A Class of Efficient Preconditioners with Multilevel Sequentially Semiseparable Matrix Structure

Yue Qiu*, Martin B. van Gijzen†, Jan-Willem van Wingerden* and Michel Verhaegen*

*Delft Center for System and Control, Delft University of Technology, 2628CD Delft, the Netherlands
†Delft Institute of Applied Mathematics, Delft University of Technology, 2628CD Delft, the Netherlands

Abstract. This paper presents a class of preconditioners for sparse systems arising from discretized partial differential equations (PDEs). In this class of preconditioners, we exploit the multilevel sequentially semiseparable (MSSS) structure of the system matrix. The off-diagonal blocks of MSSS matrices are of low-rank, which enables fast computations of linear complexity. In order to keep the low-rank property of the off-diagonal blocks, model reduction algorithm is necessary. We tested our preconditioners for 2D convection-diffusion equation, the computational results show the excellent performance of this approach.

Keywords: preconditioners, partial differential equations (PDEs), multilevel, sequentially semiseparable matrices
PACS: 15-06, 15A06, 15A23, 15B99

INTRODUCTION

Many simulations in science and engineering design [1] [2] require the solution of a large sparse system of the form

$$Ax = b\ , \tag{1}$$

where $A = [a_{ij}]$ is an $n \times n$ matrix and b is a given right-hand-side vector of compatible dimension. Often this is the most time-consuming part of a simulation. If such systems originate from discretized partial differential equations (PDEs), they are usually sparse and of large size. Many efforts have been dedicated to find efficient solution methods for such systems. Some of the most popular iterative solution methods are the conjugate gradient (CG), minimal residual (MINRES), generalized minimal residual (GMRES) and induced dimension reduction (IDR(s)) methods [3] [4] [5]. The performance of these methods highly depends on the choice of the preconditioners. In this paper, we study a new class of preconditioners based on the multilevel sequentially semiseparable matrix structure of the system [6].

Semiseparable matrices appear in several types of applications, e.g. the field of integral equations [7], Gauss-Markov processes [8], boundary value problems [9] and rational interpolation [10]. Semiseparable matrices are defined in [11] as matrices of which all sub-matrices taken from the lower-triangular or upper-triangular part are of rank 1 . The generalization of semiseparable matrices, called quasiseparable matrices, are matrices of which all sub-matrices extracted from the strictly lower-triangular or strictly upper-triangular part are of rank 1 [11], where semiseparable plus diagonal matrices are of such type. This more general type of matrices is called sequentially semi-separable (SSS) by Dewilde et al [12]. For this class, the off-diagonal blocks are of low-rank, but not necessarily limited to 1. The property of low-rank off-diagonal blocks is also investigated by Eidelman et al [13]. They also call this general class of matrices quasiseparable matrices. The class of SSS matrices is closed under basic operations as addition, multiplication and inversion. Moreover, decompositions/factorizations such as the QR [14], LU/LDU [11] [15], and ULV factorizations [16] [17] can also be computed in a structure preserving way. Many operations on SSS matrices can be performed with linear computational complexity. Examples are the matrix-matrix multiplication, the matrix-vector multiplication, matrix inversion, and the QR, LU and ULV factorizations [18] [16] [14].

Systems that arise from the discretisation of 1D partial differential equations typically have an SSS structure. Discretisations of higher dimensional (2D or 3D) partial differential equations give rise to matrices that have a so-called multi-level SSS structure [15]. For such MSSS matrices, operations like inversion and decomposition can not be performed with linear complexity. However, as discussed in [15], and as we will show in this paper, it is possible to compute an inexact LU decomposition of an MSSS matrix that can be used as a preconditioner. By numerical experiments we will show that this preconditioner can be computed at linear complexity and that the number of iterations is almost independent of the mesh size.

11th International Conference of Numerical Analysis and Applied Mathematics 2013
AIP Conf. Proc. 1558, 2253-2256 (2013); doi: 10.1063/1.4825988

The outline of this paper is as follows. Section 2 introduces multilevel sequentially semiseparable matrices and explains their relation with discretized PDEs. Section 3 discussed the MSSS preconditioner and illustrates its performance on two convection-diffusion problems. The last section gives the conclusions.

