Numerical Analysis - Part II

Anders C. Hansen

Lecture 18

Iterative methods for linear algebraic systems

Solving linear systems with iterative methods

The general *iterative* method for solving $Ax = b$ is a rule $\boldsymbol{x}^{k+1} = f_k(\boldsymbol{x}^0, \boldsymbol{x}^1, \ldots, \boldsymbol{x}^k)$. We will consider the simplest ones: *linear, one-step, stationary* iterative schemes:

$$\boldsymbol{x}^{k+1} = H\boldsymbol{x}^k + \boldsymbol{v}, \qquad \boldsymbol{x}^0, \boldsymbol{v} \in \mathbb{R}^n. \tag{1}$$

Here one chooses H and v so that x^*, a solution of $A\boldsymbol{x} = \boldsymbol{b}$, satisfies $\boldsymbol{x}^* = H\boldsymbol{x}^* + \boldsymbol{v}$, i.e. it is the fixed point of the iteration (1) (if the scheme converges). Standard terminology:

- the *iteration matrix* H,
- the *error* $\boldsymbol{e}^k := \boldsymbol{x}^* - \boldsymbol{x}^k$,
- the *residual* $\boldsymbol{r}^k := A\boldsymbol{e}^k = \boldsymbol{b} - A\boldsymbol{x}^k$.

Linear systems in elliptic PDEs

As we have seen in the previous sections linear systems $A\boldsymbol{x} = \boldsymbol{b}$, where A is a real symmetric positive (negative) definite matrix, frequently occur in numerical methods for solving elliptic partial differential equations.

Poisson's equation on a square

A typical example we already encountered is Poisson's equation on a square where the *five-point formula* approximation yields an $n \times n$ system of linear equations with $n = m^2$ unknowns $u_{p,q}$:

$$u_{p-1,q} + u_{p+1,q} + u_{p,q-1} + u_{p,q+1} - 4u_{p,q} = h^2 f(ph, qh) \qquad (2)$$

In the *natural ordering*, when the grid points are arranged by columns, A is the following block tridiagonal matrix:

$$A = \begin{bmatrix} B & I & & & \\ I & B & I & & \\ & \ddots & \ddots & \ddots & \\ & & I & B & I \\ & & & I & B \end{bmatrix}, \qquad B = \begin{bmatrix} -4 & 1 & & & \\ 1 & -4 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -4 & 1 \\ & & & 1 & -4 \end{bmatrix}. \qquad (3)$$

The matrix A is symmetric and negative definite

Lemma 1

For any ordering of the grid points, the matrix A of the system (2) is symmetric and negative definite.

Poisson's equation on a square

Note that when p or q is equal to 1 or m, then the values $u_{0,q}$, $u_{p,0}$ or $u_{p,m+1}$, $u_{m+1,q}$ are known boundary values and they should be moved to the right-hand side, thus leaving fewer unknowns on the left.

For any ordering of the grid points (ph, qh) we have shown in Lemma 1 that the matrix A of this linear system is symmetric and negative definite.

Corollary 2

For the linear system (2), for any ordering of the grid, both Jacobi and Gauss-Seidel methods converge.

Proof. By Lemma 1, A is symmetric and negative definite, hence convergence of Gauss-Seidel. To prove convergence of the Jacobi method, we need negative definiteness of the matrix $2D - A$, and that follows by the same arguments as in Lemma 1: recall that the proof operates with the modulus of the off-diagonal elements and does not depend on their sign. □

Relaxation

It is often possible to improve the efficiency of the splitting method by *relaxation*. Specifically, instead of letting $(A - B)\boldsymbol{x}^{(k+1)} = -B\boldsymbol{x}^{(k)} + \boldsymbol{b}$, we let

$$(A-B)\widehat{\boldsymbol{x}}^{(k+1)} = -B\boldsymbol{x}^{(k)}+\boldsymbol{b}, \quad \text{and then} \quad \boldsymbol{x}^{(k+1)} = \omega\widehat{\boldsymbol{x}}^{(k+1)}+(1-\omega)\boldsymbol{x}^{(k)}$$

where ω is a real constant called the *relaxation parameter*.

