Stein-based preconditioners for weak-constraint 4D-var

Davide Palitta [a], Jemima M. Tabeart [b,*,1]

[a] *Dipartimento di Matematica and AM², Alma Mater Studiorum - Università di Bologna, Piazza di Porta S. Donato, 5, I-40127, Bologna, Italy*
[b] *School of Mathematics, University of Edinburgh, Peter Guthrie Tait Road, Edinburgh, UK*

ARTICLE INFO

ABSTRACT

Algorithms for data assimilation try to predict the most likely state of a dynamical system by combining information from observations and prior models. Variational approaches, such as the weak-constraint four-dimensional variational data assimilation formulation considered in this paper, can ultimately be interpreted as a minimization problem. One of the main challenges of such a formulation is the solution of large linear systems of equations which arise within the inner linear step of the adopted nonlinear solver. Depending on the selected approach, these linear algebraic problems amount to either a saddle point linear system or a symmetric positive definite (SPD) one. Both formulations can be solved by means of a Krylov method, like GMRES or CG, that needs to be preconditioned to ensure fast convergence in terms of the number of iterations. In this paper we illustrate novel, efficient preconditioning operators which involve the solution of certain Stein matrix equations. In addition to achieving better computational performance, the latter machinery allows us to derive tighter bounds for the eigenvalue distribution of the preconditioned linear system for certain problem settings. A panel of diverse numerical results displays the effectiveness of the proposed methodology compared to current state-of-the-art approaches.

1. Introduction

Given a computational model for a dynamical system, data assimilation aims to merge observational, measured data of that system with prior model information to obtain a better estimate of the system state at a specified time. The most mature application of data assimilation is to numerical weather prediction (NWP), where it is used to obtain the initial conditions for forecasts [7], but in recent years data assimilation approaches have been studied more broadly within earth sciences, ecology, and neuroscience; see, e.g., [35,30,41]. In particular, observations $y_i \in \mathbb{R}^{p_i}$ at time $t_i \in [t_0, t_N]$ are combined with prior information $x_b \in \mathbb{R}^s$ from a model to compute the most likely state $x_i \in \mathbb{R}^s$ of the system at time t_i. It is typically assumed that the background state x_b can be written as $x_b = x_0^t + \epsilon^b$ where x_0^t denotes the true initial state of the system with the error ϵ^b. This error is distributed according to a normal distribution with error covariance matrix $B \in \mathbb{R}^{s \times s}$ and zero mean, i.e., $\epsilon^b \sim \mathcal{N}(0, B)$. Similarly, we write each observation in terms of the true initial state as $y_i = \mathcal{H}_i(x_i^t) + \epsilon_i$ with the observation error $\epsilon_i \sim \mathcal{N}(0, R_i)$ for all $i = 0, \ldots, N$. In order to map between observation and state space, we

* Corresponding author.
  E-mail addresses: davide.palitta@unibo.it (D. Palitta), tabeart@maths.ox.ac.uk (J.M. Tabeart).
[1] Now at: Mathematical Institute, University of Oxford, Andrew Wiles Building, Oxford, UK.

introduce a possibly nonlinear operator $\mathcal{H}_i : \mathbb{R}^s \to \mathbb{R}^{p_i}$ which connects the true state x_i^t at time t_i and the observational data y_i.

One of the notable aspects of the weak-constraint four-dimensional variational assimilation problem (4D-Var) is the propagation of the computed states. If x_{i-1} denotes the state variable at time t_{i-1}, this is propagated to the next observation time t_i via an imperfect forecast model $\mathcal{M}_i$ such that $x_i = \mathcal{M}_i(x_{i-1}) + \epsilon_i^m$ where $\epsilon_i^m \sim \mathcal{N}(0, Q_i)$ for all $i = 1, \ldots, N$. This is probably the main difference between the weak- and strong-constrained 4D-Var approaches. Indeed, in the latter methodology the forecast model is supposed to be exact.

The ultimate goal of weak-constrained 4D-Var is then minimizing the following functional

$$J(x) = (x_0 - x_b)^T B^{-1}(x_0 - x_b) + \sum_{i=0}^{N}(y_i - \mathcal{H}_i(x_i))^T R_i^{-1}(y_i - \mathcal{H}_i(x_i))$$

$$+ \sum_{i=1}^{N}(x_i - \mathcal{M}_i(x_{i-1}))^T Q_i^{-1}(x_i - \mathcal{M}_i(x_{i-1})), \tag{1}$$

where $x = (x_0^T, \ldots, x_N^T)^T \in \mathbb{R}^{(N+1)s}$ collects all the state variables $x_0, \ldots, x_N$.

The objective function (1) is often minimized by means of an incremental approach [8]. Roughly speaking, this consists of a Gauss-Newton scheme where at each iteration a linearised problem needs to be solved. It has been shown that such a linear, inner problem can be reformulated as a large, sparse, symmetric, but also very structured saddle point linear system; see, e.g., [12,11,18]. A more traditional approach consists of solving the SPD linear system stemming from the quadratic optimization problems arising from the adopted Gauss-Newton procedure; see e.g. [10,49].

Krylov methods like the Generalized Minimal RESidual method (GMRES) [40] and the MINimal RESidual method (MINRES) [32] are powerful tools for the solution of saddle point linear systems. See, e.g., the survey paper [2]. Similarly, the Conjugate Gradient method (CG) [21] is the most commonly used solver for SPD linear systems. In both scenarios, it is vital to choose good preconditioners for the adopted iterative scheme to ensure fast convergence in terms of both the number of iterations and wallclock times.

A variety of preconditioners for the saddle point and SPD 4D-Var problems have been proposed in the literature; see, e.g., [14,17,47,9,50]. While these operators enjoy some appealing features, e.g., they guarantee parallisability in the saddle-point context, they also neglect important features of the original linear system to achieve affordable computational costs. This worsens the capability of the preconditioners to reduce the overall iteration count. In the operational NWP setting this is particularly problematic, as in practice the maximum number of iterations is capped by a very small number compared to the dimension of the problem [7]. Therefore, any preconditioning method that reduces the iteration count without dramatically increasing the computational cost is likely to be highly beneficial.

In this work we propose to fully exploit the inherent block structure of both the saddle point and SPD formulations within a matrix-oriented GMRES/CG approach, see, e.g., [14,44,27]. Such machinery naturally leads to the design of more efficient preconditioning operators with Kronecker structure. These new preconditioners yield beneficial theoretical properties, thus achieving faster convergence in terms of number of iterations than state-of-the-art approaches, while maintaining a low computational cost and an easy-to-parallelize nature.

The framework proposed in this paper requires moderate values of p and s (e.g. $\mathcal{O}(10^3)$) to be computationally successful. We must mention that these restrictions on the problem dimensions may be unrealistic for NWP applications but can be reasonable for other data assimilation problems such as parameter estimation tasks for low-dimensional parameter domains; see, e.g., [36,23] for some examples in agriculture. The fresh methodology we present in this paper can also be successfully applied when data assimilation is combined with model order reduction. This interesting scenario sees a first reduction step aimed at reducing the state dimension. Then the reduced model is utilized within the selected data assimilation approach; see, e.g., [26]. Weak-constraint 4D-Var may be a particularly appropriate choice of data assimilation scheme to be combined with model order reduction, as it is able to take the additional model error coming from the reduction step into account explicitly. Moreover, the techniques we develop here serve as the initial step towards novel procedures tackling large-scale problems such as the ones stemming from NWP. This will be the subject of future work.

We additionally note that the weak-constraint 4D-Var problem requires more computational resource than the strong-constraint formulation, in addition to the need to prescribe model error covariance matrices. Alternative approaches have been proposed which incorporate some model error information within a strong-constraint 4D-Var approach via an inflated covariance approach, e.g. [22,15]. However, as these methods require the inversion of the inflated observation error covariance term, users are limited to the use of (approximate) observation error covariance matrices that are easy to invert. Notice that the inversion of the observation error covariance matrix is a common issue to any data assimilation approach based on quadratic minimization; see (2). In principle, our new approach is no exception. On the other hand, we are going to show that our fresh strategy allows full flexibility for all data assimilation parameters under mild assumptions on the error statistics, as well as revealing and exploiting the Kronecker structure that is obscured in the usual primal form. In our setting, the easy-to-invert nature of the observation error covariance is guaranteed by its Kronecker structure which is lost in case of inflating approaches, in general.

Here is a synopsis of the paper. In section 2 we recall the formulation of the SPD and saddle point linear systems stemming from weak-constraint 4D-Var. We briefly introduce matrix-oriented GMRES and CG in section 2.1 and in section 2.2

we describe a general preconditioning framework to be embedded in these routines. In section 3 we address the case of observation-time dependent $\mathcal{M}_i$ and propose an original, efficient preconditioning operator. The latter is very similar to the original operator with a single exception. In particular, the original forward operator is approximated by a suitable, observation-time independent one, namely $\mathcal{M}_i \equiv \widehat{\mathcal{M}}$ for all i, in the preconditioning operator. Thanks to this feature, we are able to show that the inversion of a certain matrix $\mathbf{L}$, which is the predominant computational bottleneck of state-of-the-art preconditioning procedures for 4D-Var, is in fact equivalent to solving a Stein matrix equation. In addition to leading to some insights regarding the selection of a suitable $\widehat{\mathcal{M}}$, we describe in section 3.2 how the matrix-oriented perspective allows the efficient incorporation of information from the observation term within the Schur complement of the saddle point system or, equivalently, in the preconditioner of the SPD problem, by adapting an approach proposed in [48]. We note that the observation term has often been completely neglected in state-of-the-art preconditioners, but can be incorporated approximately within the Kronecker preconditioning framework at a moderate computational cost. In section 4, some further considerations are given in the case that also the original forecast model $\mathcal{M}_i$ is observation-time independent itself, namely $\mathcal{M}_i \equiv \mathcal{M}$ for all $i = 1, \ldots, N$ in the forward operator. A number of numerical results showing the potential of our fresh, successful strategy are reported in section 5. We finish in section 6 by drawing some conclusions and presenting possible outlooks.

Throughout the paper we adopt the following notation. Capital italic letters ($\mathcal{A}$) denote block matrices whose blocks have a Kronecker structure. These blocks, and in general matrices having a Kronecker structure, are denoted by capital bold letters ($\mathbf{A}$) whereas simple capital letters (A) are used for general matrices without any Kronecker structure. I_N denotes the identity matrix of dimension N. The subscript is omitted whenever the dimension of I is clear from the context. The i-th vector of the canonical basis of $\mathbb{R}^N$ is denoted by e_i. The Kronecker product is denoted by $\otimes$, whereas $\circ$ represents the Hadamard product. Given a matrix $X \in \mathbb{R}^{n \times n}$, $\mathrm{vec}(X) \in \mathbb{R}^{n^2}$ is the vector collecting the columns of X on top of one another. For instance, the variable x in (1) can be written as $x = \mathrm{vec}([x_0, \ldots, x_N])$. To conclude, $\lambda(A)$ denotes the spectrum of the matrix A, with $\lambda_{\max}(A) = \lambda_1(A) \geq \lambda_2(A) \geq \ldots \geq \lambda_N(A) = \lambda_{\min}(A)$.

2. Linear system formulations

As previously mentioned, the vector state x which minimizes (1) can be computed by an incremental approach [8] where the cost function (1) is approximated by a quadratic function of the increment $\delta x^{(\ell)} = x^{(\ell+1)} - x^{(\ell)}$, with $x^{(\ell)}$ being the ℓ-th Gauss-Newton iterate. If $\delta x = \mathrm{vec}([\delta x_0, \ldots, \delta x_N])$, the quadratic objective function is given by

$$\delta J^{(\ell)}(\delta x) = (\delta x_0 - b_0^{(\ell)})^T B^{-1}(\delta x_0 - b_0^{(\ell)}) + \sum_{i=0}^{N} (d_i^{(\ell)} - H_i^{(\ell)} \delta x_i)^T R_i^{-1}(d_i^{(\ell)} - H_i^{(\ell)} \delta x_i)$$

$$+ \sum_{i=1}^{N} (\delta x_i - M_i^{(\ell)} \delta x_{i-1} - c_i^{(\ell)})^T Q_i^{-1}(\delta x_i - M_i^{(\ell)} \delta x_{i-1} - c_i^{(\ell)})),$$

where $b_0^{(\ell)} = x_b - x_0^{(\ell)}$, $d_i^{(\ell)} = y_i - \mathcal{H}_i(x_i^{(\ell)})$, $c_i^{(\ell)} = \mathcal{M}_i(x_{i-1}^{(\ell)}) - x_i^{(\ell)}$, and $H_i^{(\ell)}$, $M_i^{(\ell)}$ are linearizations of $\mathcal{H}_i$ and $\mathcal{M}_i$ about $x_i^{(\ell)}$, respectively. We note that B, Q_i and R_i are covariance matrices so that they are symmetric and positive semi-definite by construction. In addition, as inverse covariance matrices are required in the objective function formulation (1) we assume these matrices to be strictly positive definite. Therefore, by dropping the Gauss-Newton index (ℓ) for better readability and assuming $p_0 = \ldots = p_N = p$, CG can be employed to minimize δJ by solving the following linear system

$$\underbrace{(\mathbf{L}^T \mathbf{D}^{-1} \mathbf{L} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})}_{\mathbf{S}} \delta x = \mathbf{D}^{-1} b + \mathbf{L}^T \mathbf{H}^T \mathbf{R}^{-1} d, \tag{2}$$

where $b = \mathrm{vec}([b_0, c_1, \ldots, c_N]) \in \mathbb{R}^{(N+1)s}$, $d = \mathrm{vec}([d_0, \ldots, d_N]) \in \mathbb{R}^{(N+1)p}$, and $\mathbf{D}, \mathbf{L} \in \mathbb{R}^{(N+1)s \times (N+1)s}$, $\mathbf{R} \in \mathbb{R}^{(N+1)p \times (N+1)p}$, $\mathbf{H} \in \mathbb{R}^{(N+1)p \times (N+1)s}$ are such that

$$\mathbf{D} = \begin{pmatrix} B & & & \\ & Q_1 & & \\ & & \ddots & \\ & & & Q_N \end{pmatrix}, \mathbf{L} = \begin{pmatrix} I & & & \\ -M_1 & I & & \\ & \ddots & \ddots & \\ & & -M_N & I \end{pmatrix}, \mathbf{R} = \begin{pmatrix} R_0 & & & \\ & R_1 & & \\ & & \ddots & \\ & & & R_N \end{pmatrix}, \mathbf{H} = \begin{pmatrix} H_0 & & & \\ & H_1 & & \\ & & \ddots & \\ & & & H_N \end{pmatrix}.$$

As an alternative to the quadratic minimization (2), δx can be computed by solving the following saddle point linear system [12]

$$\underbrace{\begin{pmatrix} \mathbf{D} & 0 & \mathbf{L} \\ 0 & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^T & \mathbf{H}^T & 0 \end{pmatrix}}_{=:\mathcal{A}} \begin{pmatrix} \delta\eta \\ \delta\lambda \\ \delta x \end{pmatrix} = \begin{pmatrix} b \\ d \\ 0 \end{pmatrix}. \tag{3}$$

We note that both (2) and (3) contain a lot of inherent block structure.[2] We propose to fully exploit this structure by using matrix implementations of iterative methods and designing preconditioners with explicit Kronecker structure. We illustrate the main concept by considering a data assimilation problem where the blocks of $\mathcal{A}, \mathbf{S}$ and corresponding preconditioners $\mathcal{P}$ have Kronecker structure. This could arise naturally via consistent observation networks, with fixed observation and model error statistics, at each observation time. In the case that $Q_1 = \cdots = Q_N \equiv Q$, $R_0 = \cdots = R_N \equiv R$, and $H_1 = \cdots = H_N \equiv H$, we can write the terms above compactly by using the inherent Kronecker structure

$$\mathbf{D} = e_1 e_1^T \otimes B + (I_{N+1} - e_1 e_1^T) \otimes Q, \quad \mathbf{R} = I_{N+1} \otimes R, \quad \mathbf{H} = I_{N+1} \otimes H.$$

In the more general setting where the covariance matrices and linearised observation operator differ at each time, preconditioners with Kronecker structure can be used within the same setting. We expect the strategy presented in section 3 to be effective also in the case where we relax the Kronecker assumptions on $\mathbf{R}$, $\mathbf{H}$ and $\mathbf{D}$. This will be the subject of future work.

2.1. Matrix-oriented GMRES and CG

The Kronecker form of $\mathbf{S}$ and the blocks of the coefficient matrix $\mathcal{A}$ naturally suggests the use of matrix-oriented Krylov subspace methods to solve the linear systems (2) and (3). Depending on the adopted preconditioning operator (see section 2.2), the most popular solution schemes for solving (3) are GMRES, or MINRES if symmetry is preserved. Similarly, CG is employed for (2). It is well-known that the original vector form of such methods can be easily transformed in order to obtain a matrix-oriented formulation of these routines. These implementations can be obtained by exploiting the properties of the Kronecker product [51] and the equivalence between the 2-norm of vectors and the matrix inner product, namely $\text{vec}(A)^T \text{vec}(B) = \text{trace}((AB)^T AB)$. See, e.g., [14,44,34,27] and Appendix B.