MULTILEVEL SEQUENTIALLY SEMISEPARABLE MATRICES

The matrices in this paper will always be real and their dimensions are compatible for matrix- matrix addition, multiplication and matrix-vector multiplication when their sizes are not mentioned. To keep this paper self-contained, we review some definitions and concepts, see also [6]. The generators representation of sequentially semiseparable matrices are defined by Definition 1.

Definition 1. *Let A be an $N \times N$ matrix with the SSS structure. Let $m_1, m_2, \cdots m_n$ be positive integers with $N = m_1 + m_2 + \cdots + m_n$ such that A can be written in the following block-partitioned form:*

$$A_{ij} = \begin{cases} U_i W_{i+1} \cdots W_{j-1} V_j^T, & i < j; \\ D_i, & i = j; \\ P_i R_{i-1} \cdots R_{j+1} Q_j^T, & i > j \end{cases} \tag{2}$$

where superscript 'T' denotes the transpose of the matrix. The above representation of A is called the generators representation. The sequences $\{U_i\}_{i=1}^{n-1}$, $\{W_i\}_{i=2}^{n-1}$, $\{V_i\}_{i=2}^{n}$, $\{D_i\}_{i=1}^{n}$, $\{P_i\}_{i=2}^{n}$, $\{R_i\}_{i=2}^{n-1}$, $\{Q_i\}_{i=1}^{n-1}$ are matrices whose sizes are listed in table 1 and they are called generators of the SSS matrix A.

TABLE 1. Size of the generators of A

Generators	U_i	W_i	V_i	D_i	P_i	R_i	Q_i
Sizes	$m_i \times k_i$	$k_{i-1} \times k_i$	$m_i \times k_{i-1}$	$m_i \times m_i$	$m_i \times l_i$	$l_{i-1} \times l_i$	$m_i \times l_{i+1}$

With the generators parametrization, the SSS matrix A can be expressed by its generators and denoted as

$$A = \mathscr{S}\mathscr{S}\mathscr{S}(P_s, R_s, Q_s, D_s, U_s, W_s, V_s). \tag{3}$$

Take $n = 4$ for example, the SSS matrix A is shown by (4),

$$\begin{bmatrix} D_1 & U_1 V_2^T & U_1 W_2 V_3^T & U_1 W_2 W_3 V_4^T \\ P_2 Q_1^T & D_2 & U_2 V_3^T & U_2 W_3 V_4^T \\ P_3 R_2 Q_1^T & P_3 Q_2^T & D_3 & U_3 V_4^T \\ P_4 R_3 R_2 Q_1^T & P_4 R_3 Q_2^T & P_4 Q_3^T & D_4 \end{bmatrix} \tag{4}$$

Using this representation, the structure of SSS matrices can be exploited to do fast computation for matrix addition, multiplication and inversion by operations on its generators, with the cost of linear computational complexity. Table 2 lists references to papers that discuss how a given matrix operation can be performed using SSS matrix arithmetic.

TABLE 2. Commonly used operations on SSS matrices

operations	$A*x$	$A \pm B$	$A*B$	A^{-1}	LU	Model Reduction	$Lx = b^*$
references	[12] [13] [18]	[12] [13] [18]	[12] [13] [18]	[13]	[6] [11] [15]	[6] [18]	[18]

* L is a lower triangular SSS matrix

Remark 1. *With the operations listed in table 2, it is always possible to construct a preconditioner and solve the preconditioned system with linear complexity.*

Analogously to SSS matrices are defined in [15] by Definition 2.

Definition 2. *The matrix A is said to be a k-level MSSS matrix where all its generators are $(k-1)$-level MSSS matrices. The* 1*-level MSSS matrix is the SSS matrix that satisfies Definition 1.*

Example 1. *Any banded matrix is automatically semiseparable according to [11]. Similarly, every block banded matrix is multilevel sequentially semiseparable (MSSS). Take for example the discretized 2D Laplace equation with homogeneous Dirichlet boundary conditions by finite-element discretization on a regular grid, the stiffness matrix K is given by* $K = \begin{bmatrix} D & F & & & \\ F & D & F & & \\ & F & \ddots & \ddots & \\ & & \ddots & & F \\ & & & F & D \end{bmatrix}$*, where* $D = \begin{bmatrix} 4 & -1 & & & \\ -1 & 4 & -1 & & \\ & -1 & \ddots & & \\ & & & \ddots & -1 \\ & & & -1 & 4 \end{bmatrix}$ *and* $F = -I$ *where I is the identity matrix. The matrices D and F have the SSS matrix structure and can be denoted as*

$$D = \mathscr{S}\mathscr{S}\mathscr{S}(1,\ 0,\ -1,\ 4,\ 1,\ 0,\ -1) \text{ and } F = \mathscr{S}\mathscr{S}\mathscr{S}(0,\ 0,\ 0,\ -1,\ 0,\ 0,\ 0). \tag{5}$$

