Conjugate Gradient (CG) Method

by K. Ozawa

1 Introduction

In the series of this lecture, I will introduce the *conjugate gradient method*, which solves efficiently large scale sparse linear simultaneous equations.

2 Minimization problem

Consider the quadratic function given by

$$Q(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T A\,\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}, \qquad A \in \mathbb{R}^{n\times n}, \quad \boldsymbol{b}, \boldsymbol{x} \in \mathbb{R}^n, \tag{1}$$

where A is a symmetric positive definite matrix. This quadratic function attains its minimum at the solution of the linear algebraic equation $A\,\boldsymbol{x} = \boldsymbol{b}$, which will be denoted by $\boldsymbol{x}^*$. A variety of methods for solving this minimization problem have been derived. The general form of these algorithms is

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k\,\boldsymbol{p}_k, \qquad k = 0, 1, \ldots, \tag{2}$$

where $\boldsymbol{p}_k$ is the *direction* vector at the kth iteration. The parameter α_k is determined so as to minimize the objective function $Q(\boldsymbol{x})$, that is, α_k is the value satisfying

$$Q(\boldsymbol{x}_k - \alpha_k\,\boldsymbol{p}_k) = \min_{\alpha} Q(\boldsymbol{x}_k - \alpha\,\boldsymbol{p}_k). \tag{3}$$

Since in the expression

$$Q(\boldsymbol{x}_k - \alpha_k\,\boldsymbol{p}_k) = \frac{1}{2}(\boldsymbol{p}_k^T A\,\boldsymbol{p}_k)\,\alpha_k^2 - \boldsymbol{p}_k^T(A\,\boldsymbol{x}_k - \boldsymbol{b})\,\alpha_k + \frac{1}{2}\boldsymbol{x}_k^T(A\boldsymbol{x}_k - 2\,\boldsymbol{b}), \tag{4}$$

the coefficients of α_k^2 is positive, this function attains its minimum at

$$\alpha_k = \boldsymbol{p}_k^T(A\,\boldsymbol{x}_k - \boldsymbol{b})/(\boldsymbol{p}_k^T A\,\boldsymbol{p}_k), \tag{5}$$

for the fixed $\boldsymbol{x}_k$ and $\boldsymbol{p}_k$.

Next we consider the method for choosing the direction vectors $\boldsymbol{p}_k$.

3 Univariate iteration

If we put $\boldsymbol{p}_k = \boldsymbol{e}_i$ (the unit vector whose ith entry is and the others are 0) in (2), then with the α_k given by (5) we have

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k\,\boldsymbol{e}_i = \boldsymbol{x}_k - \frac{1}{a_{ii}}\left(\sum_{j=1}^{n} a_{ij}\,x_j^k - b_i\right)\boldsymbol{e}_i, \tag{6}$$

since

$$\boldsymbol{e}_i^T A\,\boldsymbol{e}_i = a_{ii} \qquad \text{and} \qquad \boldsymbol{e}_i^T (A\,\boldsymbol{x} - \boldsymbol{b}) = \sum_{j=1}^{n} a_{ij}\,x_j - b_i.$$

This means that the point $\boldsymbol{x}_{k+1}$ given by (6) is the minimum point which can be obtained by changing only the ith component of $\boldsymbol{x}_k$. Thus, the first n successive iterations of (6) with

$$\boldsymbol{p}_0 = \boldsymbol{e}_1, \quad \boldsymbol{p}_1 = \boldsymbol{e}_2, \ \ldots, \ \boldsymbol{p}_{n-1} = \boldsymbol{e}_n \tag{7}$$

is equivalent to the one iteration of the well-known Gauss–Seidel method.

4 Steepest descent method

On the other hand, if we choose $\boldsymbol{p}_k$ as the gradient vector of $Q(\boldsymbol{x})$ at the point $\boldsymbol{x}_k$, then we have the iteration method given by

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k\,(A\,\boldsymbol{x}_k - \boldsymbol{b}), \tag{8}$$

since

$$\boldsymbol{p}_k = \nabla Q(\boldsymbol{x}_k) = A\,\boldsymbol{x}_k - \boldsymbol{b}, \tag{9}$$

where

$$\nabla Q(\boldsymbol{x}) = \left(\frac{\partial Q}{\partial x_1}, \ldots, \frac{\partial Q}{\partial x_n} \right)^T.$$

This method is called the *steepest descent method*.

