Revista Integración
Escuela de Matemáticas
Universidad Industrial de Santander
Vol. 32, No. 1, 2014, pág. 91–100

An acceleration technique for the Gauss-Seidel method applied to symmetric linear systems

Jesús Cajigas[a,*], Isnardo Arenas[b], Paul Castillo[a]

[a] University of Puerto Rico, Department of Mathematical Sciences, Mayagüez, Puerto Rico 00681, US.
[b] The University of Texas at Dallas, Department of Mathematical Sciences, Richardson, TX 75080-3021.

Abstract. A preconditioning technique to improve the convergence of the Gauss-Seidel method applied to symmetric linear systems while preserving symmetry is proposed. The preconditioner is of the form $I + \mathcal{K}$ and can be applied an arbitrary number of times. It is shown that under certain conditions the application of the preconditioner a finite number of steps reduces the matrix to a diagonal. A series of numerical experiments using matrices from spatial discretizations of partial differential equations demonstrates that both versions of the preconditioner, point and block version, exhibit lower iteration counts than its non-symmetric version.

Keywords: Preconditioning, Gauss-Seidel method, regular splitting.
MSC2000: 65F08, 65F10, 65F50.

Una técnica de aceleración para el método Gauss-Seidel aplicado a sistemas lineales simétricos

Resumen. Se propone una técnica de precondicionamiento para mejorar la convergencia del método Gauss-Seidel aplicado a sistemas lineales simétricos pero preservando simetría. El precondicionador es de la forma $I + \mathcal{K}$ y puede ser aplicado un número arbitrario de veces. Se demuestra que bajo ciertas condiciones la aplicación del precondicionador un número finito de pasos reduce la matriz del sistema precondicionado a una diagonal. Una serie de experimentos con matrices que provienen de la discretización de ecuaciones en derivadas parciales muestra que ambas versiones del precondicionador, por punto y por bloque, muestran un menor número de iteraciones en comparación con la versión que no preserva simetría.

Palabras claves: Precondicionamiento, método de Gauss-Seidel, descomposiciones regulares.

[*] Corresponding author: *E-mail*: jesus.cajigas@upr.edu.

Received: 16 December 2013, Accepted: 20 March 2014.
To cite this article: J. Cajigas, I. Arenas, P. Castillo, An acceleration technique for the Gauss-Seidel method applied to symmetric linear systems, *Rev. Integr. Temas Mat.* 32 (2014), no. 1, 91–100.

1. Introduction

The slow convergence of the Gauss-Seidel relaxation method has prompted several researchers to develop preconditioning techniques to accelerate the convergence of this iterative method. Acceleration techniques for Jacobi and Gauss-Seidel methods applied to diagonally dominant Z matrices were proposed in [10], [8], [6] and [7]. A survey of such techniques can be found in [11]. A more recent analysis of preconditioners applied to a wider class of Z matrices can be found in [13]. Extensions of these preconditioners to other classes of matrices have also been proposed, for instance in [9, 12] for M matrices and in [14] for H matrices.

To accelerate the convergence of the Jacobi and Gauss-Seidel methods, preconditioners of the form $I + \mathcal{K}$ were originally introduced by Milaszewic in [10]. Following this idea and using a different definition of matrix $\mathcal{K}$, another preconditioner, namely $I+S_{max}$ was proposed in [7]. An extension of this preconditioner was recently proposed by the authors in [1]. The preconditioner is based on the application of a fixed but arbitrary number of $I + S_{max}$ steps. The analysis was carried out for diagonally dominant Z matrices. Numerical experiments showed good performance for a wider class of matrices including those not covered by the theoretical analysis. A block version of the preconditioner was numerically tested for the first time showing a superior performance compared to its standard point version.

