Nonlinear Dynamics (2026) 114:313
https://doi.org/10.1007/s11071-025-12123-x

RESEARCH

Construction of quadratic, conditional Lyapunov functions using SOS method for finite-dimensional shear flow models

Péter Tamás Nagy[1]

Received: 14 May 2025 / Revised: 19 November 2025 / Accepted: 5 December 2025 / Published online: 3 March 2026

Abstract
Stability analysis is a crucial tool for studying dynamical systems in mathematics, physics, and engineering. Lyapunov functions provide a framework for determining the stability of these systems, but constructing them is often challenging for non-linear systems. This paper focuses on using sum-of-squares (SOS) methods to construct quadratic Lyapunov functions for proving the local stability of finite-dimensional systems. These systems are designed to share certain properties of shear flows and are therefore described by second-order polynomials. Additionally, the system matrix of the linear part is non-normal, which presents challenges in constructing a Lyapunov function. Two established SOS-based algorithms from the literature (SOS1 and SOS2) are compared based on the size of the provable region of attraction (ROA), computational time, and the maximum allowable degrees of freedom of the system. Furthermore, this paper introduces a novel modification to the SOS2 algorithm, termed SOS2m, which aims to provide a larger ROA than the original SOS2 method with minimal additional computational cost. This new SOS2m method is shown to strike an effective balance between accuracy and computational efficiency. The presented methods, while demonstrated on shear flow models, are extendable to other dynamical systems described by polynomial functions. A further motivation for this research is to advance analysis methods for investigating subcritical laminar-turbulent transition and to develop methodologies for estimating permissible perturbation levels. To demonstrate their effectiveness, the three algorithms (SOS1, SOS2, and SOS2m) are applied to a simplified model of laminar-turbulent transition and truncated finite-dimensional reduced-order models of Poiseuille and Couette flows.

Keywords Sum of squares method · Conditional Lyapunov function · Semidefinite programming · Conditional stability · Non-linear stability · Non-normal system

1 Introduction

The investigation of stability in dynamical systems, or their controlled counterparts, is a fundamental pursuit in mathematics, physics, and engineering. One approach to proving stability is through the construction of a Lyapunov function. For finite-dimensional linear dynamical systems, this process is straightforward, involving the calculation of the eigenvalues of the system matrix. However, for non-linear systems, constructing a Lyapunov function becomes significantly more challenging. When focusing on dynamical systems described by polynomial functions – a common scenario in fluid dynamics and various other physical and engineering applications – it is often practical to assume that the Lyapunov function is also polynomial. The challenge then lies in proving that certain polynomials are positive or negative definite. Unfortunately, verifying the non-negativity of a polynomial is generally NP-hard [1]. A more tractable sufficient condition is to represent the polynomial as a sum of squares (SOS) of other polynomials, a computational technique introduced over two decades ago [2–4].

In fluid dynamics, Goulart and Chernyshenko [5] pioneered the application of the SOS method for constructing Lyapunov functions to prove the global stability of shear flows. Constructing Lyapunov functionals for such infinite-dimensional dynamical systems described by partial differential equations is extremely challenging. One possible way to simplify the problem is by projecting the dynamics onto a certain set of modes and constructing a finite-dimensional reduced-order model. However, the com-

✉ Péter Tamás Nagy
pnagy@hds.bme.hu

1 Department of Hydrodynamic Systems, Faculty of Mechanical Engineering, Budapest University of Technology and Economics, Műegyetem rkp. 3., Budapest H-1111, Budapest, Hungary

putational demand of SOS methods scales significantly with increased degrees of freedom, often preventing the inclusion of a sufficient number of modes for engineering accuracy. To address this issue, Goulart and Chernyshenko [5] proposed an "infinite-dimensional" model that incorporates the effect of otherwise truncated modes – termed "tails" – into the stability investigation. Utilizing linear eigenvalue problems, the worst-case effect of the tail can be estimated on the stability of the infinite-dimensional system. This approach was later applied to an infinite-dimensional contrived version of rotating Couette flow by Huang et al. [6], yielding improved results compared to the well-known energy method [7]. More recently, Fuentes et al. [8] provided sharper bounds on tail influences and applied the method to two-dimensional Poiseuille flow, improving the known threshold for global stability by constructing infinite-dimensional high-order Lyapunov functionals with SOS methods.

In addition to determining global stability, assessing the local stability of the laminar state equilibrium point in shear flows is an equally important scientific challenge. The threshold amplitude of the smallest perturbation that can trigger transition to turbulence is a direct measure of the stability's strength. In pipe flow, the laminar state is linearly stable for all Reynolds numbers (Re). However, it is known to be non-linearly unstable for $Re \gtrsim 2000$. The paradox of subcritical transition is that while the laminar flow is stable to infinitesimal perturbations, the threshold amplitude of perturbations rapidly decreases towards zero as Re increases, making the laminar state virtually impossible to observe in real-world experiments at high Re. In the context of controlled systems, this threshold amplitude serves as a quantitative metric to evaluate the efficacy and robustness of the controller.

Galdi and Straughan [9] showed possible analytical ways to construct an energy function to prove global or local stability and they demonstrated conditional stability in the case of a basic model for the motion of microorganisms. Unfortunately, the general application of that method is challenging. The infinite-dimensional model of Goulart and Chernyshenko [5] is arguably a more promising way to also describe the local stability of the laminar state in a shear flow. It can be expected that such methods better approximate the region of attraction (ROA) if the number of modes, and thus the degrees of freedom, is increased. This paper investigates the applicability of SOS programs to non-normal systems, focusing mainly on accuracy and determining the largest system size that can still be investigated. However, this study is confined to finite-dimensional models; the effect of the tail has not been investigated yet.

In the literature, many algorithms have been developed to estimate the ROA of an equilibrium solution. Here, some of the most known methods are presented. Jarvis-Wloszek [10] proposed such a calculation using SOS methods through the "expanding D algorithm" and "expanding interior algorithm". While Tan et al. [11] introduced a similar algorithm solved by the PENBMI solver, it was found that converting the problem into a series of semidefinite programs (SDP) is more convenient due to advancements in SDP solvers. This approach, along with similar SOS programs solved as a sequence of SDP optimizations, has been utilized for simple dynamical systems in several studies [11–15].

An alternative SOS program (referred to as SOS2 in this paper) was proposed by Liu et al. [16], building on the theoretical work of Anderson and Papachristodoulou [17] for investigating the conditional stability of flows. This approach, which focuses only on quadratic Lyapunov functions, simplifies the calculation of the ROA by solving an eigenvalue problem instead of relying on an additional SOS constraint. This method substantially reduces computational costs and enhances robustness.

There is also a convex formulation for inner-approximating a basin of attraction without constructing a Lyapunov function, introduced by Henrion and Korda [18], which avoids the necessary iteration steps of the previously mentioned SOS programs. However, in their equations, time appears as an explicit variable in the polynomials, unlike in the previous cases. Additionally, this method requires four SOS constraints, resulting in a significantly larger optimization problem. Furthermore, it requires the explicit specification of a time horizon and a target domain, parameters that play a critical role in shaping the results. It can be assumed that this method is applicable only to relatively small systems; therefore, this formulation is excluded from the comparison in this study.

Other methods also exist for calculating the ROA in dynamical systems. Without aiming for completeness, a few relevant methods applied in fluid dynamics related problems are mentioned. For instance, bounding the non-linear terms of the dynamical system with linear matrix inequalities allows for faster calculations than SOS methods, albeit with significantly lower accuracy in predicting threshold amplitudes. Two groups, Kalur et al. [15] and Liu et al. [16], developed a quadratic constrained (QC) method, modeling the non-linear term as a bounded excitation of the linear system and applying it to simple models of laminar-turbulent transition. While computationally efficient, the predicted threshold amplitude significantly underperforms, usually by at least one order of magnitude, compared to SOS methods. Toso et al. [19] applied another bounding method to low-dimensional flow models, calculating an approximately 30% higher threshold amplitude than the QC method. Additionally, Nerli et al. [20] proposed a method that employs quadratic Lyapunov functions to verify stability via non-linear optimization. Recently, Nagy [21] conducted a comprehensive analysis of this method, extending its application to significantly higher-dimensional dynamical systems. Focusing on computational speed and engineering robustness, Habib [22] proposed a rapid iterative algorithm that estimates the Local Integrity Measure (LIM) defined as the radius of the largest hypersphere entirely contained within the basin of attraction. Alternatively, with proper interpretation and necessary caveats, even neural networks can be utilized to construct Lyapunov functions [23].