5 Conjugate direction method

Let us consider the case that $\boldsymbol{p}_k$ are chosen as

$$\boldsymbol{p}_i^T A\,\boldsymbol{p}_j = 0, \quad i \neq j. \tag{10}$$

When the direction vectors $\boldsymbol{p}_k$ are taken in this way, the method is called the *conjugate direction method*. The vectors $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ satisfying (10) are said to be A-*conjugate*. The conjugate direction method converges in n steps, if no roundoff occurs. Here we show the convergence of the method.

First of all, notice that

$$\begin{aligned}(A\,\boldsymbol{x}_{k+1} - \boldsymbol{b})^T \boldsymbol{p}_j &= (A\,\boldsymbol{x}_k - \alpha_k\,A\,\boldsymbol{p}_k - \boldsymbol{b})^T \boldsymbol{p}_j \\ &= (A\,\boldsymbol{x}_k - \boldsymbol{b})^T \boldsymbol{p}_j - \alpha_k\,(A\,\boldsymbol{p}_k)^T \boldsymbol{p}_j \\ &= (A\,\boldsymbol{x}_k - \boldsymbol{b})^T \boldsymbol{p}_j,\end{aligned}$$

and as a result, we have

$$(A\,\boldsymbol{x}_{k+1} - \boldsymbol{b})^T \boldsymbol{p}_j = \begin{cases} (A\,\boldsymbol{x}_k - \boldsymbol{b})^T \boldsymbol{p}_j, & j < k, \\ 0, & j = k. \end{cases} \tag{11}$$

Thus we have

$$(A\,\boldsymbol{x}_n - \boldsymbol{b})^T \boldsymbol{p}_j = (A\,\boldsymbol{x}_{n-1} - \boldsymbol{b})^T \boldsymbol{p}_j = \cdots = (A\,\boldsymbol{x}_{j+1} - \boldsymbol{b})^T \boldsymbol{p}_j = 0, \quad j = 0, \ldots, n-1.$$

Therefore, the vector $A\,\boldsymbol{x}_n - \boldsymbol{b}$ is orthogonal to the n linearly independent vectors $\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{n-1}$, that is, $A\,\boldsymbol{x}_n - \boldsymbol{b} = 0$.

Here we consider the method of generating the conjugate vectors $\boldsymbol{p}_j$. One might think of the method of selecting eigenvectors of A as $\boldsymbol{p}_j\,(j = 1, \ldots, n)$, since the two distinct eigenvectors of A, say $\boldsymbol{p}_i$, $\boldsymbol{p}_j$, satisfy

$$\boldsymbol{p}_i^T\, A\,\boldsymbol{p}_j = \lambda_j\, \boldsymbol{p}_i^T\, \boldsymbol{p}_j = 0.$$

However, in general, the computation of all eigenvectors is far more expensive than solving linear equations, so that this method is not practical at all. The *conjugate gradient method* to be explained in the next section generates the conjugate direction vectors by using a relatively cheap procedure.

6 Conjugate gradient method

Here we show the algorithm:

CG-1

Choose a small value $\varepsilon > 0$ and an initial guess $\boldsymbol{x}_0$
$\boldsymbol{p}_0 = \boldsymbol{r}_0 = \boldsymbol{b} - A\,\boldsymbol{x}_0$, and compute $(\boldsymbol{r}_0, \boldsymbol{r}_0)$
$k = 0$
while $\|\boldsymbol{r}_k\|/\|\boldsymbol{b}\| \geq \varepsilon$ **do**
    $\alpha_k = -(\boldsymbol{r}_k, \boldsymbol{p}_k)/(\boldsymbol{p}_k, A\,\boldsymbol{p}_k)$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k\,\boldsymbol{p}_k$
    $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k + \alpha_k\,A\,\boldsymbol{p}_k$
    $\beta_k = -(\boldsymbol{p}_k, A\,\boldsymbol{r}_{k+1})/(\boldsymbol{p}_k, A\,\boldsymbol{p}_k)$
    $\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k\,\boldsymbol{p}_k$
    $k = k + 1$
end while

