Chapter 10

The Conjugate Gradient Method

The **conjugate gradient method** was introduced in [10] as a direct method for solving a linear system. Today its main use is as an iterative method for solving large sparse linear systems. On a test problem we show that it performs as well as the SOR method with optimal acceleration parameter, and we do not have to estimate any such parameter. However the conjugate gradient method is restricted to symmetric positive definite systems. We also consider the **preconditioned conjugate gradient method** which is used to speed up convergence of the conjugate gradient method.

The conjugate gradient method can also be used for minimization and is related to a minimization method known as **steepest descent**. This method and the conjugate gradient method are both minimization methods and iterative methods for solving linear equations, when the function to be minimized is quadratic.

Throughout this chapter $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ will be a symmetric positive definite matrix. Thus, $\boldsymbol{A}^T = \boldsymbol{A}$ and $\boldsymbol{y}^T\boldsymbol{A}\boldsymbol{y} > 0$ for all nonzero $\boldsymbol{y} \in \mathbb{R}^n$. We recall that $\boldsymbol{A}$ has positive eigenvalues and that the spectral (2-norm) condition number of $\boldsymbol{A}$ is given by $\kappa := \frac{M}{m}$, where M and m are the largest and smallest eigenvalue of $\boldsymbol{A}$.

10.1 Quadratic Minimization

We start by discussing some aspect of quadratic minimization and its relation to solving linear systems.

Consider for $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{b} \in \mathbb{R}^n$, the quadratic function $Q : \mathbb{R}^n \to \mathbb{R}$ given by

$$Q(\boldsymbol{y}) := \frac{1}{2}\boldsymbol{y}^T\boldsymbol{A}\boldsymbol{y} - \boldsymbol{b}^T\boldsymbol{y}. \tag{10.1}$$

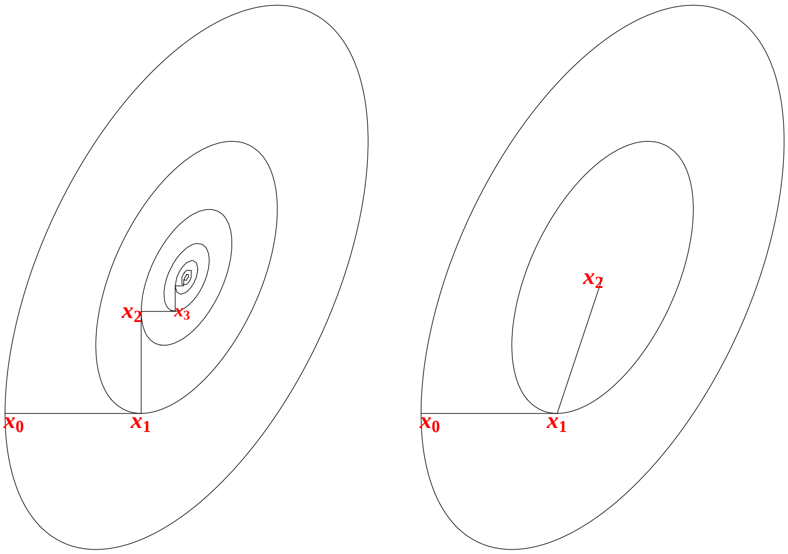

Figure 10.1: Level curves for $Q(x, y)$ given by (10.2). Also shown is a steepest descent iteration (left) and a conjugate gradient iteration (right) to find the minimum of Q. (cf Examples 10.3,10.12)

.

As an example, some level curves of

$$Q(x,y) := \frac{1}{2}\begin{bmatrix} x & y \end{bmatrix}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = x^2 - xy + y^2 \qquad (10.2)$$

are shown in Figure 10.1. The level curves are ellipses and the graph of Q is a paraboloid (cf. Exercise 10.1).

Exercise 10.1 (Paraboloid)
Let $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T$ *be the spectral decomposition of* $\boldsymbol{A}$*, i. e.,* $\boldsymbol{U}$ *is orthonormal and* $\boldsymbol{D} = \operatorname{diag}(\lambda_1, \ldots, \lambda_n)$ *is diagonal. Define new varables* $\boldsymbol{v} = [v_1, \ldots, v_n]^T := \boldsymbol{U}^T\boldsymbol{y}$*, and set* $\boldsymbol{c} := \boldsymbol{U}^T\boldsymbol{b} = [c_1, \ldots, c_n]^T$*. Show that*

$$Q(\boldsymbol{y}) = \frac{1}{2}\sum_{j=1}^{n}\lambda_j v_j^2 - \sum_{j=1}^{n} c_j v_j.$$

Minimizing a quadratic function is equivalent to solving a linear system.

Lemma 10.2 (Quadratic function)
A vector $\boldsymbol{x} \in \mathbb{R}^n$ *minimizes* Q *given by* (10.1) *if and only if* $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$*. Moreover,*

the residual $\boldsymbol{r}(\boldsymbol{y}) := \boldsymbol{b} - \boldsymbol{A}\boldsymbol{y}$ *at any* $\boldsymbol{y} \in \mathbb{R}^n$ *is equal to the negative gradient, i. e.,* $\boldsymbol{r}(\boldsymbol{y}) = -\nabla Q(\boldsymbol{y})$, *where* $\nabla := \left[\frac{\partial}{\partial y_1}, \dots, \frac{\partial}{\partial y_n}\right]^T$.

Proof. For any $\boldsymbol{y}, \boldsymbol{h} \in \mathbb{R}^n$ and $\varepsilon \in \mathbb{R}$

$$Q(\boldsymbol{y} + \varepsilon\boldsymbol{h}) = Q(\boldsymbol{y}) - \varepsilon\boldsymbol{h}^T r(\boldsymbol{y}) + \frac{1}{2}\varepsilon^2\boldsymbol{h}^T\boldsymbol{A}\boldsymbol{h}, \text{ where } r(\boldsymbol{y}) := \boldsymbol{b} - \boldsymbol{A}\boldsymbol{y}. \tag{10.3}$$

If $\boldsymbol{y} = \boldsymbol{x}$, $\varepsilon = 1$, and $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ then (10.3) simplifies to $Q(\boldsymbol{x} + \boldsymbol{h}) = Q(\boldsymbol{x}) + \frac{1}{2}\boldsymbol{h}^T\boldsymbol{A}\boldsymbol{h}$, and since $\boldsymbol{A}$ is symmetric positive definite $Q(\boldsymbol{x} + \boldsymbol{h}) > Q(\boldsymbol{x})$ for all nonzero $\boldsymbol{h} \in \mathbb{R}^n$. It follows that $\boldsymbol{x}$ is the unique minimum of Q. Conversely, if $\boldsymbol{A}\boldsymbol{x} \neq \boldsymbol{b}$ and $\boldsymbol{h} := \boldsymbol{r}(\boldsymbol{x})$, then by (10.3), $Q(\boldsymbol{x} + \varepsilon\boldsymbol{h}) - Q(\boldsymbol{x}) = -\varepsilon(\boldsymbol{h}^T\boldsymbol{r}(x) - \frac{1}{2}\varepsilon\boldsymbol{h}^T\boldsymbol{A}\boldsymbol{h}) < 0$ for $\varepsilon > 0$ sufficiently small. Thus $\boldsymbol{x}$ does not minimize Q. By (10.3) for $\boldsymbol{y} \in \mathbb{R}^n$

$$\begin{aligned}\frac{\partial}{\partial y_i}Q(\boldsymbol{y}) &:= \lim_{\varepsilon \to 0} \frac{1}{\varepsilon}\left(Q(\boldsymbol{y} + \varepsilon\boldsymbol{e}_i) - Q(\boldsymbol{y})\right) \\ &= \lim_{\varepsilon \to 0} \frac{1}{\varepsilon}\left(-\varepsilon\boldsymbol{e}_i^T\boldsymbol{r}(\boldsymbol{y})) + \frac{1}{2}\varepsilon^2\boldsymbol{e}_i^T\boldsymbol{A}\boldsymbol{e}_i\right) = -\boldsymbol{e}_i^T\boldsymbol{r}(\boldsymbol{y}), \quad i = 1, \dots, n,\end{aligned}$$

showing that $\boldsymbol{r}(\boldsymbol{y}) = -\nabla Q(\boldsymbol{y})$. □

A general class of minimization algorithms for Q and solution algorithms for a linear system is given as follows:

1. Choose $\boldsymbol{x}_0 \in \mathbb{R}^n$.
2. For $k = 0, 1, 2, \dots$

$$\boxed{\begin{array}{l}\text{Choose a “search direction” } \boldsymbol{p}_k, \\ \text{Choose a “step length” } \alpha_k, \\ \text{Compute } \boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k.\end{array}} \tag{10.4}$$

We would like to generate a sequence $\{\boldsymbol{x}_k\}$ that converges quickly to the minimum $\boldsymbol{x}$ of Q.

For a fixed direction $\boldsymbol{p}_k$ we say that α_k is **optimal** if $Q(\boldsymbol{x}_{k+1})$ is as small as possible, i.e.

$$Q(\boldsymbol{x}_{k+1}) = Q(\boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k) = \min_{\alpha \in \mathbb{R}} Q(\boldsymbol{x}_k + \alpha\boldsymbol{p}_k).$$

By (10.3) we have $Q(\boldsymbol{x}_k + \alpha\boldsymbol{p}_k) = Q(\boldsymbol{x}_k) - \alpha\boldsymbol{p}_k^T r_k + \frac{1}{2}\alpha^2\boldsymbol{p}_k^T\boldsymbol{A}\boldsymbol{p}_k$, where $\boldsymbol{r}_k := \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$. Since $\boldsymbol{p}_k^T\boldsymbol{A}\boldsymbol{p}_k \geq 0$ we find a minimum α_k by solving $\frac{\partial}{\partial\alpha}Q(\boldsymbol{x}_k + \alpha\boldsymbol{p}_k) = 0$. It follows that the optimal α_k is uniquely given by

$$\alpha_k := \frac{\boldsymbol{p}_k^T\boldsymbol{r}_k}{\boldsymbol{p}_k^T\boldsymbol{A}\boldsymbol{p}_k}. \tag{10.5}$$

10.2 Steepest Descent

In the method of **Steepest Descent**, also known as the **Gradient Method** we choose $\boldsymbol{p}_k = \boldsymbol{r}_k$, the negative gradient, and the optimal α_k. Starting from $\boldsymbol{x}_0$ and $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$ we compute for $k = 0, 1, 2 \dots$

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \Big(\frac{\boldsymbol{r}_k^T \boldsymbol{r}_k}{\boldsymbol{r}_k^T \boldsymbol{A} \boldsymbol{r}_k}\Big)\boldsymbol{r}_k. \tag{10.6}$$

Computationally, a step in the steepest descent iteration can proceed as follows

$$\boxed{\begin{aligned} \boldsymbol{t}_k &= \boldsymbol{A}\boldsymbol{r}_k, \\ \boldsymbol{\alpha}_k &= (\boldsymbol{r}_k^T \boldsymbol{r}_k)/(\boldsymbol{r}_k^T \boldsymbol{t}_k), \\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \alpha_k \boldsymbol{r}_k, \\ \boldsymbol{r}_{k+1} &= \boldsymbol{r}_k - \alpha_k \boldsymbol{t}_k. \end{aligned}} \tag{10.7}$$

Here, and in general, the following updating of the residual is used:

$$\boldsymbol{r}_{k+1} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_{k+1} = \boldsymbol{b} - \boldsymbol{A}(\boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k) = \boldsymbol{r}_k - \alpha_k \boldsymbol{t}_k, \quad \boldsymbol{t}_k := \boldsymbol{A}\boldsymbol{p}_k. \tag{10.8}$$

Example 10.3 (Steepest descent iteration)
Suppose $Q(x, y)$ is given by (10.2). *Starting with $\boldsymbol{x}_0 = [-1, -1/2]^T$ and $\boldsymbol{r}_0 = -\boldsymbol{A}\boldsymbol{x}_0 = [3/2, 0]^T$ we find*

$$\begin{aligned} &\boldsymbol{t}_0 = 3\begin{bmatrix} 1 \\ -1/2 \end{bmatrix}, \quad \alpha_0 = \frac{1}{2}, \quad \boldsymbol{x}_1 = -4^{-1}\begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \boldsymbol{r}_1 = 3 * 4^{-1}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &\boldsymbol{t}_1 = 3 * 4^{-1}\begin{bmatrix} -1 \\ 2 \end{bmatrix}, \quad \alpha_0 = \frac{1}{2}, \quad \boldsymbol{x}_2 = -4^{-1}\begin{bmatrix} 1 \\ 1/2 \end{bmatrix}, \quad \boldsymbol{r}_2 = 3 * 4^{-1}\begin{bmatrix} 1/2 \\ 0 \end{bmatrix}, \end{aligned}$$

and in general for $k \ge 1$

$$\begin{aligned} &\boldsymbol{t}_{2k-2} = 3 * 4^{1-k}\begin{bmatrix} 1 \\ -1/2 \end{bmatrix}, \quad \boldsymbol{x}_{2k-1} = -4^{-k}\begin{bmatrix} 1 \\ 2 \end{bmatrix}, \quad \boldsymbol{r}_{2k-1} = 3 * 4^{-k}\begin{bmatrix} 0 \\ 1 \end{bmatrix} \\ &\boldsymbol{t}_{2k-1} = 3 * 4^{-k}\begin{bmatrix} -1 \\ 2 \end{bmatrix}, \quad \boldsymbol{x}_{2k} = -4^{-k}\begin{bmatrix} 1 \\ 1/2 \end{bmatrix}, \quad \boldsymbol{r}_{2k} = 3 * 4^{-k}\begin{bmatrix} 1/2 \\ 0 \end{bmatrix}, \end{aligned}$$

and that $\alpha_k = 1/2$ for all k. See the left part of Figure 10.1. Since $\|\boldsymbol{r}_{j+1}\|_2/\|\boldsymbol{r}_j\|_2 = 1/2$ for all j the rate of convergence is not too impressive.

