Banner appropriate to article type will appear here in typeset article

Sparsifying the resolvent forcing mode via gradient-based optimisation

Calum S. Skene[1]†‡, Chi-An Yeh[1]¶, Peter J. Schmid[2], and Kunihiko Taira[1]

[1]Department of Mechanical and Aerospace Engineering, University of California, Los Angeles, CA 90095, USA

[2]Department of Mathematics, Imperial College London, London SW7 2AZ, United Kingdom

(Received xx; revised xx; accepted xx)

We consider the use of sparsity-promoting norms in obtaining localised forcing structures from resolvent analysis. By formulating the optimal forcing problem as a Riemannian optimisation, we are able to maximise cost functionals whilst maintaining a unit-energy forcing. Taking the cost functional to be the energy norm of the driven response results in a traditional resolvent analysis and is solvable by a singular value decomposition (SVD). By modifying this cost functional with the L_1-norm, we target spatially localised structures that provide an efficient amplification in the energy of the response. We showcase this optimisation procedure on two flows; plane Poiseuille flow at a Reynolds number of $Re = 4000$ and turbulent flow past an NACA0012 aerofoil at $Re = 23000$. In both cases, the optimisation yields sparse forcing modes that maintain important features of the structures arising from an SVD in order to provide a gain in energy. These results showcase the benefits of utilising a sparsity-promoting resolvent formulation to uncover sparse forcings, specifically with a view to using them as actuation locations for flow control.

1. Introduction

Resolvent analysis is a framework in which harmonic forcings that provide maximal amplification in their harmonic response can be determined on a frequency-by-frequency basis (Trefethen *et al.* 1993; Farrell & Ioannou 1993). By sweeping through frequencies, structural mechanisms that provide efficient means of flow amplification, as well as effective frequencies at which to provide such forcings, can be identified. While the original resolvent analysis focused on perturbation dynamics about steady states, recent studies have extended the analysis to system dynamics about the mean flow with emphasis on examining the self-sustaining fluctuations that are characteristic of turbulent flows (McKeon & Sharma 2010).

† Email address for correspondence: c.s.skene@leeds.ac.uk

‡ Present address: Department of Applied Mathematics, University of Leeds, Leeds LS2 9JT, UK

¶ Present address: Department of Mechanical and Aerospace Engineering, North Carolina State University, NC 27695, USA

Abstract must not spill onto p.2

With the resolvent analysis being able to reveal the input-output relationship with respect to the chosen base state (Jovanović & Bamieh 2005), it naturally serves as a valuable tool to design flow control techniques. Past studies including Luhar *et al.* (2014), Yeh & Taira (2019), Toedtli *et al.* (2019), and Liu *et al.* (2021) have demonstrated that physical insights revealed from resolvent analysis provide valuable guidance of developing effective and efficient actuation strategies.

Traditionally, modal analysis techniques for fluid flows (Taira *et al.* 2017, 2020) have been founded on L_2-based norms, which can lead to global spatial structures. For the resolvent analysis, this translates to having forcing modes that are spatially supported over a large region. It should however be realised that actuation inputs cannot be applied over a large spatial region in practical engineering flow control settings. In general, flow control inputs can only be introduced as localised actuation inputs. To address this point, we consider sparsity-promoting approaches to specifically target resolvent forcing modes that have spatially compact support, i.e., are spatially sparse. We also note that sparsity-promoting techniques may also help identify appropriate variables through which control inputs can be added to the flow for enhanced amplification. This piece of information is important in selecting the appropriate type of actuators to introduce control input to the flow field (Cattafesta & Sheplak 2011).

To sparsify the resolvent forcing mode, we adopt a similar approach to Foures *et al.* (2013), who used alternative norms for studying transient growth in plane Poiseuille flow. In their work, transient growth analysis has been treated as a gradient-based optimisation problem, where the goal is to find the initial condition that has the most growth as measured by an appropriate norm. Choosing the L_2-norm leads to the usual transient growth analysis (Trefethen *et al.* 1993) that can be solved using a singular value decomposition. However, choosing an alternative norm can tune the analysis to reveal different mechanisms which would be sub-optimal in terms of the L_2-norm.

Foures *et al.* (2013) found more localised transient growth mechanisms using the infinity-norm, i.e., by measuring the norm of the response by its maximum value rather than energy. The result of this is that the identified initial conditions are spatially localised in order to achieve responses that are focused around local 'hot spots.' Further to this, the non-convex nature of this optimisation problem means that there exist different branches of optimal initial conditions, with some representing local maximums of the cost functional. Physically, these localisations manifested themselves in the form of initial conditions that focused either in the middle of the channel or towards the walls.

Following this approach, our study considers resolvent analysis as an optimisation problem where forcing modes are sought that maximise a prescribed cost functional. In order to obtain spatially sparse forcing modes, we propose a gradient-based algorithm that maximises the energy of the output whilst minimising the L_1-norm of the forcing, which is also constrained to have unit energy. To provide an initial assessment of our proposed method we consider two examples. Firstly, we consider the same plane Poiseuille setup as in Foures *et al.* (2013), allowing us to qualitatively assess the differences between localisation strategies for initial conditions and for forced problems. Secondly, we consider turbulent flow past an aerofoil using the same mean-flow as Yeh & Taira (2019), providing an assessment of the method in a higher Reynolds number, turbulent flow.

The structure of the paper is as follows. Section 2 outlines the mathematical formulation of the paper and contains an introduction to the resolvent operator, a background on Riemannian optimisation and how we utilise it to find optimal, sparse resolvent modes, and a discussion of wavemakers in the context of a resolvent analysis. In section 3 we discuss the numerical setup, with the results subsequently being presented in section 4. Conclusions are offered in section 5.

2. Formulation

2.1. *The resolvent operator*

Let us consider the Navier–Stokes equations in the general, spatially discretised form

$$\boldsymbol{G}\frac{d\boldsymbol{q}}{dt} = \boldsymbol{\mathcal{N}}(\boldsymbol{q}), \tag{2.1}$$

where $\boldsymbol{q}$ is the spatially discretized state vector, and $\boldsymbol{\mathcal{N}}$ represents the right-hand side of the unforced Navier–Stokes equations. Including the mass-matrix $\boldsymbol{G}$ in equation (2.1) means that this form could represent either the compressible Navier–Stokes equations or the incompressible equations where there is no time-derivative in the continuity equation. By linearising this equation about a base flow $\mathbf{q}_0$, we can write the system in input-output form as

$$\boldsymbol{G}\frac{d\boldsymbol{q}}{dt} = \boldsymbol{L}_{\boldsymbol{q}_0}\boldsymbol{q} + \boldsymbol{B}\boldsymbol{f}, \tag{2.2}$$

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{q}, \tag{2.3}$$

where $\boldsymbol{L}_{\boldsymbol{q}_0}$ is the linearised Navier–Stokes operator (Jeun *et al.* 2016). The matrix $\boldsymbol{B}$ allows for the introduced forcing $\boldsymbol{f}$ (input) to be windowed in space or restricted to specific equations or state variables. In an analogous manner, the matrix $\boldsymbol{C}$ allows for a similar windowing to be applied to the output $\boldsymbol{y}$.

The relationship between harmonic inputs and outputs with frequency ω can be obtained by Laplace transforming the input-output system in time, giving the relation

$$\hat{\boldsymbol{y}} = \boldsymbol{C}(-i\omega\boldsymbol{G} - \boldsymbol{L}_{\boldsymbol{q}_0})^{-1}\boldsymbol{B}\hat{\boldsymbol{f}}. \tag{2.4}$$

Through this equation, the resolvent operator is defined via $\boldsymbol{\mathcal{H}}_{\boldsymbol{q}_0} \equiv \boldsymbol{C}(-i\omega\boldsymbol{G} - \boldsymbol{L}_{\boldsymbol{q}_0})^{-1}\boldsymbol{B}$. The form of equation (2.4) shows that the resolvent operator is equivalent to a transfer function which maps the forcing to its time-asymptotic response. Before we discuss the meaning of the resolvent in fluid dynamics it is worth considering the Laplace variable ω. If the operator $\boldsymbol{L}_{\boldsymbol{q}_0}$ is stable then ω is real and (2.4) is obtainable via the Fourier transform. However, if $\boldsymbol{L}_{\boldsymbol{q}_0}$ is unstable then more care is needed. Indeed, for unstable $\boldsymbol{L}_{\boldsymbol{q}_0}$ the time-asymptotic response is not given via (2.4) and is instead a combination of the exponentially growing disturbance given by the most unstable eigenvector and the forced response given by the resolvent. In order to separate these two mechanisms a complex value for ω can be used leading to the concept of a time-discounted resolvent analysis (Jovanović 2004). Choosing complex values for ω means that the imaginary part can be chosen such that the forced response 'rises above' the exponentially growing disturbance due to the unstable nature of $\boldsymbol{L}_{\boldsymbol{q}_0}$, allowing for the forced dynamics to be probed (see Yeh *et al.* (2020) for more details).

In the context of fluid dynamics, the resolvent can be interpreted in two main ways. Firstly, choosing $\boldsymbol{q}_0$ to be a steady solution to the unforced Navier–Stokes equations leads to a non-normal stability study of the flow. In this manner, the resolvent identifies forcing structures that lead to particularly efficient amplification in the dynamics despite the stable nature of the flow (Trefethen *et al.* 1993). Secondly, using a time-averaged mean-flow for $\boldsymbol{q}_0$ leads to the resolvent formulation of turbulence (McKeon & Sharma 2010). The resolvent in this instance can be used to identify the coherent structures that arise via disturbances caused by the non-linear terms.

For both steady base-flows and time-averaged mean-flows, the resolvent provides critical insights into how forcings can cause an amplification in the dynamics. This amplification can occur both from resonant frequencies, and also from particularly effective structural mechanisms. Whilst one could choose a variety of forcings $\hat{\boldsymbol{f}}$ at each frequency to determine the most effective structures, it is more efficient to directly solve for the optimal forcing. This can be mathematically formulated as

$$\boldsymbol{f}_{\text{opt}} = \arg\max_{\boldsymbol{f}} \frac{\|\boldsymbol{y}\|_{\boldsymbol{W_q}}}{\|\boldsymbol{f}\|_{\boldsymbol{W_f}}}, \tag{2.5}$$

where the norms are defined as $\|\boldsymbol{a}\|^2_{\boldsymbol{W}} = \boldsymbol{a}^H \boldsymbol{W} \boldsymbol{a}$ with $\boldsymbol{W}$ being a positive definite weight matrix. We allow for the weight matrix for the forcing ($\boldsymbol{W_f}$) and response ($\boldsymbol{W_q}$) to be different. These matrices are problem dependent, and are chosen so that the norms represent appropriate measures of the energy (see sections 3.1 and 3.2 for examples). The cost functional in this case is the gain. To link the weighted norms to the two norm, it is useful to also consider the Cholesky decomposition $\boldsymbol{W} = \boldsymbol{M}^H \boldsymbol{M}$. The optimal forcing has the corresponding output $\tilde{\boldsymbol{y}}_{\text{opt}} = \boldsymbol{\mathcal{H}} \boldsymbol{f}_{\text{opt}}$ with the amount of amplification being measured by the gain $\sigma = \|\tilde{\boldsymbol{y}}_{\text{opt}}\|_{\boldsymbol{W_q}} / \|\boldsymbol{f}_{\text{opt}}\|_{\boldsymbol{W_f}}$. This problem can be solved by taking the SVD of $\boldsymbol{M_q} \boldsymbol{\mathcal{H}} \boldsymbol{M_f}^{-1}$, whose maximum singular triplet is $(\sigma, \boldsymbol{M_q} \boldsymbol{y}_{\text{opt}}, \boldsymbol{M_f} \boldsymbol{f}_{\text{opt}})$.

