AIAA AVIATION Forum
August 2-6, 2021, VIRTUAL EVENT
AIAA AVIATION 2021 FORUM

10.2514/6.2021-2934

Input-Output Stability Analysis of a Flat-Plate, High-Speed Flow

D. Shekhtman* and N. J. Parziale†
Stevens Institute of Technology, Hoboken, NJ 07030, USA

A high-speed, flat-plate flow is used as a benchmark problem for input-output (IO) two-dimensional stability analysis. This paper expands on low-level details necessary to analyze the stability of a general 2D-flow via IO analysis. We construct the global, analytical Jacobian from a discretized version of the linearized Navier-Stokes equations. The maximization of an energy-normed input-to-output gain subject to linear state dynamics degenerates into a singular-value-decomposition problem, which yields optimal input and output directions for disturbances and perturbed variables, respectively. The input-output analysis of a high-speed flow over a flat-plate is performed in MATLAB on the optimal output directions at two nondimensional frequencies, $F = 1.6 \times 10^{-4}$ and $F = 2.2 \times 10^{-4}$. Locations of instabilities, mode shapes, and growth rates of perturbed temperature were obtained along the length of the plate and compared to those in the literature.

Nomenclature

A	= State Dynamics matrix, (-)
B	= Input Matrix, (-)
C	= Output Matrix, (-)
D	= Feedforward Matrix, (-)
$\mathcal{H}$	= State Space Transfer Function, (-)
N	= Global Decoding Matrix, Maps Conserved Variable set to Primitive Variable Set (-)
$\mathcal{M}$	= Chu Energy Norm Matrix, (-)
$\vec{q}$	= State Vector, (-)
$\vec{u}$	= Input Vector, (-)
$\vec{y}$	= Measurement Vector, (-)
M_∞	= Freestream Mach Number, (-)
U_∞	= Freestream x-velocity, (m/s)
V_∞	= Freestream y-velocity, (m/s)
μ_∞	= Freestream Dynamic Viscosity, (m/s)
mm_∞	= Freestream Molar Mass, (kg/mol)
ρ_∞	= Freestream Density, (kg/m^3)
P_∞	= Freestream Pressure, (Pa)
T_∞	= Freestream Temperature, (K)
L_{off}	= Offset Parameter, (m)
$\overline{\rho}$	= Nominal Density (kg/m^3)
$\overline{U}$	= Nominal x-Velocity, (m/s)
$\overline{V}$	= Nominal y-Velocity, (m/s)
$\overline{E}$	= Nominal Total Energy, (J)
$\overline{P}$	= Nominal Pressure, (Pa)
$\overline{T}$	= Nominal Temperature, (K)
ϵ	= Order of Perturbation, (-)
ρ_d	= Disturbed Density (kg/m^3)
u_d	= Disturbed x-Velocity, (m/s)
v_d	= Disturbed y-Velocity, (m/s)

*Graduate Student, Mechanical Engineering, Castle Point on Hudson, Hoboken, New Jersey, 07030.

†Associate Professor, Mechanical Engineering, Castle Point on Hudson, Hoboken, New Jersey, 07030, AIAA Senior Member.

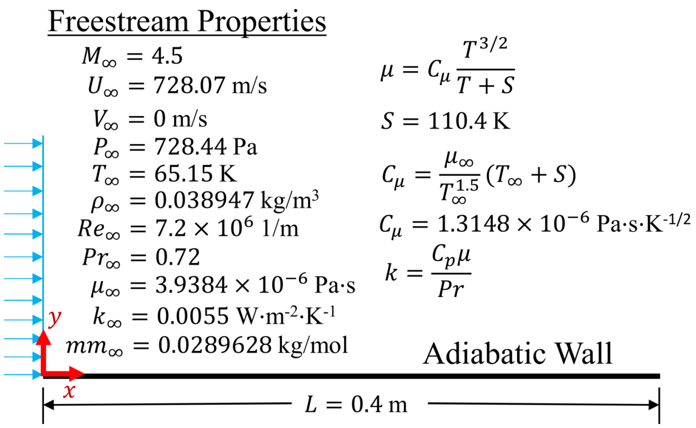

Freestream Properties
$M_\infty = 4.5$
$U_\infty = 728.07$ m/s
$V_\infty = 0$ m/s
$P_\infty = 728.44$ Pa
$T_\infty = 65.15$ K
$\rho_\infty = 0.038947$ kg/m^3
$Re_\infty = 7.2 \times 10^6$ 1/m
$Pr_\infty = 0.72$
$\mu_\infty = 3.9384 \times 10^{-6}$ Pa·s
$k_\infty = 0.0055$ W·m^{-2}·K^{-1}
$mm_\infty = 0.0289628$ kg/mol
$\mu = C_\mu \frac{T^{3/2}}{T + S}$
$S = 110.4$ K
$C_\mu = \frac{\mu_\infty}{T_\infty^{1.5}}(T_\infty + S)$
$C_\mu = 1.3148 \times 10^{-6}$ Pa·s·K$^{-1/2}$
$k = \frac{C_p \mu}{Pr}$
Adiabatic Wall
y
x
$L = 0.4$ m

Fig. 1 Flat Plate Flow Conditions from Ma *et al.* [1] and Dwivedi *et al.* [2]. An adiabatic wall is assumed. Sutherland's law is assumed to describe viscosity. Flow conditions are chosen such that the second Mack Mode is present.

E_d = Disturbed Total Energy, (J)
P_d = Disturbed Pressure, (Pa)
$\overline{\tau}_{xx}$ = x-Normal Stress, (Pa)
$\overline{\tau}_{yy}$ = y-Normal Stress, (Pa)
$\overline{\tau}_{xy}$ = xy-Shear Stress, (Pa)
μ = Dynamic Viscosity, (Pa·s)
λ = Bulk Viscosity, (Pa·s)
k = Thermal Conductivity, (W/(m·K))
T_d = Disturbed Temperature, (K)
x = Physical x-coordinate, (m)
y = Physical y-coordinate, (m)
ξ = Computational ξ-coordinate, (-)
η = Computational η-coordinate, (-)

I. Introduction

An Input-Output (IO) stability analysis is conducted for a nominal, steady, two-dimensional (2D) laminar flat-plate flow with the same freestream flow conditions as Ma and Zhong [1] and Dwivedi et al. [2]. Of greatest interest are the second mode shapes along the length of the flat plate because they contain information on growth rates and N-factors. A sketch of the flat-plate flow problem under consideration is shown in Fig. 1. The first-order disturbance equations are mapped to a uniform computational domain, finite differenced, and finally rearranged into a linear state space representation. The state space model is a linearization about a steady 2D base flow. It captures the first-order dynamics of perturbed field quantities, normally neglected by conventional, time-averaging turbulence models in CFD codes [3], and it brings to bear modern control theory for linear multivariable systems [4]. By minimizing the inverse input-to-output, energy-weighted amplification of a perturbed flow field, the problem is recast into a singular value decomposition (SVD) problem, for which an optimal disturbance direction and perturbed output direction can be determined for a particular 2D flow.

