ARCHIVIO ISTITUZIONALE
DELLA RICERCA

Alma Mater Studiorum Università di Bologna
Archivio istituzionale della ricerca

On the numerical solution of a class of systems of linear matrix equations

This is the final peer-reviewed author's accepted manuscript (postprint) of the following publication:

Published Version:
Simoncini V. (2020). On the numerical solution of a class of systems of linear matrix equations. IMA JOURNAL OF NUMERICAL ANALYSIS, 40(1), 207-225 [10.1093/imanum/dry083].

Availability:
This version is available at: https://hdl.handle.net/11585/784046 since: 2021-02-28

Published:

DOI: http://doi.org/10.1093/imanum/dry083

Terms of use:

Some rights reserved. The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/).
When citing, please refer to the published version.

(Article begins on next page)

08 October 2026

This is the final peer-reviewed accepted manuscript of:

Valeria Simoncini, On the numerical solution of a class of systems of linear matrix equations, IMA Journal of Numerical Analysis, Volume 40, Issue 1, January 2020, Pages 207–225

The final published version is available online at ***https://dx.doi.org/10.1093/imanum/dry083***

Rights / License:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. For all terms of use and more information see the publisher's website.

This item was downloaded from IRIS Università di Bologna (https://cris.unibo.it/)

When citing, please refer to the published version.

ON THE NUMERICAL SOLUTION OF A CLASS OF SYSTEMS OF LINEAR MATRIX EQUATIONS *

V. SIMONCINI†

Abstract. We consider the solution of systems of linear matrix equations in two or three unknown matrices. For dense problems we derive algorithms that determine the numerical solution by only involving matrices of the same size as those in the original problem, thus requiring low computational resources. For large and structured systems, we show how the problem properties can be exploited to design effective algorithms with low memory and operation requirements. Numerical experiments illustrate the performance of the new methods.

Key words. Linear matrix equations. Large scale equations. Schur complement. Sylvester equation.

AMS subject classifications. 65F10, 65F30, 15A06

1. Introduction.

We are interested in solving

$$\begin{aligned} A_1\mathbf{X} + \mathbf{X}A_2 + B^T\mathbf{P} &= F_1, \\ B\mathbf{X} &= F_2 \end{aligned} \tag{1.1}$$

where $A_i \in \mathbb{R}^{n_i \times n_i}$, $B \in \mathbb{R}^{m \times n_1}$ are the coefficient matrices, and F_1, F_2 are matrices of conforming dimensions. The matrices $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2}$ and $\mathbf{P} \in \mathbb{R}^{m \times n_2}$ are to be determined. In the following we shall use boldface to denote unknown (solution) matrices to be numerically computed or approximated. We assume that A_1 and $-A_2$ have disjoint spectra, ensuring that the Sylvester operator $X \mapsto A_1X + XA_2$ is invertible. Moreover, we assume that the full rank matrix B is "fat", that is it has more columns than rows, with m and n_1 of the same order of magnitude. We stress the fact that B is rectangular: the problem would be characterized by quite different properties if all coefficient matrices were square (and nonsingular). In fact, the square setting has been largely explored in the literature, both in terms of algebraic properties and computational procedures, because of its connection with invariant subspace computations; see, e.g., [5],[4],[31],[11], and references therein. Note however, that most known articles deal with the small scale case. Moreover, in the large majority of cases in the literature, one matrix term per unknown in each matrix equation is considered; see, e.g., [29]. Hence, in our setting the presence of the Sylvester operator provides additional complexity to the problem, while being encountered in applications; see below and section 2.

The system (1.1) can be generalized so as to consider three matrix equations and three unknown matrices, that is[1]

$$\begin{aligned} A_2\mathbf{X} + \mathbf{X}A_1^T + B_1^T\mathbf{P} &= F_1 \\ A_1\mathbf{Y} + \mathbf{Y}A_2^T + \mathbf{P}B_2 &= F_2 \\ B_1\mathbf{X} + \mathbf{Y}B_2^T &= F_3. \end{aligned} \tag{1.2}$$

*Version of October 8, 2018. Part of this work was supported by the Indam-GNCS 2017 Project "Metodi numerici avanzati per equazioni e funzioni di matrici con struttura". The author is a member of the GNCS-Indam activity group.

†Dipartimento di Matematica, Alma Mater Studiorum Università di Bologna, Piazza di Porta San Donato 5, I-40127 Bologna, Italy (valeria.simoncini@unibo.it), and IMATI-CNR, Pavia.

[1]Note that here A_1 and A_2 denote generic square matrices, and are not necessarily related to A_1 and A_2 in (1.1).

Here $B_1, B_2 \in \mathbb{R}^{n_1 \times n_2}$, $F_1 \in \mathbb{R}^{n_2 \times n_1}$, $F_2 \in \mathbb{R}^{n_1 \times n_2}$ and $F_3 \in \mathbb{R}^{n_1 \times n_1}$, so that $\mathbf{X} \in \mathbb{R}^{n_2 \times n_1}$, $\mathbf{Y} \in \mathbb{R}^{n_1 \times n_2}$ and $\mathbf{P} \in \mathbb{R}^{n_1 \times n_1}$.

Many application problems can be naturally formulated in the matrix setting described in (1.1) or (1.2). For instance, coupled matrix equations naturally arise in the discretization of certain systems of partial differential equations (PDEs) in the deterministic, stochastic or constrained settings, or in the control analysis of time-invariant linear dynamical systems. In section 2 we briefly describe some of these classes of problems.

Possibly for the sake of generality, little attention has been given to the matrix equation form in the past, especially in the PDE literature. The associated vector form has been preferred. Indeed, by using the Kronecker product the two matrix equations in (1.1) can be rewritten as the standard (vector) system $\mathcal{M}\mathbf{u} = b$ with

$$\begin{bmatrix} \mathcal{A} & \mathcal{B}^T \\ \mathcal{B} & \mathcal{O} \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \quad \mathcal{A} = I \otimes A_1 + A_2^T \otimes I, \quad \mathcal{B} = I \otimes B, \tag{1.3}$$

while $\mathbf{x} = \mathrm{vec}(\mathbf{X})$, $\mathbf{p} = \mathrm{vec}(\mathbf{P})$, $f_i = \mathrm{vec}(F_i)$, $i = 1, 2$, where vec is the usual function that stacks all matrix columns one after the other to form a long vector. We shall refer to this linear system as the *monolithic* equation. The advantages of (1.3) are clear: a rich literature can be employed to solve the linear system, see, e.g., [28], while the saddle point structure of $\mathcal{M}$ can be exploited to devise effective preconditioners if iterative methods are used; see, e.g., [2]. From a theoretical point of view, the Kronecker form ensures nonsingularity conditions on $\mathcal{M}$ based on the original coefficient matrices, and thus existence and uniqueness of the problem solution. Unfortunately, disadvantages are enormous: the dimension of the system in (1.3) becomes huge, as $\mathcal{A}$ has size $n_1 n_2 \times n_1 n_2$, and the same for $\mathcal{B}$. Sparse direct methods explicitly applied to $\mathcal{M}$ are unlikely to be able to exploit the Kronecker structure, while iterative methods require storing a certain number of extra working vectors of length $n_2 n_1 + m n_1$. In this paper we go further in this argumentation. We show that numerical methods that directly attack the matrix formulation, as opposed to the Kronecker form, not only use memory allocations in a more sober manner, but can also be much faster, especially if the structure is taken into account.

The idea of focussing on the matrix form of a given problem has recently proven to be very effective in various contexts. For instance, linear convection-diffusion PDEs with separable coefficients and elliptic PDEs with random inputs can be rewritten as generalized Sylvester linear matrix equations with sparse data; see, e.g., [18],[21]. Rank structure of the problem can also be exploited. We refer the reader to [26] for a discussion of typical application problems that can be conveniently put into a matrix form. Here we extend this idea to include systems of matrix equations in more than one unknown matrix. We focus on the case of two or three matrix equations with one or two matrix terms per unknown, as they arise in application problems associated with PDEs. We are unaware of numerical procedures that can effectively and explicitly handle (1.1) or (1.2) when the problem size n_i, $i = 1, 2$ is larger than 1000. The homogeneous case of (1.1), that is with $F_1 = 0, F_2 = 0$, was treated for instance in [24]. This reference will be our starting point for one of the proposed methods.

