Convergence of the Conjugate Gradient Method on Singular Systems

Ken Hayami

NII-2018-001E
June. 2018
Convergence of the Conjugate Gradient Method on Singular Systems*

Ken Hayami†

Abstract

In this short paper, we analyze the convergence of the Conjugate Gradient (CG) method in exact arithmetic, when the coefficient matrix \( A \) is symmetric positive semidefinite and the system is consistent. To do so, we diagonalize \( A \) and decompose the algorithm into the range and the null space components of \( A \). Further, we apply the analysis to the CGLS and CGNE (CG Normal Error) methods for rank-deficient least squares problems.

Key words: conjugate gradient method – singular systems – least squares problems –
Mathematical Sub Classification Number: 65F02

1 Introduction

In exact arithmetic, the Conjugate Gradient (CG) method [4] converges to the true solution within \( n \) iterations for a system of linear equations

\[
Ax = b, \quad A \in \mathbb{R}^{n \times n},
\]

if \( A \) is symmetric positive definite.

When \( A \) is symmetric positive semidefinite, the CG method still converges to a solution, provided \( b \in \mathbb{R}^n \). (This fact is stated, for instance, in Kaasschieter [5], p.266.) In this short paper, we will show this explicitly by diagonalizing \( A \) and decomposing the CG method into the \( \mathcal{R}(A) \) component and the \( \mathcal{N}(A) = \mathcal{R}(A)^\perp \) component, where \( \mathcal{R}(A) \) is the range space of \( A \), and \( \mathcal{N}(A) \) is the null space of \( A \), and \( S^\perp \) is the orthogonal complement of a subspace \( S \). This follows our analysis of the GMRES (Generalized Minimal Residual) and GCR (Generalized Conjugate Residual) methods in [3], but the analysis is much simpler in the symmetric case.

Using this fact, we will also show that the CGLS and CGNE (CG Normal Error) methods converge for least squares problems

\[
\min_{x \in \mathbb{R}^n} \|b - Ax\|_2, \quad A \in \mathbb{R}^{m \times n},
\]
even when \( A \) is rank-deficient.

*The contents of this paper was presented as a part of [2]
†National Institute of Informatics, 2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan, and SO-KENDAI (The Graduate University for Advanced Studies), e-mail: hayami@nii.ac.jp
2 The CG algorithm

The CG algorithm [4] for solving (1.1) is given as follows, where \( x_0 \) is the initial approximate solution.

The CG algorithm

\[
\begin{align*}
& r_0 = b - Ax_0 \\
& p_0 = r_0 \\
& \text{For } i = 0, 1, \ldots, \text{ until convergence, Do} \\
& \quad \alpha_i = \frac{(r_i, r_i)}{(Ap_i, p_i)} \\
& \quad x_{i+1} = x_i + \alpha_i p_i \\
& \quad r_{i+1} = r_i - \alpha_i Ap_i \\
& \quad \beta_i = \frac{(r_{i+1}, r_{i+1})}{(r_i, r_i)} \\
& \quad p_{i+1} = r_{i+1} + \beta_i p_i \\
& \text{End Do}
\end{align*}
\]

3 Decomposition of the CG algorithm into the range and null space components

In the following, we assume exact arithmetic. When \( A \) is symmetric positive definite, the CG method is known to converge to the true solution within \( n \) iterations [4]. When \( A \) is symmetric positive semidefinite, we will analyze the method as follows.

Let

\[
Q^T AQ = \Lambda = \begin{bmatrix}
\lambda_1 & 0 & \cdots & 0 \\
0 & \ddots & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & \cdots & \lambda_r
\end{bmatrix},
\]

where

\[
\Lambda_r = \begin{bmatrix}
\lambda_1 \\
\vdots \\
\lambda_r
\end{bmatrix}, \quad \lambda_1 \geq \lambda_2 \cdots \geq \lambda_r > 0, \quad r = \text{rank} A = \dim\mathcal{R}(A),
\]

and

\[
Q^T Q = I_n, \quad Q = [Q_1, Q_2],
\]

\[
Q_1 = [q_1, \cdots, q_r], \quad Q_2 = [q_{r+1}, \cdots, q_n].
\]

Here, \( q_1, \cdots, q_r \) and \( q_{r+1}, \cdots, q_n \) are orthonormal basis vectors of \( \mathcal{R}(A) \) and \( \mathcal{R}(A)^\perp = \mathcal{N}(A) \), respectively.