Exercise 10.4 (Steepest descent iteration)
Verify the numbers in Example 10.3.

10.2.1 Convergence Analysis for Steepest Descent

The steepest descent method applied to minimizing a quadratic function, or equivalently solving a symmetric positive definite linear system, always converges and we have an upper bound for the rate of convergence.

The convergence analysis is in terms of a special inner product. We define the $\boldsymbol{A}$-inner product and the corresponding $\boldsymbol{A}$-norm by

$$\langle \boldsymbol{x}, \boldsymbol{y} \rangle := \boldsymbol{x}^T \boldsymbol{A} \boldsymbol{y}, \quad \|\boldsymbol{y}\|_{\boldsymbol{A}} := \sqrt{\langle \boldsymbol{y}, \boldsymbol{y} \rangle}, \quad \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n, \tag{10.9}$$

Exercise 10.5 (The $\boldsymbol{A}$-inner product)
Show that if $\boldsymbol{A}$ is symmetric positive definite then the $\boldsymbol{A}$-inner product is an inner product.

The following theorem gives upper bounds for the $\boldsymbol{A}$-norm of the error in steepest descent .

Theorem 10.6 (Error bound for steepest descent)
For the $\boldsymbol{A}$-norms of the errors in the steepest descent method (10.6) *the following upper bounds hold*

$$\frac{||\boldsymbol{x} - \boldsymbol{x}_k||_{\boldsymbol{A}}}{||\boldsymbol{x} - \boldsymbol{x}_0||_{\boldsymbol{A}}} \le \left(\frac{\kappa - 1}{\kappa + 1}\right)^k < e^{-\frac{2}{\kappa}k}, \quad , k \ge 0, \tag{10.10}$$

where $\kappa = cond_2(\boldsymbol{A}) := M/m$ is the spectral condition number of $\boldsymbol{A}$, and M and m are the largest and smallest eigenvalue of $\boldsymbol{A}$, respectively.

In general the first upper bound in Theorem 10.6 is quite sharp. In fact for the iteration in Example 10.3 the first inequality in (10.10) is an equality (cf. Exercixe 10.7). The second inequality is sharp when κ is large.

Exercise 10.7 (A test for the error bound)
Show that in Example 10.3 that $\frac{||\boldsymbol{x}-\boldsymbol{x}_k||_{\boldsymbol{A}}}{||\boldsymbol{x}-\boldsymbol{x}_0||_{\boldsymbol{A}}} = \left(\frac{\kappa-1}{\kappa+1}\right)^k = 2^{-k}$ for $k \ge 0$.

Theorem 10.6 implies

1. Since $\frac{\kappa-1}{\kappa+1} < 1$ the steepest descent method always converges for a symmetric positive definite matrix.

2. The upper bound for the rate of convergence depends on the condition number κ. The convergence can be slow when $\frac{\kappa-1}{\kappa+1}$ is close to one, and this happens even for a moderately ill-conditioned $\boldsymbol{A}$.

Exercise 10.8 (Orthogonality in steepest descent)
Show that $\boldsymbol{r}_k^T \boldsymbol{r}_{k+1} = 0$ in the method of steepest descent. Does this mean that all the residuals are orthogonal?

10.2.2 Proof of error bound for steepest descent

For the proof of Theorem 10.6 the following inequality will be used.

Theorem 10.9 (Kantorovich inequality)
For any symmetric positive definite matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$

$$1 \le \frac{(\boldsymbol{y}^T\boldsymbol{A}\boldsymbol{y})(\boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y})}{(\boldsymbol{y}^T\boldsymbol{y})^2} \le \frac{(M+m)^2}{4Mm} \quad \boldsymbol{y} \ne \boldsymbol{0},\ \boldsymbol{y} \in \mathbb{R}^n, \tag{10.11}$$

where M *and* m *are the largest and smallest eigenvalue of* $\boldsymbol{A}$*, respectively.*

Proof. For $j = 1, \ldots, n$ let $(\lambda_j, \boldsymbol{u}_j)$ be orthonormal eigenpairs of $\boldsymbol{A}$ and $\boldsymbol{y} \in \mathbb{R}^n$. By Theorem 0.66 $(\lambda_j^{-1}, \boldsymbol{u}_j)$ are eigenpairs for $\boldsymbol{A}^{-1}$. Let $\boldsymbol{y} = \sum_{j=1}^n c_j\boldsymbol{u}_j$ be the eigenvector expansion of $\boldsymbol{y}$. By orthonormality, (cf. (6.7))

$$a := \frac{\boldsymbol{y}^T\boldsymbol{A}\boldsymbol{y}}{\boldsymbol{y}^T\boldsymbol{y}} = \sum_{i=1}^n t_i\lambda_i, \quad b := \frac{\boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y}}{\boldsymbol{y}^T\boldsymbol{y}} = \sum_{i=1}^n \frac{t_i}{\lambda_i}, \tag{10.12}$$

where

$$t_i = \frac{c_i^2}{\sum_{j=1}^n c_j^2} \ge 0, \quad i = 1, \ldots, n \text{ and } \sum_{i=1}^n t_i = 1. \tag{10.13}$$

Thus a and b are **convex combinations** of the eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{A}^{-1}$, respectively. Let c be a positive constant to be chosen later. By the geometric/arithmetic mean inequality (8.26) and (10.12)

$$\sqrt{ab} = \sqrt{(ac)(b/c)} \le (ac + b/c)/2 = \frac{1}{2}\sum_{i=1}^n t_i\bigl(\lambda_i c + 1/(\lambda_i c)\bigr) = \frac{1}{2}\sum_{i=1}^n t_i f(\lambda_i c),$$

where $f : [mc, Mc] \to \mathbb{R}$ is given by $f(x) := x + 1/x$. By (10.13)

$$\sqrt{ab} \le \frac{1}{2}\max_{mc\le x\le Mc} f(x).$$

Since $f \in C^2$ and f'' is positive it follows from Lemma 8.38 that f is a convex function. But a convex function takes it maximum at one of the endpoints of the range (cf. Exercise 10.10) and we obtain

$$\sqrt{ab} \le \frac{1}{2}\max\{f(mc), f(Mc)\}. \tag{10.14}$$

Choosing $c := 1/\sqrt{mM}$ we find $f(mc) = f(Mc) = \sqrt{\frac{M}{m}} + \sqrt{\frac{m}{M}} = \frac{M+m}{\sqrt{mM}}$. By (10.14) we obtain

$$\frac{(\boldsymbol{y}^T\boldsymbol{A}\boldsymbol{y})(\boldsymbol{y}^T\boldsymbol{A}^{-1}\boldsymbol{y})}{(\boldsymbol{y}^T\boldsymbol{y})^2} = ab \le \frac{(M+m)^2}{4Mm},$$

the upper bound in (10.11). For the lower bound we use Cauchy-Swarz inequality as follows

$$1 = \big(\sum_{i=1}^n t_i\big)^2 = \left(\sum_{i=1}^n (t_i\lambda_i)^{1/2}(t_i/\lambda_i)^{1/2}\right)^2 \le \big(\sum_{i=1}^n t_i\lambda_i\big)\big(\sum_{i=1}^n t_i/\lambda_i\big) = ab.$$

□

Exercise 10.10 (Maximum of a convex function)
Show that if $f:[a,b]\to\mathbb{R}$ is convex then $\max_{a\le x\le b} f(x) \le \max\{f(a), f(b)\}$.

Proof. Let $\boldsymbol{\epsilon}_j := x - \boldsymbol{x}_j$, $j = 0, 1, \ldots$, where $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$. It is enough to show that

$$\frac{\|\boldsymbol{\epsilon}_{k+1}\|_{\boldsymbol{A}}^2}{\|\boldsymbol{\epsilon}_k\|_{\boldsymbol{A}}^2} \le \left(\frac{\kappa-1}{\kappa+1}\right)^2, \quad k = 0, 1, 2, \ldots, \tag{10.15}$$

for then $\|\boldsymbol{\epsilon}_k\|_{\boldsymbol{A}} \le \left(\frac{\kappa-1}{\kappa+1}\right)\|\boldsymbol{\epsilon}_{k-1}\| \le \cdots \le \left(\frac{\kappa-1}{\kappa+1}\right)^k \|\boldsymbol{\epsilon}_0\|$. It follows from (10.6) that

$$\boldsymbol{\epsilon}_{k+1} = \boldsymbol{\epsilon}_k - \alpha_k \boldsymbol{r}_k, \quad \alpha_k := \frac{\boldsymbol{r}_k^T\boldsymbol{r}_k}{\boldsymbol{r}_k^T\boldsymbol{A}\boldsymbol{r}_k}.$$

We find

$$\begin{aligned}
\|\boldsymbol{\epsilon}_k\|_{\boldsymbol{A}}^2 &= \boldsymbol{\epsilon}_k^T\boldsymbol{A}\boldsymbol{\epsilon}_k = \boldsymbol{r}_k^T\boldsymbol{A}^{-1}\boldsymbol{r}_k,\\
\|\boldsymbol{\epsilon}_{k+1}\|_{\boldsymbol{A}}^2 &= (\boldsymbol{\epsilon}_k - \alpha_k\boldsymbol{r}_k)^T\boldsymbol{A}(\boldsymbol{\epsilon}_k - \alpha_k\boldsymbol{r}_k)\\
&= \boldsymbol{\epsilon}_k^T\boldsymbol{A}\boldsymbol{\epsilon}_k - 2\alpha_k\boldsymbol{r}_k^T\boldsymbol{A}\boldsymbol{\epsilon}_k + \alpha_k^2\boldsymbol{r}_k^T\boldsymbol{A}\boldsymbol{r}_k = \|\boldsymbol{\epsilon}_k\|_{\boldsymbol{A}}^2 - \frac{(\boldsymbol{r}_k^T\boldsymbol{r}_k)^2}{\boldsymbol{r}_k^T\boldsymbol{A}\boldsymbol{r}_k}.
\end{aligned}$$

Combining these and using Kantorovich inequality

$$\frac{\|\boldsymbol{\epsilon}_{k+1}\|_{\boldsymbol{A}}^2}{\|\boldsymbol{\epsilon}_k\|_{\boldsymbol{A}}^2} = 1 - \frac{(\boldsymbol{r}_k^T\boldsymbol{r}_k)^2}{(\boldsymbol{r}_k^T\boldsymbol{A}\boldsymbol{r}_k)(\boldsymbol{r}_k^T\boldsymbol{A}^{-1}\boldsymbol{r}_k)} \le 1 - \frac{4mM}{(m+M)^2} = \left(\frac{M-m}{M+m}\right)^2 = \left(\frac{\kappa-1}{\kappa+1}\right)^2$$

and (10.15) is proved.

The inequality

$$\frac{x-1}{x+1} < e^{-2/x} \quad \text{for} \quad x > 1 \tag{10.16}$$

follows from the familiar series expansion of the exponential function. Indeed, with $y = 1/x$, using $2^k/k! = 2$, $k = 1, 2$, and $2^k/k! < 2$ for $k > 2$, we find

$$e^{2/x} = e^{2y} = \sum_{k=0}^{\infty}\frac{(2y)^k}{k!} < 1 + 2\sum_{k=1}^{\infty} y^k = \frac{1+y}{1-y} = \frac{x+1}{x-1}$$

and (10.16) follows. □

10.3 The Conjugate Gradient Method

The conjugate gradient method is very similar to the steepest descent method. The only difference is that the search directions now are $\boldsymbol{A}$-orthogonal. This implies that all gradients are orthogonal, and this property have given the method its name.