Note that $\omega = 1$ corresponds to the standard "unrelaxed" iteration. Good choice of ω leads to a smaller spectral radius of the iteration matrix (compared with the "unrelaxed" method), and the smaller the spectral radius, the faster the iteration converges.

Relaxation

To this end, let us express the relaxation iteration matrix H_ω in terms of $H = -(A-B)^{-1}B$. We have

$$\widehat{\boldsymbol{x}}^{(k+1)} = H\boldsymbol{x}^{(k)} + \boldsymbol{v} \quad \Rightarrow \quad \begin{aligned} \boldsymbol{x}^{(k+1)} &= \omega\widehat{\boldsymbol{x}}^{(k+1)} + (1-\omega)\boldsymbol{x}^{(k)} \\ &= \omega H\boldsymbol{x}^{(k)} + (1-\omega)\boldsymbol{x}^{(k)} + \omega\boldsymbol{v}, \end{aligned}$$

hence

$$H_\omega = \omega H + (1-\omega)I.$$

It follows that the spectra of H_ω and H are related by the rule $\lambda_\omega = \omega\lambda + (1-\omega)$, therefore one may try to choose $\omega \in \mathbb{R}$ to minimize

$$\rho(H_\omega) = \max\{|\omega\lambda + (1-\omega)| : \lambda \in \sigma(H)\}.$$

Relaxation

In general, $\sigma(H)$ is unknown, but often we have some information about it which can be utilized to find a "good" (rather than "best") value of ω. For example, suppose that it is known that $\sigma(H)$ is real and resides in the interval $[\alpha, \beta]$ where $-1 < \alpha < \beta < 1$. In that case we seek ω to minimize

$$\max\{|\omega\lambda + (1-\omega)| : \lambda \in [\alpha, \beta]\}.$$

Relaxation

It is readily seen that, for a fixed $\lambda < 1$, the function $f(\omega) = \omega\lambda + (1-\omega)$ is decreasing, therefore, as ω increases (decreases) from 1 the spectrum of H_ω moves to the left (to the right) of the spectrum of H. It is clear that the optimal location of the spectrum $\sigma(H_\omega)$ (or of the interval $[\alpha_\omega, \beta_\omega]$ that contains $\sigma(H_\omega)$) is the one which is centralized around the origin:

$$-[\omega\alpha+(1-\omega)] = \omega\beta+(1-\omega) \quad \Rightarrow \quad \begin{aligned} \omega_{\mathrm{opt}} &= \frac{2}{2-(\alpha+\beta)} \\ -\alpha_{\omega_{\mathrm{opt}}} &= \beta_{\omega_{\mathrm{opt}}} = \frac{\beta-\alpha}{2-(\alpha+\beta)}\,. \end{aligned}$$

Minimization of quadratic function

The methods we considered so far for solving $A\boldsymbol{x} = \boldsymbol{b}$, namely Jacobi, Gauss-Seidel, and those with relaxation, fit into the scheme

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + c_k \boldsymbol{d}^{(k)} ,$$

where we were aimed at getting $\rho(H) < 1$ for the iteration matix H. Say, for Jacobi with relaxation, we set $c_k = \omega$ and $\boldsymbol{d}^{(k)} = D^{-1}(\boldsymbol{b} - A\boldsymbol{x}^{(k)})$.
For solving $A\boldsymbol{x} = \boldsymbol{b}$ with a (positive definite) matrix $A > 0$, there is a different approach to constructing good iterative methods. It is based on succesive minimization of the quadratic function

$$F(\boldsymbol{x}^{(k)}) := \|\boldsymbol{x}^* - \boldsymbol{x}^{(k)}\|_A^2 = \|\boldsymbol{e}^{(k)}\|_A^2 ,$$

since the minimizer is clearly the exact solution. Here, $\|\boldsymbol{y}\|_A := (A\boldsymbol{y}, \boldsymbol{y})^{1/2} := \sqrt{\boldsymbol{y}^T A\boldsymbol{y}}$ is a Euclidean-type distance which is well-defined for $A > 0$.