Following [3], we will decompose the CG algorithm into the \( \mathcal{R}(A) \) component and the \( \mathcal{R}(A)^\perp = \mathcal{N}(A) \) component using the following transformation:

\[
\tilde{v} = Q^T v = [Q_1, Q_2]^T v = \begin{bmatrix} Q_1^T v \\ Q_2^T v \end{bmatrix} = \begin{bmatrix} v^1 \\ v^2 \end{bmatrix}
\]
or

\[ \mathbf{v} = Q\mathbf{v} = [Q_1, Q_2] \begin{bmatrix} \mathbf{v}^1 \\ \mathbf{v}^2 \end{bmatrix} = Q_1\mathbf{v}^1 + Q_2\mathbf{v}^2. \]

Here, \( Q_1\mathbf{v}^1 \) and \( Q_2\mathbf{v}^2 \) correspond to the \( \mathcal{R}(A) \) and \( \mathcal{N}(A) \) component of \( \mathbf{v} \), respectively.

Noting that

\[ r = b - Ax, \]

\[ Q^T r = Q^T b - Q^T AQ^T x, \]

\[ \begin{bmatrix} r^1 \\ r^2 \end{bmatrix} = \begin{bmatrix} b^1 \\ b^2 \end{bmatrix} - \begin{bmatrix} \Lambda_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x^1 \\ x^2 \end{bmatrix}, \]

\[ r^1 = b^1 - \Lambda_r x^1, \]
\[ r^2 = b^2, \]
\[ (r, r) = (r^1, r^1) + (b^2, b^2), \]
\[ (Ap, p) = (p^1, \Lambda_r p^1), \]

we obtain the following.

**Decomposed CG algorithm (General case)**

<table>
<thead>
<tr>
<th>( \mathcal{R}(A) ) component</th>
<th>( \mathcal{N}(A) ) component</th>
</tr>
</thead>
<tbody>
<tr>
<td>( r_0^1 = b^1 - \Lambda_r x_0^1 )</td>
<td>( r_0^2 = b^2 )</td>
</tr>
<tr>
<td>( p_0^1 = r_0^1 )</td>
<td>( p_0^2 = b^2 )</td>
</tr>
</tbody>
</table>

For \( i = 0, 1, \ldots \), until convergence Do

\[ \alpha_i = \frac{(r^1, r^1) + (b^2 b^2)}{(p_i, \Lambda_r p_i^1)} \]
\[ x_{i+1}^1 = x_i^1 + \alpha_i p_i \]
\[ r_{i+1}^1 = r_i^1 - \alpha_i \Lambda_r p_i \]
\[ \beta_i = \frac{(r_{i+1}, r_{i+1}) + (b^2 b^2)}{(r_i, r_i) + (b^2 b^2)} \]
\[ p_{i+1}^1 = r_{i+1}^1 + \beta_i p_i \]

End Do

Further, if we assume that the system is consistent, i.e., \( b \in \mathcal{R}(A) \) or \( b^2 = 0 \), we have the following.
Decomposed CG algorithm \((b \in \mathcal{R}(A): b^2 = 0)\)

<table>
<thead>
<tr>
<th>(\mathcal{R}(A)) component</th>
<th>(\mathcal{N}(A)) component</th>
</tr>
</thead>
<tbody>
<tr>
<td>(r_0^1 = b^1 - \Lambda_r x_0^1)</td>
<td>(r_0^2 = 0)</td>
</tr>
<tr>
<td>(p_0^1 = r_0^1)</td>
<td>(p_0^2 = 0)</td>
</tr>
</tbody>
</table>

