Conjugate Gradient method

Consider the problem:
$0 < x < 1$

$$\frac{d^2U}{dx^2} = f$$

$$U(0) = U(1) = 0,$$

Find the exact solution for $f = -(k\pi)^2 \sin(k\pi x)$.
For $k = 1, 2, 3, 4, 5, 6$, use the Conjugate Gradient method with $N = 65$ and plot the norm of error as a function of the number of iterations for $n_{max} = 150$.
For $n = 1, 2, 5, 10, 20$, plot the error at grid points $i = 1$ to $i = 64$ for $k = 1, 2, 3, 4, 5, 6$.
Explain your results.

Basic conjugate gradient algorithm:
https://en.wikipedia.org/wiki/Conjugate_gradient_method#The_resulting_algorithm

Minimization Condition

The vector $\mathbf{x}^*$ is a solution to the positive definite linear system $A\mathbf{x} = \mathbf{b}$ if and only if $\mathbf{x}^*$ minimizes

$$g(\mathbf{x}) = \langle \mathbf{x}, A\mathbf{x}\rangle - 2\langle \mathbf{x}, \mathbf{b}\rangle.$$

In addition, for any $\mathbf{x}$ and $\mathbf{v} \neq \mathbf{0}$ the function $g(\mathbf{x} + t\mathbf{v})$ has its minimum when $t = \langle \mathbf{v}, \mathbf{b} - A\mathbf{x}\rangle / \langle \mathbf{v}, A\mathbf{v}\rangle$.

To begin the conjugate gradient method, we choose $\mathbf{x}$, an approximate solution to $A\mathbf{x}^* = \mathbf{b}$, and $\mathbf{v} \neq \mathbf{0}$, which gives a *search direction* in which to move away from $\mathbf{x}$ to improve the approximation. Let $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ be the residual vector associated with $\mathbf{x}$ and

$$t = \frac{\langle \mathbf{v}, \mathbf{b} - A\mathbf{x}\rangle}{\langle \mathbf{v}, A\mathbf{v}\rangle} = \frac{\langle \mathbf{v}, \mathbf{r}\rangle}{\langle \mathbf{v}, A\mathbf{v}\rangle}.$$

If $\mathbf{r} \neq \mathbf{0}$ and if $\mathbf{v}$ and $\mathbf{r}$ are not orthogonal, then $\mathbf{x} + t\mathbf{v}$ gives a smaller value for g than $g(\mathbf{x})$ and is presumably closer to $\mathbf{x}^*$ than is $\mathbf{x}$. This suggests the following method.

Let $\mathbf{x}^{(0)}$ be an initial approximation to $\mathbf{x}^*$, and let $\mathbf{v}^{(1)} \neq \mathbf{0}$ be an initial search direction. For $k = 1, 2, 3, \ldots$, we compute

$$t_k = \frac{\langle \mathbf{v}^{(k)}, \mathbf{b} - A\mathbf{x}^{(k-1)}\rangle}{\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(k)}\rangle},$$

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + t_k\mathbf{v}^{(k)}$$

and choose a new search direction $\mathbf{v}^{(k+1)}$. The object is to make this selection so that the sequence of approximations $\{\mathbf{x}^{(k)}\}$ converges rapidly to $\mathbf{x}^*$.

To choose the search directions, we view g as a function of the components of $\mathbf{x} = (x_1, x_2, \ldots, x_n)^t$. Thus,

$$g(x_1, x_2, \ldots, x_n) = \langle \mathbf{x}, A\mathbf{x}\rangle - 2\langle \mathbf{x}, \mathbf{b}\rangle = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}x_ix_j - 2\sum_{i=1}^{n} x_ib_i.$$

Taking partial derivatives with respect to the component variables x_k gives

$$\frac{\partial g}{\partial x_k}(\mathbf{x}) = 2\sum_{i=1}^{n} a_{ki}x_i - 2b_k.$$

Therefore, the gradient of g is

$$\nabla g(\mathbf{x}) = \left(\frac{\partial g}{\partial x_1}(\mathbf{x}), \frac{\partial g}{\partial x_2}(\mathbf{x}), \ldots, \frac{\partial g}{\partial x_n}(\mathbf{x})\right)^t = 2(A\mathbf{x} - \mathbf{b}) = -2\mathbf{r},$$

where the vector $\mathbf{r}$ is the residual vector for $\mathbf{x}$.

