Now consider G being invertible and denote

$$N = G^{-1}, \tag{12}$$

and (left-)multiply the r.h.s. of (11) by N. This motivates the definition of the consistent *linear fixed point iteration*

$$x_{k+1} = \underbrace{N\,H}_{=:M} x_k + Nb = Mx_k + Nb. \tag{13}$$

Lemma 1.7. *The fixed point iteration* (13) *is consistent.*

Proof. This holds true by construction. Indeed, we have for any r.h.s. b that for a fixed point of (13) there holds $x_* = G^{-1}Hx_* + G^{-1}b$. After left multiplication by G, we arrive at $Gx_* = Hx_* + b$, and hence, $(G - H)x_* = Ax_* = b$. □

Note that N is not computed, only the action $y \mapsto G^{-1}y$ or the approximate solution of $Gz = y$ needs to be computationally realized efficiently. The matrix N is often considered to be an approximate inverse to A. [4] This can be motivated by the following *residual (or correction) form* of the linear fixed point iteration (13): After construction (13) is consistent and we have (8)

$$M = I - NA, \tag{14}$$

and hence,

$$x_{k+1} = (I - NA)x_k + Nb = x_k + N(b - Ax_k) = x_k + Nr_k, \tag{15}$$

where $r_k := b - Ax_k$ is the *residual* of the k-th iterate. The iteration given in Eqn. (15) is called the *residual form*. If N is considered to be a good approximation to A^{-1} we indeed have

$$x_k + N(b - Ax_k) = x_k + \underbrace{Nb}_{\approx x_*} - \underbrace{NA}_{\approx I} x_k \approx x_k + x_* - x_k = x_*, \tag{16}$$

which motivates the choice of x_{k+1} in (15).

1.3.3 (Damped) Richardson, (damped) Jacobi, Gauss-Seidel, SOR, SSOR

(Damped) Richardson iteration: The easiest case is the splitting $A = G - H = I - (I - A)$, i.e., the approximate inverse is the identity, $(G^{-1} =)N = I$, and the iteration matrix is $M = I - A$. This leads to the *Richardson iteration*

$$x_{k+1} = x_k + (b - Ax_k) = x_k + r_k. \tag{17}$$

We see from the residual form (15) that every linear iteration can be interpreted as a Richardson iteration applied to the transformed system

$$NAx = Nb. \tag{18}$$

The matrix N is called the *preconditioner* and considered to be a reasonable approximation of the inverse A^{-1}. We emphasize that by choosing the approximate inverse $N = G^{-1}$, different iterative methods can be constructed based on the splitting $A = G - H$.
By choosing a real *relaxation or damping parameter* $\omega \in \mathbb{R}$ we get the damped version of an iteration by setting

$$x_{k+1} = x_k + \omega N(b - Ax_k) = x_k + \omega Nr_k. \tag{19}$$

The damped Richardson iteration corresponds to the choice

$$x_{k+1} = x_k + \omega(b - Ax_k). \tag{20}$$

[4] The inverse of N, an approximation of A itself, is called the preconditioner. See Ch. 1.7 for more details.

One can show that $\sigma(M^{dRich}) = \{1 - \omega\lambda : \lambda \in \sigma(A)\}$ (exercise). Convergence can be achieved by choosing ω in accordance with Th. 1.3.

The splitting methods described below will use the following decomposition of the matrix A:

$$A = D + L + U, \tag{21}$$

where D is the diagonal of A, and L and U the lower and upper part of A, respectively. If A is symmetric, then $L = U^T$.

(Damped) Jacobi iteration: Here $A = G - H = D - (-L - U)$. Thus, we have $N = D^{-1}$ (nonzero diagonal entries are assumed here) and the iteration

$$x_{k+1} = x_k + D^{-1}(b - Ax_k) = x_k + D^{-1}r_k. \tag{22}$$

Since the inversion of the diagonal matrix is trivial we have the explicit component form

$$x_{k+1,i} = \frac{1}{a_{ii}}\left(b_i - \sum_{j\neq i} a_{ij}x_{k,j}\right), \quad i = 1, \ldots, n. \tag{23}$$

The *damped Jacobi iteration* takes the form

$$x_{k+1} = x_k + \omega D^{-1}(b - Ax_k) = x_k + \omega D^{-1}r_k. \tag{24}$$

If A is SPD, then the damped Jacobi iteration converges if and only if $\frac{2}{\omega}D - A$ is positive definite (proof [3]). This is a property in the sense of 'diagonal dominance'. However, the regime of damping parameters which lead to a convergent Jacobi iteration depend on the problem even in the SPD case. This is in contrast to other methods like the so-called *Successive overrelaxation* (SOR), which converges for $\omega \in (0, 2)$ in the SPD case. However, convergence of the Jacobi method is ensured in the strictly diagonally dominant case, that is $|a_{ii}| > \sum_{j\neq i} |a_{ij}|, \quad i = 1, \ldots, n$.

