Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **4**, No. 4, pp. 473-482

DOI: 10.4208/aamm.10-m1162
August 2012

An Acceleration Method for Stationary Iterative Solution to Linear System of Equations

Qun Lin[1,*] and Wujian Peng[2]

[1] *Academy of Math and System Sciences, Chinese Academy of Sciences, Institute Computational Mathematics, Beijing 100190, China*
[2] *School of Mathematics and Information Sciences, Zhaoqing University, Zhaoqing, Guangdong 526061, China*

Received 24 March 2011; Accepted (in revised version) 5 December 2011
Available online 10 July 2012

Abstract. An acceleration scheme based on stationary iterative methods is presented for solving linear system of equations. Unlike Chebyshev semi-iterative method which requires accurate estimation of the bounds for iterative matrix eigenvalues, we use a wide range of Chebyshev-like polynomials for the accelerating process without estimating the bounds of the iterative matrix. A detailed error analysis is presented and convergence rates are obtained. Numerical experiments are carried out and comparisons with classical Jacobi and Chebyshev semi-iterative methods are provided.

AMS subject classifications: 65F10, 15A06

Key words: Iterative method, error analysis, recurrence.

1 Introduction

Linear algebraic system arises from almost every field of mathematical applications, so the problem of solving linear algebraic system is of great importance. Numerous methods have been presented for this purpose. In general, all the existing methods [1, 3, 4, 6, 9] fall into two categories: direct and iterative methods. In direct methods, one tries to decompose the coefficient matrix A in the regular system

$$Ax = b, \tag{1.1}$$

into some product form; for example in Gaussian elimination method, the coefficient

*Corresponding author.
URL: http://lsec.cc.ac.cn/∼linq/english_version.html
Email: linq@lsec.cc.ac.cn (Q. Lin), douglas_peng@yahoo.com (W. J. Peng)

matrix A is factored as $A = LU$, where L and U are lower and upper triangular matrices, we then solve the equivalent two simple systems:

$$Ly = b, \tag{1.2a}$$
$$Ux = y, \tag{1.2b}$$

which can be solved by using backward and forward substitution methods. By iterative method, one looks for a sequence of approximating solutions $\{x^{(k)}\}$ while the coefficients matrix A is unchanged or is just split by some simple procedures.

A large family of iteration methods for solving (1.1) take the form

$$Mx^{(k+1)} = Nx^{(k)} + b, \tag{1.3}$$

where

$$A = M - N, \tag{1.4}$$

is a splitting of the matrix A. For instance, the well-known Jacobi iteration is a member of this family with

$$M = D \quad \text{and} \quad N = -(L + U), \tag{1.5}$$

where D is the diagonal matrix with its entries exactly the same as those in A, and L and U are the lower and upper triangular matrices extracted directly from A:

$$L = (l_{ij})_{n*n} \qquad \text{with } l_{ij} = \begin{cases} 0, & i \le j, \\ a_{ij}, & i > j, \end{cases} \tag{1.6a}$$

$$U = (u_{ij})_{n*n} \qquad \text{with } u_{ij} = \begin{cases} 0, & i \ge j, \\ a_{ij}, & i < j. \end{cases} \tag{1.6b}$$

Another example is the Gauss-Seidel iteration in which M, N are constructed as follows

$$M = D + L, \qquad N = -U. \tag{1.7}$$

The following theorem guarantees the convergence of the iteration methods defined by (1.3).

Theorem 1.1. *Suppose A, M are invertible, and the spectral radius of matrix $M^{-1}N$ is less than 1, then the iteration sequence $\{x^{(k)}\}_{k=1}^{\infty}$ produced by* (1.3) *will converge to the solution $x = A^{-1}b$ of the linear system* (1.1) *for any starting vector $x^{(0)}$.*

The above iteration methods may be attractive because of its simplicity, however the convergence of these so-called stationary methods are usually not satisfactory. Therefore some acceleration scheme is usually applied to improve the convergence of these methods. A well-known acceleration method is the Chebyshev semi-iterative method, which is discussed in [8] as well as in [7]. The following is an introduction to this method which is a variation of that in [4].