The recent literature discusses systems with a general (large) number of matrix equations, for which different classes of approaches have to be considered; in [32] for instance, a gradient based iterative algorithm was employed, that minimizes a linear combination of the squared residual norm of each matrix equation. As already

mentioned, however, most efforts in the literature focus on a single term per unknown, with all square matrices; see, e.g., [13] and references therein.

In section 4 we discuss a nullspace based method for (1.1) that is particularly efficient whenever B has small to medium dimensions, so that a dense QR decomposition is feasible; the method can be implemented for either small or large A_2. In section 5 we derive an iterative method for (1.1) that is able to handle the large scale case for all coefficient matrices, when the right-hand sides have a low rank structure. Finally, in section 7 we discuss a Schur complement based method that is tailored towards the three matrix equations in (1.2); although the approach is also applicable to the problem (1.1), it is not competitive with respect to the previously discussed methods for that problem, whereas it is particularly efficient for the three matrix equation case.

All experiments were performed using Matlab [16] on a Dell computer with an Intel Core processor i7-3687U, with four CPUs at 2.10GHz.

2. Examples of applications. Many application problems can be formulated by means of one of the matrix equation systems above. Here we provide a few examples.

2.1. Regulator equations in constraint control. Systems of matrix equations (1.1) can arise in tracking and regulation, or in the robust model-reference control of continuous-time systems, see, e.g., [15, Th.2.6], [30, Chapter 8],[7],[8],[22, Chapter 2]. For instance, the dynamical system may represent a controllable plant subjected to step disturbances at a regulated output. Indeed, let us consider the following time-invariant dynamical linear system[2]

$$\dot{x} = Ax + Bu, \quad y = Cx, \tag{2.1}$$

while the reference model is written as $\dot{x}_m = A_m x_m + B_m u_m$, $y_m = C_m x_m$. Then the following result holds.

THEOREM 2.1. ([8]) *Assume that there exists a stabilizing gain matrix K for the system in (2.1), and that $\mathbf{X}, \mathbf{P}$ satisfy the equations*

$$A\mathbf{X} + \mathbf{X}A_m + B\mathbf{P} = 0, \quad C\mathbf{X} = C_m. \tag{2.2}$$

Let us also define $Q = -(B^TB)^{-1}B^T\mathbf{X}B_m$. When the controller $u = u_s + u_c$ with $u_s = Kx$ and $u_c = (\mathbf{P} - K\mathbf{X})x_m - Qu_m$ is applied to the unperturbed system, it holds that $\lim_{t\to\infty}(y(t) - y_m(t)) = 0$.

For $C = B^T$ the form in (1.1) is obtained.

2.2. Mixed FE formulation of the stochastic Galerkin diffusion problem. Let us consider the stochastic steady-state diffusion problem $-\nabla \cdot (c\nabla p) = f$ with zero boundary conditions; here $p = p(x, y)$ is the displacement, with $(x, y) \in D$ where D is a spatial domain, and $c : D \times \Omega \to \mathbb{R}$ is the diffusion coefficient, depending on the sample space Ω. In case the *flux* $\vec{u} := c\nabla p$ is explicitly of interest, the problem can be restated in mixed form as follows

$$c^{-1}\vec{u} - \nabla p = 0, \quad \text{in} \quad D \times \Omega \tag{2.3}$$

$$-\nabla \cdot \vec{u} = f, \quad \text{in} \quad D \times \Omega \tag{2.4}$$

$$p = 0, \quad \text{on} \quad \partial D \times \Omega. \tag{2.5}$$

[2]For the sake of a streamlined presentation a simplified model is considered.

Assume c^{-1} can be written as a truncated Karhunen-Loève expansion, that is $c^{-1} = c_0 + \sum_{r=1}^{m} \sqrt{\lambda_r} c_r(\vec{x})\xi_r(\omega)$, where ξ_r are properly selected random variables; see, e.g., [14, section 9.3]. Assume then that an appropriate class of finite elements is used for the discretization of the problem; here we follow the derivation in [9]. Then after discretization the problem reads

$$\begin{bmatrix} G_0 \otimes K_0 + \sum_{r=1}^{m} \sqrt{\lambda_r} G_r \otimes K_r & G_0^T \otimes B_0^T \\ G_0 \otimes B_0 & \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} 0 \\ f \end{bmatrix}.$$

The number m of terms in the sum is usually computed a priori, taking into account the eigenvalue decay of the covariance function of c^{-1} [14, Chapter 9]. For $m = 1$, that is assuming fast decaying eigenvalues, we obtain a pair of matrix equations very similar to the one in (1.1), that is

$$K_0 \mathbf{X} G_0 + K_1 \mathbf{X} G_1 + B_0^T \mathbf{P} G_0 = 0, \qquad B_0 \mathbf{X} G_0 = F,$$

where the operator $\mathbf{X} \mapsto K_0 \mathbf{X} G_0 + K_1 \mathbf{X} G_1$ is a generalized form of the Sylvester operator seen before.

2.3. Matrix formulation of discretized Stokes and Navier-Stokes equations. We consider the following two-dimensional steady-state Navier-Stokes equation system,

$$\begin{aligned} -\nu \Delta \vec{u} + (\vec{u} \cdot \nabla)\vec{u} + \nabla p &= \vec{f} \\ \nabla \cdot \vec{u} &= 0 \end{aligned} \quad \text{in } \Omega \tag{2.6}$$

with appropriate boundary conditions on $\partial\Omega$, where Ω is an open bounded set of $\mathbb{R}^2$. For simplicity of exposition in the following we will assume that Ω is the open unit square. The constant $\nu > 0$ is the viscosity parameter. Coordinate-wise with $\vec{u} = (u, v)$ this reads

$$-\nu \Delta u + u(u_x) + v(u_y) + \partial_x p = f_1 \tag{2.7}$$

$$-\nu \Delta v + u(v_x) + v(v_y) + \partial_y p = f_2 \tag{2.8}$$

$$\partial_x u + \partial_y v = 0. \tag{2.9}$$

A finite difference discretization of the linearized version of these equations - the Oseen equations - on a 2D staggered square grid (MAC scheme, see, e.g., [10]) leads to the standard saddle point type linear system

$$\begin{bmatrix} F_u & & B_x^T \\ & F_v & B_y^T \\ B_x & B_y & \end{bmatrix} \begin{bmatrix} u \\ v \\ p \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}, \tag{2.10}$$

where n^2 pressure unknown at cell centers are used, and $n(n-1)$ velocity unknowns at each horizontal/vertical cell side are introduced. The diagonal blocks discretize the diffusion operators as

$$F_u = A_1 \otimes I_{n-1} + I_n \otimes A_2, \quad F_v = A_2 \otimes I_n + I_{n-1} \otimes A_1.$$

In the case of constant nonzero wind coefficients, the matrices A_1 and A_2 also contain the discretization of the convection operator. More generally, for non-constant but

separable wind coefficients, certain matrices replace the identity matrices in F_u and F_v (see, e.g., [18]). The discretization of the divergence operator gives

$$B_x = I_n \otimes B, \quad B_y = B \otimes I_n,$$

where $B \in \mathbb{R}^{n\times(n-1)}$; we refer to [10] for a more detailed description of the problem discretization. By unfolding the Kronecker form of these matrices we obtain the three matrix equations in (1.2). Neglecting the non-linear term in (2.7) yields the Stokes problem, whose MAC discretization follows the same lines.

The discretization of the biharmonic equation also leads to a similar 3×3 system, see, e.g., [25, equation (1.14)], where the first two diagonal blocks refer to the mass matrix at interior and boundary nodes, respectively, while the non-diagonal elements correspond to stiffness matrices, with a possible Kronecker structure. The discretization of certain PDE-constrained optimization problems also lead to systems of equations in Kronecker form, that can be reformulated in matrix terms; see, e.g., [6].

We conclude this section with a general comment on discretization and matrix equations. Several numerical methods can take advantage of the form above when discretizing (systems of) partial differential equations. Indeed, discretizations procedures using tensor bases can appropriately handle separable coefficients in the PDE, so as to guarantee a matrix formulation of the discretized problem. This may occur for instance in finite differences with transfinite grid interpolation type methods [12], Isogeometric Analysis methodologies (see, e.g., [23]), and in certain spectral and finite volume methods (see, .e.g, [3, Polynomial approximation chapter]). Most of these strategies map the original 2D domain onto a rectangle, after which the matrix discretization is directly applicable to separable coefficient PDEs; see, e.g., the discussion in [26, Section 3].