This paper focuses on the conditional stability of finite-dimensional systems sharing certain properties of real fluid dynamical systems. Two established SOS algorithms (SOS1 and SOS2) are compared based on the size of the provable ROA, computational time, and the maximum allowable degrees of freedom of the system. Crucially, this study also proposes a novel enhancement to the SOS2 algorithm, termed SOS2m. This modification involves algorithmically updating the shape of the region where the Lyapunov derivative is negative to better align with the ROA, which, as will be shown, can significantly improve the provable ROA with minimal additional computational cost compared to SOS1.

Section 2 outlines the original SOS programs and the proposed SOS2m improvement. Section 3.1 demonstrates the application of these methods to a simple four-DoF transition model. In Section Appendix A, the construction of finite-dimensional models of Couette and Poiseuille flows with varying degrees of freedom is presented to create models with an increased number of degrees of freedom while still retaining a few important properties of shear flows. The three algorithms are compared for these systems in Section 3.2. Finally, concluding remarks are provided in Section 4.

2 Construction of quadratic Lyapunov functions

The dynamical system under investigation is described by

$$\frac{\mathrm{d}q_i}{\mathrm{d}t} = f_i(\boldsymbol{q}) \tag{1}$$

an autonomous, non-linear ordinary differential equation system, where q_i represents the dependent variables of the dynamical system forming the vector $\boldsymbol{q}$, and i ranges from 1 to n, the degrees of freedom of the system. The functions f_i describing the system are polynomials of the q_i variables. Here, only second-order polynomial f_i functions are considered; however, the presented methods are applicable to higher-order polynomials. Additionally, the origin is considered as the equilibrium point under investigation, implying that $f_i(\mathbf{0}) = 0$ for any i.

2.1 First Sum-of-Squares Program - SOS1

The first SOS program [10–13, 15], employed to calculate the ROA for conditionally stable systems, is formulated as follows:

$$V - l_1 \in \Sigma, \tag{2}$$

$$-\left[(\beta - p)s_1 + (V - \gamma)\right] \in \Sigma, \tag{3}$$

$$-\left((\gamma - V)s_2 + \dot{V}\, s_3 + l_2\right) \in \Sigma. \tag{4}$$

The notation "$\in\ \Sigma$" requires that the left-hand side of the equations be expressible as a sum-of-squares (SOS) form, ensuring positive semi-definiteness. Here, V is a polynomial in q_i representing the conditional Lyapunov function. Equation (2) ensures that V is positive definite, as l_1 is a predefined positive polynomial, set to $l_1 = 10^{-10}\boldsymbol{q}^\intercal\boldsymbol{q}$ in this study. Equation (4) mandates the decrease of the Lyapunov function (i.e., $\dot{V} < 0$) within a region where $V < \gamma$. This region defines a particular level set of V that is aimed to be an ROA of the origin. s_1, s_2 and s_3 are positive semi-definite polynomials (SOS multipliers) determined through optimization. Additionally, l_2 is a predefined positive definite polynomial, set similarly to l_1 as $l_2 = 10^{-10}\boldsymbol{q}^\intercal\boldsymbol{q}$. Equation (3) characterizes the ROA, where p is a predefined polynomial describing a shape and β determines its size. This constraint ensures that the region defined by $p < \beta$ is contained within the ROA candidate $V < \gamma$. The standard choice for p is $p = \boldsymbol{q}^\intercal\boldsymbol{q}$, describing a hypersphere. In this case, $\sqrt{\beta}$ represents its radius, also referred to as the perturbation threshold amplitude. The graphical representation of these regions is shown in Fig. 1a. The main objective is to find the largest such hypersphere, achieved by maximizing β while ensuring the constraints in Equations (2)–(4) are satisfied. This SOS algorithm is designated as SOS1 in this study.

A sum-of-squares decomposition allows for a polynomial of degree up to $2d$ to be represented as $\boldsymbol{m}^\intercal(\boldsymbol{q})\mathbf{M}\boldsymbol{m}(\boldsymbol{q})$, where $\boldsymbol{m}(\boldsymbol{q})$ includes all monomials up to order d and $\mathbf{M}$ is a positive semidefinite matrix. This leads to matrix inequality constraints during optimization. While the problem could be reformulated into a semidefinite program (SDP) solvable efficiently if all constraints were linear, Equations (3) and (4) include bilinear terms in polynomial decision variables (e.g., Vs_2, βs_1). Such a problem can be solved by methods like PENBMI [11]; however, due to significant developments in SDP solvers, it is often desirable to decompose the optimization into a series of SDPs [12, 13].

Initially, the optimization process begins with the γ-step, wherein γ is considered a parameter rather than a decision variable in the SDP optimization. Additionally, the Lyapunov function (V) remains fixed throughout this step while s_2 and s_3 (positive semidefinite polynomials) are updated by the optimizer. During the γ-steps, γ is iteratively increased using a bisection algorithm until Equation (4) is feasible, signifying that the left-hand side can be represented as a sum-of-squares expression. The bisection algorithm stops if the relative change of γ is below an initial threshold level (e.g.,

10^{-4}). The aim of the γ-step is to determine the largest γ for a given Lyapunov function V such that $\dot{V} < 0$ if $V < \gamma$. Before the first γ-step, the Lyapunov function must be initialized usually based on the linearized system.

Following the γ optimization, the process transitions to the β-step. Here, β is treated as a parameter, and a bisection algorithm is employed to maximize it. The aim of this step is to find the largest hypersphere (defined by $p = \boldsymbol{q}^\top \boldsymbol{q} < \beta$) inside the ROA candidate defined by $V < \gamma$. Initially, the stopping criterion for the bisection algorithm is that the relative change of β is smaller than 10^{-4}. This optimization continues until Equation (3) is satisfied. In the β-step, the Lyapunov function V and γ are kept static, while β and s_1 are updated.

The final phase involves the V-step, where Equations (2) to (4) are subjected to SDP conversion. Unlike previous steps, here, the decision variables previously determined ($\beta, \gamma, s_1, s_2, s_3$) are maintained constant, except for the Lyapunov function V (or its coefficients if V is parameterized, e.g., as a quadratic form), which is now allowed to vary.

This iterative sequence of γ-step, β-step, and V-step is repeated until $\sqrt{\beta}$ – the threshold amplitude – reaches its maximal value, and the relative change of $\sqrt{\beta}$ between iterations falls below a predefined tolerance level (e.g., 10^{-3} in this study). Furthermore, the relative tolerance for γ and β during their respective steps may be reduced by a factor (e.g., 1.4) after each full iteration.

In this study, the dynamical system functions (f_i) and the Lyapunov function V are restricted to be second-order polynomials. Consequently, the SOS multiplier polynomials s_1, s_2, s_3 are also defined to be second-order polynomials. If higher-order f_i functions or Lyapunov functions were used, the order of these multiplier polynomials would need to be set appropriately (typically such that the products like $\dot{V} s_3$ match the required overall polynomial degree for the SOS condition). It is noted that if only the order of s_1, s_2, s_3 polynomials is increased without increasing the order of V, the outcome does not improve.