2 The Chebyshev semi-iterative method

From now on we assume that the iterative matrix $G \equiv M^{-1}N$ is symmetric and its eigenvalues satisfy:

$$-1 < \alpha \leq \lambda_n \leq \cdots \leq \lambda_1 \leq \beta < 1. \tag{2.1}$$

Suppose $x^{(1)}, \cdots, x^{(k)}$ have been generated via the iteration (1.3), the Chebyshev semi-iterative method seeks a set of coefficients $\{a_{k,i}\}_{i=0}^{k}$ such that

$$y^{(k)} = \sum_{i=0}^{k} a_{k,i} x^{(i)}, \tag{2.2}$$

represents an improvement over $\{x^{(i)}\}_0^k$. Notice that if

$$x^{(0)} = \cdots = x^{(k)} = x,$$

then it is reasonable to assume that $y^{(k)} = x$. Hence the following constraint

$$\sum_{i=0}^{k} a_{k,i} = 1, \tag{2.3}$$

should be imposed on the coefficients.

Let x be the exact solution of the linear system. Note that

$$e^{(i)} \equiv x^{(i)} - x = G^i(x^{(0)} - x). \tag{2.4}$$

This yields

$$y^{(k)} - x = \sum_{i=0}^{k} a_{k,i}(x^{(i)} - x) = \sum_{i=0}^{k} a_{k,i} G^i (x^{(0)} - x) = \sum_{i=0}^{k} a_{k,i} G^i e^{(0)}. \tag{2.5}$$

Consequently,

$$\|y^{(k)} - x\|_2 \leq \|p_k(G)\|_2 \|e^{(0)}\|_2, \tag{2.6}$$

where

$$p_k(z) = \sum_{i=0}^{k} a_{k,i} z^i.$$

Since

$$\|p_k(G)\|_2 = \max_{\lambda_i \in \lambda(G)} |p_k(\lambda_i)| \leq \max_{\alpha \leq \lambda \leq \beta} |p_k(\lambda)|, \tag{2.7}$$

we need to create a polynomial $p_k(z)$ which is small on $[\alpha, \beta]$ subject to the constraint $p_k(1) = 1$ so that the norm of $p_k(G)$ as small as possible.

Note that when k is getting larger and larger, the summation in (2.2) would be inconvenient or even impossible; fortunately Chebyshev polynomials satisfy a three-term recurrence relation,

$$c_i(z) = 2z c_{i-1}(z) - c_{i-2}(z), \quad i \geq 2, \tag{2.8}$$

where

$$c_0(z) = 1 \quad \text{and} \quad c_1(z) = z.$$

If we define a polynomial

$$p_k(t) = \frac{c_k(\zeta(t))}{c_k(\mu)} \quad \text{or} \quad c_k(z) = p_k(t) \cdot c_k(\mu), \tag{2.9}$$

where

$$z \equiv \zeta(t) = -1 + 2 \cdot \frac{t-\alpha}{\beta-\alpha} \quad \text{and} \quad \mu = \zeta(1),$$

then $p_k(1) = 1$ and it tends to be very small on $[\alpha, \beta]$. Combining (2.8) and (2.9), we have

$$p_{k+1}(t) = \omega_1 t p_k(t) + \omega_2 p_k(t) + \omega_3 p_{k-1}(t), \tag{2.10}$$

where

$$\omega_1(k) = \frac{2\tau_1 c_k(\mu)}{c_{k+1}(\mu)}, \quad \omega_2(k) = \frac{2\tau_2 c_k(\mu)}{c_{k+1}(\mu)}, \quad \omega_3(k) = -\frac{c_{k-1}(\mu)}{c_{k+1}(\mu)},$$

with

$$\tau_1 = \frac{2}{(\beta-\alpha)} \quad \text{and} \quad \tau_2 = \frac{(\alpha+\beta)}{(\alpha-\beta)}.$$