While a resolvent analysis in this manner can provide useful information about frequencies and forcing structures that can provide a large amplification, and therefore identify good candidates for flow control, the forcing structures are often global. This means that implementing them in a practical situation is infeasible. In the present study, we present an approach to finding sparse (spatially compact) resolvent forcings that induce large amplifications in the underlying dynamics. In this manner, particularly sensitive spatial locations in the flow field are identified, providing a guide for effective and efficient actuation.

2.2. *Sparsification via Riemannian optimisation*

To seek a spatially sparse resolvent forcing mode, we first generalise the optimal forcing problem. We start by realising that finding the greatest singular value of the resolvent matrix is equivalent to maximising the gain

$$\sigma = \frac{\|\boldsymbol{\mathcal{H}} \boldsymbol{f}\|^2_{\boldsymbol{W_q}}}{\|\boldsymbol{f}\|^2_{\boldsymbol{W_f}}}. \tag{2.6}$$

Therefore, instead of carrying out an SVD, we could instead maximise the gain via a gradient ascent algorithm. It is useful to phrase this optimisation as follows; maximise the gain

$$\sigma = \|\boldsymbol{\mathcal{H}} \boldsymbol{f}\|^2_{\boldsymbol{W_q}}, \tag{2.7}$$

where the forcing is confined to the manifold given by the constraint $\|\boldsymbol{f}\|^2_{\boldsymbol{W_f}} = 1$. This is an equivalent problem to (2.6) because the resolvent is linear and hence

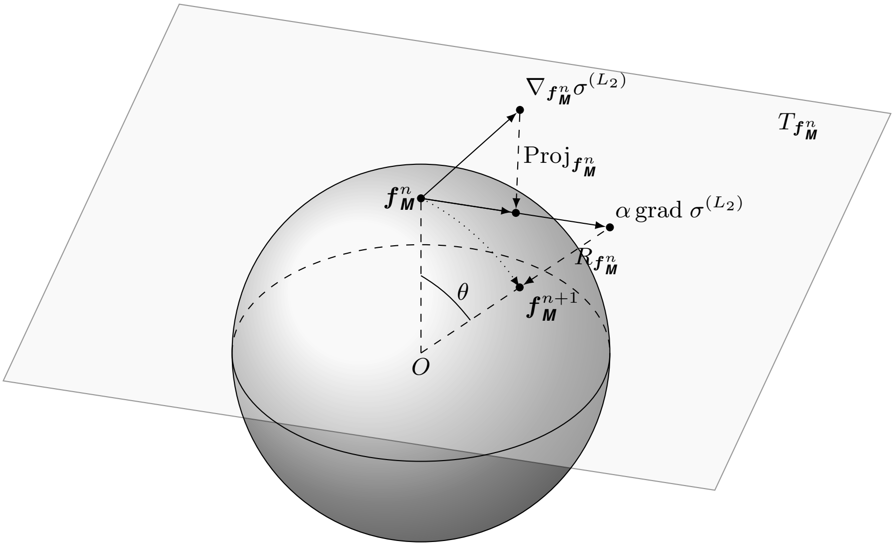


Figure 1: A sketch illustrating the concept of Riemannian optimisation. First, the Euclidean gradient $\nabla_{\boldsymbol{f}_{\boldsymbol{M}}^n}\sigma^{(L_2)}$ is found from the vector $\boldsymbol{f}_M^n$ that is situated on the hypersphere $\mathcal{S}$. This vector is then mapped to the tangent space $T_{\boldsymbol{f}_{\boldsymbol{M}}^n}\mathcal{S}$ via the projection $\mathrm{Proj}_{\boldsymbol{f}_{\boldsymbol{M}}^n}$. Next, the Riemannian gradient is extended along the tangent space by the step-size α. Finally, we map this gradient back to the manifold via the retraction $R_{\boldsymbol{f}_{\boldsymbol{M}}^n}$, yielding the updated forcing vector $\boldsymbol{f}_{\boldsymbol{M}}^{n+1}$. For varying values of α the retraction traces out a smooth curve over the manifold, displayed as a dotted line. The link between α and θ is also shown.

will produce the same gain defined by (2.6), if we choose the forcing to have a unit-energy norm. In effect, by constraining our forcing to this manifold, we are ensuring that we search for the maximum amplification in dynamics with the forcing having the same energy budget.

Whilst we could conduct an unconstrained optimisation by enforcing $\|\boldsymbol{f}\|^2_{\boldsymbol{W_f}} = 1$ with a Lagrange multiplier (Pringle *et al.* 2012), we instead take account of this constraint directly in the update step. This results in a similar approach to that of Foures *et al.* (2013), where a geometric approach was used to ensure that the unit-norm condition is satisfied when stepping in the search direction. In general, carrying out an optimisation where the input is constrained to a manifold is known as Riemannian optimisation (Absil *et al.* 2007).

Let us first discuss the optimisation problem considered thus far. We seek to maximise the gain

$$\sigma^{(L_2)} = \|\boldsymbol{M_q}\boldsymbol{\mathcal{H}}\boldsymbol{M_f}^{-1}\boldsymbol{M_f}\boldsymbol{f}\|_2, \tag{2.8}$$

subject to $\|\boldsymbol{M_f}\boldsymbol{f}\|_2 = 1$. By expressing the problem in this form, we have reformulated the problem in terms of the L_2-norm, and hence we are optimising with respect to the vector $\boldsymbol{f_M} = \boldsymbol{M_f}\boldsymbol{f}$, which we constrain to have unit L_2-norm. The manifold for this problem then becomes the complex-hypersphere $\mathcal{S} = \{\boldsymbol{y} \mid \boldsymbol{y}^H\boldsymbol{y} = 1\}$.

For an unconstrained optimisation, we generally work with the Euclidean gradient

$$\nabla_{\boldsymbol{f}_M}\sigma^{(L_2)} = \frac{\partial \sigma^{(L_2)}}{\partial \boldsymbol{f}_M}. \tag{2.9}$$

By stepping in the direction of the conjugate of this gradient, we would be increasing the value of $\sigma^{(L_2)}$, assuming that we use a sufficiently small step size for which a linear approximation is appropriate. The problem with this approach is that stepping in such a direction would most likely result in a vector that is no longer on the manifold.

To carry out a gradient descent on the hypersphere, we must therefore define the gradients appropriately. Riemannian optimisation will not work directly with the Euclidean gradient, but instead all gradients must be tangent to the hypersphere at the evaluation points. The set of all vectors tangent to the manifold at a point $\boldsymbol{x}$ is known as the tangent space $\mathrm{T}_{\boldsymbol{x}}\mathcal{S}$, with the set of all tangent spaces being referred to as the tangent bundle $\mathrm{T}\boldsymbol{x} = \sum_{\boldsymbol{x}\in\mathcal{S}} \mathrm{T}_{\boldsymbol{x}}\mathcal{S}$ (see figure 1 which schematically shows the Riemannian optimisation procedure). For the hypersphere, the Riemannian gradient can be written as

$$\operatorname{grad}\sigma^{(L_2)}(\boldsymbol{f_M}) = \left(1 - \boldsymbol{f}_{\boldsymbol{M}}^H\boldsymbol{f_M}\right)\nabla_{\boldsymbol{f}_M}\sigma^{(L_2)} = \operatorname{Proj}_{\boldsymbol{f_M}}\left(\nabla_{\boldsymbol{f_M}}\sigma^{(L_2)}\right), \tag{2.10}$$

where the function Proj is the projection that links the Riemannian gradient to the Euclidean one.

Now that we have defined appropriate gradients, we must also define how to step in the direction of steepest ascent. For the unconstrained optimisation, we may simply add a scalar multiple (the step size) of this gradient onto our current value of the forcing. However, for the Riemannian optimisation, this will result in a vector that no longer falls on the manifold, as noted above. The equivalent procedure in this case is the notion of a retraction. A retraction is a mapping $R_{\boldsymbol{x}}(\boldsymbol{\xi}) : \mathrm{T}_{\boldsymbol{x}}\mathcal{S} \to \mathcal{S}$ such that $R_{\boldsymbol{x}}(\mathbf{0}) = \boldsymbol{x}$ and $DR_{\boldsymbol{x}}(\mathbf{0}) = \mathrm{id}_{\mathrm{T}_{\boldsymbol{x}}\mathcal{S}}$. In other words, a retraction maps vectors tangent to the manifold at $\boldsymbol{x}$ to other vectors on the manifold such that for $\boldsymbol{\xi} = \mathbf{0}$ it maps $\boldsymbol{x}$ to itself, and that the derivative of the mapping at $\boldsymbol{\xi} = \mathbf{0}$ is the identity. For the hypersphere, we have the retraction

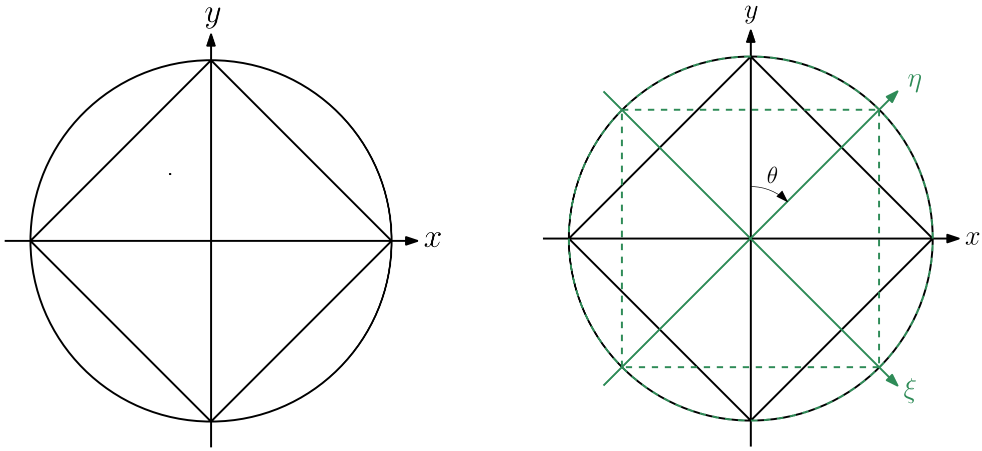
$$R_{\boldsymbol{x}}(\boldsymbol{\xi}) = \frac{\boldsymbol{x}+\boldsymbol{\xi}}{\|\boldsymbol{x}+\boldsymbol{\xi}\|}. \tag{2.11}$$

Once the gradient is found, we can then update the forcing using the map $R_{\boldsymbol{f_M}}(\alpha\operatorname{grad}\ \sigma^{(L_2)}(\boldsymbol{f_M}))$, where α denotes the step size. By writing $\cos(\theta) = 1/\sqrt{1+\alpha^2}$, we can also express the update step as

$$\boldsymbol{f}_{\boldsymbol{M}}^{\mathrm{n}+1} = R_{\boldsymbol{f}_{\boldsymbol{M}}^n}(\alpha\operatorname{grad}\sigma^{(L_2)}(\boldsymbol{f}_{\boldsymbol{M}}^n)) = \cos(\theta)\boldsymbol{f}_{\boldsymbol{M}}^{\mathrm{n}} + \sin(\theta)\operatorname{grad}\sigma^{(L_2)}(\boldsymbol{f}_{\boldsymbol{M}}^n), \tag{2.12}$$

which is exactly the geometric form used by Foures *et al.* (2013). Note that we have described a steepest ascent approach here. However many alternative gradient-based optimisation algorithms, such as the conjugate gradient and Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithms, are applicable to Riemannian optimisation with faster convergence (Boumal & Absil 2015; Huang *et al.* 2016).

