Numerical methods

Approximation of functions
OUTLINE

1. Approximation and interpolation
2. Least-square method
   - basis functions
   - design matrix
   - residual
   - weighted least squares
   - normal equation
   - Gramian matrix
   - examples
   - solution of overdetermined systems
3. Repetition
Approximation and interpolation

To approximate function \( f(x) \) means to substitute it by a function \( \varphi(x) \), which is in some sense close to function \( f(x) \).

We will deal with two basic types of approximation: **interpolation** and **least-square method**

**Definition:** **Interpolation** is such approximation, in which the function \( \varphi(x) \) goes exactly through given points \([x_i, y_i]\), where \( y_i = f(x_i) \).

Sometimes we also require that functions \( f \) and \( \varphi \) have the same derivatives in points \( x_i \).
To approximate function $f(x)$ means
to substitute it by a function $\varphi(x)$,
which is in some sense close to function $f(x)$.

We will deal with two basic types of approximation:
**interpolation** and **least-square method**

**Definition:** **Least-square method** is such approximation,
in which $\varphi(x)$ is „interlaced‟
between given points $[x_i, y_i]$ in such a way,
that the „distance‟ between functions $f$ and $\varphi$ is
in some sense minimal.
Usually the function $\varphi(x)$ does not go through
points $[x_i, y_i]$. 
OUTLINE

1. Approximation and interpolation

2. Least-square method
   - basis functions
   - design matrix
   - residual
   - weighted least squares
   - normal equation
   - Gramian matrix
   - examples
   - solution of overdetermined systems

3. Repetition
Least-square method means procedure for approximate solution of overdetermined equations or inaccurately defined linear systems based on minimization of quadrate of residuals.

**Curve fitting** is an important group of problem, which could be solved by least-square method. We will describe what is it about.
Least-square method

Let $t$ is an independent variable, e.g. time, and $y(t)$ is an unknown function of variable $t$ we want to approximate.

Suppose that we performed $m$ measurements, i.e. values $y$ were measured for specified values of $t$:

$$y_i = y(t_i), \quad i = 1, 2, \ldots, m.$$  

Our aim is to model $y(t)$ using linear combination of $n$ basis functions for some $n \leq m$:

$$y(t) \approx x_1 \varphi_1(t) + x_2 \varphi_2(t) + \cdots + x_n \varphi_n(t) := R_n(t).$$

We propose basis function based on expected course of unknown function $y(t)$. Then we have to estimate parameters $x_1, x_2, \ldots, x_n$.

Function $R_n(t)$ is called (in statistics) linear regression function.
**Design matrix**

**Design matrix** $A$ of a model is an rectangular matrix, which has $m$ rows and $n$ columns:

$$
A = \begin{pmatrix}
\varphi_1(t_1) & \varphi_2(t_1) & \cdots & \varphi_n(t_1) \\
\varphi_1(t_2) & \varphi_2(t_2) & \cdots & \varphi_n(t_2) \\
\vdots & \vdots & \ddots & \vdots \\
\varphi_1(t_m) & \varphi_2(t_m) & \cdots & \varphi_n(t_m)
\end{pmatrix}
\equiv (\varphi_1, \varphi_2, \cdots, \varphi_n),
$$

where $\varphi_i = (\varphi_i(t_1), \varphi_i(t_2), \cdots, \varphi_i(t_m))^T$ is $i$-th column of $A$.

Matrix formulation of a model is

$$y \approx Ax,$$

where $y = (y_1, y_2, \ldots, y_m)^T$ are measured data and $x = (x_1, x_2, \ldots, x_n)^T$ is a vector of unknown parameters.
**Least-square method**

**Residuals** are differences between measured and modelled data:

\[
 r_i = y_i - R_n(t_i) = y_i - \sum_{j=1}^{n} \varphi_j(t_i) x_j \equiv y_i - \sum_{j=1}^{n} a_{ij} x_j , \quad i = 1, 2, \ldots, n ,
\]

where \( a_{ij} = \varphi_j(t_i) \).

In matrix form

\[
 r = y - Ax .
\]

We want to determine the parameters \( x_j \) in such a way, that the residual will be minimal.

