A Preconditioned Multiple Shooting Shadowing Algorithm for the Sensitivity Analysis of Chaotic Systems

Karim Shawki* and George Papadakis†

Department of Aeronautics, Imperial College London, Exhibition Road, London, SW7 2AZ, UK

July 28, 2019

Abstract

We propose a preconditioner that can accelerate the rate of convergence of the Multiple Shooting Shadowing (MSS) method [1]. This recently proposed method can be used to compute derivatives of time-averaged objectives (also known as sensitivities) to system parameter(s) for chaotic systems. We propose a block diagonal preconditioner, which is based on a partial singular value decomposition of the MSS constraint matrix. The preconditioner can be computed using matrix-vector products only (i.e. it is matrix-free) and is fully parallelised in the time domain. Two chaotic systems are considered, the Lorenz system and the 1D Kuramoto Sivashinsky equation. Combination of the preconditioner with a regularisation method leads to tight bracketing of the eigenvalues to a narrow range. This combination results in a significant reduction in the number of iterations, and renders the convergence rate almost independent of the number of degrees of freedom of the system, and the length of the trajectory that is used to compute the time-averaged objective. This can potentially allow the method to be used for large chaotic systems (such as turbulent flows) and optimal control applications. The singular value decomposition of the constraint matrix can also be used to quantify the effect of the system condition on the accuracy of the sensitivities. In fact, neglecting the singular modes affected by noise, we recover the correct values of sensitivity that match closely with those obtained with finite differences for the Kuramoto Sivashinsky equation in the light turbulent regime. We notice a similar improvement for the Lorenz system as well.

Keywords:

Sensitivity analysis, Chaotic systems, Least Squares Shadowing, Preconditioning

*Email: karim.shawki14@imperial.ac.uk

†Email: g.papadakis@imperial.ac.uk

Declarations of interest: none

1 Introduction

Optimisation has a broad range of applications. In aerospace engineering for example, optimisation can be used in airfoil design [2, 3], flow control [4, 5, 6] and uncertainty quantification [7]. In practice, we are usually interested in optimising a time-averaged quantity (objective) $\bar{J}$ subject to a set of constraints. An example is the minimisation of an airfoil drag by some active control technique, subject to the governing equations, boundary and initial conditions. The sensitivity $d\bar{J}/ds$, where s is one or more control parameters, can be employed together with a gradient-based algorithm to find the optimal values of s that minimize the objective $\bar{J}$.

Traditional sensitivity analysis methods involve solving the tangent equation (or the adjoint equations for multiple control parameters s) to compute $d\bar{J}/ds$. The above methods are based on linearisation around a reference trajectory in phase space. If the dynamical system is chaotic, $d\bar{J}/ds$ diverges exponentially for long time horizons T. Here 'long time horizons' refers to time intervals much longer than the characteristic time scales of the system dynamics. Lea et al. [8] showed that the adjoint of the Lorenz system yielded $|d\bar{J}/ds| \propto e^{\lambda_{max}t}$, where λ_{max} is the largest Lyapunov exponent, that describes the dominant rate of separation of two trajectories evaluated with parameters s and $s + \delta s$. Transitional and turbulent flows are chaotic; therefore, traditional methods fail to compute the correct derivatives for long T. A recent example is the non-linear optimal control of bypass transition in a boundary layer using blowing and suction, where T had to be restricted to avoid the exponential growth of the adjoint variables during backward integration [4, 9].

Many attempts have been made to develop methods that can compute accurate sensitivities $d\bar{J}/ds$ for long T. In [8], the 'ensemble-adjoint' method was proposed, where a long T is split into many integration segments. For a given s, the adjoint method is applied to all segments, and an average is taken. The accuracy of $d\bar{J}/ds$ depends on the length of segments, and knowledge of the suitable lengths is not known a priori.

Thuburn [10] employed the Fokker-Planck equation, which governs the evolution of the probability density function in phase space when stochastic Wiener forcing terms are added to the discrete model equations. These terms are artificial but they introduce diffusion to the evolution equation, thus ensuring a smooth steady solution. He expressed the long-time sensitivity in terms of the adjoint of the Fokker-Plank equation. The issue with this method however, is its computational cost for large systems (a multi-dimensional diffusion operator must be inverted) and solution accuracy (due to the addition of the stochastic terms).

The Fluctuation-Dissipation Theorem, a powerful tool of statistical physics [11], can also be used to estimate the time-average response to forcing in nonlinear systems. The key underlying assumption is that the forcing is weak enough such that the response of the system changes linearly with the forcing. A 1D reduced order model (based on Dynamic Mode Decomposition) was recently proposed to estimate the response to external forcing of the horizontally averaged temperature in a fully turbulent, buoyancy driven flow [12]. The model was also used to derive the forcing that elicits a desired response, and this makes it suitable for control applications.

Lasagna [13] formulated a well-behaved sensitivity analysis method for unstable periodic orbits. Due to periodicity, the adjoints are bounded in time irrespective of the orbit length, and the sensitivities computed for every orbit are exact. However, finding periodic orbits can be very challenging for 3D turbulent flows.

The Least Squares Shadowing (LSS) method [14] considers a reference trajectory in phase space, evaluated at s, and finds a nearby (or 'shadowing trajectory'), evaluated at $s + \delta s$, that stays close to the reference trajectory. The proximity of the two trajectories regularises the problem and can be used to find accurate ${}^{d\bar{J}}/_{ds}$ for long T. The method relies on the shadowing lemma, that guarantees that such a nearby trajectory exists for structurally stable, uniformly hyperbolic systems [15, 16, 17]. The two trajectories start from different initial conditions, but for ergodic systems, the time-average and its sensitivity do not depend on the initial conditions. LSS results in a linear two-point boundary value problem in time, which is time-consuming to solve and has large storage requirements.

Blonigan and Wang [1] proposed the Multiple Shooting Shadowing (MSS) algorithm that results in a much smaller linear system than that of LSS. This is because the norm of the distance between the reference and shadowing trajectories is defined using the values at discrete checkpoints, not at every time step. The system can be solved iteratively using Krylov subspace methods (such as Conjugate Gradient) that only require matrix-vector products. Although computationally more efficient than standard LSS, in practice the convergence rate can be slow due to the high condition number of the system matrix.

Non-Intrusive Least Squares Shadowing (NILSS) [18] was proposed to further reduce the computational cost, and it is closely related to MSS [1]. NILSS requires the integration of one inhomogeneous tangent that represents the effect of changing s, and a set of homogeneous tangents that represent the effect of changing initial conditions. The reference trajectory is segmented and a minimisation problem is solved for the set of weights of the homogeneous tangent solutions that cancel out the growth of the inhomogeneous solution at every segment, leading to an overall bounded result. The cost of solving the NILSS matrix is negligible compared to the cost of solving the Transcription LSS matrix. The main cost is in setting up the matrix, which requires many integrations of the tangent equation, at least equal to the number of positive Lyapunov exponents. For high Re flows the cost of tangent equation integrations would be high due to the expected large number of positive Lyapunov exponents. NILSS has been applied successfully to a chaotic flow over a backwards facing step [18]. A discrete adjoint version was derived in [19] and applied for the first time to a 3D DNS simulation of turbulent channel flow at $Re_\tau = 140$.

MSS remains an attractive algorithm because it does not require the prior computation of all the positive Lyapunov exponents. However, it suffers from slow convergence, as already mentioned. In this paper, we propose a preconditioner that can significantly accelerate the convergence rate. We show that the operations to construct the preconditioner are fully parallel-in-time, require only matrix-vector products, and lead to significant overall cost savings (compared to standard

MSS). Using this preconditioner and regularising the small eigenvalues of the system matrix, leads to accurate results and convergence rates almost independent of T, and most importantly, of the number of degrees of freedom, N.

This paper is structured as follows: Section (2) presents a general overview of LSS and MSS. In Section (3), the new preconditioner for MSS is derived and is applied to the Lorenz system in Section (4), and the Kuramoto Sivashinsky equation in Section (5). The effect of ill-conditioning on the solution accuracy is explored in Section (6) and the regularisation of the preconditioned system is investigated in Section (7). The computational cost of the proposed method is discussed in Section (8) and we conclude in section (9).

2 Shadowing Methods

Consider a dynamical system described by the set of ordinary differential equations

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(\mathbf{u}, s), \quad u(0, s) = u_0(s) \tag{1}$$

where $\mathbf{u}(t, s)$ is the vector of state variables of length N, $\mathbf{f}$ is the vector of non-linear equations, and s is one (or more) system or control parameters. We define a time-averaged objective as

$$\bar{J} = \frac{1}{T} \int_0^T J(\mathbf{u}(t, s), s)\, dt \tag{2}$$

and we are interested in computing the sensitivity ${}^{d\bar{J}}/_{ds}$ over a long time horizon, $T \to \infty$. Using the chain rule, we can write

$$\frac{d\bar{J}}{ds} = \frac{1}{T} \int_0^T \frac{\partial J(\mathbf{u}(t, s), s)}{\partial u} \mathbf{v}(t, s) + \frac{\partial J}{\partial s}\, dt \tag{3}$$

where $\mathbf{v}(t, s) = {}^{du}/_{ds}$. Perturbing (1) by δs and linearizing, the tangent evolution equation for $\mathbf{v}(t, s)$ can be obtained:

$$\frac{d\mathbf{v}}{dt} = \frac{\partial f}{\partial u}\mathbf{v} + \frac{\partial \mathbf{f}}{\partial s} \tag{4}$$

with the initial condition $\mathbf{v}(0) = {}^{du_0}/_{ds}$. Equation (4) is solved for $\mathbf{v}(t)$ and the sensitivity is found using (3). An adjoint version of (4) can be derived to compute the derivative of $\bar{J}$ to many parameters simultaneously.

Chaotic systems of equations are sensitive to small changes in initial conditions and parameter values. For such systems, the solution $\mathbf{v}(t)$ grows exponentially at a rate dictated by λ_{max}, the maximum positive Lyapunov exponent. Since all chaotic systems have at least one positive λ, the sensitivity computed through (3) becomes meaningless for long T. Shadowing methods ensure that trajectories evaluated at s and $s + \delta s$ do not diverge, and therefore the solution to (4) remains bounded in time.

2.1 Least Squares Shadowing (LSS)

The Least Squares Shadowing method (LSS) was developed by Wang et al. [14] to compute the sensitivity ${}^{d\bar{J}}\!/_{ds}$ over long T for chaotic systems. LSS finds a 'shadow' trajectory that satisfies (1) at a slightly perturbed parameter value, i.e. ${}^{d\mathbf{u}'}\!/_{dt} = \mathbf{f}(\mathbf{u}', s + \delta s)$, that stays close in phase space to a reference trajectory $\mathbf{u}_{ref}$ evaluated at s, for all T. Since the solution $\mathbf{u}'(\tau(t))$ does not diverge from $\mathbf{u}_{ref}(t)$, it can be used to evaluate meaningful sensitivities ${}^{d\bar{J}}\!/_{ds}$. A time transformation $\tau(t)$ is required to avoid algebraically growing components arising from the projection of $\mathbf{v}(t)$ on the direction of $\mathbf{f}(\mathbf{u}(t))$ [20]. To obtain the shadowing trajectory, a minimisation problem is formulated and solved. Linearisation results in

$$\begin{aligned} &\underset{\mathbf{v},\eta}{\text{Minimise}} \quad \frac{1}{T}\int_0^T \|\mathbf{v}\|^2 + \alpha^2\eta^2\, dt, \\ &\text{subject to} \quad \frac{d\mathbf{v}}{dt} = \frac{\partial f}{\partial u}\mathbf{v} + \frac{\partial \mathbf{f}}{\partial s} + \eta\mathbf{f}(u_{ref}, s) \end{aligned} \tag{5}$$

where $\eta(t) = {}^{d\,({}^{d\tau}\!/_{dt}\,-\,1)}\!/_{ds}$ is a time dilatation term, and $\mathbf{v}(t) = {}^{d(\mathbf{u}'\,-\,\mathbf{u}_{ref})}\!/_{ds}$. The solutions $\mathbf{v}(t)$ and $\eta(t)$ are then used to compute the sensitivity by evaluating (see [14] for the derivation):

$$\frac{d\bar{J}}{ds} = \frac{1}{T}\int_0^T \left(\frac{\partial J}{\partial u}\mathbf{v} + \frac{\partial J}{\partial s} + \eta(J - \bar{J})\right) dt \tag{6}$$

The solution of (5) results in a two-point boundary value problem in time. Discretisation in space and time yields a linear system with $M \times N$ unknowns, where M is the number of time steps, and N is the number of degrees of freedom. The large size of the matrix therefore makes the method applicable to low-dimensional systems only.

2.2 Multiple Shooting Shadowing (MSS)

In attempt to reduce the cost of LSS for high dimensional systems, Blonigan and Wang [1] proposed to minimise $\mathbf{v}(t)$ at K discreet checkpoints as shown in Figure (1). This idea reduces the number of unknowns to $K \times N \ll M \times N$. The Multiple Shooting Shadowing (MSS) method is formulated as:

$$\underset{\mathbf{v}(t_i^+)}{\text{Minimise}} \quad \sum_{i=0}^{K} \|\mathbf{v}(t_i^+)\|_2^2 \tag{7a}$$

$$\text{subject to} \quad \mathbf{v}(t_i^+) = \mathbf{v}(t_i^-) \qquad (i = 1, 2, ..., K-1) \tag{7b}$$

$$\frac{d\mathbf{v}}{dt} - \frac{\partial f}{\partial u}\mathbf{v} - \frac{\partial \mathbf{f}}{\partial s} - \eta\mathbf{f} = 0 \qquad t_i < t < t_{i+1} \;\; (i = 0, 1, ..., K-1) \tag{7c}$$

$$\langle \mathbf{f}(\mathbf{u}(t), s), \mathbf{v}(t)\rangle = 0 \qquad t_i < t < t_{i+1} \;\; (i = 0, 1, ..., K-1) \tag{7d}$$

The constraint (7b) enforces the continuity of $\mathbf{v}(t)$ between the inner checkpoints $(t_1, t_2, ..., t_{K-1})$. The inner product (7d) ensures that $\mathbf{v}(t)$ remains normal to $\mathbf{f}(\mathbf{u}(t), s)$. This constraint arises as a direct result of solving the linear least squares minimisation problem; this is shown in Appendix A of [21].

