ON THE APPLICATION OF OPTIMAL CONTROL TECHNIQUES TO THE SHADOWING APPROACH FOR TIME AVERAGED SENSITIVITY ANALYSIS OF CHAOTIC SYSTEMS

RHYS E. GILBERT, DAVIDE LASAGNA

Abstract. Traditional sensitivity analysis methods fail for chaotic systems due to the unstable characteristics of the linearised equations. To overcome these issues two methods have been developed in the literature, one being the Shadowing approach, which results in a minimisation problem, and the other being numerical viscosity, where a damping term is added to the linearised equations to suppress the instability. The Shadowing approach is computationally expensive but produces accurate sensitivities, while numerical viscosity can produce less accurate sensitivities but with significantly reduced computational cost. However, it is not fully clear how the solutions generated by these two approaches compare to each other. In this work we aim to bridge this gap by introducing a control term, found with optimal control theory techniques, to prevent the exponential growth of solution of the linearised equations. We will refer to this method as Optimal Control Shadowing. We investigate the computational aspects and performance of this new method on the Lorenz and Kuramoto-Sivashinsky systems and compare its performance with simple numerical viscosity schemes. We show that the tangent solution generated by the proposed approach is similar to that generated by shadowing methods, suggesting that optimal control attempts to stabilize the unstable shadowing direction. Further, for the spatially-extended system, we examine the energy budget of the tangent equation and show that the control term found via the solution of the optimal control problem acts only at length scales where production of tangent energy dominates dissipation, which is not necessarily the case for the numerical viscosity methods.

1. Introduction.

Derivatives of functions are an important tool for design in the engineering sector and are used in uncertainty quantification or optimisation procedures. These derivatives are commonly known as sensitivities or linear responses. In the aerospace sector, traditionally there is one function of interest and multiple parameters and as such adjoint-based sensitivity analysis methods are employed. Shape optimisation is the main role of these derivatives and has been used extensively to optimise a design given a steady state Reynolds-Averaged Navier-Stokes (RANS) simulation Refs. [48, 72, 86]. As computational resources are becoming more accessible, higher fidelity simulations are being increasingly used such as Large Eddy Simulations (LES). These higher fidelity simulations naturally are transient and, as a consequence, time averaged sensitivities should be calculated. The approach presented in Ref. [48] breaks down for time averaged sensitivities of chaotic systems. The reason for this break down is known as the 'butterfly effect' where small perturbations to the system grow exponentially in time, see Refs. [55, 85]. The computation of the linear response is currently comprised of two approaches the ensemble and derivative operator formula. The ensemble approach computes, for a given orbit, the average perturbation, Refs. [34, 55]. Derivative operators, on the other hand, estimate the Sinai–Ruelle–Bowen (SRB) measure, Ref. [35]. A method that combines the ensemble and derivative operator formulas is the blended response algorithm, Ref. [1]. The linear response can be decomposed into contributions corresponding to the shadowing and unstable parts. It is well known that the Shadowing part is not guaranteed to compute accurate sensitivities, Ref. [62]. The saving grace is that Shadowing, when the unstable dimension is low, can produce accurate sensitivity values, Ref. [63]. Computation of the unstable component is inefficient as it scales with dimension. For this reason many methods attempt to approximate the Shadowing component.

There are currently two main leading class of methods that approximate the Shadowing component of the linear response, namely Shadowing methods and Numerical Viscosity (NV) approaches. The main difference between the two is that Shadowing methods are generally computationally expensive but, in turn, can produce accurate sensitivities with no *a priori* knowledge. By contrast, NV approaches are computationally cheaper but require selecting and tuning a numerical viscosity term. Several forms of shadowing methods have been proposed since the seminal work of Ref. [83]: Least Squares Shadowing (LSS) Refs. [11, 12, 13, 16, 26, 82, 84], Multiple Shooting Shadowing (MSS) Refs. [14, 74, 75], Non-Intrusive LSS (NILSS) Refs. [10, 17, 47, 61, 64, 65, 66] and Periodic Shadowing Refs. [53, 54]. In Shadowing methods, the tangent solution defines a perturbation to the original solution of the non-linear system that remains uniformly bounded in time, and "shadows" the original solution, hence the name. LSS, MSS and NILSS solve a minimisation problem where the L_2 norm of the tangent solution is minimised through relaxation of the initial condition of the tangent equation. MSS is a reformulation of LSS that reduces the size of the optimisation problem being solved and reduces computational costs. NILSS solves one inhomogeneous tangent equation and M homogeneous tangent equations where M is larger than the number of positive Lyapunov exponents (LEs) of the system. The final tangent solution for sensitivity analysis is formed as the linear combination of these solutions that has minimum norm.

Numerical viscosity (NV) methods, Refs. [9, 18, 36, 79, 80], on the other hand, introduce an additional numerical viscosity term into the linearised equations along with a tuning parameter which can be used to stabilise the solution. As these methods solve the non-linear solution and one linearised equation they are significantly computationally cheaper than Shadowing methods. However, one question remains open: it is not necessarily clear what term should be added to the governing equations to obtain an effective stabilisation without compromising the accuracy of the resulting solution. For instance, for Navier-Stokes problems Blonigan et al. (Ref. [18]) used the Laplacian of the adjoint field as their choice of numerical viscosity term after analysing the contributions to dissipation of adjoint energy. Such an approach targets primarily the growth of unstable perturbations at small length scales, where numerical viscosity can be effective, but can fail to control large-scale instabilities if the viscosity is not large enough.

This paper proposes to bridge the conceptual gap in understanding between Shadowing methods and NV approaches. To this end, we propose a new sensitivity technique whereby a control term is introduced in the linearised equations, as in NV approaches, but its spatio-temporal structure is not constrained *a priori*. Instead, we find the control term by utilising optimal control techniques for linear time-varying systems, (see e.g. Refs. [67, 70]), whereby the control that stabilises the tangent solution and minimises its norm, as in LSS, is found using a mathematically rigorously procedure. Given the size of the problems to which this technique is targeted to, we do not solve a Riccati equation to find the optimal feedback, Refs. [7, 46, 59, 71], but use direct-adjoint looping, Ref. [69], to find the optimal control for a given finite-span non-linear trajectory. This method will be referred to as Optimal Control Shadowing (OCS). The method depends on a single parameter that expresses the cost of applying control to the linearised equations and determines its strength, analogously to the tuning parameter in other NV methods. Note that we do not aim to improve the computational efficiency of Shadowing methods, but more fundamentally we want to use OCS to develop a better understanding of what properties and structure the control term in NV methods should have. Nevertheless we describe computational aspects associated to the solution of the optimal control problem such as the overall costs, preconditioning methods to speed-up the solution of the optimality conditions and the resulting convergence rates. We also compare the solutions generated by OCS, MSS and various NV methods. Results are presented for the Lorenz equations, Ref. [56], and the Kuramoto-Sivashinsky, Refs. [50, 51, 76, 77], partial differential equations, where we analyse the impact of the spatial domain size on the cost of the algorithm.

This paper is divided as follows: Section 2 introduces optimal control techniques into the Shadowing approach and Section 3 outlines implementation details. Section 4 investigates the behaviour of OCS on the Lorenz system. Section 5 continues to explore the behaviour of OCS along with comparison of the solutions generated by OCS, MSS and NV for the Kuramoto-Sivashinsky system. Finally, conclusions are reported in Section 6.

2. Applying optimal control techniques to the shadowing approach. In this Section we set up the problem and then derive of tangent and adjoint OCS. Finally, we provide details of a method to solve OCS that decomposes the time domain into segments to leverage distributed computing architectures. Implementation details are left for Section 3.

2.1. Problem set-up. Firstly, consider a non-linear chaotic dynamical system of the form

$$\frac{\mathrm{d}\mathbf{u}(t,p)}{\mathrm{d}t} = \mathbf{f}(\mathbf{u}(t,p),p), \tag{2.1}$$

where $\mathbf{u}(t,p) \in \mathbb{R}^{n_u}$ is the state vector, $p \in \mathbb{R}$ is a parameter of interest and $\mathbf{f}(\mathbf{u}(t,p),p) : \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}^{n_u}$ are the governing equations. In general, the function of interest is $J\left(\mathbf{u}(t,p),p\right) : \mathbb{R}^{n_u} \times \mathbb{R} \to \mathbb{R}$ and its time average is

$$\bar{J} = \frac{1}{T}\int_{t_s}^{t_f} J\left(\mathbf{u}(t,p),p\right)\mathrm{d}t, \tag{2.2}$$

where t_s is the start time, t_f is the final time and $T = t_f - t_s$ is the time horizon.

2.2. Derivation of the tangent OCS formulation. We first linearise Equation (2.1) and introduce the time dilation term $\eta\mathbf{f}$, explained in Ref. [83], which results in

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\mathbf{v} + \frac{\partial \mathbf{f}}{\partial p} + \eta\mathbf{f}, \tag{2.3}$$

where $\mathbf{v} = \frac{\partial \mathbf{u}}{\partial p}$, $\frac{\partial \mathbf{f}}{\partial \mathbf{u}}$ is the Jacobian and $\frac{\partial \mathbf{f}}{\partial p}$ is a forcing term. To remove explicit calculation of the $\eta\mathbf{f}$ term, $\mathbf{v}$ is constrained to be orthogonal to $\mathbf{f}$ as in Ref. [14]. Shadowing methods such as MSS/LSS consist of introducing a minimisation problem on the norm of $\mathbf{v}$ which is achieved through the relaxation of the initial condition of Equation (2.3). Here, to remain comparable to the NV formulation, the initial condition $\mathbf{v}(t_s) = \mathbf{0}$ is used. Introduction of a generic control term, $\mathbf{q} \in \mathbb{R}^{n_q}$, into Equation (2.3), results in the tangent equation

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\mathbf{v} + \frac{\partial \mathbf{f}}{\partial p} + \eta\mathbf{f} + \mathbf{q}, \tag{2.4a}$$

$$\mathbf{v}(t_s) = \mathbf{0}. \tag{2.4b}$$

Following the control techniques presented in Refs. [3, 6, 21, 68], which are typically used to compute the structure of the control term $\mathbf{q}$, leads to the following minimisation problem

$$\min_{\mathbf{v},\mathbf{q}} \int_{t_s}^{t_f} \mathbf{v}^T\mathbf{v} + \alpha\mathbf{q}^T\mathbf{q}dt \tag{2.5a}$$

$$s.t.\ \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\mathbf{v} + \frac{\partial \mathbf{f}}{\partial p} + \eta\mathbf{f} + \mathbf{q}, \tag{2.5b}$$

$$\mathbf{v}^T\mathbf{f}\big|_t = 0, \tag{2.5c}$$

$$\mathbf{v}(t_s) = \mathbf{0}, \tag{2.5d}$$

where α determines the cost of applying control. Small values mean control is cheap to apply and the tangent solution is over-damped, while large values mean control is expensive to apply and the tangent solution is under-damped.

For the structure of $\mathbf{q}$ to be optimal, the Pontryagin minimisation principle, see Refs. [67, 70], is used. Firstly, a set of Lagrange multipliers, $\boldsymbol{\lambda} \in \mathbb{R}^{n_\lambda}$ and $\omega \in \mathbb{R}$, are introduced for the constraints (2.5b) and (2.5c), respectively. Utilising $\boldsymbol{\lambda}$ and ω means that the constraints can be incorporated to form the Hamiltonian, $\mathcal{H}$, of Equation (2.5) resulting in

$$\mathcal{H} = \int_{t_s}^{t_f} \mathbf{v}^T\mathbf{v} + \alpha\mathbf{q}^T\mathbf{q} + \boldsymbol{\lambda}^T\left(\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} - \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\mathbf{v} - \frac{\partial \mathbf{f}}{\partial p} - \eta\mathbf{f} - \mathbf{q}\right) + \omega\left(\mathbf{f}^T\mathbf{v}\right)dt. \tag{2.6}$$

Integration of Equation (2.6) by parts leads to

$$\mathcal{H} = \int_{t_s}^{t_f} \mathbf{v}^T\mathbf{v} + \alpha\mathbf{q}^T\mathbf{q} - \frac{\mathrm{d}\boldsymbol{\lambda}^T}{\mathrm{d}t}\mathbf{v} - \boldsymbol{\lambda}^T\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\mathbf{v} + \frac{\partial \mathbf{f}}{\partial p} + \eta\mathbf{f} + \mathbf{q}\right) + \omega\left(\mathbf{f}^T\mathbf{v}\right)dt + \boldsymbol{\lambda}^T\mathbf{v}\Big|_{t_s}^{t_f}. \tag{2.7}$$

Pontryagin's minimisation principle defines the minimum of Equation (2.7) to be when all partial derivatives of $\mathcal{H}$ with respect to $\mathbf{v}$, $\boldsymbol{\lambda}$, $\mathbf{q}$, ω and η are zero. This leads to the set of first order optimality conditions

$$\frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}} = \mathbf{0} = \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} - \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\mathbf{v} - \frac{\partial \mathbf{f}}{\partial p} - \eta\mathbf{f} - \mathbf{q}, \tag{2.8a}$$

$$\frac{\partial \mathcal{H}}{\partial \mathbf{v}} = \mathbf{0} = -\frac{\mathrm{d}\boldsymbol{\lambda}}{\mathrm{d}t} - \frac{\partial \mathbf{f}}{\partial \mathbf{u}}^T\boldsymbol{\lambda} + 2\mathbf{v} + \omega\mathbf{f}, \tag{2.8b}$$

$$\frac{\partial \mathcal{H}}{\partial \mathbf{q}} = \mathbf{0} = 2\alpha\mathbf{q} - \boldsymbol{\lambda}, \tag{2.8c}$$

$$\frac{\partial \mathcal{H}}{\partial \omega} = 0 = \mathbf{f}^T\mathbf{v}\big|_t, \tag{2.8d}$$

$$\frac{\partial \mathcal{H}}{\partial \eta} = 0 = \boldsymbol{\lambda}^T\mathbf{f}\big|_t. \tag{2.8e}$$

The Hamiltonian Hessian matrix is positive-definite, which means the minimisation problem, Equation (2.5a), will be convex if the second order sufficient condition

$$\frac{\partial^2 \mathcal{H}}{\partial \mathbf{q}^2} = 2\alpha \tag{2.9}$$

is greater than zero. This leads to the constraint $\alpha > 0$.

To find the optimal control we solve Equation (2.8). Equation (2.8a) is solved forwards in time from $\mathbf{v}(t_s) = \mathbf{0}$. The Lagrange multiplier values can be selected arbitrarily and, therefore, $\boldsymbol{\lambda}^T \mathbf{v}\Big|_{t_s}^{t_f}$ in Equation (2.7) is constrained to be zero. This results in $\boldsymbol{\lambda}(t_f) = \mathbf{0}$. The co-state equation, Equation (2.8b), is solved backwards in time from t_f to t_s using $\boldsymbol{\lambda}(t_f) = \mathbf{0}$ as a terminal condition. We note at this point the main difference between OCS and NV is that NV adds a control term with a certain structure, which is generic for all initial conditions, whereas OCS adds a control term which is related to the time horizon and initial condition used.

The η term remains unknown and a closed from expression is required. This can be found by defining $\mathbf{v}'(t)$ as the solution of the tangent equation without the influence of $\eta\mathbf{f}$ and after some manipulation (details provided in Appendix A), the tangent solution at time t becomes

$$\mathbf{v}(t) = \mathbf{v}'(t) - \left. \left(\frac{\mathbf{v}'^T \mathbf{f}}{\mathbf{f}^T \mathbf{f}} \mathbf{f} \right) \right|_t . \tag{2.10}$$

Similarly, ω is also an unknown and, utilising a similar approach to η, the closed form expression is required. By defining $\boldsymbol{\lambda}'(t)$ as the solution to the co-state equation without the influence of $\omega\mathbf{f}^T$ and after manipulation (see Appendix B) the co-state solution at time t becomes

$$\boldsymbol{\lambda}(t) = \boldsymbol{\lambda}'(t) - \left. \left(\frac{\boldsymbol{\lambda}'^T \mathbf{f}}{\mathbf{f}^T \mathbf{f}} \mathbf{f} \right) \right|_t . \tag{2.11}$$

Finally, the sensitivity equation, where details of the derivation are provided in Ref. [14], is

$$\frac{\mathrm{d}\bar{J}}{\mathrm{d}p} = \frac{1}{T} \int_{t_s}^{t_f} \frac{\partial J}{\partial \mathbf{u}} \mathbf{v} + \frac{\partial J}{\partial p} + \eta \left(J - \bar{J} \right) \mathrm{d}t, \tag{2.12}$$

and can be manipulated using $\mathbf{v}'(t)$ to remove η resulting in

$$\frac{\mathrm{d}\bar{J}}{\mathrm{d}p} = \frac{1}{T} \int_{t_s}^{t_f} \frac{\partial J}{\partial \mathbf{u}} \mathbf{v}' + \frac{\partial J}{\partial p} \mathrm{d}t + \left. \frac{1}{T} \frac{\mathbf{v}'^T \mathbf{f}}{\mathbf{f}^T \mathbf{f}} \left(\bar{J} - J \right) \right|_{t_f} . \tag{2.13}$$

2.3. Derivation of the adjoint OCS formulation. It is common in engineering applications that the sensitivity of the function of interest is required with respect to multiple parameters. Typically, this is achieved using the adjoint approach. To derive the adjoint OCS formulation, additional Lagrange multipliers $\hat{\boldsymbol{\lambda}} \in \mathbb{R}^{n_{\hat{\lambda}}}$, $\hat{\mathbf{v}} \in \mathbb{R}^{n_{\hat{v}}}$, $\hat{\mathbf{q}} \in \mathbb{R}^{n_{\hat{q}}}$, $\hat{\eta} \in \mathbb{R}$ and $\hat{\omega} \in \mathbb{R}$, one for each optimality condition in Equation (2.8), are introduced. These, again, allow the incorporation of the optimality conditions, Equation (2.8), into Equation (2.12) which results in

$$\begin{aligned} \frac{\mathrm{d}\bar{J}}{\mathrm{d}p} = \int_{t_s}^{t_f} & \frac{1}{T} \frac{\partial J}{\partial \mathbf{u}} \mathbf{v} + \frac{1}{T} \eta \left(J - \bar{J} \right) + \frac{1}{T} \frac{\partial J}{\partial p} \\ & + \hat{\boldsymbol{\lambda}}^T \left(\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} - \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \mathbf{v} - \frac{\partial \mathbf{f}}{\partial p} - \eta \mathbf{f} - \mathbf{q} \right) \\ & + \left(-\frac{\mathrm{d}\boldsymbol{\lambda}^T}{\mathrm{d}t} - \boldsymbol{\lambda}^T \frac{\partial \mathbf{f}}{\partial \mathbf{u}} + 2\mathbf{v}^T + \omega \mathbf{f}^T \right) \hat{\mathbf{v}} \\ & + \left(2\alpha \mathbf{q}^T - \boldsymbol{\lambda}^T \right) \hat{\mathbf{q}} \\ & + \hat{\omega} \left(\mathbf{f}^T \mathbf{v} \right) + \hat{\eta} \left(-\mathbf{f}^T \boldsymbol{\lambda} \right) dt. \end{aligned} \tag{2.14}$$

It is worth noting that the incorporation of the optimality conditions does not modify the sensitivity in any way as all terms added are zero. Integration by parts of Equation (2.14) and grouping terms in $\mathbf{v}$, $\mathbf{q}$, $\boldsymbol{\lambda}^T$, ω and η leads to

$$
\begin{aligned}
\frac{\mathrm{d}\bar{J}}{\mathrm{d}p} = \int_{t_s}^{t_f} & \frac{1}{T}\frac{\partial J}{\partial p} - \hat{\boldsymbol{\lambda}}^T\frac{\partial \mathbf{f}}{\partial p} + \\
& \left(\frac{1}{T}\frac{\partial J}{\partial \mathbf{u}} - \frac{\mathrm{d}\hat{\boldsymbol{\lambda}}^T}{\mathrm{d}t} - \hat{\boldsymbol{\lambda}}^T\frac{\partial \mathbf{f}}{\partial \mathbf{u}} + 2\hat{\mathbf{v}}^T + \hat{\omega}\mathbf{f}^T\right)\mathbf{v} + \\
& \boldsymbol{\lambda}^T\left(\frac{\mathrm{d}\hat{\mathbf{v}}}{\mathrm{d}t} - \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\hat{\mathbf{v}} - \hat{\eta}\mathbf{f} - \hat{\mathbf{q}}\right) + \\
& \left(2\alpha\hat{\mathbf{q}}^T - \hat{\boldsymbol{\lambda}}^T\right)\mathbf{q} + \\
& \left(\frac{1}{T}\left(J - \bar{J}\right) - \hat{\boldsymbol{\lambda}}^T\mathbf{f}\right)\eta + \left(\hat{\mathbf{v}}^T\mathbf{f}\right)\omega dt + \\
& \hat{\boldsymbol{\lambda}}^T\mathbf{v}\Big|_{t_s}^{t_f} + \boldsymbol{\lambda}^T\hat{\mathbf{v}}\Big|_{t_s}^{t_f}.
\end{aligned}
\tag{2.15}
$$

By selecting the terms inside the brackets of Equation (2.15) to be zero, which removes the explicit calculation of $\mathbf{v}$, $\mathbf{q}$, $\boldsymbol{\lambda}^T$, ω and η, leads to the following adjoint optimality conditions

$$\mathbf{0} = \frac{\mathrm{d}\hat{\mathbf{v}}}{\mathrm{d}t} - \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\hat{\mathbf{v}} - \hat{\eta}\mathbf{f} - \hat{\mathbf{q}}, \tag{2.16a}$$

$$\mathbf{0} = \frac{1}{T}\frac{\partial J}{\partial \mathbf{u}} - \frac{\mathrm{d}\hat{\boldsymbol{\lambda}}^T}{\mathrm{d}t} - \hat{\boldsymbol{\lambda}}^T\frac{\partial \mathbf{f}}{\partial \mathbf{u}} + 2\hat{\mathbf{v}}^T + \hat{\omega}\mathbf{f}^T, \tag{2.16b}$$

$$\mathbf{0} = 2\alpha\hat{\mathbf{q}} - \hat{\boldsymbol{\lambda}}, \tag{2.16c}$$

$$0 = \hat{\mathbf{v}}^T\mathbf{f}\Big|_t, \tag{2.16d}$$

$$0 = \frac{1}{T}\left(J(t) - \bar{J}\right) - \hat{\boldsymbol{\lambda}}^T\mathbf{f}\Big|_t. \tag{2.16e}$$

There is still a reliance of $\mathbf{v}$ and $\boldsymbol{\lambda}^T$ in the terms $\hat{\boldsymbol{\lambda}}^T\mathbf{v}\Big|_{t_s}^{t_f}$ and $\boldsymbol{\lambda}^T\hat{\mathbf{v}}\Big|_{t_s}^{t_f}$ from Equation (2.15). The selection of the Lagrange multipliers is arbitrary and we thus set $\boldsymbol{\lambda}^T\hat{\mathbf{v}}\Big|_{t_s}^{t_f} = 0$ which results in $\hat{\mathbf{v}}(t_s) = \mathbf{0}$. In the adjoint formulation there still is a forward and backward equation which we will refer to as the tangent and co-state equations but denote the difference between those derived in Section 2.2 with a hat, $\hat{\square}$. Here, the tangent equation, Equation (2.16a), is solved forwards in time from t_s to t_f from this initial condition. Similarly, setting $\hat{\boldsymbol{\lambda}}^T\mathbf{v}\Big|_{t_s}^{t_f} = 0$ leads to $\hat{\boldsymbol{\lambda}}(t_f) = \frac{1}{T}\frac{J-\bar{J}}{\mathbf{f}^T\mathbf{f}}\mathbf{f}\Big|_{t_f}$ and the co-state equation, Equation (2.16b), is solved backwards in time from t_f to t_s from this terminal condition.