For the derivation we choose a starting vector $\boldsymbol{x}_0 \in \mathbb{R}^n$. If $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0 = \boldsymbol{0}$ then $\boldsymbol{x}_0$ is the exact solution and we are finished, otherwise we initially make a steepest descent step with $\boldsymbol{p}_0 = \boldsymbol{r}_0$. Since $\boldsymbol{p}_0^T\boldsymbol{A}\boldsymbol{p}_0$ is nonzero, $\boldsymbol{x}_1 = \boldsymbol{x}_0 + \boldsymbol{\alpha}_0\boldsymbol{p}_0$, and $\boldsymbol{r}_1 = \boldsymbol{r}_0 - \alpha_0\boldsymbol{A}\boldsymbol{p}_0$ are well defined. The choice of α_0 ensures that $\boldsymbol{r}_1$ and $\boldsymbol{r}_0$ are orthogonal. Indeed, by (10.8), $\boldsymbol{r}_1^T\boldsymbol{r}_0 = (\boldsymbol{r}_0 - \alpha_0\boldsymbol{A}\boldsymbol{p}_0)^T\boldsymbol{r}_0 = 0$ since $\alpha_0 = \frac{\boldsymbol{p}_0^T\boldsymbol{r}_0}{\boldsymbol{p}_0^T\boldsymbol{A}\boldsymbol{p}_0}$ and $\boldsymbol{p}_0 = \boldsymbol{r}_0$. For the general case we define for $j \ge 0$

$$\boldsymbol{p}_j := \boldsymbol{r}_j - \sum_{i=0}^{j-1}(\frac{\boldsymbol{r}_j^T\boldsymbol{A}\boldsymbol{p}_i}{\boldsymbol{p}_i^T\boldsymbol{A}\boldsymbol{p}_i})\boldsymbol{p}_i, \tag{10.17}$$

$$\boldsymbol{x}_{j+1} := \boldsymbol{x}_j + \alpha_j\boldsymbol{p}_j \quad \alpha_j := \frac{\boldsymbol{r}_j^T\boldsymbol{r}_j}{\boldsymbol{p}_j^T\boldsymbol{A}\boldsymbol{p}_j}, \tag{10.18}$$

$$\boldsymbol{r}_{j+1} = \boldsymbol{r}_j - \alpha_j\boldsymbol{A}\boldsymbol{p}_j. \tag{10.19}$$

We note that

1. $\boldsymbol{p}_j$ is computed by the Gram-Schmidt orthogonalization process applied to the residuals $\boldsymbol{r}_0, \dots, \boldsymbol{r}_j$ using the $\boldsymbol{A}$-inner product and are therefore $\boldsymbol{A}$-orthogonal and nonzero provided the residuals are linearly independent. The formula (10.17) will be simplified considerably, (cf. (10.23)).

2. Equation (10.19) follows from (10.8).

3. The step length α_j is optimal for all j (cf. Exercise 10.15).

Lemma 10.11 (The residuals are orthogonal)
Suppose that for some $k \ge 0$ that $\boldsymbol{x}_j$ is well defined, $\boldsymbol{r}_j \ne 0$, and $\boldsymbol{r}_i^T\boldsymbol{r}_j = 0$ for $i, j = 0, 1, \dots, k$, $i \ne j$. Then $\boldsymbol{x}_{k+1}$ is well defined and $\boldsymbol{r}_{k+1}^T\boldsymbol{r}_j = 0$ for $j = 0, 1, \dots, k$.

Proof. Since the residuals $\boldsymbol{r}_j$ are orthogonal and nonzero for $j \le k$, they are linearly independent, and it follows form the Gram-Schmidt Theorem 0.38 that $\boldsymbol{p}_k$ is nonzero and $\boldsymbol{p}_k^T\boldsymbol{A}\boldsymbol{p}_i = 0$ for $i < k$. But then $\boldsymbol{x}_{k+1}$ and $\boldsymbol{r}_{k+1}$ are well defined.

Now

$$\begin{aligned}
\boldsymbol{r}_{k+1}^T\boldsymbol{r}_j &\overset{(10.19)}{=} (\boldsymbol{r}_k - \alpha_k\boldsymbol{A}\boldsymbol{p}_k)^T\boldsymbol{r}_j \\
&\overset{(10.17)}{=} \boldsymbol{r}_k^T\boldsymbol{r}_j - \alpha_k\boldsymbol{p}_k^T\boldsymbol{A}\big(\boldsymbol{p}_j + \sum_{i=0}^{j-1}(\frac{\boldsymbol{r}_j^T\boldsymbol{A}\boldsymbol{p}_i}{\boldsymbol{p}_i^T\boldsymbol{A}\boldsymbol{p}_i})\boldsymbol{p}_i\big) \\
&\overset{\boldsymbol{p}_k^T\boldsymbol{A}\boldsymbol{p}_i = 0}{=} \boldsymbol{r}_k^T\boldsymbol{r}_j - \alpha_k\boldsymbol{p}_k^T\boldsymbol{A}\boldsymbol{p}_j = 0, \quad j = 0, 1, \ldots, k.
\end{aligned}$$

That the final expression is equal to zero follows by orthogonality and $\boldsymbol{A}$-ortogonality for $j < k$ and by the definition of α_k for $j = k$. This completes the proof. □

The expression (10.17) for $\boldsymbol{p}_k$ can be greatly simplified. All terms except the last one vanish, since by orthogonality of the residuals

$$\boldsymbol{r}_j^T\boldsymbol{A}\boldsymbol{p}_i \overset{(10.19)}{=} \boldsymbol{r}_j^T\big(\frac{\boldsymbol{r}_i - \boldsymbol{r}_{i+1}}{\alpha_i}\big) = 0, \quad i = 0, 1, \ldots, j-2.$$

For the last term with $k = j - 1$

$$\beta_k := -\frac{\boldsymbol{r}_{k+1}^T\boldsymbol{A}\boldsymbol{p}_k}{\boldsymbol{p}_k^T\boldsymbol{A}\boldsymbol{p}_k} \overset{(10.19)}{=} \frac{\boldsymbol{r}_{k+1}^T(\boldsymbol{r}_{k+1} - \boldsymbol{r}_k)}{\alpha_k\boldsymbol{p}_k^T\boldsymbol{A}\boldsymbol{p}_k} \overset{(10.18)}{=} \frac{\boldsymbol{r}_{k+1}^T\boldsymbol{r}_{k+1}}{\boldsymbol{r}_k^T\boldsymbol{r}_k}. \tag{10.20}$$

To summarize, in the **conjugate gradient method** we start with $\boldsymbol{p}_0 = \boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$ and then generate a sequence of vectors $\{\boldsymbol{x}_k\}$ as follows:

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

$$\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k, \quad \alpha_k := \frac{\boldsymbol{r}_k^T\boldsymbol{r}_k}{\boldsymbol{p}_k^T\boldsymbol{A}\boldsymbol{p}_k}, \tag{10.21}$$

$$\boldsymbol{r}_{k+1} := \boldsymbol{r}_k - \alpha_k\boldsymbol{A}\boldsymbol{p}_k, \tag{10.22}$$

$$\boldsymbol{p}_{k+1} := \boldsymbol{r}_{k+1} + \beta_k\boldsymbol{p}_k, \quad \beta_k := \frac{\boldsymbol{r}_{k+1}^T\boldsymbol{r}_{k+1}}{\boldsymbol{r}_k^T\boldsymbol{r}_k}. \tag{10.23}$$

The residuals and search directions are orthogonal and $\boldsymbol{A}$-orthogonal, respectively. The conjugate gradient method is also a direct method. Since $\dim\mathbb{R}^n = n$ the $n + 1$ residuals $\boldsymbol{r}_0, \ldots, \boldsymbol{r}_n$ cannot all be nonzero and for orthogonal residuals we find the exact solution in at most n iterations.

For computation we use the following formulas for $k = 0, 1, 2, \ldots$

$$\boxed{\begin{aligned} \boldsymbol{t}_k &= \boldsymbol{A}\boldsymbol{p}_k, \\ \alpha_k &= (\boldsymbol{r}_k^T\boldsymbol{r}_k)/(\boldsymbol{p}_k^T\boldsymbol{t}_k), \\ \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + \alpha_k\boldsymbol{p}_k, \\ \boldsymbol{r}_{k+1} &= \boldsymbol{r}_k - \alpha_k\boldsymbol{t}_k, \\ \beta_k &= (\boldsymbol{r}_{k+1}^T\boldsymbol{r}_{k+1})/(\boldsymbol{r}_k^T\boldsymbol{r}_k), \\ \boldsymbol{p}_{k+1} &:= \boldsymbol{r}_{k+1} + \beta_k\boldsymbol{p}_k. \end{aligned}} \tag{10.24}$$

Example 10.12 (Conjugate gradient iteration)
Consider (10.24) *applied to the positive definite linear system* $\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$. *Starting as in Example 10.3 with* $\boldsymbol{x}_0 = \begin{bmatrix} -1 \\ -1/2 \end{bmatrix}$ *we find* $\boldsymbol{p}_0 = \boldsymbol{r}_0 = \begin{bmatrix} 3/2 \\ 0 \end{bmatrix}$ *and then*

$$\boldsymbol{t}_0 = \begin{bmatrix} 3 \\ -3/2 \end{bmatrix}, \quad \alpha_0 = 1/2, \quad \boldsymbol{x}_1 = \begin{bmatrix} -1/4 \\ -1/2 \end{bmatrix}, \quad \boldsymbol{r}_1 = \begin{bmatrix} 0 \\ 3/4 \end{bmatrix}, \quad \beta_0 = 1/4, \quad \boldsymbol{p}_1 = \begin{bmatrix} 3/8 \\ 3/4 \end{bmatrix},$$

$$\boldsymbol{t}_1 = \begin{bmatrix} 0 \\ 9/8 \end{bmatrix}, \quad \alpha_1 = 2/3, \quad \boldsymbol{x}_2 = \boldsymbol{0}, \quad \boldsymbol{r}_2 = \boldsymbol{0}.$$

Thus $\boldsymbol{x}_2$ *is the exact solution, see the right part in Figure 10.1.*

Exercise 10.13 (Conjugate gradient iteration,II)
Do one iteration with the conjugate gradient method when $\boldsymbol{x}_0 = \boldsymbol{0}$. *(Answer:* $\boldsymbol{x}_1 = \left(\frac{\boldsymbol{b}^T\boldsymbol{b}}{\boldsymbol{b}^T\boldsymbol{A}\boldsymbol{b}}\right)\boldsymbol{b}$.)

Exercise 10.14 (Conjugate gradient iteration,III)
Do two conjugate gradient iterations for the system

$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}$$

starting with $\boldsymbol{x}_0 = \boldsymbol{0}$.

Exercise 10.15 (The cg step length is optimal)
Show that the step length α_k *in the conjugate gradient method is optimal*[12].

Exercise 10.16 (Starting value in cg)
Show that the conjugate gradient method (10.24) *for* $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ *starting with* $\boldsymbol{x}_0$ *is the same as applying the method to the system* $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{r}_0 := \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$ *starting with* $\boldsymbol{y}_0 = \boldsymbol{0}$.[13]

[12] Hint: use induction on k to show that $\boldsymbol{p}_k = \boldsymbol{r}_k + \sum_{j=0}^{k-1} a_{k,j}\boldsymbol{r}_j$ for some constants $a_{k,j}$.

[13] Hint: The conjugate gradient method for $\boldsymbol{A}\boldsymbol{y} = \boldsymbol{r}_0$ can be written $\boldsymbol{y}_{k+1} := \boldsymbol{y}_k + \gamma_k\boldsymbol{q}_k$, $\gamma_k := \frac{\boldsymbol{s}_k^T\boldsymbol{s}_k}{\boldsymbol{q}_k^T\boldsymbol{A}\boldsymbol{q}_k}$, $\boldsymbol{s}_{k+1} := \boldsymbol{s}_k - \gamma_k\boldsymbol{A}\boldsymbol{q}_k$, $\boldsymbol{q}_{k+1} := \boldsymbol{s}_{k+1} + \delta_k\boldsymbol{q}_k$, $\delta_k := \frac{\boldsymbol{s}_{k+1}^T\boldsymbol{s}_{k+1}}{\boldsymbol{s}_k^T\boldsymbol{s}_k}$. Show that $\boldsymbol{y}_k = \boldsymbol{x}_k - \boldsymbol{x}_0$, $\boldsymbol{s}_k = \boldsymbol{r}_k$, and $\boldsymbol{q}_k = \boldsymbol{p}_k$, for $k = 0, 1, 2 \ldots$.

10.4 The Conjugate Gradient Algorithm

In this section we give numerical examples and discuss implementation.

The formulas in (10.24) form a basis for an algorithm.

Algorithm 10.17 (Conjugate Gradient Iteration)
The symmetric positive definite linear system $\boldsymbol{Ax} = \boldsymbol{b}$ is solved by the conjugate gradient method. $\boldsymbol{x}$ is a starting vector for the iteration. The iteration is stopped when $||\boldsymbol{r}_k||_2/||\boldsymbol{r}_0||_2 \leq$ tol or $k >$ itmax. K is the number of iterations used.

```
function [x,K]=cg(A,b,x,tol,itmax)
r=b-A*x; p=r; rho=r'*r;
rho0=rho;
for k=0:itmax
    if sqrt(rho/rho0)<= tol
        K=k; return
    end
    t=A*p; a=rho/(p'*t);
    x=x+a*p; r=r-a*t;
    rhos=rho; rho=r'*r;
    p=r+(rho/rhos)*p;
end
K=itmax+1;
```

The work involved in each iteration is

1. one matrix times vector ($\boldsymbol{t} = \boldsymbol{Ap}$),
2. two inner products (($\boldsymbol{p}^T\boldsymbol{t}$ and $\boldsymbol{r}^T\boldsymbol{r}$),
3. three vector-plus-scalar-times-vector ($\boldsymbol{x} = \boldsymbol{x} + a\boldsymbol{p}$, $\boldsymbol{r} = \boldsymbol{r} - a\boldsymbol{t}$ and $\boldsymbol{p} = \boldsymbol{r} + (rho/rhos)\boldsymbol{p}$),

The dominating part is the computation of $\boldsymbol{t} = \boldsymbol{Ap}$.

