	1.3	0.8	0.9	1.2	2.0	3.5	4.1

$$\Rightarrow \quad \hat{a} = -0.815\,[\mathrm{m}], \quad \hat{b} = 1.428\,[-]$$

$\hat{x}_i$ [m]	$\hat{e}_{x_i}$ [m]
1.041	–2.041
0.327	–0.327
0.470	0.530
0.898	1.102
2.041	0.959
4.183	–0.183
5.040	–0.040

$$\Downarrow$$

$$\hat{e}'\hat{e} = 6.723\,[\mathrm{m}^2]$$



→ A-model: Linear Regression

A-model: Linear Regression

$z = a + bx + cy + e_z$

Example:

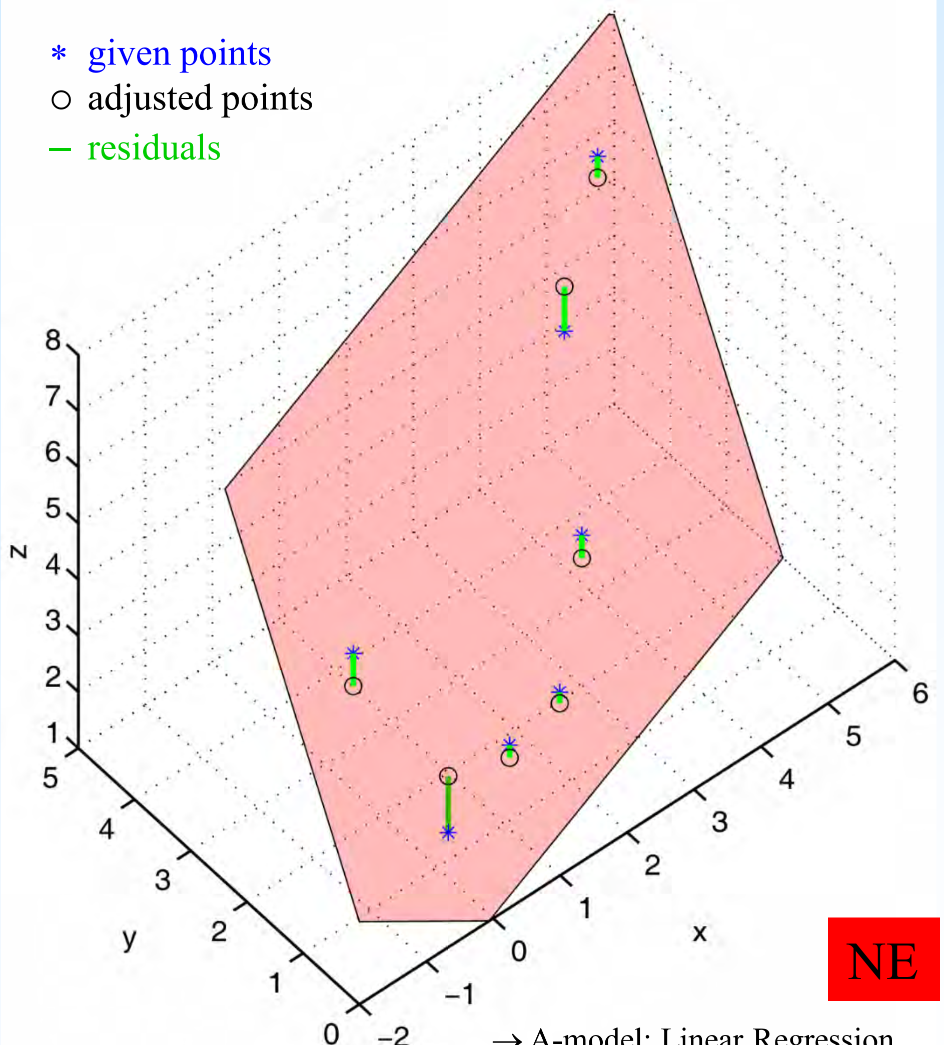
i	1	2	3	4	5	6	7
x_i [m]	–1	0	1	2	3	4	5
y_i [m]	1.3	0.8	0.9	1.2	2.0	3.5	4.1
z_i [m]	5.3	1.8	2.5	2.4	3.7	5.2	7.0

$\Downarrow$

$$\hat{a} = 0.955\ [\mathrm{m}]$$
$$\hat{b} = -0.761\ [-]$$
$$\hat{c} = 2.309\ [-]$$

$$\hat{e}'\hat{e} = 2.377\ [\mathrm{m}^2]$$

$\hat{z}_i$ [m]	$\hat{e}_{z_i}$ [m]
4.718	0.582
2.803	–1.003
2.272	0.228
2.204	0.196
3.291	0.409
5.994	–0.794
6.618	0.382



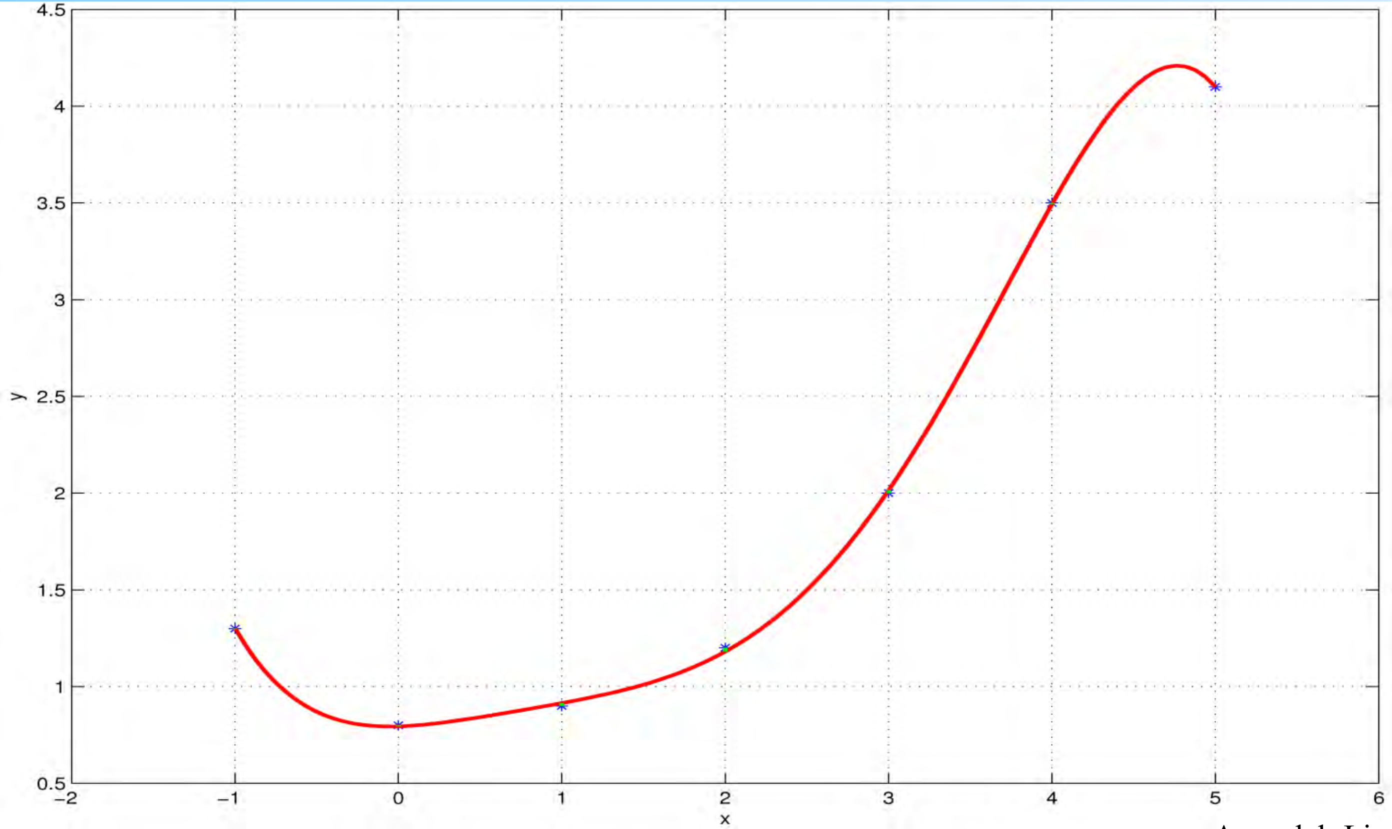
NE

→ A-model: Linear Regression

A-model: Linear Regression

Example (5^{th} order polynomial): $y_i = a_0 + a_1 x_i + a_2 x_i^2 + a_3 x_i^3 + a_4 x_i^4 + a_5 x_i^5 + e_i$

	x_i, y_i as before						
$\hat{y}_i$ [m]	1.3010	0.7942	0.9146	1.1805	2.0146	3.4942	4.1010
$\hat{e}_{y_i}$ [m]	–0.0010	0.0058	–0.0146	0.0195	–0.0146	0.0058	–0.0010



$$\hat{a}_0 = 0.7942\,[\mathrm{m}]$$
$$\hat{a}_1 = 0.0250\,[-]$$
$$\hat{a}_2 = 0.2239\,[1/\mathrm{m}]$$
$$\hat{a}_3 = -0.2078\,[1/\mathrm{m}^2]$$
$$\hat{a}_4 = 0.0898\,[1/\mathrm{m}^3]$$
$$\hat{a}_5 = -0.0104\,[1/\mathrm{m}^4]$$
$$\hat{e}'\hat{e} = 8.8\times 10^{-4}\,[\mathrm{m}^2]$$

NE

→ A-model: Linear Regression (with constraints)

A-model: Linear Regression (with constraints)

Example (5^{th} order polynomial): $f(x_i) = y_i = a_0 + a_1 x_i + a_2 x_i^2 + a_3 x_i^3 + a_4 x_i^4 + a_5 x_i^5 + e_i$

Constraints:

(a) f(x) must pass through points $P_1(x_1{=}0.5, y_1{=}7)$ and $P_2(x_2{=}4, y_2{=}15.5)$

$$y_1 = a_0 + a_1 x_1 + a_2 x_1^2 + a_3 x_1^3 + a_4 x_1^4 + a_5 x_1^5$$
$$y_2 = a_0 + a_1 x_2 + a_2 x_2^2 + a_3 x_2^3 + a_4 x_2^4 + a_5 x_2^5$$
$$\Downarrow$$

$$D_1' = \begin{bmatrix} 1 & 0.5 & 0.25 & 0.125 & 0.0625 & 0.03125 \\ 1 & 4 & 16 & 64 & 256 & 1024 \end{bmatrix}, \quad c_1 = \begin{bmatrix} 7 \\ 15.5 \end{bmatrix}$$

(b) Tangent t(x) at $P_3(x_3{=}2, f(x_3))$ must pass through point $P_4(x_4{=}4, y_4{=}-5)$

Tangent at P_3: $t(x) = f(x_3) + f'(x_3)(x - x_3)$

$$f(x_3) = a_0 + a_1 x_3 + a_2 x_3^2 + a_3 x_3^3 + a_4 x_3^4 + a_5 x_3^5$$
$$f'(x_3) = a_1 + 2a_2 x_3 + 3a_3 x_3^2 + 4a_4 x_3^3 + 5a_5 x_3^4$$

Tangent at P_3 must pass through P_4: $t(x_4) = y_4 = f(x_3) + f'(x_3)(x_4 - x_3)$

NE

→ A-model: Linear Regression (with constraints)

A-model: Linear Regression (with constraints)

$$\begin{aligned} D_2' &= [1 \quad x_4 \quad -x_3^2+2x_3x_4 \quad -2x_3^3+3x_3^2x_4 \quad -3x_3^4+4x_3^3x_4 \quad -4x_3^5+5x_3^4x_4] \\ &= [1 \quad 4 \quad 12 \quad 32 \quad 80 \quad 192] \end{aligned}$$

$$D' = \begin{bmatrix} D_1' \\ D_2' \end{bmatrix} \begin{bmatrix} 1 & 0.5 & 0.25 & 0.125 & 0.0625 & 0.03125 \\ 1 & 4 & 16 & 64 & 256 & 1024 \\ 1 & 4 & 12 & 32 & 80 & 192 \end{bmatrix}, \quad c = \begin{bmatrix} c_1 \\ y_4 \end{bmatrix} = \begin{bmatrix} 7 \\ 15.5 \\ -5 \end{bmatrix}$$

Constrained Lagrangean

$$\text{Target function } \mathcal{L}_A(x,\lambda) = \frac{1}{2}\underset{1\times1}{e'e} + \underset{1\times3}{\lambda'}(\underset{3\times6}{D'}\,\underset{6\times1}{x} - \underset{3\times1}{c}) =$$

$$= \frac{1}{2}(y-Ax)'(y-Ax) + \lambda'(D'x-c) \to \min_{x,\lambda}$$

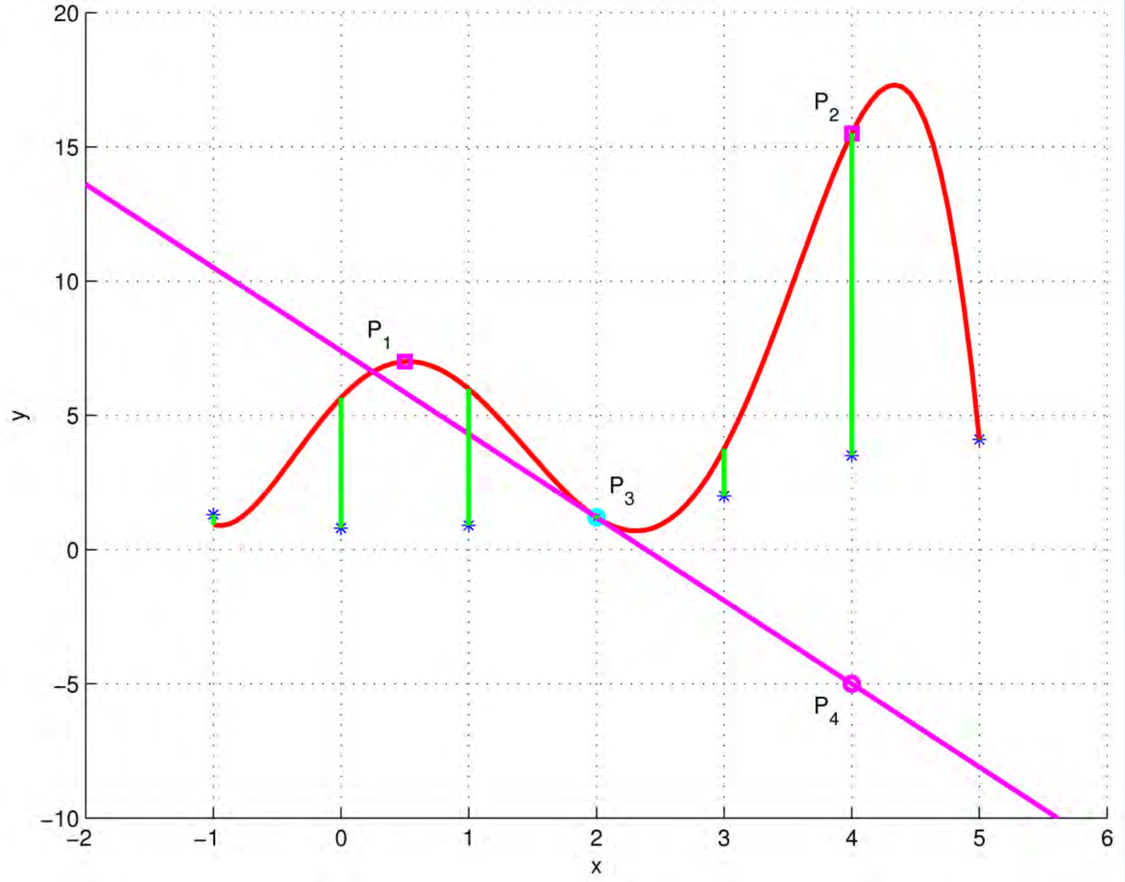
$$\Rightarrow \begin{bmatrix} \underset{6\times6}{A'A} & \underset{6\times3}{D} \\ \underset{3\times6}{D'} & \underset{3\times3}{0} \end{bmatrix} \begin{bmatrix} \underset{6\times1}{\hat{x}} \\ \underset{3\times1}{\hat{\lambda}} \end{bmatrix} = \begin{bmatrix} \underset{6\times1}{A'y} \\ \underset{3\times1}{c} \end{bmatrix} \Rightarrow \quad \hat{x}, \hat{y}, \hat{e}$$

NE

→ A-model: Linear Regression (with constraints)

A-model: Linear Regression (with constraints)

	x_i, y_i as before						
$\hat{y}_i$ [m]	0.9172	5.6562	5.9826	1.2004	3.7476	15.5000	4.0748
$\hat{e}_{y_i}$ [m]	0.3828	–4.8562	–5.0826	–0.0004	–1.7476	–12.0000	0.0252



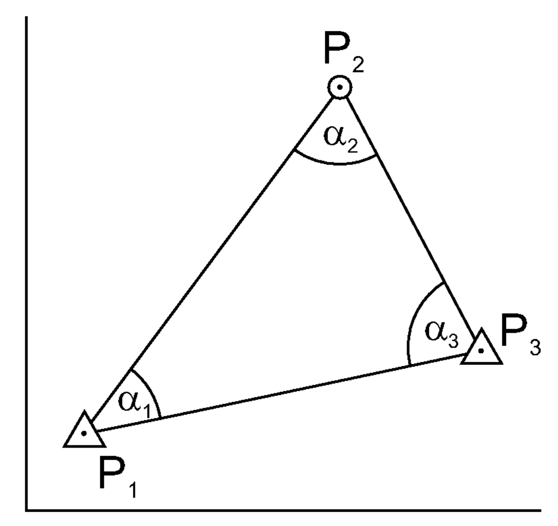
$$\hat{a}_0 = 5.6562\,[\mathrm{m}]$$
$$\hat{a}_1 = 4.9280\,[-]$$
$$\hat{a}_2 = -3.7409\,[1/\mathrm{m}]$$
$$\hat{a}_3 = -2.1978\,[1/\mathrm{m}^2]$$
$$\hat{a}_4 = 1.5346\,[1/\mathrm{m}^3]$$
$$\hat{a}_5 = -0.1975\,[1/\mathrm{m}^4]$$
$$\hat{e}'\hat{e} = 196.6168\,[\mathrm{m}^2]$$

NE

→ B-model: Principles

B-model: Principles

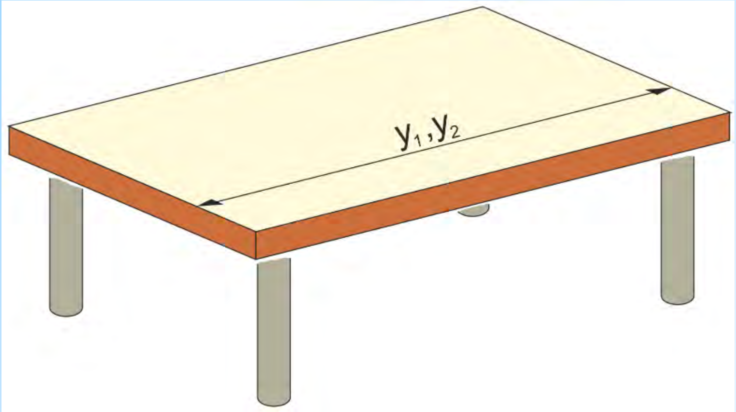
Here, no unknown parameters x (prices, table length, coordinates, etc.) exist ! The only interest is to find estimates ê so that by correcting the observations y certain condition equations between the measurements are satisfied.



Example: Find estimates $\hat{e}_1, \hat{e}_2, \hat{e}_3$ in order to correct measurements $\alpha_1,\alpha_2,\alpha_3$ so that the condition equation $(\alpha_1 - \hat{e}_1) + (\alpha_2 - \hat{e}_2) + (\alpha_3 - \hat{e}_3) = 180°$ is met.