Deriving the closed form expression for $\hat{\eta}$ term is achieved in a similar manner to the tangent formulation which results in the adjoint tangent solution at time t being

$$\hat{\mathbf{v}}(t) = \hat{\mathbf{v}}'(t) - \left(\frac{\hat{\mathbf{v}}'^T\mathbf{f}}{\mathbf{f}^T\mathbf{f}}\mathbf{f}\right)\Bigg|_t. \tag{2.17}$$

The full derivation of Equation (2.17) can be found in Appendix C. Similarly, the closed form expression for $\hat{\omega}$ in derived in a similar manner as the tangent formulation and the adjoint co-state solution at time t is

$$\hat{\boldsymbol{\lambda}}(t) = \hat{\boldsymbol{\lambda}}'(t) - \frac{\hat{\boldsymbol{\lambda}}'^T\mathbf{f}}{\mathbf{f}^T\mathbf{f}}\mathbf{f}\Bigg|_t + \frac{1}{T}\frac{J - \overline{J}}{\mathbf{f}^T\mathbf{f}}\mathbf{f}\Bigg|_t. \tag{2.18}$$

The full derivation of Equation (2.18) can be found in Appendix D. Finally, the remaining non-zero terms from Equation (2.15) lead to the adjoint sensitivity equation

$$\frac{\mathrm{d}\bar{J}}{\mathrm{d}p} = \int_{t_s}^{t_f} \frac{1}{T}\frac{\partial J}{\partial p} - \hat{\boldsymbol{\lambda}}^T\frac{\partial \mathbf{f}}{\partial p}\mathrm{d}t. \tag{2.19}$$

Once $\hat{\boldsymbol{\lambda}}$ has been found it is simple to modify Equation (2.19) for multiple parameters.

2.4. Time domain decomposition. A method for splitting the time horizon into segments is presented in Refs. [28, 29, 31, 42, 49]. The rationale as to why the time horizon is split into segments is that for large systems the method outlined in Section 2.2 may require more memory than the compute node has. Further to this, the splitting of the time horizon into segments acts to condition the system. A solution of the tangent equation still exhibits exponential growth but is limited to the growth in one segment. The growth is reduced due to the smaller segment time resulting in better conditioning. Therefore, the time domain is split into segments so that the memory requirements are reduced and each segment can be solved on a separate compute node. The splitting is derived for the tangent OCS formulation, Section 2.2. However, this is easily modified for the equations derived for the adjoint OCS formulation, Section 2.3. Various alternatives to Refs. [28, 29, 31, 42, 49] are available such as single shooting formulation Ref [20], multiple shooting for the direct solution Ref. [19], multiple shooting utilising gradient descent Ref. [2] and receding horizon optimal control Ref. [39].

The time horizon, (t_s, t_f), is split into N equal segments where each has a local time span, (t_j, t_{j+1}), where $j = 0, 1, \ldots, N-1$ and $t_s = t_0 < t_1 < \ldots < t_{N-1} < t_N = t_f$. Next, a locally defined tangent solution $\mathbf{v}_j(t) \in \mathbb{R}^{n_v}$ and control $\mathbf{q}_j(t) \in \mathbb{R}^{n_q}$ on each segment j are introduced. For this splitting to be identical to the method presented in Section 2.2, continuity in both control and tangent solutions between consecutive segments is required. This results in the following constraints $\mathbf{v}_{j-1}(t_j) = \mathbf{v}_j(t_j)$ for $j = 1, \ldots, N-1$ and $\mathbf{q}_{j-1}(t_j) = \mathbf{q}_j(t_j)$ for $j = 1, \ldots, N-1$. A graphical representation of this can be seen in Figure 1. These

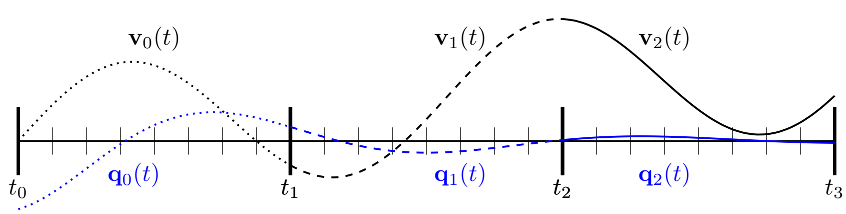


Fig. 1: Example time domain decomposition where the dotted lines represent the solution in segment 0, dashed in segment 1 and solid segment 2. Black solutions represent the tangent solution and blue the control.

conditions leads to the following minimisation problem

$$\min_{\mathbf{v}_j, \mathbf{q}_j} \sum_{j=0}^{N-1} \int_{t_j}^{t_{j+1}} \mathbf{v}_j{}^T \mathbf{v}_j + \alpha \mathbf{q}_j{}^T \mathbf{q}_j dt \tag{2.20a}$$

$$s.t. \ \frac{\mathrm{d}\mathbf{v}_j}{\mathrm{d}t} = \frac{\partial \mathbf{f}}{\partial \mathbf{u}} \mathbf{v}_j + \frac{\partial \mathbf{f}}{\partial p} + \eta_j \mathbf{f} + \mathbf{q}_j \quad j = 0, 1, \ldots N-1, \tag{2.20b}$$

$$\mathbf{v}_j^T \mathbf{f}\big|_t = 0 \quad j = 0, 1, \ldots N-1, \tag{2.20c}$$

$$\mathbf{v}(t_s) = \mathbf{0}, \tag{2.20d}$$

$$\mathbf{v}_{j-1}(t_j) = \mathbf{v}_j(t_j) \quad j = 1, \ldots, N-1, \tag{2.20e}$$

$$\mathbf{q}_{j-1}(t_j) = \mathbf{q}_j(t_j) \quad j = 1, \ldots, N-1. \tag{2.20f}$$

The Hamiltonian, $\mathcal{H}$, for this minimisation problem is formed through the introduction of locally defined Lagrange multipliers $\boldsymbol{\lambda}_j \in \mathbb{R}^{n_\lambda}$ for $j = 0, 1, \ldots, N-1$ for the tangent equation, $\omega_j \in \mathbb{R}$ for $j = 0, 1, \ldots, N-1$ for the tangent orthogonality constraint, $\Psi_j \in \mathbb{R}^{n_\Psi}$ for $j = 1, \ldots, N-1$ for the tangent continuity constraints and $\Phi_j \in \mathbb{R}^{n_\Phi}$ for $j = 1, \ldots, N-1$ for the control continuity constraint. For optimality the derivatives of the Hamiltonian with respect to $\mathbf{v}_j$, $\boldsymbol{\lambda}_j$, $\mathbf{q}_j$, η_j, ω_j, Φ_j and Ψ_j, using Pontryagin's minimisation principle, are zero. This leads to the set of first order optimality conditions

$$\frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}_j} = \mathbf{0} = \frac{\mathrm{d}\mathbf{v}_j}{\mathrm{d}t} - \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\mathbf{v}_j - \frac{\partial \mathbf{f}}{\partial p} - \eta_j \mathbf{f} - \mathbf{q}_j \quad j = 0, 1, \dots N-1, \tag{2.21a}$$

$$\frac{\partial \mathcal{H}}{\partial \mathbf{v}_j} = \mathbf{0} = -\frac{\mathrm{d}\boldsymbol{\lambda}_j}{\mathrm{d}t} - \frac{\partial \mathbf{f}}{\partial \mathbf{u}}^T \boldsymbol{\lambda}_j + 2\mathbf{v}_j + \omega_j \mathbf{f} \quad j = 0, 1, \dots N-1, \tag{2.21b}$$

$$\frac{\partial \mathcal{H}}{\partial \mathbf{q}_j} = \mathbf{0} = 2\alpha \mathbf{q}_j - \boldsymbol{\lambda}_j \quad j = 0, 1, \dots N-1, \tag{2.21c}$$

$$\frac{\partial \mathcal{H}}{\partial \omega_j} = 0 = \mathbf{f}^T \mathbf{v}_j \Big|_t \quad j = 0, 1, \dots N-1, \tag{2.21d}$$

$$\frac{\partial \mathcal{H}}{\partial \eta_j} = 0 = \boldsymbol{\lambda}_j^T \mathbf{f} \Big|_t \quad j = 0, 1, \dots N-1, \tag{2.21e}$$

$$\frac{\partial \mathcal{H}}{\partial \Psi_j} = \mathbf{0} = \mathbf{v}_{j-1}(t_j) - \mathbf{v}_j(t_j) \quad j = 1, \dots N-1, \tag{2.21f}$$

$$\frac{\partial \mathcal{H}}{\partial \Phi_j} = \mathbf{0} = \mathbf{q}_{j-1}(t_j) - \mathbf{q}_j(t_j) \quad j = 1, \dots N-1. \tag{2.21g}$$

The control continuity constraint, Equation (2.21g), can be cast onto the co-state, $\boldsymbol{\lambda}_j$, through the use of the control equation, Equation (2.21c), resulting in

$$\mathbf{0} = \boldsymbol{\lambda}_{j-1}(t_j) - \boldsymbol{\lambda}_j(t_j) \quad j = 1, \dots N-1. \tag{2.22}$$

The initial condition for the tangent equation in segment 0 is $\mathbf{v}(t_s) = \mathbf{0}$ and the terminal condition for the co-state equation in segment $N-1$ is $\boldsymbol{\lambda}(t_f) = \mathbf{0}$. These are used to ensure this method is consistent with that presented in Section 2.2. Again, removing the explicit computation of η_j results in

$$\mathbf{v}_j(t) = \mathbf{v}'_j(t) - \left(\frac{\mathbf{v}'^T_j \mathbf{f}}{\mathbf{f}^T \mathbf{f}} \mathbf{f} \right)\Bigg|_t . \tag{2.23}$$

Similarly, removal of the explicit computation of ω_j results in

$$\boldsymbol{\lambda}_j(t) = \boldsymbol{\lambda}'_j(t) - \left(\frac{\boldsymbol{\lambda}'^T_j \mathbf{f}}{\mathbf{f}^T \mathbf{f}} \mathbf{f} \right)\Bigg|_t . \tag{2.24}$$

By performing this decomposition, the sensitivity equation, (2.12), becomes

$$\frac{\mathrm{d}\bar{J}}{\mathrm{d}p} = \frac{1}{T} \sum_{j=0}^{N-1} \int_{t_j}^{t_{j+1}} \frac{\partial J}{\partial \mathbf{u}} \mathbf{v}'_j + \frac{\partial J}{\partial p} \mathrm{d}t + \frac{1}{T} \sum_{j=0}^{N-1} \frac{\mathbf{v}'^T \mathbf{f}}{\mathbf{f}^T \mathbf{f}} \left(\bar{J} - J \right)\Bigg|_{t_{j+1}} , \tag{2.25}$$

where details of this derivation can be found in Ref. [14].

3. Computational and implementation aspects. In Section 3.1 we provide a matrix-free method for solving the tangent optimality conditions, Equation (2.8). This approach is equally applicable to the optimality conditions generated for the adjoint OCS formulation, Equation (2.16). In Section 3.2, we provide a matrix-free method for the computation of the time domain decomposition optimality conditions, Equation (2.21), along with various preconditioning methods. We build on previous work undertaken in this area for optimal control of problems governed by stable advection-diffusion equations, Refs. [28, 29, 42] and extend the discussion to the performance of these methods for unstable systems such as the present ones, introducing additional requirements on the length of the segments and discussing the impact that the growth of the solution in the time horizon has on convergence of these methods.

3.1. A matrix-free method for the solution to the optimality conditions. A common method, Refs. [7, 46, 59, 71], for generating a solution that satisfies the optimality conditions, Equation (2.8), is the differential Riccati equation (DRE). The DRE generates a differential equation for a matrix, $n_q \times n_q$ in size, for the relationship between $\mathbf{v}(t)$ and $\mathbf{q}(t)$. This DRE becomes unfeasible for systems with a large number of degrees of freedom, such as unsteady turbulent flow. An iterative approach to the solution of the optimality conditions in a matrix-free sense is outlined in Ref. [69]. There are alternatives to the method proposed in Ref. [69] such as the reduced space method, Ref. [60], boundary optimal control, Ref. [52], various types of preconditioning, Refs. [4, 73], multi-grid in time, Ref. [40] and para-real/PFASST, Refs. [37, 38, 57, 58, 81]. We keep the notation general for reasons that will become clear when undertaking the time domain decomposition technique.

The method used here iterates the control term until all optimality conditions are satisfied. Initially, there is no knowledge of what the control term should be and, therefore, it is common to arbitrarily select $\mathbf{q}(t) = \mathbf{0}$. The tangent solution is found by solving Equation (2.8a), from t_s to t_f from the initial condition $\mathbf{v}(t_s) = \mathbf{0}$, the current iteration of $\mathbf{q}(t)$ along with using Equation (2.10). By finding the tangent solution in this way ensures that Equation (2.8a) and Equation (2.8d) of the optimality conditions are always satisfied. Similarly, the co-state solution is found by solving Equation (2.8b) backwards in time from t_f to t_s with $\boldsymbol{\lambda}(t_f) = \mathbf{0}$ as the terminal condition along with using the tangent solution, $\mathbf{v}(t)$, and Equation (2.11). Finding the co-state solution in this way ensures that Equation (2.8b) and Equation (2.8e) of the optimality conditions are always satisfied.

The only remaining unsatisfied optimality constraint is $\frac{\partial \mathcal{H}}{\partial \mathbf{q}} = \mathbf{0}$ because intermediate values for the control are not guaranteed to satisfy this equation. The approach taken here is to solve this unsatisfied constraint iteratively through finding a value of $\mathbf{q}$ that ensures this equation is satisfied. We first derive the linear system of the optimality constraint. This is achieved through taking the analytical solutions of the tangent and co-state equations. The tangent solution at time τ for some arbitrary control, given in terms of its state transition matrix, $\phi(t_1, t_2)$ (details of which can be found in Appendix F) is

$$\mathbf{v}(\tau) = \boldsymbol{A}\mathbf{v}(t_s) + \left(\int_{t_s}^{\tau} \eta(s)\mathrm{d}s\right)\mathbf{f}(\tau) + \boldsymbol{B}\frac{\partial \mathbf{f}}{\partial p} + \boldsymbol{B}\mathbf{q}, \tag{3.1}$$

where the linear operators $\boldsymbol{A}$ and $\boldsymbol{B}$ are defined by

$$\boldsymbol{A}\mathbf{v}(t_s) = \phi(t_s, \tau)\mathbf{v}(t_s), \tag{3.2}$$

and

$$\boldsymbol{B}\square = \int_{t_s}^{\tau} \phi(s, \tau)\square(s)\mathrm{d}s, \tag{3.3}$$

where $\square(s)$ is the value of $\square$ evaluated at time s. Similarly, the analytical form of the co-state solution at time t, written in terms of its state transition matrix, $\phi^*(t_1, t_2)$, is

$$\boldsymbol{\lambda}(t) = -2\mathbf{C}\mathbf{v}(t) + \mathbf{D}\boldsymbol{\lambda}(t_f) + \left(\int_{t}^{t_f} \omega(\tau)\mathrm{d}\tau\right)\mathbf{f}(t), \tag{3.4}$$

where the linear operators $\boldsymbol{C}$ and $\boldsymbol{D}$ are defined by

$$\mathbf{C}\mathbf{v}(t) = \int_{t}^{t_f} \phi^*\left(t, \tau\right)^{-1}\mathbf{v}(\tau)\mathrm{d}\tau, \tag{3.5}$$

and

$$\mathbf{D}\boldsymbol{\lambda}(t_f) = \phi^*\left(t, t_f\right)^{-1}\boldsymbol{\lambda}(t_f). \tag{3.6}$$

Substitution of Equation (3.1) into Equation (3.4) leads to

$$\boldsymbol{\lambda}(t) = -2\mathbf{C}\left[\boldsymbol{A}\mathbf{v}(t_s) + \left(\int_{t_s}^{t} \eta(s)\mathrm{d}s\right)\mathbf{f}(t) + \boldsymbol{B}\frac{\partial \mathbf{f}}{\partial p} + \boldsymbol{B}\mathbf{q}\right] + \mathbf{D}\boldsymbol{\lambda}(t_f) + \left(\int_{t}^{t_f} \omega(\tau)\mathrm{d}\tau\right)\mathbf{f}(t). \tag{3.7}$$

Substitution of Equation (3.7) into Equation (2.8c) and grouping term in $\mathbf{q}$ gives

$$\frac{\partial \mathcal{H}}{\partial \mathbf{q}} = \left[2\alpha\mathbf{I} + 2\mathbf{C}\mathbf{B}\right]\mathbf{q} + 2\mathbf{C}\left[\mathbf{A}\mathbf{v}(t_s) + \left(\int_{t_s}^{t} \eta(s)\mathrm{d}s\right)\mathbf{f}(t) + \mathbf{B}\frac{\partial \mathbf{f}}{\partial p}\right] - \left(\int_{t}^{t_f} \omega(\tau)\mathrm{d}\tau\right)\mathbf{f}(t) - \mathbf{D}\boldsymbol{\lambda}(t_f) = \mathbf{0}, \tag{3.8}$$

where $\mathbf{I}$ is the identity. This can be recast as a linear system of the form

$$\frac{\partial \mathcal{H}}{\partial \mathbf{q}} = \mathbf{G}(\mathbf{q}) = \mathbf{E}\mathbf{q} - \mathbf{b} = \mathbf{0},$$

where $\mathbf{G}(\mathbf{q})$ is an affine operator that computes the gradient of the Hamiltonian given a certain value of $\mathbf{q}$. The operator $\mathbf{E} = [2\alpha\mathbf{I} + 2\mathbf{C}\mathbf{B}]$ is the impact that the control has on the solution, and the known term

$$\mathbf{b} = -2\mathbf{C}\left[\mathbf{A}\mathbf{v}(t_s) + \left(\int_{t_s}^{t} \eta(s)\mathrm{d}s\right)\mathbf{f}(t) + \mathbf{B}\frac{\partial \mathbf{f}}{\partial p}\right] - \left(\int_{t}^{t_f} \omega(\tau)\mathrm{d}\tau\right)\mathbf{f}(t) + \mathbf{D}\boldsymbol{\lambda}(t_f)$$

represents the growth of the solution without control. As mentioned in Ref. [69], $\mathbf{E}$ is linear and positive definite meaning the system can be solved using a conjugate gradient method. Algorithm 3.1 shows how to compute $\mathbf{G}(\mathbf{q})$ in a matrix-free sense. The same algorithm can also be utilised to compute the know term $\mathbf{b}$,

Algorithm 3.1 A matrix-free method for the calculation of $\mathbf{G}(\mathbf{q})$.
Input: $\mathbf{q}$
Output: $\mathbf{G}(\mathbf{q})$
$\mathbf{v} \leftarrow$ Solve Equation (2.8a) from t_s to t_f using $\mathbf{v}(t_s)$ and $\mathbf{q}$
$\boldsymbol{\lambda} \leftarrow$ Solve Equation (2.8b) from t_f to t_s using $\boldsymbol{\lambda}(t_f)$ and $\mathbf{v}$
$\mathbf{G}(\mathbf{q}) \leftarrow 2\alpha\mathbf{q} - \boldsymbol{\lambda}$

which is achieved by computing $-\mathbf{G}(\mathbf{0})$. Once $\mathbf{b}$ is found, computation the action of the operator $\mathbf{E}$ is done by evaluating $\mathbf{G}(\mathbf{q})$ and adding $\mathbf{b}$. Finally, the optimal control, tangent and co-state solutions are found using Algorithm 3.2.

Algorithm 3.2 A matrix-free method for solution of the optimality conditions problem.
Set $\mathbf{b} = -\mathbf{G}(\mathbf{0})$
Solve $\mathbf{E}\mathbf{q} = \mathbf{b}$ iteratively using conjugate gradient to compute $\mathbf{q}$, using Algorithm 3.1 to compute $\mathbf{G}(\mathbf{q})$ when $\mathbf{E}\mathbf{q}$ is evaluated
$\mathbf{v} \leftarrow$ Solve Equation (2.8a) from t_s to t_f using $\mathbf{v}(t_s)$ and $\mathbf{q}$
$\boldsymbol{\lambda} \leftarrow$ Solve Equation (2.8b) from t_f to t_s using $\boldsymbol{\lambda}(t_f)$ and $\mathbf{v}$

It is worth noting that the operator $\mathbf{E}$ is never actually formed in practice and only its action on vectors is computed. This involves an evaluation of Algorithm 3.1 which requires the storage of $\mathbf{v}$, $\boldsymbol{\lambda}$ and $\mathbf{q}$ at each time step in the time horizon. Storage requirements scale with problem size, time horizon and time step.

It is well known that the convergence rate of the conjugate gradient algorithm is determined by the condition number of $\mathbf{E}$. The value chosen for α, through the $2\alpha\mathbf{I}$ term, controls the condition number of the operator $\mathbf{E}$ which is comprised of two terms, $2\alpha\mathbf{I}$ and $2\mathbf{C}\mathbf{B}$. These terms can be thought of as the cost in applying control and how well the control stabilises the solution, respectively. If $\alpha \approx 0$, then the $2\mathbf{C}\mathbf{B}$ term dominates the condition number and as such a control is applied such that

$$\mathbf{q} \approx \frac{1}{2}[\mathbf{C}\mathbf{B}]^{-1}\left(-2\mathbf{C}\left[\mathbf{A}\mathbf{v}(t_s) + \left(\int_{t_s}^{\tau} \eta(s)\mathrm{d}s\right)\mathbf{f}(\tau) + \mathbf{B}\frac{\partial \mathbf{f}}{\partial p}\right] + \left(\int_{t}^{t_f} \omega(\tau)\mathrm{d}\tau\right)\mathbf{f}(t) + \mathbf{D}\boldsymbol{\lambda}(t_f)\right), \tag{3.9}$$

which implies that the control applied is exactly the negative of the tangent equation and leads to a tangent solution that is over-damped. It is worth noting that in the limit case where $\alpha \to 0$, the norm of the resulting tangent solution $\mathbf{v}(t)$ tends to zero. Conversely, if α is large then $2\alpha\mathbf{I}$ dominates the conditioning of $\mathbf{E}$ and the control applied results in

$$\mathbf{q} \approx \frac{1}{2\alpha}\left(-2\mathbf{C}\left[\mathbf{A}\mathbf{v}(t_s) + \left(\int_{t_s}^{\tau} \eta(s)\mathrm{d}s\right)\mathbf{f}(\tau) + \mathbf{B}\frac{\partial \mathbf{f}}{\partial p}\right] + \left(\int_{t}^{t_f} \omega(\tau)\mathrm{d}\tau\right)\mathbf{f}(t) + \mathbf{D}\boldsymbol{\lambda}(t_f)\right), \tag{3.10}$$

which implies the control applied is negligible and the solution is under-damped. Therefore, selecting α in between these values leads to an adequately controlled solution.