10.4.1 Numerical Example

We test the conjugate gradient method on two examples. For a similar test for the steepest descent method see Exercise 10.22. Consider the matrix is given by the Kronecker sum $\boldsymbol{T}_2 := \boldsymbol{T}_1 \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{T}_1$ where $\boldsymbol{T}_1 = \operatorname{tridiag}_m(a, d, a)$. We recall that this matrix is symmetric positive definite if $d > 0$ and $d \geq 2|a|$. We set $h = 1/(m+1)$ and $\boldsymbol{f} = [1, \ldots, 1]^T \in \mathbb{R}^n$.

We consider two problems.

1. $a = 1/9$, $d = 5/18$, the Averaging matrix.
2. $a = -1$, $d = 2$, the Poisson matrix.

n	2 500	10 000	40 000	1 000 000	4 000 000
K	19	18	18	16	15

Table 10.18: The number of iterations K for the averaging problem on a $\sqrt{n} \times \sqrt{n}$ grid for various n

10.4.2 Implementation Issues

Note that for our test problems $\boldsymbol{T}_2$ only has $O(5n)$ nonzero elements. Therefore, taking advantage of the sparseness of $\boldsymbol{T}_2$ we can compute $\boldsymbol{t}$ in Algorithm 10.17 in $O(n)$ arithmetic operations. With such an implementation the total number of arithmetic operations in one iteration is $O(n)$. We also note that it is not necessary to store the matrix $\boldsymbol{T}_2$.

To use the Conjugate Gradient Algorithm on the test matrix for large n it is advantageous to use a matrix equation formulation. We define matrices $\boldsymbol{V}, \boldsymbol{R}, \boldsymbol{P}, \boldsymbol{B}, \boldsymbol{T} \in \mathbb{R}^{m \times m}$ by $\boldsymbol{x} = \operatorname{vec}(\boldsymbol{V})$, $\boldsymbol{r} = \operatorname{vec}(\boldsymbol{R})$, $\boldsymbol{p} = \operatorname{vec}(\boldsymbol{P})$, $\boldsymbol{t} = \operatorname{vec}(\boldsymbol{T})$, and $h^2\boldsymbol{f} = \operatorname{vec}(\boldsymbol{B})$. Then $\boldsymbol{T}_2\boldsymbol{x} = h^2\boldsymbol{f} \iff \boldsymbol{T}_1\boldsymbol{V} + \boldsymbol{V}\boldsymbol{T}_1 = \boldsymbol{B}$, and $\boldsymbol{t} = \boldsymbol{T}_2\boldsymbol{p} \iff \boldsymbol{T} = \boldsymbol{T}_1\boldsymbol{P} + \boldsymbol{P}\boldsymbol{T}_1$.

This leads to the following algorithm for testing the conjugate gradient algorithm on the matrix

$$\boldsymbol{A} = \operatorname{tridiag}_m(a, d, a) \otimes \boldsymbol{I}_m + \boldsymbol{I}_m \otimes \operatorname{tridiag}_m(a, d, a) \in \mathbb{R}^{(m^2)\times(m^2)}.$$

Algorithm 10.19 (Testing Conjugate Gradient)

```
function [V,K]=cgtest(m,a,d,tol,itmax)
R=ones(m)/(m+1)^2; rho=sum(sum(R.*R)); rho0=rho; P=R;
V=zeros(m,m); T1=sparse(tridiagonal(a,d,a,m));
for k=1:itmax
    if sqrt(rho/rho0)<= tol
        K=k; return
    end
    T=T1*P+P*T1;
    a=rho/sum(sum(P.*T)); V=V+a*P; R=R-a*T;
    rhos=rho; rho=sum(sum(R.*R)); P=R+(rho/rhos)*P;
end
K=itmax+1;
```

For both the averaging- and Poison matrix we use $tol = 10^{-8}$.

For the averaging matrix we obtain the values in Table 10.18.

The convergence is quite rapid. It appears that the number of iterations can be bounded independently of n, and therefore we solve the problem in $O(n)$ operations. This is the best we can do for a problem with n unknowns.

n	2 500	10 000	40 000	160 000
K	94	188	370	735
$K/\sqrt{n}$	1.88	1.88	1.85	1.84

Table 10.20: The number of iterations K for the Poisson problem on a $\sqrt{n} \times \sqrt{n}$ grid for various n

Consider next the Poisson problem. In in Table 10.20 we list K, the required number of iterations, and $K/\sqrt{n}$.

The results show that K is much smaller than n and appears to be proportional to $\sqrt{n}$. This is the same speed as for SOR and we don't have to estimate any acceleration parameter.

10.4.3 The Spectral Condition Numbers

We will show in Section 10.5 the following upper bound for the $\boldsymbol{A}$-norm of the error in the conjugate gradient method.

Theorem 10.21 (Error bounds for cg)
We have

$$\frac{||\boldsymbol{x} - \boldsymbol{x}_k||_{\boldsymbol{A}}}{||\boldsymbol{x} - \boldsymbol{x}_0||_{\boldsymbol{A}}} \leq 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k < 2e^{-\frac{2}{\sqrt{\kappa}}k}, \quad .k \geq 0, \tag{10.25}$$

where $\kappa = M/m$, and where M and m are the largest and smallest eigenvalue of $\boldsymbol{A}$, respectively.

The second inequality follows from (10.16). The upper bound for the rate of convergence is determined by the square root of the spectral condition number. This is is much better than the estimate (10.10) for the steepest descent method. Especially for problems with large condition numbers.

So what is the spectral condition number of $\boldsymbol{T}_2$? The eigenvalues were given in (4.22) as

$$\lambda_{j,k} = 2d + 2a\cos(j\pi h) + 2a\cos(k\pi h), \quad j,k = 1,\ldots,m. \tag{10.26}$$

For the averaging problem it follows that the largest and smallest eigenvalue of $\boldsymbol{T}_2$ are $M = \frac{5}{9} + \frac{4}{9}\cos(\pi h)$ and $m = \frac{5}{9} - \frac{4}{9}\cos(\pi h)$. Thus

$$\kappa_A = \frac{M}{m} = \frac{5+4\cos(\pi h)}{5-4\cos(\pi h)} \leq 9,$$

and the condition number is independent of n. This verifies what we observed in Table 10.18. The number of iterations can be bounded independently of the size n of the problem.

The eigenvalues for the Poisson problem can also be found from (10.26). We find $M = 4(1 + \cos(\pi h)$ and $m = 4(1 - \cos(\pi h))$ and then

$$\kappa_P = \frac{M}{m} = \frac{1 + \cos(\pi h)}{1 - \cos(\pi h)} = 1/\tan(\pi h^2/2).$$

Thus $\kappa_P \approx 4(m+1)^2/\pi^2 = O(n)$ (see also Exercise 8.35) and we solve the discrete Poisson problem in $O(n^{3/2})$ arithmetic operations. This is the same as for the SOR method and for the fast method without the FFT. In comparison the Cholesky Algorithm requires $O(n^2)$ arithmetic operations both for the averaging and the Poisson problem.

Exercise 10.22 (Program code for testing steepest descent)
Write a function K=sdtest(m,a,d,tol,itmax) to test the Steepest descent method on the matrix $\boldsymbol{T}_2$. *Make the analogues of Table 10.18 and Table 10.20. For Table 10.20 it is enough to test for say* $n = 100, 400, 1600, 2500$, *and tabulate* K/n *instead of* $K/\sqrt{n}$ *in the last row. Conclude that the upper bound* (10.10) *is realistic. Compare also with the number of iterations for the J and GS method in Table 9.1.*

Exercise 10.23 (Compare Richardson and steepest descent)
Go back and study the Richardson method (9.25) *where a constant* α *is used. Suppose we use the* α^* *in* (9.26). *What seems to be the main difference between this method and the steepest descent method?*

Exercise 10.24 (Using cg to solve normal equations)
Consider solving the least squares problem by using the conjugate gradient method on the normal equations $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}$. *Explain why only the following modifications in Algorithm 10.17 are necessary*

1. *r=A'(b-A*x); p=r;*
2. *a=rho/(t'*t);*
3. *r=r-a*A'*t;*

Note that the condition number of the normal equations is $cond_2(\boldsymbol{A})^2$, *the square of the condition number of* $\boldsymbol{A}$.

10.5 Proof of Convergence

We first show a best approximation property which will be used for the convergence analysis.

10.5.1 Krylov spaces and the Best Approximation Property

The iterates in the conjugate gradient method are $\boldsymbol{A}$-ortogonal projections into certain subspaces of $\mathbb{R}^n$ called **Krylov spaces**.

They are defined by $\mathbb{W}_0 = \{\boldsymbol{0}\}$ and

$$\mathbb{W}_k = \operatorname{span}(\boldsymbol{r}_0, \boldsymbol{A}\boldsymbol{r}_0, \boldsymbol{A}^2\boldsymbol{r}_0, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{r}_0), \quad k = 1, 2, 3, \cdots .$$

The Krylov spaces are nested subspaces

$$\mathbb{W}_0 \subset \mathbb{W}_1 \subset \mathbb{W}_2 \subset \cdots \subset \mathbb{W}_n \subset \mathbb{R}^n$$

with $\dim(\mathbb{W}_k) \le k$ for all $k \ge 0$. Moreover, If $\boldsymbol{v} \in \mathbb{W}_k$ then $\boldsymbol{A}\boldsymbol{v} \in \mathbb{W}_{k+1}$.

Lemma 10.25 (Krylov space)
We have

$$\boldsymbol{x}_k - \boldsymbol{x}_0 \in \mathbb{W}_k, \quad \boldsymbol{r}_k, \boldsymbol{p}_k \in \mathbb{W}_{k+1}, \qquad k = 0, 1, \ldots, \tag{10.27}$$

and

$$\boldsymbol{r}_k^T \boldsymbol{w} = \boldsymbol{p}_k^T \boldsymbol{A}\boldsymbol{w} = 0, \quad \boldsymbol{w} \in \mathbb{W}_k. \tag{10.28}$$

Proof. Since $\boldsymbol{p}_0 = \boldsymbol{r}_0$ (10.27) clearly holds for $k = 0$. Suppose it holds for some $k \ge 0$. Then by (10.24), $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{A}\boldsymbol{p}_k \in \mathbb{W}_{k+2}$ and $\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k \boldsymbol{p}_k \in \mathbb{W}_{k+2}$ and $\boldsymbol{x}_{k+1} - \boldsymbol{x}_0 \overset{(10.18)}{=} \boldsymbol{x}_k - \boldsymbol{x}_0 + \alpha_k \boldsymbol{p}_k \in \mathbb{W}_{k+1}$. Thus (10.27) follows by induction. Since any $\boldsymbol{w} \in \mathbb{W}_k$ is a linear combination of $\{\boldsymbol{r}_0, \boldsymbol{r}_1, \ldots, \boldsymbol{r}_{k-1}\}$ and also $\{\boldsymbol{p}_0, \boldsymbol{p}_1, \ldots, \boldsymbol{p}_{k-1}\}$, (10.28) follows. □

Theorem 10.26 (Best approximation property)
Suppose $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ is symmetric positive definite and $\{\boldsymbol{x}_k\}$ is generated by the conjugate gradient method (10.24)*. Then*

$$\|\boldsymbol{x} - \boldsymbol{x}_k\|_{\boldsymbol{A}} = \min_{\boldsymbol{w} \in \mathbb{W}_k} \|\boldsymbol{x} - \boldsymbol{x}_0 - \boldsymbol{w}\|_{\boldsymbol{A}}. \tag{10.29}$$

Proof. Fix k, let $\boldsymbol{w} \in \mathbb{W}_k$ and $\boldsymbol{u} := \boldsymbol{x}_k - \boldsymbol{x}_0 - \boldsymbol{w}$,. By (10.27) $\boldsymbol{u} \in \mathbb{W}_k$ and then (10.28) implies that $\boldsymbol{r}_k^T \boldsymbol{u} = 0$. Since $(\boldsymbol{x} - \boldsymbol{x}_k)^T \boldsymbol{A}\boldsymbol{u} = \boldsymbol{r}_k^T \boldsymbol{u}$ we find

$$\begin{aligned}
\|\boldsymbol{x} - \boldsymbol{x}_0 - \boldsymbol{w}\|_{\boldsymbol{A}}^2 &= (\boldsymbol{x} - \boldsymbol{x}_k + \boldsymbol{u})^T \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}_k + \boldsymbol{u}) \\
&= (\boldsymbol{x} - \boldsymbol{x}_k)\boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}_k) + 2\boldsymbol{r}_k^T \boldsymbol{u} + \boldsymbol{u}^T \boldsymbol{A}\boldsymbol{u} \\
&= \|\boldsymbol{x} - \boldsymbol{x}_k\|_{\boldsymbol{A}}^2 + \|\boldsymbol{u}\|_{\boldsymbol{A}}^2 \ge \|\boldsymbol{x} - \boldsymbol{x}_k\|_{\boldsymbol{A}}^2.
\end{aligned}$$

Taking square roots the result follows. □

If $\boldsymbol{x}_0 = \boldsymbol{0}$ then (10.29) says that $\boldsymbol{x}_k$ is the element in $\mathbb{W}_k$ that is closest to the solution $\boldsymbol{x}$ in the $\boldsymbol{A}$-norm. More generally, if $\boldsymbol{x}_0 \ne \boldsymbol{0}$ then $\boldsymbol{x} - \boldsymbol{x}_k =$

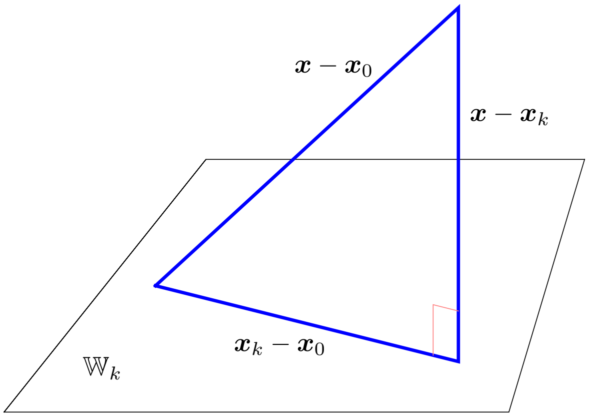
$\boldsymbol{x}-\boldsymbol{x}_0$
$\boldsymbol{x}-\boldsymbol{x}_k$
$\boldsymbol{x}_k-\boldsymbol{x}_0$
$\mathbb{W}_k$

Figure 10.2: The orthogonal projection of $\boldsymbol{x}-\boldsymbol{x}_0$ into $\mathbb{W}_k$.

