Adjoint-based optimisation of time- and span-periodic flow fields with Space–Time Spectral Method: Application to non-linear instabilities in compressible boundary layer flows

Arthur Poulain, Cédric Content, Aldo Schioppa, Pierre Nibourel, Georgios Rigas, Denis Sipp

Computers and Fluids 282 (2024) 106386

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Adjoint-based optimisation of time- and span-periodic flow fields with Space–Time Spectral Method: Application to non-linear instabilities in compressible boundary layer flows

Arthur Poulain [a], Cédric Content [a], Aldo Schioppa [a], Pierre Nibourel [a,*], Georgios Rigas [b], Denis Sipp [a]

[a] DAAA, ONERA, Institut Polytechnique de Paris, F-92190 Meudon, France
[b] Department of Aeronautics, Imperial College London, London SW7 2AZ, United Kingdom

ARTICLE INFO

Keywords:
Time Spectral Method
Algorithmic differentiation
Adjoint-based optimisation
Stability
Transition

ABSTRACT

We aim at computing time- and span-periodic flow fields in span-invariant configurations. The streamwise and cross-stream derivatives are discretised with finite volumes while time and the span-direction are handled with pseudo-spectral Fourier-collocation methods. Doing so, we extend the classical Time Spectral Method (TSM) to a Space–Time Spectral Method (S-TSM), by considering non-linear interactions of a finite number of time and span harmonics. For optimisation, we introduce an adjoint-based framework that allows efficient computation of the gradient of any cost functional with respect to a large-dimensional control parameter. Both theoretical and numerical aspects of the methodology are described: evaluation of matrix–vector products with S-TSM Jacobian (or its transpose) by algorithmic differentiation, solution of fixed-points with quasi-Newton method and de-aliasing in time and space, solution of direct and adjoint linear systems by iterative algorithms with a block-circulant preconditioner, performance assessment in CPU time and memory. We illustrate the methodology on the case of 3D instabilities (first Mack mode) triggered within a developing adiabatic boundary layer at M = 4.5. A gradient-ascent method allows to identify a finite-amplitude 3D forcing that triggers a non-linear response exhibiting the strongest time- and span-averaged drag on the flat-plate. In view of flow control, a gradient-descent method finally determines a finite amplitude 2D wall-heat flux that minimises the averaged drag of the plate in presence of the previously determined non-linear optimal forcing.

1. Introduction

The efficient computation and parametric optimisation of periodic solutions (in time and/or in a homogeneous spanwise direction) is relevant to many industrial fluid applications, such as in rotating systems (turbomachinery, propeller and fan studies, etc.), in flows undergoing bifurcations (cavity flows, bluff bodies, flutter and limit-cycle oscillations in fluid–structure interactions) or in boundary layer flows to study or control the laminar-to-turbulence transition path.

Time-periodic solutions may be obtained by time stepping the governing Navier–Stokes equations. Yet, this might be computationally expensive due to the prescribed time period being usually very large with respect to the time step of the simulation (which is severely constrained by the CFL condition). Convergence of the time stepping towards the exact periodic solution can nonetheless be accelerated with a number of techniques [1,2]. Also, in some cases, the time stepping will naturally move away from the periodic solution due to Floquet-type instabilities. In such cases, stabilising techniques may be applied to ensure convergence to the unstable periodic solution [3].

Another class of methods for calculating periodic flow solutions involves the use of spectral methods to discretise the time derivative: the use of Fourier modes to describe the time-periodic behaviour of the flow may bypass the CFL constraint and allow accurate computation of periodic solutions with a significantly reduced discretisation in time (small number of harmonics). Yet, the price to pay is that spectral methods are global in time, coupling all Fourier modes or time instants together, yielding a coupled system that is computationally challenging to solve. For this class of spectral methods, two approaches exist: the Fourier–Galerkin method, also called Harmonic Balanced Method (HBM), and the Time-Spectral Method (TSM).

* Corresponding author.
E-mail address: pierre.nibourel@onera.fr (P. Nibourel).

https://doi.org/10.1016/j.compfluid.2024.106386
Received 22 February 2024; Received in revised form 21 June 2024; Accepted 25 July 2024
Available online 27 July 2024

In the spectral Fourier–Galerkin formulation (HBM), the problem is solved in the frequency domain by expressing the solution as a truncated Fourier series. A benefit of this formulation is a direct harmonic decomposition of the solution. Rigas et al. [4] used HBM to calculate non-linear optimal disturbances in an incompressible boundary layer while Sierra-Ausin et al. [5] extended its application to compressible flows at low Mach numbers under the assumption of constant viscosity. Both studies required the computation of all non-linear interactions, which is time consuming and possible only if the non-linearity is polynomial (quadratic in incompressible Navier–Stokes equations, cubic in compressible conservative Navier–Stokes equations, the usage of the Sutherland's law being excluded therefore).

The TSM relies on solving the equations in the time domain with a pseudo-spectral Fourier collocation method, which bypasses the two above mentioned problems: since the method only relies on evaluating the residual of the governing equations at the collocations points, it is efficient and easy to implement, allows to deal with compressible conservative Navier–Stokes equations with the Sutherland's law and can be applied within a discrete framework in which the discretised residual can involve complex numerical schemes such as shock-capturing methods. Therefore, the TSM has been preferred to HBM in turbomachinery, for instance [6–8]. Yet, contrary to HBM, which allows to explicitly remove the aliasing problem associated to the non-linearity of the governing equations, the TSM requires de-aliasing techniques, which multiplies by two the number of required collocation points or Fourier modes in the case of cubic non-linearities [9].

Once formulated within TSM, the equations may be solved by explicit or implicit techniques. The former, closer to the time-stepping approaches, exhibits a large computational time cost due to a CFL constraint inversely proportional to the frequency and the number of harmonics considered. The latter is quicker, if Newton-type methods can be used; for this, the Jacobian of the TSM operator needs to be inverted, which is possible with direct LU methods only for systems with small number of degrees of freedom and small number of collocation points. If this is not the case, iterative techniques may be implemented. Sicot et al. [10] was the first to introduce a Block-Jacobi method (with a pseudo-time stepping approach) to solve for the TSM large-scale linear system. Mader and Martins [11], Kenway et al. [8] improved the method by considering a GMRES algorithm preconditioned by a Block-Jacobi solver. For low Mach number flows, Sierra-Ausin et al. [5] has compared the Block-Jacobi preconditioned GMRES algorithm for TSM and HBM (also used in Rigas et al. [4]) and showed much better convergence properties for HBM. However, HBM being formulated in frequency domain and TSM in time domain, the block-diagonal approximation involved in the Block-Jacobi method is not equivalent in HBM and TSM. It actually appears to be a much better approximation of the HBM linear operator than of the TSM linear operator, so that the two methods cannot be directly compared. Moulin [12] showed that the analog of the block-Jacobi preconditioner in HBM is a block-circulant preconditioner in TSM, which takes advantage of the block-circulant shape of the mean TSM Jacobian operator, allowing a fairer comparison of the two methods.

The objective of the present paper is first to extend the TSM formulation to a Space–Time Spectral Method (S-TSM), for which the solution is both periodic in time and in a homogeneous space direction. The starting point of the study will be a two-dimensional discretised finite-volume code; we will show how to add the spanwise and time derivatives with a spectral Fourier collocation method and how the general optimisation problem is formulated. Then, the block-circulant preconditioner [12] will be adapted to deal with both the collocation points in time and in the spanwise direction. The resulting preconditioner, that will be explicitly constructed thanks to algorithmic differentiation, is then used by a GMRES algorithm to solve the implicit Newton iterations. Following Kenway et al. [8], Rigas et al. [4], we introduce also an adjoint linear system, that will be solved again with an iterative GMRES solver based on the block-circulant conjugate transpose mean TSM operator, to perform gradient-based optimisation within a very large dimensional optimisation space. All steps will be detailed and assessed in terms of computational cost.

The method will be illustrated in the context of finite-amplitude transition in a three-dimensional compressible supersonic M = 4.5 developing boundary layer. We will optimise a finite amplitude volume force triggering a maximal mean-drag response. Then, the predictions of the S-TSM solution will be validated against a compressible direct numerical simulation (DNS), in which many more degrees of freedom can be used in the streamwise and homogeneous spanwise directions. Eventually, we will optimise a finite-amplitude wall-heat flux minimising this mean-drag in presence of the previously computed optimal finite-amplitude forcing. From a physical point of view, it can therefore be considered both as an extension of the incompressible study of Rigas et al. [4] and as the deterministic counterpart of the optimisation studies by Jahanbakhshi and Zaki [13,14] with ensemble-based methods.

The outline of the paper is as follows. The S-TSM method and gradient-based optimisation strategy are presented in Section 2, while general algorithms to solve those large-scale problems are described in Section 3. Application to the case of supersonic boundary layer flow will follow in Section 4, including the optimisation of a worst-case 3D volume force maximising the drag, the validation of the response given by the S-TSM against a DNS and the drag minimisation by a 2D wall heat-flux in presence of the instabilities triggered by the previous worst-case forcing. Computational performance is then assessed in Section 5, followed by conclusions and outlook.

2. Theory

2.1. Governing equations and formulation

We consider equations governing a state in a spanwise invariant configuration, where (x, y) are the streamwise and cross-stream coordinates, while z is the spanwise coordinate. Discretisation of the three-dimensional compressible Navier–Stokes equations in the (x, y) directions yields the following semi-discrete equations governing the state $\mathbf{q}(z, t)$:

$$\mathbf{M}\frac{\partial \mathbf{q}}{\partial t} + \mathbf{R}_{3D}(\mathbf{q}) = \mathbf{B}\mathbf{u}, \tag{1}$$

where $\mathbf{u}(z, t)$ is a vector forcing term that acts on the residual through the $\mathbf{B}$ matrix. Note that $\mathbf{u}(z, t)$ may describe either a volume source term $f(x, y, z, t)$ or a wall-boundary condition term $\phi(s, z, t)$, where s is a curvilinear abscissa along a wall. In the following, for simplicity, we will assume that the operator $\mathbf{B}$ does not involve any z-derivative. In Eq. (1), $\mathbf{M}$ designates the Hermitian mass matrix and $\mathbf{R}_{3D}(\mathbf{q})$ the discrete residual of the governing equations.

We assume that the 3D residual $\mathbf{R}_{3D}$ can be split as the sum of a 2D discretised residual $\mathbf{R}_{2D}$ and its z-derivative components $\mathbf{R}_z$ and $\mathbf{R}_{zz}$:

$$\mathbf{R}_{3D}(\mathbf{q}) = \mathbf{R}_{2D}(\mathbf{q}) + \mathbf{R}_z\left(\mathbf{q}, \frac{\partial \mathbf{q}}{\partial z}\right) + \mathbf{R}_{zz}\left(\mathbf{q}, \frac{\partial^2 \mathbf{q}}{\partial z^2}\right). \tag{2}$$

We assume that we have a code with 2D-discretised equations in $\mathbf{R}_{2D}$ (governing a 3-component spanwise invariant state), while the operators $\mathbf{R}_z$ and $\mathbf{R}_{zz}$ are available for evaluation. All these residuals are assumed to be local functions (in the span-direction), depending solely on the value of the state or of its span-derivatives at the evaluation point. The span-derivatives in the governing equations are the only quantities which make the residual be explicitly non-local in the span-direction. We will give more details about these operators in Section 4 when introducing the compressible Navier–Stokes equations.

Due to the time- and spanwise- homogeneity of the configuration, we assume that the forcing $\mathbf{u}$ and the state $\mathbf{q}$ may be searched for as periodic both in time and in the spanwise direction. We will denote ω and β as the common fundamental angular frequency and fundamental spanwise wavenumber. In the Fourier domain, the harmonic responses

of $\check{\mathbf{u}}$ and $\check{\mathbf{q}}$ are described as Fourier series truncated at N harmonics in time and M harmonics in z direction:

$$\mathbf{u}(z,t) = \sum_{n=-N}^{N}\sum_{m=-M}^{M} \check{\mathbf{u}}_{n,m} e^{i(n\omega t + m\beta z)}, \qquad \mathbf{q}(z,t) = \sum_{n=-N}^{N}\sum_{m=-M}^{M} \check{\mathbf{q}}_{n,m} e^{i(n\omega t + m\beta z)}. \tag{3}$$

The components $(\cdot)_{n,m}$ represent the nth harmonic in time and the mth harmonic in the spanwise direction associated to $e^{i(n\omega t + m\beta z)}$ in the Fourier decomposition. For the solution to be real, we have the $(-n,-m)$ component should be the complex conjugate of the (n,m) one. In particular, the averaged component over time and z-direction, $(\cdot)_{0,0}$, is real. To determine the values $\check{\mathbf{q}}_{n,m}$, we inject the ansatz Eq. (3) for the forcing $\check{\mathbf{u}}$ and the response $\check{\mathbf{q}}$ into Eq. (1). We then either project this equation by a Galerkin method in spectral space or use a collocation method in physical space (see diagram (4)). The Galerkin projection method yields the HBM method used by Rigas et al. [4] for incompressible equations and by Sierra-Ausin et al. [5] for compressible equations. It enforces the governing equations in Fourier space for $N \times M$ harmonics. In the case of fully compressible Navier–Stokes equations, this method becomes prohibitively expensive due to the triple-sum involved in the cubic term and the Sutherland's relation which is not even polynomial. The collocation method bypasses this problem by enforcing the governing equation Eq. (1) in physical space at $(2N+1)\times(2M+1)$ discrete collocation points, which have the same information according to Nyquist–Shannon criteria as the $N \times M$ equations in Fourier space. This method is either called TSM [6] or High-Dimensional Harmonic Balance method (HDHB) [15]. As this collocation method is applied here both for time and for space, we call it Space–Time Spectral Method (S-TSM).

$$\begin{array}{cccccc}
z\backslash t & -N & \cdots & n & \cdots & N \\
-M & \ddots & \cdots & \cdots & \cdots & \ddots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
m & \vdots & \ddots & \check{\mathbf{q}}_{n,m} e^{i(n\omega t + m\beta z)} & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
M & \ddots & \cdots & \cdots & \cdots & \ddots
\end{array}$$

$$\Longrightarrow \quad \begin{array}{cccccc}
z\backslash t & 0 & \cdots & k & \cdots & 2N \\
0 & \ddots & \cdots & \cdots & \cdots & \ddots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
j & \vdots & \ddots & \mathbf{q}_{k,j} = \mathbf{q}(t_k, z_j) & \ddots & \vdots \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
2M & \ddots & \cdots & \cdots & \cdots & \ddots
\end{array} \tag{4}$$

2.2. Space-Time Spectral Method (S-TSM)

Eq. (1) is enforced at all collocation points in time, $t_k = k\Delta t$ with $\Delta t = (2\pi/\omega)/(2N+1)$, and span-direction, $z_j = j\Delta z$ with $\Delta z = (2\pi/\beta)/(2M+1)$:

$$\mathbf{M}\frac{\partial \mathbf{q}_{k,j}}{\partial t} + \mathbf{R}_{3D}(\mathbf{q}_{k,j}) = \mathbf{B}\mathbf{u}_{k,j}. \tag{5}$$

In spectral collocation methods, although expression Eq. (3) holds, the variables $\mathbf{q}$ and $\mathbf{u}$ are not searched for in spectral space (as done in HBM methods) but in physical space, at the collocation points $\mathbf{q}_{k,j}$ and $\mathbf{u}_{k,j}$. The derivatives $\mathbf{D}_t$, $\mathbf{D}_z$ and $\mathbf{D}_{zz}$ matrices can then be expressed in physical space following:

$$\frac{\partial \mathbf{q}_k}{\partial t} = \sum_{k'=0}^{2N} (\mathbf{D}_t)_{kk'}\mathbf{q}_{k'}, \qquad \frac{\partial \mathbf{q}_j}{\partial z} = \sum_{j'=0}^{2M} (\mathbf{D}_z)_{jj'}\mathbf{q}_{j'}, \qquad \frac{\partial^2 \mathbf{q}_j}{\partial z^2} = \sum_{j'=0}^{2M} (\mathbf{D}_{zz})_{jj'}\mathbf{q}_{j'}. \tag{6}$$