3.2. A matrix-free method for time domain decomposition. To solve the optimality constraints, Equation (2.21) we follow the iterative approach set out in Ref. [29]. Similar in manner to the approach set out in section 3.1, we iterate the tangent and co-state interface values, $\mathbf{v}_j(t_j)$ and $\boldsymbol{\lambda}_{j-1}(t_j)$ for $j = 1, \ldots, N-1$ respectively, and control term, $\mathbf{q}(t_j)$ for $j = 0, \ldots, N-1$, until the optimality conditions are satisfied. We select the initial values of $\mathbf{v}_j(t_j) = \mathbf{0}$ and $\boldsymbol{\lambda}_{j-1}(t_j) = \mathbf{0}$ for $j = 1, \ldots, N-1$. Further, $\mathbf{q}_j(t) = \mathbf{0}$ for $j = 0, 1, \ldots, N-1$ is used. The local tangent solution is found by marching Equation (2.21a) forwards in time from t_j to t_{j+1} using the initial condition $\mathbf{v}_j(t_j)$ for $j = 0, 1, \ldots, N-1$, the control $\mathbf{q}_j$ and making use of Equation (2.23) to remove the explicit computation of η_j. Using this approach ensures that the time domain decomposition optimality constraints Equation (2.21a) and Equation (2.21d) are satisfied. The same procedure is used for the co-state solution by solving Equation (2.21b) backwards in time from t_{j+1} to t_j using the terminal condition $\boldsymbol{\lambda}_j(t_{j+1})$ for $j = 0, 1, \ldots, N-1$, the tangent solution $\mathbf{v}_j$ and Equation (2.24) to remove the explicit calculation of ω_j. Again, this approach ensures that the constraints Equation (2.21b) and Equation (2.21e) are satisfied. Therefore, the only constraints not satisfied is the control equation, Equation (2.21c), the continuity constraint on the tangent solution, Equation (2.21f), and the continuity on the control solution, Equation (2.21g). In practice, substitution of Equation (2.21g) for Equation (2.22) is undertaken. With this approach the unknowns are gathered together as

$$\mathbf{x} = \left(\mathbf{v}_1(t_1), \mathbf{q}_0, \ldots, \mathbf{v}_{j+1}(t_{j+1}), \mathbf{q}_j, \boldsymbol{\lambda}_{j-1}(t_j), \ldots, \mathbf{q}_{N-1}, \boldsymbol{\lambda}_{N-2}(t_{N-1})\right)^T.$$

Making use of the analytical form of the tangent and co-state solutions leads to

$$\begin{pmatrix} \frac{\partial \mathcal{L}}{\partial \Psi_0} \\ \frac{\partial \mathcal{L}}{\partial \mathbf{q}_0} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial \Psi_j} \\ \frac{\partial \mathcal{L}}{\partial \mathbf{q}_j} \\ \frac{\partial \mathcal{L}}{\partial \Phi_j} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial \mathbf{q}_{N-1}} \\ \frac{\partial \mathcal{L}}{\partial \Phi_{N-1}} \end{pmatrix} = \mathbf{H}\mathbf{x} = \begin{pmatrix} \mathbf{v}_1(t_1) - \mathbf{v}_0(t_1) \\ 2\alpha \mathbf{q}_0 - \boldsymbol{\lambda}_0 \\ \vdots \\ \mathbf{v}_{j+1}(t_{j+1}) - \mathbf{v}_j(t_{j+1}) \\ 2\alpha \mathbf{q}_j - \boldsymbol{\lambda}_j \\ \boldsymbol{\lambda}_{j-1}(t_j) - \boldsymbol{\lambda}_j(t_j) \\ \vdots \\ 2\alpha \mathbf{q}_{N-1} - \boldsymbol{\lambda}_{N-1} \\ \boldsymbol{\lambda}_{N-2}(t_{N-1}) - \boldsymbol{\lambda}_{N-1}(t_{N-1}) \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \tag{3.11}$$

Using a similar approach to that presented in Section 3.1 it can be shown that

$$\mathbf{H}\mathbf{x} = \mathbf{F}\mathbf{x} - \mathbf{c} = \mathbf{0}, \tag{3.12}$$

where the linear operator, $\mathbf{F}$, is block-tridiagonal,

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_{0,0} & \mathbf{F}_{0,1} & & & & & \\ \mathbf{F}_{1,0} & \ddots & & & & & \\ & & \mathbf{F}_{j,j-1} & \mathbf{F}_{j,j} & \mathbf{F}_{j,j+1} & & \\ & & & & & \ddots & \mathbf{F}_{N-2,N-1} \\ & & & & & \mathbf{F}_{N-1,N-2} & \mathbf{F}_{N-1,N-1} \end{pmatrix}, \tag{3.13}$$

and block $\mathbf{F}_{j,j}$ is

$$\mathbf{F}_{j,j} = \begin{pmatrix} \mathbf{I} & -\mathbf{B}_j & 0 \\ 0 & 2\alpha\mathbf{I} + 2\mathbf{C}_j\mathbf{B}_j & 0 \\ 0 & 2\mathbf{C}_j\mathbf{B}_j & \mathbf{I} \end{pmatrix}, \tag{3.14}$$

where $\mathbf{B}$ and $\mathbf{C}$ are defined as in Section 3.1 and the subscript $\square_j$ denotes the operator $\square$ on segment j. The full structure of $\mathbf{F}$ and $\mathbf{c}$ can be found in Appendix E.

The system, Equation (3.12), cannot be solved using conjugate gradient methods, see Refs. [29, 42], and a generic linear system solver, GMRES, is used. Algorithm 3.3 is a matrix-free method for computing

Algorithm 3.3 A matrix-free method for the calculation of $\mathbf{Hx}$.

Input: $\mathbf{x}$
Output: $\mathbf{Hx}$
for $j = 0, 1, \ldots, N-1$ **do**
  $\mathbf{v}_j(t) \leftarrow$ Solve (2.21a) from t_j to t_{j+1} using $\mathbf{v}_j(t_j)$ and $\mathbf{q}_j(t)$
  $\boldsymbol{\lambda}_j(t) \leftarrow$ Solve (2.21b) from t_{j+1} to t_j using $\boldsymbol{\lambda}_j(t_{j+1})$ and $\mathbf{v}_j(t)$
  $\mathbf{Hx} \leftarrow 2\alpha\mathbf{q}_j(t) - \boldsymbol{\lambda}_j(t)$

for $j = 1, \ldots, N-1$ **do**
  $\mathbf{Hx} \leftarrow \mathbf{v}_{j-1}(t_j) - \mathbf{v}_j(t_j)$
  $\mathbf{Hx} \leftarrow \boldsymbol{\lambda}_{j-1}(t_j) - \boldsymbol{\lambda}_j(t_j)$

$\mathbf{Hx}$. As in Section 3.1, $\mathbf{c}$ is computed by computing $\mathbf{Hx}$ with $\mathbf{x} = \mathbf{0}$. Similarly, $\mathbf{Fx}$ is found by computing $\mathbf{Hx}$ and adding $\mathbf{c}$. We, again, reiterate that at every evaluation of $\mathbf{Fx}$ in the GMRES iterative solver an evaluation of Algorithm 3.3 is required. Algorithm 3.4 is used for finding the optimal control, tangent and co-state solutions. Parallel computation can be leveraged using this method as all the information required

Algorithm 3.4 A matrix-free method for solution of the optimality conditions.

Set $\mathbf{c} = -\mathbf{H0}$
Solve $\mathbf{Fx} = \mathbf{c}$ using GMRES to compute $\mathbf{x}$ where Algorithm 3.3 is used in the evaluation of $\mathbf{Fx}$

for $j = 0, 1, \ldots, N-1$ **do**
  $\mathbf{v}_j(t) \leftarrow$ Solve (2.21a) from t_j to t_{j+1} using $\mathbf{v}_j(t_j)$ and $\mathbf{q}_j(t)$
  $\boldsymbol{\lambda}_j(t) \leftarrow$ Solve (2.21b) from t_{j+1} to t_j using $\boldsymbol{\lambda}_j(t_{j+1})$ and $\mathbf{v}_j(t)$

for each segment is contained within $\mathbf{x}$ and, therefore, computation of the tangent and co-state solutions on each segment along with computation of the constraints between consecutive segments are not reliant on the solutions of other segments.

The main limitation with this method is that the vector of unknowns, $\mathbf{x}$, is larger than $\mathbf{q}(t)$ presented in Section 3.1 due to the unknown conditions of the tangent and co-state at the segment interfaces and, therefore, does not reduce the memory limitations. One method to overcome this limitation is to store the vector $\mathbf{x}$ across compute nodes. Using this approach a parallel implementation of GMRES, see Refs. [32, 33, 87], is required for the solution of the optimality conditions. An alternative approach to remove this limitations is through using various preconditioning methods previously developed in Refs. [5, 29, 42, 49]. These preconditioning methods are presented in the following Sections.

3.2.1. Jacobi preconditioning. The authors of Refs. [5, 29, 42, 49] used the splitting $\mathbf{F} = \mathbf{M} - \mathbf{L} - \mathbf{U}$, where $\mathbf{M}$ is the block diagonal component of $\mathbf{F}$ and $\mathbf{L}$ and $\mathbf{U}$ are the block lower and block upper components, respectively. The intuitive reasoning behind these components is that $\mathbf{M}$ is related to the constraints $\frac{\partial \mathcal{L}}{\partial \mathbf{q}_j}$, $\mathbf{L}$ is related to the constraint $\frac{\partial \mathcal{L}}{\partial \Psi_j}$ and $\mathbf{U}$ is related to the constraint $\frac{\partial \mathcal{L}}{\partial \Phi_j}$. Inverting any of $\mathbf{M}$, $\mathbf{L}$ and $\mathbf{U}$ can be thought of as solving their respective constraints, for example inverting the diagonal blocks of $\mathbf{F}$ is equivalent to solving the optimal control problem on each segment, Ref. [42]. Utilising this splitting, the system, Equation (3.11), can be written as

$$\mathbf{Fx} - \mathbf{c} = \left[\mathbf{M} - (\mathbf{U} + \mathbf{L})\right]\mathbf{x} - \mathbf{c} = \mathbf{0}, \tag{3.15}$$

where left multiplication by $\mathbf{M}^{-1}$ results in

$$\left[\mathbf{I} - \mathbf{M}^{-1}(\mathbf{U} + \mathbf{L})\right]\mathbf{x} - \mathbf{M}^{-1}\mathbf{c} = \mathbf{0}. \tag{3.16}$$

Left multiplication of $\mathbf{M}^{-1}$ always satisfies the constraints $\frac{\partial \mathcal{L}}{\partial \mathbf{q}_j} = \mathbf{0}$, *i.e.* solving the optimal control problem on segment j. We utilise the method set out in section 3.1 to solve the control optimality constrain on each segment which only relies on the tangent and co-state initial and terminal values as the control is solved for in an iterative manner. Using this approach means that initial estimate $\mathbf{q}_j$ in $\mathbf{x}$ is never used. The result of this is that $\mathbf{q}_j$ can be removed from $\mathbf{x}$ simplifying the system. This feature of simplification resulting from Jacobi preconditioning is well known in the optimisation research community, see e.g. Section 4.2 of Ref. [29]. This leaves the unknowns for the Jacobi preconditioning as

$$\mathbf{x}^{\text{Jac}} = \left(\mathbf{v}_1(t_1), \ldots, \mathbf{v}_{j+1}(t_{j+1}), \boldsymbol{\lambda}_{j-1}(t_j), \ldots, \boldsymbol{\lambda}_{N-2}(t_{N-1})\right)^T,$$

which is significantly smaller than $\mathbf{x}$ and can easily fit onto one compute node to be solved with GMRES. These unknowns are the initial tangent conditions at the beginning of each segment and the co-state terminal conditions at the end of each segment. Further, the entries from $\mathbf{F}$ and $\mathbf{c}$ relating to the constraint $\frac{\partial \mathcal{L}}{\partial \mathbf{q}_j}$ can be removed which leads to the simplified Jacobi preconditioned system

$$\begin{pmatrix} \frac{\partial \mathcal{L}}{\partial \Psi_0} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial \Psi_j} \\ \frac{\partial \mathcal{L}}{\partial \Phi_j} \\ \vdots \\ \frac{\partial \mathcal{L}}{\partial \Phi_{N-1}} \end{pmatrix} = \mathbf{H}^{\text{Jac}}\mathbf{x}^{\text{Jac}} = \begin{pmatrix} \mathbf{v}_1(t_1) - \mathbf{v}_0(t_1) \\ \vdots \\ \mathbf{v}_{j+1}(t_{j+1}) - \mathbf{v}_j(t_{j+1}) \\ \boldsymbol{\lambda}_{j-1}(t_j) - \boldsymbol{\lambda}_j(t_j) \\ \vdots \\ \boldsymbol{\lambda}_{N-2}(t_{N-1}) - \boldsymbol{\lambda}_{N-1}(t_{N-1}) \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{pmatrix}. \tag{3.17}$$

Again, it can be shown that

$$\mathbf{H}^{\text{Jac}}\mathbf{x}^{\text{Jac}} = \mathbf{F}^{\text{Jac}}\mathbf{x}^{\text{Jac}} - \mathbf{c}^{\text{Jac}} = \mathbf{0}.$$

Algorithm 3.5 is a matrix-free method for the computation of $\mathbf{H}^{\text{Jac}}\mathbf{x}^{\text{Jac}}$. The term $\mathbf{c}^{\text{Jac}}$ is computed

Algorithm 3.5 A matrix-free method for the calculation of $\mathbf{H}^{\text{Jac}}\mathbf{x}^{\text{Jac}}$.

Input: $\mathbf{x}^{\text{Jac}}$
Output: $\mathbf{H}^{\text{Jac}}\mathbf{x}^{\text{Jac}}$
for $j = 0, 1, \ldots, N-1$ **do**
    Solve Algorithm 3.2 on (t_j, t_{j+1}) from $\mathbf{v}_j(t_j)$ and $\boldsymbol{\lambda}_j(t_{j+1})$
for $j = 1, \ldots, N-1$ **do**
    $\mathbf{H}^{\text{Jac}}\mathbf{x}^{\text{Jac}} \leftarrow \mathbf{v}_{j-1}(t_j) - \mathbf{v}_j(t_j)$
    $\mathbf{H}^{\text{Jac}}\mathbf{x}^{\text{Jac}} \leftarrow \boldsymbol{\lambda}_{j-1}(t_j) - \boldsymbol{\lambda}_j(t_j)$

using $\mathbf{H}^{\text{Jac}}\mathbf{x}^{\text{Jac}}$ using zero for $\mathbf{x}^{\text{Jac}}$. Similarly, $\mathbf{F}^{\text{Jac}}\mathbf{x}^{\text{Jac}}$ is computed using $\mathbf{H}^{\text{Jac}}\mathbf{x}^{\text{Jac}}$ and adding $\mathbf{c}^{\text{Jac}}$. As before, the use of Algorithm 3.5 is required for every evaluation of $\mathbf{F}^{\text{Jac}}\mathbf{x}^{\text{Jac}}$ in the iterative solver. An algorithm for computing the optimal control, tangent and co-state solutions using Jacobi preconditioning is given Algorithm 3.6. One benefit of this approach is that all the optimal control problems on each segment

Algorithm 3.6 A matrix-free method for solution of the optimality conditions.

Set $\mathbf{c}^{\text{Jac}} = -\mathbf{H}^{\text{Jac}}\mathbf{0}$
Solve $\mathbf{F}^{\text{Jac}}\mathbf{x}^{\text{Jac}} = \mathbf{c}^{\text{Jac}}$ using GMRES to compute $\mathbf{x}^{\text{Jac}}$ where Algorithm 3.5 is used in the evaluation of $\mathbf{F}^{\text{Jac}}\mathbf{x}^{\text{Jac}}$

for $j = 0, 1, \ldots, N-1$ **do**
    Solve Algorithm 3.2 on (t_j, t_{j+1}) from $\mathbf{v}_j(t_j)$ and $\boldsymbol{\lambda}_j(t_{j+1})$

can be solved in parallel, even distributed across multiple compute nodes, thus distributing computational resources and reducing the memory requirements.

3.2.2. Gauss-Seidel preconditioning. We utilise the same splitting of $\mathbf{F} = \mathbf{M} - \mathbf{L} - \mathbf{U}$ used in Section 3.2.1. We can, therefore, left multiply Equation (3.15) by $(\mathbf{M} - \mathbf{L})^{-1}$ resulting in

$$\left[\mathbf{I} - (\mathbf{M} - \mathbf{L})^{-1}\mathbf{U}\right]\mathbf{x} - (\mathbf{M} - \mathbf{L})^{-1}\mathbf{c} = \mathbf{0}. \tag{3.18}$$

Left multiplication by $(\mathbf{M}-\mathbf{L})^{-1}$ can be thought of as solving the optimal control on each segment and then applying continuity in the tangent solution between consecutive segments. We refer to this left multiplication as Forward Gauss-Seidel (FGS) preconditioning. The control optimality conditions, on each segment, are solved using the method outlined in section 3.1. As before utilising this approach results in the values of $\mathbf{q}_j$ being removed from $\mathbf{x}$. Applying continuity in the tangent solution results in the tangent interface values, $\mathbf{v}_j(t_j)$ for $j=1,\ldots,N-1$, in $\mathbf{x}$ never being used as they are updated from the preceding segment. Therefore they can also be removed from $\mathbf{x}$. This method results in the unknown terms

$$\mathbf{x}^{\mathrm{FGS}} = \left(\boldsymbol{\lambda}_{j-1}(t_j),\ldots,\boldsymbol{\lambda}_{N-2}(t_{N-1})\right)^T.$$

These unknowns are the co-state terminal conditions at the end of each segment. By utilising this left multiplication, we find that the FGS linear system satisfies the constraints $\frac{\partial\mathcal{L}}{\partial\mathbf{q}_j}=\mathbf{0}$ and $\frac{\partial\mathcal{L}}{\partial\Psi_j}=\mathbf{0}$ and, as before, we can remove these from the linear system resulting in

$$\begin{pmatrix}\frac{\partial\mathcal{L}}{\partial\Phi_1}\\ \vdots\\ \frac{\partial\mathcal{L}}{\partial\Phi_{N-1}}\end{pmatrix} = \mathbf{H}^{\mathrm{FGS}}\mathbf{x}^{\mathrm{FGS}} = \begin{pmatrix}\boldsymbol{\lambda}_0(t_1)-\boldsymbol{\lambda}_1(t_1)\\ \vdots\\ \boldsymbol{\lambda}_{N-2}(t_{N-1})-\boldsymbol{\lambda}_{N-1}(t_{N-1})\end{pmatrix} = \begin{pmatrix}\mathbf{0}\\ \vdots\\ \mathbf{0}\end{pmatrix}. \tag{3.19}$$

Again, it can be shown that

$$\mathbf{H}^{\mathrm{FGS}}\mathbf{x}^{\mathrm{FGS}} = \mathbf{F}^{\mathrm{FGS}}\mathbf{x}^{\mathrm{FGS}} - \mathbf{c}^{\mathrm{FGS}} = \mathbf{0}.$$

Algorithm 3.7 is used for computing $\mathbf{H}^{\mathrm{FGS}}\mathbf{x}^{\mathrm{FGS}}$ in a matrix-free sense. Computation of Algorithm 3.7 is

Algorithm 3.7 A matrix-free method for the calculation of $\mathbf{H}^{\mathrm{FGS}}\mathbf{x}^{\mathrm{FGS}}$.

Input: $\mathbf{x}^{\mathrm{FGS}}$
Output: $\mathbf{H}^{\mathrm{FGS}}\mathbf{x}^{\mathrm{FGS}}$
for $j=0,1,\ldots,N-1$ **do**
  Solve Algorithm 3.2 on (t_j,t_{j+1}) from $\mathbf{v}_j(t_j)$ and $\boldsymbol{\lambda}_j(t_{j+1})$
  if $j\neq N-1$ **then**
    $\mathbf{v}_{j+1}(t_{j+1}) \leftarrow \mathbf{v}_j(t_{j+1})$
for $j=1,\ldots,N-1$ **do**
  $\mathbf{H}^{\mathrm{FGS}}\mathbf{x}^{\mathrm{FGS}} \leftarrow \boldsymbol{\lambda}_{j-1}(t_j)-\boldsymbol{\lambda}_j(t_j)$

commonly referred to as instantaneous control in the literature Refs. [8, 25, 27, 43, 44, 45]. The term $\mathbf{c}^{\mathrm{FGS}}$ is computed using $\mathbf{H}^{\mathrm{FGS}}\mathbf{x}^{\mathrm{FGS}}$ using zero for $\mathbf{x}^{\mathrm{FGS}}$. Similarly, $\mathbf{F}^{\mathrm{FGS}}\mathbf{x}^{\mathrm{FGS}}$ is computed using $\mathbf{H}^{\mathrm{FGS}}\mathbf{x}^{\mathrm{FGS}}$ and adding $\mathbf{c}^{\mathrm{FGS}}$. We remind the reader that an evaluation of $\mathbf{F}^{\mathrm{FGS}}\mathbf{x}^{\mathrm{FGS}}$ in the GMRES iterative solver requires the use of Algorithm 3.7. An algorithm for computing the optimal control, tangent and co-state solutions using FGS preconditioning is given Algorithm 3.8.

Algorithm 3.8 A matrix-free method for solution of the optimality conditions.

Set $\mathbf{c}^{\mathrm{FGS}} = -\mathbf{H}^{\mathrm{FGS}}\mathbf{0}$
Solve $\mathbf{F}^{\mathrm{FGS}}\mathbf{x}^{\mathrm{FGS}} = \mathbf{c}^{\mathrm{FGS}}$ using GMRES to compute $\mathbf{x}^{\mathrm{FGS}}$

for $j=0,1,\ldots,N-1$ **do**
  Solve Algorithm 3.2 on (t_j,t_{j+1}) from $\mathbf{v}_j(t_j)$ and $\boldsymbol{\lambda}_j(t_{j+1})$

A Backwards Gauss-Seidel (BGS) preconditioning approach can be achieved by left multiplying Equation (3.15) by $(\mathbf{M}-\mathbf{U})^{-1}$. The BGS preconditioning approach can be thought of as solving the optimal control on each segment and then applying continuity on the co-state solution, leaving only the tangent continuity constraints unsatisfied. $\mathbf{H}^{\mathrm{BGS}}\mathbf{x}^{\mathrm{BGS}}$ is computed in a matrix-free sense by solving optimal control problem on the final segment, with their respective initial and terminal conditions on the tangent and co-state equations then setting $\boldsymbol{\lambda}_{N-2}(t_{N-1}) = \boldsymbol{\lambda}_{N-1}(t_{N-1})$ and then repeating this process on the previous segment. Then the discontinuity is computed on the tangent solutions between consecutive segments. The linear system $\mathbf{F}^{\mathrm{BGS}}\mathbf{x}^{\mathrm{BGS}} = \mathbf{c}^{\mathrm{BGS}}$ is solved in a similar manner as that for the FGS approach.

Finally, one drawback of the Gauss-Seidel preconditioning methods is that the parallel computation of segments is removed due to the continuity constraint between segments.

4. Performance of OCS on the Lorenz system. In this Section we restrict our investigation to the tangent OCS formulation and show the impact that the parameter α has on the sensitivity generated by OCS. Computational aspects involving the total cost of the algorithm as a function number of segments and α are discussed along with the convergence rate for different preconditioning methods for a fixed α and number of segments. For this analysis, we utilise the Lorenz 1963 system, Ref. [56], which has been extensively utilised as a benchmark for sensitivity analysis of chaotic systems in previous work, Refs. [9, 10, 13, 15, 18, 22, 23, 24, 26, 47, 53, 54, 64, 65, 74, 83, 84, 85].

4.1. Description of the Lorenz system. The Lorenz system, Ref. [56], was developed as a simplified model for atmospheric convection and is given by the following system of ordinary differential equations

$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma\,(y - x) \tag{4.1a}$$

$$\frac{\mathrm{d}y}{\mathrm{d}t} = x\,(\rho - z) - y \tag{4.1b}$$

$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z. \tag{4.1c}$$

The state vector is $\mathbf{u} = (x, y, z)^T$, where $x \in \mathbb{R}$ represents the rate of convection of the problem, $y \in \mathbb{R}$ is the horizontal temperature variation, $z \in \mathbb{R}$ is the vertical temperature variation. The parameter σ is proportional to the Prandtl number of the flow, ρ is proportional to the Rayleigh number, and β represents a physical thickness of the fluid layer. Typical values for σ, ρ and β used in other studies are 10, 28 and $\frac{8}{3}$ respectively, and shall be used here unless otherwise stated.