Example: Measure the table length twice $\rightarrow y_1,y_2$. In the ideal case we have (consistency)

$$y_1 = y_2 \sim y_1 - y_2 = 0 \sim \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = 0 \sim B'y = 0.$$

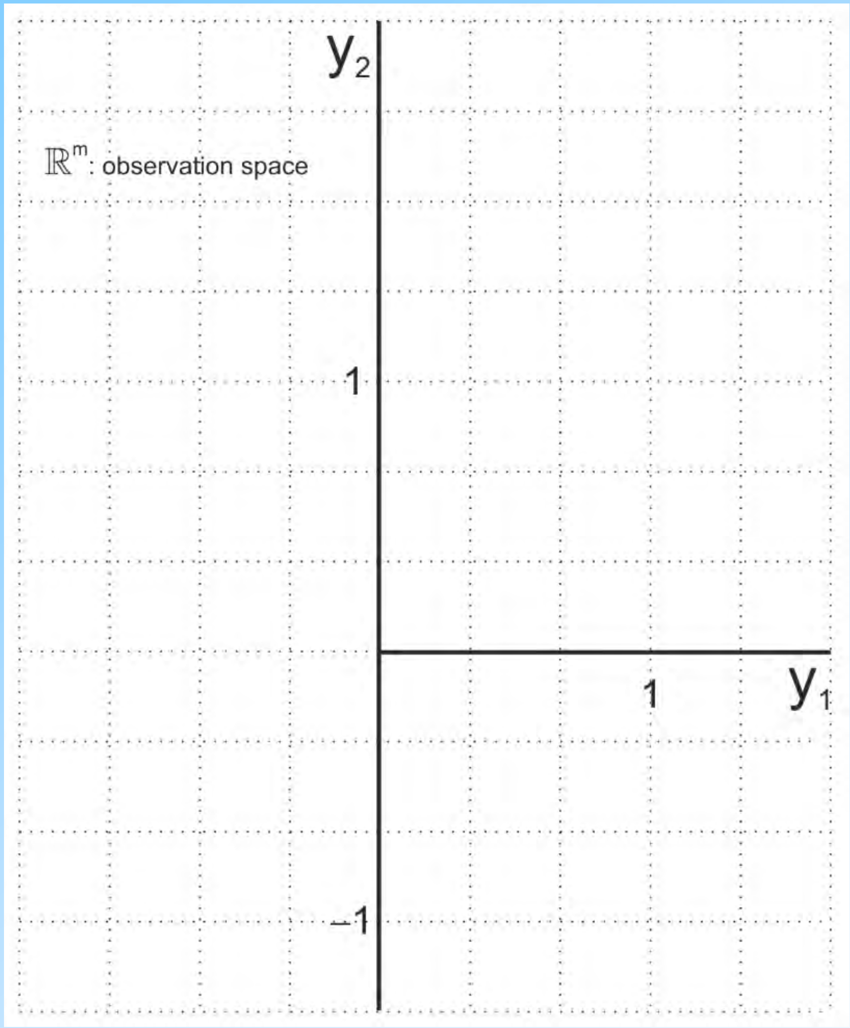


However, most probably $y_1 \neq y_2 \Rightarrow B'y \neq 0$ (inconsistency because of measurement errors) and two unknown inconsistency parameters $e=[e_1,e_2]'$ must be added in order to satisfy at the end the condition equation $B'(y - \hat{e}) = 0 \sim w := B'y = B'\hat{e}$. The term $w := B'y$ is called "misclosure".

→ B-model: Geometry

B-model: Geometry

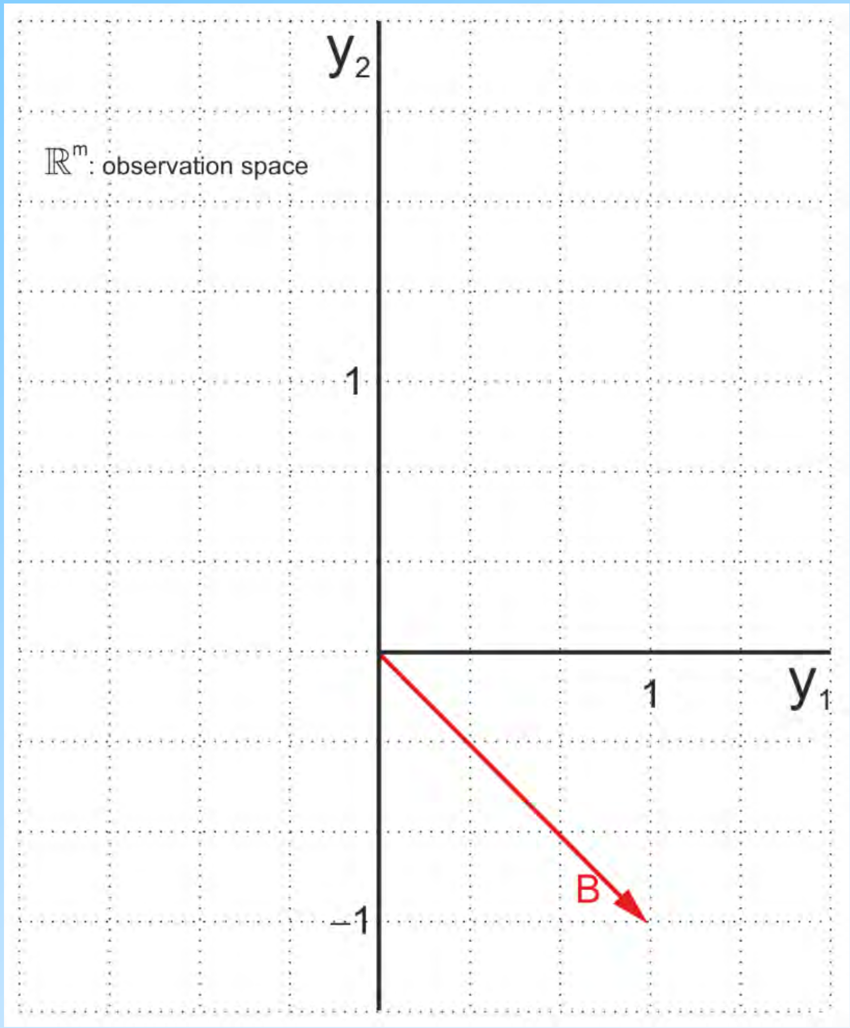
1) Create coordinate system with as many axes as observations, here m=2 → $\mathbb{R}^2$



→ B-model: Geometry

B-model: Geometry

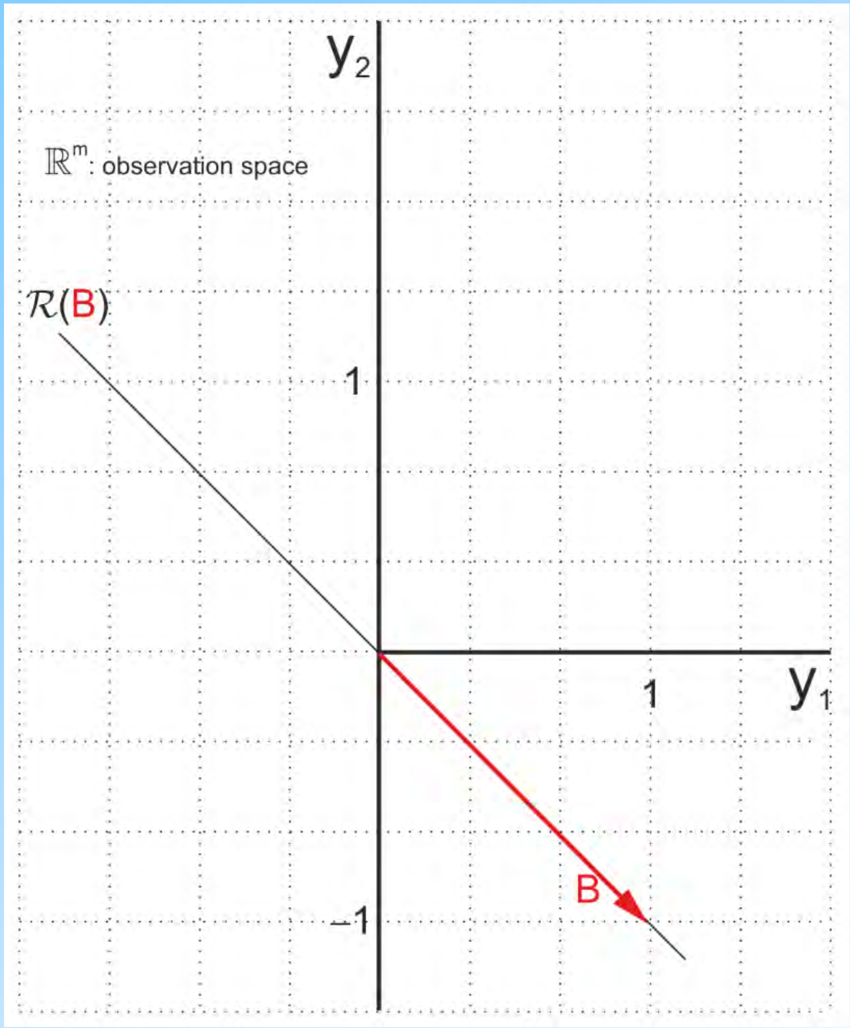
2) Plot columns of B, here B=[1,–1]′ (a single column)



→ B-model: Geometry

B-model: Geometry

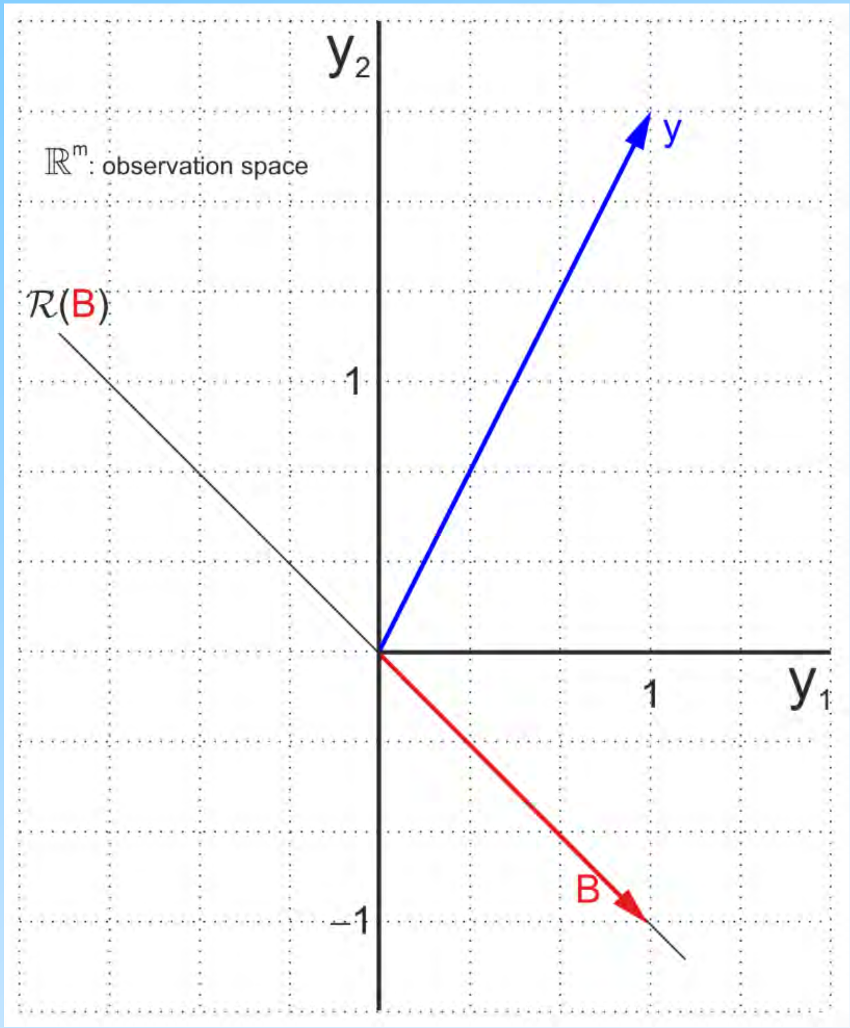
3) Draw the line in $\mathbb{R}^2$ spanned by B



→ B-model: Geometry

B-model: Geometry

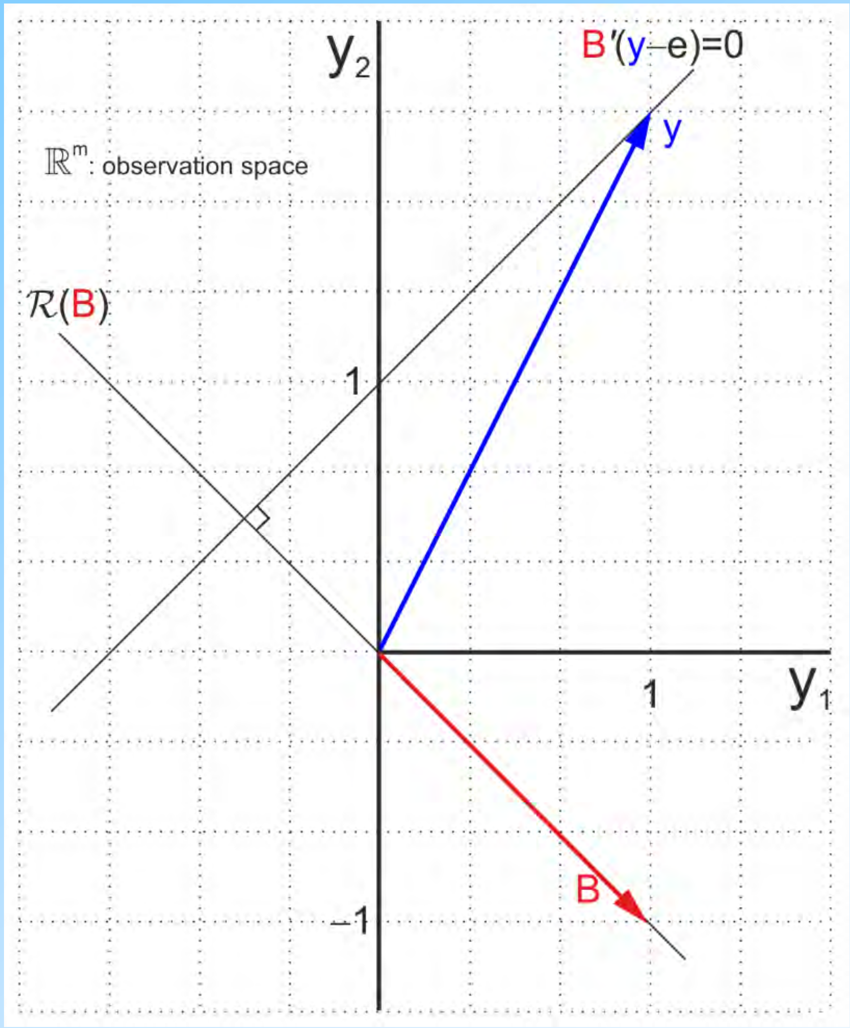
4) Plot observations y, which create (here) a point in 2d-space



→ B-model: Geometry

B-model: Geometry

5) Draw the line $B'y = B'e \sim B'(y-e)=0$

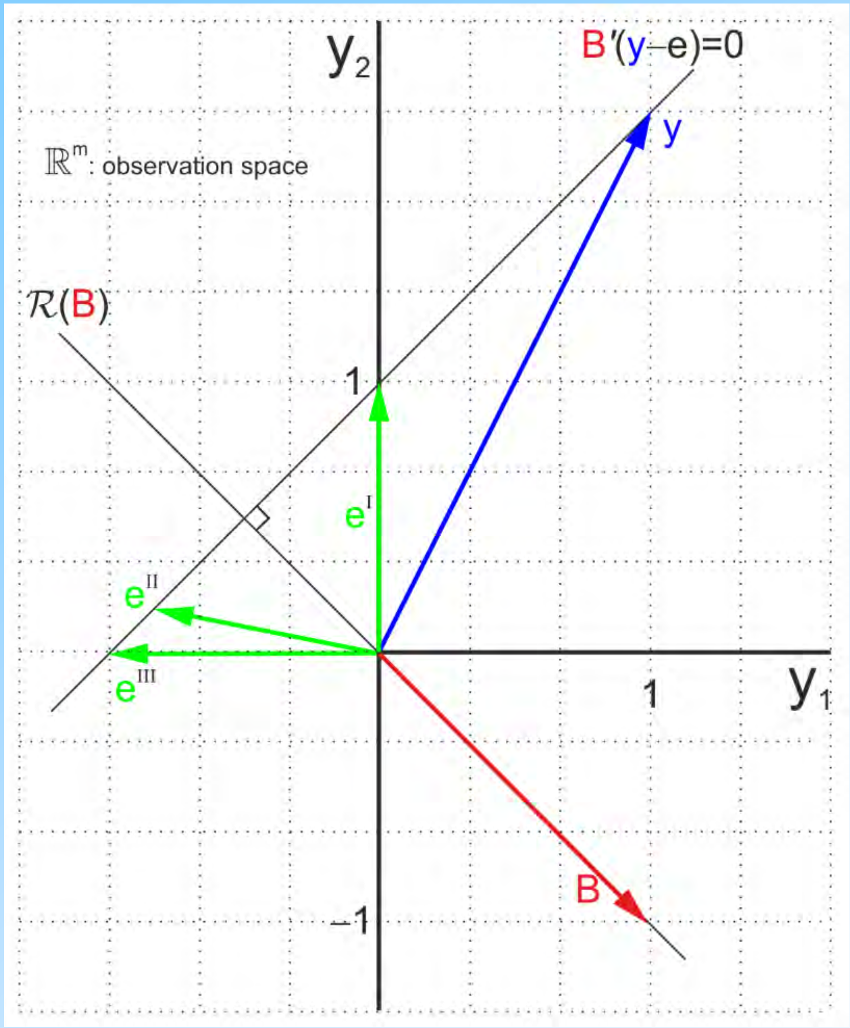


→ B-model: Geometry

 GIS

B-model: Geometry

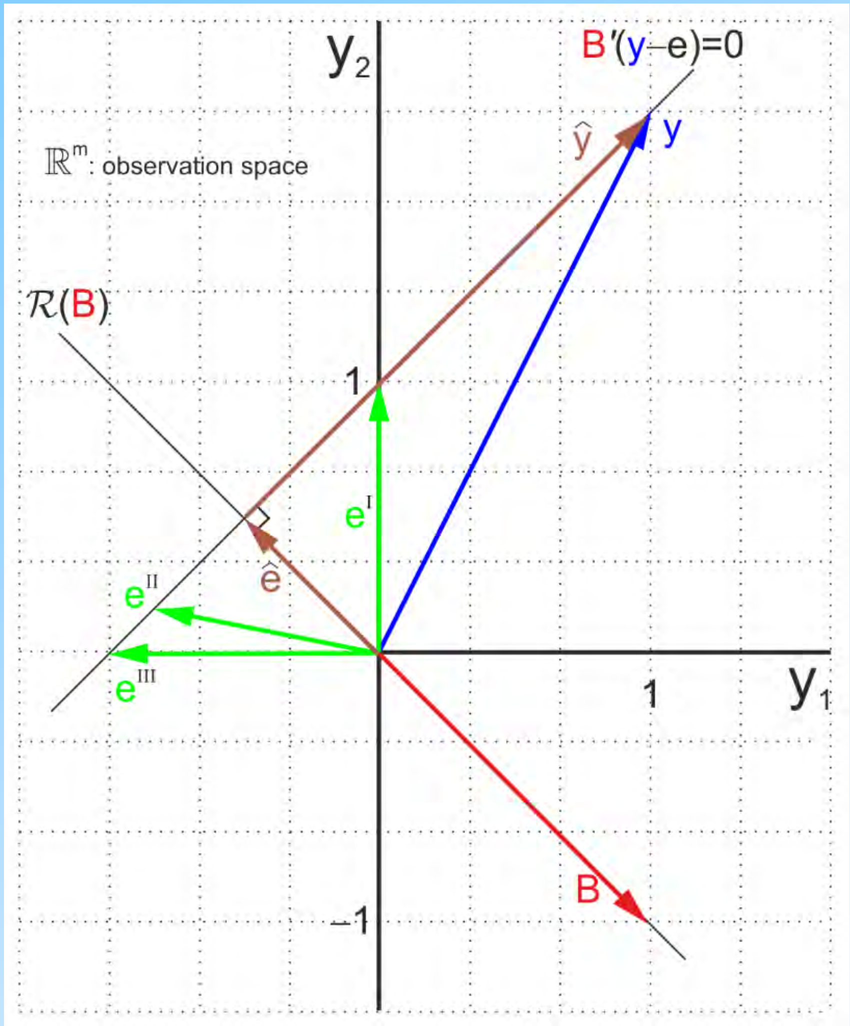
6) Find e's which satisfy $B'(y-e)=0$



→ B-model: Geometry

B-model: Geometry

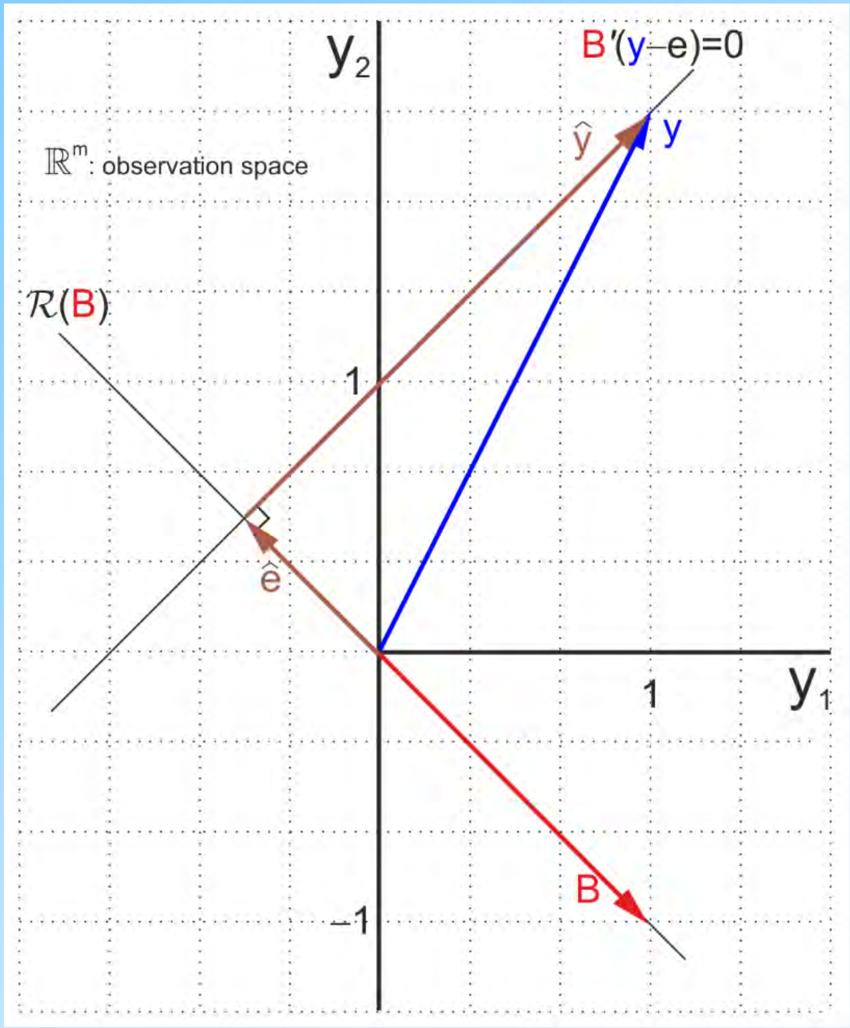
7) Find shortest $\hat{e}$ satisfying least squares postulate $\hat{e}'\hat{e}$ =min, and corresponding $\hat{y}$