Researchers that have studied the stability of flow over a flat plate include Ma and Zhong [1], Dwivedi et al. [2], Cook et al. [5], and Kuehl [6]. For laminar flat-plate base flow, the similarity solution obtained via the Illingworth transformation could be used [3, 6, 7], but some researchers preferred a computational fluid dynamics code to generate the base flow [2]. High Reynolds numbers $O(10^6)$ and high enough Mach number $M \geq 4.5$ were chosen to ensure the existence of a second-mode instability [6]. To determine instabilities, Dwivedi et al. [2] and Cook et al. [5] used IO Analysis to find optimal perturbation directions and growth rates; and Kuehl [6] used a Lyapunov stability criterion [8] on the one-dimensional (1-D), parallel flow cycle-averaged disturbance acoustic energy equation. Researchers have characterized stability using streamwise perturbed temperature profile shapes, streamwise growth rates, N-Factors, and boundary layer flow profiles.

This paper intends to (1) explain how to construct the Jacobian A from the disturbance equations, (2) prove that the SVD indeed maximizes the energy-normed gain of a flow system, (3) solve the SVD problem without the need for quadruple precision datatypes for linear algebra operations, and (4) conduct demonstrative flat-plate stability calculations in the MATLAB programming environment. This paper serves both an audience new to input-output analysis and members of the general modern controls community who study distributed linear systems. This work is also done in support of ongoing experimental research in second-mode thermoacoustic instability and separation bubble stability. Such research includes focused laser differential interferometry measurements over a sharp cone [9] and krypton tagging velocimetry of a separation bubble in a 2D wedge flow [10].

II. Governing Equations

For a gas flow with a velocity field $\vec{V} = \begin{bmatrix} u & v & w \end{bmatrix}^T$, pressure field P, temperature field T, total energy field E, and volumetric heating field $\frac{dQ}{dt}$, the compressible Navier-Stokes (NS) equations are written as follows in conservation form for a 2D-Cartesian coordinate system:

$$\frac{\partial \mathbf{U}}{\partial t} = -\frac{\partial}{\partial x}\begin{bmatrix} \rho u \\ \rho u^2 + P - \tau_{xx} \\ \rho u v - \tau_{xy} \\ (E+P)u - u\tau_{xx} - v\tau_{xy} - k\frac{\partial T}{\partial x} \end{bmatrix} - \frac{\partial}{\partial y}\begin{bmatrix} \rho v \\ \rho u v - \tau_{xy} \\ \rho v^2 + P - \tau_{yy} \\ (E+P)v - u\tau_{xy} - v\tau_{yy} - k\frac{\partial T}{\partial y} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{dQ}{dt} \end{bmatrix}, \tag{1}$$

where $\mathbf{U} = \begin{bmatrix} \rho & \rho u & \rho v & E \end{bmatrix}^{\mathbf{T}}$ is the state vector and k is the conductivity of the gas. The following constitutive equation is applied for shear stresses τ_{ij}:

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \delta_{ij}\lambda\nabla\cdot\vec{V}, \quad ij = x, y \tag{2}$$

where μ is the dynamic viscosity and λ is the bulk viscosity. Stokes' hypothesis is assumed ($\lambda = -(2/3)\mu$). A calorically perfect, ideal gas is assumed:

$$\begin{aligned} P &= \rho(\gamma-1)C_v T \\ &= (\gamma-1)\left[E - \left(\frac{\vec{V}\cdot\vec{V}}{2}\right)\rho\right], \end{aligned} \tag{3}$$

where γ is the specific heat ratio, and C_v is the specific heat at constant volume.

III. Linearized Navier-Stokes Equations

This paper focuses on the stability of 2D flows and thus neglects disturbances and perturbations in the spanwise, z-direction. Flow field variables (ρ, u, v, P, T) are disturbed by an input disturbance $\vec{u}$ resulting in a perturbed state vector $\vec{q}$. This can be interpreted as the nominal flow field $\overline{\mathbf{U}}$ being perturbed by $\vec{q}$, following a disturbance by $\vec{u}$. Via the following substitutions,

$$\rho = \overline{\rho} + \epsilon\rho_d; \quad u_i = \overline{u}_i + \epsilon u_{d,i}; \quad E = \overline{E} + \epsilon E_d; \quad P = \overline{P} + \epsilon P_d, \quad \text{and} \quad T = \overline{T} + \epsilon T_d, \tag{4}$$

in the 2D conservative NS equations [11, 12], the linearized Navier-Stokes equations are derived by keeping all terms of order ϵ: $O(\epsilon)$ [3, 13, 14], where, the power of ϵ indicates the order of perturbation. For brevity, the disturbance equations are written in tensor notation as follows. The first-order disturbance equation for continuity is

$$\frac{\partial \rho_d}{\partial t} = -\frac{\partial}{\partial x_i}(\overline{\rho}u_{di} + \rho_d\overline{u}_i). \tag{5}$$

summed over $i = (1,\ 2)$. The first-order disturbance equation for momentum is

$$\frac{\partial(\overline{\rho}u_{di})}{\partial t} = -\frac{\partial\rho_d}{\partial t}\overline{u}_i - \rho_d\frac{\partial\overline{u}_i}{\partial t} - \frac{\partial}{\partial x_j}\left[\rho_d\overline{u}_i\overline{u}_j + \overline{\rho}\ \overline{u}_i u_{dj} + \overline{\rho}u_{di}\overline{u}_j + P_d\delta_{ij}\right] + \\ + \mu\left(\frac{\partial^2 u_{dj}}{\partial x_j\partial x_i} + \frac{\partial^2 u_{di}}{\partial x_j^2}\right) + \frac{\partial\mu}{\partial x_j}\left(\frac{\partial u_{dj}}{\partial x_i} + \frac{\partial u_{di}}{\partial x_j}\right) + \frac{\partial\lambda}{\partial x_i}\frac{\partial u_{dj}}{\partial x_j} + \lambda\frac{\partial^2 u_{dj}}{\partial x_i\partial x_j} \tag{6}$$

with summation over $j = (1,\ 2)$ at a fixed i designated as $1,\ 2$. The first-order disturbance equation for total energy is