The vectors $\boldsymbol{p}_k$ generated by the algorithm satisfy the conjugacy condition

$$(\boldsymbol{p}_k, A\,\boldsymbol{p}_j) = 0, \quad 0 \leq j < k, \quad k = 1, \ldots, n-1, \tag{12}$$

and the residuals $\boldsymbol{r}_j (= \boldsymbol{b} - A\,\boldsymbol{x}_j)$ satisfy the orthogonality condition (see Appendix)

$$(\boldsymbol{r}_k, \boldsymbol{r}_j) = 0, \quad 0 \leq j < k, \quad k = 1, \ldots, n-1. \tag{13}$$

Moreover, from the definitions of α_j and β_j we have

$$(\boldsymbol{p}_j, \boldsymbol{r}_{j+1}) = (\boldsymbol{p}_j, \boldsymbol{r}_j) + \alpha_j(\boldsymbol{p}_j, A\,\boldsymbol{p}_j) = 0, \quad j = 0, 1, \ldots \tag{14}$$

$$(\boldsymbol{p}_j, A\,\boldsymbol{p}_{j+1}) = (\boldsymbol{p}_j, A\,\boldsymbol{r}_{j+1}) + \beta_j\,(\boldsymbol{p}_j, A\,\boldsymbol{p}_j) = 0, \quad j = 0, 1, \ldots \tag{15}$$

Using the relations above, we have a variant of CG-1:

CG-2

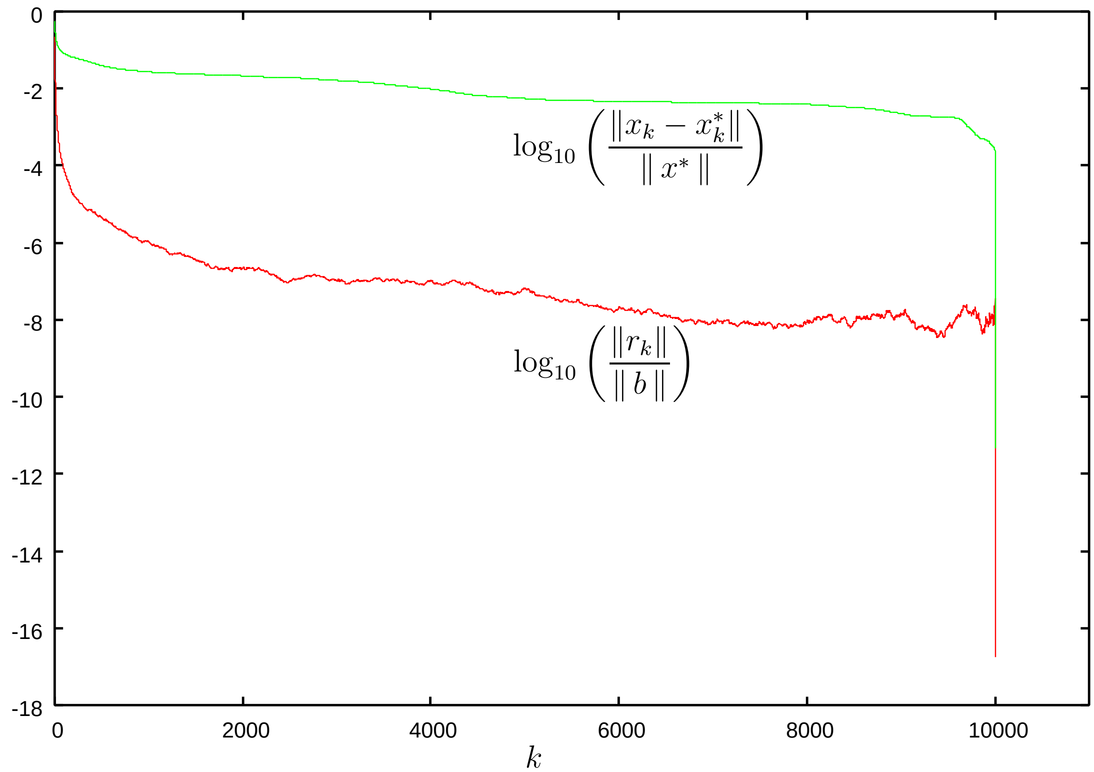
Choose a small value $\varepsilon > 0$ and an initial guess $\boldsymbol{x}_0$
$\boldsymbol{p}_0 = \boldsymbol{r}_0 = \boldsymbol{b} - A\,\boldsymbol{x}_0$, and compute $(\boldsymbol{r}_0, \boldsymbol{r}_0)$
$k = 0$
while $\|\boldsymbol{r}_k\| / \|\boldsymbol{b}\| \geq \varepsilon$ **do**
    $\alpha_k = -(\boldsymbol{r}_k, \boldsymbol{r}_k)/(\boldsymbol{p}_k, A\,\boldsymbol{p}_k)$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k\,\boldsymbol{p}_k$
    $\boldsymbol{r}_{k+1} = \boldsymbol{r}_k + \alpha_k\,A\,\boldsymbol{p}_k$
    $\beta_k = (\boldsymbol{r}_{k+1}, \boldsymbol{r}_{k+1})/(\boldsymbol{r}_k, \boldsymbol{r}_k)$
    $\boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k\,\boldsymbol{p}_k$
    $k = k + 1$
end while

