Overdetermined Systems

**Definition:** The least squares solution of the overdetermined system $Ax \approx b$ is the vector $x$ that minimizes $||r||_2 = ||b - Ax||_2$.

Define $Q(x) = Q(x_1, x_2, \ldots, x_n) = ||b - Ax||_2^2$ where $x = (x_1, \ldots, x_n)$ and $A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m (m > n)$. The least squares solution is the set of values $x_1, \ldots, x_n$ that minimize $Q(x_1, x_2, \ldots, x_n)$!

\[
Q(x_1, \ldots, x_n) = ||b - Ax||_2^2 = ||r||_2^2 = \sum_{i=1}^{m} r_i^2
\]

\[
r = b - Ax \Rightarrow r_i = b_i - (Ax)_i = b_i - \sum_{j=1}^{n} a_{ij}x_j
\]

\[
\Rightarrow Q(x_1, \ldots, x_n) = \sum_{i=1}^{m} \left( b_i - \sum_{j=1}^{n} a_{ij}x_j \right)^2
\]

If $x_1, \ldots, x_n$ are those that minimize $Q$, then:

\[
\frac{\partial Q}{\partial x_1} = 0, \frac{\partial Q}{\partial x_2} = 0, \ldots, \frac{\partial Q}{\partial x_n} = 0
\]

in order to guarantee a minimum.
\[
\frac{\partial Q}{\partial x_k} = \frac{\partial}{\partial x_k} \left( \sum_{i=1}^{m} \left( b_i - \sum_{j=1}^{n} a_{ij}x_j \right)^2 \right)
\]
\[
= \sum_{i=1}^{m} \frac{\partial}{\partial x_k} \left( b_i - \sum_{j=1}^{n} a_{ij}x_j \right)^2
\]
\[
= \sum_{i=1}^{m} 2 \left( b_i - \sum_{j=1}^{n} a_{ij}x_j \right) \frac{\partial}{\partial x_k} \left( b_i - \sum_{j=1}^{n} a_{ij}x_j \right)
\]
\[
= \sum_{i=1}^{m} -2r_i a_{ik} = -2 \sum_{i=1}^{m} [A^T]_{ki} r_i = -2[A^T r]_k = 0
\]

\[\Rightarrow [A^T r]_k = 0\]

Thus,

\[
\begin{align*}
\frac{\partial Q}{\partial x_1} &= 0 \Rightarrow [A^T r]_1 = 0 \\
\frac{\partial Q}{\partial x_2} &= 0 \Rightarrow [A^T r]_2 = 0 \\
&\vdots \\
\frac{\partial Q}{\partial x_n} &= 0 \Rightarrow [A^T r]_n = 0
\end{align*}
\]

Since \( r = b - Ax \), we have:

\[
0 = A^T r = A^T (b - Ax) = A^T b - A^T Ax \Rightarrow A^T Ax = A^T b
\]

The system above is called the normal equations system; it is a square system that has as solution the least-squares approximation of \( Ax \approx b \)

\[
\begin{pmatrix}
A_{n \times m}^{T} \\
A_{n \times m}^{T} x_{n \times 1}
\end{pmatrix}
= \begin{pmatrix}
A_{m \times 1}^{T} & b_{m \times 1}
\end{pmatrix}
\]

The normal equations always have a solution (with the simple condition that the columns of \( A \) have to be linearly independent - usually true).

**Problem:** The condition number of \( A^T A \) is the square of that of \( A \) (if \( A \) was square itself!).

**QR factorization**

An alternative method that does not suffer from this problematic conditioning is QR factorization.
Definition: An $n \times n$ matrix $Q$ is called orthonormal if and only if

$$Q^T Q = QQ^T = I$$

Theorem 1. Let $A \in \mathbb{R}^{m \times n}$ ($m > n$) have linearly independent columns. Then a decomposition $A = QR$ exists, such that $Q \in \mathbb{R}^{m \times m}$ is orthonormal and $R \in \mathbb{R}^{m \times n}$ is upper triangular, i.e.,

$$R = \begin{pmatrix} \hat{R} \\ O \end{pmatrix}$$

where $\hat{R}$ is an $n \times n$ upper triangular matrix. Additionally, given that $A$ has linearly independent columns, all diagonal elements $r_{ii} \neq 0$.

Now, let us write

$$Q = \begin{bmatrix} \hat{Q} & Q^* \end{bmatrix}$$

where $\hat{Q} \in \mathbb{R}^{m \times n}$ contains the first $n$ columns of $Q$ and $Q^* \in \mathbb{R}^{m \times (m-n)}$ contains the last $(m-n)$ columns. Respectively, we write:

$$R = \begin{pmatrix} \hat{R} \\ O \end{pmatrix}$$

where $\hat{R} \in \mathbb{R}^{n \times n}$ (and upper triangular) contains the first $n$ rows of $R$. $\hat{R}$ is also non-singular because it has linearly independent columns.