Gauss-Seidel: Choose $A = G - H = (D + L) - (-U)$, which leads to

$$x_{k+1} = x_k + (D + L)^{-1}(b - Ax_k). \tag{25}$$

This is called the *forward Gauss-Seidel* methods; a *backward* version is obtained by choosing $G = D+U$, instead of $G = D + L$. Note that, if the diagonal of A contains no zeros, the triangular matrices $D + L$ or $D + U$ are easily inverted by 'forward substitution' or 'back substitution', respectively. In component form forward Gauss-Seidel reads

$$x_{k+1,i} = \frac{1}{a_{ii}}\left(b_i - \sum_{j=1}^{i-1} a_{ij}x_{k+1,j} - \sum_{j=i+1}^{n} a_{ij}x_{k,j}\right), \quad i = 1, \ldots, n, \tag{26}$$

where the first sum on the r.h.s. contains the already computed components $x_{k+1,j}$ for $j < i$. The component form translates to matrix form again:[5]

$$x_{k+1} = (D + L)^{-1}(b - Ux_k). \tag{27}$$

The convergence property of Gauss-Seidel is contained in that of the next method (SOR), since it is a more general version of Gauss-Seidel. However, convergence of the Gauss-Seidel method is ensured in the strictly diagonal dominant case.

Successive OverRelaxation (SOR): The SOR is a variant of the Gauss-Seidel method that improves convergence rate by using a linear combination of the last iterate x_k and the Gauss-Seidel iterate x_{k+1}

[5] In the backward version L and U interchange.

(Eqn. 26). The method uses the relaxation parameter $\omega \in (0,2)$, which is often chosen heuristically in dependence of the specific problem. The iteration is given by

$$x_{k+1,i} = x^{k,i}(1-\omega) + \frac{\omega}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_{k+1,j} - \sum_{j=i+1}^{n} a_{ij} x_{k,j} \right), \quad i = 1, \ldots, n. \tag{28}$$

This can be recast in matrix notation in residual form as

$$x_{k+1} = x_k + \omega(D + \omega L)^{-1}(b - Ax_k), \tag{29}$$

which gives (forward) Gauss-Seidel for $\omega = 1$.
It can be shown that SOR converges in the SPD case for $\omega \in (0,2)$, [3].

Symmetric SOR (SSOR): Both, Gauss-Seidel and SOR depend on the numbering of the unknowns, in contrast to the Richardson and Jacobi method. Also, the iteration matrix and the approximate inverse is not symmetric even if the original matrix is symmetric. This can be overcome by the symmetric version of SOR, namely SSOR. It is constructed by two half-steps of SOR, one based on forward Gauss-Seidel, the other based on backward Gauss-Seidel. This leads to the iteration

$$x_{k+1} = x_k + \omega(2-\omega)(D + \omega U)^{-1} D (D + \omega L)^{-1}(b - Ax_k). \tag{30}$$

It can be shown that SSOR converges in the SPD case for $\omega \in (0,2)$, [3].

A collection of the different iteration matrices M and approximate inverses N are listed in Fig. 4.

method	iteration matrix $M = \mathrm{I} - NA$	approximate inverse N
damped Richardson	$M^{Rich} = \mathrm{I} - \omega A$	$\omega\,\mathrm{I}$
damped Jacobi	$M_\omega^{Jac} = \mathrm{I} - \omega D^{-1} A$	$\omega\, D^{-1}$
forward Gauss-Seidel	$M^{GS} = \mathrm{I} - (D+L)^{-1} A$	$(D+L)^{-1}$
SOR	$M_\omega^{SOR} = \mathrm{I} - \omega\,(D+\omega L)^{-1} A$	$\omega\,(D+\omega L)^{-1}$
SSOR	$M_\omega^{SSOR} = \mathrm{I} - \omega\,(2-\omega)(D+\omega U)^{-1} D\,(D+\omega L)^{-1} A$	$\omega\,(2-\omega)(D+\omega U)^{-1} D\,(D+\omega L)^{-1}$

Figure 4: Summary iterative methods, taken from [3].