The time step for the numerical simulation of the system is $\Delta t = 0.01$ time units for numerical stability and a fourth order Runge-Kutta time-stepping scheme is used. All results are generated using an initial conditions for x, y and z drawn from a uniform distribution between 0 and 1.

4.2. Numerical computation of the Lorenz system sensitivity. We select ρ to be our parameter of interest as in other studies and

$$J(\mathbf{u}) = z, \tag{4.2}$$

as the quantity of interest. To aid in the investigation into the accuracy in the time averaged sensitivity generated by OCS we first compute the impact that ρ has on the time averaged cost $\bar{J}$. We compute $\bar{J}$ for the range $\rho \in (1, 100)$ for 100 equally spaced values where each sample has random initial conditions. Each sample has a 'spin up' time of 50 time units to ensure the solution is on the attractor and the time average is computed over 1500 time units. This can be seen in Figure 2 along with a curve fit of the solution following the discontinuity at $\rho = 24$. The curve fit can then be utilised to compute the derivative of $\bar{J}$ with respect to ρ. The derivative of the curve fit at $\rho = 28$ is $\frac{\mathrm{d}\bar{J}}{\mathrm{d}\rho} \approx 1.006$ and it is this value that will be utilised for comparison throughout the remainder of this section. The authors note that the tangent and adjoint formulations are consistent with each other as $\Delta t \to 0$ along with the error between the methods decaying to zero at a rate consistent with the order of accuracy of the time-stepping scheme.

4.3. Influence of α on the sensitivity generated by OCS. We now compute the sensitivity generated by OCS for the range $\alpha \in (1, 1 \times 10^{15})$ when $T = 30$ time units, a 'spin up' time of 50 time units and the same initial conditions are used for Equation (4.1) for each value of α. We undertake this analysis to investigate how varying the cost of the control applied varies the sensitivity generated. This influence has been undertaken on three different initial conditions for the non-linear system and can be seen in Figure 3. The first observation to note is that the initial condition chosen impacts the value of the sensitivity generated. However, it is well known that Shadowing methods increase in accuracy as the time horizon is increased, Ref. [54, 82], and, therefore, this bias will reduce. Secondly, taking an average of the sensitivity over a range of initial conditions will also reduce the bias.

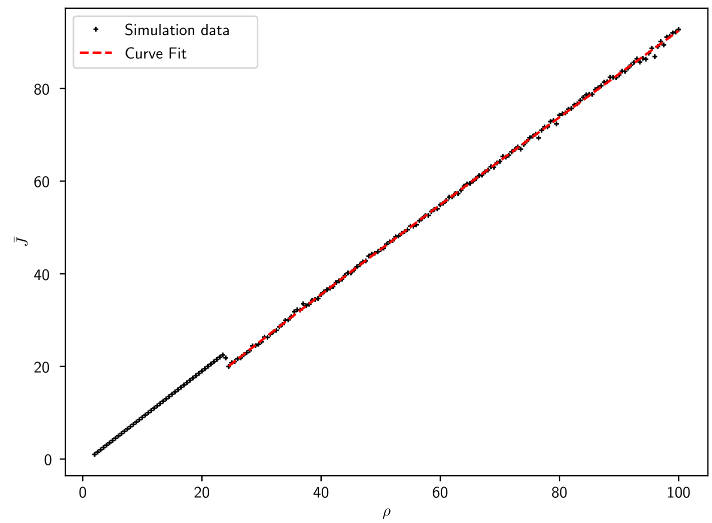


Fig. 2: $\bar{J}$ computed over 1500 time units against ρ for the Lorenz system.

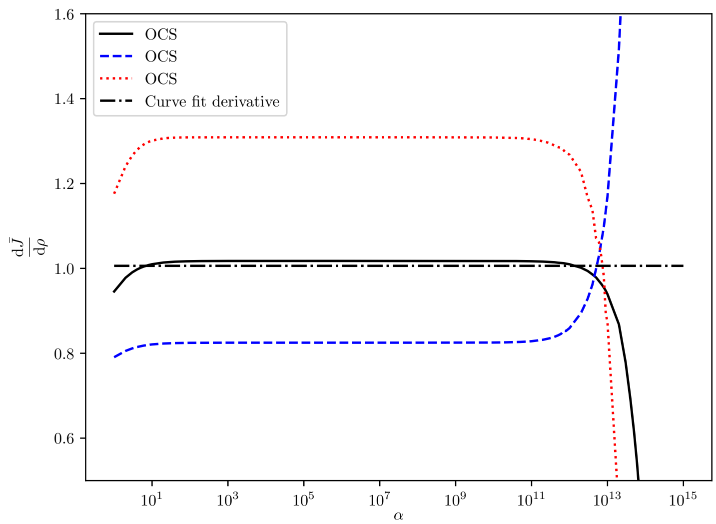
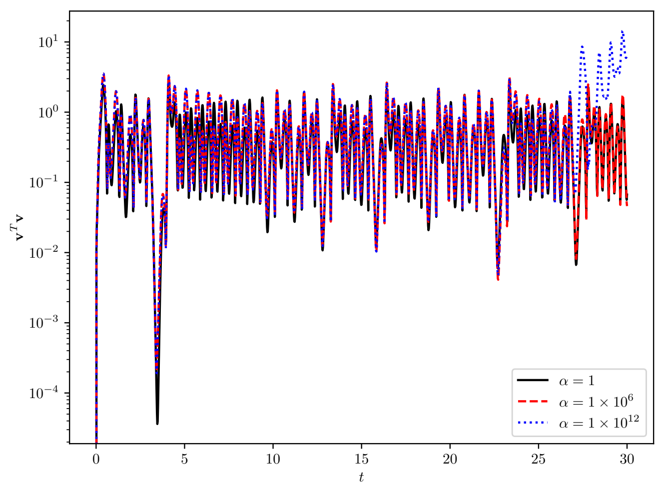


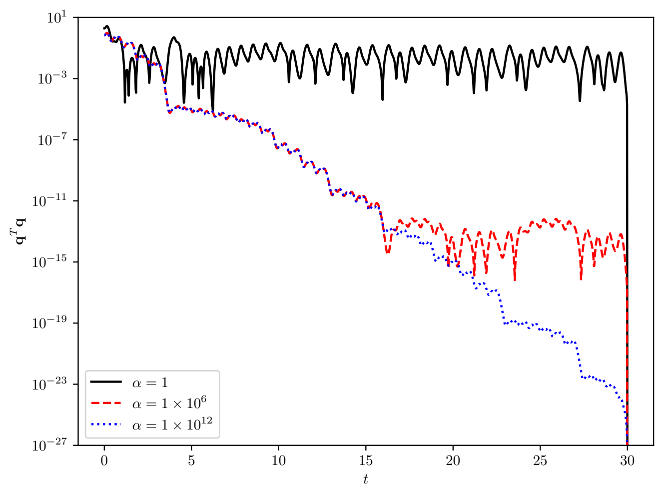
Fig. 3: The sensitivity $\frac{\mathrm{d}\bar{J}}{\mathrm{d}\rho}$ against α for OCS for three different non-linear initial conditions, solid black line, dashed blue line and dotted red line. The derivative of the curve fit derivative of Figure 2, black dash dotted line, is also shown for comparison.

It can be observed that when $\alpha \leq 10$ the sensitivity is under-predicted. This is because the control in this region is considered cheap and a large amount of control is applied. This implies that the tangent solution is significantly damped which then results in a lower sensitivity $\mathrm{d}\bar{J}/\mathrm{d}p$ (see equation (2.25)). In the limit case $\alpha \to 0$, the tangent solution $\mathbf{v}$ vanishes completely and so does the sensitivity of the time average. By contrast, when $\alpha \geq 1 \times 10^{11}$ the control applied is insufficient and is unable to stabilise the tangent solution (under-damping) which leads to inaccurate sensitivities, either much larger or much smaller. In between these regions the control applied is large enough to control the exponential growth of perturbations but small enough not to damp the entire solution. Comparisons not reported here with solutions obtained with MSS suggest that, excluding the initial and final fractions of the time horizon, the tangent solution generated by OCS converges to the true shadowing direction. This is a bounded, but unstable, solution of the linearised equations which many shadowing methods find approximations for. In the present approach, such an approximation is found by finding an appropriate stabilising control using optimal control theory. A remark on the selection of the α is in order. This parameter has units of the inverse of a time scale, but the "ideal" range in which the solution is neither over-damped nor under-damped does not seem to be related in a straightforward manner with relevant time scales of the linearised dynamics, e.g. the Lyapunov time associated to the single positive exponent of the Lorenz equations.

The squared norms of the optimal tangent, $\mathbf{v}^T\mathbf{v}$, and control, $\mathbf{q}^T\mathbf{q}$, solutions as a function of time are displayed in Figure 4 for $\alpha \in \{1, 1\times10^6, 1\times10^{12}\}$. Panel (a) shows that for all α values investigated the



(a)



(b)

Fig. 4: Comparison of the squared norms of the tangent solution, panel (a), and of the control, panel (b), for $\alpha = 1$, solid black line, $\alpha = 1\times10^6$, red dotted line, and $\alpha = 1\times10^{12}$ blue dotted line.

tangent solution is stabilised onto approximations of the shadowing direction for the initial five sixths of the time horizon. There are slight differences between the tangent solutions generated by each value of α and, therefore, the same approximation to the shadowing direction is not found between cases.

Panel (b) shows that when $\alpha = 1$ there is a constant amount of control being applied throughout the time horizon which is able to counteract the growth of perturbations. The dip in control applied at the end of the time horizon is due to the terminal condition $\boldsymbol{\lambda}(t_f) = \mathbf{0}$. When $\alpha = 1 \times 10^6$ there is significantly less control applied, and the control value decays exponentially for the first two thirds of the time horizon and then reaches a constant value for the remainder. Finally, when $\alpha = 1 \times 10^{12}$ the control follows a similar profile for the previous case except the constant value reached is several orders of magnitude lower. This exponential decay in the first two thirds of the time horizon of the control solution suggests that a solution with exponential growth is being controlled. In the final third of the time horizon there is significant difference between $\alpha = 1 \times 10^{12}$ and the other two cases. This is caused by the small value of control applied which is unable to control exponential growth in the tangent solution. This is the mechanism that causes the sensitivity values to be inaccurate for large values of α.

The implications that this has on NV methods is that the control applied by the additional term must be large initially, but might need to be weaker across the rest of the time horizon. This will have the impact of stabilising the tangent solution towards the shadowing direction, but once the tangent solution is close to the shadowing direction minimal amounts of control are required.

4.4. Convergence rates of the preconditioning methods. The stopping criteria for the GMRES method is based on the value of the residual $\mathbf{r} = \mathbf{Hx} - \mathbf{c}$ of Equation (3.12). In this analysis the stopping criteria is chosen to be 1×10^{-14} for both CG and GMRES as this value is close to machine epsilon in 64-bit double-precision floating-point arithmetic. A large stopping tolerance means that there are discontinuities between segments, and the control will not be optimal, and as such the values generated for the sensitivities can be impacted.

The preconditioning methods derived, Jacobi, FGS and BGS are compared to the no preconditioning case in Figure 5 for $T = 30$ time units, a 'spin up' time of 50 time units, $\alpha = 500$, $N = 6$ and using the same initial condition for Equation (4.1). It can be seen that applying no preconditioning has the slowest rate

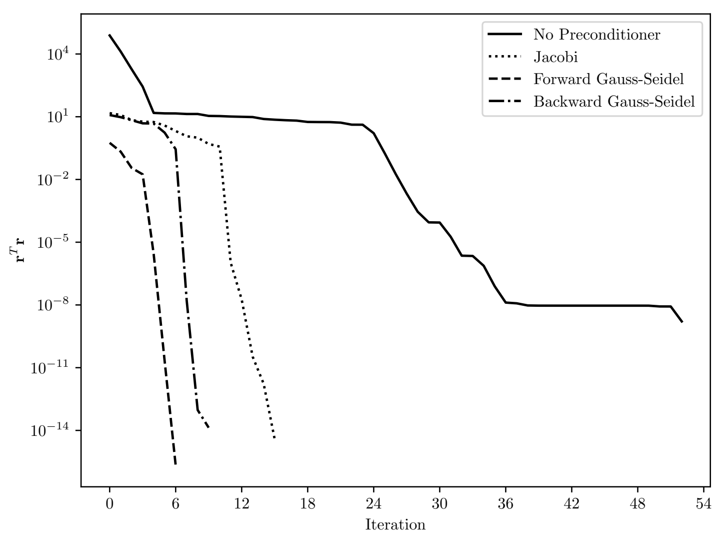


Fig. 5: The norm of the residual, $\mathbf{r}$, for no preconditioning, solid line, Jacobi preconditioning, dotted line, FGS, dashed line, and BGS, dash dotted line.

of convergence, followed by Jacobi then BGS and, finally, FGS having the fastest convergence rate of those investigated. These results are in good agreement with those presented in Figure 6 in Ref. [42], Figure 5.6 in Ref. [28] and Tables 5.2 to 5.4 in Ref. [29] despite the difference in system investigated. Although the results from Refs. [28, 29, 42] are for advection-diffusion and heat equation problems the linear unstable system investigated here still shows very similar behaviour. Therefore, it can be concluded that the difference in class of system does not impact the convergence rates of the method.

When there is no preconditioning applied, the converge rate is slow because all variables, control and interface values for the tangent and co-state, are solved together. Therefore, at each iteration the control values on each segment and tangent and co-state values at the segment interfaces are not guaranteed to be optimal. Jacobi preconditioning, on the other hand, solves the optimal control problem on each segment at each GMRES iteration with discontinuities in both the tangent and co-state solution between consecutive segments. This results in "information" being propagated between consecutive segments faster than no preconditioning. BGS solves the optimal control problem sequentially backwards in time, meaning the co-state equation always obeys continuity across the segment boundaries. This leads to an increased rate of "information" transfer compared to the Jacobi preconditioning. The reason why the FGS converges quicker than BGS is that the formulation derived in Section 2 aims to minimise the norm of the tangent solution. By ensuring that there is continuity between consecutive segments in the tangent solution, as the FGS preconditioning does, its growth is known in each segment. By knowing the growth of the tangent solution in each segment $\mathbf{q}$ can applied in a more "appropriate" fashion to better stabilise the solution, which leads to fewer iterations.

From these results it is clear that FGS requires the fewest number of iterations, but FGS does not have a favourable parallel efficiency, as the optimal control problems on the segments must be solved sequentially. In this regard, from the perspective of maximising compute resources, Jacobi preconditioning is advantageous over FGS or BGS, because the increased parallelism of the Jacobi preconditioning may result in the CPU wall-clock time of a single GMRES iteration being reduced over FGS or BGS and may lead to a faster overall solution in terms of CPU wall-clock time depending on the number of segments and other factors. Here we make use of the FGS method due to the faster convergence rates and disregard the impact on computational efficiency.

4.5. Total cost of the OCS algorithm. For a given preconditioning method, there are two factors that will impact the cost of the OCS algorithm, the first being α, due to its impact on the condition number of the linear system associated with the optimal control problem in each segment and the second being the number of segments, N. We now compute the total cost of the algorithm for $N \in \{4, 6, 8, 12, 24\}$ with $\alpha \in (1, 1 \times 10^{19})$ for $T = 30$ time units, a 'spin up' time of 50 time units using the FGS preconditioning method. The number of tangent and co-state solutions are directly tied to the number of CG and GMRES iterations. For a solution of the optimal control problem on each segment a tangent and co-state solution are required for the computation of $\mathbf{b}_j$. A forward and backward solution are then required for each call to $\mathbf{E}_j\mathbf{q}_j$. Finally, solutions to the tangent and co-state equations are required once the optimal control $\mathbf{q}$ has been found. Solving the discontinuities between segments leads to one optimal control solution required for the computation of $\mathbf{c}^{\mathrm{Jac}}$, $\mathbf{c}^{\mathrm{FGS}}$ or $\mathbf{c}^{\mathrm{BGS}}$. An optimal control solution is required every GMRES iteration for the evaluation of $\mathbf{F}^{\mathrm{Jac}}\mathbf{x}^{\mathrm{Jac}}$, $\mathbf{F}^{\mathrm{FGS}}\mathbf{x}^{\mathrm{FGS}}$ or $\mathbf{F}^{\mathrm{BGS}}\mathbf{x}^{\mathrm{BGS}}$. Finally, one more optimal control solution is required once the interface values for the tangent and co-state has been found. Therefore, the total cost of the algorithm in terms of number of tangent and co-state solutions required for convergence through can be found with

$$\text{Total Cost} = (\text{Iters}_{\text{GMRES}} + 2) \times (2 \times \text{Iters}_{\text{CG}} + 4)\,.$$

The change in cost through varying α and the number of segments for one simulation is presented in Figure 6. For large values of α fewer segments are preferred, however, for small values of α the difference between the cost generated by varying the number of segments is reduced. As can be found in Figure 6 in Ref. [42], Figure 5.6 in Ref. [28] and Tables 5.2 to 5.4 in Ref. [29] fewer segments produce lower computational cost. Based on this analysis, the fewest number of segments should be used. However, due to memory limitations this may not be feasible and a larger number of segments may be necessary. In practice, the number of segments used are selected such that the memory required for the optimal control problem on each segment is smaller than the memory each compute node has available. Finally, the method developed in Refs. [28, 29, 42] are for advection-diffusion and heat equation problems the linear unstable system investigated here still shows very similar cost. It can, therefore, be concluded that the difference in the systems has little, to no, impact on the convergence rates of the method.

5. Performance of OCS on the Kuramoto-Sivashinsky system. In this Section we investigate the performance of the tangent OCS formulation on a spatially distributed system. The Kuramoto-Sivashinsky system, Refs. [50, 51, 76, 77], has been extensively investigated for sensitivity analysis of chaotic systems, Refs. [9, 10, 11, 12, 14, 18, 53, 74, 75, 85], and we use it here as a stepping stone to larger more industrially

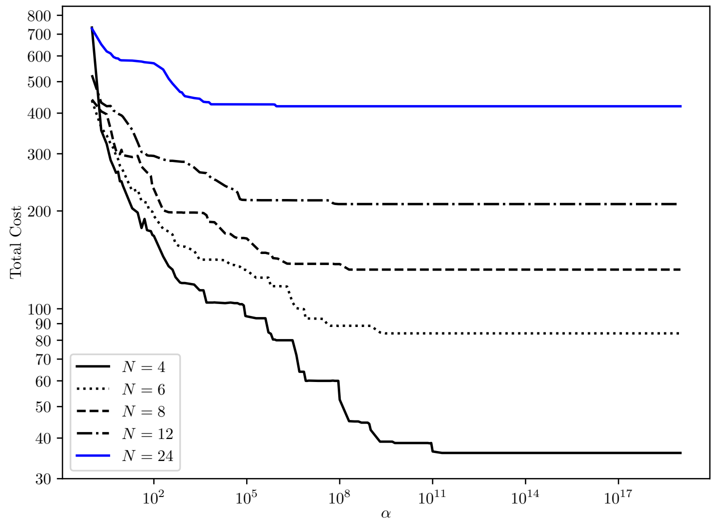


Fig. 6: Total cost of the OCS algorithm against α for 4 segments, solid black line, 6 segments, dotted black line, 8 segments, dashed black line, 12 segments, dash dotted black line, and 24 segments, blue solid line.

relevant systems. The impact that the domain size has on the convergence rate of OCS is investigated, followed by a comparison between MSS, OCS and NV methods.

5.1. Description of the Kuramoto-Sivashinsky system. The Kuramoto-Sivashinsky (KS) equation was initially introduced in Refs. [50, 51] as a method of modelling angular-phase turbulence in a system of reaction-diffusion equations. Later, the equations were derived in Refs. [76, 77] to model how the instabilities of a distributed plane flame front evolve. The KS system investigated here is given by

$$\begin{gathered}\frac{\partial u(x,t)}{\partial t} = -(u(x,t)+c)\frac{\partial u(x,t)}{\partial x} - \frac{\partial^2 u(x,t)}{\partial x^2} - \frac{\partial^4 u(x,t)}{\partial x^4}, \\ x \in (0, L), \\ u(0,t) = u(L,t),\end{gathered} \tag{5.1}$$

where $u(x,t)$ is a spatially distributed variable with L being the length of the domain. The variable c is the mean convective speed, and is used here as the parameter of interest, as in previous work, Refs [9, 10, 11, 12, 14, 18, 53, 74, 75, 85]. There have been several investigations into the modified KS system where Neumann and Dirichlet boundary conditions are used, Refs [9, 10, 11, 12, 14, 18, 53, 74, 75, 85], instead of periodic boundary conditions used here. In this Section, periodic boundary conditions are utilised as this facilitates the use of spectral methods and a wavenumber-by-wavenumber analysis of the tangent energy budget derived in Section 5.2. The solution can be expanded as

$$u(x,t) = \sum_{k=-\infty}^{\infty} \tilde{u}_k(t) e^{ik\frac{2\pi}{L}x},$$

but, in practice, a grid of $K_x = \frac{6L}{5}$ grid points is used, so that the spatial resolution is independendent of L. As the solution is real valued, this results in $\frac{K_x}{2}+1$ Fourier modes. A zero mean solution is chosen, *i.e.* $\tilde{u}_0 = 0$, which is commonly performed in the literature, Ref [30]. For the time integration a Crank-Nicolson scheme is used for the linear terms and a second order Runge-Kutta scheme is used for the non-linear terms and $\Delta t = 1 \times 10^{-3}$ time units is used for stability. The inner product between two spatially distributed variables, $a(x,t)$ and $b(x,t)$,

$$\langle a(x,t), b(x,t)\rangle = \frac{1}{L}\int_0^L a(x,t)b(x,t)\mathrm{d}x$$

is used and the norm is

$$\|a(x,t)\| = \sqrt{\langle a(x,t), a(x,t)\rangle}.$$

We consider a domain length $L = 50$ and a convective speed $c = 0.5$. This domain length is sufficiently large for the dynamics to display fully developed spatial chaos, with no qualitative change on the behaviour when larger domain sizes are considered. A typical example of the solution to the KS system for such parameter values is given in Figure 7.

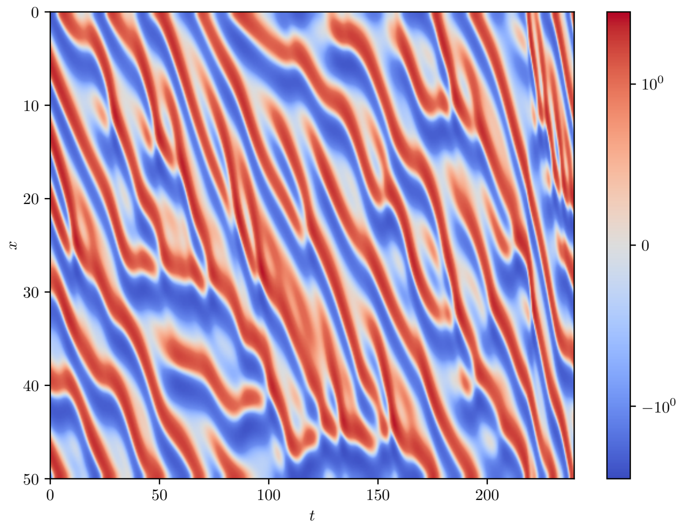


Fig. 7: Example of a typical KS solution, $u(x,t)$, for $c = 0.5$ and $L = 50$.

5.2. Derivation of the tangent energy equation. The main development of NV methods has been for fluid flow problems, Ref. [18], and has not been developed for the KS system. Therefore, we need to derive the NV approach for the KS system. The approach taken to derive the NV method in Ref. [18] consisted in selecting a term which dissipates excess adjoint energy and stabilises the solution. Here, we follow a similar approach by finding terms that contribute to the dissipation of tangent energy. We undertake this in terms of tangent energy as we are primarily investigating the tangent OCS formulation derived in Section 2.2.