3. Solvability conditions. We mentioned in the introduction that the saddle point formulation of the system of two equations allows one to give sufficient conditions for the nonsingularity of the coefficient matrix $\mathcal{M}$ in (1.3) and thus for the solvability of the associated linear system. Indeed, let H be the symmetric part of $\mathcal{A}$ in (1.3), that is $H = (\mathcal{A} + \mathcal{A}^T)/2$. If $\ker(H) \cap \ker(\mathcal{B}) = \{0\}$ then $\mathcal{M}$ is invertible; see, e.g., [2, Theorem 3.4]. As a special situation, consider the case where $\mathcal{A}$ is symmetric and nonsingular, so that $H = \mathcal{A}$, and let the columns of U_0 span the null space of B. Then the columns of $\mathcal{U}_0 = I \otimes U_0$ span the null space of $\mathcal{B}$. Therefore, the condition $\ker(H) \cap \ker(\mathcal{B}) = \{0\}$ corresponds to saying that $\mathcal{U}_0^T \mathcal{A} \mathcal{U}_0$ is nonsingular. With the aid of the properties of the Kronecker product, this latter quantity can be written in terms of the original matrices as

$$\mathcal{U}_0^T \mathcal{A} \mathcal{U}_0 = I \otimes U_0^T A_1 U_0 + A_2^T \otimes I. \tag{3.1}$$

Hence this matrix is nonsingular if and only if the spectra of $U_0^T A_1 U_0$ and $-A_2$ are disjoint. This hypothesis will be assumed throughout the rest of the paper. The above nonsingularity condition is fulfilled, for instance, if A_1 and A_2 have their field of values in the same half complex plane. Weaker assumptions can be considered.

Solvability conditions for the system of three matrix equations can be derived accordingly, by using for instance the Kronecker formulation in (2.10).

4. Null space method. The first method we consider is aimed at the coupled system (1.1), and uses dense methods, based on factorizations. Hence, the described procedure is appealing when dealing with small, possibly dense coefficient matrices.

More precisely, sizes of the order of up to a few hundreds may be appropriate for this strategy.

In the computational literature, the term "nullspace method" is often associated to one of the possible strategies for solving saddle point linear systems in (1.3); see, e.g., [2, Section 6]. Clearly, any strategy that relies on (explicitly or implicitly) computing a nullspace basis can go under this name. In our context, the nullspace under consideration is that associated with B. If a saddle point matrix in the form (1.3) were considered, then the nullspace method would rely on computing the null space of $\mathcal{B}$. Without further structure information, computing this last null space usually becomes intractable for large dimensions. Taking into account the problem structure both in the generation of the nullspace basis and later in the computation, the way we propose, allows one to drastically limit memory and computational efforts.

Let the orthonormal columns of U_0 span the null space of B, so that $BU_0 = 0$, and let the orthonormal columns of U_1 span the range of B^T. We can thus write $\mathbf{X} = U_0\hat{\mathbf{X}} + U_1\mathbf{X}_\perp$, with $[U_0, U_1]$ orthogonal matrix. Substituting $\mathbf{X}$ into the second matrix equation we obtain

$$\mathbf{X}_\perp = (BU_1)^{-1}F_2.$$

We separately multiply the first matrix equation by U_0^T and also by U_1^T, thus obtaining

$$U_0^T A_1 U_0 \hat{\mathbf{X}} + U_0^T A_1 U_1 \mathbf{X}_\perp + \hat{\mathbf{X}} A_2 = U_0^T F_1 \tag{4.1}$$

$$U_1^T A_1 U_1 \mathbf{X}_\perp + U_1^T A_1 U_0 \hat{\mathbf{X}} + \mathbf{X}_\perp A_2 + U_1^T B^T \mathbf{P} = U_1^T F_1. \tag{4.2}$$

The first matrix equation is a Sylvester equation in $\hat{\mathbf{X}}$,

$$U_0^T A_1 U_0 \hat{\mathbf{X}} + \hat{\mathbf{X}} A_2 = -U_0^T A_1 U_1 \mathbf{X}_\perp + U_0^T F_1. \tag{4.3}$$

Thanks to the discussion in section 3, the two matrices $U_0^T A_1 U_0$ and $-A_2$ have disjoint spectra, so that this equation can be solved for $\hat{\mathbf{X}}$, thus completely determining $\mathbf{X}$ as $\mathbf{X} = U_0\hat{\mathbf{X}} + U_1\mathbf{X}_\perp$. Note that since we assume that the null space of B has small dimension, the matrix $U_0^T A_1 U_0$ will be small. If A_2 has small dimensions as well, then the Sylvester equation can be solved by Schur-based methods such as the Bartels-Stewart algorithm [1]. Otherwise, an iterative procedure can be used, and this is outlined in section 4.1.

The second matrix equation in (4.2) can be used to determine $\mathbf{P}$, that is

$$\mathbf{P} = (U_1^T B^T)^{-1}(U_1^T F_1 - U_1^T A_1 U_0 \hat{\mathbf{X}} - \mathbf{X}_\perp A_2 - U_1^T A_1 U_1 \mathbf{X}_\perp).$$

The whole procedure is summarized in Algorithm 1.

Algorithm 1 (small size A_2)

1. Determine $[[U_1, U_0], R_1] = \mathrm{QR}(B^T)$
   (full QR factorization, with R_1 square $m \times m$ matrix)
2. Compute $\mathbf{X}_\perp$ by solving the system $(BU_1)\mathbf{X}_\perp = F_2$
   (that is, $R_1^T \mathbf{X}_\perp = F_2$)
3. Solve $U_0^T A_1 U_0 \hat{\mathbf{X}} + \hat{\mathbf{X}} A_2 = -U_0^T A_1 U_1 \mathbf{X}_\perp + U_0^T F_1$ for $\hat{\mathbf{X}}$
4. Compute $\mathbf{P}$ by solving the system
   $R_1 \mathbf{P} = U_1^T F_1 - U_1^T A_1 U_0 \hat{\mathbf{X}} - \mathbf{X}_\perp A_2 - U_1^T A_1 U_1 \mathbf{X}_\perp$.
5. Construct $\mathbf{X} = U_0\hat{\mathbf{X}} + U_1\mathbf{X}_\perp$

In case B^T has a large range, the computation of an orthonormal basis U_1 may be too memory consuming. In this case, it may be convenient to generate U_0 separately,

and only determine R_1 with a "Q-less" QR factorization, while implicitly dealing with U_1 as $U_1 = B^T R_1^{-1}$ whenever multiplications with U_1 are needed. Note that the triangular matrix R_1 may be sparse, if B is sparse. The computational cost is driven by the cost of dense matrix-matrix multiplications, which may significantly exceed all other costs, especially if m is significantly smaller than n_1, n_2.

In the case that $F_2 = 0$ the procedure simplifies, since then $\mathbf{X}_\perp = 0$. The first matrix equation in (4.1) thus becomes $A_1 U_0 \hat{\mathbf{X}} + U_0 \hat{\mathbf{X}} A_2 + B^T \mathbf{P} = F_1$. Solving (4.3) with right-hand side $U_0^T F_1$ yields the unique solution $\hat{\mathbf{X}}$, so that $\mathbf{X} = U_0 \hat{\mathbf{X}}$.

REMARK 4.1. *If F_1 is low rank, then depending on the spectral properties of the coefficient matrices in (4.3), the solution $\hat{\mathbf{X}}$, and thus $\mathbf{X}$, may have low numerical rank. In this case, as a consequence, also $\mathbf{P}$ will be low rank. Under these circumstances, the solution matrices can be stored by using far less memory than their dimension would suggest.*

4.1. The nullspace method. Large scale case. Whenever A_2 has large dimensions, the numerical solution of the Sylvester equation (4.3) requires an iterative solver. Various approaches can be considered, which take into account the significantly different sizes of the two coefficient matrices, see, e.g., [26, section 4.3]; here we focus on a projection-type method. The solution matrix $\hat{\mathbf{X}}$ can be approximated as $\hat{\mathbf{X}} \approx \widetilde{\mathbf{X}} W_k^T$, where the columns of W_k span an approximation space of dimension k, with k much smaller than the dimension of A_2. In other words, an approximate solution to $\mathbf{X}$ in (1.1) is sought as $\mathbf{X} \approx U_0 \widetilde{\mathbf{X}} W_k^T + U_1 \mathbf{X}_\perp = U_0 \widetilde{\mathbf{X}} W_k^T + U_1 R_1^{-T} F_2$, for some $\widetilde{\mathbf{X}}$, which is the solution of the original matrix equation projected onto a smaller subspace. The nullspace constraint already performs the basis reduction from the left.