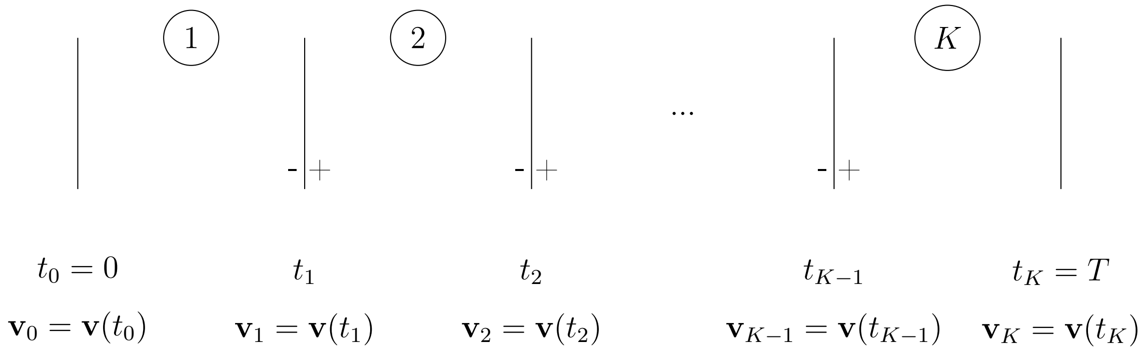

1
2
...
K
-|+
-|+
-|+
$t_0 = 0$
t_1
t_2
t_{K-1}
$t_K = T$
$\mathbf{v}_0 = \mathbf{v}(t_0)$
$\mathbf{v}_1 = \mathbf{v}(t_1)$
$\mathbf{v}_2 = \mathbf{v}(t_2)$
$\mathbf{v}_{K-1} = \mathbf{v}(t_{K-1})$
$\mathbf{v}_K = \mathbf{v}(t_K)$

Figure 1: A sketch illustrating the segmenting approach of MSS. The constraint equations are propagated forward in time in all K segments, such that $\mathbf{v}(t_i^+) = \mathbf{v}(t_i^-)$ is satisfied for $i = 1, 2, ..., K-1$.

For completeness, we provide below a short description of the solution of system (7); for all derivation details, the reader is referred to [1]. The solution $\mathbf{v}(t)$ that satisfies (7c,d) in all segments can be expressed as

$$\mathbf{v}(t) = P_t \left(\phi^{t_i,t}\mathbf{v}(t_i) + \int_{t_i}^{t} \phi^{\tau,t} \frac{\partial \mathbf{f}}{\partial s}\, d\tau \right), \qquad t_i \le t < t_{i+1} \tag{8}$$

where $\phi^{\tau,t}$ is the state transition matrix that satisfies

$$\frac{d\phi^{t,\tau}}{d\tau} = \left(\left. \frac{\partial f}{\partial u} \right|_{\tau} \right) \phi^{t,\tau} \tag{9}$$

and P_t is the projection operator (in discrete form it's a matrix)

$$P_t = I - \frac{\mathbf{f}(t)\mathbf{f}(t)^T}{\mathbf{f}(t)^T\mathbf{f}(t)} \tag{10}$$

that eliminates η from (7c) and enforces (7d). This allows us to integrate (7c) without the time dilation term $\eta\mathbf{f}$, i.e.

$$\frac{d\mathbf{v}'}{dt} - \frac{\partial f}{\partial u}\mathbf{v}' - \frac{\partial \mathbf{f}}{\partial s} = 0 \tag{11}$$

and recover $\mathbf{v}(t)$ from $\mathbf{v}(t) = P_t\mathbf{v}'(t)$.

The minimisation problem (7) can be written as a least squares problem:

$$\underset{\mathbf{v}_i}{\text{Minimise}} \;\; \frac{1}{2}\sum_{i=0}^{K} \|\mathbf{v}_i\|_2^2 \tag{12a}$$

$$\text{subject to} \;\; \mathbf{v}_{i+1} = \Phi_{i+1}\mathbf{v}_i + \mathbf{b}_{i+1} \tag{12b}$$

where $\mathbf{v}_i = \mathbf{v}(t_i^+)$, $\Phi_{i+1} = P_{t_{i+1}}\phi^{t_i,t_{i+1}}$ and $\mathbf{b}_{i+1} = P_{t_{i+1}} \int_{t_i}^{t_{i+1}} (\phi^{\tau,t}) {}^{\partial \mathbf{f}}\!/\!{}_{\partial s}\, d\tau$. Equation (12b) satisfies the original constraints (7b,c,d). System (12) can be written in matrix form as

$$\underset{\mathbf{v}_i}{\text{Minimise}} \;\; \frac{1}{2}\sum_{i=0}^{K} \|\mathbf{v}_i\|_2^2 \tag{13a}$$

$$\text{subject to} \;\; A\underline{\mathbf{v}} = \underline{\mathbf{b}} \tag{13b}$$

where

$$A = \begin{bmatrix} -\Phi_1 & I & & & \\ & -\Phi_2 & I & & \\ & & \ddots & \ddots & \\ & & & -\Phi_K & I \end{bmatrix} \quad \underline{\mathbf{v}} = \begin{bmatrix} \mathbf{v}_0 \\ \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_K \end{bmatrix} \quad \underline{\mathbf{b}} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_K \end{bmatrix} \tag{14}$$

A is a $NK \times N(K+1)$ matrix and $\underline{\mathbf{v}}$ and $\underline{\mathbf{b}}$ are vectors of length $N(K+1)$ and NK, respectively. Equation (13b) represents an under-determined system; according to (13a) we seek the solution with the minimum Euclidean norm. This a well known minimisation problem in linear algebra [22]. A set of discrete adjoint variables $\underline{\mathbf{w}} = \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_K \end{bmatrix}^T$ is introduced, and an optimality system is derived:

$$\begin{bmatrix} -I & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \underline{\mathbf{v}} \\ \underline{\mathbf{w}} \end{bmatrix} = \begin{bmatrix} 0 \\ \underline{\mathbf{b}} \end{bmatrix} \tag{15}$$

The matrix in the above equation is symmetric, but indefinite. Due to the presence of both positive and negative real eigenvalues, it is more efficient to formulate and solve a linear system with the Schur complement of (15),

$$S\underline{\mathbf{w}} = \begin{bmatrix} \Phi_1\Phi_1^T + I & -\Phi_2^T & & \\ -\Phi_2 & \Phi_2\Phi_2^T + I & -\Phi_3^T & \\ & \ddots & \ddots & \ddots \\ & & -\Phi_K & \Phi_K\Phi_K^T + I \end{bmatrix} \begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \\ \vdots \\ \mathbf{w}_K \end{bmatrix} = \underline{\mathbf{b}}, \tag{16}$$

where $S = AA^T$. The Schur complement matrix S is block tri-diagonal, symmetric and positive definite, with size $NK \times NK$. All eigenvalues of S are real and positive. The solution $\underline{\mathbf{w}}$ is substituted into (15) to obtain $\underline{\mathbf{v}}$.

Equation (16) can be solved iteratively by supplying matrix-vector products $S\underline{\mathbf{w}}^{(m)}$ at each iteration m to a Krylov subspace solver, such as Conjugate Gradient. All products can be computed by calling a time-stepper for the forward (or adjoint) equations with the appropriate initial (or terminal) conditions. For example, the product Φ_i with an arbitrary vector $\mathbf{z}_{i-1}$ requires forward integration of the homogeneous form of (11), i.e.

$$\frac{d\mathbf{v}'}{dt} - \frac{\partial f}{\partial u}\mathbf{v}' = 0 \tag{17}$$

with the initial condition $\mathbf{v}'(t_{i-1}^+) = \mathbf{z}_{i-1}$ until $t = t_i$. The projection operation is then applied, yielding $\Phi_i\mathbf{z}_{i-1} = P_{t_i}\mathbf{v}'(t_i^-)$. Similarly, the product of the transpose matrix Φ_i^T with an arbitrary $\mathbf{z}_i$ requires integration of the homogeneous adjoint equation

$$\frac{d\mathbf{w}}{dt} + \frac{\partial f}{\partial u}^T\mathbf{w} = 0 \tag{18}$$

backwards in time with the terminal condition $\mathbf{w}(t_i^-) = P_{t_i}\mathbf{z}_i$ until $t = t_{i-1}$. The product is then given by $\Phi_i^T\mathbf{z}_i = \mathbf{w}(t_{i-1}^+)$. A detailed algorithm to solve (16) is available in [1]. An adjoint algorithm for multiple control parameters is also available. The sensitivity $d\bar{J}/ds$ is obtained by integrating

$$\frac{d\bar{J}}{ds} = \frac{1}{T}\sum_{i=0}^{K-1}\int_{t_i}^{t_{i+1}} \left\langle \frac{\partial J}{\partial u}\bigg|_t, \mathbf{v}' \right\rangle dt + \frac{1}{T}\sum_{i=0}^{K-1} \frac{\langle \mathbf{f}_{i+1}, \mathbf{v}'(t_{i+1})\rangle}{\|\mathbf{f}_{i+1}\|_2^2}(\bar{J} - J_{i+1}) + \frac{\partial \bar{J}}{\partial s} \tag{19}$$

Convergence of the iterative method is however slow without preconditioning of the system. We propose an efficient preconditioner in Section (3) below.

3 A Preconditioner for the MSS Schur Complement

The convergence rate of iterative Krylov subspace solvers for symmetric, positive definite matrices (like S) depends on the distribution of the matrix eigenvalues [23]. For such systems, the eigenvalues are all positive and real. If all of them are tightly clustered around a few single points away from the origin, then one would expect fast convergence. On the other hand, widely spread eigenvalues without tight clustering can lead to slow convergence. The objective of a preconditioner is to reduce the spread of the eigenvalues, and thereby reduce the condition number, $\kappa(S) = \frac{\mu_{max}(S)}{\mu_{min}(S)}$, where $\mu_{max}(S)$ and $\mu_{min}(S)$ are the maximum and minimum eigenvalues of S, respectively.

An extensive survey of preconditioners for saddle point problems, such as the MSS KKT system (15), is available in [24]. Preconditioners can be applied to the 2×2 block system (15) or directly to the Schur complement system (16). In either case, an easily invertible approximation of S is required. There is an additional restriction, namely that the preconditioner should be matrix free, i.e. it should rely on matrix-vector products only (computing and storing S is out of the question for long trajectories and large N).

Efficient approximations can be made if one takes into account the structure of the problem. For example, in the finite element solution of the incompressible Navier-Stokes equations, the Schur complement can be interpreted as a discretisation of a second order diffusion operator. This is not surprising because pressure, that plays the role of a Lagrange multiplier that enforces incompressibility, is governed by a Poisson equation. For this operator, the action of S^{-1} can be efficiently approximated by a multigrid iteration (more details can be found in [25, 26]).

For the present case, however, such a route is not viable. In [21] a second order partial differential equation for the adjoint variable $\mathbf{w}$ is derived, but this is too complex to solve, and it is not evident how it can be simplified in order to aid the construction of a preconditioner.

Recently, McDonald et al. [27] proposed a block circulant preconditioner for the 'all-at-once' evolution of a linear system of ODEs with constant coefficients. Here 'all-at-once' means that the space/time problem is written as a monolithic linear system. The authors exploit the block Toeplitz structure of the system to develop an efficient preconditioner that results in the number of Krylov iterations being independent of the number of time-steps. Unfortunately, this approach is also unsuitable for our problem. The block Toeplitz structure of the matrix in [27] is a direct consequence of the fact that the coefficients of the ODE are constant. In our problem, however, the blocks Φ_i of matrix A in equation (14) depend on time t and the position along the reference trajectory.

In the course of this work, we tried several preconditioners that have been proposed in the literature for the solution of general saddle systems, which however do not exploit the properties of the underlying physical problem. For example, Cao et al. [28] proposed a splitting of the block matrix (15) into two matrices, one of which is used as a left preconditioner. The splitting contains an adjustable parameter, and theoretical analysis shows that the largest eigenvalue of the preconditioned system is one (independent of the value of the adjustable parameter), provided that the preconditioner is applied exactly. In practice, however, this cannot be achieved, and the preconditioner must be applied approximately. Although the method relies on matrix-vector products only, it is very time-consuming. The preconditioner of Golub et al. [29] guarantees that the eigenvalues remain bounded within two intervals $[-1, {}^{(1-\sqrt{5})}\!/_2]$ and $[1, {}^{(1+\sqrt{5})}\!/_2]$. This tight clustering of eigenvalues ensures fast convergence, but constructing the preconditioner is very expensive. Standard preconditioners for general matrices, such as incomplete LU decomposition [23], are also not suitable, and they also require storage of the matrix.

A new approach is therefore warranted. The central idea is to identify the fastest growing modes of matrix A and annihilate them. In this way, it is expected that it would be easier to minimise the norm (13a) while preserving the continuity of $\mathbf{v}(t)$ across segments. This can be achieved using partial singular value decomposition.