$(\boldsymbol{x}-\boldsymbol{x}_0)-(\boldsymbol{x}_k-\boldsymbol{x}_0)$ and $\boldsymbol{x}_k-\boldsymbol{x}_0$ is the element in $\mathbb{W}_k$ that is closest to $\boldsymbol{x}-\boldsymbol{x}_0$ in the $\boldsymbol{A}$-norm. This is the orthogonal projection of $\boldsymbol{x}-\boldsymbol{x}_0$ into $\mathbb{W}_k$, see Figure 10.2.

Exercise 10.27 (Krylov space and cg iterations)
Consider the linear system $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$ *where*

$$\boldsymbol{A}=\begin{bmatrix}2&-1&0\\-1&2&-1\\0&-1&2\end{bmatrix},\quad \textit{and}\quad \boldsymbol{b}=\begin{bmatrix}4\\0\\0\end{bmatrix}.$$

a) *Determine the vectors defining the Krylov spaces for* $k\le 3$ *taking as initial approximation* $\boldsymbol{x}=\boldsymbol{0}$. *Answer:* $[\boldsymbol{b},\boldsymbol{A}\boldsymbol{b},\boldsymbol{A}^2\boldsymbol{b}]=\begin{bmatrix}4&8&20\\0&-4&-16\\0&0&4\end{bmatrix}.$

b) *Carry out three CG-iterations on* $\boldsymbol{A}\boldsymbol{x}=\boldsymbol{b}$. *Answer:*

$$[\boldsymbol{x}_0,\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3]=\begin{bmatrix}0&2&8/3&3\\0&0&4/3&2\\0&0&0&1\end{bmatrix},$$

$$[\boldsymbol{r}_0,\boldsymbol{r}_1,\boldsymbol{r}_2,\boldsymbol{r}_3]=\begin{bmatrix}4&0&0&0\\0&2&0&0\\0&0&4/3&0\end{bmatrix},$$

$$[\boldsymbol{A}\boldsymbol{p}_0, \boldsymbol{A}\boldsymbol{p}_1, \boldsymbol{A}\boldsymbol{p}_2] = \begin{bmatrix} 8 & 0 & 0 \\ -4 & 3 & 0 \\ 0 & -2 & 16/9 \end{bmatrix},$$

$$[\boldsymbol{p}_0, \boldsymbol{p}_1, \boldsymbol{p}_2, \boldsymbol{p}_3] = \begin{bmatrix} 4 & 1 & 4/9 & 0 \\ 0 & 2 & 8/9 & 0 \\ 0 & 0 & 12/9 & 0 \end{bmatrix},$$

c) *Verify that*

- *dim*$(\mathbb{W}_k) = k$ *for* $k = 0, 1, 2, 3$.
- $\boldsymbol{x}_3$ *is the exact solution of* $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$.
- $\boldsymbol{r}_0, \dots, \boldsymbol{r}_{k-1}$ *is an orthogonal basis for* $\mathbb{W}_k$ *for* $k = 1, 2, 3$.
- $\boldsymbol{p}_0, \dots, \boldsymbol{p}_{k-1}$ *is an* $\boldsymbol{A}$*-orthogonal basis for* $\mathbb{W}_k$ *for* $k = 1, 2, 3$.
- $\{|\boldsymbol{r}_k\boldsymbol{\epsilon}\}$ *is monotonically decreasing.*
- $\{|\boldsymbol{x}_k - \boldsymbol{x}\boldsymbol{\epsilon}\}$ *is monotonically decreasing.*

10.5.2 The proof of the convergence theorem

We prove Theorem 10.21. By Theorem 10.26 $\boldsymbol{x}_k$ is the best approximation to the solution $\boldsymbol{x}$ in the $\boldsymbol{A}$-norm. We convert this best approximation property into a minimization problem involving polynomials. In the following we let Π_k denote the class of univariate polynomials of degree $\le k$ with real coefficients.

Lemma 10.28 (Krylov space and polynomials)
Suppose $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ *where* $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ *is symmetric positive definite with orthonormal eigenpairs* $(\lambda_j, \boldsymbol{u}_j)$, $j = 1, 2, \dots, n$, *and let* $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$ *with* $\boldsymbol{x}_0 \in \mathbb{R}^n$. *If* $\boldsymbol{w} = \sum_{j=0}^{k-1} a_j \boldsymbol{A}^j \boldsymbol{r}_0 \in \mathbb{W}_k$ *for some* $k \ge 0$ *then*

$$\boldsymbol{w} = P(\boldsymbol{A})\boldsymbol{r}_0, \quad P(\boldsymbol{A}) = a_0 I + a_1\boldsymbol{A} + a_2\boldsymbol{A}^2 + \cdots + a_{k-1}\boldsymbol{A}^{k-1}.$$

where $P(\boldsymbol{A})$ *is a matrix polynomial corresponding to the ordinary polynomial* $P(t) = a_0 + a_1 t + \cdots + a_{k-1}t^{k-1}$ *of degree* $\le k-1$. *Moreover,*

$$||\boldsymbol{x} - \boldsymbol{w}||_{\boldsymbol{A}}^2 = \sum_{j=1}^{n} \frac{\sigma_j^2}{\lambda_j} Q(\lambda_j)^2, \quad Q(t) := 1 - tP(t), \tag{10.30}$$

where $\boldsymbol{b} = \sum_{j=1}^{n} \sigma_j \boldsymbol{u}_j$.

Proof. Clearly $\boldsymbol{w} = P(\boldsymbol{A})\boldsymbol{r}_0$. We find

$$||\boldsymbol{x} - P(\boldsymbol{A})\boldsymbol{b}||_{\boldsymbol{A}}^2 = \boldsymbol{c}^T\boldsymbol{A}^{-1}\boldsymbol{c}, \quad \boldsymbol{c} = Q(\boldsymbol{A})\boldsymbol{b}, \quad Q(\boldsymbol{A}) = I - \boldsymbol{A}P(\boldsymbol{A}). \tag{10.31}$$

Using the eigenvector expansion for $\boldsymbol{b}$ and (0.66) we obtain

$$\boldsymbol{c} = \sum_{j=1}^{n} \sigma_j Q(\lambda_j)\boldsymbol{u}_j, \quad \boldsymbol{A}^{-1}\boldsymbol{c} = \sum_{i=1}^{n} \sigma_i \frac{Q(\lambda_i)}{\lambda_i}\boldsymbol{u}_i. \tag{10.32}$$

Combining (10.31),(10.32), and using orthonormality we find

$$||\boldsymbol{x} - \boldsymbol{w}||_{\boldsymbol{A}}^2 = \boldsymbol{c}^T\boldsymbol{A}^{-1}\boldsymbol{c} = \big(\sum_{j=1}^{n} \sigma_j Q(\lambda_j)\boldsymbol{u}_j\big)^T \big(\sum_{i=1}^{n} \sigma_i \frac{Q(\lambda_i)}{\lambda_i}\boldsymbol{u}_i\big) = \sum_{j=1}^{n} \sigma_j^2 \frac{Q(\lambda_j)^2}{\lambda_j}.$$

□

We will use the following Theorem to estimate the rate of convergence.

Theorem 10.29 (cg and best polynomial approximation)
Suppose $[a, b]$ with $0 < a < b$ is an interval containing all the eigenvalues of $\boldsymbol{A}$. Then for all $Q \in \Pi_k$ with $Q(0) = 1$ we have

$$\frac{||\boldsymbol{x} - \boldsymbol{x}_k||_{\boldsymbol{A}}}{||\boldsymbol{x} - \boldsymbol{x}_0||_{\boldsymbol{A}}} \le \max_{a \le x \le b} |Q(x)|.$$

Proof. We find $||\boldsymbol{x} - \boldsymbol{x}_0||_{\boldsymbol{A}}^2 = \boldsymbol{r}_0\boldsymbol{A}^{-1}\boldsymbol{r}_0 = \sum_{j=1}^n \frac{\sigma_j^2}{\lambda_j}$. Therefore, by the best approximation property and (10.30), for any $\boldsymbol{w} \in \mathbb{W}_k$

$$||\boldsymbol{x} - \boldsymbol{x}_k||_{\boldsymbol{A}}^2 \le ||\boldsymbol{x} - \boldsymbol{w}||_{\boldsymbol{A}}^2 \le \max_{a \le x \le b} |Q(x)|^2 \sum_{j=1}^{n} \frac{\sigma_j^2}{\lambda_j} = \max_{a \le x \le b} |Q(x)|^2 ||\boldsymbol{x} - \boldsymbol{x}_0||_{\boldsymbol{A}}^2$$

and the result follows by taking square roots. □

In the next section we use properties of the Chebyshev polynomials to show that

$$\frac{||\boldsymbol{x} - \boldsymbol{x}_k||_{\boldsymbol{A}}}{||\boldsymbol{x} - \boldsymbol{x}_0||_{\boldsymbol{A}}} \le \min_{\substack{Q \in \Pi_k \\ Q(0)=1}} \max_{m \le x \le M} |Q(x)| = \frac{2}{\big(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\big)^k + \big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\big)^k}, \tag{10.33}$$

where $\kappa = M/m$. But this implies the first inequality in (10.25). The second inequality follows from (10.16)

10.5.3 Chebyshev Polynomials

Suppose $a < b$, $c \notin [a, b]$ and $k \in \mathbb{N}$. Consider the set $\mathcal{S}_k$ of all polynomials Q of degree $\le k$ such that $Q(c) = 1$. For any continuous function f on $[a, b]$ we define

$$\|f\|_\infty = \max_{a \le x \le b} |f(x)|.$$

We want to find a polynomial $Q^* \in \mathcal{S}_k$ such that (cf. Corollary 10.29, where $c = 0 < a < b$)

$$\|Q^*\|_\infty = \min_{Q \in \mathcal{S}_k} \|Q\|_\infty.$$

We will show that Q^* is a suitably shifted and normalized version of the **Chebyshev poynomial**. The Chebyshev polynomial T_n of degree n can be defined recursively by

$$T_{n+1}(t) = 2tT_n(t) - T_{n-1}(t), \quad n \geq 1, \quad t \in \mathbb{R},$$

starting with $T_0(t) = 1$ and $T_1(t) = t$. Thus $T_2(t) = 2t^2 - 1$, $T_3(t) = 4t^3 - 3t$ etc. In general T_n is a polynomial of degree n.

There are some convenient closed form expressions for T_n.

Lemma 10.30 (Closed forms of Chebyshev polynomials)
For $n \geq 0$

1. $T_n(t) = \cos(n\mathrm{arccos}\ t)$ *for* $t \in [-1, 1]$,
2. $T_n(t) = \frac{1}{2}\left[\left(t + \sqrt{t^2-1}\right)^n + \left(t + \sqrt{t^2-1}\right)^{-n}\right]$ *for* $|t| \geq 1$.

Proof. 1. With $P_n(t) = \cos(n \arccos t)$ we have $P_n(t) = \cos n\phi$, where $t = \cos\phi$. Therefore

$$P_{n+1}(t) + P_{n-1}(t) = \cos(n+1)\phi + \cos(n-1)\phi = 2\cos\phi\cos n\phi = 2tP_n(t)$$

and it follows that P_n satisfies the same recurrence relation as T_n. Since $P_0 = T_0$ and $P_1 = T_1$ we have $P_n = T_n$ for all $n \geq 0$.