Equation (4.3) is peculiar in that the first coefficient matrix has small dimensions, therefore only the matrix A_2 requires to be reduced. According to [26, section 4.3], and setting $\widetilde{F} = -\mathbf{X}_\perp^T U_1^T A_1^T U_0 + F_1^T U_0$ in (4.3), a typical approximation space is the Krylov subspace $K_k(A_2^T, \widetilde{F}) = \text{range}([\widetilde{F}, A_2^T \widetilde{F}, \ldots, (A_2^T)^{k-1} \widetilde{F}])$, or its rational variants. We omit the algorithmic details for this approach, and instead refer to [26, section 4.3] for a description and for convergence properties.

5. Iterative solution for large scale problems. If the original problem (1.1) has truly large matrix dimensions, then the previous approaches may show high computational costs. On the other hand, the Kronecker formulation becomes completely intractable.

In the following we assume that the real part of the eigenvalues of A_1, A_2 have the same sign; without loss of generality, we can assume that this sign is positive. Moreover, we assume that the right-hand side matrix F below, is low rank. This allows us to use projection type methods, and look for low rank approximate solutions. The case of banded matrices has been recently analyzed in [19].

We remark that our analysis in this section focuses on the two matrix equations in (1.1); the current implementation does not seem to be directly applicable to (1.2).

We first notice that we can rewrite the system of matrix equations as the following generalized Sylvester equation (see also, e.g., [7])

$$\begin{bmatrix} A_1 & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{P} \end{bmatrix} + \begin{bmatrix} I_{n_1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{P} \end{bmatrix} A_2 = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}, \quad \Leftrightarrow \quad \mathcal{M}\mathbf{Z} + \mathcal{D}_0 \mathbf{Z} A_2 = F, \quad (5.1)$$

with coefficient matrices $\mathcal{M}, \mathcal{D}_0 \in \mathbb{R}^{(n_1+m)\times(n_1+m)}$ and $A_2 \in \mathbb{R}^{n_2 \times n_2}$. Note that $\mathcal{D}_0$ is highly singular. We can then transform the equation above into a standard Sylvester

equation, by writing

$$\mathbf{Z}A_2^{-1} + \mathcal{M}^{-1}\mathcal{D}_0\mathbf{Z} = \widehat{F}, \qquad \text{with} \quad \widehat{F} = \mathcal{M}^{-1}FA_2^{-1}, \tag{5.2}$$

and this is the single matrix equation we propose to solve, in place of the two original matrix equations.

Explicit computations show that

$$\mathcal{M}^{-1}\mathcal{D}_0 = \begin{bmatrix} A_1^{-1}(I - B^TS^{-1}BA_1^{-1}) & 0 \\ S^{-1}BA_1^{-1} & 0 \end{bmatrix}, \qquad S = BA_1^{-1}B, \tag{5.3}$$

and in particular, we clearly see that $\mathcal{M}^{-1}\mathcal{D}_0$ is highly singular. Note also that if A_1 is symmetric (and positive definite), then the matrix $A_1^{-1}(I - B^TS^{-1}BA_1^{-1})$ is also symmetric (and positive semidefinite). Since the eigenvalues of $A_1^{-1}(I-B^TS^{-1}BA_1^{-1})$ are contained in the spectral interval of A_1, it follows that for A_1 symmetric and positive definite the standard Sylvester equation in (5.2) admits a unique solution.

An effective strategy for solving Sylvester equations with low rank right-hand side is given by projection onto small dimensional spaces; typically, these are rational or polynomial Krylov subspaces generated by employing the available coefficient matrices. The former are (far) more effective than polynomial spaces as long as solving with shifted versions of the coefficient matrices is affordable; we refer the reader to [26] for a detailed discussion. These *nested* spaces are generated iteratively in a way so that the spaces expand with the iterations, and a better approximation is expected as the iterations proceed. In our setting, we are looking for an approximation $\widetilde{\mathbf{Z}}$ to $\mathbf{Z}$ of the form $\widetilde{\mathbf{Z}} = V_k\mathcal{Z}_kW_k^T$, where the columns of V_k, W_k span distinct subspaces to be determined next.

For a square matrix T and a tall matrix Y, let us introduce the block Krylov subspace

$$K_k(T, Y) := \text{Range}([Y, TY, T^2Y, \ldots, T^{k-1}Y]),$$

where k corresponds to the number of performed iterations. At the ith iteration a new tall matrix $T^{i-1}Y$ is added to the space by using powers of T. This gives rise to a sequence of nested spaces, that is $K_k(T, Y) \subseteq K_{k+1}(T, Y)$, in which $K_k(T, Y)$ has dimension at most kn_Y, where n_Y is the dimension of Range(Y). Computational and theoretical developments in the past decades have shown that Krylov subspaces involving shift-and-invert powers of T may be more effective than polynomial Krylov spaces for solving problems such as matrix function evaluations and matrix equations. These "second generation" Krylov spaces are called *rational* Krylov spaces. We refer the reader to [26] for a discussion on convergence properties of these spaces, together with relevant references. Let us see how to take advantage of these advanced spaces in the selection of W_k and V_k.

Let $\widehat{F} = \widehat{F}_l\widehat{F}_r^T$ be a full rank factorization of $\widehat{F}$. Within the rational Krylov subspace setting, a sometimes particularly effective choice is to have the columns of W_k span the *extended* Krylov subspace

$$\text{Range}(W_k) := K_k\left(A_2^{-T}, A_2^{-T}\widehat{F}_r\right) + K_k\left(A_2^T, \widehat{F}_r\right).$$

This is a specific rational space that involves both powers of A_2^T and A_2^{-T}. Note that Range(W_k) contains Range$(\widehat{F}^T)$=Range$(\widehat{F}_r)$.

The construction of V_k is more involved, if one is willing to use an extended type method: Firstly, the space $K_k((\mathcal{M}^{-1}\mathcal{D}_0)^{-1}, (\mathcal{M}^{-1}\mathcal{D}_0)^{-1}\widehat{F}_l)$ cannot be built, due to the singularity of the matrix $\mathcal{M}^{-1}\mathcal{D}_0$. Following a similar approach developed in [24], we propose to use the "augmented" extended Krylov space

$$\text{Range}(V_k) := K_k(\mathcal{M}^{-1}\mathcal{D}_0, \widehat{F}_l) + K_k\left((\mathcal{M}^{-1}\mathcal{D}_0 + \sigma I)^{-1}, (\mathcal{M}^{-1}\mathcal{D}_0 + \sigma I)^{-1})\widehat{F}_l\right), \quad (5.4)$$

where $\sigma > 0$ is a small parameter that makes the coefficient matrix nonsingular.

Secondly, because of the structure of $\mathcal{M}^{-1}\mathcal{D}_0$ in (5.3), the polynomial part of the space in (5.4) adds new vectors only if the first block of $\widehat{F}_l$ of n_1 rows is nonzero; otherwise, Range(V_k) will only be formed by powers of $(\mathcal{M}^{-1}\mathcal{D}_0 + \sigma I)^{-1}$. Alternatively, one can initialize the iterative procedure by appropriately selecting a starting guess $\mathbf{Z}_0$. Indeed, substituting $\mathbf{Z} = \mathbf{Z}_0 + \mathbf{Z}_\star$ in (5.1) and moving to the right-hand side all terms involving $\mathbf{Z}_0$, the equation to be solved in $\mathbf{Z}_\star$ will have the new right-hand side $F - \mathcal{M}\mathbf{Z}_0 - \mathcal{D}_0\mathbf{Z}_0A_2$. Thus $\mathbf{Z}_0$ can be chosen so that the right-hand side first block is nonzero, while the right-hand side is still low rank. Let us thus continue by assuming that the first block of $\widehat{F}_l$ is nonzero.

From a computational perspective, we stress that to generate these rational spaces, no inverses need to be explicitly computed; rather, system solves with the corresponding matrices are performed at each iteration.