From multivariable calculus, we know that the direction of greatest decrease in the value of $g(\mathbf{x})$ is the direction given by $-\nabla g(\mathbf{x})$; that is, in the direction of the residual $\mathbf{r}$.

The method that chooses

$$\mathbf{v}^{(k+1)} = \mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$$

is called the *method of steepest descent*. Although we will see in Section 10.4 that this method has merit for nonlinear systems and optimization problems, it is not used for linear systems because of slow convergence.

An alternative approach uses a set of nonzero direction vectors $\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}\}$ that satisfy

$$\langle \mathbf{v}^{(i)}, A\mathbf{v}^{(j)}\rangle = 0, \quad \text{if } i \neq j.$$

This is called an *A*-**orthogonality condition**, and the set of vectors $\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}\}$ is said to be *A*-**orthogonal**. It is not difficult to show that a set of *A*-orthogonal vectors associated with the positive definite matrix A is linearly independent. [See Exercise 13(a).] This set of search directions gives

$$t_k = \frac{\langle \mathbf{v}^{(k)}, \mathbf{b} - A\mathbf{x}^{(k-1)}\rangle}{\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(k)}\rangle} = \frac{\langle \mathbf{v}^{(k)}, \mathbf{r}^{(k-1)}\rangle}{\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(k)}\rangle}$$

and $\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + t_k\mathbf{v}^{(k)}$.

The following result shows that this choice of search directions gives convergence in at most n steps, so as a direct method it produces the exact solution, assuming that the arithmetic is exact.

A-Orthogonality Convergence

Let $\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(n)}\}$ be an A-orthogonal set of nonzero vectors associated with the positive definite matrix A, and let $\mathbf{x}^{(0)}$ be arbitrary. Define

$$t_k = \frac{\langle \mathbf{v}^{(k)}, \mathbf{b} - A\mathbf{x}^{(k-1)}\rangle}{\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(k)}\rangle} \quad \text{and} \quad \mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + t_k\mathbf{v}^{(k)},$$

for $k = 1, 2, \ldots, n$. Then, assuming exact arithmetic, $A\mathbf{x}^{(n)} = \mathbf{b}$.

Example 1 Consider the positive definite matrix

$$A = \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}.$$

Let $\mathbf{v}^{(1)} = (1, 0, 0)^t$, $\mathbf{v}^{(2)} = (-\frac{3}{4}, 1, 0)^t$, and $\mathbf{v}^{(3)} = (-\frac{3}{7}, \frac{4}{7}, 1)^t$. By direct calculation,

$$\langle \mathbf{v}^{(1)}, A\mathbf{v}^{(2)}\rangle = \mathbf{v}^{(1)t}A\mathbf{v}^{(2)} = (1, 0, 0)\begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}\begin{bmatrix} -\frac{3}{4} \\ 1 \\ 0 \end{bmatrix} = 0,$$

$$\langle \mathbf{v}^{(1)}, A\mathbf{v}^{(3)}\rangle = (1, 0, 0)\begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}\begin{bmatrix} -\frac{3}{7} \\ \frac{4}{7} \\ 1 \end{bmatrix} = 0,$$

and

$$\langle \mathbf{v}^{(2)}, A\mathbf{v}^{(3)}\rangle = \left(-\frac{3}{4}, 1, 0\right)\begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}\begin{bmatrix} -\frac{3}{7} \\ \frac{4}{7} \\ 1 \end{bmatrix} = 0.$$

Thus, $\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \mathbf{v}^{(3)}\}$ is an A-orthogonal set.

The linear system

$$\begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 24 \\ 30 \\ -24 \end{bmatrix},$$

has the exact solution $\mathbf{x}^* = (3, 4, -5)^t$. To approximate this solution, let $\mathbf{x}^{(0)} = (0, 0, 0)^t$. Since $\mathbf{b} = (24, 30, -24)^t$, we have

$$\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)} = \mathbf{b} = (24, 30, -24)^t,$$

so

$$\langle \mathbf{v}^{(1)}, \mathbf{r}^{(0)}\rangle = \mathbf{v}^{(1)t}\mathbf{r}^{(0)} = 24, \quad \langle \mathbf{v}^{(1)}, A\mathbf{v}^{(1)}\rangle = 4, \quad \text{and} \quad t_0 = \frac{24}{4} = 6.$$

