❷ Find ONB $\{\mathbf{z}_1,\ldots,\mathbf{z}_{n-k}\}$ of $\mathrm{Ker}(\mathbf{B})$ and assemble it into a matrix $\mathbf{Z}=[\mathbf{z}_1\ldots\mathbf{z}_{n-k}]\in\mathbb{K}^{n,n-k}$

$$\|\mathbf{A}-\mathbf{B}\|_F^2 \geq \|(\mathbf{A}-\mathbf{B})\mathbf{Z}\|_F^2 = \|\mathbf{A}\mathbf{Z}\|_F^2 = \sum_{i=1}^{n-k}\|\mathbf{A}\mathbf{z}_i\|_2^2 = \sum_{i=1}^{n-k}\sum_{j=1}^{r}\sigma_j^2(\mathbf{v}_j^H\mathbf{z}_i)^2 \qquad \square$$

Note: information content of a rank-k matrix $\mathbf{M}\in\mathbb{K}^{m,n}$ is equivalent to $k(m+n)$ numbers!

Approximation by low-rank matrices $\leftrightarrow$ matrix compression

5.5 Essential Skills Learned in Chapter 5

You should know:

- complexity of the direct eigensolver eig of Matlab
- what are Krylov methods, when and how to use them
- what is the singular value decomposition and how to use it
- applications of the svd: principal component analysis, extrema of quadratic forms on the unit sphere, best low rank approximation

6 Least Squares

Example 6.0.1 (linear regression).

Given: *measured* data $y_i,\mathbf{x}_i$, $y_i\in\mathbb{R}$, $\mathbf{x}_i\in\mathbb{R}^n$, $i=1,\ldots,m$, $m\geq n+1$
(y_i, $\mathbf{x}_i$ have measurement errors).

Known: without measurement errors data would satisfy
affine linear relationship $y=\mathbf{a}^T\mathbf{x}+c$, $\mathbf{a}\in\mathbb{R}^n$, $c\in\mathbb{R}$.

Goal: *estimate* parameters $\mathbf{a}$, c.

least squares estimate

$$(\mathbf{a},c) = \operatorname*{argmin}_{\mathbf{p}\in\mathbb{R}^n,q\in\mathbb{R}}\sum_{i=1}^{m}|y_i-\mathbf{p}\mathbf{x}_i-q|^2 \qquad (6.0.1)$$

linear regression for $n=2, m=8$ ▷.

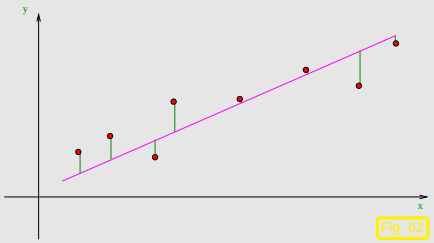


Fig. 62

Remark: In statistics we learn that the least squares estimate provides a maximum likelihood estimate, if the measurement errors are uniformly and independently normally distributed.

Example 6.0.2 (Linear data fitting). (→ Ex. 6.5.1 for a related problem)

Given: "nodes" $(t_i,y_i)\in\mathbb{K}^2$, $i=1,\ldots,m$, $t_i\in I\subset\mathbb{K}$,
basis functions $b_j: I\mapsto\mathbb{K}$, $j=1,\ldots,n$.
Find: coefficients $x_j\in\mathbb{K}$, $j=1,\ldots,n$, such that

$$\sum_{i=1}^{m}|f(t_i)-y_i|^2\to\min \quad , \quad f(t):=\sum_{j=1}^{n}x_jb_j(t)\,. \qquad (6.0.2)$$

Special case: polynomial fit: $b_j(t)=t^{j-1}$.
MATLAB-function: `p = polyfit(t,y,n);` n = polynomial degree.

◇

Remark 6.0.3 (Overdetermined linear systems).

In Ex. 6.0.1 we could try to find $\mathbf{a}$, c by solving the linear system of equations

$$\begin{pmatrix} \mathbf{x}_1^T & 1 \\ \vdots & \vdots \\ \mathbf{x}_m^T & 1 \end{pmatrix} \begin{pmatrix} \mathbf{a} \\ c \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} ,$$

but in case $m > n+1$ we encounter more equations than unknowns.

In Ex. 6.0.2 the same idea leads to the linear system

$$\begin{pmatrix} b_1(t_1) & \dots & b_n(t_1) \\ \vdots & & \vdots \\ b_1(t_m) & \dots & b_n(t_m) \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} ,$$

with the same problem in case $m > n$.

△

(Linear) least squares problem:

given: $\mathbf{A} \in \mathbb{K}^{m,n}$, $m, n \in \mathbb{N}$, $\mathbf{b} \in \mathbb{K}^m$,
find: $\mathbf{x} \in \mathbb{K}^n$ such that

$$\text{(i) } \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 = \inf\{\|\mathbf{A}\mathbf{y} - \mathbf{b}\|_2 : \mathbf{y} \in \mathbb{K}^n\}, \qquad (6.0.3)$$

(ii) $\|\mathbf{x}\|_2$ is minimal under the condition (i).

Recast as linear least squares problem, *cf.* Rem. 6.0.3:

Ex. 6.0.1: $\mathbf{A} = \begin{pmatrix} \mathbf{x}_1^T & 1 \\ \vdots & \vdots \\ \mathbf{x}_m^T & 1 \end{pmatrix} \in \mathbb{R}^{m,n+1}$, $\mathbf{b} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \in \mathbb{R}^n$, $\mathbf{x} = \begin{pmatrix} \mathbf{a} \\ c \end{pmatrix} \in \mathbb{R}^{n+1}$.

Ex. 6.0.2: $\mathbf{A} = \begin{pmatrix} b_1(t_1) & \dots & b_n(t_1) \\ \vdots & & \vdots \\ b_1(t_m) & \dots & b_n(t_m) \end{pmatrix} \in \mathbb{R}^{m,n}$, $\mathbf{b} = \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} \in \mathbb{R}^m$, $\begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$.

In both cases the residual norm $\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2$ allows to gauge the quality of the model.

Lemma 6.0.1 (Existence & uniqueness of solutions of the least squares problem)**.**
The least squares problem for $\mathbf{A} \in \mathbb{K}^{m,n}$, $\mathbf{A} \neq 0$, has a unique solution for every $\mathbf{b} \in \mathbb{K}^m$.

Proof. The proof is given by formula (6.2.4) and its derivation, see Sect. 6.2. □

MATLAB "black-box" solver for linear least squares problems:

`x = A\b` ("backslash") solves (6.0.3) for $\mathbf{A} \in \mathbb{K}^{m,n}$, $m \neq n$.

Reassuring: stable (→ Def.**??**) implementation (for dense matrices).

Remark 6.0.4 (Pseudoinverse).

By Lemma 6.0.1 the solution operator of the least squares problem (6.0.3) defines a linear mapping $\mathbf{b} \mapsto \mathbf{x}$, which has a matrix representation.