The main advantage of phrasing the optimal forcing-output problem in this way is its generality. Whilst we have shown how we can obtain the same result as the SVD (and it is actually possible to get the higher-order singular values in this manner by considering a different manifold), we are free to change how we define



(a) The unit L_2 norm (circle) and L_1 norm (diamond) with respect to the coordinate system (x, y).

(b) The L_2 norm (green-dashed circle) and L_1 norm (green-dashed diamond) with respect to the coordinate system (ξ, η).

Figure 2: Sketches of the L_2- and L_1-norms. The effect of a coordinate rotation on the norms is also demonstrated.

the gain. The SVD can only find the gain in the L_2-norm sense. This means that the input is measured by an energy norm, leading to the global structures seen in many studies. In order to introduce sparsification, we consider the use of the 1-norm.

A sketch of the unit L_2- and L_1-norms for a vector (x, y) in $\mathbb{R}^2$ is presented in figure 2a. The L_2-norm takes the form of a circle, whereas the L_1-norm yields a regular diamond inscribed within this circle. Note that the unit L_1-norm touches the unit L_2-norm at the coordinate axes. This indicates that the L_1-norm for all vectors with unit L_2-norm yields its smallest value for sparse vectors, i.e., vectors (x, y) with either x or y equal to zero. Indeed, if the square touched the circle at another location (x_0, y_0) with $x_0 \neq 0$ and $y_0 \neq 0$ then the L_2-norm would be unity whereas the L_1-norm would have a value of $|x_0|+|y_0| > 1$. Hence, optimising over the space of unit-norm forcings, whilst penalising the L_1 norm, will push the forcing vector, and hence its structure, towards more locally supported structures.

One important consideration when using an alternative norm, such as the L_1-norm, is illustrated by figure 2b. Here we have again shown the unit L_2- and L_1-norms, but this time for the coordinate system (η, ξ) which is obtained via a rotation of the coordinate system (x, y) by an angle θ. In this new coordinate system, the L_2-norm still represents a circle, which is invariant under this transformation. However, the unit L_1-norm is affected, and its square shape is rotated by the angle θ. This means that in this new coordinate system the sparse vectors, where the square touches the circle, are different to those of the original coordinate system (x, y). In other words, what is considered sparse is completely defined by how we choose to represent our vectors. In practise, we must be careful when choosing the vector of which we take the L_1-norm. We therefore choose to take the L_1-norm of a vector that leaves the L_2-norm unchanged, yet has appropriate axes for best defining the sparsity we aim to achieve. In terms of resolvent analysis, this transformation is used to maintain the physical relevance of the sparsification. Specific examples are described in sections 3.1 and 3.2.

Based on the discussion of the previous two paragraphs, we seek to maximise the new gain $\sigma^{(L_1)}$ defined by

$$\sigma^{(L_1)} = \frac{\sigma^{(L_2)}}{\|\boldsymbol{T}(\boldsymbol{f_M})\|_1} = \frac{\|\boldsymbol{M_q}\boldsymbol{\mathcal{H}}\boldsymbol{M_f}^{-1}\boldsymbol{f_M}\|_2}{\|\boldsymbol{T}(\boldsymbol{f_M})\|_1}, \tag{2.13}$$

still subject to the forcing $\boldsymbol{f_M}$ having a unit-energy norm. The transformation $\boldsymbol{T}$ in the denominator is a transformation of the vector $\boldsymbol{f_M}$ to another vector. Hence, the vector in the denominator need not be equal to the forcing vector $\boldsymbol{f_M}$ as, based on the discussion of the previous paragraph, this may not be physically relevant. However, by ensuring $\|\boldsymbol{T}(\boldsymbol{f_M})\|_2 = \|\boldsymbol{f_M}\|_2$, we maintain the geometric interpretation of sparsity illustrated by figure 2a, albeit in a much higher dimensional space. By dividing the usual gain by the 1-norm of the vector $\boldsymbol{T}(\boldsymbol{f_M})$, we are in effect promoting sparsity, with sparsity defined as a vector $\boldsymbol{T}(\boldsymbol{f_M})$ with a minimal number of non-zero entries. Optimising the gain in this form will seek a compromise between providing a large gain in energy whilst ensuring the spatial sparsity of the forcing. Indeed, the maximal nature of $\sigma^{(L_1)}$ means that obtaining a response with larger energy requires a forcing structure that is less sparse. Likewise, making the forcing more sparse leads to a less energetic response.

As in the study of Foures *et al.* (2013), who considered a similar optimisation problem for localising flow structures obtained in transient growth studies, our cost functional is non-convex. This means that any solution to the optimisation problem is only guaranteed to be a local, rather than global, maximum of the cost functional. In the case of transient growth this led to multiple branches of solutions being found during the optimisation, which could be discovered by running the problem with multiple starting guesses for the gradient-based optimisation. However, despite running multiple instances of each optimisation with different initial guesses in our following examples, no differences in the solution could be found apart from symmetries of the flow which are to be expected. Whist this does not confirm that our results are truly the global optimum, it does highlight a difference between localising forcings for driven versus initial condition based studies.

Another important factor is the realisation that the L_1-norm is notoriously hard to optimise due to its non-smoothness near the origin. Intuitively, we can visualise the problem by considering the unconstrained optimisation problem of minimising the L_1-norm of a scalar a. Using our gradient based approach, this amounts to stepping in the direction of steepest descent, which for our simple example is the sign of a. No matter how near or far we are to the optimal value of $a = 0$, this gradient will have the same magnitude. This means that we will continuously step over the optimal value, unless the step-size is perfect, leading to zig-zagging and ultimately causing the algorithm to converge rather slowly. To alleviate this behavior we replace the L_1-norm with a smooth approximation, namely $l_{1,\delta}(\boldsymbol{q}) = h_\delta(\boldsymbol{q})/\delta$ where $h_\delta(\boldsymbol{q})$ is the pseudo-Huber norm (Bube & Langan 1997; Bube & Nemeth 2007)

$$h_\delta(\boldsymbol{q}) = \sum_j \delta^2 \left(\sqrt{1 + \frac{|q_j|^2}{\delta^2}} - 1\right). \tag{2.14}$$

This pseudo-Huber norm has the property that it approximates the L_1-norm for small δ and is completely smooth. Therefore, in order to achieve convergence, we will not optimise $\sigma^{(L_1)}$ directly but perform a series of optimisations for the quantity

$$\sigma_\delta^{(L_1)} = \frac{\|\boldsymbol{M_q}\boldsymbol{\mathcal{H}}\boldsymbol{M_f}^{-1}\boldsymbol{f_M}\|_2}{l_{1,\delta}(\boldsymbol{T}(\boldsymbol{f_M}))}, \tag{2.15}$$

for decreasing values of δ. By using the optimal forcing obtained from an optimisation for the preceding one with a lower value of δ, we are able to more robustly achieve a converged optimisation for a sufficiently small δ such that our norm (2.14) is an appropriate approximation for the true L_1-norm.

Before concluding this section, it is important to note that our choice of cost functional is not unique. Indeed, other cost functionals such as $\sigma^{(L_1)} = \sigma^{(L_2)} - \mu\|\boldsymbol{T}(\boldsymbol{f_M})\|_1$ can also lead to sparse forcing modes for appropriate choices of μ. However, the fact that unit-norm forcings can lead to gains in energy many orders of magnitude larger than that of the forcing makes the choice of μ, which must balance the L_2-based gain against the L_1-based forcing, a difficult challenge. This is further complicated by the strong dependence of the gain on the forcing frequency, making a universally good way of choosing μ hard to determine. In our proposed cost functional there is no such parameter to choose, meaning that it can easily be applied to different frequencies and base-flows without change. Hence, we continue with it for the rest of the study.

2.3. *Resolvent wavemaker*

One concept that we use in our subsequent analysis is that of structural sensitivity and the wavemaker (Giannetti & Luchini 2007). The wavemaker has its origin in global stability analysis and provides a way to highlight regions in which flow field changes result in changes to global modes. Specifically, the wavemaker is the structural sensitivity to a localised flow feedback. To obtain the wavemaker, we consider the eigenvalue problem $\boldsymbol{L}\boldsymbol{x} = \lambda\boldsymbol{G}\boldsymbol{x}$. This eigenvalue problem could arise, for instance, as a global stability problem, in which case $\boldsymbol{x}$ would be the global mode, with the corresponding eigenvalue λ giving its frequency and growth rate. It can be shown (see the review of Schmid & Brandt (2014) for example) that to first order, a perturbation to the eigenvalue $\delta\lambda$ for a perturbation in the matrix $\boldsymbol{\delta L}$ is given via

$$\delta\lambda = \frac{\langle \boldsymbol{x}^\dagger, \boldsymbol{\delta L}\boldsymbol{x}\rangle}{\langle \boldsymbol{x}^\dagger, \boldsymbol{G}\boldsymbol{x}\rangle}, \tag{2.16}$$

where $\boldsymbol{x}^\dagger$ is the solution to the adjoint eigenvalue problem $\boldsymbol{L}^H\boldsymbol{x}^\dagger = \bar{\lambda}\boldsymbol{G}^H\boldsymbol{x}^\dagger$. The wavemaker is then obtained by specifying $\boldsymbol{\delta L} = \boldsymbol{I}$ and instead taking the element-wise, or Hadamard ($\odot$), product.

$$\boldsymbol{\lambda} = \frac{\bar{\boldsymbol{x}}^\dagger \odot \boldsymbol{x}}{\langle \boldsymbol{x}^\dagger, \boldsymbol{G}\boldsymbol{x}\rangle}. \tag{2.17}$$

In this way, the wavemaker $\boldsymbol{\lambda}$ can then be thought of as a vector-field $\boldsymbol{\lambda}(x, y) = (\lambda_u, \lambda_v, \lambda_w)$ whose components represent what changes to the eigenvalue occur from localised feedback at each location and state-component in the flow field.

We quickly note that there are a few ways in which the wavemaker could be perceived. Whilst we have stayed within a discrete setting Giannetti & Luchini (2007) present the wavemaker in a continuous formulation. This gives the main difference that their wavemaker is a scalar field which is defined pointwise via $\lambda(x,y) = \|\boldsymbol{x}^\dagger(x,y)\|\|\boldsymbol{x}(x,y)\|$. Hence, their wavemaker, by the Cauchy-Schwarz inequality, shows the maximum change to the eigenvalue that can be achieved via localised feedback at each spatial location. Conversely, the wavemaker presented by Schmid & Brandt (2014) is more easily related to ours via $\lambda(x,y) = \lambda_u + \lambda_v + \lambda_w$. In this manner, they obtain a complex-valued wavemaker whose real and imaginary parts show the individual changes to the real and imaginary parts of the eigenvalue. Additionally, the sign of these changes is retained, allowing for the direction of the eigenvalue perturbation to be determined. However, by keeping the values of the flow-field separate, our wavemaker definition is strongly related to that of Paladini *et al.* (2019), who introduce windowing matrices to allow for the selection of specific physical components in the resulting wavemaker. Whilst they use these matrices to isolate the contribution of the momentum to the wavemaker, we instead do this procedure for each separate velocity component. This means that for each spatial location, our wavemaker tells us how a specific eigenvalue will move for localised feedback restricted to each component of the state-vector.

