Block versions.

The Jacobi method and the Gauss-Seidel method can also be employed in block versions. Namely, let \( A \) be of the form

\[
A = \begin{pmatrix}
A_{11} & A_{12} & \cdots & A_{1p} \\
A_{21} & A_{22} & \cdots & A_{2p} \\
\vdots & \vdots & \ddots & \vdots \\
\cdots & \cdots & \cdots & A_{pp}
\end{pmatrix}
\]

where the entries \( A_{ij} \) are matrix blocks.

The block Jacobi method consists then in defining the block diagonal matrix

\[
D = \begin{pmatrix}
A_{11} & & & \\
& A_{22} & & \\
& & \ddots & \\
& & & A_{pp}
\end{pmatrix}
\]

and performing the iteration \( x_{k+1} = x_k + D^{-1}(b - Ax) \). Of course, the block Gauss-Seidel and the block SOR methods can be defined in an analogous way. Convergence theories exist for these cases as well.

If \( A \) corresponds to a FD or FE matrix like in the 2D Poisson example, with original lexicographic ordering of the unknowns, then the diagonal blocks \( A_{i,i} \) are tridiagonal matrices, and inversion of \( D \) amounts to solving small tridiagonal systems, e.g. via Cholesky decomposition. The higher effort for such a block iteration is usually paid off by a faster convergence rate.

In the context of FD methods, such a version of block relaxation scheme is also called ‘line relaxation’, where the coupling of unknowns in one of the coordinate directions is retained. Another variant is ‘alternating direction line relaxation’, where the ordering of unknowns varied in in an alternating fashion, similarly as in SSOR; cf. e.g. [15].

6 Chebyshev Acceleration and Semi-iterative Methods

We have already pointed out some difficulties in choosing the optimal relaxation parameter \( \omega_{opt} \). Chebyshev acceleration and its variants are an alternative, not uncommon tool to accelerate the convergence of a sequence \( (x_k)_{k=0}^{\infty} \). We assume that this sequence is generated by the primary iteration

\[
x_{k+1} = Mx_k + Nb
\]

We assume \( \rho(M) < 1 \), such that the primary iteration (6.1) converges. We ask: Can we construct, for every \( k \), an approximation \( y_k \) based on \( x_0, \ldots, x_k \) such that the new sequence \( (y_k)_{k=0}^{\infty} \) features faster convergence towards the fixed point \( x^* \) of (6.1)? To this end, we make the ansatz for a secondary iteration

\[
y_k = \sum_{m=0}^{k} a_{k,m} x_m
\]

for some parameters \( a_{k,m} \) to be chosen. With the polynomials

\[
p_k(t) = \sum_{m=0}^{k} a_{k,m} t^m
\]
and
\[ x_m = M^m x_0 + \sum_{\ell=1}^m M^{m-\ell} b = M^m x_0 + q_m(M)b \]
(with polynomials \(q_m \in \mathcal{P}_m\)), the secondary iteration (6.2) can be written as
\[ y_k = p_k(M)x_0 + \tilde{q}_k(M)b, \quad \tilde{q}_k(M) = \sum_{m=0}^k a_{k,m} q_m(M) \]
with matrix polynomials \(p_k \in \mathcal{P}_k\). Of course, we require that the parameters \(a_{k,m}\) be chosen such that a fixed point of (6.1) is reproduced, i.e., if we would consider \((x_k)_{k=0}^\infty\) to be the constant sequence \((x_*)_{k=0}^\infty\), then the sequence \((y_k)_{k=0}^\infty\) defined by (6.2) should be the same constant sequence. That is, we require the consistency condition
\[ 1 = \sum_{m=0}^k a_{k,m} = p_k(1) \quad \forall k \in \mathbb{N}_0 \]
i.e., all \(y_k\) are weighted means of the \(x_m, m = 0 \ldots k\). Under this assumption, we can express the error \(e_k = y_k - x_*\) in terms of the primary error \(e_k = x_k - x_*\):
\[ e_k = y_k - x_* = \sum_{m=0}^k a_{k,m} (x_m - x_*) = \sum_{m=0}^k a_{k,m} e_m = \sum_{m=0}^k a_{k,m} M^m e_0 = p_k(M)e_0 \]
This formula for the error gives an indication of how the coefficients \(a_{k,m}\), or, equivalently, the polynomials \(p_k\) should be chosen: we should choose \(p_k\) such that \(\|p_k(M)\|\) is small (minimal) in some norm of interest. Since sometimes information about the spectrum of \(M\) is available, we state:

- By Exercise 6.1, \(\sigma(p(M)) = p(\sigma(M))\) for any matrix \(M\) and any polynomial \(p\). Thus,
\[ \rho(p(M)) = \max\{|p(\lambda)| : \lambda \in \sigma(M)\} \quad (6.4) \]
- All normal matrices \(M\) satisfy \(\|M\|_2 = \rho(M)\). Hence, for normal matrices \(M\), where \(p(M)\) is also normal, there holds \(\rho(p(M)) = \|p(M)\|_2\) for all polynomials \(p\).
- Let \(A\) be SPD and let \(M\) be \(A\)-selfadjoint. Then, analogously as for the symmetric case we have \(\rho(M) = \|M\|_A\), and \(\rho(p(M)) = \|p(M)\|_A\) for all polynomials \(p\).

