Inverse Problems

Sommersemester 2022

Vesa Kaarnioja
vesa.kaarnioja@fu-berlin.de

FU Berlin, FB Mathematik und Informatik

Fourth lecture, May 16, 2022

Krylov subspace methods

Krylov subspace methods are iterative solvers for (large scale) matrix equations of the form $Ax = y$, $A \in \mathbb{R}^{n \times n}$. In general terms, the solution vector $x \in \mathbb{R}^n$ is approximated as a linear combination of vectors of the form u, Au, A^2u, $\ldots$, with some given $u \in \mathbb{R}^n$. If multiplication by A is cheap – for example, when A is sparse – Krylov subspace methods can be particularly efficient.

We consider only the most well-known Krylov subspace method, the conjugate gradient method. It is worth mentioning that other methods in this class include, e.g., the generalized minimum residual method (GMRES) and the biconjugate gradient method (BiCG).

Assumptions on A and A-dependent inner product

In what follows, we assume that the system matrix $A \in \mathbb{R}^{n\times n}$ is symmetric and positive definite:

$$A^{\mathrm{T}} = A \quad \text{and} \quad u^{\mathrm{T}} A u > 0 \quad \text{for all } u \in \mathbb{R}^n \setminus \{0\}.$$

Note that this implies that A is injective.[†] By the fundamental theorem of linear algebra, A is invertible. Furthermore, the inverse $A^{-1} \in \mathbb{R}^{n\times n}$ is also symmetric and positive definite.

We define

$$\langle u, v\rangle_A := u^{\mathrm{T}} A v \quad \text{and} \quad \|u\|_A := \sqrt{\langle u, u\rangle_A}.$$

Since A was assumed to be symmetric and positive definite, it is straightforward to check that $\langle \cdot, \cdot \rangle_A \colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ defines an inner product on $\mathbb{R}^n$. In consequence, $\|\cdot\|_A \colon \mathbb{R}^n \to \mathbb{R}$ is a norm.

[†] $Ax = Ay \Rightarrow A(x-y) = 0 \Rightarrow (x-y)^{\mathrm{T}} A(x-y) = 0 \Rightarrow x - y = 0.$

Error, residual, and minimization problem

Let $x_* = A^{-1}y \in \mathbb{R}^n$ denote the unique solution of the equation

$$Ax = y$$

for a given $y \in \mathbb{R}^n$. We define the error and residual corresponding to some approximate solution $x \in \mathbb{R}^n$ by

$$e = x_* - x \quad \text{and} \quad r = y - Ax = Ae.$$

Let $\phi\colon \mathbb{R}^n \to \mathbb{R}$ be the A-dependent quadratic functional

$$\phi(x) = \|e\|_A^2 = e^{\mathrm{T}}Ae = r^{\mathrm{T}}A^{-1}r = \|r\|_{A^{-1}}^2.$$

Since $\|\cdot\|_A$ is a norm, $\phi(x) \geq 0$ for all $x \in \mathbb{R}^n$ and

$$\phi(x) = 0 \quad \Leftrightarrow \quad e = 0 \quad \Leftrightarrow \quad x = x_*.$$

Minimizing ϕ is equivalent to solving $Ax = y$.

Minimizing ϕ in a given direction

We cannot directly evaluate the functional ϕ since this would require knowledge of the unknown x_* (or, equivalently, A^{-1}). Since our goal is to approximate the solution x_* iteratively, assuming that it is known is out of the question.

Fortunately, it turns out that if we have some initial guess $x_0 \in \mathbb{R}^n$ and some search direction $0 \neq s_0 \in \mathbb{R}^n$, we can find the minimizer of ϕ over the line

$$\mathcal{S}_0 = \{x \in \mathbb{R}^n \mid x = x_0 + \alpha s_0,\ \alpha \in \mathbb{R}\}.$$

without having to evaluate ϕ or having knowledge of x_*.

Lemma

Let $0 \neq s_0 \in \mathbb{R}^n$ be a search direction. The minimum of

$$\mathbb{R} \ni \alpha \mapsto \phi(x_0 + \alpha s_0) \in \mathbb{R}$$

is attained at

$$\alpha = \alpha_0 := \frac{s_0^{\mathrm{T}} r_0}{\|s_0\|_A^2} = \frac{s_0^{\mathrm{T}} r_0}{s_0^{\mathrm{T}} A s_0},$$

where $r_0 := y - Ax_0$ is the residual corresponding to the initial guess $x_0 \in \mathbb{R}^n$.