After k iterations, the procedure has generated W_k and V_k. To be able to determine a projected approximation $\mathcal{Z}_k$, an additional condition is required. To this end, for the sought after approximation $\widehat{\mathbf{Z}} = V_k\mathcal{Z}_kW_k^T$ we impose that the residual matrix $R_k := \widehat{\mathbf{Z}}A_2^{-1} + \mathcal{M}^{-1}\mathcal{D}_0\widehat{\mathbf{Z}} - \widehat{F}$ satisfies the following matrix Galerkin condition,

$$V_k^T R_k W_k = 0,$$

that is, the residual is orthogonal to the approximation space, in a matrix sense. Since $\widehat{F} = V_k\Phi_kW_k^T$ for some Φ_k by construction, no information is lost in the known term $\widehat{F}$ when projecting onto the reduced space via this orthogonality condition; we refer the reader to [26] for a more detailed discussion on the Galerkin condition. Explicitly writing down the residual, and taking into account both the form of $\widehat{\mathbf{Z}}$ and the orthogonality of the columns in V_k, W_k we obtain

$$V_k^T\widehat{\mathbf{Z}}A_2^{-1}W_k + V_k^T\mathcal{M}^{-1}\mathcal{D}_0\widehat{\mathbf{Z}}W_k = V_k^T\widehat{F}W_k,$$

that is

$$\mathcal{Z}_kW_k^TA_2^{-1}W_k + V_k^T\mathcal{M}^{-1}\mathcal{D}_0V_k\mathcal{Z}_k = V_k^T\widehat{F}W_k,$$

which is again a Sylvester equation, but of reduced dimension. If both V_k, W_k have small dimensions, the equation above also has small size, and classical dense solvers such as the Bartels and Stewart algorithm can be used [1]. The accuracy of the approximation is monitored by computing in a cheap manner the Frobenius norm of the residual matrix [24]. If the solution is not sufficiently good, then the two approximation spaces are expanded, and the reduced Sylvester equation is updated and solved. The complete iterative projection procedure for the Sylvester equation can be found, e.g., in [26, Algorithm 5, section 4.4.1]. Here we would like to stress that the major difference with respect to the classical Extended Krylov subspace procedure is that an *augmented* space is generated, so as to deal with a singular coefficient matrix, while maintaining the good convergence properties of rational Krylov subspaces. In the following we shall refer to this approach as the Extended Krylov ($\mathbb{EK}(\sigma)$) method with parameter σ.

6. Numerical Experiments. We report on our numerical experience with the methods we have introduced, compared with the numerical solution of the monolithic equation $\mathcal{M}\mathbf{u} = b$, by either a direct method (Matlab backslash) or by some iterative method. In particular, in this section we use both the nullspace method of section 4, and the projection method for the large scale problem described in section 5.

EXAMPLE 6.1. We consider the problem in (1.1) where A_1 (A_2) is the scaled five-point stencil finite difference discretization of the operator $\mathcal{L}_1(u) = -u_{xx} - u_{yy}$ (of the operator $\mathcal{L}_1(u) = -(e^{-10xy}u_x)_x - (e^{10xy}u_y)_y + 10(x+y)u_x$) in the unit square, with the same number of nodes in the two directions. Here $[F_1; F_2]$ is a rank-one matrix with uniformly distributed random entries, while $B = \text{bidiag}(-1, \underline{1}) \in \mathbb{R}^{(n_1-\sqrt{n_2})\times n_1}$. The convergence tolerance is set to 10^{-6}, while the shift parameter in (5.4) is $\sigma = 10^{-2}$. The numerical results are reported in Table 6.1.

n_1	n_2	size($\mathcal{A}$)	Monolithic	Direct Nullspace	Iterative $\mathbb{EK}(\sigma)$
400	100	79,000	6.9769e-02	9.4012e-02	3.1523e-02 (4)
900	225	401,625	3.4808e-01	6.3597e-01	5.0447e-02 (4)
1600	400	1272,000	1.1319e+00	4.7888e+00	7.8018e-02 (4)
2500	625	3109,375	3.1212e+00	1.5063e+01	1.5282e-01 (5)
3600	900	6453,000	1.0210e+01	3.9419e+01	2.8053e-01 (5)
4900	1225	11,962,125	3.7699e+01	1.0721e+02	1.4754e+00 (5)

TABLE 6.1

Example 6.1. Elapsed time (in parentheses is the number of iterations) for the direct solution of the monolithic equation, for the nullspace method and for the iterative method.

For this class of problems the nullspace method is never competitive, as it does not exploit the sparsity of the given data. This is instead done by the direct solver, which is able to constrain computational costs up to a problem of size 10^5. Larger sizes clearly show the limitation of the approach, compared to the new iterative method. We should keep in mind, however, that the new method is effective for low rank F, while solution methods for different structural properties remain largely unexplored.

EXAMPLE 6.2. We consider the stochastic finite element discretization of the one-dimensional Stokes problem with a stochastic component for the viscosity (see section 2). In the standard implementation, this gives the following linear system

$$\begin{bmatrix} \mathcal{H} & \mathcal{B}^T \\ \mathcal{B} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}$$

where

$$\mathcal{H} = (\nu_0 G_0 + \nu_1 G_1) \otimes A_x, \quad \mathcal{B} = G_0 \otimes B_x,$$

while $\nu = \nu_0 + \nu_1 \xi(\omega)$ is the uncertain viscosity, with ξ the random variable, while A_x, B_x are standard finite element matrices, while G_0, G_1 are associated with the discretization of the random variables. Here $G_0 = I$ and G_1 is symmetric, tridiagonal with eigenvalues in $[-\sqrt{3}, \sqrt{3}]$ [20].

Let $A = A_x$ and $B = B_x \in \mathbb{R}^{n_B \times n_1}$. We rewrite the problem as

$$\nu_0 A\mathbf{X}G_0 + \nu_1 A\mathbf{X}G_1 + B^T\mathbf{P}G_0 = F_1, \qquad B\mathbf{X}G_0 = F_2.$$

With the block formulation of the previous section and $G_0 = I$ we obtain the same form as in (5.1), that is

$$\begin{bmatrix} \nu_0 A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{P} \end{bmatrix} + \begin{bmatrix} \nu_1 A & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{P} \end{bmatrix} G_1 = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}, \quad \Leftrightarrow \quad \mathcal{M}\mathbf{Z} + \mathcal{D}_0 \mathbf{Z} G_1 = F,$$

Here we took $F = f\mathbf{1}^T$, where f is a random vector taken from a uniform distribution in the interval (0,1), and $\mathbf{1}$ is the vector of all ones. For G_1 nonsingular, we can rewrite the problem as $\mathbf{Z}G_1^{-1} + \mathcal{M}^{-1}\mathcal{D}_0\mathbf{Z} = \widetilde{F}$ and the procedure described in the previous section applies. The performance of the new method is compared with both a direct sparse method and an iterative procedure for solving the monolithic equation. Following [20], the considered iterative solver is preconditioned MINRES, where the preconditioner is given by the symmetric and positive definite matrix

$$P = \begin{bmatrix} \mu G_0 \otimes \nu_0 A & \\ & \mu G_0 \otimes B A^{-1} B^T \end{bmatrix}.$$

The (1,1) block is applied exactly, preceded by a Cholesky factorization of A. For the (2,2) block an incomplete Cholesky factorization (with zero fill-in and diagonal pivoting) of the Schur complement $BA^{-1}B^T$ is performed. For the largest problem size this last matrix could not be explicitly constructed, and the code broke down. The computational cost of building the preconditioner is not taken into account in the reported timings.

Table 6.2 shows the results for $\nu_0 = 1/10$ and $\nu_1 = 3\nu_0/10$, with data taken from [20]. We notice that for a still quite modest value of n_2, the CPU time of the sparse direct method for the monolithic equation becomes prohibitive. Preconditioned MINRES applied to the monolithic equation is more efficient, however it becomes too slow, and eventually the Schur complement matrix used to generate the preconditioner cannot be created. The matrix-oriented iterative method succeeds pretty quickly for all tested problem parameters. We also explicitly notice that the approximate solution has rank two for all values of n_1 and both values of n_2. This is to be expected, since with two terms, the truncated Karhunen-Loève expansion of the solution gives rise to a rank two solution. This inherent property cannot be exploited in the Kronecker formulation. Such a feature allows us to obtain a memory-saving approximation, $\mathbf{Z} = Z_1 Z_2^T$ with Z_1, Z_2 having only two columns.

n_1	n_2	n_B	size($\mathcal{A}$)	Monolithic direct	Monolithic MINRES	Iterative $\mathbb{EK}(\sigma)$
1256	4	389	6,580	0.1852	0.146 (11)	0.19 (2)
3526	4	990	18,064	0.9063	0.275 (11)	0.52 (2)
9812	4	2615	49,708	4.6418	0.981 (10)	2.09 (2)
n_1	n_2	n_B	size($\mathcal{A}$)	Monolithic direct	Monolithic MINRES	Iterative $\mathbb{EK}(\sigma)$
1256	165	389	271,425	2.91	1.53 (11)	0.20 (2)
3526	165	990	745,140	12.16	7.43 (11)	0.45 (2)
9812	165	2615	2050,455	-	-	1.87 (2)

TABLE 6.2

Example 6.2. One-dimensional Stokes problem with true ($n_2 = 4$) and artificial ($n_2 = 165$) number of stochastic terms.