Thus,

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + t_0\mathbf{v}^{(1)} = (0, 0, 0)^t + 6(1, 0, 0)^t = (6, 0, 0)^t.$$

Continuing, we have

$$\mathbf{r}^{(1)} = \mathbf{b} - A\mathbf{x}^{(1)} = (0, 12, -24)^t; \quad t_1 = \frac{\langle \mathbf{v}^{(2)}, \mathbf{r}^{(1)}\rangle}{\langle \mathbf{v}^{(2)}, A\mathbf{v}^{(2)}\rangle} = \frac{12}{7/4} = \frac{48}{7};$$

$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + t_1\mathbf{v}^{(2)} = (6, 0, 0)^t + \frac{48}{7}\left(-\frac{3}{4}, 1, 0\right)^t = \left(\frac{6}{7}, \frac{48}{7}, 0\right)^t;$$

$$\mathbf{r}^{(2)} = \mathbf{b} - A\mathbf{x}^{(2)} = \left(0, 0, -\frac{120}{7}\right); \quad t_2 = \frac{\langle \mathbf{v}^{(3)}, \mathbf{r}^{(2)}\rangle}{\langle \mathbf{v}^{(3)}, A\mathbf{v}^{(3)}\rangle} = \frac{-120/7}{24/7} = -5;$$

and

$$\mathbf{x}^{(3)} = \mathbf{x}^{(2)} + t_2\mathbf{v}^{(3)} = \left(\frac{6}{7}, \frac{48}{7}, 0\right)^t + (-5)\left(-\frac{3}{7}, \frac{4}{7}, 1\right)^t = (3, 4, -5)^t.$$

Since we applied the technique $n = 3$ times, this is the actual solution. ■

Before discussing how to determine the A-orthogonal set, we will continue the development. The use of an A-orthogonal set $\{\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(n)}\}$ of direction vectors gives what is

called a *conjugate direction* method. The following result concerns the orthogonality of the residual vectors $\mathbf{r}^{(k)}$ and the direction vectors $\mathbf{v}^{(j)}$.

Orthogonal Residual Vectors

The residual vectors $\mathbf{r}^{(k)}$, where $k = 1, 2, \dots, n$, for a conjugate direction method, satisfy the equations

$$\langle \mathbf{r}^{(k)}, \mathbf{v}^{(j)}\rangle = 0, \quad \text{for each } j = 1, 2, \dots, k.$$

The conjugate gradient method of Hestenes and Stiefel chooses the search directions $\{\mathbf{v}^{(k)}\}$ during the iterative process so that the residual vectors $\{\mathbf{r}^{(k)}\}$ are mutually orthogonal. To construct the direction vectors $\{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, \dots\}$ and the approximations $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots\}$, we start with an initial approximation $\mathbf{x}^{(0)}$ and use the steepest descent direction $\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}$ as the first search direction $\mathbf{v}^{(1)}$.

Assume that the conjugate directions $\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(k-1)}$ and the approximations $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(k-1)}$ have been computed with

$$\mathbf{x}^{(k-1)} = \mathbf{x}^{(k-2)} + t_{k-1}\mathbf{v}^{(k-1)},$$

where

$$\langle \mathbf{v}^{(i)}, A\mathbf{v}^{(j)}\rangle = 0 \quad \text{and} \quad \langle \mathbf{r}^{(i)}, \mathbf{r}^{(j)}\rangle = 0, \quad \text{for } i \neq j.$$

If $\mathbf{x}^{(k-1)}$ is the solution to $A\mathbf{x} = \mathbf{b}$, we are done. Otherwise, $\mathbf{r}^{(k-1)} = \mathbf{b} - A\mathbf{x}^{(k-1)} \neq \mathbf{0}$ and the orthogonality implies that $\langle \mathbf{r}^{(k-1)}, \mathbf{v}^{(i)}\rangle = 0$, for $i = 1, 2, \dots, k-1$. We then use $\mathbf{r}^{(k-1)}$ to generate $\mathbf{v}^{(k)}$ by setting

