Part 2 - Numerics

1 The Numerical Problem

1.1 Well-posedness

Definition 1.1.1. A mathematical problem in numerics consists of finding $x \in X$, where X is some space, such that

$$F(x, d) = 0.$$

for a given function $F : X \times D \to \mathbb{R}$ and given data $d \in D$

Example. 1. $F : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}, F(x, d) = \|Ax - d\|_2$ for $A \in \mathbb{R}^{m \times n}, d \in \mathbb{R}^m$. Solutions solve the lin. system $Ax = d$

2. $F : \mathbb{R} \times \mathbb{R} \to \mathbb{R}, F(x, d) = \left(\int_0^d e^{-\xi^2} \mathrm{d}\xi\right) - x$

Definition 1.1.2. A mathematical problem is called well-posed. if for all $d \in D$ a unique solution $x \in X$ exists and the solution depends continuously on $d \in D$.

Remark. Here, we will only consider well-posed problems, so that there exists

$$\varphi : D \to X, F(\varphi(d), d) = 0 \ \forall d \in D.$$

Solving a problem thus reduces to evaluating φ at $d \in D$.

Typically, there will be errors associated with the given data (e.g. Round-off errors, measurement errors). An important part of numerics is the estimation how such errors affect the computed solution. (Conditioning and stability)

We will also be concerned with computability of solutions (Algorithmics). Often, it is convenient to approximate the "true" solution $\varphi(d)$ by an easier to compute function $\varphi_\varepsilon(d)$, this leads to questions of convergence.

1.2 Numerical complexity

Of course, the number of elementary computations that have to be performed to obtain a result is important (Numerical complexity). Typically, one is interested in the scaling of the number of computations with respect to the size of the problem, e.g. , is solving $Ax = b$ for $A \in \mathbb{R}^{n\times m}$ taking $\mathcal{O}(n^2), \mathcal{O}(n^3), \mathcal{O}(n!)$ operations.

$$m(n) = \mathcal{O}(f(n)) \text{ if } |\frac{m(n)}{f(n)}| \overset{n\to\infty}{\to} c.$$

2 Matrix factorizations

We want to find L, U s.t. linear systems $Lx = b, Uy = c$ are easy to solve and $A = LU$. Then solving $Ax = b$ is easy

1. Solve $Ly = b$
2. Solve $Ux = y$

2.1 Triangular matrices

Definition 2.1.1. $L = (\ell_{ij})_{ij=1}^n \in \mathbb{R}^{n\times n}$ is called lower triangular if $\ell_{ij} = 0$ for $i < j$.
$U = (u_{ij})_{ij=1}^n \in \mathbb{R}^{n\times n}$ is called upper triangular if U^T is lower triangular.
A triangular matrix $D \in \mathbb{R}^{n\times n}$ is called normalized if $d_{ii} = 1$ for $i = 1, \ldots, n$.

Algorithm 2.1.2 (Back substitution)**.** *Let $U \in \mathbb{R}^{n\times n}$ be a regular upper triangular matrix, and let $b \in \mathbb{R}^n$. Compute $x \in \mathbb{R}^n$ by:*
for $i = n : -1 : 1$
$x_i = \left(b_i - \sum_{j=i+1}^n u_{ij}x_j\right) \frac{1}{u_{ii}}$

Remark. The total number of basic computations for backsubstitution is $\mathcal{O}(n^2)$

Lemma 2.1.3. *Let $U, V \in \mathbb{R}^{n\times n}$ be upper triangular, then UV is upper triangular. If U is also regular, then U^{-1} is upper triangular with $(U^{-1})_{ii} = \frac{1}{u_{ii}}$.*

2.2 LU-factorization

Definition 2.2.1. A factorization $A = LU$ with $L \in \mathbb{R}^{n\times n}$ lower and $U \in \mathbb{R}^{n\times n}$ upper triangular is called LU-factorization of $A \in \mathbb{R}^{n\times n}$. It is called normalized if L is normalized.

Theorem 2.2.2. *Let $A \in \mathbb{R}^{n\times n}$ be a regular matrix. TFAE*

1. *$\exists!$ normalized LU-factorization of A*
2. *All submatrices $A_k = (a_{ij})_{ij=1}^{k} \in \mathbb{R}^{k\times k}$ are regular.*

Lemma 2.2.3. *Let $A = LU$ be a normalized LU-factorization. We then have*

$$a_{ik} = u_{ik} + \sum_{j=1}^{i-1} \ell_{ij} u_{jk}, \quad a_{ki} = \ell_{ki} u_{ii} + \sum_{j=1}^{i-1} \ell_{kj} u_{ji}.$$

Solving for the respective non-trivial entries of L and U, we obtain

Algorithm 2.2.4. *Let $A \in \mathbb{R}^{n\times n}$ admit a normalized LU decomposition. The non-trivial entries of L and U can be computed by*

for *i=1:n* **do**
  for *k=i:n* **do**
    $u_{ij} = a_{ik} - \sum_{j=1}^{i-1} \ell_{ij} u_{jk}$
  end
  for *k=i+1:n* **do**
    $\ell_{ki} = \left(a_{ki} - \sum_{j=1}^{i-1} \ell_{kj} u_{ji}\right) / u_{ii}$
  end
end

Remark. The numerical complexity of the computation of an LU-decomposition of an $n \times n$ matrix is $\mathcal{O}(n^3)$.

Example. 1. The matrix

$$A = \begin{pmatrix} 1 & 1 \\ 1 & \epsilon \end{pmatrix}$$

admits a LU-decomposition and is regular. However, $A^{-1}\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}1\\0\end{pmatrix}$ and $A^{-1}\begin{pmatrix}1\\1+\epsilon\end{pmatrix} = \begin{pmatrix}0\\1\end{pmatrix}$. We thus see that a small change in the data results in a large change in the result. This is an issue of *conditioning* of the problem.

2. The matrix
$$A = \begin{pmatrix}\epsilon & 1\\ 1 & 0\end{pmatrix}$$
has no problem with conditioning. Its inverse is
$$A^{-1} = \begin{pmatrix}0 & 1\\ 1 & -\epsilon\end{pmatrix},$$
and one can easily see that solutions of the linear system $Ax = b$ change one the same order as the data b changes. The LU-factorization of A, however, is given by
$$L = \begin{pmatrix}1 & 0\\ \frac{1}{\epsilon} & 1\end{pmatrix}, \quad U = \begin{pmatrix}\epsilon & 1\\ 0 & -\frac{1}{\epsilon}\end{pmatrix}.$$
Thus, a small (e.g. roundoff) error introduced for example in between solving the two triangular systems may become large. This is an issue of *stability* of the algorithm.

2.3 Cholesky factorization

If A is symmetric, then only $n(n+1)/2$ entries of A are relevant. We may be able to take advantage of this. Assume that $A = LL^T$ for $A, L \in \mathbb{R}^{n\times n}$ and L lower triangular. We compute
$$\begin{aligned} A^T &= \left(LL^T\right)^T = LL^T = A,\\ x^TAx &= x^T\left(LL^T\right)x = \left(L^Tx\right)^T\left(L^Tx\right) = \left\|L^Tx\right\|_2^2 \geq 0. \end{aligned}$$
The matrix A must thus be symmetric for such a factorization to exist, and if A or L is regular, A must be positive definite.

Lemma 2.3.1. *If A is symmetric and positive definite, then* $\det A > 0$ *and all submatrices A_k of A are positive definite.*

Definition 2.3.2. A factorization $A = LL^T$ with lower triangular matrix $L \in \mathbb{R}^{n\times n}$ is called Cholesky factorization.

Theorem 2.3.3. *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric, positive definite matrix. Then there exists a unique Cholesky factorization $A = LL^T$ of A with $\ell_{ii} > 0$ for $i = 1 \dots n$.*

Lemma 2.3.4. *If $A = LL^T$, we have*

$$a_{ik} = \begin{cases} \ell_{ik}\ell_{kk} + \sum_{j=1}^{k-1} \ell_{ij}\ell_{kj} & \text{for } i > k, \\ \ell_{kk}^2 + \sum_{j=1}^{k-1} \ell_{kj}^2 & \text{for } i = k. \end{cases}$$

Solving for the entries of L again leads to

Algorithm 2.3.5. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric and positive definite with Cholesky factorization $A = LL^T$. The non-trivial entries of L can be computed by*

for *k=1:n* **do**
  $\ell_{kk} = \left(a_{kk} - \sum_{j=1}^{k-1} \ell_{kj}^2\right)^{1/2}$
  for *i=k+1:n* **do**
    $\ell_{ik} = \left(a_{ik} - \sum_{j=1}^{k-1} \ell_{ij}\ell_{kj}\right) / \ell_{kk}$
  end
end

3 Conditioning and stability

In the following, we consider well-posed numerical problems $\phi\colon D \to X$, where D and X are suitable spaces endowed with a norm.

3.1 Condition number

Definition 3.1.1. The (relative) condition number $\kappa_\phi(d)$ of $\phi\colon D \to X$ at $x \neq 0$ is given by

$$\kappa_\phi(d) = \inf\left\{\kappa \geq 0 : \exists \delta > 0 \text{ so that } \frac{\|\phi(d+\Delta d) - \phi(d)\|}{\|\phi(d)\|} \leq \kappa \frac{\|\Delta d\|}{\|d\|} \right.$$
$$\left. \text{for all } \Delta d \in D \text{ with } \frac{\|\Delta d\|}{\|d\|} < \delta \right\}$$

A problem is called ill-conditioned if $\kappa_\phi(d) \gg 1$.