These derivatives exhibit exponential convergence properties as the number of harmonics increases [16] and do not induce CFL-like constraints. Yet, the drawback is that the stencil couples all points together in space and time, which makes it difficult to solve. The involved coefficients are computed through successive application of the Discrete Fourier Transform (DFT), the complex multiplicative derivative in frequency domain and the Inverse Discrete Fourier Transform (IDFT). The DFT ($\mathbf{E}$) and IDFT ($\mathbf{E}^{-1}$) matrices relate spectral-domain $\check{\mathbf{q}}_n$ and time-domain $\mathbf{q}_k$ variables through $\mathbf{q}_k = \sum_n (\mathbf{E}^{-1})_{kn}\check{\mathbf{q}}_n$ and $\check{\mathbf{q}}_n = \sum_k (\mathbf{E})_{nk}\mathbf{q}_k$. These matrices read:

$$(\mathbf{E})_{nk} = \frac{1}{2N+1} e^{-(i2\pi kn)/(2N+1)}\mathbf{I}, \qquad (\mathbf{E}^{-1})_{kn} = e^{(i2\pi kn)/(2N+1)}\mathbf{I}, \tag{7}$$

with $n \in [-N, N]$ (or $[-M, M]$) and $k \in [0, 2N]$ (or $[0, 2M]$) and $\mathbf{I}$ the identity matrix of the two-dimensional discretisation. Then, for instance, the first-order time-derivative reads:

$$\mathbf{D}_t = \mathbf{E}_t^{-1}(in\omega\mathbf{I})\mathbf{E}_t, \tag{8}$$

where $(in\omega\mathbf{I})$ designates a block diagonal matrix, each block being equal to the identity matrix times a complex coefficient $in\omega$. The coefficients of the derivative matrices $\mathbf{D}_{(\cdot)}$ can be computed and stored beforehand as they only depend on the truncation levels N or M. Therefore, not every Discrete Fourier Transformation is required to be performed during the computation of the residual. In the following, the full residual may be written as:

$$\mathbf{M}\mathbf{D}_t\mathbf{q} + \mathbf{R}_{2D}(\mathbf{q}) + \mathbf{R}_z\left(\mathbf{q}, \mathbf{D}_z\mathbf{q}\right) + \mathbf{R}_{zz}\left(\mathbf{q}, \mathbf{D}_{zz}\mathbf{q}\right) = \mathbf{B}\mathbf{u}, \tag{9}$$

where, with a small abuse of notations, all matrices $\mathbf{M}$, $\mathbf{B}$ and non-linear functions $\mathbf{R}_{2D}, \mathbf{R}_z, \mathbf{R}_{zz}$ now designate block diagonal versions of the quantities. For example, for two vectors $\mathbf{q}$ and $\mathbf{q}'$, the $(k_0 j_0)^{th}$ component of $\mathbf{R}_z\left(\mathbf{q}_{k_1 j_1}, \mathbf{q}'_{k_2 j_2}\right)$ is non-zero only if $k_0 = k_1 = k_2$ and $j_0 = j_1 = j_2$. Hence, only the derivative matrices $\mathbf{D}_t$, $\mathbf{D}_z$ and $\mathbf{D}_{zz}$ correspond to non-diagonal block-quantities and are the only terms coupling the various collocation points together. This system can be rewritten in compact form as:

$$\mathbf{R}(\mathbf{q}) = \mathbf{B}\mathbf{u}. \tag{10}$$

As it stands, spectral aliasing effects associated to interaction between the spectral method and the non-linearities of the governing equations may corrupt the solution. These aliasing effects may be mitigated by taking into account a large number of harmonics in time and space. Here, we demonstrate de-aliasing in time only, without loss of generality, and the same approach can be applied in space. The first step in the de-aliasing procedure involves defining an operator $\mathbf{P}$ that prolongates a solution with N harmonics to a solution with $N' > N$ harmonics, such that $\mathbf{q}' = \mathbf{P}\mathbf{q}$. This operator adds zero-amplitude harmonics from $N+1$ to N' to a solution with N harmonics:

$$\mathbf{P} = \mathbf{E}_{N'}^{-1}\begin{pmatrix} \mathbf{I}_{N,N} \\ \mathbf{0}_{N'-N,N} \end{pmatrix}\mathbf{E}_N, \tag{11}$$

which ensures conservation of mean energy between the initial and prolongated solution:

$$\frac{1}{2N'+1}\sum_{k=0}^{2N'} \mathbf{q}'^*_k\mathbf{Q}\mathbf{q}'_k = \frac{1}{2N+1}\sum_{k=0}^{2N} \mathbf{q}^*_k\mathbf{Q}\mathbf{q}_k, \tag{12}$$

where $\mathbf{Q}$ is any chosen inner product. This prolongation operator satisfies:

$$\mathbf{P}^*\mathbf{P} = \mathbf{I}_{N,N}, \tag{13}$$

so that $\mathbf{P}^*$ can be viewed as a restriction operator from N' harmonics to N harmonics by suppressing all frequencies between N and N'. For the general space–time problem, the de-aliased S-TSM governing equations may be written as:

$$\mathbf{P}^*\mathbf{R}'(\mathbf{P}\mathbf{q}) = \mathbf{B}\mathbf{u}, \tag{14}$$

where $\mathbf{R}'$ designates the residual obtained with N' harmonics in time and M' harmonics in space, while $\mathbf{q}$, $\mathbf{B}$ and $\mathbf{u}$ are quantities pertaining only to M and N harmonics. According to Orszag and Patterson [17], $(N' = 3/2N, M' = 3/2M)$ for quadratic non-linearities, while $(N' = 2N, M' = 2M)$ for cubic ones [9].

The non-linear system Eq. (14) may be solved by an iterative Newton method where a solution $\mathbf{q}$ is improved as $\mathbf{q}+\delta\mathbf{q}$, with

$$\left[\mathbf{P}^*\mathbf{A}'\mathbf{P}\right]\delta\mathbf{q} = -\mathbf{P}^*\mathbf{R}'(\mathbf{P}\mathbf{q}) + \mathbf{B}\mathbf{u}, \tag{15}$$

where $\mathbf{A}' = \partial\mathbf{R}'/\partial\mathbf{q}'|_{\mathbf{P}\mathbf{q}}$ is the Jacobian of the residual with (N', M') harmonics. It can then be shown (in spectral space and the harmonic balanced operator $\check{\mathbf{A}}'$) that the linear operator $\mathbf{P}^*\mathbf{A}'\mathbf{P}$, is exactly equal to the Jacobian with (N, M) harmonics, $\mathbf{A}$, so that:

$$\mathbf{A}\delta\mathbf{q} = -\mathbf{P}^*\mathbf{R}'(\mathbf{P}\mathbf{q}) + \mathbf{B}\mathbf{u}. \tag{16}$$

It is important to note that the de-aliasing technique only requires the evaluation of the residual with (N', M') harmonics but the linear operator to be solved that determines the solution is still an operator of (N, M) harmonics and not (N', M') harmonics. Hence, the de-aliasing technique only weakly increases the cost of the method due to the fact that the evaluation of the residual with (N', M') harmonics exhibits a negligible increase of cost with respect to the evaluation of the residual with (N, M) harmonics. The explicit expression of the Jacobian $\mathbf{A}$ is:

$$\mathbf{A} = \mathbf{M}\mathbf{D}_t + \mathbf{A}_{2D}(\mathbf{q}) + \mathbf{A}_0(\mathbf{q}) + \mathbf{A}_z(\mathbf{q})\mathbf{D}_z + \mathbf{A}_{zz}(\mathbf{q})\mathbf{D}_{zz}, \tag{17}$$

where

$$\begin{aligned}\mathbf{A}_{2D}(\mathbf{q}) &= \left.\frac{\partial\mathbf{R}_{2D}}{\partial\mathbf{q}}\right|_{\mathbf{q}},\ \mathbf{A}_0(\mathbf{q}) = \left.\frac{\partial\mathbf{R}_z}{\partial\mathbf{q}_1}\right|_{\mathbf{q},\mathbf{D}_z\mathbf{q}} + \left.\frac{\partial\mathbf{R}_{zz}}{\partial\mathbf{q}_1}\right|_{\mathbf{q},\mathbf{D}_z\mathbf{q}},\\ \mathbf{A}_z(\mathbf{q}) &= \left.\frac{\partial\mathbf{R}_z}{\partial\mathbf{q}_2}\right|_{\mathbf{q},\mathbf{D}_z\mathbf{q}},\ \mathbf{A}_{zz}(\mathbf{q}) = \left.\frac{\partial\mathbf{R}_{zz}}{\partial\mathbf{q}_2}\right|_{\mathbf{q},\mathbf{D}_z\mathbf{q}}.\end{aligned} \tag{18}$$

2.3. S-TSM constrained optimisation and adjoint-based gradient

Considering a general cost-functional $J(\mathbf{q})$, we seek the optimal forcing $\mathbf{u}$, of given finite amplitude $\mathbf{u}^*\mathbf{Q}_u\mathbf{u} = A^2$, that maximises/minimises the cost-functional $J(\mathbf{q})$, under the constraint of the governing equation Eq. (14). For example, for a maximisation problem, we seek:

$$\underset{\mathbf{u}^*\mathbf{Q}_u\mathbf{u}=A^2}{\text{argmax}}\ J(\mathbf{q}) \quad \text{subject to} \quad \mathbf{P}^*\mathbf{R}'(\mathbf{P}\mathbf{q}) = \mathbf{B}\mathbf{u}. \tag{19}$$

The Lagrangian, which is a function of the forcing $\mathbf{u}$, the state $\mathbf{q}$ and the Lagrangian multipliers $\tilde{\mathbf{q}}$ and λ, is

$$\mathcal{L}(\mathbf{q}, [\tilde{\mathbf{q}}, \lambda], \mathbf{u}) = J(\mathbf{q}) - \tilde{\mathbf{q}}^*\left(\mathbf{P}^*\mathbf{R}'(\mathbf{P}\mathbf{q}) - \mathbf{B}\mathbf{u}\right) - \lambda(\mathbf{u}^*\mathbf{Q}_u\mathbf{u} - A^2). \tag{20}$$

By zeroing the variations of $\mathcal{L}$ with the state $\mathbf{q}$, we obtain the equation defining the adjoint state $\tilde{\mathbf{q}}$:

$$\left[\mathbf{P}^*\mathbf{A}'\mathbf{P}\right]^*\tilde{\mathbf{q}} = \frac{dJ}{d\mathbf{q}}, \tag{21}$$

or, since $\mathbf{P}^*\mathbf{A}'\mathbf{P} = \mathbf{A}$ (which can be justified in frequency space by introducing the harmonic balanced operator)

$$\mathbf{A}^*\tilde{\mathbf{q}} = \frac{dJ}{d\mathbf{q}}. \tag{22}$$

The variation of the cost-functional δJ associated to an increment $\delta\mathbf{u}$ is then obtained by the partial derivative of the Lagrangian $\mathcal{L}$ with respect to the forcing:

$$\delta J = \left(\frac{dJ}{d\mathbf{u}}\right)^*\mathbf{Q}_u\delta\mathbf{u}, \qquad \frac{dJ}{d\mathbf{u}} = \hat{\mathbf{u}} - 2\lambda\mathbf{u}, \tag{23}$$

with $\hat{\mathbf{u}} = \mathbf{Q}_u^{-1}\mathbf{B}^*\tilde{\mathbf{q}}$ being linked to the Lagrange multiplier defined in Eq. (22). At an extremum of the cost-functional, the gradient $dJ/d\mathbf{u}$ must be zero, that is $\hat{\mathbf{u}}$ and $\mathbf{u}$ should be aligned and $\mathbf{u}^*\mathbf{Q}_u\mathbf{u} = A^2$.

3. Algorithms

3.1. Solving the non-linear de-aliased S-TSM system

We follow the approach introduced in Crivellini and Bassi [18], which adds an adaptive relaxation time to the Newton algorithm to increase robustness. The final Newton algorithm is given in algorithm 1 (see Section 3.4).

3.2. Solving the constrained optimisation problem

In order to find the optimal forcing $\mathbf{u}$, several algorithms exist [19]. Here, we choose the constrained adjoint-based one proposed by Rigas et al. [4], the steps of which are described in algorithm 2 (see Section 3.4). Here, we pick $c = 0.6$ as a good trade-off between robustness and rapid convergence. The convergence criterion θ_c is equal to 1°, this value being sufficient to obtain converged solutions for this configuration. Slightly higher values of θ_c ($1° < \theta_c < 3°$) yield similar optimal forcing shape but with minor oscillations.

3.3. Solving the direct and adjoint linearised S-TSM systems

The linear operator $\mathbf{A}$ exhibits a large dimension equal to the size of the two-dimensional discretisation multiplied by the number of collocation points. Hence, a direct inversion strategy for $\mathbf{A}$ is impossible when several harmonics (M', N') are considered. An iterative solver (Section 3.3.1) is used with an appropriate preconditioning strategy (Section 3.3.2).

3.3.1. Iterative GMRES solver

The linear systems involving $\mathbf{A}$ and $\mathbf{A}^*$ are solved iteratively by a GMRES algorithm, which requires matrix–vector products, $\mathbf{A}\mathbf{x}$ or $\mathbf{A}^*\mathbf{x}$, and preconditioners that approximate $\mathbf{A}$ and $\mathbf{A}^*$ that are easy to invert. Matrix–vector products are obtained by algorithmic differentiation, which avoids the explicit construction of the operator $\mathbf{A}$ and $\mathbf{A}^*$. In the following, we use right preconditioners, $\mathbf{\Pi}$ and $\mathbf{\Pi}^*$, which are described in the next section.

3.3.2. Preconditioner

We consider the block-circulant preconditioner suggested by Moulin [12] for TSM and extend it to S-TSM. Introducing the discrete z and t Fourier-decomposition of the Jacobian $\mathbf{A}$ defined in Eq. (17) (viewed as a function of time and space due to its dependence on the periodic solution $\mathbf{q}(t_k, z_j)$), we have:

$$\mathbf{A} = \mathbf{E}_t^{-1}\mathbf{E}_z^{-1}\check{\mathbf{A}}\mathbf{E}_z\mathbf{E}_t. \tag{24}$$

The matrix $\check{\mathbf{A}}$ is the harmonic balanced operator used in Rigas et al. [4]:

$$\check{\mathbf{A}} = (in\omega)\mathbf{M} + \check{\mathbf{A}}_{2D} + \check{\mathbf{A}}_0 + \check{\mathbf{A}}_z(im\beta) + \check{\mathbf{A}}_{zz}(-m^2\beta^2). \tag{25}$$

The preconditioner is used to facilitate the solution of the system involving the harmonic balanced operator. The matrices $\check{\mathbf{A}}_{2D}$, $\check{\mathbf{A}}_0$, $\check{\mathbf{A}}_z$ and $\check{\mathbf{A}}_{zz}$ are in general full-block matrices, their diagonal terms representing the action of the time- and space-averaged Jacobian, for example $(2N+1)^{-1}(2M+1)^{-1}\sum_{k=0,j=0}^{2N,2M}\mathbf{A}_{2D}(\mathbf{q}(t_k, z_j))$ and the off-diagonal terms representing the harmonics in time and space of the Jacobian. For the preconditioner, we will here consider a block-Jacobi strategy and neglect these off-diagonal terms. Also, for minimising the implementation cost, we will approximate the averaged Jacobian by the Jacobian of the solution $\bar{\mathbf{q}}$ averaged in z and t. In such a case, for example, $\check{\mathbf{A}}_z(im\beta) = \mathbf{E}_z\mathbf{A}_z(\bar{\mathbf{q}})\mathbf{E}_z^{-1}(im\beta) = (im\beta)\mathbf{A}_z(\bar{\mathbf{q}})$ since a diagonal constant block matrix commutes with $\mathbf{E}_z^{-1}$ and $(im\beta)$. We finally obtain the following preconditioner:

$$\mathbf{\Pi} = \mathbf{E}_t^{-1}\mathbf{E}_z^{-1}\check{\mathbf{\Pi}}\mathbf{E}_z\mathbf{E}_t, \tag{26}$$

where

$$\check{\mathbf{\Pi}} = (in\omega)\mathbf{M} + \mathbf{A}_{2D}(\bar{\mathbf{q}}) + \mathbf{A}_0(\bar{\mathbf{q}}) + (im\beta)\mathbf{A}_z(\bar{\mathbf{q}}) + (-m^2\beta^2)\mathbf{A}_{zz}(\bar{\mathbf{q}}) \tag{27}$$

is block diagonal and may be inverted with standard techniques. Note that these matrix inversions exactly correspond to those used by Rigas et al. [4] in the HBM framework. In principle, a more general preconditioner could be introduced by taking into account the exact averaged Jacobian on the diagonal and the off-diagonal terms by a block Gauss–Seidel, which could potentially improve the efficiency of the preconditioner.