3.1 A preconditioner based on partial singular value decomposition

Recall that the Schur complement system (16) is $AA^T\underline{\mathbf{w}} = \underline{\mathbf{b}}$. The Singular Value Decomposition (SVD) of A reads

$$A = U\Sigma V^T \tag{20}$$

where U is a $NK \times NK$ unitary matrix, Σ is a $NK \times N(K+1)$ quasi-diagonal matrix, and V is a $N(K+1) \times N(K+1)$ also unitary matrix. The columns of U contain the left singular vectors of A, while the right singular vectors of A make up the columns of V. Σ contains the singular values of A, which we denote by $\sigma(A)$, in the $NK \times NK$ diagonal sub-matrix (the last N columns consist of zeros, which we ignore). The singular values are ordered from largest to smallest, i.e. the diagonal elements are $\Sigma_{ii} = \sigma_i \quad (i = 1 \ldots NK)$ and $\sigma_1 > \sigma_2 > \ldots > \sigma_{NK}$. Note that $\sigma(A) = \sqrt{\mu(S)}$, where $\mu(S)$ are the eigenvalues of S. Using the fact that $V^TV = I$ (since the columns of V are orthonormal) we can write S as

$$S = AA^T = U\Sigma\Sigma^TU^T = U\Sigma^2U^T \tag{21}$$

and since $U^TU = I$ we have

$$S^{-1} = U\Sigma^{-2}U^T \tag{22}$$

Equation (22) forms an exact preconditioner, which of course is not practical to compute. However, we can form an approximate preconditioner using the leading l singular modes only, i.e. perform a partial SVD. S can be written as

$$S = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1^2 & 0 \\ 0 & \Sigma_2^2 \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} = U_1 \Sigma_1^2 U_1^T + U_2 \Sigma_2^2 U_2^T \tag{23}$$

where U is partitioned as $U = \begin{bmatrix} U_1 & U_2 \end{bmatrix}$, with $U_1 = \begin{bmatrix} u_1 & u_2 & ... & u_l \end{bmatrix}$ and $U_2 = \begin{bmatrix} u_{l+1} & u_{l+2} & ... & u_{NK} \end{bmatrix}$. Matrix $\Sigma_1^2 = diag(\sigma_1^2, \sigma_2^2, ..., \sigma_l^2)$ contains the singular values corresponding to U_1 and $\Sigma_2^2 = diag(\sigma_{l+1}^2, \sigma_{l+2}^2, ..., \sigma_{NK}^2)$ contains the rest. Replacing the diagonal submatrix Σ_2^2 with the identity matrix $I_2 = I_{(NK-l)}$, we get an approximation $\hat{S}_{(l)}$

$$\hat{S}_{(l)} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1^2 & 0 \\ 0 & I_2 \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} = U_1 \Sigma_1^2 U_1^T + U_2 U_2^T \tag{24}$$

which can be easily inverted to give

$$\hat{S}_{(l)}^{-1} = M_{(l)} = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1^{-2} & 0 \\ 0 & I_2 \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} = U_1 \Sigma_1^{-2} U_1^T + U_2 U_2^T \tag{25}$$

The product $M_{(l)} S$ now becomes

$$M_{(l)} S = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \Sigma_2^2 \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix} \tag{26}$$

which indicates that $M_{(l)}$ has deflated the l largest singular values of matrix S to 1, while leaving the rest unaltered. In order to avoid computing the columns U_2 in (25), we invoke the orthogonality relation $U_2 U_2^T = I - U_1 U_1^T$, and we get

$$M_{(l)} = U_1 \Sigma_1^{-2} U_1^T + (I - U_1 U_1^T) \tag{27}$$

Therefore in order to form $M_{(l)}$, we need Σ_1 and U_1 only. We deploy $M_{(l)}$ as a left preconditioner to convert (16) into a better conditioned system of the form

$$M_{(l)} S \underline{\mathbf{w}} = M_{(l)} \underline{\mathbf{b}} \tag{28}$$

and we solve for $\underline{\mathbf{w}}$.

Preconditioning based on partial singular value decomposition has been applied in the past, for example in [30] to solve ill-conditioned least squares problems arising in image de-blurring applications. A preconditioner based on the partial Krylov-Schur decomposition of a matrix (see [31] for details) was also used in [32] to accelerate the computation of limit cycles for thermoacoustic systems. The preconditioner had a form very similar to (27), the difference being that instead of the diagonal matrix Σ_1^{-2} in the first term on the right-hand side, the inverse of an upper triangular $l \times l$ matrix was taken. The preconditioner reduced the condition number significantly, speeded up the convergence of GMRES, and lead to modest overall cost savings (mainly due to the cost of converging the eigenvalues used to form the preconditioner). Sánchez and Net [33] also used a similar preconditioner for accelerating the convergence of a multiple shooting algorithm for finding periodic orbits.

The Lanczos bidiagonalization algorithm [34] (implemented in the 'svds' command of MATLAB) can be employed to form an approximation to $M_{(l)}$ using q iterations. The algorithm requires matrix-vector products only. The approximation of $M_{(l)}$ after q iterations is

$$M_{(l)}^{(q)} = U_1^{(q)} \Sigma_1^{-2(q)} U_1^{T(q)} + (I - U_1^{(q)} U_1^{T(q)}) \tag{29}$$

Computing $U_1^{(q)}$ and $\Sigma_1^{(q)}$ using the function 'svds' of MATLAB requires evaluation of a large number of matrix-vector products Ax and $A^T z$, where x and z are algorithm-generated vectors of length $N(K+1)$ and NK respectively. More specifically, each iteration requires at least $l+2$ (the smallest permissible Krylov subspace dimension) applications of A and A^T. Even with one or two iterations of the algorithm, the cost of computing $M_{(l)}^{(q)}$ would most likely outweigh the cost savings due to preconditioning. A cheaper alternative is therefore required.

3.2 A simplified block diagonal preconditioner

We can form a much cheaper preconditioner if we consider matrix $\tilde{A}$, obtained from A by neglecting the upper diagonal, i.e.

$$\tilde{A} = \begin{bmatrix} -\Phi_1 & 0 & & & \\ & -\Phi_2 & 0 & & \\ & & \ddots & \ddots & \\ & & & -\Phi_K & 0 \end{bmatrix} \tag{30}$$

The corresponding Schur complement $\tilde{S} = \tilde{A}\tilde{A}^T$ is a block diagonal matrix that takes the form

$$\tilde{S} = \begin{bmatrix} \Phi_1\Phi_1^T & & & \\ & \Phi_2\Phi_2^T & & \\ & & \ddots & \\ & & & \Phi_K\Phi_K^T \end{bmatrix} \tag{31}$$

This form indicates that each segment is now decoupled from the two neighbouring segments on either side. A Block Diagonal Preconditioner (BDP), $\mathbf{M}_{BD}$, can be constructed that approximates the inverse, $\tilde{S}^{-1}$, i.e.

$$\mathbf{M_{BD}} \approx \tilde{S}^{-1} = \begin{bmatrix} (\Phi_1\Phi_1^T)^{-1} & & & \\ & (\Phi_2\Phi_2^T)^{-1} & & \\ & & \ddots & \\ & & & (\Phi_K\Phi_K^T)^{-1} \end{bmatrix} \tag{32}$$

Each diagonal block in (32) can be approximated as before using partial singular value decompositions i.e.

$$\mathbf{M_{BD}}_{(l)}^{(q)} = diag(\mathbf{M}_{(l),1}^{(q)}, \mathbf{M}_{(l),2}^{(q)}, ..., \mathbf{M}_{(l),K}^{(q)}) \tag{33a}$$

$$\mathbf{M}_{(l),i}^{(q)} = \underline{U}_{1,i}\underline{\Sigma}_{1,i}^{-2}\underline{U}_{1,i}^{T} + (I - \underline{U}_{1,i}\underline{U}_{1,i}^{T}) \tag{33b}$$

where i is the segment number, q is the number of iterations and l is the number of retained singular modes in each segment. $\underline{\Sigma}_{1,i}$ and $\underline{U}_{1,i}$ are the matrices corresponding to the l singular values and the left singular vectors of Φ_i, respectively. The superscript (q) and subscript (l) have been removed from $\underline{\Sigma}_{1,i}$ and $\underline{U}_{1,i}$ for clarity.

The preconditioner $\mathbf{M_{BD}}_{(l)}^{(q)}$ has several advantages. Firstly, it is much cheaper to construct than $M_{(l)}^{(q)}$; in fact, it is $O(K)$ times cheaper. To see why, recall that computing $M_{(Kl)}$ using Lanczos bidiagonalisation requires at least $Kl + 2$ matrix-vector products with A and A^T. This deflates the leading Kl eigenvalues. Preconditioning with $\mathbf{M_{BD}}_{(l)}$ is similar to preconditioning with $M_{(Kl)}$ in terms of eigenvalue deflation. However, computing $\mathbf{M_{BD}}_{(l)}$ only requires $l + 2$ matrix-vector products with Φ_i and Φ_i^T per segment (i.e. the equivalent of $l + 2$ applications of A and A^T). Secondly, the computation of $\mathbf{M_{BD}}_{(l)}^{(q)}$ is fully parallelisable in time. Each processor is assigned to one segment and computations can proceed independently, because each segment is treated separately, i.e. there is no message passing between processors. On the other hand, computing $M_{(l)}$ requires passing of all the degrees of freedom of the computational domain from the processor operating at segment $i + 1$ to the processor operating at segment i (due to the off-diagonal identities of matrix A). This can be overlapped with computation at segment i, but it is not clear a priori that message passing can be completed before the end of computation (this is probably case dependent). Thirdly, storage costs with respect to $M_{(Kl)}$ are reduced by a factor of K (for $\mathbf{M_{BD}}_{(l)}$, the singular vector matrix has Kl vectors, each with length N, while for $M_{(Kl)}$ it has also Kl vectors, but with length KN each). Finally, both S and $\mathbf{M_{BD}}_{(l)}^{(q)}$ are symmetric, positive definite matrices, making the conjugate gradient method applicable.

We use $\mathbf{M_{BD}}_{(l)}^{(q)}$ as a left preconditioner for the original system, i.e. we solve

$$\mathbf{M_{BD}}_{(l)}^{(q)} S\underline{\mathbf{w}} = \mathbf{M_{BD}}_{(l)}^{(q)}\underline{\mathbf{b}} \tag{34}$$

In the following two sections, we investigate the condition number of the matrix $\mathbf{M_{BD}}_{(l)}^{(q)} S$ and the performance of the preconditioner for two standard problems; the Lorenz system and the Kuramoto Sivashinsky equation.

4 Application to the Lorenz System

The well-known Lorenz system takes the form,

$$\begin{aligned} \frac{dx}{dt} &= \sigma(y-x) \\ \frac{dy}{dt} &= x(\rho - z) - y \\ \frac{dz}{dt} &= xy - \beta z \end{aligned} \tag{35}$$

where σ, ρ and β are system parameters. The Lorenz system is a common test case for chaotic sensitivity analysis applications. For $\sigma = 10$ and $\beta = {}^{8}/_{3}$, the largest Lyapunov exponent λ_{max} increases from $\lambda_{max} \approx 0.8$ at $\rho = 24$ to $\lambda_{max} \approx 1.7$ at $\rho = 96$ [35] (roughly a linear growth with ρ with frequent dips). The remaining exponents are $\lambda_2 = 0$ and $\lambda_3 < 0$. The objective function considered is

$$\overline{J} = \frac{1}{T}\int_0^T z\, dt \tag{36}$$

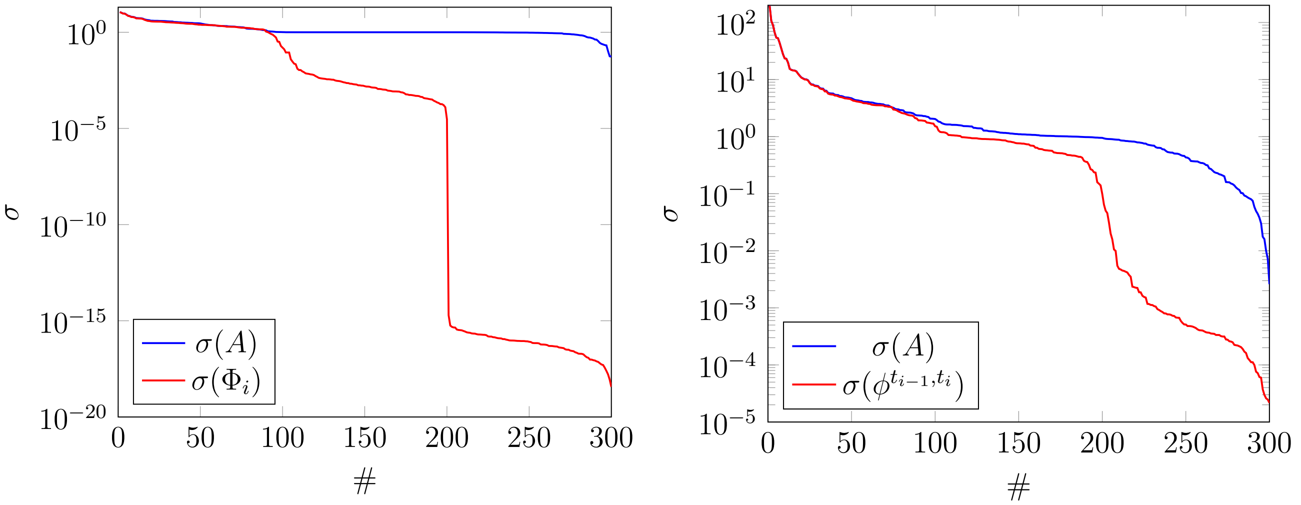
Using $\sigma = 10$ and $\beta = {}^{8}/_{3}$, sensitivities were sought with respect to ρ, i.e. ${}^{d\overline{J}}/_{d\rho}$. MATLAB's variable step Runge-Kutta solver (ode45) was used to compute the trajectory $\mathbf{u}_{ref}(t)$ and to perform all constraint and adjoint integrations.

Figure (2a) shows a comparison of the singular values of A with the union of the singular values of all Φ_i, ordered from largest to smallest. There are in total $3K = 300$ singular values for the case examined. Each segment is located at a different place on the attractor, and has 3 local finite Lyapunov exponents associated with it. It is well known that such local exponents can fluctuate significantly around the average values as the trajectory is traced [36]. We observe that $\sigma(\Phi_i)$ is close to $\sigma(A)$ for the largest K values, and thereafter the two curves start to deviate. The last K values of $\sigma(\Phi_i)$ are very small, which is expected, since $\Phi_i = P_{t_i}\phi^{t_{i-1},t_i}$ is almost singular due to the projection P_{t_i}. To remove the effect of singularity that distorts the comparison, Figure (2b) shows $\sigma(A(\phi^{t_{i-1},t_i}))$ and $\sigma(\phi^{t_{i-1},t_i})$, where $A(\phi^{t_{i-1},t_i})$ is evaluated using ϕ^{t_{i-1},t_i} in equation (14), i.e. without applying the projection P_{t_i} to the state transition matrix. The matching for the first K singular values is now much more clearly seen.

In [1] it was shown that $\mu_{max}(S) = \sigma^2_{max}(A)$ is related to the largest singular value of Φ_i across all segments, $\sigma_{max}(\Phi_i)$, as $\mu_{max}(S) = 1 + \sigma^2_{max}(\Phi_i)$, provided that $\sigma_{max}(\Phi_i) \gg 1$, in which case $\mu_{max}(S) \approx \sigma^2_{max}(\Phi_i)$. Figure (2) shows that this approximation holds for many more eigenvalues. For the Lorenz system, it holds for approximately K eigenvalues, each corresponding to the local positive Lyapunov exponent for each segment.

