

3

Least Squares

3.1 Linear Least Squares

Linear least squares problem:

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

find: $\mathbf{x} \in \mathbb{K}^n$ such that

$$(i) \|\mathbf{Ax} - \mathbf{b}\|_2 = \inf\{\|\mathbf{Ay} - \mathbf{b}\|_2 : \mathbf{y} \in \mathbb{K}^n\}, \quad (3.1.1)$$

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

D-ITET,
D-MATL

Lemma 3.1.1 (Existence & uniqueness of solutions of the linear least squares problem).

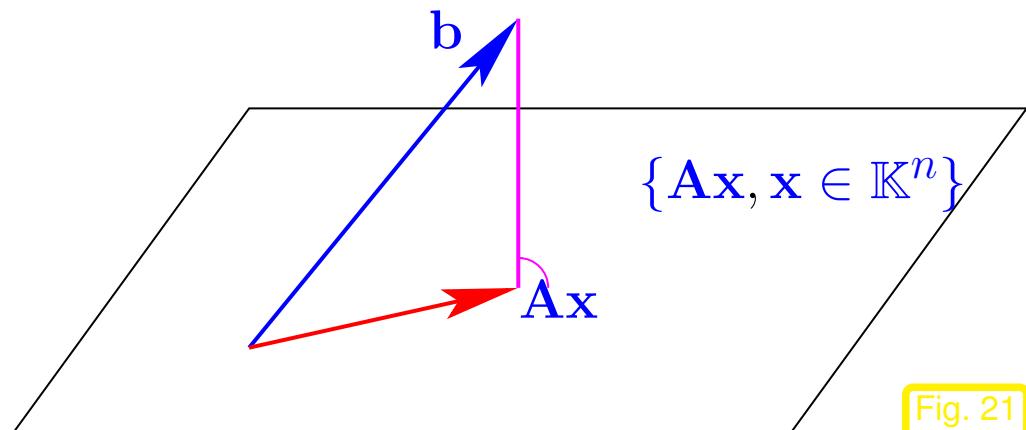
For every $\mathbf{A} \in \mathbb{K}^{m,n}$ and every $\mathbf{b} \in \mathbb{K}^m$ the linear least squares problem (3.1.1) has a unique solution.

$\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$ (“backslash”) solves (3.1.1) for $\mathbf{A} \in \mathbb{K}^{m,n}$ and $\mathbf{b} \in \mathbb{K}^m$ with $m \neq n$.

Reassuring: stable implementation (for dense matrices).

3.1.1 Normal Equations

Assume in this subsection that $\mathbf{A} \in \mathbb{K}^{m,n}$ has rank n , i.e., $\text{rank}(\mathbf{A}) = n$. This implies $m \geq n$.



Geometric interpretation of
linear least squares problem (3.1.1):

\mathbf{Ax} = orthogonal projection of \mathbf{b} on the subspace
 $\text{Im}(\mathbf{A}) := \text{Span} \{ (\mathbf{A})_{:,1}, \dots, (\mathbf{A})_{:,n} \}$.

Geometric interpretation: the least squares problem (3.1.1) amounts to searching the point $\mathbf{p} \in \text{Im}(\mathbf{A})$ nearest (w.r.t. Euclidean distance) to $\mathbf{b} \in \mathbb{K}^m$.

Hence, \mathbf{p} is the orthogonal projection of \mathbf{b} onto $\text{Im}(\mathbf{A})$, i.e., $\mathbf{b} - \mathbf{p} \perp \text{Im}(\mathbf{A})$. Note:

$$\mathbf{b} - \mathbf{p} \perp \text{Im}(\mathbf{A}) \Leftrightarrow \forall j \in \{1, \dots, n\}: \mathbf{b} - \mathbf{p} \perp (\mathbf{A})_{:,j} \Leftrightarrow \mathbf{A}^H(\mathbf{b} - \mathbf{p}) = 0. \quad (3.1.2)$$

Since $\text{rank}(\mathbf{A}) = n$, there exists an $\mathbf{x} \in \mathbb{K}^n$ such that $\mathbf{Ax} = \mathbf{p}$. Putting this into (3.1.2) results in

$\mathbf{A}^H \mathbf{Ax} = \mathbf{A}^H \mathbf{b}$

normal equation for (3.1.1) (3.1.3)

Notice: $\text{rank}(\mathbf{A}) = n \Rightarrow \mathbf{A}^H \mathbf{A} \in \mathbb{R}^{n,n}$ s.p.d. (symmetric positive definite)

Example 3.1.4 (Instability of normal equations).