3.4. End-to-end algorithm

The set of algorithms required to perform the adjoint-based optimisation with S-TSM is summarised and described below:

Algorithm 1 Iterative Newton method to solve S-TSM equations

1: Initialise state $\mathbf{q}_0$ with a guess solution.
2: $i \leftarrow 0$
3: **repeat** ▷ Newton loop for non-linear response.
4: $i \leftarrow i+1$
5: Compute local Δt based on a prescribed CFL number, that evolves as a function of the residual norm
6: Solve the linear system $\left(\frac{\mathbf{M}}{\Delta t} + \mathbf{A}(\mathbf{q}_{i-1})\right)\delta\mathbf{q}_i = -\mathbf{P}^*\mathbf{R}'(\mathbf{P}\mathbf{q}_{i-1}) + \mathbf{B}\mathbf{u}$ with algor. 3
7: $\mathbf{q}_i \leftarrow \mathbf{q}_{i-1} + \delta\mathbf{q}_i$
8: **until** $\|\delta\mathbf{q}_i\| < 10^{-5}$

Algorithm 2 Iterative method for finding optimal forcing

1: Set parameter $c \in]0;1]$: if $c \to 1$ (resp. $\to 0$) , the step-length is largest (resp. smallest)
2: Set $\epsilon \leftarrow 1$ (resp. $\leftarrow -1$) for maximisation (resp. minimisation) problem
3: Initialise the forcing $\mathbf{u}_0$ with a guess whose amplitude $\sqrt{\mathbf{u}_0^*\mathbf{Q}_u\mathbf{u}_0}$ is close to A
4: Solve for initial state $\mathbf{q}_0$: $\mathbf{P}^*\mathbf{R}'(\mathbf{P}\mathbf{q}_0) = \mathbf{B}\mathbf{u}_0$ (see algor. 1)
5: Scale $\mathbf{u}_0$ to the prescribed amplitude A
6: $i \leftarrow 0$
7: **repeat** ▷ Adjoint loop for forcing optimisation.
8: $i \leftarrow i+1$
9: Solve for updated state $\mathbf{q}_i$: $\mathbf{P}^*\mathbf{R}'(\mathbf{P}\mathbf{q}_i) = \mathbf{B}\mathbf{u}_{i-1}$ (see algor. 1)
10: Solve for adjoint state $\tilde{\mathbf{q}}_i$: $[\mathbf{A}(\mathbf{q}_i)]^*\tilde{\mathbf{q}}_i = \frac{dJ}{d\mathbf{q}}(\mathbf{q}_i)$ (see algor. 4)
11: Solve for reduced adjoint state: $\hat{\mathbf{u}}_i = \mathbf{Q}_u^{-1}\mathbf{B}^*\tilde{\mathbf{q}}_i$
12: Compute convergence angle: $\theta_i = \arccos\frac{\mathbf{u}_{i-1}^*\mathbf{Q}_u\hat{\mathbf{u}}_i}{A\gamma_i}$ where $\gamma_i^2 = \hat{\mathbf{u}}_i^*\mathbf{Q}_u\hat{\mathbf{u}}_i$
13: Compute step-length: $\lambda_i = \frac{\epsilon + c\cos\theta_i - \epsilon\sqrt{1-c^2\sin^2\theta_i}}{2A\alpha_i}$ where $\alpha_i = c/\gamma_i$
14: Update forcing: $\mathbf{u}_i = \mathbf{u}_{i-1} + \epsilon A\alpha_i(\hat{\mathbf{u}}_i - 2\lambda_i\mathbf{u}_{i-1})$
15: **until** $\theta_i < \theta_c$

Algorithm 3 Iterative solution of linear S-TSM system: $\mathbf{A}(\mathbf{q})\mathbf{x} = \mathbf{b}$, where $\mathbf{A}(\mathbf{q}) = \mathbf{M}\mathbf{D}_t + \mathbf{A}_{2\mathrm{D}}(\mathbf{q}) + \mathbf{A}_0(\mathbf{q}) + \mathbf{A}_z(\mathbf{q})\mathbf{D}_z + \mathbf{A}_{zz}(\mathbf{q})\mathbf{D}_{zz}$

1: Compute time- and -space averaged solution $\bar{\mathbf{q}}$
2: Perform LU-decomposition of $\check{\mathbf{\Pi}} = (in\omega)\mathbf{M} + \mathbf{A}_{2D}(\bar{\mathbf{q}}) + \mathbf{A}_0(\bar{\mathbf{q}}) + (im\beta)\mathbf{A}_z(\bar{\mathbf{q}}) + (-m^2\beta^2)\mathbf{A}_{zz}(\bar{\mathbf{q}})$: $\check{\mathbf{\Pi}} = \mathbf{L}\mathbf{U}$
3: Apply GMRES solver to preconditioned system: $[\mathbf{A}(\mathbf{q})]\mathbf{\Pi}^{-1}\mathbf{y} = \mathbf{b}$, where $\mathbf{\Pi}^{-1} = \mathbf{E}_t^{-1}\mathbf{E}_z^{-1}\mathbf{U}^{-1}\mathbf{L}^{-1}\mathbf{E}_z\mathbf{E}_t$
4: Compute solution: $\mathbf{x} = \mathbf{\Pi}^{-1}\mathbf{y}$

Algorithm 4 Iterative solution of adjoint linear S-TSM system: $[\mathbf{A}(\mathbf{q})]^*\mathbf{x} = \mathbf{b}$, where $\mathbf{A}(\mathbf{q}) = \mathbf{M}\mathbf{D}_t + \mathbf{A}_{2\mathrm{D}}(\mathbf{q}) + \mathbf{A}_0(\mathbf{q}) + \mathbf{A}_z(\mathbf{q})\mathbf{D}_z + \mathbf{A}_{zz}(\mathbf{q})\mathbf{D}_{zz}$

1: Compute time- and -space averaged solution $\bar{\mathbf{q}}$
2: Load the LU-decomposition of $\check{\mathbf{\Pi}} = \mathbf{L}\mathbf{U}$ performed in algor. 3
3: Apply GMRES solver to preconditioned system: $[\mathbf{A}(\mathbf{q})]^*\left(\mathbf{\Pi}^*\right)^{-1}\mathbf{y} = \mathbf{b}$, where $\left(\mathbf{\Pi}^*\right)^{-1} = \mathbf{E}_t^{-1}\mathbf{E}_z^{-1}\left(\mathbf{L}^*\right)^{-1}\left(\mathbf{U}^*\right)^{-1}\mathbf{E}_z\mathbf{E}_t$
4: Compute solution: $\mathbf{x} = \left(\mathbf{\Pi}^*\right)^{-1}\mathbf{y}$

4. Application: transition in supersonic boundary layer

4.1. Configuration

We consider the same adiabatic flat plate boundary layer configuration as Rigas et al. [4] but at a higher Mach number. The free-stream Mach number and free-stream temperature are respectively $M_\infty = 4.5$ and $T_\infty = 288$ K. All quantities are made non-dimensional with the following density, velocity, length and temperature scales: ρ_∞, U_∞, ν_∞/U_∞, T_∞. The spatial coordinates (x, y, z) then correspond to (Re_x, Re_y, Re_z), where for example $Re_x = U_\infty x/\nu_\infty$.

The flat plate geometry is studied in a rectangular computational domain. The domain starts with a thin boundary layer profile at $Re_{x,\mathrm{in}} = 0.3\times10^5$ and ends at $Re_{x,\mathrm{out}} = 3.6\times10^5$. The height of the domain is high enough in order not to affect the development of the boundary-layer or the stability analysis. In practice, the domain height is about $7\,\delta^*_{\mathrm{out}}$ with δ^*_{out} the compressible displacement thickness at the outlet. This gives $Re_{y,\mathrm{top}} = 35000$. The Cartesian mesh is equi-spaced in the x-direction and stretched in the wall-normal direction (y-direction). The mesh exhibits $(N_x, N_y) = (300, 150)$ points, which induces 225000 degrees of freedom in the state vector (due to the 5 conservative variables per grid point). This resolution, as well as the numerical schemes (see Section 4.2) are validated in Section 5.2. The following boundary conditions are applied around the rectangular domain: self-similar solution prescribed at the inlet (Dirichlet), adiabatic no-slip wall at the bottom, non-reflecting condition [20] at the top and zeroth-order extrapolation at the outlet.

4.2. BROADCAST code

The open-source BROADCAST code introduced in Poulain et al. [21] discretises the compressible Navier–Stokes equations with a Sutherland law (see Appendix A) within a finite-volume framework. The state corresponds to the conservative variables $\mathbf{q} = (\rho, \rho u, \rho v, \rho w, \rho E)$, where we recognise the density, stream-wise. cross-stream, span-wise velocities and total energy.

The $\mathbf{R}_{2D}$ residual for the inviscid flux follows a FE-MUSCL (Flux-Extrapolated-MUSCL) scheme (see [21] for more implementation details) which is a high order accurate upwind scheme [22] resulting from an upwind recursive correction to the leading truncation error term of a centred second order scheme. The order of accuracy of this high-order finite difference scheme applied in finite volume framework through the flux reconstruction has been assessed in [23] (see Fig. 6.(a) in the cited paper). This scheme has been also assessed in hypersonic flow simulations by [24] showing excellent results in accuracy and shock capturing features. In the present configuration, the FE-MUSCL scheme is chosen of the seventh order to achieve convergence (see Section 5.2). For the viscous fluxes, they are computed on a five-point compact stencil which is fourth-order accurate [25].

The $\mathbf{R}_z$ and $\mathbf{R}_{zz}$ components can be written as the sum of four functions whose expressions can be found in Appendix B:

$$\mathbf{R}_z(\mathbf{q}_1,\mathbf{q}_2) = \left.\frac{\partial\mathbf{C}_1}{\partial\mathbf{q}}\right|_{\mathbf{q}_1}\mathbf{q}_2 + \left(\left.\frac{\partial\mathbf{C}_2}{\partial\mathbf{q}}\right|_{\mathbf{q}_1}\mathbf{q}_2\right)\odot\left(\left.\frac{\partial\mathbf{C}_3}{\partial\mathbf{q}}\right|_{\mathbf{q}_1}\mathbf{q}_2\right) + \left.\frac{\partial^2\mathbf{C}_4}{\partial\mathbf{q}^2}\right|_{\mathbf{q}_1}\mathbf{q}_2\mathbf{q}_2, \quad (28)$$

$$\mathbf{R}_{zz}(\mathbf{q}_1,\mathbf{q}_2) = \left.\frac{\partial\mathbf{C}_5}{\partial\mathbf{q}}\right|_{\mathbf{q}_1}\mathbf{q}_2. \quad (29)$$

Notation $\odot$ refers to the Hadamard product (element-wise product of two matrices or vectors). A first order derivative, such as $\left.\frac{\partial\mathbf{C}_1}{\partial\mathbf{q}}\right|_{\mathbf{q}_1}$, refers to an operator that has properties similar to a Jacobian, and a second order derivative, such as $\left.\frac{\partial^2\mathbf{C}_4}{\partial\mathbf{q}^2}\right|_{\mathbf{q}_1}$ to a Hessian (which explains the two arguments).

The S-TSM implementation is based on the in-house python module from Moulin [12]. All linear systems involving sparse matrices are then solved using the PETSc software interface [26] which includes a preconditioned flexible iterative GMRES solver and the direct sparse LU solver from MUMPS [27].

The subroutines of BROADCAST code to compute the residuals and extract the derivative operators run sequentially. The linear systems are then multi-threaded with OpenMP using PETSc [28]. Furthermore, MPI parallelism is used for the parallel implementation of the S-TSM with each MPI process handling one time or spanwise collocation point associated with a round-robin strategy to minimise the memory

used and the number of parallel communications [29]. The current implementation does not support the distribution of one collocation point on multiple nodes through MPI communicators limiting then the maximum number of degrees of freedom for one collocation point.

We use algorithmic differentiation, here the *Tapenade* software [30], to access all derivatives of the discretised function $\mathbf{R}_{2D}, \mathbf{R}_z$ and $\mathbf{R}_{zz}$. The reverse mode of this software also allows to compute adjoint matrix–vector products. All matrix–vector products involving $\mathbf{A}_{2D}$ and $\mathbf{A}^*_{2D}$ are obtained with matrix-free methods. Yet, in the present version of the code, the matrices $\mathbf{A}_0, \mathbf{A}_z$ and $\mathbf{A}_{zz}$ have been stored in sparse format (see next paragraph) to access the matrix–vector products involving these matrices (such as $\mathbf{A}_0\mathbf{x}$ or $\mathbf{A}_0^*\mathbf{y}$). Thus, an improvement, mainly in memory consumption, could be done by using matrix-free methods for these matrix–vector products.

For the inversion of the preconditioner $\check{\mathbf{II}}$, the diagonal block matrices are first explicitly obtained in sparse format by successive matrix–vector products, with a multi-colouring strategy that takes advantage of the structured mesh topology [31]. In order to reduce the memory consumption when solving each block of the linear system of the preconditioner, different strategies are considered: a direct LU factorisation [27], a Block Low-Rank (BLR) approximated LU factorisation [32] (with a tolerance set at 10^{-4}), as well as a GMRES solver preconditioned by an incomplete LU (ILU). The LU-BLR factorisation is used in Section 4 and compared with the other strategies in Section 5.1 and Appendix F.

In the present version of the code, we have written all quantities with $(N' \geq N, M' \geq M)$ harmonics, keeping in mind that only the (N, M) first ones are meaningful for the solution $\mathbf{q}$. Eq. (15) for the Newton iteration then reads:

$$\mathbf{P}\mathbf{P}^*\mathbf{A}'\mathbf{x}' = -\mathbf{P}\mathbf{P}^*\mathbf{R}'(\mathbf{q}) + \mathbf{B}\mathbf{u}, \tag{30}$$

with the update $\delta\mathbf{q} = \mathbf{P}\mathbf{P}^*\mathbf{x}'$. The adjoint equation Eq. (21) is:

$$\mathbf{P}\mathbf{P}^*\mathbf{A}'^*\mathbf{y}' = \frac{dJ}{d\mathbf{q}}, \tag{31}$$

with $\tilde{\mathbf{q}} = \mathbf{P}\mathbf{P}^*\mathbf{y}'$. It can be shown that the present implementation provides the same results as the one presented in Section 2, but with an increased cost (all linear systems are solved with (N', M') harmonics instead of (N, M) harmonics). If the code was implemented in an optimised way, i.e. by writing all quantities with (N, M) harmonics and only evaluating the residual with (N', M') harmonics, we would gain $(N' - N + 1) \times (M' - M + 1)$ collocation points out of the $N' \times M'$ points considered in the present computations. Here, we have used $(N' = N, M' = 2M)$ since it was found that de-aliasing in time was not necessary here while it was mandatory for space (without de-aliasing the Newton-method fails to converge as soon as the targeted solution slightly departs from its mean). This indicates that the nonlinearity induces much stronger energy transfers towards high span-wavenumber than towards high frequency harmonics.

In the Newton algorithm 1, if the guess-solution is good enough (which is usually the case since we use continuation methods), the preconditioner may be kept fixed during the iterations, which saves a significant amount of CPU time. Hence, step 2 of algorithm 3 may be moved just after step 1 of algorithm 1. Similarly, for the optimal forcing algorithm 2, the preconditioner found in step 4 for $\mathbf{q}_0$ may be good enough for the rest of the algorithm (in steps 9 and 10, we keep the initial preconditioner based on the average of $\mathbf{q}_0$). Note however that in steps 3 of algorithms 3 and 4, the true (updated) matrices $\mathbf{A}(\mathbf{q})$ are used so that the exact solutions of the linear systems are targeted (without additional cost since matrix-products are obtained with matrix-free methods).