The number of the operations to be performed per step in this algorithm is shown in Table 1, which is a tremendous improvement over CG-1.

Table 1. Number of the operations per step in CG-2.

matrix-vector product	1
inner product	2
vector addition	3
scalar division	1
scalar comparison	1

Here we show the experimantal result for some 10000-dimensional equation.



Relative residual and error versus iteration number k.

References

[1] James M. Ortega, Introduction to Parallel and Vector Solution of Linear Systems, 1989, Prenum Press, New York.

[2] Gilbert W. Stewart, Afternotes Goes to Graduate School, 1997, SIAM, Philadelphia.

Problem

1. For the function $P(\boldsymbol{x})$ given by
$$P(\boldsymbol{x}) = \frac{1}{2}\,(\boldsymbol{x}-\boldsymbol{x}^*)^T\,A\,(\boldsymbol{x}-\boldsymbol{x}^*),$$
show that the relation
$$P(\boldsymbol{x}) = Q(\boldsymbol{x}) + \frac{1}{2}(\boldsymbol{x}^*)^T A\boldsymbol{x}^*,$$
where $\boldsymbol{x}^*$ is the solution of $A\,\boldsymbol{x} = \boldsymbol{b}$, and $Q(x)$ is the function given by (1).

2. Prove equation (4).

3. Show that the equation (9).

4. Show that the vectors $\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{n-1}$ given by (10) are linearly independent.

5. Show that equation (11) is valid.

6. Show that diagonal elements of symmetric positive definite matrices are positive.

7. For CG-1 make the table as Table 1.

8. Show that the two algorithms, CG-1 and CG-2, are equivalent.

9. Show that the matrix given by
$$A = \begin{pmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & \text{\Large 0} & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \ddots \\ & \text{\Large 0} & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{pmatrix}$$
is symmetric positive definite.

10. Solve the equation $A\,\boldsymbol{x} = \boldsymbol{b}$ by CG-1 and CG-2, and compare the CPU times for the case that A is the matrix given by above and the vector $\boldsymbol{b}$ is given by
$$\boldsymbol{b} = A\,\boldsymbol{u},$$
where the dimension of the equation is $n = 10000$ and each component of $\boldsymbol{u}$ is a random number in the range $(0, 1)$.

Appendix

1 Inner product

Let $\boldsymbol{x} = (x_1, \dots, x_n)^T$ and $\boldsymbol{y} = (y_1, \dots, y_n)^T$ be real vectors in $\mathbb{R}^n$. Then we define the inner product of $\boldsymbol{x}$ and $\boldsymbol{y}$ by

$$(\boldsymbol{x},\, \boldsymbol{y}) = \boldsymbol{x}^T \boldsymbol{y} = \sum_{i=1}^{n} x_i\, y_i. \tag{16}$$

The inner product has the following properties:

$$\begin{aligned}
&(\boldsymbol{x},\, \boldsymbol{y}) = (\boldsymbol{y},\, \boldsymbol{x}) \\
&(\alpha\, \boldsymbol{x},\, \boldsymbol{y}) = \alpha\, (\boldsymbol{x},\, \boldsymbol{y}), \quad \alpha \text{ is scalar} \\
&(\boldsymbol{x},\, \boldsymbol{y} + \boldsymbol{z}) = (\boldsymbol{x},\, \boldsymbol{y}) + (\boldsymbol{x},\, \boldsymbol{z}) \\
&(\boldsymbol{x},\, \boldsymbol{x}) \geq 0, \quad \text{`$=$' holds, only if } \boldsymbol{x} = \boldsymbol{0}.
\end{aligned} \tag{17}$$

