Journal of Computational Mathematics, Vol.20, No.4, 2002, 437–448.

A REGULARIZED CONJUGATE GRADIENT METHOD FOR SYMMETRIC POSITIVE DEFINITE SYSTEM OF LINEAR EQUATIONS*[1)]

Zhong-zhi Bai
(*LSEC, ICMSEC, Academy of Mathematics and System Sciences, Chinese Academy of Sciences, Beijing 100080, China*)

Shao-liang Zhang
(*Department of Applied Physics, University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan*)

Abstract

A class of regularized conjugate gradient methods is presented for solving the large sparse system of linear equations of which the coefficient matrix is an ill-conditioned symmetric positive definite matrix. The convergence properties of these methods are discussed in depth, and the best possible choices of the parameters involved in the new methods are investigated in detail. Numerical computations show that the new methods are more efficient and robust than both classical relaxation methods and classical conjugate direction methods.

Key words: Conjugate gradient method, Symmetric positive definite matrix, Regularization, Ill-conditioned linear system.

1. Introduction

Let $\mathbb{R}^n$ represent the real n-dimensional vector space, and $\mathbb{R}^{n\times n}$ the real $n\times n$ matrix space. In this paper, we will study iterative methods for solving the system of linear equations

$$Ax = b, \qquad A \in \mathbb{R}^{n\times n} \quad \text{and} \quad x, b \in \mathbb{R}^n, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ is a large sparse and possibly very ill-conditioned *symmetric positive definite* (SPD) matrix, $x \in \mathbb{R}^n$ the unknown vector, and $b \in \mathbb{R}^n$ a given *right-hand side* (RHS) vector.

The *conjugate gradient* (CG) method [9] is an efficient solver for approximating the solution of the system of linear equations (1), provided the coefficient matrix $A \in \mathbb{R}^{n\times n}$ is well-conditioned, or a good preconditioner is cheaply obtainable when it is ill-conditioned. A preconditioner transforms the original linear system (1) by a suitable linear transformation such that the spectral property of the matrix $A \in \mathbb{R}^{n\times n}$ is largely improved, and therefore, the convergence speed of the CG method is considerably accelerated. Two typical ways of constructing a practical preconditioner for an SPD matrix are the *symmetric successive overrelaxation* (SSOR) iteration [16, 1, 2, 3] and the *incomplete Cholesky* (IC) factorization [2, 12]. However, both SSOR and IC preconditioners are only applicable and efficient for a special class of SPD system of linear equations, e.g., a diagonally dominant or an irreducibly weakly diagonally dominant one which may come from the discretization of a second-order self-adjoint elliptic boundary value problem by the finite difference method [16, 12, 2, 15]. Moreover, the IC factorization may break down even for an SPD matrix [11]. Therefore, the existence of an IC factor can not be guaranteed even if we neglect the influence of the rounding error, needless to say its stability and accuracy.

Considering that the CG method is quite efficient for solving an SPD system of linear equations whose coefficient matrix has tightly clustered spectrum [2, 6, 15], in this paper, we present a class of *regularized conjugate gradient* (RCG) method for solving the system of linear equations (1). In the RCG method, the linear system (1) is first regularized by reasonably

* Received March 12, 2000.

[1)] Subsidized by The Special Funds For Major State Basic Research Projects G1999032803.

shifting and contracting the spectrum of the coefficient matrix $A \in \mathbb{R}^{n\times n}$, and its solution is then approximated successively by a sequence of regularized linear systems. At each step of iteration, the regularized linear system itself is iteratively solved by the CG method. Therefore, the RCG method is actually an inner/outer iterative method [13, 14, 7, 5, 8] with a standard splitting iteration as its outer iteration, and the CG iteration as its inner iteration. Evidently, this new approach is quite different from the typical ones, such as the classical relaxation methods [16] and the classical conjugate direction methods [2, 15]. Moreover, the RCG method itself can be again preconditioned by employing an IC or an SSOR preconditioner to the regularized linear system. Then, the CG method is directly applied to this preconditioned regularized linear system at each outer iterate. This naturally leads to a so-called *preconditioned regularized conjugate gradient* (PRCG) method for solving the system of linear equations (1). In actual implementation of the PRCG method, we can suitably choose the shift and contract factors in such a way that the regularized linear system has reasonably good diagonally dominant property such that the IC or the SSOR preconditioner is existent, stable, and accurate, and hence, make it a highly efficient method for solving the system of linear equations (1).

We prove the convergence and estimate the relative residual and error of both RCG and PRCG methods. In particular, we discuss the best possible choices of the shift and contract factors, as well as the best possible number of the inner CG iteration steps. Both theoretical analyses and numerical experiments show that the new regularized conjugate gradient method and its preconditioned variant converge much faster and more robust to the exact solution of the system of linear equations (1) than both classical relaxation methods and classical conjugate direction methods.

2. The Regularized Conjugate Gradient Method

For an SPD matrix $A \in \mathbb{R}^{n\times n}$, we use $\sigma(A)$ to represent its spectrum set, and $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ its smallest and largest eigenvalues, respectively. Denote $\mathcal{I}(A) = [\lambda_{\min}(A), \lambda_{\max}(A)]$. Then any $\lambda \in \sigma(A)$ satisfies $\lambda \in \mathcal{I}(A)$. The condition number $\kappa_2(A)$ of the matrix $A \in \mathbb{R}^{n\times n}$ with respect to the Euclidean norm is given by $\kappa_2(A) = \|A\|_2\|A^{-1}\|_2 = \dfrac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$. For any $z \in \mathbb{R}^n$, its A-norm is defined by $\|z\|_A = \sqrt{z^TAz}$.

We use $\nu = \frac{\mu}{\eta}$ to represent the horizontal intercept of the linear transformation $f : \mathbb{R}^1 \to \mathbb{R}^1$,

$$f(t) = \mu + \eta t, \qquad \mu, \eta \in \mathbb{R}^1, \quad \eta \neq 0. \tag{2}$$

Evidently, $f(A) = \eta A(\nu)$, where $A(\nu) = \nu I + A$, is the transformed matrix. The linear transformation (2) maps the spectrum set $\sigma(A)$ of the matrix $A \in \mathbb{R}^{n\times n}$ onto a new set $f(\sigma(A)) = \mu + \eta\sigma(A)$ which is obviously contained in the interval $\mu + \eta\mathcal{I}(A)$. If we choose the reals μ and η such that $\eta \neq 0$ and $\nu > 0$, then it immediately holds that

$$\kappa_2(f(A)) \equiv \frac{\mu + \eta\lambda_{\max}(A)}{\mu + \eta\lambda_{\min}(A)} = \frac{\nu + \lambda_{\max}(A)}{\nu + \lambda_{\min}(A)} < \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \equiv \kappa_2(A).$$

Therefore, the linear transformation (2) may considerably improve the condition number of the matrix $A \in \mathbb{R}^{n\times n}$, provided a pair of suitable constants μ and η is easily obtainable. The constant μ is called a shift, and the constant η a contractor when $|\eta| < 1$ and an amplifier when $|\eta| \geq 1$.