Minimization of quadratic function

So, at each step k, we are decreasing the A-distance between $\boldsymbol{x}^{(k)}$ and the exact solution $\boldsymbol{x}^*$. Thus, for a symmetric positive definite $A > 0$, we choose an iterative method that provides the descent condition

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + c_k \boldsymbol{d}^{(k)} \quad \Rightarrow \quad F(\boldsymbol{x}^{(k+1)}) < F(\boldsymbol{x}^{(k)}) \,. \tag{4}$$

Minimization of quadratic function

An equivalent approach is to minimize the quadratic function

$$F_1(\boldsymbol{x}) = \tfrac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} - \boldsymbol{x}^T\boldsymbol{b}\,,$$

which attains its minimum when $\nabla F_1(\boldsymbol{x}) = A\boldsymbol{x} - \boldsymbol{b} = 0$, and which does not involve the unknown $\boldsymbol{x}^*$. It is easy to check that $F_1(\boldsymbol{x}) = \frac{1}{2}F(\boldsymbol{x}) - \frac{1}{2}c$, where $c = \boldsymbol{x}^{*T} A\boldsymbol{x}^*$ is a constant independent of k, hence equivalence.

Quadratic function – Jacobi and Gauss–Seidel

Both the Jacobi and the Gauss–Seidel methods satisfy (4), precisely

$$(A\boldsymbol{e}^{(k+1)}, \boldsymbol{e}^{(k+1)}) = (A\boldsymbol{e}^{(k)}, \boldsymbol{e}^{(k)}) - (C\boldsymbol{y}^{(k)}, \boldsymbol{y}^{(k)}) < (A\boldsymbol{e}^{(k)}, \boldsymbol{e}^{(k)}),$$

where for Gauss-Seidel: $C = D > 0$, $\boldsymbol{y}^{(k)} := (L_0 + D)^{-1} A\boldsymbol{e}^{(k)}$;

and for Jacobi: $C = 2D - A > 0$, $\boldsymbol{y}^{(k)} := D^{-1} A\boldsymbol{e}^{(k)}$.

A-orthogonal projection

A-orthogonal projection method: Next, we strengthen the descent condition (4), namely given $\boldsymbol{x}^{(k)}$ and some $\boldsymbol{d}^{(k)}$ (called a *search direction*), we will seek $\boldsymbol{x}^{(k+1)}$ from the set of vectors on the line $\ell = \{\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)}\}_{\alpha \in \mathbb{R}}$ such that it makes the value of $F(\boldsymbol{x}^{(k+1)})$ not just smaller than $F(\boldsymbol{x}^{(k)})$, but as small as possible (with respect to this set), namely

$$\boldsymbol{x}^{(k+1)} := \arg\min_{\alpha} F(\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)}) \,. \tag{5}$$

Lemma 3

The minimizer in (5) is given by the formula

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}, \qquad \alpha_k = \frac{(\boldsymbol{r}^{(k)}, \boldsymbol{d}^{(k)})}{(A\boldsymbol{d}^{(k)}, \boldsymbol{d}^{(k)})}. \tag{6}$$

This choice of α_k is referred to as exact line search.

Proof. From the definition of F, it follows that in (5) we should choose the point $\boldsymbol{x}^{(k+1)} \in \ell$ that minimizes the A-distance between $\boldsymbol{x}^*$ and the points $\boldsymbol{y} \in \ell$. Geometrically, it is clear that the minimum occurs when $\boldsymbol{x}^{(k+1)}$ is the A-orthogonal projection of $\boldsymbol{x}^*$ onto the line $\ell = \{\boldsymbol{x}^{(k)} + \alpha \boldsymbol{d}^{(k)}\}$, i.e., when

$$\begin{aligned} \boldsymbol{x}^* - \boldsymbol{x}^{(k+1)} \perp_A \boldsymbol{d}^{(k)} \quad &\Rightarrow \quad A(\boldsymbol{x}^* - \boldsymbol{x}^{(k+1)}) \perp \boldsymbol{d}^{(k)} \\ &\Rightarrow \quad \boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k A \boldsymbol{d}^{(k)} \perp \boldsymbol{d}^{(k)} . \end{aligned}$$