For \(i = 0, 1, \ldots\), until convergence Do

\[
\begin{align*}
\alpha_i &= \frac{(r_i^1, r_i^1)}{(p_i^1, \Lambda_r p_i^1)} \\
x_{i+1}^1 &= x_i^1 + \alpha_i p_i^1 \\
r_{i+1}^1 &= r_i^1 - \alpha_i \Lambda_r p_i^1 \\
\beta_i &= \frac{(r_i^1, r_{i+1}^1)}{(r_i^1, r_i^1)} \\
p_{i+1}^1 &= r_{i+1}^1 + \beta_i p_i^1
\end{align*}
\]

End Do

Note that in the above, the \(\mathcal{R}(A)\) component is equivalent to CG applied to \(r_{x, i} = b^1\), where \(\Lambda_r\) is symmetric positive definite. Hence, \(x_{i+1}^1\) will converge to \(x_{i}^1 = \Lambda_r^{-1} b^1\) within \(r = \text{rank}A\) iterations. Thus, \(r_{i+1}^1\) will converge to \(0\). Therefore, the \(\mathcal{R}(A)\) component \(Q_1 x_{i}^1\) of \(x_{i}\) will converge to \(A^\dagger b\), and the \(\mathcal{R}(A)\) component of \(r_{i}\) will converge to \(0\).

As for the \(\mathcal{N}(A)\) component, since \(x_{i}^2 = x_{i}^2\) and \(r_{i}^2 = 0\) for \(i = 0, 1, \ldots\), \(Q_2 x_{i}^2 = Q_2 x_{0}^2\) and \(Q_2 r_{i}^2 = 0\) for \(i = 0, 1, \ldots\). Thus, \(x_{i}\) will converge to a solution of (1.1).

If further, \(x_0 \in \mathcal{R}(A)\) (e.g. \(x_0 = 0\)), then, \(x_{i}^2 = 0\). Thus, \(x_{i}^1 = 0\), \(Q_2 x_{i} = 0\) for \(i = 0, 1, \ldots\). Therefore, \(x_{i}\) will converge to the minimum norm solution (pseudo inverse solution) \(A^\dagger b\) of (1.1).

Analysis similar to the above was done in Washizawa et al. ([8], Subsection 4.2). Here, we have further simplified the analysis by diagonalizing \(A\).

We also obtain the following error bound:

\[
||e_k^1||_{\Lambda_r} = ||r_k^1||_{\Lambda_r^{-1}} \leq 2 \left( \frac{\sqrt{\kappa(\Lambda_r)} - 1}{\sqrt{\kappa(\Lambda_r)} + 1} \right)^k ||r_0^1||_{\Lambda_r^{-1}},
\]

where

\[
e_k^1 = x_k^1 - x_*^1, \quad \kappa(\Lambda_r) = \frac{\lambda_1}{\lambda_r}.
\]

Note here that

\[
||e^1||_{\Lambda_r}^2 = e^T \Lambda_r e^1 = e^T A e = ||e||_A
\]

and

\[
||r^1||_{\Lambda_r^{-1}}^2 = r^T \Lambda_r^{-1} r^1 = r^T A^\dagger r.
\]

Note also that

\[
A = QAQ^T, \\
A^\dagger = QA^\dagger Q^T,
\]

4
where

\[
\Lambda^t = \begin{bmatrix}
\Lambda_r^{-1} & 0 & \cdots & 0 \\
0 & \ddots & \ddots & \vdots \\
\cdots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & 0
\end{bmatrix}.
\]

Here

\[
x_* = A^t b = [Q_1, Q_2] \begin{bmatrix}
\Lambda_r^{-1} & 0 & \cdots & 0 \\
0 & \ddots & \ddots & \vdots \\
\cdots & \ddots & \ddots & 0 \\
0 & \cdots & 0 & 0
\end{bmatrix} \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix} b = Q_1 \Lambda_r^{-1} b = Q_1 x_*^t.
\]

4 Application to rank-deficient least squares problems

Next, we apply the above analysis to the least squares problem

\[
\min_{x \in \mathbb{R}^n} \| b - Ax \|_2,
\]

where now \( A \in \mathbb{R}^{m \times n} \) and \( m \geq n \), and \( A \) may not necessarily be of full rank.