Theorem 3.1.2. *$\phi\colon D \to X$ is differentiable at $d \in D$, then*

$$\kappa_\phi(d) = \frac{\|\,\mathrm{D}\,\phi(d)\|\|d\|}{\|\phi(d)\|}.$$

3.2 Stability

Definition 3.2.1. An algorithm for the (possibly approximate) computation of the numerical problem $\phi\colon D \to X$ is a map $\tilde{\phi}\colon D \to X$, which is given by the consecutive application of elementary computations, i.e.,

$$\tilde{\phi} = f_J \circ f_{J-1} \circ \cdots \circ f_2 \circ f_1.$$

Definition 3.2.2. An algorithm $\tilde{\phi}$ is called unstable if there is a perturbation $\tilde{d}$ of d such that the error introduced by inexact elementary computations in the algorithm is significantly larger than the error introduced by the perturbation itself, i.e.,

$$\frac{|\tilde{\phi}(\tilde{d}) - \phi(d)|}{|\phi(d)|} \gg \frac{|\phi(\tilde{d}) - \phi(d)|}{|\phi(d)|}.$$

An algorithm is called stable if it is not unstable.

Example. The problem of evaluating the function

$$\phi(d) = \frac{1}{d} - \frac{1}{d+1} = \frac{1}{d(d+1)}$$

is stable for large numbers d, as for $\tilde{d} = (1+\epsilon_d)d$, we have $\phi(d) - \tilde{\phi}(d) \approx \frac{2\epsilon_d d^2}{d^4}$. The outcome of the two possible algorithms below (where parenthesis indicate order of computation), however, differs substantially.

$$\tilde{\phi}_1(d) = \left(\frac{1}{d}\right) - \left(\frac{1}{d+1}\right), \quad \tilde{\phi}_2(d) = \frac{1}{(d(d+1))}$$

3.3 Condition number of linear operators

Definition 3.3.1. Given norms $\|\cdot\|_{\mathbb{R}^m}$ and $\|\cdot\|_{\mathbb{R}^n}$ on $\mathbb{R}^m$ and $\mathbb{R}^n$, respectively, the (induced) operator norm for $A \in \mathbb{R}^{m\times n}$ is given by

$$\|A\|_{op} = \sup_{x\in\mathbb{R}^n,\, \|x\|_{\mathbb{R}^n}=1} \|Ax\|_{\mathbb{R}^m}.$$

Lemma 3.3.2. *Given norms $\|\cdot\|$ on $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively, let $\|\cdot\|_{op}$ be the induced operator norm on $\mathbb{R}^{m\times n}$. We then have*

1. *$\|\cdot\|_{op}$ defines a norm $\mathbb{R}^{m\times n}$*
2. *$\|A\|_{op} = \sup_{x\in\mathbb{R}^*.|x|=1} \|Ax\| = \inf\{c > 0 : \forall x \in \mathbb{R}^n \|Ax\| \le c\|x\|\}$*
3. *for $A \neq 0$ and $x \in \mathbb{R}^n$ with $\|x\| \le 1$ and $\|Ax\| = \|A\|_{op}$ we have $\|x\| = 1$*
4. *the infimum and supremum in 2. are attained.*

Lemma 3.3.3. *Given norms $\|\cdot\|$ on $\mathbb{R}^n$ and $\mathbb{R}^m$, respectively, denote by $\|\cdot\|$ the induced operator norm on $\mathbb{R}^{m\times n}$. We then have*

1. *For $A \in \mathbb{R}^{exm}$ and $B \in \mathbb{R}^{m\times n}$ we have $\|AB\| \le \|A\|\|B\|$.*
2. *The identity matrix $\mathrm{Id} \in \mathbb{R}^{n\times n}$ satisfies $\|I_n\| = 1$.*
3. *Any operator norm $\mathbb{R}^{n\times n}$ satisfies $\|A\|_{op} \ge |\lambda|$ for all symmetric matrices $A \in \mathbb{R}^{n\times n}$ and any eigenvalue λ of A.*

Remark. The Frobenius norm $\|A\|_{\mathcal{F}}$ of a matrix $A \in \mathbb{R}^{m\times n}$ is given by

$$\|A\|_{\mathcal{F}} = \left(\sum_{i=1}^{m}\sum_{i=1}^{n} a_{ij}^2\right)^{\frac{1}{2}} = \operatorname{tr} AA^T.$$

It is not an operator norm.

Theorem 3.3.4. *Let $\|\cdot\|$ be an operator norm no $\mathbb{R}^{n\times n}$, induced by $\|\cdot\|$ on $\mathbb{R}^n$. Let $A \in \mathbb{R}^{n\times n}$ be regular and let $x, \bar{x}, b, \bar{b} \in \mathbb{R}^n$, so that*

$$Ax = b, \quad A\tilde{x} = \tilde{b}.$$

We then have

$$\frac{\|x - \tilde{x}\|}{\|x\|} \le \|A\| \left\|A^{-1}\right\| \frac{\|b - \tilde{b}\|}{\|b\|}$$

Definition 3.3.5. The condition number of a regular matrix $A \in \mathbb{R}^{n\times n}$ with respect to the operator norm induced by $\|\cdot\|$ on $\mathbb{R}^n$ is given by

$$\operatorname{cond}_{\|\|}(A) = \|A\| \left\|A^{-1}\right\|.$$

When considering ℓ^p-norms we write cond_p instead of $\operatorname{cond}_{\|\cdot\|_p}$.

4 Elimination algorithms

4.1 Gauß elimination

Algorithm 4.1.1 (Gauß elimination)**.** *Let $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$.*

1. *Set $A^{(1)} = A$ and $b^{(1)} = b$, and set $k = 1$.*
2. *For $A^{(k)}$ assume $a_{ij}^{(k)} = 0$ for $1 \le j \le k-1$ and $i \ge j+1$. Setting $\ell_{ik} = a_{ik}^{(k)}/a_{kk}^{(k)}$ for $i = k+1, \dots, n$ we define the normalized lower triangular matrix $L^{(k)} \in \mathbb{R}^{n\times n}$ by:*

$$A^{(k)} = \begin{bmatrix} a_{11}^{(1)} & \dots & & \dots & a_{1n}^{(1)} \\ & \ddots & & & \vdots \\ & & a_{kk}^{(k)} & \dots & a_{kn}^{(k)} \\ & & \vdots & & \vdots \\ & & a_{nk}^{(k)} & \dots & a_{nn}^{(k)} \end{bmatrix}, \quad L^{(k)} = \begin{bmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -\ell_{k+1,k} & & \\ & & \vdots & \ddots & \\ & & -\ell_{nk} & & 1 \end{bmatrix}$$

We then have for $A^{(k+1)} = L^{(k)}A^{(k)}$, that $a_{ij}^{(k+1)} = 0$ for $1 \le j \le k$ and $i \ge j+1$, i.e.,

$$A^{(k+1)} = \begin{bmatrix} a_{11}^{(1)} & \dots & & & \dots & a_{1n}^{(1)} \\ & \ddots & & & & \vdots \\ & & a_{kk}^{(k)} & \dots & \dots & a_{kn}^{(k)} \\ & & & a_{k+1,k+1}^{(k+1)} & \dots & a_{k+1,n}^{(k+1)} \\ & & & \vdots & & \vdots \\ & & & a_{n,k+1}^{(k+1)} & \dots & a_{nn}^{(k+1)} \end{bmatrix}.$$

Further let $b^{(k+1)} = L^{(k)} b^{(k)}$.

3. *Stop if* $k+1 = n$; *otherwise increase* $k \to k+1$ *and go to step (2).*

Theorem 4.1.2. *If* $A \in \mathbb{R}^{n\times n}$ *is regular, then the Gauß elimination algorithm is executable if and only if* A *admits an* LU*-decomposition. The algorithm then yields the normalized* LU*-decomposition of* A *with* $U = A^{(n)}$ *and* $L = \left(L^{(n-1)} \dots L^{(1)}\right)^{-1}$.

The modified right hand side $y = b^{(n)}$ *is given by* $y = L^{-1}b$ *and the solution* x *of the linear system of equations* $Ax = b$ *can be computed by solving* $Ux = y$.

4.2 Pivoting

Algorithm 4.2.1. *Gauß elimination with partial pivoting Let* $A \in \mathbb{R}^{n\times n}$ *be a regular Matrix und* $b \in \mathbb{R}^n$. *We compute the decomposition* $PA = LU$ *and* $y = U^{-1}b$ *by*

for *k=1:n-1* **do**
  find $p \in \{k, \dots, n\}$ so that $\left|a_{pk}^{(k)}\right| = \max_{i=k,\dots,n} \left|a_{ik}^{(k)}\right|$;
  exchange rows p and k in $\left[A^{(k)} \mid b^{(k)}\right]$ to obtain $\left[\tilde{A}^{(k)} \mid \tilde{b}^{(k)}\right]$;
  for *i=k+1:n* **do**
    $\ell_{ik} = \tilde{a}_{ik}^{(k)} / \tilde{a}_{kk}^{(k)}; \quad b_i^{(k+1)} = \tilde{b}_i^{(k)} - \ell_{ik}\tilde{b}_k^{(k)}$
    for *j=k+1:n* **do**
      $a_{ij}^{(k+1)} = \tilde{a}_{ij}^{(k)} - \ell_{ik}\tilde{a}_{kj}^{(k)}$
    end
  end
end

Remark. Exchange of rows can also be expressed by a permutation matrix, i.e., $\tilde{A}^{(k)} = P^{(k)} A^{(k)}$, with $P^{(k)}$ is the matrix that is obtained by exchanging rows p and k in the identity matrix.

