Kaczmarz method for saddle point systems

Jinmei Wang[1], Lizi Yin[1], and Ke Wang[2,*]

[1]School of Mathematical Sciences, University of Jinan, Jinan 250022, P.R. China

[2]Department of Mathematics, Shanghai University, Shanghai 200444, P.R. China

Abstract. The Kaczmarz method is presented for solving saddle point systems. The convergence is analyzed. Numerical examples, compared with classical Krylov subspace methods, SOR-like method (2001) and recent modified SOR-like method (2014), show that the Kaczmarz algorithm is efficient in convergence rate and CPU time.

1 Introduction

A block 2 × 2 linear system of the form

$$\begin{pmatrix} A & B \\ B^{\mathrm{T}} & 0 \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} b \\ q \end{pmatrix} \tag{1}$$

where $A\in \mathrm{R}^{m\times m}$ and $B\in \mathrm{R}^{m\times n}$, called a saddle point system, arises in a wide variety of technical and scientific applications such as constrained optimization, the finite element method to Stokes equations, fluid dynamics and weighted linear least squares problem [2]. In 2001, Golub, Wu and Yuan [6] proposed the SOR-like method for solving symmetric augmented linear system (1). Then, many improved numerical methods are suggested [1, 10, 13]. Benzi, Golub and Liesen [2] also gave a review on numerical solution of saddle point problems in 2005. Pan, Ng and Bai [9] developed a deteriorated positive-definite and skew-Hermitian splitting (DPSS) preconditioner for nonsymmetric saddle point system (1), based on which, a series of DPSS-type methods [3, 4, 12] are investigated.

The Kaczmarz algorithm is a popular iterative projection method [11] as it is simple to implement and the convergence is superior to classical splitting iterative methods such as SOR-like method. Many authors studied Kaczmarz methods for solving linear systems [7, 8, 11]. In this paper, the Kaczmarz algorithm is presented for symmetric saddle point system (1) with numerical comparisons to classical Krylov subspace methods [5] and splitting iterative methods SOR-like [6] (2001) and modified SOR-like [12] (MSOR-like, 2014).

The rest of this paper is organized as follows. In Section 2, the Kaczmarz method is presented and the convergence is analyzed. In Section 3, numerical examples are provided to show the effectiveness and efficiency of the algorithm. In Section 4, concluding remarks are drawn. Throughout the paper, I denotes the identity matrix.

2 Kaczmarz method

For saddle point problem (1), where $A\in \mathrm{R}^{m\times m}$ is symmetric positive definite, and $B\in \mathrm{R}^{m\times n}$ is of full column rank, Kaczmarz method can be proposed as follows,

---

KACZMARZ METHOD. *Given initial vectors $x_0\in R^m$ and $y_0\in R^n$, for $k = 0, 1, 2, \cdots$, the following iterative scheme is taken,*

$$\begin{cases} x_{k+1} = x_k + \dfrac{q^{(i_k)}-(B^{\mathrm{T}})^{(i_k)}x_k}{\left\|(B^{\mathrm{T}})^{(i_k)}\right\|_2^2}\left((B^{\mathrm{T}})^{(i_k)}\right)^{\mathrm{T}}, \\ y_{k+1} = y_k + \dfrac{(b-Ax_{k+1})^{(j_k)}-B^{(j_k)}y_k}{\left\|B^{(j_k)}\right\|_2^2}\left(B^{(j_k)}\right)^{\mathrm{T}}, \end{cases} \tag{2}$$

where $i_k=(k \bmod n)+1$, $j_k=(k \bmod m)+1$, $(\cdot)^{(i_k)}$ and $(\cdot)^{(j_k)}$ denote the i_kth row and j_kth row of a matrix.

The convergence result for method (2) is as the following theorem.

