1 Introduction

In many applications we have to solve a linear system
$$Au = b$$
where the matrix $A \in \mathbb{R}^{n \times n}$ and the right hand side vector $b \in \mathbb{R}^n$ are given. If $n$ is not too large (say, less than $10^4$) we can use Gaussian elimination. In Matlab this means we can use $u=A\backslash b$ which uses Gaussian elimination with partial pivoting (i.e., row permutations).

Many practical problems lead to much larger linear systems. E.g., if we consider temperatures in a unit cube $[0, 1] \times [0, 1] \times [0, 1] \subset \mathbb{R}^3$. We subdivide each edge into $N = 100$ subintervals of length $h = 1/N$. This leads to a grid of $n = N^3 = 10^6$ grid points in the volume of the cube. If we have some physical law relating the temperatures at the grid points we obtain a linear system of about $10^6$ unknowns.

If we have a “full matrix” (i.e., all entries nonzero) $A \in \mathbb{R}^{n \times n}$ with $n = 10^6$ we have a problem: Just storing the entries of the matrix would take $n^2$ times 8 bytes, i.e.
$$10^{12} \cdot 8 \text{ bytes} = 8000 \text{ Gigabytes}$$
which is not practical. Solving a linear system with Gaussian elimination takes $\frac{2}{3} n^3$ floating point operations. For $n = 10^6$ this would be $\frac{2}{3} 10^{18}$ floating point operations. Currently a very fast processor (like core i9 or Xeon) can perform up to 100 Gigaflips for one core (1 Gigaflop means $10^9$ floating point operations per second). Hence $\frac{2}{3} \cdot 10^{18}$ operations take about
$$\frac{2}{3} \cdot 10^{18} / 10^9 \text{ seconds} = \frac{2}{3} \cdot 10^9 \text{ seconds} \approx 21 \text{ years}$$
of computation time (if we had sufficient memory).

However, for most large linear systems we have a “sparse matrix” (i.e., most entries are zero): Each of the unknowns is only coupled with a small number of other unknowns, e.g., 6 other unknowns. This means that each row of the matrix contains at most 7 nonzeros. Hence the number of nonzeros is only $Cn$ rather than $n^3$. Matlab has a sparse matrix format which uses storage proportional to the number of nonzeros. If the matrix has $Cn$ nonzeros we need $C'n$ storage, and we can compute a matrix-vector product $Av$ with $C'n$ floating point operations (rather than $n^2$ as in the case of a full matrix).

For a large sparse linear system $Au = b$ there are two types of algorithms available:

direct solvers give the exact solution of the linear system in finitely many operations (up to roundoff error):
- Gaussian elimination with row pivoting (select pivot candidate with largest absolute value):
  This gives a decomposition $LU = \tilde{A}$ where $\tilde{A}$ is a row permutation of $A$, $L$ is lower triangular, $U$ is upper triangular.
- for $A$ symmetric positive definite: Cholesky decomposition $A = C^\top C$ where $C$ is upper triangular.

For a sparse matrix $A$ the factors $L, U$ (and $C$ for the Cholesky decomposition) will usually have MANY MORE NONZEROS than the original matrix $A$ (so-called “fill-in”), hence this will not be feasible for large $n$.

It turns out that using the unknowns in a different order (this corresponds to a column permutation of the matrix) can dramatically decrease the number of nonzeros in the factors $L, U$. This is what Matlab does by default if you type $u=A\backslash b$ with a matrix $A$ in sparse format.
iterative solvers start with an initial guess. Each iteration improves the error, and we iterate until the error is below a tolerance. For each iteration we use matrix-vector products like $Av$ or $A^\top v$. Hence typically each operation uses $Cn$ operations.

**Example:** For Richardson iteration we start with an initial guess $u^0$ and compute the residual $r^0 := b - Au^0$. Then we define $u^{k+1} := u^k + \alpha r^k$. Here we choose a fixed value $\alpha > 0$ which is “sufficiently small”. If the matrix $A$ has $Cn$ nonzeros each iteration costs $C'n$ operations. We need to find out how many iterations we need to reach our tolerance.

We will see: Typically the condition number of the matrix increases with $n$, e.g., $\text{cond}(A) \approx cn^\alpha$. This means that the number of iterations $k$ will also grow with $n$, i.e., $k \approx c'n^{\beta}$.

**Claim:** Typically, for problems in $\mathbb{R}^d$ with $d \geq 2$ iterative methods are competitive, and for $d \geq 3$ iterative methods are usually superior to direct methods.

We will consider linear systems $Au = b$ where

- $A \in \mathbb{R}^{n \times n}$ is sparse
- $v^\top Av > 0$ for all nonzero vectors $v \in \mathbb{R}^n$

This is satisfied in many practical applications. In particular “equilibrium problems” in elasticity, fluid dynamics and heat flow problems give linear systems where $A$ has these properties.

A very important special case occurs if we have additionally

- $A$ is symmetric

In this case $A$ is symmetric positive definite, and this will allow us to obtain more efficient algorithms.

As typical applications we will consider boundary value problems for elliptic partial differential equations:

- “diffusion problems” (e.g., equilibrium problems for elastic membrane or heat flow)
- “convection-diffusion problems”

Note that the iterative methods we discuss are useful in many other applications.

In the first case $A$ is symmetric positive definite. In the second case $A$ is no longer symmetric, but still satisfies $v^\top Av > 0$ for nonzero $v$.

We will discuss the following iterative methods:

- Richardson iteration with fixed $\alpha$
- minimal residual method and “General Minimal Residual Method” (GMRES)
- “Conjugate Gradient Method” (CG)
- Preconditioned GMRES and preconditioned CG methods

### 2 Richardson iteration for problems with “coercivity condition”

Assume that $V$ is a vector space with an inner product $(u,v)_V$ and norm $\|u\|_V$. Assume that $V$ with this norm is complete. Such a space is called a Hilbert space. We will consider real Hilbert spaces.

Examples:

- $V = \mathbb{R}^n$ with $(u,v) = \sum_{j=1}^n u_j v_j$
- $V = L^2(\Omega)$ with $(u,v) = \int_\Omega u(x)v(x)dx$
- $V = H^1(\Omega)$ with $(u,v) = \int_\Omega [u(x)v(x) + \nabla u(x) \cdot \nabla v(x)] dx$

For a function $F : V \to V$ (may be nonlinear) and $b \in V$ we want to find $u \in V$ such that $F(u) = b$
We have an initial guess $u_0 \in V$. In order to check the equation we compute the residual $r_0 := b - F(u_0)$. If $r_0 = 0$ we are done, otherwise we should use some correction based on $r_0$.

This motivates the following iterative method, called Richardson iteration: We have an initial guess $u_0$. Pick $\alpha \in \mathbb{R}$ and define \[
    u_{k+1} := u_k + \alpha (b - F(u_k)) \quad \text{for } k = 0, 1, 2, \ldots
\]

If the function $F$ satisfies a “Lipschitz condition” (1) and a “coercivity condition” (2) the problem has a unique solution, and the Richardson iteration converges to this solution for $\alpha > 0$ sufficiently small:

**Theorem 2.1.** Let $V$ be a Hilbert space. Assume that the function $F : V \to V$ satisfies with constants $L, \gamma > 0$ for all $u, v \in V$

\[
    \|F(v) - F(u)\| \leq L\|v - u\| \quad \text{"Lipschitz condition" (1)}
\]

\[
    (F(v) - F(u), v - u) \geq \gamma \|v - u\|^2 \quad \text{"coercivity condition" (2)}
\]

Then the equation $F(u) = b$ with $b \in V$ has a unique solution $u \in V$. The inverse mapping satisfies for $b, c \in V$

\[
    \|F^{-1}(c) - F^{-1}(b)\| \leq \gamma^{-1} \|c - b\|. \quad (3)
\]

**Proof.** Consider the Richardson iteration $u_{k+1} = G(u_k)$ with $G(u) := u + \alpha (b - F(u))$ with $\alpha > 0$. We claim that $G$ is a contraction if $\alpha$ is small: With $e := v - u$ we have

\[
    \|G(v) - G(u)\|^2 = \|e - \alpha (F(v) - F(u))\|^2 = \|e\|^2 - 2\alpha (F(v) - F(u), e) + \alpha^2 \|F(v) - F(u)\|^2
\]

\[
    \leq (1 - 2\alpha\gamma + \alpha^2 L^2) \|e\|^2. \quad (4)
\]

For $g(\alpha) := 1 - 2\alpha\gamma + \alpha^2 L^2$ we have $g(0) = 1$ and $g'(0) = -2\gamma < 0$, so $G$ is a contraction for sufficiently small $\alpha$. By completing the square we get

\[
    g(\alpha) = L^2 \left( \alpha - \frac{\gamma \alpha}{L} \right)^2 + 1 - \frac{\gamma^2}{L^2}
\]

and see that $g(\alpha) < 1$ for $\alpha \in (0, 2\gamma/L^2)$; we can minimize $g(\alpha)$ by choosing $\alpha = \gamma/L$ and obtain

\[
    \|G(v) - G(u)\| \leq (1 - \gamma^2/L^2)^{1/2} \|v - u\|. \quad \square
\]

By the contraction mapping theorem the equation $G(u) = u \iff F(u) = b$ has a unique solution. We obtain (3) from $\gamma \|v - u\|^2 \leq (F(v) - F(u), v - u) \leq \|F(v) - F(u)\| \|v - u\|$.

If our current approximation is $u_k$ we can compute the residual $r_k := b - F(u_k)$ and the norm $\|r_k\|$. From (1), (2) we get

\[
    \|F(u_k) - F(u_*)\| \leq L \|u_k - u_*\|
\]

\[
    \gamma \|u_k - u_*\|^2 \leq (F(u_k) - F(u_*), u_k - u_*) \leq \|F(u_k) - F(u_*)\| \cdot \|u_k - u_*\|
\]

hence we have the lower and upper bounds

\[
    \gamma \|u_k - u_*\| \leq \|r_k\| \leq L \|u_k - u_*\|. \quad (5)
\]

**Algorithm 1: Use a fixed $\alpha$.** If we know some (possibly nonoptimal) constants $L, \gamma$ satisfying (1), (2) : We can use any $\alpha \in (0, 2\gamma/L^2)$ and obtain

\[
    \|u_{k+1} - u_*\|^2 \leq (1 - 2\alpha\gamma + \alpha^2 L^2) \|u_k - u_*\|^2
\]

We get a similar estimate for the residuals: For $r_{k+1} = b - F(u_{k+1})$ we have

\[
    \|r_{k+1}\|^2 = \|r_k + F(u_k) - F(u_k + \alpha r_k)\|^2
\]

\[
    = \|r_k\|^2 - 2\alpha^{-1} \left( F(u_k + \alpha r_k) - F(u_k), \alpha r_k \right) + \|F(u_k + \alpha r_k) - F(u_k)\|^2
\]

\[
    \geq \gamma \alpha^2 \|r_k\|^2
\]

\[
    \|r_{k+1}\|^2 \leq (1 - 2\alpha\gamma + \alpha^2 L^2) \|r_k\|^2
\]

3
The best choice is \( \alpha := \gamma/L^2 \). Then we have with \( \kappa := \frac{L}{\gamma} \) the convergence estimates
\[
\|u_{k+1} - u_*\| \leq (1 - \kappa^{-2})^{1/2} \|u_k - u_*\|
\]
\[
\|r_{k+1}\| \leq (1 - \kappa^{-2})^{1/2} \|r_k\|
\]
Hence we need \( O(\kappa^2) \) iterations to achieve \( \|u_k - u_*\| \leq \) tolerance.

**Algorithm 2:** If we don’t know the constants \( \gamma, L \) we can choose for each step a parameter \( \alpha_k \) by “line search”.

Find \( \alpha_k > 0 \) such that for \( u_{k+1} = u_k + \alpha_k r_k \) the new residual \( r_{k+1} = b - F(u_{k+1}) \) has minimal norm, i.e.,
\[
g_k(\alpha) := \|b - F(u_k + \alpha r_k)\|^2 \text{ becomes minimal for } \alpha = \alpha_k.
\]
We have
\[
g_k(\alpha) = \|b - F(u_k + \alpha r_k)\|^2 \leq (1 - 2\alpha \gamma + \alpha^2 L^2) \|r_k\|^2
\]
If we perform an exact line search then
\[
\|r_{k+1}\|^2 = g_k(\alpha_k) \leq g_k\left(\frac{\gamma}{L^2}\right) \leq (1 - \kappa^{-2}) \|r_k\|^2
\]
Hence we get with (5)
\[
\|r_k\| \leq (1 - \kappa^{-2})^{k/2} \|r_0\|
\]
\[
\|u_k - u_*\| \leq \gamma^{-1} (1 - \kappa^{-2})^{k/2} \|r_0\|
\]
If we pick any \( \alpha_k \in (0, 2\gamma/L^2) \) we will have \( (1 - 2\alpha \gamma + \alpha^2 L^2) < 1 \) and hence \( \|r_{k+1}\|^2 = g_k(\alpha_k) < \|r_k\|^2 \).

In practice we can use an approximate line search: Pick \( q < 1 \) and stop the search when you achieve
\[
\|r_{k+1}\| \leq q \|r_k\|.
\]
This works if we pick \( q \geq (1 - \frac{\gamma}{L^2})^{1/2} \). Then we obtain
\[
\|r_k\| = \|b - F(u_k)\| \leq q^k \|r_0\| \to 0 \quad \text{as } k \to \infty
\]
Note that (3) implies
\[
\|u_k - u_*\| \leq \gamma^{-1} \|F(u_k) - b\|
\]
and \( \|u_k - u_*\| \leq \gamma^{-1} q^k \|r_0\| \to 0 \) as \( k \to \infty \).

**Corollary 2.2.** Let \( V \) be a Hilbert space. Assume that the function \( F: V \to V' \) satisfies with constants \( L, \gamma > 0 \) for all \( u, v \in V \)
\[
\|F(v) - F(u)\|_{V'} \leq L \|v - u\|
\]
\[
[F(v) - F(u)] \langle v - u \rangle \geq \gamma \|v - u\|^2
\]
Then the equation \( F(u) = f \) with \( f \in V' \) has a unique solution \( u \in V \).

**Proof.** By the Riesz representation theorem there is a linear mapping \( \phi: V' \to V \) such that \( \ell(v) = (\phi \ell, v) \) for all \( \ell \in V' \), \( v \in V \), and \( \|\phi \ell\|_{V'} = \|\ell\|_{V'} \). We define \( \tilde{F} := \phi \circ F: V \to V \) and can apply the previous Theorem. \( \square \)

**Corollary 2.3.** (Lax-Milgram) Let \( V \) be a Hilbert space. Assume that the bilinear form \( a: V \times V \to \mathbb{R} \) satisfies with constants \( L, \gamma > 0 \) for all \( u, v \in V \)
\[
|a(u, v)| \leq L \|u\| \|v\|
\]
\[
a(u, u) \geq \gamma \|u\|^2
\]
Then the equation there is a unique \( u \in V \) which satisfies
\[
a(u, v) = f(v) \quad \text{for all } v \in V
\]
and we have \( \|u\| \leq \gamma^{-1} \|f\|_{V'} \).

**Proof.** Define \( F: V \to V' \) by \( F(u) := a(u, \cdot \) . By the definition of \( \|\cdot\|_{V'} \) the function \( F \) satisfies the assumptions of 2.2. The estimate for \( \|u\| \) follows from \( \gamma \|u\|^2 \leq a(u, u) = f(u) \leq \|f\|_{V'} \|u\| \). \( \square \)
We can find the solution of Principle of virtual displacements: of \( L, \gamma \)

Then the equation \( Au = b \) has for \( b \in \mathbb{R}^n \) a unique solution and we have \( \| A^{-1} \|_2 \leq \gamma^{-1} \). The sharp values of \( L, \gamma \) from (9), (10) are given by

\[
L = \| A \|_2, \quad \gamma = \lambda_{\min}(A_{\text{symm}}) \quad \text{where } A_{\text{symm}} := \frac{1}{2} \left( A + A^\top \right)
\]

We can find the solution of \( Au = b \) with the following iterative methods:

1. **Richardson iteration with fixed \( \alpha \):** Let \( L := \| A \|_2 \), then for \( \alpha \in (0, 2\gamma/L^2) \) the iteration
   \[
u^{k+1} = u^k + \alpha (b - Au^k)
   \]
   converges. In particular, for \( \alpha = \gamma/L^2 \) we have \( \| u^{k+1} - u \|_2 \leq (1 - \gamma^2/L^2)^{1/2} \| u^k - u \|_2 \). A drawback of the method is that we need to know some (possibly nonoptimal) constants \( \gamma, L \) satisfying (20), \( \| A \|_2 \leq L \) in order to choose \( \alpha \) (or we have to experiment with different values of \( \alpha \)).