Figure (3) zooms in on the largest K eigenvalues for two different ΔT values. The singular values $\sigma(\Phi_i)$ approximate $\sigma(A)$ better for the larger ΔT value. This is because the diagonal blocks of matrix A become more dominant for $\Delta T = 1$ and the off-diagonal identity matrices can be neglected (refer to (30)) without impairing the accuracy of the large singular values.

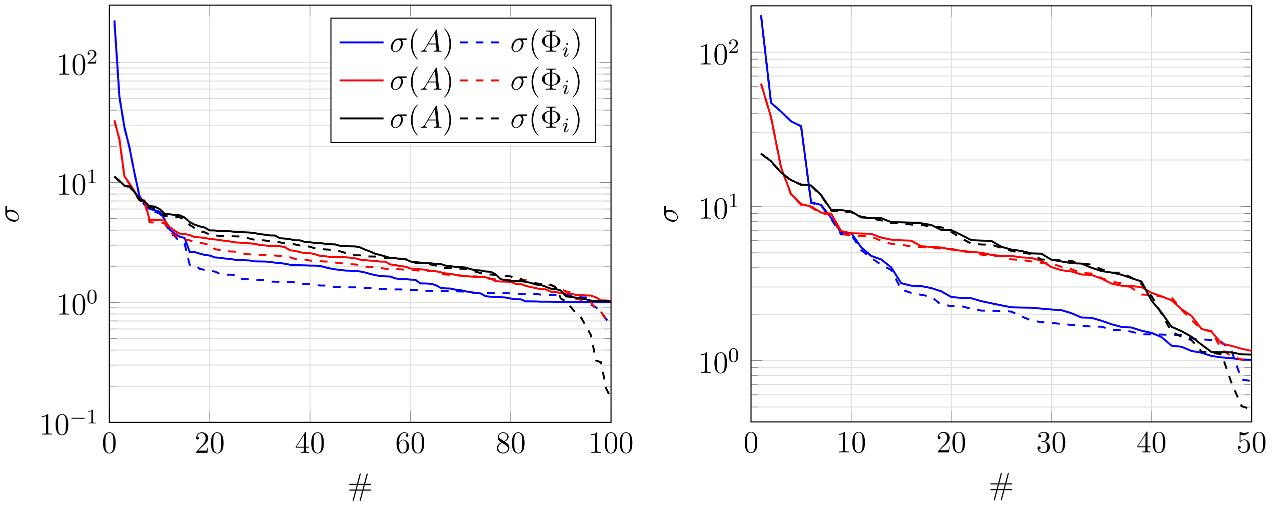
Next, we investigate the spectrum of the preconditioned system, starting with the exact preconditioner $M_{(l)}$ (27). The l largest singular values and vectors of S used to form $M_{(l)}$ have been



(a) Projections P_{t_i} applied

(b) No projections applied

Figure 2: A distribution of $\sigma(A)$, $\sigma(\Phi_i) = \sigma(P_{t_i}\phi^{t_{i-1},t_i})$ and $\sigma(\phi^{t_{i-1},t_i})$ ordered from the largest to the smallest values. Obtained for $\rho = 80$ with $T = 50$ and $\Delta T = 0.5$ ($K = 100$ segments).

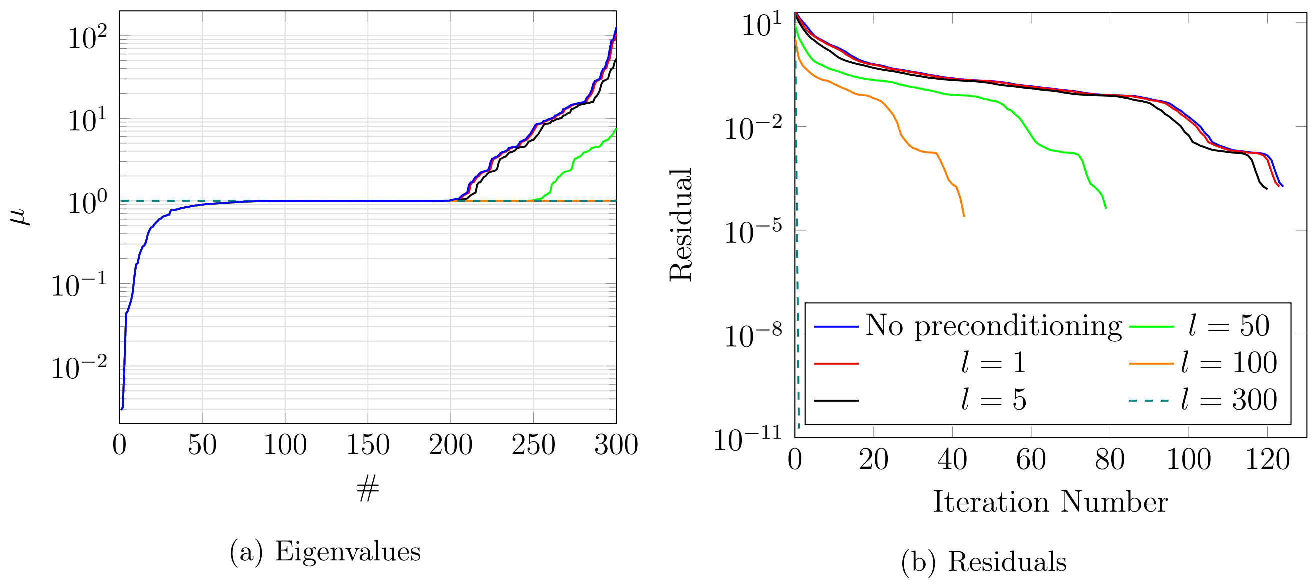


(a) $\Delta T = 0.5$ ($K = 100$ segments)

(b) $\Delta T = 1$ ($K = 50$ segments)

Figure 3: A distribution of the largest K values of $\sigma(A)$ and the largest $\sigma(\Phi_i)$ of each segment. Obtained with $T = 50$. Blue: $\rho = 40$, red: $\rho = 60$, black: $\rho = 80$

obtained iteratively until convergence. Figure (4a) shows the eigenvalues of $\mu(S)$ and $\mu(M_{(l)}S)$. For the case examined, the matrix S has approximately $K = 100$ eigenvalues $\mu(S) > 1$ (blue line), corresponding to the local positive Lyapunov exponent in each segment. Eigenvalues equal to 1 correspond to the neutrally stable exponent, and the ones with $\mu(S) < 1$ to the stable exponent. Different values of l were used to construct $M_{(l)}$ (the values are reported in panel b). We notice that $M_{(l)}$ effectively deflates the l largest eigenvalues of the preconditioned system $\mu(M_{(l)}S)$ to 1, while leaving the rest unaltered. As l increases, more and more eigenvalues are clustered closer to 1, leading to increasingly faster convergence (shown in Figure 4b). When $l = K(= 100)$, $\mu_{max}(M_{(l)}S) = 1$ and convergence still takes about 50 iterations. For the limiting



(a) Eigenvalues

(b) Residuals

Figure 4: Eigenvalues (ordered from smallest to largest) and convergence residuals for the original system S (blue line) and the preconditioned system $M_{(l)}S$ for different l (number of singular values). Obtained for $\rho = 80$ with $T = 50$ and $\Delta T = 0.5$ ($K = 100$ segments).

case of $l = 3K(= 300)$ all eigenvalues are equal to 1 (see left panel) and convergence is achieved in just one iteration, as expected. This indicates that we need not only deflate the large ($\mu{>}1$), but also to increase (regularise) the very small ($0 < \mu \ll 1$) eigenvalues for fast convergence. Such small eigenvalues make the system more singular, with implications to the accuracy of the computed sensitivity; this is explored later in Section (6).

We explore next the performance of the BDP (33). Figure (5) shows $\mu(S)$, $\mu(M_{(25)}S)$ and $\mu(\mathbf{M_{BD}}_{(1)}S)$. The number of segments is $K = 25$, and we have computed only one singular value until convergence in each segment. Matrix $\mathbf{M_{BD}}_{(1)}S$ has a very similar eigenvalue spectrum to $\mu(M_{(25)}S)$. It can be seen that $\mu_{max}(M_{(25)}S) \approx 1$, while $\mu_{max}(\mathbf{M_{BD}}_{(1)}S) \approx 2$. Most importantly, $\mathbf{M_{BD}}_{(1)}S$ has clustered the K largest eigenvalues in the interval $[1, 2]$. This indicates that the approximate BDP preconditioner encapsulates reliable information for the fastest growing modes, which results in the suppression of $\mu_{max}(\mathbf{M_{BD}}_{(1)}S)$ by four orders of magnitude. However, there is a slight reduction in $\mu_{min}(\mathbf{M_{BD}}_{(1)}S)$ with respect to $\mu_{min}(S)$ and $\mu_{min}(M_{(25)}S)$.

Figure (6) shows the convergence rates for different ρ. As expected, using the exact preconditioner $M_{(l)}$ does provide considerably faster convergence compared to the original system (16). Of course $M_{(25)}$ performs better than $\mathbf{M_{BD}}_{(1)}$ (since the latter assumes a block diagonal structure of A), but the cheaper cost of constructing and storing $\mathbf{M_{BD}}_{(1)}$ makes it a much more practical alternative.

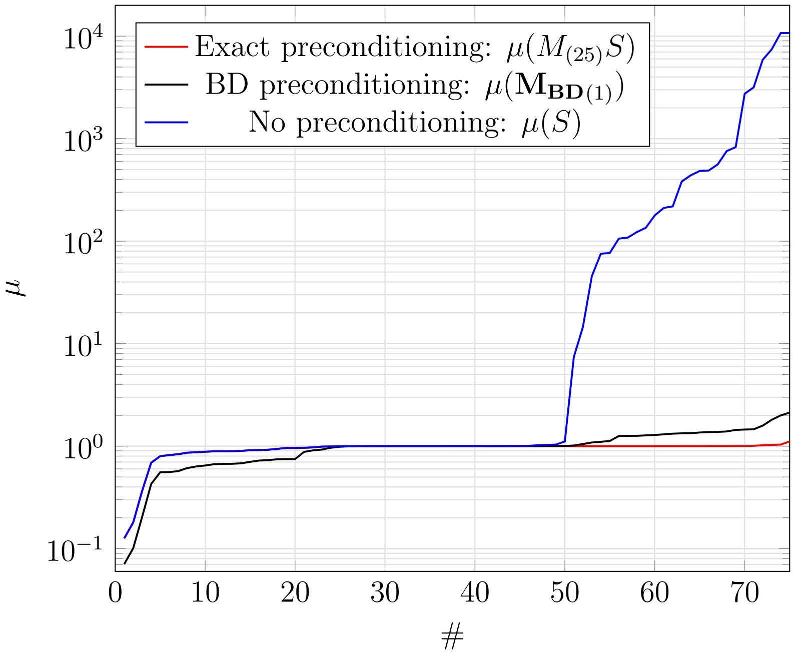


Figure 5: Eigenvalues of the original and preconditioned systems for $T = 50$, $\Delta T = 2$ ($K = 25$) and $\rho = 80$.

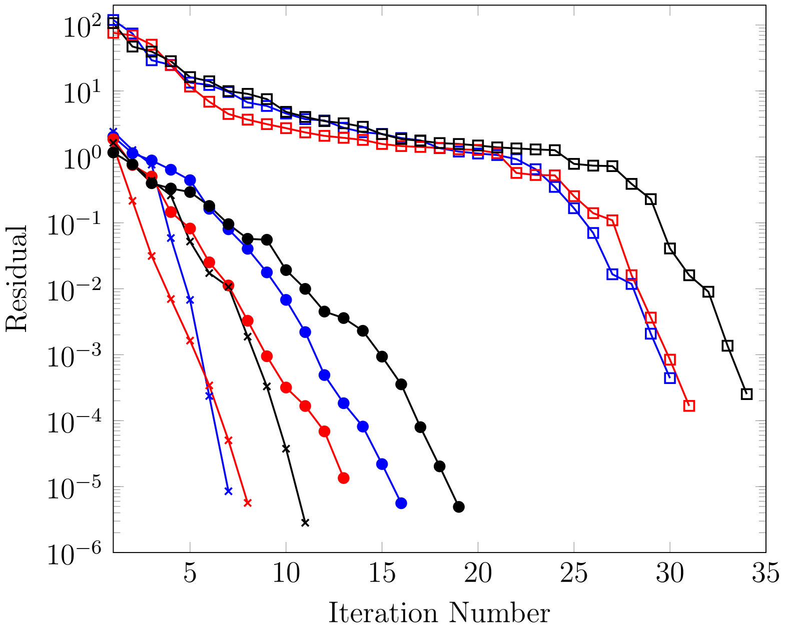


Figure 6: Convergence history for the original and preconditioned systems using $T = 50$ and $\Delta T = 2$ ($K = 25$). Blue: $\rho = 40$, red: $\rho = 60$, black: $\rho = 80$. Squares: S, crosses: $M_{(25)}S$, circles: $\mathbf{M_{BD(1)}}S$.

5 Application to the Kuramoto Sivashinsky Equation

In this section we apply the MSS method with the BDP (33) to a slightly modified version of the Kuramoto Sivashinsky (KS) equation [37]:

$$\begin{aligned}
&\frac{\partial u}{\partial t} = -(u+c)\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4} \\
&x \in [0, L] \\
&u(0,t) = u(L,t) = 0 \\
&\left.\frac{\partial u}{\partial x}\right|_{x=0} = \left.\frac{\partial u}{\partial x}\right|_{x=L} = 0
\end{aligned} \tag{37}$$

where $L = 128$ to ensure chaotic solutions [38]. The Dirichlet and Neumann boundary conditions ensure ergodicity. The spatial derivatives were discretised into $N+2$ nodes (N interior nodes and two boundary nodes) using second order central finite difference approximations on a uniform grid. The Neumann boundary conditions were enforced using ghost nodes. The MATLAB command ode45 was used for time integration. Figure (7) shows the solution to (37) for $c = 0$ and $c = 0.8$. In both cases organised structures with a dominant wavelength appear [39], corresponding to the light turbulence regime.