4.3. Finite amplitude optimal forcing

In the following, we perform non-linear optimisation (non-linear input–output analysis, algorithm 2) to identify the worst-case disturbances that trigger laminar/turbulent transition. The optimisation process consists in an iterative gradient-based method, which updates the forcing at each step to increase the mean drag-plate, subject to the constraints of the S-TSM equations and a specified forcing amplitude. This procedure involves three main steps: first, the computation of the base flow and the use of a resolvent analysis to initialise the optimisation algorithm (at low forcing amplitude) (see Section 4.3.1); second, the optimisation loop which iteratively computes the S-TSM solution, the adjoint S-TSM solution and updates the forcing up to convergence of the algorithm (see Section 4.3.2); then, progressive increase of the forcing amplitude and following of the optimal forcing solution by continuation. The initial condition of the optimisation algorithm (*i.e.* here by the base-flow and the linear optimal forcing, see Section 4.3.1) does not have any impact on the overall procedure since, at low forcing amplitude (where the initialisation step is performed), the non-linearities are weak and only one optimal solution therefore exists. Yet, when progressively increasing the amplitude, bifurcations can appear, and special care is, in principle, required in the vicinity of such points.

The general vector forcing $\mathbf{u}$ (Section 2, Section 3) is considered here to be a volume force $\mathbf{f}$ in the momentum equations (see Appendix A). The cost function $J(\mathbf{q})$ is chosen to be the mean drag-plate:

$$J(\mathbf{q}) = \left(\int_{y=0} \left(\bar{\mu}\partial_y\bar{u} - \mu_b\partial_y u_b \right) \, dx \right)^2, \tag{32}$$

where the overline notation denotes mean quantities (averaged over time and spanwise-direction). In the present implementation, we have picked a slightly simplified version of the cost-functional by neglecting the second-order term $(\bar{\mu} - \mu_b)\partial_y(\bar{u} - u_b)$ with respect to the first order term $\mu_b\partial_y(\bar{u}-u_b)+(\bar{\mu}-\mu_b)\partial_y u_b$. This approximation holds when either the relative change of the viscosity or of the friction remains weak, which is the case here.

4.3.1. Initialisation of optimisation algorithm by linear input–output analysis

Linear input–output analysis, also called resolvent analysis [33,34], around the base flow is performed to initialise the forcing in the non-linear (finite amplitude) input–output analysis (see algorithm 2). A two-dimensional steady base flow is computed by a pseudo-transient continuation method. The self-similar solution is used to initialise the calculation of the base flow. The algorithm converges in 7 iterations (high initial CFL as the self-similar solution is close to the base-flow solution) and decreases the residual L^2 norm by 12 orders of magnitude.

The optimal forcing from (linear) resolvent analysis for a particular frequency and spanwise wavenumber is the one which maximises the ratio:

$$\mu^2 = \sup_{\check{\mathbf{f}}\neq 0} \frac{\check{\mathbf{q}}^*\mathbf{Q}_q\check{\mathbf{q}}}{\check{\mathbf{f}}^*\mathbf{Q}_f\check{\mathbf{f}}}, \tag{33}$$

which is a common choice for supersonic boundary layer stability studies [21,35,36]. The measures $\mathbf{Q}_q$ and $\mathbf{Q}_f$, respectively, correspond to Chu's energy (see Appendix C) and L^2 norm, both being restricted to $Re_y \leq 18000$ to alleviate any effect from the top boundary condition. The forcing $\check{\mathbf{f}}$ is therefore only defined in this region. Since the linearised operators have already been extracted for base-flow computation, the linear input–output analysis only requires to solve a generalised eigenvalue problem to find the largest eigenvalue and its associated eigenvector [21,35]. We, therefore, use the Arnoldi algorithm of the SLEPc library [37] among the various Krylov-Schur methods [38] implemented therein. Direct LU-factorisation is used to compute the resolvent operator.

The linear amplification gain map is shown in Fig. 1. Frequencies are normalised as $F = \omega\nu_\infty/U_\infty^2$ and spanwise wavenumbers as $\beta = \beta\nu_\infty/U_\infty$. At Mach number 4.5, the strongest instability is the oblique first Mack mode [39] at similar frequency $F = 8 \times 10^{-5}$ and spanwise

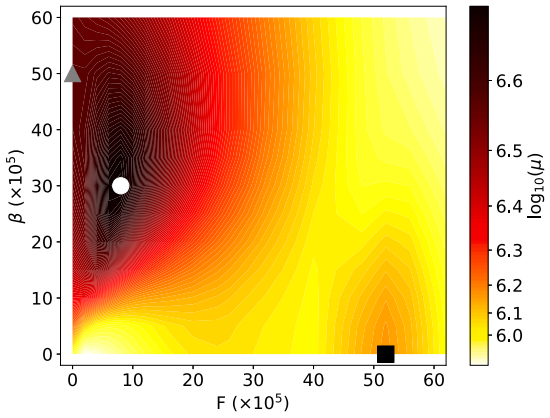


Fig. 1. Linear resolvent analysis at Mach number 4.5. White circle denotes the first Mack mode ($F = 8\times10^{-5}$, $\beta = 30\times10^{-5}$), grey triangle the streaks ($F = 0$, $\beta = 50\times10^{-5}$) and black square the second Mack mode ($F = 52\times10^{-5}$, $\beta = 0$).

wavenumber $\beta = 30\times10^{-5}$ as at Mach number 0.1 (see Appendix D). The stationary streaks (*i.e.* $F = 0$) are less dominant and maximal at $\beta = 50\times10^{-5}$, half of the spanwise wavenumber of the incompressible boundary layer. Eventually, a planar mode (*i.e.* $\beta = 0$) appears, only for Mach number greater than 4 [39], at high frequency $F = 52\times10^{-5}$: the second Mack mode. Note that these results (*e.g.* the gain values associated to each (F, β)), and also the non-linear optimisation results in Sections 4.4 and 4.3.2, depend on the extent of the domain. For example, the longer the domain, the lower the most amplified frequencies F of the first and second Mack modes will be [36].

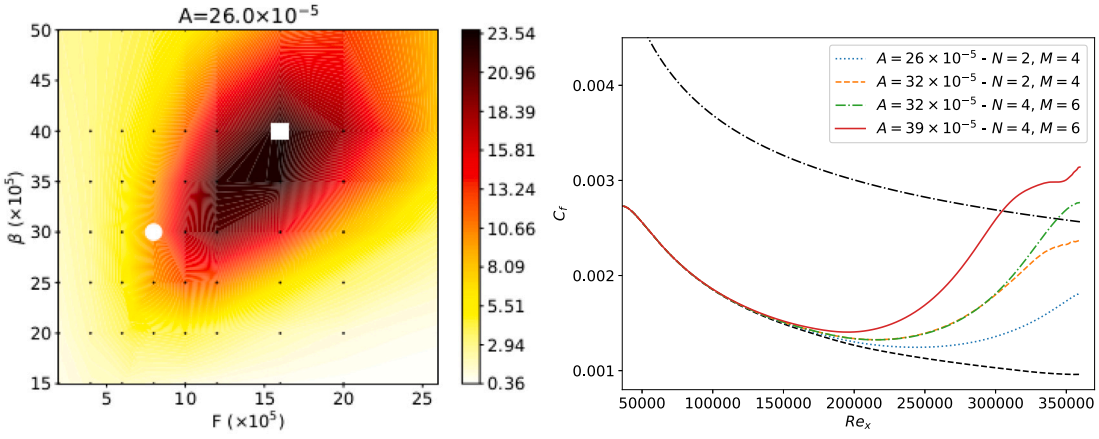
4.3.2. Optimisation results

The finite amplitude momentum forcing $\mathbf{f}$ for the non-linear input–output analysis is also applied with the same domain restriction as for the resolvent analysis. Only oblique forcing at a fundamental frequency 1ω and a fundamental spanwise wavenumber 1β is considered. Furthermore, symmetry in span is assumed so that we can prescribe $\hat{\mathbf{f}}_{1,1} = \hat{\mathbf{f}}_{1,-1}$. Therefore, the forcing is searched as two identical oblique waves of opposite angle in the $x-z$ plane.

From the linear input–output analysis (see Fig. 1), we observe that waves with frequencies close to the first Mack mode ones ($F \sim 10\times10^{-5}$) would need four levels of non-linear interactions to reach the frequency where the second Mack mode is amplified ($F \sim 50\times10^{-5}$). Rigas et al. [4] noticed for the low Mach number boundary layer that the non-linear interactions are stronger in the spanwise direction than in the time direction, meaning that more harmonics are necessary for M than N (see Appendix D) to capture the energy spread. Based on both observations above, it is expected that the second Mack mode will not be triggered by low-frequency waves in the non-linear input–output analysis for this specific configuration. We initially select $N = 2$ time harmonics and $M = 4$ spanwise harmonics from the conclusions drawn from the low Mach number boundary layer and we search for the optimal (F, β) values in the vicinity of the first Mack mode.

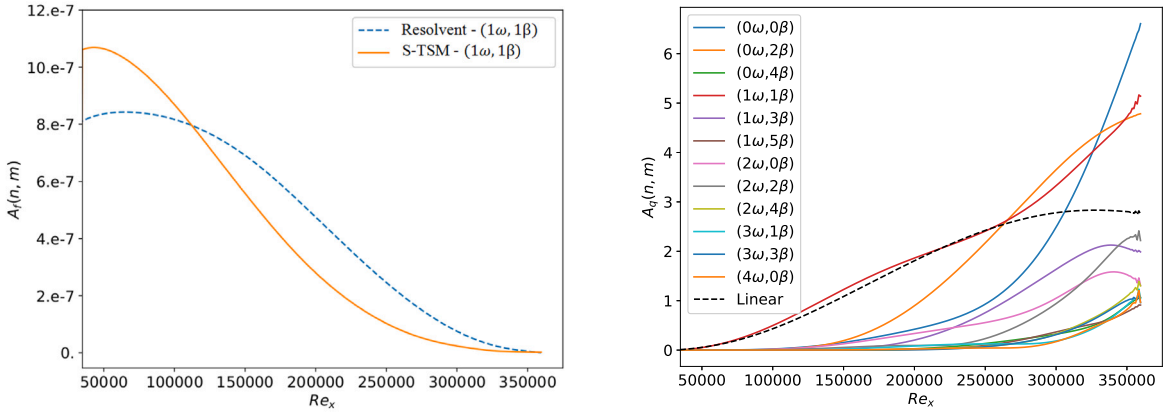
Starting at a low amplitude ($A = 10^{-5}$) for which the frequency/spanwise wavenumber of the drag peak is close to the one predicted by linear input–output analysis, larger amplitudes induce a shift of the peak value to higher frequency and spanwise wavenumber (see Fig. 2(a)). The optimal values ($F = 16\times10^{-5}$, $\beta = 40\times10^{-5}$) at the largest amplitude computed ($A = 26\times10^{-5}$) are substantially larger than the one from the resolvent analysis ($F = 8\times10^{-5}$, $\beta = 30\times10^{-5}$). For this optimal forcing, the mean-flow skin-friction coefficient evolution along the streamwise direction is plotted in Fig. 2(b). At this forcing amplitude, the skin-friction curve is between the laminar and turbulent skin-friction curves of a zero pressure gradient flat plate. Keeping F and β constant, the forcing amplitude is then increased further to reach the empirical turbulence levels. However, as we increase the forcing amplitude, the number of non-linear interactions is expected to increase. Therefore, $N = 2$ harmonics in time and $M = 4$ harmonics in spanwise direction are not sufficient to converge the non-linear response. Indeed, the solution at $A = 32\times10^{-5}$ yields different C_f predictions according to the level of truncation of N and M. By performing a number of simulations up to $N = 6$ and $M = 8$, it was found that $N = 4$ and $M = 6$ harmonics yield a converged non-linear solution at this forcing amplitude.

The spatial distribution of the optimal forcing and non-linear response for ($F = 16\times10^{-5}$, $\beta = 40\times10^{-5}$) are then analysed. The



(a) (b)

Fig. 2. (a) Square root of the cost function $\sqrt{J}$ with respect to the frequency F and the spanwise wavenumber β at Mach number 4.5 for $N = 2$ and $M = 4$ for a forcing amplitude $A = 26\times10^{-5}$. White circle denotes the maximum location of the linear optimal gain μ (first Mack mode at $F = 8\times10^{-5}$ and $\beta = 30\times10^{-5}$) while white square denotes the maximum location of the cost function J from the non-linear input–output analysis. Small black dots indicate the grid points where the non-linear input–output analysis has been performed. (b) Mean-flow skin-friction coefficient C_f at Mach number 4.5 along the streamwise direction for $F = 16\times10^{-5}$ and $\beta = 40\times10^{-5}$ at different forcing amplitudes and levels of truncation in time and spanwise directions. Black dashed line indicates the laminar skin-friction coefficient curve and dash-dotted line the turbulent curve.



(a) (b)

Fig. 3. Amplitude of the forcing and the response from the oblique forcing at $F = 16 \times 10^{-5}$, $\beta = 40 \times 10^{-5}$, $A = 32 \times 10^{-5}$, $N = 4$ and $M = 6$. (a) Amplitude of the optimal oblique forcing $A_f(1,1)$ given by the S-TSM (orange) compared with the linear forcing predicted by the resolvent analysis at the same (F, β) (blue). (b) Amplitude of the different harmonic components of the response $A_q(n,m)$ given by the S-TSM compared with the linear response predicted by the resolvent (dashed black line). Only the 12 harmonic components of largest amplitude are shown for clarity, they represent 99.4% of the total energy.

harmonic amplitude $A(n,m)$ (see in Appendix C for complete details of computation) of the forcing (Fig. 3(a)) and the response (Fig. 3(b)) are obtained for the forcing amplitude $A = 32 \times 10^{-5}$. In these plots, the harmonic amplitudes $A_q(n,m)$ (resp. $A_f(n,m)$) are given by the square root of the Chu's energy (resp. L^2 norm) in the wall-normal direction, and so they maintain a dependence on x, unlike the overall amplitude $A = 32 \times 10^{-5}$ which is defined as the integral of the L^2 norm at x-y directions.

Firstly, regarding the optimal forcing, both linear and finite amplitude are very similar, the largest difference being that the oblique waves predicted by the S-TSM are more localised and slightly more upstream in comparison to the waves from resolvent analysis. Non-linear responses are therefore similar when resulting from an optimal or a resolvent forcing (see Appendix E).

Secondly, about the non-linear response, the oblique first Mack mode forcing waves initially produce a response through the linear mechanism $(1\omega, 1\beta)$. Then, from $Re_x \sim 100000$, the non-linear interactions of the pair of symmetric oblique waves generate streaks $(1\omega, 1\beta) + (-1\omega, 1\beta) \rightarrow (0\omega, 2\beta)$ while their self-interactions produce mean-flow deviation $(1\omega, 1\beta) + (-1\omega, -1\beta) \rightarrow (0\omega, 0\beta)$. Notice that the harmonic component $(0\omega, 0\beta)$ in Fig. 3(b) represents the mean-flow deviation $\check{\mathbf{q}}_{0,0} - \mathbf{q_b}$. Higher harmonics in time $(2\omega, \ldots)$ are also produced and slowly grow in the downstream direction. As the linear amplification of the streaks is large, they grow faster than the higher harmonics in time. At $Re_x \sim 150000$, they interact with the fundamental oblique waves $(0\omega, 2\beta) + (1\omega, 1\beta) \rightarrow (1\omega, 3\beta)$ creating harmonic oblique response and at $Re_x \sim 200000$, they interact with themselves $(0\omega, 2\beta) + (0\omega, 2\beta) \rightarrow (0\omega, 4\beta)$ to generate streaks of higher harmonics. All these harmonics transfer energy to the mean-flow resulting in a growing mean-flow deviation and an increased mean-flow skin friction. The non-linear interactions at low Mach number (Fig. 8b and Fig. 15b in Rigas et al. [4]) are quite different than those observed here at Mach number 4.5. For low Mach number, the linear amplification of the streaks are larger than the oblique waves, therefore, once produced by the interaction of the oblique waves, the streaks grow much faster yielding higher harmonics in spanwise direction. They ultimately reach saturation and spread their energy to other harmonic components. At Mach number 4.5, the streaks are less amplified reaching similar level of energy than the oblique waves, producing therefore less spanwise harmonics.