Proof: The residual corresponding to $x = x_0 + \alpha s_0$ is

$$r = y - Ax = y - Ax_0 - \alpha A s_0 = r_0 - \alpha A s_0$$

and thus

$$\begin{aligned} \phi(x) = r^{\mathrm{T}} A^{-1} r &= (r_0 - \alpha A s_0)^{\mathrm{T}} A^{-1} (r_0 - \alpha A s_0) \\ &= r_0^{\mathrm{T}} A^{-1} r_0 - 2\alpha s_0^{\mathrm{T}} r_0 + \alpha^2 s_0^{\mathrm{T}} A s_0. \end{aligned}$$

Since $s_0^{\mathrm{T}} A s_0 > 0$, this is a parabola that opens upward as a function of α. The minimum is found at the zero of the derivative w.r.t. α, i.e., $-2 s_0^{\mathrm{T}} r_0 + 2\alpha s_0^{\mathrm{T}} A s_0 = 0 \Leftrightarrow \alpha = \frac{s_0^{\mathrm{T}} r_0}{s_0^{\mathrm{T}} A s_0}$. □

Sequential minimization of ϕ

Suppose that we are given a sequence of non-zero search directions $\{s_k\} \subset \mathbb{R}^n$. Then we can produce a sequence of approximate solutions by first choosing x_0 and then iteratively minimizing ϕ on the line passing through x_k in the direction of s_k as follows:

$$x_{k+1} = x_k + \alpha_k s_k, \quad \text{with } \alpha_k = \frac{s_k^{\mathrm{T}} r_k}{s_k^{\mathrm{T}} A s_k}, \quad k = 0, 1, \ldots,$$

where $r_k = y - Ax_k$ is the residual corresponding to the k^{th} iterate.

By construction, $\phi(x_{k+1}) \leq \phi(x_k)$, so $\{\phi(x_k)\}$ is a decreasing sequence of real numbers.

As yet, it is not obvious how to choose the search directions $\{s_k\}$ efficiently. The strategy will be to first consider minimization of ϕ over a hyperplane, and then seek a construction of $\{s_k\}$ for which the sequential minimization strategy coincides with minimization over a hyperplane.

Minimization of ϕ over a hyperplane

Let $\{s_0, \ldots, s_k\}$ be a set of linearly independent search directions. We consider minimization of ϕ on the hyperplane

$$\begin{aligned}\mathcal{S}_k &= \{x \in \mathbb{R}^n \mid x = x_0 + h_0 s_0 + \cdots + h_k s_k,\ h_0, \ldots, h_k \in \mathbb{R}\} \\ &= \{x \in \mathbb{R}^n \mid x = x_0 + S_k h,\ h \in \mathbb{R}^{k+1}\},\end{aligned}$$

where $x_0 \in \mathbb{R}^n$ is the initial guess and $S_k = [s_0, \ldots, s_k] \in \mathbb{R}^{n\times(k+1)}$.

Lemma

Let $\{s_0, \ldots, s_k\}$ be a set of linearly independent search directions. The function

$$\mathbb{R}^{k+1} \ni h \mapsto \phi(x_0 + S_k h) \in \mathbb{R}$$

attains its minimum at

$$h = h_* = (S_k^{\mathrm{T}} A S_k)^{-1} S_k^{\mathrm{T}} r_0,$$

where $r_0 = y - Ax_0$ is the residual corresponding to the initial guess $x_0 \in \mathbb{R}^n$.

Proof: We wish to show that $h_* = (S_k^{\mathrm{T}} A S_k)^{-1} S_k^{\mathrm{T}} r_0$ satisfies

$$h_* = \operatorname*{arg\,min}_{h \in \mathbb{R}^{k+1}} \phi(x_0 + S_k h),$$

where $S_k = [s_0, \ldots, s_k] \in \mathbb{R}^{n \times (k+1)}$ contains the search directions.

To show that the expression for h_* is well-defined, let us first show that $S_k^{\mathrm{T}} A S_k \in \mathbb{R}^{(k+1)\times(k+1)}$ is invertible. By the positive definiteness of A,

$$S_k^{\mathrm{T}} A S_k z = 0 \quad \Rightarrow z^{\mathrm{T}} S_k^{\mathrm{T}} A S_k z = 0 \quad \Rightarrow \|S_k z\|_A^2 = 0 \quad \Rightarrow S_k z = 0,$$

which means that $z = 0$ since the columns of S_k are linearly independent. Hence $S_k^{\mathrm{T}} A S_k$ is injective, and $(S_k^{\mathrm{T}} A S_k)^{-1}$ exists by the fundamental theorem of linear algebra.

The residual corresponding to $x = x_0 + S_k h$ satisfies

$$r = y - A(x_0 + S_k h) = r_0 - A S_k h,$$

thus (recall that $\phi(x) = r^{\mathrm{T}} A^{-1} r$ for $r = y - Ax$)

$$\begin{aligned} \phi(x_0 + S_k h) &= (r_0 - A S_k h)^{\mathrm{T}} A^{-1} (r_0 - A S_k h) \\ &= r_0^{\mathrm{T}} A^{-1} r_0 - 2 r_0^{\mathrm{T}} S_k h + h^{\mathrm{T}} S_k^{\mathrm{T}} A S_k h. \end{aligned}$$

We obtained

$$\phi(x_0 + S_k h) = r_0^{\mathrm{T}} A^{-1} r_0 - 2 r_0^{\mathrm{T}} S_k h + h^{\mathrm{T}} S_k^{\mathrm{T}} A S_k h.$$