2. **1-step minimum residual:** We use (8) and choose \( \alpha \) so that the norm of the new residual \( r^{k+1} = b - Au^{k+1} \) becomes minimal:
   \[
   \| r^{k+1} \|_2^2 = \| r^k - \alpha Ar^k \|_2^2 = \| r^k \|_2^2 - 2\alpha (Ar^k, r^k) + \alpha^2 \| Ar^k \|_2^2
   \]
   i.e.,
   \[
   \alpha_k := \frac{(Ar^k, r^k)}{\| Ar^k \|_2^2}
   \]
   then
   \[
   \| r^{k+1} \|_2^2 = \| r^k \|_2^2 - \frac{(Ar^k, r^k)^2}{\| Ar^k \|_2^4} \leq \left( 1 - \frac{\gamma^2}{L^2} \right) \| r^k \|_2^2
   \]
   Therefore the residuals \( r^k = A(u - u^k) \) converge and
   \[
   \| r^k \|_2 \leq (1 - \gamma^2/L^2)^{k/2} \| r^0 \|_2, \quad \| u^k - u \|_2 \leq \gamma^{-1} \left( 1 - \frac{\gamma^2}{L^2} \right)^{k/2} \| r^0 \|_2
   \]
   This method corresponds to the first step of the GMRES method, or the GMRES(1) method which is restarted after every step. The full GMRES method minimizes the residuals over multiple directions, so the norm of the residual can only be lower. Hence the above estimates for \( \| r^k \|_2 \) and \( \| u^k - u \|_2 \) also hold for the GMRES method.

3 Equilibrium Problems

3.1 Abstract setting: Symmetric bilinear form \( a(u, v) \)

We want to find a “displacement” \( u \in V \). Here \( V \) is a complete vector space with a norm \( \| v \|_V \).

In the absence of external forces the equilibrium solution minimizes the “internal energy” \( Q(u) \).

If there are external forces: Now changing the displacement from \( u \) to \( u + v \) corresponds to a work \( \ell(v) \) (“work of external forces”). Here \( \ell: V \to \mathbb{R} \) is linear and bounded:

\[
\forall v \in V: \quad |\ell(v)| \leq C_\ell \| v \|
\]

**Principle of virtual displacements:** The equilibrium solution \( u \in V \) minimizes

\[
F(u) := Q(u) - \ell(u)
\]
For a linear problem the energy $Q(u)$ is quadratic and has the form

$$Q(u) = \frac{1}{2}a(u, u)$$

where $a: V \times V \rightarrow \mathbb{R}$ is bilinear and bounded:

$$\forall u, v \in V: |a(u, v)| \leq L \|u\|_V \|v\|_V$$

(9)

The bilinear form $a$ is symmetric: $a(u, v) = a(v, u)$ for all $u, v \in V$.

We want that the energy $Q(u) = \frac{1}{2}a(u, u)$ is positive for nonzero $u \in V$, moreover we need a lower bound.

We require that the bilinear form $a(u, v)$ is coercive: there exists $\gamma > 0$ such that

$$\forall u \in V: a(u, u) \geq \gamma \|u\|_V^2$$

(10)

This property is also called “$V$-ellipticity”.

**Minimization Problem (MIN):** Find $u \in V$ such that $F(u)$ is minimal.

We claim that (MIN) is equivalent to the following linear problem:

**Variational Problem (VAR):** Find $u \in V$ such that

$$\forall v \in V: a(u, v) = \ell(v)$$

**Theorem 3.1.** The following statements are equivalent:

1. $u$ minimizes $F(u)$ over all $u \in V$
2. $u$ satisfies (VAR)

**Proof.** (1) Assume $u \in V$ solves (MIN). Then for any $v \in V$ and any $\varepsilon \in \mathbb{R}$ we have

$$F(u) \leq F(u + \varepsilon v)$$

We have by using the bilinearity and symmetry of $a(\cdot, \cdot)$, and the linearity of $\ell(\cdot)$

$$F(u + \varepsilon v) = \frac{1}{2}a(u + \varepsilon v, u + \varepsilon v) - \ell(u + \varepsilon v) = \frac{1}{2}a(u, u) + \varepsilon a(u, v) + \frac{1}{2}\varepsilon^2 a(v, v) - \ell(u) - \varepsilon\ell(v)$$

$$F(u + \varepsilon v) = Q(u) + \varepsilon [a(u, v) - \ell(v)] + \frac{1}{2}\varepsilon^2 a(v, v)$$

Consider this as a function $f(\varepsilon)$ for $\varepsilon \in \mathbb{R}$. If this is minimal for $\varepsilon = 0$ then we must have $a(u, v) - \ell(v) = 0$ (otherwise we could achieve a smaller value with $\varepsilon \neq 0$).

(2) Assume $u$ satisfies (VAR). Then we have for any nonzero $v \in V$

$$F(u + v) = F(u) + [a(u, v) - \ell(v)] + \frac{1}{2}a(v, v)$$

0 by (VAR) > 0

since by coercivity $a(v, v) \geq \gamma \|v\|_V^2$ and $\|v\|_V > 0$ for nonzero $v$.

Note that $a(\cdot, \cdot)$ is an inner product on $V$ with the corresponding norm

$$\|u\|_a := a(u, u)^{1/2}$$

and by (16), (10) we have upper and lower bounds

$$\forall u \in V: \gamma^{1/2} \|u\|_V \leq \|u\|_a \leq L^{1/2} \|u\|_V$$

□
**Theorem 3.2.** The variational problem (VAR) has a unique solution \( u \in V \).

**Proof.** Note that \( V \) with the inner product \( a(\cdot, \cdot) \) forms a Hilbert space. The functional \( \ell: V \to \mathbb{R} \) is linear and bounded:

\[
|\ell(v)| \leq C \|v\|_a
\]

with \( C := \gamma^{-1/2} C_\ell \). By the Riesz representation theorem there exists a unique \( u \in V \) such that

\[
\forall v \in V: \quad a(u,v) = \ell(v)
\]

\[\square\]

### 3.2 Example Problem: Elastic string

The internal energy of the elastic string is given by

\[
\frac{1}{2} a(u,u), \quad a(u,v) := \int_{\Omega} c(x) u'(x)v'(x) dx
\]

where the “stiffness” \( c(x) \) satisfies

\[
0 < c_{\min} \leq c(x) \leq c_{\max} \quad \text{for } x \in \Omega
\]

For a force \( f(x) \) the work of the external force is given by

\[
\ell(v) = \int_{\Omega} f(x)v(x) dx
\]

Let \( H^1(\Omega) \) denote the space

\[
H^1(\Omega) := \{ u \mid u \in L^2(\Omega) \text{ and } u' \in L^2(\Omega) \}
\]

We define the space \( V \) as

\[
V = H^1_0(\Omega) := \{ u \mid u \in H^1(\Omega), \ u = 0 \text{ on } \partial\Omega \}
\]

with the norm \( \|u\|_{H^1} \).

We have the Poincare inequality: For \( u \in H^1_0(\Omega) \)

\[
\|u\|_{L^2} \leq C \|u'\|_{L^2}
\]

Hence we can define \( \|u\|_V := \|u'\|_{L^2} \) as norm on the space \( V \):

\[
\|u'\|_{L^2} \leq \|u\|_{H^1}^2 \leq (1 + C_\Omega) \|u'\|_{L^2}^2
\]

Now we can show that \( a(\cdot, \cdot) \) satisfies the boundedness with \( C_a = c_{\max} \): By Cauchy-Schwarz

\[
a(u,v) \leq \int_{\Omega} c_{\max} \|u'| \|v'\| dx \leq c_{\max} \|u'\|_{L^2} \|v'\|_{L^2}
\]

We can also show the coercivity of \( a(\cdot, \cdot) \) with \( \gamma = c_{\min} \):

\[
a(u,u) = \int_{\Omega} c(x) u'(x)^2 dx \geq \int_{\Omega} c_{\min} u'(x)^2 dx = c_{\min} \|u'\|_{L^2}^2
\]

Assume that the given force \( f(x) \) satisfies \( f \in L^2(\Omega) \). Then we can show that functional \( \ell: V \to \mathbb{R} \) is bounded with \( C_\ell = \|f\|_{L^2} \): By Cauchy-Schwarz

\[
|\ell(v)| = \left| \int_{\Omega} f(x)v(x) dx \right| \leq \|f\|_{L^2} \|v\|_{L^2}
\]

Hence we obtain from Theorems 3.1, 3.2 that for any given load \( f \in L^2(\Omega) \) the equilibrium problem has a unique solution \( u \in V \) satisfying (VAR).
3.3 Approximation of equilibrium solution

In our example the space \( V = H_0^1(\Omega) \) is infinite dimensional. On a computer we can only compute finitely many numbers.

Choose a finite dimensional subspace \( V_N \subset V \).

**Minimization Problem (MIN\(_N\)):** Find \( u_N \in V_N \) such that \( F(u_N) \) is minimal.

We claim that (MIN) is equivalent to the following linear problem:

**Variational Problem (VAR\(_N\)):** Find \( u_N \in V_N \) such that
\[
\forall v_N \in V_N : \quad a(u_N, v_N) = \ell(v_N)
\]

Since \( V_N \subset V \) all properties of \( a(u,v) \) and \( \ell(v) \) also hold for \( u, v \in V_N \). Therefore we obtain:

- \( u_N \) satisfies (MIN\(_N\)) \( \iff \) \( u_N \) satisfies (VAR\(_N\))
- problem (VAR\(_N\)) has a unique solution \( u_N \in V_N \)

Note that \( u_N \) is the best possible approximation of \( u \) in the space \( V_N \) with respect to \( \| \cdot \|_a \):
\[
\| u - u_N \|_a = \min_{v \in V_N} \| u - v_N \|_a
\]

To prove this we have to show that \( u_N \) satisfies the normal equations, i.e., \( a(u - u_N, v_N) = 0 \) for all \( v \in V_N \):

This follows from
\[
a(u_N, v_N) \overset{(VAR\(_N\))}{=} \ell(v_N) \overset{(VAR)}{=} a(u, v_N)
\]

In order to compute \( u_N \in V_N \) we choose a basis of \( V_N \). Assume \( \dim V_N = n \) we can pick a basis \( \phi_1, \ldots, \phi_n \).

Now we can write the solution \( u_N \in V_N \) as
\[
u_N = c_1 \phi_1 + \cdots + c_n \phi_n
\]

with a coefficient vector \( \tilde{c} \in \mathbb{R}^n \). We need that (VAR\(_N\)) holds for \( v = \phi_1, \ldots, \phi_n \). This gives an \( n \times n \) linear system:
\[
A \tilde{c} = \tilde{b}
\]

where the so-called “stiffness matrix” \( A \in \mathbb{R}^{n \times n} \) is the Gram matrix with entries
\[
A_{jk} = a(\phi_k, \phi_j)
\]

and the right-hand side vector \( \tilde{b} \in \mathbb{R}^n \) has entries
\[
b_j = \ell(\phi_j).
\]

This leads to the following algorithm:

1. Compute the entries of the stiffness matrix \( A \in \mathbb{R}^{n \times n} \)
2. Compute the entries of the right-hand-side vector \( \tilde{b} \in \mathbb{R}^n \)
3. Solve the \( n \times n \) linear system \( A \tilde{c} = \tilde{b} \)

We can use Gaussian elimination to solve the linear system. For a general \( n \times n \) matrix \( A \) this requires \( O(n^3) \) operations.
3.4 Variational problem with nonsymmetric \( a(u, v) \)

We also consider the case where the bilinear form \( a(u, v) \) satisfies (9) and (10), but \( a(u, v) \) is not symmetric. Note that in this case we don’t have a minimization problem (MIN), and we don’t have a norm \( \| \cdot \|_a \). So we need different proofs.

The variational formulation (VAR) has a unique solution by the Lax-Milgram theorem (2.3).

Again, we can choose a finite dimensional subspace \( V_N \subset V \) and consider the problem (VAR\(_N\)). By Lax-Milgram this problem has a unique solution \( u_N \).

Let \( v_N \in V_N \). By (VAR) we have \( a(u, v_N) = \ell(v_N) \), by (VAR\(_N\)) we have \( a(u_N, v_N) = \ell(v_N) \). Hence

\[
\forall v_N \in V_N: \quad a(u - u_N, v_N)
\]

and for any \( v_N \in V_N \)

\[
\gamma \| u - u_N \|^2_V \leq a(u - u_N, u - u_N) = a(u - u_N, u - v_N) \leq L \| u - u_N \|_V \| u - v_N \|_V
\]

This gives the so-called “quasioptimalitity”:

\[
\| u - u_N \|_V \leq \frac{L}{\gamma} \min_{v_N \in V_N} \| u - v_N \|_V
\]

I.e., the error is up to a constant \( \frac{L}{\gamma} \) optimal. Often we know something about the smoothness of \( u \), and we can prove the existence of approximations \( v_N \) with \( \| u - v_N \|_V \leq CN^{-\alpha} \). Then we obtain for the solutions \( u_N \) of (VAR\(_N\)) the same rate: \( \| u - u_N \|_V \leq \frac{L}{\gamma} CN^{-\alpha} \). We will use this idea to obtain convergence rates for the “p-version” and “h-version” methods.

3.5 p-version

Let \( \Omega = (-1, 1) \). We want to use finite dimensional spaces \( V_N \) so that we can obtain good approximations of our solution \( u(x) \).

One good choice are polynomials: For \( N \geq 2 \) we pick the space

\[
V_N = \{ u \in \mathcal{P}_N \mid u(-1) = u(1) = 0 \}
\]

and have \( n = \text{dim } V_N = N - 1 \).

Since \( u \in V \) we have \( u(-1) = u(1) = 0 \) and therefore \( \int_\Omega u'(x)dx = u(1) - u(-1) = 0 \).

We now want to construct \( v_N \in V_N \) such that \( \| u - v_N \|_a \) is small. Note that

\[
\| u - v_N \|_a^2 \leq c_{\text{max}} \| u' - v'_N \|_{L^2}^2
\]

Therefore we have to find \( q_{N-1} := v'_N \in \mathcal{P}_{N-1} \) such that \( \| u' - q_{N-1} \|_{L^2} \) is small. We define \( q_{N-1} \) as the \( L^2 \) projection of \( u' \) onto the space \( \mathcal{P}_{N-1} \). Then we must have \( \int_\Omega (u' - q_{N-1})r \ dx = 0 \) for all \( r \in \mathcal{P}_{N-1} \). In particular for \( r(x) = 1 \) we obtain \( \int_\Omega (u' - q_{N-1})dx = 0 \), hence \( \int_\Omega q_{N-1}(x)dx = 0 \). We now define \( v_N \) as the antiderivative

\[
v_N(x) := \int_{-1}^x q_{N-1}(t)dt,
\]

then clearly \( v_N \in \mathcal{P}_N \) and \( v_N(-1) = 0 \), \( v_N(1) = 0 \), i.e., \( v_N \in V_N \).

Now we can give an upper bound for \( \| u' - q_{N-1} \|_{L^2} \). Since \( q_{N-1} \) is the best \( L^2 \) approximation we have for any \( \tilde{q}_{N-1} \in \mathcal{P}_{N-1} \)

\[
\| u' - q_{N-1} \|_{L^2} \leq \| u' - \tilde{q}_{N-1} \|_{L^2} \leq \| u' - \tilde{q}_{N-1} \|_{L^2}
\]

where we use the weight function \( w(x) = (1 - x^2)^{-1/2} \geq 1 \) on \( \Omega = (-1, 1) \).

The function \( u' \) has a Chebyshev expansion

\[
u' = \sum_{k=0}^{\infty} a_k T_k
\]
and we choose \( \tilde{q}_{N-1} \in \mathcal{P}_{N-1} \) as the truncated Chebyshev expansion (this minimizes \( \| u' - \tilde{q}_{N-1} \|_{L^2} \))

\[
\tilde{q}_{N-1} := \sum_{k=0}^{N-1} a_k T_k
\]

Then we have

\[
\| u' - \tilde{q}_{N-1} \|_{L^2}^2 = \sum_{k=0}^{N-1} |a_k|^2 \| T_k \|_{L^2}^2 / \pi
\]

Therefore we obtain the upper bound:

\[
\| u - u_N \|_2^2 \leq c_{\max} \pi \sum_{k=N}^{\infty} |a_k|^2
\]  
(12)

where \( a_k \) are the Chebyshev coefficients of the function \( u'(x) \).

We can use this to get results for an arbitrary interval \( \Omega = (x_{\text{left}}, x_{\text{right}}) \).

By using the results for the decay of \( |a_k| \) we obtain the following convergence rates:

- **if** \( u \) **has derivatives** \( u', \ldots, u^{(m)} \) **and** \( u^{(m)} \) **has bounded variation**, with \( m \geq 1 \):
  \[
  \| u - u_N \|_{H^1} \leq CN^{-m}
  \]

**Example:** for \( -u'' = \delta_0 \) on \( \Omega = (-1, 1) \) we obtain \( u(x) = (1 - |x|)/2 \). Here \( m = 1 \) since \( u' \in BV \).

Hence \( \| u - u_N \|_{H^1} \leq CN^{-1} \)

- **if** \( u \) **is analytic on** \( \Omega \):
  \[
  \| u - u_N \|_{H^1} \leq Ce^{-\beta N}
  \]
  with \( \beta > 0 \).

- **if** \( u \) **has the form** \( u(x) = c |x - x_0|^\alpha + \text{smoother terms with } \alpha > 1/2 \) **and** \( x_0 \in \Omega \):
  \[
  \| u - u_N \|_{H^1} \leq CN^{-(\alpha - 1/2)}
  \]

Note that we used the estimate \( \| u'' - \tilde{q}_{N-1} \|_{L^2} \leq \| u' - \tilde{q}_{N-1} \|_{L^2} \) in (11). Therefore (12) does not always give sharp results. E.g., for a singularity \( |x - x_{\text{left}}| \alpha \) at the endpoint of the interval one can obtain with a different proof that \( \| u - u_N \|_{H^1} \leq CN^{-2(\alpha - 1/2)} \).