Let

$$f(t) = t p_k(t) = \sum_{j=0}^{k} a_{k,j} t^{j+1}. \tag{2.11}$$

From (1.3), one has

$$x^{(k+1)} = G x^{(k)} + M^{-1} b. \tag{2.12}$$

Thus replacing t^j by $x^{(j)}$ in $f(t)$ yields a vector in the form $Gy^{(k)} + M^{-1}b$. Consequently, a three-term recurrence among the $\{y^{(k)}\}$ is developed:

$$y^{(k+1)} = (\omega_1(k) G + \omega_2(k) I) y^{(k)} + \omega_3(k) y^{(k-1)} + \omega_1(k) M^{-1} b, \tag{2.13}$$

where $y^{(0)} = x^{(0)}$, $y^{(1)} = x^{(1)}$, $G = M^{-1}N$ and I is the identity matrix. From (2.6) it is easy to see that

$$\|y^{(k)} - x\|_2 \leq \frac{\|x - x^{(0)}\|_2}{|c_k(\mu)|}. \tag{2.14}$$

So the larger the μ is , the faster the series $\{y^{(k)}\}$ converge to the exact solution x.

Remark 2.1. 1. Exact lower and upper bounds of α and β are difficult to obtain except in a few well structured problems; It is observed that the convergence behavior of Chebyshev accelerating scheme is very sensitive to the accurate estimation of the extreme eigenvalues of the iterative matrix;

2. Chebyshev polynomial has the so-called min-max property in $[-1, 1]$, but it grows much faster than other polynomials of the same order on points outside of the

interval $[-1,1]$, this may cause the accumulating error grows very rapidly for values between -1 and α, β and 1 , thus it is not necessarily a good idea to use higher order Chebyshev polynomials in the accelerating scheme;

3. Calculation of higher order polynomial function values always has the risk of excessive rounding-off errors, which in turn will bring unexpected consequence to the solutions in this case. As a matter of fact, our experiments show increased errors instead of improvements in relative error when the iteration number exceeds 10.

Based on the above observation, we present a so-called restarting scheme without using Chebyshev polynomials to work as the accelerating process for stationary iterative methods as stated in the sections followed.

3 A restarted iterative scheme

In this section we will present another acceleration method for the iteration family defined by (1.3). Our idea looks similar with Chebyshev semi-iterative method, however instead of using successive refinement upon each iteration, we do refinement once after a fixed number of iteration, and the improved approximation will be used again to get more accurate approximation. So the whole process takes two loops: the inner loop for basic iteration defined by (1.3) and the outer loop is imposed on the refinement.

3.1 Construction of iterative scheme

Let k be a fixed integer (usually $k \leq 12$ is good enough), $x^{(1)}$, $x^{(2)}$, $\cdots$, $x^{(k)}$ are the approximation to the solution x of system (1.1) generated by (1.3). The goal is to seek a set of parameter $\{a_i\}_{i=0}^{k}$ such that the combination

$$y^{(i)} = \sum_{j=0}^{k} a_j x^{(j)}, \tag{3.1}$$

is a better improvement over $x^{(i)}$, $(i = 1,2,\cdots,k)$. For the same reason as in Chebyshev semi-iterative method, we impose a constraint $\sum_{i=0}^{k} a_i = 1$, so that when $x^{(0)} = x^{(1)} = \cdots = x^{(k)} = x$ will give us

$$y^{(i)} = x. \tag{3.2}$$

To carry out the discussion of iterative scheme, we now introduce some "Chebyshev-like" polynomials.

Let $p_k(x)$ is any polynomial with degree k which satisfies the following condition:

1. $\max\limits_{x\in[-1,\alpha]} |p_k(x)| < \epsilon$, where $\alpha < 1$ and $\epsilon \ll 1$;
2. $p_k(1) = 1$;
3. $p_k(x)$ increases monotonically in $[\alpha, 1]$.

We can easily obtain such kind of polynomials by standard approximation techniques: First define a piecewise continuous function which vanishes on $[-1,\alpha]$ and is linear on $[\alpha,1]$, then use interpolation techniques to get a good approximating polynomial.