Definition 6.0.2 (Pseudoinverse). *The pseudoinverse $\mathbf{A}^+ \in \mathbb{K}^{n,m}$ of $\mathbf{A} \in \mathbb{K}^{m,n}$ is the matrix representation of the (linear) solution operator $\mathbb{R}^m \mapsto \mathbb{R}^n$, $\mathbf{b} \mapsto \mathbf{x}$ of the least squares problem* (6.0.3) $\|\mathbf{A}\mathbf{x} - \mathbf{b}\| \to \min$, $\|\mathbf{x}\| \to \min$.

MATLAB:

`P = pinv(A)` computes the pseudoinverse.

△

Remark 6.0.5 (Conditioning of the least squares problem).

6.0
p. 33

Definition 6.0.3 (Generalized condition (number) of a matrix, → Def. 2.4.6).
Let $\sigma_1 \geq \sigma_2 \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_p = 0$, $p := \min\{m, n\}$, be the singular values (→ Def. 5.4.2) of $\mathbf{A} \in \mathbb{K}^{m,n}$. Then

$$\operatorname{cond}_2(\mathbf{A}) := \frac{\sigma_1}{\sigma_r}$$

is the generalized condition (number) (w.r.t. the 2-norm) of $\mathbf{A}$.

Theorem 6.0.4. *For $m \geq n$, $\mathbf{A} \in \mathbb{K}^{m,n}$, $\operatorname{rank}(\mathbf{A}) = n$, let $\mathbf{x} \in \mathbb{K}^n$ be the solution of the least squares problem $\|\mathbf{A}\mathbf{x} - \mathbf{b}\| \to \min$ and $\widehat{\mathbf{x}}$ the solution of the perturbed least squares problem $\|(\mathbf{A} + \Delta\mathbf{A})\widehat{\mathbf{x}} - \mathbf{b}\| \to \min$. Then*

$$\frac{\|\mathbf{x} - \widehat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2} \lesssim \left(2\operatorname{cond}_2(\mathbf{A}) + \operatorname{cond}_2^2(\mathbf{A}) \frac{\|\mathbf{r}\|_2}{\|\mathbf{A}\|_2 \|\mathbf{x}\|_2} \right) \frac{\|\Delta\mathbf{A}\|_2}{\|\mathbf{A}\|_2}$$

holds, where $\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{b}$ is the residual.

This means: if $\|\mathbf{r}\|_2 \ll 1$ ➤ condition of the least squares problem $\approx \operatorname{cond}_2(\mathbf{A})$
if $\|\mathbf{r}\|_2$ "large" ➤ condition of the least squares problem $\approx \operatorname{cond}_2^2(\mathbf{A})$

6.0
p. 33

△

6.1 Normal Equations

Setting: $\mathbf{A} \in \mathbb{R}^{m,n}$, $m \geq n$, with full rank $\operatorname{rank}(\mathbf{A}) = n$.

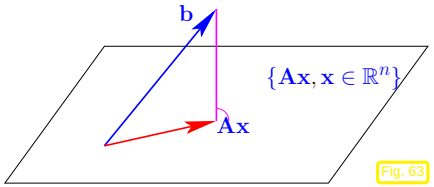


Fig. 63

Geometric interpretation of linear least squares problem (6.0.3):

$\mathbf{x} \mathrel{\hat{=}}$ orthogonal projection of $\mathbf{b}$ on the subspace $\operatorname{Im}(\mathbf{A}) := \operatorname{Span}\{(\mathbf{A})_{:,1}, \ldots, (\mathbf{A})_{:,n}\}$.

Geometric interpretation: the least squares problem (6.0.3) amounts to searching the point $\mathbf{p} \in \operatorname{Im}(\mathbf{A})$ nearest (w.r.t. Euclidean distance) to $\mathbf{b} \in \mathbb{R}^m$.

Geometric intuition, see Fig. 63: $\mathbf{p}$ is the orthogonal projection of $\mathbf{b}$ onto $\operatorname{Im}(\mathbf{A})$, that is $\mathbf{b} - \mathbf{p} \perp \operatorname{Im}(\mathbf{A})$. Note the equivalence

$$\mathbf{b} - \mathbf{p} \perp \operatorname{Im}(\mathbf{A}) \;\Leftrightarrow\; \mathbf{b} - \mathbf{p} \perp (\mathbf{A})_{:,j}\,, \quad j = 1, \ldots, n \;\Leftrightarrow\; \mathbf{A}^H(\mathbf{b} - \mathbf{p}) = 0\,,$$

Representation $\mathbf{p} = \mathbf{A}\mathbf{x}$ leads to normal equations (6.1.2).

Solve (6.0.3) for $\mathbf{b} \in \mathbb{R}^m$

$$\mathbf{x} \in \mathbb{R}^n\!: \quad \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \to \min \;\Leftrightarrow\; f(\mathbf{x}) := \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \to \min\,. \tag{6.1.1}$$

A quadratic functional, *cf.* (4.1.1)

$$f(\mathbf{x}) = \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 = \mathbf{x}^H(\mathbf{A}^H\mathbf{A})\mathbf{x} - 2\mathbf{b}^H\mathbf{A}\mathbf{x} + \mathbf{b}^H\mathbf{b}\,.$$

Minimization problem for f ➢ study gradient, *cf.* (4.1.4)

$$\mathbf{grad}\, f(\mathbf{x}) = 2(\mathbf{A}^H\mathbf{A})\mathbf{x} - 2\mathbf{A}^H\mathbf{b}\,.$$

$$\mathbf{grad}\, f(\mathbf{x}) \overset{!}{=} 0\!: \quad \boxed{\mathbf{A}^H\mathbf{A}\mathbf{x} = \mathbf{A}^H\mathbf{b}} \quad = \text{normal equation of (6.1.1)} \tag{6.1.2}$$

Notice: $\operatorname{rank}(\mathbf{A}) = n \;\Rightarrow\; \mathbf{A}^H\mathbf{A} \in \mathbb{R}^{n,n}$ s.p.d. (→ Def. 2.6.2)

Remark 6.1.1 (Conditioning of normal equations).

Caution: danger of instability, with SVD $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^H$

$$\operatorname{cond}_2(\mathbf{A}^H\mathbf{A}) = \operatorname{cond}_2(\mathbf{V}\Sigma^H\mathbf{U}^H\mathbf{U}\Sigma\mathbf{V}^H) = \operatorname{cond}_2(\Sigma^H\Sigma) = \frac{\sigma_1^2}{\sigma_n^2} = \operatorname{cond}_2(\mathbf{A})^2\,.$$

➢ For fairly ill-conditioned $\mathbf{A}$ using the normal equations (6.1.2) to solve the linear least squares problem (6.1.1) numerically may run the risk of huge amplification of roundoff errors incurred during the computation of the right hand side $\mathbf{A}^H\mathbf{b}$: potential instability (→ Def. **??**) of normal equation approach.

△

Example 6.1.2 (Instability of normal equations).

Caution: loss of information in the computation of $\mathbf{A}^H\mathbf{A}$, e.g.