→ B-model: Geometry

B-model: Geometry

8) Remove e's longer than $\hat{e}$



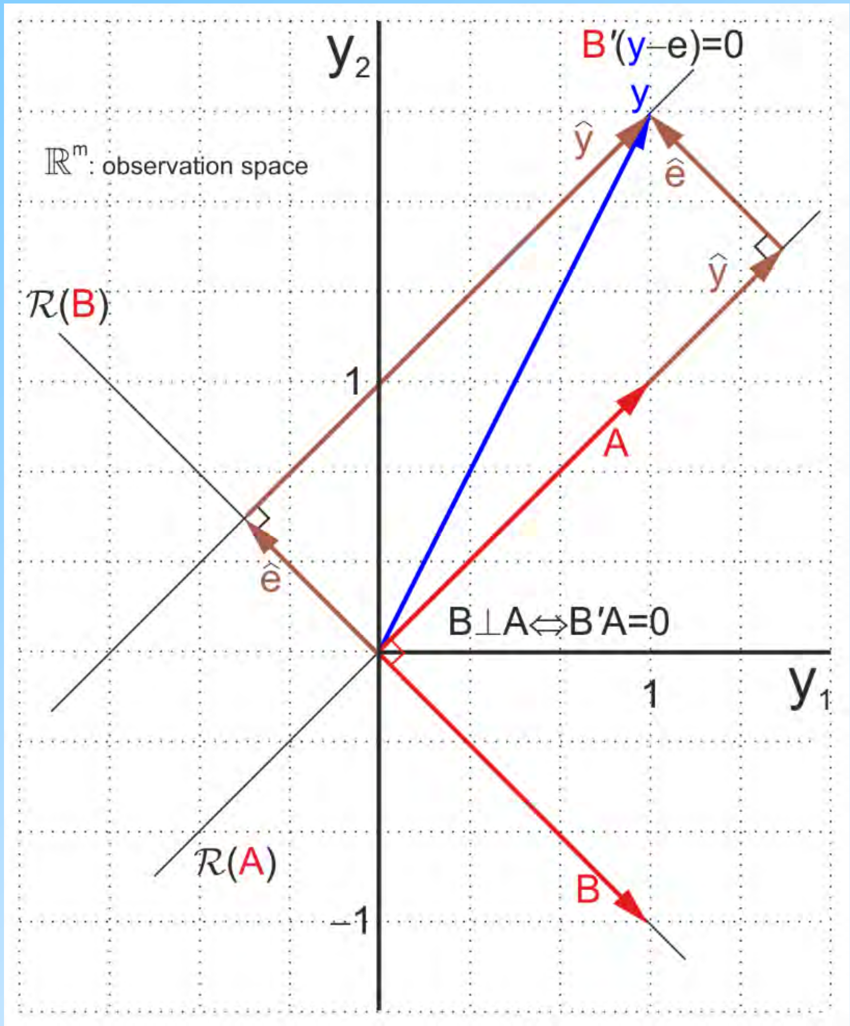
$$\Rightarrow \hat{e} = P_B y = B(B'B)^{-1}B'y$$ LS-Estimate of e

$$\Rightarrow \hat{y} = P_B^{\perp} y = \left[I_m - B(B'B)^{-1}B'\right]y$$ LS-Estimate of y

→ A-model & B-model: Geometry

A-model & B-model: Geometry

9) Connect A-model (parameter adjustment) and B-model (condition adjustment)



$$\Rightarrow \hat{e} = P_B y = B(B'B)^{-1}B'y \qquad \text{LS-Estimate of } e$$

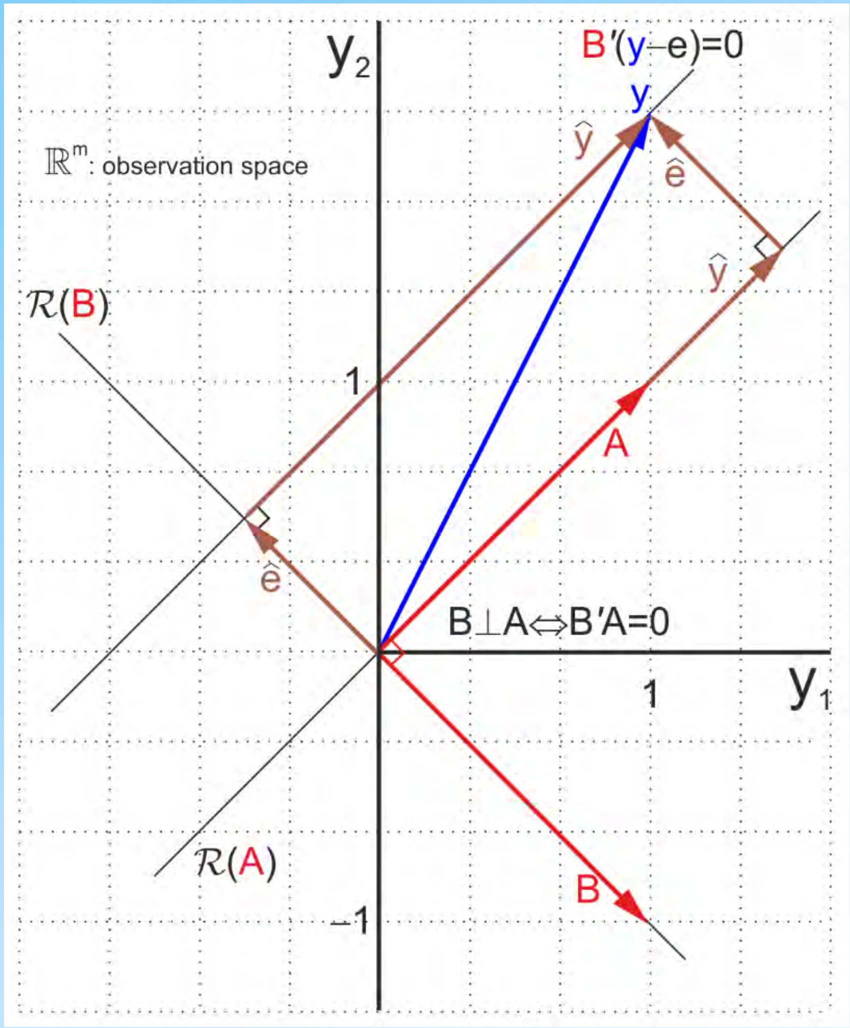
$$\Rightarrow \hat{y} = P_B^{\perp} y = \left[I_m - B(B'B)^{-1}B'\right]y \qquad \text{LS-Estimate of } y$$

$$\Rightarrow P_B = P_A^{\perp} \ , \ P_B^{\perp} = P_A$$

→ A-model & B-model: Link

A-model & B-model: Link

9) Connect A-model (parameter adjustment) and B-model (condition adjustment)



A-model: $y = Ax + e$

B-model: $B'y = B'e \sim B'(y-e) = 0$

$\Downarrow$

A-model $\rightarrow$ B-model

$B'y = B'Ax + B'e = B'e$

$\Leftrightarrow B'A = 0 \sim B \perp A$

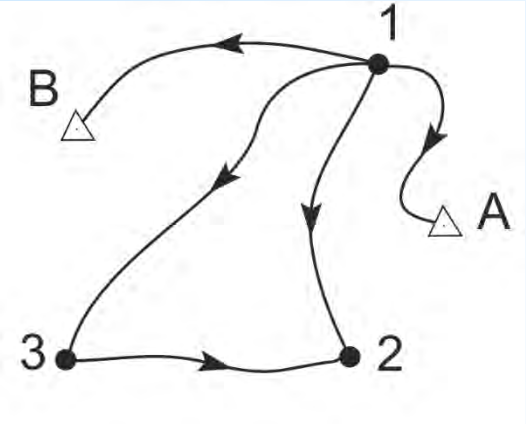
B forms a basis for the left nullspace $\mathcal{N}(A')$ of A

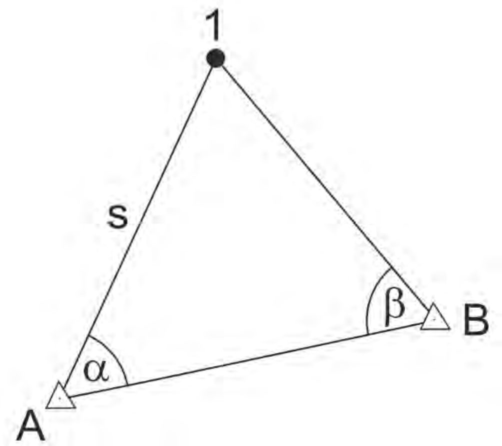
→ B-model: Calculus

B-model: Calculus

In principle, B' in the B-model is different from D' in the A-model because no x exist. While B' works on the level of conditions between the measurements, matrix D' has to do with constraints for parameters $\Rightarrow$ no datum problem in the B-model.
m: number of observations, r: number of conditions = number of observations which could be deleted without making the network collapse.

Example: In the levelling network five points A,1,2,3,B are connected by levelling lines. Points A and B are benchmarks with heights given. Obviously point heights 1,2,3 can still be determined even if e.g. levelling lines B-1 and 2-3 are discarded ($\rightarrow$ r = 2): The height network will not collapse.





Example: In the triangle, angles α,β and distance s have been measured in order to compute coordinates for point 1. Again, angle β could be omitted; α and s would do the job because A and B are benchmarks with given coordinates. One condition equation (which one ?) is the correct answer.

→ B-model: Calculus

B-model: Calculus

Most often r is hard to find; r equals the redundancy, $r = m - n$, which involves the A-model again, however. In the B-model all observations must be involved or processed, and all condition equations must be linearly independent. Starting from $B'(y-e) = B'c$ we define the Lagrange-, cost or target function $\mathcal{L}_B(e,\lambda)$ and look for those $\hat{e}$ and $\hat{\lambda}$ which minimize $\mathcal{L}_B(\hat{e},\hat{\lambda})$.

$$\mathcal{L}_B(e,\lambda) = \frac{1}{2}\underset{1\times 1}{e'e} + \underset{1\times r}{\lambda'}(\underbrace{B'y - B'c}_{r\times 1} - \underset{r\times m}{B'}\ \underset{m\times 1}{e}) = \frac{1}{2}\underset{1\times 1}{e'e} + \underset{1\times r}{\lambda'}(\underset{r\times 1}{w} - \underset{r\times m}{B'}\ \underset{m\times 1}{e}) \to \min_{e,\lambda}$$

The B-model equations $B'(y-e) = B'c$ are included as constraints always in their homogeneous form. The quantity $w = B'y - B'c$ is the misclosure, λ are additional unknowns, so-called Lagrange multipliers, which are used in constrained optimization problems.

Necessary conditions for a minimum:

$$\left.\begin{aligned} \frac{\partial \mathcal{L}_B(e,\lambda)}{\partial e}(\hat{e},\hat{\lambda}) &= \hat{e} - B\hat{\lambda} \overset{!}{=} 0 \\ \frac{\partial \mathcal{L}_B(e,\lambda)}{\partial \lambda}(\hat{e},\hat{\lambda}) &= w - B'\hat{e} \overset{!}{=} 0 \end{aligned}\right\} \Rightarrow \begin{bmatrix} \underset{m\times m}{I} & \underset{m\times r}{-B} \\ \underset{r\times m}{-B'} & \underset{r\times r}{0} \end{bmatrix} \begin{bmatrix} \underset{m\times 1}{\hat{e}} \\ \underset{r\times 1}{\hat{\lambda}} \end{bmatrix} = \begin{bmatrix} \underset{m\times 1}{0} \\ \underset{r\times 1}{-w} \end{bmatrix}$$

→ B-model: Calculus

B-model: Calculus

$$\Rightarrow \hat{\lambda} = (B'B)^{-1} w \Rightarrow \hat{e} = B\hat{\lambda} = B(B'B)^{-1} w$$

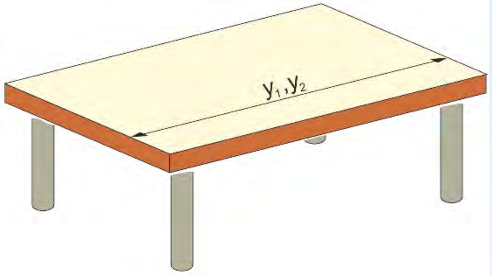
Remark 1:
Constraints $w - B'e = 0$ can also be added with sign reversed, i.e. the Lagrange function is then formulated as $\mathcal{L}_B(e,\lambda) = \frac{1}{2} e'e - \lambda'\left(w - B'e\right)$. This will change the sign of $\hat{\lambda}$ only but not of $\hat{e}$. Please note that this will also change the sign in one of the computational checks.

Remark 2:
The partial derivative of $\mathcal{L}_B$ w.r.t. λ returns always the homogeneous constraints $w - B'e = 0$.

→ B-model: Example table length

B-model: Example table length

Example: Measuring the side length of a table twice (y_1,y_2) with a tape rule: side length (x) not relevant in B-model.



Ideal case: $y_1 = y_2 \sim y_1 - y_2 = 0 \sim \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = B'y = 0$

Reality: measurements contain errors (inconsistencies), i.e.

$$y_1 \neq y_2 \sim y_1 - y_2 \neq 0 \sim \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = B'y = w \neq 0$$

misclosure w

Add (unknown) inconsistencies e=[e_1,e_2]'

$$B'(y-e) = B'y - B'e = w - B'e = 0$$

$$\Rightarrow \hat{\lambda} = (B'B)^{-1} w$$

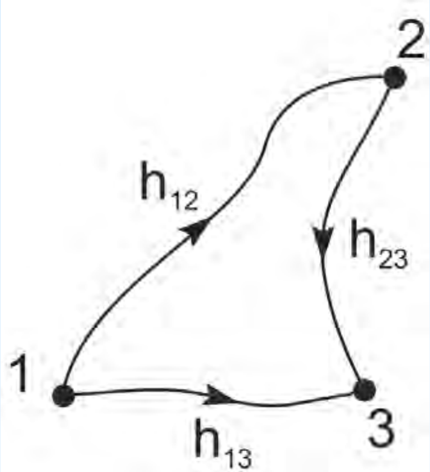
$$\Rightarrow \hat{e} = B\hat{\lambda} = B(B'B)^{-1} w = B(B'B)^{-1} B'y = \frac{1}{2} \begin{bmatrix} y_1 - y_2 \\ -(y_1 - y_2) \end{bmatrix}$$

$$\Rightarrow \hat{e}'\hat{e} = \frac{1}{2}(y_1 - y_2)^2$$

→ B-model: Example height network

B-model: Example height network

m = 3 observations, r = 1 condition equation



dot indicates inconsistency

$$h_{12} + h_{23} - h_{13} \doteq 0 \quad \Rightarrow$$

Condition equation: loop

$$h_{12} - e_{12} + h_{23} - e_{23} - h_{13} + e_{13} = 0$$

no dot means consistency

$$\begin{bmatrix} 1 & -1 & 1 \end{bmatrix}\left(\begin{bmatrix} h_{12} \\ h_{13} \\ h_{23} \end{bmatrix} - \begin{bmatrix} e_{12} \\ e_{13} \\ e_{23} \end{bmatrix}\right) = 0 \sim B'(y-e) = B'y - B'e = w - B'e = 0$$

$$\Rightarrow B(B'B)^{-1} = \frac{1}{3}\begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \quad , \quad B'y = w = h_{12} + h_{23} - h_{13}$$

$$\hat{e} = \frac{1}{3}\begin{bmatrix} h_{12} - h_{13} + h_{23} \\ h_{13} - h_{12} - h_{23} \\ h_{12} - h_{13} + h_{23} \end{bmatrix} \quad , \quad \hat{y} = \frac{1}{3}\begin{bmatrix} 2h_{12} + h_{13} - h_{23} \\ h_{12} + 2h_{13} + h_{23} \\ h_{13} - h_{12} + 2h_{23} \end{bmatrix}$$

$$B'(y-\hat{e}) = B'\hat{y} = 0 \quad \checkmark$$

$$\hat{e}'\hat{e} - w'\hat{\lambda} = 0 \quad \checkmark$$

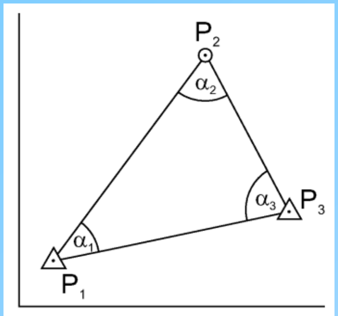
Computational checks

Note: $y_{A-model} \neq y_{B-model}$ and $\hat{y}_{A-model} \neq \hat{y}_{B-model}$ because original $y_{A-model}$ was modified by subtracting H_1. But

$$\hat{y}_{A-model} - \begin{bmatrix} H_1 \\ H_1 \\ 0 \end{bmatrix} = \hat{y}_{B-model}$$

→ B-model: Example trigonometric network

B-model: Example trigonometric network



Q: Is it possible to calculate coordinates for P_2 using the B-model ?

A: No, because it is not made for this purpose; it does not hunt for coordinates ! What can be done with the B-model is to find corrections e_1, e_2, e_3 to the measured angles so that in the end $(\alpha_1 - \hat{e}_1) + (\alpha_2 - \hat{e}_2) + (\alpha_3 - \hat{e}_3) = 180°$ is satisfied. With

$y = [\alpha_1 \quad \alpha_2 \quad \alpha_3]'$, $B' = [1 \quad 1 \quad 1]$ and the misclosure $w = B'y - 180°$ we get

$$\hat{e} = B(B'B)^{-1} w = \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}(\alpha_1 + \alpha_2 + \alpha_3 - 180°) = \frac{1}{3}\begin{bmatrix} w \\ w \\ w \end{bmatrix} = \begin{bmatrix} \hat{e}_1 \\ \hat{e}_2 \\ \hat{e}_3 \end{bmatrix}$$

LS-Estimate of e

$$\hat{y} = y - \hat{e} = \frac{1}{3}\begin{bmatrix} 3\alpha_1 - w \\ 3\alpha_2 - w \\ 3\alpha_3 - w \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 2\alpha_1 - \alpha_2 - \alpha_3 + 180° \\ 2\alpha_2 - \alpha_1 - \alpha_3 + 180° \\ 2\alpha_3 - \alpha_1 - \alpha_2 + 180° \end{bmatrix}$$

LS-Estimate of y

$B'\hat{y} - 180° = 0 \quad \checkmark$

Main check: "Corrected" observations must satisfy the condition equation !

$\rightarrow$ Taylor series expansion/Linearization

Taylor series expansion/Linearization

Approximate a given irrational function y=f($\mathbf{x}$) at a point of expansion $\mathbf{x}=\mathbf{x}_0$ using much simpler functions: Taylor's polynomials $P_n(\mathbf{x})$ after Brooke Taylor (1685-1731). $\mathbf{x}$ may be a single independent variable $\mathbf{x}$=x or a p-dimensional vector of independent variables $\mathbf{x}=[x_1,\ldots,x_p]'$.