By the linear transformation (2), we can rewrite the system of linear equations (1) as

$$(\nu I + A)x = \nu x + b, \tag{3}$$

where $I \in \mathbb{R}^{n\times n}$ is the identity matrix and $\nu \geq 0$ a constant. Hence, the system of linear equations (1) is equivalent to the system of linear equations (3).

The basic idea of our new *regularized conjugate gradient* (RCG) method is as follows: Given a starting vector $x^{(0)} \in \mathbb{R}^n$; Suppose that we have got approximations $x^{(0)}, x^{(1)}, \ldots, x^{(k)}$ to the solution x^* of the system of linear equations (1), then the next approximation $x^{(k+1)}$ to x^* is obtained through solving the system of linear equations

$$(\nu I + A)x = \nu x^{(k)} + b \tag{4}$$

iteratively, with the CG method, to certain arithmetic precision. More precisely, this RCG method can be described as follows:

Method 2.1. (The RCG method).

```
 1. Input the largest admissible number of iteration steps $k_{\max}$ and the stopping tolerance $\varepsilon$ of
    the RCG method
 2. Input the largest admissible number of iteration steps $\ell_{\max}$ and the stopping tolerance $\epsilon$ of
    the inner CG iteration
 3. Input the starting vector $x$ and the iteration parameter $\nu$
 4. Set $k := 0$
 5. Compute $r = b - Ax$, $\rho^{(0)} = \|r\|_2^2$, and $\vartheta^{(0)} = \sqrt{\rho^{(0)}}$
 6. If $\vartheta^{(0)} \le \varepsilon\|b\|_2$ and $k \ge k_{\max}$, then GoTo 20
 7.     Set $\ell := 1$ and $y := x$
 8.     Do While $\vartheta^{(\ell-1)} > \epsilon\vartheta^{(0)}$ and $\ell < \ell_{\max}$
 9.         If $\ell = 1$ then Set $\beta := 0$ and $p := r$, Else
                Compute $\beta = \rho^{(\ell-1)}/\rho^{(\ell-2)}$ and $p = r + \beta p$
10.         Compute $w = \nu p + Ap$
11.         Compute $\alpha = \rho^{(\ell-1)}/p^T w$
12.         Compute $y = y + \alpha p$
13.         Compute $r = r - \alpha w$
14.         Compute $\rho^{(\ell)} = \|r\|_2^2$ and $\vartheta^{(\ell)} = \sqrt{\rho^{(\ell)}}$
15.         Set $\ell := \ell + 1$
16.     EndDo
17.     Set $x := y$
18.     Set $k := k + 1$
19. GoTo 5
20. Continue
21. EndDo
```

Lines 8-16 in Method 2.1 is actually the classical CG method applied to the system of linear equations (4) at the k-th outer iterate. Evidently, when $\mu = 0$ and $\eta = 1$, or equivalently, when $\nu = 0$, Method 2.1 automatically recovers the classical CG method [9, 6] for the system of linear equations (1), provided we set $\varepsilon = \epsilon$ and $k_{\max} = \ell_{\max}$.

The input of the RCG method is the initial iterate x, the RHS vector b, the largest admissible numbers of iteration steps $k_{\max}$ and $\ell_{\max}$, and the stopping tolerances ε and ϵ for the inner CG iteration and the RCG method, respectively, and a routine which computes the action of the matrix $A \in \mathbb{R}^{n\times n}$ on a vector. Note that the matrix A itself need not be formed or stored, only a routine for matrix-vector multiplications is required.

According to the costs, we need to store only the five vectors x, y, w, p and r. Each inner CG iteration requires a single matrix-vector multiplication (to compute Ap), two inner products (one for $p^T w$ and one to compute $\rho^{(\ell)} = \|r\|_2^2$), and three operations of the form $u + \xi v$, where u, v are vectors and ξ is a scalar. Each outer iteration requires one additional matrix-vector multiplication and one operation of the form $u - v$ (to compute $r = b - Ax$), and one inner product (to compute $\rho^{(0)} = \|r\|_2^2$). Therefore, if we assume that the number of nonzeros on the i-th row of the matrix $A \in \mathbb{R}^{n\times n}$ is $\omega^{(i)}$, then each inner CG iteration requires

$$W_{\text{cg}} = \sum_{i=1}^{n}(2\omega^{(i)} - 1) + 7n + 1 = 2\sum_{i=1}^{n}\omega^{(i)} + 6n + 1$$

flops, and the corresponding outer iteration requires

$$W_{\text{outer}} = \sum_{i=1}^{n}(2\omega^{(i)} - 1) + 3n - 1 = 2\sum_{i=1}^{n}\omega^{(i)} + 2n - 1$$

flops. Furthermore, if we assume that the k-th iterate of the RCG method requires $m^{(k)}$ steps of inner CG iteration to reach the exitting tolerance ϵ, then the total flops of the k-th iterate of the RCG method is

$$W_{\text{rcg}}^{(k)} = m^{(k)}W_{\text{cg}} + W_{\text{outer}} = 2(m^{(k)} + 1)\sum_{i=1}^{n}\omega^{(i)} + 2(3m^{(k)} + 1)n + m^{(k)} - 1.$$

In particular, in the case of $\omega^{(i)} = \omega (i = 1, 2, \ldots, n)$ and $m^{(k)} = m (k = 1, 2, \ldots)$, the flops of each global iterate of the RCG method is

$$W_{\text{rcg}} = 2((m+1)\omega + 3m + 1)n + m - 1. \tag{5}$$

3. Convergence Analysis

For a nonnegative integer k, we use $\mathcal{P}^{(k)}$ to denote the set of polynomials of degree at most k such that $p^{(k)}(0) = 1$ holds for any $p^{(k)} \in \mathcal{P}^{(k)}$. In addition, we define quantities

$$\begin{aligned}
\widetilde{\gamma}^{(k)}(\nu) &= \min_{p^{(k)} \in \mathcal{P}^{(k)}} \max_{z \in \sigma(A(\nu))} |p^{(k)}(z)|, \\
\widetilde{\delta}^{(k)}(\nu) &= \frac{\nu}{\nu + \lambda_{\min}(A)} + \frac{\lambda_{\max}(A)}{\nu + \lambda_{\max}(A)} \sqrt{\kappa_2(A(\nu))}\widetilde{\gamma}^{(k)}(\nu), \\
\widetilde{\theta}^{(k)}(\nu) &= \frac{\nu}{\nu + \lambda_{\min}(A)} + \frac{\lambda_{\max}(A)}{\nu + \lambda_{\min}(A)} \sqrt{\kappa_2(A(\nu))}\widetilde{\gamma}^{(k)}(\nu), \\
\gamma^{(k)}(\nu) &= 2\left(\frac{\sqrt{\kappa_2(A(\nu))} - 1}{\sqrt{\kappa_2(A(\nu))} + 1}\right)^k, \\
\delta^{(k)}(\nu) &= \frac{\nu}{\nu + \lambda_{\min}(A)} + \frac{\lambda_{\max}(A)}{\nu + \lambda_{\max}(A)} \sqrt{\kappa_2(A(\nu))}\gamma^{(k)}(\nu), \\
\theta^{(k)}(\nu) &= \frac{\nu}{\nu + \lambda_{\min}(A)} + \frac{\lambda_{\max}(A)}{\nu + \lambda_{\min}(A)} \sqrt{\kappa_2(A(\nu))}\gamma^{(k)}(\nu).
\end{aligned}$$

These notations will be used throughout the remainder of this paper.