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ \delta & 0 \\ 0 & \delta \end{pmatrix} \;\Rightarrow\; \mathbf{A}^H\mathbf{A} = \begin{pmatrix} 1+\delta^2 & 1 \\ 1 & 1+\delta^2 \end{pmatrix}$$

```
>> A = [1          1;...
        sqrt(eps)  0;...
        0          sqrt(eps)];
>> rank(A)
     ans =  2
>> rank(A'*A)
     ans =  1
```

If $\delta < \sqrt{\texttt{eps}} \;\Rightarrow\; 1+\delta^2 = 1$ in $\mathbb{M}$, i.e. $\mathbf{A}^H\mathbf{A}$ "numeric singular", though $\operatorname{rank}(\mathbf{A}) = 2$, see Sect. **??**, in particular Rem. **??**.

◇

Another *reason not to compute* $\mathbf{A}^H\mathbf{A}$, when both m, n large:

$$\mathbf{A} \text{ sparse} \;\not\Rightarrow\; \mathbf{A}^T\mathbf{A} \text{ sparse}$$

- Potential memory overflow, when computing $\mathbf{A}^T\mathbf{A}$
- Squanders possibility to use efficient sparse direct elimination techniques, see Sect. 2.5.3

A way to avoid the computation of $\mathbf{A}^H\mathbf{A}$:

Expand normal equations (6.1.2): introduce residual $\mathbf{r} := \mathbf{A}\mathbf{x} - \mathbf{b}$ as new unknown:

$$\mathbf{A}^H\mathbf{A}\mathbf{x} = \mathbf{A}^H\mathbf{b} \quad \Leftrightarrow \quad \mathbf{B}\begin{pmatrix}\mathbf{r}\\ \mathbf{x}\end{pmatrix} := \begin{pmatrix}-\mathbf{I} & \mathbf{A}\\ \mathbf{A}^H & 0\end{pmatrix}\begin{pmatrix}\mathbf{r}\\ \mathbf{x}\end{pmatrix} = \begin{pmatrix}\mathbf{b}\\ 0\end{pmatrix} . \tag{6.1.3}$$

More general substitution $\mathbf{r} := \alpha^{-1}(\mathbf{A}\mathbf{x} - \mathbf{b})$, $\alpha > 0$ to improve the condition:

$$\mathbf{A}^H\mathbf{A}\mathbf{x} = \mathbf{A}^H\mathbf{b} \quad \Leftrightarrow \quad \mathbf{B}_\alpha\begin{pmatrix}\mathbf{r}\\ \mathbf{x}\end{pmatrix} := \begin{pmatrix}-\alpha\mathbf{I} & \mathbf{A}\\ \mathbf{A}^H & 0\end{pmatrix}\begin{pmatrix}\mathbf{r}\\ \mathbf{z}\end{pmatrix} = \begin{pmatrix}\mathbf{b}\\ 0\end{pmatrix} . \tag{6.1.4}$$

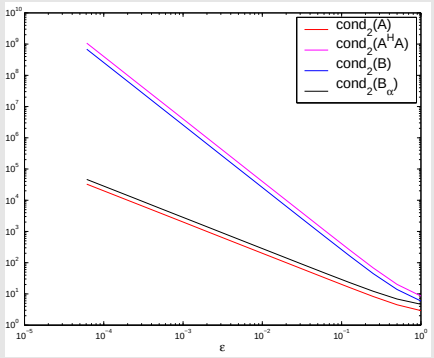
For $m, n \gg 1$, $\mathbf{A}$ sparse, both (6.1.3) and (6.1.4) lead to large sparse linear systems of equations, amenable to sparse direct elimination techniques, see Sect. 2.5.3

Example 6.1.3 (Condition of the extended system).

Consider (6.1.3), (6.1.4) for

$$\mathbf{A} = \begin{pmatrix}1+\epsilon & 1\\ 1-\epsilon & 1\\ \epsilon & \epsilon\end{pmatrix} .$$

Plot of different condition numbers in dependence on ϵ ($\alpha = \|\mathbf{A}\|_2/\sqrt{2}$)



6.2 Orthogonal Transformation Methods

Consider the linear least squares problem (6.0.3)

given $\mathbf{A} \in \mathbb{R}^{m,n}$, $\mathbf{b} \in \mathbb{R}^m$ find $\mathbf{x} = \operatorname{argmin}_{\mathbf{y}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{y} - \mathbf{b}\|_2$.

Assumption: $m \geq n$ and $\mathbf{A}$ has full (maximum) rank: $\operatorname{rank}(\mathbf{A}) = n$.

Recall Thm. **??**: orthogonal (unitary) transformations (→ Def. **??**) leave 2-norm invariant.

Idea: Transformation of $\mathbf{A}\mathbf{x} - \mathbf{b}$ to simpler form by *orthogonal* row transformations:

$$\operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{y} - \mathbf{b}\|_2 = \operatorname*{argmin}_{\mathbf{y}\in\mathbb{R}^n} \left\|\widetilde{\mathbf{A}}\mathbf{y} - \widetilde{\mathbf{b}}\right\|_2 ,$$

where $\widetilde{\mathbf{A}} = \mathbf{Q}\mathbf{A}$, $\widetilde{\mathbf{b}} = \mathbf{Q}\mathbf{b}$ with orthogonal $\mathbf{Q} \in \mathbb{R}^{m,m}$.

As in the case of LSE (→ Sect. **??**): "simpler form" = triangular form.

QR-decomposition

QR-decomposition: $\mathbf{A} = \mathbf{Q}\mathbf{R}$, $\mathbf{Q} \in \mathbb{K}^{m,m}$ unitary, $\mathbf{R} \in \mathbb{K}^{m,n}$ (regular) upper triangular matrix.

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 = \left\|\mathbf{Q}(\mathbf{R}\mathbf{x} - \mathbf{Q}^H\mathbf{b})\right\|_2 = \left\|\mathbf{R}\mathbf{x} - \widetilde{\mathbf{b}}\right\|_2 \quad , \quad \widetilde{\mathbf{b}} := \mathbf{Q}^H\mathbf{b} .$$

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \to \min \quad \Leftrightarrow \quad \left\| \begin{pmatrix}\mathbf{R}\\ \mathbf{0}\end{pmatrix} \begin{pmatrix}x_1\\ \vdots\\ x_n\end{pmatrix} - \begin{pmatrix}\widetilde{b}_1\\ \vdots\\ \\ \\ \vdots\\ \widetilde{b}_m\end{pmatrix} \right\|_2 \to \min .$$

$$\mathbf{x} = \begin{pmatrix}\mathbf{R}\end{pmatrix}^{-1} \begin{pmatrix}\widetilde{b}_1\\ \vdots\\ \widetilde{b}_n\end{pmatrix} \quad , \quad \text{residuum} \quad \mathbf{r} = \mathbf{Q}\begin{pmatrix}0\\ \vdots\\ 0\\ \widetilde{b}_{n+1}\\ \vdots\\ \widetilde{b}_m\end{pmatrix} .$$

Note: residual norm readily available $\|\mathbf{r}\|_2 = \sqrt{\widetilde{b}_{n+1}^2 + \cdots + \widetilde{b}_m^2}$.