Consider the splitting $A = -L + D - U$ of a non singular matrix A where $-L$ and $-U$ are the strict lower and upper triangular parts of A and D its main diagonal. Recalling that the Gauss-Seidel iteration is a fixed point iteration of the form

$$x_{n+1} \; = \; M^{-1}Nx_n + M^{-1}b,$$

where $M = D - L$ and $N = U$. The aim of the $I + S_{max}$ preconditioner is to reduce the spectral radius of the iteration matrix $M^{-1}N$ by annihilating, on each row of U, the entry with greatest magnitude. Unfortunately, this technique does not preserve symmetry when applied to a symmetric matrix; and, if such preconditioner is applied to both sides of A, that is $(I+S_{max})A(I+S_{max})^T$, symmetry is obviously preserved, but new non zero entries are introduced. In this work a preconditioner, S, that applies Kotakemori's idea while preserving symmetry is introduced. As in [1], the proposed preconditioner can be applied a fixed but arbitrary number of steps. A block version is also numerically tested and compared against its point version.

2. Symmetric preconditioner

The proposed preconditioner is of the form $S = I + \mathcal{K}$ and takes as a starting point Kotakemori's idea and the product SAS^T to preserve symmetry. The entries of matrix $\mathcal{K}$ are defined as follows:

$$\mathcal{K}_{i,j} \; = \; \begin{cases} -\dfrac{a_{i,k_i} + \mathcal{K}_{k_i,k_{k_i}} a_{i,k_{k_i}}}{a_{k_i,k_i} + \mathcal{K}_{k_i,k_{k_i}} a_{k_i,k_{k_i}}} & \text{if } j = k_i, \\ \\ 0 & \text{if } j \neq k_i, \end{cases} \tag{1}$$

where

$$k_i = \min\{j \mid |a_{i,j}| = \max_{k>i} |a_{i,k}|\}.$$

For $k_i = n$ the value $\mathcal{K}_{n,k_n}$ is assumed to be zero.

Lemma 2.1. *Let A be a non-singular symmetric matrix and S its symmetric preconditioner; then SAS^T is a non-singular symmetric matrix.*

Proof. First notice that symmetry is trivially preserved. Since

$$\det(SAS^T) = \det(S)\det(A)\det(S^T)$$

and neither one of these determinants is zero, then SAS^T is non-singular. ☑

Lemma 2.2. *Let A be a symmetric positive definite matrix and S its symmetric preconditioner; then SAS^T is a positive definite matrix.*

Proof. Let x be a non-zero vector. By the non-singularity of S^T the vector $y = S^Tx$ is also a non null vector. Since A is symmetric positive definite we have

$$x^TSAS^Tx \;=\; \langle S^Tx, AS^Tx\rangle \;=\; \langle y, Ay\rangle \;>\; 0,$$

which proves that SAS^T is also symmetric positive definite. ☑

2.1. Recursive preconditioner $S^{[k]}$

Following the idea presented in [1] we define a *recursive* preconditioner $S^{[k]}$ by applying a fixed but arbitrary number of times the preconditioner S as follows. Let A be a symmetric matrix; we set $S^{[0]} = I$ and $A_0 = (S^{[0]})A(S^{[0]})^T = A$ and for all $k \geq 1$ we define

$$A_{k+1} \;=\; \left(S^{[k]}\right) A_k \left(S^{[k]}\right)^T,$$

where $S^{[k]}$ is the preconditioner S applied to matrix A_k. We now state our main result.

Theorem 2.3. *Let A be a non-singular $n \times n$ symmetric-matrix; then there exists $k_A \in \mathbb{N}$ such that A_{k_A} is a diagonal matrix.*

Proof. We proceed by induction. Let A be a non-singular 2×2 symmetric matrix. Then

$$S^{[1]} \;=\; \begin{pmatrix} 1 & -\frac{a_{1,2}}{a_{2,2}} \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad A_1 \;=\; \begin{pmatrix} a_{1,1} - \frac{a_{1,2}a_{2,1}}{a_{2,2}} & 0 \\ 0 & a_{2,2} \end{pmatrix}.$$

Now, suppose that for any $n \times n$ symmetric-matrix A, there exists k_n such that A_{k_n} is a diagonal matrix. Let A be an $(n+1) \times (n+1)$ symmetric matrix; then, it can be partitioned as follows:

$$A \;=\; \left(\begin{array}{c|ccc} a_{1,1} & a_{1,2} & \dots & a_{1,n+1} \\ \hline a_{2,1} & & & \\ \vdots & & A_{2,2} & \\ a_{n+1,1} & & & \end{array}\right).$$