Equation (4.1) is equivalent to the normal equation (of the first kind):

\[
A^T A x = A^T b.
\]

Here, \( A^T A \) is square, symmetric, and positive semidefinite. Moreover, (4.2) is consistent, since \( A^T b = A^T b \in \mathcal{R}(A^T A) = \mathcal{R}(A^T) \). Thus, we may apply the Conjugate Gradient method to (4.2) with \( A := A^T A \) and \( b := A^T b \) to obtain the following CGLS(CGNR(CG Normal Residual)) method[1, 7].

CGLS method

Choose \( x_0 \).

\[
\begin{align*}
q_i &= A p_i, \\
\alpha_i &= \gamma_i / \| q_i \|_2^2, \\
x_{i+1} &= x_i + \alpha_i p_i, \\
r_{i+1} &= r_i - \alpha_i q_i, \\
s_{i+1} &= A^T r_{i+1}, \\
\gamma_{i+1} &= \| s_{i+1} \|^2_2, \\
\beta_i &= \gamma_{i+1} / \gamma_i, \\
p_{i+1} &= s_{i+1} + \beta_i p_i.
\end{align*}
\]

For \( i = 0, 1, 2, \ldots \), until \( \gamma_i < \epsilon \) Do

End Do
In the above algorithm, the $\mathcal{R}(A^T)$ component of the solution $Q_1 x^k_1$ converges to $(A^T A)^T A^T b = A^T_1 b$, and $A^T r_k$ converges to $0$.

The $\mathcal{N}(A^T)$ component of the solution remains $x^2_k = x^2_0$.

Thus, if $x_0 \in \mathcal{R}(A^T)$ (e.g. $x_0 = 0$), $x^2_k = 0$, so that the solution $x_k$ converges to the minimum norm solution: $x_* = A^T b$.

This was shown by Kammerer and Nashed [6] (Theorem 5.1) in a Hilbert space setting. Here we gave a simpler derivation using matrix analysis.

We also obtain the following error bound:

$$||r_k||_{\mathcal{R}(A)}||_2 = e^T_k A^T A e_k \leq 2 \left( \frac{\sigma_1 - \sigma_r}{\sigma_1 + \sigma_r} \right)^k ||r_0||_{\mathcal{R}(A)}||_2,$$

where $\sigma_1 \geq \cdots \geq \sigma_r > 0$ are the nonzero singular values of $A$.

Similar analysis can be performed for the under-determined least squares problem:

$$\min_{\{x: A x = b\}} ||x||_2$$

with $m \leq n$, which is equivalent to the normal equation of the second kind:

$$A A^T y = b, \quad x = A^T y.$$

Applying CG to this equation with $A : A A^T$ gives the CGNE method [7]. Assume $b \in \mathcal{R}(A A^T) = \mathcal{R}(A)$. Further assume that the initial approximate solution satisfies $y_0 \in \mathcal{R}(A)$, which is equivalent to $x_0 \in \mathcal{R}(A^T A) = \mathcal{R}(A^T)$ (e.g. $y_0 = 0$, $x_0 = 0$). Then, $y_k$ will converge to $(A A^T)^T b$, and the solution $x_k$ will converge to $A^T (A A^T)^T b = A^T_1 b$, the min-norm solution.

We obtain the following error bound:

$$r^T_k (A A^T)^T r_k \leq 2 \left( \frac{\sigma_1 - \sigma_r}{\sigma_1 + \sigma_r} \right)^k r^T_0 (A A^T)^T r_0.$$

## 5 Concluding remark

In this paper, we analyzed the convergence of the CG method in exact arithmetic, when the coefficient matrix $A$ is symmetric positive semidefinite and the system is consistent. To do so, we diagonalized $A$ and decomposed the algorithm into the $\mathcal{R}(A)$ component and its orthogonal complement space $\mathcal{N}(A)$ component. Further, we applied the analysis to the CGLS and CGNE methods for rank-deficient least squares problems.

## 6 Acknowledgenets

This work was supported by the Grant-in-Aid for Scientific Research (Fundamental Research (C), Grant No. 15K04768) of the Ministry of Education, Culture, Sports, Science and Technology, Japan.
References