2. Fix t with $|t| \geq 1$ and let $x_n := T_n(t)$ for $n \geq 0$. The recurrence relation for the Chebyshev polynomials can then be written

$$x_{n+1} - 2tx_n + x_{n-1} = 0 \text{ for } n \geq 1, \text{ with } x_0 = 1, x_1 = t. \tag{10.34}$$

To solve this difference equation we insert $x_n = z^n$ into (10.34) and obtain $z^{n+1} - 2tz^n + z^{n-1} = 0$ or $z^2 - 2tz + 1 = 0$. The roots of this equation are

$$z_1 = t + \sqrt{t^2-1}, \quad z_2 = t - \sqrt{t^2-1} = \left(t + \sqrt{t^2-1}\right)^{-1}.$$

Now z_1^n, z_2^n and more generally $c_1z_1^n + c_2z_2^n$ are solutions of (10.34) for any constants c_1 and c_2. We find these constants from the initial conditions $x_0 = c_1 + c_2 = 1$ and $x_1 = c_1z_1 + c_2z_2 = t$. Since $z_1 + z_2 = 2t$ the solution is $c_1 = c_2 = \frac{1}{2}$. □

We show that the unique solution to our minimization problem is

$$Q^*(x) = \frac{T_k(u(x))}{T_k(u(c))}, \qquad u(x) = \frac{b + a - 2x}{b - a}. \tag{10.35}$$

Clearly $Q^* \in \boldsymbol{x}_k$.

Theorem 10.31 (A minimal norm problem)
Suppose $a < b$, $c \notin [a,b]$ and $k \in \mathbb{N}$. If $Q \in \mathcal{S}_k$ and $Q \neq Q^*$ then $\|Q\|_\infty > \|Q^*\|_\infty$.

Proof. Recall that a nonzero polynomial p of degree k can have at most k zeros. If $p(z) = p'(z) = 0$, we say that p has a double zero at z. Counting such a zero as two zeros it is still true that a nonzero polynomial of degree k has at most k zeros.

$|Q^*|$ takes on its maximum $1/|T_k(u(c))|$ at the $k+1$ points $\mu_0, \ldots, \mu_k$ in $[a,b]$ such that $u(\mu_i) = \cos(i\pi/k)$ for $i = 0, 1, \ldots, k$. Suppose $Q \in S_k$ and that $\|Q\| \leq \|Q^*\|$. We have to show that $Q \equiv Q^*$. Let $f \equiv Q - Q^*$. We want to show that f has at least k zeros in $[a,b]$. Since f is a polynomial of degree $\leq k$ and $f(c) = 0$, this means that $f \equiv 0$ or equivalently $Q \equiv Q^*$.

Consider $I_j = [\mu_{j-1}, \mu_j]$ for a fixed j. Let

$$\sigma_j = f(\mu_{j-1}) f(\mu_j).$$

We have $\sigma_j \leq 0$. For if say $Q^*(\mu_j) > 0$ then

$$Q(\mu_j) \leq \|Q\|_\infty \leq \|Q^*\|_\infty = Q^*(\mu_j)$$

so that $f(\mu_j) \leq 0$. Moreover,

$$-Q(\mu_{j-1}) \leq \|Q\|_\infty \leq \|Q^*\|_\infty = -Q^*(\mu_{j-1}).$$

Thus $f(\mu_{j-1}) \geq 0$ and It follows that $\sigma_j \leq 0$. Similarly, $\sigma_j \leq 0$ if $Q^*(\mu_j) < 0$.

If $\sigma_j < 0$, f must have a zero in I_j since it is continuous. Suppose $\sigma_j = 0$. Then $f(\mu_{j-1}) = 0$ or $f(\mu_j) = 0$. If $f(\mu_j) = 0$ then $Q(\mu_j) = Q^*(\mu_j)$. But then μ_j is a maximum or minimum both for Q and Q^*. If $\mu_j \in (a,b)$ then $Q'(\mu_j) = Q^{*\prime}(\mu_j) = 0$. Thus $f(\mu_j) = f'(\mu_j) = 0$, and f has a double zero at μ_j. We can count this as one zero for I_j and one for I_{j+1}. If $\mu_j = b$, we still have a zero in I_j. Similarly, if $f(\mu_{j-1}) = 0$, a double zero of f at μ_{j-1} appears if $\mu_{j-1} \in (a,b)$. We count this as one zero for I_{j-1} and one for I_j.

In this way we associate one zero of f for each of the k intervals I_j, $j = 1, 2, \ldots, k$. We conclude that f has at least k zeros in $[a,b]$. □

Exercise 10.32 (An explicit formula for the Chebyshev polynomial)
Show that

$$T_n(t) = \cosh(n \mathit{arccosh}\ t) \text{ for } |t| \geq 1,$$

where arccosh is the inverse function of $\cosh x := (e^x + e^{-x})/2$.

Theorem 10.31 with $a = m$, $b = M$, and $c = 0$ implies that the minimizing polynomial in (10.33) is given by

$$Q^*(x) = T_k\left(\frac{M+m-2x}{M-m}\right) / T_k\left(\frac{M+m}{M-m}\right), \tag{10.36}$$

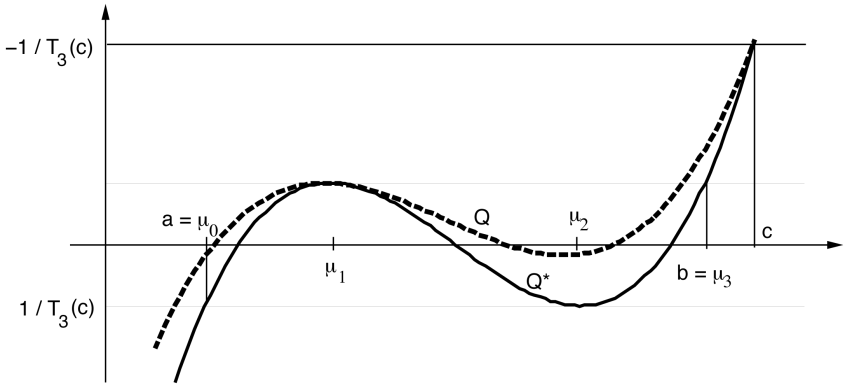

Figure 10.3: This is an illustration of the proof of Theorem 10.31 for $k = 3$. $f \equiv Q - Q^*$ has a double zero at μ_1 and one zero between μ_2 and μ_3.

where m and M, the smallest and largest eigenvslue of $\boldsymbol{A}$. By Lemma 10.30

$$\max_{m \le x \le M} \left| T_k\left(\frac{M+m-2x}{M-m}\right)\right| = \max_{-1\le t\le 1} \left|T_k(t)\right| = 1. \tag{10.37}$$

Moreover with $t = (M+m)/(M-m)$ we have

$$t + \sqrt{t^2-1} = \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}, \quad \kappa = M/m.$$

Thus again by Lemma 10.30 we find

$$T_k\left(\frac{M+m}{M-m}\right) = T_k\left(\frac{\kappa+1}{\kappa-1}\right) = \frac{1}{2}\left[\left(\frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1}\right)^k + \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k\right] \tag{10.38}$$

and (10.33) and the first inequality in follows.

10.5.4 Monotonicity of the error

The error analysis for the conjugate gradient method is based on the $\boldsymbol{A}$-norm. We end this chapter by considering the Euclidian norm of the error, and show that it is strictly decreasing.

Theorem 10.33 (The error in cg is strictly decreasing)
Let in the conjugate gradient method m be the smallest integer such that $\boldsymbol{r}_{m+1} = \boldsymbol{0}$. For $k \le m$ we have $\|\boldsymbol{\epsilon}_{k+1}\|_2 < \|\boldsymbol{\epsilon}_k\|_2$. More precisely,

$$\|\boldsymbol{\epsilon}_k\|_2^2 - \|\boldsymbol{\epsilon}_{k+1}\|_2^2 = \frac{\|\boldsymbol{p}_k\|_2^2}{\|\boldsymbol{p}_k\|_{\boldsymbol{A}}^2}(\|\boldsymbol{\epsilon}_k\|_{\boldsymbol{A}}^2 + \|\boldsymbol{\epsilon}_{k+1}\|_{\boldsymbol{A}}^2)$$

where $\boldsymbol{\epsilon}_j = \boldsymbol{x} - \boldsymbol{x}_j$ *and* $\boldsymbol{A}\boldsymbol{x} = b$..

Proof. For $j \le m$

$$\boldsymbol{\epsilon}_j = \boldsymbol{x}_{m+1} - \boldsymbol{x}_j = \boldsymbol{x}_m - \boldsymbol{x}_j + \alpha_m \boldsymbol{p}_m = \boldsymbol{x}_{m-1} - \boldsymbol{x}_j + \alpha_{m-1}\boldsymbol{p}_{m-1} + \alpha_m \boldsymbol{p}_m = ...$$

so that

$$\boldsymbol{\epsilon}_j = \sum_{i=j}^{m} \alpha_i \boldsymbol{p}_i, \quad \alpha_i = \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\boldsymbol{p}_i^T \boldsymbol{A} \boldsymbol{p}_i}. \tag{10.39}$$

By (10.39) and $\boldsymbol{A}$-orthogonality

$$\|\boldsymbol{\epsilon}_j\|_{\boldsymbol{A}}^2 = \boldsymbol{\epsilon}_j \boldsymbol{A} \boldsymbol{\epsilon}_j = \sum_{i=j}^{m} \alpha_i^2 \boldsymbol{p}_i^T \boldsymbol{A} \boldsymbol{p}_i = \sum_{i=j}^{m} \frac{(\boldsymbol{r}_i^T \boldsymbol{r}_i)^2}{\boldsymbol{p}_i^T \boldsymbol{A} \boldsymbol{p}_i}. \tag{10.40}$$

By (10.23) and Lemma 10.25

$$\boldsymbol{p}_i^T \boldsymbol{p}_k = (\boldsymbol{r}_i + \beta_{i-1}\boldsymbol{p}_{i-1})^T \boldsymbol{p}_k = \beta_{i-1}\boldsymbol{p}_{i-1}^T \boldsymbol{p}_k = \cdots = \beta_{i-1}\cdots\beta_k(\boldsymbol{p}_k^T \boldsymbol{p}_k),$$

and since $\beta_{i-1}\cdots\beta_k = (\boldsymbol{r}_i^T \boldsymbol{r}_i)/(\boldsymbol{r}_k^T \boldsymbol{r}_k)$ we obtain

$$\boldsymbol{p}_i^T \boldsymbol{p}_k = \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\boldsymbol{r}_k^T \boldsymbol{r}_k} \boldsymbol{p}_k^T \boldsymbol{p}_k, \quad i \ge k. \tag{10.41}$$

Since

$$\|\boldsymbol{\epsilon}_k\|_2^2 = \|\boldsymbol{\epsilon}_{k+1} + \boldsymbol{x}_{k+1} - \boldsymbol{x}_k\|_2^2 = \|\boldsymbol{\epsilon}_{k+1} + \alpha_k \boldsymbol{p}_k\|_2^2,$$

we obtain

$$\begin{aligned}
\|\boldsymbol{\epsilon}_k\|_2^2 - \|\boldsymbol{\epsilon}_{k+1}\|_2^2 =& \alpha_k\big(2\boldsymbol{p}_k^T \boldsymbol{\epsilon}_{k+1} + \alpha_k \boldsymbol{p}_k^T \boldsymbol{p}_k\big) \\
\overset{(10.39)}{=}& \alpha_k\big(2\sum_{i=k+1}^{m} \alpha_i \boldsymbol{p}_i^T \boldsymbol{p}_k + \alpha_k \boldsymbol{p}_k^T \boldsymbol{p}_k\big) = \big(\sum_{i=k}^{m} + \sum_{i=k+1}^{m}\big)\alpha_k \alpha_i \boldsymbol{p}_i^T \boldsymbol{p}_k \\
\overset{(10.41)}{=}& \big(\sum_{i=k}^{m} + \sum_{i=k+1}^{m}\big) \frac{\boldsymbol{r}_k^T \boldsymbol{r}_k}{\boldsymbol{p}_k^T \boldsymbol{A} \boldsymbol{p}_k} \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\boldsymbol{p}_i^T \boldsymbol{A} \boldsymbol{p}_i} \frac{\boldsymbol{r}_i^T \boldsymbol{r}_i}{\boldsymbol{r}_k^T \boldsymbol{r}_k} \boldsymbol{p}_k^T \boldsymbol{p}_k \\
\overset{(10.40)}{=}& \frac{\|\boldsymbol{p}_k\|_2^2}{\|\boldsymbol{p}_k\|_{\boldsymbol{A}}^2}\big(\|\boldsymbol{\epsilon}_k\|_{\boldsymbol{A}}^2 + \|\boldsymbol{\epsilon}_{k+1}\|_{\boldsymbol{A}}^2\big).
\end{aligned}$$

and the Theorem is proved. □

10.6 Preconditioning

For problems $\boldsymbol{Ax} = \boldsymbol{b}$ of size n, where both n and $\mathrm{cond}_2(\boldsymbol{A})$ are large, it is often possible to improve the performance of the conjugate gradient method by using a technique known as **preconditioning**. Instead of $\boldsymbol{Ax} = \boldsymbol{b}$ we consider an equivalent system $\boldsymbol{BAx} = \boldsymbol{Bb}$, where $\boldsymbol{B}$ is nonsingular and $\mathrm{cond}_2(\boldsymbol{BA})$ is smaller than $\mathrm{cond}_2(\boldsymbol{A})$. The matrix $\boldsymbol{B}$ will in many cases be the inverse of another matrix, $\boldsymbol{B} = \boldsymbol{M}^{-1}$. We cannot use CG on $\boldsymbol{BAx} = \boldsymbol{Bb}$ directly since $\boldsymbol{BA}$ in general is not symmetric even if both $\boldsymbol{A}$ and $\boldsymbol{B}$ are. But if $\boldsymbol{B}$ (and hence $\boldsymbol{M}$) is symmetric positive definite then we can apply CG to a symmetrized system and then transform the recurrence formulas to an iterative method for the original system $\boldsymbol{Ax} = \boldsymbol{b}$. This iterative method is known as the **preconditioned conjugate gradient method**. We shall see that the convergence properties of this method is determined by the eigenvalues of $\boldsymbol{BA}$.