Whilst the previous paragraph talked about wavemakers in terms of an eigenvalue problem, it can also be directly formulated for an SVD-based resolvent analysis (Qadri & Schmid 2017). Indeed, by realising that taking the SVD of the matrix $\boldsymbol{K} = \boldsymbol{M_q}\mathcal{H}\boldsymbol{M_f}^{-1}$ is equivalent to taking the eigenvalues of the matrix $\boldsymbol{K}^H\boldsymbol{K}$, the same procedure that yields (2.16) can be applied, resulting in

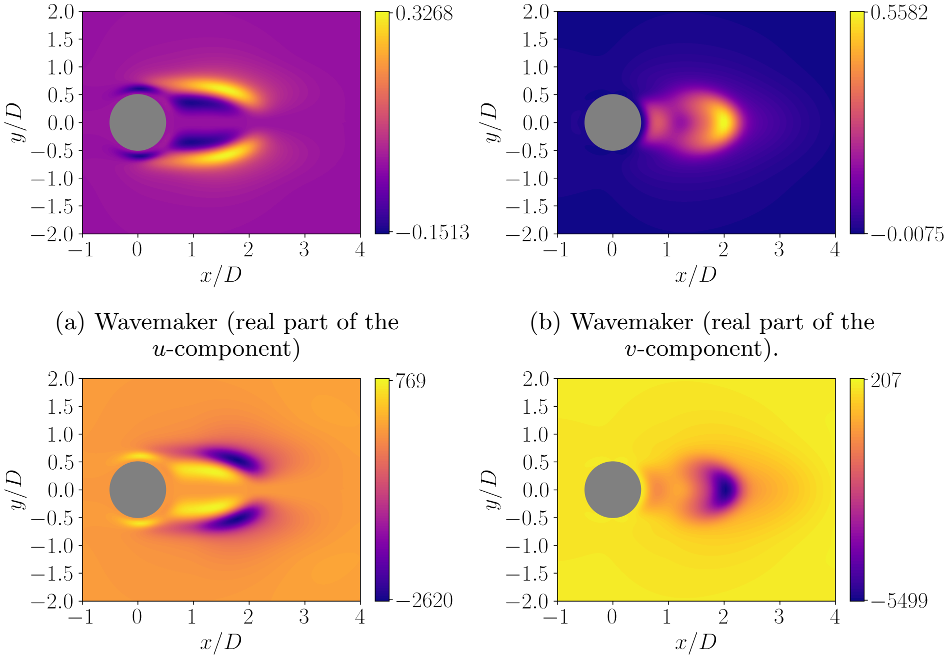
$$\delta\sigma = \sigma^2 \mathrm{Real}\left(\langle \boldsymbol{f}, \boldsymbol{\delta L q}\rangle_{\boldsymbol{W_f}}\right), \tag{2.18}$$

where $\boldsymbol{L}$ stands for the linearised Navier–Stokes operator (Fosas de Pando *et al.* 2014; Fosas de Pando & Schmid 2017; Qadri & Schmid 2017). Again, taking $\boldsymbol{\delta L} = \boldsymbol{I}$ and using the Hadamard product yields

$$\boldsymbol{\sigma} = \sigma^2 \mathrm{Real}\left(\bar{\boldsymbol{f}} \odot \boldsymbol{W_f q}\right). \tag{2.19}$$

The resolvent wavemaker $\boldsymbol{\sigma}$ is then analogous to the eigenvalue-based wavemaker, i.e., for localised feedback at each spatial location and component of the state-vector, the resolvent wavemaker will indicate how the singular value will be perturbed.

An example of the wavemaker and resolvent wavemaker is shown for cylinder flow in figure 3. It is important to note that for the eigenvalue-based wavemaker the frequency is set by the eigenvalues. However, our definition of the resolvent wavemaker allows any frequency to be specified. Therefore, we concentrate on $St = 0.162$, which is the frequency at which the most unstable eigenvalue is found. We observe that these wavemakers have similar structures but with different gains. The fact that they have similar structures is not surprising, since the resolvent forcing and response modes are qualitatively similar to the direct and adjoint eigenvectors, respectively. However, the signs of the structures are often different. This indicates that a localised feedback affects the eigenvalue perturbation differently from the singular value perturbation, highlighting the importance of formulating a resolvent wavemaker in order to quantify the effect of localised feedback for resolvent analyses.



(a) Wavemaker (real part of the u-component)

(b) Wavemaker (real part of the v-component).

(c) Resolvent wavemaker (u-component) at $St = 0.162$.

(d) Resolvent wavemaker (v-component) at $St = 0.162$.

Figure 3: The eigenvalue-based wavemakers and resolvent wavemakers for cylinder flow at $Re = 100$ shown for illustration. Computations performed with the immersed boundary projection method *ibmos* (Fosas de Pando 2020).

3. Numerical setup

This section describes the numerical setup for our flow examples. In addition to the details given in this section, all Riemannian optimisations are managed using the python package *pyManopt* (Townsend *et al.* 2016), the python extension of the MATLAB package *Manopt* (Boumal *et al.* 2014). The optimisations are all conducted using the conjugate gradient algorithm.

3.1. *Plane Poiseuille flow*

Firstly, we consider plane Poiseuille flow to compare the differences in localisation strategies between initial-condition-based (transient growth) and driven (resolvent) studies. The present setup follows that of Foures *et al.* (2013) in which a domain covers $(x, y) \in [0, 2\pi] \times [0, 2]$ at a Reynolds number of $Re = 4000$. No-slip boundary conditions are applied at $y = 0$ and 2 and periodic boundary conditions are applied at $x = 0$ and 2π. The base flow is analytically provided as $u = y(2 - y)$. We conduct the numerical simulations using the python package *ibmos* developed by Fosas de Pando (2020). This is an immersed boundary projection code based on the formulation of Taira & Colonius (2007) with specific formulation for optimisation and stability analyses. The package solves the nonlinear incompressible Navier–Stokes equations and directly provides the linearised and adjoint codes necessary for conducting a resolvent analysis. For the plane Poiseuille examples, the matrix $\boldsymbol{B}$ is chosen so that the forcing is only added to the momentum equations. Similarly, the matrix $\boldsymbol{C}$ is chosen so that only velocity components constitute the output.

As detailed in section 2.2, there is some consideration in choosing the vector for our L_1-norm, $\boldsymbol{T}(\boldsymbol{f_M})$. The obvious choice would be to use the same vector used for the unit energy norm, $\mathbf{f_M}$, for the L_1-norm. However, as the x- and y-components of the velocity occur in different locations of $\mathbf{f_M}$ this would result in a sparsification that not only sparsifies the forcing mode in space, but also sparsifies between the x- and y-components of velocity. In other words, if the sparse procedure were to locate a single spatial point for the forcing mode, it would also be advantageous to completely align the velocity vector with the coordinate axes at this point in order to achieve a further reduction in the L_1-norm. As we are primarily interested in localisation in space, as opposed to sparsifying the velocity vector itself, we therefore design a vector for the L_1-norm optimisation that does not result in this unwanted sparsification. This is particularly pertinent to applications of this method to flow-actuation, where the directional information obtained by keeping the x-, y- and possibly z-components of velocity independent of the sparsification procedure will provide additional insight into actuator design.

To this end, we consider a vector of the following form: $\boldsymbol{T}(\boldsymbol{f_M}) = \boldsymbol{M}(\boldsymbol{u} \odot \boldsymbol{u} + \boldsymbol{v} \odot \boldsymbol{v})^{1/2}$, where $\odot$ is the Hadamard product and the square root is taken componentwise. This vector has the same 2-norm as $\mathbf{f_M}$, but groups local contributions of the forcing mode to the total energy together. Hence, the L_1-norm of this vector is small when the forcing is localised in space, but without penalising among individual components of the velocity vector. It should be noted that some additional care may be needed when designing this vector depending on the specific numerical implementation. For example, as our immersed boundary implementation uses a staggered mesh with the x-components of velocity lying on the east and west faces of the cell whilst the y-components lie on the north and south faces, we form the vector $\boldsymbol{T}(\boldsymbol{f_M})$ on the cell centres by averaging the kinetic energy contributions from the cell-edges. The weight matrices are chosen to incorporate the grid spacing (see Taira & Colonius (2007) for more information), so that the forcing and response are measured in terms of the kinetic energy.

3.2. *Flow past an aerofoil*

Secondly, we also consider a spanwise-periodic turbulent flow over a canonical aerofoil obtained from a large-eddy simulation (LES) with a Vremen sub-grid scale model (Vreman 2004). The LES is conducted using the finite-volume solver *CharLES* that solves the compressible Navier–Stokes equations with second-order spatial and third-order temporal accuracies (Khalighi *et al.* 2011; Brès *et al.* 2017). The linearisation is performed within the same solver (Sun *et al.* 2017), considering the time- and spanwise-averaged turbulent flow over the aerofoil as the base flow.

The resolvent analysis is performed on a separate mesh from that used by the LES. The mesh for the resolvent analysis has a two-dimensional rectangular domain with the extent of $x/L_c \in [-15, 16]$ and $y/L_c \in [-6, 5]$, comprising approximately 0.11 million cells and giving the resulting discretised linear operator a dimension of 540840×540840. Compared to the LES mesh, the mesh for resolvent analysis is coarser over the aerofoil and in the wake, but is much finer in the upstream region of the aerofoil in order to resolve the forcing mode structures. The convergence of resolvent norm with respect to the domain extent and grid resolution has been reported in detail in Yeh & Taira (2019). At the far-field boundary and over the aerofoil, Dirichlet conditions are specified for the density and velocities and Neumann conditions are prescribed for the pressure in $\boldsymbol{q}$. At the outlet boundary, Neumann conditions are provided for all flow variables. The base-flow is two-dimensional, however, in contrast to the plane Poiseuille case, and we allow the perturbations to be three-dimensional by adopting a bi-global setting that decomposes $\boldsymbol{q}$ into spanwise Fourier modes with the wavenumber β.

Even though the linear operator is sparse, its large dimension requires special care. To efficiently deal with this operator, the python bindings for `PETSc` (Balay *et al.* 2021*a*,*b*, 1997), `petsc4py` (Dalcin *et al.* 2011) are used. This enables us to carry out the required linear algebra manipulations in parallel whilst keeping our code within the python environment. Specifically, `PETSc` is used together with the external library `MUMPS` (Amestoy *et al.* 2001, 2019) in order to provide the LU decomposition of the resolvent operator, and hence evaluate the action of the resolvent operator (and its adjoint) on a vector. To compare our sparse method with a traditional resolvent analysis, the SVD of the resolvent operator is found using a Lanczos SVD solver provided by the python bindings for `SLEPc` (Hernandez & Vidal 2005), `slepc4py` (Dalcin *et al.* 2011).