Isosurfaces of the harmonic components and the full flow reconstruction in real space are plotted in Fig. 4 at the forcing amplitude $A = 32 \times 10^{-5}$ where the skin-friction overshoots above the turbulent skin-friction. One may observe that the combination of oblique waves (Fig. 4(a)) and streaks (Fig. 4(b)) dominate the shape of the instantaneous disturbance flow-field $\mathbf{q} - \mathbf{q_b}$ (Fig. 4(c)). Furthermore, initial stages of the laminar to turbulent transition are observed through staggered Λ-shaped vortices (Fig. 4(c)). The staggered Λ-shaped vortices are due to a path to transition called *O-regime* [40] initially discovered by [41,42]. This regime requires only the interaction of obliques waves, which generate streamwise vortices (nonlinear mechanism) and then streamwise streaks (linear lift-up mechanism), that finally undergo secondary instabilities to produce staggered Λ-vortices (contrary to the Herbert regime which involves the interaction of an oblique wave and a superharmonic planar Tollmien-Schlichting wave to produce staggered structures [43]). Hence, the Λ-structures arise from a sub-harmonic instability compared to streaks. The O-regime is a common path to transition in case of oblique first Mack mode breakdown [44,45]. The staggered maxima of the Λ–structures are connected by the legs and indicate the constant phase lines of the oblique waves [44,46]. One can see at the end of the domain the beginning of the split-up of the staggered maxima and the base flow distortion. As the computational domain is quite short (to reduce computational costs and facilitate convergence), only this first stage of transition can be qualitatively seen here with S-TSM. The next stages of transition are described in next section where a DNS is performed to validate the non-linear response from the S-TSM implementation.

4.3.3. Validation against DNS

The non-linear response obtained by the non-linear input–output analysis of the supersonic boundary layer is validated through DNS in this section. The DNS is performed with the in-house research solver FastS [47], which has been used extensively for hypersonic transition studies [48,49]. For the set-up of the three-dimensional DNS, boundary conditions equivalent to those of the input–output analysis are applied. The main DNS domain in the streamwise direction extends up to $Re_x = 10^6$, followed by a sponge region (stretching of the mesh) to avoid reflections. The spanwise domain length is twice the wavelength of the fundamental mode $\beta = 40 \times 10^{-5}$ and spanwise periodic boundary conditions are enforced. The mesh grid size is $(N_x, N_y, N_z) = (1618, 160, 96)$ with $\max(\Delta x^+, \Delta y^+_{wall}, \Delta z^+) = (2, 0.3, 6)$ sufficient to obtain converged results. The numerical scheme is a variant of a second-order upwind AUSM+(P) [50] associated with a fifth-order MUSCL reconstruction to limit the numerical dissipation. A similar shock-capture method [24] as

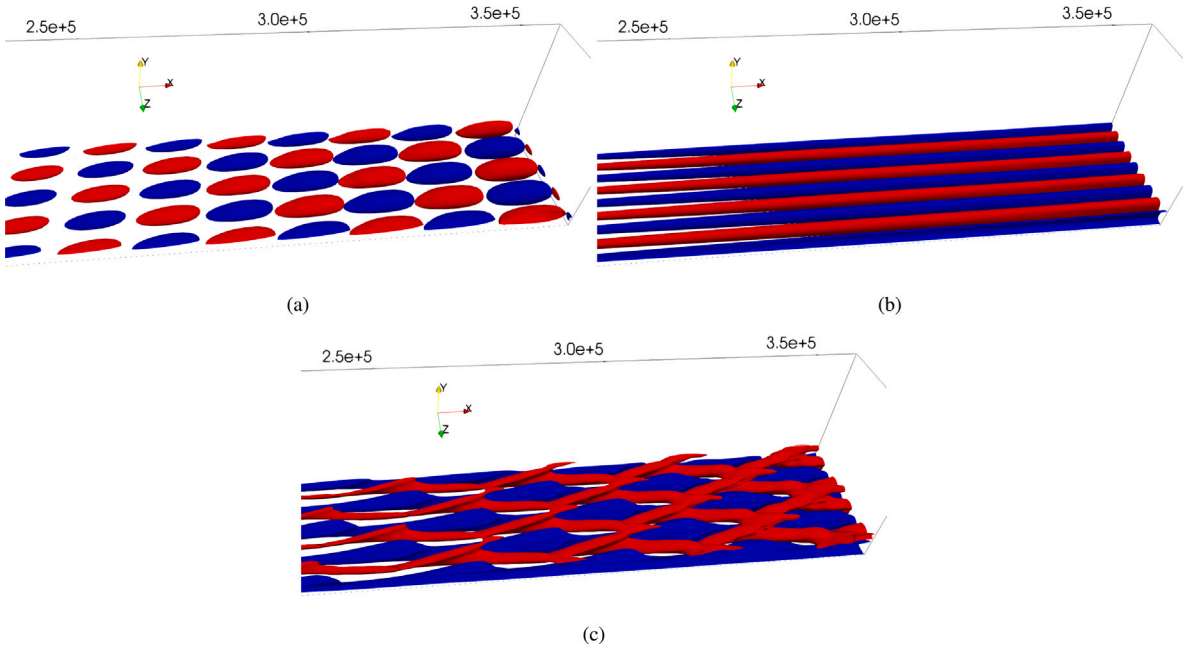


Fig. 4. Isosurfaces of streamwise velocity fluctuations $u' = \pm 0.1$ of the response downstream $Re_x = 250000$ resulting from the oblique forcing at $F = 16 \times 10^{-5}$, $\beta = 40 \times 10^{-5}$, $A = 32 \times 10^{-5}$, $N = 4$ and $M = 6$. (a) Fundamental oblique harmonic $u'_{1,1}$ $(1\omega, 1\beta)$. (b) Streak harmonic $u'_{0,2}$ $(0\omega, 2\beta)$. (c) Instantaneous disturbance flow-field $u - u_b$ (sum of all harmonics).

for the two-dimensional discretisation (BROADCAST) is used (reduction of the reconstruction order based on the variables state for the shock-capture in FastS).

DNS results are obtained by applying the same optimal fundamental forcing calculated by the non-linear input–output analysis using S-TSM ($F = 16 \times 10^{-5}$, $\beta = 40 \times 10^{-5}$) (see Section 4.3.2) consisting of symmetric oblique waves $(1\omega, \pm 1\beta)$. The instantaneous flow field up to $Re_x = 10^6$ is represented in Fig. 5. The streamwise velocity plot near the wall (see Fig. 5(a)) illustrates that the flow presents a sub-harmonic sinuous mode as the sinuous fluctuations of the low-speed streaks (in black) are associated with staggered (in x) varicose oscillations of the high-speed streaks (in white) [51]. Mutual and self-interactions of the streamwise vortex/streaks and the oblique waves [40] result in the staggered Λ-vortices (see Fig. 5(b)). Iso-surface of Q-criterion (see Fig. 5(c)) illustrates qualitatively all the stages of transition to turbulence. After the connection of their staggered maxima, the Λ-structures lift-up due to the streamwise vortices, making the near wall region vulnerable to instabilities [44]. Then, the formation of ring-shaped vortices in the outer region is observed, before saturation and the development of hairpin vortices implying the onset of turbulent breakdown.

S-TSM and DNS results for mean-flow skin-friction and amplitude of the different harmonics are compared in Fig. 6 up to the end of the S-TSM computational domain. Good agreement is observed even for the harmonics generated after two and three levels of non-linear interactions corresponding to $(1\omega, 3\beta)$ and $(2\omega, 4\beta)$. Small discrepancies close to the outlet at $Re_x = 3.6 \times 10^5$ are due to the presence of the boundary condition in the S-TSM.

In summary, the non-linear response from the S-TSM implementation has been validated for compressible flows. The validation of the full non-linear input–output analysis, which also includes the optimisation of the forcing to get the worst-case disturbances, has been performed on the same boundary layer configuration but at low Mach number to compare with Rigas et al. [4] and is presented in Appendix D.

4.4. Finite-amplitude optimal heat-flux control

In the previous sections, the worst-case disturbances have been identified by solving the drag maximisation problem, the solution of which is the finite-amplitude optimal forcing. Here, we seek the optimal control to minimise the maximum drag. In principle, we would like to target the coupled max–min problem that may be solved by alternate maximisation/minimisation steps up to convergence as done in Zuccher et al. [52]. Here, we restrict the procedure to the first maximisation step (Section 4.3.2) and minimisation step (present section).

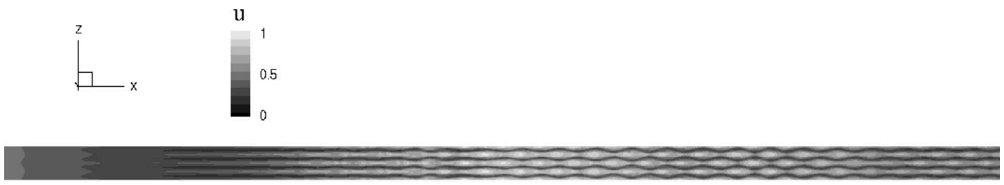
For the control to be close to realistic applications, we consider heat-flux actuators positioned at the wall surface acting only on the mean-flow. Hence, in this section, for the minimisation problem (resp. min–max problem), the general forcing $\mathbf{u}$ used through Sections 2 and 3 is a heat-flux at the wall acting on the mean-flow (resp. a volume force in the momentum equations).

During the optimisation, we allow any heat-flux harmonic components $(0\omega, m\beta)$ to amplify. The spanwise homogeneous mean-flow component $(0\omega, 0\beta)$ is by far the strongest and a small contribution $(0\omega, 2\beta)$ designed to damp the streaks arises. However, as the latter is small and reduces only by further 3% the cost function $\sqrt{J}$, only the spanwise homogeneous mean-flow heat-flux is retained here. Starting from a low amplitude, the optimal control amplitude is increased and results are given at the largest amplitude before the controlled mean flow deforms too much the boundary layer. Heat-flux control amplitude is computed through the energy coefficient C_θ based on the ratio of energy injected at the wall over the free-stream energy deficit,

$$C_\theta = \frac{\int_{y=0} \lambda |\frac{\partial T}{\partial y}| \, dx}{\int_{x=x_{out}} (\rho_\infty E_\infty U_\infty - \rho E u) \, dy}, \tag{34}$$

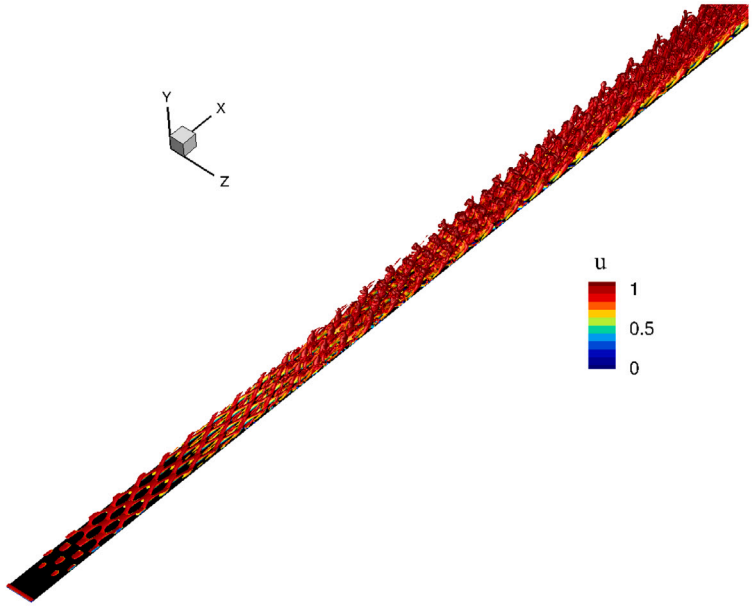
with $\phi_w = \lambda \frac{\partial T}{\partial y}$ the heat-flux injected (the uncontrolled case being adiabatic) where $\lambda = \mu c_p / Pr$ with μ the dynamic viscosity, P_r the Prandtl number and c_p the isobaric heat capacity.

The optimal mean-flow heat-flux at $C_\theta = 1.65 \times 10^{-2}$ is plotted in Fig. 7(a) for the case excited by the optimal fundamental forcing at



(a)

(b)



(c)

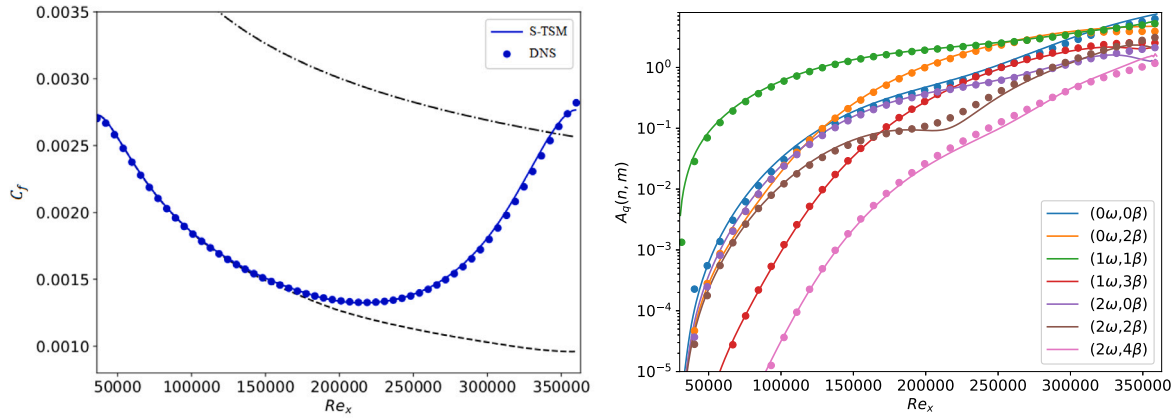
Fig. 5. Instantaneous streamwise velocity field u at (a) $Re_y = 1000$ and (b) $Re_y = 5000$. (c) Iso-surface of Q-criterion at 2×10^{-9} coloured by streamwise velocity u.

$F = 16 \times 10^{-5}$, $\beta = 40 \times 10^{-5}$, $A = 26 \times 10^{-5}$, $N = 2$ and $M = 4$. It is made of a strong heating peak close to the inlet, a weak cooling region downstream before a weak heating zone until the end of the domain. Spatial local stability analysis has been performed on the local mode corresponding to the first Mack mode (fundamental forcing). It is found that the location of the maximum of the amplification rate of the local mode is close to the maximum of positive heat-flux and branch II (downstream end of the unstable region of the local mode) corresponds to the location of the change from a positive (heating) to a negative (cooling) heat-flux. Note that branch I (upstream beginning of the unstable region) is located upstream of the inlet. Despite the short domain studied here, the trend of the optimal heat-flux is relatively similar to the optimal heat-flux from Jahanbakhshi and Zaki [14] computed for a transition case induced by the non-linear interactions of first and second Mack modes. They showed that alternation of almost spanwise homogeneous heating and cooling upstream, and spanwise periodic heat-flux patterns downstream, was optimal to delay transition.

Finally, we compute the mean-flow skin-friction coefficient (Fig. 7(b)) for two cases: the controlled flow with the frozen forcing (minimisation problem) and the flow with the new non-linear optimal forcing computed around the controlled mean-flow (minmax problem). The heat-flux control reduces the skin-friction from $Re_x \sim 200000$ to the end of the domain (little overshoot and oscillations due to the outlet boundary condition). The new optimal forcing under a controlled mean-flow leads to substantially smaller skin-friction levels than without control.