We would like to point out that none of the Krylov routines used to obtain the results in section 5 are equipped with low-rank truncations as suggested in [14,44,27]. These truncations steps are essential to obtain a feasible storage demand when very large dimensional problems are considered. Here we suppose p, s, and N to be moderate, say $\mathcal{O}(10^3)$, so that issues related to the memory consumption in our solvers do not occur in general. Avoiding the employment of any low-rank truncation will be also crucial to obtain the results stated in Proposition 2 and section 4.1.

2.2. Preconditioning operators

It is well-known that Krylov subspace techniques require effective preconditioning operators to obtain fast convergence in terms of the number of iterations.

In the 4D-Var context, many authors considered preconditioners for (2) of the form

$$\widehat{\mathbf{S}} := \mathbf{L}^T \mathbf{D}^{-1} \mathbf{L}, \tag{4}$$

which neglect the second term $\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}$ in the definition of $\mathbf{S}$. See, e.g., [14,17,47]. This leads to an easier-to-invert preconditioning operator[3] as $\widehat{\mathbf{S}}^{-1} = \mathbf{L}^{-1} \mathbf{D} \mathbf{L}^{-T}$. A key limitation of this preconditioner is that computation of the inverse operators $\mathbf{L}^{-1}$ and $\mathbf{L}^{-T}$ requires many serial matrix products and is thus not parallelisable. One of the main strategies to overcome this issue is the introduction of a further layer of approximation related to employing an operator $\widehat{\mathbf{L}} \approx \mathbf{L}$ in the definition of $\widehat{\mathbf{S}}$, such that multiplication of a vector by $\widehat{\mathbf{S}}^{-1} = \widehat{\mathbf{L}}^{-1} \mathbf{D} \widehat{\mathbf{L}}^{-T}$ can be distributed over multiple processors. Different options for the selection of $\widehat{\mathbf{L}}$ can be found in, e.g., [12,17,14,47].

For saddle-point linear systems of the form (3), some of the most commonly-used preconditioners are the *block diagonal* and *block triangular* preconditioners. See, e.g., [3,29,17,14]. In particular, the block diagonal preconditioner is defined as follows

$$\mathcal{P}_{\mathcal{D}} := \begin{pmatrix} \mathbf{D} & & \\ & \mathbf{R} & \\ & & \widehat{\mathbf{S}} \end{pmatrix}, \tag{5}$$

where $\widehat{\mathbf{S}}$ is again such that $\widehat{\mathbf{S}} \approx \mathbf{S} = \mathbf{L}^T \mathbf{D}^{-1} \mathbf{L} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}$ is an approximation to the Schur complement $\mathbf{S}$ of $\mathcal{A}$, and it is often of the form (4).

Similarly, the block triangular preconditioner is defined as follows

$$\mathcal{P}_{\mathcal{T}} := \begin{pmatrix} \mathbf{D} & 0 & \mathbf{L} \\ & \mathbf{R} & \mathbf{H} \\ & & \widehat{\mathbf{S}} \end{pmatrix}. \tag{6}$$

---

[2] Notice that the coefficient matrix in (2) is the Schur complement of $\mathcal{A}$. This motivates the use of $\mathbf{S}$ for the former.

[3] Notice that, due to its structure, $\mathbf{L}$ is always nonsingular, regardless of the M_i's.

A different class of preconditioning operators for data assimilation problems is given by the *inexact constraint* preconditioner

$$
\mathcal{P}_{\mathcal{C}} := \begin{pmatrix} \mathbf{D} & 0 & \widehat{\mathbf{L}} \\ 0 & \mathbf{R} & 0 \\ \widehat{\mathbf{L}}^T & 0 & 0 \end{pmatrix},
\tag{7}
$$

which does not involve the inexact Schur complement $\widehat{\mathbf{S}}$.

Clearly, the effectiveness of the preconditioning operators $\widehat{\mathbf{S}}$, $\mathcal{P}_{\mathcal{D}}$, $\mathcal{P}_{\mathcal{T}}$, and $\mathcal{P}_{\mathcal{C}}$ significantly depends on the adopted approximations $\widehat{\mathbf{L}}$ and $\widehat{\mathbf{S}}$. In this paper we introduce novel tools which allow for more successful selections of $\widehat{\mathbf{L}}$ and $\widehat{\mathbf{S}}$. In particular, in section 3.1 we propose a novel approximation $\widehat{\mathbf{L}} \approx \mathbf{L}$ which amounts to a Stein operator. We will show that the inversion of such $\widehat{\mathbf{L}}$ is still computationally affordable by exploiting its matrix equation structure, while it leads to a dramatic decrease in the iteration count. Moreover, we extend this matrix-oriented method to preconditioning operators $\widehat{\mathbf{S}}$ which explicitly take into account information from the observation term $\mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}$ of the Schur complement $\mathbf{S}$ (see section 3.2) by adapting a low-rank correction approach that was proposed in [48]. The original techniques proposed in this paper lead to the design of preconditioning operators with better theoretical properties (section 4.1) and more competitive computational records (section 5).

We conclude this section by presenting a novel result related to the use of $\mathcal{P}_{\mathcal{D}}$ in our setting. We report the proof of the following theorem in Appendix A.

Theorem 1. *If* $\mathbf{b} = (b^T, d^T, 0)^T$ *denotes the right-hand side in* (3)*, then the orthonormal basis vectors* $\{v_1, \dots, v_m\}$ *of the Krylov subspace* $K_m(\mathcal{A}\mathcal{P}_{\mathcal{D}}^{-1}, \mathbf{b}) = span\{\mathbf{b}, \mathcal{A}\mathcal{P}_{\mathcal{D}}^{-1}\mathbf{b}, \dots, (\mathcal{A}\mathcal{P}_{\mathcal{D}}^{-1})^{m-1}\mathbf{b}\}$ *computed by GMRES are such that*

$$
v_{2k-1} = \begin{bmatrix} u_{2k-1} \\ w_{2k-1} \\ 0 \end{bmatrix}, \quad and \quad v_{2k} = \begin{bmatrix} 0 \\ 0 \\ z_{2k} \end{bmatrix}, \quad for\ any\ k \geq 1.
$$

The zero block structure of the basis vectors illustrated in Theorem 1 can be exploited to design more efficient implementations of the preconditioning step involving $\mathcal{P}_{\mathcal{D}}$. For instance, we can invert the (inexact) Schur complement $\widehat{\mathbf{S}}$ only for alternate iterations. Similarly, the linear systems with $\mathbf{D}$ and $\mathbf{R}$ play a role only in case of an odd iteration index. The GMRES orthogonalization step can also benefit from Theorem 1, as there is no need to explicitly perform the orthonormalization of the blocks which necessarily have to be zero in the current iteration.

We take advantage of these observations to obtain all the results related to the performance achieved by $\mathcal{P}_{\mathcal{D}}$, which are reported in section 5.

3. A new preconditioning operator

In this section we present the main contribution of this paper. In particular, we propose to use the following operator

$$
\widehat{\mathbf{L}} = I_{N+1} \otimes I_s - \Sigma \otimes \widehat{M} = \begin{pmatrix} I & & & \\ -\widehat{M} & I & & \\ & \ddots & \ddots & \\ & & -\widehat{M} & I \end{pmatrix},
\tag{8}
$$

in place of $\mathbf{L}$ within the selected preconditioning framework. The matrix $\widehat{M}$ in (8) is chosen to be some representative value of the M_i's defining $\mathbf{L}$.

In the numerical experiments in section 5 we consider a number of options for $\widehat{M}$ including, one of the M_i's (e.g. the smallest/largest in norm or condition number, first/last in the sequence), possibly cycling on the index i, and the Karcher matrix mean [5] when the M_i's are all SPD.[4]

The employment of the operator $\widehat{\mathbf{L}}$ described in (8) in the definition of $\widehat{\mathbf{S}}$, $\mathcal{P}_{\mathcal{D}}$, $\mathcal{P}_{\mathcal{T}}$, and $\mathcal{P}_{\mathcal{C}}$ leads to novel preconditioning operators for (2) and (3) that can significantly outperform other state-of-the-art approaches. In the next proposition we provide some indications of when we might expect using our fresh approach to be particularly effective. To this end, we present theoretical bounds on the eigenvalues of $\widehat{\mathbf{L}}^{-T} \mathbf{L}^T \mathbf{L} \widehat{\mathbf{L}}^{-1}$. See Appendix A for the proof.

Proposition 2. *Let* $D_i = \widehat{M} - M_i$ *and* $\widehat{\mathbf{L}}$ *as in* (8)*. The eigenvalues of* $\widehat{\mathbf{L}}^{-T} \mathbf{L}^T \mathbf{L} \widehat{\mathbf{L}}^{-1}$ *can be bounded above by*

$$
1 + \frac{N}{2} \left(\rho_N + \sqrt{\rho_N^2 + 4\rho_N} \right),
\tag{9}
$$

---

[4] Alternative matrix means can be used depending on the problem at hand.

where

$$\rho_N = \begin{cases} N \cdot \max_{m=1,\dots,N} \lambda_{max}(D_m^T D_m), & \text{if } \lambda_{\max}(\widehat{M}^T \widehat{M}) = 1, \\[2mm] \frac{1 - \lambda_{\max}^N(\widehat{M}^T \widehat{M})}{1 - \lambda_{\max}(\widehat{M}^T \widehat{M})} \cdot \max_{m=1,\dots,N} \lambda_{max}(D_m^T D_m), & \text{otherwise}. \end{cases} \qquad (10)$$

Due to the multiple levels of approximation used to obtain the result of Proposition 2, the bounds are likely to be loose in practice. However, the qualitative information encoded in (9) may provide a way to select a 'good' choice of $\widehat{M}$, and an indication of when the preconditioner (8) is likely to be effective.

In particular, Proposition 2 indicates that the best results are likely to be obtained when $\max_{i,j} \|M_i - M_j\|$ is small. If the difference between the linearised model operators is large then the maximum eigenvalue of the difference terms cannot all be kept small. Similarly, the spectral norm of $\widehat{M}$ itself must also be small. If not, then the sum in (32) will blow up rapidly even for moderate values of N. Both of these observations provide insight into a heuristic way to select $\widehat{M}$: begin by choosing M_i with smallest norm. If all the values of $||M_i||$ are similar, then it is likely that the D_i term becomes more important – we can then choose $\widehat{M}$ to be the value of M_i that minimises the average value of $\|D_i D_i^T\|$. See section 5 for a panel of diverse numerical experiments displaying such trends.

To obtain a successful preconditioning strategy, operating with $\widehat{\mathbf{L}}$ in (8) must not be computationally demanding. In particular, the application of the preconditioners $\widehat{\mathbf{S}}$, $\mathcal{P}_{\mathcal{D}}$, $\mathcal{P}_{\mathcal{T}}$, and $\mathcal{P}_{\mathcal{C}}$ always requires the inversion of $\widehat{\mathbf{L}}$. The efficient computation of $\widehat{\mathbf{L}}^{-1}$ will be the subject of the next section.

3.1. On the inversion of the Stein operator $\widehat{\mathbf{L}}$

Thanks to the properties of the Kronecker product, see, e.g., [42], the action of $\widehat{\mathbf{L}}$ in (8) on a vector $z = \text{vec}(Z)$ can be written as follows

$$\widehat{\mathbf{L}}z = \text{vec}(Z - \widehat{M} Z \Sigma^T).$$

A linear operator of the form

$$\mathfrak{L}: \quad \begin{matrix} \mathbb{R}^{s \times (N+1)} & \to & \mathbb{R}^{s \times (N+1)} \\ Z & \mapsto & Z - \widehat{M} Z \Sigma^T \end{matrix}$$

is called a Stein operator in the matrix equation literature; see, e.g., [42]. Therefore, $\widehat{\mathbf{L}}z = \text{vec}(\mathfrak{L}(Z))$. Due to this relation, hereafter, with abuse of notation, we say that also $\widehat{\mathbf{L}}$ amounts to a Stein operator. The inversion of $\widehat{\mathbf{L}}$ is thus equivalent to inverting $\mathfrak{L}$, and hence to solving a so-called Stein matrix equation

$$\text{vec}(Z) = \widehat{\mathbf{L}}^{-1} \text{vec}(V) \iff Z - \widehat{M} Z \Sigma^T = V. \qquad (11)$$

Similarly,

$$\text{vec}(Z) = \widehat{\mathbf{L}}^{-T} \text{vec}(V) \iff Z - \widehat{M}^T Z \Sigma = V. \qquad (12)$$

Different numerical methods have been proposed in the literature for the efficient solution of Stein matrix equations. See, e.g., [1,25] and [42, Section 6].

In our setting, we need to solve equations (11) and (12) several times. Indeed, depending on the adopted preconditioning scheme, a couple of Stein equations have to be solved at each GMRES/CG iteration. By fully exploiting the structure of the coefficient matrices defining the Stein equations, we illustrate a novel solution procedure that remarkably reduces the computational cost of the preconditioning steps. Our original scheme requires some minor precomputation that can be performed once prior to the start of the adopted Krylov iterative scheme.

We first notice that we can write

$$\Sigma = C - e_1 e_{N+1}^T, \qquad C = \begin{pmatrix} 0 & & & 1 \\ 1 & 0 & & \\ & \ddots & \ddots & \\ & & 1 & 0 \end{pmatrix}. \qquad (13)$$

Thanks to its circulant structure, C can be cheaply diagonalized by the fast Fourier transform (FFT), namely $C = F^{-1} \Pi F$ where

$$\Pi = \text{diag}(\pi_1, \dots, \pi_{N+1}), \quad (\pi_1, \dots, \pi_{N+1})^T = FCe_1,$$

and F denotes the discrete Fourier matrix. The observation in (13) leads to the following result. See Appendix A for its proof.

Algorithm 1 Solution of the Stein equation $Z - \widehat{M}Z\Sigma^T = V$.

---

input : $P \in \mathbb{C}^{s \times (N+1)}$ and $\Lambda, T, U \in \mathbb{C}^{s \times s}$ as in Proposition 3, $V \in \mathbb{R}^{s \times (N+1)}$.
output: $Z \in \mathbb{R}^{s \times (N+1)}$ solution to $Z - \widehat{M}Z\Sigma^T = V$.

1 Compute $Y = P \circ (T^{-1}VF^T)$
2 Compute $W = P \circ (U^{-1}(\Lambda Y F^{-1} e_{N+1}) e_1^T F^T)$
3 Set $Z = T^{-1}(Y - W)F^{-T}$

Algorithm 2 Solution of the Stein equation $X - \widehat{M}^T X\Sigma = U$.

---

input : $P \in \mathbb{C}^{s \times (N+1)}$ and $\Lambda, T, U \in \mathbb{C}^{s \times s}$ as in Proposition 3, $V \in \mathbb{R}^{s \times (N+1)}$.
output: $Z \in \mathbb{R}^{s \times (N+1)}$ solution to $Z - \widehat{M}^T Z\Sigma = V$.

1 Compute $G = P \circ (T^T V F^{-1})$
2 Compute $H = P \circ (U^{-1}(\Lambda G F^T e_1) e_{N+1}^T F^{-1})$
3 Set $Z = T^{-T}(G - H)F$

Proposition 3. *Let $\widehat{M} = T\Lambda T^{-1}$ be the eigendecomposition of $\widehat{M}$ with $\Lambda = diag(\lambda_1, \ldots, \lambda_s)$ and $P \in \mathbb{C}^{s \times (N+1)}$ be such that $P_{i,j} = 1/(1 - \lambda_i \pi_j)$. Moreover, let $U = I + diag((P(\Lambda Fe_1 \circ F^{-T} e_{N+1})))$ where $\circ$ denotes the Hadamard product. Then the solution Z to the Stein equation in* (11) *can be written as*

$$Z = T(Y - W)F^{-T}, \tag{14}$$

where

$$Y = P \circ (T^{-1}VF^T), \quad and \quad W = P \circ (U^{-1}(\Lambda Y F^{-1} e_{N+1}) e_1^T F^T).$$

Similarly, the solution Z to (12) *is such that*

$$Z = T^{-T}(G - H)F, \tag{15}$$

where

$$G = P \circ (T^T V F^{-1}), \quad and \quad H = P \circ (U^{-1}(\Lambda G F^T e_1) e_{N+1}^T F^{-1}).$$

The computational cost of the solution of the Stein equations (11)–(12) by (14)–(15) amounts to $\mathcal{O}(s^3(N+1)\log(N+1))$ floating point operations: the cubic term s^3 arises from the eigendecomposition of $\widehat{M}$, while the use of the FFT, namely computing the action of the matrices F and F^{-1} in (14)–(15), leads to the polylogarithmic term in $N + 1$. Even though the eigendecomposition of $\widehat{M}$ can be computed once, prior to the start of the Krylov routine, the approach presented in Proposition 3 requires the matrix $\widehat{M}$ to be of moderate size. On the other hand, by fully exploiting the circulant-plus-low-rank structure of Σ, we can afford sizable values of N. See [33, Section 5] for constructions similar to the ones stated in Proposition 3 derived for the solution of certain Sylvester equations.