1.3.4 Remarks on optimal damping parameter

In general it is not possible to determine the optimal damping parameter analytically. However, in the case of SPD matrices the optimal damping parameter for the Richardson and Jacobi method takes the form $\omega_{opt} = 2/(\lambda_{min} + \lambda_{max})$, where $\lambda_{min/max}$ is the minimum and maximum eigenvalue of A, respectively. For so-called *consistently ordered* SPD matrices [1] (e.g. K_{2d} Poisson matrix Fig. 1), it is possible to determine the optimal damping parameter for the SOR method analytically by exploiting the so-called *Young theorem*, [1, 3]: $\omega_{opt} = 2/(1 + \sqrt{1-\beta^2})$, $\beta := \rho(M^{Jac})$. In the case of the K_{2d} matrix of the uniformly discretized Laplace operator in two dimensions (cf. Fig. 1) the spectrum of M^{Jac} can be computed analytically, leading to $\beta = \rho(M^{Jac}) = 1 - ch^2 + \mathcal{O}(h^3)$, $c > 0$, where $h = 1/(n+1)$ is the mesh size. In this case the optimal damping parameters for Jacobi and SOR can be computed, as well as the convergence factor (rate) of the optimally damped SOR method, which turns out to be $\rho(M^{SOR,\omega_{opt}}) = 1 - \widetilde{c}h + \mathcal{O}(h^2)$. Note, that the latter spectral radius indicates a significant improvement for the convergence rate compared to Jacobi.
Fig. 5 shows a comparison of the convergence for the $2d$ Poisson problem for different methods. The left figure shows convergence for all methods, whereas in the right figure Jacobi and Gauss-Seidel do not converge practically speaking. However, from the left figure one can observe that Gauss-Seidel converges at twice the convergence rate of Jacobi. [6] The optimally damped SOR is faster in both cases, whereas the CG method is obviously superior here. Convergence gets slower as problem size increases.

[6] For consistently ordered matrices there holds $\rho(M^{GS}) = \rho(M^{Jac})^2$.

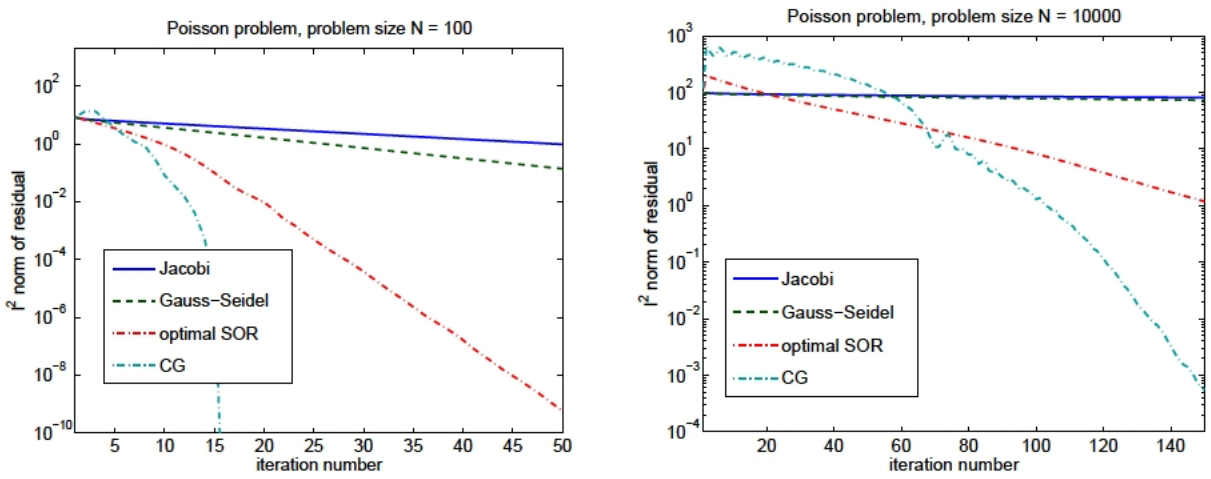

Figure 5: Comparison for the $2d$ Poisson problem for Jacobi, Gauss-Seidel, optimally damped SOR and CG (Conjugate Gradient, see Sec.1.5), taken from [3].