By the induction hypothesis, there exists an integer k_1 such that after applying k_1 steps of the symmetric preconditioner to matrix A the sub-block $A_{2,2}$ reduces to a diagonal matrix. We now consider entry $(1, n+1)$ of the preconditioned matrix A_{k_1}. We have two cases:

- Case $A_{k_1}(1, n+1) = 0$. We can consider the following block partition of A_{k_1}:

$$A_{k_1} = \left(\begin{array}{ccc|c} & & & 0 \\ & A_1 & & \vdots \\ & & & 0 \\ \hline 0 & \dots & 0 & * \end{array}\right).$$

  Using the hypothesis of induction one more time, there exists an integer k_2 such that after applying k_2 steps of the symmetric preconditioner to matrix A_{k_1}, $A_{k_1+k_2}$ is a diagonal matrix.

- Case $A_{k_1}(1, n+1) \neq 0$. Consider the block partition

$$A_{k_1} = \left(\begin{array}{cccc|c} & & & & * \\ & & & & 0 \\ & & A_1 & & \vdots \\ & & & & 0 \\ \hline * & 0 & \dots & 0 & * \end{array}\right).$$

  We consider two possible sub-cases: first, suppose that after a finite number of preconditioning steps, k_2, entry $(1, n+1)$ becomes 0, that is $A_{k_1+k_2}(1, n+1) = 0$; the result follows since we fall into previous case. On the other hand, if for any number $k \geq 1$ of preconditioning steps entry $(1, n+1) \neq 0$, this means that $S^{[k]}$ acts only over the block A_1, for any $k \geq 1$. However, by the induction hyphotesis there exists a number k_2 such that matrix A_1 becomes diagonal. Therefore, after $k_1 + k_2 + 1$ steps entry $(1, n+1)$ becomes 0 which is a contradiction. ☑

3. Computational issues

Taking advantage of the structure of matrix $S = I + \mathcal{K}$, only two values per row are needed to store this preconditioner: for row i, we store the value $\mathcal{K}_{i,k_i}$ in $Sarray[i]$ and the index value k_i in $Jarray[i]$; both values have been defined in Equation (1). Using this data structure, preconditioner $S^{[k]}$ requires a memory space $\mathcal{O}(2kn)$ instead of $\mathcal{O}(kn^2)$. An important detail to consider is that only S is stored but not S^T, which is needed in the computation of the solution of the original problem. Since $k_i > i$, the computation is done starting from row $n-1$, as shown in Algorithm 3.1.

Algorithm 3.1 Computation of S

```
$Sarray[n] \leftarrow 0$
**for** $i = n-1 : 1$ **do**
    $Jarray[i] \leftarrow k_i$ where $k_i = \min\{j \mid |a_{i,j}| = \max_{k>i} |a_{i,k}|\}$
    $k_{k_i} = Jarray[k_i]$
    $Sarray[i] \leftarrow -\dfrac{a_{i,k_i} + Sarray[k_i] a_{i,k_{k_i}}}{a_{k_i,k_i} + Sarray[k_i] a_{k_i,k_{k_i}}}$
**end for**
```

To find the index value k_i a search above the main diagonal on each row is conducted. This search takes $\mathcal{O}(nnz(i))$ time and the entire process has a complexity $\mathcal{O}(nnz)$ where nnz is the total number of non-zero entries in matrix A. For matrices arising from discretizations of partial differential equations the complexity is linear $\mathcal{O}(n)$ with respect to the number of unknowns, n.

The assembly of the preconditioned matrix SAS^T is presented in Algorithm 3.2. Given the structure of S, row i of (SA) is a linear combination of two rows in A:

$$(SA)_{i,*} = A_{i,*} + Sarray[i] \cdot A_{k_i,*},$$

as shown in lines $2-3$ of Algorithm 3.2. Column j of SAS^T is a linear combination of two columns of (SA):

$$(SAS^T)_{*,j} = (SA)_{*,j} + Sarray[j] \cdot (SA)_{*,k_j},$$

lines $4-6$ of Algorithm 3.2. Finally, since we use a storage scheme for sparse matrix a compression of the matrix is performed in line 7.