These considerations suggest that a reasonable procedure is to seek \(p_k \in \mathcal{P}_k\) (the linear space of polynomials of degree \(\leq k\)) as the solution of the following minimization problem:
\[ \min_{p_k \in \mathcal{P}_k, \quad p_k(1)=1} \max_{\lambda \in \sigma(M)} |p_k(\lambda)| \]
Since this problem is still hard to solve, we will settle for less: If \(\Gamma \subset \mathbb{C}\) is a closed set such that \(\sigma(M) \subset \Gamma\), then we could seek to solve the following minimization problem:
\[ \min_{p_k \in \mathcal{P}_k, \quad p_k(1)=1} \max_{z \in \Gamma} |p_k(z)| \quad (6.5) \]
Of course, this still requires some \textit{a priori} knowledge about the location of the spectrum. Here we consider the case that \(\sigma(A) \subset \Gamma = [a, b]\), an interval on the real line. In this case, the minimization problem (6.5) can be solved explicitly (Corollary 6.3 ahead). As we will see in Section 6.2, the numerical realization can also be achieved in an efficient way.

\textbf{Exercise 6.1} Let \(M\) be an arbitrary matrix, and let \(p\) be a polynomial. Show (e.g., using the Schur or Jordan form): \(\sigma(p(M)) = p(\sigma(M))\).
6.1 Chebyshev polynomials

The Chebyshev polynomials of the first kind, \( T_k \in P_k \), are defined by the three-term recurrence

\[
T_0(\xi) = 1, \quad T_1(\xi) = \xi, \quad T_{k+1} = 2\xi T_k(\xi) - T_{k-1}(\xi), \quad k \geq 1 \tag{6.6}
\]

It can be checked (e.g., by induction) that these polynomials can be expressed in closed form:

\[
T_k(\xi) = \begin{cases} 
\cos(k \arccos(\xi)), & |\xi| \leq 1 \\
\frac{1}{2} \left[ (\xi + \sqrt{\xi^2 - 1})^k + (\xi + \sqrt{\xi^2 - 1})^{-k} \right], & |\xi| \geq 1
\end{cases} \tag{6.7}
\]

Among the numerous remarkable properties of Chebyshev polynomials, we note that they are the solutions of an optimization problem of the form considered in (6.5):

**Theorem 6.2** Let \([\alpha, \beta] \subset \mathbb{R}\) be a non-empty interval, and let \(\gamma\) be any real scalar outside this interval. Then the minimum

\[
\min_{p \in P_k, p(\gamma) = 1} \max_{t \in [\alpha, \beta]} |p(t)| \tag{6.8}
\]

is attained by the polynomial\(^{24}\)

\[
p(t) = C_k(t) = \frac{T_k(1 + 2 \frac{t - \beta}{\beta - \alpha})}{T_k(1 + 2 \frac{\alpha - \beta}{\beta - \alpha})} \tag{6.9}
\]

Furthermore, the minimizer is unique.

\(^{24}\) For \(t \in [\alpha, \beta]\) we have \(1 + 2 \frac{t - \beta}{\beta - \alpha} \in [-1, 1] \).
6.2 Chebyshev acceleration for \( \sigma(M) \subset (-1, 1) \)

We now assume \( \sigma(M) \subset (-1, 1) \) (convergent primary iteration with \( M \) having a real spectrum). Moreover, we assume that parameters \(-1 < \alpha < \beta < 1\) are known such that \( \sigma(M) \subset [\alpha, \beta] \). With these parameters \( \alpha, \beta \) and \( \gamma = 1 \), we use the polynomials \( p_k(t) = C_k(t) \) explicitly given in Theorem 6.2 to define our the secondary iteration (6.2). This results in a ‘Chebyshev-accelerated’ iteration scheme. Note that this is a consistent choice since \( T_k(1) = 1 \) for all \( k \).

Improved convergence behavior of the Chebyshev iterates in self-adjoint cases.

To quantify the convergence behavior of the secondary Chebyshev iteration, we consider the case that the iteration matrix \( M \) is self-adjoint with respect to the energy product \( (\cdot, \cdot)_B \) for some SPD matrix \( B \), i.e., \( (Mx, y)_B \equiv (x, My)_B \ (M^* = M) \). Any such matrix has a real spectrum. Additionally, we now assume knowledge of \( \rho \in (0, 1) \) such that \( \sigma(M) \subset [-\rho, \rho] \). We then take the polynomials \( p_k \) defining the secondary iteration (6.3), as \( p_k(t) = C_k(t) \) with \( \alpha = -\rho, \beta = \rho \) and \( \gamma = 1 \), i.e.,

\[
y_k = \sum_{m=0}^k a_{k,m} x_m = p_k(M) x_0 + d_k, \quad a_{k,m} = \text{coefficients of } p_k,
\]