2.2 Second sum-of-squares program - SOS2

An alternative methodology was proposed by Anderson and Papachristodoulou [17] for investigating local stability. This method, focusing on quadratic Lyapunov functions, was further developed by Liu et al. [16], leading to the following SOS program:

$$V - l_1 \in \Sigma, \tag{5}$$

$$-\left(\dot{V} + (\delta^2 - \boldsymbol{q}^\top \boldsymbol{q}) s_1 + l_2\right) \in \Sigma. \tag{6}$$

In this study, we refer to this approach as SOS2. The first equation (5) ensures the positive definiteness of the Lyapunov function V, akin to the requirement in Equation (2). The terms l_1 and l_2 are the same predefined positive definite polynomials as in SOS1, and s_1 is a positive semidefinite polynomial (SOS multiplier) determined by the optimization. The second equation (6) ensures that the derivative of V, $\dot{V}$, remains negative within a hypersphere defined by $\boldsymbol{q}^\top \boldsymbol{q} < \delta^2$. However, this region $\boldsymbol{q}^\top \boldsymbol{q} < \delta^2$ where $\dot{V} < 0$ is not necessarily the provable ROA itself. The ROA is a level set $V < \gamma$ that must be contained within this region. For quadratic Lyapunov functions, $V = \boldsymbol{q}^\top \mathbf{P} \boldsymbol{q}$ (where $\mathbf{P}$ is a positive definite matrix, as required by Eq. (5)), the largest such level set $V < \gamma$ that is guaranteed to be within $\boldsymbol{q}^\top \boldsymbol{q} < \delta^2$ (where $\dot{V} < 0$) can be determined. The value of γ is found by identifying the largest γ such that the ellipsoid $\boldsymbol{q}^\top \mathbf{P} \boldsymbol{q} < \gamma$ is contained in the sphere $\boldsymbol{q}^\top \boldsymbol{q} < \delta^2$. This is given by $\gamma = \delta^2 \lambda_{\min}(\mathbf{P})$, where $\lambda_{\min}(\mathbf{P})$ is the smallest eigenvalue of $\mathbf{P}$. This subset, where $V < \gamma$ (and thus $\dot{V} < 0$), forms a superellipsoidal region in the state space, which is the provable ROA. In practice, this ROA is often characterized by the radius $\sqrt{\beta}$ of the largest inscribed hypersphere $\boldsymbol{q}^\top \boldsymbol{q} < \beta$. This can be computed as:

$$\sqrt{\beta} = \frac{\sqrt{\gamma}}{\sqrt{\lambda_{\max}(\mathbf{P})}} = \delta \sqrt{\frac{\lambda_{\min}(\mathbf{P})}{\lambda_{\max}(\mathbf{P})}}. \tag{7}$$

The different regions are illustrated in Fig. 1b. The orange dashed curve represents the boundary of the region $\boldsymbol{q}^\top \boldsymbol{q} < \delta^2$ where the Lyapunov function's derivative is negative ($\dot{V} < 0$). The blue curve represents the provable ROA (the level set $V < \gamma$). The yellow curve represents the largest hypersphere $\boldsymbol{q}^\top \boldsymbol{q} < \beta$ inside this ROA.

A key distinction between SOS1 and SOS2 (as formulated here) is that SOS2 inherently uses quadratic Lyapunov functions (for the eigenvalue-based γ calculation), resulting in a superellipsoidal ROA. SOS1 could, in principle, use higher-order Lyapunov functions for more complex ROA shapes, but this study confines SOS1 to quadratic polynomials as well for a fair comparison of the underlying SOS program structures for quadratic V.

The objective in SOS2 is also to maximize $\sqrt{\beta}$. Since β depends on δ and $\mathbf{P}$, the optimization typically aims to maximize δ subject to the SOS constraints (5)–(6) being feasible. Similar to SOS1, Equation (6) introduces a non-linear constraint due to the $\delta^2 s_1$ term (bilinearity in δ^2 and the coefficients of s_1). To address this, δ^2 is treated as a parameter that is iteratively maximized using a bisection algorithm until Equation (6) (and (5)) is feasible for V (i.e., $\mathbf{P}$) and s_1. In this study, s_1 is assumed to be of second order, consistent with the quadratic V and second-order f_i. Unlike SOS1, which involves multiple nested iteration loops and step types, the SOS2 program as described here typically requires a simpler iterative search for δ (and optimization for V and s_1 at each δ). This eigenvalue-based method for determining γ simplifies

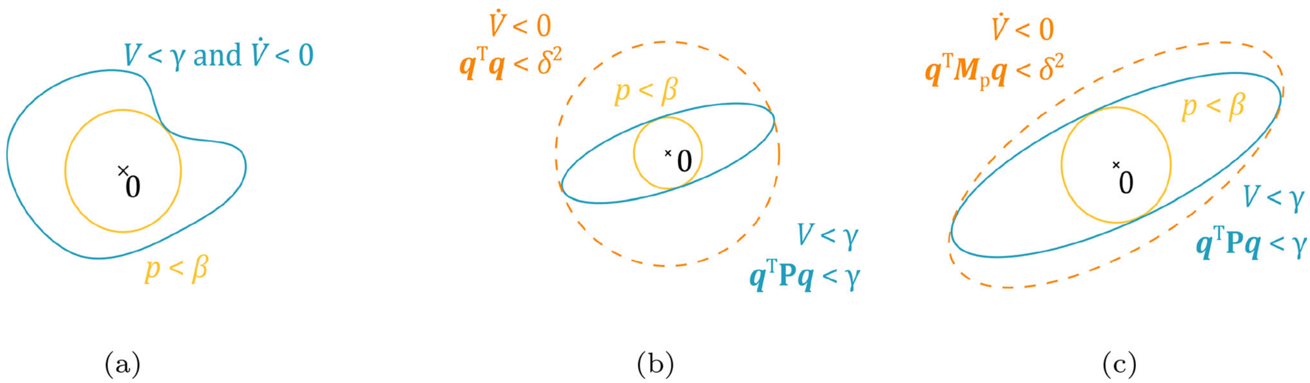

Fig. 1 Graphical representation of the state space regions for the three SOS methods. The blue curve represents the boundary of the ROA of the equilibrium point (origin). The dashed orange curve indicates the boundary of the region where the Lyapunov function derivative $\dot{V}$ is negative according to the SOS2 and SOS2m methods. The yellow circle (or ellipse in the general case, depicted as a circle here for simplicity) denotes the largest inscribed hypersphere $p < \beta$ within the ROA. (a) SOS1 method, (b) SOS2 method, (c) SOS2m method

the optimization process and can enhance robustness. However, SOS2 is specifically tailored for quadratic Lyapunov functions and, due to its structural simplification (one SOS condition for $\dot{V}$ related to a fixed shape $\boldsymbol{q}^\top\boldsymbol{q} < \delta^2$), theoretically might predict a smaller provable ROA compared to the more flexible SOS1 framework.

2.3 The improved second sum-of-squares program - SOS2m

The main drawback of the SOS2 algorithm is that the region where $\dot{V}$ is negative (i.e., $\boldsymbol{q}^\top\boldsymbol{q} < \delta^2$) is assumed to be a hypersphere, while the ROA itself ($V = \boldsymbol{q}^\top\mathbf{P}\boldsymbol{q} < \gamma$) is a hyperellipsoid for quadratic Lyapunov functions. We hypothesize that if the shape of the region where $\dot{V}$ is constrained to be negative aligns more closely with the actual shape of the Lyapunov level sets (the ROA), the provable ROA could be enlarged. This concept is visualized in Fig. 1c, where the region for $\dot{V} < 0$ is defined by $\boldsymbol{q}^\top\mathbf{M}_\mathrm{P}\boldsymbol{q} < \delta^2$ instead of $\boldsymbol{q}^\top\boldsymbol{q} < \delta^2$ (as in Fig. 1b). Here, $\mathbf{M}_\mathrm{P}$ is a positive definite matrix chosen to be similar to the Lyapunov matrix $\mathbf{P}$, potentially leading to a larger provable ROA. Achieving this alignment involves modifying Equation (6) while retaining Equation (5):

$$V - l_1 \in \Sigma, \tag{8}$$

$$-\left(\dot{V} + (\delta^2 - \boldsymbol{q}^\top\mathbf{M}_\mathrm{P}\boldsymbol{q})s_1 + l_2\right) \in \Sigma. \tag{9}$$

This modified approach is termed SOS2m. In Equation (9), the matrix $\mathbf{M}_\mathrm{P}$ is ideally chosen to be similar to the matrix $\mathbf{P}$ that defines the Lyapunov function $V = \boldsymbol{q}^\top\mathbf{P}\boldsymbol{q}$. Directly setting $\mathbf{M}_\mathrm{P} = \mathbf{P}$ would introduce a more complex bilinear (or rather, polynomial matrix inequality) optimization problem that cannot be converted directly to a SDP. To circumvent this issue, a straightforward iterative algorithm is proposed to progressively refine $\mathbf{M}_\mathrm{P}$:

1. Initialize $\mathbf{M}_\mathrm{P} = \mathbf{I}$ (identity matrix), making the first iteration of SOS2m equivalent to the SOS2 program.
2. Solve the SOS program defined by (8) and (9) by maximizing δ (using bisection) to find the optimal $V = \boldsymbol{q}^\top\mathbf{P}_\mathrm{new}\boldsymbol{q}$ and s_1.
3. Update $\mathbf{M}_\mathrm{P} \leftarrow \mathbf{P}_\mathrm{new}$.
4. Repeat steps 2 and 3. The iterations aim for the convergence of the resulting ROA size (e.g., $\sqrt{\beta}$ calculated via Eq. (12)) towards its maximum value or until it stabilizes.