### 3.6 h-version

We have a domain \( \Omega \subset \mathbb{R}^d \) and we will consider the cases \( d = 1, d = 2 \) and \( d = 3 \).

**Error** \( \| f' - p' \|_{L^2([0,h])} \) **for linear interpolation on an interval**

Let \( f \in C([0,h]) \). Let \( p(x) = f(0) + \frac{f(h) - f(0)}{h} x \) denote the polynomial \( p \in \mathcal{P}_1 \) interpolating \( f \) at the nodes 0, h. If \( f'' \) is continuous we know that \( |f(x) - p(x)| \leq \frac{h^2}{2} \| f'' \|_{\infty} \) for \( x \in [0, h] \) from the classical formula for the interpolation error, i.e., the error is of order \( O(h^2) \). The error \( |f' - p'| \) for the derivatives is of order \( O(h^1) \):

1. Consider a function \( f(t) \) on the interval \([0, 2\pi]\) with Fourier series \( f = \sum_{k=-\infty}^{\infty} c_k e^{ikt} \). The mean value of \( f \) on \([0, 2\pi]\) is \( c_0 \), hence the Parseval identity for \( f(t) - c_0 = \sum_{k \neq 0} c_k e^{ikt} \) gives
   \[
   \| f - c_0 \|_{L^2([0,2\pi])}^2 = 2\pi \sum_{k \neq 0} |c_k|^2 \leq 2\pi \sum_{k=-\infty}^{\infty} k^2 |c_k|^2 = \| f' \|_{L^2([0,2\pi])}^2
   \]  
(13)

Here the last equality is the Parseval identity for \( f'(t) = \sum_{k=-\infty}^{\infty} c_k k e^{ikt} \).

2. Consider a function \( g(x) \) on an interval \([0, h]\). Let \( c_0 = h^{-1} \int_0^h g(x) dx \) denote the mean value on \([0, h]\).
Let \( f(t) := g(t \frac{h}{2\pi}) \), then \( f'(t) = g'(t \frac{h}{2\pi}) \cdot \frac{h}{2\pi} \). Then (13) gives with the change of variables \( x = t \frac{h}{2\pi} \)

\[
\| g - c_0 \|_{L^2([0,h])}^2 \leq \left( \frac{h}{2\pi} \right)^2 \| g' \|_{L^2([0,h])}^2
\]  
(14)
(3) Consider a function \( u(x) \) on an interval \([0,h]\). Let \( p \in P_1 \) denote the interpolating polynomial for the nodes 0, h. Then
\[
p'(x) = \frac{u\left(\frac{h}{2}\right) - u\left(-\frac{h}{2}\right)}{h} = h^{-1} \int_0^h u'(t)dt,
\]
i.e., \( p' \) is the mean value \( c_0 \) of the function \( g(x) := u'(x) \). Hence (14) gives
\[
\left\| u' - p' \right\|_{L^2([0,h])} \leq \frac{h}{2\pi} \left\| u'' \right\|_{L^2([0,h])} \tag{15}
\]

1-dimensional case

For \( d = 1 \) the domain is an interval \( \Omega = (x_{\text{left}}, x_{\text{right}}) \). We use a partition
\[
x_{\text{left}} = x^0 < x^1 < \cdots < x^{N-1} < x^N = x_{\text{right}}
\]
which divides the interval \((a,b)\) into \( N \) subintervals and define
\[
V_N := \left\{ u \in C(\bar{\Omega}) \mid u\big|_{[x^i-1,x^i]} \text{ is linear for } j = 1, \ldots, N; u|_{\partial \Omega} = 0 \right\}
\]
Let \( h_j := x^j - x^{j-1} \) and \( h_{\max} := \max_{j=1,\ldots,N} h_j \).
The dimension of \( V_N \) is \( n := N - 1 \). A function \( v \in V_N \) is specified by its nodal values \( v_j := v(x^j) \), \( j = 1, \ldots, n \). We use the nodal basis functions \( \phi_1, \ldots, \phi_n \) for \( V_N \): The function \( \phi_j \) has nodal values
\[
\phi_j(x^k) = \begin{cases} 1 & \text{if } k = j \\ 0 & \text{if } k \neq j \end{cases}
\]
The entries of the stiffness matrix \( A \) and the right hand side vector \( b \) are given by
\[
a_{jk} = a(\phi_k, \phi_j), \quad b_j = \ell(\phi_j)
\]
Note that \( a(\phi_k, \phi_j) = 0 \) if \(|j - k| \geq 2\). Hence the matrix \( A \) is tridiagonal, and the linear system \( Ac = b \) can be solved with \( O(n) \) operations.

For a function \( u \) on \( \bar{\Omega} \) we denote by \( u_I \in V_N \) the piecewise linear interpolation: \( u_I := u(x^1)\phi_1 + \cdots + u(x^n)\phi_n \). Then we apply (15) on each subinterval \([x^{j-1},x^j]\) and obtain
\[
\left\| u' - u'_I \right\|_{L^2(\Omega)} \leq \frac{h_{\max}}{2\pi} \left\| u'' \right\|_{L^2(\Omega)}
\]
This implies \( \left\| u' - u'_I \right\|_2 \leq Ch_{\max} \). Therefore we obtain for the solutions \( u_N \) of (VAR\(_N\)) from the quasioptimality
\[
\left\| u - u_N \right\|_V \leq C'h_{\max}
\]
For equidistant nodes with \( h_j = (x_{\text{right}} - x_{\text{left}})/N \) this gives \( \left\| u - u_N \right\|_V \leq C''N^{-1} \).

2-dimensional case

In two dimensions we assume that \( \Omega \) is a polygon. We divide \( \Omega \) into triangles \( T_j \), \( j = 1, \ldots, M \). The triangles should have size \( h = N^{-1} \): we want that there are constants \( c, C \) such
- each triangle fits into a circle of radius \( Ch \)
- we can fit a circle of radius \( ch \) into the interior of each triangle

We define \( V_N \) as
\[
V_N := \left\{ u \in C(\bar{\Omega}) \mid u\big|_{T_j} \text{ is linear for } j = 1, \ldots, M; u|_{\partial \Omega} = 0 \right\}
\]
Let \( x^1, \ldots, x^n \) denote the vertices of the mesh in the interior of \( \Omega \). A function \( v \in V_N \) is specified by the nodal values \( v_j := v(x^j), j = 1, \ldots, n \). We use the nodal basis functions \( \phi_1, \ldots, \phi_n \) for \( V_N \): The function \( \phi_j \) has nodal values \( \phi_j(x^k) = \begin{cases} 1 & \text{if } k = j \\ 0 & \text{if } k \neq j \end{cases} \).

For a function \( u \) on \( \Omega \) we denote by \( u_I \in V_N \) the piecewise linear interpolation:

\[
u_I = u(x^1)\phi_1 + \cdots + u(x^n)\phi_n.
\]

One can prove that

\[
\| \nabla u - \nabla u_I \|_{L^2(\Omega)} \leq C h |u|_{H^2(\Omega)} \quad \text{where } |u|^2_{H^2(\Omega)} := \sum_{j,k=1}^2 \left\| \frac{\partial^2 u}{\partial x_i \partial x_j} \right\|^2_{L^2(\Omega)}
\]

Therefore we obtain for the solutions \( u_N \) of \((\text{VAR}_N)\) from the quasioptimality

\[
\| u - u_N \|_V \leq C' h
\]

3-dimensional case

In two dimensions we assume that \( \Omega \) is a polyhedron. We divide \( \Omega \) into tetrahedra \( T_j, j = 1, \ldots, M \). The tetrahedra should have size \( h = N^{-1} \): they should fit inside a sphere of radius \( Ch \), and they should contain a sphere of radius \( ch \). We can now proceed as in the 2-dimensional case and obtain from the quasioptimality that \( \| u - u_N \|_V \leq C' h \).

### 4 Solution of the linear system: Direct solvers versus iterative solvers

In many applications we have to solve a linear system \( Ax = b \) with \( A \in \mathbb{R}^{n \times n} \) and \( b \in \mathbb{R}^n \) given. If \( n \) is large the solution of the linear system takes a lot of operations, and standard Gaussian elimination may take too long.

But in many cases most entries of the matrix \( A \) are zero and \( A \) is a so-called sparse matrix. This means each equation only couples very few of the \( n \) unknowns \( x_1, \ldots, x_n \). A typical example are discretizations of partial differential equations, see next section for an example.

**Direct solvers** will give the exact solution after finitely many operations (if we ignore roundoff errors).

**Gaussian elimination with partial pivoting:** This gives a decomposition \( LU = \begin{bmatrix} \text{row } p_1 \text{ of } A \\ \vdots \\ \text{row } p_n \text{ of } A \end{bmatrix} \) where \( L \) is lower triangular, \( U \) is upper triangular.

**Cholesky decomposition:** We need that \( A \) is symmetric positive definite. This gives a decomposition \( A = LL^\top \) where \( L \) is lower triangular.

Cholesky decomposition takes about half the number of operations of Gaussian elimination.

**Cost for Gaussian elimination and Cholesky algorithm:**

- for full matrices finding the decomposition takes \( Cn^3 \) operations.
  Once have the decomposition solving the linear system for a given vector \( b \) takes \( n^2 \) operations.

- for band matrices with bandwidth \( m \), i.e., \( A_{ij} = 0 \) for \( |i - j| > m \):
  Finding the decomposition takes \( Cm^2n \) operations, solving a linear system then takes \( Cmn \) operations.

In Matlab we should initialize the matrix \( A \) as a sparse matrix structure. Then Matlab will only use storage and operations to compute the nonzero elements of \( L, U \).

For a matrix \( A \) with bandwidth \( m \) the factors \( L, U \) will also have bandwidth \( m \).

For a general sparse matrix \( A \), the factors \( L, U \) will usually have additional nonzero elements at locations where \( A \) had zero elements. This is called **fill-in**, and this increases the number of operations.
Reordering: If you use the Matlab command \( x = A \backslash b \) (where the matrix has sparse array type) then Matlab will try to renumber the unknowns in such a way that the amount of fill-in will minimized. This can substantially reduce the number of operations.

The command `spparms('spumoni',2)` makes Matlab print out details about the algorithms used for each following \( \backslash \) command. Try the following example for solving \(-\Delta u = 1\) on the square \( \Omega = (0,1)^2 \):

```matlab
N = 50;
A = delsq(numgrid('S',N+1));
figure(1); spy(A)
figure(1)
n = size(A,1);
b = ones(n,1)/N^2;
spparms('spumoni',2) % print out details about what "\" does
u = A\b;
figure(2); surf(reshape(u,N-1,N-1))
```

There are also versions of the \( \text{lu} \) and \( \text{chol} \) commands that use reordering to minimize fill-in.

5 Convection diffusion problem in \( \mathbb{R}^d \)

We consider a typical application problem which leads to a large sparse linear system. Equilibrium problems for elastic deformations or heat transfer lead to elliptic differential equations.

Boundary value problem

In the convection diffusion problem we have a domain \( \Omega \subset \mathbb{R}^d \). For \( d = 1 \) we consider an interval, for \( d = 2 \) we consider a polygon, for \( d = 3 \) we consider a polyhedron.

We want to find a function \( u(x) \) for \( x \in \Omega \) such that

\[-\Delta u + b \cdot \nabla u = f \quad \text{in } \Omega, \quad u = 0 \quad \text{on the boundary } \partial \Omega \]

where \( b \in \mathbb{R}^n \) is a constant vector, and \( f \) is a given function on \( \Omega \). This is called a boundary value problem.

Variational formulation

We first want to find the “variational formulation”: If we multiply the PDE by a “test function” \( v \) which is zero on the boundary and integrate over \( \Omega \) we obtain after using the first Green formula

\[
\int_{\Omega} \nabla u \cdot \nabla v \, dx + \int_{\Omega} (b \cdot \nabla u) v \, dx = \int_{\partial \Omega} (\partial_n u) v \, ds
\]

We use the Hilbert space \( V = H^1_0(\Omega) = \{ u \mid \int_{\Omega} \left( \| \nabla u \|_2^2 + |u|^2 \right) \, dx < \infty, \ u|_{\partial \Omega} = 0 \} \) with the norm

\[
\| u \|^2_V = \int_{\Omega} \left( \| \nabla u \|_2^2 + |u|^2 \right) \, dx = \| \nabla u \|_{L^2(\Omega)}^2 + \| u \|_{L^2(\Omega)}^2.
\]

Note that \( \int_{\Omega} (b \cdot \nabla u) u \, dx = \int_{\Omega} (b_1 u_{x_1} + b_2 u_{x_2}) u \, dx = 0 \): For \( \int_{x_2 = a}^{b(x_2)} \int_{x_1 = a(x_2)}^{b(x_1)} u_{x_1} u \, dx \, dx_2 \) integration by parts gives for the inner integral \( \int_{x_1 = a(x_2)}^{b(x_1)} u_{x_1} u \, dx_1 = - \int_{x_1 = a(x_2)}^{b(x_2)} u_{x_1} u \, dx_1 \) since \( u \) is zero on the boundary.

We then obtain that \( a(\cdot, \cdot) : V \times V \to \mathbb{R} \) is a bilinear form satisfying for all \( u, v \in V \)

\[
a(u, v) \leq L_a \| u \|_V \| v \|_V \quad \text{(16)}
\]

\[
a(u, u) \geq \gamma_a \| u \|_V^2 \quad \text{(17)}
\]
The first inequality follows from the Cauchy-Schwartz inequality. For the second inequality we use \( a(u, v) = \int \nabla u \cdot \nabla u \, dx \) and the Poincare inequality \( \|u\|_{L^2(\Omega)} \leq C_\Omega \|\nabla u\|_{L^2(\Omega)} \).

We obtain that \( \ell: V \to \mathbb{R} \) is a linear functional such that

\[
|\ell(v)| \leq C_\ell \|v\|_V.
\]

The variational formulation is: Find \( u \in V \) such that

\[
\forall v \in V : \quad a(u, v) = \ell(v)
\]

By the Lax-Milgram theorem (see Appendix A below) this variational problem has a unique solution \( u \in V \).

**Finite element discretization**

We choose a finite dimensional subspace \( V_h \subset V \).

For \( d = 2 \) the domain \( \Omega \) is a polygon, and we divide it into a mesh of triangles. (For \( d = 1 \) we divide the interval into subintervals, for \( d = 3 \) we divide the polyhedron into a mesh of tetrahedra).

Then we define \( V_h \) as the space of piecewise linear functions on the mesh which are continuous in \( \Omega \) and are zero on the boundary \( \partial \Omega \).

The discrete problem is: Find \( u_h \in V_h \) such that

\[
\forall v_h \in V_h : \quad a(u_h, v_h) = \ell(v_h)
\]

(18)

Since \( V_h \subset V \) the inequalities (16), (17) are satisfied for \( u, v \in V_h \). Hence by the Lax-Milgram theorem the discrete problem has a unique solution \( u_h \).

We can specify a function \( v_h \in V_h \) by specifying the values \( v_1, \ldots, v_n \) at the interior nodes \( x_1, \ldots, x_n \) of the mesh. The basis function \( \phi_j \) is the function in \( V_h \) with \( \phi_j(x_j) = 1 \) and \( \phi(x_k) = 0 \) for \( k \neq j \). We can then write \( u_h \) as

\[
u_h = u_1 \phi_1 + \cdots + u_n \phi_n
\]

where \( u = \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} \in \mathbb{R}^n \) is the coefficient vector. Now (18) for \( v_h = \phi_1, \ldots, \phi_n \) gives the linear system

\[
Au = b, \quad A_{jk} = a(\phi_k, \phi_j), \quad b_j = \ell(\phi_j).
\]

Therefore the finite element method involves the following steps:

- pick a mesh on \( \Omega \)
- assemble the stiffness matrix \( A \) and the right hand side vector \( b \)
- solve the linear system \( Au = b \)

**Work for direct solvers**

For \( d = 1 \) we obtain a tridiagonal matrix \( A \). Hence the work is proportional to \( N = 1/h \).

As a simple example for \( d = 3 \) consider the cube \( \Omega = (0, 1)^3 \). Let \( h = 1/N \) with positive integer \( N \). By using uniform grids \( x_1 = j_1/N, x_2 = j_2/N, x_3 = j_3/N \) with \( j_1, j_2, j_3 \in \{0, \ldots, N\} \) for each coordinate we can subdivide \( \Omega \) into \( N^3 \) smaller cubes. We can then subdivide each of the smaller cubes into tetrahedra. We have \( n = (N-1)^3 \) interior nodes with \( j_1, j_2, j_3 \in \{1, \ldots, N-1\} \), and we can order them lexicographically by \((j_1, j_2, j_3): (1, 1, 1), \ldots, (1, N-1), (1, 2, 1), \ldots, (N, N, N) \). Then the resulting stiffness matrix \( A \) has size \( n \times n \) with \( n = (N-1)^3 \), and bandwidth \((N-1)^2\).
This will also hold for a more general domain $\Omega \subset \mathbb{R}^d$, assuming that all triangles/tetrahedra are of “size $h$, up to a constant”: We will have $n = \dim V_h \approx ch^{-d}$ and a bandwidth $m \approx c' h^{1-d}$. Therefore the work for Gaussian elimination is with $h \approx cn^{-1}$

$$m^2n \approx C \left( N^{d-1} \right)^2 N^d = CN^{3d-2}$$

For $d = 2$ we have therefore $O(N^4)$ operations. For $d = 3$ we have $O(N^7)$ operations. Using Matlab’s reordering algorithms reduces the work to $N^3$ for $d = 2$, but for $d = 3$ it does not improve the rate $O(N^7)$.