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



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

D-ITET,
D-MATL

```

1 >> A = [ 1           1 ; ...
2               sqrt(eps) 0 ; ...
3               0           sqrt(eps) ];
4 >> rank(A)
5   ans = 2
6 >> rank(A'*A)
7   ans = 1

```



3.1.2 Orthogonal Transformation Methods

Idea: Note that $\|\mathbf{Ax} - \mathbf{b}\|_2 = \|\mathbf{U}(\mathbf{Ax} - \mathbf{b})\|_2 = \|(\mathbf{U}\mathbf{A})\mathbf{x} - \mathbf{Ub}\|_2$ for all unitary $\mathbf{U} \in \mathbb{K}^{m,m}$.

3.1.2.1 Solving linear least squares problem via QR decomposition

Assume in this subsection that $\mathbf{A} \in \mathbb{K}^{m,n}$ has rank n , i.e., $\text{rank}(\mathbf{A}) = n$. This implies $m \geq n$.

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

D-ITET,
D-MATL

$$\|\mathbf{Ax} - \mathbf{b}\|_2 = \|\mathbf{Q}(\mathbf{Rx} - \mathbf{Q}^H \mathbf{b})\|_2 = \|\mathbf{Rx} - \tilde{\mathbf{b}}\|_2 \quad \text{with} \quad \tilde{\mathbf{b}} := \mathbf{Q}^H \mathbf{b} .$$

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

$$\mathbf{x} = \left(\begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \tilde{b}_1 \\ \vdots \\ \tilde{b}_n \end{pmatrix} , \text{ residuum } \mathbf{r} = \mathbf{Q} \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \tilde{b}_{n+1} \\ \vdots \\ \tilde{b}_m \end{pmatrix} \right).$$

3.1.2.2 Solving linear least squares problem via SVD

In this subsection the general setting of the linear least squares problem (3.1.1) is considered!

Denote by r the rank of $\mathbf{A} \in \mathbb{K}^{m,n}$, i.e., $r = \text{rank}(\mathbf{A}) \leq \min(m, n)$.

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

$$\left(\begin{array}{|c|} \hline \mathbf{A} \\ \hline \end{array} \right) = \left(\begin{array}{|c|c|} \hline \mathbf{U}_1 & \mathbf{U}_2 \\ \hline \end{array} \right) \left(\begin{array}{|c|c|} \hline \Sigma_r & 0 \\ 0 & 0 \\ \hline \end{array} \right) \left(\begin{array}{|c|} \hline \mathbf{V}_1^H \\ \hline \mathbf{V}_2^H \\ \hline \end{array} \right), \quad (3.1.6)$$

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

$$\|\mathbf{Ax} - \mathbf{b}\|_2 = \left\| [\mathbf{U}_1 \quad \mathbf{U}_2] \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{V}_1^H \\ \mathbf{V}_2^H \end{pmatrix} \mathbf{x} - \mathbf{b} \right\|_2 = \left\| \begin{pmatrix} \Sigma_r \mathbf{V}_1^H \mathbf{x} \\ 0 \end{pmatrix} - \begin{pmatrix} \mathbf{U}_1^H \mathbf{b} \\ \mathbf{U}_2^H \mathbf{b} \end{pmatrix} \right\|_2 \quad (3.1.7)$$

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

➤ possibly *underdetermined* linear system $\Sigma_r \mathbf{V}_1^H \mathbf{x} = \mathbf{U}_1^H \mathbf{b} . \quad (3.1.8)$