The following lemma gives the sharpest possible bounds about the quantities $\widetilde{\gamma}^{(k)}(\nu)$, $\widetilde{\delta}^{(k)}(\nu)$, and $\widetilde{\theta}^{(k)}(\nu)$.

Lemma 3.1. *[4, 6, 15] If $A \in \mathbb{R}^{n \times n}$ is an SPD matrix, and ν satisfies $\lambda_{\min}(A) + \nu > 0$, then $\widetilde{\gamma}^{(k)}(\nu) \le \gamma^{(k)}(\nu)$, and hence, $\widetilde{\delta}^{(k)}(\nu) \le \delta^{(k)}(\nu)$ and $\widetilde{\theta}^{(k)}(\nu) \le \theta^{(k)}(\nu)$.*

For the relation between the relative residual in the Euclidean norm and the relative error in the A-norm, we have the following result.

Lemma 3.2. *[10] Let $A \in \mathbb{R}^{n \times n}$ be an SPD matrix. Then for any $z \in \mathbb{R}^n$,*

$$\|A^{\frac{1}{2}} z\|_2 = \|z\|_A$$

and

$$\sqrt{\lambda_{\min}(A)}\|z\|_A \le \|Az\|_2 \le \sqrt{\lambda_{\max}(A)}\|z\|_A.$$

In addition, the classical error bound for the CG method applied to the solution x^* of the system of linear equations (1) is given in the following lemma.

Lemma 3.3. *[10, 6, 15] Let $A \in \mathbb{R}^{n \times n}$ be an SPD matrix. If the CG method is started from an initial iterate $x^{(0)} \in \mathbb{R}^n$, then after k steps of iterates, it generates an approximation $x^{(k)}$ to the solution x^* of the system of linear equations (1), which satisfies*

$$\|x^{(k)} - x^*\|_A \le \widetilde{\gamma}^{(k)}(0)\|x^{(0)} - x^*\|_A \le \gamma^{(k)}(0)\|x^{(0)} - x^*\|_A,$$

where

$$\widetilde{\gamma}^{(k)}(0) = \widetilde{\gamma}^{(k)}(\nu)|_{\nu=0} = \min_{p^{(k)} \in \mathcal{P}^{(k)}} \max_{z \in \sigma(A)} |p^{(k)}(z)|,$$

and

$$\gamma^{(k)}(0) = \gamma^{(k)}(\nu)|_{\nu=0} = 2\left(\frac{\sqrt{\kappa_2(A)} - 1}{\sqrt{\kappa_2(A)} + 1}\right)^k.$$

With Lemmas 3.1-3.3 in hands, we can now demonstrate a precise estimate about the convergence speed of the RCG method.

Theorem 3.1. *Let $A \in \mathbb{R}^{n \times n}$ be an SPD matrix and $\nu \ge 0$ be a constant. If the RCG method is started from an initial iterate $x^{(0)} \in \mathbb{R}^n$, and applies $m^{(k)}$ steps of CG iteration to get the next approximation $x^{(k+1)}$ to the solution x^* of the system of linear equations (1), then it holds that:*

(a) $\|b-Ax^{(k+1)}\|_2 \leq \widetilde{\delta}^{(m^{(k)})}(\nu)\|b-Ax^{(k)}\|_2 \leq \delta^{(m^{(k)})}(\nu)\|b-Ax^{(k)}\|_2$;

(b) $\|x^{(k+1)}-x^*\|_2 \leq \widetilde{\theta}^{(m^{(k)})}(\nu)\|x^{(k)}-x^*\|_2 \leq \theta^{(m^{(k)})}(\nu)\|x^{(k)}-x^*\|_2$.

Proof. Denote $b^{(k)}(\nu)=\nu x^{(k)}+b$, $x^{(*,k)}$ the exact solution of the system of linear equations (4), i.e., it satisfies $A(\nu)x^{(*,k)}=b^{(k)}(\nu)$, and $y^{(k,m^{(k)})}$ the final result of the inner CG iteration at the k-th outer iterate of the RCG method. Then from Lemma 3.3 we know that

$$\|y^{(k,m^{(k)})}-x^{(*,k)}\|_{A(\nu)} \leq \widetilde{\gamma}^{(m^{(k)})}(\nu)\|y^{(k,0)}-x^{(*,k)}\|_{A(\nu)}$$

holds. Furthermore, since $y^{(k,0)}=x^{(k)}$ and $y^{(k,m^{(k)})}=x^{(k+1)}$, the above estimate immediately leads to

$$\|x^{(k+1)}-x^{(*,k)}\|_{A(\nu)} \leq \widetilde{\gamma}^{(m^{(k)})}(\nu)\|x^{(k)}-x^{(*,k)}\|_{A(\nu)}. \tag{6}$$

Define the residual vector of the RCG method at the k-th outer iterate by $r^{(k)}$, i.e., $r^{(k)}=b-Ax^{(k)}$. Then we have

$$x^{(k+1)}-x^{(k)}=A^{-1}(Ax^{(k+1)}-Ax^{(k)})=A^{-1}(r^{(k)}-r^{(k+1)}).$$

This equality and actual computations straightforwardly yield

$$\begin{array}{rcl} A(\nu)x^{(k+1)}-b^{(k)}(\nu) & = & (\nu I+A)x^{(k+1)}-(\nu x^{(k)}+b) \\ & = & \nu(x^{(k+1)}-x^{(k)})-r^{(k+1)} \\ & = & \nu A^{-1}(r^{(k)}-r^{(k+1)})-r^{(k+1)} \\ & = & \nu A^{-1}r^{(k)}-(I+\nu A^{-1})r^{(k+1)} \\ & = & \nu A^{-1}r^{(k)}-A(\nu)A^{-1}r^{(k+1)}. \end{array}$$

Therefore,

$$\begin{array}{rcl} r^{(k+1)} & = & A(\nu)^{-1}A[\nu A^{-1}r^{(k)}+(b^{(k)}(\nu)-A(\nu)x^{(k+1)})] \\ & = & A(\nu)^{-1}[\nu r^{(k)}+A(b^{(k)}(\nu)-A(\nu)x^{(k+1)})] \end{array} \tag{7}$$

and

$$\begin{array}{rcl} x^*-x^{(k+1)} & = & A^{-1}r^{(k+1)} \\ & = & A(\nu)^{-1}[\nu(x^*-x^{(k)})+(b^{(k)}(\nu)-A(\nu)x^{(k+1)})]. \end{array} \tag{8}$$