**Work of direct solvers for the convection-diffusion problem:**

<table>
<thead>
<tr>
<th></th>
<th>$d = 1$</th>
<th>$d = 2$</th>
<th>$d = 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian elimination (using band structure)</td>
<td>$N^1$</td>
<td>$N^4$</td>
<td>$N^7$</td>
</tr>
<tr>
<td>Gaussian elimination with reordering</td>
<td>$N^1$</td>
<td>$N^3$</td>
<td>$N^7$</td>
</tr>
</tbody>
</table>

**Estimates** $(A\bar{u}, \bar{v}) \leq L \|\bar{u}\|_2 \|\bar{v}\|_2$ and $(A\bar{u}, \underline{\bar{u}}) \geq \gamma \|\bar{u}\|_2^2$ for the stiffness matrix $A$

A function $v_h \in V_h$ is given by a coefficient vector $\bar{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$. For the function $v_h$ we have the norms $\|v_h\|_{L^2}$ and $\|\nabla v_h\|_{L^2}$. How are these norms related to the norm $\|\bar{v}\|_2$ the coefficient vector $\bar{v}$?

We assume that all triangles are of “size $h$, up to a constant”. More precisely: We assume that a circle of radius $c_0 h$ fits inside each triangle, and each triangle fits inside a circle of radius $C_0 h$. Then one can show that there exist constants $c_1, c_2, c_3$ depending on $c_0$ and $C_0$ such that

$$c_1 h^{d/2} \|\bar{v}\|_2 \leq \|v_h\|_{L^2} \leq c_2 h^{d/2} \|\bar{v}\|_2$$

This implies

$$\|v_h\|_{H^1}^2 = \|\nabla v_h\|_{L^2}^2 + \|v_h\|_{L^2}^2 \leq c' h^{d-2} \|\bar{v}\|_2^2$$

Hence we obtain for functions $u_h, v_h \in V_h$ with coefficient vectors $\bar{u}, \bar{v}$

$$(A\bar{u}, \bar{v}) = \int \bar{u} \cdot \nabla \bar{v} \, dx, \quad (A\bar{u}, \underline{\bar{u}}) = \int \frac{1}{2} (A + A^T)\bar{u} \cdot \bar{v} \, dx$$

Hence we obtain

$$L = \|A\|_2 \leq Ch^{d-2}, \quad \gamma = \lambda_{\min} \left( \frac{1}{2} (A + A^T) \right) \geq C' h^d$$

We can split the bilinear form $a(u, v)$ into a diffusion part and a convection part

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v \, dx + \int_\Omega (b \cdot \nabla u) v \, dx,$$

hence we have $A = A_{\text{diff}} + A_{\text{conv}}$ with

$$|(A_{\text{diff}} \bar{u}, \bar{v})| = \left| \int_\Omega \nabla u_h \cdot \nabla v_h \, dx \right| \leq \|\nabla u_h\|_{L^2} \|\nabla v_h\|_{L^2} \leq c_2 h^{d-2} \|\bar{u}\|_2 \|\bar{v}\|_2$$

$$|(A_{\text{conv}} \bar{u}, \bar{v})| = \left| \int_\Omega \nabla u_h \cdot (b v_h) \, dx \right| \leq \|\nabla u_h\|_{L^2} \|b\|_2 \|v_h\|_{L^2} \leq c_3 \|b\|_2 h^{d-1} \|\bar{u}\|_2 \|\bar{v}\|_2$$
We want to solve the linear system $Au = b$ where $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^{n}$. We assume that the symmetric part of $A$ is positive definite. Then we have

\[ (Au, u) \geq \gamma \|u\|_2^2 \quad \text{and} \quad |(Au, v)| \leq L \|u\|_2 \|v\|_2 \quad \text{for all } u, v \in \mathbb{R}^{n} \]

with $\gamma = \lambda_{\min}\left(\frac{1}{2}(A + A^T)\right) > 0$ and $L = \|A\|_2$.

The current guess is $u^{(k)}$. We compute the residual $r^{(k)} := b - Au^{(k)}$ and define the new guess as

\[ u^{(k+1)} := u^{(k)} + \alpha_k r^{(k)} \]

where we choose $\alpha_k$ such that the new residual has $r^{(k+1)} := b - Au^{(k+1)}$ has minimum norm $\|r^{(k+1)}\|_2$, yielding

\[ \alpha_k := \frac{(Ar^{(k)}, r^{(k)})}{\|Ar^{(k)}\|_2^2} \]

Note that each step requires one matrix-vector product $Ar^{(k)}$ (and a few inner products of vectors).

If the matrix $A$ satisfies (19), (20) we obtained earlier

\[ \|r^{(k+1)}\|_2 \leq (1 - K^{-1})^{1/2} \|r^{(k)}\|_2 \quad \text{with } K = \left(\frac{L}{\gamma}\right)^2 \]

implying

\[ \|r^{(k)}\|_2 \leq (1 - K^{-1})^{k/2} \|r^{(0)}\|_2, \quad \|u^{(k)} - u\|_2 \leq \gamma^{-1} (1 - K^{-1})^{k/2} \|r^{(0)}\|_2 \]

Let $\kappa := L/\gamma$. Since $\|A^{-1}\|_2 \leq \gamma^{-1}$ we have $\text{cond}_2(A) \leq \kappa$:

\[ \text{cond}_2(A) = \|A\|_2 \|A^{-1}\|_2 \leq L\gamma^{-1} = \kappa. \]

If the matrix $A$ is symmetric we have $\text{cond}_2(A) = \kappa$:

\[ L = \|A\|_2 = \lambda_{\max}(A), \quad \gamma = \lambda_{\min}(A), \quad \text{cond}_2(A) = \|A\|_2 \|A^{-1}\|_2 = L\gamma^{-1} = \kappa. \]

Assume that we have $\|r^{(k+1)}\| \leq q \|r^{(k)}\|$ with $q = 1 - \varepsilon$. In order to achieve $\|r^{(k)}\|_2 \leq \delta$ we need to pick $k$ such that

\[ \|r^{(k)}\|_2 \leq q^k \|r^{(0)}\|_2 \leq \delta, \quad \text{hence } k \geq \frac{\log(\delta/\|r^{(0)}\|_2)}{\log q} \]

For $q = 1 - \varepsilon$ the first order Taylor approximation gives $\log(1 - \varepsilon) \approx -\varepsilon$, hence we need approximatively

\[ k \geq \varepsilon^{-1} \log \left(\frac{\|r^{(0)}\|_2}{\delta}\right) = \varepsilon^{-1} C_{\delta} \]

steps for the iterative method.

Here we have $q = (1 - K^{-1})^{1/2} \approx 1 - \frac{1}{2} K^{-1}$ using Taylor. Hence we need

\[ k \geq 2C_{\delta} K = 2C_{\delta} \kappa^2 \]

steps for the iterative method.

Note that for the convection diffusion problem we have $\kappa = Ch^{-2}$ and $q \leq 1 - ch^4$. Therefore it would seem that we need $Ch^{-4}$ steps of our iterative method. But it turns out that this estimate is too pessimistic. Actually we have $q \leq 1 - ch^2$ and we need only $Ch^{-2}$ steps of our iterative method as we will see in the next section.
7 Sharper estimates for the convergence factor

Symmetric case

Recall that

$$\|r^{(k+1)}\|_2 = \|(I - \alpha A)r^{(k)}\|_2 \leq \|I - \alpha A\|_2 \|r^{(k)}\|_2.$$  

If \(A\) is symmetric, then also \(I - \alpha A\) is symmetric and we have with the eigenvalues \(\lambda_1, \ldots, \lambda_n\) of \(A\)

$$\|I - \alpha A\|_2 = \max_{j=1, \ldots, n} |1 - \alpha \lambda_j|$$  \hspace{1cm} (23)

If \(A\) is positive definite, the eigenvalues are positive. We will have \(\|I - \alpha A\| < 1\) if \(\alpha > 0\) and \(1 - \alpha \lambda_{\max} > -1\), i.e.,

$$\alpha \in (0, 2/\lambda_{\max})$$

We can minimize (23) by choosing \(\alpha\) such that

$$-1 + \alpha \lambda_{\max} = -(1 - \alpha \lambda_{\min})$$

yielding

$$\alpha = \frac{2}{\lambda_{\max} + \lambda_{\min}}, \quad \|I - \alpha A\|_2 = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} = 1 - \frac{2}{\kappa + 1} \quad \text{with} \quad \kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$$

So the convergence factor is \(q = 1 - \frac{2}{\kappa + 1}\) and the number of iterations is proportional to \(\kappa = \text{cond}_2(A)\) (and not \(\kappa^2\) as the earlier estimate (21) would suggest).

Nonsymmetric case

We can write \(A\) as a sum of a symmetric part \(H\) and antisymmetric part \(S\):

$$A = H + S, \quad H := \frac{1}{2}(A + A^T), \quad S := \frac{1}{2}(A - A^T)$$

We assume that \(A\) is positive definite, i.e., \((Av, v) = (Hv, v) > 0\) for \(v \neq 0\). Let \(u\) denote the current guess, and \(r := b - Au\) the residual.

The next approximation is \(u^{\text{new}} = u + \alpha r\), with the residual \(r^{\text{new}} = b - Au^{\text{new}} = (I - \alpha A)r\). Hence

$$\|r^{\text{new}}\|^2 = ((I - \alpha A)r, (I - \alpha A)r) = \|r\|^2 - 2\alpha (Ar, r) + \alpha^2 \|Ar\|^2$$

Note that both \(\|Ar\|\) and \((Ar, r)^{1/2} = (Hr, r)^{1/2}\) define norms on \(\mathbb{R}^n\). Therefore there exists \(C > 0\) such that

$$\|Ar\|^2 \leq C (Ar, r) \quad \text{for all} \quad r \in \mathbb{R}^d$$  \hspace{1cm} (24)

Then

$$\|r^{\text{new}}\|^2 \leq \|r\|^2 + [-2\alpha + C\alpha^2] (Ar, r)$$

The bound is minimal for \(\alpha = C^{-1}\), and with this we get

$$\|r^{\text{new}}\|^2 \leq \|r\|^2 - C^{-1} (Ar, r) \leq \left[1 - \frac{\gamma}{C}\right] \|r\|^2$$

It remains to find \(C\) such that (24) holds: with \(v = Ar\) we get

$$(Ar, r) = (Hr, r) = (HA^{-1}v, A^{-1}v)$$

Then we obtain \((v, v) \leq C(A^{-T}HA^{-1}v, v)\) with \(C = \lambda_{\min}(B)^{-1} = \lambda_{\max}(B^{-1})\) since \(B\) is symmetric. Hence we need an estimate \((w, B^{-1}w) \leq C (w, w)\) with \(B^{-1} = AH^{-1}A^T\): Using \(A^T = H - S\) we get

$$(w, B^{-1}w) = ((H - S)w, H^{-1}(H - S)w) = (Hw, w) - (Sw, w) - (Hw, H^{-1}Sw) + (Sw, H^{-1}Sw) \leq$$

$$\lambda_{\max}(H) \|w\|^2 + \lambda_{\min}(H) \|Sw\|^2$$
Since \((Sw, Sw) = (-S^2w, w) \leq \rho(S)^2 \|w\|^2\) we obtain
\[
C = \lambda_{\text{max}}(H) + \frac{\rho(S)^2}{\lambda_{\text{min}}(H)}
\]

Note:
- the eigenvalues of \(H\) are real and positive.
- the eigenvalues of \(S\) are of the form \(+\alpha_j i\) with \(\alpha_j \geq 0\).

**Proof:** Let \(\mu_j\) denote the eigenvalues of \(S\). The matrix \(S^2\) is symmetric and has real eigenvalues \(\mu_j^2 \leq 0\) because \((S^2w, w) = -(Sw, Sw) \leq 0\). Since the matrix \(S\) is real, taking the complex conjugate of \(Sw = \mu_j v\) gives \(\overline{Sw} = \mu_j \overline{v}\). Hence \(\overline{\mu_j}\) is also an eigenvalue of \(S\).

**Theorem 7.1.** Let \(A \in \mathbb{R}^{n \times n}\), let \(H := \frac{1}{2}(A + A^\top)\), \(S := \frac{1}{2}(A - A^\top)\). If \(A\) is positive definite, i.e., \(\lambda_{\text{min}}(H) > 0\) the 1-step minimum residual iteration satisfies
\[
\| r^{(k+1)} \|_2 \leq (1 - K^{-1})^{1/2} \| r^{(k)} \|_2 \quad K := \text{cond}_2(H) + \left( \frac{\rho(S)}{\lambda_{\text{min}}(H)} \right)^2
\]

**Note:** The number of iterations is proportional to \(K\). In our earlier estimate (22) we had \(K = \left( \frac{\|A\|_2}{\gamma} \right)^2\) whereas we now obtain
\[
K = \frac{\|H\|_2}{\gamma} + \left( \frac{\|S\|_2}{\gamma} \right)^2
\]

This shows that for a symmetric matrix \(A = H\) the number of steps is proportional to the condition number. If we have nonsymmetric \(A = H + S\) then \(K\) increases by \((\|S\|_2/\gamma)^2\). So we see that the quadratic term \((\|A\|_2/\gamma)^2\) in our earlier estimate is actually only caused by the antisymmetric part \(S\).

**Application to convection diffusion problem**
Recall that \(A = A_{\text{diff}} + A_{\text{conv}}\) with the symmetric matrix \(H = A_{\text{diff}}\) and the antisymmetric matrix \(S = A_{\text{conv}}\) and
\[
C_1 h^d \|u\|^2_2 \leq (A_{\text{diff}} u, u) \leq C_2 h^{d-2} \|u\|^2_2
\]
\[
| (A_{\text{conv}} u, v) | \leq C_3 h^{d-1} \|u\|_2 \|v\|_2
\]

Therefore we have
\[
\lambda_{\text{min}}(H) \geq C_1 h^d, \quad \lambda_{\text{max}}(H) \leq C_2 h^{d-2}, \quad \rho(S) = \|S\|_2 \leq C_3 h^{d-1}
\]
yielding with (25)
\[
K := \text{cond}_2(H) + \left( \frac{\rho(S)}{\lambda_{\text{min}}(H)} \right)^2 \leq \frac{C_2}{C_1} h^{-2} + \left( \frac{C_3}{C_1} h^{-1} \right)^2 = Ch^{-2}, \quad q = (1 - C^{-1} h^2)^{1/2}
\]

This means that we need \(C'h^{-2}\) steps of the iterative method to reduce the norm of the residual by a fixed factor.

**Note:** \(C_3\) is proportional to \(\|b\|_2\), so we obtain \(K = \left( C + C' \|b\|_2^2 \right) h^{-2}\). So for a problem with strong convection the number of iterations can be very large.

Recall that the stiffness matrix \(A\) is of size \(n \times n\) with \(cn\) nonzero elements where \(n \approx ch^{-d} \approx cN^d\) (for the meshsize \(h \approx 1/N\)).

Therefore the work of a matrix-vector product is given by the number of nonzero matrix elements \(cN^d\).

The work of one step of the 1-step min. res. method is one matrix-vector product, and some inner products, so the work per step is \(c'N^d\).
The number of steps is proportional to \( h^{-2} \approx N^2 \) if we want to achieve a residual with \( \|r\|_2 \leq \delta \). Hence the total work for our iterative method with \( q = 1 - ch^2 \) is

\[ CN^2 N^d \]

If we had an iterative method with \( q = 1 - ch \) we would obtain a total work of \( C N N^d \) instead.

**Summary:** **Work for solving convection-diffusion problem in** \( \Omega \subset \mathbb{R}^d \) **with meshsize** \( h = 1/N \)

The number of unknowns is \( n = O(N^d) \). An ideal algorithm would use \( O(n) = O(N^d) \) operations.

<table>
<thead>
<tr>
<th>( d = 1 )</th>
<th>( d = 2 )</th>
<th>( d = 3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gaussian elimination (using band structure)</td>
<td>( N^1 )</td>
<td>( N^4 )</td>
</tr>
<tr>
<td>Gaussian elimination with reordering</td>
<td>( N^1 )</td>
<td>( N^3 )</td>
</tr>
<tr>
<td>GMRES(1) method: ( q = 1 - ch^2 )</td>
<td>( N^3 )</td>
<td>( N^4 )</td>
</tr>
<tr>
<td>GMRES(1) method: symmetric ( A ): ( q = 1 - ch )</td>
<td>( N^2 )</td>
<td>( N^3 )</td>
</tr>
<tr>
<td>GMRES(1) method: symmetric ( A ) with multigrid preconditioner</td>
<td>( N^1 )</td>
<td>( N^2 )</td>
</tr>
</tbody>
</table>

Note that for \( d = 1 \) using an iterative method is pointless. For \( d = 2 \) the direct solver with reordering is better than the 1-step min. res. method. For \( d = 3 \) the iterative method is clearly better than the direct method.

In the case of a symmetric matrix we can construct an iterative method with \( q = 1 - ch \). This is the **conjugate gradient method** which we will discuss next.