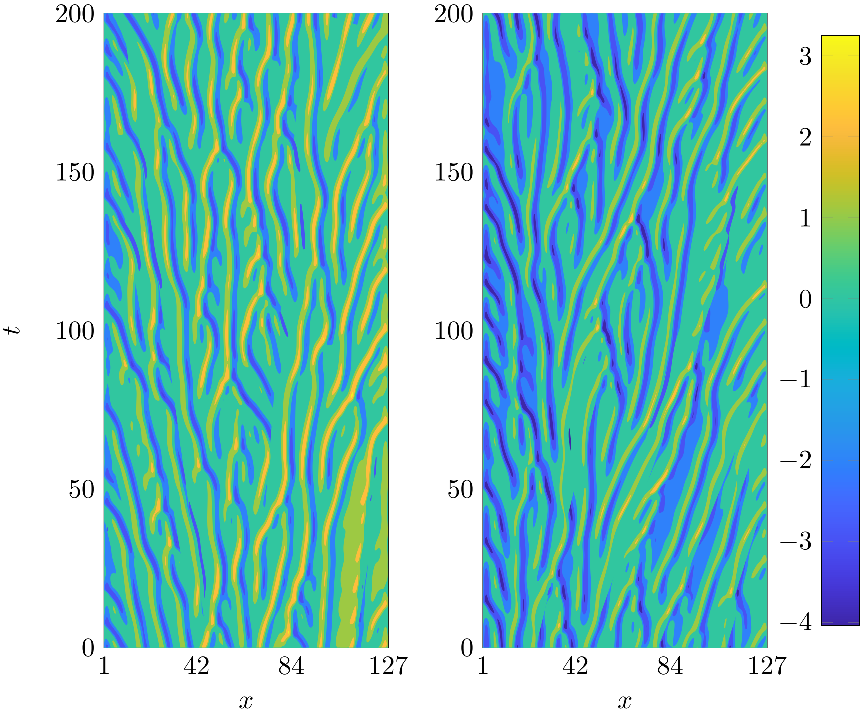


Figure 7: Space-time plot of the solution $\mathbf{u}(x,t)$ for $L = 128$ using $N = 255$ nodes in the x-direction (left: $c = 0$, right: $c = 0.8$). The integration time interval is $[-1000, 200]$ with initial condition $\mathbf{u}_0 = 1$.

Two objective functions were considered:

$$\langle \bar{u} \rangle = \frac{1}{TL}\int_0^T \int_0^L u\,dx\,dt \tag{38a}$$

$$\langle \overline{u^2} \rangle = \frac{1}{TL}\int_0^T \int_0^L u^2\,dx\,dt \tag{38b}$$

and their sensitivities to parameter c, ${}^{d\langle \bar{u} \rangle}/{}_{dc}$ and ${}^{d\langle \overline{u^2} \rangle}/{}_{dc}$, were sought. The MSS segment size for all the cases studied was $\Delta T = 10$ (based on $\lambda_{max} \approx 0.1$ for $c = 0$ and $c = 0.8$ [37]) unless otherwise stated.

The ideal use of the BDP (33) involves choosing the parameters q and l such that the total number of matrix-vector products involving Φ_i and Φ_i^T is minimised. Considering that there are 15 positive Lyapunov exponents for $c = 0.8$ [37], it seems reasonable to choose $l = 15$ in each segment to construct (33). We explore the effect of different values of l later.

Figure (8) shows $\sigma(A)$ and the union of the computed 15 singular values of $\sigma(\Phi_i)$ for each segment, ordered from largest to smallest. Both the converged values and the values with $q = 1$ and $q = 2$ iterations are shown. It is clear that we can obtain accurate approximations to $\sigma(A)$ even with just 1 iteration when we evaluate $\sigma(\Phi_i)$. The curves start to deviate when the singular values approach unity.

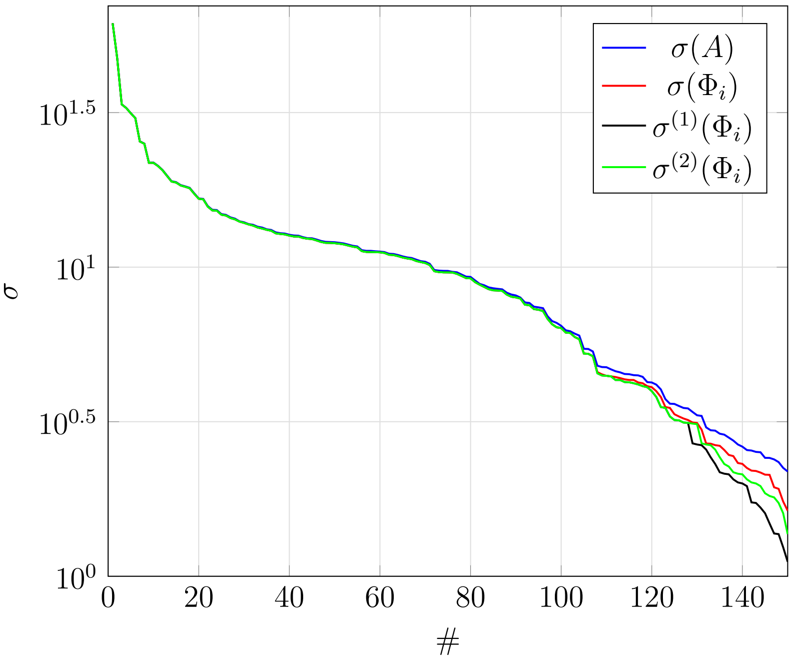


Figure 8: $\sigma(A)$ (blue line) and the largest 15 $\sigma(\Phi_i)$ for all segments ordered from largest to smallest (red: exact, black: $q = 1$ iteration, green: $q = 2$ iterations). Computed for $N = 127$, $c = 0.8$, $T = 100$ ($K = 10$).

Figure (9) shows the eigenvalues of the preconditioned system. Using $\mathbf{M_{BD}}_{(l)}^{(q)}$ has reduced the maximum eigenvalue (and therefore the condition number κ) by more than 2 orders of magnitude. The spectra of the inexact BDP systems $\mathbf{M_{BD}}_{(15)}^{(1)}S$ and $\mathbf{M_{BD}}_{(15)}^{(2)}S$ are very similar to the spectra of the exact BDP system $\mathbf{M}_{\mathbf{BD}(15)}S$, (with $\mu_{max}(\mathbf{M}_{\mathbf{BD}}{}_{(15)}^{(1)}) \approx 15$, $\mu_{max}(\mathbf{M}_{\mathbf{BD}}{}_{(15)}^{(2)}S) \approx 12$ and $\mu_{max}(\mathbf{M}_{\mathbf{BD}(15)}S) \approx 6$).

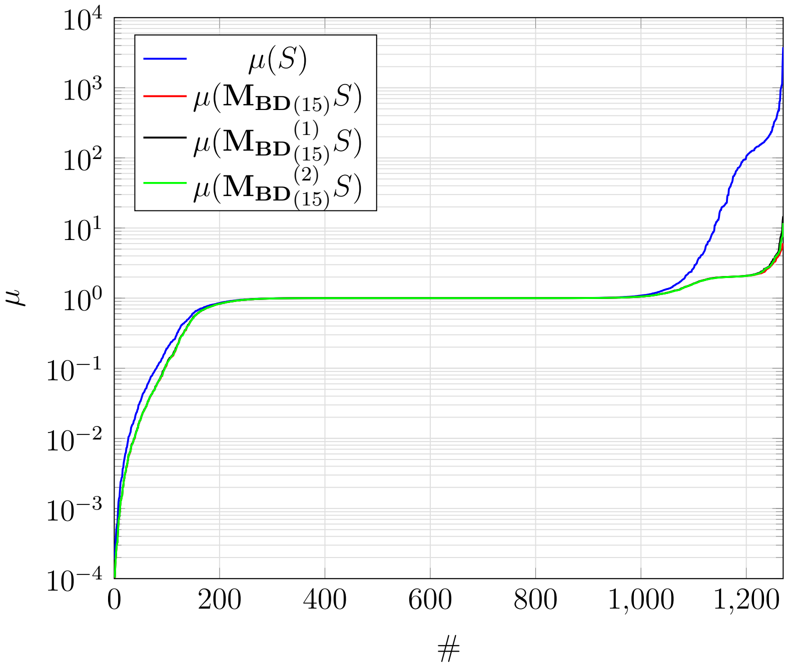
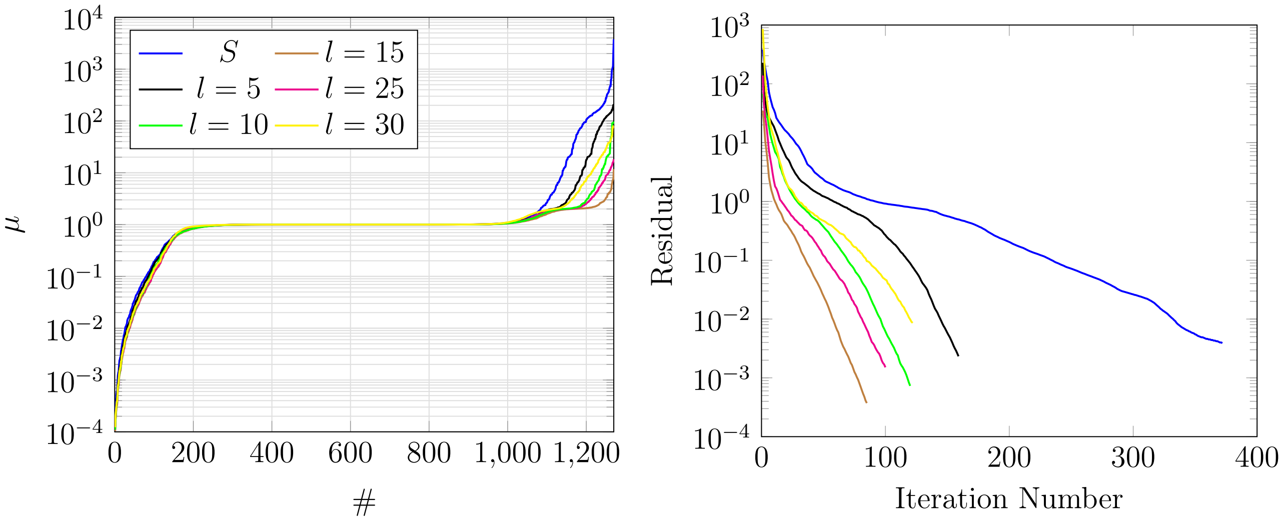


Figure 9: Eigenvalues of the original system (blue line), the exact BDP ($\mathbf{M}_{\mathbf{BD}(15)}$) and inexact BDP ($\mathbf{M}_{\mathbf{BD}}{}_{(15)}^{(1)}$,$\mathbf{M}_{\mathbf{BD}}{}_{(15)}^{(2)}$) using $l = 15$.

We study next the effect of l on the spectra of $\mathbf{M}_{\mathbf{BD}}{}_{(l)}^{(q)}S$ and on the convergence rate. The value $q = 2$ was kept constant. Results for different l are shown in Figure (10). Increasing l up to $l = 15$ improves the clustering of eigenvalues and leads to faster convergence rates (panel b).



(a) Eigenvalues

(b) Residuals

Figure 10: Eigenvalues and residuals of the original system S (blue line), and of the BDP system ($\mathbf{M}_{\mathbf{BD}}{}_{(l)}^{(2)}S$) for $N = 127$, $T = 100$ and $c = 0.8$. The preconditioners were constructed for different l, and their residuals were found with a regularisation value $\gamma = 0.01$.

Interestingly, a further increase to $l = 25$ or $l = 30$ increases the condition number and slows down the convergence (Figure 10b). This indicates that after a certain value of l, adding more singular modes starts to provide unreliable information to the preconditioner $\mathbf{M}_{\mathbf{BD}}{}_{(l)}^{(q)}$. This is because as l increases, the singular values approach unity, and $\sigma(\Phi_i)$ starts to diverge from $\sigma(A)$. This was also observed clearly in Figure (2) for the Lorenz system.

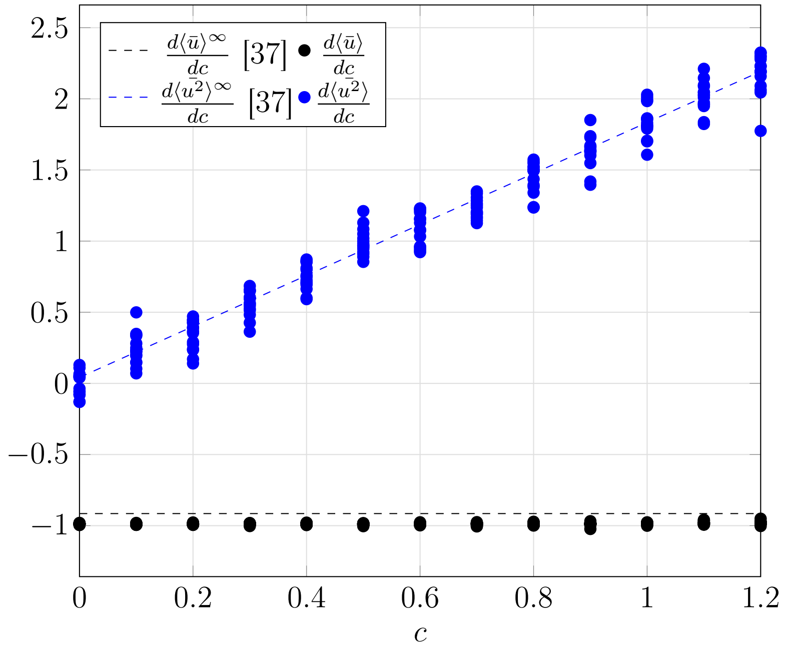


Figure 11: Sensitivities of $\langle\bar{u}\rangle$ and $\langle\overline{u^2}\rangle$ to parameter c. The dashed lines [37] are shown for reference and were obtained by differentiating curve fits for $T \to \infty$. The black dots $({}^{d\langle\bar{u}\rangle}/{}_{dc})$ and the blue dots $({}^{d\langle\overline{u^2}\rangle}/{}_{dc})$ were obtained for $T = 100$ trajectories with random $\mathbf{u}_0$ using MSS for $N = 127$ and $N = 255$, respectively.