The procedures for solving this Stein equation and its transpose are summarized in Algorithm 1 and 2, respectively, and they rely on the results presented in Proposition 3. Notice that only matrix-matrix multiplications and entry-wise operations are performed in Algorithm 1 and 2 making the preconditioning step easy to parallelize. See, e.g., [19].

3.2. Influence of Schur complement approximations

The quality of the Schur complement approximation $\widehat{\mathbf{S}}$ plays an important role in determining the effectiveness of the preconditioning operators for (2) and (3). In this work we make use of two choices of $\widehat{\mathbf{S}}$. We briefly consider classic approximations of the form $\widehat{\mathbf{S}} = \widehat{\mathbf{L}}^\top \mathbf{D}^{-1} \widehat{\mathbf{L}}$ as studied in [14,17] but involving the new approach for $\widehat{\mathbf{L}}$ illustrated in the previous section. The second option is motivated by a low-rank approximation proposed in [48] and includes information from the observation term explicitly. If

$$\widehat{\mathbf{S}} = \mathbf{L}^\top \mathbf{D}^{-1} \mathbf{L}, \tag{16}$$

then the eigenvalues of $\widehat{\mathbf{S}}^{-1}\mathbf{S}$ are given by $(N+1)(s-p)$ unit eigenvalues, and $p(N+1)$ eigenvalues given by $1 + \lambda(\mathbf{L}^{-1}\mathbf{D}\mathbf{L}^{-T}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})$. We note that $\mathbf{L}^{-1}\mathbf{D}\mathbf{L}^{-T}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ is of rank $p(N+1)$ with non-negative eigenvalues, meaning that the minimum eigenvalue of $\widehat{\mathbf{S}}^{-1}\mathbf{S}$ is 1. However, the remaining non-unit eigenvalues can be large, for example in the case that $\mathbf{R}$ is ill-conditioned; see, e.g., [46]. An alternative choice of $\widehat{\mathbf{S}}$ which allows any extreme eigenvalues arising from the observation term to be accounted for within the preconditioner, comes from considering a low-rank update to (16) of the form

$$\widehat{\mathbf{S}} = \mathbf{L}^\top \mathbf{D}^{-1} \mathbf{L} + \mathbf{K}_r \mathbf{K}_r^T, \tag{17}$$

where $\mathbf{K}_r = I_{N+1} \otimes V_r \Upsilon_r^{1/2} \in \mathbb{R}^{(N+1)s \times (N+1)r}$ and $V_r \Upsilon_r^{1/2} \in \mathbb{R}^{s \times r}$ is constructed from the leading r terms of the eigendecomposition $V \Upsilon V^T = H^T R^{-1} H$. This approach has been studied theoretically in [48], where it was proved that in addition to increasing the number of unit eigenvalues, increasing r reduces the largest eigenvalues of $\widehat{\mathbf{S}}^{-1}\mathbf{S}$.

A similar low-rank update approach can also be considered when using an approximation $\widehat{\mathbf{L}}$ to $\mathbf{L}$. In this setting the smallest eigenvalue of $\widehat{\mathbf{S}}^{-1}\mathbf{S}$ can now be smaller than one, and including more information from the observation term is not guaranteed to reduce bounds on the largest eigenvalue of the preconditioned system. However, this approach has been found to perform well for a number of problems, particularly where $\widehat{\mathbf{L}}$ is a spectrally good approximation to $\mathbf{L}$. In what follows we apply the low-rank update to an approximate first term.

In a true low-rank approach ($r << p$) computational efficiency is ensured by applying the Woodbury identity. This avoids applying the inverse of $\mathbf{D}$, which is expensive and allows the re-use of parallelisable or inexpensive approximations to $\widehat{\mathbf{L}}^{-1}$. As the setting considered in this paper requires blocks that are not too large ($\mathcal{O}(10^3)$), it is not unreasonable to compute a full decomposition of $H^T R^{-1} H$. We therefore propose using the low-rank approach and Woodbury implementation with large values of $r \leq p$, i.e.,

$$\widehat{\mathbf{S}}^{-1} = \widehat{\mathbf{L}}^{-1} \mathbf{D} \widehat{\mathbf{L}}^{-T} - \widehat{\mathbf{L}}^{-1} \mathbf{D} \widehat{\mathbf{L}}^{-T} \mathbf{K}_r \left(I_{r(N+1)} + \mathbf{K}_r^T \widehat{\mathbf{L}}^{-1} \mathbf{D} \widehat{\mathbf{L}}^{-T} \mathbf{K}_r \right)^{-1} \mathbf{K}_r^T \widehat{\mathbf{L}}^{-1} \mathbf{D} \widehat{\mathbf{L}}^{-T}. \tag{18}$$

3.2.1. Algorithmic considerations

While the use of the preconditioner $\widehat{\mathbf{S}}^{-1}$ in (18) leads to great gains in the convergence properties of the selected preconditioned iterative scheme, especially for $r \approx p$ – see section 5 – it also poses some computational challenges. We now demonstrate how to implement (18) in a feasible way by exploiting the Kronecker structure of the new preconditioner.

One benefit of using the Woodbury formulation is that the efficient implementations of $\widehat{\mathbf{L}}^{-1}\mathrm{vec}(V)$ and $\widehat{\mathbf{L}}^{-T}\mathrm{vec}(Z)$ that were introduced in section 3.1 can be reused to apply $\hat{\mathbf{L}}^{-1}\mathbf{D}\widehat{\mathbf{L}}^{-T}$. Similarly, we can exploit the Kronecker structure of $\mathbf{K}_r = I_{N+1} \otimes V_r \Upsilon_r^{1/2}$ to cheaply apply this operator and its transpose.

Therefore the main computational bottleneck of the relation (18) is the solution of the $r(N+1) \times r(N+1)$ linear system with $I_{r(N+1)} + \mathbf{K}_r^T \widehat{\mathbf{L}}^{-1} \mathbf{D} \widehat{\mathbf{L}}^{-T} \mathbf{K}_r$. We obtain computational gains by first transforming this problem into an equivalent one, and then solving the transformed problem iteratively using an inner matrix-oriented CG problem.

Solving a transformed problem:

We can make considerable computational savings by solving a transformed problem that exploits the identity plus Kronecker structure of $\widehat{\mathbf{L}}$. Writing $\widehat{M} = T \Lambda T^{-1}$, we define

$$\widetilde{\mathbf{L}} = I_{N+1} \otimes I_s - \Sigma \otimes \Lambda, \quad \widetilde{\mathbf{S}} = \widetilde{\mathbf{L}}^\top \widetilde{\mathbf{D}}^{-1} \widetilde{\mathbf{L}} + \widetilde{\mathbf{K}}_r \widetilde{\mathbf{K}}_r^T,$$

where $\widetilde{\mathbf{D}} = e_1 e_1^T \otimes T^{-1} B T^{-T} + (I_{N+1} - e_1 e_1^T) \otimes T^{-1} Q T^{-T}$, and $\widetilde{\mathbf{K}}_r = I_{N+1} \otimes V_r \Upsilon_r^{1/2}$ with V_r, Υ_r coming now from the eigendecomposition of $\widetilde{H}^T R^{-1} \widetilde{H}$, $\widetilde{H} = HT$. The computation of $\widetilde{\mathbf{S}}^{-1}$ now involves the solution of a linear system with $I_{r(N+1)} + \widetilde{\mathbf{K}}_r^T \widetilde{\mathbf{L}}^{-1} \widetilde{\mathbf{D}} \widetilde{\mathbf{L}}^{-T} \widetilde{\mathbf{K}}_r$ whose action is performed by following Algorithm 3. Notice that the cost of the latter algorithm is now linear in s and polylogarithmic in $N+1$ thanks to the semidiagonalization of $\widehat{\mathbf{L}}$.

For the SPD problem (2), we apply the preconditioner $\widehat{\mathbf{S}} = (I_{N+1} \otimes T^{-T})\widetilde{\mathbf{S}}(I_{N+1} \otimes T^{-1})$ as

$$\widehat{\mathbf{S}}^{-1}\mathrm{vec}(V) = (I_{N+1} \otimes T)(\widetilde{\mathbf{S}}^{-1}\mathrm{vec}(T^T V)). \tag{19}$$

We note the equality here, and that the only assumption required is that the full eigendecomposition of $\widehat{M}$ is available.

Similarly, for the saddle-point linear system (3), we still write $\mathcal{P}_D = \mathcal{T}\widetilde{\mathcal{P}}_D\mathcal{T}^T$ and $\mathcal{P}_T = \mathcal{T}\widetilde{\mathcal{P}}_T\mathcal{T}^T$ where

$$\widetilde{\mathcal{P}}_D = \begin{pmatrix} \widetilde{\mathbf{D}} & & \\ & \mathbf{R} & \\ & & \widetilde{\mathbf{S}} \end{pmatrix}, \qquad \widetilde{\mathcal{P}}_T = \begin{pmatrix} \widetilde{\mathbf{D}} & 0 & \widetilde{\mathbf{L}} \\ & \mathbf{R} & \widetilde{\mathbf{H}} \\ & & \widetilde{\mathbf{S}} \end{pmatrix},$$

and

$$\mathcal{T} = \begin{pmatrix} I_{N+1} \otimes T & & \\ & I_{N+1} \otimes I_p & \\ & & I_{N+1} \otimes T^{-T} \end{pmatrix}. \tag{20}$$

At the jth GMRES iteration we perform

$$\mathcal{P}_D^{-1}\mathrm{vec}(V) = \mathcal{T}^{-T}(\widetilde{\mathcal{P}}_D^{-1}(\mathcal{T}^{-T}\mathrm{vec}(V))), \tag{21}$$

and $\widetilde{\mathcal{P}}_D^{-1}$ is computed by following the strategy presented in the previous sections. Notice that $\mathcal{T}$ is block diagonal with blocks having a Kronecker form. This rich structure can be exploited to cheaply perform the transformations involving $\mathcal{T}$ itself. The same approach is adopted for the block triangular preconditioner $\mathcal{P}_T$. The inexact constraint preconditioner $\mathcal{P}_C$ would not benefit from the semi-diagonalization of $\widehat{\mathbf{L}}$ as its definition does not include the approximate Schur complement $\widehat{\mathbf{S}}$. We note at this stage that the transformed preconditioner is equivalent to (18).

Algorithm 3 Computing the action of $I_{r(N+1)} + \widetilde{\mathbf{K}}_r^T \widetilde{\mathbf{L}}^{-1} \widetilde{\mathbf{D}} \widetilde{\mathbf{L}}^{-T} \widetilde{\mathbf{K}}_r$.

---
input : $Z \in \mathbb{C}^{r \times (N+1)}$, V_r, $\Upsilon_r \in \mathbb{R}^{s \times r}$, and Λ, $\widetilde{B}$, $\widetilde{Q} \in \mathbb{R}^{s \times s}$.
output: $X \in \mathbb{R}^{r \times (N+1)}$ such that $\mathrm{vec}(X) = (I_{r(N+1)} + \widetilde{\mathbf{K}}_r^T \widetilde{\mathbf{L}}^{-1} \widetilde{\mathbf{D}} \widetilde{\mathbf{L}}^{-T} \widetilde{\mathbf{K}}_r) \mathrm{vec}(Z)$.

1 Solve $Y - \Lambda Y \Sigma = V_r \Upsilon_r Z$ by means of Algorithm 2 with $T = I$
2 Compute $W = \widetilde{B} Y e_1 e_1^T + \widetilde{Q} Y (I_{N+1} - e_1 e_1^T)$
3 Solve $U - \Lambda U \Sigma^T = W$ by means of Algorithm 1 with $T = I$
4 Set $X = Z + \Upsilon_r V_r^T U$

Inner matrix-oriented CG:
It now remains to solve $(I_{r(N+1)} + \widetilde{\mathbf{K}}_r^T \widetilde{\mathbf{L}}^{-1} \widetilde{\mathbf{D}} \widetilde{\mathbf{L}}^{-T} \widetilde{\mathbf{K}}_r)^{-1} x$ efficiently. A naive strategy would consist of assembling the coefficient matrix first, by possibly exploiting the Kronecker structure of the involved factors $\widetilde{\mathbf{K}}_r$, $\widetilde{\mathbf{L}}$, and $\widetilde{\mathbf{D}}$. To this end, the scheme proposed in [20] could be employed with straightforward modifications. Even though this step can be carried out prior to the GMRES/CG iterations, the construction of the dense matrix $\widetilde{\mathbf{K}}_r^T \widetilde{\mathbf{L}}^{-1} \widetilde{\mathbf{D}} \widetilde{\mathbf{L}}^{-T} \widetilde{\mathbf{K}}_r$ requires the solution of $r(N+1)$ matrix equations making this task computationally unaffordable.

We thus pursue a different path. Since **D** and **R** are SPD by construction $I_{r(N+1)} + \widetilde{\mathbf{K}}_r^T \widetilde{\mathbf{L}}^{-1} \widetilde{\mathbf{D}} \widetilde{\mathbf{L}}^{-T} \widetilde{\mathbf{K}}_r$ is SPD as well. Moreover, the Kronecker structure of the latter matrix can be exploited to cheaply compute its action as mentioned above. We therefore propose using an iterative method to approximate the solution of $(I_{r(N+1)} + \widetilde{\mathbf{K}}_r^T \widetilde{\mathbf{L}}^{-1} \widetilde{\mathbf{D}} \widetilde{\mathbf{L}}^{-T} \widetilde{\mathbf{K}}_r)^{-1} x$ within the preconditioner by means of a matrix-oriented CG method. This iterative method only requires applications of the operator $I_{r(N+1)} + \widetilde{\mathbf{K}}_r^T \widetilde{\mathbf{L}}^{-1} \widetilde{\mathbf{D}} \widetilde{\mathbf{L}}^{-T} \widetilde{\mathbf{K}}_r$. Therefore, the overall scheme for solving (2) and (3) can be seen as an inner-outer iteration [43] whenever the approximation (18) with $r > 0$ is adopted within the selected preconditioning framework. In particular, the outer Krylov routine (GMRES/CG) is preconditioned with a scheme involving a second, inner Krylov method (CG). Notice that the use of CG within the preconditioning step requires the employment of a flexible variant of the outer Krylov method as we are using different approximate preconditioners for each outer iteration; see [43]. The matrix-oriented implementation of flexible GMRES and CG can be easily obtained from their standard form [38,31].

We stress once again that combining the inner matrix-oriented CG method with the semi-diagonalised approach to solve for $\widetilde{\mathbf{S}}^{-1}$ significantly lowers the computational cost of the preconditioning step, especially for the case $r > 0$. Indeed, in this case, the cost of the CG iterations involved in the computation of $\widetilde{\mathbf{S}}^{-1}$ is linear in s and polylogarithmic in N thanks to the semi-diagonalization of $\widehat{\mathbf{L}}$; see Algorithm 3. We note that by using an inner iterative solver we obtain an approximation to the 'true' preconditioner $\widetilde{\mathbf{S}}$. However, our numerical experiments in section 5 reveal that we can obtain near optimal performance using this nested approach for a reasonable choice of tolerance within the inner matrix-oriented CG problem.

As previously mentioned, matrix CG is often equipped with some low-rank truncations to reduce the overall memory demand; see, e.g., [27]. However, storage will not be an issue in our context thanks to the modest problem dimensions we consider. Moreover, the introduction of any low-rank truncation would worsen the performance of the preconditioning step in general. Therefore, we perform no low-rank truncations within the inner matrix CG.

We would like to mention that, similarly to the outer Krylov routine, the inner matrix CG can also be preconditioned to achieve a faster convergence in terms of the number of iterations. However, we were not able to design an effective preconditioning operator for $I_{r(N+1)} + \widetilde{\mathbf{K}}_r^T \widetilde{\mathbf{L}}^{-1} \widetilde{\mathbf{D}} \widetilde{\mathbf{L}}^{-T} \widetilde{\mathbf{K}}_r$ with a reasonable computational cost. A number of natural preconditioning strategies did not yield improvement in terms of computational speed compared to a plain, unpreconditioned matrix CG implementation. We therefore present an unpreconditioned inner matrix CG in all the numerical experiments reported in section 5.

One may consider performing the FFT transformations involved in the solution of the Stein equations outside the GM-RES/CG iteration, thinking that this would further decrease the computational cost of the preconditioning steps. However, this would also introduce complex arithmetic in the GMRES/CG iteration, increasing the cost of the overall scheme. Moreover, the application of the FFT can be cheaply performed without forming the discrete Fourier matrix F. In particular, in all our numerical tests we employed the Matlab `fft` and `ifft` functions. In light of these considerations, we confine the use of FFT to the preconditioning step only.