$$\mathbf{v}^{(k)} = \mathbf{r}^{(k-1)} + s_{k-1}\mathbf{v}^{(k-1)}.$$

We want to choose s_{k-1} so that

$$\langle \mathbf{v}^{(k-1)}, A\mathbf{v}^{(k)}\rangle = 0.$$

Since

$$A\mathbf{v}^{(k)} = A\mathbf{r}^{(k-1)} + s_{k-1}A\mathbf{v}^{(k-1)}$$

and

$$\langle \mathbf{v}^{(k-1)}, A\mathbf{v}^{(k)}\rangle = \langle \mathbf{v}^{(k-1)}, A\mathbf{r}^{(k-1)}\rangle + s_{k-1}\langle \mathbf{v}^{(k-1)}, A\mathbf{v}^{(k-1)}\rangle,$$

we will have $\langle \mathbf{v}^{(k-1)}, A\mathbf{v}^{(k)}\rangle = 0$ when

$$s_{k-1} = -\frac{\langle \mathbf{v}^{(k-1)}, A\mathbf{r}^{(k-1)}\rangle}{\langle \mathbf{v}^{(k-1)}, A\mathbf{v}^{(k-1)}\rangle}.$$

It can also be shown that with this choice of s_{k-1} we have $\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(i)}\rangle = 0$, for each $i = 1, 2, \dots, k-2$. Thus, $\{\mathbf{v}^{(1)}, \dots \mathbf{v}^{(k)}\}$ is an A-orthogonal set.

Having chosen $\mathbf{v}^{(k)}$, we compute

$$t_k = \frac{\langle \mathbf{v}^{(k)}, \mathbf{r}^{(k-1)} \rangle}{\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(k)} \rangle} = \frac{\langle \mathbf{r}^{(k-1)} + s_{k-1}\mathbf{v}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle}{\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(k)} \rangle}$$

$$= \frac{\langle \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle}{\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(k)} \rangle} + s_{k-1}\frac{\langle \mathbf{v}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle}{\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(k)} \rangle}.$$

By the orthogonality result, $\langle \mathbf{v}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle = 0$, so

$$t_k = \frac{\langle \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle}{\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(k)} \rangle}. \tag{7.5}$$

Thus,

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + t_k\mathbf{v}^{(k)}.$$

To compute $\mathbf{r}^{(k)}$, we multiply by A and subtract $\mathbf{b}$ to obtain

$$A\mathbf{x}^{(k)} - \mathbf{b} = A\mathbf{x}^{(k-1)} - \mathbf{b} + t_k A\mathbf{v}^{(k)}$$

or

$$\mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} - t_k A\mathbf{v}^{(k)}.$$

Thus,

$$\langle \mathbf{r}^{(k)}, \mathbf{r}^{(k)} \rangle = \langle \mathbf{r}^{(k-1)}, \mathbf{r}^{(k)} \rangle - t_k\langle A\mathbf{v}^{(k)}, \mathbf{r}^{(k)} \rangle = -t_k\langle \mathbf{r}^{(k)}, A\mathbf{v}^{(k)} \rangle.$$

Further, from Eq. (7.5),

$$\langle \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle = t_k\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(k)} \rangle,$$

so

$$s_k = -\frac{\langle \mathbf{v}^{(k)}, A\mathbf{r}^{(k)} \rangle}{\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(k)} \rangle} = -\frac{\langle \mathbf{r}^{(k)}, A\mathbf{v}^{(k)} \rangle}{\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(k)} \rangle}$$

$$= \frac{(1/t_k)\langle \mathbf{r}^{(k)}, \mathbf{r}^{(k)} \rangle}{(1/t_k)\langle \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle} = \frac{\langle \mathbf{r}^{(k)}, \mathbf{r}^{(k)} \rangle}{\langle \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle}.$$

In summary, we have the formulas:

$$\mathbf{r}^{(0)} = \mathbf{b} - A\mathbf{x}^{(0)}; \quad \mathbf{v}^{(1)} = \mathbf{r}^{(0)};$$

and, for $k = 1, 2, \ldots, n$,

$$t_k = \frac{\langle \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle}{\langle \mathbf{v}^{(k)}, A\mathbf{v}^{(k)} \rangle},$$