The Hessian of $h \mapsto \phi(x_0 + S_k h)$ is $2 S_k^{\mathrm{T}} A S_k$, which is positive definite since

$$u^{\mathrm{T}} (S_k^{\mathrm{T}} A S_k) u = (S_k u)^{\mathrm{T}} A (S_k u) \geq 0 \quad \text{for all } u \in \mathbb{R}^{k+1},$$

where equality holds iff $S_k u = 0 \Leftrightarrow u = 0$. Hence $h \mapsto \phi(x_0 + S_k h)$ is convex, and we can find its unique minimizer by solving the zero point of its gradient:

$$\begin{aligned} 0 = \nabla_h \phi(x_0 + S_k h) &= 2 S_k^{\mathrm{T}} A S_k h - 2 S_k^{\mathrm{T}} r_0 \\ \Leftrightarrow \quad h &= (S_k^{\mathrm{T}} A S_k)^{-1} S_k^{\mathrm{T}} r_0. \quad \square \end{aligned}$$

Numerical example

Let us consider minimization with the *steepest descent* directions

$$s_k = -\nabla \phi(x_k) = 2(y - Ax_k), \quad k = 0, 1, \ldots . \tag{1}$$

In general, the convergence of the sequence $\{x_k\}$ toward the global minimizer $x_* = A^{-1}y$ can be fairly slow. We demonstrate this with the following example.

Let

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 5 \end{bmatrix} \quad \text{and} \quad y = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Now

$$\phi(x) = x_1^2 + 5x_2^2.$$

We plot the level contours of ϕ and the sequence $\{x_k\}_{k=0}^{5}$ starting from $x_0 = (1, 0.3)^{\mathrm{T}}$. The true solution $x_* = (0, 0)^{\mathrm{T}}$ is marked with a blue cross.

We also illustrate minimization over the hyperplanes $\mathcal{S}_0$ and $\mathcal{S}_1$, i.e., $x_0 + S_0 h_*$ and $x_0 + S_1 h_*$ with $S_0 = [s_0] \in \mathbb{R}^{2\times 1}$ and $S_1 = [s_0, s_1] \in \mathbb{R}^{2\times 2}$, where s_0 and s_1 were computed using the sequential method (1).

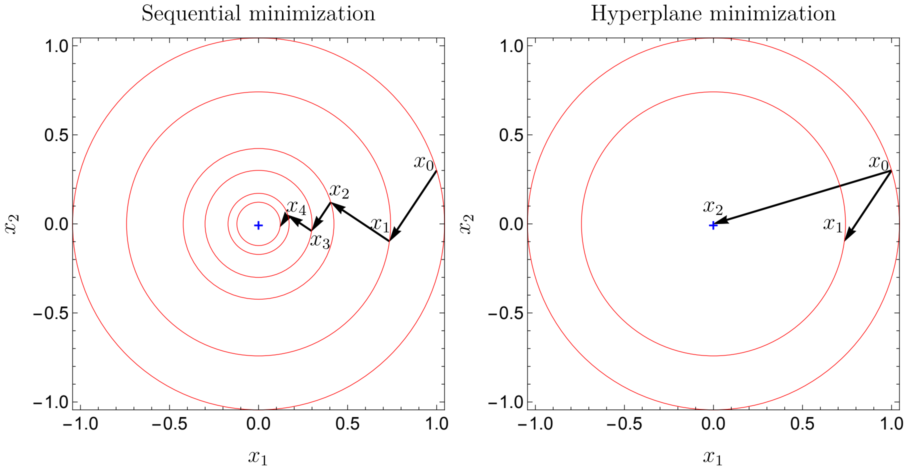


Figure: Left: Minimization using steepest descent search directions $s_k = -\nabla\phi(x_k)$ and the sequential minimization technique. Right: Minimization over the hyperplanes $\mathcal{S}_0$ and $\mathcal{S}_1$ spanned by the steepest descent directions from the left picture. Notably, the hyperplane method converges to the actual solution $x_* = (0,0)^{\mathrm{T}}$ (marked with a blue cross) when $k = n = 2$.

Finding a minimizer of ϕ over the hyperplane

$$\mathcal{S}_k = \{x \in \mathbb{R}^n \mid x = x_0 + S_k h,\ \ h \in \mathbb{R}^{k+1}\}$$

generally involves inverting a $(k+1) \times (k+1)$ matrix, which we would like to avoid.

On the other hand, minimizing ϕ sequentially over the directions $s_0, \ldots, s_k$ may not result in as good approximation as doing the minimization over the whole hyperplane $\mathcal{S}_k$ at once.

However, it turns out that sequential minimization can be used to produce the minimizer over $\mathcal{S}_k$ as long as the search directions $\{s_0, \ldots, s_k\}$ are chosen in a clever way.

Goal: *choose* $\{s_0, \ldots, s_k\}$ *so that each iteration of the sequential minimization algorithm coincides with the minimizer over the respective hyperplanes* $\mathcal{S}_0$, …, $\mathcal{S}_k$.

A-conjugate search directions

We say that non-zero vectors $\{s_0, \ldots, s_k\} \subset \mathbb{R}^n$ are A-conjugate if

$$\langle s_i, s_j \rangle_A = s_i^{\mathrm{T}} A s_j = 0 \quad \text{whenever } i \neq j.$$

That is, $\{s_0, \ldots, s_k\}$ are A-conjugate if they are orthogonal with respect to the inner product $\langle \cdot, \cdot \rangle_A$.