Meanwhile, the matrix K has the (2-level) MSSS matrix structure and can be denoted by

$$K = \mathscr{M}\mathscr{S}\mathscr{S}\mathscr{S}(I,\ 0,\ F,\ D,\ I,\ 0,\ F). \tag{6}$$

MSSS PRECONDITIONERS

In this section we will describe how the MSSS structure of a matrix can be exploited to construct an efficient preconditioner. We will illustrate the approach with experiments on a convection-diffusion problem.

To construct a preconditioner, we factorize the system matrix with MSSS matrix techniques. The decomposition method proposed in [15] results in a growth of the rank of the generators. As a result, the decomposition of an MSSS matrix is not of order N complexity. The solution proposed in [15] is to use a model order reduction algorithm to reduce the rank of the sub-blocks, to keep the decomposition operation of order N. However, as a results of the model order reduction, the decomposition is only approximate. The model reduction algorithm plays a key rule in MSSS arithmetic. In [15] Gondzio et al used the method in [19] to solve the PDE-constrained optimization problem. We use a novel model reduction algorithm described in [6] (available at https://www.dropbox.com/sh/pj6gqw6pxzn69qz/TXyboPQEhy) to construct an efficient preconditioner for the 2D convection-diffusion problem. The idea of the novel model reduction algorithm is to use the low-rank property of the controllability and observability gramians of linear time-varying (LTV) systems to approximate the gramians with low numerical rank. With the approximate gramians of LTV systems, it is possible to perform the balanced truncation to get a LTV system of reduced order, which corresponds to the reduced generators of an SSS matrix. The model reduction in [19] is to approximate the Hankel blocks of an SSS matrix. Compared with this model reduction algorithm, the novel model reduction algorithm needs less floating point operations. Meanwhile, the novel model reduction algorithm leads to satisfied accuracy. Numerical experiments show the efficiency of our method.

Example 2. *Let* $\Omega = [0,\ 1]^2$ *and consider the problem*

$$-\varepsilon\nabla^2 u + \overrightarrow{\omega}.\nabla u = 0 \text{ in } \Omega \tag{7}$$

with $u = u_D$ *on* Ω_D *and* $\frac{\partial u}{\partial n} = u_N$ *on* Ω_N *where* $u_D = u_N = (2x-1)^2(2y-1)^2$, $\Omega_D = \{0\} \times [0,\ \frac{1}{2}]$, $\Omega_N = [0,\ \frac{1}{2}] \times \{0\}$ *and* ε *is a positive scalar,* $\overrightarrow{\omega}$ *is the unit directional vector,* $\overrightarrow{\omega} = (cos(\theta),\ sin(\theta))^T$.

The discretized 2D convection-diffusion equation yields a nonsymmetric linear system, which we solve with the induced dimension reduction (IDR(s)) method in combination with the MSSS preconditoner. The experiments have been performed on a laptop with Intel Core 2 Duo 2.4 GHZ and 4Gb memory using Matlab R2010b. The parameters are set to $\theta = \frac{\pi}{5}$, $s = 4$, the maximum semiseparable order is listed in the brackets followed after the mesh size and the computational results for different ε are shown in table 3 and 4

Remark 2. *Table 3 and 4 show that the number of iterations is almost independent of the mesh size, and that the computing time of both the computation of the preconditioner and of the iterative solution increases linearly with the number of equations.*

TABLE 3. Number of iterations and time of preconditioning and IDR(s) solver when $\varepsilon = 10^{-1}$

mesh size h	iterations	Preconditioning (seconds)	IDR(s) (seconds)	total (seconds)
$2^{-5}(3)$	4	0.56	0.20	0.76
$2^{-6}(3)$	4	1.91	0.36	2.27
$2^{-7}(4)$	4	6.34	0.77	7.11
$2^{-8}(4)$	4	23.91	2.09	26.00
$2^{-9}(4)$	5	92.08	9.95	102.03