4. Observation-time independent $\mathcal{M}$

If the model forecast $\mathcal{M}$ takes a constant value between each observation time, the same is true for its linearization. In this case, all the matrices M_i in the definition of **L** are the same, namely $M = M_i$ for all $i = 1, \dots, N+1$. In this case, the operator

$$\mathbf{L} = \begin{pmatrix} I & & & \\ -M & I & & \\ & \ddots & \ddots & \\ & & -M & I \end{pmatrix} = I_{N+1} \otimes I_s - \Sigma \otimes M, \tag{22}$$

is a Stein operator itself and we can thus use $\widehat{\mathbf{L}} = \mathbf{L}$ in our preconditioning strategy. In this easier setting, the latter choice leads to narrow eigenvalue distributions of the preconditioned coefficient matrices – see section 4.1 – while maintaining high computational efficiency.

In contrast to what happens in the more general case discussed at the end of section 3.2.1, we can now perform the transformation based on the eigenvector matrix T once before the Krylov routine starts, and not every time the preconditioner is applied. For instance, if $M = T\Lambda T^{-1}$, (2) can be written as

$$\widetilde{\mathbf{S}}\widetilde{\delta x} = f, \tag{23}$$

where $\widetilde{\mathbf{S}} = \widetilde{\mathbf{L}}^T \widetilde{\mathbf{D}}^{-1} \widetilde{\mathbf{L}} + \widetilde{\mathbf{H}}^T \mathbf{R} \widetilde{\mathbf{H}}$, with $\widetilde{\mathbf{D}}$, $\widetilde{\mathbf{H}}$, and $\widetilde{\mathbf{L}}$ as in section 3.2.1, and $f = (I \otimes T^T)(\mathbf{D}^{-1}b + \mathbf{L}^T\mathbf{H}^T\mathbf{R}^{-1}d)$. Once $\widetilde{\delta x}$ is computed, we retrieve the actual solution by performing $\delta x = (I_{N+1} \otimes T)\widetilde{\delta x}$.

The same approach can be followed for the saddle point linear system (3). We can write

$$\mathcal{A} = \begin{pmatrix} e_1 e_1^T \otimes B + (I_{N+1} - e_1 e_1^T) \otimes Q & & I_{N+1} \otimes I_s - \Sigma \otimes M \\ 0 & I_{N+1} \otimes R & I_{N+1} \otimes H \\ I_{N+1} \otimes I_s - \Sigma^T \otimes M^T & I_{N+1} \otimes H^T & 0 \end{pmatrix}$$

$$= \mathcal{T} \underbrace{\begin{pmatrix} e_1 e_1^T \otimes \widetilde{B} + (I_{N+1} - e_1 e_1^T) \otimes \widetilde{Q} & & I_{N+1} \otimes I_s - \Sigma \otimes \Lambda \\ 0 & I_{N+1} \otimes R & I_{N+1} \otimes \widetilde{H} \\ I_{N+1} \otimes I_s - \Sigma^T \otimes \Lambda & I_{N+1} \otimes \widetilde{H}^T & 0 \end{pmatrix}}_{\widetilde{\mathcal{A}}} \mathcal{T}^T,$$

where $\widetilde{B}$, $\widetilde{Q}$, $\widetilde{H}$, and $\mathcal{T}$ are as in section 3.2.1. In place of (3) we can thus solve the transformed system

$$\widetilde{\mathcal{A}} \begin{pmatrix} \widetilde{\delta \eta} \\ \widetilde{\delta \lambda} \\ \widetilde{\delta x} \end{pmatrix} = \begin{pmatrix} \widetilde{b} \\ \widetilde{d} \\ 0 \end{pmatrix}, \tag{24}$$

where

$$\begin{pmatrix} \widetilde{\delta \eta} \\ \widetilde{\delta \lambda} \\ \widetilde{\delta x} \end{pmatrix} = \mathcal{T}^T \begin{pmatrix} \delta \eta \\ \delta \lambda \\ \delta x \end{pmatrix}, \quad \text{and} \quad \begin{pmatrix} \widetilde{b} \\ \widetilde{d} \\ 0 \end{pmatrix} = \mathcal{T}^{-1} \begin{pmatrix} b \\ d \\ 0 \end{pmatrix}.$$

Once $(\widetilde{\delta \eta}, \widetilde{\delta \lambda}, \widetilde{\delta x})^T$ is computed, we retrieve the original solution by $(\delta \eta, \delta \lambda, \delta x)^T = \mathcal{T}^{-T}(\widetilde{\delta \eta}, \widetilde{\delta \lambda}, \widetilde{\delta x})^T$.

The preconditioning operators for (23) and (24) can be obtained by mimicking what we presented in the previous sections. The major difference is the cheaper inversion of the Stein operator $\widetilde{\mathbf{L}} = I_{N+1} \otimes I_s - \Sigma \otimes \Lambda$ which is now "semi"-diagonalized. The cost of computing $\widetilde{\mathbf{L}}^{-1}$ is thus linear in s and polylogarithmic in $N + 1$.

4.1. Spectral results

In this section we present bounds on the eigenvalues of the preconditioned systems using our new approach when $\widehat{\mathbf{L}} = \mathbf{L}$.

The spectral properties of linearised data assimilation problem (2) have been studied in [45] for the unpreconditioned 3D-Var formulation, and in [46] for strong-constraint 4D-Var preconditioned with the exact first term. Bounds on the spectrum of the (preconditioned) Hessian $\mathbf{S}$ for the weak-constraint problem can be obtained using the same theoretical approaches and replacing $\mathbf{B}$ in those bounds with $\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}$.

Proposition 4. *Let* $\widehat{\mathbf{S}} = \mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}$. *Then* $\widehat{\mathbf{S}}^{-1}\mathbf{S} = \mathbf{I}_{s(N+1)} + \mathbf{L}^{-1}\mathbf{D}\mathbf{L}^{-T}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ *and the eigenvalues of* $\widehat{\mathbf{S}}^{-1}\mathbf{S}$ *are bounded as follows*

$$\lambda(\widehat{\mathbf{S}}^{-1}\mathbf{S}) \in \left[1, \frac{\lambda_{\max}(\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})}{\lambda_{\min}(\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L})}\right]. \tag{25}$$

We note that $\widehat{\mathbf{S}}^{-1}\mathbf{S}$ *has* $(s - p)(N + 1)$ *unit eigenvalues.*

Proof. Apply [46, Theorem 4] replacing $\mathbf{B}^{-1}$ with $\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}$. □

In the case where the low-rank update to the Schur complement preconditioner presented in section 3.2 is applied with $\widehat{\mathbf{L}} = \mathbf{L}$, a bound on the maximum eigenvalue is controlled by the largest neglected eigenvalue that is not included in the approximation $\mathbf{K}_r$.

Proposition 5 *([48]). Let* $\widehat{\mathbf{S}} = \mathbf{L}^T\mathbf{D}^{-1}\mathbf{L} + \mathbf{K}_r\mathbf{K}_r^T$ *with* $\mathbf{K}_r$ *defined as in section 3.2. Then* $\mathbf{S}^{-1}\mathbf{S} = \mathbf{I}_{s(N+1)} + (\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L} + \mathbf{K}_r\mathbf{K}_r^T)^{-1}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ *and the eigenvalues of* $\widehat{\mathbf{S}}^{-1}\mathbf{S}$ *are bounded between*

$$\lambda(\widehat{\mathbf{S}}^{-1}\mathbf{S}) \in \left[1, \frac{\lambda_{r+1}}{\lambda_{\min}(\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L})}\right], \tag{26}$$

where λ_{r+1} *is the* $(r + 1)$*th largest eigenvalue of* $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$, *i.e. the largest eigenvalue that is neglected by the low-rank approximation* $\mathbf{K}_r\mathbf{K}_r^T$ *to* $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$. *We note that* $\widehat{\mathbf{S}}^{-1}\mathbf{S}$ *has* $(s + r - p)(N + 1)$ *unit eigenvalues.*

Corollary 6. *If $r = p$ then $\widehat{\mathbf{S}}^{-1}\mathbf{S} = \mathbf{I}_n$ and the eigenvalues of the preconditioned system are all units.*

Approximations of $\widehat{\mathbf{S}}$, either by using randomised approximations of $\mathbf{K}_r$ as proposed in [48], or inner iterative methods to compute (18) as proposed in this paper may lead to eigenvalues of the preconditioned system that are smaller than 1 or larger than the theoretical upper bound given by Proposition 5. We will study the performance of these approximate preconditioners in section 5.

The spectral properties of the preconditioned saddle point problem (3) have been studied in [18,14,48,11], although typically by considering approximations $\widehat{\mathbf{R}}$, $\widehat{\mathbf{D}}$ and $\widehat{\mathbf{L}}$ rather than using the exact forward model matrices. In this work we instead propose using the exact covariance matrices within the preconditioner, i.e. $\widehat{\mathbf{R}} = \mathbf{R}$ and $\widehat{\mathbf{D}} = \mathbf{D}$. This is possible due to the exploitation of the Kronecker structure and the use of matrix iterative methods.

Let $\lambda(\widehat{\mathbf{S}}^{-1}\mathbf{S}) \in [\lambda_{\mathbf{S}}, \Lambda_{\mathbf{S}}]$. In what follows we consider how each of the three preconditioners for the saddle point problem (3) are affected by the approximation of $\widehat{\mathbf{S}}$ to $\mathbf{S}$. We now state bounds on the eigenvalues of the preconditioned saddle point system using each of the preconditioners introduced in section 2.2 with $\widehat{\mathbf{L}} = \mathbf{L}$.

Proposition 7. *With the definitions as stated above, the eigenvalues of $\mathcal{P}_D^{-1}\mathcal{A}$ are real, and satisfy:*

$$\lambda(\mathcal{P}_D^{-1}\mathcal{A}) \in \left[\frac{1 - \sqrt{1 + 4\Lambda_{\mathbf{S}}}}{2}, \frac{1 - \sqrt{1 + 4\lambda_{\mathbf{S}}}}{2} \right] \cup \{1\} \cup \left[\frac{1 + \sqrt{1 + 4\lambda_{\mathbf{S}}}}{2}, \frac{1 + \sqrt{1 + 4\Lambda_{\mathbf{S}}}}{2} \right].$$

Proof. The result directly comes from [37, Theorem 4.2.1]. □

We can see that obtaining an improved estimate of the Schur complement (in a spectral sense) will lead to tighter bounds on the eigenvalues of the preconditioned system when using $\mathcal{P}_D^{-1}\mathcal{A}$.

Corollary 8. *If $\widehat{\mathbf{S}} = \mathbf{S}$*

$$\lambda(\mathcal{P}_D^{-1}\mathcal{A}) \in \left\{ \frac{1 - \sqrt{5}}{2}, 1, \frac{1 + \sqrt{5}}{2} \right\}.$$

The quality of the Schur complement approximation also affects the bounds on the eigenvalues of the block triangular preconditioner.

Proposition 9. *With the definitions as stated above, the eigenvalues of $\mathcal{P}_{\mathcal{T}}^{-1}\mathcal{A}$ are given by $(s + p)(N + 1)$ units, and the remaining $s(N + 1)$ eigenvalues are given by the eigenvalues of $\mathbf{S}\widehat{\mathbf{S}}^{-1}$.*

Proof. We consider the product

$$
\mathcal{A}\mathcal{P}_{\mathcal{T}}^{-1} = \begin{pmatrix} \mathbf{D} & 0 & \mathbf{L} \\ 0 & \mathbf{R} & \mathbf{H} \\ \mathbf{L}^T & \mathbf{H}^T & 0 \end{pmatrix} \begin{pmatrix} \mathbf{D}^{-1} & 0 & \mathbf{D}^{-1}\mathbf{L}\widehat{\mathbf{S}}^{-1} \\ 0 & \mathbf{R}^{-1} & \mathbf{R}^{-1}\mathbf{H}\widehat{\mathbf{S}}^{-1} \\ 0 & 0 & -\widehat{\mathbf{S}}^{-1} \end{pmatrix}
$$
$$
= \begin{pmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \mathbf{L}^T\mathbf{D}^{-1} & \mathbf{H}^T\mathbf{R}^{-1} & \mathbf{S}\widehat{\mathbf{S}}^{-1} \end{pmatrix}.
$$
(27)

The eigenvalues of $\mathcal{A}\mathcal{P}_{\mathcal{T}}^{-1}$, and by similarity $\mathcal{P}_{\mathcal{T}}^{-1}\mathcal{A}$ are therefore given by 1 and the eigenvalues of $\mathbf{S}\widehat{\mathbf{S}}^{-1}$. □

Corollary 10. *If $\widehat{\mathbf{S}} = \mathbf{S}$, then $\lambda(\mathcal{P}_{\mathcal{T}}^{-1}\mathcal{A}) = 1$.*

Proposition 11. *With the definitions as stated above, the eigenvalues of $\mathcal{P}_C^{-1}\mathcal{A}$ consist of $(2s - p)(N + 1)$ unit eigenvalues, with the remaining $2p(N + 1)$ eigenvalues given by*

$$\lambda(\mathcal{P}_C^{-1}\mathcal{A}) = 1 \pm \sqrt{\lambda_i(\mathbf{R}^{-1}\mathbf{H}\mathbf{L}^{-1}\mathbf{D}\mathbf{L}^{-T}\mathbf{H}^T)}i$$
(28)

Proof. The proof comes from [13, Appendix A]. □

5. Numerical experiments

5.1. Experimental framework

In this section we display some numerical results achieved by the novel preconditioning framework we presented in this paper for the two problems of interest (2) and (3). Our matrix-oriented strategy is compared to state-of-the-art vector-oriented approaches designed for linear systems stemming from data assimilation problems. In particular, we consider the scheme from [47] where a user-specified parameter k defines the approximation $\widehat{\mathbf{L}}^{-1}$. This is used within the Hessian/Schur complement approximation $\widehat{\mathbf{S}} = \widehat{\mathbf{L}}^T \mathbf{D}^{-1} \widehat{\mathbf{L}}$; see [47, Section 4] for further details.[5] We note that the approach of [47] is designed to increase parallelisability of the application of $\widehat{\mathbf{L}}$ and hence a preconditioner. All experiments presented in this section are performed in serial; by exploiting parallel architectures we expect to see large decreases in wallclock times for the approach of [47] when $k << N + 1$, which is not the case for our new strategy.

We also compare matrix-oriented CG with the improved Schur complement (as presented in section 3.2) against the limited memory preconditioner (LMP) approach of [9], where an alternative identity plus low-rank preconditioner is applied as a second level preconditioner. This method can only be implemented in the vectorised setting, and requires the use of $\widehat{\mathbf{L}} \equiv \mathbf{L}$. Hence, in its current formulation, the LMP approach cannot be easily parallelised, making comparison of wallclock times with the matrix-oriented approach more meaningful than for the approach of [47].

As previously mentioned, we employ a matrix-oriented implementation of CG (respectively GMRES) whereas the standard *vector* form of CG (resp. GMRES) is adopted whenever a preconditioning strategy different from the one introduced in this paper is considered. This is mainly due to the possibility of using existing code for the preconditioners in [14,17,47]. Indeed, these routines have been designed for standard GMRES and CG and not for their matrix-oriented counterpart. Notice, however, that the two GMRES/CG implementations are equivalent in exact arithmetic as we do not perform any low-rank truncation within matrix GMRES/CG. On the other hand, the matrix-oriented form of GMRES/CG may present some computational advantages due to the Kronecker form of the blocks of the coefficient matrix $\mathcal{A}$ in (3) and $\mathbf{S}$ in (2). See, e.g., Table 5.

In all the results reported here, the algorithms have been stopped as soon as the iterative method (either in matrix or vector form) relative residual norm becomes smaller than 10^{-8}. The same threshold has been used for the inner CG relative residual norm when we adopt the strategy presented in section 3.2.1.

In what follows, the matrix-oriented implementation of CG (resp. GMRES) will be denoted by MATCG (resp. MATGMRES) whereas its standard, vector counterpart by VECCG (resp VECGMRES).

All the experiments have been run using Matlab (version 2022a) on a machine with a 1.8GHz Intel quad-core i7 processor with 15GB RAM on an Ubuntu 20.04.2 LTS operating system.

For both the problem settings we considered, we used data assimilation terms based on those introduced in [47], which we now present briefly. For all experiments the dimension of the state is $s = 1000$ and the number of observations at each observation time is given by $p = 500$. The background error covariance matrix and model error covariance matrices are produced using an adapted SOAR correlation function [47, Equation (15)] with parameters $L_B = 0.6$, $L_Q = 0.75$, $\sigma_B = 0.5$, $\sigma_Q = 0.2$, 100 non-zero entries per row for B and 120 for Q. The observation error covariance matrix $R \in \mathbb{R}^{500 \times 500}$ is produced using the block approach of [47]. The observation operator $H \in \mathbb{R}^{500 \times 1000}$ has a single unit entry per row, arranged in ascending column order. Each of these terms is repeated in a Kronecker structure to obtain $\mathbf{D} = e_1 e_1^T \otimes B + (I_{N+1} - e_1 e_1^T) \otimes Q \in \mathbb{R}^{1000(N+1) \times 1000(N+1)}$, $\mathbf{R} = I_{N+1} \otimes R \in \mathbb{R}^{500(N+1) \times 500(N+1)}$, and $\mathbf{H} = I_{N+1} \otimes H \in \mathbb{R}^{500(N+1) \times 1000(N+1)}$. We discuss the two classes of model matrices in the relevant section.