Starting from the KS system, the linearised tangent equation with control term included is

$$\frac{\partial v(x,t)}{\partial t} = \underbrace{-c\frac{\partial v(x,t)}{\partial x} - \frac{\partial (u(x,t)v(x,t))}{\partial x} + \eta f(x,t)}_{C(x,t)} - \underbrace{\frac{\partial^2 v(x,t)}{\partial x^2}}_{P(x,t)} - \underbrace{\frac{\partial^4 v(x,t)}{\partial x^4}}_{D(x,t)} - \underbrace{\frac{\partial u(x,t)}{\partial x}}_{I(x,t)} + \underbrace{q(x,t)}_{Q(x,t)}, \tag{5.2}$$
$$x \in [0, L),$$
$$v(0,t) = v(L,t),$$

where $C(x,t)$ can be seen as a combination of terms that are producing and dissipating tangent energy (with $f(x,t)$ the right hand side of Equation (5.1)), $P(x,t)$ purely contributes to production of tangent energy, $D(x,t)$ is a dissipation term, $I(x,t)$ is an inhomogeneous term, and $Q(x,t)$ is the contribution to the tangent energy of the control term. Similar to the nonlinear KS equation, the tangent formulation has periodic boundary conditions. Therefore, the tangent and control variables can be expanded in a Fourier series as

$$v(x,t) = \sum_{k=-\infty}^{\infty} \tilde{v}_k(t) e^{ik\frac{2\pi}{L}x} \quad \text{and} \quad q(x,t) = \sum_{k=-\infty}^{\infty} \tilde{q}_k(t) e^{ik\frac{2\pi}{L}x}.$$

Let $\epsilon(x,t) = \frac{1}{2}v(x,t)^2$ denote the local tangent energy. An evolution equation for this variable is found by multiplication of Equation (5.2) by $v(x,t)$, resulting in

$$\frac{\partial \epsilon(x,t)}{\partial t} = -2\epsilon(x,t)\frac{\partial u(x,t)}{\partial x} - (u(x,t)+c)\frac{\partial \epsilon(x,t)}{\partial x} - v(x,t)\frac{\partial^2 v(x,t)}{\partial x^2} + v(x,t)\eta f(x,t) - v(x,t)\frac{\partial^4 v(x,t)}{\partial x^4} - v(x,t)\frac{\partial u(x,t)}{\partial x} + v(x,t)q(x,t), \tag{5.3}$$

Defining the tangent energy, $\mathcal{E}(t) = \frac{1}{L}\int_0^L \epsilon(x,t)\mathrm{d}x$, as the domain average of the local tangent energy, results in the scalar evolution equation

$$\frac{\mathrm{d}\mathcal{E}(t)}{\mathrm{d}t} = \frac{1}{L}\int_0^L \left(-2\epsilon(x,t)\frac{\partial u(x,t)}{\partial x} - (u(x,t)+c)\frac{\partial \epsilon(x,t)}{\partial x} - v(x,t)\frac{\partial^2 v(x,t)}{\partial x^2} + v(x,t)\eta f(x,t) - v(x,t)\frac{\partial^4 v(x,t)}{\partial x^4} - v(x,t)\frac{\partial u(x,t)}{\partial x} + v(x,t)q(x,t)\right)\mathrm{d}x. \tag{5.4}$$

Integration of the third term on the right hand side of Equation (5.4) by parts results in

$$\int_0^L -v(x,t)\frac{\partial^2 v(x,t)}{\partial x^2}\mathrm{d}x = \int_0^L \left(\frac{\partial v(x,t)}{\partial x}\right)^2 \mathrm{d}x,$$

which can be seen to be a production term as it is always positive. Similarly, integration of the fourth term by parts twice leads to

$$\int_0^L -v(x,t)\frac{\partial^4 v(x,t)}{\partial x^4}\mathrm{d}x = \int_0^L -\left(\frac{\partial^2 v(x,t)}{\partial x^2}\right)^2 \mathrm{d}x,$$

which is always negative and is a dissipation term. The first numerical viscosity term considered here is

$$q(x,t) = \mu_P \frac{\partial^2 v(x,t)}{\partial x^2}$$

which modifies the production of tangent energy where $\mu_P > 0$ is a tuning parameter and shall be referred to as NV_P henceforth. When $\mu_P > 1$ the control applied by NV_P counteracts the influence of $P(x,t)$. Similarly, modification of the dissipation of tangent energy leads to

$$q(x,t) = -\mu_D \frac{\partial^4 v(x,t)}{\partial x^4}$$

where $\mu_D > 0$ is a scaling factor for the artificial viscosity term and the approach shall be referred to as NV_D.

5.3. Numerical computation of the sensitivity for the Kuramoto-Sivashinsky system. The functional of interest used for sensitivity analysis is

$$J\left(u(x,t)\right) = \|u(x,t)\|^2, \tag{5.5}$$

which represents the energy density of the solution variable $u(x,t)$. The value of $\bar{J}$ is found using a random initial condition for Equation (5.1) for a time horizon of $T \in \{2\times10^3, 2\times10^4\}$ time units following a 'spin up' time of 1000 time units. This is undertaken over 50 samples. The change in mean value $\bar{J}$ and standard deviation as a function of c for $c \in (0.0, 1.0)$ using 100 equally spaced points is shown in Figure 8, where a curve fit through the data is also shown. We note that the literature uses Neumann and Dirichlet boundary conditions and the resulting correlation between c and $\bar{J}$ is stronger, see Figure 7 in Ref. [14]. Figure 8 indicates that as the time horizon increases the average value of $\bar{J}$ is approximately constant in this range of c values. This is shown by the mean converging to the curve fit as T increases along with the standard deviation decreasing. The derivative of the curve fit when $c = 0.5$ is $\frac{\mathrm{d}\bar{J}}{\mathrm{d}c} \approx 6.725\times10^{-2}$, i.e. approximately zero and shall be used for comparison throughout this Section.

We then solve the tangent OCS problem for $\alpha \in (10, 1\times10^{10})$ using $T = 240$ time units, preceded by a 'spin up' time of 1000 time units, $L = 50$ and $K_x = 60$. This procedure is repeated for $\mu_P \in (0,5)$ and

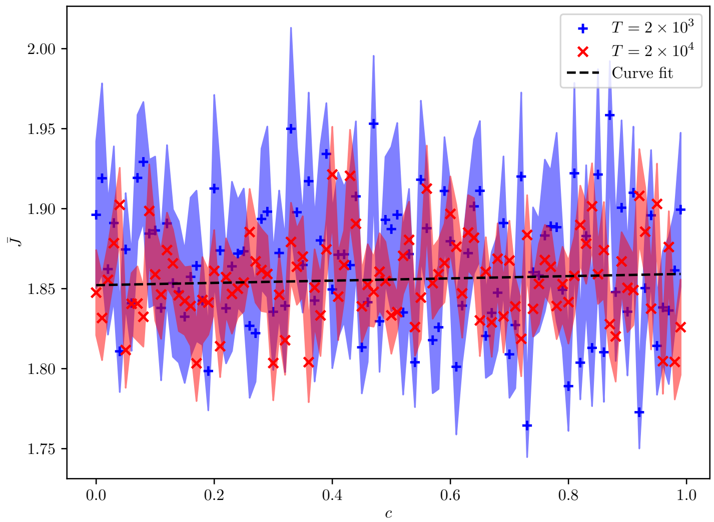
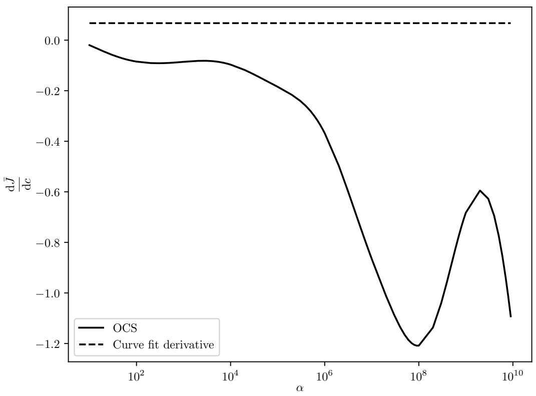


Fig. 8: Mean and standard deviation of $\bar{J}$ against c for the KS system from numerical simulation when $T = 2 \times 10^3$, blue crosses, $T = 2 \times 10^4$, red crosses, and a curve fit through when $T = 2 \times 10^4$, black dashed line.

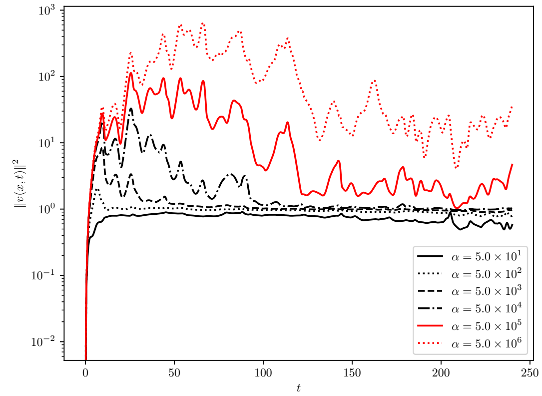
$\mu_D \in (0, 50)$ using the same conditions. In Figure 9 we show the sensitivity generated by tangent OCS, NV$_P$ and NV$_D$ in the left column for the range of tuning parameters along with, in the right column, $\|v(x,t)\|^2$ for selected values of the tuning parameters. We find that there is some bias with the OCS solution and the curve fit derivative, Figure (9a) which is caused by the selection of the initial condition on $u(x, t_s)$, as shown previously for the Lorenz system. We note that even though there is little correlation between c and $\bar{J}$ we still observe that the influence of α shows similar features as those presented for the Lorenz case and this lack of correlation is not a major limitation in the analysis. When $\alpha < 5 \times 10^2$ the solution is over damped. Again when $\alpha > 5 \times 10^4$ the solution is under damped. In between these values the sensitivities are in good agreement with the curve fit derivative. Above this value the sensitivities become inaccurate. This, as we will show later, is due to the control applied being too small and allowing the tangent solution to experience exponential growth, Figure (9b). We also observed that from $\alpha > 5 \times 10^3$ the tangent solution can experience significant transient growth in the first fraction of the time span, before control is able to stabilize it. One interesting feature to note is that the norm of the tangent solution can be quite large, and still produce sensitivity values in good agreement with the curve fit derivative.

For the NV methods, we find that when $\mu_P < 1$ the sensitivities generated by NV$_P$ are inaccurate, Figure (9c), which is due to the control being inadequate and not able to stabilise the tangent solution, as can be seen in panel (d). When μ_P is increased above 2 the sensitivity generated becomes in better agreement with the curve fit derivative. This is due to the control being sufficient to damp the exponential growth of the tangent solution. Similar behaviour can be seen for the NV$_D$ approach too.

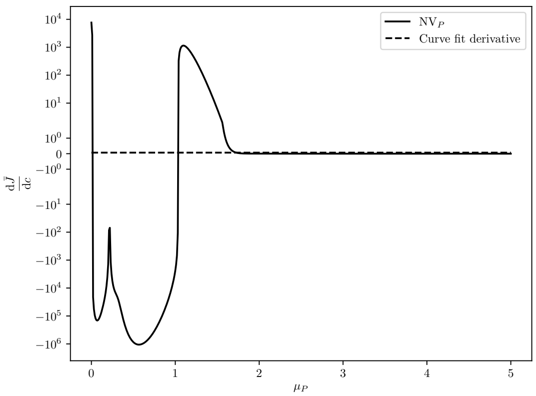
5.4. Influence of the domain size on the convergence rate of the OCS algorithm. We now investigate the influence of the size of the domain, L, has on the convergence properties of CG and GMRES. The spatial resolution, $\Delta x = \frac{L}{K_x} = \frac{5}{6}$, is kept constant between all domain sizes investigated. Each simulation uses a time horizon of $T = 240$ time units, following a 'spin up' time of 1000 time units, $N = 20$ and the FGS preconditioning method. Increasing the domain size increases the size of the control variable, $\mathbf{q}$, which results in the size of the linear systems generated by the optimal control problem on each segment and the continuity constraints between consecutive segments becoming larger. The convergence properties of average number of CG iterations across all segments per GMRES iteration, panel (a), and GMRES, panel (b), for $L \in \{64, 128, 256, 512\}$ and $\alpha \in (1, 1 \times 10^{19})$ are presented in Figure 10. In general a small value of α leads to an increase in the number of CG iterations required for all domain sizes, Figure 10a. There is a very small influence that the domain size has on the average number of CG iterations. This behaviour is also



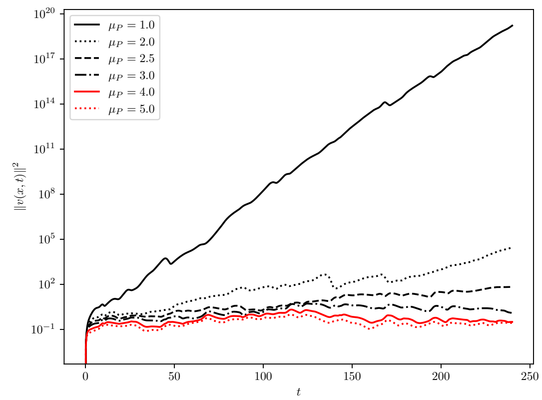
(a) $\frac{\mathrm{d}\bar{J}}{\mathrm{d}c}$ against α generated by OCS, solid line, and the curve fit derivative, dashed line.



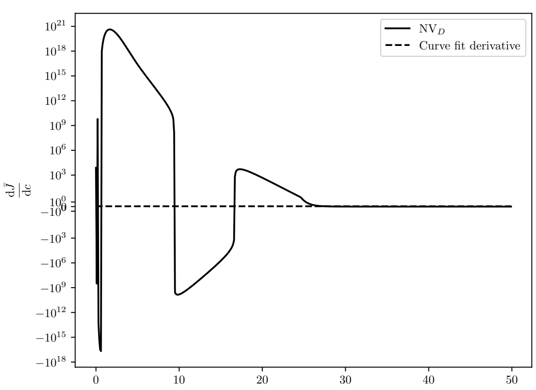
(b) $\|v(x,t)\|^2$ for $\alpha \in \{50, 500, 5\times10^3, 5\times10^4, 5\times10^5, 5\times10^6\}$



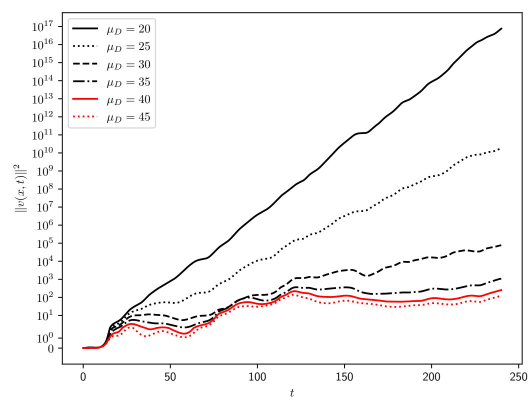
(c) $\frac{\mathrm{d}\bar{J}}{\mathrm{d}c}$ against μ_P generated by NV_P, solid line, and the curve fit derivative, dashed line.



(d) $\|v(x,t)\|^2$ for $\mu_P \in \{1, 2, 2.5, 3, 4, 5\}$



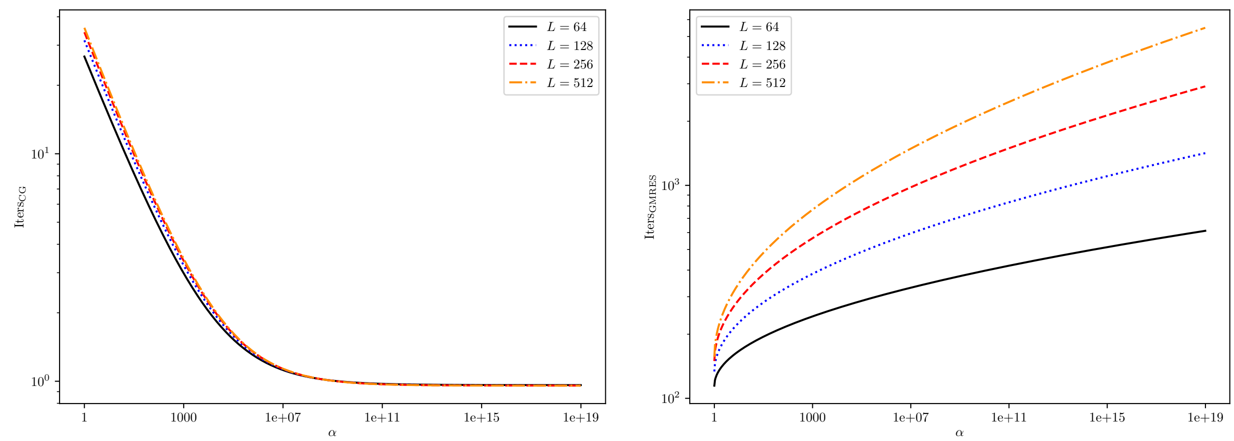
(e) $\frac{\mathrm{d}\bar{J}}{\mathrm{d}c}$ against μ_D generated by NV_D, solid line, and the curve fit derivative, dashed line.



(f) $\|v(x,t)\|^2$ for $\mu_D \in \{20, 25, 30, 35, 40, 45\}$

Fig. 9: Sensitivity $\frac{\mathrm{d}\bar{J}}{\mathrm{d}c}$, left column, and squared norm of the tangent solution $\|v(x,t)\|^2$, right column, for different values of tuning parameters generated by OCS, top row, NV_P, middle row, and NV_D, bottom row.

.



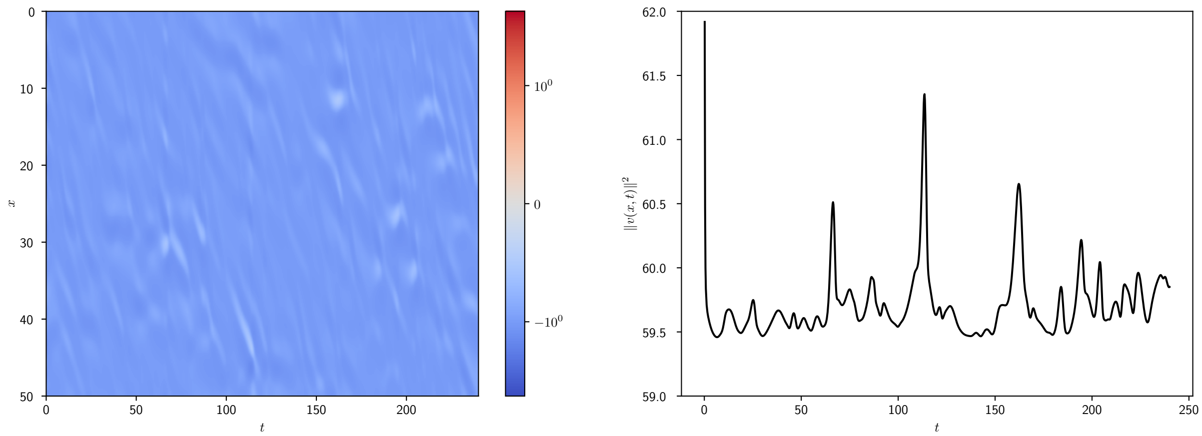
(a) (b)

Fig. 10: Average number of CG iterations per GMRES iteration, panel (a), and GMRES iterations, panel (b), for $L = 64$, black solid line, $L = 128$, blue dotted line, $L = 256$, red dashed line, and $L = 512$, orange dash dotted line, for a range of α values.

seen, for a different system, in Table 5.7 in Ref. [29]. In general increasing α leads to a larger number of GMRES iterations, Figure 10b. This is because the discontinuities between segments becomes larger when α is increased which results in more work being required to enforce continuity. Increasing the domain size increases the system solved by GMRES which results in a larger number of GMRES iterations and this relationship can be seen to be linear. From this analysis we expect that when applying these methods to larger systems, such as fluid flow problem, that the majority of the computational time is dedicated to the GMRES iterations.

5.5. Spatio-temporal structure of the tangent solutions. A MSS solution, left, on a time horizon of 240 time units following a 'spin up' time of 1000 time units using $K_x = 60$ and $N = 60$ from a random initial condition $u(x, t_s)$ is shown in Figure 11 along with the norm of the tangent solution, right, for clarity. The tangent solutions, left column, generated by OCS for the same conditions when $\alpha \in \{50, 5000, 5 \times 10^6\}$ and NV_p, right column, for $\mu_P \in \{1, 2.5, 5\}$ are shown in Figure 12. We refer the reader to Figures 9b and 9d for the solution norms of these. When $\alpha = 50$ there are some similarities between MSS, Figure 11a, and the OCS, Figure 12a. One thing to note is that OCS produces a tangent solution with a smaller norm, solid black line in Figure 9b, than MSS, Figure 11b. We argue that this is because the small value of α leads to a large amount of control being applied which damps the solution. OCS has zero norm which then rises, this is common between all OCS solutions which is due to the control stabilising the solution onto the approximate shadowing direction. Increasing α to 5000 results in a larger solution norm, dashed black line in Figure 9b. The large initial fluctuations in the tangent solution is caused by the reduced amount of control being applied when $\alpha = 5000$ compared to the previous case. We find that the tangent solution for $\alpha = 5 \times 10^6$, Figure 12c, does not initially resemble that of MSS, Figure 11a, but following a transient period it does show similar features. Finally, when $\alpha = 5 \times 10^6$ we find that the solution norm is significantly larger than MSS. This is caused by the control term being too small. This results in the tangent solution, Figure 12e, having no resemblance to MSS. The behaviour that increasing α leads to larger norm is not seen in the literature. This is because the equations that optimal control have been applied to in the literature, Refs [7, 8] for example, do not exhibit exponential growth whereas the tangent equation, in our case, does.

We find that when $\mu_P = 1$ the control term is unable to stabilise the tangent solution, which results in the solution norm growing exponentially, solid black line in Figure 9d. One other impact of this is that this solution, Figure 12b, has no similarities with that generated by MSS. Increasing μ_P to 2.5 leads to a solution where the control applied results in a better stabilised tangent solution, dashed black line in Figure 9d,



(a) (b)

Fig. 11: $v(x,t)$, panel (a), and $\|v(x,t)\|^2$, panel (b), of the tangent solution generated by MSS.

shown by the reduced norm. For this value of μ_P the tangent solution is better controlled, Figure 12d, and is starting to exhibit some features that resemble that of MSS. Increasing μ_P to 5, leads to solution that has been stabilised, dotted red line in Figure 9d. Further, this norm is smaller than that generated by MSS, Figure 11b. When $\mu_P = 5$ the tangent solution, Figure 12f, has some features of the MSS solution, Figure 11a, but there is still a large difference, seen in the large diagonal banding in the solution. We see similar behaviour for NV_D as we do for NV_P.