1.4 Gradient methods and steepest descent variants

1.4.1 Gradient methods

We consider here the case of a SPD matrix $A \in \mathbb{R}^{n\times n}$. Gradient methods are based on the observation that critical points of a function correspond to solutions of the linear system.

Theorem 1.8. *Let $A \in \mathbb{R}^{n\times n}$ be SPD. The function*

$$\Phi(x) := \frac{1}{2}x^T Ax - b^T x \tag{31}$$

fulfills[7]

$$\nabla\Phi(x) = Ax - b. \tag{32}$$

Further, the (global) minimizer of Φ, which is given through $\nabla\Phi(x) = 0$, solves the linear system $Ax = b$.

Proof. Looking at the linear parts in the perturbation $\delta \in \mathbb{R}^n$ in the expression $\Phi(x+\delta) - \Phi(x)$ and using the symmetry of A reveals $\nabla\Phi(x) = Ax - b$. Further, since the Hessian of (31) is A and positive definite (by assumption), the function (31) is strictly convex and we conclude that the solution of $Ax = b$ is the unique minimizer of (31). □

The following lemma identifies the minimization of the function Φ as the minimization of the energy error.

Lemma 1.9. *Minimizing* (31) *is equivalent to minimizing the error $x - x_*$ in the energy norm induced by A, that is $\|x - x_*\|_A^2$.*

Proof. We have

$$\begin{aligned}
\Phi(x) - \Phi(x_*) &= \frac{1}{2}x^T Ax - x^T b - \frac{1}{2}x_*^T Ax_* + x_*^T b \\
&= \frac{1}{2}(x-x_*)^T A(x-x_*) + \underbrace{\frac{1}{2}\left(x^T Ax_* + x_*^T Ax\right) - x_*^T Ax_* - x^T b + x_*^T b}_{=0,\ \text{since } Ax_*=b} \\
&= \frac{1}{2}(x-x_*)^T A(x-x_*) \\
&= \frac{1}{2}\|x-x_*\|_A^2.
\end{aligned} \tag{33}$$

[7] If A is not symmetric, there holds $\nabla\Phi(x) = \frac{1}{2}(A + A^T)x - b$.

□

One class of numerical methods for minimization are so-called *line search methods* which take the form

$$x_{k+1} = x_k + \alpha_k d_k, \tag{34}$$

where the update on the r.h.s. consists of the *search direction* $d_k \in \mathbb{R}^n$ and the *step length* $\alpha_k \in \mathbb{R}$. Once a search direction is chosen, the step length is determined by a one-dimensional minimization problem (*line search*)

$$\alpha_k = \arg\min_{\alpha \in \mathbb{R}} \Phi(x_k + \alpha d_k). \tag{35}$$

Lemma 1.10. *In the present case of the quadratic function* Φ *in* (31) *it is possible to solve* (35) *explicitly as*

$$\alpha_k = \frac{(b - Ax_k)^T d_k}{d_k^T A d_k} = \frac{r_k^T d_k}{d_k^T A d_k}. \tag{36}$$

Proof. Exercise. □

Many different search directions could be considered. For instance, if we choose the k-th coordinate unit vectors $d_k = e_k = (0, \dots 0, \underbrace{1}_{k-th}, 0, \dots, 0)^T$ for the k-th update, we have

$$x_{k+1} = x_k + \frac{(r_k)_k}{a_{kk}} d_k, \tag{37}$$

which exactly corresponds to the k-th update in the inner loop of a Gauss-Seidel step, thus, n of these updates, successively applied and with cyclic choice of unit vectors as search directions, result in a single Gauss-Seidel step.
If we choose $d_k = r_k$ but $\alpha_k \equiv 1$ (or ω) we arrive at the (damped) Richardson iteration.

1.4.2 The steepest descent method

In the *steepest descent (SD) method* the step length is locally optimized corresponding to (36) for the choice $d_k = r_k = -\nabla\Phi(x_k)$, i.e.,

$$\alpha_k = \frac{r_k^T r_k}{r_k^T A r_k} = \frac{\|r_k\|_2^2}{\|r_k\|_A^2}. \tag{38}$$

Algorithm 3: Steepest descent for solving $Ax = b$, A SPD.