5.2. Results for Lorenz96

Our first example is the Lorenz96 problem [28], a nonlinear set of coupled ODEs that is often used as a data assimilation test problem due to its chaotic nature.

Consider s equally spaced points on the unit line, e.g. $\Delta x = \frac{1}{s}$. For $i = 1, \ldots, s$, we consider

$$\frac{dx_i}{dt} = (x_{i+1} - x_{i-2})x_{i-1} - x_i + 8,$$

with periodic boundary conditions (i.e. $x_{-1} = x_{s-1}, x_0 = x_s$). We discretize the equations above by using the numerical implementation of [10], which integrates the model forward in time using a fourth-order Runge-Kutta scheme. We consider $s = 1000$, and unless otherwise mentioned, we use $\Delta t = 10^{-6}$.

As illustrated in Proposition 2, we expect the strategy proposed in section 3 to work well whenever the selected $\widehat{M}$ has small spectral norm and is such that the matrices $D_i = \widehat{M} - M_i$ have small spectral norm for all $i = 1, \ldots, N$ as well. In Table 1 we report $\|\widehat{M}\|$ and $\max_m \lambda_{\max}(D_m^T D_m)$ for $N = 10$ and different choices of $\widehat{M}$ varying Δt. In particular, we consider as $\widehat{M}$ the symmetrised first and the last matrices in the block subdiagonal of $\mathbf{L}$, namely $Sym(M_1) = 1/2(M_1 + M_1^T)$ and

---

[5] Notice that choosing $\widehat{\mathbf{L}} = \mathbf{L}$ is equivalent to setting $k = N + 1$ in [47].

Table 1

Example 5.2. $\|\widehat{M}\|$, $\max_m \lambda_{\max}(D_m^T D_m)$ and the upper bound (9) of Proposition 2 for different selection of $\widehat{M}$ and $N = 10$.

Δt	$\widehat{M}$	$Sym(M_1)$	$Sym(M_{10})$	$Karch(M_i)$
1×10^{-6}	$\|\widehat{M}\|$	1.000	1.0023	1.0011
	$\max_m \lambda_{\max}(D_m^T D_m)$	1.7641×10^{-5}	1.7641×10^{-5}	1.3981×10^{-5}
	Upper bound (9)	1.0012	1.0012	1.0001
1×10^{-3}	$\|\widehat{M}\|$	0.9980	1.0110	1.0046
	$\max_m \lambda_{\max}(D_m^T D_m)$	1.401×10^{-2}	1.401×10^{-2}	1.009×10^{-2}
	Upper bound (9)	1.3796	1.3912	1.3261
5×10^{-2}	$\|\widehat{M}\|$	0.9980	1.0110	1.0046
	$\max_m \lambda_{\max}(D_m^T D_m)$	1.401×10^{-2}	1.401×10^{-2}	1.009×10^{-2}
	Upper bound (9)	66.9385	11278	320.97
1×10^{-1}	$\|\widehat{M}\|$	0.8932	8.2359	1.5188
	$\max_m \lambda_{\max}(D_m^T D_m)$	7.9840	10.8191	7.7113
	Upper bound (9)	516.7533	2.148×10^{10}	8828.74

Table 2

Example 5.2. Iterations (top) and wallclock time (bottom) to convergence for the Lorenz96 problem with $\Delta t = 10^{-6}$ for $N = 10$ for different preconditioners. Values are averaged over 10 realisations. The pre-computation for the Karcher mean took 28.7684 seconds.

	$\widehat{S}$	$\mathcal{P}_{\mathcal{D}}$	$\mathcal{P}_{\mathcal{T}}$	$\mathcal{P}_{\mathcal{C}}$
MATCG/MATGMRES, $\widehat{M} = Sym(M_1)$, $r = 0$	25.7	45.0	29.0	46.0
MATCG/MATGMRES, $\widehat{M} = Sym(M_{10})$, $r = 0$	25.8	45.0	29.0	46.0
MATCG/MATGMRES, $\widehat{M} = karch(M_i)$, $r = 0$	26.3	45.0	28.1	46.0
MATCG/MATGMRES, $\widehat{M} = Sym(M_1)$, $r = p$	3.8	7.0	6.0	-
MATCG/MATGMRES, $\widehat{M} = Sym(M_{10})$, $r = p$	3.2	7.0	6.0	-
MATCG/MATGMRES, $\widehat{M} = karch(M_i)$, $r = p$	3.5	7.0	6.0	-
VECCG/VECGMRES [47], $k = 3$	529.5	358.0	188.7	66.7
MATCG/MATGMRES, $\widehat{M} = Sym(M_1)$, $r = 0$	9.3816	7.5924	4.9876	7.0108
MATCG/MATGMRES, $\widehat{M} = Sym(M_{10})$, $r = 0$	9.3436	7.7803	5.0595	6.9689
MATCG/MATGMRES, $\widehat{M} = karch(M_i)$, $r = 0$	9.4084	7.6916	4.8374	6.8653
MATCG/MATGMRES, $\widehat{M} = Sym(M_1)$, $r = p$	3.3120	3.2902	3.6184	-
MATCG/MATGMRES, $\widehat{M} = Sym(M_{10})$, $r = p$	3.0478	3.5083	3.7205	-
MATCG/MATGMRES, $\widehat{M} = karch(M_i)$, $r = p$	3.4022	3.4488	3.7182	-
VECCG/VECGMRES [47], $k = 3$	222.9978	153.9620	91.1391	19.7812

$Sym(M_{10}) = 1/2(M_{10} + M_{10}^T)$, and the Karcher mean[6] of the symmetric parts of the matrices M_i, $i = 1, \ldots, 10$, namely $Sym(M_i) = 1/2(M_i + M_i^T)$, as these are all SPD. In what follows, we denote the Karcher mean by $Karch(M_i)$. We also report the computed upper bound of Proposition 2.

For $\Delta t \leq 1 \times 10^{-3}$ the aforementioned selections of $\widehat{M}$ lead to very similar results. As $\|\widehat{M}\|$ is close to one and $\max_m \lambda_{\max}(D_m^T D_m)$ is rather small, the upper bound on the eigenvalues of $\widehat{L}^{-T} L^T L \widehat{L}^{-1}$ is also close to 1. As Δt increases, the norm of the linearised M_i operators moves further away from 1, leading to increases in the upper bound. For the Lorenz96 problem, $\|M_i\|$ increases monotonically with i, meaning that for larger choices of Δt the norm of $Sym(M_{10})$ is much larger than 1. We see that in the case $\Delta t = 10^{-1}$ the choice of $\widehat{M}$ makes a large difference to the upper bound in Proposition 2. In section 3, we proposed to select $\widehat{M}$ with the smallest norm in order to minimise the upper bound on the preconditioned spectrum, or by minimising $\max_m \lambda_{\max}(D_m^T D_m)$ in the case that the norms have similar values. This approach is supported by the results of Table 1, and motivates the selection of $\widehat{M} = Sym(M_1)$ for this problem for the experiments that follow.

In Table 2 we report the performance achieved by our matrix-oriented preconditioning frameworks: $\widehat{S}$ (with both $r = 0$ and $r = p$ in (18)) for (2) and $\mathcal{P}_{\mathcal{D}}$, $\mathcal{P}_{\mathcal{T}}$ (with both $r = 0$ and $r = p$ in (18)), and $\mathcal{P}_{\mathcal{C}}$ for (3), varying $\widehat{M}$. We compare these results with those attained by the strategy proposed in [47], where $\widehat{L}$ is chosen as follows:

$$\text{the } (i, j)\text{th block of } \widehat{L} = \begin{cases} \mathbf{I}, & i = j, \\ -M_i, & i = j \text{ and } i - k\lfloor \frac{i}{k} \rfloor, \\ 0, & \text{otherwise.} \end{cases}$$

As this $\widehat{L}$ does not have the form of a Stein equation, it is applied using VECCG/VECGMRES, with the parameter $k = 3$.

---

[6] The Karcher mean is computed by means of the routine `positive_definite_karcher_mean` included in the Matlab toolbox Manopt 6.0 [6]. As optimization procedure, we adopted the Barzilai-Borwein approach presented in [24] within `positive_definite_karcher_mean`.

Table 3

Example 5.2. Iterations (top) and wallclock time (bottom) to convergence for the Lorenz96 problem with $N = 100$ for different preconditioners. 10 realisations. VECGM-RES with $\mathcal{P}_D$ and $\widehat{\mathbf{L}}^{-1}$ computed as in [47] ($k = 3$) did not converge in 1000 iterations.

	$\widehat{\mathbf{S}}$	$\mathcal{P}_D$	$\mathcal{P}_T$	$\mathcal{P}_C$
MATCG/MATGMRES, $\widehat{M} = Sym(M_1)$, $r = 0$	213	239	173.5	242.5
MATCG/MATGMRES, $\widehat{M} = Sym(M_1)$, $r = p$	9	15	14	-
VECCG/VECGMRES [47], $k = 3$	955.4	-	938.1	570.9
MATCG/MATGMRES, $\widehat{M} = Sym(M_1)$, $r = 0$	827.34	499.70	356.71	504.56
MATCG/MATGMRES, $\widehat{M} = Sym(M_1)$, $r = p$	94.80	105.70	116.43	-
VECCG/VECGMRES [47], $k = 3$	4757	5054	5553	2243

Table 4

Example 5.2. Iterations (left) and wallclock time (right) to convergence for the Lorenz96 problem with $N = 100$ for different preconditioners which approximate $\widehat{\mathbf{S}}$. Values are averaged over 10 realisations.

Preconditioner	Iterations	Wallclock time
MATCG, $\widehat{M} = Sym(M_1)$, $r = 0$	212.5	845.71
MATCG, $\widehat{M} = Sym(M_1)$, $r = p$	9	98.98
VECCG, $k = N + 1$, $r = 0$	178.5	967.73
VECCG, $k = N + 1$, $r = 10$	126.2	958.20
VECCG, $k = N + 1$, $r = 10(N + 1)$, LMP	28	119.22

From the results in Table 2 we can see that the use of the $\widehat{\mathbf{L}}$ proposed in section 3 is very effective in reducing the overall iteration count for all $\widehat{\mathbf{S}}$, $\mathcal{P}_D$, $\mathcal{P}_T$, and $\mathcal{P}_C$. The number of iterations achieved by $\widehat{\mathbf{S}}$, $\mathcal{P}_D$ and $\mathcal{P}_T$ with $r = p$ is remarkably small, especially when compared to the one attained by employing the $\widehat{\mathbf{L}}$ coming from [47] with $k = 3$. These small numbers of iterations impact on the wallclock time of the overall solution problem too with $\widehat{\mathbf{S}}$, $r = p$, with $\widehat{M} = Sym(M_{10})$ being the fastest approach we tested. However, we note that the approach of [47] is designed to increase parallelisability of preconditioners, and significant speed up is to be expected for this preconditioner in a parallel setting (all experiments presented here are performed in serial).

In Table 3 we report the results obtained for $N = 100$ for a selection of parameters. For our new strategy, we document the results achieved using $\widehat{M} = Sym(M_1)$. We can see that $\widehat{\mathbf{S}}$, $\mathcal{P}_D$, and $\mathcal{P}_T$ with $r = p$ lead to a very small number of MATCG/MATGMRES iterations also for this problem setting. Moreover, they scale much better than the strategy from [47] in terms of computational timing in this serial setting. However, we recall from Table 1 that the norm of the difference between the blocks M_i is small for $\Delta t = 10^{-6}$. This is the best scenario for our novel preconditioning approach.

In Table 4 we consider the performance of different approximations $\widehat{\mathbf{S}}$ for the SPD problem (2). In addition to the low-rank approach presented in section 3.2 we consider the limited memory preconditioning (LMP) approach studied in [9]. The LMP approach approximates eigenvalues of $\mathbf{I} + \mathbf{D}^{1/2}\mathbf{L}^{-T}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{L}^{-1}\mathbf{D}^{1/2}$, i.e. symmetrically preconditioning with the exact $\mathbf{L}^T\mathbf{D}^{-1}\mathbf{L}$ operator. Spectral information is typically approximated using randomised numerical linear algebra approaches. In addition to requiring spectral information of a much large linear system than the approach considered in section 3.2, LMP requires the use of $\widehat{\mathbf{L}} = \mathbf{L}$, meaning that it cannot be readily applied in the matrix-oriented approach. Hence, computing the full spectrum of $\mathbf{I} + \mathbf{D}^{1/2}\mathbf{L}^{-T}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{L}^{-1}\mathbf{D}^{1/2}$ is prohibitively costly both in terms of storage and computation. We therefore compare MATCG preconditioned with $\widehat{\mathbf{S}}$ as in (18) with $r = 0, r = p$ with VECCG preconditioned in two different ways. In the first place, we use $\widehat{\mathbf{S}}$ as in (17) where the inverse of the exact $\mathbf{L}$ is computed by means of the algorithm in [47] by setting $k = N + 1$. The low-rank term $\mathbf{K}_r\mathbf{K}_r^T$ is constructed by considering $r = 10$, or $r = p$ eigenpairs of $H^T R H$. The second preconditioning approach for VECCG is given by LMP. Also in this case $\mathbf{L}^{-1}$ is computed by following [47] with $k = N + 1$. In LMP, the rank of the low-rank approximation to $\mathbf{D}^{1/2}\mathbf{L}^{-T}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{L}^{-1}\mathbf{D}^{1/2}$ is set to $10(N + 1)$. Notice that this means that we are employing the same number of eigenpairs as in (17) for $r = 10$. Indeed, if $V_r \Upsilon_r V_r^T \approx H^T R H$, $V_r \in \mathbb{R}^{p \times 10}$, then $\mathbf{K}_r = I_{N+1} \otimes V_r \Upsilon_r^{1/2}$ has rank $10(N + 1)$.

We see that in terms of iterations the LMP approach results in much larger reductions than the approach of section 3.2 for the same number of eigenpairs. However, by exploiting the Kronecker structure of the new preconditioning approach, we can incorporate many more terms, leading to very small number of iterations for MATCG with $r = p$. We also note the improvement in wallclock times when using the matrix-oriented approach, with the fastest times occurring for MATCG with $r = p$.

Fig. 1 shows how the Kronecker preconditioners perform as Δt increases, and the difference between linearised model operators increases. We compare against the approach of [47] using $k = N + 1$, i.e. $\widehat{\mathbf{L}} \equiv \mathbf{L}$. For the SPD problem, we also plot convergence for the vectorised approach with LMP. For $\Delta t \in [10^{-6}, 10^{-2}]$ both MATCG/MATGMRES and VECCG/VECGMRES behave similarly, with only a small increase in iterations with Δt. However, for larger values of Δt the number of iterations required to reach convergence increases for all methods, with the largest impact seen for MATCG/MATGMRES. The use of $r = p$ within MATCG/MATGMRES is much more sensitive to the choice of Δt, with a steady increase in the number of
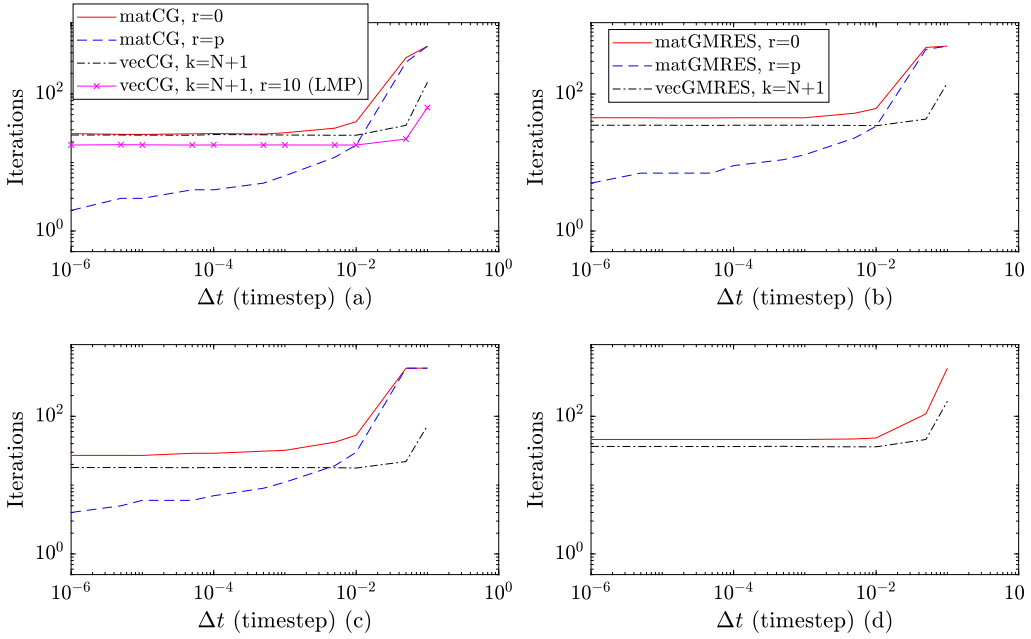
Fig. 1. Example 5.2. Iterations to reach convergence with changing time discretization, Δt, for the Lorenz96 problem for different choices of preconditioner for $N = 10$. Panel (a) shows $\widehat{\mathbf{S}}$, (b) shows $\mathcal{P}_\mathcal{D}$, (c) shows $\mathcal{P}_\mathcal{T}$ and (d) shows $\mathcal{P}_\mathcal{C}$. For all panels the red solid line represents MATGMRES/MATCG with $r = 0$ for $\widehat{M} = Sym(M_1)$, blue dashed line represents MATGMRES/MATCG with $r = p$, and black dot-dashed line VECGMRES/VECCG for $k = N + 1$. For panel (a) the cyan solid line with cross markers shows VECCG for $k = N + 1$ and $r = 10(N + 1)$ using the LMP approach. We report averaged behaviour over 10 realisations. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

iterations required to reach convergence as Δt increases. For $\Delta t = 5 \times 10^{-2}$ there is negligible benefit in terms of iterations to the inclusion of observation information within the Schur complement. For large values of Δt, the difference between $\widehat{\mathbf{L}} = I_{N+1} \otimes I_s - \Sigma \otimes \widehat{M}$ and $\mathbf{L}$ increases, meaning that including $r > 0$ factors coming from the observation term is obtaining an improved estimate of the wrong preconditioner. In the future it might be possible to design alternative additional terms that can correct for this discrepancy. We note that for LMP as $\widehat{\mathbf{L}} \equiv \mathbf{L}$ the 'correct' low-rank update is used for all choices of Δt.