We can represent the A-conjugacy condition compactly using the matrix $S_k = [s_0, \ldots, s_k] \in \mathbb{R}^{n \times (k+1)}$:

$$S_k^{\mathrm{T}} A S_k = \begin{bmatrix} s_0^{\mathrm{T}} \\ \vdots \\ s_k^{\mathrm{T}} \end{bmatrix} [As_0, \ldots, As_k] = \mathrm{diag}(d_0, \ldots, d_k) \in \mathbb{R}^{(k+1) \times (k+1)}, \quad (2)$$

where $d_j = s_j^{\mathrm{T}} A s_j > 0$, $j = 0, \ldots, k$, since A was assumed to be positive definite.

The following theorem provides the connection between sequential minimization and minimization over hyperplanes, when the search directions are chosen to be A-conjugate.

Theorem

Let $x_0 \in \mathbb{R}^n$ be an initial guess and suppose that the search directions $\{s_0, \ldots, s_k\} \subset \mathbb{R}^n$ are non-zero and A-conjugate. Then the sequential minimizer of ϕ over these directions, i.e., $x_{k+1} \in \mathbb{R}^n$ obtained by the iteration

$$x_{j+1} = x_j + \alpha_j s_j, \quad \textit{with } \alpha_j = \frac{s_j^{\mathrm{T}} r_j}{s_j^{\mathrm{T}} A s_j}, \quad j = 0, \ldots, k,$$

is the minimizer of ϕ on the hyperplane

$$\mathcal{S}_k = \{x \in \mathbb{R}^n \mid x = x_0 + S_k h, \ h \in \mathbb{R}^{k+1}\},$$

where $S_k = [s_0, \ldots, s_k] \in \mathbb{R}^{n \times (k+1)}$. That is to say,

$$x_{k+1} = x_0 + S_k h_* = x_0 + S_k (S_k^{\mathrm{T}} A S_k)^{-1} S_k^{\mathrm{T}} r_0,$$

where $r_0 = y - A x_0$ is the residual corresponding to the initial guess x_0.

Proof. Let $a_j = (\alpha_0, \ldots, \alpha_j)^{\mathrm{T}} \in \mathbb{R}^{j+1}$, where $\alpha_j = \frac{s_j^{\mathrm{T}} r_j}{s_j^{\mathrm{T}} A s_j}$ are the line search parameters of the sequential minimization algorithm. Then

$$x_j = x_0 + \sum_{i=0}^{j-1} \alpha_i s_i = x_0 + S_{j-1} a_{j-1}, \quad j = 1, \ldots, k+1.$$

The residual corresponding to x_j is

$$r_j = y - A x_j = (y - A x_0) - A S_{j-1} a_{j-1} = r_0 - A S_{j-1} a_{j-1}$$

and hence

$$s_j^{\mathrm{T}} r_j = s_j^{\mathrm{T}} r_0 - s_j^{\mathrm{T}} A S_{j-1} a_{j-1} = s_j^{\mathrm{T}} r_0 - \underbrace{s_j^{\mathrm{T}} [A s_0, \ldots, A s_{j-1}]}_{=0} a_{j-1},$$

since $s_j^{\mathrm{T}} A s_i$, $i < j$, due to A-conjugacy. Thus we obtain the simplified expression

$$\alpha_j = \frac{s_j^{\mathrm{T}} r_j}{s_j^{\mathrm{T}} A s_j} = \frac{s_j^{\mathrm{T}} r_0}{s_j^{\mathrm{T}} A s_j}, \quad j = 0, \ldots, k.$$

When the search directions are A-conjugate, we obtained for the line search parameters of the sequential minimization parameter that

$$\alpha_j = \frac{s_j^{\mathrm{T}} r_j}{s_j^{\mathrm{T}} A s_j} = \frac{s_j^{\mathrm{T}} r_0}{s_j^{\mathrm{T}} A s_j}, \quad j = 0, \ldots, k.$$

On the other hand, since $\{s_0, \ldots, s_k\}$ are A-conjugate, we have that

$$\begin{aligned}(S_k^{\mathrm{T}} A S_k)^{-1} &= \mathrm{diag}(s_0^{\mathrm{T}} A s_0, \ldots, s_k^{\mathrm{T}} A s_k)^{-1} \qquad \text{(cf. (2))}\\ &= \mathrm{diag}\left(\frac{1}{s_0^{\mathrm{T}} A s_0}, \ldots, \frac{1}{s_k^{\mathrm{T}} A s_k}\right).\end{aligned}$$

Especially, this means that the the minimizer h^* of $\phi(x_0 + S_k h)$ over the hyperplane $\mathcal{S}_k$ is given by

$$h^* = (S_k^{\mathrm{T}} A S_k)^{-1} S_k^{\mathrm{T}} r_0 = \mathrm{diag}\left(\frac{1}{s_0^{\mathrm{T}} A s_0}, \ldots, \frac{1}{s_k^{\mathrm{T}} A s_k}\right) \begin{bmatrix} s_0^{\mathrm{T}} r_0 \\ \vdots \\ s_k^{\mathrm{T}} r_0 \end{bmatrix} = \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_k \end{bmatrix} = a_k.$$

In consequence, $x_{k+1} = x_0 + S_k a_k = x_0 + S_k h_*$. □

If the search directions are chosen to be A-conjugate, the residuals satisfy a useful geometric property.