Data: Matrix $A \in \mathbb{R}^{n\times n}$ SPD, r.h.s. $b \in \mathbb{R}^n$, initial vector $x_0 \in \mathbb{R}^n$
Result: $x \in \mathbb{R}^n$ approximate solution
Initialization: Set $k \leftarrow 0$
while *Convergence criterion not satisfied* **do**

$$r_k \leftarrow b - Ax_k$$
$$\alpha_k \leftarrow \frac{r_k^T r_k}{r_k^T A r_k}$$
$$x_{k+1} \leftarrow x_k + \alpha_k r_k$$
$$k \leftarrow k + 1$$

end
$x \leftarrow x_k$

The choice $d_k = r_k = -\nabla\Phi(x_k)$, namely the negative gradient direction, yields the fastest rate of decrease *locally* (direction of steepest descent).[8] The steepest descent algorithm is given in Alg. 3. We note, that the number of matrix-vector multiplications as given in the pseudo-algorithm Alg. 3 is not optimal, in fact, it can be reduced to just one matrix-vector multiplication inside the loop by precomputing $p_k := Ar_k$ and using the fact $r_{k+1} = r_k - \alpha_k p_k$. *(exercise)*
As *convergence criterion* one can measure the (relative) residual length, i.e., $\|r_k\|$ (or $\|r_k\|/\|b\|$ resp.) and terminate if the error is below a prescribed threshold $0 < \epsilon \ll 1$. The length of successive iterates, i.e., $\|x_{k+1} - x_k\|$ or $\|x_{k+1} - x_k\|_A$, would yield a similar criterion. Furthermore, a maximum iteration number should be prescribed.
It is notable that in the steepest descent method *consecutive search directions are orthogonal* to each other w.r.t. the Euclidean inner product.

Lemma 1.11. *Consecutive search directions in the SD method are orthogonal, that is, $d_{k+1}^T d_k = 0$, see also Fig. 6.*

Proof. This can be seen by observing that $d_{k+1} = b - Ax_{k+1} = b - A(x_k + \alpha_k r_k) = r_k - \alpha_k A r_k$. Inserting the step length (38) yields

$$d_{k+1}^T d_k = (r_k - \alpha_k A r_k)^T r_k = r_k^T r_k - \alpha_k r_k^T A r_k = r_k^T r_k - \frac{r_k^T r_k}{r_k^T A r_k} r_k^T A r_k = 0. \tag{39}$$

□

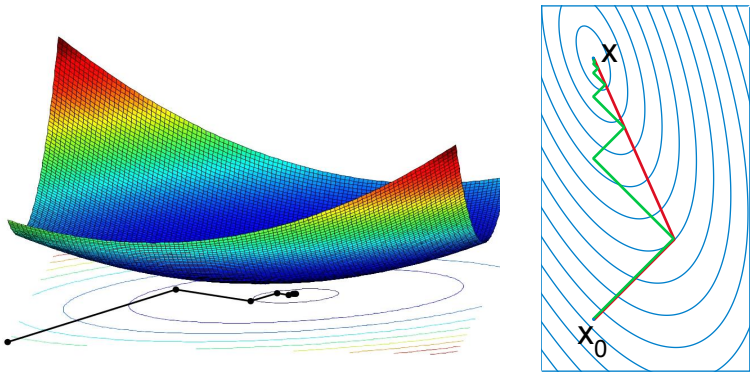

Figure 6: Two dimensional illustration of the convergence process for the minimization of the functional Φ from (31) corr. to a SPD system $Ax = b$. Consecutive search directions (right figure, green lines) in the SD method are orthogonal.

1.4.3 Convergence of the SD method

In the SPD case the SD method converges and the speed of convergence depends on the spectrum of A, more precisely, on the condition number $\kappa_2(A) = \lambda_{max}/\lambda_{min}$. The following theorem estimates the error in the energy norm.

Theorem 1.12. *The SD method applied to the SPD system $Ax = b$ yields the following estimate for the error $\varepsilon_k := x_* - x_k$ in the energy norm*

$$\|\varepsilon_k\|_A \leq \left(\frac{\kappa_2(A) - 1}{\kappa_2(A) + 1}\right)^k \|\varepsilon_0\|_A . \tag{40}$$

Hence, $\varepsilon_k \to 0$ for $k \to \infty$. □

[8] $\nabla\Phi(x_k)$ would yield the direction of steepest ascent.