Because

$$r^{(k)}=b-Ax^{(k)}=(\nu x^{(k)}+b)-(\nu x^{(k)}+Ax^{(k)})=b^{(k)}(\nu)-A(\nu)x^{(k)},$$

from (6), and by making use of Lemma 3.2 we have

$$\begin{array}{rcl} \|b^{(k)}(\nu)-A(\nu)x^{(k+1)}\|_2 & = & \|A(\nu)(A(\nu)^{-1}b^{(k)}(\nu)-x^{(k+1)})\|_2 \\ & \leq & \sqrt{\lambda_{\max}(A(\nu))}\|x^{(*,k)}-x^{(k+1)}\|_{A(\nu)} \\ & \leq & \sqrt{\lambda_{\max}(A(\nu))}\widetilde{\gamma}^{(m^{(k)})}(\nu)\|x^{(*,k)}-x^{(k)}\|_{A(\nu)} \\ & \leq & \sqrt{\dfrac{\lambda_{\max}(A(\nu))}{\lambda_{\min}(A(\nu))}}\widetilde{\gamma}^{(m^{(k)})}(\nu)\|A(\nu)(x^{(*,k)}-x^{(k)})\|_2 \\ & = & \sqrt{\kappa_2(A(\nu))}\widetilde{\gamma}^{(m^{(k)})}(\nu)\|b^{(k)}(\nu)-A(\nu)x^{(k)}\|_2 \\ & = & \sqrt{\kappa_2(A(\nu))}\widetilde{\gamma}^{(m^{(k)})}(\nu)\|r^{(k)}\|_2. \end{array} \tag{9}$$

Now, through taking $\|\cdot\|_2$-norms on both sides of (7), substituting (9) into the obtained inequality, and then applying Lemma 3.1 to the resulted estimate, we obtain

$$\begin{array}{rcl} \|r^{(k+1)}\|_2 & \leq & \nu\|A(\nu)^{-1}\|_2\|r^{(k)}\|_2+\|A(\nu)^{-1}A\|_2\|b^{(k)}(\nu)-A(\nu)x^{(k+1)}\|_2 \\ & \leq & [\nu\|A(\nu)^{-1}\|_2+\|A(\nu)^{-1}A\|_2\sqrt{\kappa_2(A(\nu))}\widetilde{\gamma}^{(m^{(k)})}(\nu)]\|r^{(k)}\|_2 \\ & = & \left(\dfrac{\nu}{\nu+\lambda_{\min}(A)}+\dfrac{\lambda_{\max}(A)}{\nu+\lambda_{\max}(A)}\sqrt{\kappa_2(A(\nu))}\widetilde{\gamma}^{(m^{(k)})}(\nu)\right)\|r^{(k)}\|_2 \\ & = & \widetilde{\delta}^{(m^{(k)})}(\nu)\|r^{(k)}\|_2 \leq \delta^{(m^{(k)})}(\nu)\|r^{(k)}\|_2. \end{array}$$

Similarly, through taking $\|\cdot\|_2$-norms on both sides of (8) we can obtain

$$\begin{array}{rcl} \|x^*-x^{(k+1)}\|_2 & \leq & \|A(\nu)^{-1}\|_2[\nu\|x^*-x^{(k)}\|_2+\|b^{(k)}(\nu)-A(\nu)x^{(k+1)}\|_2] \\ & \leq & \|A(\nu)^{-1}\|_2[\nu\|x^*-x^{(k)}\|_2+\sqrt{\kappa_2(A(\nu))}\widetilde{\gamma}^{(m^{(k)})}(\nu)\|r^{(k)}\|_2] \\ & \leq & \|A(\nu)^{-1}\|_2[\nu+\|A\|_2\sqrt{\kappa_2(A(\nu))}\widetilde{\gamma}^{(m^{(k)})}(\nu)]\|x^*-x^{(k)}\|_2 \\ & = & \widetilde{\theta}^{(m^{(k)})}(\nu)\|x^*-x^{(k)}\|_2 \leq \theta^{(m^{(k)})}(\nu)\|x^*-x^{(k)}\|_2. \end{array}$$

Theorem 3.1 presents upper bounds for the reduction rates of both relative residual and relative error of the RCG method. Obviously, the residual reduction ratio $\delta^{(m^{(k)})}(\nu)$ at the k-th step of the RCG method can be decomposed into two parts: one is $\delta_{\text{outer}}(\nu) = \frac{\nu}{\nu+\lambda_{\min}(A)}$ which is the contract factor of the outer iteration, and another is $\delta_{\text{inner}}^{(m^{(k)})}(\nu) = \frac{\lambda_{\max}(A)}{\nu+\lambda_{\max}(A)}\sqrt{\kappa_2(A(\nu))}\gamma^{(m^{(k)})}(\nu)$ which is the contract factor of the inner CG iteration. Similar observation holds for the error reduction ratio $\theta^{(m^{(k)})}(\nu)$. The size of the quantity $\delta_{\text{inner}}^{(m^{(k)})}(\nu)$ depends upon the number of the inner CG iteration steps $m^{(k)}$, and affects the convergence speed of the RCG method itself. In description of the RCG Method 2.1, this quantity is automatically governed by the exitting tolerance ϵ and the admissible number of iteration steps $\ell_{\max}$ of the inner CG iteration. Hence, good choices of both ϵ and $\ell_{\max}$ are crucial for guaranteeing the RCG method to achieve high computing efficiency in actual applications.

The identity (7) and the exitting criterion of the inner CG iteration in Method 2.1 provide with us one possible way to choose ϵ and $\ell_{\max}$. Note that the fastest residual reduction ratio of the RCG method is limited by the quantity $\delta_{\text{outer}}(\nu)$. Therefore, it is reasonable for us to choose ϵ and $\ell_{\max}$ such that the RCG method keeps a prescribed residual reduction ratio, say $\phi = \psi + (1-\psi)\delta_{\text{outer}}(\nu)$, where $\psi \in [0,1)$ is a constant.

From (7) we have

$$\begin{aligned}\|r^{(k+1)}\|_2 &\leq \nu\|A(\nu)^{-1}\|_2\|r^{(k)}\|_2 + \|A(\nu)^{-1}A\|_2\|b^{(k)}(\nu) - A(\nu)x^{(k+1)}\|_2 \\ &= \delta_{\text{outer}}(\nu)\|r^{(k)}\|_2 + \frac{\lambda_{\max}(A)}{\nu+\lambda_{\max}(A)}\|b^{(k)}(\nu) - A(\nu)x^{(k+1)}\|_2,\end{aligned}$$

and from the definition of Method 2.1 we have

$$\|b^{(k)}(\nu) - A(\nu)x^{(k+1)}\|_2 \leq \epsilon\|r^{(k)}\|_2. \tag{10}$$

Hence, it follows that

$$\|r^{(k+1)}\|_2 \leq \left(\delta_{\text{outer}}(\nu) + \frac{\lambda_{\max}(A)\epsilon}{\nu+\lambda_{\max}(A)}\right)\|r^{(k)}\|_2.$$

Now, if we take ϵ so small such that $\epsilon < \frac{\psi}{\kappa_2(A)}$ and

$$\delta_{\text{outer}}(\nu) + \frac{\lambda_{\max}(A)\epsilon}{\nu+\lambda_{\max}(A)} \leq \phi,$$

or equivalently,

$$\nu \leq \frac{\lambda_{\max}(A)(\psi-\epsilon)}{\kappa_2(A)\epsilon - \psi}, \tag{11}$$

then the residual of the RCG method in the Euclidean norm satisfies

$$\|r^{(k+1)}\|_2 \leq \phi\|r^{(k)}\|_2, \qquad k = 0,1,2,\ldots. \tag{12}$$

For the number $m^{(k)}$ of the inner CG iteration, from (9) and (10), and according to Lemma 3.1 we immediately know that (12) and (11) hold when

$$2\sqrt{\kappa_2(A(\nu))}\left(\frac{\sqrt{\kappa_2(A(\nu))}-1}{\sqrt{\kappa_2(A(\nu))}+1}\right)^{m^{(k)}} \leq \epsilon.$$

It follows straightforwardly from this restriction that

$$\ell_{\max} = \frac{\ln\left(\dfrac{\epsilon}{2\sqrt{\kappa_2(A(\nu))}}\right)}{\ln\left(\dfrac{\sqrt{\kappa_2(A(\nu))}-1}{\sqrt{\kappa_2(A(\nu))}+1}\right)}.$$

Moreover, considering that

$$\delta_{\text{inner}}^{(\ell_{\max})}(\nu) \leq \psi(1-\delta_{\text{outer}}(\nu)),$$

we immediately have $\epsilon \leq \frac{\psi}{2\kappa_2(A)}$.