This gives expression for α_k in (6). □

The steepest descent method

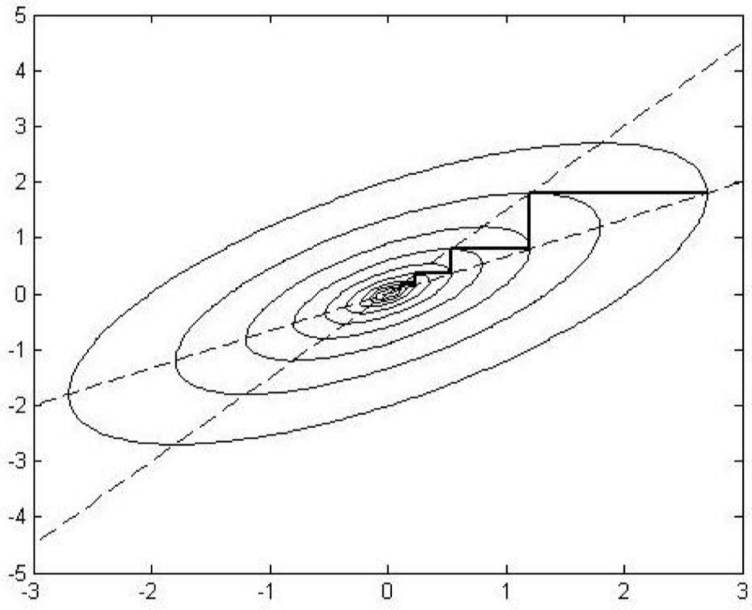
The steepest descent method: This method takes $\boldsymbol{d}^{(k)} = -\nabla F_1(\boldsymbol{x}^{(k)}) = \boldsymbol{b} - A\boldsymbol{x}^{(k)}$ for every k, the reason being that, locally, the negative gradient of a quadratic function shows the direction of the (locally) steepest descent at a given point. Thus, the iterations have the form

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k(\boldsymbol{b} - A\boldsymbol{x}^{(k)}), \qquad k \geq 0\,. \tag{7}$$

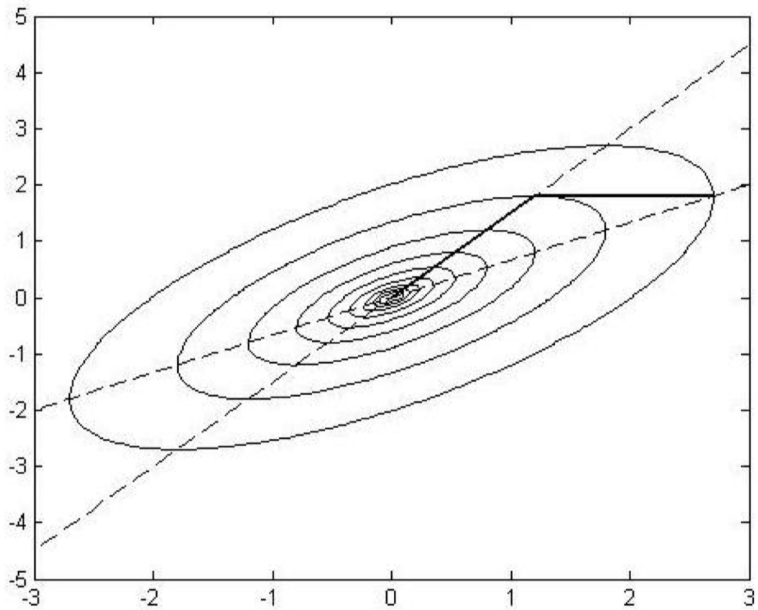
It can be proved that the sequence $(\boldsymbol{x}^{(k)})$ converges to the solution $\boldsymbol{x}^*$ of the system $A\boldsymbol{x} = \boldsymbol{b}$ as required, but usually the speed of convergence is rather slow.