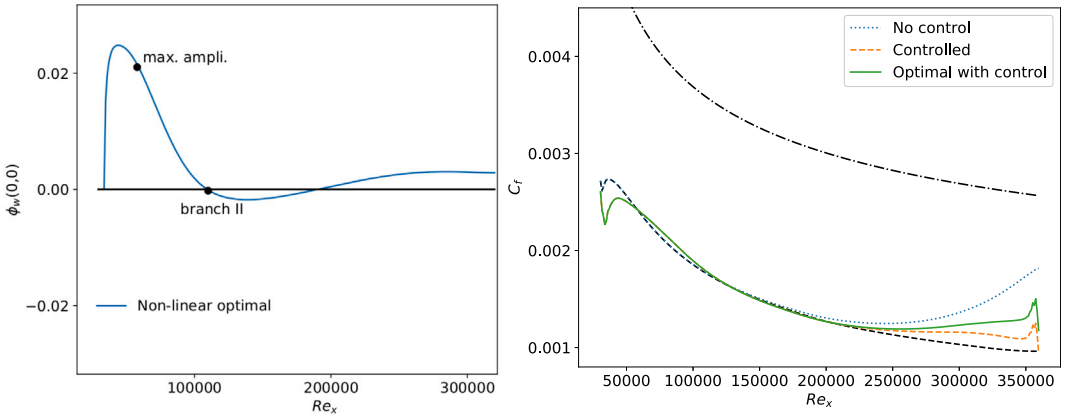
5. Computational performance

The computational performance of the non-linear input–output analysis through S-TSM are evaluated within the ONERA HPC environment: Intel Xeon (Cascade Lake - 6240R, 2.4 GHz) with multi-threading on 6 cores. The case considered is the supersonic boundary layer at Mach number 4.5 at the forcing amplitude where the laminar-to-turbulent transition occurs *i.e.* $F = 16{\times}10^{-5}$, $\beta = 40{\times}10^{-5}$, $A = 32{\times}10^{-5}$,



(a) (b)

Fig. 6. Comparison between S-TSM (solid lines) and DNS (dots) non-linear responses with the optimal forcing obtained for $F = 16\times10^{-5}$, $\beta = 40\times10^{-5}$, $A = 32\times10^{-5}$, $N = 4$ and $M = 6$. (a) Mean-flow skin-friction coefficient C_f along the streamwise direction. Black dashed line indicates the laminar skin-friction coefficient curve and dash-dotted line the turbulent curve. (b) Amplitude of the different harmonic components of the response $A_q(n,m)$. Only the 7 harmonics of largest amplitude are shown for clarity (although the DNS contains many more).



(a) (b)

Fig. 7. (a) Optimal mean-flow heat-flux $\boldsymbol{\phi}_w$ at $C_\theta = 1.65\times10^{-2}$ to delay the laminar to turbulent transition induced by the optimal fundamental forcing at $F = 16\times10^{-5}$, $\beta = 40\times10^{-5}$, $A = 26\times10^{-5}$, $N = 2$ and $M = 4$. Locations of branch II and maximal amplification rate from a local stability analysis of the first Mack mode are also indicated. (b) Mean-flow skin-friction coefficient C_f along the streamwise direction. Comparison without control (dotted lines), with control at $C_\theta = 1.65\times10^{-2}$ under frozen optimal forcing (dashed lines) and under the new optimal forcing computed around the controlled mean-flow (solid lines). Black dashed line indicates the laminar skin-friction coefficient curve and dash-dotted line the turbulent curve.

$N = 4$ and $M = 6$. The performance of the preconditioning strategy and the scalability with the two-dimensional number of degrees of freedom of the S-TSM are studied. As the parallel implementation of the S-TSM relies on one MPI process per collocation point, scalability cannot be evaluated with respect to the number of MPI cores for a given case. Moreover, the following results depend on the case explored as, from Eq. (27), the preconditioner $\check{\boldsymbol{\Pi}}$ explicitly depends on ω, β (high frequency and spanwise wavenumber reduce its sparsity) and the mean-flow $\bar{\mathbf{q}}$ (high mean-flow deviation deteriorates the approximation $\boldsymbol{\Pi} = \mathbf{A}(\bar{\mathbf{q}}) \approx \mathbf{A}(\mathbf{q})$).

5.1. Preconditioning strategy

First, the preconditioning strategy performances are evaluated on the non-linear input–output for $N_x = 300$ points (mesh used for the present work). The LU factorisation of $\check{\boldsymbol{\Pi}}$ is compared with the Block Low-Rank approximated LU factorisation (LU-BLR) [32] of $\check{\boldsymbol{\Pi}}$; factorisations applied on approximated block-circulant preconditioner $\check{\boldsymbol{\Pi}}$ computed with lower order convective schemes (reducing then the stencil and therefore the amount of off-diagonal components in the operator) are also considered. Elapsed time, RAM consumed and number of GMRES iterations to converge the GMRES algorithm are displayed in Table 1.

The LU-BLR considerably decreases the total RAM (10.1 Gb against 15.7 Gb per collocation point) by reducing the RAM of the LU factorisation. Furthermore, this approximated LU does not degrade much the preconditioner as the number of GMRES iterations to converge remains similar (202 against 195). Thanks to the time saved during the LU factorisation, it manages to reach a very similar elapsed time (409 s against 404 s). Then, about the lower order approximation strategy to build $\check{\boldsymbol{\Pi}}$, the RAM decreases because of sparser preconditioner. However, unexpectedly, the number of GMRES iterations decreases for a preconditioner more approximated with respect to the linear system. Both these factors strongly reduce the elapsed time. One possible clue to justify it may be that $\check{\boldsymbol{\Pi}}$ is computed from $\boldsymbol{\Pi} = \mathbf{A}(\bar{\mathbf{q}})$ where the mean flow $\bar{\mathbf{q}}$ acts as a low-pass filter. Nonetheless, computations carried

Table 1
Elapsed time (seconds) to perform one Newton iteration (including the preconditioning) and RAM consumption per collocation point (Gigabytes) for different preconditioning strategies.

Preconditioning strategy	Elapsed time	Elapsed time for LU	Total RAM	RAM for LU	Number of GMRES iterations
LU on $\check{\mathbf{I}}\check{\mathbf{I}}$ computed at order 7	404	103	15.7	14.1	195
LU on $\check{\mathbf{I}}\check{\mathbf{I}}$ computed at order 5	320	81	12.2	10.9	168
LU on $\check{\mathbf{I}}\check{\mathbf{I}}$ computed at order 3	253	57	7.3	6.1	153
LU-BLR on $\check{\mathbf{I}}\check{\mathbf{I}}$ computed at order 7	409	86	10.1	8.5	202
LU-BLR on $\check{\mathbf{I}}\check{\mathbf{I}}$ computed at order 5	311	79	8.6	7.2	173
LU-BLR on $\check{\mathbf{I}}\check{\mathbf{I}}$ computed at order 3	255	58	6.2	5	154

Table 2
Number of Newton iterations, averaged number of GMRES iterations per Newton iteration to converge the non-linear solution, elapsed time (seconds) and total RAM consumption (Gigabytes) for different forcing amplitudes A with $n_{dof} = 225000$ degrees of freedom for the case $F = 16\times10^{-5}$, $\beta = 40\times10^{-5}$, $N = 2$ and $M = 4$ (85 collocation points) running on 510 cores. The non-linear solution is computed from the base-flow for the lowest forcing amplitude and from the previous lower amplitude case for higher amplitudes. The optimal forcing is updated at each forcing amplitude. Starting from the linear forcing, it converges in 6 optimisation iterations at $A = 1\times10^{-5}$ and then requires about two optimisation iterations to update between two different forcing amplitudes.

Forcing amplitude A	Number of Newton iterations	Number of GMRES iterations per Newton ite.	Elapsed Time (s)	Total RAM (Gb)
1×10^{-5}	8	13	239	1301
5×10^{-5}	8	17	264	1295
10×10^{-5}	9	18	298	1299
15×10^{-5}	9	22	335	1307
20×10^{-5}	11	32	520	1312
26×10^{-5}	11	82	1203	1343

Table 3
Convergence of the total amplitude A_q defined in Eq. (C.5) for different levels of refinement in the streamwise direction and numerical scheme orders. N_c/ξ is the number of cells per fundamental (first Mack mode) wavelength in the streamwise direction. RAM consumption per collocation point (Gigabytes) and elapsed time (seconds) to construct the matrix and perform the LU-BLR factorisation of the block-circulant preconditioner are indicated as well as the elapsed time (seconds) per GMRES iteration.

n_{dof}	N_c/ξ	Scheme order	Total amplitude A_q	RAM for LU	Elapsed time for LU	Elapsed time per GMRES iteration
150000	19	7	2648.2	6	59	1.4
225000	29	7	2651.5	8.5	86	1.6
150000	19	5	2627.9	5.2	43	1.0
225000	29	5	2645.0	7.4	66	1.5
337500	43	5	2651.6	9.4	101	2.3
225000	29	3	2490.0	4.5	33	1.2
450000	57	3	2614.9	7.8	62	2.6
675000	86	3	2628.3	10.7	85	4

out on a different case (including less non-linear interactions) yields opposite results (see Appendix F) which confirms our conservative approach using LU-BLR without low-order approximation.

5.2. Memory and time performance

Here, the memory and time scalability performance of the S-TSM algorithm with respect to the number of degrees of freedom of the two-dimensional discretisation are assessed.

For the case considered, the system solves $(2N+1)\times(4M+1) = 225$ collocation points. For the mesh with $N_x = 300$ *i.e.* $n_{dof} = 225000$ degrees of freedom, the elapsed time to compute one GMRES iteration is around 1.6 s. With the GMRES relative tolerance set to 10^{-6} and BLR approximation for LU factorisation set to 10^{-4}, less than 250 GMRES iterations are required to converge the linear system and 14 Newton iterations to converge the non-linear solution when $A = 32\times10^{-5}$. This represents a total elapsed time (without the optimisation of the forcing) on 1350 cores of about 1 h and 35 min. Total RAM consumed is 3.5 Tb. However, these figures are valid for a specific configuration. As shown in Table 2 (realised with 85 collocation points), the number of GMRES iterations, as well as Newton iterations, strongly increases with the forcing amplitude when stronger non-linear interactions arise, increasing the total time and slightly the RAM consumption.

The elapsed time (Fig. 8(a)) and RAM usage (Fig. 8(b)) are evaluated with respect to the number of degrees of freedom for the construction and LU factorisation of the operator. The block-circulant preconditioner $\check{\mathbf{I}}\check{\mathbf{I}}$ scales similarly to the two-dimensional Jacobian $\mathbf{A}_{2D}$ both for time and memory. The block-circulant preconditioner $\check{\mathbf{I}}\check{\mathbf{I}}$ requires larger memory and time for the construction of the operator and its LU-BLR factorisation because of the additional spanwise contributions $\mathbf{A}_z$ (Eq. (27)) which append non-zero components to the two-dimensional operator (see Table B.4 of the spanwise contributions). To compute the total memory to perform the non-linear input–output analysis through S-TSM, the memory required for the preconditioner, for the Krylov subspaces of the GMRES algorithm and for the construction of $\mathbf{A}_z$ for the adjoint product are summed. This yields a similar scalability and the full algorithm consumes about three times more memory per collocation point than the two-dimensional code BROADCAST in this configuration.

Finally, a convergence analysis is performed with respect to the order of the numerical convective scheme. To assess that all harmonics are well resolved, the total amplitude A_q of the fluctuations is computed following Eq. (C.5) for different refinements in the streamwise direction and scheme orders (see Table 3). With the seventh order scheme, convergence is obtained for around 29 cells per wavelength of the fundamental mode (harmonic $(1\omega, 1\beta)$). Decreasing to the fifth order, 29 cells is enough to obtain similar results but convergence is reached at 43 cells per wavelength. For the third order scheme, it is necessary to considerably increase the number of cells per wavelength to at least 86 to reach similar total amplitude. Therefore, for converged results, it is beneficial to use a high order scheme (either 5th or 7th in this case) as even if the elapsed time for the LU-factorisation is similar to a low order scheme, their RAM consumption per collocation point is smaller (around 8 Gb against 10.7 Gb). Furthermore, the elapsed time per GMRES iteration, which is the predominant time constraint as around 200 GMRES iterations may be performed to solve one linear system at high forcing amplitudes, is much shorter (around 1.5 s against 4 s).

6. Conclusion

In the present article, we have developed a methodology to compute efficiently periodic solutions in both time and a span-direction. The starting point is an existing two-dimensional solver, in which we add spectral Fourier collocation discretisations for the time and spanwise directions. The solutions are obtained by use of a de-aliased quasi-Newton method, the S-TSM linear system coupling all collocation points being solved with an iterative preconditioned GMRES solver. The existence of an efficient preconditioner is a key ingredient for the efficacy of the method. Here, we have extended the block-circulant preconditioner introduced by [12] in the TSM case to the Space-TSM case. For gradient-based optimisation with many control parameters,

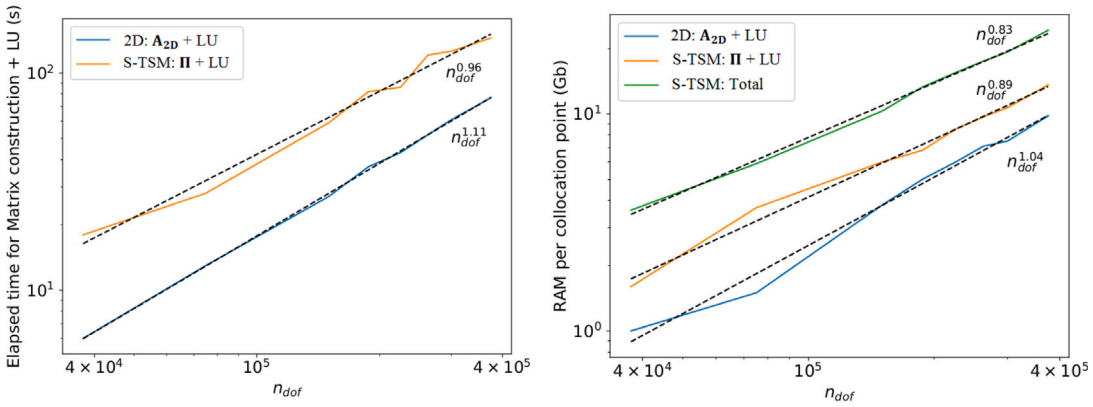


Fig. 8. Elapsed time and RAM consumption per collocation point. (a) Elapsed time (seconds). (b) RAM consumption per collocation point (Gigabytes). Comparison between the matrix construction and the LU-BLR factorisation of the two-dimensional Jacobian $\mathbf{A}_{2D}$ and the block-circulant preconditioner $\check{\mathbf{\Pi}}$. Total RAM used for S-TSM computation (construction and LU-factorisation of $\check{\mathbf{\Pi}}$ + construction of $\mathbf{A}_z$ for adjoint + Krylov subspace for GMRES) is also displayed.

we have introduced an adjoint linear system (based on the conjugate-transpose of the S-TSM operator) that may be solved with the same tools.

These methods have been implemented in the BROADCAST solver, which solves the 2D compressible Navier-Stakes equations within a finite-volume approach. The direct and adjoint matrix–vector products required in the GMRES solver are obtained, on the fly (without explicit construction of the matrices), by Automatic Differentiation of the discretised equations. Only the sparse matrices involved in the preconditioner are explicitly built since they are required by the Block Low-Rank approximated LU solver.

Finally, we have considered the laminar/turbulent transition within a supersonic flat-plate boundary layer at Mach number 4.5. The optimal finite-amplitude disturbance corresponding to an oblique breakdown scenario involving the first Mack mode have been determined based on a gradient-ascent method. Injecting the optimal forcing into a compressible DNS solver led to the same solution as the one provided by S-TSM, which validated the implementation. Also, a gradient-descent technique allowed us to determine the associated optimal finite-amplitude wall heat-flux distribution to delay transition as much as possible.

We have finally assessed the memory and time performances of the preconditioner and the S-TSM solver. The approach is very efficient for small amplitude forcing; yet, as the amplitude increases, more iterations are progressively needed for the solution of the large scale linear systems, to handle the progressively more energetic harmonics in the solution. This is obviously linked to the choice of the block-circulant preconditioner, which overlooks the off-diagonal terms of the harmonic balanced system, the latter precisely representing these harmonics. One possible way to improve the preconditioner would be to take into account these off-diagonal terms, for example with a block Gauss–Seidel method instead of a block-Jacobi solver. Also, more advanced iterative solvers, such as flexible inner-outer GMRES algorithms can be used [53]. Also, the overall cost can still be significantly reduced by implementing in a more efficient way the de-aliasing technique, since the (N', M') harmonics are strictly only required when the nonlinear residual is evaluated, while the large-scale direct and adjoint linear systems can be solved with only (N, M) harmonics. This means that the de-aliasing technique has a nearly negligible cost. Note also that, when the forcing amplitude increases, de-aliasing is absolutely mandatory in space (none of the problems could be solved when $M' = M$), but can be switched-off for time ($N' = N$). This is due to the fact that the non-linearities and the energy transfers seem much stronger in space than in time.