Theorem 2.1. *Suppose that saddle system* (1) *is consistent. Then the iteration sequence $\left\{\begin{pmatrix} x_k \\ y_k \end{pmatrix}\right\}_{k=0}^{\infty}$, generated by the Kaczmarz method* (2) *starting from an initial guess $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ with x_0 in in the column space of B and y_0 in the column space of B^{T}, converges to the unique least-norm solution $\begin{pmatrix} x^* \\ y^* \end{pmatrix} = \begin{pmatrix} A & B \\ B^{\mathrm{T}} & 0 \end{pmatrix}^{\dagger}\begin{pmatrix} b \\ q \end{pmatrix}$ of* (1), *where $(\cdot)^{\dagger}$ indicates the Moore-Penrose inverse of a matrix. Moreover, the solution error for the iteration sequence is*

* Corresponding author: kwang@shu.edu.cn

$$\|x_{k+1}-x^*\|_2^2 \le \left(1-\frac{\lambda_{\min}(BB^{\mathrm{T}})}{\|B\|_F^2}\right)\|x_k-x^*\|_2^2,$$

$$\|y_{k+1}-y^*\|_2 = \sqrt{1-\frac{\lambda_{\min}(B^{\mathrm{T}}B)}{\|B\|_F^2}}\,(\|y_k-y^*\|_2 + \sqrt{\lambda_{\max}(A^{\mathrm{T}}A)}\|x_k-x^*\|_2),$$

where $\lambda_{\min}(\cdot)$ *and* $\lambda_{\max}(\cdot)$ *are the the smallest and largest nonzero eigenvalues of a matrix*.

3 Numerical experiments

Two numerical examples are discussed using Kaczmarz algorithm (2) compared with the classical SOR-like method [6], recent MSOR-like method [12] and Krylov subspace methods with preconditioner

$$P=\begin{pmatrix}\hat{A} & 0\\ 0 & \hat{S}\end{pmatrix} \tag{3}$$

where $\hat{A}$ = diag(A) and $\hat{S} = B^{\mathrm{T}}\hat{A}^{-1}B$ [5] for solving symmetric saddle system (1). All runs are performed in MATLAB 7.12 on an Intel Core CPU 2.80 GHz (8.00 GB RAM) Windows 7 system.

The first example is for solving weighted least squares problem [10].

Example 3.1. The weighted linear least squares problem

Find: x^* so that: $\|Bx^*-b\|_{A^{-1}}^2 = \min_{x\in R^n} \|Bx-b\|_{A^{-1}}^2$, induces the following saddle point system

$$\begin{pmatrix}A & B\\ B^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}z\\ x\end{pmatrix}=\begin{pmatrix}b\\ 0\end{pmatrix}, \tag{4}$$

where $z = A^{-1}(b-Bx)$.

With $A = \mathrm{tridiag}(1,2,1) \in R^{m\times m}$ and $B = I_m$, the Kaczmarz method is applied to problem (4). Take $Q = B^{\mathrm{T}}B$ as in [6] and the optimal parameter ω_b for SOR-like and MSOR-like methods. The numerical results are listed in Table 1, where the norm of absolute residual vectors is defined as

$$r=\sqrt{\|b-Bx_k-Az_k\|_2^2+\|B^{\mathrm{T}}z_k\|_2^2},$$

the initial vector is taken as

$$\begin{pmatrix}z_0\\ x_0\end{pmatrix}=0,$$

and all runs terminate if $r \le 10^{-7}$.

Table 1 shows that Kaczmarz method converges faster than MSOR-like and SOR-like methods with the optimal parameter ω_b, that is, Kaczmarz method needs much fewer iterations and much less CPU time and and has much higher precision than MSOR-like and SOR-like methods which have similar convergence.

Next example in [1] is for solving the Stokes problem.

Example 3.2. Consider Stokes problem: find **u** and **p** such that

$$\begin{cases}-\mu\Delta\mathbf{u}+\nabla\mathbf{p}=f, & \text{in } \Omega,\\ \nabla\mathbf{u}=g, & \text{in } \Omega,\\ \mathbf{u}=0, & \text{on } \partial\Omega,\\ \int_\Omega \mathbf{p}(x)\,dx=0, & \end{cases} \tag{5}$$

where $\Omega = (0,1)\times(0,1) \subset \mathrm{R}^2$, $\partial\Omega$ is the boundary of Ω, Δ is the componentwise Laplace operator, **u** is a vector-valued function representing the velocity, and **p** is a scalar function representing the pressure. By discretizing (5) with the upwind scheme, the following saddle point system of linear equations is obtained:

$$\begin{pmatrix}A & B\\ B^{\mathrm{T}} & 0\end{pmatrix}\begin{pmatrix}\mathbf{u}\\ \mathbf{p}\end{pmatrix}=\begin{pmatrix}f\\ g\end{pmatrix}. \tag{6}$$