Suppose $\boldsymbol{B}$ is symmetric positive definite. By Theorem 3.31 there is a nonsingular matrix $\boldsymbol{C}$ such that $\boldsymbol{B} = \boldsymbol{C}^T\boldsymbol{C}$. ($\boldsymbol{C}$ is only needed for the derivation and will not appear in the final formulas). Now

$$\boldsymbol{BAx} = \boldsymbol{Bb} \Leftrightarrow \boldsymbol{C}^T(\boldsymbol{CAC}^T)\boldsymbol{C}^{-T}\boldsymbol{x} = \boldsymbol{C}^T\boldsymbol{Cb} \Leftrightarrow (\boldsymbol{CAC}^T)\boldsymbol{y} = \boldsymbol{Cb}, \;\&\; \boldsymbol{x} = \boldsymbol{C}^T\boldsymbol{y}.$$

We have 3 linear systems

$$\boldsymbol{Ax} = \boldsymbol{b} \tag{10.42}$$

$$\boldsymbol{BAx} = \boldsymbol{Bb} \tag{10.43}$$

$$(\boldsymbol{CAC}^T)\boldsymbol{y} = \boldsymbol{Cb}, \;\&\; \boldsymbol{x} = \boldsymbol{C}^T\boldsymbol{y}. \tag{10.44}$$

Note that (10.42) and (10.44) are symmetric positive definite linear systems. In addition to being symmetric positive definite the matrix $\boldsymbol{CAC}^T$ is similar to $\boldsymbol{BA}$. Indeed,

$$\boldsymbol{C}^T(\boldsymbol{CAC}^T)\boldsymbol{C}^{-T} = \boldsymbol{BA}.$$

Thus $\boldsymbol{CAC}^T$ and $\boldsymbol{BA}$ have the same eigenvalues. Therefore if we apply the conjugate gradient method to (10.44) then the rate of convergence will be determined by the eigenvalues of $\boldsymbol{BA}$.

We apply the conjugate gradient method to $(\boldsymbol{CAC}^T)\boldsymbol{y} = \boldsymbol{Cb}$. Denoting the search direction by $\boldsymbol{q}_k$ and the residual by $\boldsymbol{z}_k = \boldsymbol{Cb} - \boldsymbol{CAC}^T\boldsymbol{y}_k$ we obtain the following from (10.21), (10.22), and (10.23).

$$\begin{aligned}
\boldsymbol{y}_{k+1} &= \boldsymbol{y}_k + \alpha_k\boldsymbol{q}_k, \quad \alpha_k = \boldsymbol{z}_k^T\boldsymbol{z}_k/\boldsymbol{q}_k^T(\boldsymbol{CAC}^T)\boldsymbol{q}_k,\\
\boldsymbol{z}_{k+1} &= \boldsymbol{z}_k - \alpha_k(\boldsymbol{CAC}^T)\boldsymbol{q}_k,\\
\boldsymbol{q}_{k+1} &= \boldsymbol{z}_{k+1} + \beta_k\boldsymbol{q}_k, \quad \beta_k = \boldsymbol{z}_{k+1}^T\boldsymbol{z}_{k+1}/\boldsymbol{z}_k^T\boldsymbol{z}_k.
\end{aligned}$$

With

$$\boldsymbol{x}_k := \boldsymbol{C}^T\boldsymbol{y}_k, \quad \boldsymbol{p}_k := \boldsymbol{C}^T\boldsymbol{q}_k, \quad \boldsymbol{s}_k := \boldsymbol{C}^T\boldsymbol{z}_k, \quad \boldsymbol{r}_k := \boldsymbol{C}^{-1}\boldsymbol{z}_k \tag{10.45}$$

this can be transformed into

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k, \quad \alpha_k = \frac{\boldsymbol{s}_k^T \boldsymbol{r}_k}{\boldsymbol{p}_k^T \boldsymbol{A} \boldsymbol{p}_k}, \tag{10.46}$$

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{A} \boldsymbol{p}_k, \tag{10.47}$$

$$\boldsymbol{s}_{k+1} = \boldsymbol{s}_k - \alpha_k \boldsymbol{B} \boldsymbol{A} \boldsymbol{p}_k, \tag{10.48}$$

$$\boldsymbol{p}_{k+1} = \boldsymbol{s}_{k+1} + \beta_k \boldsymbol{p}_k, \quad \beta_k = \frac{\boldsymbol{s}_{k+1}^T \boldsymbol{r}_{k+1}}{\boldsymbol{s}_k^T \boldsymbol{r}_k}. \tag{10.49}$$

Here $\boldsymbol{x}_k$ will be an approximation to the solution $\boldsymbol{x}$ of $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, $\boldsymbol{r}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$ is the residual in the original system, and $\boldsymbol{s}_k = \boldsymbol{B}\boldsymbol{b} - \boldsymbol{B}\boldsymbol{A}\boldsymbol{x}_k$ is the residual in the preconditioned system. This follows since by (10.45)

$$\boldsymbol{r}_k = \boldsymbol{C}^{-1}\boldsymbol{z}_k = \boldsymbol{b} - \boldsymbol{C}^{-1}\boldsymbol{C}\boldsymbol{A}\boldsymbol{C}^T\boldsymbol{y}_k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_k$$

and $\boldsymbol{s}_k = \boldsymbol{C}^T\boldsymbol{z}_k = \boldsymbol{C}^T\boldsymbol{C}\boldsymbol{r}_k = \boldsymbol{B}\boldsymbol{r}_k$. We start with $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$, $\boldsymbol{p}_0 = \boldsymbol{s}_0 = \boldsymbol{B}\boldsymbol{r}_0$ and obtain the following preconditioned conjugate gradient algorithm for determining approximations $\boldsymbol{x}_k$ to the solution of a symmetric positive definite system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$.

Algorithm 10.34 (Preconditioned conjugate gradient)
The symmetric positive definite linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is solved by the preconditioned conjugate gradient method on the system $\boldsymbol{B}\boldsymbol{A}\boldsymbol{x} = \boldsymbol{B}\boldsymbol{b}$, where $\boldsymbol{B}$ is symmetric positive definite . $\boldsymbol{x}$ is a starting vector for the iteration. The iteration is stopped when $||\boldsymbol{r}_k||_2/||\boldsymbol{r}_0||_2 \le$ tol or $k >$ itmax. K is the number of iterations used.

```
function [x,K]=pcg(A,B,b,x,tol,itmax)
r=b-A*x; p=B*r; s=p; rho=s'*r;
rho0=rho;
for k=0:itmax
    if sqrt(rho/rho0)<= tol
        K=k; return
    end
    t=A*p; a=rho/(p'*t);
    x=x+a*p; r=r-a*t;
    w=B*t; s=s-a*w;
    rhos=rho; rho=s'*r;
    p=r+(rho/rhos)*p;
end
K=itmax+1;
```

This algorithm is quite similar to Algorithm 10.17. It differs in the calculation of ρ. The main additional work is contained in $w = B * t$. We'll discuss this further in connection with an example. There the inverse of $\boldsymbol{B}$ is known and we have to solve a linear system to find $\boldsymbol{w}$.

We have the following convergence result for this algorithm.

Theorem 10.35 (Error bound preconditioned cg)
Suppose we apply a symmetric positive definite preconditioner $\boldsymbol{B}$ to the symmetric positive definite system $\boldsymbol{Ax} = \boldsymbol{b}$. Then the quantities $\boldsymbol{x}_k$ computed in Algorithm 10.34 satisfy the following bound:

$$\frac{||\boldsymbol{x} - \boldsymbol{x}_k||_{\boldsymbol{A}}}{||\boldsymbol{x} - \boldsymbol{x}_0||_{\boldsymbol{A}}} \le 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k, \quad \textit{for} \quad k \ge 0,$$

where $\kappa = \lambda_{max}/\lambda_{min}$ is the ratio of the largest and smallest eigenvalue of $\boldsymbol{BA}$.

Proof. Since Algorithm 10.34 is equivalent to solving (10.44) by the conjugate gradient method Theorem 10.21 implies that

$$\frac{||\boldsymbol{y} - \boldsymbol{y}_k||_{\boldsymbol{CAC}^T}}{||\boldsymbol{y} - \boldsymbol{y}_0||_{\boldsymbol{CAC}^T}} \le 2\left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^k, \quad \text{for} \quad k \ge 0,$$

where $\boldsymbol{y}_k$ is the conjugate gradient approximation to the solution $\boldsymbol{y}$ of (10.44) and κ is the ratio of the largest and smallest eigenvalue of $\boldsymbol{CAC}^T$. Since $\boldsymbol{BA}$ and $\boldsymbol{CAC}^T$ are similar this is the same as the κ in the theorem. By (10.45) we have

$$\begin{aligned}\|\boldsymbol{y} - \boldsymbol{y}_k\|^2_{\boldsymbol{CAC}^T} &= (\boldsymbol{y} - \boldsymbol{y}_k)^T(\boldsymbol{CAC}^T)(\boldsymbol{y} - \boldsymbol{y}_k)\\ &= (\boldsymbol{C}^T(\boldsymbol{y} - \boldsymbol{y}_k))^T\boldsymbol{A}(\boldsymbol{C}^T(\boldsymbol{y} - \boldsymbol{y}_k)) = \|\boldsymbol{x} - \boldsymbol{x}_k\|^2_{\boldsymbol{A}}\end{aligned}$$

and the proof is complete. □

We conclude that $\boldsymbol{B}$ should satisfy the following requirements for a problem of size n:

1. The eigenvalues of $\boldsymbol{BA}$ should be located in a narrow interval. Preferably one should be able to bound the length of the interval independently of n.
2. The evaluation of $\boldsymbol{Bx}$ for a given vector $\boldsymbol{x}$ should not be expensive in storage and arithmetic operations, ideally $O(n)$ for both.

10.7 Preconditioning Example

Consider the problem

$$\begin{array}{rcll} -\frac{\partial}{\partial x}\big(c(x,y)\frac{\partial u}{\partial x}\big) - \frac{\partial}{\partial y}\big(c(x,y)\frac{\partial u}{\partial y}\big) &=& f(x,y) & (x,y) \in \Omega = (0,1)^2 \\ u(x,y) &=& 0 & (x,y) \in \partial\Omega. \end{array} \tag{10.50}$$

Here Ω is the open unit square while $\partial\Omega$ is the boundary of Ω. The functions f and c are given and we seek a function $u = u(x,y)$ such that (10.50) holds. We assume that c and f are defined and continuous on Ω and that $c(x,y) > 0$ for all $(x,y) \in \Omega$. The problem (10.50) reduces to the Poisson problem in the special case where $c(x,y) = 1$ for $(x,y) \in \Omega$.