Implementation: successive orthogonal row transformations (by means of Householder reflections (**??**) for general matrices, and Givens rotations (**??**) for banded matrices, see Sect. **??** for details) of augmented matrix $(\mathbf{A},\mathbf{b}) \in \mathbb{R}^{m,n+1}$, which is transformed into $(\mathbf{R},\widetilde{\mathbf{b}})$

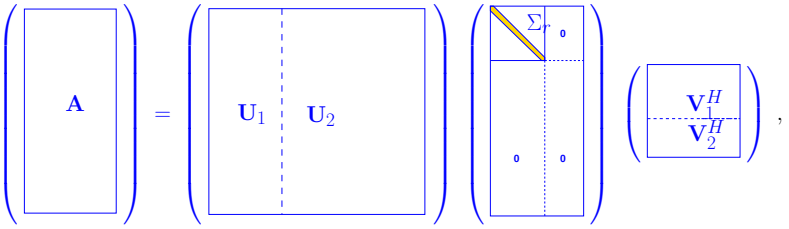
$\mathbf{Q}$ need not be stored !

➤ A QR-based algorithm is implemented in the least-squares-solver of the MATLAB-operator "\" (for dense matrices).

Alternative: Solving linear least squares problem by SVD

Most general setting: $\mathbf{A} \in \mathbb{K}^{m,n}$, $\operatorname{rank}(\mathbf{A}) = r \le \min\{m,n\}$):

SVD: $\mathbf{A} = [\,\mathbf{U}_1 \;\; \mathbf{U}_2\,] \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{pmatrix}$



(6.2.1)

$\mathbf{U}_1 \in \mathbb{K}^{m,r}$, $\mathbf{U}_2 = \mathbb{K}^{m,m-r}$, $\Sigma_r = \operatorname{diag}(\sigma_1,\ldots,\sigma_r) \in \mathbb{R}^{r,r}$, $\mathbf{V}_1 \in \mathbb{K}^{n,r}$, $\mathbf{V}_2 \in \mathbb{K}^{n,n-r}$, the columns of $\mathbf{U}_1,\mathbf{U}_2,\mathbf{V}_1,\mathbf{V}_2$ are orthonormal.

$$\|\mathbf{Ax}-\mathbf{b}\|_2 = \left\| [\mathbf{U}_1\;\mathbf{U}_2] \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{pmatrix} \mathbf{x} - \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} \Sigma_r \mathbf{V}_1^H \mathbf{x} \\ 0 \end{pmatrix} - \begin{pmatrix} \mathbf{U}_1^H \mathbf{b}_1 \\ \mathbf{U}_2^H \mathbf{b}_2 \end{pmatrix} \right\|_2 \quad (6.2.2)$$

Logical strategy: choose $\mathbf{x}$ such that the first r components of $\begin{pmatrix} \Sigma_r \mathbf{V}_1^H \mathbf{x} \\ 0 \end{pmatrix} - \begin{pmatrix} \mathbf{U}_1^H \mathbf{b}_1 \\ \mathbf{U}_2^H \mathbf{b}_2 \end{pmatrix}$ vanish:

➢ *underdetermined* linear system $\Sigma_r \mathbf{V}_1^H \mathbf{x} = \mathbf{U}_1^H \mathbf{b}_1$. (6.2.3)

To fix a unique solution we appeal to the minimal norm condition in (6.0.3): solution $\mathbf{x}$ of (6.2.3) is unique up to contributions from $\operatorname{Ker}(\mathbf{V}_1) = \operatorname{Im}(\mathbf{V}_2)$. Since $\mathbf{V}$ is orthogonal, the minimal norm solution is obtained by setting contributions from $\operatorname{Im}(\mathbf{V}_2)$ to zero, which amounts to choosing $\mathbf{x} \in \operatorname{Im}(\mathbf{V}_1)$.

➤ solution $\boxed{\mathbf{x} = (\mathbf{V}_1 \Sigma_r^{-1} \mathbf{U}_1^H)\mathbf{b}_1}$, $\|\mathbf{r}\|_2 = \left\|\mathbf{U}_2^H \mathbf{b}_2\right\|_2$. (6.2.4)

Practical implementation:

"numerical rank" test: $r = \max\{i\colon \sigma_i/\sigma_1 > \texttt{tol}\}$

Code 6.2.1: Solving LSQ problem via SVD

```
function y = lsqsvd(A,b)
[U,S,V] = svd(A,0);
sv = diag(S);
r = max(find(sv/sv(1) > eps));
y = V(:,1:r)*(diag(1./sv(1:r))*...
          (U(:,1:r)'*b));
```


Remark 6.2.2 (Pseudoinverse and SVD). → Rem. 6.0.4

The solution formula (6.2.4) directly yields a representation of the pseudoinverse $\mathbf{A}^+$ (→ Def. 6.0.2) of any matrix $\mathbf{A}$:

Theorem 6.2.1 (Pseudoinverse and SVD).
If $\mathbf{A} \in \mathbb{K}^{m,n}$ *has the SVD decomposition* (6.2.1), *then* $\mathbf{A}^+ = \mathbf{V}_1 \Sigma_r^{-1} \mathbf{U}_1^H$ *holds.*

△

Remark 6.2.3 (Normal equations vs. orthogonal transformations method).

Superior numerical stability (→ Def. **??**) of orthogonal transformations methods:

➤ Use orthogonal transformations methods for least squares problems (6.0.3), whenever $\mathbf{A} \in \mathbb{R}^{m,n}$ *dense* and n small.

SVD/QR-factorization cannot exploit sparsity:

➤ Use normal equations in the expanded form (6.1.3)/(6.1.4), when $\mathbf{A} \in \mathbb{R}^{m,n}$ sparse (→ Def. 2.5.1) and m,n big.

△

Example 6.2.4 (Fit of hyperplanes).

This example studies the power and versatility of orthogonal transformations in the context of (generalized) least squares minimization problems.

The Hesse normal form of a hyperplane $\mathcal{H}$ (= affine subspace of dimension $d-1$) in $\mathbb{R}^d$ is:

$$\mathcal{H} = \{\mathbf{x} \in \mathbb{R}^d : c + \mathbf{n}^T\mathbf{x} = 0\}\ , \quad \|\mathbf{n}\|_2 = 1\ . \tag{6.2.5}$$

► Euclidean distance of $\mathbf{y} \in \mathbb{R}^d$ from the plane: $\operatorname{dist}(\mathcal{H}, \mathbf{y}) = |c + \mathbf{n}^T\mathbf{y}|$. (6.2.6)

Goal: given the points $\mathbf{y}_1, \ldots, \mathbf{y}_m$, $m > d$, find $\mathcal{H} \leftrightarrow \{c \in \mathbb{R}, \mathbf{n} \in \mathbb{R}^d, \|\mathbf{n}\|_2 = 1\}$, such that

$$\sum_{j=1}^{m} \operatorname{dist}(\mathcal{H}, \mathbf{y}_j)^2 = \sum_{j=1}^{m} |c + \mathbf{n}^T\mathbf{y}_j|^2 \to \min\ . \tag{6.2.7}$$