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + t_k\mathbf{v}^{(k)},$$

$$\mathbf{r}^{(k)} = \mathbf{r}^{(k-1)} - t_k A\mathbf{v}^{(k)},$$

$$s_k = \frac{\langle \mathbf{r}^{(k)}, \mathbf{r}^{(k)} \rangle}{\langle \mathbf{r}^{(k-1)}, \mathbf{r}^{(k-1)} \rangle},$$

$$\mathbf{v}^{(k+1)} = \mathbf{r}^{(k)} + s_k\mathbf{v}^{(k)}. \tag{7.6}$$

We will now extend the conjugate gradient method to include *preconditioning*. If the matrix A is ill-conditioned, the conjugate gradient method is highly susceptible to rounding errors. So, although the exact answer should be obtained in n steps, this is not usually the case. As a direct method the conjugate gradient method is not as good as Gaussian elimination with pivoting. The main use of the conjugate gradient method is as an iterative method applied to a better-conditioned system. In this case an acceptable approximate solution is often obtained in about $\sqrt{n}$ steps.

To apply the method to a better-conditioned system, we want to select a nonsingular conditioning matrix C so that

$$\tilde{A} = C^{-1}A(C^{-1})^t$$

is better conditioned. To simplify the notation, we will use the matrix C^{-t} to refer to $(C^{-1})^t$.

Consider the linear system

$$\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}},$$

where $\tilde{\mathbf{x}} = C^t\mathbf{x}$ and $\tilde{\mathbf{b}} = C^{-1}\mathbf{b}$. Then

$$\tilde{A}\tilde{\mathbf{x}} = (C^{-1}AC^{-t})(C^t\mathbf{x}) = C^{-1}A\mathbf{x}.$$

Thus, we could solve $\tilde{A}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$ for $\tilde{\mathbf{x}}$ and then obtain $\mathbf{x}$ by multiplying by C^{-t}. However, instead of rewriting Eqs. (7.6) using $\tilde{\mathbf{r}}^{(k)}$, $\tilde{\mathbf{v}}^{(k)}$, $\tilde{t}_k$, $\tilde{\mathbf{x}}^{(k)}$, and $\tilde{s}_k$, we incorporate the preconditioning implicitly.

Since

$$\tilde{\mathbf{x}}^{(k)} = C^t\mathbf{x}^{(k)},$$

we have

$$\tilde{\mathbf{r}}^{(k)} = \tilde{\mathbf{b}} - \tilde{A}\tilde{\mathbf{x}}^{(k)} = C^{-1}\mathbf{b} - (C^{-1}AC^{-t})C^t\mathbf{x}^{(k)} = C^{-1}(\mathbf{b} - A\mathbf{x}^{(k)}) = C^{-1}\mathbf{r}^{(k)}.$$

Let $\tilde{\mathbf{v}}^{(k)} = C^t\mathbf{v}^{(k)}$ and $\mathbf{w}^{(k)} = C^{-1}\mathbf{r}^{(k)}$. Then

$$\tilde{s}_k = \frac{\langle \tilde{\mathbf{r}}^{(k)}, \tilde{\mathbf{r}}^{(k)} \rangle}{\langle \tilde{\mathbf{r}}^{(k-1)}, \tilde{\mathbf{r}}^{(k-1)} \rangle} = \frac{\langle C^{-1}\mathbf{r}^{(k)}, C^{-1}\mathbf{r}^{(k)} \rangle}{\langle C^{-1}\mathbf{r}^{(k-1)}, C^{-1}\mathbf{r}^{(k-1)} \rangle},$$

so

$$\tilde{s}_k = \frac{\langle \mathbf{w}^{(k)}, \mathbf{w}^{(k)} \rangle}{\langle \mathbf{w}^{(k-1)}, \mathbf{w}^{(k-1)} \rangle}. \tag{7.7}$$

Finally, the third iteration gives

$$\mathbf{u} = A\mathbf{v}^{(3)} = (0.1014898976, -0.1040922099, -0.0286253554)^t;$$

$$t_3 = 1.192628008;$$

$$\mathbf{x}^{(3)} = (2.999999998, 4.000000002, -4.999999998)^t;$$

$$\mathbf{r}^{(3)} = (0.36 \times 10^{-8}, 0.39 \times 10^{-8}, -0.141 \times 10^{-8})^t.$$

Since $\mathbf{x}^{(3)}$ is nearly the exact solution, rounding error did not significantly affect the result. In Example 1 of Section 7.5, the Gauss-Seidel method required 34 iterations, and the SOR method, with $\omega = 1.25$, required 14 iterations for an accuracy of 10^{-7}. It should be noted, however, that in this example, we are really comparing a direct method to iterative methods. ■

The next example illustrates the effect of preconditioning on a poorly conditioned matrix. In this example and subsequently, we use $D^{-1/2}$ to represent the diagonal matrix whose entries are the reciprocals of the square roots of the diagonal entries of the coefficient matrix A.