Now we construct our iterative method in the following:

Restarted Iterative Scheme:

Step 1 Construct a "Chebyshev-like" polynomial

$$p_k(x)=\sum_{i=0}^{k}a_i x^i,$$

where k is the degree of the $p_k(x)$.

Step 2 For i = 1 to k
do standard iteration (1.3)
end.

Step 3 Construct improved approximation

$$y^{(k)}=\sum_{i=0}^{k}a_i x^{(i)},$$

where a_i $(i=0,1,\cdots,k)$ are the coefficients of $p_k(x)$.

Step 4 Compute the relative error in L_2 norm for the residuals of $y^{(k)}$. If not satisfying, set $x^{(0)}$ as $y^{(j)}$ and repeat Step 2 to Step 4.

Apparently there are two loops in the above scheme: the inner loop in Step 2 and the outer loop which consists of Steps 2, 3 and 4. Unlike Chebyshev semi-iterative method, a successive refinement after each standard iteration (1.3) is not needed here, this eliminates the possibility of big accumulated rounding error; furthermore, the polynomials do not have to meet any recurrence relationship.

3.2 Error analysis

We will see in this section that the error formula is much more precise than what one can get from Chebyshev semi-iterative method. We assume that the matrix $G\equiv M^{-1}N$ is symmetric with its eigenvalues $\{\lambda_i\}_{i=1}^n$ satisfy (2.1).

Let

$$e^{(i)}\equiv x^{(i)}-x,\quad e_y^{(i)}\equiv y^{(i)}-x,\quad i=0,1,2,\cdots.$$

Let (λ_i,v_i) be the eigenpair of matrix $G\equiv M^{-1}N$. Note that $\{v_i\}_1^n$ form a basis of R^n and we can further assume that it is an orthonormal basis of R^n since G is symmetric. Suppose $e^{(0)}$ can be expressed under this basis as the following

$$e^{(0)}=\sum_{i=1}^{n}d_i v_i. \tag{3.3}$$

Then we have the following result.

Theorem 3.1. *Let $p_k(x)$ be a polynomial with degree k, (λ_i, v_i) be the eigenpair of G which is similar to a symmetric matrix, $\{d_i\}_1^n$ be defined in* (3.3)*. Then*

$$\|e_y^{(r)}\|^2 = \sum_{i=1}^{n}[d_i^2 p_k^{2r}(\lambda_i)], \quad r = 1, 2, \cdots, \tag{3.4}$$

Proof. Noting that

$$x^{(i)} - x = (M^{-1}N)^i(x^{(0)} - x) = G^i(x^{(0)} - x), \tag{3.5}$$

we have

$$y^{(r)} - x = \sum_{j=0}^{k} a_j(x^{(j)} - x) = \sum_{j=0}^{k} a_j G^j(x^{(0)} - x) = p_k(G)(x^{(0)} - x). \tag{3.6}$$

Thus

$$\|y^{(r)} - x\|_2 \leq \|p_k(G)\|_2 \|x^{(0)} - x)\|_2. \tag{3.7}$$

Note the definition of $e_y^{(i)}$ and $e^{(0)}$, (3.6) and (3.7) can be rewritten as

$$e_y^{(r)} = p_k(G)e_0, \quad \|e_y^{(r)}\| \leq \|p_k(G)\|_2 \|e_0\|. \tag{3.8}$$

As we have noted before, the error estimation (3.8) is too rough. Hence we need to explore the structure with more detail.

From (3.3) and (3.6), we have

$$\begin{aligned} e_y^{(r)} &= p_k(G)e_0 = p_k(G)\sum_{i=1}^{n} d_i v_i \\ &= \sum_i d_i p_k(G)v_i = \sum_i d_i \Big(\sum_j a_j \lambda_j^j \Big) v_i = \sum_i d_i p_k(\lambda_i) v_i, \end{aligned} \tag{3.9}$$

which gives that

$$\|e_0\|^2 = \sum_{i=1}^{n} d_i^2, \quad \|e_y^{(r)}\|^2 = \sum_{i=1}^{n} d_i^2 p_k^2(\lambda_i). \tag{3.10}$$

Now if we reset $x^{(0)}$ as the improved approximation $y^{(i)}$, then do the same iteration and refinement to get another improved approximation $y^{(r+1)}$, following the same analysis we get

$$\|e_y^{(r+1)}\|^2 = \sum_{i=1}^{n}[d_i^2 p_k^4(\lambda_i)], \tag{3.11}$$

where $d_i (i = 1, 2, \cdots)$ are the original values.