5.3. Results for heat equation

We now present an example with $M_i = M$ for all i. This simpler setting allows us to validate the theoretical properties of our new approach, and consider computational aspects such as the cost of the approach proposed in section 3.2 and scaling of our methods with increasing observation times. Our second numerical example comes from [47, Section 6.1]. For this example, $\mathbf{L}$ is a Stein operator of the form (22) where the matrix M amounts to the discrete operator stemming from the discretization of the one-dimensional heat equation on the unit line

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$$

with Dirichlent boundary conditions $u(0, t) = u_0$, $u(1, t) = u_1$ for all $t \in (0, 1]$. By discretising the equation above by means of the forward Euler method in time and second-order central differences in space, M can be written as follows

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1-2r & r & 0 & \cdots & 0 & 0 \\ 0 & r & \ddots & \ddots & & \vdots & \vdots \\ 0 & 0 & \ddots & \ddots & \ddots & 0 & 0 \\ \vdots & & \ddots & \ddots & \ddots & r & 0 \\ 0 & \cdots & 0 & 0 & r & 1-2r & 0 \\ 0 & \cdots & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad r = \frac{\Delta t}{(\Delta x)^2}.$$

For this example we use $\Delta x = 10^{-3}$ and $\Delta t = 4 \times 10^{-7}$.

Table 5
Example 5.3. Iterations and wallclock time to convergence for $N = 10$ and $\widehat{\mathbf{L}} \equiv \mathbf{L}$ for the objective function formulation (2) (column 1) and the saddle point formulation (3) with different preconditioners (columns 2-4) averaged over 10 realisations.

	$\widehat{\mathbf{S}}$	$\mathcal{P}_{\mathcal{D}}$	$\mathcal{P}_{\mathcal{T}}$	$\mathcal{P}_{\mathcal{C}}$
Iterations – vecCG/vecGMRES	18.4	36.6	18.8	37.6
Iterations – matCG/matGMRES	18.4	36.6	18.8	37.6
Wallclock time – vecCG/vecGMRES	6.2475	7.7263	4.1889	0.4013
Wallclock time – matCG/matGMRES	1.3356	0.8472	0.8493	0.5713

We begin by comparing the performance of the matrix and vector oriented approaches when using the exact $\widehat{\mathbf{L}} \equiv \mathbf{L}$ in all four choices of preconditioner proposed in this paper, and $r = 0$ in (18). For vecCG/vecGMRES, $\mathbf{L}^{-1}$ is computed by using the procedure coming from [47] for $k = N + 1$. We remind the reader that the two versions of CG/GMRES are equivalent in exact arithmetic. However, the matrix oriented approaches present some computational advantages for this example. Table 5 demonstrates the remarkable gain in efficiency that occurs for $\widehat{\mathbf{S}}$, $\mathcal{P}_{\mathcal{D}}$, and $\mathcal{P}_{\mathcal{T}}$ for $N = 10$. We notice that the strategy based on $\mathcal{P}_{\mathcal{C}}$ is faster than the ones related to the block diagonal and block triangular preconditioners, in spite of the fact that it requires a larger number of iterations. This is due to the small N selected in this example which makes the inversion of $\mathbf{D}$ the most expensive step in the preconditioners that involve $\widehat{\mathbf{S}}$. We recall that $\mathcal{P}_{\mathcal{C}}$ avoids the application of $\mathbf{D}^{-1}$. Moreover, for this choice of N, vecGMRES is faster than matGMRES when $\mathcal{P}_{\mathcal{C}}$ is adopted as preconditioning operator. This is related to the cost of the eigendecomposition of M described at the end of section 3.1. This step cubically depends on s; taking about 0.2 s for this problem, namely about $1/2$ of the overall running time achieved by matGMRES. Nevertheless, we anticipate the cost of this eigendecomposition to be amortised for larger choices of N (see e.g. Fig. 2).

We now focus on the performance achieved by the novel Schur complement approximation (17) and its implementation illustrated in section 3.2.1. To this end, we consider only $\widehat{\mathbf{S}}$ and $\mathcal{P}_{\mathcal{D}}$. Similar results have also been obtained for $\mathcal{P}_{\mathcal{T}}$ but are not presented here. In Table 6 (left), for different values of N, we report the iteration count and the overall running time of matCG/matGMRES when the approximate Schur complement $\widehat{\mathbf{S}}$ is adopted for $\mathbf{S}$ and $\mathcal{P}_{\mathcal{D}}$ for $r = 0$ and $r = p$. We notice that choosing $r = p$ in (17) leads to a remarkable decrease in the number of iterations needed to converge. Moreover, the number of CG/GMRES iterations we perform turns out to be N-independent for both problem.

We notice that the number of iterations required to solve the matCG formulation is smaller than for matGMRES, leading to smaller wallclock times. We note that for $r = p$ we are approximating the inverse of $\mathbf{S}$ to a small tolerance, and hence obtain convergence in a single iteration of matCG. Wallclock times for the case $r = p$ are comparable for both problems. Even though the use of inner CG introduces some inexactness in the preconditioning step, we notice that the number of iterations performed by matCG/matGMRES with $\widehat{\mathbf{S}}$ and $r = p$ is independent of N and equal to the number of iterations expected in case of an exact computation of $\widehat{\mathbf{S}}^{-1}$.

The large reduction in the iteration count also leads to a significant speed-up of the overall solution process, which is not obvious in general. Indeed, the use of (18) for large values of r can be computationally demanding due to the need to solve the linear system with $I_{r(N+1)} + \mathbf{K}_r^T \mathbf{L}^{-1} \mathbf{D} \mathbf{L}^{-T} \mathbf{K}_r$. However, thanks to the matrix CG strategy presented in section 3.2.1, which takes full advantage of the semi-diagonalization of $\mathbf{L}$, dealing with $\widehat{\mathbf{S}}$ by (18) turns out to be computationally affordable for $r = p$. In Table 6 (far right), we report the number of CG iterations and the related running time needed to approximately invert $\widehat{\mathbf{S}}$ within the matGMRES iteration. We remind the reader that, in light of Theorem 1, $\widehat{\mathbf{S}}^{-1}$ has to be computed only every other GMRES iteration. From the results in Table 6 (right), we can see that the CG steps correspond to a small proportion of the overall GMRES running time for small N. However, as N increases, the cost of the inner CG iteration becomes a larger proportion of the overall wallclock time. The number of CG iterations increases with N, leading to a more demanding preconditioning step for larger numbers of observation times. In this scenario, equipping the inner CG solve with effective preconditioning operators may be largely beneficial. However, as we previously mentioned, natural preconditioning candidates were not able to reduce the CG iteration count without significantly increasing its computational cost per iteration. We will explore this challenging topic of designing bespoke preconditioners for the inner CG solver in the future.

We conclude the heat equation example by comparing the novel preconditioning operators developed in this paper with state-of-the-art approaches. In particular, we consider matCG equipped with $\widehat{\mathbf{S}}$ and matGMRES equipped with $\mathcal{P}_{\mathcal{D}}$, $\mathcal{P}_{\mathcal{T}}$ (with both $r = 0$ and $r = p$ in (17)), and $\mathcal{P}_{\mathcal{C}}$. Using $k = N + 1$ within the strategy of [47] becomes infeasible for large values of N. We therefore use the strategy coming from [47] with $k = 3$ to approximate $\widehat{\mathbf{L}}^{-1}$ as a comparison. This is implemented with vecCG/vecGMRES and is otherwise equipped with the same preconditioning frameworks as our novel approach.

Fig. 2 (top) shows the number of iterations to reach convergence with an increasing number of observation times N. We can observe that $\widehat{\mathbf{S}}$, $\mathcal{P}_{\mathcal{D}}$ and $\mathcal{P}_{\mathcal{T}}$ require a very small number of iterations whenever we select $r = p$ in (17). The performance of these operators is optimal as the number of performed CG/GMRES iterations is constant with increasing N. We recall that such optimality is not guaranteed, as the linear system with $I_{r(N+1)} + \mathbf{K}_r^T \mathbf{L}^{-1} \mathbf{D} \mathbf{L}^{-T} \mathbf{K}_r$ involved in (18) is solved iteratively to a relative residual tolerance of 10^{-8}. As previously mentioned, such *inexactness* does not allow us to claim that we are working within the scenarios depicted in Corollary 8–10 – for which we would be able to guarantee an N-independent number of CG/GMRES iterations – even though $r = p$ in (17). Nevertheless, for this example the overall solution process does demonstrate N independence of iterations to reach convergence.

Table 6

Example 5.3. Iterations and wallclock time to convergence for $\widehat{\mathbf{S}}$ (first two columns) $\mathcal{P}_{\mathcal{D}}$ (columns 3-6) with the two different choices $r = 0$ and $r = p$ in (17) as N varies. In columns 5 and 6 we report the iterations and wallclock time needed by inner CG method to solve the linear system with $I_{r(N+1)} + \mathbf{K}_r^T \mathbf{L}^{-1} \mathbf{DL}^{-T} \mathbf{K}_r$ involved in (18) when $r > 0$ in $\mathcal{P}_{\mathcal{D}}$.

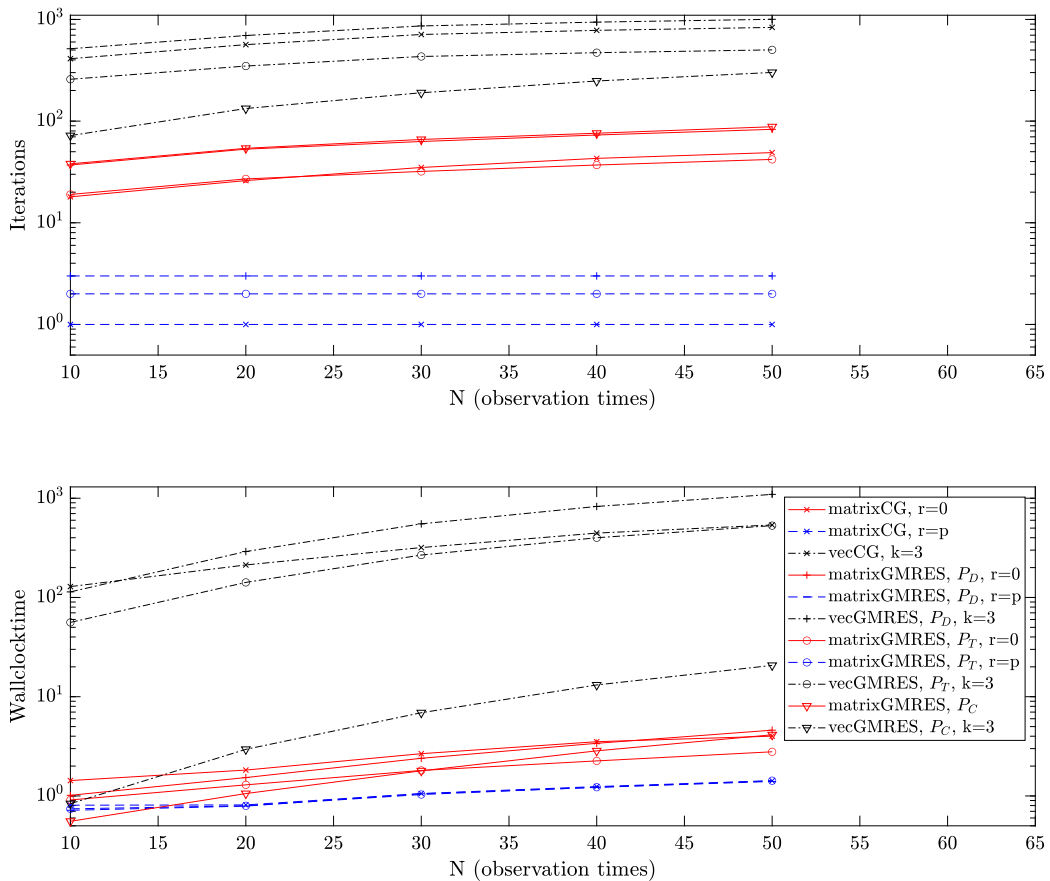| | MATCG | | MATGMRES | | Inner CG 2nd GMRES it. | |
N	Its.	Time	Its.	Time	Its.	Time
10 ($r=0$)	18.3	1.1732	36.4	1.4561		
($r=p$)	1	0.6683	3	0.6332	20	0.0598
20 ($r=0$)	26.1	1.8153	51.8	2.1692		
($r=p$)	1	0.8008	3	0.7577	30.3	0.1420
30 ($r=0$)	34.2	2.3684	63.0	2.9455		
($r=p$)	1	0.9045	3	0.9111	40	0.2674
40 ($r=0$)	42.1	2.9996	74.2	4.0273		
($r=p$)	1	1.0495	3	1.0598	49.3	0.4227
50 ($r=0$)	49.5	3.5876	83.0	5.1370		
($r=p$)	1	1.2258	3	1.2289	59	0.5731
60 ($r=0$)	57.1	4.2990	92.6	6.7560		
($r=p$)	1	1.5402	3	1.5267	67	0.8732

Fig. 2. Example 5.3. Iterations (top) and wallclock time (bottom) to reach convergence with increasing problem size (number of observation times) for the heat equation problem for different choices of preconditioner. Red solid lines denote MATCG/MATGMRES with $r = 0$, blue dashed lines denote MATCG/MATGMRES with $r = p$, and black dot-dashed lines denote VECCG/VECGMRES with $k = 3$. Crosses denote $\widehat{\mathbf{S}}$, pluses denote $\mathcal{P}_{\mathcal{D}}$, circles denote $\mathcal{P}_{\mathcal{T}}$ and triangle denote $\mathcal{P}_{\mathcal{C}}$. We report averaged behaviour over 10 realisations.

Competitive performance is attained also when $r = 0$ in the $\widehat{\mathbf{S}}$, $\mathcal{P}_{\mathcal{D}}$, and $\mathcal{P}_{\mathcal{T}}$ preconditioning frameworks compared to the $\mathbf{L}^{-1}$ approximation with $k = 3$. The large difference in iterations for the strategy coming from [47] also leads to much longer wallclock times.

We notice that the use of $\mathcal{P}_{\mathcal{C}}$ leads to a number of GMRES iterations which is always very similar to the one achieved by $\mathcal{P}_{\mathcal{D}}$ with $r = 0$. The performance of $\widehat{\mathbf{S}}$ is also similar to the performance of $\mathcal{P}_{\mathcal{T}}$ in the $r = 0$ setting.

In Fig. 2 (bottom) we report the computational time of the overall CG/GMRES solution process for all the preconditioning operators we mentioned above. We can see that, except for $\mathcal{P}_{\mathcal{C}}$ with small N, vecGMRES is much slower than matGMRES equipped with our novel preconditioning strategies. In particular, from the results depicted in Fig. 2 (bottom) we see that selecting $r = p$ in $\widehat{\mathbf{S}}$, $\mathcal{P}_{\mathcal{D}}$ and $\mathcal{P}_{\mathcal{T}}$ is a favourable choice over $r = 0$ also in terms of running time, with competitive scaling with N when using the improved Schur complement approximation.