To fix a unique solution we appeal to the **minimal norm condition** in (3.1.1): solution \mathbf{x} of (3.1.8) is unique up to contributions from $\text{Ker}((\mathbf{V}_1)^H) = \text{Im}(\mathbf{V}_2)$. Since \mathbf{V} is orthogonal, the minimal norm solution is obtained by setting contributions from $\text{Im}(\mathbf{V}_2)$ to zero, which amounts to choosing

$$\mathbf{x} \in \text{Im}(\mathbf{V}_1).$$

► solution $\mathbf{x} = (\mathbf{V}_1 \Sigma_r^{-1} \mathbf{U}_1^H) \mathbf{b} , \quad \|\mathbf{r}\|_2 = \|\mathbf{U}_2^H \mathbf{b}\|_2 . \quad (3.1.9)$

Code 3.1: Solving LSQ problem via SVD

```

1 function y = lsqsvd(A,b)
2 [U,S,V] = svd(A,0);
3 sv = diag(S);
4 r = max(find(sv/sv(1) > eps));
5 y = V(:,1:r)*(diag(1./sv(1:r)) * ...
   (U(:,1:r)'*b));
6
```

Practical implementation:

“numerical rank” test:

$$r = \max\{i: \sigma_i/\sigma_1 > \text{tol}\}$$

3.2 Non-linear Least Squares

Non-linear least squares problem

Given: $m, n \in \mathbb{N}$ with $m > n$, $D \subset \mathbb{R}^n$, $F = (F_1, \dots, F_m) : D \subset \mathbb{R}^n \rightarrow \mathbb{R}^m$.

Find: $\mathbf{x}^* \in D$: $\|F(\mathbf{x}^*)\|_2 = \inf_{x \in D} \|F(x)\|_2$ (3.2.1)

Terminology: $D \hat{=} \text{parameter space}$, $x_1, \dots, x_n \hat{=} \text{parameter}$.

We assume

- the unique existence of a solution $\mathbf{x}^* \in D$ of the non-linear least squares problem (3.2.1),
- “independence for each parameter”: \mathbf{x}^* is an interior point of D , F is continuously differentiable in a neighbourhood of \mathbf{x}^* and $\text{rank}(F'(\mathbf{x}^*)) = n$. This implies

\exists neighbourhood $\mathcal{U}(\mathbf{x}^*)$ of \mathbf{x}^* : $\forall \mathbf{x} \in \mathcal{U}(\mathbf{x}^*)$: $\text{rank}(F'(\mathbf{x})) = n$. (3.2.2)

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

3.2.1 Newton method

Define auxiliary function $\Phi: D \rightarrow [0, \infty)$ by $\Phi(x) := \frac{1}{2}\|F(\mathbf{x})\|_2^2$ for all $x \in D$. Then

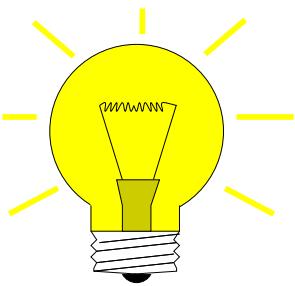
$$\Phi(\mathbf{x}^*) = \min_{x \in D} \Phi(x) \quad \Rightarrow \quad (\mathbf{grad} \Phi)(\mathbf{x}^*) = 0 \quad .$$

Simple idea: use (under suitable assumptions on F) Newton's method to determine a zero of $\mathbf{grad} \Phi: D \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$ given by $(\mathbf{grad} \Phi)(x) = (\Phi'(x))^T = (F'(x))^T F(x)$ for all $x \in D$.

► Newton iterates $\mathbf{x}^{(k)} \in D, k \in \mathbb{N}_0$, satisfy for all $k \in \mathbb{N}_0$:

$$\underbrace{\left((F'(\mathbf{x}^{(k)}))^T F'(\mathbf{x}^{(k)}) + \sum_{j=1}^m F_j(\mathbf{x}^{(k)}) (\mathbf{Hess} F_j)(\mathbf{x}^{(k)}) \right)}_{=(\mathbf{Hess} \Phi)(\mathbf{x}^{(k)})=(\mathbf{grad} \Phi)'(\mathbf{x}^{(k)})} \left(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)} \right) = - \underbrace{(F'(\mathbf{x}^{(k)}))^T F(\mathbf{x}^{(k)})}_{=(\mathbf{grad} \Phi)(\mathbf{x}^{(k)})} \quad (3.2.4)$$