We can derive least-square method by solving quadratic minimization problem:

\[
 \| r \|^2 := \sum_{i=1}^{m} r_i^2 \rightarrow \min .
\]
Least-square method

Principle of least squares

\[ r_i = y_i - R_n(t_i) \]

\[ \sum_{i=1}^{m} r_i^2 \rightarrow \text{min} \]
Approximate solution of overdetermined system $A x = y$
(i.e. we have more equations than unknowns),
which minimize the residual $r = y - Ax$,
is called

solution of linear system
by least squares.
Least-square method

Sometimes we have to use **weighted linear least squares**:

If measurements are not equally reliable, then we can assign to each measurement the weight \( w_i > 0 \) and then we minimize sum of weighted quadrates

\[
\| r \|_w^2 := \sum_{i=1}^{m} w_i r_i^2 \rightarrow \min.
\]

If, e.g. the error of \( i \)-th measurement is approximately equal to \( e_i \), we will choose \( w_i = 1/e_i \).

Each method for solving unweighted LSM is possible to use also for weighted LSM: it is enough to multiply \( y_i \) and \( i \)-th row of \( A \) by \( \sqrt{w_i} \).
OUTLINE

1. Approximation and interpolation
2. Least-square method
   - basis functions
   - design matrix
   - residual
   - weighted least squares
   - normal equation
   - Gramian matrix
   - examples
   - solution of overdetermined systems
3. Repetition
Normal equations

Solution of minimization problem

\[ \|r\|^2 := \sum_{i=1}^{m} r_i^2 \rightarrow \min \]

have to fulfill the necessary condition for extrema:

\[
\frac{\partial \|r\|^2}{\partial x_k} = \sum_{i=1}^{m} \left( y_i - \sum_{j=1}^{n} a_{ij} x_j \right)^2 = 0, \quad k = 1, 2, \ldots, n.
\]

After derivation we obtain

\[
\frac{\partial \|r\|^2}{\partial x_k} = 2 \sum_{i=1}^{m} \left( y_i - \sum_{j=1}^{n} a_{ij} x_j \right) (-a_{ik}) = 0
\]

and then

\[
\sum_{j=1}^{n} \left( \sum_{i=1}^{m} a_{ik} a_{ij} \right) x_j = \sum_{i=1}^{m} a_{ik} y_i, \quad k = 1, 2, \ldots, n,
\]

which could be written in matrix form as \(A^T Ax = A^T y\).
Normal equations

Linear system \( A^T A x = A^T y \) is known as **normal equations**.

If the columns of matrix \( A \) are linearly independent, then the matrix \( G := A^T A \) is positive definite and solution \( x^* \) of normal equations is the only solution of minimization problem, i.e. it holds

\[
\|y - Ax^*\|^2 = \min_x \|y - Ax\|^2.
\]
Normal equations

If we express the normal equations using vectors \( \varphi_i \), we get

\[
\begin{pmatrix}
(\varphi_1, \varphi_1) & (\varphi_1, \varphi_2) & \cdots & (\varphi_1, \varphi_n) \\
(\varphi_2, \varphi_1) & (\varphi_2, \varphi_2) & \cdots & (\varphi_2, \varphi_n) \\
\vdots & \vdots & \ddots & \vdots \\
(\varphi_n, \varphi_1) & (\varphi_n, \varphi_2) & \cdots & (\varphi_n, \varphi_n)
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
\vdots \\
x_n
\end{pmatrix}
=
\begin{pmatrix}
(\varphi_1, y) \\
(\varphi_2, y) \\
\vdots \\
(\varphi_n, y)
\end{pmatrix},
\]

where

\[
(\varphi_k, \varphi_j) = \sum_{i=1}^{m} \varphi_k(t_i)\varphi_j(t_i) \quad \text{and} \quad (\varphi_k, y) = \sum_{i=1}^{m} \varphi_k(t_i)y_i
\]

are scalar products of vectors \( \varphi_k, \varphi_j \) and \( \varphi_k, y \).

Matrix \( G \) of this system is called the **Gramian matrix** of a set of vectors \( \varphi_j, j = 1, 2, \ldots, n \).
During the design of approximation $R_n(t)$ we should select functions $\varphi_i(t)$ in such a way, that columns $\varphi_i$ of matrix $A$ to be linearly independent.