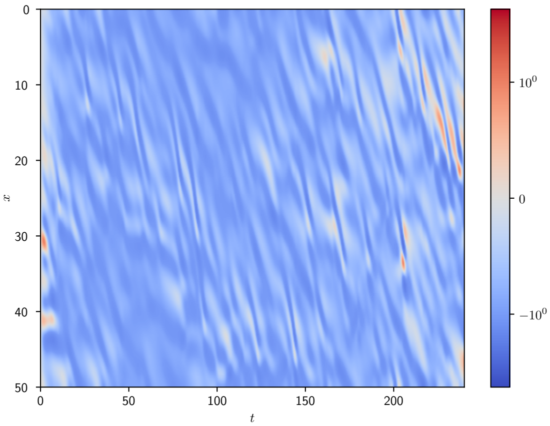
5.5.1. Quantifying the similarities between tangent solutions. To gain a better understanding of the similarities in the tangent solutions generated by OCS and NV with MSS we make use of

$$\theta_X(t) = \arccos\left(\frac{\langle v_{\text{MSS}}(x,t), v_X(x,t)\rangle}{\|v_{\text{MSS}}(x,t)\|\|v_X(x,t)\|}\right),$$

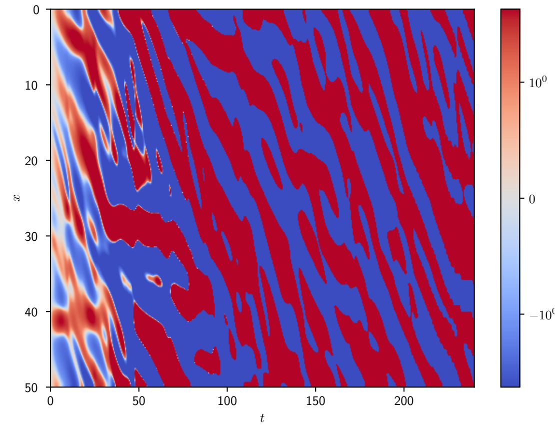
where $v_X(x,t)$ is the tangent solution generated by OCS or NV_P and $v_{\text{MSS}}(x,t)$ to MSS. We use the solution generated by MSS as a reference for the shadowing direction. Similar behaviour is seen between the NV_P and NV_D methods. A value of $\theta_X(t)$ close to zero shows that the solutions are well aligned with each other and are similar whereas values close to $\frac{\pi}{2}$ indicates the solutions are dissimilar. We show the comparison between MSS and OCS, θ_{OCS}, for a range of α values in panel (a) and the comparison between NV_P and MSS, θ_{NV_P}, for a range of μ_P values in panel (b) of Figure 13. When $\alpha \in \{5, 50, 500, 5000\}$ there is initially little similarity between the tangent solution generated by MSS and OCS. This is because there is a difference in the initial conditions between the two methods. This suggests that OCS stabilises the tangent solution to one similar to MSS. When $\alpha = 5$ the tangent solution gets close to the MSS solution rapidly and throughout the time horizon diverges away. This is because the exponential growth of perturbations are damped yet control is continued to be applied throughout the time horizon meaning the solution is controlled further. Increasing α to 50 leads to an increased time the OCS solution takes to get close to MSS. Once the solution reaches something resembling MSS little control is applied. This behaviour is seen when increasing α to 500 or 5000. If α is increased further then there is no resemblance between the tangent solutions generated by OCS and MSS. This is because the control applied is unable to control the growth of perturbations.

For NV_P, we find that regardless of which value of μ_P chosen there is no similarity with the tangent solution generated by MSS as the angle between solutions is, on average, $\frac{\pi}{2}$. This result is surprising even though the sensitivity values generated with both methods are in good agreement with each other.

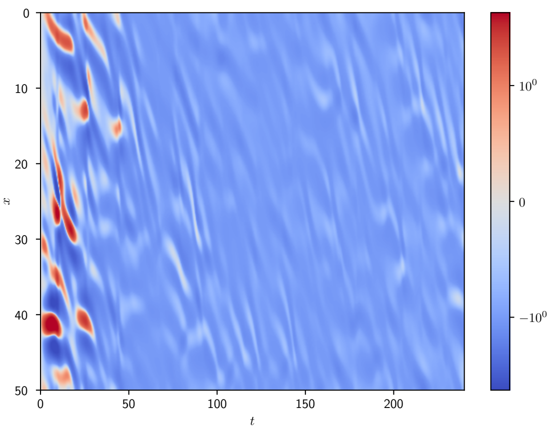
5.6. Spatio-temporal structures of the control produced by OCS and NV. We now show the control solution to aid in the explanation of the analysis derived for the tangent solutions. We show the spatio-temporal structures of $q(x,t)$ and its squared norm generated by OCS for $\alpha \in \{50, 5000, 5\times10^6\}$ in the



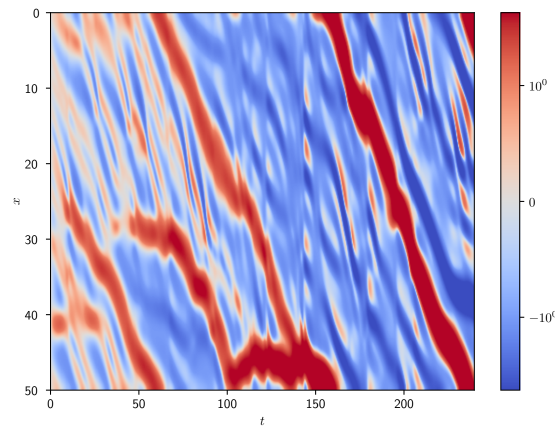
(a) $v(x,t)$ generated by OCS for $\alpha = 50$.



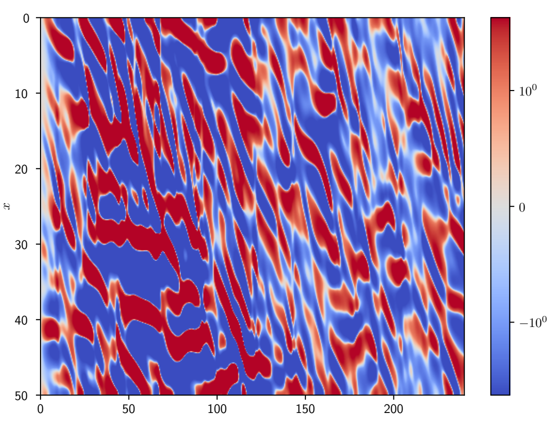
(b) $v(x,t)$ generated by NV_P for $\mu_P = 1$.



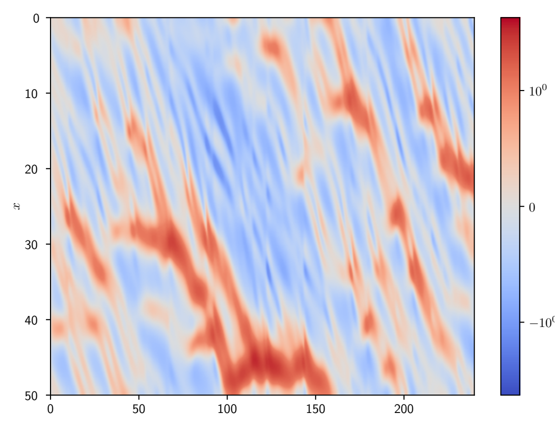
(c) $v(x,t)$ generated by OCS for $\alpha = 5000$.



(d) $v(x,t)$ generated by NV_P for $\mu_P = 2.5$.

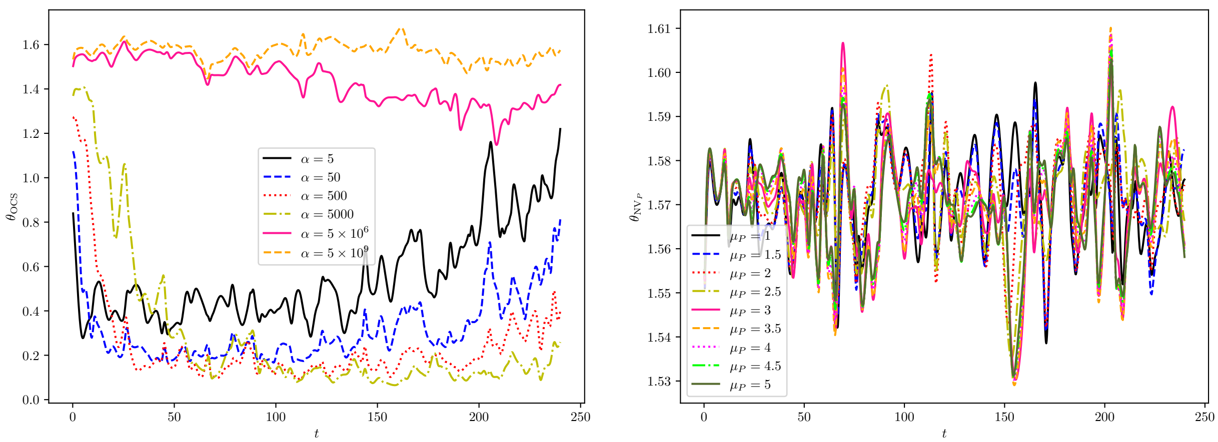


(e) $v(x,t)$ generated by OCS for $\alpha = 5 \times 10^6$.



(f) $v(x,t)$ generated by NV_P for $\mu_P = 5$.

Fig. 12: $v(x,t)$ generated by OCS, left column, for $\alpha \in \{50, 5000, 5 \times 10^6\}$ and NV_p, right column, for $\mu_P \in \{1, 2.5, 5\}$.



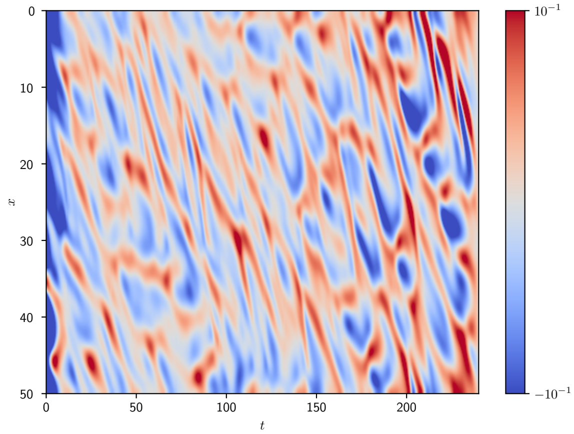
(a) (b)

Fig. 13: θ_{OCS} for $\alpha = 5$, solid black line, $\alpha = 50$, dashed blue, $\alpha = 500$, dotted red, $\alpha = 5000$, dash dotted yellow, $\alpha = 5 \times 10^6$, solid pink, and $\alpha = 5 \times 10^9$, dashed orange, panel (a), and θ_{NV_P} for $\mu_P = 1$, solid black line, $\mu_P = 1.5$, dashed blue, $\mu_P = 2$, dotted red, $\mu_P = 2.5$, dash dotted yellow, $\mu_P = 3$, solid pink, $\mu_P = 3.5$, dashed orange, $\mu_P = 4$, dotted magenta, $\mu_P = 4.5$, dash dotted lime, and $\mu_P = 5$, solid green, panel (b).

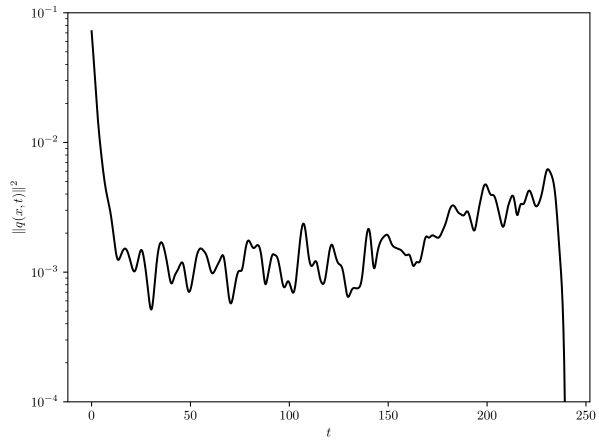
left and right columns, respectively, of Figure 14. We find that when $\alpha = 50$ the amount of control applied is constant throughout the time horizon, Figure 14b. The drop towards the end of the time horizon is due to the control being related to the co-state solution which has zero terminal condition. We also find that the control has very similar spatio-temporal structures in the entire time domain, as observed in the tangent solution, Figure 12a. Increasing α to 5000 there is exponential decay in the amount of control applied over the first half of the time horizon, Figure 14d. Following this the amount of control applied is constant. We find that the majority of the control is applied in the first half of the time horizon, Figure 14c. Finally, increasing α to 5×10^6 the control applied is several orders of magnitude lower than the other cases.

We repeat this for $\mu_P \in \{1, 2.5, 5\}$ using the same conditions as before. These results can be seen in Figure 15. As the control term, $q(x,t) = \mu_P \frac{\partial^2 v(x,t)}{\partial x^2}$, is linked to the tangent solution we find that when $\mu_P = 1$ the control is unable stabilise the solution and the amount applied increases exponentially throughout the time horizon, Figure 15b. We also find control is applied at shorter wavelengths, Figure 15a, compared to the tangent solution, Figure 12b. Increasing μ_P to 2.5 applies smaller amount of control than $\mu_P = 1$, Figure 15d. This may seem counter-intuitive, but as $q(x,t)$ is directly related to the tangent solution a control term that is able to stabilise the solution will result in less control being applied. Finally, increasing μ_P to 5 reduces the control applied even more, Figure 15f. We note that this behaviour is also seen for the NV_D case. We find that when $\mu_P = 5$, Figure 15e, the resulting control being applied has similar features as when $\alpha = 50$, Figure 14a. Further, NV_P applies orders of magnitude more control than OCS. As a final remark, the magnitude of the control applied by NV_P has high amplitude oscillations, Figure 15f, whereas that applied by OCS, Figure 14d, does not exhibit this behaviour.

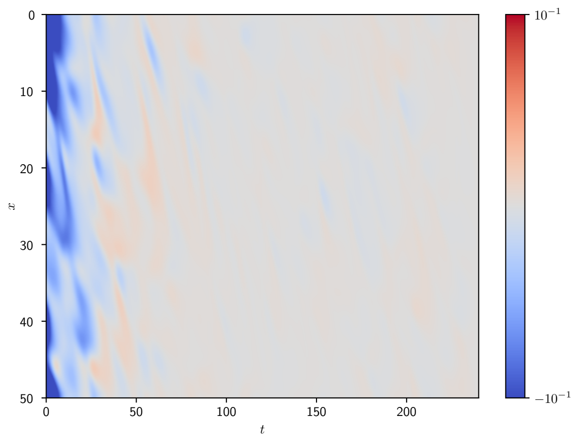
5.7. Wavenumber analysis of the tangent equation. In this Section we investigate the tangent equation, Section 5.2, on a wavenumber by wavenumber basis for the solutions generated by MSS, OCS and the two NV methods. We solve the tangent equation using spectral methods and, therefore, each term in Equation (5.2) can be represented in terms of its spectral decomposition. This is repeated for the terms, $C(x,t)$, $P(x,t)$, $D(x,t)$, $I(x,t)$ and $Q(x,t)$ in Equation (5.2) using the same approach. We then compute the time average of the absolute value of these on a wavenumber by wavenumber basis, *e.g.* $\overline{|P_k|}$. We compute an average of these terms using 50 samples from different initial conditions, $L = 50$, $K_x = 60$, $T = 240$ and a 'spin up' time of 1000 time units. We observe that results on different domain sizes collapse when the wavenumbers are scaled by $\frac{k}{L}$. Therefore, the analysis drawn for $L = 50$ is applicable to all domain sizes



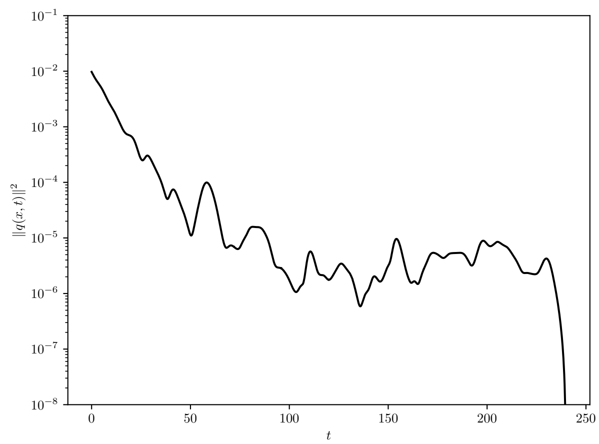
(a) $q(x,t)$ generated by OCS for $\alpha = 50$.



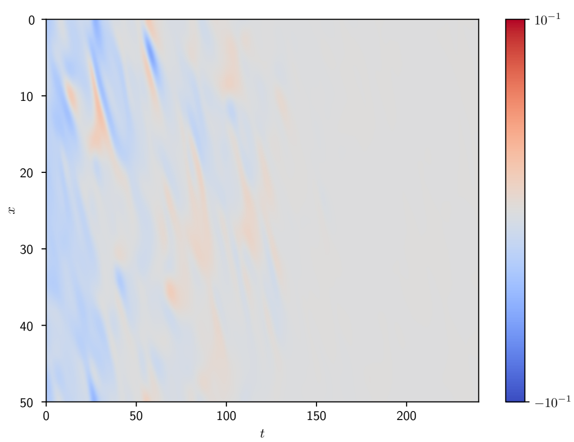
(b) $\|q(x,t)\|^2$ generated by OCS for $\alpha = 50$.



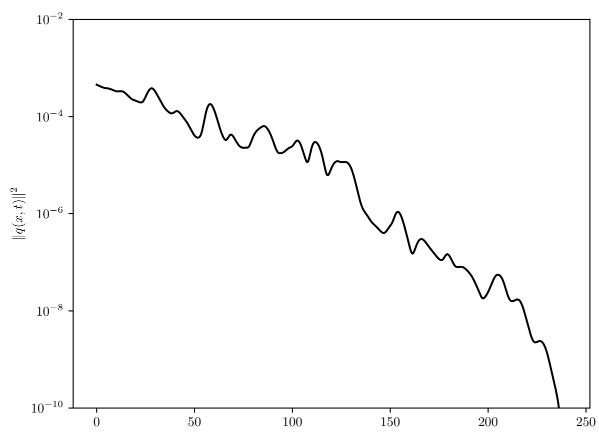
(c) $q(x,t)$ generated by OCS for $\alpha = 5000$.



(d) $\|q(x,t)\|^2$ generated by OCS for $\alpha = 5000$.

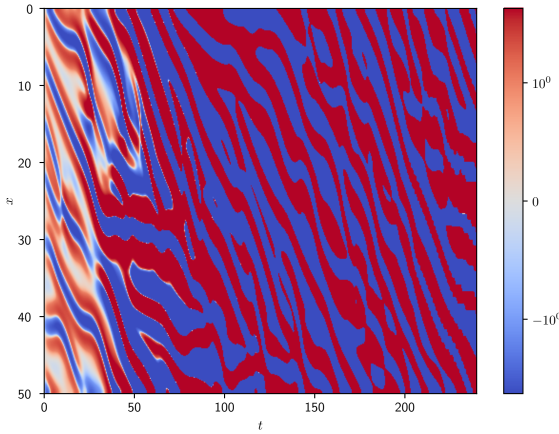


(e) $q(x,t)$ generated by OCS for $\alpha = 5 \times 10^6$.

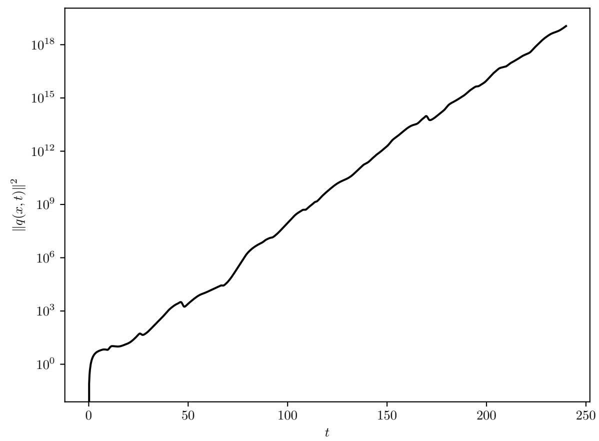


(f) $\|q(x,t)\|^2$ generated by OCS for $\alpha = 5 \times 10^6$.

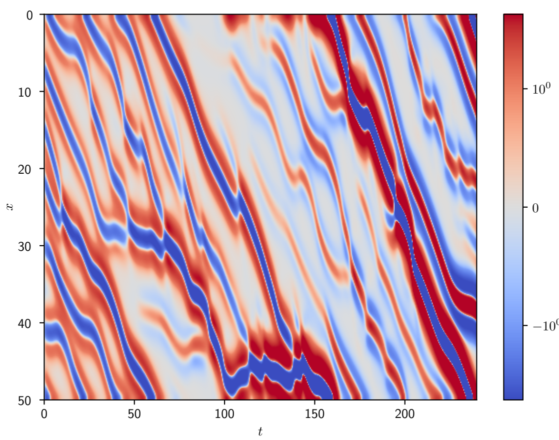
Fig. 14: $q(x,t)$, left column, and $\|q(x,t)\|^2$, right column, of the control solution generated by OCS for $\alpha = 50$, top row, $\alpha = 5000$, middle row, and $\alpha = 5 \times 10^6$, bottom row.



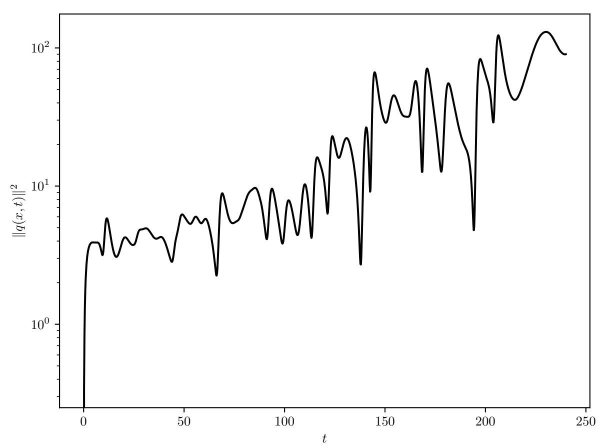
(a) $q(x,t)$ generated by NV_P for $\mu_P = 1$.



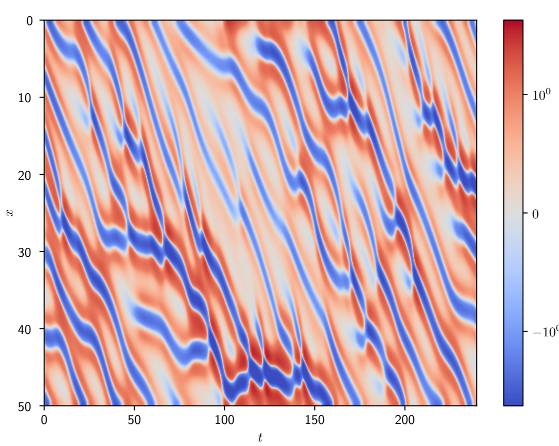
(b) $\|q(x,t)\|^2$ generated by NV_P for $\mu_P = 1$.



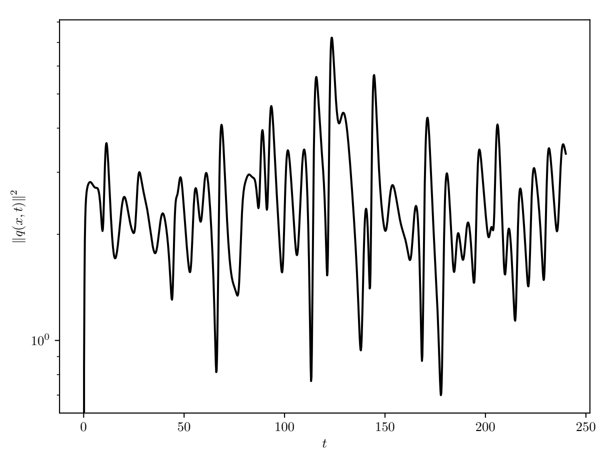
(c) $q(x,t)$ generated by NV_P for $\mu_P = 2.5$.



(d) $\|q(x,t)\|^2$ generated by NV_P for $\mu_P = 2.5$.



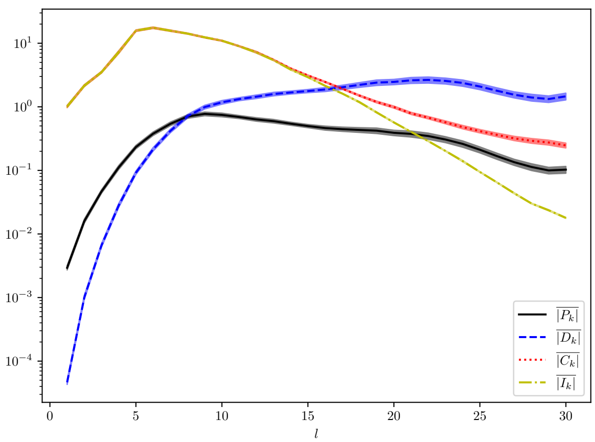
(e) $q(x,t)$ generated by NV_P for $\mu_P = 5$.