$$\frac{\partial E_d}{\partial t} = -\frac{\partial}{\partial x_i}\left[\left(\overline{E} + \overline{P}\right)u_{di} + (E_d + P_d)\,\overline{u}_i\right] + \overline{\rho}\dot{Q}_d + \rho_d\dot{\overline{Q}} + \frac{\partial k}{\partial x_i}\frac{\partial T_d}{\partial x_i} + k\frac{\partial^2 T_d}{\partial x_i^2} + \\ + \frac{\partial}{\partial x_i}\left(\overline{\tau}_{ij}u_{dj}\right) + \frac{\partial\overline{u}_j}{\partial x_i}\left\{\lambda\frac{\partial u_{dk}}{\partial x_k}\delta_{ij} + \mu\left(\frac{\partial u_{dj}}{\partial x_i} + \frac{\partial u_{di}}{\partial x_j}\right)\right\} + \overline{u}_j\left\{\frac{\partial\lambda}{\partial x_j}\frac{\partial u_{dk}}{\partial x_k} + \lambda\frac{\partial^2 u_{dk}}{\partial x_i\partial x_k}\right\} + \\ + \frac{\partial\mu}{\partial x_i}\left(\frac{\partial u_{dj}}{\partial x_i} + \frac{\partial u_{di}}{\partial x_j}\right) + \mu\left(\frac{\partial^2 u_{dj}}{\partial x_j^2} + \frac{\partial^2 u_{di}}{\partial x_i\partial x_j}\right) \tag{7}$$

summed over $i,\ j,\ k = (1,\ 2)$. In this paper, nominal z-direction flow is zero ($\overline{W} = 0$) and derivatives of nominal variables with respect to the spanwise direction, z, are also zero $\left(\frac{\partial(\,)}{\partial z} = 0\right)$.

From the ideal gas law, the disturbance pressure P_d is defined as

$$P_d = (\gamma-1)E_d - (\gamma-1)\left[\left(\frac{\overline{U}^2 + \overline{V}^2}{2}\right)\rho_d + \overline{\rho}\overline{U}u_d + \overline{\rho}\overline{V}v_d\right], \tag{8}$$

and disturbance temperature T_d is defined as

$$T_d = \frac{1}{\overline{\rho}C_v}\left(E_d - \rho_d\left(C_v\overline{T} + \frac{\overline{U}^2 + \overline{V}^2}{2}\right) - \overline{\rho}\overline{U}u_d - \overline{\rho}\overline{V}v_d\right). \tag{9}$$

Eqs. (5), (6), and (7) are all linear partial differential equations (PDEs) with respect to disturbance variables. However, they are formidable because the disturbance terms have variable coefficients [3]. Using finite differences, the disturbance equations can be converted into a form suitable for a linear state space representation. Explicit derivatives are shown in Eqs. (5), (6), and (7) to make it easier to apply the finite difference stencils shown in Fig. 4.

IV. Domain Mapping and Grid Generation

For non-orthogonal geometries and flows arising from grid transformations such as the Illingworth-transformation, domain mapping topologically orthogonalizes a geometry, while retaining its physical information. It conducts all numerical differentiation within an orthogonal, uniformly, spaced grid called the computational domain and maps these derivatives to a physical grid, via the chain rule. In the computational domain, finite differences are performed on physical quantities with respect to natural coordinates, ξ and η. In this manner, the full accuracies of the finite difference formulas are retained [11]. For the flat-plate grid shown in Fig. 2, the transformation formulas are listed as

$$x = L\xi \quad \text{and} \quad y = y_u\left[\frac{(b+1) - (b-1)\left(\frac{b+1}{b-1}\right)^{1-\eta}}{1 + \left(\frac{b+1}{b-1}\right)^{1-\eta}}\right], \tag{10}$$

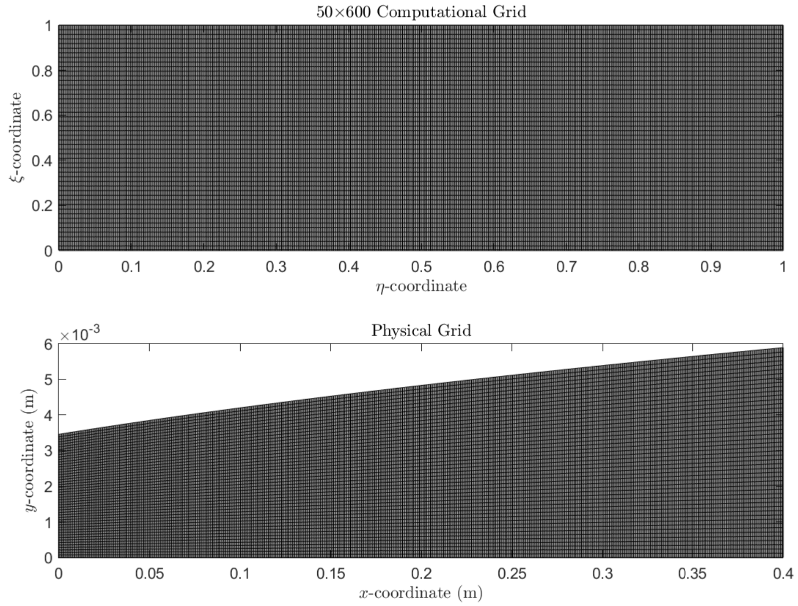


Fig. 2 Typical Computational and Physical Grids for Flat Plate with $b = 10$ and $L_{off} = 0.21$ m. Mesh sizes considered were 50×600, 100×1200, and 100×2400.

which has an inverse transformation of

$$\xi = x/L \quad \text{and} \quad \eta = 1 - \frac{\left[1 - \ln\left(\dfrac{b + 1 - \dfrac{y}{y_u}}{b - 1 + \dfrac{y}{y_u}}\right)\right]}{\ln\left(\dfrac{b+1}{b-1}\right)}. \tag{11}$$

The computational coordinates ξ and η range between 0 and 1. The upper bound of the domain, y_u, scales a wall clustering function found in Anderson *et al.* [11]. The parameter b controls the degree of clustering in the y-direction. As b approaches 1, grid clustering increases. Eq. (10) can be used to describe general 2D channel flow. The inverse mapping, Eq. (11) provides the much-desired derivatives of natural coordinates (ξ, η) with respect to physical coordinates (x, y). For a flat plate, a suitable choice for the upper domain boundary is

$$y_u = \frac{M_\infty^2 \sqrt{\mu_\infty}}{\sqrt{\rho_\infty U_\infty}} \sqrt{x + L_{off}}, \tag{12}$$

where M_∞ is the freestream Mach number, μ_∞ is the freestream dynamic viscosity, ρ_∞ is the freestream density, U_∞ is the freestream velocity, and L_{off} is an offset parameter. $L_{off} > 0$ guarantees a finite derivative at $x = 0$.