Corollary

If the non-zero search directions $\{s_j\}_{j=0}^{k} \subset \mathbb{R}^n$ are A-conjugate, then the residual $r_{k+1} = y - Ax_{k+1}$ satisfies

$$r_{k+1} \perp \mathrm{span}\{s_0, \dots, s_k\},$$

where the orthogonality is in the sense of the standard Euclidean dot product $\langle z, w \rangle = z^{\mathrm{T}} w$.

Proof. Since $x_{k+1} = x_0 + S_k h_*$, it holds that

$$r_{k+1} = (y - Ax_0) - AS_k h_* = r_0 - AS_k h_*.$$

In consequence,

$$[r_{k+1}^{\mathrm{T}} s_0, \dots, r_{k+1}^{\mathrm{T}} s_k] = r_{k+1}^{\mathrm{T}} S_k = r_0^{\mathrm{T}} S_k - h_*^{\mathrm{T}} S_k^{\mathrm{T}} A S_k = 0,$$

because $h_*^{\mathrm{T}} = ((S_k^{\mathrm{T}} A S_k)^{-1} S_k^{\mathrm{T}} r_0)^{\mathrm{T}} = r_0^{\mathrm{T}} S_k (S_k^{\mathrm{T}} A S_k)^{-1}$. □

Construction of A-conjugate search directions

There are many ways to construct a set of A-conjugate search directions. We obtain the conjugate gradient algorithm with the following choice of Krylov subspaces.

Definition

The k^{th} Krylov subspace of A with the initial vector $r_0 = y - Ax_0$ is defined as

$$\mathcal{K}_k = \mathcal{K}_k(A, r_0) = \operatorname{span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}, \quad k = 1, 2, \ldots .$$

Note that $A(\mathcal{K}_k) \subset \mathcal{K}_{k+1}$. Furthermore,

- $\mathcal{K}_{k-1} \subset \mathcal{K}_k$ (Krylov subspaces are nested).
- $\dim \mathcal{K}_k \leq k$ (dimension of the k^{th} Krylov subspace is at most k).
- $\dim \mathcal{K}_k \leq \dim \mathcal{K}_{k-1} + 1$ (dimension of the successive Krylov is at most one higher than that of the former).

N.B. If r_0 is an eigenvector of A, then $\dim \mathcal{K}_k = 1$ for all $k \geq 1$. However, it turns out that this will not be an issue.

Construction of the conjugate gradient algorithm

We construct a sequence of A-conjugate search directions inductively.
Idea: given a set of A-conjugate search directions, we either construct a new A-conjugate search direction or the previous iterate is already the global minimizer x_*, i.e., the unique solution of $Ax = y$.

1. Choose an initial guess $x_0 \in \mathbb{R}^n$.
2. If $r_0 = y - Ax_0 = 0$, then $x_* = x_0$ and we are done. Otherwise, set $s_0 = r_0$. Then the single search direction $\{s_0\}$ is automatically A-conjugate and $\mathcal{K}_1 = \operatorname{span}\{s_0\} = \operatorname{span}\{r_0\}$.
3. Suppose that we have non-zero and A-conjugate search directions $\{s_j\}_{j=0}^{k-1}$, $k \geq 1$, such that

   $$\mathcal{K}_m = \operatorname{span}\{s_0, \ldots, s_{m-1}\} = \operatorname{span}\{r_0, \ldots, r_{m-1}\}, \ m = 1, \ldots, k, \quad (3)$$

   where $r_j = y - Ax_j$, $j = 0, \ldots, k-1$, are residuals corresponding to the iterates $\{x_j\}_{j=0}^{k-1}$ of the sequential minimization algorithm.
   If $r_k = 0$, then $x_* = x_k$ and we are done. Otherwise, we try to choose another non-zero A-conjugate search direction $s_k \in \mathbb{R}^n$ such that (3) remains valid with k replaced by $k+1$.

Suppose that $r_k \neq 0$. Then

$$r_k = y - Ax_k = y - A(x_{k-1} + \alpha_{k-1} s_{k-1}) = r_{k-1} - \alpha_{k-1} A s_{k-1},$$

where $r_{k-1}, s_{k-1} \in \mathcal{K}_k$ by assumption, and the new residual $r_k \in \mathcal{K}_{k+1}$. Moreover, $r_k \perp \{s_0, \ldots, s_{k-1}\}$ (recall the corollary about residuals from earlier) and $\mathcal{K}_k = \mathrm{span}\{s_0, \ldots, s_{k-1}\}$, it must hold that

$$\mathcal{K}_{k+1} = \mathrm{span}\{s_0, \ldots, s_{k-1}, r_k\} = \mathrm{span}\{r_0, \ldots, r_{k-1}, r_k\}.$$

We seek the new search direction s_k via the ansatz

$$s_k = r_k + \beta_{k-1} s_{k-1}, \quad \beta_{k-1} \in \mathbb{R}.$$

Evidently, $s_k \in \mathcal{K}_{k+1}$ and, moreover,

$$\mathcal{K}_{k+1} = \mathrm{span}\{s_0, \ldots, s_{k-1}, r_k\} = \mathrm{span}\{s_0, \ldots, s_{k-1}, s_k\}.$$

We can solve the undetermined coefficient β_{k-1} by enforcing the A-conjugacy condition.