If it to be the contrary than it is possible to show that the minimization problem has infinite number of solutions which is obviously not desirable.
Normal equations

Two important special cases, for which the columns of matrix $A$ are linearly independent:

1. $\varphi_j(t)$ is a polynomial of degree $j-1$, e.g. $\varphi_j(t) = t^{j-1}$, $j = 1, 2, \ldots, n$;

2. for $n=2N+1$, where $N$ is a whole nonnegative integer, we chose

$$\begin{align*}
\varphi_1(t) &= 1, \\
\varphi_{2k}(t) &= \cos kt, \\
\varphi_{2k+1}(t) &= \sin kt, \quad k = 1, 2, \ldots, N,
\end{align*}$$

and „time of measurement“ $t_i$ we chose from interval $(c, c + 2\pi)$, where $c$ is an arbitrary number.

Approximation $R_n(t)$ is in the first case an algebraic polynomial and in the second case a trigonometric polynomial.
Normal equations

If $m=n$ and the matrix $A$ is regular, then $x^* = A^{-1}y$ and $r = 0$,
i.e. $R_n(t_i) = y_i$, $i = 1, 2, \ldots, m$.

However if measured data $y_i$ contains errors, then it is not practical,
that function $R_n(t)$ follows those errors.

On the contrary, we want that $R_n(t)$ authentically
reconstructs the unknown function $y(t)$,
therefore it is desirable,
that $R_n(t)$ **smooths** measured data.

This is possible only if the number of measurements $m$ is much larger than
the number of design parameters $n$,
i.e. for $m \gg n$. 
1. Approximation and interpolation

2. Least-square method
   - basis functions
   - design matrix
   - residual
   - weighted least squares
   - normal equation
   - Gramian matrix
   - examples
   - solution of overdetermined systems

3. Repetition
Example

For data given by the table

<table>
<thead>
<tr>
<th>$t_i$</th>
<th>0</th>
<th>0.5</th>
<th>1</th>
<th>1.5</th>
<th>2</th>
<th>2.5</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_i$</td>
<td>3.57</td>
<td>2.99</td>
<td>2.62</td>
<td>2.33</td>
<td>2.22</td>
<td>2.10</td>
<td>2.05</td>
</tr>
</tbody>
</table>

estimate an approximation $R_2(t) = x_1 + x_2 e^{-t}$

using least squares.

Obviously $\varphi_1(t) = 1$ and $\varphi_2(t) = e^{-t}$.

Normal equations are

$$
\begin{pmatrix}
\sum_{i=1}^{7} 1 \cdot 1 & \sum_{i=1}^{7} 1 \cdot e^{-t_i} \\
\sum_{i=1}^{7} e^{-t_i} \cdot 1 & \sum_{i=1}^{7} e^{-t_i} \cdot e^{-t_i}
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2
\end{pmatrix}
= 
\begin{pmatrix}
\sum_{i=1}^{7} 1 \cdot y_i \\
\sum_{i=1}^{7} e^{-t_i} \cdot y_i
\end{pmatrix}.
$$
Example

For data given by the table

<table>
<thead>
<tr>
<th>$t_i$</th>
<th>0</th>
<th>0,5</th>
<th>1</th>
<th>1,5</th>
<th>2</th>
<th>2,5</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_i$</td>
<td>3,57</td>
<td>2,99</td>
<td>2,62</td>
<td>2,33</td>
<td>2,22</td>
<td>2,10</td>
<td>2,05</td>
</tr>
</tbody>
</table>

estimate an approximation $R_2(t) = x_1 + x_2 e^{-t}$ using least squares.

After computations of sums, we get system

$$
\begin{pmatrix}
7 & 2,4647 \\
2,4647 & 1,5805
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2
\end{pmatrix}
= 
\begin{pmatrix}
17,88 \\
7,4422
\end{pmatrix},
$$

which solution is $x_1 \doteq 1,9879$ and $x_2 \doteq 1,6087$.