Note: (6.2.7) $\neq$ linear least squares problem due to constraint $\|\mathbf{n}\|_2 = 1$.

$$(6.2.7) \quad \Leftrightarrow \quad \left\| \underbrace{\begin{pmatrix} 1 & y_{1,1} & \cdots & y_{1,d} \\ 1 & y_{2,1} & \cdots & y_{2,d} \\ \vdots & \vdots & & \vdots \\ 1 & y_{m,1} & \cdots & y_{m,d} \end{pmatrix}}_{=:\mathbf{A}} \begin{pmatrix} c \\ n_1 \\ \vdots \\ n_d \end{pmatrix} \right\|_2 \to \min \quad \text{under constraint} \quad \|\mathbf{n}\|_2 = 1\ .$$

Step ❶: QR-decomposition (→ Section **??**)

$$\mathbf{A} := \begin{pmatrix} 1 & y_{1,1} & \cdots & y_{1,d} \\ 1 & y_{2,1} & \cdots & y_{2,d} \\ \vdots & \vdots & & \vdots \\ 1 & y_{m,1} & \cdots & y_{m,d} \end{pmatrix} = \mathbf{QR} \quad , \quad \mathbf{R} := \begin{pmatrix} r_{11} & r_{12} & \cdots & \cdots & r_{1,d+1} \\ 0 & r_{22} & \cdots & \cdots & r_{2,d+1} \\ \vdots & & \ddots & & \vdots \\ 0 & & & & r_{d+1,d+1} \\ 0 & \cdots & & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & & \cdots & 0 \end{pmatrix} \in \mathbb{R}^{m,d+1}\ .$$

$$\|\mathbf{Ax}\|_2 \to \min \quad \Leftrightarrow \quad \|\mathbf{Rx}\|_2 = \left\| \begin{pmatrix} r_{11} & r_{12} & \cdots & \cdots & r_{1,d+1} \\ 0 & r_{22} & \cdots & \cdots & r_{2,d+1} \\ \vdots & & \ddots & & \vdots \\ 0 & & & & r_{d+1,d+1} \\ 0 & \cdots & & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & \cdots & & \cdots & 0 \end{pmatrix} \begin{pmatrix} c \\ n_1 \\ \vdots \\ n_d \end{pmatrix} \right\|_2 \to \min\ . \tag{6.2.8}$$

Step ❷ Note that necessarily (why?)

$$c \cdot r_{11} + n_1 \cdot r_{12} + \cdots + r_{1,d+1} \cdot n_d = 0\ .$$

This insight converts (6.2.8) to

$$\left\| \begin{pmatrix} r_{22} & r_{23} & \cdots & \cdots & r_{2,d+1} \\ 0 & r_{33} & \cdots & \cdots & r_{3,d+1} \\ \vdots & & \ddots & & \vdots \\ 0 & & & & r_{d+1,d+1} \end{pmatrix} \begin{pmatrix} n_1 \\ \vdots \\ \vdots \\ n_d \end{pmatrix} \right\|_2 \to \min \quad , \quad \|\mathbf{n}\|_2 = 1\ . \tag{6.2.9}$$

(6.2.9) = problem of type (5.4.5), minimization on the Euclidean sphere.

➢ Solve (6.2.9) using SVD !

Note: Since $r_{11} = \left\|(\mathbf{A})_{:,1}\right\|_2 = \sqrt{d+1} \neq 0 \quad \Rightarrow \quad c = -r_{11}^{-1} \sum_{j=1}^{d} r_{1,j+1} n_j$.

MATLAB-function:

For $\mathbf{A} \in \mathbb{K}^{m,n}$ find $\mathbf{n} \in \mathbb{R}^d$, $\mathbf{c} \in \mathbb{R}^{n-d}$ such that $\left\| \mathbf{A} \begin{pmatrix} \mathbf{c} \\ \mathbf{n} \end{pmatrix} \right\|_2 \to \min$

Code 6.2.5: (Generalized) distance fitting a hyperplane

```
function [c,n] = clsq(A,dim);
[m,p] = size(A);
if p < dim+1, error('not enough unknowns'); end;
if m < dim, error('not enough equations'); end;
m = min(m, p);
```


◇

6.3 Total Least Squares

Given: overdetermined linear system of equations $\mathbf{Ax} = \mathbf{b}$, $\mathbf{A} \in \mathbb{R}^{m,n}$, $\mathbf{b} \in \mathbb{R}^m$, $m \geq n$.

Known: LSE solvable $\Leftrightarrow$ $\mathbf{b} \in \operatorname{Im}(\mathbf{A})$, *if* $\mathbf{A}$, $\mathbf{b}$ *were not perturbed*,

but $\mathbf{A}$, $\mathbf{b}$ are perturbed (measurement errors).

Sought: *Solvable* overdetermined system of equations $\widehat{\mathbf{A}}\mathbf{x} = \widehat{\mathbf{b}}$, $\widehat{\mathbf{A}} \in \mathbb{R}^{m,n}$, $\widehat{\mathbf{b}} \in \mathbb{R}^m$, "nearest" to $\mathbf{Ax} = \mathbf{b}$.

☞ least squares problem "turned upside down": now we are allowed to tamper with system matrix and right hand side vector!

Total least squares problem:

Given: $\mathbf{A} \in \mathbb{R}^{m,n}$, $m \geq n$, $\operatorname{rank}(\mathbf{A}) = n$, $\mathbf{b} \in \mathbb{R}^m$,

find: $\widehat{\mathbf{A}} \in \mathbb{R}^{m,n}$, $\widehat{\mathbf{b}} \in \mathbb{R}^m$ with

$$\left\| \underbrace{\begin{bmatrix} \mathbf{A} & \mathbf{b} \end{bmatrix}}_{=:\mathbf{C}} - \underbrace{\begin{bmatrix} \widehat{\mathbf{A}} & \widehat{\mathbf{b}} \end{bmatrix}}_{=:\widehat{\mathbf{C}}} \right\|_F \to \min \quad , \quad \widehat{\mathbf{b}} \in \operatorname{Im}(\widehat{\mathbf{A}}) \, . \tag{6.3.1}$$

$$\widehat{\mathbf{b}} \in \operatorname{Im}(\widehat{\mathbf{A}}) \;\Rightarrow\; \operatorname{rank}(\widehat{\mathbf{C}}) = n \quad \blacktriangleright \quad (6.3.1) \;\Rightarrow\; \min_{\operatorname{rank}(\widehat{\mathbf{C}})=n} \left\| \mathbf{C} - \widehat{\mathbf{C}} \right\|_F \, .$$

Thm. 5.4.7 ➤ use the SVD decomposition of $\mathbf{C}$ to construct $\widehat{\mathbf{C}}$:

$$\mathbf{C} = \mathbf{U}\Sigma\mathbf{V}^H = \sum_{j=1}^{n+1} \sigma_j (\mathbf{U})_{:,j} (\mathbf{V})_{:,j}^H$$

$$\widehat{\mathbf{C}} = \sum_{j=1}^{n} \sigma_j (\mathbf{U})_{:,j} (\mathbf{V})_{:,j}^H \;\Rightarrow\; \widehat{\mathbf{C}}(\mathbf{V})_{:,n+1} = 0 \, .$$