Remark. There is also total pivoting where also columns are exchanged, but this is usually not done because it is very slow.

Theorem 4.2.2. *For $A \in R^{n\times n}$ regular, the Gauß-elimination algorith mwith partial pivoting can be executed. It yields the LU-decomposition $PA = LU$ with $|l_{ij}| \leq 1$ for all $1 \leq j \leq n$ and the modified rhs $b^{(n)} = L^{-1}Pb$. Here $P = P^{(n-1)}P^{(n-2)} \dots P^{(1)}$ where $P^{(k)}$ is a permutation matrix to permute the rows in the k-th step.*

Proof.

- Assume the alg. can not be executed at the k-th step. Then the matrix must have $a_{jk} = 0$for $j \geq k$. Such a matrix can not be regular. However up to that step only regular lower triangular and regular permutation matrices were applied. This is a contradiction.
- The statement about $|l_{ij}| \leq 1$follows immediately by $l_{ik} = a_{ik}^{(k)}/a_{pk}^{(k)}$ but $|a_{pk}^{(k)}| \geq |a_{ik}^{(k)}|$ due to pivoting.
- Noting that $(P^{(k)})^{-1} = P^{(k)}$, we compute

  $$
  \begin{aligned}
  A^{(1)} &= A \\
  A^{(2)} &= L^{(1)}P^{(1)}A^{(1)} = L^{(1)}P^{(1)}A \\
  A^{(3)} &= L^{(2)}P^{(2)}A^{(1)} = L^{(2)}P^{(2)}L^{(1)}P^{(1)}A = L^{(2)}\left(P^{(2)}L^{(1)}P^{(2)}\right)P^{(2)}P^{(1)}A \\
  A^{(4)} &= L^{(3)}\left(P^{(3)}L^{(2)}P^{(3)}\right)\left(P^{(3)}P^{(2)}L^{(1)}P^{(2)}P^{(3)}\right)\left(P^{(3)}P^{(2)}P^{(1)}A\right) \\
  &\dots
  \end{aligned}
  $$

  Setting $\hat{L}^{(k)} = P^{(n-1)}P^{(n-2)} \dots P^{(k+1)}L^{(k)}P^{(k+1)} \dots P^{(n-2)}P^{(n-1)}$ we get $A^{(n)} = \hat{L}^{(n-1)} \dots \hat{L}^{(1)}PA$.
  Since $\hat{L}^{(k)}$ has the same structure as $L^{(k)}$ we get the desired decomposition. The statement about $b^{(n)}$ is then also clear.

□

Remark. We still need $\mathcal{O}(n^3)$ operations for this algorithm.

5 Least squares problems

5.1 Gaußian normal equation

We look at overdetermined problems, i.e. for $A \in \mathbb{R}^{m\times n}$ $m \geq n, b \in \mathbb{R}^m$ find $x \in \mathbb{R}^n$ s.t. $Ax \approx b$ in a good way.

Example. Linear regression for measurements (t_i, y_i).

Definition 5.1.1. Given $A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$, we define the least squares problem of

$$\text{minimize } x \mapsto \|Ax - b\|_2^2$$

The vector $r = Ax - b \in \mathbb{R}^m$ is called residual.

Theorem 5.1.2. *Solutions of the least squares problem are given by the solutions of the Gaußian normal equation*

$$A^T Ax = A^T b$$

In particular, a solution exists. If $z \in \mathbb{R}^n$ is a further solution, then $Ax = Az$ and the residuals agree.

Proof. We note that $\mathbb{R}^m = Im(A) \oplus ker\left(A^T\right)$ and $Im(A) \perp ker\left(A^T\right)$ Given $b \in \mathbb{R}^m$ we thus have uniquely determinend, orthogonal vectors $y \in Im(A), r \in ker\left(A^T\right)$ with $y \cdot r = 0, b = y + r$. Further, we have $x \in \mathbb{R}^n$: $y = Ax$. This yields

$$A^T b = A^T y + A^T r = A^T y = A^T Ax.$$

Now let $z \in \mathbb{R}^n$ and compute

$$\begin{aligned}
\|b - Az\|_2^2 &= \|b - Ax + A(x - z)\|_2^2 \\
&= \|b - Ax\|_2^2 + 2\underbrace{(b - Ax)}_{=r} \cdot (A(x - z)) + \|A(x - z)\|_2^2 \\
&= \|b - Ax\|_2^2 + 2\underbrace{A^T r}_{=0} \cdot (x - z) + \|A(x - z)\|_2^2 \\
&= \|b - Ax\|_2^2 + \|A(x - z)\|_2^2 \geq \|b - Ax\|_2^2
\end{aligned}$$

Thus x is a minimizer and a solution to the least squares problem.
If z is also a minimizer , then above we need an equality and thus $\|A(x - z)\|_2^2 = 0 \implies A(x - z) = 0$, thus $x - z \in ker(A) \overset{\text{Ex.}}{=} ker\left(A^T A\right)$. In particular, if z is a minimizer, it also solves the Gauß normal equation. We also get $Ax = Az$. □

Remark. We have used that $ker\left(A^T A\right) = ker(A)$. This is an exercise.

Lemma 5.1.3. *A^TA is symmetric and pos. semidefinite. It is pos. def. iff $ker\,(A) = \{0\}$, i.e. A is injective.*

Proof. Exercise. □

Remark. $\mathrm{cond}_2\left(A^TA\right) = \|A^TA\|_2\|\left(A^TA\right)^{-1}\|_2 = \frac{\lambda_{\max}\left(A^TA\right)}{\lambda_{\min}\left(A^TA\right)} = \mathrm{cond}_2^2\,(A)$ if $A \in R^{n\times n}$. So conditioning of A^TA is usually not good. We thus typically do not use the Gaußian normal equation to solve least squares problems.

5.2 Householder trasformations

Note that for Q orthogonal, i.e. $Q \in O\,(n)$ we have

$$\|A\,(Ax-b)\,\|_2^2 = \|Ax-b\|_2^2$$

Remark. For $Q \in O\,(n)$ we have $\mathrm{cond}_2 = 1$.

Definition 5.2.1. Given $v \in \mathbb{R}^l$, $\|v\|_2 = 1$, the matrix $P_v = \mathbb{1} - 2vv^T$ is called Householder trasformation.

Lemma 5.2.2. *Every Householder transfomration $P_v = \mathbb{1} - 2vv^T$ is symmetric and orthogonal. We have $P_vv = -v$ and $P_vw = w$ for $v \cdot w = 0$*

Lemma 5.2.3. *Let $x \in \mathbb{R}^\ell\backslash\{0\}$ and $x \notin \mathrm{span}\,\{e_1\}$. With $\sigma = \mathrm{sign}\,(x_1)$ if $x_1 \neq 0$ and $\sigma = 1$ otherwise, set*

$$v = \frac{x + \sigma\|x\|_2 e_1}{\|x + \sigma\|x\,\|_2 e_1\|_2}$$

We then have

$$P_vx = \left(I_\ell - 2vv^T\right)x = -\sigma\|x\|e_1.$$

5.3 QR-Decomposition

Theorem 5.3.1. *Let $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ and $\mathrm{rank}\,A = n$. Then there exist $Q \in O(m)$ and a generalized upper triangular matrix $R \in \mathbb{R}^{m\times n}$, i.e, we have $r_{ij} = 0$ for $i > j$, such that*

$$A = QR = Q \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ & r_{22} & \dots & r_{2n} \\ & & \ddots & \vdots \\ & & & r_{nn} \\ & & & \end{bmatrix}$$

We further have $|r_{ii}| > 0$ *for all* $1 \leq i \leq n$. *This factorization is called* QR*-decomposition.*

Theorem 5.3.2. *Let* $A \in \mathbb{R}^{m \times n}$ *with* $m \geq n$ *and* $\operatorname{rank} A = n$. *Using the* QR*-decomposition* $A = QR$ *and*

$$Q^T b = \begin{bmatrix} c \\ d \end{bmatrix}, \quad Q^T A = R = \begin{bmatrix} \hat{R} \\ 0 \end{bmatrix}$$

with $c \in \mathbb{R}^n, d \in \mathbb{R}^{m-n}$ *and an upper triangular matrix* $\hat{R} \in \mathbb{R}^{n \times n}$ *the solution of the least squares problem defined by* A *and* b *is given by the solution* x *of* $\hat{R}x = c$.

Remark. For $A \in \mathbb{R}^{n \times n}$ we have $\operatorname{cond}_2(R) = \operatorname{cond}_2(A)$. The QR-decomposition thus yields a stable algorithm to compute the solution of a least squares problem.

6 Iterative methods for linear systems

Especially if a matrix is sparse, i.e., a significant number of its entries are zero, it is often advantageous to use an iterative method, where in each iteration the matrix is simply applied to a vector, to approximate the solution of a linear system. Such methods are often based on Banach fixed point theorem. We recall

Theorem 6.0.1. *if* $\Phi \colon \mathbb{R}^n \to \mathbb{R}^n$ *is a contraction, i.e., there exists* $q \in [0, 1)$ *so that* $||\Phi(x) - \Phi(y)|| \leq q||x - y||$ *for some norm* $|| \cdot ||$, *then* Φ *admits a unique fixed point* $x^* \in \mathbb{R}^n$, *i.e.,* $\Phi(x^*) = x^*$. *For any starting value* $x^0 \in \mathbb{R}^n$, *the fixed point iteration* $x^{k+1} = \Phi\left(x^k\right)$ $(k = 0, 1, 2, \dots)$, *defines a sequence converging to* x^* *with the property that*

$$\left\|x^k - x^*\right\| \leq \frac{q^k}{1 - q} \left\|x^1 - x^0\right\|$$

6.1 Linear iteration methods

Assume that the map $\Phi\colon \mathbb{R}^n \to \mathbb{R}^n$ is given by $\Phi(x) = Mx + s$ with a matrix $M \in \mathbb{R}^{n\times n}$. We then see that Φ is a contraction if a norm exists such that the corresponding operator norm satisfies $||M|| < 1$.