The reason is that the iteration (7) decreases the value of $F(\boldsymbol{x}^{(k+1)})$ locally, relatively to $F(\boldsymbol{x}^{(k)})$, but the global decrease, with respect to $F(\boldsymbol{x}^{(0)})$, is often not that large. The use of *conjugate directions* provides a method with a global minimization property.

Steepest descent and conjugate gradient



(a) Worst case scenario of steepest descent



(b) Conjugate gradient method applied to the same problem as in (a)

Conjugate directions

Let's revisit equation (6) for a general direction $\boldsymbol{d}$ (i.e., not necessarily equal to the negative gradient). Assume $\boldsymbol{x} = \boldsymbol{x}^{(k)}$, and let $\boldsymbol{e}^{(k)} = \boldsymbol{x}^* - \boldsymbol{x}^{(k)}$ be the error and $\boldsymbol{r}^{(k)} = \boldsymbol{b} - A\boldsymbol{x}^{(k)} = A\boldsymbol{e}^{(k)}$ be the residual. Then we can write $\langle \boldsymbol{r}^{(k)}, \boldsymbol{d} \rangle = \langle \boldsymbol{e}^{(k)}, \boldsymbol{d} \rangle_A$, and so for a general search direction $\boldsymbol{d}$ with an exact line search, the iterate takes the form $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \frac{\langle \boldsymbol{e}^{(k)}, \boldsymbol{d} \rangle_A}{\langle \boldsymbol{d}, \boldsymbol{d} \rangle_A} \boldsymbol{d}$. By substracting $\boldsymbol{x}^*$, the iterates in terms of the error $\boldsymbol{e}^{(k+1)}$ are given by:

$$\boldsymbol{e}^{(k+1)} = \boldsymbol{e}^{(k)} - \frac{\langle \boldsymbol{e}^{(k)}, \boldsymbol{d} \rangle_A}{\langle \boldsymbol{d}, \boldsymbol{d} \rangle_A} \boldsymbol{d}. \tag{8}$$

Geometrically, this means that $\boldsymbol{e}^{(k+1)}$ is the projection of $\boldsymbol{e}^{(k)}$ onto the hyperplane that is A-orthogonal to $\boldsymbol{d}$, i.e., we have

$$\langle \boldsymbol{e}^{(k+1)}, \boldsymbol{d} \rangle_A = 0. \tag{9}$$

Conjugate directions

Definition 4 (Conjugate directions)

The vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ are *conjugate* with respect to a symmetric positive definite matrix A if they are nonzero and A-orthogonal: $\langle \boldsymbol{u}, \boldsymbol{v} \rangle_A := \langle \boldsymbol{u}, A\boldsymbol{v} \rangle = 0$.

Conjugate gradient - Warm up

Theorem 5

Let $\boldsymbol{d}^{(0)}, \boldsymbol{d}^{(1)}, \ldots, \boldsymbol{d}^{(n-1)}$ be n nonzero pairwise conjugate directions, and consider the sequence of iterates

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}, \qquad \alpha_k = \frac{\langle \boldsymbol{r}^{(k)}, \boldsymbol{d}^{(k)} \rangle}{\langle \boldsymbol{d}^{(k)}, A\boldsymbol{d}^{(k)} \rangle}.$$

Let $\boldsymbol{r}^{(k)} = \boldsymbol{b} - A\boldsymbol{x}^{(k)}$ be the residual. Then for each $k = 1, \ldots, n$, $\boldsymbol{r}^{(k)}$ is orthogonal to $\operatorname{span}\{\boldsymbol{d}^{(0)}, \ldots, \boldsymbol{d}^{(k-1)}\}$. In particular $\boldsymbol{r}^{(n)} = 0$.