6. Conclusions and outlook

To fully exploit the rich Kronecker structure of the matrices stemming from weak-constraint 4D-Var problems, matrix-oriented Krylov methods can be employed to solve both (2) and (3). The use of such machinery naturally leads to the design of new preconditioning approaches. In particular, by selecting a fresh option for the operator $\widehat{\mathbf{L}}$ whose inversion can be recast in terms of the solution of a Stein matrix equation, we designed improved preconditioners able to drastically reduce the Krylov iteration count for certain problems. Our new approach also allows for the efficient inclusion of information from the observation term of the Schur complement $\mathbf{S}$, leading to more accurate approximations $\widehat{\mathbf{S}}$.

In the case of observation-time independent forecast models $\mathcal{M}$, our new preconditioning frameworks achieve optimal performance in terms of the number of iterations, remarkably without increasing the computational cost of the overall solution process.

The implementation presented in this paper requires a number of assumptions on the structure of the data assimilation system, which we hope to relax in future work. Firstly, the machinery developed here relies on having a moderate spatial dimension s. This assumption is crucial for the Stein equation solution scheme presented in section 3.1. We plan to extend the preconditioning framework presented in this paper to the case of sizable s in near future. This can be achieved, e.g., by using projection-based methods for large-scale Stein equations. The approach currently also requires that a number of other components of the assimilation problem have a strict Kronecker structure, meaning that the model error and the observing system are constant for all observation times. As reported in section 3, we could approximate each term at the preconditioning level by means of some Kronecker forms. However, the selection of such approximations may be cumbersome. These aspects will be investigated elsewhere.

CRediT authorship contribution statement

Jemima M. Tabeart: Conceptualization, Methodology, Investigation, Software, Writing. **Davide Palitta:** Conceptualization, Methodology, Investigation, Writing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Jemima M Tabeart reports financial support was provided by Engineering and Physical Sciences Research Council.

Data availability

The code is publicly available at https://github.com/JemimaT/Stein4DVar.

Appendix A

Here we report the proof of Theorem 1.

Proof. We first write

$$
\mathcal{A}\mathcal{P}_{\mathcal{D}}^{-1} = \begin{bmatrix} I & 0 & \mathbf{L}\widehat{\mathbf{S}}^{-1} \\ 0 & I & \mathbf{H}\widehat{\mathbf{S}}^{-1} \\ \mathbf{L}^T\mathbf{D}^{-1} & \mathbf{H}^T\mathbf{R}^{-1} & 0 \end{bmatrix}.
$$

We show the statement by induction on $k \geq 1$.

For $k = 1$, we define

$$
v_1 = \frac{1}{\sqrt{\|b\|^2 + \|d\|^2}} \begin{bmatrix} b \\ d \\ 0 \end{bmatrix}.
$$

Then,

$$
\widetilde{v}_2 = \mathcal{A}\mathcal{P}_{\mathcal{D}}^{-1} v_1 = \frac{1}{\sqrt{\|b\|^2 + \|d\|^2}} \begin{bmatrix} b \\ d \\ \mathbf{L}^T\mathbf{D}^{-1}b + \mathbf{H}^T\mathbf{R}^{-1}d \end{bmatrix},
$$

and the latter vector needs to be orthogonalized with respect to v_1. A direct computation shows that the outcome of this orthogonalization is

$$
\widehat{v}_2 = \frac{1}{\sqrt{\|b\|^2 + \|d\|^2}} \begin{bmatrix} 0 \\ 0 \\ \mathbf{L}^T\mathbf{D}^{-1}b + \mathbf{H}^T\mathbf{R}^{-1}d \end{bmatrix},
$$

and $v_2 = \widehat{v}_2/\|\widehat{v}_2\|$.

We now assume that the result has been shown for a certain $\bar{k} > 1$ and we prove the inductive step for $\bar{k} + 1$.

It holds

$$
\widetilde{v}_{2(\bar{k}+1)-1} = \widetilde{v}_{2\bar{k}+1} = \mathcal{A}\mathcal{P}_{\mathcal{D}}^{-1} v_{2\bar{k}} = \begin{bmatrix} \mathbf{L}\widehat{\mathbf{S}}^{-1}z_{2\bar{k}} \\ \mathbf{H}\widehat{\mathbf{S}}^{-1}z_{2\bar{k}} \\ 0 \end{bmatrix}.
$$

Then the orthogonalization step is such that

$$
\widehat{v}_{2(\bar{k}+1)-1} = \widetilde{v}_{2(\bar{k}+1)-1} - \sum_{j=1}^{\bar{k}} \alpha_j v_{2j-1} - \sum_{j=1}^{\bar{k}} \beta_j v_{2j},
$$

and the only term that may potentially contribute to the third block of $\widehat{v}_{2(\bar{k}+1)-1}$ is

$$
\sum_{j=1}^{\bar{k}} \beta_j v_{2j} = \begin{bmatrix} 0 \\ 0 \\ \sum_{j=1}^{\bar{k}} \beta_j z_{2j} \end{bmatrix}.
$$

However, all the scalars β_j's are zero since

$$
\beta_j = \widetilde{v}_{2(\bar{k}+1)-1}^T v_{2j} = \left[(\mathbf{L}\widehat{\mathbf{S}}^{-1}z_{2\bar{k}})^T, (\mathbf{H}\widehat{\mathbf{S}}^{-1}z_{2\bar{k}})^T, 0 \right] \begin{bmatrix} 0 \\ 0 \\ z_{2j} \end{bmatrix} = 0.
$$

Therefore, $v_{2(\bar{k}+1)-1} = \widehat{v}_{2(\bar{k}+1)-1}/\|\widehat{v}_{2(\bar{k}+1)-1}\|$ has a third zero block.

To conclude, if $v_{2(\bar{k}+1)-1} = [u_{2(\bar{k}+1)-1}^T, w_{2(\bar{k}+1)-1}^T, 0]^T$, then

$$
\widetilde{v}_{2(\bar{k}+1)} = \mathcal{A}\mathcal{P}_{\mathcal{D}}^{-1} v_{2(\bar{k}+1)-1} = \begin{bmatrix} u_{2(\bar{k}+1)-1} \\ w_{2(\bar{k}+1)-1} \\ \mathbf{L}^T\mathbf{D}^{-1}u_{2(\bar{k}+1)-1} + \mathbf{H}^T\mathbf{R}^{-1}w_{2(\bar{k}+1)-1} \end{bmatrix},
$$

and orthonormalizing such a vector with respect to the computed basis, and in particular $v_{2(\bar{k}+1)-1}$, leads to a $v_{2(\bar{k}+1)}$ whose third block is the only nonzero block. Hence the result of Theorem 1 holds by induction. □

Here we report the proof of Proposition 2.

Proof. We begin by observing that $\widehat{\mathbf{L}}^{-T}\mathbf{L}^T\mathbf{L}\widehat{\mathbf{L}}^{-1} = I + A(\widehat{M})$ where the (i, j)th block of $A(\widehat{M})$ is given by

$$\begin{cases} \sum_{k=i}^{N} \widehat{M}^{(k-i)T} D_k^T D_k \widehat{M}^{k-i} & \text{if } i = j, \\ D_{i-1}\widehat{M}^{i-j-1} + \sum_{k=i}^{N} \widehat{M}^{(k-i)T} D_k^T D_k \widehat{M}^{k-j} & \text{if } i > j, \\ \widehat{M}^{(j-i-1)T} D_{j-1}^T + \sum_{k=j}^{N} \widehat{M}^{(k-i)T} D_k^T D_k \widehat{M}^{k-j} & \text{if } j > i. \end{cases} \tag{29}$$

We then write $A(\widehat{M}) = \sum_{m=1}^{N} A_m$ where A_m contains terms which depend only on D_m and powers of $\widehat{M}$ and their transposes only. We bound the eigenvalues of $A(\widehat{M})$ above by applying [4, Equation 5.12.2] to obtain

$$\lambda_{\max}(A(\widehat{M})) \le 1 + \sum_{k=1}^{N} \lambda_{\max}(A_k). \tag{30}$$

We now bound the eigenvalues of A_m. For $m = 1, \ldots, N$, the (i, j)th block of A_m is given by

$$\begin{cases} \widehat{M}^{(m-i)T} D_m^T D_m \widehat{M}^{m-j} & \text{if } i, j \le m, \\ \widehat{M}^{(m-i)T} D_m^T & \text{if } j = m+1 > i, \\ D_m \widehat{M}^{m-i} & \text{if } i = m+1 > j. \end{cases} \tag{31}$$

For all choices of m, A_m has $m^2 - 1$ non-zero blocks, and has rank $2s$. If $0_\ell \in \mathbb{R}^\ell$ denotes the zero vector of length ℓ, the $(m-2)s$ eigenvectors corresponding to zero take the form $(e_i, -\widehat{M}e_i, 0_{(m-2)s})$, or $(0_s, e_i, -\widehat{M}e_i, 0_{(m-1)s}, \ldots, (0_{(m-1)s}, e_i, -\widehat{M}e_i, 0_s)$.

The non-zero eigenvalues of A_m can be found by solving the $s \times s$ system

$$\left(D_m \left(\sum_{k=0}^{N-1} \widehat{M}^k \widehat{M}^{kT} \right) D_m^T \right) v = \frac{\mu^2}{\mu+1} v,$$

i.e.

$$\mu = 0.5(\rho \pm \sqrt{\rho^2 + 4\rho}), \tag{32}$$

where ρ are the eigenvalues of $(D_m(\sum_{k=0}^{m-1} \widehat{M}^k \widehat{M}^{kT}) D_m^T)$.

By the monotonicity of (32), the largest value of μ occurs for the largest value of ρ with the positive option, and the smallest value of μ occurs for the largest value of ρ taking the negative option. Therefore an upper bound for ρ provides us with an upper bound for μ, and hence $\lambda_{\max}(A(\widehat{M}))$.

By similarity

$$\max(\rho_m) = \lambda_{\max}(D_m^T D_m (\sum_{k=0}^{m-1} \widehat{M}^k \widehat{M}^{kT}))$$

$$\le \lambda_{\max}(D_m^T D_m)\lambda_{\max}(\sum_{k=0}^{m-1} \widehat{M}^k \widehat{M}^{kT}))$$

$$\le \lambda_{\max}(D_m^T D_m) \sum_{k=0}^{m-1} \lambda_{\max}(\widehat{M}^T \widehat{M})^k.$$

A loose upper bound can be obtained by defining

$$\rho_N = \max_m \lambda_{\max}(D_m^T D_m) \sum_{k=0}^{N-1} \lambda_{\max}(\widehat{M}^T \widehat{M})^k.$$

Moreover,

$$\sum_{k=0}^{N-1} \lambda_{\max}(\widehat{M}^T \widehat{M})^k = \begin{cases} N, & \text{if } \lambda_{\max}(\widehat{M}^T \widehat{M}) = 1, \\ \frac{1-\lambda_{\max}^N(\widehat{M}^T \widehat{M})}{1-\lambda_{\max}(\widehat{M}^T \widehat{M})}, & \text{otherwise.} \end{cases}$$

For every choice of m it holds $\mu_m \le 0.5(\rho_N + \sqrt{\rho_N^2 + 4\rho_N})$, therefore

$$\lambda_{max}(A(\widehat{M})) \le 1 + \frac{N}{2}(\rho_N + \sqrt{\rho_N^2 + 4\rho_N}). \quad \square$$

Here we report the proof of Proposition 3.

Proof. We show (14). The proof for (15) is analogous.

By plugging (13) into (11) we get

$$Z - \widehat{M}ZC^T + \widehat{M}Ze_{N+1}e_1^T = V.$$

Premultiplying by T^{-1} and postmultiplying by F^T yields

$$\widetilde{Z} - \Lambda\widetilde{Z}\Pi + \Lambda\widetilde{Z}(F^{-T}e_{N+1})(e_1^T F^T) = T^{-1}VF^T, \qquad \widetilde{Z} := T^{-1}ZF^T,$$

whose Kronecker form is given by

$$(I_{N+1} \otimes I_s - \Pi \otimes \Lambda + (Fe_1)(e_{N+1}^T F^{-1}) \otimes \Lambda)\text{vec}(\widetilde{Z}) = \text{vec}(T^{-1}VF^T).$$

If $\mathbf{G} := I_{N+1} \otimes I_s - \Pi \otimes \Lambda \in \mathbb{C}^{(N+1)s \times (N+1)s}$, $\mathbf{M} := Fe_1 \otimes \Lambda \in \mathbb{C}^{(N+1)s \times s}$, and $\mathbf{N} := F^{-T}e_{N+1} \otimes I_s \in \mathbb{C}^{(N+1)s \times s}$, the Sherman-Morrison-Woodbury formula [16, Equation (2.1.4)] shows that

$$\text{vec}(\widetilde{Z}) = \mathbf{G}^{-1}\text{vec}(T^{-1}VF^T) - \mathbf{G}^{-1}\mathbf{M}(I + \mathbf{N}^T\mathbf{G}^{-1}\mathbf{M})^{-1}\mathbf{N}^T\mathbf{G}^{-1}\text{vec}(T^{-1}VF^T).$$

Once $\widetilde{Z}$ is computed, we retrieve Z by performing $Z = T\widetilde{Z}F^{-T}$.

We now derive a cheap procedure for the computation of $\widetilde{Z}$ which does not involve the construction of any large matrix. We first notice that, since $\mathbf{G}$ is diagonal it holds

$$\text{vec}(Y) = \mathbf{G}^{-1}\text{vec}(T^{-1}VF^T) \quad \Longleftrightarrow \quad Y = P \circ (T^{-1}VF^T),$$

where $P \in \mathbb{C}^{s \times (N+1)}$ is such that $P_{i,j} = 1/(1 - \lambda_i\pi_j)$.

Moreover, by exploiting the Kronecker structure of $\mathbf{N}$, we have

$$\mathbf{N}^T\mathbf{G}^{-1}\text{vec}(T^{-1}VF^T) = ((e_{N+1}^T F^{-1}) \otimes I_s)\text{vec}(Y) = YF^{-T}e_{N+1}.$$

We now focus on the computation of $\mathbf{N}^T\mathbf{G}^{-1}\mathbf{M}$. We remind the reader that

$$\mathbf{G} = \begin{pmatrix} I_s - \pi_1\Lambda & & \\ & \ddots & \\ & & I_s - \pi_{N+1}\Lambda \end{pmatrix}, \ \mathbf{N} = \begin{pmatrix} (F^{-T}e_{N+1})_1 I_s \\ \vdots \\ (F^{-T}e_{N+1})_{N+1} I_s \end{pmatrix}, \ \mathbf{M} = \begin{pmatrix} (Fe_1)_1\Lambda \\ \vdots \\ (Fe_1)_{N+1}\Lambda \end{pmatrix},$$

and a direct computation shows that

$$\mathbf{N}^T\mathbf{G}^{-1}\mathbf{M} = \sum_{j=1}^{N+1}(I_s - \pi_j\Lambda)^{-1}\Lambda(Fe_1)_j(F^{-T}e_{N+1})_j = \text{diag}(P(\Lambda Fe_1 \circ F^{-T}e_{N+1})).$$

The formulation above provides a cheap expression for the construction of $\mathbf{N}^T\mathbf{G}^{-1}\mathbf{M}$ along with illustrating its diagonal structure, hence solving the linear system with $U = I + \text{diag}(P(\Lambda Fe_1 \circ F^{-T}e_{N+1}))$ does not significantly increase the cost of computing Z.

Returning to the computation of $\widetilde{Z}$, we have

$$\begin{aligned} \mathbf{G}^{-1}\mathbf{M}(I + \mathbf{N}^T\mathbf{G}^{-1}\mathbf{M})^{-1}\mathbf{N}^T\mathbf{G}^{-1}\text{vec}(T^{-1}VF^T) &= \mathbf{G}^{-1}\mathbf{M}U^{-1}YF^{-T}e_{N+1} \\ &= P \circ (\Lambda U^{-1}YF^{-T}e_{N+1}e_1^T F^T) \\ &= W. \end{aligned}$$

Combining the steps above yields the statement in (14). □

Appendix B

For the sake of completeness, in Algorithm 4 we report the pseudocode of the matrix-oriented GMRES method applied to (3). The m-th basis vector of the Krylov subspace $K_m(\mathcal{A}, \mathbf{b})$ is represented in terms of the matrices $\mathcal{V}_{1,m} \in \mathbb{R}^{s \times (N+1)}$, $\mathcal{V}_{2,m} \in \mathbb{R}^{p \times (N+1)}$, and $\mathcal{V}_{3,m} \in \mathbb{R}^{s \times (N+1)}$, namely

$$v_m = \text{vec}\begin{bmatrix} \mathcal{V}_{1,m} \\ \mathcal{V}_{2,m} \\ \mathcal{V}_{3,m} \end{bmatrix}.$$

Algorithm 4 Matrix-oriented GMRES for (3).

input : $B, Q_1, \ldots, Q_N \in \mathbb{R}^{s \times s}$, $R_0, \ldots, R_N \in \mathbb{R}^{p \times p}$, $H_0, \ldots, H_N \in \mathbb{R}^{p \times s}$, $M_1, \ldots, M_N \in \mathbb{R}^{s \times s}$ $[b_0, c_1, \ldots, c_N] \in \mathbb{R}^{s \times (N+1)}$, $[d_0, \ldots, d_N] \in \mathbb{R}^{p \times (N+1)}$, $m_{\max}$, $\varepsilon > 0$

output: $\delta \Theta_m, \delta X_m \in \mathbb{R}^{s \times (N+1)}$, $\delta \Lambda_m^T \in \mathbb{R}^{p \times (N+1)}$ such that $\mathrm{vec}(\delta \Theta_m^T, \delta \Lambda_m^T, \delta X_m^T)^T$ is an approximate solution to (3).