We want to choose $\beta_{k-1} \in \mathbb{R}$ in $s_k = r_k + \beta_{k-1} s_{k-1}$ so that

$$\begin{aligned} 0 = s_j^{\mathrm{T}} A s_k &= s_j^{\mathrm{T}} A r_k + \beta_{k-1} s_j^{\mathrm{T}} A s_{k-1} \\ &= (A s_j)^{\mathrm{T}} r_k + \beta_{k-1} s_j^{\mathrm{T}} A s_{k-1} \end{aligned} \tag{4}$$

for all $j = 0, \dots, k-1$. Since $\{s_0, \dots, s_{k-2}\} \subset \mathcal{K}_{k-1}$, we have

$$\{A s_0, \dots, A s_{k-2}\} \subset \mathcal{K}_k = \operatorname{span}\{s_0, \dots, s_{k-1}\},$$

and thus the vectors $\{A s_0, \dots, A s_{k-2}\}$ are orthogonal to r_k (again, recall the corollary about residuals from earlier). Thus (4) is satisfied automatically for $j = 0, \dots, k-2$ and we only need to ensure that the case corresponding to $j = k-1$ is satisfied.

Solving the remaining equation for β_{k-1} results in

$$s_{k-1}^{\mathrm{T}} A r_k + \beta_{k-1} s_{k-1}^{\mathrm{T}} A s_{k-1} = 0 \quad \Leftrightarrow \quad \beta_{k-1} = -\frac{s_{k-1}^{\mathrm{T}} A r_k}{s_{k-1}^{\mathrm{T}} A s_{k-1}}.$$

Thus we obtain the update rule

$$s_k = r_k + \beta_{k-1} s_{k-1}, \quad \beta_{k-1} = -\frac{s_{k-1}^{\mathrm{T}} A r_k}{s_{k-1}^{\mathrm{T}} A s_{k-1}}.$$

Putting everything together: preliminary conjugate gradient algorithm

Let $A \in \mathbb{R}^{n\times n}$ be a symmetric and positive definite matrix. The solution of the system $Ax = y$ is the minimizer of the quadratic functional $\phi(x)$ defined earlier. We can proceed as follows:

1. Let $x_0 \in \mathbb{R}^n$ be an initial guess.
2. Set $k = 0$, $r_0 = y - Ax_0$, and $s_0 = r_0$. Note that $\mathcal{K}_1 = \mathrm{span}\{s_0\} = \mathrm{span}\{r_0\}$ is trivially A-conjugate.

   Repeat until the chosen stopping criterion is satisfied:

   3. The minimizer of ϕ in hyperplane $\mathcal{K}_{k+1}$ is given by the line search step

   $$x_{k+1} = x_k + \alpha_k s_k, \quad \alpha_k = \frac{s_k^{\mathrm{T}} r_k}{s_k^{\mathrm{T}} A s_k}. \tag{5}$$

      (Recall that as long as $\{s_0, \dots, s_k\}$ are A-conjugate, the sequential minimization algorithm produces the minimizer in hyperplane $\mathcal{K}_{k+1}$.)

   4. Update residual $r_{k+1} = y - Ax_{k+1} = y - Ax_k - \alpha_k A s_k = r_k - \alpha_k A s_k$.
   5. The next A-conjugate search direction is given by the update

   $$s_{k+1} = r_{k+1} + \beta_k s_k, \quad \beta_k = -\frac{s_k^{\mathrm{T}} A r_{k+1}}{s_k^{\mathrm{T}} A s_k}. \tag{6}$$

   6. Set $k \leftarrow\ k+1$.

   end

The conjugate gradient algorithm is usually presented in slightly different form. Assuming that the iteration has not yet converged at the iterate x_k, we can deduce the following formulae for (5) and (6).

Simplifying (5): Since $r_k \perp s_{k-1}$, we have that

$$s_k^{\mathrm{T}} r_k \overset{(6)}{=} (r_k + \beta_{k-1} s_{k-1})^{\mathrm{T}} r_k = \|r_k\|^2 \quad \Rightarrow \quad \alpha_k \overset{(5)}{=} \frac{\|r_k\|^2}{s_k^{\mathrm{T}} A s_k}. \tag{7}$$

Simplifying (6): since $r_{k+1} \perp \operatorname{span}\{s_0, \ldots, s_k\} = \mathcal{K}_{k+1} \ni r_k$ (corollary on residuals with A-conjugate directions) and $r_{k+1} = r_k - \alpha_k A s_k$ (see step 4 on previous slide), then

$$\|r_{k+1}\|^2 = r_{k+1}^{\mathrm{T}}(r_k - \alpha_k A s_k) \overset{(7)}{=} -\frac{\|r_k\|^2}{s_k^{\mathrm{T}} A s_k} r_{k+1}^{\mathrm{T}} A s_k \overset{(6)}{=} \beta_k \|r_k\|^2$$

and thus

$$\beta_k = \frac{\|r_{k+1}\|^2}{\|r_k\|^2}.$$

We can plug these formulae for α_k and β_k into the preliminary conjugate gradient algorithm, which leads to the “standard form” of the method.