Seeking approximation is

$R_2(t) \doteq 1,99 + 1,61 e^{-t}$ and $\|r\| \doteq 0,0651$. 
Example

For data given by the table

\[
\begin{array}{c|ccccccc}
 t_i & 0 & 0.5 & 1 & 1.5 & 2 & 2.5 & 3 \\
 y_i & 3.57 & 2.99 & 2.62 & 2.33 & 2.22 & 2.10 & 2.05 \\
\end{array}
\]

gradually we will approximate the data by polynomials of the first, second and third degree.

As basis functions we chose

\[ \varphi_j(t) = t^{j-1}, \ j = 1, 2, \ldots, n \ \text{and} \ n = 2, 3, 4. \]
Polynomial of first degree

For \( \varphi_1(t) = 1 \) and \( \varphi_2(t) = t \) we obtain normal equations

\[
\begin{pmatrix} 7 & 10.5 \\ 10.5 & 22.75 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 17.88 \\ 23.45 \end{pmatrix},
\]

which has solution

\( x_1 = 3.28 \) and \( x_2 = -0.48 \) therefore

\[
R_2(t) = 3.28 - 0.48t \quad \|r\| = 0.4756.
\]

Approximation by linear polynomial is not good.
**Example**

**Polynomial of second degree**

Normal equations

\[
\begin{pmatrix}
7 & 10.5 & 22.75 \\
10.5 & 22.75 & 55.125 \\
22.75 & 55.125 & 142.1875
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3
\end{pmatrix}
= 
\begin{pmatrix}
17.88 \\
23.45 \\
49.0625
\end{pmatrix}
\]

have solution

\[x_1 = 3.53, \ x_2 = -1.09, \ x_3 = 0.20,\]

\[R_3(t) = 3.53 - 1.09t + 0.2t^2, \quad \|r\| = 0.1006.\]

The residual is smaller but still large than in the previous example.
Example

Polynomial of the third degree

Normal equations

\[
\begin{pmatrix}
7 & 10,5 & 22,75 & 55,125 \\
10,5 & 22,75 & 55,125 & 142,1875 \\
22,75 & 55,125 & 142,1875 & 381,2813 \\
55,125 & 142,1875 & 381,2813 & 1049,5469 \\
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_2 \\
x_3 \\
x_4 \\
\end{pmatrix}
=
\begin{pmatrix}
17,88 \\
23,45 \\
49,0625 \\
116,78 \\
\end{pmatrix}
\]

have solution

\[x_1 \doteq 3,57, \quad x_2 \doteq -1,35, \quad x_3 \doteq 0,43, \quad x_4 \doteq -0,05,\]

\[R_4(t) \doteq 3,57 - 1,35t + 0,43t^2 - 0,05t^3, \quad \|r\| \doteq 0,0360.\]
Example

If we enlarge the degree of the polynomial, we could see that polynomial $R_7(t)$ of the sixth degree passes through all points $[t_i, y_i]$, so we will obtain interpolating polynomial.

Notice the elements of Gramian matrices: the larger degree, the larger maximal coefficient.

This means that the condition number of Gramian matrices is growing.

Table shows the condition numbers $\kappa_2(G)$:

<table>
<thead>
<tr>
<th>$n$</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\kappa_2(G)$</td>
<td>16</td>
<td>4,27 $\cdot$ 10²</td>
<td>1,91 $\cdot$ 10⁴</td>
<td>1,20 $\cdot$ 10⁶</td>
<td>1,17 $\cdot$ 10⁸</td>
<td>2,31 $\cdot$ 10¹⁰</td>
</tr>
</tbody>
</table>
OUTLINE

1. Approximation and interpolation
2. Least-square method
   - basis functions
   - design matrix
   - residual
   - weighted least squares
   - normal equation
   - Gramian matrix
   - examples
     - solution of overdetermined systems
3. Repetition
Solution of overdetermined systems

Normal equations are not suitable for solving large overdetermined systems because the condition number of Gramian matrix is considerably larger than condition number of design matrix $A$, because

$$\kappa_2(A^T A) = [\kappa_2(A)]^2.$$

The better way is to use singular value decomposition or QR decomposition.
If the number of equations $M$ is less than the number of unknowns $N$, or if $M = N$ but equations are linearly dependent, then the system has no solution or it has more than one solution.