with

\[
p_k(t) = C_k(t) = \frac{1}{T_k(1/\rho)} T_k(t/\rho)
\]
We then obtain from Corollary 6.3
\[
\rho(p_k(M)) = \max_{\lambda \in \sigma(M)} |p_k(\lambda)| \leq \max_{\lambda \in [-\rho, \rho]} |p_k(\lambda)| = \frac{2c^k}{1 + c^{2k}} \leq 2c^k, \quad c = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}, \quad \kappa = \frac{1 + \rho}{1 - \rho}
\]
The assumption that $M$ is $B$-selfadjoint implies $\|p(M)\|_B = \rho(p(M))$ for all polynomials $p$. Hence, for the primary and secondary errors $e_k, \tilde{e}_k$ we have
\[
\left(\frac{\|e_k\|_B}{\|e_0\|_B}\right)^{1/2} \leq \|M^k\|_B^{1/2} \leq \rho(M),
\]
\[
\left(\frac{\|\tilde{e}_k\|_B}{\|e_0\|_B}\right)^{1/2} \leq \|p_k(M)\|_B^{1/2} = (\rho(p_k(M)))^{1/2} \leq 2^{1/2}c
\]
In Remark 5.7, we have seen that the convergence factors
\[
q_{\text{primary}} = \limsup_{k \to \infty} \left(\frac{\|e_k\|_B}{\|e_0\|_B}\right)^{1/2} \leq \rho(M),
\]
\[
q_{\text{Cheb}} = \limsup_{k \to \infty} \left(\frac{\|\tilde{e}_k\|_B}{\|e_0\|_B}\right)^{1/2} \leq c
\]
are good measures for the asymptotic behavior of the convergence speed. To compare $q_{\text{primary}}$ with $q_{\text{Cheb}}$, let us assume that the parameter $\rho$ is the best possible choice, i.e., $\rho = \rho(M)$. The interesting case is $\rho = 1 - \delta$ for small $\delta > 0$. We then have $c = \frac{1-1/\sqrt{\kappa}}{1+1/\sqrt{\kappa}}$ and $\kappa = \frac{1+\rho}{1-\rho}$. This leads to $c = 1 - c'\sqrt{\delta}$ with some $c' > 0$. We therefore arrive at
\[
q_{\text{primary}} \leq 1 - \delta, \quad q_{\text{Cheb}} \leq 1 - c'\sqrt{\delta} \quad (6.12)
\]
In practice we have\(^{25}\) $q_{\text{primary}} = \rho$, such that for small $\delta$, Chebyshev acceleration will noticeably improve the convergence behavior.

**Numerical realization.**

At first glance, a drawback of the Chebyshev acceleration appears to be that the definition of $y_k$ according to (6.10) requires knowledge of all primary iterates $x_0, \ldots, x_k$. In view of storage restrictions, this may be difficult to realize in practice. However, a clever rewriting and exploiting of the three-term recurrence for the Chebyshev polynomials $T_k$ removes this restriction.

Since the $T_k(\xi)$ satisfy the three-term recurrence (6.6), and so do the polynomials $p_k$ from (6.11):
\[
\mu_{k+1} p_{k+1}(t) = \frac{2}{\rho} \mu_k t p_k(t) - \mu_{k-1} p_{k-1}(t), \quad k \geq 1, \quad \text{with} \quad \mu_k = T_k(1/\rho) \quad (6.13)
\]
with initial functions
\[
p_0(t) = 1, \quad p_1(t) = \frac{T_1(t/\rho)}{T_1(1/\rho)} = \frac{t/\rho}{1/\rho} = t,
\]
i.e., $a_{0,0} = 1$ and $a_{1,0} = 0$, $a_{1,1} = 1$. We also observe the important property
\[
\mu_{k+1} = \frac{2}{\rho} \mu_k - \mu_{k-1} \quad (6.14)
\]
which can be seen directly from the properties of the Chebyshev polynomials or by observing that our requirement $p_k(1) = 1$ for all $k$ enforces this in view of (6.13).

\(^{25}\)Convince yourself that $q_{\text{primary}} = \rho$ unless $e_0$ has no component in the invariant subspace associated with the dominant eigenvalue.
We are now ready to implement Chebyshev acceleration. With \( x_\ast = \lim_{k \to \infty} x_k \) we obtain from the error equation \( \tilde{e}_k = p_k(M)e_0 \):

\[
y_{k+1} = x_\ast + \tilde{e}_{k+1} = x_\ast + p_{k+1}(M)e_0 = x_\ast + 2 \frac{\mu_k}{\rho \mu_{k+1}} Mp_k(M)e_0 - \frac{\mu_{k-1}}{\mu_{k+1}} p_{k-1}(M)e_0
\]

\[
= x_\ast + 2 \frac{\mu_k}{\rho \mu_{k+1}} M\tilde{e}_k - \frac{\mu_{k-1}}{\mu_{k+1}} \tilde{e}_{k-1} = x_\ast + 2 \frac{\mu_k}{\rho \mu_{k+1}} M(y_k - x_\ast) - \frac{\mu_{k-1}}{\mu_{k+1}} (y_{k-1} - x_\ast)
\]

\[
= 2 \frac{\mu_k}{\rho \mu_{k+1}} My_k - \frac{\mu_{k-1}}{\mu_{k+1}} y_{k-1} + \frac{1}{\mu_{k+1}} (\mu_{k+1} - 2 \frac{\mu_k}{\rho} M + \mu_{k-1}) x_\ast
\]

We now exploit the fact that \( x_\ast \) is a fixed point of the basic iteration, i.e., \( x_\ast = Mx_\ast + Nb \). This together with (6.14) allows us to remove the appearance of \( x_\ast \) and to obtain a direct three-term recurrence for the \( y_k \), without explicit use of the primary iterates \( x_k \):

\[
y_{k+1} = 2 \frac{\mu_k}{\rho \mu_{k+1}} My_k - \frac{\mu_{k-1}}{\mu_{k+1}} y_{k-1} + 2 \frac{\mu_k}{\rho \mu_{k+1}} Nb, \quad \text{with} \quad y_0 = x_0, \quad y_1 = x_1 = Mx_0 + Nb
\]

We collect these findings in Alg. 6.1.