For nonzero vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, if $(\boldsymbol{x},\, \boldsymbol{y}) = \boldsymbol{0}$ then the vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ are said to be *orthogonal*. If the vector $\boldsymbol{x}$ is orthogonal with the linearly independent n vectors $\boldsymbol{p}_i$ $(i = 1, \dots, n)$ in $\mathbb{R}^n$, then $\boldsymbol{x} = \boldsymbol{0}$, since from the assumption we have

$$\begin{pmatrix} \boldsymbol{p}_1^T \\ \boldsymbol{p}_2^T \\ \vdots \\ \boldsymbol{p}_n^T \end{pmatrix} \boldsymbol{x} = \boldsymbol{0},$$

and the matrix in the left-hand side is nonsingular, so that $\boldsymbol{x} = \boldsymbol{0}$.

Next we show the set of n vectors $\boldsymbol{p}_i$ $(i = 1, \dots, n)$ which are orthogonal with each other are linearly independent. Since, if not so, that is, if $\boldsymbol{p}_i$ $(i = 1, \dots, n)$ are linearly dependent, then there exist constants c_i $(i = 1, \dots, n)$, not necessarily all zero, such that

$$c_1 \boldsymbol{p}_1 + \cdots + c_n \boldsymbol{p}_n = \boldsymbol{0}.$$

From this we have for all j

$$\boldsymbol{0} = (c_1 \boldsymbol{p}_1 + \cdots + c_n \boldsymbol{p}_n,\, \boldsymbol{p}_j) = c_j\, (\boldsymbol{p}_j,\, \boldsymbol{p}_j),$$

which means $c_j = 0$. This contradicts the assumption.

2 Eigenvalues and eigenvectors of real symmetric matrices

When a real matrix A satisfies $a_{ij} = a_{ji}$, that is, $A^T = A$, the matrix A is called *real symmetric matrix*. The eigenvalues and eigenvectors of real symmetric matrices have the following properties:

1. Eigenvalues are real
   Let λ and $\boldsymbol{x}$ be any eigenvalue end eigenvector of A, respectively. Then we have

$$(\overline{A\,\boldsymbol{x}},\,\boldsymbol{x}) = (\overline{\lambda\boldsymbol{x}},\,\boldsymbol{x}) = \bar{\lambda}\,(\bar{\boldsymbol{x}},\,\boldsymbol{x}),$$

   where $^{-}$ denotes complex conjugate. On the other hand, from the property of inner product we have

$$(\overline{A\,\boldsymbol{x}},\,\boldsymbol{x}) = (\bar{\boldsymbol{x}},\,A\boldsymbol{x}) = \lambda\,(\bar{\boldsymbol{x}},\,\boldsymbol{x}).$$

   These two expressions mean $\bar{\lambda} = \lambda$, since $(\bar{\boldsymbol{x}},\,\boldsymbol{x}) \neq 0$.

2. Orthogonality of eigenvectors
   Let λ_i and λ_j be the eigenvalues of A and assume $\lambda_i \neq \lambda_j$. Then we have

$$(A\boldsymbol{x_i},\,\boldsymbol{x_j}) = \lambda_i(\boldsymbol{x_i},\,\boldsymbol{x_j}),$$

   and

$$(A\boldsymbol{x}_i,\,\boldsymbol{x}_j) = (\boldsymbol{x}_i,\,A\boldsymbol{x}_j) = \lambda_j(\boldsymbol{x}_i,\,\boldsymbol{x}_j),$$

   since A is symmetric, which implies

$$(\lambda_i - \lambda_j)\,(\boldsymbol{x}_i,\,\boldsymbol{x}_j) = 0.$$

   Thus we have $(\boldsymbol{x}_i,\,\boldsymbol{x}_j) = 0$.

3 Quadratic form

For a real symmetric matrix $A = (a_{ij})$ and a real vector $\boldsymbol{x} = (x_1, \ldots, x_n)^T$, the quantity given by

$$Q\,(\boldsymbol{x}) = \boldsymbol{x}^T\,A\,\boldsymbol{x} = (\boldsymbol{x},\,A\,\boldsymbol{x}) = \sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}\,x_i\,x_j \tag{18}$$

is called *quadratic form*. For any $\boldsymbol{x} \neq \boldsymbol{0}$, if $Q\,(\boldsymbol{x}) > 0\,(\geq 0)$, then the matrix A is said to be *positive (semi-) definite*. Positive definite matrices have the following properties:

1. The diagonal elements of symmetric positive definite matrix is positive.
   This is clear from

$$a_{ii} = (\boldsymbol{e}_i, A\,\boldsymbol{e}_i).$$

2. All the eigenvalues of symmetric positive definite matrix A are positive.
   For any $\boldsymbol{x} \neq 0$, if we transform $\boldsymbol{x}$ by $\boldsymbol{y} = T\,\boldsymbol{x}$, where $T = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n)$, and $\boldsymbol{u}_i$ is the eigenvector of A corresponding to λ_i, then

$$0 < \boldsymbol{x}^T\,A\,\boldsymbol{x} = \boldsymbol{y}^T\,(T^T A\,T)\,\boldsymbol{y} = \boldsymbol{y}^T\,(T^{-1}A\,T)\,\boldsymbol{y} = \sum_{i=1}^{n}\lambda_i\,y_i^2,$$

   which means $\lambda_i > 0$ for all i.

4 Theorem

Here we show again the algorithm CG-1:

1: Choose a small value $\varepsilon > 0$ and initial guess $\boldsymbol{x}^0$
2: $\boldsymbol{p}_0 = \boldsymbol{r}_0 = \boldsymbol{b} - A\,\boldsymbol{x}_0$, and compute $(\boldsymbol{r}_0, \boldsymbol{r}_0)$
3: $k = 0$
4: **while** $(\boldsymbol{r}_k, \boldsymbol{r}_k) \geq \varepsilon$ **do**
5: $\quad \alpha_k = -(\boldsymbol{r}_k, \boldsymbol{p}_k)/(\boldsymbol{p}_k, A\,\boldsymbol{p}_k)$
6: $\quad \boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k\,\boldsymbol{p}_k$
7: $\quad \boldsymbol{r}_{k+1} = \boldsymbol{r}_k + \alpha_k\,A\,\boldsymbol{p}_k$
8: $\quad \beta_k = -(\boldsymbol{p}_k, A\,\boldsymbol{r}_{k+1})/(\boldsymbol{p}_k, A\,\boldsymbol{p}_k)$
9: $\quad \boldsymbol{p}_{k+1} = \boldsymbol{r}_{k+1} + \beta_k\,\boldsymbol{p}_k$
10: $\quad k = k + 1$
11: **end while**

Theorem Let A be an $n \times n$ symmetric positive definite matrix, and $\boldsymbol{x}^*$ be the solution of the equation $A\,\boldsymbol{x} = \boldsymbol{b}$. Then the vectors $\boldsymbol{p}_k$ generated by CG-1 algorithm satisfy

$$\boldsymbol{p}_k^T A\,\boldsymbol{p}_j = 0, \qquad 0 \leq j < k, \quad k = 1, \ldots, n-1, \tag{19}$$

$$\boldsymbol{r}_k^T \boldsymbol{r}_j = 0, \qquad 0 \leq j < k, \quad k = 1, \ldots, n-1, \tag{20}$$

and $\boldsymbol{p}_k \neq 0$ unless $\boldsymbol{x}_k = \boldsymbol{x}^*$.

Proof By the definitions of α_j, β_j and those of $\boldsymbol{p}_j, \boldsymbol{r}_{j+1}$, we have

$$\boldsymbol{p}_j^T \boldsymbol{r}_{j+1} = \boldsymbol{p}_j^T \boldsymbol{r}_j + \alpha_j\,\boldsymbol{p}_j^T A\,\boldsymbol{p}_j = 0, \qquad j = 0, 1, \ldots, \tag{21}$$

$$\boldsymbol{p}_j^T A\,\boldsymbol{p}_{j+1} = \boldsymbol{p}_j^T A\,\boldsymbol{r}_{j+1} + \beta_j\,\boldsymbol{p}_j^T A\,\boldsymbol{p}_j = 0, \qquad j = 0, 1, \ldots \tag{22}$$