Theorem 6.1.1. *For $M \in \mathbb{R}^{n\times n}$ we have*

$$\begin{aligned}\rho(M) &= \max\{|\lambda| : \lambda \in \mathbb{C} \text{ is a complex eigenvalue of } M\}\\ &= \inf\{\|M\|_{op} : \|\cdot\|_{op} \text{ is an induced norm on } \mathbb{C}^{n\times n}\}.\end{aligned}$$

Corollary 6.1.2. *If $\rho(M) < 1$ then $\Phi\colon x \mapsto Mx + s$ defines a contraction.*

Example. The Richardson method to approximate the solution of a linear system of equations $Ax = b$ is, given $\omega > 0$, defined by $M = I_n - \omega A$ and $c = \omega b$, i.e.,

$$x^{k+1} = Mx^k + c = x^k - \omega\left(Ax^k - b\right).$$

If A is symmetric and positive definite, then all Eigenvalues of A are positive, and for ω sufficiently small we have $\rho\left(I_n - \omega A\right) < 1$. If $x^{k+1} = x^k$, then x^k is a solution of $Ax = b$.

6.2 Jacobi and Gauß-Seidel method

We consider the additive split $A = L + U + D$, where L is strictly lower triangular, U is strictly upper triangular, and D is diagonal. We consider methods $x^{k+2} = Mx^k + c$.

Definition 6.2.1. The Jacobi and Gauß-Seidel method are defined by

$$\begin{aligned}M^J &= -D^{-1}(A - D), & c^J &= D^{-1}b\\ M^{GS} &= -(L + D)^{-1}U, & c^{GS} &= (L + D)^{-1}b.\end{aligned}$$

Definition 6.2.2. A matrix $A \in \mathbb{R}^{n\times n}$ is called diagonally dominant if, for $i = 1, 2, \ldots, n$ we have

$$\sum_{j=1,\ldots,j\neq i} |a_{ij}| \leq |a_{ii}|$$

and if this inequality is strict for some $i_0 \in \{1, 2, \ldots, n\}$. If the inequality is strict for all $i = 1, 2, \ldots, n$, then A is called strictly diagonally dominant.

Remark. We can quickly see that the Jacobi and Gauß-Seidel method define converging sequences x^k if A is strictly diagonally dominant. Unfortunately, this is usually too much to ask in practice.

Definition 6.2.3. A matrix $A \in \mathbb{R}^{n\times n}$ is called reducible, if disjoint, non-empty index sets $I, J \subset \{1, 2, \ldots, n\}$ exist, so that $I \cup J = \{1, 2, \ldots, n\}$ and $a_{ij} = 0$ for all pairs $(i, j) \in I \times J$. Otherwise A is called irreducible.

Theorem 6.2.4. *If A is irreducible and diagonally dominant, then the Jacobi and Gauß-Seidel methods are executable and convergent.*

7 Polynomial interpolation

We consider the problem of finding a polynomial of degree no more than n, i.e, a function in

$$P_n = \left\{ \sum_{i=0}^{n} a_i x^i : a_0, a_1, \ldots, a_n \in \mathbb{R} \right\}$$

interpolating a given set of pairs of nodes and values.

7.1 Lagrange interpolation

Definition 7.1.1. Given nodes $a \leq x_0 < x_1 < \cdots < x_n \leq b$ and values $y_0, y_1, \ldots, y_n$, the Lagrange interpolation problem consists in finding a polynomial $p \in P_n$ so that $p(x_i) = y_i$ for $i = 0, 1, \ldots, n$.

Definition 7.1.2. Given nodes $x_0 < x_1 < \cdots < x_n$, the corresponding Lagrange polynomials $L_0, L_1, \ldots, L_n \in P_n$ are given by

$$L_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j} = \frac{(x - x_1)}{(x_i - x_1)} \cdots \frac{(x - x_{i-1})}{(x_i - x_{i-1})} \frac{(x - x_{i+1})}{(x_i - x_{i+1})} \cdots \frac{(x - x_n)}{(x_i - x_n)}.$$

Remark. We have $L_i(x_j) = \delta_{ij}$ for $0 \leq i, j \leq n$.

Theorem 7.1.3. *The Lagrange interpolation problem is uniquely solved by taking*

$$p = \sum_{i=0}^{n} y_i L_i.$$

The polynomial p is calle (Lagrange) interpolation polynomial.

7.2 Interpolation error

If $y_j = f(x_j)$ with nodes x_j and a given function f, we may ask how far the interpolation polynomial p is from the function f.

Theorem 7.2.1. *Let $f \in C^{n+1}([a,b])$, $a \leq x_1 < x_2, \ldots, < x_n \leq b$ and let $f(x_i) = y_i$ for $i = 0, 1, \ldots, n$. If $p \in P_n$ is the Lagrange interpolation polynomial, then for $x \in [a,b]$ there exists $\xi \in [a,b]$, so that*

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=0}^{n} (x - x_j).$$

Corollary 7.2.2. *The interpolation error satisfies*

$$\|f - p\|_{C^0([a,b])} \leq \frac{\left\|f^{(n+1)}\right\|_{C^0([a,b])}}{(n+1)!} (b-a)^{n+1}.$$

Remark. If the norms of the derivatives of f does not grow too quickly as $n \to \infty$, then the preceding corollary yields uniform convergence of the interpolating polynomial when increasing its order. This estimate, however, may by not optimal as it does not consider the effect of choosing the nodes in an optimal way. When, for example, taking evenly spaced nodes, the interpolation of

$$f \colon [-1,1] \to \mathbb{R}, \quad f(x) = \frac{1}{1 + 25x^2}$$

yields larger and larger oscillation when the order of the polynomial is increased.

7.3 Chebyshev nodes

We may decrease the interpolation error by finding nodes which minimize the values of the nodal polynomial

$$w(x) = \prod_{j=0}^{n} (x - x_j)$$

on $[a, b]$. We consider here the case $[a, b] = [-1, 1]$.

Definition 7.3.1. For $n \in \mathbb{N}_0$, the n-th Chebyshev polynomial on $t \in [-1, 1]$ is given by

$$T_n(t) = \cos(n \arccos t),$$

The roots of a Chebyshev polynomial are called Chebyshev nodes.

Lemma 7.3.2. *1. We have $|T_n(t)| \leq 1$ for all $t \in [-1, 1]$.*

2. With $T_0(t) = 1$ and $T_1(t) = t$, we have

$$T_{n+1}(t) = 2tT_n(t) - T_{n-1}(t)$$

for $t \in [-1, 1]$. In particular, this yields $T_n \in P_n|_{[-1,1]}$ and $n \geq 1$ we have $T_n(t) = 2^{n-1}t^n + q_{n-1}(t)$ with $q_{n-1} \in P_{n-1}|_{[-1,1]}$.

3. For $n \geq 1$ the polynomial T_n has the n roots $t_j = \cos((j + 1/2)\pi/n)$, $j = 0, 1, \ldots, n - 1$, and $n + 1$ extremal points $s_j = \cos(j\pi/n)$, $j = 0, 1, \ldots, n$.

Theorem 7.3.3. *Let $t_0, t_1, \ldots, t_n \in [-1, 1]$ be the roots of the Chebyshev polynomial T_{n+1}. We then have*

$$\min_{x_0,\ldots,x_n \in [-1,1]]} \max_{x \in [-1,1]} \prod_{j=0}^{n} |x - x_j| = \max_{x \in [-1.1]} \prod_{j=0}^{n} |x - t_j| = 2^{-\pi}.$$

8 Discrete Fourier transforms

8.1 Trigonometric interpolation

Definition 8.1.1. For $m \in \mathbb{N}, n = 2m$, Nodes $x_j = \frac{2\pi j}{n}$ and values $y_j \in \mathbb{R}$, $j = 0, \ldots, n-1$ we find $a_l, b_l \in \mathbb{R}, l = 1, \ldots, m-1$ and $a_0, a_m \in \mathbb{R}$ s.t. for

$$T(x) = \frac{a_0}{2} + \sum_{l=1}^{m-1} \left(a_l \cos(lx) + b_l \sin(lx)\right) + \frac{a_m}{2} \cos(mx)$$

we have $T(x_j) = y_j$ for $j = 0, \ldots, n-1$. This is called real trigonometric interpolation problem.