**Algorithm 6.1 Chebyshev acceleration**

% input: primary iteration \( x_{k+1} = Mx_k + Nb \); 
% assumption: \( \sigma(M) \subset [-\rho, \rho] \subset (-1,1) \)

1: Choose \( x_0 \in \mathbb{R}^n \)
2: \( y_0 = x_0, \ y_1 = x_1 = Mx_0 + Nb \)
3: \( \mu_0 = 1, \ \mu_1 = 1/\rho \); 
4: for \( k = 1, 2, \ldots \) do 
5: \( \mu_{k+1} = \frac{2}{\rho} \mu_k - \mu_{k-1} \) % use recursion for \( \mu_k \) instead of definition \( \mu_k = T_k(1/\rho) \)
6: \( y_{k+1} = 2 \frac{\mu_k}{\rho \mu_{k+1}} My_k - \frac{\mu_{k-1}}{\mu_{k+1}} y_{k-1} + 2 \frac{\mu_k}{\rho \mu_{k+1}} Nb \)
7: end for 

### 6.3 Numerical example

We illustrate the Chebyshev acceleration again for the model problem of Example 2.2. Since we require the iteration matrix to have real spectrum, we would like it to be selfadjoint. For the model problem of Example 2.2, the iteration matrix \( M^{Jac} = I - D^{-1}A \) of the Jacobi method is indeed symmetric.\(^{26}\) For Chebyshev acceleration of the Jacobi method we assume that the parameter \( \rho \) has been chosen as \( \rho = \rho(M^{Jac}) \).

We also employ Chebyshev acceleration for the Gauss-Seidel method. Since \( \sigma(M^{GS}) \) is not necessarily real, we consider its symmetric variant, i.e., SSOR with \( \omega = 1 \). From Corollary 5.32 we know that \( \rho(M^{GS}) = \rho(M^{SOR(1)}) = \rho(M^{Jac})^2 \). For our calculations, we employ \( \rho = \rho(M^{Jac})^4 \) since we heuristically expect \( \rho(M^{SOR(1)}) \leq \rho(M^{GS})^2 \) (note: SSOR(1) is effectively two Gauss-Seidel steps). Fig. 6.14 shows the performance of various iterative methods including the Chebyshev accelerated versions of the Jacobi method and of SSOR(1). We observe that Chebyshev acceleration does indeed significantly improve the convergence.

\(^{26}\) For an arbitrary SPD matrix \( A \), \( M^{Jac} \) is \( A \)-selfadjoint; cf. the proof of Theorem 5.15.
A brief discussion is in order. We have already seen that \( \rho(M^\text{Jac}) = 1 - ch^2 + O(h^3) \). Hence, from our discussion in (6.12) we infer that \( \limsup_{k \to \infty} (\rho(p_k(M^\text{Jac})))^{1/k} = 1 - c'h + O(h^{3/2}) \) for some suitable \( c' \).

For the Chebyshev acceleration based on the SSOR(1) method, we have chosen \( \rho = \rho(M^\text{Jac})^4 \); again we get \( \rho = 1 - c''h^2 + O(h^3) \) and thus conclude that the Chebyshev accelerated SSOR(1) has a contraction rate of \( 1 - c'''h^2 + O(h^{3/2}) \).

Note that, in practice, good estimates for the spectrum of the iteration matrix are required. To obtain a bound on the largest eigenvalue of the iteration matrix, one possibility is to perform a few steps of a simple vector iteration; i.e., the parameter \( \rho \) in Fig. 6.15 is estimated with 10 steps of a simple vector iteration with starting vector \((1, 1 \ldots 1)^T\).
Figure 6.15: Chebyshev acceleration for Poisson problem (Example 2.2) of symmetric Gauss-Seidel (SSOR(1)) and SSOR(ω_{opt}) (ω_{opt} = optimal choice for SOR)

To illustrate the influence of the quality of available bounds for the spectrum of the iteration matrix, in Fig. 6.15 we show the performance of SSOR(1) with ρ = ρ^2(Jac) instead of ρ = ρ^4(Jac). We observe a significant deterioration of the performance, in spite of the fact that for h = 10^{-2} (i.e., N = 10,000) the values ρ^2(Jac) and ρ^4(Jac) are rather close, suggesting that the Chebyshev acceleration is quite sensitive. We also note that in the case N = 10,000 that the estimate obtained with merely 10 vector iterations is rather poor.

Figure 6.14 also contains the results for the Conjugate Gradient (CG) method introduced in the following chapter. The convergence behavior of CG observed for this example is more irregular, and typically of superlinear nature (with an acceleration effect in the later iteration steps). Figure 6.15 also contains the results for Chebyshev acceleration applied to SSOR(ω_{opt}), with ω_{opt} = optimal damping parameter for SOR according to Thm. 5.33. The combination of these techniques results in a very good performance, especially for a larger problem size.

In the numerical examples, we have used Alg. 6.1 to accelerate the SSOR iteration. Alg. 6.1 assumes σ(M) ⊂ [−ρ, ρ]. However, one could easily improve the performance of the accelerated SSOR method since the spectrum in fact satisfies σ(M^{SSOR(1)}) ⊂ [0, 1) as the following exercise shows.

**Exercise 6.4** Let A be SPD. Show: the spectrum σ(M^{SSOR(1)}) of the symmetric Gauss-Seidel method satisfies σ(M^{SSOR(1)}) ⊂ [0, 1). *(Hint: Show A ≤ N^{-1} and appeal to Exercise 5.17.)*

**Exercise 6.5** Formulate the Chebyshev acceleration algorithm for the general case that the iteration matrix M satisfies σ(M) ⊂ [α, β]. Let M^{SSOR} be the iteration matrix for the symmetric Gauss-Seidel iteration applied to the matrix A for the 2D Poisson problem. Compare the convergence behavior of the SSOR(1) method with the accelerated version. For the latter, obtain an estimate ρ for ρ(M^{SSOR}) by a few steps of vector iteration. Note that Exercise 6.4 implies σ(M^{SSOR}) ⊂ [0, 1).
7 Gradient Methods

Motivated by the rather slow convergence of classical iterative methods, and in view of the sensitivity of acceleration with respect to estimated parameters, a variety of alternative methods have been proposed. We first consider methods applicable to SPD matrices \( A \), which often arise as a result of the discretization of elliptic operators, e.g., the matrices of Examples 2.1, 2.2. Later we shall relax this condition and consider nonsymmetric equations. Recall that an SPD matrix \( A > 0 \) satisfies

\[
\langle x, A x \rangle = \langle x, x \rangle_A = \| x \|_A^2 \quad \forall x \neq 0
\]

In particular, all eigenvalues of \( A \) are positive.