Pseudocode for the conjugate gradient algorithm

```
Given: symmetric, positive definite system matrix $A \in \mathbb{R}^{n\times n}$,
data $y \in \mathbb{R}^n$.
  1. Choose initial guess $x_0 \in \mathbb{R}^n$.
  2. Set $k = 0$, $r_0 = y - Ax_0$, $s_0 = r_0$;
     Repeat until the chosen stopping rule is satisfied:
        3. $\alpha_k = \|r_k\|^2/(s_k^{\mathrm{T}} A s_k)$;
        4. $x_{k+1} = x_k + \alpha_k s_k$;
        5. $r_{k+1} = r_k - \alpha_k A s_k$;
        6. $\beta_k = \|r_{k+1}\|^2/\|r_k\|^2$;
        7. $s_{k+1} = r_{k+1} + \beta_k s_k$;
        8. $k \leftarrow k + 1$;
     end
```

Let us revisit the simple optimization example from earlier.

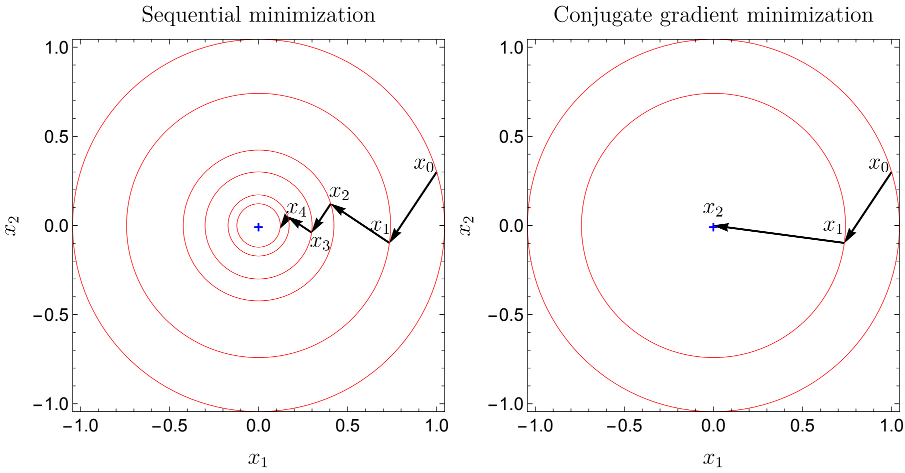


Figure: Left: Minimization using steepest descent search directions $s_k = -\nabla\phi(x_k)$. Right: In the linear case, the conjugate gradient method iteratively finds the optima over the Krylov subspaces $\mathcal{K}_1$ and $\mathcal{K}_2$. The CG method converges to the actual solution $x_* = (0,0)^{\mathrm{T}}$ (marked with a blue cross) when $k = n = 2$.

Conjugate gradient method for inverse problems

According to the previous construction, if the conjugate gradient method is applied to the equation

$$Ax = y,$$

where $A \in \mathbb{R}^{n \times n}$ is symmetric and positive definite, an exact solution (up to rounding errors) is achieved in at most n iteration steps, i.e., $x_n = x_* = A^{-1}y$. However, the algorithm typically converges satisfactorily much quicker. A (pessimistic) convergence rate is proved in the first exercise of week 4.

With ill-posed problems, one should be more cautious and terminate the iterations well before convergence to avoid fitting the solution to noise. In fact, since the conjugate gradient method often converges very fast, one should be extremely cautious.

Let us consider a general ill-posed matrix equation

$$Ax = y,$$

where $A \in \mathbb{R}^{m \times n}$ and $y \in \mathbb{R}^m$ are given.

- If $m = n$ and there is some available prior information suggesting that A is, at least in theory, positive (semi-)definite, one can apply the conjugate gradient algorithm directly on the original equation.
- More generally, one may still consider the normal equation

  $$A^{\mathrm{T}} Ax = A^{\mathrm{T}} y,$$

  which corresponds to solving the original equation in the sense of least squares.

The system matrix $A^{\mathrm{T}}A = (A^{\mathrm{T}}A)^{\mathrm{T}} \in \mathbb{R}^{n\times n}$ is symmetric and

$$u^{\mathrm{T}}A^{\mathrm{T}}Au = \|Au\|^2 > 0 \quad \text{for all } u \in \mathbb{R}^n \setminus \mathrm{Ker}(A).$$

Thus the conditions of the conjugate gradient algorithm are almost satisfied, and one may look for the solution of the inverse problem by using the conjugate gradient algorithm with A replaced by $A^{\mathrm{T}}A$ and y by $A^{\mathrm{T}}y$.[†]

As a stopping criterion, one may try, e.g., the Morozov principle for the original equation: terminate the iteration when

$$\|y - Ax_k\| \leq \varepsilon$$

for some $\varepsilon > 0$, which measures the amount of noise in y in some sense.