As in the previous case, we need to be careful regarding the choice of the state vector for the L_1-norm. To sparsify the location of any momentum input, rather than the individual components of the momentum, we must design our L_1-norm such that the momentum components are together. This requires some care for the aerofoil case, since it is compressible and the modes are not measured via an L_2-norm but via the Chu-norm (Chu 1965)

$$\|\boldsymbol{q}\|_E^2 = \int_\Omega \left(\frac{RT_0}{\rho_0}|\rho'|^2 + \rho_0\|\boldsymbol{u}'\|^2 + \frac{R\rho_0}{(\gamma-1)T_0}|T'|^2 \right) \mathrm{d}V, \tag{3.1}$$

which represents the energy contained in a perturbation in the absence of compression work. In defining the Chu-norm, we have used a dash $'$ to denote quantities derived from our state vector $\boldsymbol{q}$. Similarly a subscript 0 is used to denote quantities derived from the base-flow used for linearisation. This integral is discretised to $\|\boldsymbol{q}\|_E^2 = \boldsymbol{q}^H \boldsymbol{W}_E \boldsymbol{q}$ with $\boldsymbol{W}_E$ as a positive definite weight matrix. Taking the Cholesky decomposition $\boldsymbol{W}_E = \boldsymbol{M}^H \boldsymbol{M}$ gives the matrices needed for the resolvent description ($\boldsymbol{M}_{\boldsymbol{q}} = \boldsymbol{M}_{\boldsymbol{f}} = \boldsymbol{M}$). Hence, to keep momentum grouped in our sparsification, we split the components of the norm matrix $\boldsymbol{M}$ to form the state $\boldsymbol{T}(\boldsymbol{f}_{\boldsymbol{M}}) = (\boldsymbol{M}_\rho \boldsymbol{\rho}', \boldsymbol{M}_{\mathrm{KE}} \boldsymbol{\kappa}', \boldsymbol{M}_T \boldsymbol{T}')$ where $\boldsymbol{\kappa}' = \sqrt{(\rho \boldsymbol{u})' \odot (\rho \boldsymbol{u})' + (\rho \boldsymbol{v})' \odot (\rho \boldsymbol{v})' + (\rho \boldsymbol{w})' \odot (\rho \boldsymbol{w})'}$. Note that in defining $\boldsymbol{\kappa}'$, we have used the notation $(\rho \boldsymbol{u})' = \rho_0 \boldsymbol{u}' + \rho' \boldsymbol{u}_0$ for the streamwise linearised momentum component, with similar definitions for the spanwise $(\rho \boldsymbol{v})'$ and transverse $(\rho \boldsymbol{w})'$ linearised momentum components. This vector has the same L_2-norm as our full state vector. However, when taking the L_1-norm the kinetic energy is now grouped, ensuring that all velocity components are treated equally. It should be pointed out that now we have two additional components in the norm, specifically ρ' and T'. By not including these in the same component of the vector for the L_1-norm, they are treated separately by the sparsification. In essence, this means that the sparse resolvent does not only sparsify the spatial structure of any forcing but also sparsifies the actuation mechanism by choosing between a velocity-based, density-based or temperature-based forcing.

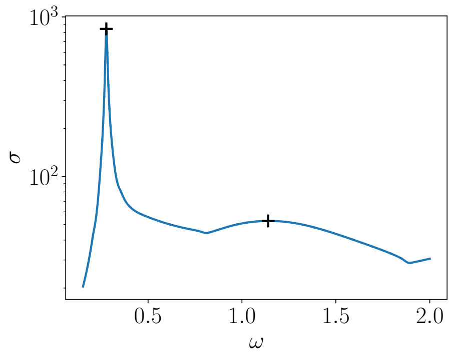


Figure 4: The optimal gains over frequency for Poiseuille flow. Our analysis focuses on the optimal gain which occurs at $\omega = 0.278$, and on the second peak at $\omega = 1.14$ (both shown with a black +)

4. Results

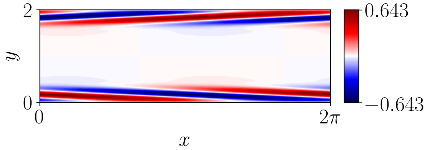
4.1. *Plane Poiseuille flow*

Let us first consider plane Poiseuille flow. This canonical example provides a good comparison with the work of Foures *et al.* (2013) and highlights the differences between using an alternative norm for a resolvent analysis and a transient growth study. It is important to note that as plane Poiseuille flow is a parallel flow, we could have proceeded with a local analysis, i.e. we could specify the streamwise wavenumber α and search for modes of the form

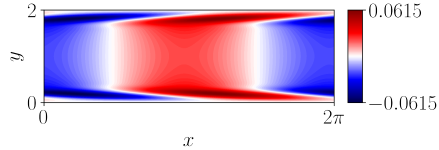
$$\boldsymbol{f}(x,y) = \boldsymbol{f}_\alpha(y)e^{i\alpha x} \quad \text{and} \quad \boldsymbol{u}(x,y) = \boldsymbol{u}_\alpha(y)e^{i\alpha x}. \tag{4.1}$$

However, as we are using a global (2D) analysis, the wavenumbers that our forcing and response can consist of are set by the aspect ratio of the domain. Taking $x \in [0, 2\pi]$ with periodic boundary conditions requires our wavenumbers to be integer, i.e., $\alpha \in \mathbb{N}$. Another artifact of using a 2D code for a parallel-flow is that the results do not change if the forcing and response modes are translated along the x-axis.

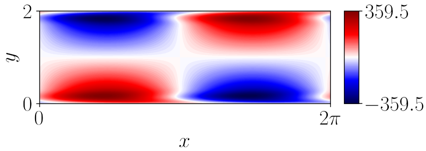
The gains obtained from a full resolvent analysis (i.e. by using an SVD) are shown in figure 4. This figure shows a strong peak at $\omega = 0.278$, followed by another peak at $\omega = 1.14$. Examining the forcing and response modes at these two frequencies (shown in figures 5 and 6, respectively) we observe that the first peak is associated with $\alpha = 1$ structures whereas the second peak corresponds to a higher wavenumber of $\alpha = 2$. This can be seen as a consequence of the flow being parallel and hints that by performing a two-dimensional analysis the optimal response is obtained at the wavenumber that has the maximum response from the one-dimensional analysis. With this in mind, the qualitative shape of the gain distribution agrees with those obtained in (Schmid & Henningson 2001) if the effects of perturbations in the spanwise direction, not considered in our analysis, are neglected. In both cases, the forcing mode consists of structures slanted against the shear, indicating that an Orr-mechanism is responsible for the gain in dynamics. Another interesting observation can be made by examining the phase velocity $k = \omega/\alpha$. We see that the phase velocity of the second peak is twice that of the first peak. The fact that the second peak is a faster disturbance



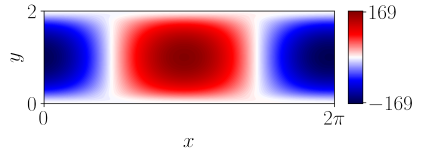
(a) Real part of the full forcing mode. (u-velocity component)



(b) Real part of the full forcing mode. (v-velocity component)

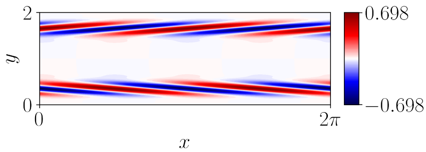


(c) Real part of the full response mode. (u-velocity component)

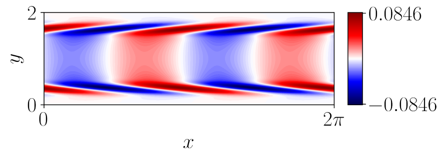


(d) Real part of the full response mode. (v-velocity component)

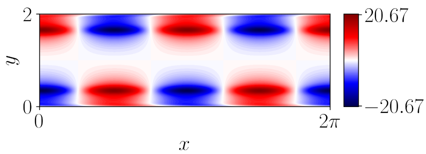
Figure 5: The real parts of the forcing and response mode obtained by SVD of the resolvent at $\omega = 0.278$. The forcing is unit-norm whereas the response mode has norm equal to the gain.



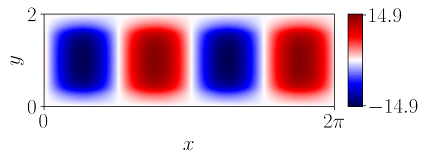
(a) Real part of full forcing mode. (u-velocity component)



(b) Real part of full forcing mode. (v-velocity component)



(c) Real part of full response mode. (u-velocity component)



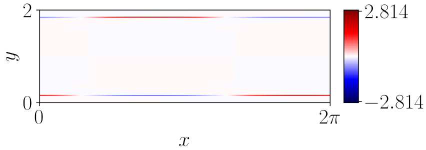
(d) Real part of full response mode. (v-velocity component)

Figure 6: The real parts of the forcing and response mode obtained by SVD of the resolvent at $\omega = 1.14$. The forcing is unit-norm whereas the response mode has norm equal to the gain.

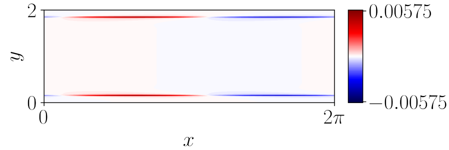
is also evident from the forcing mode being situated more centrally in the y-direction where the base-flow has a higher velocity.

We now turn our attention to the resolvent analysis results from the sparse optimisation procedure. The sparse forcing mode obtained for $\omega = 0.278$ is shown in figure 7. Firstly, it is clear from this figure that the forcing mode is more sparse than the full resolvent analysis. Indeed, instead of a series of slanted structures angled against the shear we now have thin stripes parallel to the walls. Interestingly, even though our vector $\boldsymbol{T}(\boldsymbol{f_M})$ was carefully chosen not to sparsify the separate velocity components at a given spatial location, the sparse forcing mode consists mainly of a u-component, indicating that this forcing is primarily in the direction of the wall.

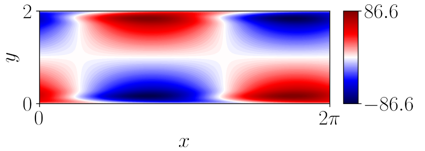
A striking feature of the sparse forcing mode is that it has maintained its



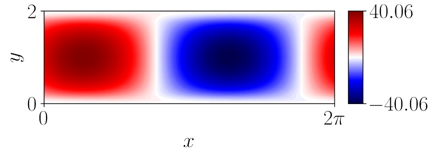
(a) Real part of the sparse forcing mode (u-velocity component).



(b) Real part of the sparse forcing mode (v-velocity component).



(c) Real part of the sparse response mode (u-velocity component).



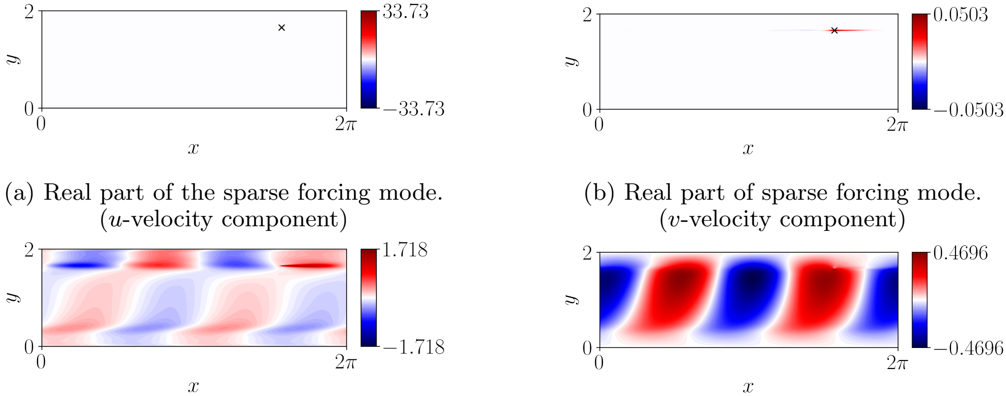
(d) Real part of the sparse response mode (v-velocity component).

Figure 7: The real parts of the forcing and response mode obtained by sparsification at $\omega = 0.278$. The forcing is unit-norm whereas the response mode has norm equal to the gain.

$\alpha = 1$ structure, i.e. it is still 2π-periodic in the x-direction. This is particularly enlightening since the strip-structure is not as sparse as the forcing mode could be, which would consist of just one element of the kinetic energy vector being filled, i.e. a single spatial location forcing. Therefore, the fact that the sparse procedure has chosen a less sparse structure indicates that forcing with this spatial wavenumber is crucial in achieving a high gain at this frequency. The location of these stripes can be hypothesised to be intrinsically linked to the $\alpha = 1$ structure using the concept of critical layers. A critical layer occurs at y^* where the base-flow velocity $U(y^*)$ is equal to the phase velocity k of a disturbance, and is central in causing instability in plane Poiseuille flow. Using the phase velocity k for a disturbance at $\omega = 0.278$ and $\alpha = 1$ implies a critical layer at $y^* \approx 0.150$, which is close to the y-location of the stripes which occur at $y = 0.155$. Hence, the sparse forcing mechanism can be summarised as forcing along (or just above) and parallel to the critical layer, with the v-velocity component, which does not contribute to this critical layer, being negligible.