The aim of gradient methods is to minimize the quadratic functional

\[
\phi(x) = \frac{1}{2} \langle A x, x \rangle - \langle b, x \rangle
\]

for some \( b \in \mathbb{R}^n \). We compute the gradient of \( \phi \) as the residual

\[
\nabla \phi(x) = A x - b \tag{7.2}
\]

Moreover, the Hessian matrix of \( \phi \) is given by the Jacobian of \( \nabla \phi \),

\[
H(x) = J(\nabla \phi(x)) \equiv A > 0
\]

Thus, the functional \( \phi \) has a unique minimum at \( x_* \), the stationary point of \( \phi \), satisfying \( \nabla \phi(x_*) = A x_* - b = 0 \). We conclude that

\[
For A > 0, solving A x = b is equivalent to finding the minimum of \( \phi(x) \) from (7.1).
\]

Exercise 7.1 Let \( x_* \) be the solution of \( A x = b, \ A > 0 \). Show that

\[
\phi(x) - \phi(x_*) = \frac{1}{2} \langle x - x_*, x - x_* \rangle = \frac{1}{2} \| x - x_* \|_A^2
\]

and conclude again that \( \phi \) has indeed a unique minimum.

Exercise 7.1 also shows that

\[
\text{Minimization of } \phi(x) \text{ over any subdomain } D \subset \mathbb{R}^n \\
\text{is equivalent to minimization of the error } e = x - x_* \text{ in the energy norm.}
\]

Remark 7.2 The equivalence of solving \( A x = b \) with \( A > 0 \) and minimization of \( \phi(x) \) from (7.1) is formally analogous to the variational formulation of an elliptic PDE. The simplest 2D example is the Poisson equation (2.2) with homogeneous Dirichlet boundary conditions, with the corresponding (energy) functional

\[
\phi(u) = \frac{1}{2} a(u, u) - (f, u)_{L^2(\Omega)}, \quad a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v \, dx \tag{7.4}
\]

Within the Sobolev space \( H^1_0(\Omega) \), the unique minimum of \( \phi(u) \) from (7.4) is attained for \( u_* \) = weak solution of the boundary value problem (2.2). In this context, \( \nabla u \) may be considered as an analog of the discrete object \( A^{1/2} x \), and \( \| u \|_{H^1} = a(u, u)^{1/2} = \| \nabla u \|_{L^2(\Omega)} \) is the corresponding energy norm. The integration-by-parts identity \( \int_{\Omega} (-\Delta u) v \, dx = \int_{\Omega} \nabla u \cdot \nabla v \, dx = a(u, v) \) (for \( u, v \in H^1_0(\Omega) \)) corresponds to the identity \( \langle Ax, y \rangle = \langle A^{1/2} x, A^{1/2} y \rangle = \langle x, y \rangle_A \).

\[27\] For general \( A \in \mathbb{R}^{n \times n} \) we have \( \nabla \phi(x) = (\text{Re} A) x - b \), where \( \text{Re} A = \frac{1}{2}(A + A^T) \) is the symmetric, or ‘real part’ of \( A \).
Simple iterative schemes for minimizing $\phi$ from (7.1) are ‘descent methods’ and proceed as follows:

Starting from an initial vector $x_0$, the iteration is given by

$$ x_{k+1} = x_k + \alpha_k d_k $$  \hspace{1cm} (7.5)

where the search direction $d_k \in \mathbb{R}^n$ and the step length $\alpha_k \in \mathbb{R}$ are to be chosen. Typically, once a search direction $d_k \neq 0$ is chosen, the step length $\alpha_k$ is taken as the minimizer of the one-dimensional minimization problem (‘line search’):

Find the minimizer $\alpha_k \in \mathbb{R}$ of $\alpha \mapsto \phi(x_k + \alpha d_k)$.  \hspace{1cm} (7.6)

This minimization problem is easily solved since it is quadratic and convex in $\alpha$: Define $\psi(\alpha) = \phi(x_k + \alpha d_k)$.

Then, the chain rule gives

$$ \psi'(\alpha) = \nabla \phi(x_k + \alpha d_k)^T d_k = (A(x_k + \alpha d_k) - b)_k = (-r_k + \alpha A d_k, d_k) = \alpha (A d_k, d_k) - (r_k, d_k) $$

where the residual $r_k$ is, as before, defined as

$$ r_k = -\nabla \phi(x_k) = b - A x_k $$  \hspace{1cm} (7.7)

Moreover, $\psi(\alpha)$ is, of course, convex: $\psi''(\alpha) \equiv (A d_k, d_k) > 0$. The condition on a minimizer $\alpha_k$ of $\psi$ is $\psi'(\alpha_k) = 0$. Thus,

$$ \alpha_k = \frac{(d_k, r_k)}{\|d_k\|^2_A} $$  \hspace{1cm} (7.8)

**Exercise 7.3** Show that the choice of $\alpha_k$ in (7.8) leads to an approximation $x_{k+1} = x_k + \alpha_k d_k$ such that

$$ \phi(x_{k+1}) - \phi(x_k) = -\frac{1}{2} \frac{|(d_k, r_k)|^2}{\|d_k\|^2_A} $$  \hspace{1cm} (7.9)