1 Compute $\beta = \sqrt{\|[b_0, c_1, \ldots, c_N]\|_F^2 + \|[d_0, \ldots, d_N]\|_F^2}$ and set $\mathcal{V}_{1,1} = [b_0, c_1, \ldots, c_N]/\beta$, $\mathcal{V}_{2,1} = [d_0, \ldots, d_N]/\beta$, and $\mathcal{V}_{3,1} = 0$

2 **for** $m = 1, 2, \ldots,$ *till* $m_{\max}$ **do**

3   Set

$$\widehat{\mathcal{V}}_{1,m+1} = [B\mathcal{V}_{1,m}e_1, Q_1\mathcal{V}_{1,m}e_2, \ldots, Q_N\mathcal{V}_{1,m}e_{N+1}] + [\mathcal{V}_{3,m}e_1, \mathcal{V}_{3,m}e_2 - M_1\mathcal{V}_{3,m}e_1, \ldots, \mathcal{V}_{3,m}e_{N+1} - M_N\mathcal{V}_{3,m}e_N]$$

$$\widehat{\mathcal{V}}_{2,m+1} = [R_0\mathcal{V}_{2,m}e_1, \ldots, R_N\mathcal{V}_{2,m}e_{N+1}] + [H_0\mathcal{V}_{3,m}e_1, \ldots, H_N\mathcal{V}_{3,m}e_{N+1}]$$

$$\widehat{\mathcal{V}}_{3,m+1} = [\mathcal{V}_{1,m}e_1 - M_1^T\mathcal{V}_{1,m}e_2, \ldots, \mathcal{V}_{1,m}e_N - M_N^T\mathcal{V}_{1,m}e_{N+1}, \mathcal{V}_{1,m}e_N] + [H_0^T\mathcal{V}_{2,m}e_1, \ldots, H_N^T\mathcal{V}_{2,m}e_{N+1}]$$

4   Set $(\underline{T}_m)_{j,m} = 0$ for $j = 1, \ldots, m+1$

5   **for** $\ell = 1, 2$ **do**

6   Compute

$$(\underline{T}_m)_{j,m} = (\underline{T}_m)_{j,m} + \sqrt{\mathrm{trace}(\widehat{\mathcal{V}}_{1,m+1}^T \mathcal{V}_{1,j})^2 + \mathrm{trace}(\widehat{\mathcal{V}}_{2,m+1}^T \mathcal{V}_{2,j})^2 + \mathrm{trace}(\widehat{\mathcal{V}}_{3,m+1}^T \mathcal{V}_{3,j})^2}, \quad j = 1, \ldots, m$$

7   Set $\widehat{\mathcal{V}}_{i,m+1} = \widehat{\mathcal{V}}_{i,m+1} - \sum_{j=1}^m (\underline{T}_m)_{j,m}\mathcal{V}_{i,j}$, for $i = 1, 2, 3$

8   Compute $(\underline{T}_m)_{m+1,m} = \sqrt{\|\widehat{\mathcal{V}}_{1,m+1}\|_F^2 + \|\widehat{\mathcal{V}}_{2,m+1}\|_F^2 + \|\widehat{\mathcal{V}}_{3,m+1}\|_F^2}$

9   Set $\mathcal{V}_{i,m+1} = \widehat{\mathcal{V}}_{i,m+1}/(\underline{T}_m)_{m+1,m}$, $i = 1, 2, 3$

10   Solve $y_m = \arg\min_{y \in \mathbb{R}^m} \|\underline{T}_m y - \beta e_1\|$

11   Compute the residual norm $\|r_m\|$

12   **if** $\|r_m\| \le \varepsilon\beta$ **then**

13   Go to 14

14 Set $\delta\Theta_m = \sum_{j=1}^m \mathcal{V}_{1,j}(e_i^T y_m)$, $\delta\Lambda_m = \sum_{j=1}^m \mathcal{V}_{2,j}(e_i^T y_m)$, and $\delta X_m = \sum_{j=1}^m \mathcal{V}_{3,j}(e_i^T y_m)$

Algorithm 5 Matrix-oriented CG for (2).

input : $B, Q_1, \ldots, Q_N \in \mathbb{R}^{s \times s}$, $R_0, \ldots, R_N \in \mathbb{R}^{p \times p}$, $H_0, \ldots, H_N \in \mathbb{R}^{p \times s}$, $M_1, \ldots, M_N \in \mathbb{R}^{s \times s}$ $[b_0, c_1, \ldots, c_N] \in \mathbb{R}^{s \times (N+1)}$, $[d_0, \ldots, d_N] \in \mathbb{R}^{p \times (N+1)}$, $m_{\max}$, $\varepsilon > 0$

output: $\delta X_m \in \mathbb{R}^{s \times (N+1)}$ approximate solution to (2)

1 Set $\mathcal{R}_0 = W_0 = [Bb_0, Q_1c_1, \ldots, Q_Nc_N] + [H_0^T R_0^{-1}d_0 - M_1^T H_1^T R_1^{-1}d_1, \ldots, H_{N-1}^T R_{N-1}^{-1}d_{N-1} - M_N^T H_N^T R_N^{-1}d_N, H_N^T R_N^{-1}d_N]$, $\delta X_0 = 0$, and compute $\rho_0 = \|\mathcal{R}_0\|_F^2$

2 **for** $m = 1, 2, \ldots,$ *till* $m_{\max}$ **do**

3   Set

4
$$\begin{aligned}
\mathcal{W}_m = \ & [B^{-1}W_{m-1}e_1 - M_1^T Q_1^{-1}(W_{m-1}e_2 - M_1 W_{m-1}e_1), \\
& Q_1^{-1}(W_{m-1}e_2 - M_1 W_{m-1}e_1) - M_2^T Q_2^{-1}(W_{m-1}e_3 - M_2 W_{m-1}e_2), \ldots, Q_N^{-1}(W_{m-1}e_{N+1} - M_N W_{m-1}e_N)] \\
& + [H_0^T R_0^{-1} H_0 W_{m-1}e_1, \ldots, H_N^T R_N^{-1} H_N W_{m-1}e_{N+1}]
\end{aligned}$$

5      $\alpha_m = \rho_{m-1}/\mathrm{trace}(\mathcal{W}_m^T W_{m-1})$

6      $\delta X_m = \delta X_{m-1} + \alpha_m W_{m-1}$

7      $\mathcal{R}_m = \mathcal{R}_{m-1} - \alpha_k \mathcal{W}_m$

8      $\rho_m = \|\mathcal{R}_m\|_F^2$

9   **if** $\sqrt{\rho_m} \le \varepsilon\rho_0$ **then**

10   Return δX_m

11     $\beta_m = \rho_m/\rho_{m-1}$

12     $W_m = \mathcal{R}_m + \beta_m W_{m-1}$

The residual norm in line 11 can be cheaply computed by following, e.g., the classic Givens rotations approach presented in [39, Section 6.5.3].

Similarly, in Algorithm 5 we report the pseudocode of the matrix-oriented CG method applied to (2).

In what follows, $(A)_{i,j}$ will denote the (i, j)-th entry of the matrix A.

References

[1] A. Barraud, A numerical algorithm to solve $A^T X A - X = Q$, IEEE Trans. Autom. Control 22 (1977) 883–885.

[2] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.

[3] M. Benzi, A.J. Wathen, Some preconditioning techniques for saddle point problems, in: Model Order Reduction: Theory, Research Aspects and Applications, in: Math. Ind., vol. 13, Springer, Berlin, 2008, pp. 195–211.

[4] D.S. Bernstein, Matrix Mathematics, Princeton University Press, 2009.

[5] D.A. Bini, B. Iannazzo, Computing the Karcher mean of symmetric positive definite matrices, Linear Algebra Appl. 438 (2013) 1700–1710.

[6] N. Boumal, B. Mishra, P.-A. Absil, R. Sepulchre, Manopt, a Matlab toolbox for optimization on manifolds, J. Mach. Learn. Res. 15 (2014) 1455–1459.

[7] A. Carrassi, M. Bocquet, L. Bertino, G. Evensen, Data assimilation in the geosciences: an overview of methods, issues, and perspectives, Wiley Interdiscip. Rev.: Clim. Change 9 (2018) e535.

[8] P. Courtier, J.-N. Thépaut, A. Hollingsworth, A strategy for operational implementation of 4D-var, using an incremental approach, Q. J. R. Meteorol. Soc. 120 (1994) 1367–1387.

[9] I. Daužickaitè, A.S. Lawless, J.A. Scott, P.J. van Leeuwen, Randomised preconditioning for the forcing formulation of weak constraint 4d-var, preprint, arXiv:2101.07249, 2021.

[10] A. El-Said, Conditioning of the weak-constraint variational data assimilation problem for numerical weather prediction, PhD thesis, University of Reading, 2015.

[11] M. Fisher, S. Gratton, S. Gürol, Y. Trémolet, X. Vasseur, Low rank updates in preconditioning the saddle point systems arising from data assimilation problems, Optim. Methods Softw. 33 (2018) 45–69.

[12] M. Fisher, S. Gürol, Parallelization in the time dimension of four-dimensional variational data assimilation, Q. J. R. Meteorol. Soc. 143 (2017) 1136–1147.

[13] M. Fisher, Y. Tremolet, H. Auvinen, D. Tan, P. Poli, Weak-Constraint and Long-Window 4D-Var, ECMWF, Reading, UK, 2012.

[14] M.A. Freitag, D.L.H. Green, A low-rank approach to the solution of weak constraint variational data assimilation problems, J. Comput. Phys. 357 (2018) 263–281.

[15] I. Gejadze, H. Oubanas, V. Shutyaev, Implicit treatment of model error using inflated observation-error covariance, Q. J. R. Meteorol. Soc. 143 (2017) 2496–2508.

[16] G.H. Golub, C.F. Van Loan, Matrix Computations, fourth ed., Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 2013.

[17] S. Gratton, S. Gürol, E. Simon, P.L. Toint, Guaranteeing the convergence of the saddle formulation for weakly constrained 4D-var data assimilation, Q. J. R. Meteorol. Soc. 144 (2018) 2592–2602.

[18] D. Green, Model order reduction for large-scale data assimilation problems, PhD thesis, University of Bath, 2019.

[19] A. Gupta, V. Kumar, Scalability of Parallel Algorithms for Matrix Multiplication, 1993 International Conference on Parallel Processing - ICPP'93, vol. 3, IEEE, 1993, pp. 115–123.

[20] Y. Hao, V. Simoncini, The Sherman–Morrison–Woodbury formula for generalized linear matrix equations and applications, Numer. Linear Algebra Appl. 28 (2021) e2384.

[21] M.R. Hestenes, E. Stiefel, Methods of conjugate gradients for solving linear systems, J. Res. Natl. Bur. Stand. 49 (1952) 409–436.

[22] K. Howes, A.M. Fowler, A. Lawless, Accounting for model error in strong-constraint 4d-var data assimilation, Q. J. R. Meteorol. Soc. 143 (2017) 1227–1240.

[23] J. Huang, J.L. Gómez-Dans, H. Huang, H. Ma, Q. Wu, P.E. Lewis, S. Liang, Z. Chen, J.-H. Xue, Y. Wu, F. Zhao, J. Wang, X. Xie, Assimilation of remote sensing into crop growth models: current status and perspectives, Agric. For. Meteorol. 276-277 (2019) 107609.

[24] B. Iannazzo, M. Porcelli, The Riemannian Barzilai–Borwein method with nonmonotone line search and the matrix geometric mean computation, IMA J. Numer. Anal. 38 (2017) 495–517.

[25] K. Jbilou, A. Messaoudi, A computational method for symmetric Stein matrix equations, in: Numerical Linear Algebra in Signals, Systems and Control, in: Lect. Notes Electr. Eng., vol. 80, Springer, Dordrecht, 2011, pp. 295–311.

[26] M. Kärcher, S. Boyaval, M.A. Grepl, K. Veroy, Reduced basis approximation and a posteriori error bounds for 4d-var data assimilation, Optim. Eng. 19 (2018) 663–695.

[27] D. Kressner, M. Plešinger, C. Tobler, A preconditioned low-rank CG method for parameter-dependent Lyapunov matrix equations, Numer. Linear Algebra Appl. 21 (2014) 666–684.

[28] E.N. Lorenz, Predictability: a problem partly solved, in: Seminar on Predictability, 4–8 September 1995, in: Reading, vol. 1, ECMWF, 1995, pp. 1–18.

[29] M.F. Murphy, G.H. Golub, A.J. Wathen, A note on preconditioning for indefinite linear systems, SIAM J. Sci. Comput. 21 (2000) 1969–1972.

[30] G. Nakamura, R. Potthast, Inverse Modeling, IOP Publishing, 2015.

[31] Y. Notay, Flexible conjugate gradients, SIAM J. Sci. Comput. 22 (2000) 1444–1460.

[32] C.C. Paige, M.A. Saunders, Solutions of sparse indefinite systems of linear equations, SIAM J. Numer. Anal. 12 (1975) 617–629.

[33] D. Palitta, Matrix equation techniques for certain evolutionary partial differential equations, J. Sci. Comput. 87 (2021).

[34] D. Palitta, P. Kürschner, On the convergence of Krylov methods with low-rank truncations, Numer. Algorithms 88 (2021) 1383–1417.

[35] E. Pinnington, T. Quaife, E. Black, Impact of remotely sensed soil moisture and precipitation on soil moisture prediction in a data assimilation system with the jules land surface model, Hydrol. Earth Syst. Sci. 22 (2018) 2575–2588.

[36] X. Quan, B. He, X. Li, Z. Tang, Estimation of grassland live fuel moisture content from ratio of canopy water content and foliage dry biomass, IEEE Geosci. Remote Sens. Lett. 12 (2015) 1903–1907.

[37] T. Rees, Preconditioning iterative methods for PDE constrained optimization, PhD thesis, Citeseer, 2010.

[38] Y. Saad, A flexible inner-outer preconditioned GMRES algorithm, SIAM J. Sci. Comput. 14 (1993) 461–469.

[39] Y. Saad, Iterative Methods for Sparse Linear Systems, second ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 2003.

[40] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput. 7 (1986) 856–869.

[41] S.J. Schiff, Neural Control Engineering: the Emerging Intersection Between Control Theory and Neuroscience, MIT Press, 2011.

[42] V. Simoncini, Computational methods for linear matrix equations, SIAM Rev. 58 (2016) 377–441.

[43] V. Simoncini, D.B. Szyld, Flexible inner-outer Krylov subspace methods, SIAM J. Numer. Anal. 40 (2002) 2219–2239.

[44] M. Stoll, T. Breiten, A low-rank in time approach to PDE-constrained optimization, SIAM J. Sci. Comput. 37 (2015) B1–B29.

[45] J.M. Tabeart, S.L. Dance, S.A. Haben, A.S. Lawless, N.K. Nichols, J.A. Waller, The conditioning of least-squares problems in variational data assimilation, Numer. Linear Algebra Appl. 25 (2018) e2165.

[46] J.M. Tabeart, S.L. Dance, A.S. Lawless, N.K. Nichols, J.A. Waller, New bounds on the condition number of the Hessian of the preconditioned variational data assimilation problem, Numer. Linear Algebra Appl. 29 (2022) e2405.

[47] J.M. Tabeart, J.W. Pearson, Saddle point preconditioners for weak-constraint 4d-var, preprint, arXiv:2105.06975, 2022.

[48] J.M. Tabeart, J.W. Pearson, Using low-rank observation information to precondition weak-constraint 4D-var, 2023, in preparation.

[49] Y. Trémolet, Accounting for an imperfect model in 4d-var, Q. J. R. Meteorol. Soc. 132 (2006) 2483–2504.

[50] J. Tshimanga, S. Gratton, A.T. Weaver, A. Sartenaer, Limited-memory preconditioners, with application to incremental four-dimensional variational data assimilation, Q. J. R. Meteorol. Soc. 134 (2008) 751–769.

[51] C.F. Van Loan, The ubiquitous Kronecker product, J. Comput. Appl. Math. 123 (2000) 85–100, Numerical Analysis 2000. Vol. III: Linear Algebra.