Definition 8.1.2. The complex trigonometric interpolation problem consists in finding $\beta_k \in \mathbb{C}, k = 0, \ldots, n-1$ s.t. for $x_j = \frac{2\pi j}{n}, y_j \in \mathbb{C}, j = 0, \ldots, n-1$ and

$$p(x) = \beta_0 + \beta_1 e^{ix} + \ldots + \beta_{n-1} e^{i(n-1)} = \sum_{k=1}^{n-1} \beta_k e^{ikx}$$

s.t. $p(x_j) = y_j$ for $j = 0, \ldots, n-1$

Theorem 8.1.3. *Fix $n = 2m, y_0, \ldots, y_{n-1} \in \mathbb{R}$. The coefficients $\beta_k, k = 0, \ldots, n-1$ solve the complex trig-interpol problem iff the coefficients $a_l, b_l, l = 1, \ldots, m-1$ given by $a_0 = 2\beta_0, a_l = \beta_l + \beta_{2m-l}, b_l = i(\beta_l - \beta_{2m-l}), a_m = 2\beta_m$ solve the real trig-interpol problem given by $y_1, \ldots, y_{n-1}$*

Proof. Ingedients: $e^{-ilx} = e^{\frac{-i2\pi lj}{n}} = e^{\frac{i2\pi(n-l)j}{n}} = e^{i(n-l)x_j}$
$e^{ix} = \cos(x) + i \sin(x)$.
This implies

$$\begin{aligned}
&\frac{a_0}{2} + \sum_{l=1}^{m-1} \left(a_l \cos(lx_j) + b_l \sin(lx_j)\right) + \frac{a_m}{2} \cos(mx_j) \\
&= \underbrace{\frac{a_0}{2}}_{\beta_0} + \sum_{l=1}^{m-1} \underbrace{\frac{a_l - ib_l}{2}}_{\beta_l} e^{ilx_j} + \sum_{l=1}^{m-1} \underbrace{\frac{a_l + ib_l}{2}}_{\beta_{n-l}} e^{i(n-l)x_j} + \underbrace{\frac{a_m}{2}}_{\beta_{n-1}} \frac{e^{imx_j} + e^{-imx_j}}{2}
\end{aligned}$$

□

8.2 Fourier basis

If we write $p(x_j) = y_j$ in vectorial form, we obtain

$$y = \begin{pmatrix} y_0 \\ \vdots \\ y_{n-1} \end{pmatrix} = \sum_{k=0}^{n-1} \beta_k \begin{pmatrix} e^{ikx_0} \\ \vdots \\ e^{ikx_{n-1}} \end{pmatrix} = \sum_{k=0}^{n-1} \beta_k \omega^k \text{ with } \omega^k = \begin{pmatrix} e^{ikx_0} \\ \vdots \\ e^{ikx_{n-1}} \end{pmatrix}$$

For this to have a solution, we need $(\omega^0, \ldots, \omega^{n-1})$ to be a basis of $\mathbb{C}^n$.

Definition 8.2.1. For $n \in \mathbb{N}$ let $\omega_n = e^{\frac{i2\pi}{n}}$ be the n-th root of unity and for $k = 0, \ldots, n-1$ let

$$\omega^k = \begin{pmatrix} \omega_n^{0k} \\ \omega_n^{1k} \\ \vdots \\ \omega_n^{(n-1)k} \end{pmatrix}$$

Then $(\omega^0, \ldots, \omega^{n-1})$ are called Fourier basis of $\mathbb{C}^n$.

Proposition 8.2.2. *The Fourier basis is an orthogonal basis of $\mathbb{C}^n$, i.e. $\omega^k \cdot \omega^l = n\delta_{kl}$.*

Proof. Exercise. □

Lemma 8.2.3. *The transformation from Fourier to canonical basis is performed by the matrix*

$$T_n = \begin{pmatrix} \omega^0 & \ldots & \omega^{n-1} \end{pmatrix} \in \mathbb{C}^{n \times n}$$

with inverse $T_n^{-1} = \frac{1}{n}\overline{T_n}^T$, i.e. for $y = \sum_{i=0}^{n-1} y_i e_i \in \mathbb{C}^n$ we have

$$y = \sum_{k=0}^{n-1} \beta_k \omega^k \text{ with } \beta = \frac{1}{n}\overline{T_n}^T y$$

Proof. Clear, if noting that $\frac{1}{\sqrt{n}} T_n \in U(n)$. □

Remark. $y \mapsto \beta = \frac{1}{n}\overline{T_n}^T y$ is called discrete Fourier transform,
$\beta \mapsto y = T_n \beta$ is called inverse DFT.
As the transformation is unitary (modular constant) it is stable.

8.3 Fast Fourier transform

Example. Lets look at the inverse Fourier transform $y = T_8\beta$, noting that $\omega_8^l = \left(e^{\frac{i2\pi}{8}}\right)^l = e^{\frac{li2\pi}{8}} = e^{\frac{(l \bmod 8)2\pi}{8}} = \omega_8^{l \bmod 8}$ We have

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y6 \\ y7 \end{pmatrix} = \begin{pmatrix} \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 \\ \omega_8^0 & \omega_8^1 & \omega_8^2 & \omega_8^3 & \omega_8^4 & \omega_8^5 & \omega_8^6 & \omega_8^7 \\ \vdots & \vdots & \omega_8^4 & \omega_8^6 & \omega_8^0 & \omega_8^2 & \vdots & \vdots \\ & & \omega_8^6 & \omega_8^1 & \vdots & \omega_8^7 & & \\ & & \omega_8^0 & \omega_8^4 & & \omega_8^4 & & \\ & & & \omega_8^7 & & \omega_8^1 & & \\ & & & \omega_8^2 & & \omega_8^6 & & \\ \omega_8^0 & \omega_8^7 & \omega_8^6 & \omega_8^5 & \omega_8^4 & \omega_8^3 & \omega_8^2 & \omega_8^1 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \vdots \\ \\ \\ \\ \\ \\ \beta_7 \end{pmatrix}$$

Sorting by odd and even indices yiels

$$\left(\begin{array}{c} y_0 \\ y_1 \\ y_2 \\ y_3 \\ \hline y_4 \\ y_5 \\ y_6 \\ y_7 \end{array}\right) = \left(\begin{array}{cccc|cccc} \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 \\ \omega_8^0 & \omega_8^2 & \omega_8^4 & \omega_8^6 & \omega_8^1 & \omega_8^3 & \omega_8^5 & \omega_8^7 \\ \omega_8^0 & \omega_8^4 & \omega_8^0 & \omega_8^4 & \omega_8^2 & \omega_8^6 & \omega_8^2 & \omega_8^6 \\ \omega_8^0 & \omega_8^6 & \omega_8^4 & \omega_8^2 & \omega_8^3 & \omega_8^1 & \omega_7^0 & \omega_8^5 \\ \hline \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^0 & \omega_8^4 & \omega_8^4 & \omega_8^4 & \omega_8^4 \\ \omega_8^0 & \omega_8^2 & \omega_8^4 & \omega_8^6 & \omega_8^5 & \omega_8^7 & \omega_8^1 & \omega_8^3 \\ \omega_8^0 & \omega_8^4 & \omega_8^0 & \omega_8^4 & \omega_8^6 & \omega_8^2 & \omega_8^6 & \omega_8^2 \\ \omega_8^0 & \omega_8^6 & \omega_8^4 & \omega_8^2 & \omega_8^7 & \omega_8^5 & \omega_8^3 & \omega_8^1 \end{array}\right) \begin{pmatrix} \beta_0 \\ \beta_2 \\ \beta_4 \\ \beta_6 \\ \beta_1 \\ \beta_3 \\ \beta_5 \\ \beta_7 \end{pmatrix} = \begin{pmatrix} T_4 & D_4T_4 \\ T_4 & -D_4T_4 \end{pmatrix} \begin{pmatrix} \beta_0 \\ \beta_2 \\ \beta_4 \\ \beta_6 \\ \beta_1 \\ \beta_3 \\ \beta_5 \\ \beta_7 \end{pmatrix}$$

with

$$D_4 = \begin{pmatrix} \omega_8^0 & & & \\ & \omega_8^1 & & \\ & & \omega_8^2 & \\ & & & \omega_8^3 \end{pmatrix}$$

Thus we have

$$\begin{pmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{pmatrix} = T_4 \begin{pmatrix} \beta_0 \\ \beta_2 \\ \beta_4 \\ \beta_6 \end{pmatrix} + D_4T_4 \begin{pmatrix} \beta_1 \\ \beta_3 \\ \beta_5 \\ \beta_7 \end{pmatrix}$$

$$\begin{pmatrix} y_4 \\ y_5 \\ y_6 \\ y_7 \end{pmatrix} = T_4 \begin{pmatrix} \beta_0 \\ \beta_2 \\ \beta_4 \\ \beta_6 \end{pmatrix} - D_4 T_4 \begin{pmatrix} \beta_1 \\ \beta_3 \\ \beta_5 \\ \beta_7 \end{pmatrix}$$

Theorem 8.3.1. *For $\beta \in \mathbb{C}^{2m}$ let $D_m \in \mathbb{C}^{m \times m}$ the diagonal matrix with entries $(D_m)_{ll} = \omega_{2m}^l, l = 0, \ldots, m-1$.*
Then $y = T_{2m}\beta$ is given by $\begin{pmatrix} y^1 \\ y^2 \end{pmatrix}$ with $y^1, y^2 \in \mathbb{C}^m$ given by

$$y^1 = T_m \beta^{even} + D_m T_m \beta^{odd}$$
$$y^2 = T_m \beta^{even} - D_m T_m \beta^{odd}$$

where

$$\beta^{even} = \begin{pmatrix} \beta_0 \\ \beta_2 \\ \vdots \\ \beta_{2m-2} \end{pmatrix}, \beta^{odd} = \begin{pmatrix} \beta_1 \\ \beta_3 \\ \vdots \\ \beta_{2m-1} \end{pmatrix}$$

Proof. Analogous to example. □

We have taken a problem of size n with complexity $\mathcal{A}(n)$ and split it into 2 problems of size $\frac{n}{2}$ with complexity $\mathcal{A}\left(\frac{n}{2}\right)$ + putting together with complexity $\frac{3n}{2}$. If $n = 2^l$ we can iterate

$$\begin{aligned} \mathcal{A}(n) &\to 2\mathcal{A}\left(\frac{n}{2}\right) + \frac{3n}{2} \to 2\left(\mathcal{A}\left(\frac{n}{4}\right) + \frac{3n}{4}\right) + \frac{3n}{2} \\ &\to \ldots \to 2^l \mathcal{A}(1) + l\frac{3n}{2} \end{aligned}$$

So we get from $\mathcal{O}(n^2)$ to $\mathcal{O}(n\mathcal{O}(1) + \log_2(n)\, n) = \mathcal{O}(n \log_2(n))$

9 Numerical Quadrature

9.1 Quadrature rules

We want to approximate $\int_a^b f(x)\,\mathrm{d}x =: I(f)$.