Thus, if $d_k$ is chosen such that $(d_k, r_k) \neq 0$, then $\phi(x_{k+1}) < \phi(x_k)$.  \textit{■}

**Remark 7.4** Some of the basic iterative methods are related to descent methods, or descent methods ‘in disguise’:

For $A \in \mathbb{R}^{n \times n}$, consider the case where the first $n$ search directions $d_0, \ldots, d_{n-1}$ are chosen as the unit vectors, $d_k = (0, \ldots, 1, \ldots, 0)^T$. Then we obtain from (7.8):

$$ \alpha_k = \frac{(d_k, r_k)}{\|d_k\|^2_A} = \frac{(r_k)_k}{A_{k,k}} $$

hence

$$ x_{k+1} = x_k + \frac{(r_k)_k}{A_{k,k}} d_k $$

which exactly corresponds to the $k$-th update in the inner loop of a Gauss-Seidel step (5.9). Note that $n$ of these updates results in a single Gauss-Seidel step, and further steps are obtained by repeating this procedure with cyclic choice of search directions (= unit vectors).

In the simplest version of the Richardson iteration (5.6) we simply take $d_k = r_k$ and $\alpha_k = 1$, but this does not minimize $\phi$ in the search direction $r_k$. The locally optimal choice (7.8) corresponds to the of the relaxation parameter $\alpha_k = \frac{|r_k|^2}{\|r_k\|^2_A}$ and leads to the steepest descent method discussed in the sequel.  \textit{■}
7.1 The Method of Steepest Descent (SD) for SPD systems

We need to specify the search direction \( d_k \) in the iteration (7.5). As shown in Exercise 7.1, most choices of \( d_k \) will lead to \( \phi(x_{k+1}) - \phi(x_k) < 0 \). The steepest descent method is a ‘greedy’ algorithm in that it chooses \( d_k \) as the local direction of steepest descent, which is given by

\[
d_k = -\nabla \phi(x_k) = r_k
\]

This choice of search direction, together with the step length \( \alpha_k \) given by (7.8), leads to the Steepest Descent Algorithm formulated in Alg. 7.1.  

---

**Algorithm 7.1 Steepest Descent**

1: Choose \( x_0 \in \mathbb{R}^n \)
2: for \( k = 0, 1, \ldots \) do
3: \( r_k = b - Ax_k \)
4: \( \alpha_k = \frac{(r_k, r_k)}{(r_k, Ar_k)} \)
5: \( x_{k+1} = x_k + \alpha_k r_k \)
6: end for

---

Orthogonal search directions.

A consequence of our choice for the step length \( \alpha_k \) in (7.8) is that

\[ \text{In SD, consecutive search directions are orthogonal to each other.} \]

To see this, we observe

\[
d_{k+1} = b - A x_{k+1} = b - A(x_k + \alpha_k d_k) = r_k - \alpha_k A d_k
\]

Inserting the definition of \( \alpha_k \) given by (7.8) gives, with \( d_k = r_k \),

\[
(d_{k+1}, d_k) = (r_k, d_k) - \alpha_k (A d_k, d_k) = (r_k, r_k) - \frac{(r_k, r_k)}{(r_k, A r_k)} (r_k, A r_k) = 0 \tag{7.10}
\]

This characteristic can be seen in Fig. 7.16.

---

28 Basically, the idea of SD can be applied to any minimization problem.
7.1 The Method of Steepest Descent (SD) for SPD systems

Convergence of the SD method.

In order to quantify the speed of convergence of the steepest descent method, we use the Kantorovich inequality as technical tool. Let $A$ be any real SPD matrix, and $\lambda_{\text{max}}$ and $\lambda_{\text{min}}$ its largest and smallest eigenvalues. Then, for all $x \in \mathbb{R}^n$,

$$
\frac{(A x, x)(A^{-1} x, x)}{(x, x)^2} \leq \frac{(\lambda_{\text{min}} + \lambda_{\text{max}})^2}{4 \lambda_{\text{min}} \lambda_{\text{max}}} \tag{7.11}
$$

For a proof see [15, Lemma 5.1].

Now we study the magnitude of the error vectors $e_k = x_r - x_k$ in the energy norm $\| \cdot \|_A$. Note that $A e_k = -r_k$, where $r_k = b - A x_k$ is the k-th residual. From (7.3) and (7.9) we obtain with $d_k = r_k$:

$$
\frac{1}{2} \|e_{k+1}\|_A^2 = \phi(x_{k+1}) - \phi(x_k) = (\phi(x_{k+1}) - \phi(x_k)) + (\phi(x_k) - \phi(x_*)) = -\frac{1}{2} \frac{(r_k, r_k)^2}{\|r_k\|_A^2} + \frac{1}{2} \|e_k\|_A^2 \tag{7.12}
$$

Now we use the Kantorovich inequality (7.11) and identity $r_k = -A^{-1} e_k$ to estimate

$$
\frac{|(r_k, r_k)|^2}{\|r_k\|_A^2} = \frac{|(r_k, r_k)|^2}{(A r_k, r_k)} \geq \frac{4 \lambda_{\text{min}} \lambda_{\text{max}}}{(\lambda_{\text{min}} + \lambda_{\text{max}})^2} (A^{-1} r_k, r_k) = \frac{4 \lambda_{\text{min}} \lambda_{\text{max}}}{(\lambda_{\text{min}} + \lambda_{\text{max}})^2} (e_k, A e_k) = \frac{4 \lambda_{\text{min}} \lambda_{\text{max}}}{(\lambda_{\text{min}} + \lambda_{\text{max}})^2} \|e_k\|_A^2
$$