3.2.2 Gauss-Newton method



Idea: local linearization of F : $F(x) \approx F(y) + F'(y)(x - y)$

➤ sequence of *linear* least squares problems

► Gauss-Newton iteration

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

$$\mathbf{x}^{(k+1)} := \mathbf{x}^{(k)} - \mathbf{s} \quad , \quad \mathbf{s} := \operatorname{argmin}_{\mathbf{v} \in \mathbb{R}^n} \|F(\mathbf{x}^{(k)}) - F'(\mathbf{x}^{(k)}) \mathbf{v}\|_2 . \quad (3.2.6)$$

linear least squares problem

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

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

$$\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$$



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

Code 3.2: template for Gauss-Newton method

```

1 function  $\mathbf{x} = \text{gn}(\mathbf{x}, \mathbf{F}, \mathbf{J}, \text{tol})$ 
2  $\mathbf{s} = \mathbf{J}(\mathbf{x}) \backslash \mathbf{F}(\mathbf{x});$  %
3  $\mathbf{x} = \mathbf{x} - \mathbf{s};$ 
4 while ( $\text{norm}(\mathbf{s}) > \text{tol} * \text{norm}(\mathbf{x})$ ) %
5    $\mathbf{s} = \mathbf{J}(\mathbf{x}) \backslash \mathbf{F}(\mathbf{x});$  %
6    $\mathbf{x} = \mathbf{x} - \mathbf{s};$ 
7 end

```

Gauss-Newton method vs. Newton method:

Advantage of the Gauss-Newton method : second derivative of \mathbf{F} not needed.

Drawback of the Gauss-Newton method : no local quadratic convergence.

3.2.2.1 An example: non-linear least squares data fitting

Given: data points $(t_i, y_i) \in \mathbb{R}^2, i \in \{1, \dots, m\}$, with measurements errors and a nonlinear and smooth function $f: \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}$

Goal roughly: Find parameter $\mathbf{x}^* \in \mathbb{R}^n$: $\forall i \in \{1, \dots, n\}$: $y_i \approx f(t_i, \mathbf{x}^*)$

Goal precise: Find parameter $\mathbf{x}^* \in \mathbb{R}^n$: $\underbrace{\sum_{i=1}^n |f(t_i, \mathbf{x}^*) - y_i|^2}_{=\|F(\mathbf{x}^*)\|_2^2} = \inf_{x \in \mathbb{R}^n} \underbrace{\sum_{i=1}^n |f(t_i, \mathbf{x}) - y_i|^2}_{=\|F(\mathbf{x})\|_2^2}$

Determine parameters by non-linear least squares data fitting:

$$\mathbf{x}^* = \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} \sum_{i=1}^m |f(t_i, \mathbf{x}) - y_i|^2 = \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} \|F(\mathbf{x})\|_2^2 \quad (3.2.7)$$

with $F: \mathbb{R}^n \rightarrow \mathbb{R}^m$ given by $F(\mathbf{x}) := \begin{pmatrix} f(t_1, \mathbf{x}) - y_1 \\ \vdots \\ f(t_m, \mathbf{x}) - y_m \end{pmatrix}$ for all $\mathbf{x} \in \mathbb{R}^n$.

Example 3.2.3 (Non-linear data fitting).

Non-linear data fitting problem (3.2.7) for $n = 3$ and $f(t, x) = x_1 + x_2 \exp(-x_3 t)$ for all $t \in \mathbb{R}$ and all $x = (x_1, x_2, x_3) \in \mathbb{R}^3$. Then $F: \mathbb{R}^3 \rightarrow \mathbb{R}^m$ and