## 8 The Conjugate Gradient Method

### The minimization problem

We are given a **symmetric positive definite matrix** \( A \in \mathbb{R}^{n \times n} \) and a right hand side vector \( b \in \mathbb{R}^n \). We want to solve the **linear system**

\[
\text{Find } u_* \in \mathbb{R}^n \text{ such that } Au_* = b.
\]

Since \( A \) is nonsingular there exists a unique solution. The linear system is equivalent to the minimization problem

\[
\text{Find } u_* \in \mathbb{R}^n \text{ such that } F(u) = \frac{1}{2} u^\top Au - b^\top u \text{ is minimal.}
\]

Note that \( (u,v)_A := (Au,v) = v^\top Au \) defines an inner product on \( \mathbb{R}^n \), and a norm \( \|u\|^2_A = u^\top Au \). Note that

\[
\|u - u_*\|^2_A = (u,u)_A - 2(u_*u)_A + (u_*u)_A = 2F(u) + \|u_*\|^2_A
\]

Hence minimizing \( F(u) \) is equivalent to minimizing \( \|u - u_*\|_A \).

In applications \( F(u) \) corresponds to an energy (internal energy minus work of external forces) which is minimized by the solution. Therefore \( \|u\|_A \) is also called the energy norm, and this norm is the “natural” way to measure errors for this problem.

### Subspace corrections

**Subspace correction with** \( \tilde{V} = \text{span}\{d_0\} \)

We start at \( u_0 \in \mathbb{R}^n \) and pick a search direction \( d_0 \). We define \( u^\text{new} := u_0 + \alpha_0 d_0 \) where \( \alpha_0 \) is chosen such that \( \|u^\text{new} - u_*\|_A \) becomes minimal: The **normal equations** state that \( (u^\text{new} - u_* , d_0)_A = 0 \) or

\[
(d_0,d_0)_A \alpha_0 = (u_* - u_0, d_0)_A = (A(u_* - u_0), d_0)_A = (r_0, d_0)_A, \quad \text{hence } \alpha_0 = \frac{(d_0,r_0)}{(d_0,d_0)_A} \tag{26}
\]

with the residual \( r_0 := b - Au_0 \). In this case we perform a subspace correction with the 1-dimensional subspace \( \tilde{V} = \text{span}\{d_0\} \).
Subspace correction with $\tilde{V} = \text{span}\{d_0, \ldots, d_k\}$

We start at $u_0 \in \mathbb{R}^n$ and pick linearly independent vectors $d_0, \ldots, d_k \in \mathbb{R}^n$. We define

$$u_{\text{new}} := u_0 + \alpha_0 d_0 + \cdots + \alpha_k d_k$$

where $\alpha_0, \ldots, \alpha_k$ are chosen such that $\|u_{\text{new}} - u_*\|_A$ becomes minimal: The normal equations state that $(u_{\text{new}} - u_*, d_j)_A = 0$ for $j = 0, \ldots, k$. Therefore we can find $\alpha_0, \ldots, \alpha_k$ by solving the linear system

$$
\begin{bmatrix}
(d_0, d_0)_A & \cdots & (d_k, d_0)_A \\
\vdots & \ddots & \vdots \\
(d_0, d_k)_A & \cdots & (d_k, d_k)_A
\end{bmatrix}
\begin{bmatrix}
\alpha_0 \\
\vdots \\
\alpha_k
\end{bmatrix} =
\begin{bmatrix}
(r_0, d_0) \\
\vdots \\
(r_0, d_k)
\end{bmatrix}
$$

(27)

Here we used $(u_*, u_*)_A = (A(u_*, u_*)_A = (r_0, d_j)$ on the right hand side. Note that the normal equations state that $(A(u_{\text{new}} - u_*), d_j)_A = 0$ for $j = 0, \ldots, k$, hence the new residual $r_{\text{new}} := b - Au_{\text{new}} = A(u_* - u_{\text{new}})$ satisfies

$$r_{\text{new}} \perp d_0, \ldots, d_k$$

Steepest descent method

The current guess is $u_j$. We choose the direction vector $d_0$ to be the steepest descent direction of the function $F(u)$: The gradient is $\nabla F(u) = Au - b$, so the steepest descent direction is given by the residual $r_j = b - Au_j$. We define $u_{j+1} := u_j + \alpha_j r_j$ where $\alpha_j$ is chosen so that $\|u_{j+1} - u_*\|_A$ becomes minimal:

For $k = 0, 1, 2, \ldots$ do

$$r_k := b - Au_k, \quad \alpha_k := \frac{(r_k, r_k)_A}{(r_k, r_k)_A}, \quad u_{k+1} := u_k + \alpha_k r_k$$

Convergence: The errors satisfy

$$u_{k+1} - u_* = u_k + \alpha_k A(u_* - u_k) - u_* = (I - \alpha_k A)(u_k - u_*)$$

Since $A$ is symmetric we have an orthonormal basis $v_1, \ldots, v_n$ of eigenvectors with $Av_j = \lambda_j v_j$. We write the old error $u_k - u_*$ using this basis

$$u_k - u_* = \sum_{j=1}^n c_j v_j, \quad \|u_k - u_*\|_A^2 = \sum_{j=1}^n |c_j|^2 \lambda_j$$

and get for the new error

$$u_{k+1} - u_* = (I - \alpha A)(u_k - u_*) = \sum_{j=1}^n c_j (1 - \alpha \lambda_j) v_j$$

$$\|u_{k+1} - u_*\|_A^2 = \sum_{j=1}^n |1 - \alpha \lambda_j|^2 |c_j|^2 \lambda_j \leq \left( \max_{j=1, \ldots, n} |1 - \alpha \lambda_j| \right)^2 \|u_k - u_*\|_A^2$$

The value of $\alpha$ which minimizes $\max_{j=1, \ldots, n} |1 - \alpha \lambda_j|$ is $\alpha_* = \frac{2}{\lambda_{\text{min}} + \lambda_{\text{max}}}$ where

$$q_* = \max_{j=1, \ldots, n} |1 - \alpha_* \lambda_j| = \frac{\lambda_{\text{max}} - \lambda_{\text{min}}}{\lambda_{\text{max}} + \lambda_{\text{min}}} = 1 - \frac{2}{\kappa + 1} \quad \text{with } \kappa := \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}} = \text{cond}_2(A)$$

Since $\alpha_k$ is chosen such that $\|u_{k+1} - u_*\|_A$ is minimal we obtain for $u_{k+1} = u_k + \alpha_k r_k$ the bounds

$$\|u_{k+1} - u_*\|_A \leq \left( 1 - \frac{2}{\kappa + 1} \right) \|u_k - u_*\|_A, \quad \|u_{k+1} - u_*\|_A \leq \left( 1 - \frac{2}{\kappa + 1} \right) \|u_0 - u_*\|_A$$

This means that we need $C_{\delta k}$ iterations to achieve $\|u_k - u_*\|_A \leq \delta$. 

20
Conjugate gradient method, version 0

We start with an initial guess $u_0$.

For $k = 0, 1, 2, \ldots$ do

- Let $r_k := b - Au_k$
  - If $r_k = 0$ stop (since $u_k$ is the exact solution)
- Perform a subspace correction with $V_k := \text{span}\{r_0, \ldots, r_k\}$:
  - Solve (27) (with $d_j := r_j$), let $u_{k+1} := u_0 + \alpha_0 r_0 + \cdots + \alpha_k r_k$

The normal equations are in this case

$$
\begin{align*}
(\alpha_{k+1} - u_*) & = 0 \\
(r_{k+1}, r_j)
\end{align*}
$$

(28)

i.e., we have $r_{k+1} \perp V_k$. As long as we have $r_k \neq 0$ the vectors $r_0, \ldots, r_k$ are therefore linearly independent.

Observation 1: We will have $r_k = 0$, i.e., $u_k = u_*$ for $k \geq K$ with some $K \leq n$.

But the CG method is typically used as an iterative method with $k \ll n$ iterations to find an approximate solution $u_k$, rather than the exact solution.

Observation 2: We have

$$V_k = \text{span}\{r_0, Ar_0, \ldots, A^k r_0\}
$$

(29)

Proof: It is obvious for $k = 0$. Assume it holds for $k - 1$. We have

$$r_k = b - Au_k = b - Au_0 - A (\alpha_0 r_0 + \cdots + \alpha_{k-1} r_{k-1})
$$

$$\in \text{span}\{r_0, \ldots, A^{k-1} r_0\}
$$

hence $r_k \in \text{span}\{r_0, Ar_0, \ldots, A^k r_0\}$.

The subspace $V_k = \text{span}\{r_0, Ar_0, \ldots, A^k r_0\}$ is called a Krylov space, and it plays a central role in understanding CG, GMRES and related iterative methods.

Conjugate gradient method, version 1

We can make the method more efficient by orthogonalizing the vectors $r_0, r_1, \ldots, r_k$ with respect to $(\cdot, \cdot)_A$ using the Gram-Schmidt method, yielding vectors $d_0, d_1, \ldots, d_k$ such that

$$V_k = \text{span}\{r_0, \ldots, r_k\} = \text{span}\{d_0, \ldots, d_k\}
$$

and $(d_j, d_k)_A = 0$ for $j \neq k$. We let $u_{k+1} = u_0 + \alpha_0 d_0 + \cdots + \alpha_k d_k$ where $\alpha_0, \ldots, \alpha_k$ are chosen such that $\|u_j - u_*\|_A = \|\alpha_0 d_0 + \cdots + \alpha_k d_k - (u_* - u_0)\|_A$ is minimal. Since the directions $d_0, \ldots, d_k$ are $A$-orthogonal (a.k.a. “conjugate”) the normal equations (27) decouple:

$$(d_j, d_j)_A \alpha_j = (r_0, d_j) \quad \text{for } j = 0, \ldots, k
$$

We can write the right hand side as

$$(u_* - u_0, d_j)_A = \left( u_* - \underbrace{(u_0 + \alpha_0 d_0 + \cdots + \alpha_{j-1} d_{j-1})}_{\bot_A d_j} \right)_A = (r_j, d_j)
$$

yielding $\alpha_j = \frac{(r_j, d_j)}{(d_j, d_j)_A}$ and $u_{k+1} = u_0 + \alpha_0 d_0 + \cdots + \alpha_{k-1} d_{k-1} + \alpha_k d_k = u_k + \alpha_k d_k$.

Therefore the algorithm can be written as follows:
For $k = 0, 1, 2, \ldots$ do:

- the new steepest descent direction is given by the residual:
  
  \[
  r_k := b - Au_k, \quad \text{if } r_k = 0: \text{ stop (since } u_k \text{ is exact solution)}
  \]

- modify this to make it conjugate to all previous search directions $d_{k-1}, d_{k-2}, \ldots, d_0$:
  
  \[
  d_k := r_k - \frac{(r_k, d_{k-1})_A}{(d_{k-1}, d_{k-1})_A} d_{k-1} - \frac{(r_k, d_{k-2})_A}{(d_{k-2}, d_{k-2})_A} d_{k-2} - \cdots - \frac{(r_k, d_0)_A}{(d_0, d_0)_A} d_0
  \]  
  \[
  (30)
  \]

- perform optimal step in direction $d_k$: (actually minimizes $\|(u_0 + \alpha_0 d_0 + \cdots + \alpha_k d_k) - u^*\|_A$ over all $\alpha_0, \ldots, \alpha_k$)

  \[
  \alpha_k := \frac{(r_k, d_k)}{(d_k, d_k)_A}, \quad u_{k+1} := u_k + \alpha_k d_k
  \]

  \[
  (31)
  \]

There is one additional simplification: By (29) we have $Ad_{k-2} \in V_{k-1}$, and by the normal equations $r_k \perp V_{k-1}$:

\[
(r_k, d_{k-2})_A = (r_k, Ad_{k-2}) = 0
\]

By this argument all the red terms in (30) are zero and we have

\[
\frac{d_k}{d_{k-1}} := r_k - \frac{(r_k, d_{k-1})_A}{(d_{k-1}, d_{k-1})_A} d_{k-1}
\]

\[
(32)
\]

where we **orthogonalize only with respect to the previous direction** $d_{k-1}$.

### Final version of the conjugate gradient method

By (32) we have $(d_k, r_k) = (r_k, d_k)$ as $(r_k, d_{k-1}) = 0$ by the normal equations $r_k \perp V_{k-1}$. Hence

\[
\alpha_k = \frac{(r_k, r_k)}{(d_k, d_k)_A}
\]

\[
(33)
\]

We have from (31) that $\alpha_k Ad_k = r_k - r_{k+1}$ and hence

\[
\alpha_k r_{k+1}^T Ad_k = r_{k+1}^T (r_k - r_{k+1}) = -r_{k+1}^T r_{k+1}
\]

since $(r_k, r_{k+1}) = 0$ by the normal equations $r_{k+1} \perp V_k$. Using this in the numerator and (33) in the denominator we get

\[
\frac{-r_{k+1}^T r_{k+1}}{r_k^T r_k} = \frac{\alpha_k r_{k+1}^T Ad_k}{\alpha_k d_k^T Ad_k} = \frac{(r_{k+1}, d_k)_A}{(d_k, d_k)_A}
\]

so that we can write (32) for $k+1$ as

\[
\beta_k := \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}, \quad d_{k+1} := r_{k+1} + \beta_k d_k
\]

We then have the following algorithm:

For $k = 0, 1, 2, \ldots$ do:

\[
\alpha_k := \frac{r_k^T r_k}{d_k^T (Ad_k)}, \quad u_{k+1} := u_k + \alpha_k d_k, \quad r_{k+1} := r_k - \alpha_k (Ad_k), \quad \text{if } r_k = 0: \text{ stop}
\]

\[
\beta_k := \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}, \quad d_{k+1} := r_{k+1} + \beta_k d_k
\]

### The cost of each step:
Proof of the error estimate for the Conjugate Gradient Method

We have \( u_k = u_0 + w \) where \( w \in V_{k-1} = \text{span} \{ r_0, Ar_0, \ldots, A^{k-1}r_0 \} \) is chosen such that \( \| u_k - u_* \|_A \) is minimal. Therefore

\[
    u_k - u_* = u_0 - u_\ast + \sum_{j=0}^{k-1} \beta_j A^j (u_* - u_0) = p(A)(u_* - u_0) \quad \text{with } p(\lambda) = 1 + \beta_0 \lambda + \cdots + \beta_k \lambda^k
\]

Using the eigenvectors \( v_1, \ldots, v_n \) of \( A \) we can write the initial error as \( u_0 - u_* = \sum_{j=1}^n c_j v_j \). Then

\[
    \| u_k - u_* \|_A^2 = \sum_{j=1}^k p(\lambda_j)^2 |c_j|^2 \leq \left( \max_{j=1,\ldots,n} |p(\lambda_j)| \right)^2 \| u_0 - u_* \|_A^2
\]

We now try to choose a polynomial \( p(\lambda) = 1 + \beta_0 \lambda + \cdots + \beta_k \lambda^{k+1} \) which makes \( q := \max_{j=1,\ldots,n} |p(\lambda_j)| \) small. We want a polynomial \( p \in \mathcal{P}_k \) with

\[
    p(0) = 1, \quad q := \max_{\lambda \in [\lambda_1,\lambda_n]} |p(\lambda)| \text{ is small.}
\]

We can actually determine the polynomial \( p \) which minimizes \( q \). We start with the Chebyshev polynomial \( T_k(x) \) which has \( \max_{x \in [-1,1]} |T_k(x)| = 1 \). We then use a linear change of variables \( x \in [-1,1] \) to \( \lambda \in [\lambda_1,\lambda_n] \)

\[
    \lambda = \frac{\lambda_1 + \lambda_n}{2} + x \frac{\lambda_n - \lambda_1}{2}, \quad x = \frac{2\lambda - \lambda_1 - \lambda_n}{\lambda_n - \lambda_1} =: g(\lambda)
\]

\[
    \tilde{p}(\lambda) := T_k(g(\lambda)), \quad p(\lambda) := \frac{\tilde{p}(\lambda)}{\tilde{p}(0)}
\]

\[
    q = \max_{\lambda \in [\lambda_1,\lambda_n]} |p(\lambda)| = \frac{1}{|\tilde{p}(0)|}, \quad \tilde{p}(0) = T_k \left( \frac{-\lambda_n + \lambda_1}{\lambda_n - \lambda_1} \right)
\]

Note that \( x := \frac{\lambda_n + \lambda_1}{\lambda_n - \lambda_1} = \frac{k+1}{k-1} > 1 \). We need a lower bound \( |T_k(-x)| = |T_k(x)| \geq \cdots \). Note that with \( x = \cos t = \frac{1}{2}(z + z^{-1}) \)

\[
    T_k(x) = \cos (kt) = \frac{1}{2} \left( z^k + z^{-k} \right)
\]

Note that for \( z := (\kappa^{1/2} + 1) / (\kappa^{1/2} - 1) \) we have

\[
    \frac{1}{2} \left( z + z^{-1} \right) = \frac{1}{2} \left( \frac{(\kappa^{1/2} + 1)^2 + (\kappa^{1/2} - 1)^2}{(\kappa^{1/2} + 1)(\kappa^{1/2} - 1)} \right) = \frac{\kappa + 1}{\kappa - 1}
\]

hence with \( \rho := z^{-1} = 1 - \frac{2}{\kappa^{1/2} + 1} \)

\[
    T_k(x) = \frac{1}{2} \left( z^k + z^{-k} \right) = \frac{1}{2} \left( \rho^{-k} + \rho^k \right)
\]
yielding
\[ q = |T_k(x)|^{-1} = \frac{2}{\rho^{-k} + \rho^k} \leq 2\rho^k = 2 \left(1 - \frac{2}{\kappa^{1/2} + 1}\right)^k \]
\[ \|u_k - u^*\|_A \leq 2 \left(1 - \frac{2}{\kappa^{1/2} + 1}\right)^k \|u_0 - u^*\|_A \]

Note that for \( k = 1 \) we have \( q = \frac{2}{\rho^{-1} + \rho} = \frac{2}{z + z^{-1}} = \frac{\kappa - 1}{\kappa + 1} = 1 - \frac{2}{\kappa + 1} \) which is the bound we obtained for the steepest descent method.