Together with (7.12) this yields

$$
\|e_{k+1}\|_A^2 \leq \|e_k\|_A^2 \left(1 - \frac{4 \lambda_{\text{min}} \lambda_{\text{max}}}{(\lambda_{\text{min}} + \lambda_{\text{max}})^2}\right) = \|e_k\|_A^2 \left(\frac{\lambda_{\text{max}} - \lambda_{\text{min}}}{\lambda_{\text{max}} + \lambda_{\text{min}}}\right)^2
$$

$$
= \left(\frac{\lambda_{\text{max}} - \lambda_{\text{min}}}{\lambda_{\text{max}} + \lambda_{\text{min}}}\right)^2 \|e_k\|_A^2 = \left(\frac{\kappa_2(A) - 1}{\kappa_2(A) + 1}\right)^2 \|e_k\|_A^2 \tag{7.13}
$$

with the condition number $\kappa_2(A) = \lambda_{\text{max}}/\lambda_{\text{min}}$. From this reasoning we obtain

**Theorem 7.5** For the SD iteration applied to an SPD system $A x = b$, the error in the energy norm is bounded by

$$
\|e_k\|_A \leq \left(\frac{\kappa_2(A) - 1}{\kappa_2(A) + 1}\right)^k \|e_0\|_A \tag{7.14}
$$

I.e., the asymptotic convergence rate is bounded by $\frac{\kappa_2(A) - 1}{\kappa_2(A) + 1}$.

Hence $e_k \rightarrow 0$ as $k \rightarrow \infty$. Evidently, the speed of convergence depends on the spectrum $A$. In particular, when the condition number $\kappa_2(A)$ is large, the contours of the functional $\phi$, which are elliptic in shape, are long drawn-out, and the poor convergence rate suggested by (7.13) is graphically explained by ‘zig-zag’-paths similar to the one shown in Fig. 7.17. This illustrates a worst case, which occurs for an initial error close to the eigenvector associated with $\lambda_{\text{max}}$.

---

29 The Kantorovich inequality is an example for a ‘strengthened Cauchy-Schwarz (CS) inequality’. Using CS together with $\|A\|_2 = \lambda_{\text{max}}$, $\|A^{-1}\|_2 = 1/\lambda_{\text{min}}$ we would obtain the elementary, larger bound $\lambda_{\text{max}}/\lambda_{\text{min}} = \kappa_2(A)$ on the right-hand side of (7.11). In (7.14), this would result in the larger factor $\left(\frac{\kappa_2(A) - 1}{\kappa_2(A) + 1}\right)^k$. For $\kappa = \kappa_2(A) \rightarrow \infty$ the Kantorovich inequality gains a factor $\approx 1/4$, and with $\varepsilon = 1/k$ this gives the following bounds for the asymptotic convergence rates: $\approx 1 - 2 \varepsilon$ (Kantorovich) vs. $1 - \varepsilon/2$ (CS).

30 The best case: For $e_0 = 0$ any eigenvector of $A$, the SD iteration always would find the exact solution $x_r$ in one step, independent of the problem dimension (simple proof!). This is, of course, in no way a practical situation. Moreover, as illustrated in Fig. 7.17, a small deviation from the eigenvector associated with $\lambda_{\text{max}}$ will lead to a very poor convergence behavior for the case of large $\kappa_2(A)$. 

---

**Iterative Solution of Large Linear Systems** Ed. 2011
Example 7.6 The performance of the SD method can now be compared with that of the Jacobi and Gauss-Seidel methods. The convergence paths for these three methods are shown in Fig. 7.18. The system considered is again that of Example 2.2, i.e., the matrix $A$ arises from discretizing the 2D Poisson equation with the 5 point finite difference stencil over an $8 \times 8$ mesh or an $11 \times 11$ mesh. The right-hand side $b$ is taken as $b = (1, 1, \ldots, 1)^T$ and the starting vector is $x_0 = b$. Note that we are only plotting a 2D projection of the solution vector in Fig. 7.18, and therefore the steepest descent orthogonality property is not graphically observed. Note that the SD iteration slows down with increasing $k$.

The SD method does not prove a great improvement over the classical iterative methods. Nevertheless, it comes with a number of new concepts including formulating the problem as a minimization procedure and considering the relationship between consecutive search directions. These concepts will be extended in the following to generate more successful iterative procedures.

---

**Figure 7.17:** SD convergence path for a $2 \times 2$ matrix $A$ with $\kappa_2 \approx 25$

**Figure 7.18:** Convergence paths for the Jacobi, Gauss-Seidel and SD methods for the Poisson problem from Example 2.2. Left: case $N = 49$. Right: case $N = 100$. 
7.2 Nonsymmetric steepest descent algorithms

In the steepest descent algorithm we have required \( A \) to be SPD in order for the functional \( \phi \) to have a unique minimum at the solution of \( Ax = b \). Variations on the steepest descent algorithm for nonsymmetric systems have also been developed, see [15]. The most general, but by far not computationally cheapest or most efficient requires only that \( A \) be non-singular. Then, since \( A^T A \) is SPD, Alg. 7.1 can be applied to the normal equations

\[
A^T A x = A^T b
\]

This procedure is called the residual norm steepest descent method, and the functional being minimized in this case is

\[
\psi(x) = \frac{1}{2}(A x, A x) - (x, A^T b)
\]

This method minimizes the \( \ell_2 \)-norm of the residual, \( \| A x - b \|^2_2 \). However, in view of the convergence result (7.13), it is now the condition number of \( A^T A \), which is typically much larger than that of \( A \), that controls the convergence rate of the iteration.\(^{31}\)

7.3 Gradient methods as projection methods

One of the main characteristics of the SD Method is that consecutive search directions (i.e., the residuals) are orthogonal, (7.10), which implies that the \( \ell_2 \)-projection of the new residual onto the previous one is zero. Another way of putting it is: the approximation \( x_{k+1} \) is defined as the solution of

\[
\text{Find } x_{k+1} \in x_k + \text{span}\{r_k\} \text{ such that } r_{k+1} \perp r_k
\]

This is a local condition concerning only consecutive residuals; the ‘search history’, i.e., the information about the previous search directions \( r_0, \ldots, r_{k-1} \) in not exploited. One may hope that including this information in the method leads to faster convergence.