[†] Small remark on implementation: matrix-matrix products are typically far more expensive to compute than matrix-vector products. For example, instead of computing expressions like `residual = A'*y - A'*A*x0` when implementing the conjugate gradient method in MATLAB, one should use parentheses to parse the computation like `residual = A'*y - A'*(A*x0)`.

Numerical example: backward heat equation revisited

Let us revisit the backward heat equation:

$$\begin{cases} \partial_t u(x,t) = \partial_x^2 u(x,t) & \text{for } (x,t) \in (0,\pi) \times \mathbb{R}_+, \\ u(0,\cdot) = u(\pi,\cdot) = 0 & \text{on } \mathbb{R}_+, \\ u(\cdot,0) = f & \text{on } (0,\pi), \end{cases}$$

where $f\colon (0,\pi) \to \mathbb{R}$ is the initial heat distribution.

Inverse problem: Reconstruct the initial state f based on noisy measurements of $u(\cdot, T)$ at time $T > 0$.

Let $x_j = jh$, $j = 0, \ldots, 100$ with $h = \pi/100$, and denote $U(t) = (U_j(t))_{j=1}^{99}$ and $F = (f(x_j))_{j=1}^{99}$. At time $t = T > 0$, the discretized heat distribution $U := U(T)$ is given by

$$U = AF,$$

where $A = \mathrm{e}^{TB} \in \mathbb{R}^{99\times 99}$ and $B = h^{-2}\mathrm{tridiag}(1,-2,1) \in \mathbb{R}^{99\times 99}$.

As ground truth, we take

$$f(x) = \begin{cases} 1 & \text{if } x \in [1,2], \\ 0 & \text{if } x \in (0,1) \cup (2,\pi). \end{cases}$$

We assume that the simulated data $U = U(T) \in \mathbb{R}^{99}$ at time $T = 0.1$ is contaminated with mean-zero Gaussian noise with standard deviation 0.01, and that the discrepancy between the measured data and the underlying "exact" data equals the square root of the expected value of the squared norm of the noise vector, i.e.,

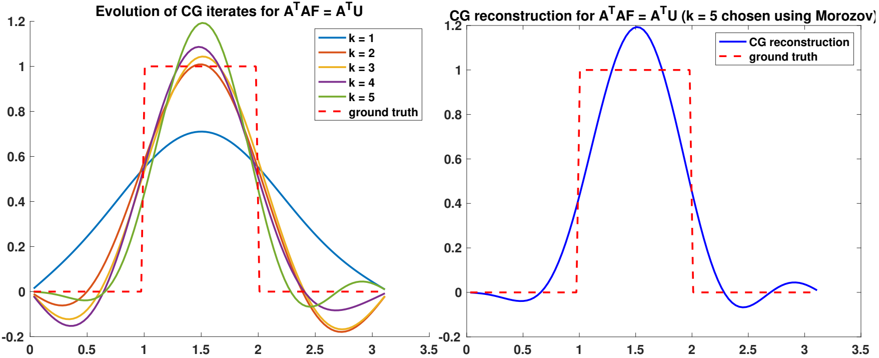
$$\varepsilon = \sqrt{99 \cdot 0.01^2} \approx 0.0995.$$

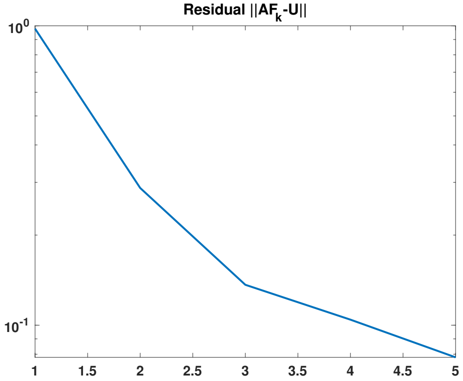
We use the conjugate gradient method to solve the normal equation

$$A^{\mathrm{T}} A F = A^{\mathrm{T}} U,$$

and terminate the algorithm for the first CG iterate F_k such that

$$\|AF_k - U\| \leq \varepsilon.$$


Evolution of CG iterates for $A^TAF = A^TU$
k = 1
k = 2
k = 3
k = 4
k = 5
ground truth
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.5
1
1.5
2
2.5
3
3.5
CG reconstruction for $A^TAF = A^TU$ (k = 5 chosen using Morozov)
CG reconstruction
ground truth
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.5
1
1.5
2
2.5
3
3.5


Residual ||AF_k-U||
10^0
10^-1
1
1.5
2
2.5
3
3.5
4
4.5
5

Although we have simply scratched the surface by covering some of the basic ideas surrounding the conjugate gradient scheme and demonstrating how an “early stopping rule” can provide reasonable solutions for inverse problems, the regularizing properties of the conjugate gradient method have been analyzed more explicitly in the literature. A classic textbook specifically about this subject is:

M. Hanke. *Conjugate gradient type methods for ill-posed problems*. Pitman Research Notes in Mathematics Series, 327.