**Preconditioned steepest descent method**

Let us review the steepest descent method in a general setting: We have a Hilbert space \( V \) with inner product \((u, v)_V \) and norm \( \|u\|_V = (u, u)^{1/2}_V \). We are given a symmetric bilinear form \( a(u, v) \) with
\[ \forall u, v \in V: \quad a(u, u) \geq \gamma \|u\|_V^2, \quad |a(u, v)| \leq L \|u\|_V \|v\|_V \]
and a linear functional \( \ell \in V' \). Let \( f(u) := \frac{1}{2}a(u, u) - \ell(u) \). We want to find \( u \in V \) such that \( f(u) \) is minimal. This is equivalent to (VAR): find \( u \in V \) such that
\[ \forall v \in V: \quad a(u, v) = \ell(v) \]

For the current approximation \( u_k \) consider the residual \( \ell - a(u_k, \cdot) \in V' \). By the Riesz representation theorem there is a unique \( \tilde{r}_k \in V \) such that
\[ \forall v \in V: \quad (\tilde{r}_k, v)_V = \ell(v) - a(u_k, v) \]

With \( \alpha_k > 0 \) we define the next iteration as
\[ u_{k+1} := u_k + \alpha_k \tilde{r}_k \]

**Method 1:** We can use any fixed \( \alpha \in (0, 2L^{-1}) \). For \( \alpha = \frac{2}{\gamma + L} \) we obtain \( \|u_{k+1} - u^*\|_V \leq q \|u_k - v\|_V \) and
\[ \|u_{k+1} - u^*\|_A \leq q \|u_k - v\|_A \]
with \( q = \left(1 - \frac{2}{\kappa + 1}\right)^{1/2} \).

**Method 2:** We choose \( \alpha_k \) such that \( f(u_{k+1}) \) is minimal. This is equivalent to
\[ a(u_k + \alpha \tilde{r}_k, \tilde{r}_k) = \ell(\tilde{r}_k), \quad \text{i.e.,} \quad \alpha \cdot a(\tilde{r}_k, \tilde{r}_k) = \ell(\tilde{r}_k) - a(u_k, \tilde{r}_k) = (\tilde{r}_k, \tilde{r}_k)_V \]

hence
\[ \alpha_k = \frac{(\tilde{r}_k, \tilde{r}_k)_V}{a(\tilde{r}_k, \tilde{r}_k)} \]

Minimizing \( f(u) \) is equivalent to minimizing \( \|u - u^*\|_A \). Hence we obtain the same bound as for \( \alpha = \frac{2}{\gamma + L} \) and have \( \|u_{k+1} - u^*\|_A \leq q \|u_k - v\|_A \) with \( q = \left(1 - \frac{2}{\kappa + 1}\right)^{1/2} \).

Now we consider the case \( V = \mathbb{R}^n \), let \( (u, v) := \sum_{j=1}^n u_j v_j \). Let \( b \in \mathbb{R}^n \) and \( A \in \mathbb{R}^{n \times n} \) such that its symmetric part is positive definite. Then we define
\[ \ell(v) := (b, v), \quad a(u, v) := (Au, v) \]

Then (VAR) is the linear system \( Au = b \).

Let \( B \in \mathbb{R}^{n \times n} \) be symmetric positive definite, then \( (u, v)_V := (Bu, v) \) is an inner product for \( V \). We can therefore use the above results with this inner product \((u, v)_V\).

The residual \( \tilde{r}_k \) satisfies \( \ell(\tilde{r}_k, v)_V = \ell(v) - a(u_k, v) \):
\[ (B\tilde{r}_k, v) = (b, v) - (Au_k, v), \quad \text{hence} \quad \tilde{r}_k = B^{-1}r_k \quad \text{with} \quad r_k := b - Au_k \]

We have
\[ \alpha_k = \frac{(\tilde{r}_k, \tilde{r}_k)_V}{a(\tilde{r}_k, \tilde{r}_k)} = \frac{(r_k, \tilde{r}_k)}{(A\tilde{r}_k, \tilde{r}_k)} \]
Preconditioned conjugate gradient method

Note that our algorithm uses two inner products on $\mathbb{R}^n$: $(u, v)_A := v^T A u$ and $(u, v)_B := v^T B u$ with the matrix $B := I$. For a given $\tilde{b} \in \mathbb{R}^n$ we want to find $u_* \in \mathbb{R}^n$ such that $F(u) := \frac{1}{2} (u, u)_A - (\tilde{b}, u)_B$ is minimal. This is equivalent to finding $u_* \in \mathbb{R}^n$ such that

$$\forall v \in \mathbb{R}^n : \quad (u_*, v)_A = (\tilde{b}, v)_B$$  \hspace{1cm} (34)

For a given approximation $u$ we can determine the residual $\tilde{r}$ satisfying

$$\forall v \in \mathbb{R}^n : \quad (\tilde{r}, v)_B = (\tilde{b}, v)_B - (u, v)_A$$  \hspace{1cm} (35)

So far we considered $B = I$. But we can generalize the conjugate gradient method for an arbitrary symmetric positive definite matrix $B$. Note that $(u, u)_A$ can be bounded from above and below by $(u, u)_B$: There exist $\lambda_{\min}, \lambda_{\max}$ such that

$$\lambda_{\min} (u, u)_B \leq (u, u)_A \leq \lambda_{\max} (u, u)_B$$

and the ratio

$$\tilde{\kappa} := \frac{\lambda_{\max}}{\lambda_{\min}}$$

measures how close $(u, u)_A$ and $(u, u)_B$ are to each other. Our error estimate gives for this generalized CG method

$$\|u_k - u_*\|_A \leq 2 \left( 1 - \frac{2}{\tilde{\kappa}^{1/2} + 1} \right)^k \|u_0 - u_*\|_A$$

This means that we need $C \tilde{\kappa}^{1/2}$ iterations to achieve $\|u_k - u_*\|_A \leq \delta$.

For many problems the original CG method performs poorly because $(u, u)_A$ and $(u, u)_B$ are “far apart”, i.e., the condition number $\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$ is large. The idea of the preconditioned conjugate gradient method (PCG) is to use a symmetric positive definite matrix $B$ such that

- The bilinear form $(\cdot, \cdot)_B$ is “close” to the bilinear form $(\cdot, \cdot)_A$:

$$\tilde{\lambda}_{\min} (u, u)_B \leq (u, u)_A \leq \tilde{\lambda}_{\max} (u, u)_B \quad \text{with} \quad \tilde{\kappa} := \frac{\tilde{\lambda}_{\max}}{\tilde{\lambda}_{\min}} \ll \kappa$$

- It is “easy” to find $B^{-1} v$, i.e., to solve a linear system $Bu = v$

Note that (34), (35) give

$$Au_* = B\tilde{b} \quad \text{i.e.,} \quad \tilde{b} = B^{-1} b$$

$$B\tilde{r} = B\tilde{b} - Au \quad \text{i.e.,} \quad \tilde{r} = B^{-1} (b - Au)$$

So we must be able to compute $B^{-1} v$ in order to implement the PCG. Let $r_k := b - Ax_k$. We then have $(x_k, v)_A - (\tilde{b}, v)_B = (\tilde{r}, v)_B$ with $\tilde{r}_k := B^{-1} r_k$, and $(\tilde{r}_k, \tilde{r}_k)_B = (B\tilde{r}_k, \tilde{r}_k) = (r_k, r_k)$. Therefore we obtain the algorithm for the preconditioned conjugate gradient method:

$$r_0 := (b - Ax_0), \quad r_0 := B^{-1} r_0, \quad d_0 := \tilde{r}_0.$$

For $k = 0, 1, 2, \ldots$ do

$$\alpha_k := \frac{r_k^T \tilde{r}_k}{d_k^T (Ad_k)}, \quad u_{k+1} := u_k + \alpha_k d_k, \quad r_{k+1} := r_k - \alpha_k (Ad_k), \quad \tilde{r}_{k+1} := B^{-1} r_{k+1}$$

$$\beta_k := \frac{r_k^T \tilde{r}_{k+1}}{r_k^T \tilde{r}_k}, \quad d_{k+1} := \tilde{r}_{k+1} + \beta_k d_k$$

The cost of each step:

- 1 matrix vector product: compute $Ad_k$
- 2 dot products: compute $d_k^T (Ad_k), r_k^T \tilde{r}_{k+1}$
- 1 preconditioning step: compute $\tilde{r}_{k+1} := B^{-1} r_{k+1}$, i.e., solve $B\tilde{r}_{k+1} = r_{k+1}$

Error estimate:

$$\|u_k - u_*\|_A \leq 2 \left( 1 - \frac{2}{\tilde{\kappa}^{1/2} + 1} \right)^k \|u_0 - u_*\|_A$$
Choosing a good preconditioner

Choosing a good preconditioner $B$ can make a huge difference and enable us to solve much larger problems. The preconditioner matrix $B$ should somehow “mimic” the matrix $A$, but $Bv = f$ should be much cheaper to solve than $Av = f$.

There is no universal recipe for a good preconditioner. This depends on the original problem which the matrix $A$ describes (e.g., elliptic PDE). We give three examples for choosing a preconditioner:

1. **Incomplete Cholesky decomposition**: For a sparse matrix $A$ we can compute the Cholesky decomposition $A = C^T C$: use $C = \text{chol}(A)$ in Matlab. The matrix $C$ has typically many more nonzeros than the matrix $A$ (so-called “fill-in”), and this makes the direct solver expensive. We can “cheat” in the Cholesky algorithm and only allow a limited amount of fill-in: this is the incomplete Cholesky decomposition. It gives an upper triangular matrix $\tilde{C}$, but $\tilde{C}^T \tilde{C}$ is not equal to $A$. In Matlab use $C = \text{i chol}(A)$ (there are options to specify how much fill-in to allow).

   Then use $B = \tilde{C}^T \tilde{C}$: We can solve $Bv = f$ as follows: solve $\tilde{C}^T y = f$ by forward substitution, then solve $\tilde{C} v = y$ by back substitution.

   For the diffusion problem we get instead of $\kappa = c h^{-2}$ now $\tilde{\kappa} = \tilde{c} h^{-2}$ with $\tilde{c} < c$. So this can reduce the number of iterations, but the total cost still grows with the same power of $N$.

2. **Use “similar problem” for which there is a fast algorithm**

   Example: Solve elliptic PDE with nonconstant coefficient $c(x)$ on square $\Omega = (0, 1)^2$

   \[-\text{div}(c(x) \nabla u) = f(x)\]

   using a uniform mesh.

   For $c(x) = 1$ we can use FFT to solve this problem: In 1D use discrete sine transform. In 2D, 3D apply this for each direction. Since the discrete problem is translation invariant for a uniform mesh: for the new basis the matrix becomes diagonal. Hence the work is $C n \log n$.

3. **Multigrid**: Consider the diffusion problem in $\mathbb{R}^d$. Consider as an example a polygonal domain $\Omega \subset \mathbb{R}^2$.

   We obtain our mesh by successive “uniform mesh refinement”: We start with a coarse mesh corresponding to a space $V_0 \subset H^1_0(\Omega)$ with triangles of size~$h_0$. Then we bisect each edge and divide each triangle into 4 subtriangles. This defines a space $V_1$. Another refinement step (divide each triangle into 4 subtriangles) gives a space $V_2$. After $L$ refinement steps we obtain a space $V_L$ with triangles of size $h_L \sim 2^{-L} h_0$.

   This gives the nested spaces

   \[ V_0 \subset V_1 \subset \cdots \subset V_L, \quad N_\ell := \dim V_\ell \]

   We use nodal basis for $V_\ell$: a function $v_\ell \in V_\ell$ corresponds to a vector $\vec{v} \in \mathbb{R}^{N_\ell}$ containing the nodal values of $v_\ell$. Since $V_\ell \subset V_{\ell+1}$ we can evaluate $v_\ell$ at the nodes of the next finer mesh and obtain a vector $\vec{w} \in \mathbb{R}^{N_{\ell+1}}$. A node $j$ on the finer mesh is either a node $k$ on the coarser mesh, or a midpoint of an edge with nodes $k_1, k_2$ on the coarse mesh. Hence we have

   \[ w_j = \begin{cases} v_k & \text{if node } j \text{ on the fine mesh is node } k \text{ on the coarse mesh} \\ \frac{1}{2}(v_{k_1} + v_{k_2}) & \text{if node } j \text{ on the fine mesh is the midpoint of nodes } k_1, k_2 \text{ on the coarse mesh} \end{cases} \]

   This mapping corresponds to a matrix $J_\ell$ where row $j$ contains either only a 1 in column $k$, or two entries $\frac{1}{2}$ in columns $k_1, k_2$:

   \[ \vec{w} = J_\ell \vec{v}, \quad J_\ell \in \mathbb{R}^{N_{\ell+1} \times N_\ell} \]

   Now we define the preconditioning operator $M_L = B^{-1}$ on the fine mesh as follows: **“BPX preconditioner”**

   \[ M_\ell \vec{f} := \begin{cases} I & \text{if } \ell = 0 \\ 2^{(d-2)\ell} I + J_{\ell-1}M_{\ell-1} J_{\ell-1}^T & \text{otherwise} \end{cases} \quad (36) \]

   One can prove [Bramble, Pasciak, Xu 1990]
Method 2: \[
\| \alpha \|_{H^1(\Omega)}^2 \leq (B\tilde{V}, \tilde{V}) \leq C_2 \| v_k \|_{H^1(\Omega)}^2
\]
with positive constants \(C_1, C_2\) independent of \(h\).

This means that BPX provides optimal preconditioning for any variational problem in \(H^1(\Omega)\). Note that the BPX preconditioner does not depend on the coefficients \(c(x), b(x)\) of the bilinear form
\[
a(u, v) = \int_{\Omega} [c(x) \nabla u \cdot \nabla u + (b(x) \cdot \nabla u) v] \, dx
\]
Note that there are other preconditioners (e.g., multigrid V-cycle) which use the specific form of \(a(u, v)\). These preconditioners will usually give a better performance.

9 Preconditioned GMRES(1) algorithm

Recall the variational problem (VAR): We have a space \(V\) with inner product \((u, v)_V\) and norm \(\| u \|_V = (u, u)_V^{1/2}\). We are given a bilinear form \(a(u, v)\) with
\[
\forall u, v \in V: \quad a(u, u) \geq \gamma \| u \|_V^2, \quad |a(u, v)| \leq L \| u \|_V \| v \|_V
\]
and a linear functional \(\ell \in V'\). We want to find \(u \in V\) such that
\[
\forall v \in V: \quad a(u, v) = \ell(v)
\]
Define \(\tilde{b} \in V\) such that \((\tilde{b}, \cdot)_V = \ell\). Define \(\tilde{A}: V \to V\) such that \((\tilde{A}u, \cdot)_V = a(u, \cdot)\). Then (VAR) can be written as \(\tilde{A}u = \tilde{b}\).

The Lax-Milgram theorem states that this problem has a unique solution \(u\) which can be approximated by Richardson iteration: For the current approximation \(u_k\) consider the residual \(\ell - a(u_k, \cdot) \in V'\). By the Riesz representation theorem there is a unique \(\tilde{r}_k \in V\) such that
\[
\forall v \in V: \quad (\tilde{r}_k, v)_V = \ell(v) - a(u_k, v)
\]
With \(\alpha_k > 0\) we define the next iteration as
\[
u_{k+1} := u_k + \alpha_k \tilde{r}_k
\]

Method 1: We can use any fixed \(\alpha \in (0, 2\gamma/L^2)\). For \(\alpha = \gamma/L^2\) we obtain \(\| u_{k+1} - u \|_V \leq q \| u_k - v \|_V\), \(\| \tilde{r}_{k+1} \|_V \leq q \| \tilde{r}_k \|_V\) with \(q = (1 - \kappa^2)^{1/2}\).

Method 2: We choose \(\alpha_k\) such that \(\| \tilde{r}_{k+1} \|_V\) is minimal. We have \(\tilde{r}_{k+1} = \tilde{r}_k - \alpha \tilde{A} \tilde{r}_k\):
\[
(\tilde{r}_{k+1}, v)_V = \ell(v) - a(u_k + \alpha \tilde{r}_k, v)
\]
\[
= (\tilde{b} - \tilde{A}(u_k + \alpha \tilde{r}_k), v)_V = (\tilde{r}_k - \alpha \tilde{A} \tilde{r}_k, v)_V
\]
\[
\| \tilde{r}_{k+1} \|_V^2 = \| \tilde{r}_k \|_V^2 - 2\alpha (\tilde{A} \tilde{r}_k, \tilde{r}_k)_V + \alpha^2 \| \tilde{A} \tilde{r}_k \|_V^2
\]
Hence
\[
\alpha_k = \frac{(\tilde{A} \tilde{r}_k, \tilde{r}_k)_V}{\| \tilde{A} \tilde{r}_k \|_V^2}
\]
Now we consider the case \(V = \mathbb{R}^n\), let \((u, v) := \sum_{j=1}^n u_j v_j\). Let \(b \in \mathbb{R}^n\) and \(A \in \mathbb{R}^{n \times n}\) such that its symmetric part is positive definite. Then we define
\[
\ell(v) := (b, v), \quad a(u, v) := (Au, v)
\]
Then (VAR) is the linear system \( Au = b \).