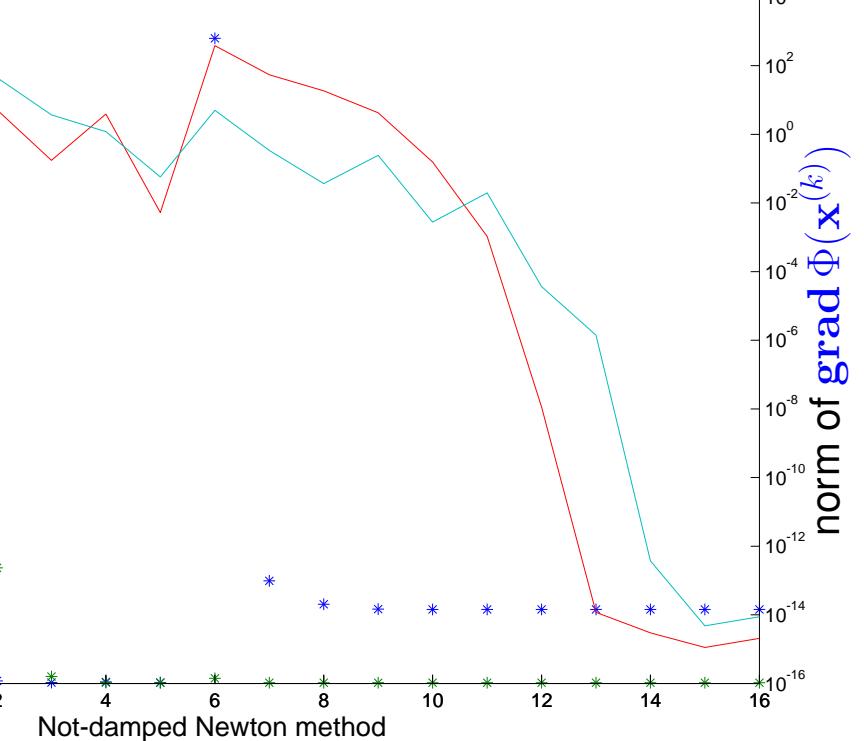
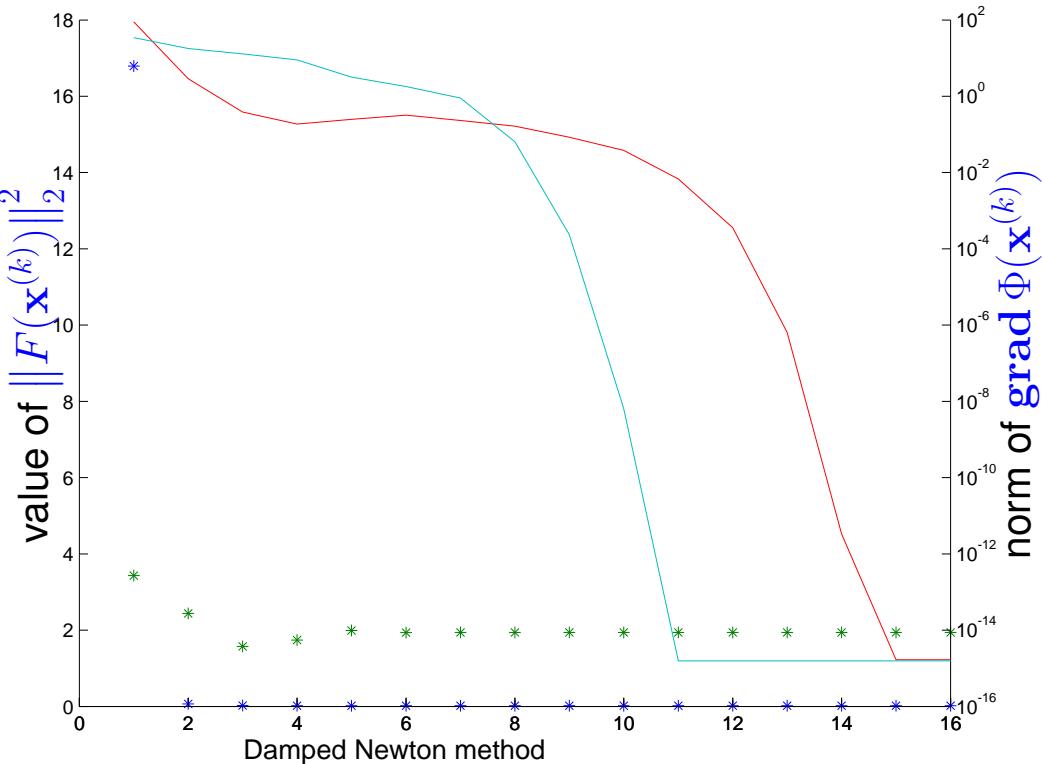
$$F(\mathbf{x}) = \begin{pmatrix} x_1 + x_2 \exp(-x_3 t_1) - y_1 \\ \vdots \\ x_1 + x_2 \exp(-x_3 t_m) - y_m \end{pmatrix}, \quad F'(\mathbf{x}) = \begin{pmatrix} 1 & e^{-x_3 t_1} & -x_2 t_1 e^{-x_3 t_1} \\ \vdots & \vdots & \vdots \\ 1 & e^{-x_3 t_m} & -x_2 t_m e^{-x_3 t_m} \end{pmatrix}$$