Further evidence for the importance of $\alpha = 1$ forcing is shown in the response modes, which are also displayed in figure 7. The figure shows that the response modes stemming from the sparse forcing mode have the same structure as those from the full resolvent. As well as reinforcing that the $\alpha = 1$ forcing is critical for providing optimal amplification at this frequency, this observation also highlights the low-rank nature of the resolvent at this frequency. Even though the forcing shape is qualitatively different in the sparse case, the shape of the response is identical, disregarding arbitrary phase shifts, albeit with a lower magnitude. This agrees with previous observations, such as that of Rosenberg *et al.* (2019) where it is shown that when there is a large separation in singular values the shape of the forcing is less critical in exciting the dominant response. The lower magnitude is to be expected since our sparse forcing mode sacrifices some amount of energy to achieve a more localised spatial structure.

To provide a comparison between the results at different wavenumbers, we also carry out the sparse optimisation procedure at $\omega = 1.14$. Figure 8 shows the results, which differ quite significantly from the case of $\omega = 0.278$. In this case,



(a) Real part of the sparse forcing mode. (u-velocity component)

(b) Real part of sparse forcing mode. (v-velocity component)

(c) Real part of sparse response mode. (u-velocity component)

(d) Real part of sparse response mode. (v-velocity component)

Figure 8: The real parts of the forcing and response mode obtained using sparsification at $\omega = 1.14$. The forcing is unit-norm whereas the response mode has norm equal to the gain. The location of the sparse forcing mode is shown with $\times$.

the sparsification procedure has resulted in a single spatial forcing in u, with a negligible v-component which can be safely disregarded. In fact, the structure of the v-component is an artifact of the optimisation procedure which initially converged to a critical-layer mechanism similar to the previous case, before converging to a single spatial location. The reason for the different structure in this case can be attributed to the higher rank nature of the resolvent at this frequency. For $\omega = 0.278$, $\sigma_1^{(L_2)}/\sigma_2^{(L_2)} \approx 31$ whereas for $\omega = 1.14$, $\sigma_1^{(L_2)}/\sigma_2^{(L_2)} \approx 2$. The effect is that, even though an $\alpha = 2$ forcing is optimal, there is a less clear distinction between this forcing and the higher-order singular vectors. The result is that, unlike the previous case, there is less of a need for a specific α wavenumber to provide the optimal gain and, the sparsification procedure can take advantage of this to further sparsify the forcing structure. This is also evident in the response modes which are quite different from the SVD results. Finally, it is worth noting that the asymmetry of the forcing mode, with the single spatial location being located above the centreline, is due to our optimisation procedure converging to a local maximum. The reflection of this point about the centreline would also achieve the same value of the cost functional, hence representing another local optimum. Similar behaviour has been reported in the work of Foures *et al.* (2013).

4.2. *Flow past an aerofoil*

Secondly, we consider flow past a NACA 0012 aerofoil at an angle of attack of 9°, a chord-based Reynolds number of $Re = 23000$ and a free stream Mach number of $M = 0.3$ (see section 3.2 for the numerical details). In contrast to the plane Poiseuille example, this flow is unsteady and turbulent. Therefore, the mean-flow is used for linearisation. This time-averaged base-flow is shown in figure 9. Similarly to the work of Yeh & Taira (2019), and Ribeiro *et al.* (2020) who considered a resolvent analysis with the same base flow, we consider the resolvent modes at spanwise wavenumbers $\beta = 0$ and $20/\pi$. As the linear operator is unstable, a discounting parameter of $\alpha = 0.63$ is used (Jovanović

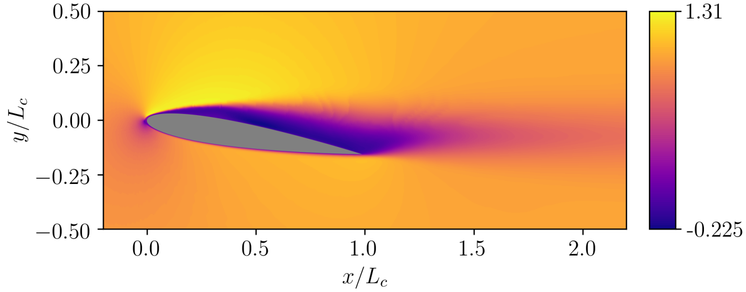
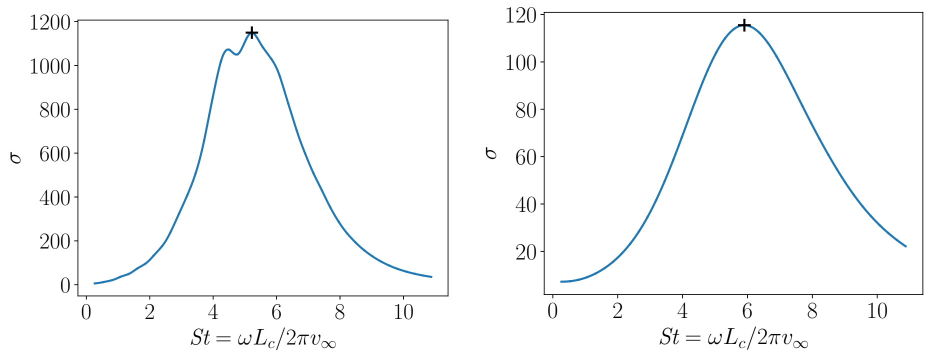


Figure 9: The streamwise velocity component of the base flow.



(a) $\beta = 0$

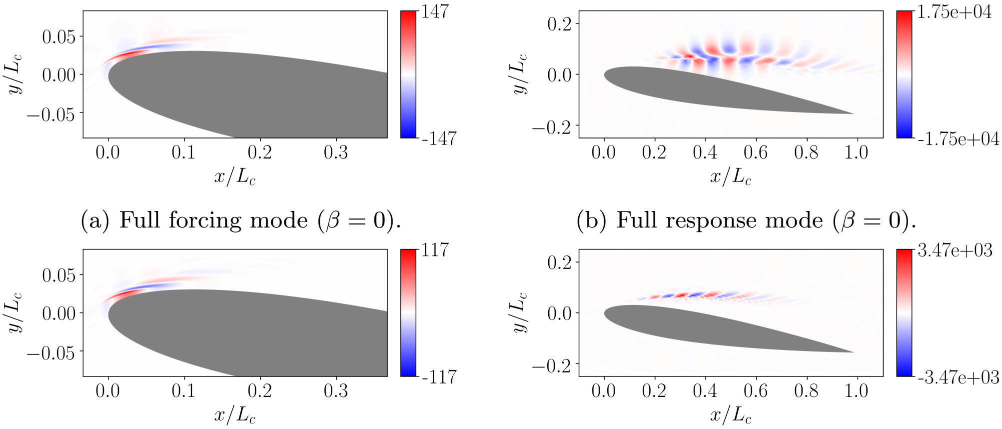
(b) $\beta = 20\pi$

Figure 10: The optimal gains against the Strouhal number for flow past an aerofoil. Our analysis will focus on the optimal gain which occurs at $St \approx 5.22$ for $\beta = 0$ and $St \approx 5.90$ for $\beta = 20\pi$ (both highlighted with a black $+$).

2004). The gain-frequency relationships are shown in figure 10. Similarly to the previous examples, we focus our subsequent analysis on the frequencies at which the peak gain is obtained.

Let us begin our analysis by briefly examining the modes obtained from a full resolvent analysis for our chosen parameters. For full details, see the paper by Yeh & Taira (2019). The spanwise linearised momentum component of the forcing mode and its corresponding response for our two spanwise wavenumber choices are showcased in figure 11. In both cases, the forcing is similar, consisting of slanted structures near the leading edge of the aerofoil on the suction side. The response modes are both located in the shear layer further downstream of the leading edge but differ in their spatial structures. For $\beta = 0$, there is a larger spatial support with the mode shape extending both vertically and horizontally about the shear layer, whereas for $\beta = 20\pi$ the response aligns much more tightly with the shear. This agrees with the findings of Yeh & Taira (2019) who state that for an increased forcing frequency or wavenumber the shear layer is needed to support the resulting smaller-scale fluctuations.

Now that we have characterised the L_2-norm SVD-based results, we turn our attention towards the sparse-optimisation-based modes. Figure 12 shows the sparse forcing and response modes. In both spanwise wavenumber cases, the optimisation procedure has identified a single spatial momentum-based structure



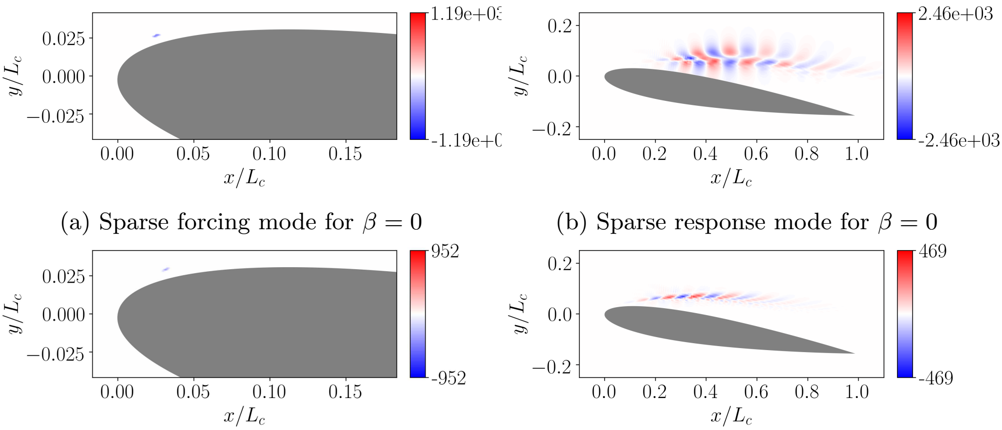
(a) Full forcing mode ($\beta = 0$).

(b) Full response mode ($\beta = 0$).

(c) Full forcing mode ($\beta = 20\pi$).

(d) Full response mode ($\beta = 20\pi$).

Figure 11: The full resolvent modes for aerofoil flow at $St = 5.22$ for $\beta = 0$ and $St = 5.90$ for $\beta = 20\pi$. The linearised component of the streamwise momentum is shown.



(a) Sparse forcing mode for $\beta = 0$

(b) Sparse response mode for $\beta = 0$

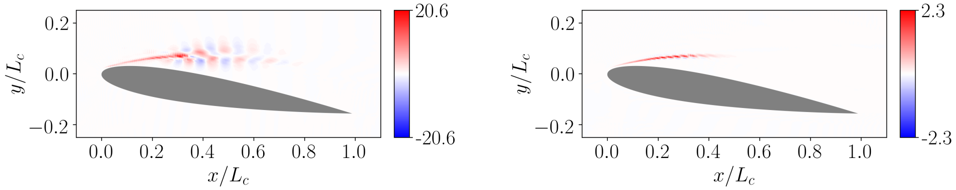
(c) Sparse forcing mode for $\beta = 20\pi$

(d) Sparse response mode for $\beta = 20\pi$

Figure 12: The sparse resolvent modes for aerofoil flow at $St = 5.22$ for $\beta = 0$ and $St = 5.90$ for $\beta = 20\pi$. The linearised component of the streamwise momentum is shown.

for the forcing mode, with the density and pressure contributions being negligible. This illustrates the effectiveness of the sparse procedure, which in this case was not only able to sparsify the spatial structure, but has also sparsified the physical makeup of the forcing, indicating that a momentum-based forcing provides the optimal sparse gain in dynamics. It is also worth highlighting that, even though we have only a single-location forcing that, the response mode is qualitatively the same as the full case.