Definition 9.1.1. A quadrature formula on $[a,b]$ is a linear map

$$Q : C([a,b]) \to \mathbb{R},$$
$$Q(f) = \sum_{i=0}^{n} \omega_i f(x_i)$$

with nodes $x_i, i = 0, \ldots, n$ and quadrature weights $\omega_i, i = 0, \ldots, n$. The number $\|Q\| = \frac{1}{b-a} \sum_{i=0}^{n} |\omega_i|$ is called its stability indicator.

Example. For $a = x_0 < x_1 < \ldots < x_n = b$ we can approximate the Riemann integral by

$$\int_a^b f(x)\,\mathrm{d}x \approx \sum_{i=0}^{n-1} (x_{i+1} - x_i) f(x_i)$$

Remark. We always have

$$|Q(f)| \leq \|Q\| (b-a) \|f\|_{C^0([a,b])}$$

Definition 9.1.2. A quadrature formula is called exact of degree r, if $Q(p) = I(p)$ for all $p \in \mathcal{P}_r$

Theorem 9.1.3. *Assume Q is exact of degree $r \geq 0$. We then have*

$$\sum_{i=0}^{n} \omega_i = b - a$$

and for $f \in C^0([a,b])$ we have

$$|I(f) - Q(f)| \leq (1 + \|Q\|)(b-a) \min_{p \in \mathcal{P}_r} \|f - p\|_{C^0([a,b])}$$

If also $\omega_i \geq 0$ for $i = 0, \ldots, n$ then $|Q| = 1$

Proof. We have $\sum \omega_i = Q(1) \overset{r \geq 0}{=} I(1) = b - a$. Take now $f \in C^0, p \in \mathcal{P}_r$ with $I(p) = Q(p)$. We have

$$\begin{aligned} |I(f) - Q(f)| &= |I(f-p) - Q(f-p)| \leq |I(f-p)| + |Q(f-p)| \\ &\leq (b-a)\|f-p\|_{C^0} + (\sum |\omega_i|)\|f-p\|_{C^0} \\ &\leq (1 + \|Q\|)(b-a)\|f-p\|_{C^0} \end{aligned}$$

taking min over $p \in \mathcal{P}_r$ yields the result. □

Remark. 1. Result from polynomial interpolation yields

$$|I(f) - Q(f)| \leq (1 + \|Q\|)(b-a) \frac{\|f^{(r+1)}\|_{C^0}}{(r+1)!} (b-a)^{r+1}$$

2. Symmetry may further improve things: Q exact of degree $2q, q \in \mathbb{N}_0$, weights & nodes symetric wrt. $\frac{b+a}{2}$ yields exact of degree $2q+1$.

3. Transformation to other invervals by affine map.

9.2 Newton-Cotes formulas

Consider nodes $x_i, i = 0, \ldots, n$ and for $f \in C^0([a,b])$ the Lagrange interp. polynomial $p = \sum_{i=0}^{n} f(x_i) L_i$, with L_i the Lagrange polynomial for the point x_i. We now approximate the integral of f by the integral of p, which yields

$$\int_a^b p(x)\,\mathrm{d}x = \sum_{i=0}^{n} f(x_i) \underbrace{\int_a^b L_i(x)\,\mathrm{d}x}_{=:\omega_i} = \sum_{i=0}^{n} f(x_i)\,\omega_i = Q(f)$$

These are called Newton-Cotes formulas.

Theorem 9.2.1. *Given nodes $x_0, \ldots, x_n$, weights $\omega_i = \int_a^b L_i(x)\,dx, i = 0, \ldots, n$ the the resultion Newton-Cotes formula is exact of degree n.*

Example. 1. $n = 0, x_0 = \frac{b+a}{2}$: Midpointrule. $\to \omega_0 = b - a$. Is exact of degree $r = 1$ (symmetry).

2. $n = 1, x_0 = a, x_1 = b$: Trapezoidal rule. $\to \omega_0 = \omega_1 = \frac{b-a}{2}$. Exact of deg $r = 1$.

3. $n = 2, x_0 = a, x_1 = \frac{b+a}{2}, x_2 = b$: Simpsons rule. $\to \omega_0 = \frac{b-a}{6} = \omega_2, \omega_1 = \frac{2(b-a)}{3}$. Is exact of deg $r = 3$ (symmetry).

4. $n \geq 7$ is not good as weights can become negative.

9.3 Summed quadrature rule

Definition 9.3.1. Let $a = a_0 < a_1 < \ldots < a_N = b$ a unif. partition of $[a, b]$ and $Q_l : C^0([a_{l-1}, a_l]) \to \mathbb{R}$ be a quad. rule on $[a_{l-1}, a_l]$, $l = 1, \ldots, N$. Then

$$Q^N(f) = \sum_{l=1}^{N} Q_l\left(f|_{[a_{l-1},a_l]}\right)$$

is a summed quadrature rule.

Example. With trapezoidal rule on all $[a_{l-1}, a_l]$ we get

$$\begin{aligned} Q^N(f) &= \sum_{l=1}^{N} \frac{a_l - a_{l-1}}{2} \left(f(a_l) + f(a_{l-1})\right) \\ &= \frac{b-a}{2n} \left(f(a_0) + 2f(a_1) + \ldots + 2f(a_{N-1}) + f(a_N)\right) \end{aligned}$$

Theorem 9.3.2. *If all quadr. formulas in each partition are exact of deg $r \geq 0$, then we have*

$$|I(f) - Q^N(f)| \leq (b-a)^{r+2} \left(1 + \max_{l=1,\ldots,N} \|Q_l\|\right) \frac{N^{-(r+1)}}{(r+1)!} \|f^{(r+1)}\|_{C^0([a,b])}$$

Proof. Calculation. □

Definition 9.3.3. Q^N is called convergent of order $s \geq 0$ if

$$|Q^N(f) - I(f)| = \mathcal{O}(h^s)$$

for $f \in C^s([a, b])$, where $h = \frac{b-a}{n}$.

Example. Summed trapezoidal rule is convegent of order $s = 2$.

9.4 Gauß-quadrature

Lemma 9.4.1. *A quadrature formula with $n + 1$ nodes and wights has at most exactness of degree $2n + 1$.*

Proof. Consider $Q(f) = \sum_{i=0}^{n} \omega_i f(x_i)$ and let
$p(x) = \prod_{i=0}^{n} (x - x_i)^2 \in \mathcal{P}_{2n+2}$ so that $Q(p) = 0$, but $\int_a^b p(x)\,\mathrm{d}x \neq 0$ as $p(x) \geq 0$ and there exist points where $p(x) > 0$. □

Lemma 9.4.2. *A quadrature formula with $n+1$ nodes and weights $(x_i, \omega_i), i = 0, \ldots, n$ is exact of degree n iff $\omega_i = \int_a^b L_i(x)\,dx$.*
If it is exact of degree $2n$, then $\omega_i > 0, i = 0, \ldots, n$.

Proof. Exercise. □

For Gauß-quadrature, we cosntruct $n+1$ nodes, s.t. the maximal degree of exactness $r = 2n+1$ is attained. More generally, we consider integrals of the form

$$I_\omega(f) = \int_a^b f(x)\,\omega(x)\,\mathrm{d}x$$

with $\omega \geq 0, \omega \in C^0([a,b])$ weight function that defines a scalar product

$$(f,g)_\omega = \int_a^b f(x)\,g(x)\,\omega(x)\,\mathrm{d}x$$

on $C^0([a,b])$.

Theorem 9.4.3. *There exist orthogonal polynomials $(\pi_j)_{j=0}^n$ s.t. $\pi_j \in \mathcal{P}_j$ and $(\pi_j, \pi_i)_\omega = \delta_{jk}$ for $0 \leq j,k \leq n$. In particular we have*

$$(\pi_j, p)_\omega = 0 \ \forall p \in \mathcal{P}_{j-1}$$

and the polynomials are a basis of $\mathcal{P}_n$.

Proof. Exercise. (Gram-Schmidt) □

Lemma 9.4.4. *The roots of any ortho pol $\pi_j, 0 \leq j \leq n$ are simple, real and contained in (a,b).*

Proof. Assume the statement was false for a $j \in \{0, \ldots, n\}$. If π_j has a root $z \in \mathbb{R} \setminus (a,b)$. Then $p(x) = \frac{\pi(x)}{x-z} \in \mathcal{P}_{j-1}$ and we get

$$0 = (\pi_j, p) = \int_a^b \frac{\pi_j^2(x)}{x-z}\omega(x)\,\mathrm{d}x$$

which is not possible as $x - z \neq 0$ on (a,b) and $\pi_j \neq 0$.
If z is a multiple root or if $z \in \mathbb{C} \setminus \mathbb{R}$, then $\overline{z}$ is also a root and let
$p(x) := \frac{\pi_j(x)}{(x-z)(x-\overline{z})} = \frac{\pi_j(x)}{|x-z|^2} \in \mathcal{P}_{j-2}$ and we get again a contradiction to $(\pi_j, p)_\omega = 0$. □

Example. 1. $\omega(x) = \frac{1}{\sqrt{1-x^2}}$ on $(-1,1)$ we get the Chebyshev-polynomials as the orthogonal family.