Let \( B \in \mathbb{R}^{n \times n} \) by symmetric positive definite, then \((u,v)_V := (Bu,v)\) is an inner product for \( V \). We can therefore use the above results with this inner product \((u,v)_V\). We have \((\tilde{b},v)_V = (b,v)\) and \((\tilde{A}u,v)_V = (Au,v)\), hence \( \tilde{b} = B^{-1}b \) and \( \tilde{A} = B^{-1}A \).

The residual \( \tilde{r}_k \) satisfies \((\tilde{r}_k,v)_V = \ell(v) - a(u_k,v)\):
\[
(B\tilde{r}_k,v) = (b,v) - (Au_k,v), \quad \text{hence} \quad \tilde{r}_k = B^{-1}(b-Au_k)
\]
This gives
\[
\alpha_k = \left( \frac{(\tilde{A}\tilde{r}_k,\tilde{r}_k)}{(\tilde{A}r_k,\tilde{A}r_k)} \right)_V = \frac{(BB^{-1}A\tilde{r}_k,\tilde{r}_k)}{(BB^{-1}A\tilde{r}_k,B^{-1}A\tilde{r}_k)} = \frac{(A\tilde{r}_k,\tilde{r}_k)}{(A\tilde{r}_k,B^{-1}A\tilde{r}_k)}
\]
The algorithm requires one matrix-vector product \( Av \) and one preconditioning operation \( B^{-1}v \) per step:

**Preconditioned GMRES(1) algorithm:**

Start with an initial guess \( u_0 \) and compute
\[
r_0 := b - Au_0, \quad \tilde{r}_0 := B^{-1}r_0
\]
For \( k = 0,1,2,\ldots \)

\[
\text{compute} \ A\tilde{r}_k, \quad \text{compute} \ B^{-1}A\tilde{r}_k \quad \alpha_k := \frac{(A\tilde{r}_k,\tilde{r}_k)}{(A\tilde{r}_k,B^{-1}A\tilde{r}_k)} \\
\quad u_{k+1} := u_k + \alpha_k \tilde{r}_k, \quad \tilde{r}_{k+1} := \tilde{r}_k - \alpha_k B^{-1}A\tilde{r}_k
\]
We can also use \( r_{k+1} := r_k - \alpha_k A\tilde{r}_k \) to obtain the actual residuals for each iteration. We can terminate the iteration if \( \|r_{k+1}\| \leq \text{tolerance} \).

10 Optimization

10.1 Introduction: Local minima, convexity

Consider a function \( f : D \to \mathbb{R} \) on a domain \( D \subset \mathbb{R}^n \). We want to find a local minimum \( x_\star \in D \).

We assume that subroutines to evaluate \( f \) and the gradient \( F =: \nabla f \) are available. We want to find a good approximation for \( x_\star \) using a small number of function evaluations of \( f \) and \( F \).

In general there may be multiple local minima. There may also be other critical points with \( F(x) = 0 \) which are not local minima, e.g., local maxima, saddle points.

For a local minimum \( x_\star \) we usually have that \( f \) is convex in a neighborhood of \( x_\star \).

A function \( f \) is **convex** on a convex set \( D \) iff for \( x,y \in D \)
\[
f((1-t)x + ty) \leq (1-t)f(x) + tf(y) \quad \text{for} \ 0 < t < 1
\]
The function is **strictly convex** if this holds with “\(<\)" instead of “\(\leq\)". If \( f \) is strictly convex in \( D \) there is at most one local minimum in \( D \).

Assume that \( f \) is convex with gradient \( F =: \nabla f \). Then \( f \geq \text{any tangent plane}:
\[
f(y) \geq f(x) + (F(x),y-x) \\
f(x) \geq f(y) + (F(y),x-y)
\]
We can relate the error in the function value to the error in the argument:

\[(F(y) - F(x), y - x) \geq 0\]

Assume that \(F\) is Lipschitz, i.e., \(\|F(y) - F(x)\| \leq L \|y - x\|\) for all \(x, y \in D\). Intuitively this gives an upper bound \(L\) for “curvature”: for \(g(t) := f(x_0 + tv)\) with \(\|v\| = 1\) we always have \(g''(t) \leq L\).

Then one obtains with a similar argument the “co-coercivity” property

\[(F(y) - F(x), y - x) \geq L^{-1} \|F(y) - F(x)\|^2\]

It will be useful to have a lower bound \(\gamma > 0\) for “curvature”: This corresponds to the coercivity condition

\[(F(y) - F(x), y - x) \geq \gamma \|y - x\|^2\]

In this case \(f\) is called strongly convex. Intuitively, for \(g(t) := f(x_0 + tv)\) with \(\|v\| = 1\) we always have \(g''(t) \geq \gamma\). Another way to express this: the function \(f(x) - \frac{\gamma}{2}(x, x)\) is convex.

10.2 Steepest descent method for strongly convex functions

We want to find the minimum of a function \(f\) defined on \(\mathbb{R}^n\). We will assume the function \(f\) is “strongly convex”, and the gradient \(F := \nabla f\) is Lipschitz: there exist \(L, \gamma > 0\) such that for all \(x, y \in \mathbb{R}^n\)

\[\|F(y) - F(x)\| \leq L \|y - x\| \quad (37)\]

\[(F(y) - F(x), y - x) \geq \gamma \|y - x\|^2 \quad (38)\]

By Theorem 2.1 the equation \(F(x_*) = 0\) has unique solution \(x_*\). We claim that \(x_*\) is the unique minimum point in \(\mathbb{R}^n\): For \(v \in \mathbb{R}^n\) let \(g(t) := f(x_* + tv)\), then

\[f(x_* + v) - f(x_*) = g(1) - g(0) = \int_0^1 g'(t) dt = \int_0^1 (F(x_* + tv), v) dt\]

\[= \int_0^1 t^{-1} (F(x_* + tv) - F(x_*), tv) dt \geq \gamma \int_0^1 t^{-1} t^2 \|v\|^2 dt = \frac{\gamma}{2} \|v\|^2\]

We can relate the error in the function value to the error in \(x\):

**Lemma 10.1.** Assume \(F := \nabla f\) satisfies (37), (38) with \(\gamma \geq 0\). If \(F(x_*) = 0\) we have for \(x \in \mathbb{R}^n\)

\[\frac{\gamma}{2} \|x - x_*\|^2 \leq f(x) - f(x_*) \leq \frac{L}{2} \|x - x_*\|^2 \quad (39)\]

If \(F(x)\) is known we obtain the improved lower bound

\[\frac{1}{2(L - \gamma)} \left[\|F(x)\|^2 + \gamma L \|x - x_*\|^2 - 2\gamma (F(x), x - x_*)\right] \leq f(x) - f(x_*) \quad (40)\]

**Proof.** Consider \(g(t) := f(x_* + tv)\) then \(g'(t) = (F(x_* + tv), v)\). Hence

\[f(x_* + v) - f(x_*) = g(1) - g(0) = \int_0^1 g'(t) dt = \int_0^1 (F(x_* + tv) - F(x_*), v) dt\]

Since

\[\gamma t^2 \|v\|^2 \leq (F(x_* + tv) - F(x_*), tv) \leq L t^2 \|v\|^2 \quad (41)\]

we obtain

\[\frac{\gamma}{2} \|v\|^2 \leq \int_0^1 \gamma t \|v\|^2 dt \leq f(x) - f(x_*) \leq \int_0^1 L t \|v\|^2 dt = \frac{L}{2} \|v\|^2\]

and let \(v := x - x_*\).
For the second statement we first consider the case $\gamma = 0$: assume that $f$ is convex and $F = \nabla f$ is Lipschitz. Let $F_0 := F(x)$ and consider $f$ on the line in the steepest descent direction: for $t \geq 0$

$$g(t) := f(x - tF_0), \quad g'(t) = -(F(x - tF_0), F_0)$$

Note that

$$(F(x - tF_0), F_0) = (F_0, F_0) + (F(x - tF_0) - F_0, F_0) \geq \|F_0\|^2 - \|F(x - tF_0) - F(x)\| \cdot \|F_0\| \geq (1 - Lt) \|F_0\|^2 \leq Lt \|F_0\|$$

$$g(s) - g(0) = \int_0^s g'(t) dt \leq \int_0^s (Lt - 1) \|F_0\|^2 dt$$

$$g(L^{-1}) - g(0) \leq \left[ \frac{L}{2} t^2 - t \right]_0^{L^{-1}} \|F_0\|^2 = - \frac{1}{2L} \|F_0\|^2$$

This gives (40) for $\gamma = 0$:

$$f(x) - f(x_*) \geq f(x) - f(x - L^{-1}F_0) \geq \frac{1}{2L} \|F(x)\|^2 \tag{42}$$

Now we consider the case $\gamma > 0$: If $f$ satisfies (38), then $\tilde{f}(x) := f(x) - \frac{\gamma}{2} \|x - x_*\|^2$ is convex, and $\tilde{F}(x) := \nabla \tilde{f}(x) = F(x) - \gamma(x - x_*)$ is Lipschitz with $\tilde{L} := L - \gamma$. Using (42) gives

$$\tilde{f}(x) - \tilde{f}(x_*) \geq \frac{1}{2\tilde{L}} \|\tilde{F}(x)\|^2 = \frac{1}{2(L - \gamma)} \|F(x) - \gamma(x - x_*)\|^2$$

$$f(x) - \frac{\gamma}{2} \|x - x_*\|^2 - f(x_*) \geq \frac{1}{2(L - \gamma)} \left[ \|F(x)\|^2 - 2\gamma(F(x), x - x_*) + \gamma^2 \|x - x_*\|^2 \right]$$

which is (40).

\[ \square \]

**Steepest Descent Algorithm**

We want to approximate $x_*$ with an iterative method. We assume that we have subroutines to evaluate $f$ and the gradient $F$. We start with an initial guess $x_0$.

Let $x_k$ denote the current guess. We choose as “search direction” the steepest descent direction $-\nabla f(x_k)$:

$$v := -F(x_k) \tag{43}$$

which is actually the residual $r_k = 0 - F(x_k)$. If $F(x_k) = 0$ we stop since we found the unique minimum of $f$. Otherwise we have a descent direction, since the function $g(t) := f(x_k + tv)$ satisfies

$$g'(0) = (\nabla f(x_k), v) = -\|F(x_k)\|^2 < 0 \tag{44}$$

We define

$$x_{k+1} := x_k + \alpha_kv$$

**Method 1:** We choose a fixed $\alpha > 0$ for all iterations.

**Method 2:** Find $\alpha_k > 0$ such that $\|F(x_{k+1})\|$ is minimal.

**Method 3:** Find $\alpha_k > 0$ such that $f(x_{k+1})$ is minimal: “exact line search”

Find $\alpha > 0$ such that $f(x_0 + \alpha v)$ is minimal \tag{45}

$$x_{k+1} := x_k + \alpha_kv \tag{46}$$

Note that $G(t) := g'(t) = (F(x_0 + tv), v)$ satisfies because of (37), (38)

$$|G(t) - G(s)| \leq L \|v\|^2 |t - s|$$

$$(G(t) - G(s))(t - s) \geq \gamma \|v\|^2 (t - s)^2$$

I.e., $g(t)$ satisfies the 1-dimensional versions of (37), (38). Hence $g(t)$ has a unique minimum $\alpha_* \in \mathbb{R}$. Because of (44) we must have $\alpha_*> 0$. 

30
Lemma 10.2. which contains

Now the co-coercivity of

Using this in (47) gives

Proof.

The function

In particular, for

Remark. Finding the minimum of a function \( g(x) \) on an interval \([a, b]\): There are efficient methods with superlinear convergence available. In Matlab you can use \texttt{fminbnd}. Example: \( f=x(\sin(x); \ x=[31, f(4, 6)] \)

We want to show that our iterative method \( x_{k+1} = G(x_k) \) converges: From (4)

\[
\|x_{k+1} - x_*\|^2 = \|x_k - x_*\|^2 - 2\alpha (F(x_k) - F(x_*), x_k - x_*) + \alpha^2 \|F(x_k) - F(x_*)\|^2 \leq (1 - 2\alpha\gamma + \alpha^2 L^2) \|x_k - x_*\|^2
\] (47)

Method 1

For Method 1 with a fixed \( \alpha \) we obtained with \( \kappa := \frac{L}{\gamma} \)

For \( 0 < \alpha < \frac{2\gamma}{L^2} \) we have \( \|x_{k+1} - x_*\| \leq q \|x_k - x_*\| \) with \( q = (1 - 2\alpha\gamma + \alpha^2 L^2)^{1/2} < 1 \).

For \( \alpha = \frac{\gamma}{L^2} \) we have \( \|x_{k+1} - x_*\| \leq q_* \|x_k - x_*\| \) with \( q_* = (1 - \kappa^{-2})^{1/2} \), i.e., we need \( O(\kappa^2) \) iterations.

Actually we can obtain from (47) a better estimate, see [Nesterov: “Introductory lectures on convex optimization”] and [Theorems 7.4.4. and 14.2.2 in “Convexity and Optimization” by L.-Å. Lindahl].

We used in 47 the Lipschitz property to get the upper bound \( \alpha^2 \|F(x_k) - F(x_*)\|^2 \leq \alpha^2 L^2 \|x_k - x_*\|^2 \), and because of this we obtained terms with \( \kappa^2 \). The idea is to use a lower bound for \( (F(x_k) - F(x_*), x_k - x_*) \) which contains \( \|F(x_k) - F(x_*)\|^2 \), and can absorb the \( \alpha^2 \|F(x_k) - F(x_*)\|^2 \) term.

Lemma 10.2. Assume \( F := \nabla f \) satisfies (37), (38). Then

\[
(F(x + v) - F(x), v) \geq \frac{\gamma L}{\gamma + L} \|v\|^2 + \frac{1}{\gamma + L} \|F(x + v) - F(x)\|^2
\] (48)

Proof. The function \( g(x) := f(x) - \frac{1}{2}\gamma \|x\|^2 \) is convex and \( \nabla g \) satisfies a Lipschitz condition with \( \tilde{L} = L - \gamma \). Now the co-coercivity of \( g \) with \( y = x + v \) gives the result.

Now we obtain

Theorem 10.3. Assume \( F := \nabla f \) satisfies (37), (38). Then the steepest descent iteration with fixed \( \alpha \) converges if \( \alpha \) is sufficiently small: For \( 0 < \alpha \leq \frac{2}{\gamma + L} \)

\[
\|x_{k+1} - x_*\|^2 \leq \left( 1 - \alpha \frac{2\gamma L}{\gamma + L} \right) \|x_k - x_*\|^2
\]

In particular, for \( \alpha = \frac{2}{\gamma + L} \) we have with \( \kappa := \frac{L}{\gamma} \)

\[
\|x_{k+1} - x_*\| \leq \left( 1 - \frac{2}{1 + \kappa} \right) \|x_k - x_*\|
\]

Proof. We get from (48) with \( x = x_* \) and \( x + v = x_k \)

\[
(F(x_k), x_k - x_*) \geq \frac{\gamma L}{\gamma + L} \|x_k - x_*\|^2 + \frac{1}{\gamma + L} \|F(x_k)\|^2
\]

Using this in (47) gives

\[
\|x_{k+1} - x_*\|^2 \leq \|x_k - x_*\|^2 - 2\alpha \left[ \frac{\gamma L}{\gamma + L} \|x_k - x_*\|^2 + \frac{1}{\gamma + L} \|F(x_k)\|^2 \right] + \alpha^2 \|F(x_k) - F(x_*)\|^2
\]

\[
= \left( 1 - \frac{2\alpha\gamma L}{\gamma + L} \right) \|x_k - x_*\|^2 + \alpha \left( \alpha - \frac{2}{\gamma + L} \right) \|F(x_k)\|^2
\]
For $\alpha \leq \frac{2}{\gamma + L}$ we get

$$
\|x_{k+1} - x_*\|^2 \leq \left(1 - \alpha \frac{2\gamma L}{\gamma + L}\right) \|x_k - x_*\|^2
$$ (49)

For $\alpha = \frac{2}{\gamma + L}$ we get

$$
\|x_{k+1} - x_*\|^2 \leq \left(1 - \frac{4\gamma L}{(L + \gamma)^2}\right) \|x_k - x_*\|^2
$$

Since

$$
\frac{(L + \gamma)^2 - 4\gamma L}{(L + \gamma)^2} = \frac{(L - \gamma)^2}{(L + \gamma)^2}
$$

we obtain

$$
\|x_{k+1} - x_*\|^2 \leq \left(1 - \frac{2\gamma}{L + \gamma}\right)^2 \|x_k - x_*\|^2
$$

Therefore we need $O(\kappa)$ rather than $O(\kappa^2)$ iterations. For $\alpha = \frac{2}{\gamma + L}$ we obtain with (39)

$$
\|x_k - x_*\| \leq \left(1 - \frac{2}{\kappa + 1}\right)^k \|x_0 - x_*\| 
$$ (50)

$$
f(x_k) - f(x_*) \leq \frac{L}{2} \|x_k - x_*\|^2 \leq \frac{L}{2} \left(1 - \frac{2}{\kappa + 1}\right)^{2k} \|x_0 - x_*\|^2
$$ (51)

For the norm of the new residual we obtain from (6) and (48)

$$
\|F(x_{k+1})\|^2 = \|r_k + F(x_k) - F(x_k + \alpha r_k)\|^2 = \|r_k\|^2 - 2\alpha^{-1} \left(F(x_k + \alpha r_k) - F(x_k), \alpha r_k\right) + \|F(x_k + \alpha r_k) - F(x_k)\|^2
$$

$$
\leq \|r_k\|^2 - 2\alpha^{-1} \left[\frac{\gamma L}{\gamma + L} \|\alpha r_k\|^2 + \frac{1}{\gamma + L} \|F(x_k + \alpha r_k) - F(x_k)\|^2\right] + \|F(x_k + \alpha r_k) - F(x_k)\|^2
$$

$$
= \left(1 - \frac{2\alpha\gamma L}{\gamma + L}\right) \|r_k\|^2 + \left(1 - \alpha^{-1} \frac{2}{\gamma + L}\right) \|F(x_k + \alpha r_k) - F(x_k)\|^2
$$

Note that for $\alpha \leq \frac{2}{\gamma + L}$ the second term is $\leq 0$, hence

$$
\|F(x_{k+1})\|^2 \leq \left(1 - \frac{2\alpha\gamma L}{\gamma + L}\right) \|F(x_k)\|^2
$$

For $\alpha = \frac{2}{\gamma + L}$ we obtain

$$
\|F(x_{k+1})\| \leq \left(1 - \frac{2}{\kappa + 1}\right) \|F(x_k)\|
$$

**Method 2**

If we choose $\alpha_k$ to minimize $\|F(x_{k+1})\|$ we will do at least as well for $\alpha = \frac{2}{\gamma + L}$, and therefore obtain

$$
\|F(x_{k+1})\| \leq \left(1 - \frac{2}{\kappa + 1}\right) \|F(x_k)\|
$$

Using (5) we obtain for the errors

$$
\|x_k - x_*\| \leq \gamma^{-1} \|F(x_k)\| \leq \gamma \left(1 - \frac{2}{\kappa + 1}\right)^k \|F(x_0)\|
$$
10.3 Convergence for exact line search

We saw: if we choose a fixed value for $\alpha$ we can achieve $f(x_k) - f(x_*) \leq Cq^{2k}$ with $q = 1 - \frac{2}{\kappa + 1}$.