Ad proof. The proof uses the *Kantorovich inequality* as a first ingredient: Let B be SPD and $n \times n$ with extreme eigenvalues $\lambda_{min}, \lambda_{max} > 0$. Then

$$\frac{\|x\|_B^2 \|x\|_{B^{-1}}^2}{\|x\|^4} \leq \frac{(\lambda_{min} + \lambda_{max})^2}{4\lambda_{max}\lambda_{min}}. \tag{41}$$

Secondly, we have by Lemma 1.9 that $\frac{1}{2}\|\varepsilon_k\|_A^2 = \Phi(x_k) - \Phi(x_*)$. Further, by elementary calculation, one gets for $x_{k+1} = x_k + \alpha_k r_k$ with optimal α_k that $\Phi(x_{k+1}) - \Phi(x_k) = -\frac{1}{2}\frac{\|r_k\|^4}{\|r_k\|_A^2}$. All together this leads to

$$\tfrac{1}{2}\|\varepsilon_{k+1}\|_A^2 = \Phi(x_{k+1}) - \Phi(x_*) = \Big(\Phi(x_{k+1}) - \Phi(x_k)\Big) + \Big(\Phi(x_k) - \Phi(x_*)\Big) = -\tfrac{1}{2}\frac{\|r_k\|^4}{\|r_k\|_A^2} + \tfrac{1}{2}\|\varepsilon_k\|_A^2. \tag{42}$$

By the Kantorovich inequality (41) we have

$$\frac{\|r_k\|^4}{\|r_k\|_A^2} \geq \frac{4\lambda_{max}\lambda_{min}}{(\lambda_{min} + \lambda_{max})^2} \|r_k\|_{A^{-1}}^2 = \frac{4\lambda_{max}\lambda_{min}}{(\lambda_{min} + \lambda_{max})^2} \|\varepsilon_k\|_A^2. \tag{43}$$

We can therefore simplify (42) to

$$\|\varepsilon_{k+1}\|_A^2 \leq \Big(1 - \frac{4\lambda_{max}\lambda_{min}}{(\lambda_{min} + \lambda_{max})^2}\Big) \|\varepsilon_k\|_A^2 = \Big(\frac{\lambda_{max} - \lambda_{min}}{\lambda_{min} + \lambda_{max}}\Big)^2 \|\varepsilon_k\|_A^2 = \Big(\frac{\kappa_2(A) - 1}{\kappa_2(A) + 1}\Big)^2 \|\varepsilon_k\|_A^2. \tag{44}$$

□

If the condition number is large, the contour lines of the quadratic functional Φ in (31) are elongated ellipses and the orthogonality of the consecutive search directions yields an inefficient 'zig-zag' path, see Fig. 6 where a two-dimensional ($n = 2$) example is illustrated.

1.4.4 Variants of the SD method

Residual Norm Steepest Descent. The SD method described above requires A to be SPD. If $A \in \mathbb{R}^{n\times n}$ is nonsingular but not symmetric one can rewrite the problem to the (in the present case) mathematically equivalent *normal equations* [9]

$$A^T Ax = A^T b, \tag{45}$$

and apply the steepest descent method (Alg. 3) to the system (45). It is called the *residual norm steepest descent method* [1] because it minimizes the residual norm $\|b - Ax\|_2^2$. In fact, if one defines for nonsingular square matrix A the quadratic functional

$$\widetilde{\Phi}(x) = \frac{1}{2}\|b - Ax\|_2^2, \tag{46}$$

its gradient is $\nabla\widetilde{\Phi}(x) = A^T Ax - A^T b$, thus critical points solve (45). Since, $A^T A$ is SPD iff A is nonsingular, the minimizer is unique and coincides with the solution of $Ax = b$.
Often the condition number of $A^T A$ is much larger[10] than that of A, which makes methods via the normal equation often numerically inefficient. We will overcome this by the *generalized minimal residual* (GMRES) method in chapter 1.6.

Minimal Residual (MR) Iteration. One variant is to use the steepest descent direction for the functional (31) but the step length is chosen in a line search to minimize the squared norm of the new residual $\|b - Ax_{k+1}\|_2^2 = \|b - A(x_k + \alpha_k r_k)\|_2^2 = \|r_k - \alpha_k Ar_k\|_2^2$, resulting in $\alpha_k = r_k^T Ar_k / \|Ar_k\|_2^2$. One can show that this iteration converges if the matrix is not necessarily symmetric [1]. A general version *GMRES* for only nonsingular requirement will be discussed in Ch. 1.6.

[9] Known from linear least squares problems for overdetermined systems (regression).
[10] Typically near the squared value $\kappa_2(A)^2$.