2. $\omega(x) = 1$ on $[-1,1]$

$$P_n(x) = \frac{1}{2^n n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n} \left(x^2-1\right)^n$$

Theorem 9.4.5. *Let π_{n+1} the $n+1$-th orthogonal polynomial w.r.t. the weight function $\omega \in C^0((a,b))$. Using its roots $(x_i)_{i=0,\dots,n}$ and weights*

$$\omega_i = \int_a^b L_i(x)\,\omega(x)\,dx \text{ for } i = 0,\dots,n$$

we obtain an quadrature formula

$$Q_\omega f = \sum_{i=0}^{n} \omega_i f(x_i)$$

with $Q_\omega p = I_\omega p = \int_a^b p(x)\,\omega(x)\,dx$ for all $p \in \mathcal{P}_{2n+1}$

Proof. The formula is well defined (correct number of nodes , all inside (a,b)) and it immediately satisfies

$$Q_\omega r = I_\omega r \ \forall r \in \mathcal{P}_n.$$

Now for $p \in \mathcal{P}_{2n+1}$ we obtain by poly. division $q, r \in \mathcal{P}_n$ with

$$p = q\pi_{n+1} + r$$

Since $(q, \pi_{n+1})_\omega = 0$ we obtain

$$I_\omega p = \underbrace{\int_a^b q(x)\,\pi_{n+1}(x)\,\omega(x)\,\mathrm{d}x}_{=(q,\pi_{n+1})_\omega=0} + \int_a^b r(x)\,\omega(x)\,\mathrm{d}x = I_\omega r$$

and

$$Q_\omega p = \sum_{i=0}^{n} \omega_i \left(q(x_i) \underbrace{\pi_{n+1}(x_i)}_{=0} + r(x_i) \right) = \sum_{i=0}^{n} \omega_i r(x_i) = Q_\omega r \overset{r\in\mathcal{P}_n}{=} I_\omega r$$

□

Example. For $\omega(x) = 1$ on $[-1,1]$ we get $P_0(x) = 1, P_1(x) = x, P_2(x) = \frac{1}{2}(3x^2-1), P_3(x) = \frac{1}{2}(5x^3-3x)$.
$n = 0, x_0 = 0, \omega_0 = 2$
$n = 1, x_0 = -\sqrt{1/3}, x_1 = \sqrt{1/3}, \omega_0 = 1, \omega_1 = 1$
$n = 2, x_0 = -\sqrt{3/5}, x_1 = 0, x_2 = \sqrt{3/5}, \omega_0 = 5/9, \omega_1 = 8/9, \omega_2 = 5/9$

10 Nonlinear problems

10.1 Rootfinding

Consider $U \subseteq \mathbb{R}^n, f : U \to \mathbb{R}^n$ we are looking to find $x^* \in U$ s.t. $f(x^*) = 0$. Typically you cannot find an exact solution, so we need to look for a sequence $(x_k)_{k\in\mathbb{N}_0}$, s.t. $x_k \to x^*$. A method then iteratively generates the sequence from a starting value $x_0 \in U$.

Definition 10.1.1. A numerical method that yields a sequence $(x_k)_{k\in\mathbb{N}_0}$ of approximation for a numerical problem is called

1. globally convergent if the sequence $(x_k)_{k\in\mathbb{N}_0}$ converges to a solution x^* for any starting value $x_0 \in U$
2. locally convergent if for every solution $x^* \in U$ there exists $\varepsilon > 0$ s.t. $x_k \to x^*\ \forall x_0 \in B_\varepsilon(x^*) \cap U$.

Definition 10.1.2. A locally convergent method is called convergent of order $\alpha \geq 1$ if $\exists q \in \mathbb{R}$ s.t. for any solution $x^* \in U$, every starting point $x_0 \in B_\varepsilon(x^*) \cap U$ and the sequence $(x_k)_{k\in\mathbb{N}_0}$ generated by the method we have

$$\limsup_{k\to\infty} \frac{\delta_{k+1}}{\delta_k^\alpha} = q \text{ for } \delta_k = \|x^* - x_k\|.$$

(and, if $\alpha = 1$ we also have $q < 1$)

Remark. $\alpha = 1, q < 1$: linear method
$\alpha = 2$: quadratic method
$\alpha = 1, q = 0$: superlinear
$\alpha = 1, q = 1$: sublinear

Algorithm 10.1.3 (Bisection method)**.** *Let $f \in C^0([a,b]), f(a) f(b) \leq 0$. and let $\varepsilon_{stop} > 0$. Set $a_0, b_0 = a, b$ and $k = 0$.*

1. Set $c_k = \frac{a_k+b_k}{2}$.

2. Set $(a_{k+1}, b_{k+1}) = \begin{cases} (a_k, c_k) & \text{if } f(a_k) f(c_k) \leq 0 \\ (c_k, b_k) & \text{otherwise} \end{cases}$

3. Stop, if $b_{k+1} - a_{k+1} \leq \varepsilon_{\text{stop}}$, otherwise set $k \to k+1$ and go to Step 1.

Theorem 10.1.4. *The bisection method stops after $J \leq 1 + \log_2 \frac{b-a}{\varepsilon_{stop}}$ steps. Taking $x_k = c_k$ as approximating sequence, it is globally convergent of order $\alpha = 1$ with $q = \frac{1}{2}$.*

Remark. 1. A variant is false position rule, where instead of midpoint we take where the affine line through $f(a_k), f(b_k)$ intersects 0.

2. Both work only in 1d.

Algorithm 10.1.5. *Let $f \in C^0([a,b]), f(a) f(b) \leq 0$ and $\varepsilon_{stop} > 0$, set $x_0 = a, x_1 = b, k = 1$.*

1. If $f(x_k) \neq f(x_{k-1})$ set

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k)$$

2. Stop if $|x_{k+1} - x_k| < \varepsilon_{\text{stop}}$, otherwise set $k \to k+1$, go to step 1.

Remark. 1. Can also take $|f(x_{k+1})| < \varepsilon_{\text{stop}}$ as stopping criterion.

2. $\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$ is an approx of $(f'(x_k))^{-1}$

Now using a Taylor approximation of $f \in C^1(U, \mathbb{R}^n)$ around $x \in U$ yields

$$0 = f(x^*) = f(x) + Df(x)(x^* - x) + \varphi(\|x^* - x\|)$$

Ignoring the small (when x^* close to x) term $\varphi(\|x^* - x\|)$ we get

$$x^* \approx x - Df(x)^{-1}(f(x))$$

, an iteration based on this is

$$x_{k+1} = x_k - Df(x_k)^{-1} f(x_k).$$

this is Newton's method.

Algorithm 10.1.6 (Newton)**.** *Let* $f \in C^1(U, \mathbb{R}^n), x_0 \in U, \varepsilon_{stop} > 0, k = 0$

1. If $Df(x_k)$ is regular, set

$$x_{k+1} = x_k - Df(x_k)^{-1} f(x_k)$$

2. Stop, if $\|x_{k+1} - x_k\| < \varepsilon_{\text{stop}}$, otherwise set $k \to k+1$ and go to step 1.

Theorem 10.1.7. *Let* $f \in C^2(U, \mathbb{R}^n)$, $x^* \in U$ *a root of* f, *s.t.* $Df(x^*)$ *is regular. Then there exists* $\varepsilon > 0$ *s.t. for all* $x_0 \in B_\varepsilon(x^*) \cap U$ *the Newton method is executable and it converges. For iterates* $(x_k)_{k \in \mathbb{N}_0}$ *we have*

$$\|x^* - x_{k+1}\| \leq c\|x^* - x_k\|^2$$

with constant $c \geq 0$

Proof. Since $\det(Df(x^*)) \neq 0$, and $x \mapsto \det(Df(x))$ is cont., we have $\tilde{\varepsilon} > 0$ s.t. $\det(Df(x)) \neq 0$ and $\|Df(x)^{-1}\| \leq c_1 \ \forall x \in B_{\tilde{\varepsilon}}(x^*) \subseteq U$.
Now assume $x_k \in B_{\tilde{\varepsilon}}(x^*)$ for $k \geq 0$. The Taylor expansion yields

$$0 = f(x^*) = f(x_k) + Df(x_k)(x^* - x_k) + \varphi(\|x^* - x_k\|)$$

with $\varphi : \mathbb{R} \to \mathbb{R}, \varphi(t) \leq c_2 t^2$ for all $|t| \leq c_3$.
This now yields

$$\|f(x_k) - Df(x_k)(x^* - x_k)\| \leq c_2\|x^* - x_k\|^2$$

Using the iteration, we get

$$x^* - x_{k+1} = Df(x_k)^{-1}(f(x_k) + Df(x_k)(x^* - x_k))$$

so

$$\begin{aligned}\|x^* - x_{k+1}\| &= \|Df(x_k)^{-1}(f(x_k) + Df(x_k)(x^* - x_k))\| \\ &\leq \|Df(x_k)^{-1}\| \|f(x_k) + Df(x_k)(x^* - x_k)\| \\ &\leq c_1 c_2 \|x^* - x_k\|^2\end{aligned}$$

Taking $\varepsilon \leq \min\left\{\frac{1}{c_1 c_2}, \tilde{\varepsilon}, \frac{1}{2}\right\}$, we get as long as $x_k \in B_\varepsilon(x^*)$ that

$$\|x^* - x_{k+1}\| \leq c_1 c_2 \varepsilon \|x^* - x_k\| \leq \varepsilon \leq \tilde{\varepsilon}$$

so $x_{k+1} \in B_{\tilde{\varepsilon}}(x^*)$. So the method is well defined and quadratically convergent if $x_0 \in B_\varepsilon(x^*)$. □

10.2 Gradient flows

Instead of finding roots, we want to find minimizers of $g : U \subseteq \mathbb{R}^n \to \mathbb{R}$. This is a related problem to root finding, as the minimizer will (assuming regularity) be a critical point.