The iteration process is terminated when the relative change in $\sqrt{\beta}$ (calculated using the generalized formulation below) falls below a threshold (e.g., 10^{-3}).

The inclusion of $\mathbf{M}_\mathrm{P}$ in Equation (9) necessitates a modified approach to calculating γ, which defines the ROA level set $V < \gamma$. Now, γ is expressed as:

$$\gamma = \delta^2\,\lambda_{\min}(\mathbf{P}, \mathbf{M}_\mathrm{P}) \tag{10}$$

with $\lambda_{\min}(\mathbf{P}, \mathbf{M}_\mathrm{P})$ representing the smallest eigenvalue of the generalized eigenvalue problem:

$$\mathbf{P}\,\boldsymbol{q} = \lambda\,\mathbf{M}_\mathrm{P}\,\boldsymbol{q}. \tag{11}$$

This modification influences the computation of the threshold amplitude $\sqrt{\beta}$ (the radius of the largest inscribed hypersphere in the ROA $V < \gamma$) as follows:

$$\sqrt{\beta} = \sqrt{\frac{\gamma}{\lambda_{\max}(\mathbf{P})}} = \delta\sqrt{\frac{\lambda_{\min}(\mathbf{P}, \mathbf{M}_\mathrm{P})}{\lambda_{\max}(\mathbf{P})}}. \tag{12}$$

In scenarios where $\mathbf{M}_\mathrm{P}$ is the identity matrix (i.e., the first iteration, or if SOS2 is used), this calculation aligns with Equation (7). However, as $\mathbf{M}_\mathrm{P}$ iteratively approximates $\mathbf{P}$, $\lambda_{\min}(\mathbf{P}, \mathbf{M}_\mathrm{P})$ is expected to increase (approaching 1 if $\mathbf{M}_\mathrm{P} \approx \mathbf{P}$ and $\mathbf{P}$ is scaled appropriately), thereby potentially expanding the provable ROA. This enhancement is visualized by comparing Fig. 1bwith Fig. 1c, illustrating the potentially expanded bounds of the ROA achieved through the iterative modification of $\mathbf{M}_\mathrm{P}$.

To ensure reproducibility, the source code for all three algorithms (SOS1, SOS2, and SOS2m) has been released and is available at https://github.com/pnagyhds/SOS_conditional_Lyapunov.

3 Comparison of the methods

3.1 Simple model of laminar-turbulent transition

First, the three SOS algorithms are compared using a low-order model of transition proposed by Waleffe [24]. To adapt the model for analysis around the origin of the state space, and because the laminar equilibrium point in the original model was non-zero, the last state variable was shifted as $q_4 = m - 1$ (assuming m was the original variable), following the adjustments made by Kalur et al. [15] and Henningson [25]. Consequently, the modified dynamical system is expressed as:

$$\frac{\mathrm{d}\boldsymbol{q}}{\mathrm{d}t} = \frac{1}{Re}\begin{bmatrix} -\lambda_w & Re & 0 & 0 \\ 0 & -\mu_w & 0 & 0 \\ 0 & 0 & -\nu_w & 0 \\ 0 & 0 & 0 & -\sigma_w \end{bmatrix}\boldsymbol{q} + \begin{bmatrix} -\gamma_w q_3^2 + q_2 q_4 \\ \delta_w q_3^2 \\ \gamma_w q_1 q_3 - \delta_w q_2 q_3 \\ -q_2 q_4 \end{bmatrix}. \tag{13}$$

The parameters $\lambda_w, \mu_w, \nu_w, \sigma_w$ represent the decay rates due to viscosity, while γ_w, δ_w describe the non-linear interaction between rolls (q_2) and streaks (q_1), as detailed in Waleffe [24]. The system parameters are set as $\lambda_w = \mu_w = \sigma_w = 10$, $\nu_w = 15$, $\delta_w = 1$, and $\gamma_w = 0.1$ or $\gamma_w = 0.5$, as in [15, 24]. A further parameter in the model is the Reynolds number, Re, which represents the ratio of inertial to viscous terms in the Navier-Stokes equation. For small Reynolds numbers, dissipative terms dominate, and the origin is a global attractor. As the Reynolds number increases, the behavior of the system significantly changes and becomes complex. The non-normality of the system matrix, associated with a large condition number of the corresponding eigenvector matrix, increases as the Reynolds number increases.

A fundamental characteristic of fluid dynamics systems, including this simplified transition model, is the energy conservation of the non-linear terms, in accordance with the Reynolds-Orr identity [7]. This model exhibits such a characteristic, enabling the assessment of energy stability through the eigenvalues of the symmetric part of the system matrix. When all eigenvalues of the symmetric part of the system matrix are negative, the system is deemed globally stable since the kinetic energy function serves as an unconditional Lyapunov function. The energy stability limit Re_E is analytically derived in [24] as:

$$Re_E = 2\sqrt{\lambda_w \, \mu_w}, \tag{14}$$

resulting in $Re_E = 20$ for the specified parameter sets. Below this value, there is no necessity for calculating an alternative Lyapunov function to kinetic energy for global stability. Beyond this threshold, global stability may either be established with Lyapunov functions of order higher than quadratic or, as pursued in this study, the ROA can be determined using quadratic Lyapunov functions. The upper limit above which the ROA calculation becomes impractical (or yields a vanishing ROA) is when the system becomes linearly unstable, typically denoted by $Re > Re_L$. In the case of a linearly unstable system, the ROA (around the stable origin, if it were) vanishes. The calculation of the ROA is meaningful within these two limits, $Re_E < Re < Re_L$. However, the linear component of (13) always has negative eigenvalues, so it is linearly stable for all Reynolds numbers. The investigation is carried out for $30 \leq Re \leq 100$ with a step size of 10, similarly to [15].

All SOS programs were implemented in Matlab, utilizing the SOSTOOLS library [26] to convert the problems into SDPs, which were solved by a state-of-the-art SDP solver, MOSEK 10.1. The calculations were performed on a computer equipped with an Intel i5-4590S processor running at 3.0 GHz and 8 GB of RAM at 1600 MHz.

The radius of the ROA, $\sqrt{\beta}$, calculated by the algorithms is depicted in Figs. 2a and 2b. Typically, SOS2 predicts a smaller threshold amplitude than SOS1, except at $Re = 30$ and $\gamma_w = 0.1$, where it unexpectedly surpasses SOS1. This anomaly suggests a potential sensitivity of the SOS1 method to specific parameters or its iterative optimization procedure. The findings of Kalur et al. [15], who applied the SOS1 algorithm to the same problem, are also illustrated in Fig. 2a (labeled "SOS1-K"), closely aligning with our SOS1 results, including the behavior in the low Reynolds number region. This consistency suggests that the discrepancy is likely inherent to the algorithmic approach rather than a specific implementation error.

The newly developed SOS2m method demonstrates superior performance, particularly for $\gamma_w = 0.1$, where it matches or exceeds the SOS1 method and significantly outperforms SOS2. At $\gamma_w = 0.5$, while still surpassing SOS2, SOS2m slightly lags behind SOS1, with the gap widening as the Reynolds number increases.