In the later case the space of solutions is given by a particular solution added to the any linear combination of $N - M$ vectors.

The task to find the space of solution for matrix $A$ is possible to solve using **singular value decomposition of matrix $A$**.
If the number of equations $M$ is greater than the number of unknowns $N$, then in general there is no solution vector and the system is called \textit{overdetermined}.

However, we could find a best “compromise” solution, which is “the closest” solution that satisfy all equations.

If „the closest“ we define in a sense of least square, i.e. the sum of square of residuals is the smallest possible, then the overdetermined system is reduced to solvable problem called the \textit{least square method}. 
Reduced system of equations could be written as system $N \times N$ equation

$$\left( A^T \cdot A \right) \cdot x = \left( A^T \cdot b \right).$$

This equations we call **normal equations** of a least square problem.

Singular value decomposition has many common features with the least square method, which we show later.

Direct solution of normal equations is in general not the best way to find a solution of least square problem.
SVD is based on the following theorem of linear algebra:

Each matrix $A$ of type $M \times N$, which the number of rows $M$ is greater or equal to the number of columns $N$, could be decomposed to a product of matrix with orthogonal columns $U$ of type $M \times N$, diagonal matrix $W$ of type $N \times N$ with positive or zero entries (singular values) and transpose orthogonal matrix $V$ of type $N \times N$.

$$
\begin{pmatrix}
A \\
\end{pmatrix}
= 
\begin{pmatrix}
U \\
\end{pmatrix}
\cdot 
\begin{pmatrix}
w_1 \\
\vdots \\
w_N \\
\end{pmatrix}
\cdot 
\begin{pmatrix}
V^T \\
\end{pmatrix}
$$
Orthogonality of matrices $\mathbf{U}$ and $\mathbf{V}$ could be written as

\[
\begin{pmatrix}
\mathbf{U}^T
\end{pmatrix}
\cdot
\begin{pmatrix}
\mathbf{U}
\end{pmatrix}
=
\begin{pmatrix}
\mathbf{V}^T
\end{pmatrix}
\cdot
\begin{pmatrix}
\mathbf{V}
\end{pmatrix}
=
\begin{pmatrix}
1
\end{pmatrix}
\]
Singular value decomposition

SVD could be done also if $M < N$.

In such a case the singular values $w_j$ for $j = M+1, \ldots, N$
are all zero
as well as corresponding columns of matrix $U$.

There is a lot of algorithms of SVD,
proven is subroutine \texttt{svdcmp} from \textit{Numerical Recipes}. 
If we have more equations than unknowns we are seeking solution in a least square sense.

We are solving system written as:

\[
\begin{pmatrix}
A & x & b
\end{pmatrix}
\]
After SVD of matrix $\mathbf{A}$ we have
the solution in a form

$$
\mathbf{x} = \mathbf{V} \cdot \text{diag}(1/w_j) \cdot \mathbf{U}^T \cdot \mathbf{b}
$$

In this case usually it is not necessary set to zero values $w_j$
however the unusually small values indicate
that the data are not sensitive to some parameters.
Another useful method for solving least squares is using QR decomposition of \(m \times n\) matrix \(A\) of normal equations, with \(m \geq n\).

**QR decomposition** is the product of an \(m \times m\) unitary matrix \(Q\) and an \(m \times n\) upper triangular matrix \(R\) which can be computed using e.g. the Gram–Schmidt process.

As the bottom \((m-n)\) rows of matrix \(R\) consist entirely of zeroes, it is often useful to partition \(R\), or both \(R\) and \(Q\):

\[
A = QR = Q \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = [Q_1, Q_2] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1,
\]

where \(R_1\) is an \(n \times n\) upper triangular matrix, \(0\) is an \((m-n) \times n\) zero matrix, \(Q_1\) is \(m \times n\), \(Q_2\) is \(m \times (m-n)\), and \(Q_1\) and \(Q_2\) both have orthogonal columns.
OUTLINE

1. Approximation and interpolation
2. Least-square method
   - basis functions
   - design matrix
   - residual
   - weighted least squares
   - normal equation
   - Gramian matrix
   - examples
   - solution of overdetermined systems
3. Repetition