Algorithm 3.2 Computation of SAS^T

```
1: **for** $i = 0 : i < n$ **do**
2:     $row = A_{i,*}$
3:     $row = row_{i,*} + Sarray[i] \cdot A_{k_i,*}$
4:     **for** $j = 0 : n-1$ **do**
5:         $row_j = row_j + Sarray[j] \cdot row_{k_j}$
6:     **end for**
7:     Compress resulting matrix
8: **end for**
```

4. Numerical experiments

The purpose of this section is to carry out a series of experiments to assess the quality of the proposed preconditioner applied to matrices obtained from the discretization of partial differential equations using the Finite Volume method and the Local Discontinuous Galerkin (LDG) method [3, 2], which is a high order discontinuous finite element method. The symmetric version, $S^{[k]}$, and its non symmetric version $P^{[k]}$, analyzed in

[1], are compared as well as its point and block versions. We use a stopping criteria based on relative residual norms, that is

$$\|b - Ax_n\|_2 \ \leq \ r\|b - Ax_0\|_2,$$

where the relative tolerance r is chosen according to the problem. Additionally, a maximum number of iterations of 5000 is used.

4.2. Porous media problem

In this experiment, matrix A is obtained from the discretization of the Laplacian operator $-\nabla \circ \mathcal{D}\nabla p$ in $\Omega = [0,1] \times [0,1]$ using the Finite Volume method. Dirichlet boundary conditions are imposed at $x = 0$, $p = 1$, and at $x = 1$, $p = 0$. No flow boundary conditions are imposed at $y = 0$ and $y = 1$. The permeability tensor takes the values $\mathcal{D} = 10^{-6} I_d$ and $\mathcal{D} = I_d$ according to the permeability field distribution proposed in [5]. Starting from a spatial uniform discretization of 20×20 rectangular cells and by performing three consecutive global refinement, matrices A_i, $i = 1, \ldots, 4$ of order $400, 1600, 6400$ and 25600 respectively, are considered in this set of experiments. The right hand side is chosen such that the vector x^* with random generated entries in $(0,1)$ is the exact solution of the linear system.

Table 1 show the number of iterations until convergence for both preconditioners using different number of preconditioning steps. For all matrices, the total number of iterations using preconditioner $S^{[k]}$ is lower, almost by a factor of 2, compared to those obtained for the non-symmetric preconditioner $P^{[k]}$.

	$P^{[1]}$	$S^{[1]}$	$P^{[5]}$	$S^{[5]}$	$P^{[10]}$	$S^{[10]}$	$P^{[15]}$	$S^{[15]}$	$P^{[20]}$	$S^{[20]}$
A_1	431	**398**	183	**111**	112	**63**	89	**44**	75	**35**
A_2	1787	**1665**	784	**477**	466	**261**	368	**182**	303	**140**
A_3	–	–	2850	**1745**	1709	**972**	1363	**686**	1131	**533**
A_4	–	–	–	–	–	**3334**	4639	**2393**	3879	**1885**

Table 1. Iteration count for isotropic permeability field.

In Table 2 we show the spectral radius of the preconditioned Gauss-Seidel iteration matrix for the first three matrices. Our numerical results indicate that when $S^{[k]}$ is applied recursively the spectral radius is decreasing faster than when $P^{[k]}$ is applied. This behavior agrees with the low iteration count obtained when the symmetric preconditioner is used.

	0	$P^{[1]}$	$S^{[1]}$	$P^{[5]}$	$S^{[5]}$	$P^{[10]}$	$S^{[10]}$	$P^{[15]}$	$S^{[15]}$	$P^{[20]}$	$S^{[20]}$
A_1	0.987	0.978	**0.976**	0.947	**0.912**	0.912	**0.843**	0.889	**0.782**	0.868	**0.724**
A_2	0.997	0.996	**0.995**	0.989	**0.982**	0.981	**0.966**	0.976	**0.951**	0.971	**0.936**
A_3	0.999	0.999	**0.999**	0.998	**0.996**	0.996	**0.992**	0.995	**0.989**	0.993	**0.986**

Table 2. Spectral radius of preconditioned Gauss-Seidel iteration matrix.