In the experiment, the right hand side of f and g is chosen such that the exact solution of (6)

$$\begin{pmatrix}\mathbf{u}^*\\ \mathbf{p}^*\end{pmatrix}=\begin{pmatrix}1\\ 1\\ \vdots\\ 1\end{pmatrix}\in R^{m+n},$$

with

$$A=\begin{pmatrix}\boldsymbol{I}\otimes\boldsymbol{T}+\boldsymbol{T}\otimes\boldsymbol{I} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{I}\otimes\boldsymbol{T}+\boldsymbol{T}\otimes\boldsymbol{I}\end{pmatrix}\in R^{2q^2\times 2q^2},$$
$$B=I\in R^{2q^2\times 2q^2},$$

and

$$T=\frac{1}{h^2}\cdot\mathrm{tridiag}(-1,2,-1)\in R^{q\times q},$$

Table 1: Iterations (IT), CPU time (CPU) and absolute residual (r) for Example 3.1

$m(n)$	Kaczmarz			MSOR-like			SOR-like		
	IT	CPU	r	IT	CPU	r	IT	CPU	r
20	20	0.0004	7.7×10^{-15}	152	0.0027	9.6×10^{-8}	165	0.0031	9.7×10^{-8}
200	200	0.0177	5.7×10^{-12}	1746	0.2118	9.9×10^{-8}	1733	0.2077	9.9×10^{-8}
2000	2000	29.2106	3.8×10^{-9}	18404	627.1635	1.0×10^{-7}	18386	626.4071	1.0×10^{-7}

where $\otimes$ denotes the Kronecker product, $h=\frac{1}{q+1}$ is the discretization mesh size, and the relative error is defined as

$$\mathrm{ERR}=\frac{\sqrt{\|\mathbf{u}_k-\mathbf{u}^*\|_2^2+\|\mathbf{p}_k-\mathbf{p}^*\|_2^2}}{\sqrt{\|\mathbf{u}^*\|_2^2+\|\mathbf{p}^*\|_2^2}}.$$

With $m = n = 2q^2$, the initial vector

$$\begin{pmatrix} \mathbf{u}_0 \\ \mathbf{p}_0 \end{pmatrix} = 0$$

and all runs terminated if ERR$\leq 10^{-7}$, the numerical results are listed in Table 2, where $Q = B^{\mathrm{T}}B$ and $\omega = \omega_b$ (the optimal) for SOR-like and MSOR-like.

From the numerical results in Tables 2 and 3, it can be seen that the Kaczmarz method is superior to MSOR-like and SOR-like methods with the optimal parameter ω_{opt} and the classical Krylov subspace methods, which is also shown from Table 1 in Example 3.1. Obviously, the Kaczmarz method requires fewer iterations and less CPU time and has higher precision than MSOR-like and SOR-like methods.

Table 2: Iterations (IT), CPU time (CPU) and absolute residual (r) for Example 3.2

$m(n)$	Kaczmarz			MSOR-like			SOR-like		
	IT	CPU	r	IT	CPU	r	IT	CPU	r
242	483	0.0327	0	1281	0.1222	1.0×10^{-7}	4120	0.1383	1.0×10^{-7}
648	1295	0.1522	0	2987	6.6767	1.0×10^{-7}	9680	3.9641	1.0×10^{-7}
1250	2499	2.1855	0	5419	48.9726	1.0×10^{-7}	19786	32.5917	1.0×10^{-7}