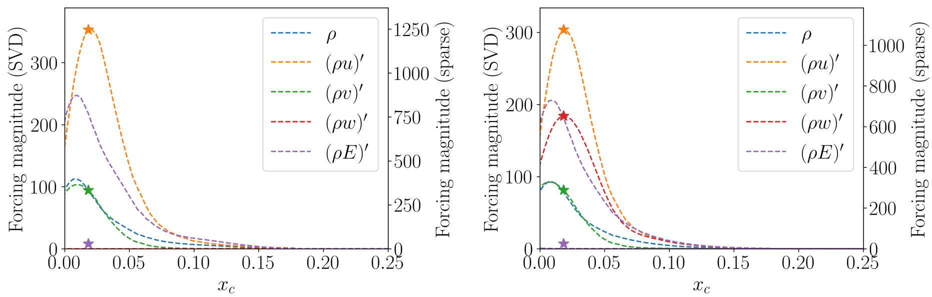
To provide additional insight into the chosen spatial location, we now examine the resolvent wavemakers, which are shown in figure 13. For both values of β, the wavemakers display a large positive region in the mean-flow shear layer. This is not unexpected, since regions of shear translate to regions of high non-



(a) Wavemaker ($\beta = 0$).

(b) Wavemaker ($\beta = 20\pi$).

Figure 13: The resolvent wavemakers for aerofoil flow at $St = 5.22$ for $\beta = 0$ and $St = 5.90$ for $\beta = 20\pi$. The linearised component of the streamwise momentum is shown.



(a) Surface forcing on the suction side of the aerofoil ($\beta = 0$).

(b) Surface forcing on the suction side of the aerofoil ($\beta = 20\pi$).

Figure 14: The surface forcing on the suction side of the aerofoil at $St = 5.22$ for $\beta = 0$ and $St = 5.90$ for $\beta = 20\pi$. The sparse components are shown with a star.

normality in the linearised Navier–Stokes operator, which underpins sensitive areas for forcing. In fact, in the full-SVD forcing modes, we directly see this, as the forcing modes in both cases are primarily located in this shear region. The sparse forcing locations are also situated in this region and are located near the maximum value of the full-SVD forcing mode. Whilst this may show that choosing the largest value of the forcing mode is a good candidate for the sparse forcing mode, we emphasise that the optimisation procedure is not biased by any knowledge of the full forcing mode, and all physical mechanisms and spatial locations are weighted equally.

Whilst forcing in the shear region may provide the optimal response, it is rather impractical for flow actuation purposes. Therefore, we conclude this section by using the windowing matrix $\boldsymbol{B}$ to conduct our analysis on the surface of the aerofoil. Figure 14 shows the forcing distribution along the suction side of the aerofoil as a function of the distance along the chord x_c. In the full resolvent analysis, most of the forcing is concentrated near the leading edge of the aerofoil, agreeing with the non-windowed case. In both spanwise wavenumber cases, the sparse mode is once again a single-point momentum-based forcing and is located at the maximum value for the kinetic energy of the full forcing mode. This provides the optimal compromise between forcing with $(\rho u)'$ at its maximum value and $(\rho v)'$ at its maximum, which in the full case is located to the left of $(\rho u)'$. Even though there is a $(\rho E)'$-component, this is simply a consequence of the kinetic part of the energy, since $(\rho E)' = \rho' \|\boldsymbol{u}_0\|^2/2 + \rho_0 \boldsymbol{u}' \cdot \boldsymbol{u}_0 + P'/(\gamma - 1)$, and there is no thermodynamic contribution to the linearised total-energy. The importance of grouping momentum together into one coherent strategy is evident from the figure, as the directional information of the actuation is crucial in both the full and sparse resolvent analyses to achieve the optimal gain. This information would otherwise be lost. Moreover, by grouping the momentum together we strike a compromise between choosing a forcing that is optimal for each isolated velocity component.

5. Conclusion

By reformulating an optimal-input analysis as a Riemannian optimisation problem, we are able to tailor a resolvent analysis to uncover sparse forcing modes and their corresponding response modes. By designing a cost functional based on the ratio of the energy gain to the L_1-norm of the forcing, we are able to find forcing modes that provide the largest gain whilst being spatially sparse. To test the method within the context of resolvent analyses performed on steady base-flows and time-averaged mean-flows, we considered two flow examples: plane Poiseuille flow in the linearly stable regime and the turbulent flow past an aerofoil.

For plane Poiseuille flow, two forcing frequencies were considered. At the first frequency, located at the maximum gain of the full-SVD analysis, the sparse forcing mode consisted of an $\alpha = 1$ stripe at a single y-location, just above the critical layer. Conversely, for the second frequency located at the second peak of the full analysis, the forcing consisted of a single spatial forcing near the $\alpha = 2$ critical layer. For the first case, utilising an $\alpha = 1$ forcing is critical in obtaining a high gain, and therefore the optimisation only sparsifies the forcing in the y-direction. However, in the second case, there is a much lower separation in the effectiveness of different forcing mechanisms, as indicated by the ratio of the singular values, meaning that the sparse procedure is able to sparsify further whilst still providing a large gain.

In the turbulent flow past an aerofoil, all sparse modes consisted of single spatial locations, with the sparsification procedure also identifying momentum-based forcing as the optimum physical mechanism. For two different spanwise wavenumbers an analysis of the resolvent wavemakers shows that forcing in the shear layer provides the optimal gain, with the sparse procedure focusing on the location of the maximal value of the full forcing modes. To identify an implementable actuator position, we also considered a windowed analysis where the forcing modes are confined to the surface of the aerofoil. Again, we achieve single-point momentum-based actuation positions which are found to be a compromise among the optimal locations for each independent velocity component. This emphasises the importance of designing an appropriate vector for the L_1-norm, as the directional information would have been lost had we not grouped momentum together.

Overall, the sparse optimisation procedure provides an unbiased optimal sparsification of the flow and is able to adapt to the different forcing strategies available at different frequencies. Although the aerofoil results show that choosing the maxima of the SVD is a good candidate for a sparse forcing vector, the plane Poiseuille example shows that both single-point and multi-point forcing modes can be found depending on the low-rank nature and physical mechanisms furnished by the resolvent. Based on our results, it can be postulated that in more complex systems, such as those stemming from aeroacoustic or combustion problems, where multiple physical mechanisms are at play, the sparse resolvent would be able to adapt to the optimal physical mechanisms present at each frequency, and would even be able to combine in an optimal sparse way these different mechanisms in order to achieve the largest gain. Investigating the sparse optimisation procedure on these types of flows would therefore be an interesting future direction of study. Furthermore, as the choice of cost functional for the purpose of sparsification is not unique, the design of other functionals, such as those that could allow for a tuning of sparsity versus gain, provides another area for future investigation.

Although we have not considered them in our study, we further note that recent efforts have been made to extend resolvent analysis to both periodic flows (Padovan *et al.* 2020) and also to the non-linear regime (Rigas *et al.* 2021). As the techniques we have presented carry over to both these cases without significant modification, these present interesting avenues for future investigations. Moreover, the usefulness of Riemannian optimisation in tailoring input-output analyses to specific flow applications is not limited to our sparse analysis. As well as being able to design cost functionals in order to uncover different aspects of the resolvent, the manifold to which we confine the forcing modes can be changed. The result is a rich landscape of possibilities in which resolvent analyses can be extended, with the traditional SVD-based approach being just one such choice.

Funding. KT and PJS thank the support from the Army Research Office (grant: W911NF-21-1-0060, program officer: Matthew J. Munson). KT also acknowledges the support from the Office of Naval Research (grant: N0014-19-1-2460, program officer: David R. Gonzalez).

Declaration of interests. The authors report no conflict of interest.

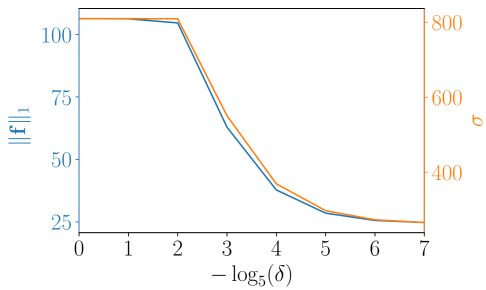
Author ORCID. C. S. Skene, https://orcid.org/0000-0003-0994-2013;
C.-A Yeh, https://orcid.org/0000-0003-0426-8381;
P. J. Schmid, https://orcid.org/0000-0002-2257-8490;
K. Taira, https://orcid.org/0000-0002-3762-8075;

Appendix A. Convergence results

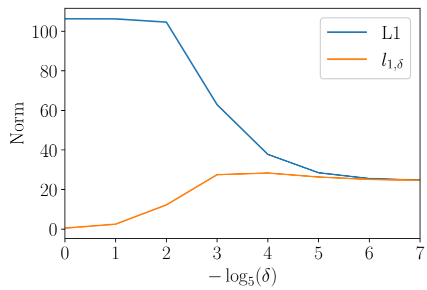
In this appendix, we consider the numerical details of the optimisation procedure. In order to provide an overview we will present the results stemming from the plane Poiseuille example at $\omega = 0.278$ considered in section 4 which is representative of all cases considered in this paper.

Figure 15 shows how the results of the optimisation procedure depend on δ. We can see from figures 15a and 15c that there is an initial region in which the optimisation procedure has a large L_1-norm and that the gain is in line with that obtained from the SVD. After $\delta = 2^{-5}$ the pseudo-Huber norm starts to behave more like the L_1-norm, as shown in figures 15b and 15d, and both the gain and the L_1-norm of the forcing decrease. At $\delta = 5^{-7}$ the pseudo-Huber norm and the L_1-norm have converged, and we can stop the optimisation procedure. These figures also highlight the strong dependence of the gain on the L_1 norm of the forcing for these examples, with the gain decreasing almost exactly in line with the L_1 norm.

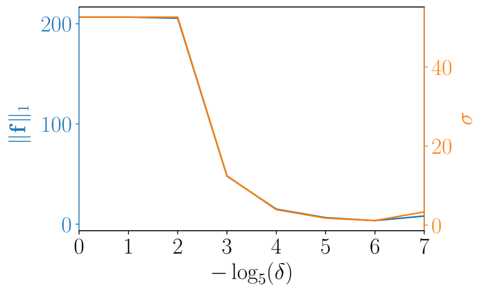
Also shown in figures 16a and 16b is the norm of the gradient provided by the optimisation procedure as a function of the number of iterations. For the case of $\omega = 0.278$, the number of iterations is in line with the work of Foures *et al.* (2013), with perhaps a few more iterations needed in our case. The number of iterations required as well as the non-smoothness of the gradient norms is indicative of



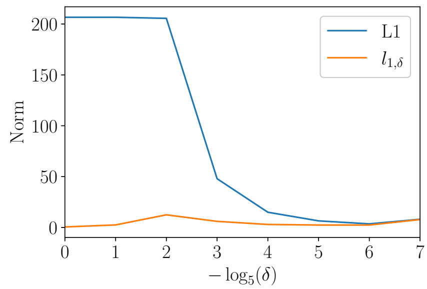
(a) L_1 norm and gains with delta ($\omega = 0.278$)



(b) Convergence of the pseudo-Huber norm to the true L_1 norm ($\omega = 0.278$)

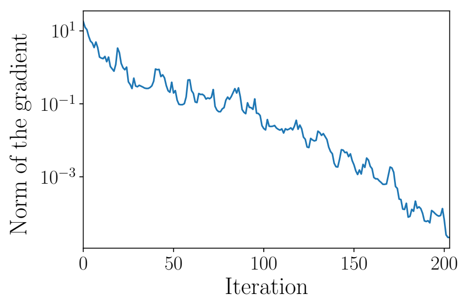


(c) L_1 norm and gains with delta ($\omega = 1.14$)

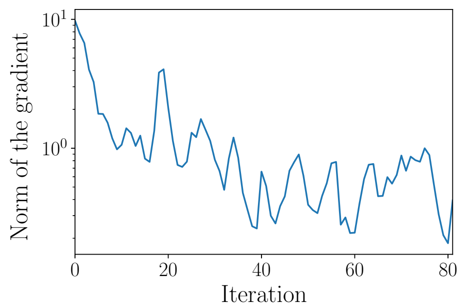


(d) Convergence of the pseudo-Huber norm to the true L_1 norm ($\omega = 1.14$)

Figure 15: Dependence of the results on δ for the plane Poiseuille flow examples



(a) Convergence behaviour of the conjugate gradient algorithm for plane Poiseuille flow at $\omega = 0.278$ and $\delta = 5^{-7}$.