Since our interest is the performance of the preconditioner on sparse matrices, we consider the standard Compressed Sparse Row, CSR, storage scheme. In Table 3 we show the ratio of the total number of non-zero coefficients between the preconditioned and the original matrix as a measure of memory storage. Compared to preconditioner $P^{[k]}$, the symmetric version $S^{[k]}$ requires more storage for the same number of preconditioning steps.

	$P^{[1]}$	$S^{[1]}$	$P^{[5]}$	$S^{[5]}$	$P^{[10]}$	$S^{[10]}$	$P^{[15]}$	$S^{[15]}$	$P^{[20]}$	$S^{[20]}$
A_1	**1.4**	1.8	**6.07**	18.7	**17.22**	61.8	**27.48**	72.7	**31.66**	72.8
A_2	**1.4**	1.9	**6.59**	20.7	**22.34**	94.1	**48.77**	204.5	**78.46**	287.3
A_3	**1.4**	1.9	**6.94**	20.8	**24.39**	109.2	**58.01**	289.7	**107.75**	547.7
A_4	**1.4**	1.9	**7.02**	20.6	**24.84**	110.5	**61.15**	316.3	**119.50**	678.7

Table 3. Ratio of non-zeros coefficients relative to non-preconditioned matrix.

4.3. Random isotropic permeability field

We now consider a more challenging problem by using a random isotropic permeability field. The permeability tensor of cell i is of the form $D_i = 10^{\alpha_i} I$ where $\alpha_i \in (-8, 1)$. The right hand side was chosen as in the previous experiment. In Table 4 we report the operation count for both preconditioners, again the symmetric version $S^{[k]}$ exhibits lower iteration counts than its non symmetric version $P^{[k]}$.

	$P^{[1]}$	$S^{[1]}$	$P^{[5]}$	$S^{[5]}$	$P^{[10]}$	$S^{[10]}$	$P^{[15]}$	$S^{[15]}$	$P^{[20]}$	$S^{[20]}$
A_1	–	–	2330	**1464**	808	**310**	411	**115**	248	**57**
A_2	–	–	–	**4076**	3241	**1179**	1869	**579**	1258	**332**
A_3	–	–	–	–	–	**4764**	–	**2570**	–	**1620**
A_4	–	–	–	–	–	–	–	–	–	**4560**

Table 4. Iteration count for random isotropic permeability field.

5. Block version of $S^{[k]}$

Following the construction process done for the point version of the preconditioner, a block version of it is now defined. We assume that matrix A has been partitioned with a block structure, then $\mathcal{K}$ can be defined as follows:

$$\mathcal{K}_{i,j} = \begin{cases} -(A_{i,k_i} + \mathcal{K}^T_{k_i,k_{k_i}} A_{i,k_{k_i}})(A_{k_i,k_i} + \mathcal{K}^T_{k_i,k_{k_i}} A_{k_i,k_{k_i}})^{-1} & \text{if } j = k_i, \\ 0 & \text{if } j \neq k_i, \end{cases} \tag{2}$$

where

$$k_i = \min\left\{ j : \| A_{i,j} \| = \max_{k>i} \| A_{i,k} \| \right\},$$

and $\|\|$ is a matrix norm. The appropriate selection of which norm to choose depends specifically on the problem. For illustration purposes we have compared different norms on a particular problem. We considered matrices obtained from the LDG spatial discretization of a diffusion problem in a three dimensional domain using polynomials of degree one. As described in [4], these matrices have a natural sparse block structure where each block is a 4 × 4 dense block. The matrix is symmetric positive definite, its block dimensions are 2523 × 2523 and its global matrix dimensions are 10092 × 10092. We consider the standard norms $||\cdot||_F$, $||\cdot||_1$, $||\cdot||_\infty$ and $||\cdot||_m$ which is defined as:

$$\| A \|_m = \max\{|a_{ij}|,\ \ i,j = 1,2,...,n\}$$

We have set the relative tolerance to 10^{-9}. Since the right hand side is know we use as initial guess the null vector. Table 5 shows the dependence of the iteration count with respect to the matrix norm used. For this particular example, although a slight variations in the iteration count is observed, the infinite norm seems to produce the lowest iteration count.