One independent variable $\mathbf{x}$=x :

Taylor's polynomial $P_n(x)$ of f(x) at $x=x_0$ is a polynomial of degree n that coincides with f(x) in the first n derivatives. x_0 is called "Taylor point (of expansion)".

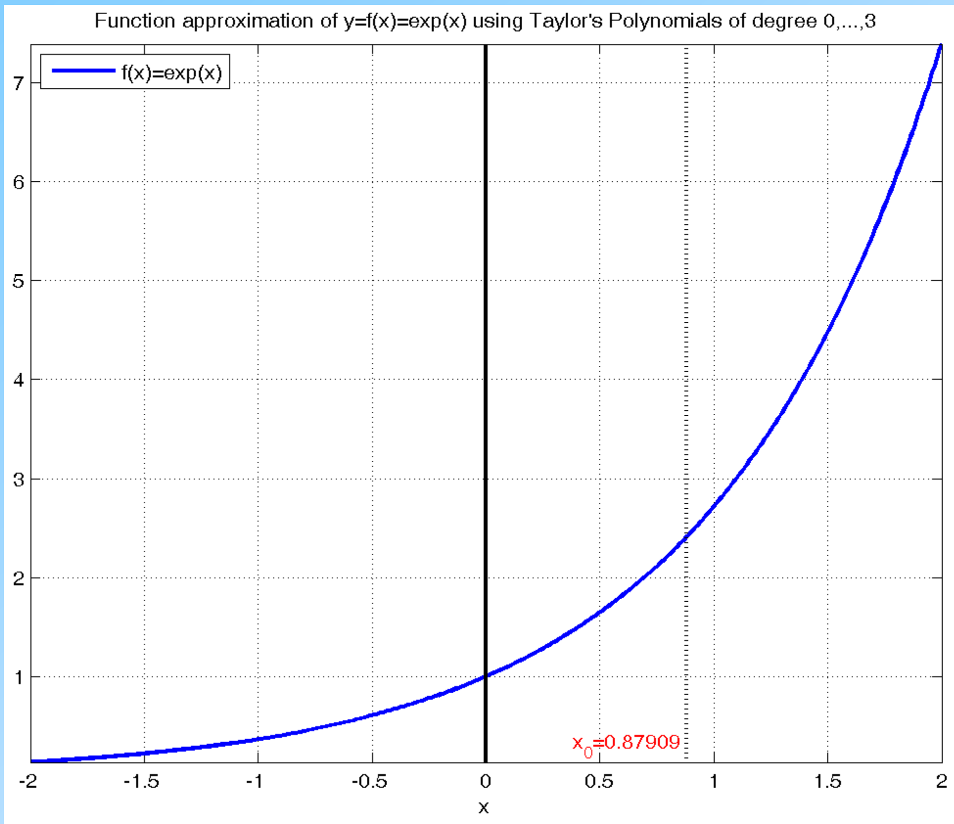
1st derivative, not transposed !

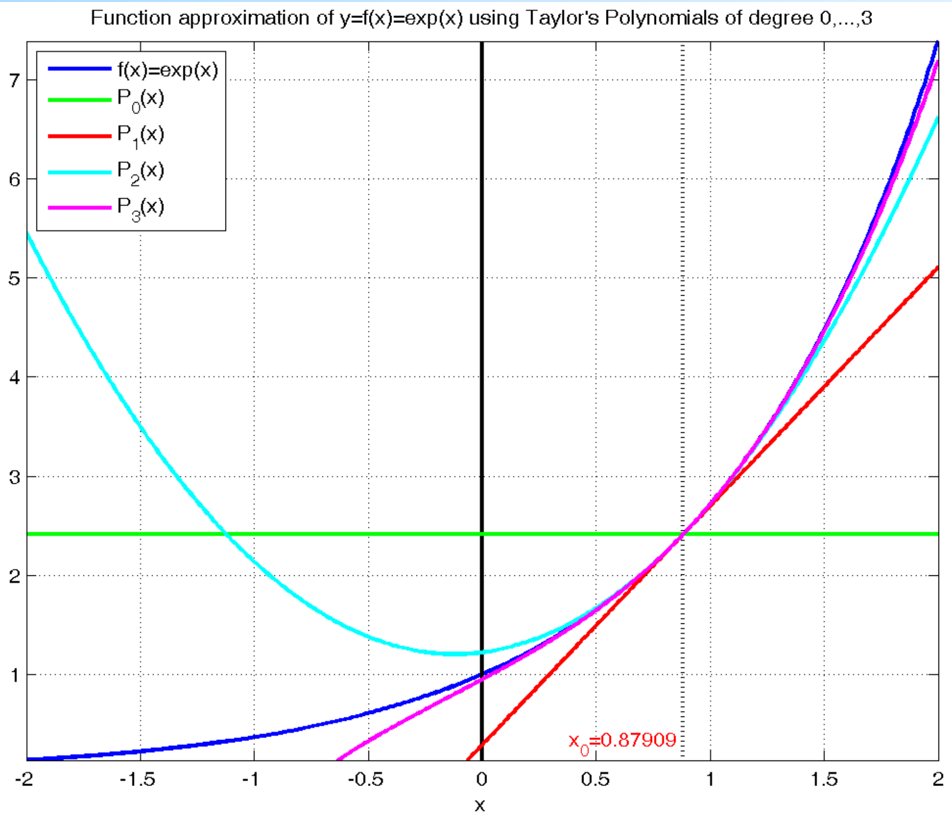
$$f(x) = \underbrace{\underbrace{\underbrace{f(x_0)}_{P_0:\text{ constant polynomial}} + \left.\frac{f'(x)}{1!}\right|_{x=x_0}(x-x_0)}_{P_1:\text{ linear polynomial}} + \left.\frac{f''(x)}{2!}\right|_{x=x_0}(x-x_0)^2}_{P_2:\text{ quadratic polynomial}} + \ldots + \left.\frac{f^{(n)}(x)}{n!}\right|_{x=x_0}(x-x_0)^n + \ldots$$

→ Taylor series expansion/Linearization

Taylor series expansion/Linearization

Approximation of $y=f(x)=\exp(x)$ at $x=x_0=0.87909$

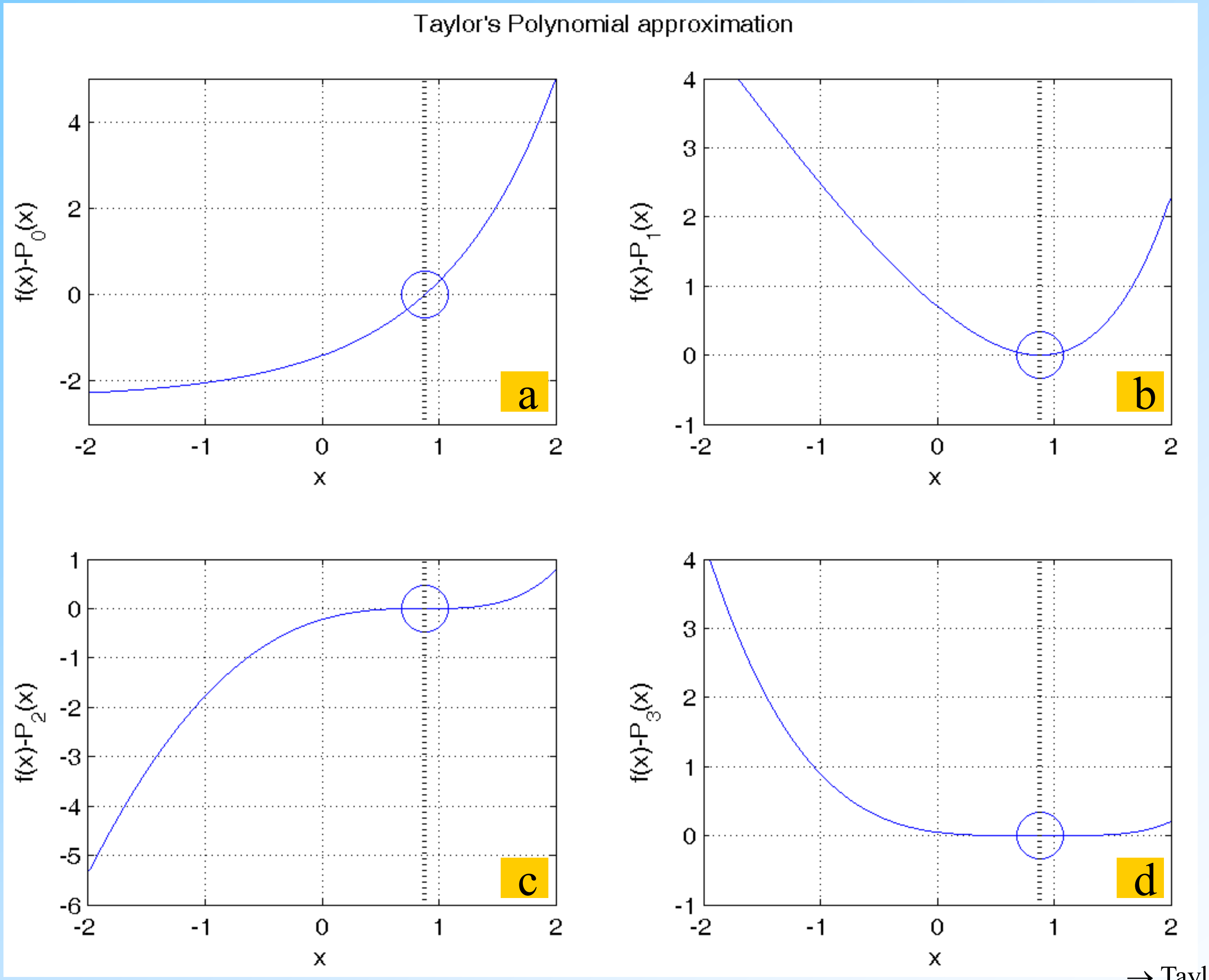




Constant, linear, quadratic and cubic polynomial at $x=x_0=0.87909$

→ Taylor series expansion/Linearization

Taylor series expansion/Linearization



Quality of approximation using
(a) constant,
(b) linear,
(c) quadratic,
(d) cubic polynomial

→ Taylor series expansion/Linearization

Taylor series expansion/Linearization

Two independent variables $\mathbf{x}=[x_1,x_2]'$:

Taylor's polynomial $P_n(x_1,x_2)$ of $f(x_1,x_2)$ at $x_1 = x_1^0,\ x_2 = x_2^0$ is a polynomial of degree n that coincides with $f(x_1,x_2)$ in the first n derivatives. $x_1^0,\ x_2^0$ is called "Taylor point (of expansion)".

$$f(x_1,x_2) = f(x_1^0,x_2^0) + \left.\frac{\partial f(x_1,x_2)}{\partial x_1}\right|_{x_1=x_1^0} (x_1 - x_1^0) + \left.\frac{\partial f(x_1,x_2)}{\partial x_2}\right|_{x_1=x_1^0} (x_2 - x_2^0) + \ .\ .\ .$$

[redacted]

Example 1 (Taylor point a_0,b_0 given):

$$f(a,b) = ae^{bx} = a_0 e^{b_0 x} + \left.\frac{\partial(ae^{bx})}{\partial a}\right|_{\substack{a=a_0\\ b=b_0}} (a-a_0) + \left.\frac{\partial(ae^{bx})}{\partial b}\right|_{\substack{a=a_0\\ b=b_0}} (b-b_0) + \ \ldots$$

$$= a_0 e^{b_0 x} + e^{b_0 x} (a-a_0) + a_0 x e^{b_0 x} (b-b_0) + \ \ldots.$$

$$= a_0 e^{b_0 x} + e^{b_0 x} \Delta a + a_0 x e^{b_0 x} \Delta b + \ \ldots.$$

$$= a_0 e^{b_0 x} + \left[\ e^{b_0 x} \quad a_0 x e^{b_0 x}\ \right] \begin{bmatrix} \Delta a \\ \Delta b \end{bmatrix} + \ \ldots.$$

→ Taylor series expansion/Linearization

GIS

Taylor series expansion/Linearization

Example 2a: Distance observation equation (Taylor point $x_A^0, y_A^0, x_B^0, y_B^0$ given); explicit differentiation

$$s_{AB} := s(x_A, y_A, x_B, y_B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2} =$$

$$= s_{AB}^0 + \frac{x_A^0 - x_B^0}{s_{AB}^0}\Delta x_A + \frac{y_A^0 - y_B^0}{s_{AB}^0}\Delta y_A + \frac{-(x_A^0 - x_B^0)}{s_{AB}^0}\Delta x_B + \frac{-(y_A^0 - y_B^0)}{s_{AB}^0}\Delta y_B + \ldots$$

$$= s_{AB}^0 + \begin{bmatrix} \frac{x_A^0 - x_B^0}{s_{AB}^0} & \frac{y_A^0 - y_B^0}{s_{AB}^0} & \frac{x_B^0 - x_A^0}{s_{AB}^0} & \frac{y_B^0 - y_A^0}{s_{AB}^0} \end{bmatrix} \begin{bmatrix} \Delta x_A \\ \Delta y_A \\ \Delta x_B \\ \Delta y_B \end{bmatrix} + \ldots$$

$$s_{AB}^0 = s(x_A^0, y_A^0, x_B^0, y_B^0) = \sqrt{(x_A^0 - x_B^0)^2 + (y_A^0 - y_B^0)^2}$$

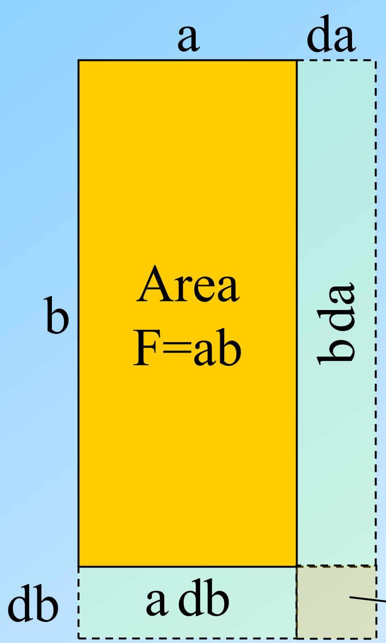
NE

→ Taylor series expansion/Linearization

Taylor series expansion/Linearization

Example 2b: Distance observation equation (Taylor point $x_A^0, y_A^0, x_B^0, y_B^0$ given); implicit differentiation

Start from implicit functional relationship between observation and unknowns, $s_{AB}^2 = (x_A - x_B)^2 + (y_A - y_B)^2$, and compute its total differential.



$da\,db \approx 0$

What is the total differential, e.g. of area F ?

Total differential dF of area F: Small change of F due to a small change (da,db) of all ("total") independent variables (here: a and b).

$$F = ab \Rightarrow dF = \frac{\partial F}{\partial a} da + \frac{\partial F}{\partial b} db = b\,da + a\,db = \begin{bmatrix} b & a \end{bmatrix} \begin{bmatrix} da \\ db \end{bmatrix}$$

→ Taylor series expansion/Linearization

Taylor series expansion/Linearization

Total differential of $s_{AB}^2 := (x_A - x_B)^2 + (y_A - y_B)^2$

$$
\begin{aligned}
2s_{AB}\,ds_{AB} &= \frac{\partial(s_{AB}^2)}{\partial x_A}dx_A + \frac{\partial(s_{AB}^2)}{\partial y_A}dy_A + \frac{\partial(s_{AB}^2)}{\partial x_B}dx_B + \frac{\partial(s_{AB}^2)}{\partial y_B}dy_B \\
&= 2(x_A - x_B)dx_A + 2(y_A - y_B)dy_A - 2(x_A - x_B)dx_B - 2(y_A - y_B)dy_B
\end{aligned}
$$

$$
ds_{AB} = \frac{x_A - x_B}{s_{AB}}dx_A + \frac{y_A - y_B}{s_{AB}}dy_A - \frac{x_A - x_B}{s_{AB}}dx_B - \frac{y_A - y_B}{s_{AB}}dy_B +
$$

Introduce given approximate coordinates and switch from $d \to \Delta$

$$
\Delta s_{AB} = \frac{x_A^0 - x_B^0}{s_{AB}^0}\Delta x_A + \frac{y_A^0 - y_B^0}{s_{AB}^0}\Delta y_A - \frac{x_A^0 - x_B^0}{s_{AB}^0}\Delta x_B - \frac{y_A^0 - y_B^0}{s_{AB}^0}\Delta y_B
$$

$$
\Delta s_{AB} := s_{AB} - s_{AB}^0
$$

$$
s_{AB}^0 = s(x_A^0, y_A^0, x_B^0, y_B^0) = \sqrt{(x_A^0 - x_B^0)^2 + (y_A^0 - y_B^0)^2}
$$

→ Taylor series expansion/Linearization

Taylor series expansion/Linearization

Remark: If one (or more) variables, e.g. y_A (y_A,x_B), is (are) taken as constant, differentiation with respect to that (these) variable(s) is not nessecary.

$$s(x_A, y_A, x_B, y_B) = s_{AB}^0 + \frac{x_A^0 - x_B^0}{s_{AB}^0}\Delta x_A + \frac{-(x_A^0 - x_B^0)}{s_{AB}^0}\Delta x_B + \frac{-(y_A - y_B^0)}{s_{AB}^0}\Delta y_B + \ldots$$

$$= s_{AB}^0 + \begin{bmatrix} \frac{x_A^0 - x_B^0}{s_{AB}^0} & \frac{x_B^0 - x_A^0}{s_{AB}^0} & \frac{y_B^0 - y_A^0}{s_{AB}^0} \end{bmatrix} \begin{bmatrix} \Delta x_A \\ \Delta x_B \\ \Delta y_B \end{bmatrix} + \ldots$$

$$s_{AB}^0 = s(x_A^0, y_A, x_B^0, y_B^0) = \sqrt{(x_A^0 - x_B^0)^2 + (y_A - y_B^0)^2}$$

$$s(x_A, y_A, x_B, y_B) = s_{AB}^0 + \frac{x_A^0 - x_B}{s_{AB}^0}\Delta x_A + \frac{-(y_A - y_B^0)}{s_{AB}^0}\Delta y_B + \ldots$$

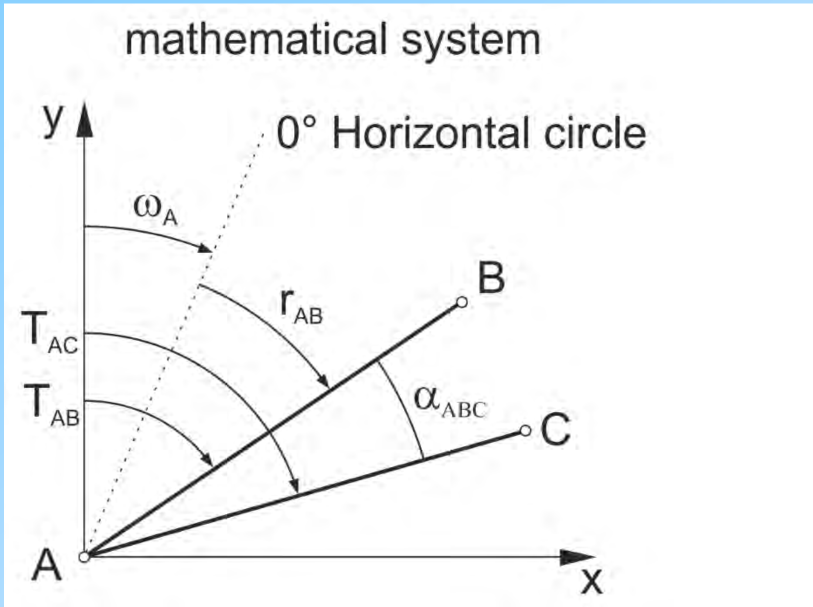
$$= s_{AB}^0 + \begin{bmatrix} \frac{x_A^0 - x_B^0}{s_{AB}^0} & \frac{y_B^0 - y_A^0}{s_{AB}^0} \end{bmatrix} \begin{bmatrix} \Delta x_A \\ \Delta y_B \end{bmatrix} + \ldots$$