The above analysis is summarized in the following theorem.

Theorem 3.2. *Let $A \in \mathbb{R}^{n\times n}$ be an SPD matrix and $\nu \geq 0$ a constant satisfying $\nu \leq \frac{\lambda_{\max}(A)(\psi-\epsilon)}{\lambda_{\max}(A)\epsilon-\psi}$. If the RCG method is started from an initial iterate $x^{(0)} \in \mathbb{R}^n$, and each inner CG iteration is exitted once its current residual reaches a reduction ratio $\epsilon \leq \frac{\psi}{2\kappa_2(A)}$ or its current iteration step reaches a maximum limit*

$$\ell_{\max} = \ln\left(\frac{\epsilon}{2\sqrt{\kappa_2(A(\nu))}}\right) / \ln\left(\frac{\sqrt{\kappa_2(A(\nu))}-1}{\sqrt{\kappa_2(A(\nu))}+1}\right),$$

where $\psi \in [0,1)$ is a prescribed constant, then the iterate sequence $\{x^{(k)}\}_{k=0}^{\infty}$ generated by the RCG method converges to the unique solution x^* of the system of linear equations (1). Moreover, it holds that

$$\|b-Ax^{(k+1)}\|_2 \leq \phi\|b-Ax^{(k)}\|_2, \qquad k=0,1,2,\ldots,$$

where $\phi = \psi + (1-\psi)\delta_{\text{outer}}(\nu)$.

Another route for determining the parameters ν and ϵ (through $\ell_{\max}$) is as follows. According to both convergence speed (see Theorem 3.1 (a)) and computational workload (see (5)), we can choose ν and $\ell_{\max}$ such that the computational efficiency $\mathcal{E}_{\text{rcg}}(\ell_{\max},\nu)$ corresponding to the RCG method is maximized, where

$$\mathcal{E}_{\text{rcg}}(\ell_{\max},\nu) = -\frac{\ln\left(\delta^{(\ell_{\max})}(\nu)\right)}{W_{\text{rcg}}(\ell_{\max})},$$

with

$$W_{\text{rcg}}(\ell_{\max}) = (2\omega n+6n+1)\ell_{\max} + (2\omega n+2n-1)$$

being defined by (5), and

$$\delta^{(\ell_{\max})}(\nu) = \frac{\nu}{\nu+\lambda_{\min}(A)} + \frac{2\lambda_{\max}(A)\sqrt{\kappa_2(A(\nu))}}{\nu+\lambda_{\max}(A)}\left(\frac{\sqrt{\kappa_2(A(\nu))}-1}{\sqrt{\kappa_2(A(\nu))}+1}\right)^{\ell_{\max}}.$$

Through simply denoting $\mathcal{E}_{\text{rcg}}(\ell_{\max},\nu)$ by $\mathcal{E}(\ell_{\max},\nu)$, and letting $s=\ell_{\max}$ and

$$\nu = \frac{\lambda_{\max}(A)-t^2\lambda_{\min}(A)}{t^2-1},$$

we can rewrite $\mathcal{E}_{\text{rcg}}(\ell_{\max},\nu)$ in the variables s and t as

$$\mathcal{E}(s,t) = -\frac{\ln\delta(s,t)}{W(s)},$$

where

$$\delta(s,t) = \frac{1}{\kappa-1}\left(\kappa - t^2 + \frac{2\kappa(t-1)^{s+1}}{t(t+1)^{s-1}}\right), \quad W(s) = c(n)s + d(n),$$

with

$$\kappa = \kappa_2(A), \quad c(n) = 2\omega n+6n+1 \quad \text{and} \quad d(n) = 2\omega n+2n-1.$$

Now, straightforwardly computing the partial derivatives of the function $\mathcal{E}(s,t)$ with respect to s and t, respectively, yields that a maximum point of $\mathcal{E}(s,t)$ is a solution of the following system of nonlinear equations:

$$\begin{cases} \widetilde{g}_1(s,t) &=& W(s)\dfrac{\partial\delta(s,t)}{\partial s} - c(n)\delta(s,t)\ln(\delta(s,t)) = 0,\\ \widetilde{g}_2(s,t) &=& \dfrac{\partial\delta(s,t)}{\partial t} = 0, \end{cases}$$

which is equivalent to

$$\begin{cases} g_1(s,t) &=& 2W(s)\dfrac{(\kappa-1)\delta(s,t)+t^2-\kappa}{(t+1)^2}\ln\left(\dfrac{t-1}{t+1}\right)\\ && -c(n)(\kappa-1)\delta(s,t)\ln(\delta(s,t)) = 0,\\ g_2(s,t) &=& \kappa(t^2+st(t+1)^2+1)\left(\dfrac{t-1}{t+1}\right)^s - t^3 = 0. \end{cases} \tag{13}$$

We can verify that this system of nonlinear equations has at least one positive solution, which could be solved by the classical Newton method [10].

Theorem 3.3. *Let $A \in \mathbb{R}^{n\times n}$ be an SPD matrix, with at most ω nonzeros on each of its rows. Then the best possible number $\ell^*_{\max}$ of the inner CG iteration steps and the best possible parameter ν^* of the matrix transformation are determined by the smallest positive root of the system of nonlinear equations (13), where*

$$\ell^*_{\max} = \frac{\ln\left(\dfrac{\psi t^*}{2\kappa}\right)}{\ln\left(\dfrac{t^*-1}{t^*+1}\right)} \quad \textit{and} \quad \nu^* = \frac{(\kappa_2(A)-(t^*)^2)\lambda_{\min}(A)}{(t^*)^2-1}.$$

In this case, a lower bound of ϵ is given by

$$\epsilon^* = 2t^*\left(\frac{t^*-1}{t^*+1}\right)^{\ell^*_{\max}}.$$

Therefore, for a prescribed constant $\psi \in [0,1)$, if the RCG method is started from an initial iterate $x^{(0)} \in \mathbb{R}^n$, and each inner CG iteration is exitted once its current residual reaches a reduction ratio $\epsilon = \min\left\{\frac{\psi}{2\kappa_2(A)}, \epsilon^*\right\}$ or its current iteration steps reaches a maximum limit $\ell_{\max} = \max\left\{\frac{\ln\left(\frac{\epsilon^*}{2t^*}\right)}{\ln\left(\frac{t^*-1}{t^*+1}\right)}, \ell^*_{\max}\right\}$, then the iterate sequence $\{x^{(k)}\}_{k=0}^{\infty}$ generated by the RCG method converges to the unique solution x^* of the system of linear equations (1), and it holds that

$$\|b - Ax^{(k+1)}\|_2 \le \phi\|b - Ax^{(k)}\|_2, \qquad k = 0,1,2,\ldots,$$

where $\phi = \psi + (1-\psi)\delta_{\mathrm{outer}}(\nu^*)$.