To further exercise the algorithm, we have artificially enlarged the size of the G matrices. More precisely we have taken the matrices G_0, G_1 obtained by a choice of

the parameters associated with the stochastic space so as to give $n_2 = 165$ (this would in fact give a larger number of G terms, which we omit in our analysis). The results of these experiments are reported in Table 6.2. The size of B is unchanged and it is not reported in the table. While for n_2 small the solution of the monolithic equation by means of preconditioned MINRES was still competitive, this is no longer so for larger n_2, eventually leading to a memory failure for the largest problem considered. The new approach remains competitive throughout. Note that both iterative methods are mesh independent, that is their iteration number to reach convergence does not depend on the problem size. As already mentioned, for the new method this is related to the particular form of the problem.

n_1	n_2	size($\mathcal{A}$)	Monolithic direct	Monolithic MINRES	Iterative $\mathbb{EK}(\sigma)$
2512	4	11,604	0.55	0.12 (12)	0.28 (2)
7052	4	32,168	3.73	0.36 (12)	1.22 (2)
19624	4	88,956	11.93	1.51 (12)	4.37 (2)
n_1	n_2	size($\mathcal{A}$)	Monolithic direct	Monolithic MINRES	Iterative $\mathbb{EK}(\sigma)$
2512	165	478 665	7.60	3.16 (17)	0.33 (2)
7052	165	1 326 930	34.08	15.52 (18)	1.32 (2)
19624	165	3 669 435	–	–	5.69 (3)

TABLE 6.3

Example 6.3. Two-dimensional Stokes problem with true ($n_2 = 4$) and artificial ($n_2 = 165$) number of stochastic terms.

EXAMPLE 6.3. We consider the two-dimensional counterpart of the problem in Example 6.2. This gives the following linear system

$$\begin{bmatrix} \mathcal{H} & \mathcal{B}^T \\ \mathcal{B} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}$$

where

$$\mathcal{H} = \text{blkdiag}((\nu_0 G_0 + \nu_1 G_1) \otimes A_x, (\nu_0 G_0 + \nu_1 G_1) \otimes A_y), \quad \mathcal{B} = [G_0 \otimes B_x, G_0 \otimes B_y];$$

here A_x, A_y, B_x, B_y are standard finite element matrices, and G_0, G_1 are associated with the discretization of the random variables. Again, $G_0 = I$ and G_1 is symmetric, tridiagonal with eigenvalues in $[-\sqrt{3}, \sqrt{3}]$.

Let $A = \text{blkdiag}(A_x, A_y)$ and $B = [B_x; B_y]$; Here A_y and A_x have the same dimensions, and the same holds for B_x and B_y, with data dimensions the same as those in the 1D case of Example 6.2. We rewrite the problem as

$$\nu_0 A\mathbf{X}G_0 + \nu_1 A\mathbf{X}G_1 + B^T\mathbf{P}G_0 = F_1, \qquad B\mathbf{X}G_0 = F_2.$$

With the block formulation of the previous section and $G_0 = I$ we obtain the same form as in (5.1), that is

$$\begin{bmatrix} \nu_0 A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{P} \end{bmatrix} + \begin{bmatrix} \nu_1 A & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{P} \end{bmatrix} G_1 = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}, \quad \Leftrightarrow \quad \mathcal{M}\mathbf{Z} + \mathcal{D}_0\mathbf{Z}G_1 = F.$$

We considered $F = \mathbf{1}\mathbf{1}^T$. Once again, for G_1 nonsingular, we can rewrite the problem as $\mathbf{Z}G_1^{-1} + \mathcal{M}^{-1}\mathcal{D}_0\mathbf{Z} = \widetilde{F}$ and the procedure described in the previous section applies. Table 6.3 shows the results for $\nu_0 = 1/10$ and $\nu_1 = 3\nu_0/10$, with data taken from [20].

Also in the two-dimensional case, preconditioned MINRES is the best performing method, as long as memory requirements remain limited. The direct method on the monolithic equation becomes very inefficient also for moderate dimensions. The new method is to be preferred when dimensions become truly large, also because of the low memory requirements. Note that both iterative methods are mesh independent also in this case.

7. Schur complement method. In this section we derive a matrix-based Schur complement approach for solving the given system of matrix equations. We first apply the Schur complement approach to the problem (1.1), and then generalize it to (1.2). This methodology is yet another way of attacking the solution of saddle point problems, see, e.g., [2, section 5]. Once again, we describe how to exploit the structure so as to avoid dealing with huge dimensional matrices; this turns out to be crucial in the case of (1.2). The key idea is to exploit a matrix-oriented version of standard Krylov subspace methods, where the coefficient matrix is an operator acting on a matrix. This way, computations with matrices of the original sizes can be performed. Moreover, the structure of the original matrices, such as low rank or banded structure, can be exploited to significantly lower the computational costs and memory requirements of the method; see [19] for one such example. For reference, the general matrix-oriented version of CG for the matrix equation $\mathcal{L}(X) = F$ is described in Algorithm 2, for $\mathcal{L}$ being self-adjoint and positive definite.

Algorithm 2 Given the operator $\mathcal{L}$, the matrix F and the approximation X_0
Set $R_0 = F - \mathcal{L}(X_0)$, $P_0 = R_0$
For $k = 0, 1, 2, ...$
$W_k = \mathcal{L}(P_k)$
$\alpha_k = \frac{\|R_k\|_F^2}{\langle P_k, W_k \rangle_F}$
$X_{k+1} = X_k + P_k \alpha_k$
$R_{k+1} = R_k - W_k \alpha_k$
If $\|R_{k+1}\|_F / \|R_0\|_F <$ tol then Stop
$\beta_k = \frac{\|R_{k+1}\|_F^2}{\|R_k\|_F^2}$
$P_{k+1} = R_{k+1} + P_k \beta_k$

In the following we derive the specific forms of $\mathcal{L}$ and F when the Schur complement formula is applied to both cases of two and three matrix equations. The Kronecker (vector) version (1.3) of the matrix problem (1.1) can be solved by means of a Schur complement approach. Indeed, for $\mathcal{A}$ nonsingular in (1.3) we can write $\mathbf{x} = \mathcal{A}^{-1}(f_1 - \mathcal{B}^T \mathbf{p})$, and substituting into the second block of equations yields the well known Schur complement linear system $\mathcal{B}\mathcal{A}^{-1}\mathcal{B}^T \mathbf{p} = \mathcal{B}\mathcal{A}^{-1} f_1 - f_2$ to be solved in $\mathbf{p}$. In the following we show that this system can be implicitly posed in a matrix-oriented framework directly from (1.1), thus avoiding the Kronecker formulation.

Let us consider the linear operator

$$\mathcal{L}_{12}(X) = A_1 X + X A_2.$$

The matrix $\mathbf{X} = \mathcal{L}_{12}^{-1}(D)$ is the action of the inverse operator to a matrix D, and it corresponds to the solution of the linear matrix equation $A_1 \mathbf{X} + \mathbf{X} A_2 = D$. Then from the first equation in (1.1) we obtain

$$\mathbf{X} = \mathcal{L}_{12}^{-1}(F_1 - \mathbf{P} B)$$

which, substituted into the second matrix equation in (1.1) yields

$$B\mathcal{L}_{12}^{-1}(B^T\mathbf{P}) = -F_2 + B\mathcal{L}_{12}^{-1}(F_1). \tag{7.1}$$

Explicit computation shows that $\mathrm{vec}(\mathcal{L}_{12}^{-1}(\mathbf{P}B)B^T) = \mathcal{B}\mathcal{A}^{-1}\mathcal{B}^T\mathbf{p}$, therefore the matrix operator $B\mathcal{L}_{12}^{-1}(B^T\cdot) : P \mapsto B\mathcal{L}_{12}^{-1}(B^TP)$ is symmetric and positive definite if $\mathcal{A}$ is symmetric and positive definite. Since the equation (7.1) is mathematically equivalent to the vector Schur complement equation, it is possible to solve the linear matrix equation by using a matrix-oriented iterative solver, as desired. The application of the operator $P \mapsto B\mathcal{L}_{12}^{-1}(B^TP)$, that is $W = B\mathcal{L}_{12}^{-1}(B^TP)$ can be performed in three steps as follows:

1. Given P, compute $\hat{P} = B^TP$
2. Solve the Sylvester equation $A_1\mathbf{V} + \mathbf{V}A_2 = \hat{P}$ for $\mathbf{V}$
3. Compute $W = B\mathbf{V}$

All other operations can be performed as matrix-matrix products and sums, as described in Algorithm 2.