(b) Convergence behaviour of the conjugate gradient algorithm for plane Poiseuille flow at $\omega = 1.14$ and $\delta = 5^{-7}$.

Figure 16

the difficulty of the gradient-based optimisation. This is especially highlighted by figure 16b, where the optimisation terminates earlier due to a stagnation of the cost functional. This showcases the importance of using both a relaxation parameter δ as well as an optimisation procedure such as the conjugate gradient algorithm in order to achieve converged results.

REFERENCES

Absil, P.-A., Mahony, R. & Sepulchre, R. 2007 *Optimization Algorithms on Matrix Manifolds*. USA: Princeton University Press.

Amestoy, P.R., Buttari, A., L'Excellent, J.-Y. & Mary, T. 2019 Performance and Scalability of the Block Low-Rank Multifrontal Factorization on Multicore Architectures. *ACM Transactions on Mathematical Software* **45**, 2:1–2:26.

Amestoy, P.R., Duff, I. S., Koster, J. & L'Excellent, J.-Y. 2001 A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications* **23** (1), 15–41.

Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., May, D. A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B. F., Zampini, S., Zhang, H. & Zhang, H. 2021*a* PETSc users manual. *Tech. Rep.* ANL-95/11 - Revision 3.15. Argonne National Laboratory.

Balay, S., Abhyankar, S., Adams, M. F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Gropp, W. D., Kaushik, D., Knepley, M. G., May, D. A., McInnes, L. C., Mills, R. T., Munson, T., Rupp, K., Sanan, P., Smith, B. F., Zampini, S., Zhang, H. & Zhang, H. 2021*b* PETSc Web page. Https://petsc.org/.

Balay, S., Gropp, W. D., McInnes, L. C. & Smith, B. F. 1997 Efficient management of parallelism in object oriented numerical software libraries. In *Modern Software Tools in Scientific Computing* (ed. E. Arge, A. M. Bruaset & H. P. Langtangen), pp. 163–202. Birkhauser Press.

Boumal, N. & Absil, P.-A. 2015 Low-rank matrix completion via preconditioned optimization on the Grassmann manifold. *Linear Algebra and its Applications* **475**, 200–239.

Boumal, N., Mishra, B., Absil, P.-A. & Sepulchre, R. 2014 Manopt, a Matlab toolbox for optimization on manifolds. *Journal of Machine Learning Research* **15**, 1455–1459.

Brès, G. A., Ham, F. E., Nichols, J. W. & Lele, S. K. 2017 Unstructured large-eddy simulations of supersonic jets. *AIAA Journal* **55** (4), 1164–1184.

Bube, K. P. & Langan, R. T. 1997 Hybrid $\ell 1$ / $\ell 2$ minimization with applications to tomography. *GEOPHYSICS* **62** (4), 1183–1195.

Bube, K. P. & Nemeth, T. 2007 Fast line searches for the robust solution of linear systems in the hybrid $\ell 1$ / $\ell 2$ and Huber norms. *GEOPHYSICS* **72** (2), A13–A17.

Cattafesta, L. N. & Sheplak, M. 2011 Actuators for active flow control. *Annu. Rev. Fluid Mech.* **43**, 247–272.

Chu, B. T. 1965 On the energy transfer to small disturbances in fluid flow (part I). *Acta Mechanica* **1** (3), 215–234.

Dalcin, L. D., Paz, R. R., Kler, P. A. & Cosimo, A. 2011 Parallel distributed computing using python. *Advances in Water Resources* **34** (9), 1124 – 1139, New Computational Methods and Software Tools.

Farrell, B. F. & Ioannou, P. J. 1993 Stochastic forcing of the linearized Navier–Stokes equations. *Physics of Fluids A: Fluid Dynamics* **5** (11), 2600–2609.

Fosas de Pando, M. 2020 IBMOS: Immersed boundary method optimization and stability. https://doi.org/10.5281/zenodo.3757783.

Fosas de Pando, M. & Schmid, P. J. 2017 Optimal frequency-response sensitivity of compressible flow over roughness elements. *Journal of Turbulence* **18** (4), 338–351.

Fosas de Pando, M., Schmid, P. J. & Lele, S. K. 2014 Parametric sensitivity for large-scale aeroacoustic flows. In *Proceedings of the 2014 Summer Program*, pp. 365–374. Center for Turbulence Research, Stanford University.

Foures, D. P. G., Caulfield, C. P. & Schmid, P. J. 2013 Localization of flow structures using ∞-norm optimization. *Journal of Fluid Mechanics* **729**, 672–701.

Giannetti, F. & Luchini, P. 2007 Structural sensitivity of the first instability of the cylinder wake. *Journal of Fluid Mechanics* **581**, 167–197.

Hernandez, V.and Roman, J. E. & Vidal, V. 2005 Slepc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Trans. Math. Softw.* **31** (3), 351–362.

Huang, W., Absil, P.-A. & Gallivan, K. A. 2016 A Riemannian BFGS method for nonconvex optimization problems. In *Numerical Mathematics and Advanced Applications ENUMATH 2015* (ed. B. Karasözen, M. Manguoğlu, M. Tezer-Sezgin, S. Göktepe &

Ö. Uğur), *Lecture Notes in Computational Science and Engineering* 112, pp. 627–634. Cham: Springer International Publishing.

Jeun, J., Nichols, J. W. & Jovanović, M. R. 2016 Input-output analysis of high-speed axisymmetric isothermal jet noise. *Physics of Fluids* **28** (4), 047101, arXiv: https://doi.org/10.1063/1.4946886.

Jovanović, M. R. 2004 Modeling, analysis, and control of spatially distributed systems. Ph. D. thesis, University of California, Santa Barbara.

Jovanović, M. R. & Bamieh, B. 2005 Componentwise energy amplification in channel flows. *Journal of Fluid Mechanics* **534**, 145–183.

Khalighi, Y., Nichols, J. W., Ham, F., Lele, S. K. & Moin, P. 2011 Unstructured large eddy simulation for prediction of noise issued from turbulent jets in various configurations. AIAA Paper 2011-2886.

Liu, Q., Sun, Y., Yeh, C.-A., Ukeiley, L. S., Cattafesta, L. N. & Taira, K. 2021 Unsteady control of supersonic turbulent cavity flow based on resolvent analysis. *Journal of Fluid Mechanics* **925**, A5.

Luhar, M., Sharma, A. S. & McKeon, B. J. 2014 Opposition control within the resolvent analysis framework. *Journal of Fluid Mechanics* **749**, 597–626.

McKeon, B. J. & Sharma, A. S. 2010 A critical-layer framework for turbulent pipe flow. *Journal of Fluid Mechanics* **658**, 336–382.

Padovan, A., Otto, S. E. & Rowley, C. W. 2020 Analysis of amplification mechanisms and cross-frequency interactions in nonlinear flows via the harmonic resolvent. *Journal of Fluid Mechanics* **900**, A14.

Paladini, E., Beneddine, S., Dandois, J., Sipp, D. & Robinet, J.-C. 2019 Transonic buffet instability: From two-dimensional airfoils to three-dimensional swept wings. *Phys. Rev. Fluids* **4**, 103906.

Pringle, C. C. T., Willis, A. P. & Kerswell, R. R. 2012 Minimal seeds for shear flow turbulence: using nonlinear transient growth to touch the edge of chaos. *Journal of Fluid Mechanics* **702**, 415–443.

Qadri, U. A. & Schmid, P. J. 2017 Frequency selection mechanisms in the flow of a laminar boundary layer over a shallow cavity. *Phys. Rev. Fluids* **2**, 013902.

Ribeiro, J. H. M., Yeh, C.-A. & Taira, K. 2020 Randomized resolvent analysis. *Phys. Rev. Fluids* **5**, 033902.

Rigas, G., Sipp, D. & Colonius, T. 2021 Nonlinear input/output analysis: application to boundary layer transition. *Journal of Fluid Mechanics* **911**, A15.

Rosenberg, K., Symon, S. & McKeon, B. J. 2019 Role of parasitic modes in nonlinear closure via the resolvent feedback loop. *Phys. Rev. Fluids* **4**, 052601.

Schmid, P. J. & Brandt, L. 2014 Analysis of Fluid Systems: Stability, Receptivity, Sensitivity: Lecture notes from the FLOW-NORDITA Summer School on Advanced Instability Methods for Complex Flows, Stockholm, Sweden, 2013. *Applied Mechanics Reviews* **66** (2), 024803.

Schmid, P. J. & Henningson, D. S. 2001 *Stability and Transition in Shear Flows*, *Applied Mathematical Sciences*, vol. 142. New York, NY: Springer-Verlag.

Sun, Y., Taira, K., Cattafesta, L. N. & Ukeiley, L. S. 2017 Biglobal instabilities of compressible open-cavity flows. *Journal of Fluid Mechanics* **826**, 270–301.

Taira, K., Brunton, S. L., Dawson, S. T. M., Rowley, C. W., Colonius, T., McKeon, B. J., Schmidt, O. T., Gordeyev, S., Theofilis, V. & Ukeiley, L. S. 2017 Modal analysis of fluid flows: An overview. *AIAA J.* **55** (12), 4013–4041.

Taira, K. & Colonius, T. 2007 The immersed boundary method: A projection approach. *Journal of Computational Physics* **225** (2), 2118 – 2137.

Taira, K., Hemati, M. S., Brunton, S. L., Sun, Y., Duraisamy, K., Bagheri, S., Dawson, S. & Yeh, C.-A. 2020 Modal analysis of fluid flows: Applications and outlook. *AIAA J.* **58** (3), 998–1022.

Toedtli, S. S., Luhar, M. & McKeon, B. J. 2019 Predicting the response of turbulent channel flow to varying-phase opposition control: Resolvent analysis as a tool for flow control design. *Phys. Rev. Fluids* **4**, 073905.

Townsend, J., Koep, N. & Weichwald, S. 2016 Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *Journal of Machine Learning Research* **17** (137), 1–5.

TREFETHEN, L. N., TREFETHEN, A. E., REDDY, S, C. & DRISCOLL, T. A. 1993 Hydrodynamic stability without eigenvalues. *Science* **261** (5121), 578–584.

VREMAN, A. W. 2004 An eddy-viscosity subgrid-scale model for turbulent shear flow: Algebraic theory and applications. *Physics of Fluids* **16** (10), 3670–3681.

YEH, C.-A., BENTON, S. I., TAIRA, K. & GARMANN, D. J. 2020 Resolvent analysis of an airfoil laminar separation bubble at $Re = 500000$. *Phys. Rev. Fluids* **5** (8), 1–24.

YEH, C.-A. & TAIRA, K. 2019 Resolvent-analysis-based design of airfoil separation control. *Journal of Fluid Mechanics* **867**, 572–610.