Table 3: Krylov subspace methods for Example 3.2

$m(n)$	242			648			1250		
method	IT	CPU	r	IT	CPU	r	IT	CPU	r
bicg	55	0.0789	3.7×10^{-8}	186	1.3151	3.2×10^{-8}	401	9.0979	5.9×10^{-8}
bicgstab	139	0.1394	4.6×10^{-8}	772	4.4161	9.0×10^{-8}	-	-	-
bicgstabl	60	0.1362	4.4×10^{-8}	276	3.3722	9.2×10^{-8}	623	24.9574	9.8×10^{-8}
cgs	-	-	-	-	-	-	-	-	-
gmres	-	-	-	-	-	-	-	-	-
lsqr	9	0.0539	8.0×10^{-11}	10	0.4050	3.5×10^{-10}	-	-	-
minres	1602	0.5708	9.9×10^{-8}	3115	8.4035	9.9×10^{-8}	5114	48.5307	1.0×10^{-7}
pcg	-	-	-	-	-	-	-	-	-
qmr	57	0.0903	1.1×10^{-8}	189	1.3779	4.0×10^{-8}	414	9.5015	9.9×10^{-8}
symmlq	4084	1.8005	9.6×10^{-8}	5062	13.3150	9.6×10^{-8}	4089	39.1793	9.3×10^{-8}
tf qmr	-	-	-	-	-	-	-	-	-

All tests use Matlab built-in iterative solvers and the preconditioner is taken as (3).

4 Conclusion

In this paper, the Kaczmarz method is suggested for solving symmetric saddle point system, which is a popular iterative projection method and does not need multiplications of matrix and vector (different to classical splitting iterative method), thus easy to implement and more efficient. Numerical examples show that the Kaczmarz algorithm is much faster than the classical Krylov subspace methods and splitting iterative methods SOR-like and recent MSOR-like and has higher precision for a special case where $B = I$. As for general cases, additional techniques are expected, which will be presented in further work.

Acknowledgement

This work was supported by 2019 Shandong Postgraduate Education Quality Curriculum Project, Jinan University Graduate Teaching Reform Research Project (jdyy903), Jinan University Teaching Reform Research Project (jzc1903), and Key Scientific Research Project of Colleges and Universities in Henan Province (20B110012), China.

References

1. Z.-Z. Bai, B.N. Parlett, Z.-Q. Wang, On generalized successive overrelaxation methods for augmented linear systems, Numerische Mathematik 102 (2005) 1-38.
2. M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numerica 14 (2005) 1-137.
3. Y. Cao, Regularized DPSS preconditioners for non-Hermitian saddle point problems, Applied Mathematics Letters 84 (2018) 96-102.
4. Y. Cao, J.-L. Dong, Y.-M. Wang, A relaxed deteriorated PSS preconditioner for nonsym-metricsaddle point problems from the steady Navier-Stokes equation, Journal of Computational and Applied Mathematics 273 (2015) 41-60.
5. J. Chiu, L. Davidson, A. Dutta, J. Gou, K.C. Loy, M. Thom, D. Trenev, Efficient and robust solution strategies for saddle-point systems, University of Minnesota, Institute for Mathematics and Its Applications (2014). Retrieved from the University of Minnesota Digital Conservancy. http://hdl.handle.net/11299/181496.

6. G.H. Golub, X. Wu, J.-Y. Yuan, SOR-like methods for augmented systems, BIT 41 (2001) 71-85.
7. Y. Liu, C.-Q. Gu, Variant of greedy randomized Kaczmarz for ridge regression, Applied Numerical Mathematics 143 (2019) 223-246.
8. Y.-Q. Niu, B. Zheng, A greedy block Kaczmarz algorithm for solving large-scale linear systems, Applied Mathematics Letters 104 (2020) 106294.
9. J.-Y. Pan, M.K. Ng, Z.-Z. Bai, New preconditioners for saddle point problems, Applied Mathematics and Computation 172 (2006) 762-771.
10. Y. Wei, X. Yu, R. Zhang, Preconditioned conjugate gradient method and generalized successive over relaxation method for the weighted least squares problems, International Journal of Computer Mathematics 81 (2004) 203-214.
11. J.-J. Zhang, A new greedy Kaczmarz algorithm for the solution of very large linear systems. Applied Mathematics Letters 91 (2019) 207-212.
12. J. Zhang, C. Gu, A variant of the deteriorated PSS preconditioner for nonsymmetric saddle point problems, BIT 56 (2016) 587-604.
13. B. Zheng, K. Wang, Y. Wu, SSOR-like methods for saddle point problems, International Journal of Computer Mathematics 86 (2009) 1405-1423.