If $(\mathbf{V})_{n+1,n+1} \neq 0$, then

$$\widehat{\mathbf{A}}\mathbf{x} = \widehat{\mathbf{b}} \quad \text{with} \quad \mathbf{x} = (\mathbf{v})_{n+1,n+1}^{-1} (\mathbf{V})_{\cdot,n+1} \, .$$

Code 6.3.2: Total least squares via SVD

```
function x = lsqtotal(A,b);
[m,n]=size(A);
[U, Sigma, V] = svd([A,b]);
s = V(n+1,n+1);
if s == 0 ,
  error('No solution')
end
x = -V(1:n,n+1)/s;
```

6.4 Constrained Least Squares

Given: $\mathbf{A} \in \mathbb{R}^{m,n}$, $m \geq n$, $\operatorname{rank}(\mathbf{A}) = n$, $\mathbf{b} \in \mathbb{R}^m$,
$\mathbf{C} \in \mathbb{R}^{p,n}$, $p < n$, $\operatorname{rank}(\mathbf{C}) = p$, $\mathbf{d} \in \mathbb{R}^p$

Find: $\mathbf{x} \in \mathbb{R}^n$ with $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \to \min$, $\boxed{\mathbf{C}\mathbf{x} = \mathbf{d}}$. (6.4.1)

Linear constraint

Solution via normal equations

Idea: coupling the constraint using the Lagrange multiplier $\mathbf{m} \in \mathbb{R}^p$

$$\mathbf{x} = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \max_{\mathbf{m}\in\mathbb{R}^p} L(\mathbf{x},\mathbf{m}) \, , \quad L(\mathbf{x},\mathbf{m}) := \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2 + \mathbf{m}^H(\mathbf{C}\mathbf{x} - \mathbf{d}) \, .$$

Necessary (and sufficient) condition for the solution (→ Section 6.1)

$$\frac{\partial L}{\partial \mathbf{x}}(\mathbf{x},\mathbf{m}) = \mathbf{A}^H(\mathbf{A}\mathbf{x} - \mathbf{b}) + \mathbf{C}^H\mathbf{m} \overset{!}{=} 0 \quad , \quad \frac{\partial L}{\partial \mathbf{m}}(\mathbf{x},\mathbf{m}) = \mathbf{C}\mathbf{x} - \mathbf{d} \overset{!}{=} 0.$$

$$\begin{pmatrix} \mathbf{A}^H\mathbf{A} & \mathbf{C}^H \\ \mathbf{C} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{m} \end{pmatrix} = \begin{pmatrix} \mathbf{A}^H\mathbf{b} \\ \mathbf{d} \end{pmatrix}$$ Extended normal equations (saddle point problem)

Algorithm (based on block-LU-decomposition):

$$\begin{pmatrix} \mathbf{A}^H\mathbf{A} & \mathbf{C}^H \\ \mathbf{C} & 0 \end{pmatrix} = \begin{pmatrix} \mathbf{R}^H & 0 \\ \mathbf{G} & -\mathbf{S}^H \end{pmatrix} \begin{pmatrix} \mathbf{R} & \mathbf{G}^H \\ 0 & \mathbf{S} \end{pmatrix} \quad , \quad \mathbf{R}, \mathbf{S} \in \mathbb{R}^{n,n} \text{ upper triangular matrix, } \mathbf{G} \in \mathbb{R}^{p,n} \, .$$

$\mathbf{R}$ from $\mathbf{R}^H\mathbf{R} = \mathbf{A}^H\mathbf{A}$ → Cholesky decomposition → Sect. 2.6,
$\mathbf{G}$ from $\mathbf{R}^H\mathbf{G}^H = \mathbf{C}^H$ → n forward substitution → Sect. 2.2,
$\mathbf{S}$ from $\mathbf{S}^H\mathbf{S} = \mathbf{G}\mathbf{G}^H$ → Cholesky decomposition → Sect. 2.6.

Caution Sect. 6.1: the computation of $\mathbf{A}^H\mathbf{A}$ can be expensive and problematic!
(remedy through introduction of a new unknown $\mathbf{r} = \mathbf{A}\mathbf{x} - \mathbf{b}$, cf. (6.1.3))

$$\begin{pmatrix} -\mathbf{I} & \mathbf{A} & 0 \\ \mathbf{A}^H & 0 & \mathbf{C}^H \\ 0 & \mathbf{C} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{r} \\ \mathbf{x} \\ \mathbf{m} \end{pmatrix} = \begin{pmatrix} \mathbf{b} \\ 0 \\ \mathbf{d} \end{pmatrix} \, . \tag{6.4.2}$$

Solution via SVD:

① Compute orthonormal basis of $\operatorname{Ker}(\mathbf{C})$ using SVD (→ Section 6.2):

$$\mathbf{C} = \mathbf{U}\,[\Sigma\; 0] \begin{bmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{bmatrix} \quad , \quad \mathbf{U} \in \mathbb{R}^{p,p}, \Sigma \in \mathbb{R}^{p,p}, \mathbf{V}_1 \in \mathbb{R}^{n,p}, \mathbf{V}_2 \in \mathbb{R}^{n,n-p}$$

$\blacktriangleright$ $\mathrm{Ker}(\mathbf{C}) = \mathrm{Im}(\mathbf{V}_2)$.

and the particular solution

$$\mathbf{x}_0 := \mathbf{V}_1 \Sigma^{-1} \mathbf{U}^H \mathbf{d} \, .$$

Representation of the solution $\mathbf{x}$ of (6.4.1): $\mathbf{x} = \mathbf{x}_0 + \mathbf{V}_2 \mathbf{y}, \quad \mathbf{y} \in \mathbb{R}^{n-p}$.

② Insert this representation in (6.4.1) ➤ standard linear least squares

$$\|\mathbf{A}(\mathbf{x}_0 + \mathbf{V}_2\mathbf{y}) - \mathbf{b}\|_2 \to \min \quad \Leftrightarrow \quad \|\mathbf{A}\mathbf{V}_2\mathbf{y} - (\mathbf{b} - \mathbf{A}\mathbf{x}_0)\| \to \min \, .$$

Exercise 6.4.1. Given a regular tridiagonal matrix $\mathbf{T} \in \mathbb{R}^{n,n}$, develop an algorithm for solving the linear least squares problem

$$\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2 \, ,$$

where

$$\mathbf{A} = \begin{pmatrix} \mathbf{T}^{-1} \\ \vdots \\ \mathbf{T}^{-1} \end{pmatrix} \in \mathbb{R}^{pn,n} \, .$$

6.5 Non-linear Least Squares

Example 6.5.1 (Non-linear data fitting (parametric statistics)).

Given: data points (t_i, y_i), $i = 1, \ldots, m$ with measurements errors.