We can verify the following:

$$\hat{Q}^T \hat{Q} = I_{n \times n} \quad (\text{although } \hat{Q} \hat{Q}^T \neq I_{m \times m})$$

Proof.

$$[\hat{Q}^T \hat{Q}]_{ij} = \sum_{k=1}^{m} [\hat{Q}^T]_{ik}[\hat{Q}]_{kj}$$

$$= \sum_{k=1}^{m} [\hat{Q}]_{ki}[\hat{Q}]_{kj} = \sum_{k=1}^{m} [Q]_{ki}[Q]_{kj}$$

$$= [Q^T Q]_{ij} = [I_{m \times m}]_{ij}$$

The factorization $A = \hat{Q} \hat{R}$ is the so-called economy size $QR$ factorization. Once we have $\hat{Q}$ and $\hat{R}$ computed, we observe that the normal equations can be written as:
\[ \begin{align*}
A^T Ax &= A^T b \\
\Rightarrow \hat{R}^T \hat{Q}^T \hat{Q} \hat{R} &= \hat{R}^T \hat{Q}^T b \\
\Rightarrow \hat{R}^T \hat{R} &= \hat{R}^T \hat{Q}^T b \\
\Rightarrow \hat{R}x &= \hat{Q}^T b \quad (1)
\end{align*} \]

The last equality follows because \( \hat{R} \) is invertible.

**Benefit:** We can show that \( \text{cond}(A^T A) = [\text{cond}(\hat{R})]^2 \), thus equation (1) is much better conditioned than the normal equations system!

**Numerical Integration**

We seek an algorithm to approximate the definite integral:

\[ I = \int_a^b f(x) \, dx \]

or, the area below the graph of \( y = f(x) \). Of course, in the fortuitous case where we know a function \( F(x) \) (the anti-derivative of \( f \)), such that \( F'(x) = f(x) \), we can write:

\[ \int_a^b f(x) \, dx = F(b) - F(a) \]

For example, \( \text{arctan}(x)' = 1/(1 + x^2) \), thus

\[ \int_a^b \frac{dx}{1 + x^2} = \text{arctan}(b) - \text{arctan}(a) \]

However, this is not a practical algorithm, since:

- The anti-derivative is not generally known.
- Often, the anti-derivative may be significantly more expensive to evaluate than \( f(x) \) itself. For example, compare \( f(x) = 1/(1 + x^2) \) (easy) with \( F(x) = \text{arctan}(x) \) (expensive).

Our general solution methodology will be as follows:

- Subdivide the interval of integration using the \( n + 1 \) points \( \{x_i\}_{i=0}^n \) with

\[ a = x_0 < x_1 < x_2 < \ldots < x_{n-1} < x_n = b \]
In each interval \([x_i, x_{i+1}]\), approximate \(f(x)\) with some simpler function, say a polynomial \(P_i(x)\) which is easy to integrate. Approximate

\[
I_i = \int_{x_i}^{x_{i+1}} f(x)\,dx \approx \int_{x_i}^{x_{i+1}} P_i(x)\,dx
\]

- Compute the integral

\[
I = \int_a^b f(x)\,dx = \sum_{i=0}^{n-1} I_i \approx \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} P_i(x)\,dx
\]

**Example:** The rectangle rule: at each interval \([x_i, x_{i+1}]\) use the approximation \(P_i(x) = f(x_i)\) (the left end point!)

Thus, we approximate:

\[
I_i = \int_{x_i}^{x_{i+1}} f(x)\,dx \approx \int_{x_i}^{x_{i+1}} f(x_i)\,dx = f(x_i)(x_{i+1} - x_i)
\]

In the case where \(x_{i+1} - x_i = h = \text{constant}\), we can write

\[
I = \int_a^b f(x)\,dx = \sum_{i=0}^{n-1} I_i \approx \sum_{i=0}^{n-1} f(x_i) \cdot h = \frac{b-a}{n} \sum_{i=0}^{n-1} f(x_i)
\]

As in the case of interpolation, we can assess the error incurred by this approximation. There are two errors we actually focus on:

- The local error \(|\int_{x_i}^{x_{i+1}} (f(x) - P_i(x))|\) at each subinterval \([x_i, x_{i+1}]\).
- The global error for the entire integral \(\int_a^b f(x)\,dx\).