Norm	$S^{[10]}$	$S^{[15]}$	$S^{[20]}$	$S^{[25]}$
$\|\|\cdot\|\|_1$	2868	1738	1220	930
$\|\|\cdot\|\|_F$	2811	1701	1194	**902**
$\|\|\cdot\|\|_\infty$	**2759**	1687	**1175**	**902**
$\|\|\cdot\|\|_m$	2892	**1170**	1250	960

Table 5. Number of iterations until convergence for the LDG stiffness matrix.

Tables 6 and 7 show the amount of memory needed by the preconditioned matrix for all different block norms, in Megabytes and total number of non-zero entries, respectively. In general we observe no substantial difference for a fixed number of preconditioning steps k. For k less or equal to 10 the infinite norm yield less fill in, while for k equal to 25 the Frobenius norm, the preconditioned matrix requires less memory. However is important to observe that as we perform more preconditioning steps the preconditioned matrix becomes denser no matter what norm is used; therefore asymptotically the choice of the block norm becomes irrelevant.

Norm	$S^{[1]}$	$S^{[5]}$	$S^{[10]}$	$S^{[15]}$	$S^{[20]}$	$S^{[25]}$
$\|\|\cdot\|\|_1$	4.72	30.59	76.07	**120.03**	**163.15**	206.61
$\|\|\cdot\|\|_F$	4.63	30.28	76.09	120.48	164.38	**205.68**
$\|\|\cdot\|\|_\infty$	**4.48**	**29.52**	**75.50**	120.87	164.04	205.76
$\|\|\cdot\|\|_M$	4.68	30.22	76.58	122.48	167.67	208.93

Table 6. Memory storage (in Megabytes) for preconditioned matrix using different block norms.

Norm	$S^{[1]}$	$S^{[5]}$	$S^{[10]}$	$S^{[15]}$	$S^{[20]}$	$S^{[25]}$
$\|\|\cdot\|\|_1$	29697	195487	487021	**768863**	**1045271**	1323833
$\|\|\cdot\|\|_F$	29121	193517	487187	771715	1053125	**1317861**
$\|\|\cdot\|\|_\infty$	**28143**	**188659**	**483421**	774241	1050943	1318403
$\|\|\cdot\|\|_M$	29415	193129	490303	784525	1074251	1338707

Table 7. Total number of non zero blocks of preconditioned matrix using different norms.

5.4. Block $S^{[k]}$ versus block $P^{[k]}$

We now compare the block version of both preconditioners: symmetric $S^{[k]}$ versus non-symmetric $P^{[k]}$, [1]. The linear system used for this experiment is the same of the previous section. Figure 1 shows the behavior of the Euclidean norm of the residuals. Clearly we observe that for a given number of preconditioning steps, k, $S^{[k]}$ requires less iterations than $P^{[k]}$ almost by a factor of two.

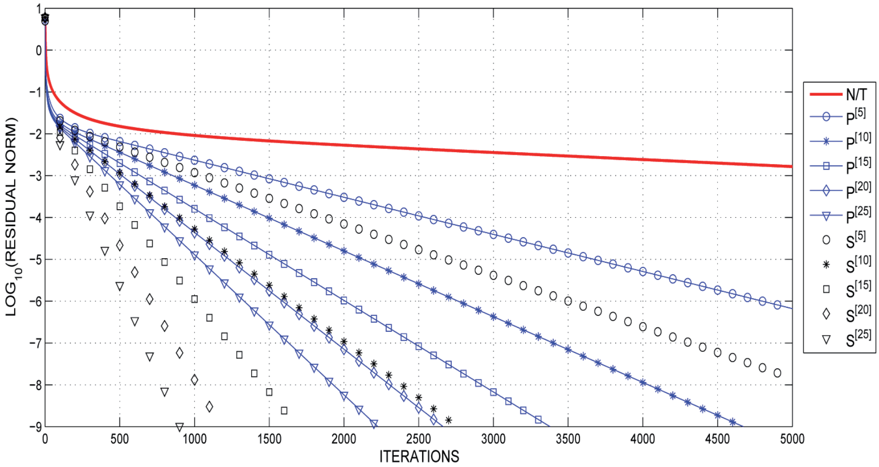

Figure 1. Residual history of $S^{[k]}$ and $P^{[k]}$ preconditioners for different number of steps k.