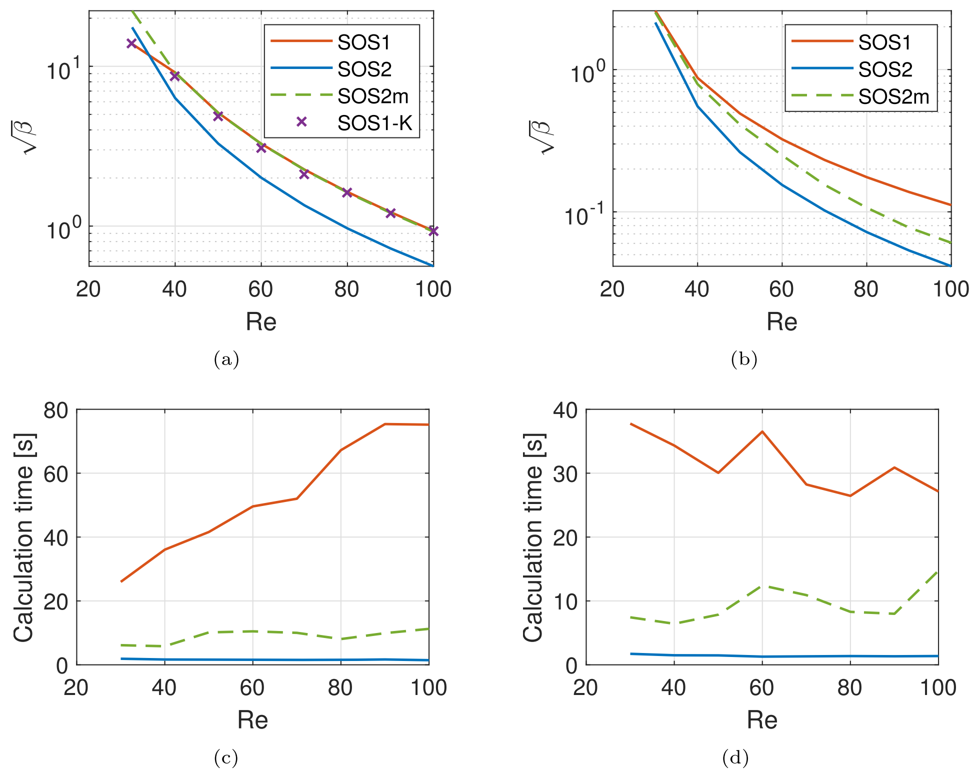

Fig. 2 Threshold amplitude $\sqrt{\beta}$ as a function of Reynolds number for different SOS algorithms: (a) $\gamma_w = 0.1$, (b) $\gamma_w = 0.5$. Computational time as a function of Reynolds number: (c) $\gamma_w = 0.1$, (d) $\gamma_w = 0.5$. The curve labeled "SOS1-K" corresponds to solutions obtained using the SOS1 algorithm, as calculated by Kalur et al. [15]

In terms of computational efficiency (Figs. 2c and 2d), SOS2 emerges as the fastest, displaying minimal sensitivity to parameter variations. This speed is attributed to the algorithm's simpler structure, which typically requires a single main optimization loop for δ (with V and s_1 optimized within), in contrast to the more complex iterative nature of the SOS1 (multiple nested loops for γ, β, V) and SOS2m (iterative updates for $\mathbf{M}_{\mathrm{P}}$) methods. These latter algorithms exhibit higher computational times that vary with system parameters, largely dependent on the number of iterations required for convergence. Specifically, for the SOS1 algorithm with $\gamma_w = 0.1$, computational time escalates with increasing Reynolds numbers, whereas for $\gamma_w = 0.5$, it decreases, averaging about half the time required for $\gamma_w = 0.1$. The SOS2m method, generally faster than SOS1, does not show a discernible trend relative to Reynolds number in these tests and, like SOS2, its computation time remains relatively stable across different γ_w values, though higher than SOS2 due to its iterative nature.

3.2 Comparison of SOS algorithms for finite-dimensional models of Couette and Poiseuille flow

In order to compare the algorithms (SOS1, SOS2, SOS2m) for dynamical systems with higher degrees of freedom, reduced-order models of two-dimensional Couette and Poiseuille flows are constructed using the Galerkin projection method. It is important to clarify that the models investigated do not describe the full dynamics of these flows but are designed to retain some key properties of shear flows. Specifically, these simple dynamical models are subcritical, their non-linear terms conserve kinetic energy (by construction of the Galerkin projection from the Navier-Stokes equations), and the matrix describing the linear part of the dynamics becomes increasingly non-normal as the Reynolds number increases. The construction method is detailed in the Appendix A.

To generate systems of increasing size, the number of streamwise modes N_x is set to 1 (i.e., fundamental mode $j_m = \pm 1$ and mean $j_m = 0$), while the number of wall-normal modes N_y is incremented from 4 upwards. This results in real-valued systems with degrees of freedom $n = (1 + 2N_x) \times N_y$, leading to the series $n = 12, 15, 18, 21, 24$, and so forth.

First, the threshold amplitude levels $\sqrt{\beta}$ are compared among the different models and are illustrated in Figs. 3a and 3b. Since the main aim is comparison, the threshold amplitudes are normalized by the result of the best-performing algorithm for each case, making the results more readable. For reproducibility, the threshold kinetic energy (proportional to $\beta = \sum q_i^2$ at the edge of the ROA hypersphere), normalized by the kinetic energy of the base flow, is also presented in Figs. 3e and 3f.

For these models, the new SOS2m algorithm generally outperforms the others in terms of the size of the provable

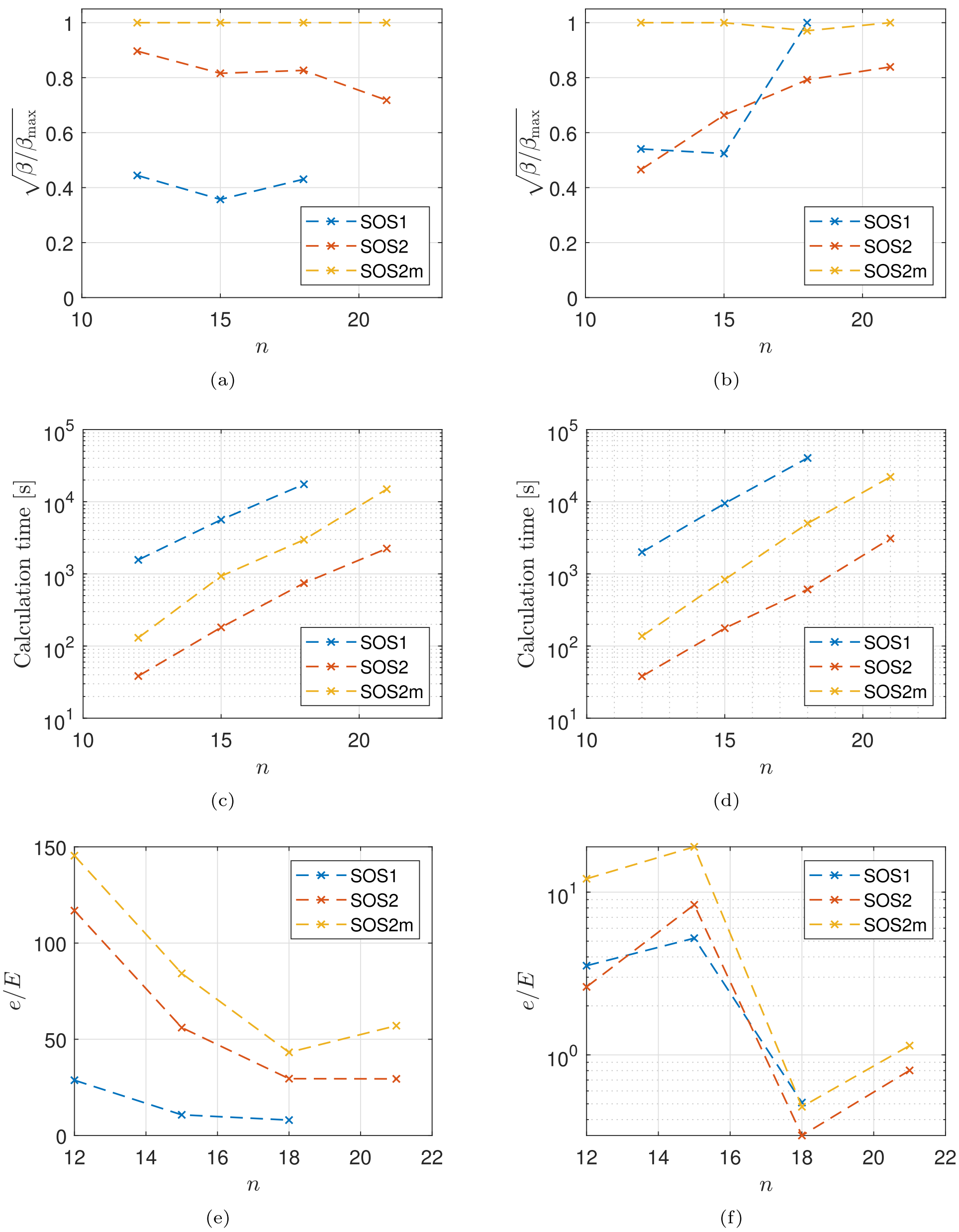

Fig. 3 Comparison of SOS algorithms for finite-dimensional models of Poiseuille and Couette flows. (a,b) Relative threshold amplitude $\sqrt{\beta}/\max(\sqrt{\beta})$ as a function of the number of degrees of freedom n. (c,d) Computational time (in seconds). (e,f) Threshold kinetic energy $e = \frac{1}{2}\sum q_i^2$ (normalized by base flow kinetic energy E) corresponding to β (a,c,e) Models based on Poiseuille flow. (b,d,f) Models based on Couette flow. All results at $Re = 200$

ROA, except for one case at $n = 18$ for the Couette flow. However, as the number of degrees of freedom increased from 15 to 18 for Couette flow, a significant reduction in the threshold amplitude is observed in Fig. 3f for all methods, which is attributed to a substantial change in the behavior of the underlying dynamical system itself as more modes are included, rather than a failing of one particular algorithm.