Theorem 3.3 presents an intuitive understanding and theoretical illustration about iteration parameters in the RCG method. However, since an approximate solution of the system of nonlinear equations (13) and the extreme eigenvalues of the matrix $A \in \mathbb{R}^{n\times n}$ are not trivially obtainable, this theorem has little applicability in actual computations.

4. The Preconditioned RCG Method

To further reduce the condition number $\kappa_2(A(\nu))$, and hence improve the performance of the RCG method, we can precondition the system of linear equations (4) by an SPD matrix $M(\nu) \in \mathbb{R}^{n\times n}$, termed as the preconditioner, that is close to the matrix $A(\nu) \in \mathbb{R}^{n\times n}$ and may be obtained by the IC factorization [12, 15], then the eigenvalues of the matrix $(M(\nu)^{-1}A(\nu))$ will be clustered near one, and by Theorem 3.1, the RCG method that employs the inner CG iteration to the preconditioned system of linear equations

$$M(\nu)^{-1}A(\nu)x = M(\nu)^{-1}(\nu x^{(k)} + b)$$

at the k-th outer iterate will have considerably fast convergence rate. This motivates the following *preconditioned regularized conjugate gradient* (PRCG) method.

Method 4.1. (The PRCG method).

```
 1. Input the largest admissible number of iteration steps $k_{\max}$ and the stopping tolerance $\varepsilon$ of
    the RCG method
 2. Input the largest admissible number of iteration steps $\ell_{\max}$ and the stopping tolerance $\epsilon$ of
    the inner CG iteration
 3. Input the starting vector $x$ and the iteration parameter $\nu$
 4. Set $k := 0$
 5. Compute $r = b - Ax$, $\rho^{(0)} = \|r\|_2^2$, and $\vartheta^{(0)} = \sqrt{\rho^{(0)}}$
 6. If $\vartheta^{(0)} \le \varepsilon\|b\|_2$ and $k \ge k_{\max}$, then GoTo 22
 7.    Set $\ell := 1$ and $y := x$
 8.    Do While $\vartheta^{(\ell-1)} > \epsilon\vartheta^{(\ell)}$ and $\ell < \ell_{\max}$
 9.       Solve $M(\nu)z = r$
10.       Compute $\tau^{(\ell-1)} = z^T r$
11.       If $\ell = 1$ then Set $\beta := 0$ and $p := z$, Else
             Compute $\beta = \tau^{(\ell-1)}/\tau^{(\ell-2)}$ and $p = z + \beta p$
12.       Compute $w = \nu p + Ap$
```

```
13.         Compute $\alpha = \tau^{(\ell-1)}/p^T w$
14.         Compute $y = y + \alpha p$
15.         Compute $r = r - \alpha w$
16.         Compute $\rho^{(\ell)} = \|r\|_2^2$ and $\vartheta^{(\ell)} = \sqrt{\rho^{(\ell)}}$
17.         Set $\ell := \ell + 1$
18.     EndDo
19.     Set $x := y$
20.     Set $k := k + 1$
21. GoTo 5
22. Continue
23. EndDo
```

The input of the PRCG method is the same as that for the RCG method and the routine to solve a linear system with the preconditioner $M(\nu)$ as its coefficient matrix. Aside from the preconditioner, the arguments of the PRCG method are the same as those to the RCG method. The cost of the PRCG method is identical to the RCG method with the addition of the application of the preconditioner $M(\nu)$ in line 9 and the additional inner product required to compute $\tau^{(\ell-1)}$ in line 10.

Analogously to Theorem 3.1, a precise estimate about the convergence speed of the PRCG method is described by the following theorem.

Theorem 4.1. *Let $A \in \mathbb{R}^{n\times n}$ be an SPD matrix, $\nu \geq 0$ a constant, and $M(\nu) \in \mathbb{R}^{n\times n}$ a preconditioner of the matrix $A(\nu) \in \mathbb{R}^{n\times n}$. If the PRCG method is started from an initial iterate $x^{(0)} \in \mathbb{R}^n$, and applied $m^{(k)}$ steps of the preconditioned CG iteration to get the next approximation $x^{(k+1)}$ to the solution x^* of the system of linear equations (1), then it holds that:*

(a) $\|b - Ax^{(k+1)}\|_2 \leq \overline{\delta}^{(m^{(k)})}(\nu)\|b - Ax^{(k)}\|_2 \leq \widehat{\delta}^{(m^{(k)})}(\nu)\|b - Ax^{(k)}\|_2$;

(b) $\|x^{(k+1)} - x^*\|_2 \leq \overline{\theta}^{(m^{(k)})}(\nu)\|x^{(k)} - x^*\|_2 \leq \widehat{\theta}^{(m^{(k)})}(\nu)\|x^{(k)} - x^*\|_2$,

where

$$\begin{aligned}
\overline{\gamma}^{(k)}(\nu) &= \min_{p^{(k)}\in\mathcal{P}^{(k)}} \max_{z\in\sigma(M(\nu)^{-1}A(\nu))} |p^{(k)}(z)|,\\
\overline{\delta}^{(k)}(\nu) &= \frac{\nu}{\nu+\lambda_{\min}(A)} + \frac{\lambda_{\max}(A)}{\nu+\lambda_{\max}(A)}\kappa_2(M(\nu))\sqrt{\kappa_2(M(\nu)^{-1}A(\nu))}\,\overline{\gamma}^{(k)}(\nu),\\
\overline{\theta}^{(k)}(\nu) &= \frac{\nu}{\nu+\lambda_{\min}(A)} + \frac{\lambda_{\max}(A)}{\nu+\lambda_{\min}(A)}\kappa_2(M(\nu))\sqrt{\kappa_2(M(\nu)^{-1}A(\nu))}\,\overline{\gamma}^{(k)}(\nu),\\
\widehat{\gamma}^{(k)}(\nu) &= 2\left(\frac{\sqrt{\kappa_2(M(\nu)^{-1}A(\nu))}-1}{\sqrt{\kappa_2(M(\nu)^{-1}A(\nu))}+1}\right)^k,\\
\widehat{\delta}^{(k)}(\nu) &= \frac{\nu}{\nu+\lambda_{\min}(A)} + \frac{\lambda_{\max}(A)}{\nu+\lambda_{\max}(A)}\kappa_2(M(\nu))\sqrt{\kappa_2(M(\nu)^{-1}A(\nu))}\,\widehat{\gamma}^{(k)}(\nu),\\
\widehat{\theta}^{(k)}(\nu) &= \frac{\nu}{\nu+\lambda_{\min}(A)} + \frac{\lambda_{\max}(A)}{\nu+\lambda_{\min}(A)}\kappa_2(M(\nu))\sqrt{\kappa_2(M(\nu)^{-1}A(\nu))}\,\widehat{\gamma}^{(k)}(\nu).
\end{aligned}$$

5. Numerical Results

In this section, we compute three examples of the system of linear equations (1) to test numerical behaviours of the RCG and PRCG methods. Each iterate is started from an initial vector having all entries equal to zero, and terminated once the current iterate attains a prescribed stopping tolerance ε or a prescribed largest iteration step $k_{\max}$. The new methods are compared with the CG method and the SGS (*Symmetric Gauss-Seidel*) method or SSOR method of optimal relaxation factor (SSOR(ω_{opt}) method), from aspects of both number of the iteration steps (denoted by "IT") and precision of the approximated solutions (denoted by "ERROR" and defined by ERROR $= \frac{\|x^{(it)}-x^*\|_2}{\|x^*\|_2}$).