Proof. Since $\boldsymbol{r}^{(k)} = A\boldsymbol{e}^{(k)}$, it suffices to show that $\boldsymbol{e}^{(k)}$ is A-orthogonal to $\operatorname{span}\{\boldsymbol{d}^{(0)}, \ldots, \boldsymbol{d}^{(k-1)}\}$. The proof is by induction on k. For $k = 0$ there is nothing to prove. Assume the statement is true for $k \geq 0$, and consider the equation (8) (with $\boldsymbol{d} = \boldsymbol{d}^{(k)}$). From the induction hypothesis, and the fact that the $\boldsymbol{d}^{(i)}$ are pairwise conjugate directions, we see that $\boldsymbol{e}^{(k+1)}$ is A-orthogonal to $\boldsymbol{d}^{(0)}, \ldots, \boldsymbol{d}^{(k-1)}$. Furthermore, we have already seen in (9) that $\langle \boldsymbol{e}^{(k+1)}, \boldsymbol{d}^{(k)} \rangle_A = 0$. Thus this shows that $\boldsymbol{e}^{(k+1)}$ is A-orthogonal to $\boldsymbol{d}^{(0)}, \ldots, \boldsymbol{d}^{(k)}$ as desired. □

Conjugate gradient - Warm up

So, if a sequence $(\boldsymbol{d}^{(k)})$ of conjugate directions is at hands, we have an iterative procedure with good approximation properties.
The (A-orthogonal) basis of conjugate directions is constructed by A-orthogonalization of the sequence $\{\boldsymbol{r}_0, A\boldsymbol{r}_0, A^2\boldsymbol{r}_0, ..., A^{n-1}\boldsymbol{r}_0\}$ with $\boldsymbol{r}_0 = \boldsymbol{b} - A\boldsymbol{x}_0$. This is done in the way similar to orthogonalization of the monomial sequence $\{1, x, x^2, ..., x^{n-1}\}$ using a recurrence relation.

Remark 6

It is possible to extend the methods for solving $A\boldsymbol{x} = \boldsymbol{b}$ with symmetric positive definite A to any other matrices by a simple trick. Suppose we want to solve $B\boldsymbol{x} = \boldsymbol{c}$, where $B \in \mathbb{R}^{n \times n}$ is nonsingular. We can convert the above system to the symmetric and positive definite setting by defining $A = B^T B$, $\boldsymbol{b} = B^T \boldsymbol{c}$ and then solving $A\boldsymbol{x} = \boldsymbol{b}$ with the conjugate gradient algorithm (or any other method for positive definite A).

The conjugate gradient method

Here it is.

(A) For any initial vector $\boldsymbol{x}^{(0)}$, set $\boldsymbol{d}^{(0)} = \boldsymbol{r}^{(0)} = \boldsymbol{b} - A\boldsymbol{x}^{(0)}$;

(B) For $k \geq 0$, calculate $\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \alpha_k \boldsymbol{d}^{(k)}$ and the residual

$$\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \alpha_k A\boldsymbol{d}^{(k)}, \quad \text{with}$$

$$\alpha_k := \{\, \boldsymbol{r}^{(k+1)} \perp \boldsymbol{d}^{(k)}\} = \frac{(\boldsymbol{r}^{(k)}, \boldsymbol{d}^{(k)})}{(A\boldsymbol{d}^{(k)}, \boldsymbol{d}^{(k)})}, \quad k \geq 0\,. \tag{10}$$

(C) For the same k, the next conjugate direction is the vector

$$\boldsymbol{d}^{(k+1)} = \boldsymbol{r}^{(k+1)} + \beta_k \boldsymbol{d}^{(k)}, \quad \text{with}$$

$$\beta_k := \{\, \boldsymbol{d}^{(k+1)} \perp A\boldsymbol{d}^{(k)}\} = -\frac{(\boldsymbol{r}^{(k+1)}, A\boldsymbol{d}^{(k)})}{(\boldsymbol{d}^{(k)}, A\boldsymbol{d}^{(k)})}, \quad k \geq 0\,. \tag{11}$$