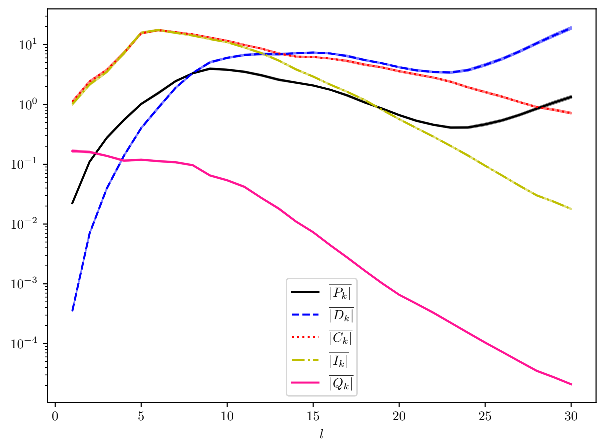
(f) $\|q(x,t)\|^2$ generated by NV_P for $\mu_P = 5$.

Fig. 15: $q(x,t)$, left column, and $\|q(x,t)\|^2$, right column, of the tangent solution generated by NV_P for $\mu_P = 1$, top row, $\mu_P = 2.5$, middle row, and $\mu_P = 5$, bottom row.

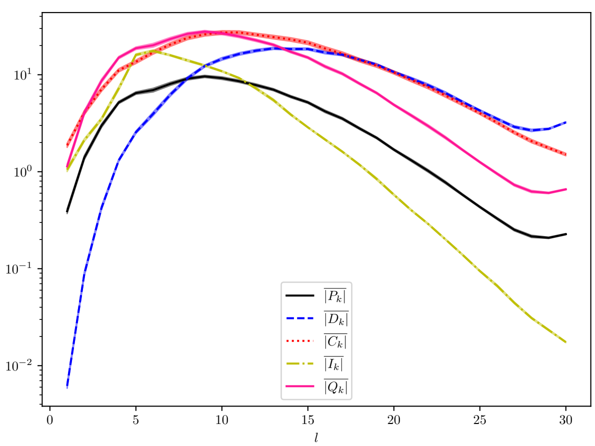
under the correct scaling. We also restrict ourselves to one value of the tuning parameter in OCS and the NV methods which are $\alpha = 500$, $\mu_P = 2.5$ and $\mu_D = 31.0$. We compute the mean and standard error of these results for 50 different initial conditions which is shown in Figure 16 for MSS, panel (a), OCS, panel (b), NV$_P$, panel (c), and NV$_D$, panel (d). One general trend that is observed is that production of tangent energy



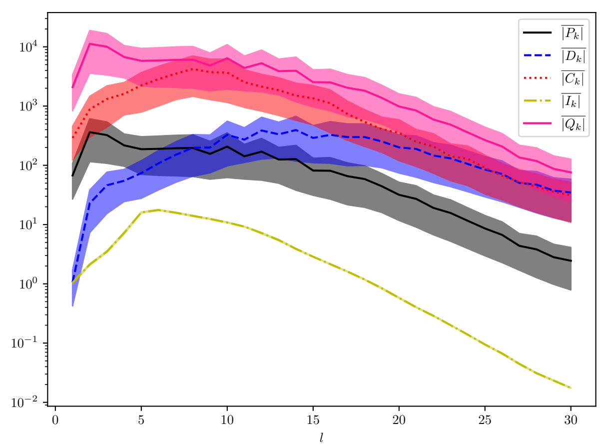
(a) Time averaged tangent spatial modes produced by MSS.



(b) Time averaged tangent spatial modes produced by OCS.



(c) Time averaged tangent spatial modes produced by NV$_P$.



(d) Time averaged tangent spatial modes produced by NV$_D$.

Fig. 16: Comparison of the mean and standard error of the time averaged tangent spatial modes produced by various methods over 50 samples. Panel (a) is the tangent solution generated by MSS, panel (b) by OCS, panel (c) by NV$_P$ and panel (d) by NV$_D$.

is dominated by the longest wavelengths. Further, dissipation of tangent energy is dominant at intermediate wavelengths, and that dissipation dominates production from around wavenumber $k = 8$. It is well known that the stable covariant Lyapunov vectors (CLVs) have stronger signatures as high wavenumbers, see Figure 3 in Ref. [78], and more stable CLVs have higher wavenumbers. We argue that the wavenumber at which dissipation dominates production is related to the most dominant wavenumber of the CLV with the smallest negative (closest to zero) LE. The inhomogeneous and convective terms both act on the longest length scales of the system and are reduced at shorter length scales for all cases.

The main difference between OCS and the two NV methods investigated is the length scales at which the control terms acts. We find that for OCS the control term acts predominantly at wavenumbers where production dominates dissipation and is reduced elsewhere. Further, the magnitude of the control term is several orders of magnitude lower than the other terms. This suggests that OCS applies control at length scales that require stabilising. We argue these long wavelengths are related to the unstable CLVs of the system, see Figure 3 in Ref. [78]. Both NV methods, on the other hand, have a large amount of control applied across all wavenumbers peaking at $k = 8$ and, therefore, damping across a large range of length scales, even when dissipation dominates production. This behaviour suggests that NV applies control that does not consider the unstable, neutral or stable sub-spaces of the tangent space, by contrast, OCS, we argue, applies control predominantly on the unstable sub-space. The wider implications of this is that knowing at what length scales production of tangent energy dominates dissipation or the length scales associated to the unstable sub-space could lead to the selection of a more targeted control term.

6. Conclusion.

We developed Optimal Control Shadowing, OCS, as a method to bridge the gap in understanding between the computationally expensive and accurate Shadowing methods, and the computationally cheap but less accurate numerical viscosity, NV, methods. The tangent OCS formulation, valid for a single parameter of interest, and the adjoint OCS formulation, valid for multiple parameters of interest, were derived. For large systems, such as fluid flows, solving the optimal control problem is a computationally challenging task, both in terms of storage and CPU usage. To partly overcome this limitation, a method for decomposing the time domain into segments such that the solutions can be distributed across multiple compute nodes was utilised. Finally, various algorithms and preconditioning methods for the numerical solution of the optimality conditions were presented.

The accuracy of the method depends on a single tuning parameter, α. Small values of α means control is inexpensive to apply and leads to the solution being over-damped resulting in inaccurate sensitivity values. In these conditions, the tangent solution derived from OCS was significantly different to that obtained using state-of-the-art shadowing methods such as MSS. Large values of α corresponds to control being expensive to apply, resulting in under-damping of the solution which exhibits exponential growth and leads to inaccurate sensitivities. Again, in these conditions strong differences with results from MSS were observed. Values of α between the over- or under- damped regions produced sensitivities in good agreement with the expected sensitivity values. These values also produced similar solutions to those generated by MSS. NV methods, on the other hand, did not show any similarities with MSS regardless of which tuning parameter value selected.

In general, fewer segments used to decompose the time domain results in an algorithm with smaller computational cost. In practice, however, fewer segments results in the memory requirements of the optimal control problem on each segment potentially being larger than the compute node they are allocated to. Therefore, the number of segments should be selected such that the memory requirements for the optimal control problem is below the memory limit of each node. The value of α controls the condition number of the optimal control problem and, as a consequence, the rate of convergence of iterative algorithms used for its solution. Among the preconditioners examines, Forward Gauss-Seidel (FGS) preconditioning produced the fastest convergence rate, followed by Backward Gauss-Seidel (BGS) preconditioning and then Jacobi preconditioning. The reason why FGS was fastest was due to an increased rate of "information" transfer between segments. The limitation of FGS and BGS is that the segments must be solved sequentially whereas Jacobi and no preconditioning can be solved in parallel. Therefore, the increased rate of convergence of the FGS method may have a slower wall-clock time than Jacobi due to the parallel nature of Jacobi preconditioning.

Through investigating the control term applied by OCS and both NV methods we found that there were significant differences. Namely, OCS applies the majority of the control initially and reduces the amount throughout the time horizon. Both NV methods, on the other hand, apply control proportional to the tangent solution and apply orders of magnitude more control than is applied by OCS. Through examination of the wavenumber decomposition of the control terms for OCS and NV it was determined that OCS applies the majority of the control in wavelengths where production of tangent energy dominates dissipation, whereas NV methods apply control across a wide range of length scales. We argue that this relationship is related to the CLVs and knowledge of them along with knowing the length scales at which production of tangent energy dominates dissipation could be utilised to derive NV terms that apply control more selectively.

Appendix A. Derivation for the closed form expression of η.

Following Ref. [14] we remove the explicit calculation of η in the following way. Firstly, the tangent equation, Equation (2.8a), can be rewritten utilising the state transition matrix, $\phi(t_1, t_2)$, where an introduction to the state transition matrix can be found in Ref. [41] and details are provided in Appendix F. The solution to the tangent equation at time t is

$$\mathbf{v}(t) = \boldsymbol{A}\mathbf{v}(t_s) + \left(\int_{t_s}^{t} \eta(\tau)\mathrm{d}\tau\right) \mathbf{f}|_t + \boldsymbol{B}\frac{\partial \mathbf{f}}{\partial p} + \boldsymbol{B}\mathbf{q}, \tag{A.1}$$

where

$$\boldsymbol{A}\mathbf{v}(t_s) = \phi(t_s, t)\mathbf{v}(t_s), \tag{A.2}$$

and

$$\boldsymbol{B}\square = \int_{t_s}^{t} \phi(\tau, t) \, \square|_\tau \, \mathrm{d}\tau. \tag{A.3}$$

By defining the solution to the tangent equation without the influence of the $\eta\mathbf{f}$ term as

$$\mathbf{v}'(t) = \boldsymbol{A}\mathbf{v}(t_s) + \boldsymbol{B}\frac{\partial \mathbf{f}}{\partial p} + \boldsymbol{B}\mathbf{q}, \tag{A.4}$$

Equation (A.1) can be rewritten as follows

$$\mathbf{v}(t) = \mathbf{v}'(t) + \left(\int_{t_s}^{t} \eta(\tau)\mathrm{d}\tau\right) \mathbf{f}|_t \,. \tag{A.5}$$

Taking the inner product of Equation (A.5) with $\mathbf{f}$ and utilising the constraint, Equation (2.8d), results in

$$\mathbf{v}^T\mathbf{f}\big|_t = 0 = \mathbf{v}'^T\mathbf{f}\big|_t + \left(\int_{t_s}^{t} \eta(\tau)\mathrm{d}\tau\right) \mathbf{f}^T\mathbf{f}\big|_t \,. \tag{A.6}$$

Manipulation of Equation (A.6) leads to the closed form expression of η,

$$\int_{t_s}^{t} \eta(\tau)\mathrm{d}\tau = -\left.\frac{\mathbf{v}'^T\mathbf{f}}{\mathbf{f}^T\mathbf{f}}\right|_t, \tag{A.7}$$

which can be combined with Equation (A.5) to give the solution to the tangent solution at time t

$$\mathbf{v}(t) = \mathbf{v}'(t) - \left.\left(\frac{\mathbf{v}'^T\mathbf{f}}{\mathbf{f}^T\mathbf{f}}\mathbf{f}\right)\right|_t. \tag{A.8}$$

Appendix B. Derivation for the closed form expression of ω. The closed form expression for ω by found by following a similar procedure to that of the derivation for the closed form expression for η. Firstly, Equation (2.8b) can be written in terms of the adjoint state transition matrix, $\phi^*(t_1, t_2)$, which details can again be found in Appendix F. The solution to the co-state equation at time t can be written

$$\boldsymbol{\lambda}(t) = -2\mathbf{C}\mathbf{v} + \mathbf{D}\boldsymbol{\lambda}(t_f) + \left(\int_{t}^{t_f} \omega(\tau)\mathrm{d}\tau\right) \mathbf{f}|_t \,, \tag{B.1}$$

where

$$\mathbf{C}\mathbf{v} = \int_{t}^{t_f} \phi^*(t, \tau)^{-1} \mathbf{v}(\tau)\mathrm{d}\tau, \tag{B.2}$$

and

$$\mathbf{D}\boldsymbol{\lambda}(t_f) = \phi^*(t, t_f)^{-1} \boldsymbol{\lambda}(t_f). \tag{B.3}$$

Defining the solution to the co-state equation without the influence of the $\omega\mathbf{f}$ as

$$\boldsymbol{\lambda}'(t) = -2\mathbf{C}\mathbf{v} + \mathbf{D}\boldsymbol{\lambda}(t_f), \tag{B.4}$$

leads to Equation (B.1) being

$$\boldsymbol{\lambda}(t) = \boldsymbol{\lambda}'(t) + \left(\int_t^{t_f} \omega(\tau)\mathrm{d}\tau \right) \mathbf{f}\big|_t . \tag{B.5}$$

By taking the inner product of Equation (B.5) with $\mathbf{f}$ and using the co-state constraint, Equation (2.8e), results in

$$\boldsymbol{\lambda}^T\mathbf{f}\Big|_t = 0 = \boldsymbol{\lambda}'^T\mathbf{f}\big|_t + \left(\int_t^{t_f} \omega(\tau)\mathrm{d}\tau \right) \mathbf{f}^T\mathbf{f}\big|_t . \tag{B.6}$$

Manipulation of Equation (B.6) leads to the closed form expression of ω,

$$\int_t^{t_f} \omega(\tau)\mathrm{d}\tau = - \left. \frac{\boldsymbol{\lambda}'^T\mathbf{f}}{\mathbf{f}^T\mathbf{f}} \right|_t , \tag{B.7}$$

and can be substituted into Equation (B.5) to give the co-state solution at time t

$$\boldsymbol{\lambda}(t) = \boldsymbol{\lambda}'(t) - \left. \left(\frac{\boldsymbol{\lambda}'^T\mathbf{f}}{\mathbf{f}^T\mathbf{f}} \mathbf{f} \right) \right|_t . \tag{B.8}$$

Appendix C. Derivation of the closed form expression of $\hat{\eta}$. The adjoint tangent equation, Equation (2.16a), can be written using its state transition matrix, $\hat{\phi}(t_1, t_2)$. The solution to the adjoint tangent equation at time t is

$$\hat{\mathbf{v}}(t) = \hat{\boldsymbol{A}}\hat{\mathbf{v}}(t_s) + \left(\int_{t_s}^{t} \hat{\eta}(\tau)\mathrm{d}\tau \right) \mathbf{f}\big|_t + \hat{\boldsymbol{B}}\hat{\mathbf{q}}, \tag{C.1}$$

where

$$\hat{\boldsymbol{A}}\hat{\mathbf{v}}(t_s) = \hat{\phi}(t_s, t)\hat{\mathbf{v}}(t_s), \tag{C.2}$$

and

$$\hat{\boldsymbol{B}}\hat{\mathbf{q}} = \int_{t_s}^{t} \hat{\phi}(\tau, t)\, \hat{\mathbf{q}}\big|_\tau \,\mathrm{d}\tau. \tag{C.3}$$

By defining the solution to the tangent equation without the influence of the $\hat{\eta}\mathbf{f}$ term as

$$\hat{\mathbf{v}}'(t) = \hat{\boldsymbol{A}}\hat{\mathbf{v}}(t_s) + \hat{\boldsymbol{B}}\hat{\mathbf{q}}, \tag{C.4}$$

Equation (C.1) can be rewritten as

$$\hat{\mathbf{v}}(t) = \hat{\mathbf{v}}'(t) + \left(\int_{t_s}^{t} \hat{\eta}(\tau)\mathrm{d}\tau \right) \mathbf{f}\big|_t . \tag{C.5}$$

Taking the inner product of Equation (C.5) with $\mathbf{f}$ and utilising constraint Equation (2.16d) results in

$$\hat{\mathbf{v}}^T\mathbf{f}\big|_t = 0 = \hat{\mathbf{v}}'^T\mathbf{f}\big|_t + \left(\int_{t_s}^{t} \hat{\eta}(\tau)\mathrm{d}\tau \right) \mathbf{f}^T\mathbf{f}\big|_t . \tag{C.6}$$

Manipulation of Equation (C.6) leads to the closed form expression for $\hat{\eta}$,

$$\int_{t_s}^{t} \hat{\eta}(\tau)\mathrm{d}\tau = - \left. \frac{\hat{\mathbf{v}}'^T\mathbf{f}}{\mathbf{f}^T\mathbf{f}} \right|_t , \tag{C.7}$$

which can be substituted combined with Equation (C.4) to give the adjoint tangent solution at time t

$$\hat{\mathbf{v}}(t) = \hat{\mathbf{v}}'(t) - \left. \left(\frac{\hat{\mathbf{v}}'^T\mathbf{f}}{\mathbf{f}^T\mathbf{f}} \mathbf{f} \right) \right|_t . \tag{C.8}$$

Appendix D. Derivation of the closed form expression of $\hat{\omega}$. The adjoint co-state equation, Equation (2.16b), can be written in terms of its adjoint state transition matrix, $\hat{\phi}^*(t_1, t_2)$, resulting in

$$\hat{\boldsymbol{\lambda}}(t) = -\hat{\boldsymbol{C}}\left[2\hat{\mathbf{v}}\right] - \hat{\boldsymbol{C}}\left[\frac{1}{T}\frac{\partial J}{\partial \mathbf{u}}^{T}\right] + \hat{\boldsymbol{D}}\boldsymbol{\lambda}(t_f) + \left(\int_t^{t_f} \hat{\omega}(\tau)\mathrm{d}\tau\right) \mathbf{f}\big|_t\,, \tag{D.1}$$

where

$$\hat{\boldsymbol{C}}\Box = \int_t^{t_f} \hat{\phi}^*(t,\tau)^{-1}\, \Box|_\tau\, \mathrm{d}\tau, \tag{D.2}$$

and

$$\hat{\boldsymbol{D}}\boldsymbol{\lambda}_{t_f} = \hat{\phi}^*\left(t, t_f\right)^{-1} \hat{\boldsymbol{\lambda}}(t_f). \tag{D.3}$$

Defining the solution to the co-state equation without the influence of the $\hat{\omega}\mathbf{f}$ as

$$\hat{\boldsymbol{\lambda}}'(t) = -\hat{\boldsymbol{C}}\left[2\hat{\mathbf{v}}\right] - \hat{\boldsymbol{C}}\left[\frac{1}{T}\frac{\partial J}{\partial \mathbf{u}}^{T}\right] + \hat{\mathbf{D}}\hat{\boldsymbol{\lambda}}(t_f), \tag{D.4}$$

Equation (D.1) can be rewritten as

$$\hat{\boldsymbol{\lambda}}(t) = \hat{\boldsymbol{\lambda}}'(t) + \left(\int_t^{t_f} \hat{\omega}(\tau)\mathrm{d}\tau\right) \mathbf{f}\big|_t\,. \tag{D.5}$$

By taking the inner product of Equation (D.5) with $\mathbf{f}$ and using the co-state constraint, Equation (2.16e), results in

$$\hat{\boldsymbol{\lambda}}^T\mathbf{f}\Big|_t - \frac{1}{T}\left(J - \bar{J}\right)\Big|_t = 0 = \hat{\boldsymbol{\lambda}}'^T\mathbf{f}\Big|_t + \left(\int_t^{t_f} \hat{\omega}(\tau)\mathrm{d}\tau\right) \mathbf{f}^T\mathbf{f}\big|_t - \frac{1}{T}\left(J - \bar{J}\right)\Big|_t. \tag{D.6}$$

The closed form expression for $\hat{\omega}$ is

$$\int_t^{t_f} \hat{\omega}(\tau)\mathrm{d}\tau = -\frac{\hat{\boldsymbol{\lambda}}'^T\mathbf{f}}{\mathbf{f}^T\mathbf{f}}\bigg|_t + \frac{1}{T}\frac{J - \bar{J}}{\mathbf{f}^T\mathbf{f}}\bigg|_t. \tag{D.7}$$

and can be substituted into Equation (D.5) to give the adjoint co-state solution at time t

$$\hat{\boldsymbol{\lambda}}(t) = \hat{\boldsymbol{\lambda}}'(t) - \frac{\hat{\boldsymbol{\lambda}}'^T\mathbf{f}}{\mathbf{f}^T\mathbf{f}}\mathbf{f}\bigg|_t + \frac{1}{T}\frac{J - \bar{J}}{\mathbf{f}^T\mathbf{f}}\mathbf{f}\bigg|_t. \tag{D.8}$$

Appendix E. Analytical structure of the linear system from time domain decomposition. The analytical structure of $\mathbf{F}$ is

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_{0,0} & \mathbf{F}_{0,1} & & & & & \\ \mathbf{F}_{1,0} & & & & & & \\ & & \ddots & & & & \\ & & \mathbf{F}_{j,j-1} & \mathbf{F}_{j,j} & \mathbf{F}_{j,j+1} & & \\ & & & & \ddots & & \\ & & & & & & \mathbf{F}_{N-2,N-1} \\ & & & & & \mathbf{F}_{N-1,N-2} & \mathbf{F}_{N-1,N-1} \end{pmatrix}, \tag{E.1}$$

with

$$\mathbf{F}_{0,0} = \begin{pmatrix} \mathbf{I} & \mathbf{A}_0 \\ 0 & 2\alpha\mathbf{I} + 2\mathbf{C}_0\mathbf{B}_0 \end{pmatrix}, \tag{E.2}$$

$$\mathbf{F}_{0,1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -\mathbf{D}_0 \end{pmatrix}, \tag{E.3}$$

$$\mathbf{F}_{1,0} = \begin{pmatrix} -\mathbf{A}_1 & 0 \\ 2\mathbf{C}_1\mathbf{A}_1 & 0 \\ 2\mathbf{C}_1\mathbf{A}_1 & 0 \end{pmatrix}, \tag{E.4}$$

$$\mathbf{F}_{j,j-1} = \begin{pmatrix} -\mathbf{A}_j & 0 & 0 \\ 2\mathbf{C}_j\mathbf{A}_j & 0 & 0 \\ 2\mathbf{C}_j\mathbf{A}_j & 0 & 0 \end{pmatrix}, \tag{E.5}$$

$$\mathbf{F}_{j,j} = \begin{pmatrix} \mathbf{I} & -\mathbf{B}_j & 0 \\ 0 & 2\alpha\mathbf{I} + 2\mathbf{C}_j\mathbf{B}_j & 0 \\ 0 & 2\mathbf{C}_j\mathbf{B}_j & \mathbf{I} \end{pmatrix}, \tag{E.6}$$

$$\mathbf{F}_{j,j+1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -\mathbf{D}_j \\ 0 & 0 & -\mathbf{D}_j \end{pmatrix}, \tag{E.7}$$

$$\mathbf{F}_{N-2,N-1} = \begin{pmatrix} 0 & 0 \\ 0 & -\mathbf{D}_{N-2} \\ 0 & -\mathbf{D}_{N-2} \end{pmatrix}, \tag{E.8}$$

$$\mathbf{F}_{N-1,N-2} = \begin{pmatrix} 2\mathbf{C}_{N-1}\mathbf{A}_{N-1} & 0 & 0 \\ 2\mathbf{C}_{N-1}\mathbf{A}_{N-1} & 0 & 0 \end{pmatrix}, \tag{E.9}$$

$$\mathbf{F}_{N-1,N-1} = \begin{pmatrix} 2\alpha\mathbf{I} + 2\mathbf{C}_{N-1}\mathbf{B}_{N-1} & 0 \\ 2\mathbf{C}_{N-1}\mathbf{B}_{N-1} & \mathbf{I} \end{pmatrix}. \tag{E.10}$$