When comparing the original SOS1 and SOS2 algorithms, the SOS2 algorithm often demonstrates comparable or even better performance for Poiseuille flow models as n increases, while for Couette flow models, the relative performance varies. This result can be surprising because, theoretically, the SOS1 algorithm, with its more flexible formulation, should predict larger (or equal) ROAs. The observed discrepancy where SOS2 might outperform or be more robust for larger n might arise because the SDP problem associated with the SOS1 algorithm involves more decision variables and more complex bilinear constraints that are handled iteratively. This increases the complexity of each SDP step and the overall iteration, potentially leading to premature termination, local optima in the iterative search, or numerical difficulties (infeasibility reported by the solver not due to true mathematical infeasibility but due to solver tolerances or problem scaling) for larger systems. This indicates that advancements in SDP solvers or more sophisticated techniques for handling the bilinearities in SOS1 might enable it to consistently predict larger ROAs in the future. However, it was also noted that in one case involving the four-dimensional Waleffe model

(Section 3.1), the SOS1 algorithm predicted a smaller ROA than SOS2m and even SOS2 at $Re = 30$, $\gamma_w = 0.1$.

Additionally, for $n > 18$ in the case of SOS1, and $n > 21$ for SOS2 and SOS2m algorithms, the solver failed to find feasible solutions even for very small parameters (γ, δ, or β). This issue is related to the rapid growth in the number of decision variables (coefficients of V and s_k polynomials) and constraints in the SDP. For quadratic V and quadratic s_k with n variables, the number of monomial terms in the resulting SOS conditions (which are quartic in q_i) scales roughly as n^4, and the size of the Gram matrices in the SDP formulation scales accordingly. The SDP might become so large that even state-of-the-art solvers like MOSEK cannot handle it efficiently or run into memory/time limitations. Since SOS2 and SOS2m involve fewer SOS constraints than SOS1, they appear applicable to slightly larger dynamical systems in practice.

Next, the necessary computational time for these models was evaluated (Figs. 3c and 3d). For this comparison, a computer with an Intel Core i7-7700 processor at 3.6 GHz and 32 GB of RAM at 2400 MHz was used. The trends in computational time are similar to those observed in the four-dimensional Waleffe model. The SOS2 algorithm is the fastest. SOS2m is slower due to its additional iterative loop for $\mathbf{M}_{\mathrm{P}}$. SOS1 is generally the slowest due to its three nested iterative loops for γ, β, and V. The computational time scales roughly as the seventh power of the number of degrees of freedom ($O(n^7)$) in all cases. This scaling behavior is consistent with theoretical expectations for interior-point methods applied to SDPs arising from SOS problems.

For a fourth-order polynomial, appearing in the SOS constraints in the case of quadratic V, with n distinct variables, the number of monomials is proportional to $D \propto n^2$. In solving a semidefinite program (SDP) using the primal–dual interior point method [5, 27], the number of computations per iteration scales as $\mathcal{O}(D^3)$. Additionally, the number of iterations required scales as $\mathcal{O}(\sqrt{D})$. Combining these factors, the total computational time scales approximately as $\mathcal{O}(D^{3.5})$. Since $D \propto n^2$, the computational time scales as $\mathcal{O}(n^7)$, indicating that as the number of degrees of freedom increases, the computational effort required grows significantly. The results are presented for $Re = 200$ in all cases, but similar conclusions regarding relative performance and scaling apply at other Reynolds numbers investigated.

4 Conclusion

The computation of regions of attraction for dynamical systems using SOS methods has been the focus of this study, with comparisons made between two established algorithms (SOS1 and SOS2) and a newly proposed modification (SOS2m). The analysis covered finite-dimensional dynamical models with up to $n = 21$ degrees of freedom, designed to mimic key characteristics of shear flows.

This study serves as a crucial intermediate step toward understanding and predicting subcritical transition in shear flows by exploring state-of-the-art methods for constructing conditional Lyapunov functions. The proposed algorithms are applied to finite-dimensional dynamical models that exhibit key characteristics of shear flows, such as energy conservation in the nonlinear components (for inviscid dynamics, or specific structure for viscous Galerkin models) and increasing non-normality of the linearized dynamics at higher Reynolds numbers. Additionally, the findings may have broader applicability to other dynamical systems that can be described using polynomial representations.

The comparison of the three algorithms revealed that the widely used SOS1 algorithm, often expected to predict the largest ROA, can underperform or face numerical challenges when the number of degrees of freedom exceeds approximately 10. Specifically, while SOS1 predicts a ROA comparable to the SOS2 algorithm for Couette flow-based dynamical systems, it often estimates a significantly smaller ROA for Poiseuille flow-based models in our tests, particularly as n grows. Furthermore, SOS2 generally requires an order of magnitude less computational time. This unexpected finding regarding ROA size for SOS1 can be attributed to SOS2's simpler structure with fewer constraints, which reduces the complexity of the SDP problem, potentially making it easier to solve and less prone to getting stuck in poor local optima of the iterative search procedure that SOS1 employs.

Based on these results, the SOS1 algorithm might be preferred for very low-dimensional systems where computational time is not a primary limitation and the order of the Lyapunov function polynomial can be increased (as SOS1 is more readily adaptable to higher-order V, while SOS2 and SOS2m, as formulated with eigenvalue calculations for γ, are intrinsically tied to quadratic V). For higher-dimensional systems (e.g., $n > 10$), the SOS2 algorithm offers a practical and more robust alternative due to its reduced computational demand and simpler implementation.

The SOS2 algorithm can be further enhanced with the proposed new variant, SOS2m, introduced in this paper (Section 2.3). Although this modification requires additional computational time compared to SOS2 (due to the iterative refinement of $\mathbf{M}_{\mathrm{P}}$), it generally predicts a ROA comparable to or larger than SOS1 for small systems and consistently larger ROAs than SOS2 for larger systems, while still being significantly less computationally intensive than SOS1. Moreover, implementing the SOS2 and SOS2m algorithms is arguably simpler compared to the multi-loop structure of SOS1.

In summary, for investigating the conditional stability of shear flows (and other polynomial systems) with moderately higher degrees of freedom using quadratic Lyapunov functions, the SOS2 and particularly the new SOS2m algorithms appear preferable. If accuracy in estimating the ROA is a priority and computational cost is a concern, the SOS2m method is recommended as it often provides the best balance. Otherwise, the original SOS2 algorithm provides a faster and efficient alternative for estimating the ROA when its inherent approximations are acceptable.

Appendix A Construction of finite-dimensional models of Couette and Poiseuille flow

Here, the Stokes modes will form the basis for the reduced-order models. The Stokes equations are similar to the Navier-Stokes equations for an incompressible fluid but with the non-linear convection term neglected. The Stokes equations in non-dimensional form are given by:

$$\frac{\partial u_i}{\partial t} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j \partial x_j} \tag{A1}$$

and the incompressibility condition:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{A2}$$

where u_i represents the non-dimensional velocity components, p is the non-dimensional pressure, and x_i are the spatial coordinates, with x_1 as the streamwise and x_2 as the wall-normal coordinate. In both flow configurations, the geometry is a rectangle where $x_1 \in [0, L_x]$ and $x_2 \in [-1, 1]$. In this case, Stokes eigenfunctions can be calculated by assuming a temporal ansatz of the form:

$$u_i(\boldsymbol{x}, t) = \hat{u}_i(\boldsymbol{x})\mathrm{e}^{\lambda t} \tag{A3}$$

and solving the eigenvalue problem:

$$\lambda \hat{u}_i = -\frac{\partial \hat{p}}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 \hat{u}_i}{\partial x_j \partial x_j}, \tag{A4}$$

subject to:

$$\frac{\partial \hat{u}_i}{\partial x_i} = 0 \tag{A5}$$

for λ. The eigenvalues λ are negative real numbers representing the dissipation rate of the corresponding mode. Furthermore, the eigenvectors (Stokes modes) are orthogonal, which is advantageous for the Galerkin projection.