For Method 3 we actually choose $\alpha_k$ to minimize $f(x_{k+1})$ at each step. So it should be easy to prove $f(x_k) - f(x_*) \leq Cq^{2k}$ with $q = 1 - \frac{2}{\kappa + 1}$ in this case as well.

However, this is surprisingly difficult. This does NOT follow from the previous results!

So far we proved results $\|x_{k+1} - x_*\| \leq q \|x_k - x_*\|$ and $\|r_{k+1}\| \leq q \|r_k\|$. What we need is a result

$$f(x_{k+1}) - f(x_*) \leq q [f(x_k) - f(x_*)]$$

The steepest descent method goes back to Cauchy, but this “obvious” result was only proved in 2016: [Theorem 1.2 in “On the worst-case complexity of the gradient method with exact line search for smooth strongly convex functions” by E. de Klerk, F. Glineur, A. B. Taylor 2016]

**Theorem 10.4.** Assume $F := \nabla f$ satisfies (37), (38). Then the steepest descent iteration with exact line search satisfies with $\kappa := L/\gamma$

$$f(x_{k+1}) - f(x_*) \leq \left(1 - \frac{2}{\kappa + 1}\right)^2 [f(x_k) - f(x_*)]$$

We need the following Lemma:

**Lemma 10.5.** Assume $F := \nabla f$ satisfies (37), (38). Then for any $x, y \in \mathbb{R}^n$

$$f(y) - f(x) - (F(x), y - x) \geq \frac{1}{2(L - \gamma)} \left[\|F(y) - F(x)\|^2 + \gamma L \|y - x\|^2 - 2\gamma (F(y) - F(x), y - x)\right]$$

(52)

**Proof.** Let $\tilde{f}(y) := f(y) - (F(x), y - x)$. Then $\tilde{F}(y) := \nabla \tilde{f}(y) = F(y) - F(x)$ and $\tilde{F}(x) = 0$. Now use (40) for $\tilde{f}$ with $x$ in place of $x_*$.

**Proof of Theorem 10.4:** We use (52) for $x_0, x_1, x_*$. With $f_0 := f(x_0)$, $F_0 := F(x_0)$ etc. we get

$$f_0 - f_1 - (F_1, x_0 - x_1) \geq \frac{1}{2(L - \gamma)} \left[\|F_0 - F_1\|^2 + \gamma L \|x_0 - x_1\|^2 - 2\gamma (F_0 - F_1, x_0 - x_1)\right]$$

(53)

$$f_* - f_0 - (F_0, x_* - x_0) \geq \frac{1}{2(L - \gamma)} \left[\|F_* - F_0\|^2 + \gamma L \|x_* - x_0\|^2 - 2\gamma (F_* - F_0, x_* - x_0)\right]$$

(54)

$$f_* - f_1 - (F_1, x_* - x_1) \geq \frac{1}{2(L - \gamma)} \left[\|F_* - F_1\|^2 + \gamma L \|x_* - x_1\|^2 - 2\gamma (F_* - F_1, x_* - x_1)\right]$$

(55)

Note that we have after exact line search $(F_0, F_1) = 0$, hence

$$-(F_0, F_1) \geq 0$$

(56)

We have $x_1 = x_0 - \alpha F_0$, hence $(x_0 - x_1, F_1) = \alpha (F_0, F_1) = 0$, hence

$$(F_1, x_0 - x_1) \geq 0$$

(57)

Now we multiply (53) by $\frac{L - \gamma}{L + \gamma}$, (54) by $2\gamma \frac{L - \gamma}{(L + \gamma)^2}$, (55) by $\frac{2\gamma}{L + \gamma}$, (56) by $\frac{2}{L + \gamma}$, (57) by 1 and add this together. We obtain

$$f_1 - f_* \leq \left(\frac{L - \gamma}{L + \gamma}\right)^2 (f_0 - f_1) - \frac{\gamma L (L + 3\gamma)}{2(L + \gamma)^2} \|x_0 - \frac{L + \gamma}{L + 3\gamma} x_* - \frac{3L + 3\gamma}{L + 3\gamma} F_0 - \frac{L + \gamma}{L + 3\gamma} F_1\|^2$$

$$- \left(\frac{2L^2}{L + 2L^2 - 3\gamma^2}\right) \|x_1 - x_* - \frac{(L - \gamma)^2}{2\gamma (L + \gamma)} F_0 - \frac{L + \gamma}{2\gamma L} F_1\|^2$$

$$f_1 - f_* \leq \left(\frac{L - \gamma}{L + \gamma}\right)^2 (f_0 - f_1)$$

□
Therefore we obtain for exact line search

\[
f(x_k) - f(x^*) \leq \left(1 - \frac{2}{\kappa + 1}\right)^{2k} [f(x_0) - f(x^*)]
\]  

(58)

We use (39)

\[
\frac{\gamma}{2} \|x_k - x^*\|^2 \leq [f(x_k) - f(x^*)]
\]

and get the error bound

\[
\|x_k - x^*\| \leq \left(\frac{2}{\gamma}\right)^{1/2} \left(1 - \frac{2}{\kappa + 1}\right)^k [f(x_0) - f(x^*)]^{1/2}
\]

(59)

**Remark 10.6.** Assume that the derivative of \( F(x) \) exists: The **Hessian** \( H(x) \) contains the 2nd order partial derivatives of \( f \)

\[
H(x) := DF(x) = \begin{bmatrix}
\frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\
\vdots & \ddots & \vdots \\
\frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n^2}
\end{bmatrix}
\]

The matrix \( H(x) \) is symmetric and positive definite: \((H(x)v, v) \geq \gamma \|v\|^2\), i.e., \( \lambda_{\text{min}}(H(x)) \geq \gamma \).

As the iterative method (Method 1 or Method 2 or Method 3) converges we have \( x_k \to x^* \). For \( x \) close to \( x^* \) we have from Taylor’s theorem

\[
F(x) = F(x^*) + H(x^*)(x - x^*) + O(\|x - x^*\|)
\]

As \( k \to \infty \) the function \( F(x) \) gets closer to the linear function \( H(x^*)(x - x^*) \). Hence we expect that the iteration (Method 1 or Method 2 or Method 3) asymptotically behaves like the iteration for the linear problem with a symmetric positive definite matrix \( A = H(x^*) \), i.e., convergence with factor \( q = 1 - \frac{2}{\kappa + 1} \) where \( \kappa = \text{cond}_2(A) \). But the above results (50), (51), (58), (59) are much stronger: we obtain the sharp convergence results with factor \( q = 1 - \frac{2}{\kappa + 1} \) for all \( k = 0, 1, 2, \ldots \).

### 11 Summary for Richardson iteration

#### 11.1 General case: Solve \( F(u) = b \) where \( F \) satisfies Lipschitz, coercivity

We considered a function \( F: V \to V \). For a given \( b \in V \) we want to find \( u \in V \) such that

\[
F(u) = b
\]

We assume

\[
\|F(v) - F(u)\| \leq L \|v - u\| \quad \text{"Lipschitz condition"} \quad (60)
\]

\[
(F(v) - F(u), v - u) \geq \gamma \|v - u\|^2 \quad \text{"coercivity condition"} \quad (61)
\]

We pick an initial guess \( u_0 \in V \). For the current approximation \( u_k \) we compute the residual \( r_k := b - F(u_k) \) and define with \( \alpha > 0 \) the **Richardson iteration**

\[
u_{k+1} := u_k + \alpha r_k
\]

We have

\[
\gamma \|u_k - u^*\| \leq \|r_k\| \leq L \|u_k - u^*\|
\]
Method 1: Use fixed $\alpha$

For $\alpha \in (0, 2\gamma/L^2)$ we have $q := (1 - 2\alpha \gamma + \alpha^2 L^2)^{1/2} < 1$ and
\[
\|u_k - u_*\| \leq q^k \|u_0 - u_*\| \\
\|r_k\| \leq q^k \|r_0\|
\]
The best choice is $\alpha = \gamma/L^2$, then we have with $\kappa := \frac{L}{\gamma}$ the factor $q = (1 - \kappa^{-2})^{1/2}$. Hence we need $O(\kappa^2)$ iterations to achieve $\|u_k - u_*\| \leq$ tolerance or $\|r_k\| \leq$ tolerance.

Method 2: Use line search to find $\alpha_k$ minimizing $\|r_{k+1}\|$ 

If we perform exact line search we obtain
\[
\|r_k\| \leq (1 - \kappa^{-2})^{k/2} \|r_0\| \\
\|u_k - u_*\| \leq \gamma^{-1} (1 - \kappa^{-2})^{k/2} \|r_0\|
\]
Hence we need $O(\kappa^2)$ iterations to achieve $\|u_k - u_*\| \leq$ tolerance or $\|r_k\| \leq$ tolerance.

11.2 Special case: minimize $f(u)$ where $F = \nabla f$ and $F$ satisfies Lipschitz, coercivity

Here $b = 0$. For a given approximation $x_k$ we can compute $r_k := b - F(u_k)$ and $\|r_k\|$. We can also compute $f(u_k)$. We have
\[
\frac{\gamma}{2} \|u_k - u_*\|^2 \leq f(u_k) - f(u_*) \leq \frac{L}{2} \|u_k - u_*\|^2
\]

Method 1: Use fixed $\alpha$

For $\alpha \in \left(0, \frac{2}{\gamma + L}\right]$ we have $q := \left(1 - \alpha \frac{2\gamma L}{\gamma + L}\right)^{1/2} < 1$ and
\[
\|u_k - u_*\| \leq q^k \|u_0 - u_*\| \\
\|r_k\| \leq q^k \|r_0\| \\
f(x_k) - f(x_*) \leq \frac{L}{2} q^{2k} \|u_0 - u_*\|^2
\]
The best choice is $\alpha = \frac{2}{\gamma + L}$, then we have with $\kappa := \frac{L}{\gamma}$ the factor $q = 1 - \frac{2}{\kappa + 1}$. Hence we need $O(\kappa)$ iterations to achieve $\|u_k - u_*\| \leq$ tolerance or $\|r_k\| \leq$ tolerance or $f(x_k) - f(x_*) \leq$ tolerance.

Method 2: Use line search to find $\alpha_k$ minimizing $\|r_{k+1}\|$

If we perform exact line search we obtain
\[
\|r_k\| \leq \left(1 - \frac{2}{\kappa + 1}\right)^k \|r_0\| \\
\|u_k - u_*\| \leq \gamma^{-1} \left(1 - \frac{2}{\kappa + 1}\right)^k \|r_0\| \\
f(x_k) - f(x_*) \leq \frac{L}{2} \gamma^{-2} \left(1 - \frac{2}{\kappa + 1}\right)^{2k} \|r_0\|^2
\]
Hence we need $O(\kappa)$ iterations to achieve $\|u_k - u_*\| \leq$ tolerance or $\|r_k\| \leq$ tolerance or $f(x_k) - f(x_*) \leq$ tolerance.
Method 3: Use line search to find $\alpha_k$ minimizing $f(u_{k+1})$

If we perform exact line search we obtain

\[ f(x_k) - f(x^*_k) \leq \left( 1 - \frac{2}{\kappa + 1} \right)^k [f(x_0) - f(x^*_k)] \]

\[ \|u_k - u^*_k\| \leq \left( \frac{2}{\gamma} \right)^{1/2} \left( 1 - \frac{2}{\kappa + 1} \right)^k [f(x_0) - f(x^*_k)]^{1/2} \]

Hence we need $O(\kappa)$ iterations to achieve $\|u_k - u^*_k\| \leq \text{tolerance}$ or $f(x_k) - f(x^*_k) \leq \text{tolerance}$.

11.3 Special case: linear $F(u) = Au - b$ with $(Au, u) > 0$, nonsymmetric $A$

Here we have $L = \|A\|_2$ and $\gamma = \lambda_{\min} \left( \frac{1}{2} (A + A^T) \right)$.

For Method 2 (minimizing $\|r_{k+1}\|$) we obtain an explicit formula for $\alpha_k$:

\[ \alpha_k := \frac{(Ar_k, r_k)}{\|Ar_k\|^2} \]

In the linear case we can prove a sharper result: With the symmetric part $H$ and the antisymmetric part $S$ given by

\[ H := \frac{1}{2} (A + A^T), \quad S := \frac{1}{2} (A - A^T) \]

we define

\[ K = \frac{\|H\|_2}{\gamma} + \left( \frac{\|S\|_2}{\gamma} \right)^2 \]

and obtain with $q := (1 - K^{-1})^{1/2}$

\[ \|r_k\| \leq q^k \|r_0\| \]

\[ \|u_k - u^*_k\| \leq \gamma^{-1} q^k \|r_0\| \]

We will need $O(K)$ iterations. If the antisymmetric part $S$ is small compared to $H$ this will give a sharper estimate.

11.4 Special case: linear $F(u) = Au - b$ with $(Au, u) > 0$, symmetric $A$

The linear system $Au = b$ is equivalent to minimizing $f(u) = \frac{1}{2} (Au, u) - (b, u)$.

$\|v\|_A := (Av, v)^{1/2}$ is a norm and

\[ f(u) - f(u^*_k) = \frac{1}{2} \|u - u^*_k\|^2_A \]

We have $L = \|A\|_2 = \lambda_{\max}(A)$ and $\gamma = \lambda_{\min}(A)$. Hence $\kappa := \frac{L}{\gamma} = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} = \text{cond}_2(A)$.

Method 1 (fixed $\alpha$): For $\alpha \in \left( 0, \frac{2}{\lambda_{\max}} \right)$ we obtain

\[ q := \max \{|1 - \alpha \lambda_{\min}|, |1 - \alpha \lambda_{\max}|\} < 1 \]

\[ \|u_k - u^*_k\| \leq q^k \|u_0 - u^*_k\| \]

\[ \|u_k - u^*_k\|_A \leq q^k \|u_0 - u^*_k\|_A \]

\[ \|r_k\| \leq q^k \|r_0\| \]

The best value is $\alpha = \frac{2}{\lambda_{\min} + \lambda_{\max}}$ and we have $q = 1 - \frac{2}{\kappa + 1}$. Hence we need $O(\kappa)$ iterations to achieve a given tolerance.
For Method 2 (minimizing $\|r_{k+1}\|$) we have the explicit formula

$$\alpha_k := \frac{(A_{rk}, r_k)}{\|A_{rk}\|^2}$$

and obtain with $q = 1 - \frac{2}{\kappa + 1}$

$$\|r_k\| \leq q^k \|r_0\|$$
$$\|u_k - u_*\| \leq \gamma^{-1} q^k \|r_0\|$$

For Method 3 (minimizing $f(u_{k+1})$) we have the explicit formula

$$\alpha_k := \frac{\|r_k\|^2}{(A_{rk}, r_k)}$$

and obtain with $q = 1 - \frac{2}{\kappa + 1}$

$$\|u_k - u_*\|_A \leq q^k \|u_0 - u_*\|_A$$
$$\|u_k - u_*\| \leq \gamma^{-1/2} q^k \|u_0 - u_*\|_A$$