Finally we compare the point versus the block version of the same symmetric preconditioner $S^{[k]}$. Using the same matrix as before, Table 8 shows that the block version improves the convergence of the Gauss-Seidel even further.

	$S^{[1]}$	$S^{[5]}$	$S^{[10]}$	$S^{[15]}$	$S^{[20]}$	$S^{[25]}$
Point	–	–	–	3696	2770	2207
Block	–	–	**2759**	**1687**	**1175**	**902**

Table 8. Point vs. Block version of preconditioner $S^{[k]}$ using the infinite norm.

6. Concluding remarks

We have developed $S^{[k]}$, a symmetric version of the recursive preconditioner $P^{[k]}$ analyzed in [1], for accelerating the convergence of the Gauss-Seidel method applied to symmetric linear systems with the property of preserving symmetry. Both, point and block versions of the symmetric preconditioner have been implemented. Numerical experiments carried out on matrices obtained from different discretization of two and three dimensional elliptic problems show lower iteration counts for the new preconditioner compared to those required by the non-symmetric recursive preconditioner $P^{[k]}$ for both, point and block versions. However, if memory is a concern, $S^{[k]}$ requires more memory than $P^{[k]}$.

References

[1] Arenas I., Castillo P., and Yong X., "An extension of the $I + S_{max}$ preconditioner for the Gauss-Seidel method", *Rev. Integr. Temas Mat.* 31 (2013), no. 1, 1–14.

[2] Castillo P., "Performance of discontinuous Galerkin methods for elliptic PDEs", *SIAM J. Sci. Comput.* 24 (2002), no. 2, 524–547.

[3] Castillo P., Cockburn B., Perugia I., and Schötzau D., "An a priori error analysis of the Local Discontinuous Galerkin method for elliptic problems", *SIAM J. Numer. Anal.*, 38 (2000), no. 5, 1676–1706.

[4] Castillo P. and Sequeira F., "Computational aspects of the Local Discontinuous Galerkine method on unstructured grids in three dimensions", *Math. Comput. Modelling* 57 (2013), no. 9-10, 2279–2288.

[5] Durlofsky L-J., "Accuracy of mixed and control volume finite element approximations to Darcy velocity and related quantities", *Water Resources Research* 30 (1994), no. 4, 965–973.

[6] Kohno T., Kotakemori H., Niki H., and Usui M., "Improving the modified Gauss-Seidel method for Z-matrices", *Linear Algebra Appl.* 267 (1997), 113–123.

[7] Kotakemori H., Harada K., Morimoto M., and Niki H., "A comparison theorem for the iterative method with the preconditioner $(I + S_{max})$", *J. Comput. Appl. Math.* 145 (2002), no. 2, 373–378.

[8] Kotakemori H., Niki H., and Okamoto N., "Accelerated iterative method for Z-matrices", *J. Comput. Appl. Math.* 75 (1996), no. 1, 87–97.

[9] Liu Q., Chen G., and Cai J., "Convergence analysis of the preconditioned Gauss-Seidel method for H-matrices", *Comput. Math. Appl.* 56 (2008), 2048–2053.

[10] Milaszewicz J.P., "Improving Jacobi and Gauss-Seidel iterations", *Linear Algebra Appl.* 93 (1987), 161–170.

[11] Niki H., Harada K., Morimoto M., and Sakakihara M., "The survey of preconditioners used for accelerating the rate of convergence in the Gauss-Seidel method", *J. Comput. Appl. Math.*, 2004, no. 164-165, 587–600. Proceedings of the 10th International Congress on Computational and Applied Mathematics.

[12] Shao J-L., Huang T-Z., and Zhang G-F., "Linear system based approach for solving some related problems of M-matrices", *Linear Algebra Appl.* 432 (2010), no. 1, 327–337.

[13] Shen H., Shao X., Huang Z., and Li C., "Preconditioned Gauss Seidel iterative method for Z matrices linear systems", *Bull. Korean Math. Soc.* 48 (2011), no. 2, 303–314.

[14] Zheng B. and Miao S-X., "Two new modified Gauss Seidel methods for linear systems with M-matrices", *J. Comput. Appl. Math.* 233 (2009), 922–930.