As for the Poisson problem we solve (10.50) numerically on a grid of points

$$\{(jh,kh) :\ j,k = 0,1,\ldots,m+1\}, \quad \text{where} \quad h = 1/(m+1),$$

and where m is a positive integer. Let (x,y) be one of the interior grid points. For univariate functions f,g we use the central difference approximations

$$\begin{aligned}\frac{\partial}{\partial t}\Big(f(t)\frac{\partial}{\partial t}g(t)\Big) &\approx \Big(f(t+\frac{h}{2})\frac{\partial}{\partial t}g(t+h/2) - f(t-\frac{h}{2})\frac{\partial}{\partial t}g(t-\frac{h}{2})\Big)/h\\ &\approx \Big(f(t+\frac{h}{2})\big(g(t+h)-g(t)\big) - f(t-\frac{h}{2})\big(g(t)-g(t-h)\big)\Big)/h^2\end{aligned}$$

to obtain

$$\frac{\partial}{\partial x}\big(c\frac{\partial u}{\partial x}\big)_{j,k} \approx \frac{c_{j+\frac12,k}(v_{j+1,k}-v_{j,k}) - c_{j-\frac12,k}(v_{j,k}-v_{j-1,k})}{h^2}$$

and

$$\frac{\partial}{\partial y}\big(c\frac{\partial u}{\partial y}\big)_{j,k} \approx \frac{c_{j,k+\frac12}(v_{j,k+1}-v_{j,k}) - c_{j,k-\frac12}(v_{j,k}-v_{j,k-1})}{h^2},$$

where $c_{p,q} = c(ph,qh)$ and $v_{j,k} \approx u(jh,kh)$. With these approximations the discrete analog of (10.50) turns out to be

$$\begin{aligned} -(\boldsymbol{P}_h v)_{j,k} &= h^2 f_{j,k} \quad j,k = 1,\ldots,m\\ v_{j,k} &= 0 \qquad\quad j = 0, m+1 \text{ all } k \text{ or } k = 0, m+1 \text{ all j},\end{aligned} \tag{10.51}$$

where

$$\begin{aligned} -(\boldsymbol{P}_h v)_{j,k} = \ &(c_{j,k-\frac12} + c_{j-\frac12,k} + c_{j+\frac12,k} + c_{j,k+\frac12})v_{j,k}\\ &- c_{j,k-\frac12}v_{j,k-1} - c_{j-\frac12,k}v_{j-1,k} - c_{j+\frac12,k}v_{j+1,k} - c_{j,k+\frac12}v_{j,k+1}\end{aligned} \tag{10.52}$$

and $f_{j,k} = f(jh,kh)$.

As before we let $\boldsymbol{V} = (v_{j,k}) \in \mathbb{R}^{m\times m}$ and $\boldsymbol{F} = (f_{j,k}) \in \mathbb{R}^{m\times m}$. The corresponding linear system can be written $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ where $\boldsymbol{x} = \text{vec}(\boldsymbol{V})$, $\boldsymbol{b} = h^2\text{vec}(\boldsymbol{F})$, and the n-by-n coefficient matrix $\boldsymbol{A}$ is given by

$$\begin{aligned} a_{i,i} &= c_{j_i,k_i-\frac12} + c_{j_i-\frac12,k_i} + c_{j_i+\frac12,k_i} + c_{j_i,k_i+\frac12}, && i = 1,2,\ldots,n\\ a_{i+1,i} &= a_{i,i+1} = -c_{j_i+\frac12,k_i}, && i \bmod m \neq 0\\ a_{i+m,i} &= a_{i,i+m} = -c_{j_i,k_i+\frac12}, && i = 1,2,\ldots,n-m\\ a_{i,j} &= 0 && \text{otherwise,}\end{aligned} \tag{10.53}$$

where (j_i, k_i) with $1 \le j_i, k_i \le m$ is determined uniquely from the equation $i = j_i + (k_i - 1)m$ for $i = 1, \ldots, n$. When $c(x, y) = 1$ for all $(x, y) \in \Omega$ then we recover the Poisson matrix.

In general we cannot write $\boldsymbol{A}$ as a matrix equation of the form (4.15). But we can show that $\boldsymbol{A}$ is symmetric and it is positive definite as long as the function c is positive on Ω.

Theorem 10.36 (Positive definite matrix)
If $c(x, y) > 0$ for $(x, y) \in \Omega$ then the matrix $\boldsymbol{A}$ given by (10.53) is symmetric positive definite.

Proof.
To each $x \in \mathbb{R}^n$ there corresponds a matrix $\boldsymbol{V} \in \mathbb{R}^{m \times m}$ such that $x = \operatorname{vec}(\boldsymbol{V})$. We claim that

$$x^T \boldsymbol{A}\boldsymbol{x} = \sum_{j=1}^{m}\sum_{k=0}^{m} c_{j,k+\frac{1}{2}}\big(v_{j,k+1} - v_{j,k}\big)^2 + \sum_{k=1}^{m}\sum_{j=0}^{m} c_{j+\frac{1}{2},k}\big(v_{j+1,k} - v_{j,k}\big)^2, \quad (10.54)$$

where $v_{0,k} = v_{m+1,k} = v_{j,0} = v_{j,m+1} = 0$ for $j, k = 0, 1, \ldots, m+1$. Since $c_{j+\frac{1}{2},k}$ and $c_{j,k+\frac{1}{2}}$ correspond to values of c in Ω for the values of j, k in the sums it follows that they are positive and from (10.54) we see that $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} \ge 0$ for all $x \in \mathbb{R}^n$. Moreover if $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = 0$ then all quadratic factors are zero and $v_{j,k+1} = v_{j,k}$ for $k = 0, 1, \ldots, m$ and $j = 1, \ldots, m$. Now $v_{j,0} = v_{j,m+1} = 0$ implies that $\boldsymbol{V} = \boldsymbol{0}$ and hence $x = 0$. Thus $\boldsymbol{A}$ is symmetric positive definite.

It remains to prove (10.54). From the connection between (10.52) and (10.53) we have

$$\begin{aligned}
\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} &= \sum_{j=1}^{m}\sum_{k=1}^{m} -(\boldsymbol{P}_h v)_{j,k} v_{j,k} \\
&= \sum_{j=1}^{m}\sum_{k=1}^{m}\Big(c_{j,k-\frac{1}{2}} v_{j,k}^2 + c_{j-\frac{1}{2},k} v_{j,k}^2 + c_{j+\frac{1}{2},k} v_{j,k}^2 + c_{j,k+\frac{1}{2}} v_{j,k}^2 \\
&\qquad - c_{j,k-\frac{1}{2}} v_{j,k-1} v_{j,k} - c_{j,k+\frac{1}{2}} v_{j,k} v_{j,k+1} \\
&\qquad - c_{j-\frac{1}{2},k} v_{j-1,k} v_{j,k} - c_{j+\frac{1}{2},k} v_{j,k} v_{j+1,k}\Big).
\end{aligned}$$

Using the homogenous boundary conditions we have

$$\begin{aligned}
\sum_{j=1}^{m}\sum_{k=1}^{m} c_{j,k-\frac{1}{2}} v_{j,k}^2 &= \sum_{j=1}^{m}\sum_{k=0}^{m} c_{j,k+\frac{1}{2}} v_{j,k+1}^2,\\
\sum_{j=1}^{m}\sum_{k=1}^{m} c_{j,k-\frac{1}{2}} v_{j,k-1} v_{j,k} &= \sum_{j=1}^{m}\sum_{k=0}^{m} c_{j,k+\frac{1}{2}} v_{j,k+1} v_{j,k},\\
\sum_{j=1}^{m}\sum_{k=1}^{m} c_{j-\frac{1}{2},k} v_{j,k}^2 &= \sum_{k=1}^{m}\sum_{j=0}^{m} c_{j+\frac{1}{2},k} v_{j+1,k}^2,\\
\sum_{j=1}^{m}\sum_{k=1}^{m} c_{j-\frac{1}{2},k} v_{j-,k} v_{j,k} &= \sum_{k=1}^{m}\sum_{j=0}^{m} c_{j+\frac{1}{2},k} v_{j+1,k} v_{j,k}.
\end{aligned}$$

It follows that

$$\begin{aligned}
\boldsymbol{x}^T \boldsymbol{A}\boldsymbol{x} = &\sum_{j=1}^{m}\sum_{k=0}^{m} c_{j,k+\frac{1}{2}} \left(v_{j,k}^2 + v_{j,k+1}^2 - 2v_{j,k}v_{j,k+1}\right)\\
&+ \sum_{k=1}^{m}\sum_{j=0}^{m} c_{j+\frac{1}{2},k} \left(v_{j,k}^2 + v_{j+1,k}^2 - 2v_{j,k}v_{j+1,k}\right)
\end{aligned}$$

and (10.54) follows. □ □

10.7.1 Applying Preconditioning

Consider solving $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{A}$ is given by (10.53) and $b \in \mathbb{R}^n$. Since $\boldsymbol{A}$ is positive definite it is nonsingular and the system has a unique solution $x \in \mathbb{R}^n$. Moreover we can use either Cholesky factorization or the block tridiagonal solver to find $\boldsymbol{x}$. Since the bandwidth of $\boldsymbol{A}$ is $m = \sqrt{n}$ both of these methods require $O(n^2)$ arithmetic operations for large n.

If we choose $c(x, y) \equiv 1$ in (10.50), we get the Poisson problem. With this in mind, we may think of the coefficient matrix $\boldsymbol{A}_p$ arising from the discretization of the Poisson problem as an approximation to the matrix (10.53). This suggests using $\boldsymbol{B} = \boldsymbol{A}_p^{-1}$, the inverse of the discrete Poisson matrix as a preconditioner for the system (10.51).

Consider Algorithm 10.34. With this preconditioner the calculation $\boldsymbol{w} = \boldsymbol{B}\boldsymbol{t}$ takes the form $\boldsymbol{A}_p\boldsymbol{w}_k = \boldsymbol{t}_k$.

In Section 5.2 we developed a Simple fast Poisson Solver, Cf. Algorithm 5.1. This method can be utilized to solve $\boldsymbol{A}_p\boldsymbol{w} = \boldsymbol{t}$.

Consider the specific problem where

$$c(x, y) = e^{-x+y} \text{ and } f(x, y) = 1.$$

n	2500	10000	22500	40000	62500
K	222	472	728	986	1246
$K/\sqrt{n}$	4.44	4.72	4.85	4.93	4.98
K_{pre}	22	23	23	23	23

Table 10.37: The number of iterations K (no preconditioning) and K_{pre} (with preconditioning) for the problem (10.50) using the discrete Poisson problem as a preconditioner.

We have used Algorithm 10.17 (conjugate gradient without preconditioning), and Algorithm 10.34 (conjugate gradient with preconditioning) to solve the problem (10.50). We used $\boldsymbol{x}_0 = 0$ and $\epsilon = 10^{-8}$. The results are shown in Table 10.37.

Without preconditioning the number of iterations still seems to be more or less proportional to $\sqrt{n}$ although the convergence is slower than for the constant coefficient problem. Using preconditioning speeds up the convergence considerably. The number of iterations appears to be bounded independently of n. This illustrates that preconditioning is useful when solving nontrivial problems.

Using a preconditioner increases the work in each iteration. For the present example the number of arithmetic operations in each iteration changes from $O(n)$ without preconditioning to $O(n^{3/2})$ or $O(n \log_2 n)$ with preconditioning. This is not a large increase and both the number of iterations and the computing time is reduced significantly.

Let us finally show that the number $\kappa = \lambda_{max}/\lambda_{min}$ which determines the rate of convergence for the preconditioned conjugate gradient method applied to (10.50) can be bounded independently of n.

Theorem 10.38 (Eigevalues of preconditioned matrix)
Suppose $0 < c_0 \le c(x,y) \le c_1$ *for all* $(x,y) \in [0,1]^2$. *For the eigenvalues of the matrix* $\boldsymbol{BA} = \boldsymbol{A}_p^{-1}\boldsymbol{A}$ *just described we have*

$$\kappa = \frac{\lambda_{max}}{\lambda_{min}} \le \frac{c_1}{c_0}.$$

Proof.

Suppose $\boldsymbol{A}_p^{-1}\boldsymbol{Ax} = \lambda x$ for some $\boldsymbol{x} \in \mathbb{R}^n \setminus \{0\}$. Then $\boldsymbol{Ax} = \lambda \boldsymbol{A}_p x$. Multiplying this by $\boldsymbol{x}^T$ and solving for λ we find

$$\lambda = \frac{x^T \boldsymbol{Ax}}{x^T \boldsymbol{A}_p x}.$$

We computed $\boldsymbol{x}^T\boldsymbol{Ax}$ in (10.54) and we obtain $\boldsymbol{x}^T\boldsymbol{A}_p\boldsymbol{x}$ by setting all the c's there equal to one

$$\boldsymbol{x}^T\boldsymbol{A}_p x = \sum_{i=1}^{m}\sum_{j=0}^{m}\bigl(v_{i,j+1} - v_{i,j}\bigr)^2 + \sum_{j=1}^{m}\sum_{i=0}^{m}\bigl(v_{i+1,j} - v_{i,j}\bigr)^2.$$

Thus $\boldsymbol{x}^T\boldsymbol{A}_p x > 0$ and bounding all the c's in (10.54) from below by c_0 and above by c_1 we find

$$c_0(x^T\boldsymbol{A}_p x) \le x^T\boldsymbol{A}\boldsymbol{x} \le c_1(x^T\boldsymbol{A}_p x)$$

which implies that $c_0 \le \lambda \le c_1$ for all eigenvalues λ of $\boldsymbol{B}\boldsymbol{A} = \boldsymbol{A}_p^{-1}\boldsymbol{A}$. □

Using $c(x, y) = e^{-x+y}$ as above, we find $c_0 = e^{-2}$ and $c_1 = 1$. Thus $\kappa \le e^2 \approx 7.4$, a quite acceptable matrix condition which explains the convergence results from our numerical experiment.

10.8 Review Questions

10.8.1 Does the steepest descent and conjugate gradient method always converge?

10.8.2 What kind of orthogonalities occur in the conjugate gradient method?

10.8.3 What is a Krylow space?

10.8.4 What is a convex function?

10.8.5 How do SOR and conjugate gradient compare?