Here we assume, as an induction hypothesis, that (19) and (20) hold for some $k < n-1$. Then we must show these hold for $k+1$. Since $\boldsymbol{p}_0 = \boldsymbol{r}_0$, these hold for $k = 1$. For any $j < k$, using the 7th and 9th lines in CG-1, we have

$$\begin{aligned} \boldsymbol{r}_j^T\,\boldsymbol{r}_{k+1} &= \boldsymbol{r}_j^T\,(\boldsymbol{r}_k + \alpha_k\,A\,\boldsymbol{p}_k) \\ &= \boldsymbol{r}_j^T\,\boldsymbol{r}_k + \alpha_k\,\boldsymbol{p}_k^T A\,\boldsymbol{r}_j \\ &= \boldsymbol{r}_j^T\,\boldsymbol{r}_k + \alpha_k\,\boldsymbol{p}_k^T A\,(\boldsymbol{p}_j - \beta_{j-1}\,\boldsymbol{p}_{j-1}) = 0, \end{aligned} \tag{23}$$

since all three terms in the last line vanish by induction hypothesis. Moreover, we have from (21) and (22)

$$\begin{aligned} \boldsymbol{r}_k^T \boldsymbol{r}_{k+1} &= (\boldsymbol{p}_k - \beta_{k-1}\boldsymbol{p}_{k-1})^T\,\boldsymbol{r}_{k+1} \\ &= -\beta_{k-1}\,\boldsymbol{p}_{k-1}^T\,\boldsymbol{r}_{k+1} \\ &= -\beta_{k-1}\,\boldsymbol{p}_{k-1}^T\,(\boldsymbol{r}_k + \alpha_k\,A\,\boldsymbol{p}_k) \\ &= 0. \end{aligned}$$

Thus we have shown that (20) is true also for $k+1$. Next we have for any $j < k$

$$\boldsymbol{p}_j^T A\,\boldsymbol{p}_{k+1} = \boldsymbol{p}_j^T A\,(\boldsymbol{r}_{k+1} + \beta_k\,\boldsymbol{p}_k) = \boldsymbol{p}_j^T A\,\boldsymbol{r}_{k+1} = \alpha_j^{-1}\,(\boldsymbol{r}_{j+1} - \boldsymbol{r}_j)^T\,\boldsymbol{r}_{k+1} = 0, \tag{24}$$

if $\alpha_j \neq 0$, which we will show later. Therefore (19) is true also for $k+1$.

Finally, we show $\alpha_j \neq 0$. By the definition of $\boldsymbol{p}_j$ and (22) we have

$$\boldsymbol{r}_j^T\boldsymbol{p}_j = \boldsymbol{r}_j^T(\boldsymbol{r}_j + \beta_{j-1}\,\boldsymbol{p}_{j-1}) = \boldsymbol{r}_j^T\boldsymbol{r}_j.$$

Hence we have

$$\alpha_j = -\boldsymbol{r}_j^T\boldsymbol{r}_j/\boldsymbol{p}_j^T A\,\boldsymbol{p}_j.$$

Therefore, if $\alpha_j = 0$, then $\boldsymbol{r}_j = 0$, that is, $\boldsymbol{x}_j = \boldsymbol{x}^*$ so that the process stops with $\boldsymbol{x}_j$.

5 Differentiation by vectors

Here we define the differentiation of the scalar-valued function $J(\boldsymbol{x})$ with respect to its vector argument by

$$J'(\boldsymbol{x}) := \begin{pmatrix} \frac{\partial J}{\partial x_1} \\ \frac{\partial J}{\partial x_2} \\ \vdots \\ \frac{\partial J}{\partial x_n} \end{pmatrix}. \tag{25}$$

According to this definition, we have

$$\begin{aligned} Q'(\boldsymbol{x}) &= \left(\frac{1}{2}\,\boldsymbol{x}^T A\,\boldsymbol{x} - \boldsymbol{b}^T\,\boldsymbol{x}\right)' \\ &= A\,\boldsymbol{x} - \boldsymbol{b}, \end{aligned} \tag{26}$$

since

$$\begin{aligned} Q(\boldsymbol{x}) &= \frac{1}{2}\sum_{i=1}^{n} a_{ii}\,x_i^2 + \sum_{i\neq j} a_{ij}\,x_i\,x_j, \\ \boldsymbol{b}^T\boldsymbol{x} &= \sum_{i=1}^{n} b_i\,x_i, \end{aligned}$$

and

$$\begin{aligned} \frac{\partial}{\partial x_k}Q(\boldsymbol{x}) &= a_{kk}\,x_k + \sum_{j\neq k} a_{kj}\,x_j = \sum_{j=1}^{n} a_{kj}\,x_j, \\ \frac{\partial}{\partial x_k}(\boldsymbol{b}^T\boldsymbol{x}) &= b_k. \end{aligned}$$