The method presented in this article, while a valuable tool for optimisation and physical analysis, is not without its limitations. First, the developed framework cannot handle (as it stands) multiple incommensurable frequencies or wavenumbers, which somewhat restricts its applicability to simple cases. Second, the memory cost of the method is much higher than the cost involved in a DNS, inducing that the number of harmonics studied is limited. The method is thus restricted to the study of early stages of transition when the dimensionality of the flow is still low. Third, the linear solvers involved in the Newton-method and in the adjoint computation being based on preconditioners assuming low-amplitude fluctuations (mean-flow preconditioner), the convergence properties progressively degrade as the solution deviates from its mean (large amplitude forcing). Finally, the presented method is restricted (as it stands) to the case where the solution is fully triggered by the imposed forcing (amplifier flow [54]) and not by an intrinsic instability mechanism (oscillator flow [54]). Yet, all of these limitations can in principle be improved or even fixed for some of them: the first and fourth points can easily be handled. The third one (preconditioner) is still an active research area.

CRediT authorship contribution statement

Arthur Poulain: Writing – original draft, Methodology, Investigation, Conceptualization. **Cédric Content:** Supervision, Software, Methodology, Conceptualization. **Aldo Schioppa:** Validation, Investigation. **Pierre Nibourel:** Validation, Supervision. **Georgios Rigas:** Supervision, Methodology, Formal analysis, Conceptualization. **Denis Sipp:** Supervision, Methodology, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work is funded by the French Agency for Innovation and Defence (AID) as part of the UK-FR PhD program. G. R. also acknowledges funding from the Air Force Office of Scientific Research (AFOSR),

United States/European Office of Aerospace Research and Development (EOARD) (Award FA8655-21-1-7009). We are grateful to J. Moulin for sharing the TSM module.

Appendix A. Compressible Navier–Stokes equations

In expanded form, if $\mathbf{u}$ designates for example a volume force $\mathbf{f}$ in the momentum equations (so that $\mathbf{B} = \mathbf{M}\mathbf{P}_{\mathbf{wf}}$ with $\mathbf{M}$ the Hermitian mass matrix and $\mathbf{P}_{\mathbf{wf}}$ the prolongation/restriction matrix to prolongate/restrict the forcing to specific components or flow regions), the continuous Navier–Stokes equations read

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0, \tag{A.1}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v} + p\mathbf{I} - \boldsymbol{\tau}) = \mathbf{f}, \tag{A.2}$$

$$\frac{\partial (\rho E)}{\partial t} + \nabla \cdot ((\rho E + p)\mathbf{v} - \boldsymbol{\tau} \cdot \mathbf{v} - \lambda \nabla T) = 0, \tag{A.3}$$

with $E = p/(\rho(\gamma-1)) + \frac{1}{2}\mathbf{v}\cdot\mathbf{v}$, $\boldsymbol{\tau} = \mu(\nabla\mathbf{v} + (\nabla\mathbf{v})^T) - \frac{2}{3}\mu(\nabla\cdot\mathbf{v})\mathbf{I}$, μ the dynamic viscosity, $\mathbf{I}$ the identity matrix, $\lambda = \mu c_p / Pr$, c_p the isobaric heat capacity and P_r the Prandtl number ($P_r = 0.72$). To close the system, two more equations are required. First, one assumes a homogeneous, thermally and calorically perfect gas. The perfect gas law is

$$p = \rho r T, \tag{A.4}$$

with $r = 287.1$ J kg^{-1} K^{-1} the specific gas constant.

Then, the Sutherland's law is selected to link the viscosity μ to the temperature [55].

$$\mu(T) = \mu_{ref} \left(\frac{T}{T_{ref}} \right)^{3/2} \frac{T_{ref} + S}{T + S}, \tag{A.5}$$

with $S = 110.4$ K the Sutherland's temperature, $\mu_{ref} = 1.716^{-5}$ kg m^{-1} s^{-1} and $T_{ref} = 273.15$ K.

Appendix B. Spanwise contributions of the Cartesian Navier–Stokes equations

The spanwise contributions $\mathbf{R}_z$ and $\mathbf{R}_{zz}$ gather all the z-derivative components in the residual $\mathbf{R}_{3D}$. The matrices for the z-derivatives are derived as

$$\frac{\partial \mathbf{C}_1}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial z} = \left(\sum_k \boldsymbol{\alpha}_k(\mathbf{q}) \frac{\partial \mathbf{a}_k(\mathbf{q})}{\partial \mathbf{q}} \right) \frac{\partial \mathbf{q}}{\partial z}, \tag{B.1}$$

$$\frac{\partial^2 \mathbf{C}_4}{\partial \mathbf{q}^2} \frac{\partial \mathbf{q}}{\partial z} \frac{\partial \mathbf{q}}{\partial z} = \left(\sum_l \lambda_l(\mathbf{q}) \frac{\partial^2 \mathbf{b}_l(\mathbf{q})}{\partial \mathbf{q}^2} \right) \frac{\partial \mathbf{q}}{\partial z} \frac{\partial \mathbf{q}}{\partial z}, \tag{B.2}$$

$$\frac{\partial \mathbf{C}_2}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial z} \odot \frac{\partial \mathbf{C}_3}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial z} = \sum_m \boldsymbol{\gamma}_m(\mathbf{q}) \frac{\partial \mathbf{c}_m(\mathbf{q})}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial z} \odot \frac{\partial \mathbf{d}_m(\mathbf{q})}{\partial \mathbf{q}} \frac{\partial \mathbf{q}}{\partial z}, \tag{B.3}$$

$$\frac{\partial \mathbf{C}_5}{\partial \mathbf{q}} \frac{\partial^2 \mathbf{q}}{\partial z^2} = \left(\sum_l \lambda_l(\mathbf{q}) \frac{\partial \mathbf{b}_l(\mathbf{q})}{\partial \mathbf{q}} \right) \frac{\partial^2 \mathbf{q}}{\partial z^2}, \tag{B.4}$$

with the expressions of $\boldsymbol{\alpha}_k, \mathbf{a}_k, \lambda_l, \mathbf{b}_l, \boldsymbol{\gamma}_m, \mathbf{c}_m$ and $\mathbf{d}_m$ listed in the Table B.4.

Table B.4
Spanwise contributions of the Cartesian Navier–Stokes equations.

Continuity equation

k	α_k	a_k	l	λ_l	b_l	m	γ_m	c_m	d_m
1	1	ρw							

Momentum equation in x-direction

k	α_k	a_k	l	λ_l	b_l	m	γ_m	c_m	d_m
1	1	$\rho u w - \mu \frac{\partial w}{\partial x}$	1	$-\mu$	u	1	-1	μ	u
2	$\frac{2}{3}\frac{\partial \mu}{\partial x}$	w							
3	$\frac{2}{3}\mu$	$\frac{\partial w}{\partial x}$							

Momentum equation in y-direction

k	α_k	a_k	l	λ_l	b_l	m	γ_m	c_m	d_m
1	1	$\rho v w - \mu \frac{\partial w}{\partial y}$	1	$-\mu$	v	1	-1	μ	v
2	$\frac{2}{3}\frac{\partial \mu}{\partial y}$	w							
3	$\frac{2}{3}\mu$	$\frac{\partial w}{\partial y}$							

Momentum equation in z-direction

k	α_k	a_k	l	λ_l	b_l	m	γ_m	c_m	d_m
1	1	$\rho w^2 + p + \frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)$	1	$-\frac{4}{3}\mu$	w	1	$-\frac{4}{3}$	μ	w
2	$-\frac{\partial \mu}{\partial x}$	u							
3	$-\frac{\partial \mu}{\partial y}$	v							
4	$-\mu$	$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$							

Energy equation

k	α_k	a_k	l	λ_l	b_l	m	γ_m	c_m	d_m
1	1	$(\rho E + p)w$	1	$-\lambda$	T	1	-1	λ	T
2	$\frac{2}{3}\frac{\partial(\mu u)}{\partial x}$	w	2	$-\mu u$	u	2	$-\mu$	u	u
3	$\frac{2}{3}\mu u$	$\frac{\partial w}{\partial x}$	3	$-\mu v$	v	3	$-u$	μ	u
4	$-\frac{\partial(\mu w)}{\partial x}$	u	4	$-\frac{4}{3}\mu w$	w	4	$-\mu$	v	v
5	$-\mu w$	$\frac{\partial u}{\partial x}$				5	$-v$	μ	v
6	$\frac{2}{3}\frac{\partial(\mu v)}{\partial y}$	w				6	$-\frac{4}{3}\mu$	w	w
7	$\frac{2}{3}\mu v$	$\frac{\partial w}{\partial y}$				7	$-\frac{4}{3}w$	μ	w
8	$-\frac{\partial(\mu w)}{\partial y}$	v							
9	$-\mu w$	$\frac{\partial v}{\partial y}$							
10	$-\mu\frac{\partial w}{\partial x}$	u							
11	$-u$	$\mu\frac{\partial w}{\partial x}$							
12	$-\mu\frac{\partial w}{\partial y}$	v							
13	$-v$	$\mu\frac{\partial w}{\partial y}$							
14	$\frac{2}{3}\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)$	w							
15	$\frac{2}{3}w\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)$	μ							
16	$\frac{2}{3}\mu w$	$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}$							

Appendix C. Amplitude of the harmonic components

The general solution writes

$$\check{\mathbf{q}}(x, y, z, t) = \sum_{n=-N}^{N} \sum_{m=-M}^{M} \check{\mathbf{q}}_{n,m}(x, y) e^{i(n\omega t + m\beta z)}. \tag{C.1}$$

In the case of a z-symmetrical solution, Eq. (C.1) expands as

$$\begin{aligned} \check{\mathbf{q}} = \mathbf{q}_\mathbf{b} + (\check{\mathbf{q}}_{0,0} - \mathbf{q}_\mathbf{b}) + \sum_{n=1}^{N} \left(\check{\mathbf{q}}_{n,0} e^{in\omega t} + c.c. \right) + \sum_{m=1}^{M} \left(\check{\mathbf{q}}_{0,m} e^{im\beta z} + c.c. \right) \\ + \sum_{n=1}^{N} \sum_{m=1}^{M} \left(\check{\mathbf{q}}_{n,m} e^{i(n\omega t + m\beta z)} + \check{\mathbf{q}}_{n,-m} e^{i(n\omega t - m\beta z)} + c.c. \right). \end{aligned} \tag{C.2}$$

For compressible flow, a common choice for the energy of the perturbations written $\mathbf{Q}_q$ is Chu's energy [56] in order to take into account the pressure ($\check{p}$) and entropy ($\check{s}$) disturbances:

$$E_{\text{Chu}} = \check{\mathbf{q}}^* \mathbf{Q}_{\text{Chu}} \check{\mathbf{q}} = \frac{1}{2} \int_\Omega \left(\rho_b |\check{\mathbf{v}}|^2 + \frac{1}{\gamma p_b} \check{p}^2 + \gamma(\gamma-1) M^4 p_b \check{s}^2 \right) d\Omega. \tag{C.3}$$

Chu's energy is the sum of the kinetic energy of the perturbation and a thermodynamic component (potential energy from compression and from heat exchange) with appropriate coefficients to exclude the conservative compression work [57] in order to obtain a norm which does not increase in time in the absence of sources of energy [56]. Matrix $\mathbf{Q}_q = \mathbf{Q}_{\text{Chu}}$ for Chu's energy norm is block-diagonal whose writing form with conservative variables is detailed in Bugeat et al. [35].

Therefore, the amplitude of each harmonic component of the response is defined using Chu's energy norm $\mathbf{Q}_q$ with the appropriate coefficient given by Eq. (C.2):

$$A_q(n,m) = \begin{cases} \sqrt{(\check{\mathbf{q}}_{0,0} - \mathbf{q}_\mathbf{b})^* \mathbf{Q}_q (\check{\mathbf{q}}_{0,0} - \mathbf{q}_\mathbf{b})}, & \text{if } (n,m) = (0,0), \\ \sqrt{2\check{\mathbf{q}}_{n,m}^* \mathbf{Q}_q \check{\mathbf{q}}_{n,m}}, & \text{if } (n,m) \in (1\ldots N, 0) \cup (0, 1\ldots M) \\ \sqrt{4\check{\mathbf{q}}_{n,m}^* \mathbf{Q}_q \check{\mathbf{q}}_{n,m}}, & \text{if } (n,m) \in (1\ldots N, 1\ldots M). \end{cases} \tag{C.4}$$

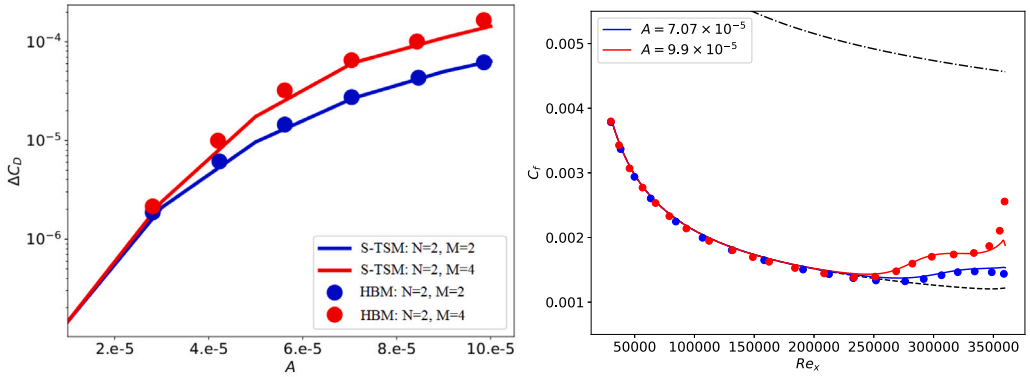


Fig. D.9. Comparison of the drag increase and the mean-flow skin-friction coefficient at different forcing amplitude between the S-TSM at Mach number 0.1 (solid lines) and the incompressible HBM from [4] (dots). (a) Drag increase ΔC_D with respect to the forcing amplitude A for $N = 2$ in time and two different levels of truncation in spanwise direction: $M = 2$ (blue) and $M = 4$ (red). (b) Mean-flow skin-friction coefficient C_f along the streamwise direction for $N = 2$ and $M = 4$ at two different forcing amplitudes: $A = 7.07 \times 10^{-5}$ (blue) and $A = 9.9 \times 10^{-5}$ (red). Black dashed line indicates the laminar skin-friction coefficient curve and dash-dotted line the turbulent curve.

The total amplitude of the system is then defined as

$$A_q = \sqrt{\sum_{n\geq 0, m\geq 0} A_q(n,m)^2}. \tag{C.5}$$

Similarly, the amplitude of the harmonic components of the forcing is defined as in Eq. (C.4) with the L^2 norm $\mathbf{Q}_u$ and the total forcing amplitude A reads

$$A = \sqrt{\sum_{n>0, m>0} A_f(n,m)^2}. \tag{C.6}$$

Appendix D. Low Mach number boundary layer

For comparison with Rigas et al. [4], the same adiabatic flat plate configuration is considered (same domain as for the supersonic boundary layer, see Section 4.1). The free-stream Mach number is set to 0.1 and the subsonic version of the inlet (Blasius solution with non-reflecting subsonic inlet) and outlet (non-reflecting condition with the free-stream pressure taken as reference as zero pressure gradient is assumed) boundary conditions are used. The base-flow is converged in 7 Newton iterations and the resolvent analysis is again performed for the initialisation of the non-linear input–output analysis.