Our first example is the system of linear equations (1) of which the coefficient matrix is the Hilbert matrix

$$A = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n-1} & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \cdots & \frac{1}{n} & \frac{1}{n+1} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \cdots & \frac{1}{n+1} & \frac{1}{n+2} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{1}{n-1} & \frac{1}{n} & \frac{1}{n+1} & \cdots & \frac{1}{2n-3} & \frac{1}{2n-2} \\ \frac{1}{n} & \frac{1}{n+1} & \frac{1}{n+2} & \cdots & \frac{1}{2n-2} & \frac{1}{2n-1} \end{pmatrix} \in \mathbb{R}^{n\times n}, \tag{14}$$

and the RHS vector is chosen such that the exact solution x^* having all entries equal to 1. It is well known that this matrix is very ill-conditioned when n is slightly large. The total numbers of iteration steps (IT) and the relative error of the approximated solutions (ERROR) corresponding to the RCG, CG and SGS methods are listed in Table 5.1. Evidently, the RCG method converges very faster and produces a much more accurate solution to the system of linear equations than the other two methods.

Table 5.1. Total iteration numbers and relative errors for linear system (14).

n	RCG method				CG method		SGS method	
	$\nu = 0.0006$		$\nu = 0.001$					
	IT	ERROR	IT	ERROR	IT	ERROR	IT	ERROR
400	56	5.67E-03	85	5.39E-03	149	2.98E-01	$\geq$ 2E+05	–
600	69	4.98E-03	96	4.89E-03	148	2.41E-01	$\geq$ 2E+05	–
800	67	5.66E-03	102	5.25E-03	894	7.55E-01	$\geq$ 2E+05	–
1000	70	5.59E-03	103	5.24E-03	647	4.94E-01	$\geq$ 2E+05	–

The iteration parameters are: $\varepsilon = 10^{-6}$, $\epsilon = 0.01$, $\psi = 0.1$, $k_{\max} = 1000$ and $\ell_{\max} = 20$.

Our second example is the system of linear equations (1) of which the coefficient matrix $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ is given by

$$a_{ij} = \begin{cases} (n+1-i)^2 - \dfrac{12i^2}{n(n+1)(2n+1)}\left(\dfrac{3n(n+1)}{2n+1} - 2i\right), & \text{for } i = j, \\ -\dfrac{12ij}{n(n+1)(2n+1)}\left(\dfrac{3n(n+1)}{2n+1} - i - j\right), & \text{for } i \neq j, \end{cases} \tag{15}$$

and the RHS vector is chosen such that the exact solution x^* having all entries equal to 1. The spectrum of this matrix is $\sigma(A) = \{1, 2^2, \ldots, n^2\}$ and its condition number is $\kappa_2(A) = n^2$. Therefore, it is very ill-conditioned when n is quite large. The total number of iteration steps (IT) and the relative error of the approximated solution (ERROR) corresponding to the RCG and CG methods are listed in Table 5.2. Again, it is obvious that the RCG method has much better convergence property than the CG method in both steps of the iterates and precisions of the solutions.

Table 5.2. Total iteration numbers and relative errors for linear system (15).

n	RCG method						CG method	
	$\nu = 0.05$		$\nu = 0.1$		$\nu = 0.3$			
	IT	ERROR	IT	ERROR	IT	ERROR	IT	ERROR
300	4012	5.81E-06	4312	1.23E-05	4537	2.81E-05	4937	1.83E-05
400	7037	4.03E-05	5412	5.39E-06	5962	4.01E-05	7162	3.55E-05
500	6236	1.29E-04	6815	3.79E-05	8362	1.17E-04	6437	3.00E-05
600	8237	1.88E-04	9030	7.79E-06	8962	3.46E-05	9081	1.47E-04

The iteration parameters are: $\varepsilon = 10^{-7}$, $\epsilon = 0.01$, $\psi = 0.1$, $k_{\max} = 1000$ and $\ell_{max} = 25$.

At last, we consider the Poisson's equation

$$\begin{cases} \dfrac{\partial^2 u}{\partial^2 t_1^2} + \dfrac{\partial^2 u}{\partial^2 t_2^2} = \widetilde{f}(t_1, t_2), & \text{in} \quad \Omega, \\ u(t_1, t_2) = 0, & \text{on} \quad \partial\Omega, \end{cases} \tag{16}$$

where $\Omega = (0,1) \times (0,1)$ is a unit square in $\mathbb{R}^2$, $\partial\Omega$ is the boundary of the domain Ω, and $\widetilde{f} : \mathbb{R}^2 \to \mathbb{R}^2$ is a given function of variables t_1 and t_2. We discretize the unit square with mesh spacing h and use $u_{i,j}$ to denote an approximation to the solution of (16) at the grid point (ih, jh). Then, by approximating the derivatives of (16) by the usual second-order difference approximations we obtain the system of linear equations

$$\begin{cases} u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j} = h^2 \widetilde{f}_{i,j}, & i,j = 1,2,\ldots,N, \\ u_{i,0} = u_{i,1} = u_{0,j} = u_{1,j} = 0, & \end{cases}$$

where $(N+1)h = 1$.

This system can be written in the form (1) by letting

$$x^T = (u_{1,1}, \ldots, u_{1,N}, u_{2,1}, \ldots, u_{2,N}, \ldots, u_{N,1}, \ldots, u_{N,N}),$$

with the coefficient matrix

$$A = \begin{pmatrix} T & -I & & & \\ -I & T & -I & & \\ & \ddots & \ddots & \ddots & \\ & & -I & T & -I \\ & & & -I & T \end{pmatrix} \in \mathbb{R}^{n\times n}, \quad T = \begin{pmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 4 & -1 \\ & & & -1 & 4 \end{pmatrix} \in \mathbb{R}^{N\times N}, \quad (17)$$

and the RHS vector $b \in \mathbb{R}^n$ consisting of the quantities $-h^2\widetilde{f}_{i,j}$ in the proper positions, where $n = N^2$.

The spectrum of the matrix A is given by

$$4 + 2(\cos(k\pi h) + \cos(j\pi h)), \quad k, j = 1, 2, \ldots, N,$$

and the largest and smallest eigenvalues are, respectively,

$$\lambda_{\max}(A) = 4(1 + \cos(\pi h)), \qquad \lambda_{\min}(A) = 4(1 - \cos(\pi h)).$$

Therefore, the condition number of the matrix A is

$$\kappa_2(A) = \frac{1+\cos(\pi h)}{1-\cos(\pi h)} \approx \frac{4}{\pi^2}h^{-2} + \mathcal{O}(1) = \frac{4}{\pi^2}(N+1)^2 + \mathcal{O}(1) \approx \frac{4}{\pi^2}n + \mathcal{O}(1).$$

In our computations, we choose $\widetilde{f} : \mathbb{R}^2 \to \mathbb{R}^2$ such that the exact solution x^* of the system of linear equations (17) having all entries equal to 1. The IC factorization with no additional fill-in (IC(0)) [1, 12, 11, 2, 15] is used to precondition both RCG and CG methods. The resulted PRCG method is compared with the resulted PCG method and the SSOR(ω_{opt}) method, as well as the RCG method.