Grid transformation formulas for the differentiation of a general, scalar field function $\phi = \phi(x, y, z, t)$ are listed below. Going from the uniform two-dimensional grid in ξ and η to the physical grid in x and y, first order derivatives are

$$\frac{\partial \phi}{\partial x_i} = \frac{\partial \phi}{\partial \xi}\frac{\partial \xi}{\partial x_i} + \frac{\partial \phi}{\partial \eta}\frac{\partial \eta}{\partial x_i}, \tag{13}$$

where $i = 1, 2$; and second order derivatives are

$$\begin{aligned}\frac{\partial^2\phi}{\partial x_i\partial x_j} =&\frac{\partial^2\phi}{\partial\xi^2}\left(\frac{\partial\xi}{\partial x_i}\frac{\partial\xi}{\partial x_j}\right)+\frac{\partial^2\phi}{\partial\eta^2}\left(\frac{\partial\eta}{\partial x_i}\frac{\partial\eta}{\partial x_j}\right)+\frac{\partial^2\phi}{\partial\xi\partial\eta}\left(\frac{\partial\eta}{\partial x_i}\frac{\partial\xi}{\partial x_j}+\frac{\partial\eta}{\partial x_j}\frac{\partial\xi}{\partial x_i}\right)+\\ &\left(\frac{\partial^2\xi}{\partial x_i\partial x_j}\right)\frac{\partial\phi}{\partial\xi}+\left(\frac{\partial^2\eta}{\partial x_i\partial x_j}\right)\frac{\partial\phi}{\partial\eta},\end{aligned} \tag{14}$$

where $i = 1, 2$ and $j = 1, 2$. Here, subscripts do not imply summation. Using Eqs. (13) and (14), many different canonical 2D geometries can be accommodated. Numerical discretization is achieved through the application of finite difference formulas in the computational domain. On the boundaries of the domain, second-order accurate finite-differencing is used with the corresponding stencil shown in the bottom of Fig. 4. For the domain interior, fourth-order finite differencing is used. The stencil for fourth-order-accurate numerical discretization is shown in the top of Fig. 4. Finite differencing formulas may be obtained from Anderson *et al.* [11], and fourth-order differencing formulas can be obtained via a finite differencing calculator [15, 16].

V. Linear State Space Formulation

The state space representation has a state equation,

$$\dot{\vec{q}} = A\vec{q} + B\vec{u}, \tag{15}$$

and an output equation,

$$\vec{y} = C\vec{q} + D\vec{u}. \tag{16}$$

Matrices A, B, C, D are respectively called the state matrix, input matrix, output matrix, and feedthrough matrix [4, 17]. In this paper, $B = A$ for full-state disturbance, $C = I$ for full-state measurement, and $D = 0$ for zero direct transmission. Vectors $\vec{q}$, $\vec{u}$, and $\vec{y}$ are respectively the disturbed state vector, the disturbance input vector, and the measurement vector. Matrix A is also identified as the Jacobian matrix, containing the linear dynamics of the disturbance equations to second-order accuracy on the boundaries and fourth-order accuracy in the interior. By applying the Laplace transform on Eqs. 15–16 and setting the complex variable $s = j\omega$, the state space model is expressed in frequency space as

$$\begin{aligned}\vec{q}(\omega) &= C(j\omega I - A)^{\dagger}B\vec{u}(\omega)\\ &= \mathcal{H}\vec{u}(\omega),\end{aligned} \tag{17}$$

where $\mathcal{H}$ is the state space transfer function. For a system of first-order ordinary differential equations (ODEs) denoted as $\frac{\partial\vec{q}}{\partial t} = \vec{f}(\vec{q})$, such as those formed by finite-differenced versions of Eqs. (5–7), matrix A is defined as

$$A = \frac{\partial\vec{f}}{\partial\vec{q}}\ , \text{ whose elements are } A_{ps} = \frac{\partial f_p}{\partial q_s}. \tag{18}$$

Matrix A also incorporates boundary conditions (BC), which are Dirichlet for perturbed velocity components and Neumann for perturbed density, total energy, pressure, and temperature quantities. Matrix A is composed of pointwise/elemental Jacobian matrixes, $A_{ij}^{(e)}$ for each point ij: second-order accurate $A_{ij}^{(e)_2}$ for the domain boundaries and fourth-order accurate $A_{ij}^{(e)_4}$ for the domain interior. This is done for both numerical stability and simplicity.

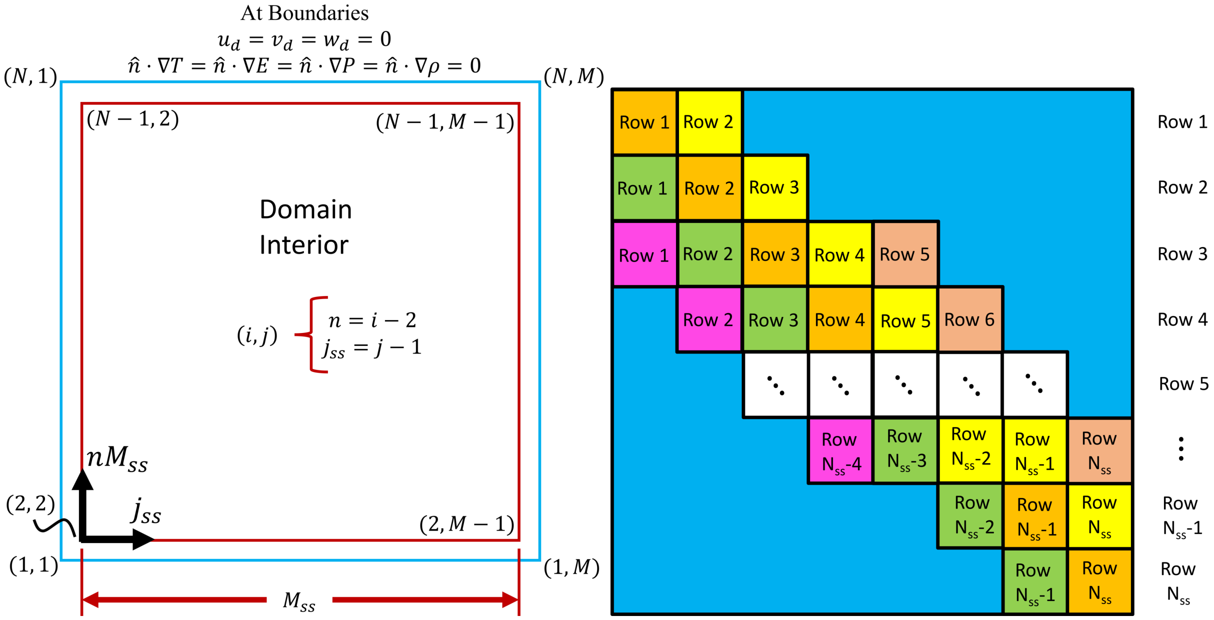
Using the top stencil shown in Fig. 4 for the domain interior, the fourth-order accurate pointwise Jacobian is