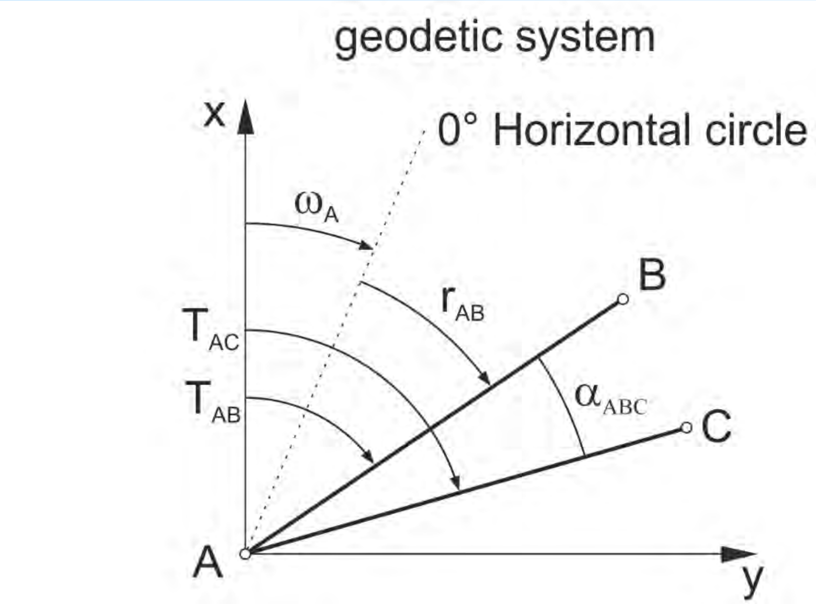
$$s_{AB}^0 = s(x_A^0, y_A, x_B, y_B^0) = \sqrt{(x_A^0 - x_B)^2 + (y_A - y_B^0)^2}$$

→ Taylor series expansion/Linearization

Taylor series expansion/Linearization

Example 3: bearing (direction) r_{AB}, grid bearing T_{AB} and angle α_{ABC} observation equation





$$r_{AB} := r(x_A, y_A, x_B, y_B) = T_{AB} - \omega_A$$

$$T_{AB} := T(x_A, y_A, x_B, y_B) = \arctan \frac{x_B - x_A}{y_B - y_A}$$

$$T_{AB} := T(x_A, y_A, x_B, y_B) = \arctan \frac{y_B - y_A}{x_B - x_A}$$

$$\alpha_{ABC} := \alpha(x_A, y_A, x_B, y_B, x_C, y_C) = T_{AC} - T_{AB}$$

$\rightarrow$ Taylor series expansion/Linearization

Taylor series expansion/Linearization

Mathematical system:

$$T_{AB} = T^0_{AB} - \frac{y^0_B - y^0_A}{\left(s^0_{AB}\right)^2}\Delta x_A + \frac{x^0_B - x^0_A}{\left(s^0_{AB}\right)^2}\Delta y_A + \frac{y^0_B - y^0_A}{\left(s^0_{AB}\right)^2}\Delta x_B - \frac{x^0_B - x^0_A}{\left(s^0_{AB}\right)^2}\Delta y_B \quad + \ldots$$

$$= T^0_{AB} + \frac{1}{\left(s^0_{AB}\right)^2}\begin{bmatrix} -(y^0_B - y^0_A) & x^0_B - x^0_A & y^0_B - y^0_A & -(x^0_B - x^0_A) \end{bmatrix}\begin{bmatrix} \Delta x_A \\ \Delta y_A \\ \Delta x_B \\ \Delta y_B \end{bmatrix} + \ldots$$

$$T^0_{AB} = T(x^0_A, y^0_A, x^0_B, y^0_B) = \arctan\frac{y^0_B - y^0_A}{x^0_B - x^0_A}$$

$$\text{MATLAB}: T_{AB}[\text{rad}] = \text{atan}\left[(y_B - y_A)/(x_B - x_A)\right] \text{ (+quadrant rule)}$$

$$T_{AB}[\text{rad}] = \text{atan2}\left[(y_B - y_A),(x_B - x_A)\right] \text{(quadrant rule built-in)}$$

Physical units: Theodolite observations (angles, directions) are usually given in [deg] or [gon] while computed T_{AB} is in [rad] $\rightarrow$ Convert angles to [rad]:

$$[\text{rad}] = [\text{deg}]\frac{\pi}{180°} \quad \text{or } [\text{rad}] = [\text{gon}]\frac{\pi}{200^g}$$

→ Linearization & adjustment

Linearization & adjustment (A-model)

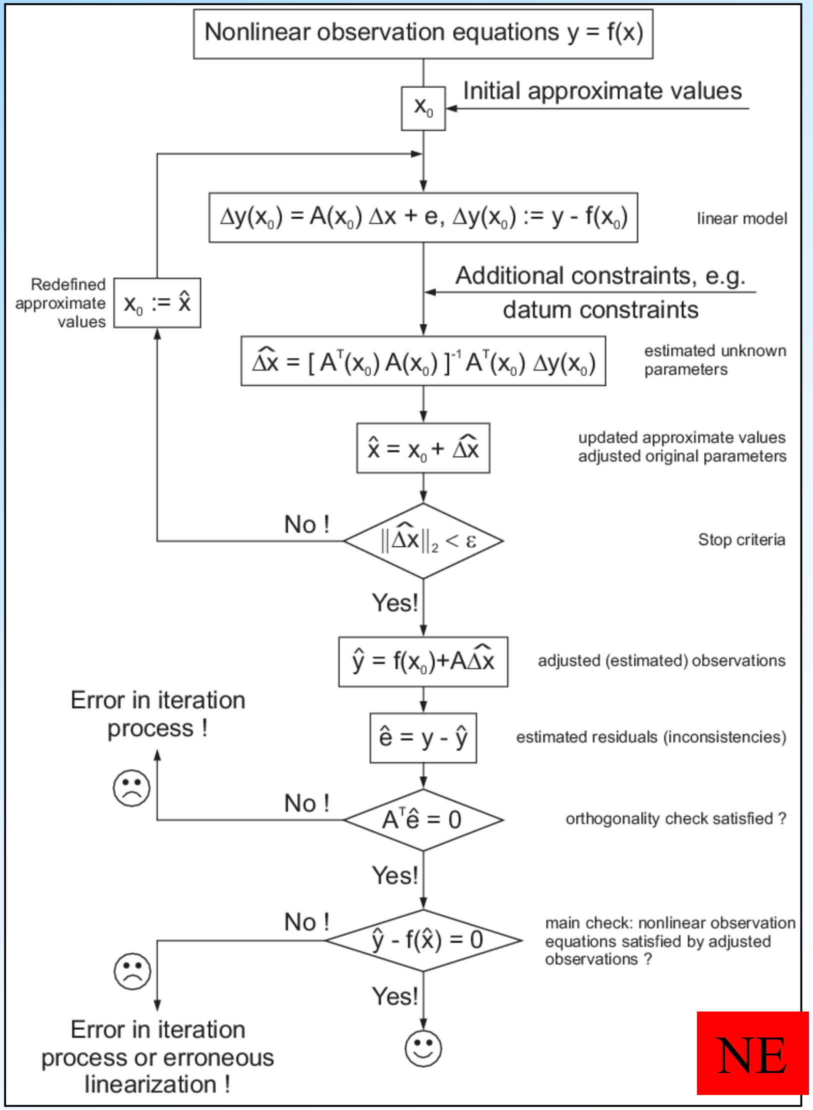
As we are dealing with linear models in adjustment applications, e.g. with the A-model $y=Ax+e$, the question arises how to connect linearization and adjustment.
Answer: Approximate non-linear observation equations f(x) by its linear part

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0) = f(x_0) + f'(x_0)\Delta x$$

of Taylor's polynomial and assume that terms of higher order are small ! Then put

$$\Delta y = f(x) - f(x_0),\ A(x_0) = f'(x_0)$$

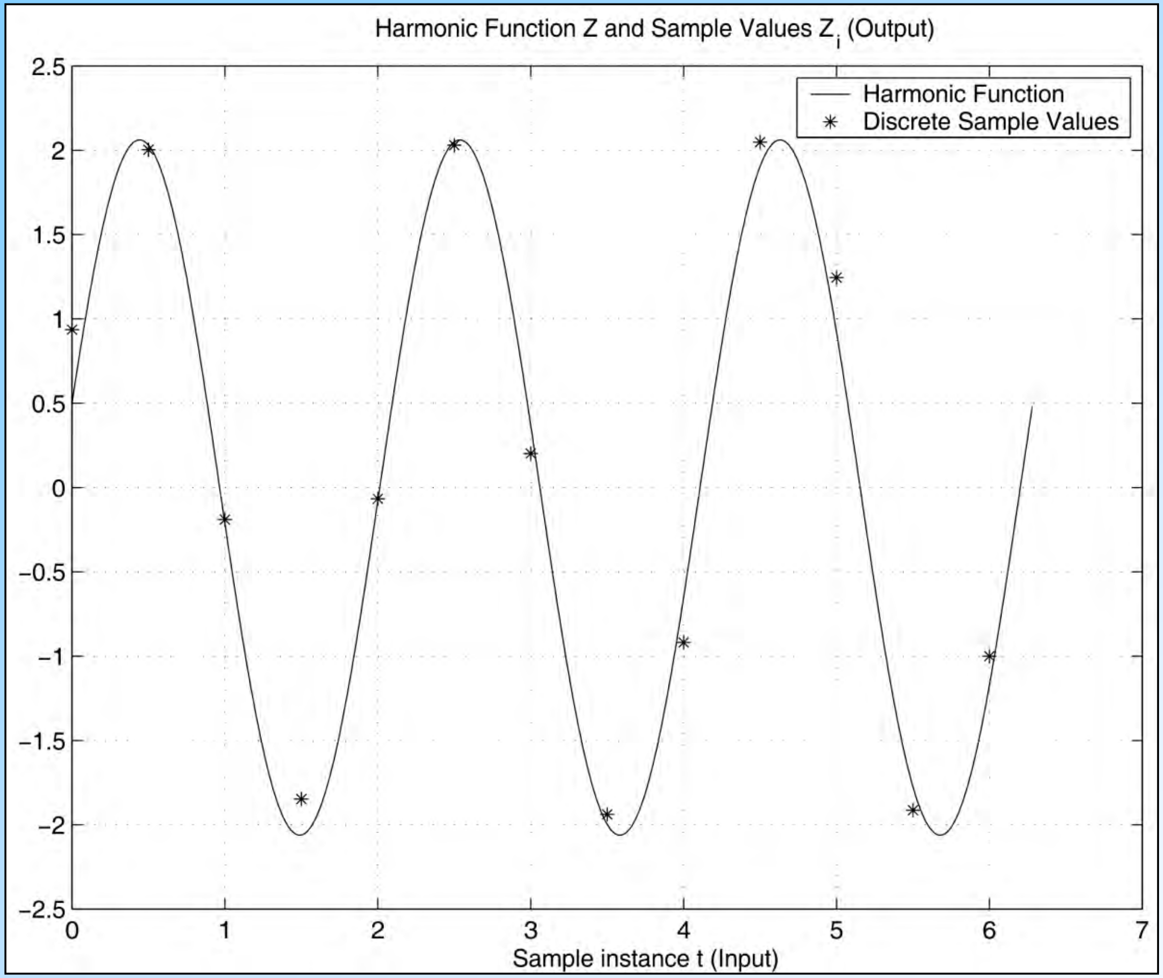
and identify $y_{\text{linear model}} \equiv \Delta y,\ x_{\text{linear model}} \equiv \Delta x.$
Drawback: Vector of inconsistencies e sucks up neglected higher order terms !
Solution: Iterative scheme in order to sharpen up Taylor point and thus to reduce the magnitude of (neglected) higher order terms in each iteration step.



→ Example 1: Non-linear regression

Example 1: Non-linear regression

Non-linear regression of a harmonic oscillation



Input: t_i, $i=1,\ldots,m > 3$
Output: $Z_i=Z(t_i)$

Model:

$$Z_i = a\sin\omega t_i + b\cos\omega t_i + e_i$$

Unknowns: a, b, ω

→ Example 2: Non-linear regression

Example 2: Non-linear regression

Fit to the data $u=[1,2,4]'$, $v(u)=[1.9,1.1,0.25]'$ an exponential function $v=ae^{bu}$.
Linearization at given Taylor point (a_0,b_0: $a=a_0+\Delta a$, $b=b_0+\Delta b$):

$$
\begin{aligned}
v &= v(a_0,b_0) + \left.\frac{\partial v}{\partial a}\right|_{\substack{a=a_0\\b=b_0}}(a-a_0) + \left.\frac{\partial v}{\partial b}\right|_{\substack{a=a_0\\b=b_0}}(b-b_0) + \text{higher order terms} \\
&= v_0 + \left.\begin{bmatrix}\frac{\partial v}{\partial a} & \frac{\partial v}{\partial b}\end{bmatrix}\right|_{\substack{a=a_0\\b=b_0}} \begin{bmatrix}\Delta a\\ \Delta b\end{bmatrix} + \text{higher order terms} \\
&= v_0 + \begin{bmatrix} e^{b_0 u} & a_0 u e^{b_0 u}\end{bmatrix} \begin{bmatrix}\Delta a\\ \Delta b\end{bmatrix} + \text{higher order terms} \Rightarrow \\
\Delta v := v - v_0 &= A \quad \Delta x \quad + e \\
(\Delta y &= A \quad \Delta x \quad + e)
\end{aligned}
$$

$\Rightarrow$ Iterative least-squares adjustment (updating approximate values a_0,b_0 until convergence is achieved) in order eliminate (neglected) higher order terms in the inconsistencies

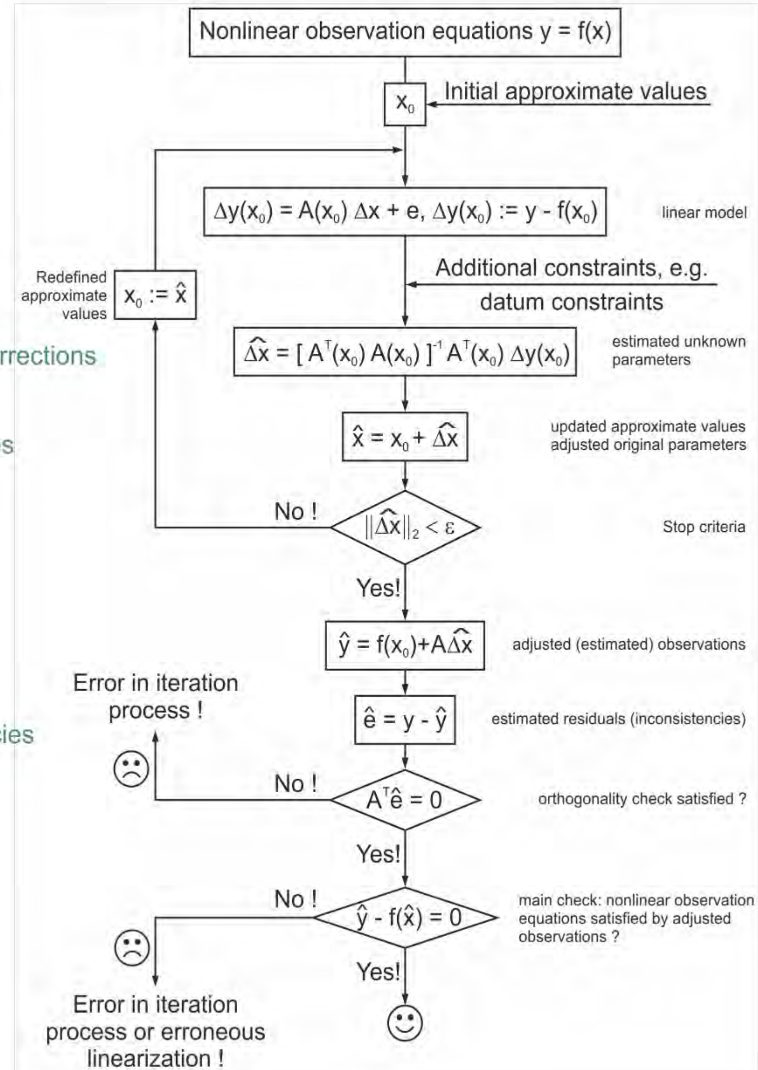
Checks: 1) Orthogonality check: $A'\hat{e} = 0$

2) Main check: $v - \hat{e} - \hat{a}e^{\hat{b}u} = 0 \quad \forall u, v$

→ Example 2: Non-linear regression

Example 2: Non-linear regression

```
normdx_hat=inf;eps=1e-12;           % while loop initialization
counter=0;itermax=100;

a0=…;b0=…;                          % Initial approximate values
while normdx_hat > eps && counter < itermax
  counter=counter+1;
  A=[exp(u*b0),a0*u.*exp(u*b0)];    % Design matrix
  dy=v-a0*exp(b0*u);                % Reduced observations

  dx_hat=A\dy;                      % LS-estimate of unknown corrections

  a0=a0+dx_hat(1);b0=b0+dx_hat(2);  % Updated approximate values

  normdx_hat=norm(dx_hat(1:2));     % Stop criteria

end

dy_hat=A*dx_hat;
v_hat=dy_hat+a0*exp(b0*u);          % Adjusted observations

e_hat=v-v_hat;                      % LS estimate of inconsistencies

check1=A'*e_hat;                    % Orthogonality check
if norm(check1) > 1e-10
  error('Orthogonality check failure');
end
check2=v-e_hat-a0*exp(b0*u);        % Main check
if norm(check2) > 1e-10
  error('Main check failure');
end
%
```



→ Example3: Non-linear regression with constraint(s)

Example 3: Non-linear regression with constraint(s)

Fit to the data $u=[1,2,4]'$, $v(u)=[1.9,1.1,0.25]'$ an exponential function $v=ae^{bu}$ which passes through the point $v(\tilde{u}=3)=0.75 \equiv \tilde{v}=\hat{a}\exp(\hat{b}\tilde{u})$.

Linearization at given Taylor point (a_0,b_0) as before ! But: Constraint is also non-linear and must be linearized, too, in order to be incorporated in the form $D'x=c$ $(D'\Delta x=c)$:

$$\begin{aligned}
\tilde{v} &= \tilde{v}(a_0,b_0) + \left.\frac{\partial\tilde{v}}{\partial a}\right|_{\substack{a=a_0\\ b=b_0}} (a-a_0) + \left.\frac{\partial\tilde{v}}{\partial b}\right|_{\substack{a=a_0\\ b=b_0}} (b-b_0) + \text{higher order terms}\\
&= \tilde{v}_0 + \left.\begin{bmatrix} \dfrac{\partial\tilde{v}}{\partial a} & \dfrac{\partial\tilde{v}}{\partial b} \end{bmatrix}\right|_{\substack{a=a_0\\ b=b_0}} \begin{bmatrix} \Delta a \\ \Delta b \end{bmatrix} + \text{higher order terms}\\
&= \tilde{v}_0 + \begin{bmatrix} e^{b_0\tilde{u}} & a_0\tilde{u}e^{b_0\tilde{u}} \end{bmatrix} \begin{bmatrix} \Delta a \\ \Delta b \end{bmatrix} + \text{higher order terms} \Rightarrow
\end{aligned}$$

$$\Delta\tilde{v} := \tilde{v} - \tilde{v}_0 = D' \quad \Delta x$$

$$(c = D' \quad \Delta x)$$

→ Non-linear function fit with constraints: Example

Example 3: Non-linear regression with constraint(s)

Constrained Lagrangean: $\mathcal{L}_A(\Delta x, \lambda) = \frac{1}{2}(\Delta y - A\Delta x)'(\Delta y - A\Delta x) + \lambda'(D'\Delta x - c) = \min_{\Delta x, \lambda}$

Extended normal equations:
$$\begin{bmatrix} \underset{2\times 2}{A'A} & \underset{2\times 1}{D} \\ \underset{1\times 2}{D'} & \underset{1\times 1}{0} \end{bmatrix} \begin{bmatrix} \underset{2\times 1}{\widehat{\Delta x}} \\ \underset{1\times 1}{\hat{\lambda}} \end{bmatrix} = \begin{bmatrix} \underset{2\times 1}{A'\Delta y} \\ \underset{1\times 1}{c} \end{bmatrix}$$

$\Rightarrow$ Iterative least-squares adjustment (updating approximate values a_0, b_0 until convergence is achieved) in order to eliminate (neglected) higher order terms in the inconsistencies