for all $t \in \mathbb{R}$ and all $x = (x_1, x_2, x_3) \in \mathbb{R}^3$.

Numerical experiment:

convergence of the Newton method and the Gauss-Newton method for different initial values

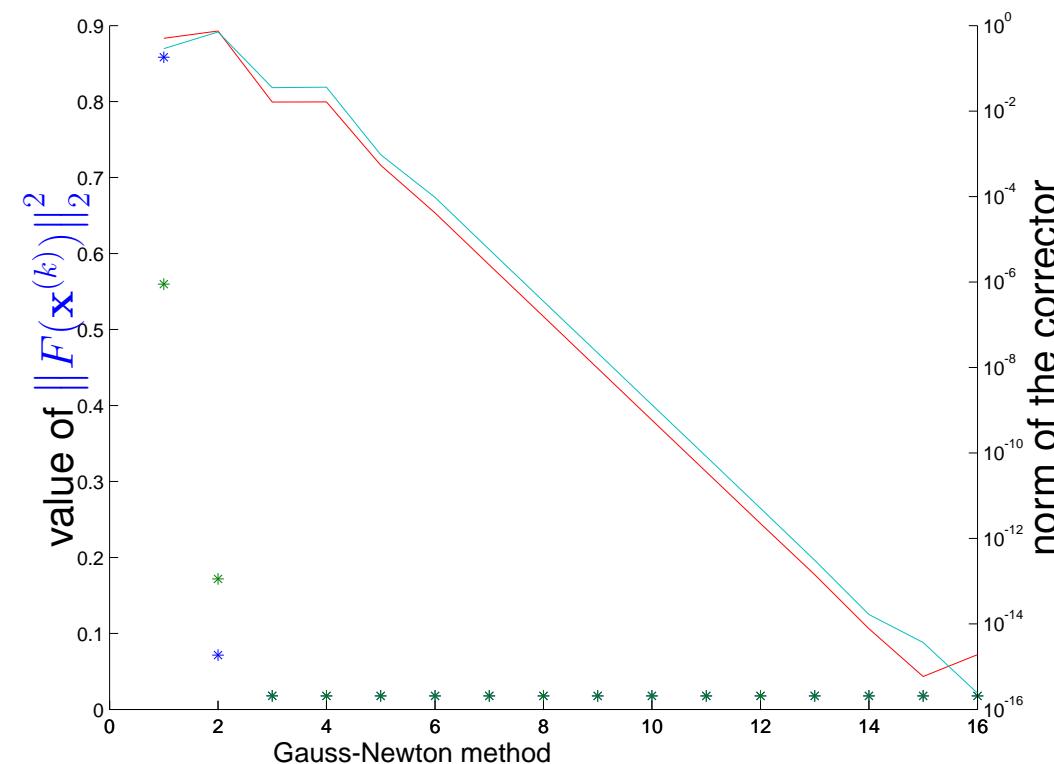
```
rand('seed', 0);
t = (1:0.3:7)';
y = x(1) + x(2)*exp(-x(3)*t);
y = y+0.1*(rand(length(y), 1)-0.5);
```



Convergence behaviour of the Newton method:

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

Gauss-Newton method:
initial value $(1.8, 1.8, 0.1)$ (red curve),
initial value $(1.5, 1.5, 0.1)$ (cyan curve),
convergence in both cases.
Notice: **linear convergence**.



X

Part II

Integration of Ordinary Differential Equations