$$A_{ij}^{(e)_4} = \begin{bmatrix} A_1 & A_2 & A_3 & \dots & A_{24} & A_{25} \end{bmatrix}, \tag{19}$$

where the subscript represents a point in the stencil. Each submatrix A_k represents the contribution of point k to $A_{ij}^{(e)}$. Mathematically, this is expressed as

$$A_k = \frac{\partial\vec{f}_{ij}}{\partial\vec{q}_k}, \tag{20}$$

where $\vec{f}_{ij}$ is the finite differenced NS equation at point ij and $\vec{q}_k$ is the set of conserved quantities at point k. These matrices are conveniently obtained via the use of MATLAB's Symbolic Toolbox [18], using the partial differentiation



(a) Index Transformation of a 2D Domain with BC's.

(b) Sparse layout of Jacobian Matrix A.

Fig. 3 Vectorization of 2D matrix representing 2D domain. (a) Boundary conditions for a 2D domain and index transformation for vectorization from i, j to n, j_{ss}. (b) Resulting sparseness of Jacobian Matrix A due to vectorization-transformation in Fig. 3a.

Index Transformation for Fourth-Order Differencing Stencil

$$k = \begin{array}{|c|c|c|c|c|}\hline 21 & 22 & 23 & 24 & 25 \\ \hline 16 & 17 & 18 & 19 & 20 \\ \hline 11 & 12 & 13 & 14 & 15 \\ \hline 6 & 7 & 8 & 9 & 10 \\ \hline 1 & 2 & 3 & 4 & 5 \\ \hline \end{array}$$

$$S_{ij} = \begin{array}{|c|c|c|c|c|}\hline (n+2)M_{ss}+j_{ss}-2 & (n+2)M_{ss}+j_{ss}-1 & (n+2)M_{ss}+j_{ss} & (n+2)M_{ss}+j_{ss}+1 & (n+2)M_{ss}+j_{ss}+2 \\ \hline (n+1)M_{ss}+j_{ss}-2 & (n+1)M_{ss}+j_{ss}-1 & (n+1)M_{ss}+j_{ss} & (n+1)M_{ss}+j_{ss}+1 & (n+1)M_{ss}+j_{ss}+2 \\ \hline nM_{ss}+j_{ss}-2 & nM_{ss}+j_{ss}-1 & nM_{ss}+j_{ss} & nM_{ss}+j_{ss}+1 & nM_{ss}+j_{ss}+2 \\ \hline (n-1)M_{ss}+j_{ss}-2 & (n-1)M_{ss}+j_{ss}-1 & (n-1)M_{ss}+j_{ss} & (n-1)M_{ss}+j_{ss}+1 & (n-1)M_{ss}+j_{ss}+2 \\ \hline (n-2)M_{ss}+j_{ss}-2 & (n-2)M_{ss}+j_{ss}-1 & (n-2)M_{ss}+j_{ss} & (n-2)M_{ss}+j_{ss}+1 & (n-2)M_{ss}+j_{ss}+2 \\ \hline \end{array}$$

$$slice_k = (c_l S_{ijk} + 1) : 1 : c_l S_{ijk}$$

Index Transformation for Second-Order Differencing Stencil

$$k = \begin{array}{|c|c|c|}\hline 7 & 8 & 9 \\ \hline 4 & 5 & 6 \\ \hline 1 & 2 & 3 \\ \hline \end{array}$$

$$S_{ij} = \begin{array}{|c|c|c|}\hline (n-1)M_{ss}+j_{ss}-1 & (n+1)M_{ss}+j_{ss} & (n-1)M_{ss}+j_{ss}+1 \\ \hline nM_{ss}+j_{ss}-1 & nM_{ss}+j_{ss} & nM_{ss}+j_{ss}+1 \\ \hline (n-1)M_{ss}+j_{ss}-1 & (n-1)M_{ss}+j_{ss} & (n-1)M_{ss}+j_{ss}+1 \\ \hline \end{array}$$

$$slice_k = (c_l S_{ijk} + 1) : 1 : c_l S_{ijk}$$

Fig. 4 Indexed slices of State Vector $\vec{q}$ defined using the Index transformation from i, j to n, j_{ss}. Index k refers to a point in the differencing stencil. These slices are critical in the construction of the Jacobian matrix A. Slices are in MATLAB notation. The parameter c_l represents the number of state space variables (perturbation variables) per point (i, j).

described in Eq. (18). For the domain boundaries, the stencil shown in the bottom of Fig. 4 is used for the second-order accurate pointwise Jacobian:

$$A_{ij}^{(e)_2} = \begin{bmatrix} A_1 & A_2 & A_3 & A_4 & A_5 & A_6 & A_7 & A_8 & A_9 \end{bmatrix}. \tag{21}$$

On the boundaries of the interior domain, boundary conditions are applied on individual submatrices A_k. BC submatrices $A_{BC,k}$ are formed from the submatrices, A_k, of domain boundary points and are added to the submatrices of adjacent points. BC submatrices zero out perturbed velocity contributions on the boundaries, enforcing Dirichlet conditions, and they assume equality between perturbed density, pressure, temperature, and energy variables on the boundaries with points on the edges of the interior domain, resulting in Neumann conditions. Using MATLAB notation [19], a typical BC submatrix $A_{BC,k}$ is written as a concatenation: $A_{BC,k} = [A_k(:,1), \mathrm{zeros}(4,2), A_k(:,4)]$.