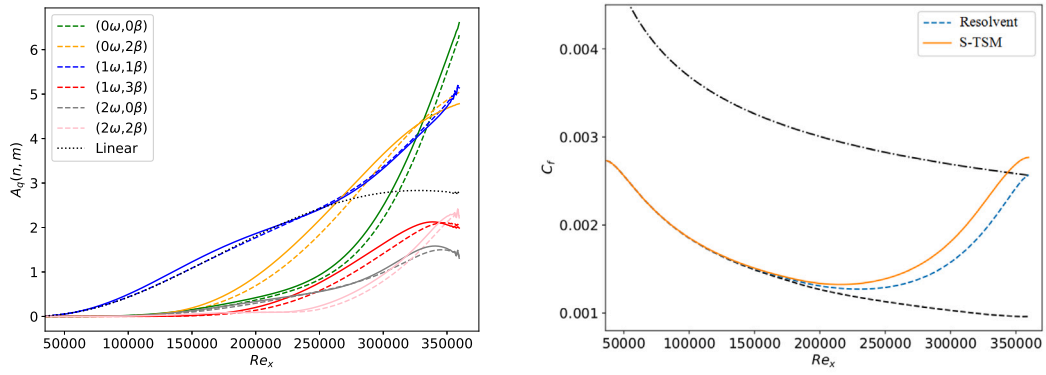
Rigas et al. [4] showed that the most dangerous symmetrical fundamental forcing in the non-linear framework was a pair of oblique TS waves whose frequency $F = 11.7 \times 10^{-5}$ and spanwise wavenumber $\beta = 33.4 \times 10^{-5}$ were close to the linear ones. Starting from the forcing predicted by the resolvent analysis at a small amplitude $A = 10^{-5}$ and the level of truncation of $N = 2$ in time and $M = 2$ in spanwise direction, the non-linear input–output analysis is then repeated with an increasing amplitude A by initialising the optimal forcing and the non-linear solution from the one obtained at the previous lower amplitude. Evolution of the drag increase ΔC_D with the forcing amplitude A is plotted in Fig. D.9(a). It is noticed that above $A = 3 \times 10^{-5}$, the non-linear interactions are not fully captured with $M = 2$ harmonics. Increasing the spanwise level of truncation up to $M = 4$ allows to take into account the higher harmonics generated by the $(0\omega, 2\beta)$ streaks. In Fig. D.9(b), the skin-friction of the mean-flow is plotted for both methods at two different forcing amplitudes departing from the laminar flow. Except for the values close to the outlet, both methods converge to the same skin-friction coefficient. The discrepancies are explained by the different outlet boundary conditions applied between the present finite volume compressible framework and the finite element incompressible framework from Rigas et al. [4]. Except these small discrepancies, the S-TSM compares well with the incompressible HBM from Rigas et al. [4] and the full non-linear input–output developed in this work is validated.

Appendix E. Non-linear response for optimal and resolvent forcings

The non-linear response from non-linear input–output analysis is similar with the one obtained without adjoint optimisation of the forcing *i.e.* with the initial linear resolvent forcing plotted in Fig. E.10. Because of the upstream location of the optimal fundamental forcing (see Fig. 3), the harmonics are generated slightly upstream (see Fig. E.10(a)) and follow a similar amplification to eventually result in larger mean-flow deviation. Furthermore, one may notice that the harmonics spread energy back to the fundamental mode from $Re_x = 260000$ by comparing the linear and non-linear responses of the resolvent forcing. In comparison with the resolvent forcing, the optimal forcing has increased the integral of the skin-friction of the mean-flow deviation by 32% and the transition onset $Re_{x,t}$ based on the location of the minimum of the mean-flow skin-friction coefficient by 6% (see Fig. E.10(b)).

Appendix F. Preconditioner approximated at lower order

To evaluate the approximation of $\check{\mathbf{II}}$ built with a lower order convective scheme, the case studied in this work is compared to a case with 2 harmonics in time for the TSM (no spectral derivatives in the spanwise direction) at low forcing amplitude. The comparison of the computational performance between both cases highlights two opposite behaviours (see Table F.5). The large system case converges quicker with the low order preconditioner while the small system converges slower, both requiring less memory. A third preconditioning strategy is also studied with an inner GMRES (nested GMRES at relative tolerance 10^{-4}) preconditioned by an ILU(2). Parametric studies on the level of sparsity for the ILU and the tolerance of the inner GMRES were performed and only the best result in term of RAM consumption is shown. However, the nested GMRES strategy is not optimal for this case as for the same RAM used as the low order preconditioner, it is slower by almost a factor five.



(a) (b)

Fig. E.10. Comparison of the non-linear responses obtained with the optimal forcing from non-linear S-TSM (solid lines) and from linear resolvent analysis (dashed lines) for $F = 16 \times 10^{-5}$, $\beta = 40 \times 10^{-5}$, $A = 32 \times 10^{-5}$, $N = 4$ and $M = 6$. (a) Amplitude of the different harmonic components of the response $A_q(n,m)$. The linear response predicted by the resolvent analysis is indicated in dotted black line. (b) Mean-flow skin-friction coefficient C_f along the streamwise direction. Black dashed line indicates the laminar skin-friction coefficient curve and dash-dotted line the turbulent curve.

Table F.5
Elapsed time (seconds) to perform a Newton iteration (including the preconditioning) and RAM consumption per collocation point (Gigabytes) for different preconditioning strategies. The case studied in this work (225000 degrees of freedom multiplied by 225 collocation points) *i.e.* the case where laminar-turbulent transition occurs is compared with a case with a coarser mesh and only 5 collocation points at a lower forcing amplitude including less non-linear interactions.

Case	Preconditioning strategy	Elapsed time	Total RAM	GMRES iterations
225000 dof × 225 collocation points	LU on $\tilde{\mathbf{H}}$ computed at order 7	404	15.7	195
	LU on $\tilde{\mathbf{H}}$ computed at order 5	320	12.2	168
150000 dof × 5 collocation points	LU on $\tilde{\mathbf{H}}$ computed at order 7	187	8.5	4
	LU on $\tilde{\mathbf{H}}$ computed at order 5	201	5.9	70
	inner GMRES on $\tilde{\mathbf{H}}$ preconditioned with ILU(2)	920	5.9	6

References

[1] Frantz RA, Loiseau J-C, Robinet J-C. Krylov methods for large-scale dynamical systems: Application in fluid dynamics. Appl Mech Rev 2023;75(3):030802.
[2] Slater JC. A numerical method for determining nonlinear normal modes. Nonlinear Dyn 1996;10:19–30.
[3] Shaabani-Ardali L, Sipp D, Lesshafft L. Time-delayed feedback technique for suppressing instabilities in time-periodic flow. Phys Rev Fluids 2017;2(11):113904.
[4] Rigas G, Sipp D, Colonius T. Nonlinear input/output analysis: application to boundary layer transition. J Fluid Mech 2021;911.
[5] Sierra-Ausin J, Citro V, Giannetti F, Fabre D. Efficient computation of time-periodic compressible flows with spectral techniques. Comput Methods Appl Mech Engrg 2022;393:114736.
[6] Gopinath A, Jameson A. Time spectral method for periodic unsteady computations over two-and three-dimensional bodies. In: 43rd AIAA aerospace sciences meeting and exhibit. 2005, p. 1220.
[7] Hall KC, Thomas JP, Clark WS. Computation of unsteady nonlinear flows in cascades using a harmonic balance technique. AIAA J 2002;40(5):879–86.
[8] Kenway GK, Kennedy GJ, Martins JR. Scalable parallel approach for high-fidelity steady-state aeroelastic analysis and adjoint derivative computations. AIAA J 2014;52(5):935–51.
[9] LaBryer A, Attar P. High dimensional harmonic balance dealiasing techniques for a Duffing oscillator. J Sound Vib 2009;324(3–5):1016–38.
[10] Sicot F, Puigt G, Montagnac M. Block-Jacobi implicit algorithms for the time spectral method. AIAA J 2008;46(12):3080–9.
[11] Mader CA, Martins JR. Derivatives for time-spectral computational fluid dynamics using an automatic differentiation adjoint. AIAA J 2012;50(12):2809–19.
[12] Moulin J. On the flutter bifurcation in laminar flows: linear and nonlinear modal methods [Ph.D. thesis], Institut polytechnique de Paris; 2020.
[13] Jahanbakhshi R, Zaki TA. Nonlinearly most dangerous disturbance for high-speed boundary-layer transition. J Fluid Mech 2019;876:87–121.
[14] Jahanbakhshi R, Zaki TA. Optimal heat flux for delaying transition to turbulence in a high-speed boundary layer. J Fluid Mech 2021;916:A46.
[15] Liu L, Thomas JP, Dowell EH, Attar P, Hall KC. A comparison of classical and high dimensional harmonic balance approaches for a Duffing oscillator. J Comput Phys 2006;215(1):298–320.
[16] Canuto C, Hussaini MY, Quarteroni A, Thomas Jr A, et al. Spectral methods in fluid dynamics. Springer Science & Business Media; 2012.
[17] Orszag SA, Patterson GS. Numerical simulation of three-dimensional homogeneous isotropic turbulence. Phys Rev Lett 1972;28:76–9.
[18] Crivellini A, Bassi F. An implicit matrix-free discontinuous Galerkin solver for viscous and turbulent aerodynamic simulations. Comput & Fluids 2011;50(1):81–93.
[19] Kerswell R. Nonlinear nonmodal stability theory. Annu Rev Fluid Mech 2018;50:319–45.
[20] Poinsot T, Lele S. Boundary conditions for direct simulations of compressible viscous flows. J Comput Phys 1992;101(1):104–29.
[21] Poulain A, Content C, Sipp D, Rigas G, Garnier E. BROADCAST: A high-order compressible CFD toolbox for stability and sensitivity using Algorithmic Differentiation. Comput Phys Comm 2023;283:108557.
[22] Cinnella P, Content C. High-order implicit residual smoothing time scheme for direct and large eddy simulations of compressible flows. J Comput Phys 2016;326:1–29.
[23] Petropoulos I, Costes M, Cinnella P. Development and analysis of high-order vorticity confinement schemes. Comput & Fluids 2017;156:602–20.
[24] Sciacovelli L, Passiatore D, Cinnella P, Pascazio G. Assessment of a high-order shock-capturing central-difference scheme for hypersonic turbulent flow simulations. Comput & Fluids 2021;230:105134.
[25] Shen Y, Zha G, Chen X. High order conservative differencing for viscous terms and the application to vortex-induced vibration flows. J Comput Phys 2009;228(22):8283–300.
[26] Balay S, Abhyankar S, Adams M, Brown J, Brune P, Buschelman K, et al. PETSc users manual. Argonne National Laboratory; 2019, ANL-95/11.
[27] Amestoy P, Duff I, L'Excellent J-Y, Koster J. A fully asynchronous multifrontal solver using distributed dynamic scheduling. SIAM J Matrix Anal Appl 2001;23(1):15–41.
[28] Lange M, Gorman G, Weiland M, Mitchell L, Southern J. Achieving efficient strong scaling with PETSc using hybrid MPI/OpenMP optimisation. In: Kunkel JM, Ludwig T, Meuer HW, editors. Supercomputing. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer; 2013, p. 97–108.
[29] Mundis N, Mavriplis D. Toward an optimal solver for time-spectral fluid-dynamic and aeroelastic solutions on unstructured meshes. J Comput Phys 2017;345:132–61.
[30] Hascoet L, Pascual V. The tapenade automatic differentiation tool: principles, model, and specification. ACM Trans Math Softw 2013;39(3):1–43.
[31] Mettot C. Linear stability, sensitivity, and passive control of turbulent flows using finite differences [Ph.D. thesis], Palaiseau, Ecole polytechnique; 2013.
[32] Mary T. Block low-rank multifrontal solvers: complexity, performance, and scalability [Ph.D. thesis], Université Paul Sabatier-Toulouse III; 2017.

[33] Sipp D, Marquet O. Characterization of noise amplifiers with global singular modes: the case of the leading-edge flat-plate boundary layer. Theor Comput Fluid Dyn 2013;27(5):617–35.
[34] Trefethen LN, Trefethen AE, Reddy SC, Driscoll TA. Hydrodynamic stability without eigenvalues. Science 1993;261(5121):578–84.
[35] Bugeat B, Chassaing J-C, Robinet J-C, Sagaut P. 3D global optimal forcing and response of the supersonic boundary layer. J Comput Phys 2019;398:108888.
[36] Nibourel P, Leclercq C, Demourant F, Garnier E, Sipp D. Reactive control of second Mack mode in a supersonic boundary layer with free-stream velocity/density variations. J Fluid Mech 2023;954:A20.
[37] Roman J, Campos C, Romero E, Tomás A. SLEPc users manual. Report No. DSIC-II/24/02, Valencia, Spain: D. Sistemes Informàtics i Computació Universitat Politècnica de València; 2015.
[38] Hernández V, Román J, Tomás A, Vidal V. Krylov-Schur methods in SLEPc. Tech. Rep. STR-7, Universitat Politecnica de Valencia; 2007.
[39] Mack L. The inviscid stability of the compressible laminar boundary layer. Space Prog Summary 1963;37:23.
[40] Jiang X, Lee C, Chen X, Smith C, Linden P. Structure evolution at early stage of boundary-layer transition: simulation and experiment. J Fluid Mech 2020;890:A11.
[41] Goldstein ME, Choi S-W. Nonlinear evolution of interacting oblique waves on two-dimensional shear layers. J Fluid Mech 1989;207:97–120.
[42] Schmid P, Henningson D. A new mechanism for rapid transition involving a pair of oblique waves. Phys Fluids A 1992;4(9):1986–9.
[43] Sayadi T, Hamman CW, Moin P. Direct numerical simulation of complete H-type and K-type transitions with implications for the dynamics of turbulent boundary layers. J Fluid Mech 2013;724:480–509.
[44] Unnikrishnan S, Gaitonde D. First-mode-induced nonlinear breakdown in a hypersonic boundary layer. Comput & Fluids 2019;191:104249.
[45] Zhou T, Liu Z, Lu Y, Wang Y, Yan C. Direct numerical simulation of complete transition to turbulence via first- and second-mode oblique breakdown at a high-speed boundary layer. Phys Fluids 2022;34(7):074101.
[46] Mayer CS, Von Terzi DA, Fasel HF. Direct numerical simulation of complete transition to turbulence via oblique breakdown at Mach 3. J Fluid Mech 2011;674:5–42.
[47] Péron S, Renaud T, Mary I, Benoit C, Terracol M. An immersed boundary method for preliminary design aerodynamic studies of complex configurations. In: 23rd AIAA computational fluid dynamics conference. 2017, p. 3623.
[48] Lugrin M, Beneddine S, Leclercq C, Garnier E, Bur R. Transition scenario in hypersonic axisymmetrical compression ramp flow. J Fluid Mech 2021;907:A6.
[49] Nibourel P. Robust reactive control of transition to turbulence of a supersonic boundary layer [Ph.D. thesis], Institut polytechnique de Paris; 2023.
[50] Laurent C. Étude d'écoulements transitionnels et hors équilibre par des approches DNS et RANS. [Ph.D. thesis], Ecole nationale supérieure d'arts et métiers-ENSAM; 2012.
[51] Andersson P, Brandt L, Bottaro A, Henningson DS. On the breakdown of boundary layer streaks. J Fluid Mech 2001;428:29–60.
[52] Zuccher S, Luchini P, Bottaro A. Algebraic growth in a Blasius boundary layer: optimal and robust control by mean suction in the nonlinear regime. J Fluid Mech 2004;513:135–60.
[53] Jadoui M, Blondeau C, Martin E, Renac F, Roux F-X. Comparative study of inner–outer Krylov solvers for linear systems in structured and high-order unstructured CFD problems. Comput & Fluids 2022;244:105575.
[54] Sipp D, Marquet O, Meliga P, Barbagallo A. Dynamics and control of global instabilities in open-flows: a linearized approach. Appl Mech Rev 2010;63(3).
[55] Sutherland W. LII. The viscosity of gases and molecular force. Lond Edinburgh Dublin Phil Mag J Sci 1893;36(223):507–31.
[56] George J, Sujith R. On Chu's disturbance energy. J Sound Vib 2011;330(22):5280–91.
[57] Hanifi A, Schmid P, Henningson D. Transient growth in compressible boundary layer flow. Phys Fluids 1996;8(3):826–37.