Given the linearity of the eigenvalue problem and assuming periodicity in the x_1 direction, the problem can be decomposed using a complex Fourier series in space. The $\hat{u}_i$ velocity field can be expressed as a sum of modes of the form:

$$\tilde{u}_{i,j_m}(x_2)\mathrm{e}^{\mathrm{i} j_m \alpha_0 x_1}, \tag{A6}$$

where $\alpha_0 = 2\pi/L_x$ is the fundamental wavenumber associated with the periodicity length L_x in the x_1-direction, and j_m represents the streamwise mode index. While theoretically j_m ranges from $-\infty$ to ∞, in practice, it is truncated to finite limits $-N_x \leq j_m \leq N_x$ for numerical computations. Substituting the complex wave form (A6) into Equations (A4) and (A5) leads to the following one-dimensional eigenvalue problem for each j_m mode (parameterized by $\alpha = j_m \alpha_0$):

$$\begin{bmatrix} L_d & 0 & -\mathrm{i}\alpha \\ 0 & L_d & -D_{x_2} \\ \mathrm{i}\alpha & D_{x_2} & 0 \end{bmatrix} \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{p} \end{bmatrix} = \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \\ \tilde{p} \end{bmatrix} \tag{A7}$$

where $L_d = (D_{x_2}^2 - \alpha^2)$ is the Laplace operator and D_{x_2} is the differentiation operator with respect to x_2. The problem (A7) can be discretized using the Chebyshev collocation method. The required boundary conditions involve stationary walls at $x_2 = \pm 1$, implying $\tilde{u}_i(\pm 1) = 0$ for all velocity components i. These conditions are enforced by modifying the discretized matrix system by removing rows and columns. In this study, 100 Chebyshev collocation points are employed, a choice deemed sufficiently accurate based on prior research [28]. For each streamwise wavenumber $j_m\alpha_0$, the first N_y wall-normal modes (eigenfunctions $\tilde{u}_{i,j_m,k}(x_2)$ with corresponding eigenvalues $\lambda_{j_m,k}$) with the largest eigenvalues (least negative, i.e., least dissipative) are considered as a basis for the Galerkin projection. This results in a total of $n \approx (2N_x + 1)N_y$ modes.

The actual velocity field is approximated as a linear combination of these selected Stokes modes $\hat{\boldsymbol{u}}_{k_m}(\boldsymbol{x})$:

$$\boldsymbol{u}(\boldsymbol{x}, t) = \sum_{k_m=1}^{n} q_{k_m}(t)\hat{\boldsymbol{u}}_{k_m}(\boldsymbol{x}), \tag{A8}$$

where k_m is a global index for the modes, and $q_{k_m}(t)$ are the time-dependent complex amplitudes (modal coefficients) that describe the state of the resulting dynamical system.

The ordinary differential equations for q_{k_m} are obtained by applying the Galerkin projection method to the perturbed Navier-Stokes equation:

$$\frac{\partial u_i}{\partial t} + U_j\frac{\partial u_i}{\partial x_j} + u_j\frac{\partial U_i}{\partial x_j} + u_j\frac{\partial u_i}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j \partial x_j}. \tag{A9}$$

where U_i represents the i-th velocity component of the laminar base flow and u_i is the perturbation velocity field. The

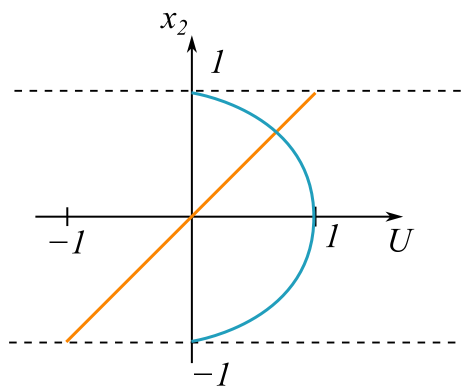

Fig. 4 The streamwise velocity component $U_1(x_2)$ of the base flows. Orange: Couette flow, blue: Poiseuille flow

Galerkin projection yields a right-hand side for the evolution of $q_{i'}$ (using i' as generic index for $f_{i'}$) in the form:

$$f_{i'} = A_{i'j'}q_{j'} + Q_{i'j'k'}q_{j'}q_{k'}, \tag{A10}$$

where the coefficients $A_{i'j'}$ (linear terms) and $Q_{i'j'k'}$ (quadratic terms) are computed from inner products involving the base flow U_i and the basis modes $\hat{u}_{i,k_m}$. Specifically:

$$A_{i'j'} = \left\langle -U_k \frac{\partial \hat{u}_l^{(j')}}{\partial x_k} - \hat{u}_k^{(j')} \frac{\partial U_l}{\partial x_k} + \frac{1}{Re} \frac{\partial^2 \hat{u}_l^{(j')}}{\partial x_k \partial x_k}, \hat{u}_l^{(i')} \right\rangle \tag{A11}$$

$$Q_{i'j'k'} = \left\langle -\hat{u}_k^{(j')} \frac{\partial \hat{u}_l^{(k')}}{\partial x_k}, \hat{u}_l^{(i')} \right\rangle \tag{A12}$$

where $\langle \cdot, \cdot \rangle$ denotes the appropriate inner product over the domain Ω, $\hat{u}_l^{(j')}$ is the l-th component of the j'-th basis mode, and summation over repeated spatial indices k, l is implied. In both flow configurations, only the streamwise component U_1 of the base velocity field is non-zero. For Poiseuille flow, this component is:

$$U_1(x_2) = 1 - x_2^2, \tag{A13}$$

while for Couette flow, it is:

$$U_1(x_2) = x_2. \tag{A14}$$

The base flow velocity profiles are illustrated in Fig. 4.

The modes are substituted in the form (A6), and the integrals over x_2 are evaluated numerically (e.g., using Chebyshev collocation points and associated quadrature rules), while integrals over x_1 (for periodic directions) are often evaluated analytically due to the Fourier basis. Since the Stokes eigenfunctions $\hat{u}_{i,j_m}$ for $j_m \neq 0$ are complex, the resulting $A_{i'j'}$ matrix and $Q_{i'j'k'}$ tensor are also complex. These cannot be directly handled by the SOSTOOLS implementation used, which expects real-valued polynomial systems. The issue is resolved by transforming the system of complex q_{k_m} into a real-valued one. Let i_0 represent the indices of the real-valued modes ($j_m = 0$), i_c the complex-valued modes (for $j_m > 0$), and i_{cc} their corresponding complex conjugates (for $j_m < 0$). By relating $q_{j_m<0} = q^*_{j_m>0}$ and defining real variables like $q_{\text{real},j_m} = (q_{j_m} + q^*_{j_m})/\sqrt{2}$ and $q_{\text{imag},j_m} = (q_{j_m} - q^*_{j_m})/(\mathrm{i}\sqrt{2})$, a transformation matrix $\mathbf{T}$ can be constructed to map the complex amplitudes to real ones. The specific transformation matrix is:

$$\mathbf{T} = \begin{bmatrix} \mathbf{I}_{N_0\times N_0} & \mathbf{0}_{N_0\times N_c} & \mathbf{0}_{N_0\times N_c} \\ \mathbf{0}_{N_c\times N_0} & \frac{1}{\sqrt{2}}\mathbf{I}_{N_c\times N_c} & \frac{1}{\sqrt{2}}\mathbf{I}_{N_c\times N_c} \\ \mathbf{0}_{N_c\times N_0} & \frac{1}{\mathrm{i}\sqrt{2}}\mathbf{I}_{N_c\times N_c} & -\frac{1}{\mathrm{i}\sqrt{2}}\mathbf{I}_{N_c\times N_c} \end{bmatrix} \tag{A15}$$