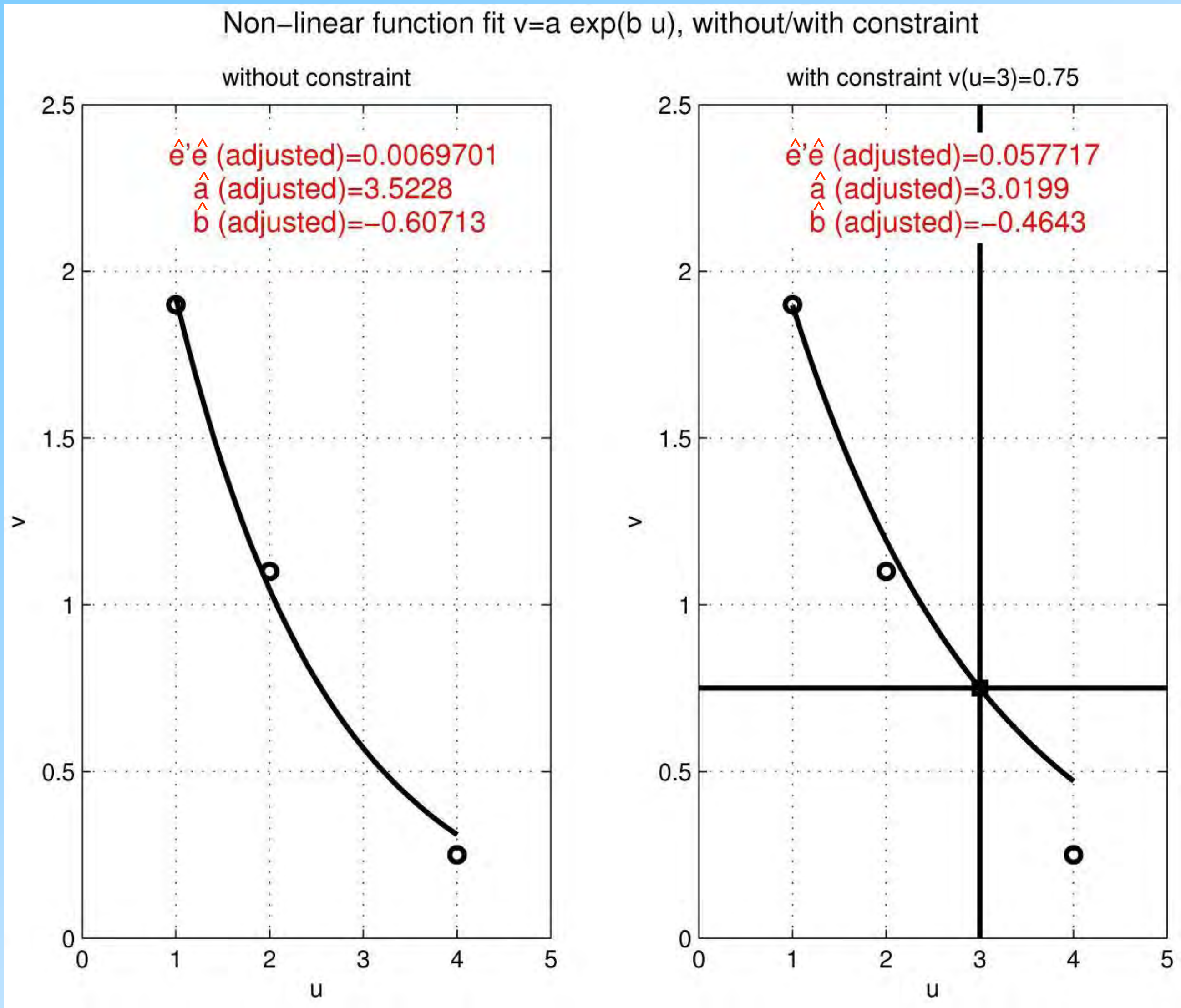
Checks:

1) Orthogonality check: $\hat{e}'\hat{e} + \Delta y' A \widehat{\Delta x} + c'\hat{\lambda} - \Delta y'\Delta y = 0$

2) Main check: $v - \hat{e} - \hat{a}e^{\hat{b}u} = 0 \quad \forall u, v, \tilde{u}, \tilde{v}$

→ Examples: Non-linear regression without/with constraint(s)

Examples: Non-linear regression without/with constraint(s)

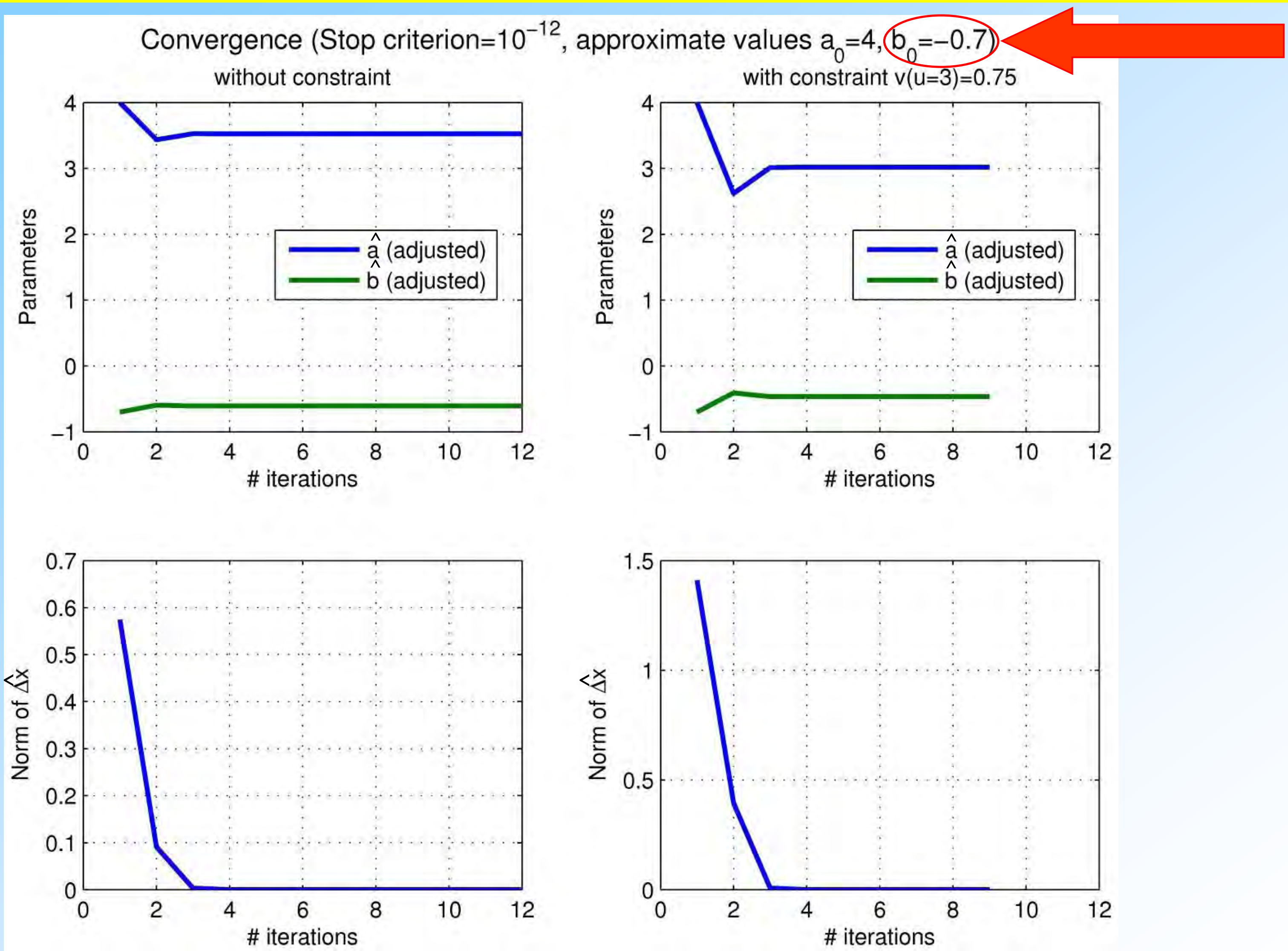


MATLAB Code: see webpage "Non-linear function fit"

NE

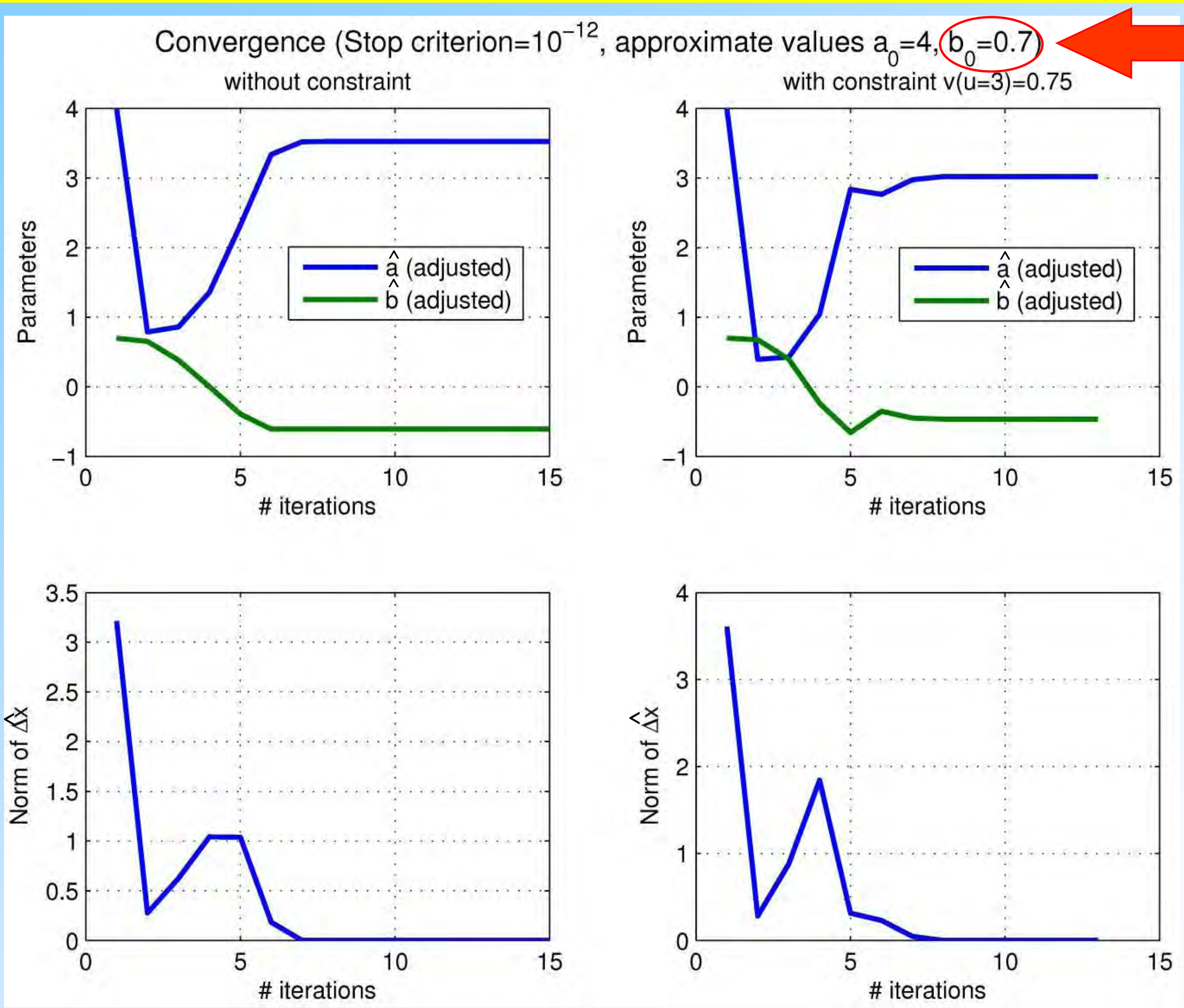
→ Examples: Non-linear regression without/with constraint(s)

Examples: Non-linear regression without/with constraint(s)



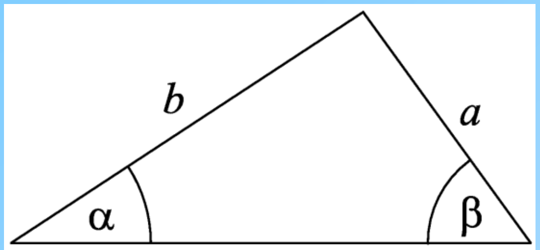
→ Examples: Non-linear regression without/with constraint(s)

Examples: Non-linear regression without/with constraint(s)



→ Linearization in the B-model

Linearization in the B-model



$$\frac{(a-e_a)}{\sin(\alpha-e_\alpha)} = \frac{(b-e_b)}{\sin(\beta-e_\beta)}$$

$$f(e_a, e_b, e_\alpha, e_\beta) = (a-e_a)\sin(\beta-e_\beta) - (b-e_b)\sin(\alpha-e_\alpha) = 0$$

Initial Taylor point: $e^0 = [e_a^0, e_b^0, e_\alpha^0, e_\beta^0]' = [0, 0, 0°, 0°]'$

$$\begin{aligned}
f(e) &= f(e^0) + \left.\frac{\partial f}{\partial e_a}\right|_0 (e_a - e_a^0) + \left.\frac{\partial f}{\partial e_b}\right|_0 (e_b - e_b^0) + \left.\frac{\partial f}{\partial e_\alpha}\right|_0 (e_\alpha - e_\alpha^0) + \left.\frac{\partial f}{\partial e_\beta}\right|_0 (e_\beta - e_\beta^0) \\
&= f(e^0) + \left.\frac{\partial f}{\partial e_a}\right|_0 e_a + \ldots + \left.\frac{\partial f}{\partial e_\beta}\right|_0 e_\beta - \left.\frac{\partial f}{\partial e_a}\right|_0 e_a^0 - \ldots - \left.\frac{\partial f}{\partial e_\beta}\right|_0 e_\beta^0 \\
&= f(e^0) + \left[-\frac{\partial f}{\partial e_a}, -\frac{\partial f}{\partial e_b}, -\frac{\partial f}{\partial e_\alpha}, -\frac{\partial f}{\partial e_\beta}\right]_0 \begin{bmatrix} e_a^0 \\ e_b^0 \\ e_\varepsilon^0 \\ e_\beta^0 \end{bmatrix} - \left[-\frac{\partial f}{\partial e_a}, -\frac{\partial f}{\partial e_b}, -\frac{\partial f}{\partial e_\alpha}, -\frac{\partial f}{\partial e_\beta}\right]_0 \begin{bmatrix} e_a \\ e_b \\ e_\alpha \\ e_\beta \end{bmatrix} \\
&= f(e^0) + \qquad B' \qquad e^0 - \qquad B' \qquad e \\
&= w - B'e = 0 \quad \text{with } w := f(e^0) + B'e^0
\end{aligned}$$

→ Mixed model (Gauß-Helmert model, Total Least Squares)

Mixed model (Gauß-Helmert model, Total Least Squares)

Quite often it is not really justified to consider only inconsistencies in the "left hand side y". For example, in function fitting problems – one example is the straight line fit – the values x_i where measurements y_i are taken may also be corrupted by errors. We then have the situation to find inconsistencies e_x and e_y in both x and y according to the model $y_i = a + b(x_i - e_{x_i}) + e_{y_i}, \quad i = 1,\ldots,m$.
Obviously, unknown parameters a and b exists as well as unknown inconsistencies e_x and e_y, and b and e_x are linked in a non-linear fashion, in addition.

Two different approaches can be used in order to solve the problem, i.e. in order to minimize the square sum of <u>all</u> inconsistencies, and – at the same time – determine both a and b.

The first approach is to reformulate the above equation as a (non-linear) condition equation with unknowns, i.e. $f_i(a, b, e_{x_i}, e_{y_i}) = y_i - e_{y_i} - [a + b(x_i - e_{x_i})] = 0$, and to linearize it. This leads to the "general model of adjustment", also called "mixed model" (because of a mixture of unknown parameters – non-stochastic quantities – and inconsistencies – stochastic quantities –), or "Gauß-Helmert-model". Robert Friedrich Helmert (1843-1917) was a famous German geodesist.

→ Mixed model (Gauß-Helmert model, Total Least Squares)

Mixed model (Gauß-Helmert model, Total Least Squares)

Introduce (initial) approximate values $a_0, b_0, e_{x_i}^0 = 0, e_{y_i}^0 = 0 \quad \forall i = 1,\ldots,m$ and compute the Taylor series expansion up to the linear term:

$$f_i(a, b, e_{x_i}, e_{y_i}) =$$

$$= f_i(a_0, b_0, e_{x_i}^0, e_{y_i}^0) + \left.\frac{\partial f_i}{\partial a}\right|_0 (a - a_0) + \left.\frac{\partial f_i}{\partial b}\right|_0 (b - b_0) + \left.\frac{\partial f_i}{\partial e_{x_i}}\right|_0 (e_{x_i} - e_{x_i}^0) + \left.\frac{\partial f_i}{\partial e_{y_i}}\right|_0 (e_{y_i} - e_{y_i}^0)$$

$$= f_i(a_0, b_0, e_{x_i}^0, e_{y_i}^0) - \left.\frac{\partial f_i}{\partial e_{x_i}}\right|_0 e_{x_i}^0 - \left.\frac{\partial f_i}{\partial e_{y_i}}\right|_0 e_{y_i}^0 + \left.\frac{\partial f_i}{\partial a}\right|_0 \Delta a + \left.\frac{\partial f_i}{\partial b}\right|_0 \Delta b + \left.\frac{\partial f_i}{\partial e_{x_i}}\right|_0 e_{x_i} + \left.\frac{\partial f_i}{\partial e_{y_i}}\right|_0 e_{y_i}$$

$$= \quad w_i \quad -\Delta a - (x_i - e_{x_i}^0)\Delta b + b_0 e_{x_i} - e_{y_i}$$

$$= w_i + \begin{bmatrix} -1 & -(x_i - e_{x_i}^0) \end{bmatrix} \begin{bmatrix} \Delta a \\ \Delta b \end{bmatrix} + \begin{bmatrix} b_0 & -1 \end{bmatrix} \begin{bmatrix} e_{x_i} \\ e_{y_i} \end{bmatrix}$$

$$= w_i + \quad A_i \quad \begin{bmatrix} \Delta a \\ \Delta b \end{bmatrix} + \quad B_i' \quad \begin{bmatrix} e_{x_i} \\ e_{y_i} \end{bmatrix} = 0 \quad , \quad w_i = y_i - (a_0 + b_0 x_i)$$

NE

→ Mixed model (Gauß-Helmert model, Total Least Squares)

Mixed model (Gauß-Helmert model, Total Least Squares)

Target function $\mathcal{L}(e,\Delta\xi,\lambda) = \frac{1}{2}\underset{1\times1}{e'e} + \underset{1\times m}{\lambda'}(\underset{m\times1}{w} + \underset{m\times n}{A}\underset{n\times1}{\Delta\xi} + \underset{m\times t}{B'}\underset{t\times1}{e}) \rightarrow \underset{e,\Delta\xi,\lambda}{\min} \quad (n=2, t=2m)$

$$\underset{m\times t}{B'} = \left[\underbrace{\begin{matrix} b_0 & 0 & \cdots & 0 \\ 0 & b_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & b_0 \end{matrix}}_{m\times m} \; \underbrace{\begin{matrix} -1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -1 \end{matrix}}_{m\times m}\right]$$

$$\underset{m\times n}{A} = -\begin{bmatrix} 1 & x_1 - e^0_{x_1} \\ \vdots & \vdots \\ 1 & x_m - e^0_{x_m} \end{bmatrix}, \quad \underset{n\times 2}{\Delta\xi} = \begin{bmatrix} \Delta a \\ \Delta b \end{bmatrix}$$

$$\underset{t\times1}{e} = [\underbrace{e_{x_1} \ldots e_{x_m}}_{e'_x} \quad \underbrace{e_{y_1} \ldots e_{y_m}}_{e'_y}]' = [e'_x \quad e'_y]'$$

$$\underbrace{\begin{bmatrix} \underset{t\times t}{I} & \underset{t\times n}{0} & \underset{t\times m}{B} \\ \underset{n\times t}{0} & \underset{n\times n}{0} & \underset{n\times m}{A'} \\ \underset{m\times t}{B'} & \underset{m\times n}{A} & \underset{m\times m}{0} \end{bmatrix}}_{t+m+n\times t+m+n} \underbrace{\begin{bmatrix} \underset{t\times1}{\hat{e}} \\ \underset{n\times1}{\widehat{\Delta\xi}} \\ \underset{m\times1}{\hat{\lambda}} \end{bmatrix}}_{t+m+n\times1} = \underbrace{\begin{bmatrix} \underset{t\times1}{0} \\ \underset{n\times1}{0} \\ \underset{m\times1}{-w} \end{bmatrix}}_{t+m+n\times1} \xRightarrow[\|\hat{e}-e_0\|<\varepsilon,\ \|\widehat{\Delta\xi}\|<\varepsilon]{\text{Iteration needed until}} \begin{bmatrix} \underset{t\times1}{\hat{e}} \\ \underset{n\times1}{\widehat{\Delta\xi}} \\ \underset{m\times1}{\hat{\lambda}} \end{bmatrix} = \begin{bmatrix} \underset{t\times t}{I} & \underset{t\times n}{0} & \underset{t\times m}{B} \\ \underset{n\times t}{0} & \underset{n\times n}{0} & \underset{n\times m}{A'} \\ \underset{m\times t}{B'} & \underset{m\times n}{A} & \underset{m\times m}{0} \end{bmatrix}^{-1} \begin{bmatrix} \underset{t\times1}{0} \\ \underset{n\times1}{0} \\ \underset{m\times1}{-w} \end{bmatrix}$$

→ Mixed model (Gauß-Helmert model, Total Least Squares)

Mixed model (Gauß-Helmert model, Total Least Squares)