A matrix equation solve in step 2 needs to be employed at each iteration. If the size of the Sylvester equation is such that a dense solver such as the Bartels-Stewart method is used, then it may be convenient to first perform a Schur decomposition of the matrices A_1 and A_2, and perform the whole computation by the iterative solver with the transformed problem. More precisely, let

$$A_1^* = Q_1R_1Q_1^*, \quad A_2 = Q_2R_2Q_2^* \tag{7.2}$$

be the two Schur decompositions, with Q_i unitary and R_i block upper triangular[3]. Here "$*$" denotes conjugate transposition. Then by multiplying (1.1) by Q_1^* and Q_2 from the left and from the right, respectively, we obtain

$$R_1^*\widehat{\mathbf{X}} + \widehat{\mathbf{X}}R_2 + (Q_1^*B^*)\mathbf{P} = (Q_1^*F_1Q_2), \quad BQ_1\widehat{\mathbf{X}} = (F_2Q_2).$$

The Schur complement approach can thus be applied to the transformed problem, and a matrix-oriented Krylov subspace method employed for its solution.

It is important to realize that, although the computation of two Schur decompositions may appear costly, this is performed on matrices that may be relatively small, compared with the original problem in Kronecker form. For instance, matrices of size $n_1 = n_2 = 500$, $m = 400$ are associated with a Kronecker form having a coefficient matrix of size $n_1n_2 + n_1m = 450\,000$.

For the system of two matrix equations this approach is not competitive with respect to the strategies developed in the previous sections, both in the small and large scale cases. Therefore, we shall mainly focus on its performance for the system of three matrix equations, for which the previous methods did not give a natural generalizations.

REMARK 7.1. *If the coefficient matrix (the operator in our case) are well conditioned, the* CG *method can conveniently exploit the structure in the data, whenever present. Therefore, if the right-hand side is low rank, the first few iterates can be well represented by low rank matrices. The same holds for banded matrices. We refer the interested reader to the discussion in [19].*

[3]Here we assume complex arithmetic. If real arithmetic is preferred, then real orthogonal matrices can be obtained, at the cost of a *block* upper triangular matrix R.

7.1. The Schur complement method for the three matrix equation case. The procedure can be naturally extended to the system of three matrix equations in (1.2), while exploiting the Sylvester structure of the "diagonal" blocks. Indeed, the coefficient operator can be written as

$$(\mathbf{X}, \mathbf{Y}, \mathbf{P}) \mapsto (\mathcal{L}_{21}(\mathbf{X}) + B^T\mathbf{P},\, \mathcal{L}_{12}(\mathbf{Y}) + \mathbf{P}B,\, B\mathbf{X} + \mathbf{Y}B^T)$$

with obvious notation for $\mathcal{L}_{12}$. Proceeding like in the previous case, we obtain the matrix-oriented Schur complement equation

$$B\mathcal{L}_{21}^{-1}(B^T\mathbf{P}) + \mathcal{L}_{12}^{-1}(\mathbf{P}B)B^T = -F_3 + B\mathcal{L}_{21}^{-1}(F_1) + \mathcal{L}_{12}^{-1}(F_2)B^T, \tag{7.3}$$

to be solved for $\mathbf{P}$. Expanding the left-hand side operator via the Kronecker product shows that the coefficient matrix is symmetric and positive definite as along as A_2 and A_1 are.

Solving (7.3) requires the application of the operator coefficient, which involves two Sylvester equation solves with the same coefficient matrices. As in the case of two matrix equations, an a-priori Schur decomposition of the matrices A_1, A_2 can be particularly advantageous, since here the transformation can be exploited in two of the three matrix equations. As before, using the decompositions in (7.2) we can rewrite (1.2) as

$$R_2\widehat{\mathbf{X}} + \widehat{\mathbf{X}}R_1^* + (Q_2^*B_1^TQ_1)\widehat{\mathbf{P}} = (Q_2^*F_1Q_1) \tag{7.4}$$

$$R_1\widehat{\mathbf{Y}} + \widehat{\mathbf{Y}}R_2^* + \widehat{\mathbf{P}}(Q_1^*B_2Q_2) = Q_1^*F_2Q_2 \tag{7.5}$$

$$(Q_1^*B_1Q_2)\widehat{\mathbf{X}} + \widehat{\mathbf{Y}}Q_2^*B_2^TQ_1 = Q_1^*F_3Q_1, \tag{7.6}$$

where $\widehat{\mathbf{X}} = Q_2^*\mathbf{X}Q_1$, $\widehat{\mathbf{Y}} = Q_1^*\mathbf{Y}Q_2$ and $\widehat{\mathbf{P}} = Q_1^*\mathbf{P}Q_1$.

The use of a matrix-oriented iterative solver such as CG for approximating $\mathbf{P}$ affects the accuracy of the third matrix equation, that is the CG residual norm corresponds to the matrix residual norm associated with the third equation. The solution matrices $\mathbf{X}$ and $\mathbf{Y}$ are recovered a-posteriori by explicitly using the first two matrix equations. This implies that the first two matrix equations are solved much more accurately than the third one, that is, the associated residual norms are much smaller.

We proceed with an experiment with the Schur complement method. Comparisons are performed with respect to the iterative solver preconditioned MINRES on the monolithic equation.

EXAMPLE 7.2. We consider the MAC finite difference discretization of the Stokes operator on a staggered grid; this corresponds to the linear version of (2.7). There are n_1^2 pressure unknowns at cell centers and $n_1(n_1-1)$ velocity unknowns on each of horizontal and vertical cell sides, so that $n_2 = n_1 - 1$. Both $A_1 \in \mathbb{R}^{n_1\times n_1}$ and $A_2 \in \mathbb{R}^{n_2\times n_2}$ have diffusion coefficient equal to $\nu = 10^{-2}$; $B \in \mathbb{R}^{n_1\times n_2}$ and $B = B_1 = B_2$. Runs were obtained for $n_1 \in \{100, 200, \ldots, 500\}$. The right-hand side accounts both for the forcing term as well as for the boundary conditions. In this case, F_1 is a rank-one matrix, while F_2 and F_3 are the zero matrices. Experimental comparisons of the Schur complement method and preconditioned MINRES are displayed in Table 7.1: CPU time (in seconds) is reported, together with the number of iterations in parenthesis. In the Schur complement approach, the CG method in Algorithm 2 is used. Moreover, the eigendecomposition of the symmetric matrices $A_1 \in \mathbb{R}^{n_1\times n_1}$ and $A_2 \in \mathbb{R}^{n_2\times n_2}$, $n_2 = n_1 - 1$ is computed once for all. This way, the Sylvester equations to be solved at

dim	MINRES	Schur compl. method
29 800	0.221 (23)	0.018 (10)
119 600	1.568 (23)	0.055 (11)
269 400	4.866 (23)	0.148 (11)
479 200	11.457 (23)	0.378 (12)
749 000	29.533 (25)	2.128 (12)

TABLE 7.1

Example 7.2. Performance of the Schur complement method compared with preconditioned MINRES. *Here dim*$=3n_1(n_1-1)$ *is the dimension of the monolithic equation, with* $n_1 \in \{100, 200, \ldots, 500\}$.

each CG iteration only involve diagonal coefficient matrices of eigenvalues; see (7.4). The computational cost of the eigendecomposition is included in the reported timings. For MINRES a block diagonal preconditioner is considered, $P = \mathrm{blkdiag}(F_x, F_y, M_p)$, where M_p is the mass matrix. For this discretization M_p is a multiple of the identity matrix. This is known to be an optimal preconditioner for the problem, in the sense that the number of MINRES iterations is bounded independently of the problem size, that is, of the mesh parameter. This property can be clearly seen in our experiments. The matrix P was factorized as $P = P_1 P_1^T$, where the factor P_1 was obtained by computing the (complete) Cholesky factorization of the two blocks F_x, F_y. The cost of building the preconditioner is not taken into account in the reported performance. We also stress that the factor P_1 is banded with a narrow bandwidth, so that solving with P_1 is rather cheap. The iteration number in Table 7.1 shows that the performance of both methods is independent of the problem dimension. In terms of CPU time, the Schur complement method is clearly superior to preconditioned MINRES, with one order of magnitude lower timings.