Generally if we repeat the above process r times, then the error estimate will have the following form

$$\|e_y^{(r)}\|^2 = \sum_{i=1}^{n} [d_i^2 p_k^{2r}(\lambda_i)], \quad r = 1, 2, \cdots. \tag{3.12}$$

This completes the proof. □

Remark 3.1. *One is easy to construct a polynomial $p_k(z)$ such that $|p_k(x)| < 10^{-1}$ if $|x| \leq 0.95$ but k does not need to be very high. In our experiments, we found that $k = 7$ is good enough.*

Remark 3.2. *From Remark 2.1 one can see that if the largest eigenvalue of G in magnitude is not greater than 0.95, then for each loop of the above restarted iteration scheme, the norm of the error would be reduced at least 90%. This means that it is possible for us to get a very good approximation to the exact solution after just a few iterations, which makes it a very good solver for large scale systems!*

From Remark 3.2 one can see that the above restarted iteration scheme is extremely efficient for those systems in whose

$$G \equiv M^{-1}N$$

has a spectral radius not greater than a number less than but close to 1.

4 Numerical experiments

In this section we examine the results of some numerical examples to verify the conclusion in Theorem 3.1. Note that from Remark 3.2 we conclude that if we choose $p_k(x)$ satisfying

$$|p_k(x)| \leq 0.1 \quad \text{for } |x| \leq 0.95,$$

then by the error formula (3.4) we see that the norm of $e_y^{(r)}$ will decrease at least 90%. The following examples show that restarted iterative method behaviors exactly like what is predicted. All experiments have been done with Matlab.

Example 4.1. Generation of the system: we generate a system $Ax = b$ of size 150 with solution $x = \sin(\pi x)e^{1+x}$, $A = M - N$ and $M^{-1}N$ being symmetric and $0 \leq \lambda \leq 0.96$ for $\lambda \in \lambda(M^{-1}N)$, $\lambda(G)$ denotes the spectrum of matrix G. k is chosen as 7.

Remark 4.1. 1. The spectrum of $M^{-1}N$ is chosen to fall between 0 and 0.96 for a better demonstration of the restarted scheme;

2. The relative residual $\|b - A * x\|/\|b\|$ is used as the stopping criteria and for observation as well.

Example 4.2. Generation of the system: we generate a system $Ax = b$ of size 1000 with solution x as in Example 4.1, $A = M - N$ and $M^{-1}N$ being symmetric and $0 \leq \lambda \leq 0.99$ for $\lambda \in \lambda(M^{-1}N)$, $\lambda(G)$ denotes the spectrum of matrix G. k is chosen as 7.

Table 1: Example 4.1. Comparison between algorithms.

Jacobi		Accelerated Jacobi		Chebyshev semi-iter.	
iter#	rel_err	iter#*	rel_err	iter#	rel_err
10	1.6732e-02	5	5.5102e-06	10	6.377
50	2.8303e-03	6	7.1765e-07	50	1.5738e-1
100	2.4902e-04	7	1.7439e-07	100	1.2924e-3
150	2.6169e-05	8	3.0166e-08	150	1.2812e-05
200	2.9986e-06	9	6.1530e-09	200	1.7618e-06
250	3.5303e-07	10	1.1449e-09	250	1.2632e-08
300	4.1790e-08	11	2.2409e-10	300	1.1395e-09
350	4.9405e-09	12	4.2512e-11	340	8.1047e-12
400	5.8215e-10	13	1.0266e-11		
450	6.8269e-11	14	2.6330e-12		
500	7.9035e-12	15	4.1299e-13		

*: Each iteration of accelerated Jacobi method needs $k(=7)$ Jacobi iterations.