In order to program the Jacobian in a computer script, an index transformation is necessary to vectorize the interior of the computational domain, as shown in Fig. 3a. The interior of an $N \times M$ computational domain is $(N_{ss} = N - 2) \times (M_{ss} = M - 2)$. The indices i, j denoting row and column transform into $j_{ss} = j - 1$ and $n = i - 2$, where $0 \leq n \leq N - 1$ and $1 \leq j_{ss} \leq M_{ss}$. Slices of the computational domain matrix, which are listed in Fig. 4 map the elemental pointwise Jacobians to locations in the global Jacobian matrix. As a result of the aforementioned mapping, Fig. 3b shows that the Jacobian matrix is sparse and ill-conditioned (especially for sizes greater than $\left((M_{ss} \cdot N_{ss} \cdot c_l)^2 = 3000^2\right)$, requiring special storage like MATLAB's sparse data type, and tight tolerance on linear algebra solvers, such as *lsqminnorm* [20], for repeatable, grid-independent solutions.

VI. Optimal Controls Problem

Following Dwivedi *et al.* [2], the Chu energy norm is evaluated as

$$\begin{aligned} E_{Chu} &= \int_V \left(\frac{1}{2}\overline{\rho}\left(u_d^2 + v_d^2\right) + \frac{\overline{P}}{2}\left(\frac{\rho_d}{\overline{\rho}}\right)^2 + \frac{(C_v \overline{T})}{2}\left(\frac{T_d}{\overline{T}}\right)^2 \right) dV \\ &= \int_V \left(\frac{1}{2}\overline{\rho}\left(u_d^2 + v_d^2\right) + \frac{\overline{P}}{2}\left(\frac{\rho_d}{\overline{\rho}}\right)^2 + \frac{(C_v \overline{T})}{2}\left(\frac{T_d}{\overline{T}}\right)^2 \right) |J| d\xi d\eta \\ &= \int_V \left(\frac{1}{2}\overline{\rho}\left(u_d^2 + v_d^2\right) + \frac{\overline{P}}{2}\left(\frac{\rho_d}{\overline{\rho}}\right)^2 + \frac{(C_v \overline{T})}{2}\left(\frac{T_d}{\overline{T}}\right)^2 \right) \frac{d\xi d\eta}{\left|\left(\frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial y} - \frac{\partial \xi}{\partial y}\frac{\partial \eta}{\partial x}\right)\right|} \\ &= \frac{1}{2}(C\vec{q})^H N^H \mathcal{M} N C \vec{q}, \end{aligned} \tag{22}$$

where $|J|$ is the determinant of the geometric Jacobian, $J = \dfrac{\partial(x,y)}{\partial(\xi,\eta)}$*, locally evaluated at a discrete point ij as

$$|J_{ij}| = \frac{1}{\left|\left(\frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial y} - \frac{\partial \xi}{\partial y}\frac{\partial \eta}{\partial x}\right)\right|_{ij}}; \tag{23}$$

*$|J|$ is evaluated as $1/|J^{-1}|$. Since $JJ^{-1} = I$, $|J||J^{-1}| = 1$, and thus, $|J| = 1/|J^{-1}|$.

and the pointwise decoding matrix N_{ij} and weighting matrix $\mathcal{M}_{ij}$ for a 3D disturbance analysis are respectively

$$N_{ij} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \dfrac{1}{\overline{\rho}} & 0 & 0 & 0 \\ 0 & 0 & \dfrac{1}{\overline{\rho}} & 0 & 0 \\ 0 & 0 & 0 & \dfrac{1}{\overline{\rho}} & 0 \\ \left[-\left(\dfrac{\overline{U}^2+\overline{V}^2}{2\overline{\rho}C_v}\right)-\dfrac{\overline{T}}{\overline{\rho}}\right] & \dfrac{-\overline{U}}{C_v\overline{T}} & \dfrac{-\overline{V}}{C_v\overline{T}} & 0 & \dfrac{1}{C_v\overline{T}} \end{bmatrix} \tag{24}$$

and

$$\mathcal{M}_{ij} = |J_{ij}| \begin{bmatrix} \dfrac{\overline{P}}{\overline{\rho}^2} & 0 & 0 & 0 & 0 \\ 0 & \overline{\rho} & 0 & 0 & 0 \\ 0 & 0 & \overline{\rho} & 0 & 0 \\ 0 & 0 & 0 & \overline{\rho} & 0 \\ 0 & 0 & 0 & 0 & \dfrac{\left(C_v\overline{T}\right)}{\overline{T}^2} \end{bmatrix}. \tag{25}$$

The maximization of the square of the energy-normed gain of a flow system,

$$G^2 = \frac{(C\vec{q})^H P(C\vec{q})}{(\vec{u})^H P\vec{u}}. \tag{26}$$

is subject to the linear state space model Eq. (15) of the discretized, compressible flow disturbance equations. This is mathematically stated as the minimization of the Lagrangian $\mathcal{L}$ [21], or more specifically, the reciprocal square gain $G^{-2} = \dfrac{(\vec{u})^H P\vec{u}}{(C\vec{q})^H P(C\vec{q})}$ subject to linear flow system dynamics:

$$\mathcal{L} = \frac{(\vec{u})^H P\vec{u}}{(C\vec{q})^H P(C\vec{q})} - \vec{\Lambda}^H((j\omega I - A)\vec{q} - B\vec{u}), \tag{27}$$

where Λ is a Lagrangian multiplier and $P = N^H \mathcal{M} N$. Similar to Dwivedi et al. [2], we will now prove that the singular value decomposition (SVD) maximizes the Chu-normed amplification of a disturbance vector, and in the process, we derive equations to obtain the SVD. Differentiating the Lagrangian $\mathcal{L}$ and transposing yields three equations:[†]

$$\frac{\partial\mathcal{L}}{\partial\vec{\Lambda}} = 0 = ((j\omega I - A)\vec{q} - B\vec{u}) \tag{28}$$

$$\frac{\partial\mathcal{L}}{\partial\vec{u}} = 0 = \frac{2P\vec{u}}{(C\vec{q})^H P(C\vec{q})} + B^H\vec{\Lambda} \tag{29}$$

$$\frac{\partial\mathcal{L}}{\partial\vec{q}} = 0 = -\frac{(\vec{u})^H P\vec{u}}{\left[(C\vec{q})^H P(C\vec{q})\right]^2} 2C^H PC\vec{q} - (j\omega I - A)^H\vec{\Lambda}. \tag{30}$$

[†] Differentiation is conducted in a manner of Murray [22] in his derivation of a linear quadratic regulator controller.