7.2. The Schur complement method. Considerations for the large-scale setting. If the involved coefficient matrices are themselves very large, the Bartels-Stewart algorithm becomes expensive. An iterative method can thus be used to approximately solve the two Sylvester equations associated to the operator-matrix operation. This gives rise to a typical inner-outer iteration, where not only the coefficient matrix is not know explicitly, but it cannot even be applied with high accuracy. In this setting, one usually talks about *inexact* procedures, and the convergence theory of the underlying method no longer applies. Several strategies have been proposed, some of which quite successful, to make the whole procedure robust. In the symmetric and positive definite case, the algorithm in [17] provides a good trade-off between efficiency and robustness of the inexact procedure. We also recall that if a method such as GMRES is employed with inexact operator-matrix products, the inexactness of this operator can be actually *increased* as convergence takes place, further lowering the computational cost; see, e.g., [27] and references therein.

8. Conclusions. We have proposed several algorithms for solving systems of two and three matrix equations, stemming from the numerical modelling of real application problems. Our results show that by taking into account the original problem structure, we are able to solve very large problems in a few seconds.

In the three matrix equation case we have shown the feasibility of an iterative method based on the Schur complement. With the given generality of the dimensions the derivation of a direct procedure was considered to be too cumbersome to be pursued.

The discussed problems can be generalized in several directions. In connection to (1.1) we can readily handle the case in which the coefficient matrix in the second equation is not the transpose of that in the first equation. Moreover, we can transform the generalized case, that is that having nonsingular coefficient matrices on both sides of $\mathbf{X}$, say, into the considered setting by one sided multiplication.

The case of a multi-term system is by far more challenging. This may arise by either including many terms in one of the main variables ($\mathbf{X}$ and/or $\mathbf{Y}$), or by adding more matrix equations with the corresponding number of unknowns. The first setting is typical of stochastic partial differential equations, and will be the focus of future research.

Acknowledgements. We thank two reviewers for helpful comments. We thank Catherine Powell and Howard Elman for providing us with the data for the stochastic Stokes 1D and 2D problems, and for the (Navier-)Stokes problem, respectively.

REFERENCES

[1] R. H. Bartels and G. W. Stewart. Algorithm 432: Solution of the Matrix Equation $AX + XB = C$. *Comm. of the ACM*, 15(9):820–826, 1972.
[2] Michele Benzi, Gene H. Golub, and Jörg Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.
[3] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang. *Spectral methods*. Scientific Computation. Springer-Verlag, Berlin, 2006.
[4] Feng Ding and Tongwen Chen. On iterative solutions of general coupled matrix equations. *SIAM J. Control Optim.*, 44:2269–2284, 2006.
[5] Andrii Dmytryshyn. *Tools for Structured Matrix Computations: Stratifications and Coupled Sylvester Equations*. PhD thesis, Department of Computer Science, Umea University, 2015.
[6] S. Dolgov and M. Stoll. Low-rank solution to an optimization problem constrained by the Navier-Stokes equations. *SIAM J. Sci. Comput.*, 39(1):A255–A280, 2017.
[7] Guangren Duan. A note on combined generalized Sylvester matrix equations. *Journal of Control Theory and Applications*, 4:397–400, 2004.
[8] Guangren Duan and Biao Zhang. Robust model-reference control for descriptor linear systems subject to parameter uncertainties. *Journal of Control Theory and Applications*, 5(3):213–220, 2007.
[9] H. C. Elman, D. G. Furnival, and C. E. Powell. H(div) preconditioning for a mixed finite element formulation of the diffusion problem with random data. *Math. of Comp.*, 79(270):733–760, 2010.
[10] Howard C. Elman. Preconditioning for the steady-state Navier-Stokes equations with low viscosity. *SIAM J. Sci. Comput.*, 20(4):1299–1316, 1999.
[11] B. Kågström and P. Poromaa. LAPACK-Style algorithms and software for solving the generalized Sylvester equation and estimating the separation between regular matrix pairs. *ACM Transactions on Mathematical Software*, 22(1):78–103, 1996. Also as LAPACK Working Note 75.
[12] Patrick M. Knupp and Stanly Steinberg. *The fundamentals of grid generation*. Knupp, 1992.
[13] Sheng-Kun Li. Iterative Hermitian R-conjugate solutions to general coupled Sylvester matrix equations. *Filomat*, 31(7):2061–2072, 2017.
[14] G. J. Lord, C. E. Powell, and T. Shardlow. *An introduction to computational stochastic PDEs*. Cambridge University Press, 2014.
[15] M. Mariton. *Jump Linear Systems in Automatic Control*. Marcel Dekker, 1990.
[16] The MathWorks, Inc. *MATLAB 7*, r2017b edition, 2017.
[17] Y. Notay. Flexible conjugate gradients. *SIAM J. Sci. Comput.*, 23(4):1444–1460, 2000.
[18] D. Palitta and V. Simoncini. Matrix-equation-based strategies for convection-diffusion equations. *BIT Numer. Math.*, 56:751–776, 2016.
[19] D. Palitta and V. Simoncini. Numerical methods for large-scale Lyapunov equations with symmetric banded data. arXiv 1711.04187, Dipartimento di Matematica, Università di Bologna, 2017. To appear in SIAM J. Scientific Computing.
[20] C. Powell and D. Silvester. Preconditioning steady-state Navier-Stokes equations with random data. *SIAM Journal Sci. Comp.*, 34(5):A2482–A2506, 2012.

[21] C. E. Powell, D. Silvester, and V. Simoncini. An efficient reduced basis solver for stochastic Galerkin matrix equations. *SIAM J. Sci. Comput.*, 39(1):A141–A163, 2017.
[22] A. Saberi, A. A. Stoorvogel, and P. Sannuti. *Control of Linear Systems with Regulation and Input Constraints*. Communications and Control Engineering. Springer-Verlag, New York, 1999.
[23] G Sangalli and M Tani. Isogeometric preconditioners based on fast solvers for the sylvester equation. *SIAM Journal on Scientific Computing*, 38(6):A3644–A3671, 2016.
[24] S. Shank and V. Simoncini. Krylov subspace methods for large scale constrained Sylvester equations. *SIAM J. Matrix Anal. Appl.*, 34(4):1448–1463, 2013.
[25] D. Silvester and M. Mihajlovic. A black-box multigrid preconditioner for the biharmonic equation. *BIT Numerical Mathematics*, 44:151–163, 2004.
[26] V. Simoncini. Computational methods for linear matrix equations. *SIAM Review*, 58(3):377–441, Sept 2016.
[27] V. Simoncini and D. B. Szyld. Theory of inexact Krylov subspace methods and applications to scientific computing. *SIAM J. Sci. Comput.*, 25(2):454–477, 2003.
[28] V. Simoncini and D. B. Szyld. Recent computational developments in Krylov subspace methods for linear systems. *Num. Lin. Alg. with Appl.*, 14(1):1–59, 2007.
[29] Qing-Wen Wang, Hua-Sheng Zhang, and Guang-Jing Song. A new solvable condition for a pair of generalized Sylvester equations. *Electronic Journal of Linear Algebra*, 18:289–301, June 2009.
[30] W. Murray Wonham. *Linear Multivariable control: a geometric approach*. Springer, second edition, 1979.
[31] Ai-Guo Wu, Gang Feng, Guang-Ren Duan, and Wei-Jun Wu. Iterative solutions to coupled Sylvester-conjugate matrix equations. *Computers and Mathematics with Applications*, 60:5466, 2010.
[32] Bin Zhou, Guang-Ren Duan, and Zhao-Yan Li. Gradient based iterative algorithm for solving coupled matrix equations. *Systems & Control Letters*, 58(5):327–333, May 2009.