where N_0 is the number of real-valued modes (e.g., $j_m = 0$ modes) and N_c is the number of unique complex-valued modes (e.g., for $j_m > 0$, so the total number of complex modes is $2N_c$). Let $\boldsymbol{q}_{\text{complex}}$ be the vector of original complex modal coefficients and $\boldsymbol{q}_{\text{real}} = \mathbf{T}\boldsymbol{q}_{\text{complex}}$ (or $\boldsymbol{q}_{\text{complex}} = \mathbf{T}^{-1}\boldsymbol{q}_{\text{real}}$). Applying this transformation to the coefficient matrix and tensor results in real-valued $\tilde{A}_{ij}$ and $\tilde{Q}_{ijk}$ for the dynamics of $\boldsymbol{q}_{\text{real}}$:

$$\tilde{A} = \mathbf{T}\mathbf{A}\mathbf{T}^{-1} \tag{A16}$$

$$\tilde{Q}_{ijk} = T_{im}Q_{mol}\,(T^{-1})_{oj}\,(T^{-1})_{lk}. \tag{A17}$$

It is noted that this transformation could be avoided by decomposing the complex modes into two real-valued basis functions (e.g., sine and cosine streamwise dependence) before calculating the Galerkin integrals; however, this approach can complicate the evaluation of the integrals compared to using complex exponentials.

Acknowledgements This paper was supported by the János Bolyai Research Scholarship (BO/00364/24) of the Hungarian Academy of Sciences.

Author Contributions PTN conceptualized the research and wrote, edited the main manuscript.

Funding Open access funding provided by Budapest University of Technology and Economics.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Murty, K.G., Kabadi, S.N.: Some NP-complete problems in quadratic and nonlinear programming. Math. Program. **39**, 117 (1987). https://doi.org/10.1007/BF02592948
2. Nesterov, Y.: *Squared functional systems and optimization problems* (Springer,), pp. 405–440 (2000)
3. Papachristodoulou, A., Prajna, S.: On the construction of lyapunov functions using the sum of squares decomposition. Proceedings of the 41st IEEE Conference on Decision and Control **3**, 3482–3487 (2002)
4. Parrilo, P.A.: Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. Ph.D. thesis, California Institute of Technology (2000)
5. Goulart, P.J., Chernyshenko, S.: Global stability analysis of fluid flows using sum-of-squares. Physica D **241**(6), 692–704 (2012). https://doi.org/10.1016/j.physd.2011.12.008. (https://www.sciencedirect.com/science/article/pii/S0167278911003575)
6. Huang, D., Chernyshenko, S., Goulart, P., Lasagna, D., Tutty, O., Fuentes, F.: (2015) Sum-of-squares of polynomials approach to nonlinear stability of fluid flows: An example of application. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **471**(2183) . https://doi.org/10.1098/rspa.2015.0622
7. Orr, W.M.: The Stability or Instability of the Steady Motions of a Perfect Liquid and of a Viscous Liquid. Part II: A Viscous Liquid. Proc. R. Irish Acad. **27**, 69–138 (1907). http://www.jstor.org/stable/20490591
8. Fuentes, F., Goluskin, D., Chernyshenko, S.: Global stability of fluid flows despite transient growth of energy. Phys. Rev, Lett (2022). https://doi.org/10.1103/PhysRevLett.128.204502
9. Galdi, G.P., Straughan, B.: Exchange of stabilities, symmetry, and nonlinear stability. Arch. Ration. Mech. Anal. **89**(3), 211–228 (1985). https://doi.org/10.1007/BF00276872
10. Jarvis-Wloszek, Z.W.: Lyapunov Based Analysis and Controller Synthesis for Polynomial Systems using Sum-of-Squares Optimization. Ph.D. thesis, University of California (2003)
11. Tan, W., Packard, A.: Stability region analysis using polynomial and composite polynomial Lyapunov functions and sum-of-squares programming. IEEE Trans. Autom. Control **53**(2), 565–571 (2008). https://doi.org/10.1109/TAC.2007.914221
12. Topcu, U., Packard, A., Seiler, P.: Local stability analysis using simulations and sum-of-squares programming. Automatica **44**(10), 2669–2675 (2008). https://doi.org/10.1016/j.automatica.2008.03.010
13. Khodadadi, L., Samadi, B., Khaloozadeh, H.: Estimation of region of attraction for polynomial nonlinear systems: A numerical method. ISA Trans. **53**(1), 25–32 (2014). https://doi.org/10.1016/j.isatra.2013.08.005
14. Meng, F., Wang, D., Yang, P., Xie, G., Guo, F.: Application of Sum-of-Squares Method in Estimation of Region of attraction for Nonlinear Polynomial Systems. IEEE Access **8**, 14234–14243 (2020). https://doi.org/10.1109/aCCESS.2020.2966566
15. Kalur, A., Seiler, P., Hemati, M.S.: Nonlinear stability analysis of transitional flows using quadratic constraints. Physical Review Fluids **6**(4), 1–20 (2021). https://doi.org/10.1103/PhysRevFluids.6.044401. arXiv:2004.05440
16. Liu, C., Gayme, D.F.: Input-output inspired method for permissible perturbation amplitude of transitional wall-bounded shear flows. Physical Review E **102**(6), 1–16 (2020). https://doi.org/10.1103/PhysRevE.102.063108. arXiv:2006.16484
17. Anderson, J., Papachristodoulou, A.: Advances in computational lyapunov analysis using sum-of-squares programming. Discrete & Continuous Dynamical Systems-Series B **20**(8) (2015). https://doi.org/10.3934/dcdsb.2015.20.2361
18. Henrion, D., Korda, M.: Convex computation of the region of attraction of polynomial control systems. IEEE Trans. Autom. Control **59**(2), 297–312 (2014). https://doi.org/10.1109/TAC.2013.2283095
19. Toso, L.F., Drummond, R., Duncan, S.R.: Regional stability analysis of transitional fluid flows. IEEE Control Systems Letters **6**, 2287 2292 (2022). https://doi.org/10.1109/lcsys.2022.3145233
20. Nerli, A., Camarri, S., Salvetti, M.V.: A conditional stability criterion based on generalized energies. J. Fluid Mech. **581**, 277–286 (2007). https://doi.org/10.1017/S002211200700609X
21. Nagy, P.T.: Constructing quadratic Lyapunov functions for conditionally stable fluid dynamics systems. J. Fluid Mech. **1014**, A25 (2025). https://doi.org/10.1017/jfm.2025.10272
22. Habib, G.: Dynamical integrity assessment of stable equilibria: a new rapid iterative procedure. Nonlinear Dyn. **106**(3), 2073–2096 (2021). https://doi.org/10.1007/s11071-021-06936-9
23. Petridis, V., Petridis, S.: *Construction of neural network based lyapunov functions*, in *The 2006 IEEE International Joint Conference on Neural Network Proceedings* (IEEE, 2006), pp. 5059–5065
24. Waleffe, F.: Transition in shear flows. Nonlinear normality versus non-normal linearity. Physics of Fluids **7**(12), 3060–3066 (1995). https://doi.org/10.1063/1.868682
25. Henningson, D.: Comment on Transition in shear flows. Nonlinear normality versus non-normal linearity" [Phys. Fluids 7 , 3060 (1995)] . Physics of Fluids **8**(8), 2257–2258 (1996). https://doi.org/10.1063/1.869011
26. Papachristodoulou, A., Anderson, J., Valmorbida, G., Prajna, S., Seiler, P., Parrilo, P.A.: *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*. http://arxiv.org/abs/1310.4716 (2013). Available from http://www.cds.caltech.edu/sostools and http://www.mit.edu/~parrilo/sostools
27. Parrilo, P.A.: Semidefinite programming relaxations for semialgebraic problems. Math. Program. **96**, 293–320 (2003). https://doi.org/10.1007/s10107-003-0387-5
28. Nagy, P.T., Pa l, G., Kiss, M.: Imposing a constraint on the discrete Reynolds Orr equation demonstrated in shear flows. Physics of Fluids **35**(3), 034115 (2023). https://doi.org/10.1063/5.0142781. https://pubs.aip.org/aip/pof/article-pdf/doi/10.1063/5.0142781/16799795/034115_1_online.pdf