Inserting Eqs. (30) and (28) into Eq. (29) yields

$$\begin{aligned}
P\vec{u} &= -\frac{1}{2}(C\vec{q})^H P(C\vec{q}) B^H \vec{\Lambda} = -\frac{1}{2}(C\vec{q})^H P(C\vec{q}) B^H (j\omega I - A)^{-H} \left(\frac{-2G^{-2}}{(C\vec{q})^H P(C\vec{q})} C^H P C\vec{q}\right) \\
&= B^H (j\omega I - A)^{-H} G^{-2} C^H P C\vec{q} \\
&= B^H (j\omega I - A)^{-H} G^{-2} C^H P \,\mathcal{H}\vec{u} \\
&= G^{-2} \left[C(j\omega I - A)^{\dagger} B\right]^H P \,\mathcal{H}\vec{u} \\
&= G^{-2} \mathcal{H}^H P \,\mathcal{H}\vec{u}
\end{aligned} \tag{31}$$

A dagger denotes the generalized inverse (Moore-Penrose Inverse). Multiplying both sides of Eq. (31) by $\vec{u}^H$ recovers the definition of the inverse square gain function G^{-2}, noting that $\vec{q} = H\vec{u}$:

$$G^{-2} = \frac{(\vec{u})^H P\vec{u}}{(C\vec{q})^H P(C\vec{q})}. \tag{32}$$

The system of equations, Eqs. (28-30), are degenerate and loop about the definition of the gain function G:

$$G^2 = \frac{(C\vec{q})^H P(C\vec{q})}{(\vec{u})^H P\vec{u}}. \tag{33}$$

Therefore, only an optimal disturbance direction and an optimal output direction can be determined. Optimal directions and gains can be obtained either via singular value decomposition or an equivalent least squares minimization. According to Skogestad *et al.* [4], the SVD determines the optimal input direction to maximize an amplification gain defined as the ratio of two 2-norms. The solution of Eqs. 30 constitute the SVD of matrix $\mathcal{M}^{1/2}NH = U\Sigma V^H$. Their solution requires the solution to linear systems of equations. Due to the linearly dependent, rank deficient, and poorly conditioned nature of the fluid system, the generalized inverse (denoted by the dagger symbol) is used in both the SVD proof and the SVD solving algorithm below. The generalized inverse is yet another SVD / least squares operation [23]. It does not impose strict boundary conditions. The generalized inverse optimally attempts to satisfy boundary conditions by minimizing residuals on boundary points. Hence, the generalized inverse will not artificially stabilize a system in order to meet a boundary condition that the system refuses. Instead, it will capture an unattenuated fluid instability, like Mack's second mode. In MATLAB, the generalized inverse is best implemented as a least squares operation via the command *lsqminnorm* [20].

To determine the singular value decomposition of large, sparse matrices, a slightly modified version of the algorithm from Dwivedi *et al.* [2] is used. It is as follows

1) Guess $\vec{u}_{opt}$ as $\vec{u}^{(n)}$. Let $n = 1$. A suitable guess could a vector consisting of ones.
2) Normalize $\vec{u}^{(n)}$.
3) Evaluate $\vec{q}^{(n)} = (j\omega I - A)^{\dagger} B\vec{u}^{(n)} = \text{lsqminnorm}(j\omega I - A, B\vec{u}^{(n)}, \text{tol})$.
4) Normalize $\vec{q}^{(n)}$.
5) Evaluate $\left(G^{-2}\right)^{(n)} = \dfrac{(\vec{u}^{(n)})^H P\vec{u}^{(n)}}{(C\vec{q}^{(n)})^H P(C\vec{q}^{(n)})}$.
6) Evaluate $\Lambda^{(n)} = ((j\omega I - A)^H)^{\dagger} \dfrac{\left(G^2\right)^{(n)}}{(C\vec{q}^{(n)})^H P(C\vec{q}^{(n)})} 2C^H PC\vec{q}^{(n)}$

   or $\Lambda^{(n)} = \text{lsqminnorm}\left((j\omega I - A)^H, \dfrac{\left(G^2\right)^{(n)}}{(C\vec{q}^{(n)})^H P(C\vec{q}^{(n)})} 2C^H PC\vec{q}^{(n)}, \text{tol}\right)$.
7) Evaluate $\vec{u}^{(n+1)} = -\dfrac{(C\vec{q}^{(n)})^H P(C\vec{q}^{(n)})}{2} P^{\dagger} B^H \Lambda^{(n)} = -\dfrac{(C\vec{q}^{(n)})^H P(C\vec{q}^{(n)})}{2}\ \text{lsqminnorm}(P, B^H \Lambda^{(n)}, \text{tol})$.
8) Set $n = n + 1$. Return to Step 2 and repeat.

The tolerance, tol, is specified to be 1×10^{-200}. At least 5−10 iterations are required for convergence. Due to its similarity with power iteration for matrix eignvalues, the numerical algorithm is henceforth dubbed SVD power iteration.

VII. Results

The base flow, as shown in Fig. 5, is obtained from a similarity solution, using the Illingworth transformation of the compressible boundary layer equations with flow conditions listed in Fig. 1. This base flow is linearly interpolated to a mapped nonorthogonal grid, described by Eqs. 10–12 and shown in Fig. 6. This grid is then used in the calculation of the Jacobian, which stores the linear dynamics of the NS equations. Interpolation eliminates the singularity of the Illingworth transformation at $x = 0$.

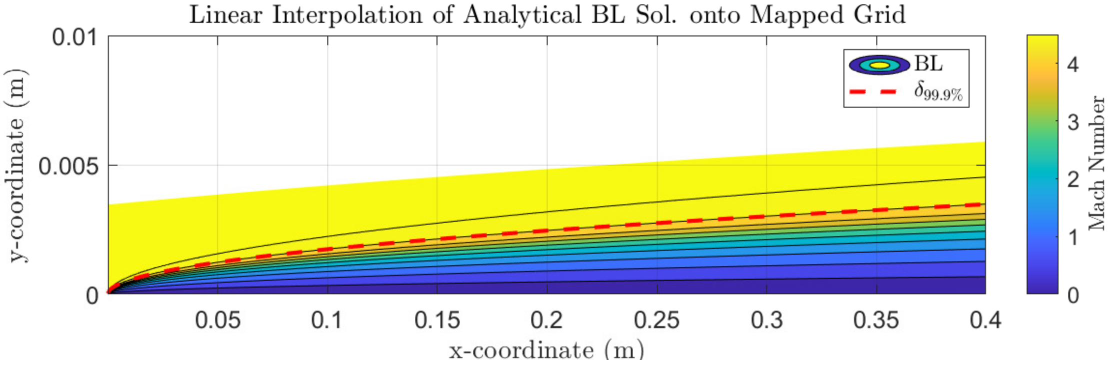


Fig. 5 Flat Plate Base Flow for Input-Output Analysis with a 100×2400 **nonorthogonal grid.**