The system of equations to be solved is not only sparse but also has dimension $t+m+n \times t+m+n$, which, in turn, can be reduced to dimension $m+n \times m+n$ if $\hat{e}$ is eliminated. From the equations

$$\left.\begin{array}{lll} \hat{e}+B\hat{\lambda}=0 & \Rightarrow - & B'\hat{e}-B'B\hat{\lambda}=0 \\ B'\hat{e}+A\widehat{\Delta\xi}+w=0 & \Rightarrow & B'\hat{e}+A\widehat{\Delta\xi}=-w \end{array}\right\} \Rightarrow B'B\hat{\lambda}-A\widehat{\Delta\xi}=w$$

$$A'\hat{\lambda}=0 \qquad\qquad -A'\hat{\lambda}=0$$

we get $$\Rightarrow \underbrace{\begin{bmatrix} \underset{m\times m}{B'B} & \underset{m\times n}{-A} \\ \underset{n\times m}{-A'} & \underset{n\times n}{0} \end{bmatrix}}_{m+n\times m+n} \underbrace{\begin{bmatrix} \underset{m\times 1}{\hat{\lambda}} \\ \underset{n\times 1}{\widehat{\Delta\xi}} \end{bmatrix}}_{m+n\times 1} = \underbrace{\begin{bmatrix} \underset{m\times 1}{w} \\ \underset{n\times 1}{0} \end{bmatrix}}_{m+n\times 1}$$

$$\widehat{\Delta\xi}=-[A'(B'B)^{-1}A]^{-1}A'(B'B)^{-1}w$$

$$\hat{\lambda}=(B'B)^{-1}\{w-A[A'(B'B)^{-1}A]^{-1}A'(B'B)^{-1}w\}$$

$$\hat{e}=-B(B'B)^{-1}\{w-A[A'(B'B)^{-1}A]^{-1}A'(B'B)^{-1}w\}$$

The redundancy is $r=m-2.$

→ Mixed model (Gauß-Helmert model, Total Least Squares): Example 1

Mixed model (Gauß-Helmert model): Example 1

i	1	2	3	4	5	6	7
x_i [m]	–1	0	1	2	3	4	5
y_i [m]	1.3	0.8	0.9	1.2	2.0	3.5	4.1

$\Rightarrow$

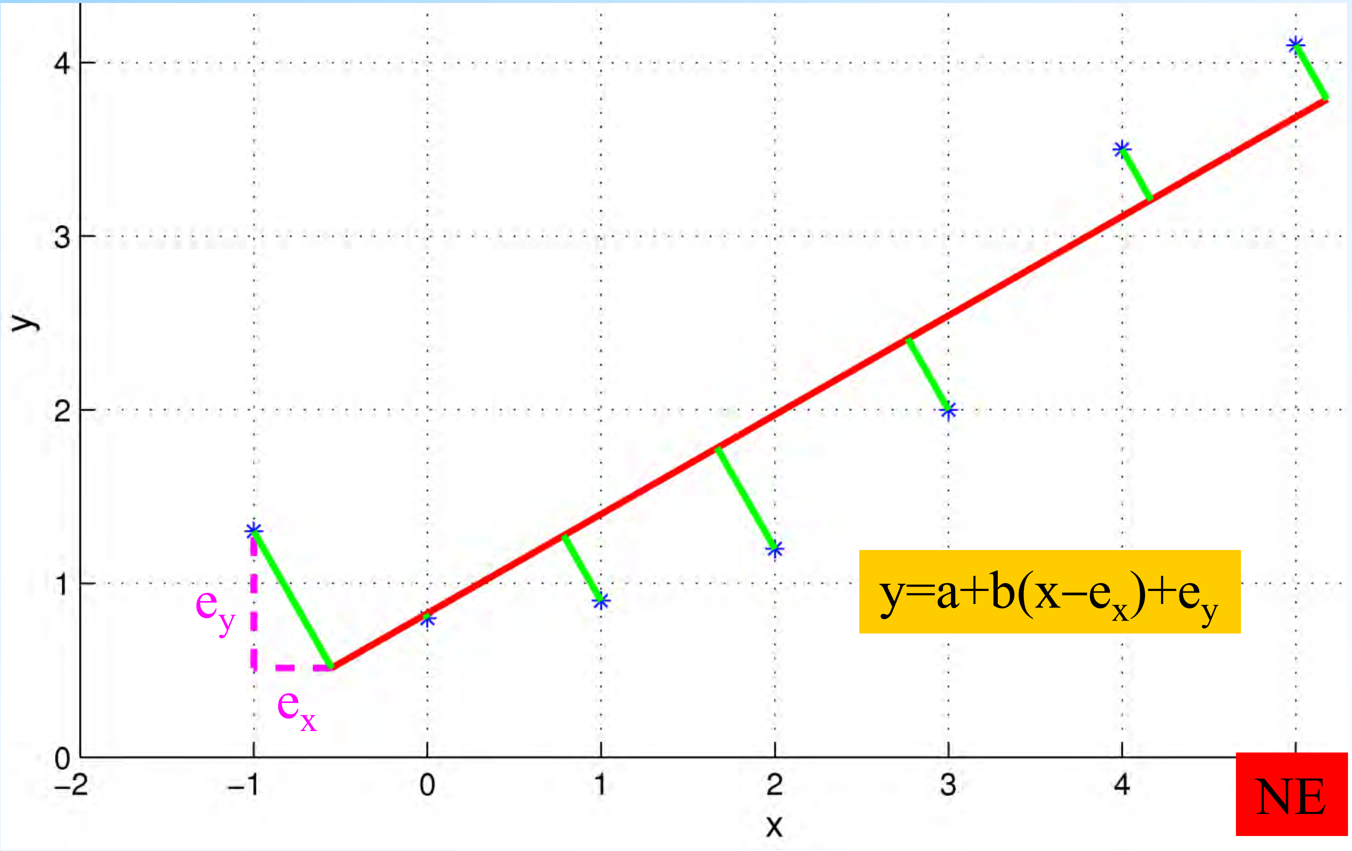
$$\hat{a} = 0.829\,[\mathrm{m}]$$
$$\hat{b} = 0.571\,[-]$$

Remark: Closed ODF solution exists!

$\hat{e}_{y_i}$ [m]	$\hat{e}_{x_i}$ [m]
0.786	–0.449
–0.022	0.012
–0.377	0.215
–0.582	0.332
–0.409	0.234
0.291	–0.166
0.313	–0.179

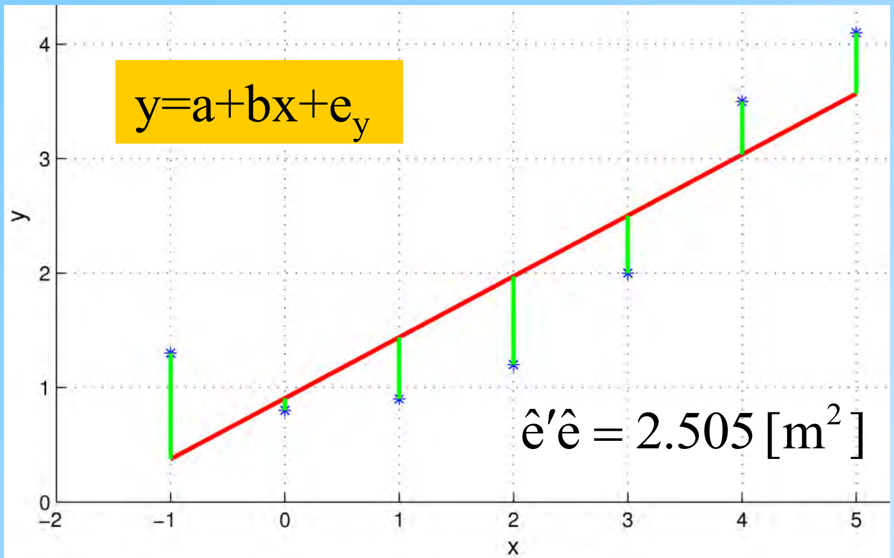
$\Downarrow$

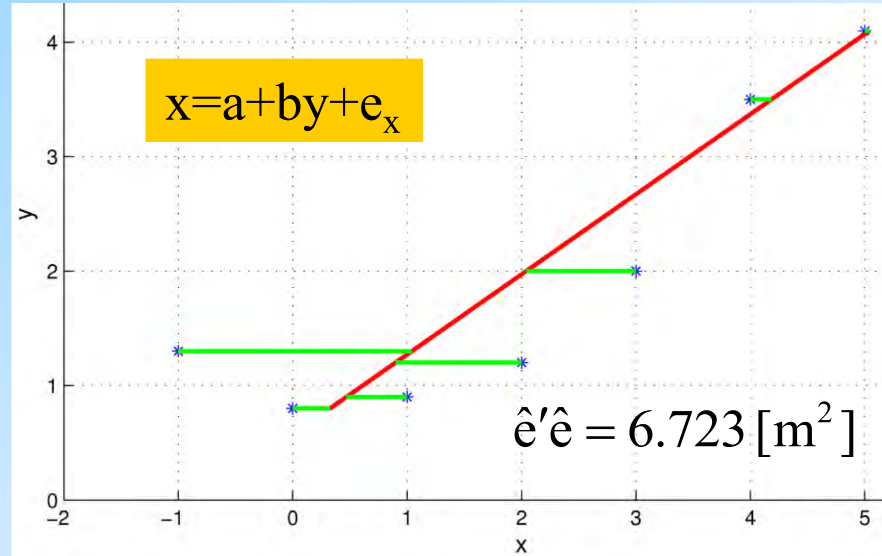
$$\hat{e}'\hat{e} = 1.921\,[\mathrm{m}^2]$$

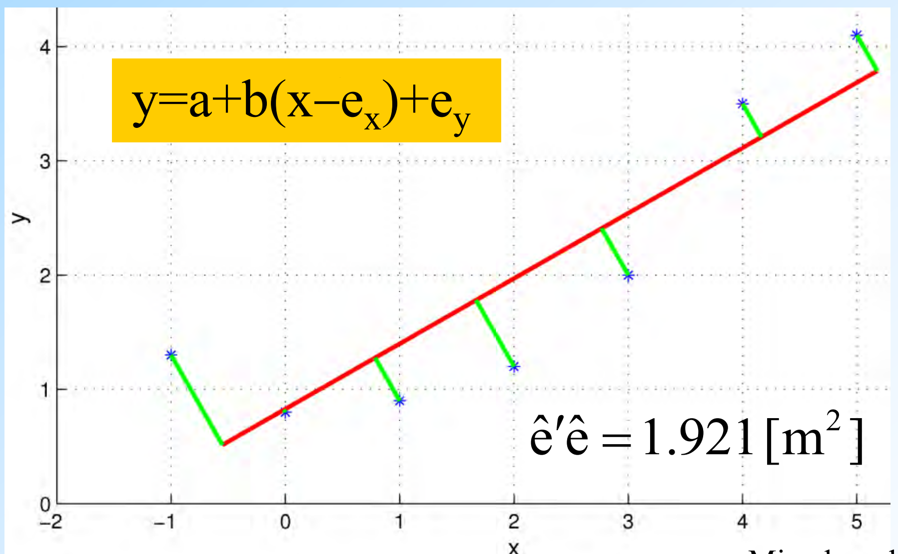


→ Comparison line fits

Comparison line fits







NE

→ Mixed model (Gauß-Helmert model): Example 2

Mixed model (Gauß-Helmert model): Example 2

i	1	2	3	4	5	6	7
x_i [m]	–1	0	1	2	3	4	5
y_i [m]	1.3	0.8	0.9	1.2	2.0	3.5	4.1
z_i [m]	5.3	1.8	2.5	2.4	3.7	5.2	7.0

Remark: Closed ODF solution exists!

Adjustment model

$$n_x(x-e_x)+n_y(y-e_y)+n_z(z-e_z)+d=0$$

$$n_x^2+n_y^2+n_z^2=1$$

$$\Downarrow$$

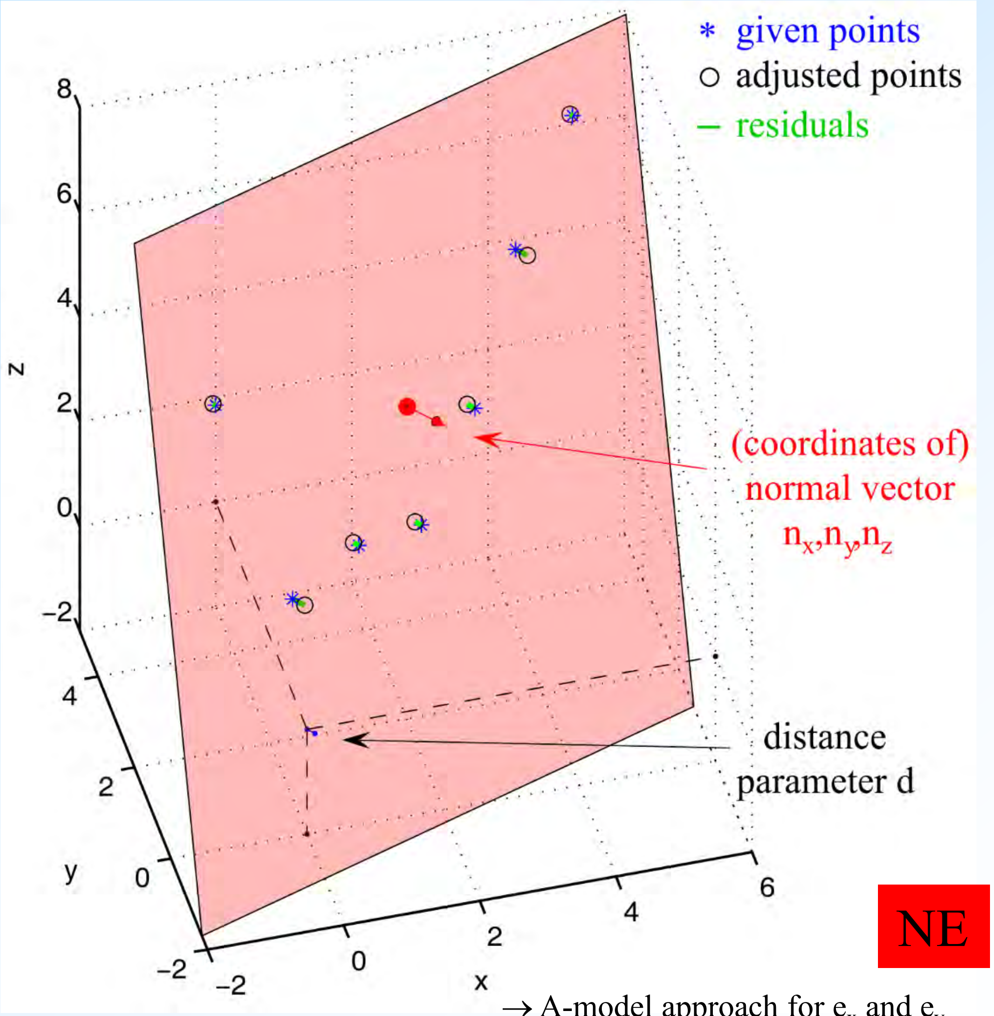
$$\hat{n}_x = 0.322 \pm 0.059$$

$$\hat{n}_y = -0.888 \pm 0.018$$

$$\hat{n}_z = 0.330 \pm 0.065$$

$$\hat{d} = -0.207\,[\mathrm{m}] \pm 0.252\,[\mathrm{m}]$$

$$\hat{e}'\hat{e} = 0.299\,[\mathrm{m}^2]$$



→ A-model approach for e_x and e_y

A-model approach for e_x and e_y

The second approach for processing simultaneously inconsistencies e_x and e_y is to introduce so called pseudo-observations, which will lead to the standard A-model. Starting from $y_i = a + b(x_i - e_{x_i}) + e_{y_i}, i = 1, \ldots, m,$ the (unknown) quantities $x_i - e_{x_i}$ are substituted by new (unknown) quantities $\bar{x}_i = x_i - e_{x_i}$, and the pseudo-observation equations $x_i = \bar{x}_i + e_{x_i}$ are added to the problem; instead of m equations there are now 2m observation equations:

$$\left.\begin{aligned} y_i &= a + b\bar{x}_i + e_{y_i} \\ x_i &= \bar{x}_i + e_{x_i} \end{aligned}\right. \quad i = 1, \ldots, m$$

As slope b and "coordinates" $\bar{x}_i$ are non-linearly connected, approximate quantities for a, b and $\bar{x}_i$, i.e. a_0, b_0 and $\bar{x}_i^0$ are introduced and linearization can start. As initial approximate values for $\bar{x}_i^0$ original observation values x_i are used.

$$y_i = a_0 + b_0\bar{x}_i^0 + \Delta a + \bar{x}_i^0\Delta b + b_0\Delta\bar{x}_i + e_{y_i} \Rightarrow \underbrace{y_i - (a_0 + b_0\bar{x}_i^0)}_{\Delta y_i} = \Delta a + \bar{x}_i^0\Delta b + b_0\,\Delta\bar{x}_i + e_{y_i}$$

$$x_i = \bar{x}_i^0 + \Delta\bar{x}_i + e_{x_i} \Rightarrow \underbrace{x_i - \bar{x}_i^0}_{\Delta x_i} = \Delta\bar{x}_i + e_{x_i}$$

NE

→ A-model approach for e_x and e_y

A-model approach for e_x and e_y

The final step is to establish the A-model equations (n=2) and to solve the least squares problem iteratively as usual.

$$\underset{m\times 1}{\Delta y} = \begin{bmatrix} y_1 - (a_0 + b_0 \bar{x}_1^0) \\ \vdots \\ y_m - (a_0 + b_0 \bar{x}_m^0) \end{bmatrix}, \underset{m\times 1}{\Delta x} = \begin{bmatrix} x_1 - \bar{x}_1^0 \\ \vdots \\ x_m - \bar{x}_m^0 \end{bmatrix}, \underset{m\times 1}{\Delta \bar{x}} = \begin{bmatrix} \Delta \bar{x}_1 \\ \vdots \\ \Delta \bar{x}_m \end{bmatrix}, \underset{m\times 1}{e_y} = \begin{bmatrix} e_{y_1} \\ \vdots \\ e_{y_m} \end{bmatrix}, \underset{m\times 1}{e_x} = \begin{bmatrix} e_{x_1} \\ \vdots \\ e_{x_m} \end{bmatrix}$$

$$\underset{2m\times 1}{\begin{bmatrix} \Delta y \\ \Delta x \end{bmatrix}} = \underset{2m\times m+n}{\begin{bmatrix} A_1 \\ A_2 \end{bmatrix}} \underset{m+n\times 1}{\begin{bmatrix} \Delta a \\ \Delta b \\ \Delta \bar{x} \end{bmatrix}} + \underset{2m\times 1}{\begin{bmatrix} e_y \\ e_x \end{bmatrix}} \quad (\sim y = Ax + e)$$

$$\underset{m\times m+n}{A_1} = \begin{bmatrix} 1 & \bar{x}_1^0 & b_0 & 0 & \cdots & 0 \\ 1 & \bar{x}_2^0 & 0 & b_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \bar{x}_m^0 & 0 & 0 & \cdots & b_0 \end{bmatrix}, \underset{m\times m+n}{A_2} = \begin{bmatrix} 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}$$

NE

→ Weighted Least Squares

Weighted Least Squares

Weighted least squares deals with the question how to work with measurements of different quality, i.e. how to take into account observations which originate from instruments of different quality categories.

Answer: Different observations are assigned different weights; precise observations obtain higher weights as compared to those of low precision. Weights are arranged in a (diagonal and positive definite) m×m weight matrix

$$\underset{m\times m}{P} = \begin{bmatrix} p_{11} & & & 0 \\ & p_{22} & & \\ & & \ddots & \\ 0 & & & p_{mm} \end{bmatrix}$$

metric (matrix) of the observation space

and the "old" target function with $P=I_m$, i.e. $\mathcal{L}_A(x) = \frac{1}{2} e'e = \frac{1}{2} e' \, I \, e$

is replaced by the "new" target function $\mathcal{L}_A(x) = \frac{1}{2} e' \, P \, e$.

The letter P for the weight matrix originates from the Latin word "pondus".