Example 3 The linear system $A\mathbf{x} = \mathbf{b}$ with

$$A = \begin{bmatrix} 0.2 & 0.1 & 1 & 1 & 0 \\ 0.1 & 4 & -1 & 1 & -1 \\ 1 & -1 & 60 & 0 & -2 \\ 1 & 1 & 0 & 8 & 4 \\ 0 & -1 & -2 & 4 & 700 \end{bmatrix} \quad \text{and} \quad \mathbf{b} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

has the solution

$$\mathbf{x}^* = (7.859713071, 0.4229264082, -0.07359223906, -0.5406430164, 0.01062616286)^t.$$

The matrix A is symmetric and positive definite but is ill-conditioned with condition number $K_\infty(A) = 13961.71$. We will use tolerance 0.01 and compare the results obtained from the Jacobi, Gauss-Seidel, and SOR (with $\omega = 1.25$) iterative methods and from the conjugate gradient method with $C^{-1} = I$. Then we precondition by choosing C^{-1} as $D^{-1/2}$, the diagonal matrix whose diagonal entries are the reciprocal of the positive square roots of the diagonal entries of the positive definite matrix A. The results are presented in Table 7.5. The preconditioned conjugate gradient method gives the most accurate approximation with the smallest number of iterations. ■

The preconditioned conjugate gradient method is often used in the solution of large linear systems in which the matrix is sparse and positive definite. These systems must be solved to approximate solutions to boundary-value problems in ordinary-differential equations (Sections 11.3, 11.5, 11.6). The larger the system, the more impressive the conjugate gradient method becomes since it significantly reduces the number of iterations required. In these systems, the preconditioning matrix C is approximately equal to L in the Choleski factorization LL^t of A. Generally, small entries in A are ignored and Choleski's method is applied to obtain what is called an incomplete LL^t factorization of A. Thus, $C^{-t}C^{-1} \approx A^{-1}$ and a good approximation is obtained. More information about the conjugate gradient method can be found in Kelley [Kelley].

Table 7.5

Method	Number of Iterations	$\mathbf{x}^{(k)}$	$\lVert \mathbf{x}^* - \mathbf{x}^{(k)} \rVert_\infty$
Jacobi	49	(7.86277141, 0.42320802, −0.07348669, −0.53975964, 0.01062847)^t	0.00305834
Gauss-Seidel	15	(7.83525748, 0.42257868, −0.07319124, −0.53753055, 0.01060903)^t	0.02445559
SOR($\omega = 1.25$)	7	(7.85152706, 0.42277371, −0.07348303, −0.53978369, 0.01062286)^t	0.00818607
Conjugate Gradient	5	(7.85341523, 0.42298677, −0.07347963, −0.53987920, 0.008628916)^t	0.00629785
Conjugate Gradient (Preconditioned)	4	(7.85968827, 0.42288329, −0.07359878, −0.54063200, 0.01064344)^t	0.00009312

Exercise Set 7.7

1. The linear system

$$x_1 + \tfrac{1}{2}x_2 = \tfrac{5}{21},$$
$$\tfrac{1}{2}x_1 + \tfrac{1}{3}x_2 = \tfrac{11}{84}$$

has solution $(x_1, x_2)^t = (\frac{1}{6}, \frac{1}{7})^t$.

   a. Solve the linear system using Gaussian elimination with two-digit rounding arithmetic.

   b. Solve the linear system using the conjugate gradient method ($C = C^{-1} = I$) with two-digit rounding arithmetic.

   c. Which method gives the better answer?

   d. Choose $C^{-1} = D^{-1/2}$. Does this choice improve the conjugate gradient method?

2. The linear system

$$0.1x_1 + 0.2x_2 = 0.3,$$
$$0.2x_1 + 113x_2 = 113.2$$

has solution $(x_1, x_2)^t = (1, 1)^t$. Repeat the directions for Exercise 1 on this linear system.