Table 2: Example 4.2. Comparison between algorithms.

Jacobi		Accelerated Jacobi		Chebyshev semi-iter.	
iter#	rel_err	iter#*	rel_err	iter#	rel_err
200	4.1933e-04	5	8.516e-04	300	7.3532e-03
300	9.0763e-05	10	8.1074e-05	400	1.5552e-03
500	6.0424e-06	15	1.0509e-05	500	7.479e-05
700	5.0311e-07	20	1.5709e-06	600	1.0599e-05
900	4.7894e-08	25	2.5607e-07	700	2.0927e-06
1100	5.1402e-09	30	4.5088e-08	800	4.4771e-07
1300	6.0986e-10	35	8.5377e-09	900	4.3248e-08
1500	7.7409e-11	40	1.7185e-09	1000	3.1937e-09
1700	1.0119e-11	45	3.631e-10	1100	4.4871e-10
1900	1.2622e-12	50	7.9013e-11	1200	9.3449e-11

*: Each iteration of accelerated Jacobi method needs $k(=7)$ Jacobi iterations.

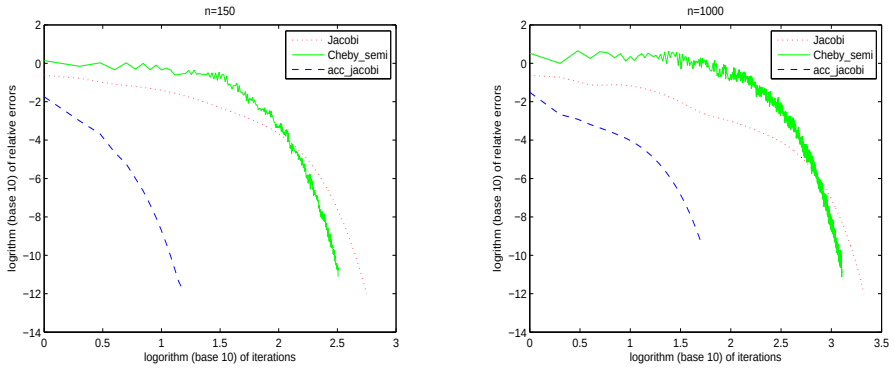

Figure 1: Comparison between Jacobi, accelerated Jacobi and Chebyshev semi-iterative methods.

Fig. 1 shows the graphic view of the relative errors via the iteration numbers of the three methods mentioned above. It seems obvious that our proposed method has

much better numerical performance than the other two methods while Chebyshev semi-iterative method does not show too much advantages over Jacobi method.

5 Summary and Comments

We studied an accelerative scheme for stationary iterative method in this paper, which is based on polynomial accelerative technique as well as restarting strategy. Numerical experiments are carried out to verify the efficiency of the proposed method. It is observed that our method behaves much better than the classical Jacobi method and the Chebyshev semi-iterative method.

References

[1] O. AXELSSON, Iterative Solution Methods, Cambridge University Press, 1994.
[2] S. CHANDRASEKAREN AND I. C. F IPSEN, *On the sensitivity of solution components in linear systems of equations*, SIAM J. Matrix. Appl., 16 (1995), pp. 93–112.
[3] B. N. DATTA, Numerical Linear Algebra and Applications, Brooks/Cole Publishing Company, Pacific Grove, California, 1995.
[4] G. H. GOLUB AND CHARLES F. VAN LOAN, Matrix Computation (Third edition), The John Hopkins University Press, Baltimore and London, 1996.
[5] L. A. HAGEMAN AND D. M. YOUNG, Applied Iterative Methods, Academic Press, New York, 1981.
[6] W. HACKBUSCH, Iterative Solution of Large Sparse System of Equations, Springer-Verlag, New York, 1994.
[7] G. H. GOLUB AND R. S. VARGA, *Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Riechardson iterative methods, Parts I and II*, Numer. Math., 3 (1961), pp. 147–156, 157–168.
[8] R. S VARGA, Matrix Iterative Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1962.
[9] D. M. YOUNG, Iterative Solution of Large Linear Systems, Academic Press, New York, 1971.