→ Weighted Least Squares

Weighted Least Squares

$$\min_x \mathcal{L}_A(x) = \min_x \frac{1}{2} e'Pe = \frac{1}{2}\min_x e'Pe = \frac{1}{2}\min_x (y - Ax)'P(y - Ax)$$

$$= \frac{1}{2}\min_x (y' - x'A')P(y - Ax) =$$

$$= \frac{1}{2}\min_x (y'Py - \underbrace{y'PAx}_{=x'A'Py} - x'A'Py + x'A'PAx)$$

$$= \frac{1}{2}\min_x (y'Py - 2x'A'Py + x'A'PAx)$$

Necessary condition for a minimum:

$$\boxed{\frac{\partial \mathcal{L}_A(x)}{\partial x}(\hat{x}) = -A'Py + A'PA\hat{x} \overset{!}{=} 0}$$

Sufficient condition for a minimum:

$$\frac{\partial^2 \mathcal{L}_A(x)}{\partial x^2}(\hat{x}) = A'PA > 0 \quad \checkmark$$

→ Weighted Least Squares

Weighted Least Squares

$\Rightarrow$	$\hat{x} = (A'PA)^{-1} A'Py$	Weighted LS-estimate of x
$\Rightarrow$	$\hat{y} = A\hat{x} = A(A'PA)^{-1} A'Py = P_{A,(PA)^\perp} y = P_A y$	Weighted LS-estimate of y
$\Rightarrow$	$\hat{e} = y - \hat{y} = [I - P_{A,(PA)^\perp}]y = P_{(PA)^\perp,A} y = P_A^\perp y$	Weighted LS-estimate of e

with orthogonality check $A'P\hat{e} = 0.$

In the case of condition adjustment we have the (new) target function

$$\min_{e,\lambda} \mathcal{L}_B(e,\lambda) = \min_{e,\lambda} \left[\frac{1}{2} e'Pe + \lambda'(B'y - B'e) \right] = \min_{e,\lambda} \left[\frac{1}{2} e'Pe + \lambda'(w - B'e) \right]$$

leading to the normal equations/solutions/check

$$\begin{aligned} P\hat{e} - B\hat{\lambda} &= 0 \\ B'y - B'\hat{e} &= 0 \end{aligned} \quad \Rightarrow \quad \begin{aligned} &\hat{e} = P^{-1}B\hat{\lambda} = P^{-1}B(B'P^{-1}B)^{-1} w \\ &\hat{\lambda} = (B'P^{-1}B)^{-1} w, \quad \hat{y} = y - \hat{e} \end{aligned}$$

$$\hat{e}'P\hat{e} - w'\hat{\lambda} = 0$$

→ Weighted Least Squares

Weighted Least Squares

Every LS-problem with weight matrix $\mathrm{P} \neq \mathrm{I}$ can be transformed into an unweighted LS-problem. This transformation is called homogenization.
Start from $\tilde{y} = \tilde{A}\tilde{x} + \tilde{e}$ with weight matrix $\tilde{P} \neq I$. Then, the least-squares estimate for $\tilde{x}$ is

$$\hat{\tilde{x}} = (\tilde{A}'\tilde{P}\tilde{A})^{-1}\tilde{A}'\tilde{P}y.$$

For the reason that the weight matrix was supposed to be positive definite, it can be decomposed into the product of two matrices G, i.e. $\tilde{P} = G\,G'$ (Cholesky decomposition). G is lower triangular. If $\tilde{P}$ is diagonal, G is nothing else but

$$G = \sqrt{\tilde{P}} = \mathrm{diag}(\sqrt{\tilde{p}_{ii}}) \;,\; i = 1,\ldots,m$$

Proof: Put $A = G'\tilde{A}$, $y = G'\tilde{y}$, $e = G'\tilde{e}$ and compute the traditional unweighted LS-quantities, e.g. $\hat{x} = (A'A)^{-1}A'y$ and substitute A, y and e

$$\boxed{\hat{x} = (A'\underbrace{G\,G'}_{\tilde{P}}A)^{-1}A'\underbrace{G\,G'}_{\tilde{P}}y = (A'\tilde{P}A)^{-1}A'\tilde{P}y = \hat{\tilde{x}}}$$

MATLAB: G=chol($\tilde{P}$,'lower');

→ Weighted Least Squares: Example

Weighted Least Squares: Example

Given m=2 inconsistent observations, y_1 and y_2, for one unknown x (n=1) in the model y=x (direct observations) and 2 weights $p_1 \equiv p_{11}$, $p_2 \equiv p_{22}$, we have

$$\underset{m\times 1}{\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}} = \begin{bmatrix} x \\ x \end{bmatrix} = \underset{m\times n}{\begin{bmatrix} 1 \\ 1 \end{bmatrix}} x + \begin{bmatrix} e_{y_1} \\ e_{y_2} \end{bmatrix} = Ax + e, \quad \underset{m\times n}{A} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \underset{m\times m}{P} = \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix}$$

$$A'P = \begin{bmatrix} p_1 & p_2 \end{bmatrix} \qquad A'PA = p_1 + p_2 \qquad (A'PA)^{-1} = \frac{1}{p_1 + p_2}$$

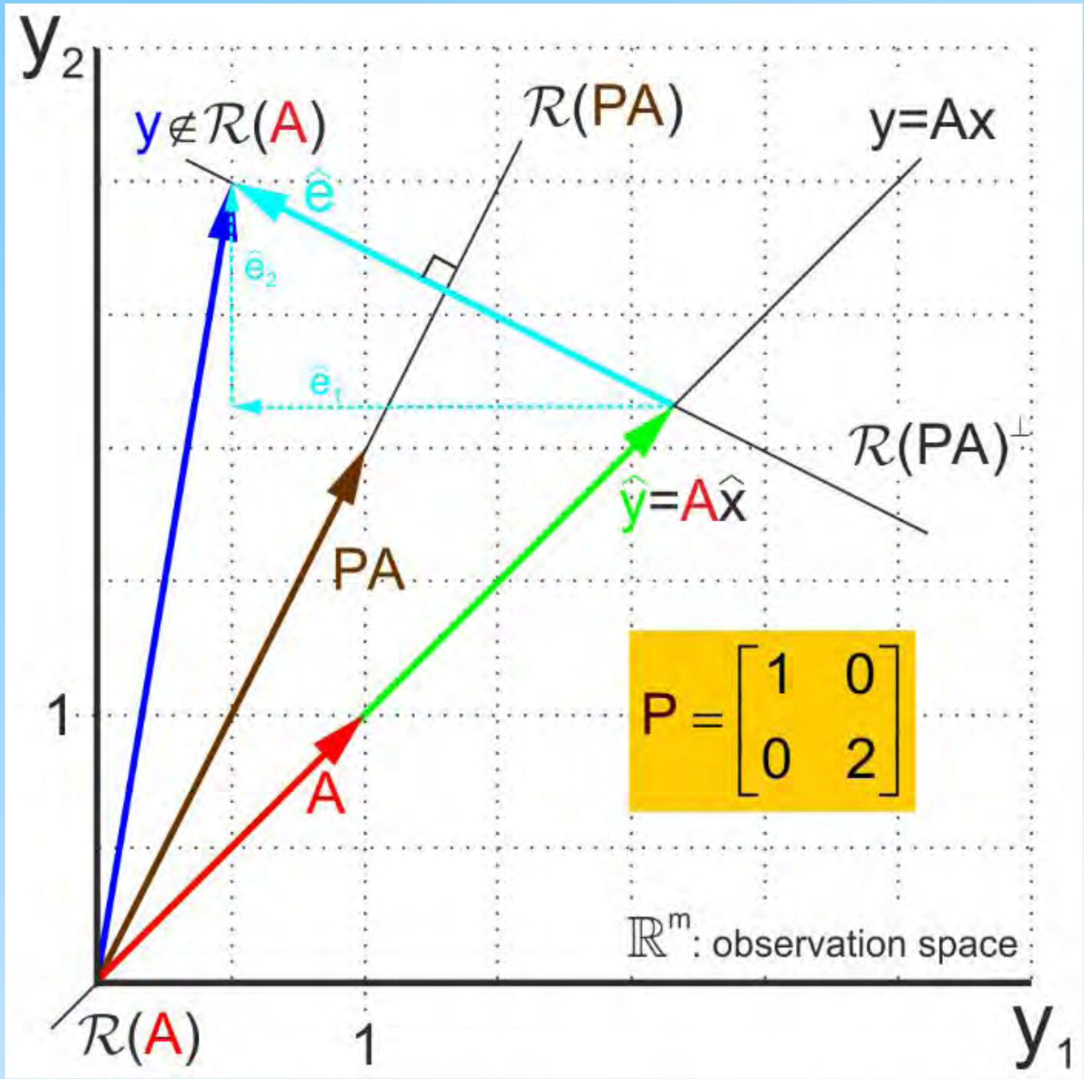
$$A'Py = p_1 y_1 + p_2 y_2 \qquad \boxed{\hat{x} = \frac{p_1 y_1 + p_2 y_2}{p_1 + p_2} \quad \text{"P-weighted average"}}$$

$$\hat{y} = A\hat{x} = \frac{p_1 y_1 + p_2 y_2}{p_1 + p_2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\hat{e} = y - \hat{y} = \begin{bmatrix} y_1 - \dfrac{p_1 y_1 + p_2 y_2}{p_1 + p_2} \\ y_2 - \dfrac{p_1 y_1 + p_2 y_2}{p_1 + p_2} \end{bmatrix} = \begin{bmatrix} \dfrac{p_2 (y_1 - y_2)}{p_1 + p_2} \\ -\dfrac{p_1 (y_1 - y_2)}{p_1 + p_2} \end{bmatrix}$$

→ Weighted Least Squares: Example

Weighted Least Squares: Example



Assume that y_2 results from observing x using a good quality instrument and y_1 from a less precise instrument. Then y_2 should obtain more importance as compared to y_1, i.e. $p_2 > p_1$! The minimization process now tries harder to make e_2 smaller than e_1, because y_2 should get a smaller correction. And indeed

$$|\hat{e}_2| = p_1 \frac{|-(y_1 - y_2)|}{p_1 + p_2} <$$

$$|\hat{e}_1| = p_2 \frac{|y_1 - y_2|}{p_1 + p_2}$$

because $p_2 > p_1$ or $p_1 < p_2$

→ Weighted Least Squares: Example straight line fit, mixed model

Weighted Least Squares: Example straight line fit, mixed model

i	1	2	3	4	5	6	7
x_i [m]	–1	0	1	2	3	4	5
y_i [m]	1.3	0.8	0.9	1.2	2.0	3.5	4.1
p_x	3	9	8	4	5	7	10
p_y	2	8	7	5	10	8	6

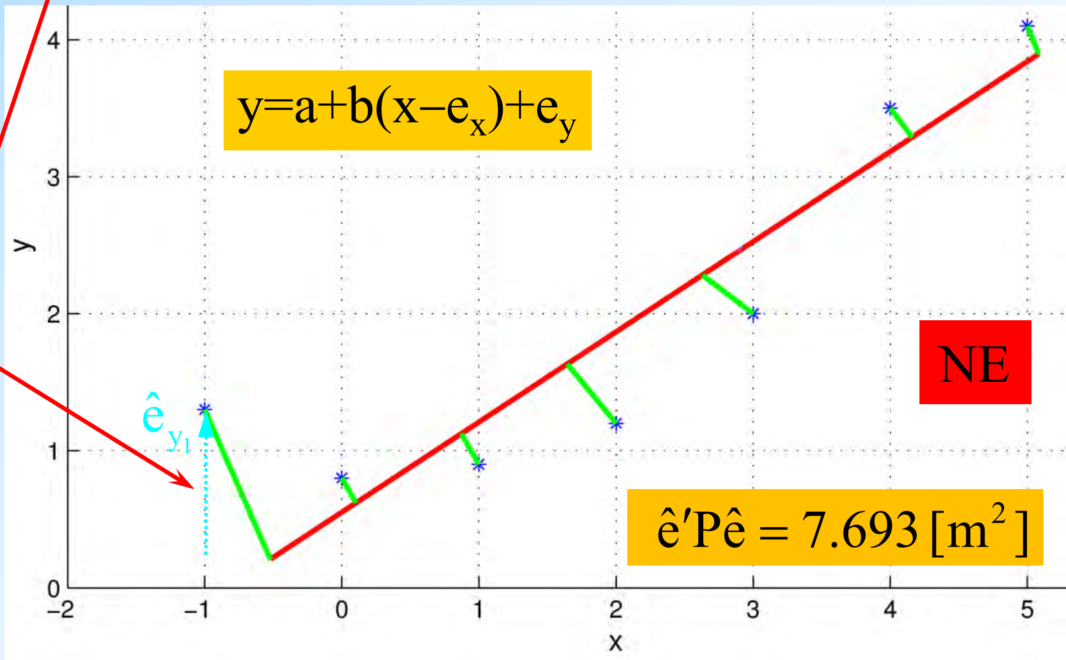
same as before (x_i, y_i)

High (low) weights make small (big) absolute $\hat{e}$'s (p_x, p_y)

$\hat{e}_{y_i}$ [m]	$\hat{e}_{x_i}$ [m]
1.092	–0.479
0.180	–0.105
–0.224	0.129
–0.433	0.356
–0.282	0.370
0.212	–0.159
0.205	–0.081

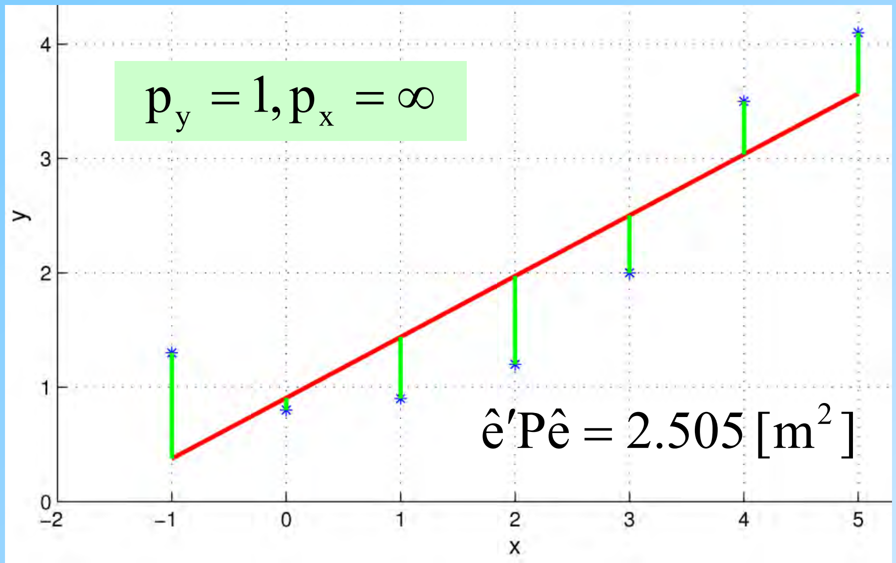
$\hat{a} = 0.55$

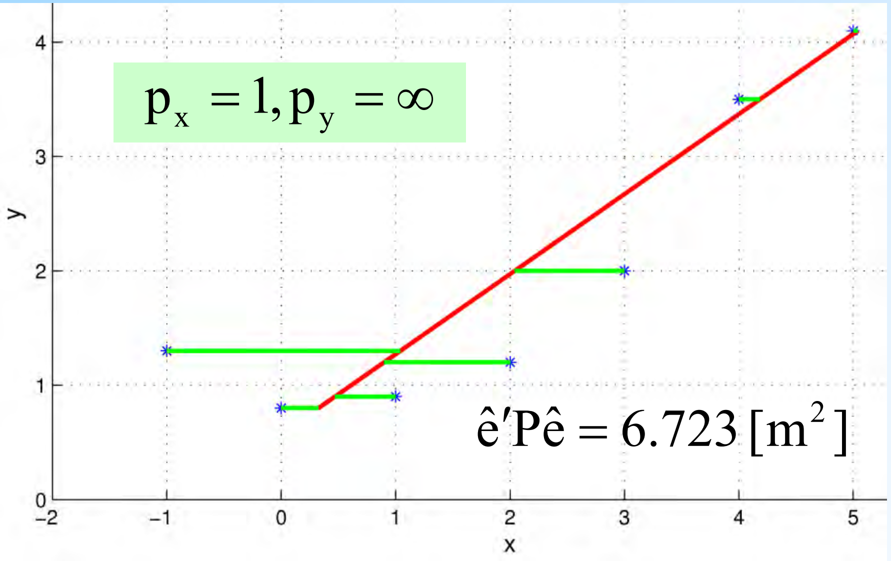
$\hat{b} = 0.66$

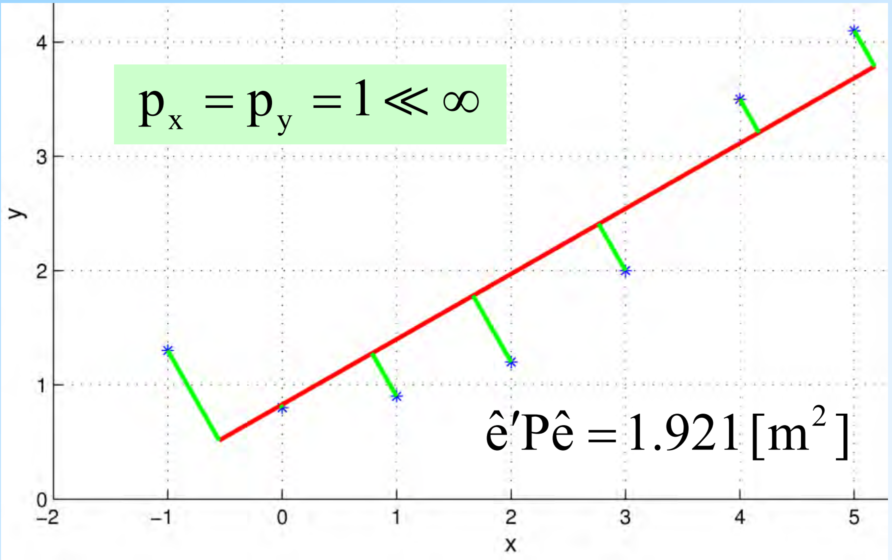


→ Comparison line fits

Comparison line fits







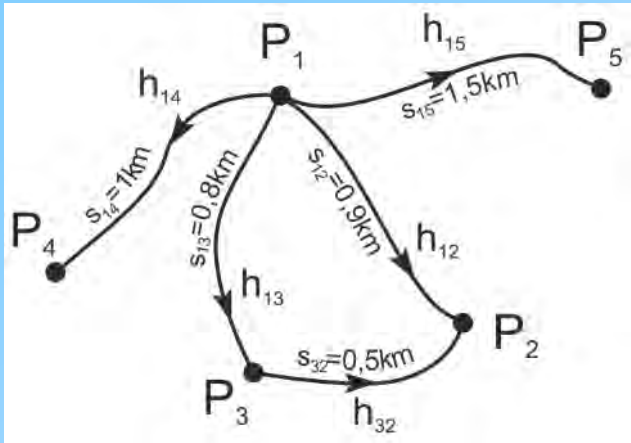
All three cases:
Mixed model
$y=a+b(x-e_x)+e_y$

NE

→ Weighted Least Squares: Example free network adjustment

Weighted Least Squares: Example free network adjustment

Free adjustment of a 5-point levelling network



$i-j$	h_{ij} [m]	p_{ij}	H^0 [m]	$h_{ij}-(H_j^0-H_i^0)$ [m]
1 – 2	14.301	1 / 0.9	$H_1^0 = 93.459$	0.001
1 – 3	9.995	1 / 0.8	$H_2^0 = 107.759$	–0.005
1 – 4	7.006	1 / 1	$H_3^0 = 103.459$	0.006
1 – 5	17.500	1 / 1.5	$H_4^0 = 100.459$	0.003
3 – 2	4.299	1 / 0.5	$H_5^0 = 110.956$	–0.001

In order to determine the absolute heights H_i, height differences h_{ij} have been observed with weights $p_{ij} = s_{ij}^{-1}$. From given approximate heights H_i^0 the reduced observation vector y is computed via $h_{ij} - (H_j^0 - H_i^0) = \Delta H_j - \Delta H_i$.

For the vector of unknowns $x = [\Delta H_1, \Delta H_2, \Delta H_3, \Delta H_4, \Delta H_5]'$ the rank deficient design matrix A is

$$A = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \\ 0 & 1 & -1 & 0 & 0 \end{bmatrix}.$$

NE

→ Weighted Least Squares: Example free network adjustment

Weighted Least Squares: Example free network adjustment

Using the constraint $D_1'\hat{x} = [1 \quad 1 \quad 1 \quad 1 \quad 1]\hat{x} = 0$ the free levelling network solution is found from the extended normal equations

$$\begin{bmatrix} A'PA & D_1 \\ D_1' & 0 \end{bmatrix}\begin{bmatrix} \hat{x} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} N & D_1 \\ D_1' & 0 \end{bmatrix}\begin{bmatrix} \hat{x} \\ \hat{\lambda} \end{bmatrix} = \begin{bmatrix} A'Py \\ 0 \end{bmatrix}$$

after two iterations
(stop criterion $\|\hat{x}\|_2 < 10^{-10}$)

$\hat{e}'P\hat{e} = 22.3\,\mathrm{mm}^2$

$\hat{x}$ [mm]	$\hat{H}$ [m]	$\hat{e}$ [mm]	$\hat{h}$ [m]
–0.9	93.4581	2.9	14.2981
–2.8	107.7562	–2.6	9.9975
–3.4	103.4556	0.0*	7.0060
5.1	100.4641	0.0*	17.5000
2.1	110.9581	–1.6	4.3006

* For the reason that points 4 and 5 are polar points their estimated residuals turn out to be always zero !

→ Weighted Least Squares: Example free network adjustment