Known: $y = f(t, \mathbf{x})$ through a function $f : \mathbb{R} \times \mathbb{R}^n \mapsto \mathbb{R}$ depending non-linearly and smoothly on parameters $\mathbf{x} \in \mathbb{R}^n$.

Example: $f(t) = x_1 + x_2 \exp(-x_3 t), \quad n = 3.$

Determine parameters by non-linear least squares data fitting:

$$\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \sum_{i=1}^{m} |f(t_i, \mathbf{x}) - y_i|^2 = \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \tfrac{1}{2} \|F(\mathbf{x})\|_2^2 \, , \tag{6.5.1}$$

with $F(\mathbf{x}) = \begin{pmatrix} f(t_1, \mathbf{x}) - y_1 \\ \vdots \\ f(t_m, \mathbf{x}) - y_m \end{pmatrix}$.

◇

Non-linear least squares problem

Given: $F : D \subset \mathbb{R}^n \mapsto \mathbb{R}^m, \quad m, n \in \mathbb{N}, \; m > n.$

Find: $\mathbf{x}^* \in D$: $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x}\in D} \Phi(\mathbf{x})$, $\Phi(\mathbf{x}) := \frac{1}{2}\|F(\mathbf{x})\|_2^2$. (6.5.2)

Terminology: $D \mathrel{\hat{=}}$ parameter space, $x_1, \ldots, x_n \mathrel{\hat{=}}$ parameter.

As in the case of linear least squares problems (→ Rem. 6.0.3): a non-linear least squares problem is related to an overdetermined non-linear system of equations $F(\mathbf{x}) = 0$.

As for non-linear systems of equations (→ Chapter 3): existence and uniqueness of $\mathbf{x}^*$ in (6.5.2) has to be established in each concrete case!

We require "independence for each parameter":

$\exists$ neighbourhood $\mathcal{U}(\mathbf{x}^*)$ such that $DF(\mathbf{x})$ has full rank $n \quad \forall \, \mathbf{x} \in \mathcal{U}(\mathbf{x}^*)$. (6.5.3)

(It means: the columns of the Jacobi matrix $DF(\mathbf{x})$ are linearly independent.)

If (6.5.3) is not satisfied, then the parameters are redundant in the sense that fewer parameters would be enough to model the same dependence (locally at $\mathbf{x}^*$).

6.5.1 (Damped) Newton method

$$\Phi(\mathbf{x}^*) = \min \quad \Rightarrow \quad \mathbf{grad}\,\Phi(\mathbf{x}) = 0, \quad \mathbf{grad}\,\Phi(\mathbf{x}) := (\tfrac{\partial\Phi}{\partial x_1}(\mathbf{x}), \ldots, \tfrac{\partial\Phi}{\partial x_n}(\mathbf{x}))^T \in \mathbb{R}^n.$$

Simple idea: use Newton's method (→ Sect. 3.4) to determine a zero of $\mathbf{grad}\,\Phi : D \subset \mathbb{R}^n \mapsto \mathbb{R}^n$.

Newton iteration (3.4.1) for non-linear system of equations $\mathbf{grad}\,\Phi(\mathbf{x}) = 0$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - H\Phi(\mathbf{x}^{(k)})^{-1} \mathbf{grad}\,\Phi(\mathbf{x}^{(k)}) \, , \quad (H\Phi(\mathbf{x}) = \text{Hessian matrix}) \, . \tag{6.5.4}$$

Expressed in terms of $F : \mathbb{R}^n \mapsto \mathbb{R}^n$ from (6.5.2):

chain rule (3.4.2) ➤ $\mathbf{grad}\,\Phi(\mathbf{x}) = DF(\mathbf{x})^T F(\mathbf{x})$,

product rule (3.4.3) ➤ $H\Phi(\mathbf{x}) := D(\mathbf{grad}\,\Phi)(\mathbf{x}) = DF(\mathbf{x})^T DF(\mathbf{x}) + \sum_{j=1}^{m} F_j(\mathbf{x}) D^2 F_j(\mathbf{x})$,

$$\Updownarrow$$

$$(H\Phi(\mathbf{x}))_{i,k} = \sum_{j=1}^{n} \frac{\partial^2 F_j}{\partial x_i \partial x_k}(\mathbf{x}) F_j(\mathbf{x}) + \frac{\partial F_j}{\partial x_k}(\mathbf{x}) \frac{\partial F_j}{\partial x_i}(\mathbf{x}) \,.$$

For Newton iterate $\mathbf{x}^{(k)}$: Newton correction $\mathbf{s} \in \mathbb{R}^n$ from LSE

$$\left(DF(\mathbf{x}^{(k)})^T DF(\mathbf{x}^{(k)}) + \sum_{j=1}^{m} F_j(\mathbf{x}^{(k)}) D^2 F_j(\mathbf{x}^{(k)}) \right) \mathbf{s} = -DF(\mathbf{x}^{(k)})^T F(\mathbf{x}^{(k)}) \,. \tag{6.5.5}$$

Remark 6.5.2 (Newton method and minimization of quadratic functional).

Newton's method (6.5.4) for (6.5.2) can be read as *successive minimization* of a local quadratic approximation of Φ:

$$\Phi(\mathbf{x}) \approx Q(\mathbf{s}) := \Phi(\mathbf{x}^{(k)}) + \mathbf{grad}\,\Phi(\mathbf{x}^{(k)})^T \mathbf{s} + \frac{1}{2} \mathbf{s}^T H\Phi(\mathbf{x}^{(k)}) \mathbf{s} \,, \tag{6.5.6}$$

$$\mathbf{grad}\, Q(\mathbf{s}) = 0 \quad \Leftrightarrow \quad H\Phi(\mathbf{x}^{(k)})\mathbf{s} + \mathbf{grad}\,\Phi(\mathbf{x}^{(k)}) = 0 \quad \Leftrightarrow \quad \text{(6.5.5)} \,.$$

➢ Another model function method (→ Sect. 3.3.2) with quadratic model function for Q. △

6.5.2 Gauss-Newton method

Idea: local linearization of F: $F(x) \approx F(y) + DF(\mathbf{y})(\mathbf{x} - \mathbf{y})$

➢ sequence of *linear* least squares problems

$$\operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \|F(\mathbf{x})\|_2 \text{ approximated by } \underbrace{\operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \|F(\mathbf{x}_0) + DF(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)\|_2}_{(\spadesuit)} \,,$$

where $\mathbf{x}_0$ is an approximation of the solution $\mathbf{x}^*$ of (6.5.2).

$$(\spadesuit) \quad \Leftrightarrow \quad \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{A}\mathbf{x} - \mathbf{b}\| \quad \text{with} \quad \mathbf{A} := DF(\mathbf{x}_0) \in \mathbb{R}^{m,n} \,, \quad \mathbf{b} := F(\mathbf{x}_0) - DF(\mathbf{x}_0)\mathbf{x}_0 \in \mathbb{R}^m \,.$$

This is a linear least squares problem of the form (6.0.3).