The subscripts, $\mathbf{A}_j$, represent the operator $\mathbf{A}$ on segment j. Finally, the analytical structure of $\mathbf{c}$ is

$$\mathbf{c} = \left(\mathbf{c}_0, \ldots, \mathbf{c}_j, \ldots, \mathbf{c}_{N-1}\right)^T, \tag{E.11}$$

with

$$\mathbf{c}_0 = \begin{pmatrix} \mathbf{B}_0 \frac{\partial \mathbf{f}}{\partial p} + \left(\int_{t_0}^{t} \eta(\tau)\mathrm{d}\tau\right)\mathbf{f}(t) \\ 2\mathbf{C}_0\left[\mathbf{B}_0\frac{\partial \mathbf{f}}{\partial p} + \left(\int_{t_0}^{t_1}\eta(\tau)\mathrm{d}\tau\right)\mathbf{f}(t_1)\right] - \left[\int_{t_0}^{t_1}\omega_0(\tau)\mathrm{d}\tau\right]\mathbf{f}(t_1) \end{pmatrix}, \tag{E.12}$$

$$\mathbf{c}_j = \begin{pmatrix} \mathbf{B}_j \frac{\partial \mathbf{f}}{\partial p} + \left(\int_{t_j}^{t_{j+1}} \eta(\tau)\mathrm{d}\tau\right)\mathbf{f}(t) \\ 2\mathbf{C}_j\left[\mathbf{B}_j\frac{\partial \mathbf{f}}{\partial p} + \left(\int_{t_j}^{t}\eta(\tau)\mathrm{d}\tau\right)\mathbf{f}(t)\right] - \left[\int_{t_j}^{t}\omega_j(\tau)\mathrm{d}\tau\right]\mathbf{f}(t) \\ 2\mathbf{C}_j\left[\mathbf{B}_j\frac{\partial \mathbf{f}}{\partial p} + \left(\int_{t_j}^{t_{j+1}}\eta(\tau)\mathrm{d}\tau\right)\mathbf{f}(t_{j+1})\right] - \left[\int_{t_j}^{t_{j+1}}\omega_j(\tau)\mathrm{d}\tau\right]\mathbf{f}(t_{j+1}) \end{pmatrix}, \tag{E.13}$$

$$\mathbf{c}_{N-1} = \begin{pmatrix} 2\mathbf{C}_{N-1}\left[\mathbf{B}_{N-1}\frac{\partial \mathbf{f}}{\partial p} + \left(\int_{t_{N-1}}^{t}\eta(\tau)\mathrm{d}\tau\right)\mathbf{f}(t)\right] - \left[\int_{t_{N-1}}^{t}\omega_{N-1}(\tau)\mathrm{d}\tau\right]\mathbf{f}(t) \\ 2\mathbf{C}_{N-1}\left[\mathbf{B}_{N-1}\frac{\partial \mathbf{f}}{\partial p} + \left(\int_{t_{N-1}}^{t_f}\eta(\tau)\mathrm{d}\tau\right)\mathbf{f}(t_f)\right] - \left[\int_{t_{N-1}}^{t_f}\omega_{N-1}(\tau)\mathrm{d}\tau\right]\mathbf{f}(t_f) \end{pmatrix}. \tag{E.14}$$

Appendix F. Properties of the state transition matrix. Given a system of the form

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{A}\mathbf{v} + \mathbf{g}, \tag{F.1}$$

where $\mathbf{v} \in \mathbb{R}^n$ is the state, $\mathbf{A} \in \mathcal{R}^{n\times n}$ and $\mathbf{g} \in \mathbb{R}^n$ are known and continuous in time, the homogeneous equation is defined as

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = \mathbf{A}\mathbf{v}, \tag{F.2}$$

where the is a linear combination of a set of solutions $\mathbf{v}_j$ that satisfy Equation (F.2). Define the fundamental matrix solution, $\mathbf{X}|_t \in \mathbb{R}^{n\times n}$, as a matrix whose columns are solutions to Equation (F.2). A general solution is $\mathbf{v}(t) = \mathbf{X}|_t\,\mathbf{c}$, where $\mathbf{c}$ is a vector of arbitrary weights. If $\mathbf{v}(t_1)$ is known then $\mathbf{c} = \mathbf{X}^{-1}\mathbf{v}\big|_{t_1}$. One other important feature is $\mathbf{X}|_{t_0} = \mathbf{I}$. The difference between the solution at time t_1 and t_2 of Equation (F.2) and Equation (F.1) is the influence of $\mathbf{g}$ which can be written as

$$\mathbf{X}^{-1}\mathbf{v}\big|_{t_1}^{t_2} = \int_{t_1}^{t_2} \mathbf{X}^{-1}\mathbf{g}\big|_\tau\,\mathrm{d}\tau. \tag{F.3}$$

The solution to Equation (F.1) at t_2 is

$$\mathbf{v}(t_2) = \mathbf{X}|_{t_2}\left[\mathbf{X}^{-1}\mathbf{v}\big|_{t_1} + \int_{t_1}^{t_2} \mathbf{X}^{-1}\mathbf{g}\big|_\tau\,\mathrm{d}\tau\right]. \tag{F.4}$$

The state transition matrix is defined as

$$\phi(t_1,t_2) = \mathbf{X}|_{t_2}\,\mathbf{X}^{-1}\big|_{t_1} \tag{F.5}$$

and can be thought of moving the solution $\mathbf{v}(t_1)$ to $\mathbf{v}(t_2)$ under the influence of Equation (F.2) and is written $\mathbf{v}(t_2) = \phi(t_1,t_2)\mathbf{v}(t_1)$. The state transition matrix also satisfies Equation (F.2) and has the following relationships

$$\phi(t,t) = I \tag{F.6}$$

$$\phi(t_1,t_2)\phi(t_0,t_1) = \phi(t_0,t_2), \tag{F.7}$$

$$\phi(t_1,t_2) = \phi(t_2,t_1)^{-1} \tag{F.8}$$

and

$$\phi^*(t_1,t_2) = \phi(t_1,t_2)^{-T}, \tag{F.9}$$

where $\phi^*(t_1,t_2)$ is the adjoint state transition matrix.

References.

[1] Rafail V Abramov and Andrew J Majda. "New Approximations and Tests of Linear Fluctuation-Response for Chaotic Nonlinear Forced-Dissipative Dynamical Systems". In: *Journal of Nonlinear Science* 18.3 (2008), pp. 303–341. URL: https://doi.org/10.1007/s00332-007-9011-9.

[2] Usman Ali and Yorai Wardi. "Multiple Shooting Technique for Optimal Control Problems with Application to Power Aware Networks". In: *IFAC-PapersOnLine* 48.27 (2015), pp. 286–290. ISSN: 24058963.

[3] Brian D. O. Anderson and John B. Moore. *Optimal Control: Linear Quadratic Methods*. USA: Prentice-Hall, Inc., 1990.

[4] Andrew T. Barker and Martin Stoll. "Domain decomposition in time for PDE-constrained optimization". In: *Computer Physics Communications* 197 (2015), pp. 136–143.

[5] Martin Berggren and Matthias Heinkenschloss. “Parallel Solution of Optimal-Control Problems by Time-Domain Decomposition”. In: 1997.

[6] John T. Betts. *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming.* Society for Industrial and Applied Mathematics, 2010.

[7] Thomas R. Bewley, Paolo Luchini, and Jan Pralits. “Methods for solution of large optimal control problems that bypass open-loop model reduction”. In: *Meccanica* 51.12 (2016), pp. 2997–3014.

[8] Thomas R. Bewley, Parviz Moin, and Roger Temam. “DNS-based predictive control of turbulence: An optimal benchmark for feedback algorithms”. In: *Journal of Fluid Mechanics* 447 (2001), pp. 179–225.

[9] Manav Bhatia and David Makhija. “Sensitivity analysis of time-averaged quantities of chaotic systems”. In: *AIAA Journal* 57.5 (2019), pp. 2088–2099.

[10] Patrick J. Blonigan. “Adjoint sensitivity analysis of chaotic dynamical systems with non-intrusive least squares shadowing”. In: *Journal of Computational Physics* 348 (2017), pp. 803–826.

[11] Patrick J. Blonigan, Steven A. Gomez, and Qiqi Wang. “Least squares shadowing for sensitivity analysis of turbulent fluid flows”. In: *52nd Aerospace Sciences Meeting* (Jan. 2014), pp. 1–23.

[12] Patrick J. Blonigan and Qiqi Wang. “Least squares shadowing sensitivity analysis of a modified Kuramoto-Sivashinsky equation”. In: *Chaos, Solitons and Fractals* 64.1 (2014), pp. 16–25.

[13] Patrick J. Blonigan and Qiqi Wang. “Multigrid-in-time for sensitivity analysis of chaotic dynamical systems”. In: *Numerical Linear Algebra with Applications* 24 (2017), pp. 1–27.

[14] Patrick J. Blonigan and Qiqi Wang. “Multiple shooting shadowing for sensitivity analysis of chaotic dynamical systems”. In: *Journal of Computational Physics* 354 (2018), pp. 447–475.

[15] Patrick J. Blonigan et al. “A non-intrusive algorithm for sensitivity analysis of chaotic flow simulations”. In: *AIAA SciTech Forum - 55th AIAA Aerospace Sciences Meeting* (2017), pp. 1–8.

[16] Patrick J. Blonigan et al. “Least-squares shadowing sensitivity analysis of chaotic flow around a two-dimensional airfoil”. In: *AIAA Journal* 56.2 (2018), pp. 658–672.

[17] Patrick J. Blonigan et al. “Toward a chaotic adjoint for LES”. In: *arXiv* (2017).

[18] Patrick J. Blonigan et al. “Towards adjoint sensitivity analysis of statistics in turbulent flow simulation”. In: *2012 Summer Program, Center for Turbulence Research, Stanford Univ., Stanford, CA, 2012* (2012), pp. 229–239.

[19] Hans G. Bock and Karl J. Plitt. “Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems.” In: *IFAC Proceedings Series* 17.2 (1985), pp. 1603–1608.

[20] J. Frédéric Bonnans. “The shooting approach to optimal control problems”. In: *IFAC Proceedings Volumes (IFAC-PapersOnline)* 11 (2013), pp. 281–292.

[21] Arthur E. Bryson and Yu-Chi Ho. *Applied Optimal Control*. New York: John Wiley and Sons, 1975.

[22] Nisha Chandramoorthy, Luca Magri, and Qiqi Wang. “Variational optimization and data assimilation in chaotic time-delayed systems with automatic-differentiated shadowing sensitivity”. In: (2020). URL: http://arxiv.org/abs/2011.08794.

[23] Nisha Chandramoorthy and Qiqi Wang. “On the probability of finding nonphysical solutions through shadowing”. In: *Journal of Computational Physics* 440 (2021), p. 110389.

[24] Nisha Chandramoorthy et al. “Feasibility analysis of ensemble sensitivity computation in turbulent flows”. In: *AIAA Journal* 57 (10 2019), pp. 4514–4526.

[25] Yong Chang and S. Scott Collis. “Active control of turbulent channel flows based on Large Eddy simulation”. In: *Proceedings of the 1999 3rd ASME/JSME Joint Fluids Engineering Conference, FEDSM’99, San Francisco, California, USA, 18-23 July 1999 (CD-ROM)* (1999).

[26] Mario Chater et al. “Least squares shadowing method for sensitivity analysis of differential equations”. In: *SIAM Journal on Numerical Analysis* 55.6 (2017), pp. 3030–3046.

[27] Haecheon Choi, Michael Hinze, and Karl Kunisch. “Instantaneous control of backward-facing step flows”. In: *Applied Numerical Mathematics* 31 (2 1999), pp. 133–158.

[28] Agata Comas. “Time Domain Decomposition Methods for Second Order Linear Quadratic Optimal Control Problems”. Master’s Thesis. Rice University, 2004.

[29] Agata Comas. “Time-Domain Decomposition Preconditioners for the Solution of Discretized Parabolic Optimal Control Problems”. PhD Thesis. Rice University, 2005.

[30] Predrag Cvitanović, Ruslan L. Davidchack, and Evangelos Siminos. “On the State Space Geometry of the Kuramoto–Sivashinsky Flow in a Periodic Domain”. In: *SIAM Journal on Applied Dynamical Systems* 9 (1 Jan. 2010), pp. 1–33.

[31] Xiaodi Deng. “A Parallel-In-Time Gradient-Type Method Doctor of Philosophy A Parallel-In-Time Gradient-Type Method For Optimal Control Problems”. PhD thesis. Rice University, 2017.
[32] Byron DeVries et al. “Parallel Implementations of FGMRES for Solving Large, Sparse Non-symmetric Linear Systems”. In: *Procedia Computer Science* 18 (2013). 2013 International Conference on Computational Science, pp. 491–500. ISSN: 1877-0509.
[33] Jocelyne Erhel. “A parallel GMRES version for general sparse matrices”. In: *ETNA* 3 (Aug. 1998).
[34] Gregory L. Eyink, Tom W. N. Haine, and Daniel J. Lea. “Ruelle’s linear response formula, ensemble adjoint schemes and Lévy flights”. In: *Nonlinearity* 17.5 (2004), pp. 1867–1889.
[35] Stefano Galatolo and Isaia Nisoli. “An Elementary Approach to Rigorous Approximation of Invariant Measures”. In: *SIAM Journal on Applied Dynamical Systems* 13.2 (2014), pp. 958–985. DOI: 10.1137/130911044. URL: https://doi.org/10.1137/130911044.
[36] Anirban Garai and Scott M Murman. “Stabilization of the Adjoint for Turbulent Flows”. In: *AIAA Journal* 59.6 (June 2021), pp. 2001–2013.
[37] Sebastian Götschel and Michael L. Minion. “An efficient parallel-in-time method for optimization with parabolic pdes”. In: *SIAM Journal on Scientific Computing* 41.6 (2019), pp. C603–C626.
[38] Sebastian Götschel and Michael L. Minion. “Parallel-in-time for parabolic optimal control problems using pfasst”. In: *Lecture Notes in Computational Science and Engineering* 125.May (2018), pp. 363–371.
[39] Lars Grüne. “Numerical Methods for Nonlinear Optimal Control Problems”. In: *Encyclopedia of Systems and Control*. London: Springer London, 2019, pp. 1–8.
[40] Stefanie Günther, Nicolas R. Gauger, and Jacob B. Schroder. “A non-intrusive parallel-in-time approach for simultaneous optimization with unsteady PDEs”. In: *Optimization Methods and Software* 34.6 (2019), pp. 1306–1321.
[41] Jack K. Hale. *Ordinary Differential Equations*. Dover Books on Mathematics Series. Dover Publications, 2009. ISBN: 9780486472119.
[42] Matthias Heinkenschloss. “A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems”. In: *Journal of Computational and Applied Mathematics* 173.1 (2005), pp. 169–198.
[43] Michael Hinze and Karl Kunisch. “On suboptimal control strategies for the Navier-Stokes equations”. In: *ESAIM: Proceedings* 4 (1998), pp. 181–198.
[44] Michael Hinze and Stefan Volkwein. “Analysis of instantaneous control for the Burgers equation”. In: *Nonlinear Analysis, Theory, Methods and Applications* 50 (1 2002), pp. 1–26.
[45] L. Stephen Hou and Y. Yan. “Dynamics and approximations of a velocity tracking problem for the Navier-Stokes flows with piecewise distributed controls”. In: *SIAM Journal on Control and Optimization* 35 (6 1997).
[46] Oscar Hugues-Salas and Stephen P. Banks. “Optimal control of nonhomogeneous chaotic systems”. In: *IFAC Proceedings Volumes (IFAC-PapersOnline)* 1.PART 1 (2006), pp. 203–208.
[47] Francisco Huhn and Luca Magri. “Optimisation of chaotically perturbed acoustic limit cycles”. In: *Nonlinear Dynamics* 100.2 (2020), pp. 1641–1657.
[48] Antony Jameson. “Aerodynamic Shape Optimization Using the Adjoint Method”. In: *VKI Lecture Series on Aerodynamic Drag Prediction and Reduction, von Karman Institute of Fluid Dynamics, Rhode St Genese*. 2003, pp. 3–7.
[49] Nathaniel Kroeger. “ADMM Based Methods for Time-Domain Decomposition Formulations of Optimal Control Problems”. Master’s Thesis. Rice University, 2020.
[50] Yoshiki Kuramoto. “Diffusion-Induced Chaos in Reaction Systems”. In: *Progress of Theoretical Physics Supplement* 64 (1978), pp. 346–367.
[51] Yoshiki Kuramoto and Toshio Tsuzuki. “Persistent Propagation of Concentration Waves in Dissipative Media Far from Thermal Equilibrium”. In: *Progress of Theoretical Physics* 55.2 (Feb. 1976), pp. 356–369.
[52] John E. Lagnese and Guenter Leugering. “Time-domain decomposition of optimal control problems for the wave equation”. In: *Systems and Control Letters* 48.3-4 (2003), pp. 229–242.
[53] Davide Lasagna. “Sensitivity and stability of long periodic orbits of chaotic systems”. In: *Physical Review E* 102 (5 2020).

[54] Davide Lasagna, Ati Sharma, and Johan Meyers. "Periodic shadowing sensitivity analysis of chaotic systems". In: *Journal of Computational Physics* 391.September 2019 (2019), pp. 119–141.

[55] Daniel J. Lea, Myles R. Allen, and Thomas W. N. Haine. "Sensitivity analysis of the climate of a chaotic system". In: *Tellus, Series A: Dynamic Meteorology and Oceanography* 52 (5 2000), pp. 523–532.

[56] Edward N. Lorenz. "Deterministic Nonperiodic Flow". In: *Journal of the Atmospheric Sciences* 20.2 (Mar. 1963), pp. 130–141.

[57] Yvon Maday, Mohamed-Kamel Riahi, and Julien Salomon. "Parareal in Time Intermediate Targets Methods for Optimal Control Problems". In: *Control and Optimization with PDE Constraints* (2013), pp. 79–92.

[58] Yvon Maday, Julien Salomon, and Gabriel Turinici. "Monotonic Parareal Control for Quantum Systems". In: *SIAM Journal on Numerical Analysis* 45.6 (Jan. 2007), pp. 2468–2482.

[59] Bruce C. Moore. "Principal Component Analysis in Linear Systems: Controllability, Observability, and Model Reduction". In: *IEEE Transactions on Automatic Control* 26.1 (1981), pp. 17–32.

[60] Hiro Mukai. "Parallel Algorithms for Unconstrained Optimization." In: *Proceedings of the IEEE Conference on Decision and Control* 1 (1979), pp. 451–454.

[61] Angxiu Ni. *Adjoint shadowing directions in hyperbolic systems for sensitivity analysis*. 2018. arXiv: 1807.05568 [math.DS].

[62] Angxiu Ni. "Approximating Linear Response by Nonintrusive Shadowing Algorithms". In: *SIAM Journal on Numerical Analysis* 59.6 (2021), pp. 2843–2865. DOI: 10.1137/20M1388255. URL: https://doi.org/10.1137/20M1388255.

[63] Angxiu Ni. "Fast adjoint differentiation of chaos via computing unstable perturbations of transfer operators". In: *arXiv* (2022).

[64] Angxiu Ni and Chaitanya Talnikar. "Adjoint sensitivity analysis on chaotic dynamical systems by Non-Intrusive Least Squares Adjoint Shadowing (NILSAS)". In: *Journal of Computational Physics* 395 (2019), pp. 690–709.

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

[66] Angxiu Ni et al. "Sensitivity analysis on chaotic dynamical systems by Finite Difference Non-Intrusive Least Squares Shadowing (FD-NILSS)". In: *Journal of Computational Physics* 394 (2019), pp. 615–631.

[67] Roel Nottrot. *Optimal Processes on Manifolds*. Vol. 963. Lecture Notes in Mathematics. Berlin, Heidelberg: Springer Berlin Heidelberg, 1982.

[68] *Optimal Control: Calculus of Variations, Optimal Control Theory and Numerical Methods*. International Series of Numerical Mathematics. Birkhäuser Basel, 1993.

[69] Bernard Pagurek and C. Murray Woodside. "The conjugate gradient method for optimal control problems with bounded control variables". In: *Automatica* 4 (5-6 Nov. 1968), pp. 337–349.

[70] Lev S. Pontryagin. *Mathematical Theory of Optimal Processes*. Classics of Soviet Mathematics. Taylor & Francis, 1987.

[71] James B. Rawlings, David Q. Mayne, and Moritz M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2017. ISBN: 9780975937730.

[72] James Reuther et al. "Constrained multipoint aerodynamic shape optimization using an adjoint formulation and parallel computers". In: *35th Aerospace Sciences Meeting and Exhibit* (1997).

[73] Christian E. Schaerer, Tarek Mathew, and Marcus Sarkis. "Block iterative algorithms for the solution of parabolic optimal control problems". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 4395 LNCS (2007), pp. 452–465.

[74] Karim Shawki and George Papadakis. "A preconditioned Multiple Shooting Shadowing algorithm for the sensitivity analysis of chaotic systems". In: *Journal of Computational Physics* 398 (2019).

[75] Karim Shawki and George Papadakis. "Feedback control of chaotic systems using multiple shooting shadowing and application to Kuramoto Sivashinsky equation". In: *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 476.2240 (2020).

[76] Gregory I. Sivashinsky. "Nonlinear analysis of hydrodynamic instability in laminar flames-I. Derivation of basic equations". In: *Acta Astronautica* 4.11-12 (Nov. 1977), pp. 1177–1206.

[77] Gregory I. Sivashinsky and Daniel M. Michelson. “On Irregular Wavy Flow of a Liquid Film Down a Vertical Plane”. In: *Progress of Theoretical Physics* 63.6 (June 1980), pp. 2112–2114.

[78] Kazumasa A. Takeuchi et al. “Hyperbolic decoupling of tangent space and effective dimension of dissipative systems”. In: *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* 84.4 (2011), pp. 1–19.

[79] Chaitanya Talnikar and Qiqi Wang. “Adjoint-based trailing edge shape optimization of a transonic turbine vane using large eddy simulations”. In: *arXiv* (Nov. 2020), pp. 1–34. URL: http://arxiv.org/abs/2011.06744.

[80] Chaitanya Talnikar, Qiqi Wang, and Gregory M. Laskowski. “Unsteady adjoint of pressure loss for a fundamental transonic turbine vane”. In: *Journal of Turbomachinery* 139.3 (2017).

[81] Andreas Thune. “A Parallel in Time Method for Optimal Control Algorithm”. Master’s Thesis. University of Oslo, 2017.

[82] Qiqi Wang. “Convergence of the least squares shadowing method for computing derivative of ergodic averages”. In: *SIAM Journal on Numerical Analysis* 52.1 (2014), pp. 156–170.

[83] Qiqi Wang. “Forward and adjoint sensitivity computation of chaotic dynamical systems”. In: *Journal of Computational Physics* 235 (2013), pp. 1–13.

[84] Qiqi Wang, Rui Hu, and Patrick J. Blonigan. “Least Squares Shadowing sensitivity analysis of chaotic limit cycle oscillations”. In: *Journal of Computational Physics* 267 (2014), pp. 210–224.

[85] Qiqi Wang et al. “Towards scalable parallel-in-time turbulent flow simulations”. In: *Physics of Fluids* 25.11 (Nov. 2013), p. 110818.

[86] Markus Widhalm, Arno Ronzheimer, and Martin Hepperle. “Comparison between gradient-free and adjoint based aerodynamic optimization of a flying wing transport aircraft in the preliminary design”. In: *Collection of Technical Papers - AIAA Applied Aerodynamics Conference* 1.June 2014 (2007), pp. 727–747.

[87] Lilia Ziane Khodja et al. “Parallel sparse linear solver with GMRES method using minimization techniques of communications for GPU clusters”. In: *The Journal of Supercomputing* 69 (Mar. 2014).