Figure (11) shows ${}^{d\langle\bar{u}\rangle}/{}_{dc}$ and ${}^{d\langle\overline{u^2}\rangle}/{}_{dc}$ for 20 and 15 different initial conditions $\mathbf{u}_0$ respectively. The values of parameter c examined are between 0 and 1.2, which correspond to the light turbulence regime. Each data point was computed for a randomly generated initial condition vector $0 < \mathbf{u}_0 < 1$. To obtain $\mathbf{u}(x,t)$, (37) was integrated in the interval $[-1000, 100]$, and MSS was applied to $\mathbf{u}(x,t)$ in the interval $[0, 100]$. The reference data [37] is the derivative of the curve fit of $\langle\bar{u}\rangle$ and $\langle\overline{u^2}\rangle$ vs. c, obtained for very long trajectories ($T = 2000$). Figure (11) shows that MSS slightly under-predicts ${}^{d\langle\bar{u}\rangle}/{}_{dc}$ (similar to [37, 1]). This difference will be discussed in the next section.

6 Effect of the System Condition on the Accuracy of the Computed Sensitivity

In Section (3), a block diagonal preconditioner was presented that can deflate large singular values. While deflating is essential for accelerating convergence, as shown in Figures 6 and 10, very small singular values are present for quasi-hyperbolic systems and cause significant problems in solution accuracy and convergence.

Consider again the sensitivity $d\langle\bar{u}\rangle/dc$ shown in Figure (11). There is a constant bias of about 8% for all simulations (20 random $\mathbf{u}_0$ for each value of c). This bias has been observed previously for the KS in [37, 1]. We can gain helpful insight by considering the analytical solution to the minimisation problem (13), expressed in terms of the singular values and left and right singular vectors of matrix A as follows:

$$\underline{\mathbf{v}}_l = \sum_{i=1}^{l} \left(\frac{u_i^T \underline{\mathbf{b}}}{\sigma_i} \right) v_i \tag{39}$$

This expression indicates that the minimal norm solution is a linear combination of the right singular vectors v_i. The coefficients $u_i^T\underline{\mathbf{b}}/\sigma_i$ are obtained by projecting the right hand side $\underline{\mathbf{b}}$ to the left singular vectors u_i^T and dividing by the singular value σ_i. For $l = NK$, i.e. using all singular modes, (39) yields the same solution obtained by solving (15) iteratively. However, terminating the summation at a value of $l < NK$ allows us to study the effect of a smaller group of singular modes. We can also use a single value of index i to compute the contribution of an individual singular mode to the solution $\underline{\mathbf{v}}$, and therefore to the sensitivity. These properties make the decomposition (39) a very useful tool.

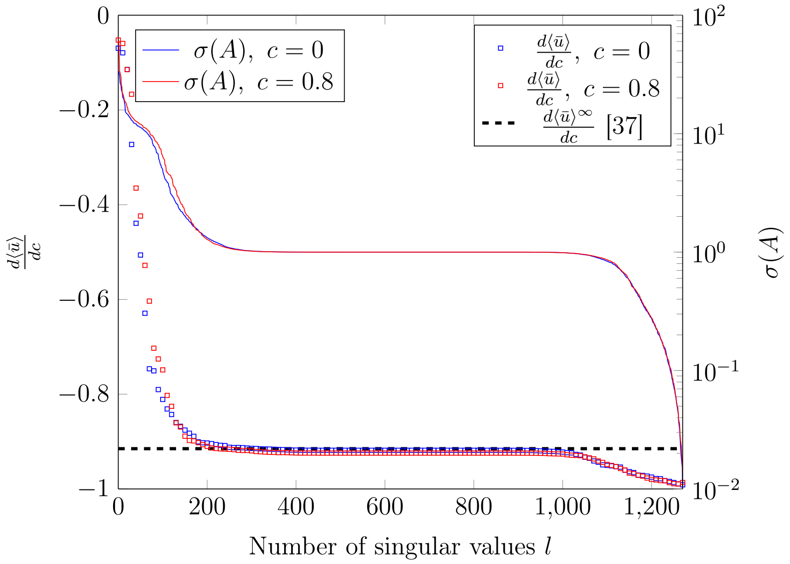
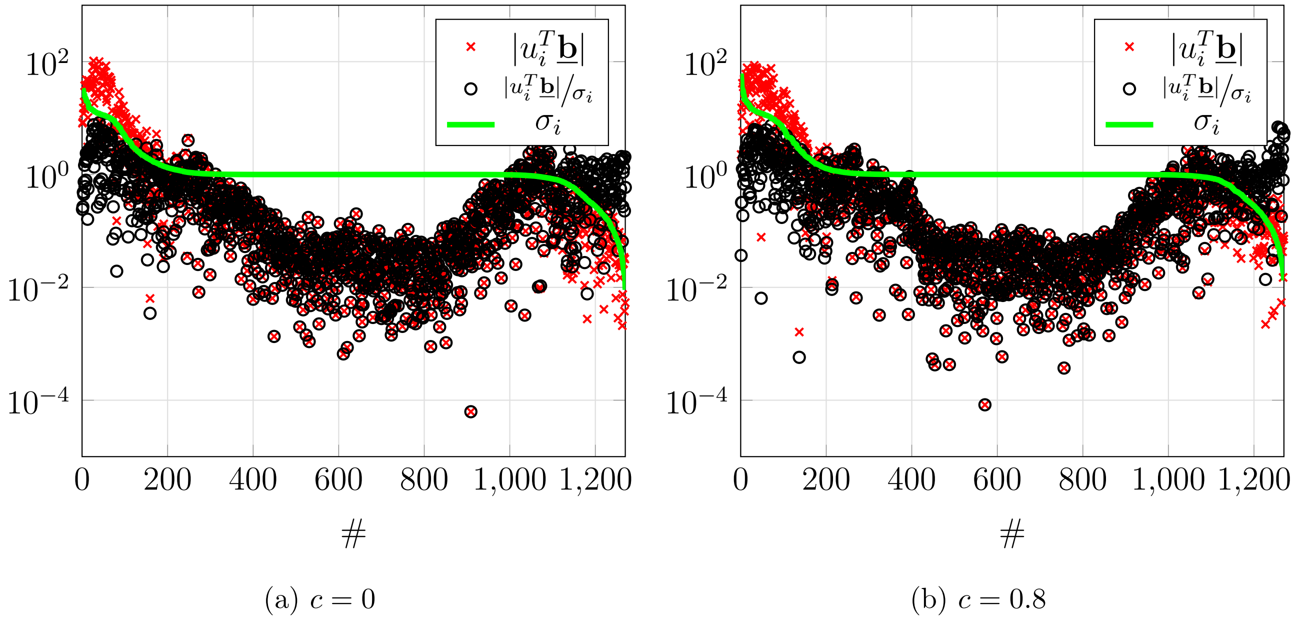


Figure 12: Sensitivities computed using equation (39) for different values of l (KS equation, $T = 100$, $N = 127$, $K = 10$ segments). The solid lines show $\sigma(A)$ (right vertical axis).

Figure (12) shows $d\langle\bar{u}\rangle/dc$ for the KS equation, obtained when different values of l are used to terminate the summation (39). On the same plot, we superimpose the corresponding singular values (right vertical axis). A very interesting behaviour can be noticed. Summing modes with $\sigma \geq 1$ leads to an error in $d\langle\bar{u}\rangle/dc$ of less than 1% of $d\langle\bar{u}\rangle^\infty/dc$. When l is between 250 to 1000, with $\sigma(A) \approx 1$, the sensitivity remains almost constant. However, including in the summation terms with very small values of $\sigma(A)$ degrades the accuracy of the solution, as seen from the divergence of the squares from the dashed line. The final value is equal to the one shown in Figure (11).

We can gain more insight by plotting the coefficients $|u_i^T\underline{\mathbf{b}}|/\sigma_i$, the projections $|u_i^T\underline{\mathbf{b}}|$, and σ_i together. This is known as a discrete Picard plot [40], and is shown in Figure (13). In order to



(a) $c = 0$

(b) $c = 0.8$

Figure 13: Spectral coefficients for $T = 100$ (KS equation)

interpret this plot, we recall that very small singular values indicate an ill-conditioned system, i.e. the solution is very sensitive to small changes in the right hand side $\underline{\mathbf{b}}$. Let's decompose $\underline{\mathbf{b}}$ into an unknown error-free part $\hat{\underline{\mathbf{b}}}$, and a random error part $\underline{\mathbf{e}}$ (for example due to the spatial discretisation and the time advancement scheme), i.e. $\underline{\mathbf{b}} = \hat{\underline{\mathbf{b}}} + \underline{\mathbf{e}}$, with $\|\underline{\mathbf{e}}\|_2 \ll \|\hat{\underline{\mathbf{b}}}\|_2$. Substituting in (39) with $l = NK$ we get

$$\underline{\mathbf{v}} = \sum_{i=1}^{NK} \left(\frac{u_i^T \hat{\underline{\mathbf{b}}}}{\sigma_i} \right) v_i + \left(\frac{u_i^T \underline{\mathbf{e}}}{\sigma_i} \right) v_i = \hat{\underline{\mathbf{v}}} + \underline{\mathbf{v}_e} \tag{40}$$

This equation indicates that very small σ_i can amplify the error component of the solution, $\underline{\mathbf{v}_e} = \sum_{i=1}^{NK} \left(u_i^T \underline{\mathbf{e}}/\sigma_i \right) v_i$. In order to have a meaningful solution, the projection of the error component to the left singular vector, $u_i^T \underline{\mathbf{e}}$, should decay to 0 faster than σ_i. This is known as the Picard condition [41]. Inspection of Figure (13a) shows that, although the values of $u_i^T \underline{\mathbf{b}}$ are quite spread out, it is clear that they decay by 3-4 orders of magnitude for i between $1-600$. The largest values, of order $O(10^2)$, correspond to small i, i.e. to the largest singular values. This indicates that the largest contribution to $\underline{\mathbf{b}}$ originates from the most rapidly growing modes. For $i > 600$, the values of $u_i^T \underline{\mathbf{b}}$ remain between $10^{-2} - 10^0$, i.e. at least 2 orders of magnitude smaller than the maximum. We expect that the small values of $u_i^T \underline{\mathbf{b}}$ originate from the random error $\underline{\mathbf{e}}$. As long as $\sigma_i \approx 1$, their contribution is innocuous, but when σ_i is reduced to 10^{-2}, they are significantly amplified (as demonstrated from the variation of $u_i^T \underline{\mathbf{e}}/\sigma_i$) and contaminate the solution. This explains the sensitivity trend in Figure (11). The same mechanism is valid for $c = 0.8$ (panel b), and in fact for all c in the light turbulent regime.

We have performed simulations for larger c, in the convection dominated regime, and we also observed deviations of the computed sensitivity from the solution obtained by finite differences. However, the aforementioned justification could not account for these deviations. For an explanation for the observed differences in that regime, refer to [37].

We repeated the above exercise also for the Lorenz system. We computed again ${d\bar{J}}/{d\rho}$ (for $T = 100$ and $K = 200$) using equation (39) and compared with finite differences. The green dots in Figure (14) show the values of ${d\bar{J}}/{d\rho}$ computed using all singular modes ($l = NK = 600$). The black dots show ${d\bar{J}}/{d\rho}$ computed using $l = 500$ (this value of l removes all singular values smaller than 1). Each dot is the average of 10 sensitivities computed using different trajectories with random initial conditions $\mathbf{u}_0$. For $l = NK$, there is a deviation with respect to the finite difference values (blue squares) of up to 3.75% (for $\rho = 26$). Note also the different trends for small values of ρ. On the other hand, the sensitivities computed using $l = 500$ follow the same trend and the error is reduced to less than 1%.

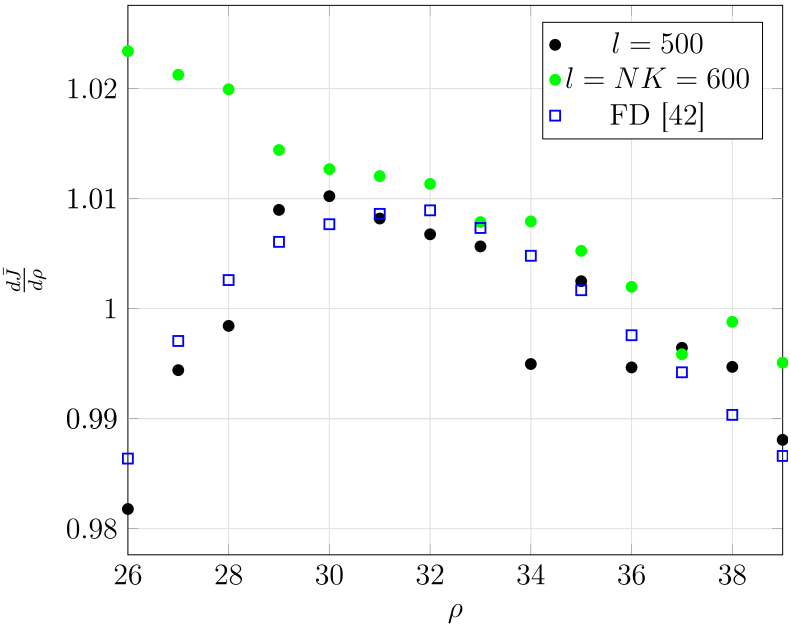


Figure 14: Sensitivity ${d\bar{J}}/{d\rho}$ against ρ for the Lorenz system. Sensitivities shown in filled dots were computed using equation (39) with two different values of l. The finite difference (FD) data points (open squares) were digitised from figure (10) of [42] and are for $T = 1000$.

7 Regularisation of the Preconditioned System

To mitigate the effect of ill-conditioning on the sensitivity ${d\bar{J}}/{ds}$ for a given system, we can either use (39) with a suitable choice of l or regularise the MSS system. Regularisation has a similar filtering effect, i.e. it damps the contribution of very small singular values. The difference is that selecting a particular value of l results in a sharp filter, while regularisation has a smoother filter kernel, as explained in [41]. Tikhonov regularisation is one of the most widely used regularisation techniques. The idea is to solve a regularised version of (16):

$$(\gamma I + S)\underline{\mathbf{w}} = \underline{\mathbf{b}} \tag{41}$$

where I is the identity matrix, and $\gamma > 0$ is an appropriately chosen parameter. It has been employed in [1] to improve the conditioning of S for Dowell's Plate equation ($N = 4$) and the KS equation ($N \geq 127$). Parameter γ shifts all $\mu(S)$ to $\mu(S) + \gamma$. A large γ however relaxes the continuity constraint (13b) and the sensitivities computed through (19) become inaccurate. If γ is chosen adequately, it can improve the accuracy and accelerate the convergence simultaneously (both for very little additional cost).