Note: (6.5.3) $\Rightarrow$ $\mathbf{A}$ has full rank, if $\mathbf{x}_0$ sufficiently close to $\mathbf{x}^*$.

Note: Approach different from local quadratic approximation of Φ underlying Newton's method for (6.5.2), see Sect. 6.5.1, Rem. 6.5.2.

Gauss-Newton iteration

Initial guess $\mathbf{x}^{(0)} \in D$

$$\mathbf{x}^{(k+1)} := \mathbf{x}^{(k)} - \mathbf{s} \quad , \quad \mathbf{s} := \operatorname*{argmin}_{\mathbf{x}\in\mathbb{R}^n} \left\| F(\mathbf{x}^{(k)}) - DF(\mathbf{x}^{(k)})\mathbf{s} \right\|_2 \,. \tag{6.5.7}$$

linear least squares problem

MATLAB-\ used to solve linear least squares problem in each step:

for $\mathbf{A} \in \mathbb{R}^{m,n}$

x = A\b

$\Updownarrow$

$\mathbf{x}$ minimizer of $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2$ with minimal 2-norm

Code 6.5.4: template for Gauss-Newton method

```
1 function x = gn(x,F,J,tol)
2 s = J(x)\F(x);                    %
3 x = x-s;
4 while (norm(s) > tol*norm(x)) %
5   s = J(x)\F(x);                  %
6   x = x-s;
7 end
```

Comments on Code 6.5.2:

☞ Argument x passes initial guess $\mathbf{x}^{(0)} \in \mathbb{R}^n$, argument F must be a *handle* to a function $F : \mathbb{R}^n \mapsto \mathbb{R}^m$, argument J provides the Jacobian of F, namely $DF : \mathbb{R}^n \mapsto \mathbb{R}^{m,n}$, argument `tol` specifies the tolerance for termination

☞ Line 4: iteration terminates if relative norm of correction is below threshold specified in `tol`.

Summary:

Advantage of the Gauss-Newton method : second derivative of F not needed.
Drawback of the Gauss-Newton method : no local quadratic convergence.

Example 6.5.5 (Non-linear data fitting (II)). → Ex. 6.5.1

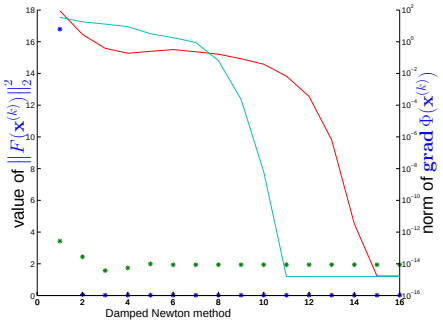
Non-linear data fitting problem (6.5.1) for $f(t) = x_1 + x_2 \exp(-x_3 t)$.

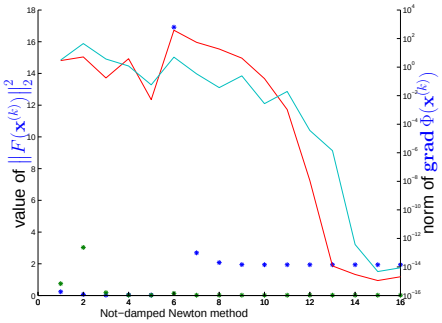
$$F(\mathbf{x}) = \begin{pmatrix} x_1 + x_2 \exp(-x_3 t_1) - y_1 \\ \vdots \\ x_1 + x_2 \exp(-x_3 t_m) - y_m \end{pmatrix} : \mathbb{R}^3 \mapsto \mathbb{R}^m \,, \; DF(\mathbf{x}) = \begin{pmatrix} 1 & e^{-x_3 t_1} & -x_2 t_1 e^{-x_3 t_1} \\ \vdots & \vdots & \vdots \\ 1 & e^{-x_3 t_m} & -x_2 t_m e^{-x_3 t_m} \end{pmatrix}$$

Numerical experiment:

convergence of the Newton method, damped Newton method (→ Section 3.4.4) and Gauss-Newton method for different initial values

```
rand('seed',0);
t = (1:0.3:7)';
y = x(1) + x(2)*exp(-x(3)*t);
y = y+0.1*(rand(length(y),1)-0.5);
```





Convergence behaviour of the Newton method:

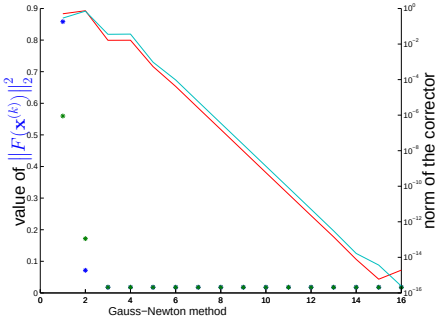
initial value $(1.8, 1.8, 0.1)^T$ (red curve) ➤ Newton method caught in local minimum,
initial value $(1.5, 1.5, 0.1)^T$ (cyan curve) ➤ fast (locally quadratic) convergence.

Gauss-Newton method:

initial value $(1.8, 1.8, 0.1)^T$ (red curve),
initial value $(1.5, 1.5, 0.1)^T$ (cyan curve),

convergence in both cases.

Notice: linear convergence.



◇

6.5.3 Trust region method (Levenberg-Marquardt method)

As in the case of Newton's method for non-linear systems of equations, see Sect. 3.4.4: often overshooting of Gauss-Newton corrections occurs.

Remedy as in the case of Newton's method: damping.

6.5
p. 366

Idea: damping of the Gauss-Newton correction in (6.5.7) using a penalty term

instead of $\left\| F(\mathbf{x}^{(k)}) + DF(\mathbf{x}^{(k)})\mathbf{s} \right\|^2$ minimize $\left\| F(\mathbf{x}^{(k)}) + DF(\mathbf{x}^{(k)})\mathbf{s} \right\|^2 + \lambda \|\mathbf{s}\|_2^2$.

$\lambda > 0 \hat{=}$ penalty parameter (how to choose it ? ⟶ heuristic)

$$\lambda = \gamma \left\| F(\mathbf{x}^{(k)}) \right\|_2 \quad , \quad \gamma := \begin{cases} 10 & \text{, if } \left\| F(\mathbf{x}^{(k)}) \right\|_2 \geq 10 \, , \\ 1 & \text{, if } 1 < \left\| F(\mathbf{x}^{(k)}) \right\|_2 < 10 \, , \\ 0.01 & \text{, if } \left\| F(\mathbf{x}^{(k)}) \right\|_2 \leq 1 \, . \end{cases}$$

Modified (regularized) equation for the corrector $\mathbf{s}$:

$$\left(DF(\mathbf{x}^{(k)})^T DF(\mathbf{x}^{(k)}) + \lambda \mathbf{I} \right) \mathbf{s} = -DF(\mathbf{x}^{(k)}) F(\mathbf{x}^{(k)}) \, . \tag{6.5.8}$$

6.6 Essential Skills Learned in Chapter 6

6.5
p. 365

6.6
p. 36

You should know:

- several possibilities to solve linear least squares problems
- how to solve non-linear least squares problems

6.5
p. 366

6.6
p. 36