Krylov subspace methods are based on this idea: the approximation \( x_{k+1} \) is constructed such that the residual \( r_{k+1} \) is ‘orthogonal’ (in some appropriate sense to be specified below) to all previous residuals, search directions or a related set of vectors.

Remark 7.7 Brief review on orthogonal projectors:

Let \( \mathbb{R}^n = K \oplus K^\perp \), with \( x \perp y \) for all \( x \in K \) and \( y \in K^\perp \). This is an orthogonal subspace decomposition of \( \mathbb{R}^n \). Let

\[
K = \text{span}\{u_1, \ldots, u_m\}, \quad K^\perp = \text{span}\{v_1, \ldots, v_{n-m}\}
\]

with \((u_i, u_j) = \delta_{i,j},\ (v_i, v_j) = \delta_{i,j},\ \text{and } (u_i, v_j) = 0\). The union of the \( u_i \) and \( v_j \) is an orthogonal basis of the full space \( \mathbb{R}^n \). For each \( x \in \mathbb{R}^n \) we consider the corresponding Fourier expansion

\[
x = \sum_{i=1}^m (u_i, x) u_i + \sum_{j=1}^{n-m} (v_j, x) v_j =: P x + Q x
\]

This defines a pair \((P, Q)\) of orthogonal projectors. \( P \) projects onto \( K \) along \( K^\perp \), and vice versa.

\(^{31}\) (7.13) represents only an upper bound for the error. Nevertheless, the bound describes the overall convergence behavior quite realistically.
From (7.15) and with

\[
U = \begin{pmatrix}
  u_1 & u_2 & \cdots & u_m
\end{pmatrix} \in \mathbb{R}^{n \times m},
\]

\[
V = \begin{pmatrix}
  v_1 & v_2 & \cdots & v_{n-m}
\end{pmatrix} \in \mathbb{R}^{n \times (n-m)},
\]

we obtain the matrix representation for the projectors \( P, Q \in \mathbb{R}^{n \times n} \):

\[
\sum_{i=1}^{m} u_i (u_i^T x) = \sum_{i=1}^{m} (u_i u_i^T) x = U U^T x = P x \in K
\]

and analogously for the orthogonal complement,

\[
\sum_{j=1}^{n-m} v_j (v_j^T x) = \sum_{j=1}^{n-m} (v_j v_j^T) x = V V^T x = Q x \in K^\perp
\]

From the orthonormality relations \( U^T U = I_{m \times m}, \ V^T V = I_{(n-m) \times (n-m)}, \) and \( U^T V = 0_{m \times (m-n)}, \ V^T U = 0_{(n-m) \times m} \) we obtain the characterizing identities of a pair of orthogonal projectors:

\[
PP = P = P^T, \quad QQ = Q = Q^T, \quad PQ = Q P = 0_{n \times n}
\]

I.e., an orthogonal projector is idempotent (projector property) and symmetric. We also note the Pythagorean identity

\[
\|x\|_2^2 \equiv \|P x\|_2^2 + \|Q x\|_2^2
\]

Exercise 7.8 [See Exercise 5.14] \(^{32}\)

Consider a decomposition \( \mathbb{R}^n = K \oplus K^\perp_A \) analogously as above, with \((\cdot, \cdot)\) throughout replaced by \((\cdot, \cdot)_A\), and \(A\)-conjugate bases \(\{u_1, \ldots, u_m\}, \ \{v_1, \ldots, v_{n-m}\}\), i.e., \((u_i, u_j)_A = \delta_{i,j}, \ (v_i, v_j)_A = \delta_{i,j}, \) and \((u_i, v_j)_A = 0\). I.e., \(U\) and \(V\) are \(A\)-conjugate matrices, satisfying \(U^T A U = I_{m \times m}, \ V^T A V = I_{(n-m) \times (n-m)}, \) and \(U^T A V = 0_{m \times (m-n)}, \ V^T A U = 0_{(n-m) \times m}\).

Show that the corresponding pair \((P, Q)\) of ‘\(A\)-conjugate’ projectors onto \(K\) and \(K^\perp_A\) is given by

\[
P = U U^T A, \quad Q = V V^T A
\]

and \(P\) and \(Q\) satisfy

\[
PP = P = A^A, \quad QQ = Q = A^A, \quad PQ = Q P = 0_{n \times n}
\]

where \(M^A\) is the adjoint of a matrix \(M \in \mathbb{R}^{n \times n}\) w.r.t. \((\cdot, \cdot)_A\), i.e.,

\[
M^A = A^{-1} M^T A
\]

Note the Pythagorean identity

\[
\|x\|_A^2 \equiv \|P x\|_A^2 + \|Q x\|_A^2
\]

\(^{32}\) For \(x \in \mathbb{R}^n\) one may also denote \(x^A = x^T A = (A x)^T, \) with \((x, x)_A = x^A x, \) but this notation is not standard and we do not use it in the following.