In this section, we apply Tikhonov regularization to the preconditioned system (34), to simultaneously regularize small $\mu(S)$ and to deflate large $\mu(S)$. We have two options: Regularise the original system and then apply preconditioning, i.e. solve

$$\mathbf{M_{BD}}_{(l)}^{(q)} (\gamma I + S)\, \underline{\mathbf{w}} = \mathbf{M_{BD}}_{(l)}^{(q)} \underline{\mathbf{b}} \Longrightarrow \left(\gamma \mathbf{M_{BD}}_{(l)}^{(q)} + \mathbf{M_{BD}}_{(l)}^{(q)} S\right) \underline{\mathbf{w}} = \mathbf{M_{BD}}_{(l)}^{(q)} \underline{\mathbf{b}} \tag{42}$$

or apply preconditioning first and then regularise, i.e. solve

$$\left(\gamma I + \mathbf{M_{BD}}_{(l)}^{(q)} S\right) \underline{\mathbf{w}} = \mathbf{M_{BD}}_{(l)}^{(q)} \underline{\mathbf{b}} \tag{43}$$

When the exact preconditioner $M_{(l)}$ (27) is used and γ is much smaller than the largest l retained singular values, the two options are almost identical. For $\mathbf{M_{BD}}_{(l)}^{(q)}$, the second option guarantees the clustering of eigenvalues inside the tight range, $\left[\gamma + \sigma_{min}(\mathbf{M_{BD}}_{(l)}^{(q)} S), \gamma + \sigma_{max}(\mathbf{M_{BD}}_{(l)}^{(q)} S)\right]$. Such tight clustering is conducive to rapid convergence of iterative subspace solvers. The second option physically means that we first deflate the rapidly growing modes within each segment and then slightly relax the continuity constraint between segments. This leads to smaller iterations because now the growth in each segment has been suppressed.

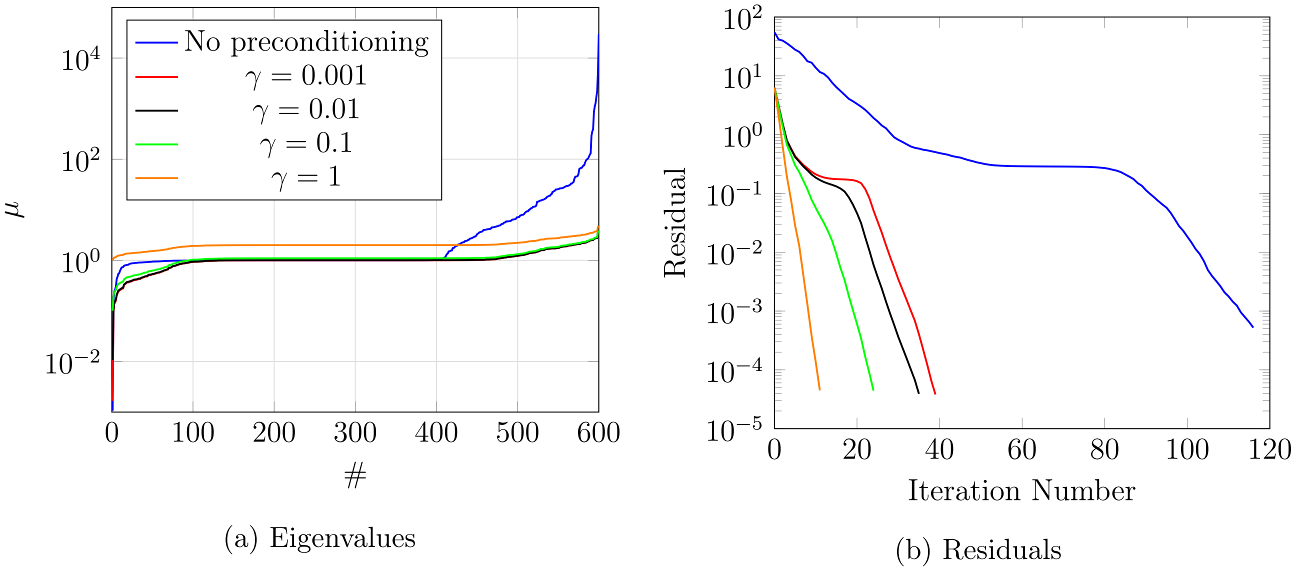


Figure 15: Eigenvalues and residuals of the original system S and the preconditioned system $\gamma I + \mathbf{M_{BD}}_{(1)} S$ for different γ. A Lorenz system trajectory length $T = 200$ with $\Delta T = 1$ was used for $\rho = 40$.

The combined effect of regularization and preconditioning for the Lorenz system can be seen in Figure (15a). Without regularization, the condition number $\kappa(S) \approx 3 \times 10^7$ while $\kappa(\mathbf{M_{BD}}_{(1)} S) \approx$

5.6×10^3, i.e. a reduction of 4 orders of magnitude in κ. When using $\gamma = 1$, $\kappa(I + \mathbf{M_{BD(1)}}S) \approx 4$, and convergence is obtained in 12 iterations only (Figure 15b).

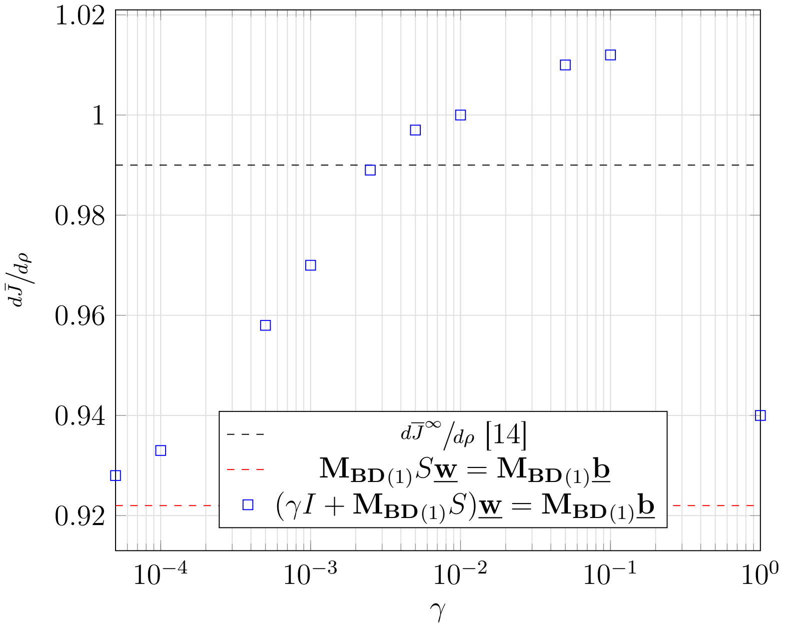


Figure 16: Sensitivities for the Lorenz system ($T = 200$, $\Delta T = 1$ and $\rho = 40$) for different γ.

The sensitivities $d\bar{J}/d\rho$ computed for different γ are shown in Figure (16). For reference, $d\bar{J}^{\infty}/d\rho$ [14] and the solution for $\gamma = 0$ (which is affected by the ill-conditioning of S) are shown. There is a range $0.001 \leq \gamma \leq 0.1$ which provides a good balance between filtering out the noisy singular values while keeping $\|\underline{\mathbf{b}} - A\underline{\mathbf{v}}\|_2$ close to zero. In this range, the solution error is $O(\leq 2\%)$. Further increase to $\gamma = 1$ relaxes the constraint $A\underline{\mathbf{v}} = \underline{\mathbf{b}}$ significantly and results in a solution error of $O(5\%)$. A method to estimate the optimal value of γ based on the L-curve criterion is proposed in [43, 44].

Ideally, the convergence rate should be independent of the number of segments K (and therefore T) and the number of degrees of freedom, N. This would make MSS applicable to large systems. Figure (17) shows that indeed the convergence rate is almost independent of T for the Lorenz system. Theoretical analysis shows that for a linear system with a symmetric, positive definite matrix $\mathscr{A}$, the $\mathscr{A}$-norm of the error at iteration m, $\|r_m\|_{\mathscr{A}}$, satisfies (refer to [24])

$$\frac{\|r_m\|_{\mathscr{A}}}{\|r_0\|_{\mathscr{A}}} \leq 2 \left(\frac{\sqrt{\kappa(\mathscr{A})} - 1}{\sqrt{\kappa(\mathscr{A})} + 1} \right)^m \tag{44}$$

This error bound is independent of the number of unknowns, and provided that $\kappa(\mathscr{A})$ is suppressed by preconditioning and regularisation, the number of iterations becomes independent of T and N. This is clearly demonstrated in Figure (17). Tests showed that using equation (44) to predict the number of iterations that will result in a pre-specified normalised residual (set to 10^{-5}) overestimated the actual iterations needed in practise. This shows that (44) indeed provides an upper bound. The bound becomes more accurate as $\kappa(\mathscr{A})$ decreases. Table (1) shows that for the chosen $\gamma = 0.1$, $d\bar{J}/d\rho \to 0.99$ (the infinite time averaged sensitivity $d\bar{J}^{\infty}/d\rho$).

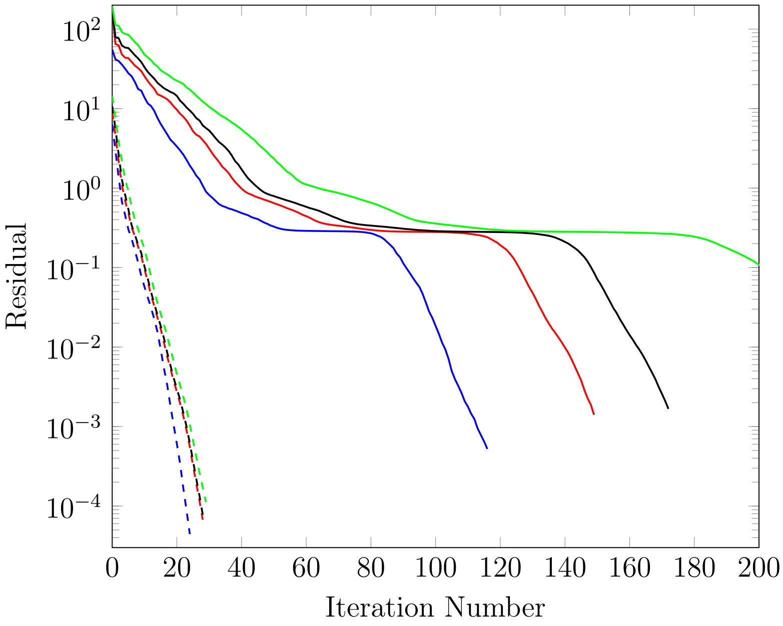


Figure 17: Residuals for S (solid lines) and the BDP system $\gamma I + \mathbf{M}_{\mathbf{BD}(1)} S$ (dashed lines) with $\gamma = 0.1$. The segment size is $\Delta T = 1$ and $\rho = 40$ (Lorenz system). Blue: $T = 200$, red: $T = 300$, black: $T = 500$, green: $T = 1000$.

	$T = 200$	$T = 300$	$T = 500$	$T = 1000$
$d\bar{J}/ds$	1.01	0.97	0.97	0.99

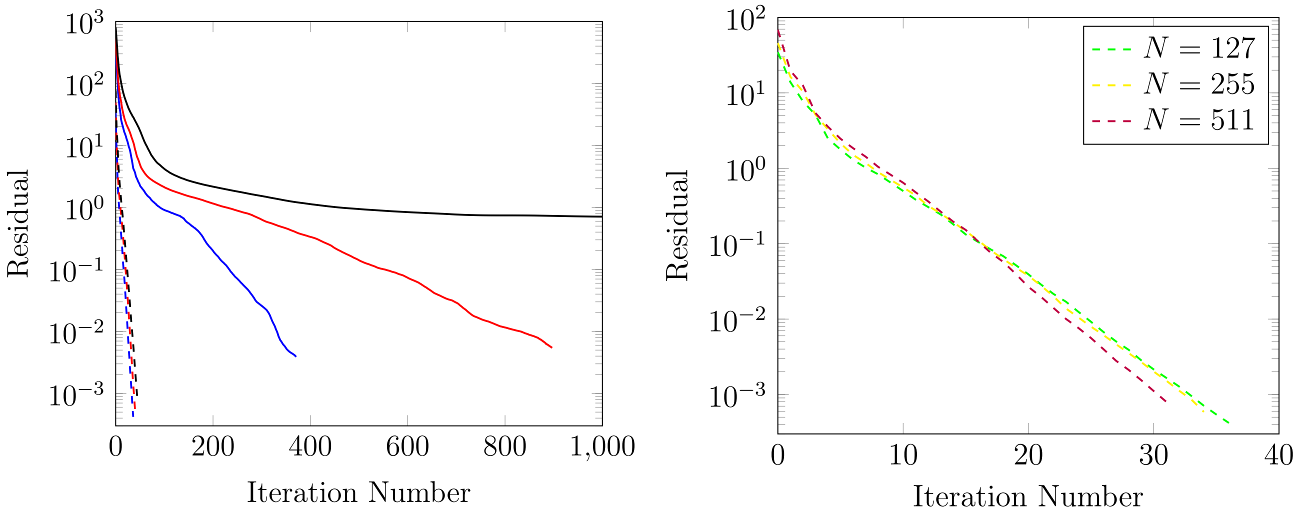
Table 1: A table showing $d\bar{J}/d\rho$ for $\rho = 40$ (Lorenz system) using different trajectory lengths. A regularization value $\gamma = 0.1$ was used.

Figure (18) shows the convergence rates for varying T (left panel) and N (right panel) for the Kuramoto Sivashinsky equation. The figure demonstrates again that the combination of regularisation and preconditioning (dashed lines) renders convergence almost independent on T (with $N = 127$) and N (with $T = 100$). The fast convergence is a direct result of the clustering of eigenvalues in a tight range and the suppression of κ.