Algorithm 10.2.1 (Gradient descent method)**.** *Let $g \in C^1(U), x_0 \in U, \sigma \in (0,1)$ and $\varepsilon_{stop} > 0$.*

1. Set $d_k = -\nabla g(x_k)$ and find
$$\max\left\{a_k \in \left\{2^{-l} : l \in \mathbb{N}_0\right\} : \underbrace{g(x_k + \alpha_k d_k) \leq g(x_k) - \sigma\alpha_k \|d_k\|^2}_{\text{Armijo-condition}}\right\}.$$
2. Set $x_{k+1} = x_k + \alpha_k d_k$
3. Stop, if $\|\alpha_k d_k\| \leq \varepsilon_{\text{stop}}$ otherwise set $k \to k+1$ and got to Step 1.

Theorem 10.2.2. *Let $g \in C^1(U)$ $x_0 \in U, V \subset\subset U$, convex, s.t.*

$$\tilde{V} = \{x \in U : g(x) \leq g(x_0)\} \subseteq V.$$

Settting $m = \max_{x \in \overline{V}} \|\nabla g(x)\|$ and $W = \{x + s : x \in V, s \in B_m(0)\}$ assume then $g \in C^2(\overline{W})$. Then for the iterates $(x_k)_{k \in \mathbb{N}_0}$ of the gradient method we have

$$\nabla g(x_k) \to 0 \text{ as } k \to \infty \text{ and } \alpha_k > (1-\sigma)\frac{1}{\gamma} \text{ with } \gamma = \sup_{x \in W} \|D^2 g(x)\|$$

Proof. The sequence $(g(x_k))_{k\in\mathbb{N}_0}$ is monotonically decreasing, therefore $(x_k)_{k\in\mathbb{N}_0} \subseteq V$ and $g(x_k) \geq -c_0 = \min_{x\in\overline{V}} g(x)$ for all $k \in \mathbb{N}_0$.
From the Armijo-condition, we have

$$\begin{aligned} g(x_0) &\geq g(x_1) + \sigma\alpha_0 \|\nabla g(x_0)\|^2 \\ &\geq g(x_2) + \sigma\alpha_1 \|\nabla g(x_1)\|^2 + \sigma\alpha_0 \|\nabla g(x_0)\|^2 \\ &\geq \ldots \geq g(x_{l+1}) + \sigma \sum_{k=0}^{l} \alpha_k \|\nabla g(x_k)\|^2 \geq -c_0 + \sigma \sum_{k=0}^{l} \alpha_k \|\nabla g(x_k)\|^2 \end{aligned}$$

The sequence $\sum_{k=0}^{l} \alpha_l \|\nabla g(x_k)\|^2$ is thus bounded and therefore $\alpha_k \|\nabla g(x_k)\|^2 \to 0$.
We need to show that $\alpha_k \geq \delta > 0 \ \forall k \in \mathbb{N}_0$ and some $\delta > 0$. Then we are done.
For any $k \in \mathbb{N}_0$, we have that either $\alpha_k = 1$ or that the Armijo-condition is violated for $2\alpha_k$, which implies

$$2\sigma\alpha_k \|\nabla g(x_k)\|^2 > g(x_k) - g(x_k + 2\alpha_k d_k)$$

The Taylor expansion implies that there exists $\xi \in W$ s.t.

$$g(x_k + 2\alpha_k d_k) = g(x_k) + \nabla g(x_k)(2\alpha_k d_k) + \frac{1}{2}(2\alpha_k)^2 \left(d_k, D^2 g(x_k)\, d_k\right)$$

Using $d_k = -\nabla g(x_k)$ and $(d_k, D^2 g(x_k)\, d_k) \leq \gamma \|d_k\|^2$ we have

$$\begin{aligned} & 2\sigma\alpha_k \|d_k\|^2 > 2\alpha_k \|d_k\|^2 - 2\gamma\alpha_k^2 \|d_k\|^2 \\ \implies & (1-\sigma)\,\alpha_k < \gamma\alpha_k^2 \\ \implies & \alpha_k > (1-\sigma)\,\frac{1}{\gamma} \end{aligned}$$

□

11 The Conjugate gradient method

11.1 Quadratic minimization

Let $A \in \mathbb{R}^{n\times n}$ symmetric, pos. def. Then the solution $x^* \in \mathbb{R}^n$ of $Ax = b$ is the unique minimizer of

$$\phi(x) = \frac{1}{2}\|b - Ax\|_{A^{-1}}^2 = \frac{1}{2}(\underbrace{A^{-1}}_{\text{also pos def, ...}} (b - Ax))\,(b - Ax) \geq 0$$

Taking a starting value $\tilde{x} \in \mathbb{R}^n$ and a search direction $\tilde{d} \in \mathbb{R}^n$ we can find a new value $\tilde{x} + \tilde{\alpha}\tilde{d}$ by minimizing $\tilde{\psi} : t \mapsto \phi\left(\tilde{x} + t\tilde{d}\right)$. We have in our case $\tilde{\psi}(t) = \tilde{\phi}(\tilde{x}) - t(b - A\tilde{x})\tilde{d} + \frac{t^2}{2}\left(A\tilde{d}\right) \cdot \tilde{d}$, so the minimizer is $\tilde{\alpha} = \frac{(b-A\tilde{x})\cdot\tilde{d}}{\left(A\tilde{d}\right)\cdot\tilde{d}}$.
If $\tilde{d} = -\nabla\phi(\tilde{x}) = b - A\tilde{x}$ then we get

$$\tilde{x}^{\text{new}} = \tilde{x} + \tilde{\alpha}\tilde{d} = \tilde{x} + \tilde{\alpha}(b - A\tilde{x})$$

which is a step in the Richardson method.

Remark. One can show that $\|x_k - x^*\|_A \leq \left(\frac{\kappa-1}{\kappa+1}\right)^k \|x_0 - x^*\|_A$ with $\kappa = \operatorname{cond}(A)$.

11.2 Conjugate gradients

Definition 11.2.1. $x, y \in \mathbb{R}^n$ are called A-conjugate, if $(x, Ay) = 0$.

Lemma 11.2.2. *Let $d_0, d_1, \ldots, d_k \in \mathbb{R}^n \setminus \{0\}$ be pairwise A-conjugate, i.e. $d_i \cdot (Ad_j) = 0$ for $i \neq j$. Take $x_0 \in \mathbb{R}^n$ and set x_{j+1} as the minimizer of ϕ from 11.1 in the direction of d_j, i.e.*

$$x_{j+1} = x_j + \alpha_j d_j = x_0 + \sum_{l=0}^{j} \alpha_l d_l$$
$$\text{with } \alpha_j = \frac{d_j \cdot (b - Ax_{j-1})}{d_j \cdot Ad_j} = \frac{d_j \cdot (b - Ax_0)}{d_j \cdot Ad_j}$$

for $j = 1, \ldots, k$. Then x_{j+1} is the minimizer of ϕ in the set

$$x_0 + \text{span}\{d_0, \ldots, d_j\}.$$

Proof. Bunch of linear algebra, basically show that partial derivatives $\frac{\partial}{\partial\alpha_j}\phi\left(x_0 + \sum_{l=0}^{j}\alpha_l d_l\right) = \psi_i'(\alpha_i) = 0$. □

To compute A-conj. search directions consider the residual $r_k = b - Ax_k$

Lemma 11.2.3. *For $x_0 \in \mathbb{R}^n, r_0 = b - Ax_0, d_0 = r_0$, we iteratively set*

$$r_{k+1} = r_k - \alpha_k Ad_k$$
$$d_{k+1} = r_{k+1} - \beta_k d_k$$
$$\text{where } \alpha_k = \frac{d_k \cdot r_k}{d_k \cdot Ad_k}, \beta_k = \frac{d_k \cdot Ar_{k+1}}{d_k \cdot Ad_k}$$

Then $d_0, \ldots, d_k$ are a set of non-zero A-conj vectors, until $r^{k+1} = 0$. The Ksylov space $K_k(A, r_0) = span\left\{r_0, Ar_0, \ldots A^{k-1}r_0\right\}$ satisfies

$$K_k(A, r_0) = span\{d_0, \ldots, d_{k-1}\} = span\{r_0, \ldots r_{k-1}\}$$

and r_k is orthogonal to this space.

Proof. Longer computation. □

This yiels

Algorithm 11.2.4 (CG-Method)**.** *Let $A \in \mathbb{R}^{n\times n}$ symmetric, pos. def., $b \in \mathbb{R}^n, x_0 \in \mathbb{R}^n, \varepsilon_{stop} > 0$. Set $d_0 = r_0 = b - Ax_0, k = 0$.*

1. Set $x_{k+1} = x_k + \alpha_k d_k$
   $r_{k+1} = r_k Ad_k$,
   $d_{k+1} = r_{k+1} - \beta_k d_k$ where
   $\alpha_k = \frac{\|r_k\|^2}{d_k \cdot Ad_k}, \beta_k = \frac{\|r_{k+1}\|^2}{\|r_k\|^2}$
2. Stop if $\|r_{k+1}\| \leq \varepsilon_{\text{stop}}$, otherwise $k \to k+1$, go to 1.

Theorem 11.2.5. *We get*

$$\|x^* - x_k\|_A \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k \|x^* - x_k\|_A$$

where $\kappa = cond(A)$.