TABLE 4. Number of iterations and time of preconditioning and IDR(s) solver when $\varepsilon = 10^{-2}$

mesh size h	iterations	Preconditioning (seconds)	IDR(s) (seconds)	total (seconds)
$2^{-5}(3)$	2	0.65	0.16	0.81
$2^{-6}(3)$	3	1.99	0.29	2.28
$2^{-7}(3)$	3	5.97	0.67	6.64
$2^{-8}(4)$	4	23.27	1.71	24.98
$2^{-9}(4)$	4	90.05	6.36	96.41

CONCLUSIONS

In this paper, we have exploited the MSSS structure of the system matrix to construct an efficient preconditioner. The MSSS structure has low-rank off-diagonal blocks, which enables fast computations of linear complexity. In order to keep the off-diagonal blocks of low-rank, a novel model reduction algorithm is used. Numerical experiment of 2D convection-diffusion problem shows the efficiency of this approach. This approach can be extended to preconditioning for more complicated 3D problems. Moreover, our method is also efficient for preconditioning of PDE-constrained optimization problems, we refer [6] for further results.

REFERENCES

1. L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, *Real-time PDE-constrained Optimization*, Society for Industrial and Applied Mathematics, 2007.
2. T. Rees, H. Dollar, and A. Wathen, *SIAM Journal on Scientific Computing* **32**, 271–298 (2010).
3. G. Golub, and C. Van Loan, *Matrix computations*, Johns Hopkins University Press, 1996.
4. Y. Saad, *Iterative methods for sparse linear systems*, Society for Industrial and Applied Mathematics, 2003.
5. P. Sonneveld, and M. van Gijzen, *SIAM Journal on Scientific Computing* **31**, 1035–1062 (2008).
6. Y. Qiu, M. van Gijzen, J. van Wingerden, and M. Verhaegen, Efficient preconditioners for PDE-constrained optimization problems with a multi-level sequentially semi-separable matrix structure, Tech. Rep. 13-04, Delft Institution of Applied Mathematics, Delft University of Technology (2013), URL `https://www.dropbox.com/sh/pj6gqw6pxzn69qz/TXyboPQEhy`.
7. R. Gonzales, J. Eisert, I. Koltracht, M. Neumann, and G. Rawitscher, *Journal of Computational Physics* **134**, 134–149 (1997).
8. A. Kavcic, and J. Moura, *IEEE Transactions on Information Theory* **46**, 1495–1509 (2000), ISSN 0018-9448.
9. L. Greengard, and V. Rokhlin, *Communications on Pure and Applied Mathematics* **44**, 419–452 (1991).
10. M. Van Barel, D. Fasino, L. Gemignani, and N. Mastronardi, "Orthogonal rational functions and diagonal-plus-semiseparable matrices," in *International Symposium on Optical Science and Technology*, 2002, pp. 162–170.
11. R. Vandebril, M. Van Barel, and N. Mastronardi, *Matrix computations and semiseparable matrices: linear systems*, Johns Hopkins University Press, 2007.
12. S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, X. Sun, A. van der Veen, and D. White, *SIAM Journal on Matrix Analysis and Applications* **27**, 341–364 (2005).
13. Y. Eidelman, and I. Gohberg, *Integral Equations and Operator Theory* **34**, 293–324 (1999).
14. Y. Eidelman, I. Gohberg, and V. Olshevsky, *Linear Algebra and Its Applications* **404**, 305–324 (2005).
15. J. Gondzio, and P. Zhlobich, *arXiv preprint arXiv:1112.6018* (2011).
16. S. Chandrasekaran, M. Gu, and T. Pals, *SIAM Journal on Matrix Analysis and Applications* **28**, 603–622 (2006).
17. A. van der Veen, *Time-Varying System Theory and Computational Modeling*, Ph.D. thesis, Delft University of Technology (1993).
18. S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, and A. van der Veen, Fast stable solvers for sequentially semi-separable linear systems of equations, Tech. rep., Lawrence Livermore National Laboratory (2003).
19. Y. Eidelman, and I. Gohberg, *Calcolo* **42**, 187–214 (2005).