8 Computational Cost of the Method

In this section, we consider the computational cost of the method. We report only the cost of constructing the preconditioner and of solving the Schur complement, since these operations account for most of the wall time. We use the total number of operations involving Φ_i and Φ_i^T (per segment) to quantify the cost. We assume that these operations can be done in parallel using K processors (one processor allocated to each segment). We also assume that message passing can be overlapped with computation, and that it is completed before the end of computation.

The cost (number of applications of Φ_i and Φ_i^T) of constructing one block of the preconditioner, $\mathbf{M}^{(q)}_{(l),i}$, is $2q(l+2)$, where $l+2$ denotes the selected size of the subspace. The factor 2 appears



(a) Varying T with $N = 127$

(b) Varying N with $T = 100$

Figure 18: Residuals of the original system (solid lines) and of $\gamma I + \mathbf{M_{BD}}^{(q)}_{(l)} S$ with $\gamma = 0.09$, $q = 2$ and $l = 15$ (dashed lines). The segment size is $\Delta T = 10$ for all cases. Blue: $T = 100$, red: $T = 200$, black: $T = 500$.

because the partial singular value decomposition of Φ_i requires matrix-vector products with both Φ_i and Φ_i^T. One iteration of the subspace solver for the solution of the linear system requires the application of A and A^T once, i.e. a total of $2m$ matrix-vector products per segment are required, where m is the number of iterations. This cost is only approximate, as we have ignored the time taken to orthogonalise the subspace every time a new vector is added.

Table (2) shows a cost comparison for different T (for the KS Equation) with and without preconditioning and regularisation (the values correspond to Figure 18a). We have chosen $q = 2$ and $l = 15$, so the preconditioner cost (per segment) is $2q(l + 2) = 68$ Φ_i and Φ_i^T applications. It can be seen that the combination of preconditioning and regularisation results in significant savings; the cost is reduced by a factor of 35 for $T = 500$. Note that the condition number is reduced by between 5 to 7 orders of magnitude (depending on T) and remains almost constant. The number of iterations depends very weakly on T, and the cost increases almost linearly with T (while keeping ΔT constant). The minimum and maximum eigenvalues are also reported in Table (2), and this information can be used to assess the individual effects of preconditioning and regularisation. Preconditioning results in a reduction of μ_{max} by two orders of magnitude (and a corresponding reduction in κ). Regularisation raises μ_{min} by three to five orders of magnitude.

We have not attempted to minimise the cost, but there is scope for significant reduction. For example, for $T = 500$, we have chosen $\gamma = 0.09$, which produces a value for the sensitivity $^{d\langle \bar{u}\rangle^{\infty}}/_{dc}$ accurate to 1%. Increasing to $\gamma = 0.25$ and using $q = 1$, still gives an acceptable sensitivity (the error is 8%), but the number of iterations is reduced to 35, and the total cost is 116 Φ_i and Φ_i^T applications per segment.

	$T = 100$ $(K = 10)$	$T = 200$ $(K = 20)$	$T = 500$ $(K = 50)$
Cost of solving $\left(\gamma I + \mathbf{M_{BD}}_{(l)}^{(q)} S\right) \underline{\mathbf{w}} = \mathbf{M_{BD}}_{(l)}^{(q)} \underline{\mathbf{b}}$			
μ_{max}, μ_{min}	11.87, 0.090	11.87, 0.090	13.37, 0.090
Condition number, κ	132	132	149
Number of iterations, m	38	42	46
Cost of iterations	76	84	92
Cost of constructing $\mathbf{M_{BD}}_{(l)}^{(q)}$	68	68	68
Total cost	144	152	160
Cost of solving $S\underline{\mathbf{w}} = \underline{\mathbf{b}}$			
μ_{max}, μ_{min}	3800, 1.90×10^{-4}	3800, 1.80×10^{-5}	4900, 4.90×10^{-6}
Condition number, κ	2.0×10^{7}	2.1×10^{8}	1.0×10^{9}
Number of iterations, m	371	897	2,790
Total cost	742	1,794	5,580

Table 2: A table showing the cost (total number of Φ_i and Φ_i^T applications per segment) for the cases shown in Figure (18a). The preconditioner was constructed using $q = 2$, $l = 15$ and $\gamma = 0.09$. The relative residual $\|r_m\|_2/\|r_0\|_2 \approx 1 \times 10^{-5}$ for all cases.

9 Conclusions

We proposed a block diagonal preconditioner to accelerate the convergence rate for the solution of the linear system arising from the application of the Multiple Shooting Shadowing algorithm. The preconditioner is based on the partial singular value decomposition of the diagonal blocks of the Schur complement. It was applied to the Lorenz system and the Kuramoto Sivashinsky equation.

The number of singular modes to retain in the partial SVD, l, is case dependent. A well-chosen value is required for fast convergence. If the number of positive Lyapunov exponents is known, it can be used to inform the choice of l. Strictly speaking however, this is not necessary. A self-adaptive algorithm is currently being investigated to estimate l that requires no prior knowledge of the number of positive Lyapunov exponents.

When the preconditioner was combined with a regularisation method, the condition number was significantly suppressed, and the convergence rate was found to be weakly dependent on the number of degrees of freedom and the length of the trajectory. The total number of operations was significantly reduced as a result. This paves the way to apply MSS to higher dimensional chaotic systems for sensitivity and optimal control applications. Furthermore, it was shown that a large condition number can affect the accuracy of the computed sensitivity, and can explain an 8% bias in the Kuramoto-Sivashinsky equation in the light turbulence regime. It can also explain a smaller bias for the Lorenz system.

Some open issues remain. Apart from the value of l mentioned above, the question on how to choose an appropriate value of the regularisation parameter γ that can provide an adequate balance between solution accuracy and rate of convergence is still open. Some ideas to compute the optimal γ are provided in [43, 44]. The value is case dependent, and more simulations are required to improve our understanding, especially for high-dimensional systems.

Acknowledgements

The first author wishes to acknowledge the financial support of Al-Alfi Foundation in the form of a PhD scholarship.

References

[1] Patrick J. Blonigan and Qiqi Wang. Multiple shooting shadowing for sensitivity analysis of chaotic dynamical systems. *Journal of Computational Physics*, 354:447–475, 2018.

[2] Antony Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3):233–260, 9 1988.

[3] Wei Liao and Her Mann Tsai. Aerodynamic shape optimization on overset grids using the adjoint method. *International Journal for Numerical Methods in Fluids*, 62(12):1332–1356, 1 2010.

[4] Dandan Xiao and George Papadakis. Nonlinear optimal control of bypass transition in a boundary layer flow. *Physics of Fluids*, 29(5):054103, 5 2017.

[5] Paolo Luchini and Alessandro Bottaro. Adjoint Equations in Stability Analysis. *Annual Review of Fluid Mechanics*, 46(1):493–517, 2014.

[6] Sebastian Peitz, Sina Ober-Blöbaum, and Michael Dellnitz. Multiobjective Optimal Control Methods for Fluid Flow Using Reduced Order Modeling. *arXiv:1510.05819*, pages 1–31, 10 2015.

[7] Haitao Liao. Uncertainty Quantification and Bifurcation Analysis of an Airfoil with Multiple Nonlinearities. *Mathematical Problems in Engineering*, 2013:1–12, 2013.

[8] D J Lea, M R Allen, and T W N Haine. Sensitivity analysis of the climate of a chaotic system. *Quarterly Journal of the Royal Meteorological Society*, 128(August):2587–2605, 2002.

[9] Dandan Xiao and George Papadakis. Nonlinear optimal control of transition due to a pair of vortical perturbations using a receding horizon approach. *Journal of Fluid Mechanics*, 861:524–555, 2019.

[10] J. Thuburn. Climate sensitivities via a Fokker–Planck adjoint approach. *Quarterly Journal of the Royal Meteorological Society*, 131(605):73–92, 2005.

[11] R Kubo. The fluctuation-dissipation theorem. *Reports on Progress in Physics*, 29(1):255, 1966.

[12] M. A. Khodkar and Pedram Hassanzadeh. Data-driven reduced modelling of turbulent Rayleigh–Benard convection using DMD-enhanced fluctuation–dissipation theorem. *Journal of Fluid Mechanics*, 852(May):R3, 10 2018.

[13] Davide Lasagna. Sensitivity analysis of chaotic systems using unstable periodic orbits. *SIAM Journal on Applied Dynamical Systems*, 17(1):547–580, 2018.

[14] Qiqi Wang, Rui Hu, and Patrick Blonigan. Least Squares Shadowing sensitivity analysis of chaotic limit cycle oscillations. *Journal of Computational Physics*, 267:210–224, 2014.

[15] Rufus Bowen. ω-Limit sets for Axiom A diffeomorphisms. *Journal of Differential Equations*, 18(2):333–339, 7 1975.

[16] Sergei Yu Pilyugin. *Shadowing in Dynamical Systems*, volume 1706 of *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, 1999.

[17] Philip Holmes, John L. Lumley, Gahl Berkooz, and Clarence W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, Cambridge, 2nd edition, 2012.

[18] Angxiu Ni and Qiqi Wang. Sensitivity analysis on chaotic dynamical systems by Non-Intrusive Least Squares Shadowing (NILSS). *Journal of Computational Physics*, 347:56–77, 2017.

[19] Patrick J. Blonigan. Adjoint sensitivity analysis of chaotic dynamical systems with non-intrusive least squares shadowing. *Journal of Computational Physics*, 348:803–826, 2017.

[20] Qiqi Wang, Steven A. Gomez, Patrick J. Blonigan, Alastair L. Gregory, and Elizabeth Y. Qian. Towards scalable parallel-in-time turbulent flow simulations. *Physics of Fluids*, 25(11), 2013.

[21] Patrick Blonigan, Steven Gomez, and Qiqi Wang. Least Squares Shadowing for Sensitivity Analysis of Turbulent Fluid Flows. *52nd Aerospace Sciences Meeting*, (January):1–24, 2014.

[22] William H. Press, Saul A Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes*. Cambridge University Press, New York, 3rd edition, 2007.

[23] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2nd edition, 2003.

[24] Michele Benzi, Gene H. Golub, and Jörg Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 5 2005.

[25] A. J. Wathen. Preconditioning. *Acta Numerica*, 24:329–376, 2015.

[26] Howard C. Elman, David J. Silvester, and Andrew J. Wathen. *Finite Elements and Fast Iterative Solvers : with Applications in Incompressible Fluid Dynamics*. OUP Oxford, 2nd edition, 2014.

[27] Eleanor McDonald, Jennifer Pestana, and Andy Wathen. Preconditioning and Iterative Solution of All-at-Once Systems for Evolutionary Partial Differential Equations. *SIAM Journal on Scientific Computing*, 40(2):A1012–A1033, 1 2018.

[28] Yang Cao, Mei-Qun Jiang, and Ying-Long Zheng. A splitting preconditioner for saddle point problems. *Numerical Linear Algebra with Applications*, 18(5):875–895, 10 2011.

[29] Gene H. Golub, Chen Greif, and James M. Varah. An Algebraic Analysis of a Block Diagonal Preconditioner for Saddle Point Systems. *SIAM Journal on Matrix Analysis and Applications*, 27(3):779–792, 2005.

[30] Mansoor Rezghi and S. M. Hosseini. Lanczos based preconditioner for discrete ill-posed problems. *Computing*, 88(1-2):79–96, 6 2010.

[31] G. Stewart. A Krylov–Schur algorithm for large eigenproblems. *SIAM Journal on Matrix Analysis and Applications*, 23(3):601–614, 2002.

[32] Iain Waugh, Simon Illingworth, and Matthew Juniper. Matrix-free continuation of limit cycles for bifurcation analysis of large thermoacoustic systems. *Journal of Computational Physics*, 240:225–247, 2013.

[33] Juan Sánchez and Marta Net. On the Multiple Shooting Continuation of Periodic Orbits by Newton-Krylov Methods. *International Journal of Bifurcation and Chaos*, 20(01):43–61, 1 2010.

[34] James Baglama and Lothar Reichel. Augmented Implicitly Restarted Lanczos Bidiagonalization Methods. *SIAM Journal on Scientific Computing*, 27(1):19–42, 1 2005.

[35] Jan Frøyland and Knut H. Alfsen. Lyapunov-exponent spectra for the Lorenz model. *Physical Review A*, 29(5):2928–2931, 5 1984.

[36] Christopher L. Wolfe and Roger M. Samelson. An efficient method for recovering Lyapunov vectors from singular vectors. *Tellus A*, 59(3):355–366.

[37] Patrick J. Blonigan and Qiqi Wang. Least squares shadowing sensitivity analysis of a modified Kuramoto-Sivashinsky equation. *Chaos, Solitons and Fractals*, 64(1):16–25, 2014.

[38] James M. Hyman and Basil Nicolaenko. The Kuramoto-Sivashinsky equation: A bridge between PDE'S and dynamical systems. *Physica D: Nonlinear Phenomena*, 18(1-3):113–126, 1 1986.

[39] Philip Holmes, John L. Lumley, Gahl Berkooz, and Clarence W. Rowley. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry.* Cambridge Monographs on Mechanics. Cambridge University Press, 2 edition, 2012.

[40] Shervan Erfani, Ali Tavakoli, and Davod Khojasteh Salkuyeh. An efficient method to set up a Lanczos based preconditioner for discrete ill-posed problems. *Applied Mathematical Modelling*, 37(20):8742 – 8756, 2013.

[41] Per Christian Hansen. Regularization Tools: A Matlab package for analysis and solution of discrete ill-posed problems. *Numerical Algorithms*, 6(1):1–35, 3 1994.

[42] Davide Lasagna, Ati Sharma, and Johan Meyers. Periodic Shadowing Sensitivity Analysis of Chaotic Systems. arxiv.org/abs/1806.02077. 2018.

[43] D. Calvetti, G.H. Golub, and L. Reichel. Estimation of the L-Curve via Lanczos Bidiagonalization. *BIT Numerical Mathematics*, 39(4):603–619, 1999.

[44] D. Calvetti, L. Reichel, and A. Shuibi. L-Curve and curvature bounds for Tikhonov regularization. *Numerical Algorithms*, 35(2-4):301–314, 2004.