For different ν and h, the total CPU times (TIMING) and the relative error (ERROR) corresponding to the PRCG, PCG and the SSOR(ω_{opt}) methods are listed in Table 5.3, and those corresponding to the RCG method are listed in Table 5.4. Here, each TIMING and each ERROR in Tables 5.3 and 5.4 is, respectively, an average of 30 repeated running timings and errors of the same program.

Table 5.3. CPU times and relative errors for linear system (17).

h^{-1}	PRCG method				PCG method		SSOR(ω_{opt}) method	
	$\nu = 0.095 \times h^2$		$\nu = 0.0006$				$\omega_{opt} = \frac{2}{1+\sin(\pi h)}$	
	TIMING	ERROR	TIMING	ERROR	TIMING	ERROR	TIMING	ERROR
14	0.1106	6.41E-07	0.0740	1.95E-06	0.0970	2.38E-06	0.2348	1.07E-07
16	0.1499	2.02E-06	0.0992	3.02E-06	0.1626	2.63E-06	–	–
24	0.7177	2.11E-06	0.6406	2.78E-06	0.7568	2.52E-06	–	–
32	2.6791	4.22E-06	2.0184	4.33E-06	2.7011	3.87E-06	–	–
40	6.0413	5.65E-06	5.6947	5.21E-06	6.0635	5.77E-06	–	–
48	13.1693	7.72E-06	10.2591	7.07E-06	14.1659	6.19E-06	–	–
56	29.1890	7.35E-06	22.8013	6.59E-06	30.1533	7.15E-06	–	–
64	69.3848	7.72E-06	41.7172	7.65E-06	74.6402	6.58E-06	–	–

The iteration parameters are: $\varepsilon = 10^{-6}$, $\epsilon = 0.01$, $\psi = 0.1$, $k_{\max} = 1000$ and $\ell_{\max} = 30$.

From Table 5.3, we clearly see that the PRCG method outperform the PCG and SSOR(ω_{opt}) methods in the senses of both CPU times and relative errors. Through comparing Table 5.3 with Table 5.4, we observe that when the mesh spacing h is not too small (e.g., $h = \frac{1}{48}$, $\frac{1}{40}$, or more), the RCG method with a suitably chosen ν outperforms both PRCG and PCG methods; and when the mesh spacing h is quite small (e.g., $h = \frac{1}{56}, \frac{1}{64}$ or less), the PRCG method outperforms the RCG method for a small parameter ν (e.g., $\nu = 0.095h^2$ or $\nu = 0.0006$), however, this situation could be reversed for a reasonably large parameter ν (e.g., $\nu = 0.006$). Therefore, the new RCG and PRCG methods are more efficient and robust than both the CG method and the optimal SSOR method.

Table 5.4. CPU times and relative errors of the RCG method for linear system (17).

h^{-1}	$\nu = 0.095 \times h^2$		$\nu = 0.0006$		$\nu = 0.006$		$\nu = 0.06$	
	TIMING	ERROR	TIMING	ERROR	TIMING	ERROR	TIMING	ERROR
14	0.0196	2.042E-07	0.0199	2.44E-07	0.0270	4.61E-07	0.0432	2.05E-06
16	0.0391	5.59E-07	0.0325	3.27E-07	0.0442	1.63E-07	0.0742	3.12E-06
24	0.1588	3.15E-07	0.1940	1.97E-07	0.2531	1.01E-06	0.4281	5.66E-06
32	0.6096	4.30E-07	0.6033	2.42E-07	0.7986	1.64E-06	1.3799	7.62E-06
40	1.9803	1.53E-07	1.6684	9.77E-08	2.1876	2.46E-07	3.6309	9.42E-06
48	8.9936	1.28E-07	4.5574	2.19E-07	4.4441	5.99E-07	8.1461	9.42E-06
56	36.2691	7.03E-08	20.2750	1.13E-07	16.5696	1.81E-07	19.5382	2.56E-06
64	444.1473	2.82E-07	387.4398	1.44E-07	18.7799	1.12E-07	45.6351	2.87E-07

The iteration parameters are: $\varepsilon = 10^{-6}$, $\epsilon = 0.01$, $\psi = 0.1$, $k_{\max} = 1000$ and $\ell_{\max} = 30$.

References

[1] O. Axelsson, A generalized SSOR method, *BIT,* **12** (1972), 443-467.

[2] O. Axelsson, Iterative Solution Methods, Cambridge University Press, Cambridge, 1994.

[3] Z.Z. Bai, A class of modified block SSOR preconditioners for symmetric positive definite systems of linear equations, *Advances in Computational Mathematics,* **10** (1999), 169-186.

[4] J.W. Daniel, The conjugate gradient method for linear and nonlinear operator equations, *SIAM Journal on Numerical Analysis,* **4** (1967), 10-26.

[5] R.S. Dembo, S.C. Eisenstat, T. Steihaug, Inexact Newton methods, *SIAM Journal on Numerical Analysis,* **19** (1982), 400-408.

[6] G.H. Golub, C.F. Van Loan, Matrix Computations, 3rd Edition, The Johns Hopkins University Press, Baltimore and London, 1996.

[7] G.H. Golub, M.L. Overton, Convergence of a two-stage Richardson iterative procedure for solving systems of linear equations, In: Numerical Analysis (Proceedings of the Ninth Biennial Conference, Dundee, Scotland, 1981)(G.A. Watson, ed.), Lecture Notes Math. 912, Springer, New York Heidelberg Berlin, pp128-139.

[8] G.H. Golub, M.L. Overton, The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems, *Numerische Mathematik,* **53** (1988), 571-593.

[9] M.R. Hestenes, E.L. Stiefel, Methods of conjugate gradients for solving linear systems, *Journal of Research of the National Bureau Standards, Section B,* **49** (1952), 409-436.

[10] C.T. Kelley, Iterative Methods for Linear and Nonlinear Equations, SIAM, Philadelphia, 1995.

[11] T.A. Manteuffel, An incomplete factorization technique for positive definite linear systems, *Mathematics of Computations,* **34** (1980), 473-497.

[12] J.A. Meijerink, H.A. van der Vorst, An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix, *Mathematics of Computations,* **31** (1977), 148-162.

[13] N.K. Nichols, On the convergence of two-stage iterative processes for solving linear equations, *SIAM Journal on Numerical Analysis,* **10** (1973), 460-469.

[14] N.K. Nichols, On the local convergence of certain two step iterative procedures, *Numerische Mathematik,* **24** (1975), 95-101.

[15] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing Company, Boston, 1996.

[16] R.S. Varga, Matrix Iterative Analysis, Prentice Hall, Englewood Cliffs, N.J., 1962.