Input-output analysis was conducted at two forcing frequencies ω, corresponding to nondimensional forcing frequencies $F = 1.6 \times 10^{-4}$ and $F = 2.2 \times 10^{-4}$, where $F = \dfrac{\omega \mu_\infty}{\rho_\infty U_\infty^2}$. These nondimensional frequencies are used in Dwivedi *et al.* [2] as means to determine whether a stability code can accurately predict mode shapes at $F = 1.6 \times 10^{-4}$ and $F = 2.2 \times 10^{-4}$. In Figs. 6, the optimal output directions for temperature perturbation at maximum amplification are obtained via input-output analysis for a 100×2400 nonorthogonal grid and are mapped back to the grid for a physical, spatial picture that features disturbance locations. In Figs. 7 and 8, the optimal output direction for near-wall temperature disturbance is shown with respect to x-coordinate at $F = 1.6 \times 10^{-4}$ and $F = 2.2 \times 10^{-4}$. The optimal output direction for near-wall temperature disturbance and nondimensional growth rates overlap in nondimensional circular frequency RF, where the nondimensional streamwise coordinate is $R = \sqrt{Re_\infty x}$. Growth rates evaluated via the following formula:

$$\alpha_{T d} dR = \frac{\partial [\ln(T_d)]}{\partial x} \frac{\sqrt{x}}{\sqrt{Re_\infty}} dx^*, \tag{34}$$

but it should be noted that $\frac{dx}{dR} = \frac{2\sqrt{x}}{\sqrt{Re_\infty}}$, so $dx \neq L^* dx^*$ but rather $dx = 2L^* dx^*$, where $L^* = \sqrt{\dfrac{x}{Re_\infty}}$ and dx^* is a nondimensional variable. If however we let $dx = L^* dx^*$, $dR^* = 0.5 dx^*$, and this is what we use to be consistent with Ma *et al.* [1] and Dwivedi *et al.* [2].

Grid independence of mode shapes with respect to grid size and discretization is displayed in Fig. 9. The offset parameter L_{off} was varied to observe dependence on grid height on convergence. The purpose of the grid independence study was to determine the reliability of the iterative SVD-solver algorithm in Section VI, considering the large dimensions of the sparse matrices involved.

VIII. Discussion

From Figs. 6 and 7, the locations of second mode, flat-plate instabilities can be deduced and match well with the results of Ma *et al.* [1] and Dwivedi *et al.* [2]. In bottom figure of Fig. 7, the optimal output directions at maximum amplification for temperature disturbance at both $F = 1.6 \times 10^{-4}$ and $F = 2.2 \times 10^{-4}$ overlap when plotted over the nondimensional coordinate RF. A peak disturbance occurs at $RF = 0.2355$. This closely agrees with the physical behavior predicted by Ma *et al.* [1] that the peak temperature perturbation should occur at a nondimensional frequency of $RF = 0.2324$. From the middle plot of Fig. 7, the disturbance is not attenuated at $x = 0.4$ m, indicating a thermoacoustic instability for the nondimensional forcing frequency of $F = 1.6 \times 10^{-4}$ akin to that studied by Kuehl [6] and Ma *et al.*

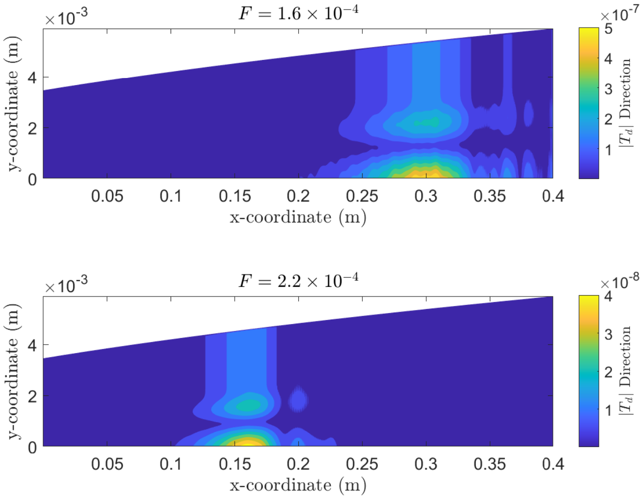


Fig. 6 Output direction for Complex Norm of Temperature perturbation at maximum amplification mapped to a 100×2400 nonorthogonal grid. (*Top*) $F = 1.6 \times 10^{-4}$. (*Bottom*) $F = 2.2 \times 10^{-4}$.

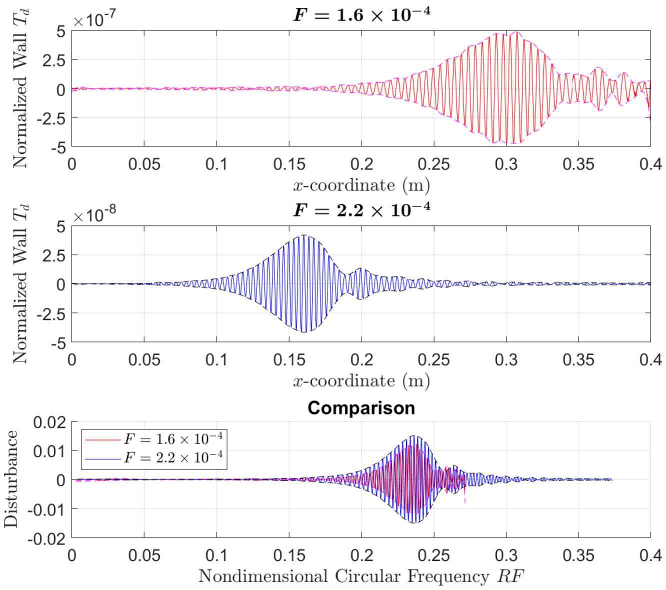


Fig. 7 Optimal Output Directions for Near-wall Perturbation Quantities over Length of Flat Plate for Two Disturbance Frequencies ($F = 2.2 \times 10^{-4}$ (*Top*) and $F = 1.6 \times 10^{-4}$ (*Middle*)), using a Nonorthogonal Grid Discretization of 100×2400. Of interest is the spatial growth of the temperature disturbance T_d with respect to streamwise coordinate x. (*Bottom*) Overlap of Wall Temperature Perturbation at both disturbance frequencies when plotted over nondimensional circular frequency RF, which in this case acts as a nondimensional x-coordinate.

[1]. The flat plate needs to be longer to attenuate this instability. The use of the least-squares, linear-equation solver for a rank-deficient linear system avoided the premature enforcement of Dirichlet and Neumann BCs at the right-hand boundary, thus producing a more physical result. It effectively applied an optimal outlet condition that best satisfied the

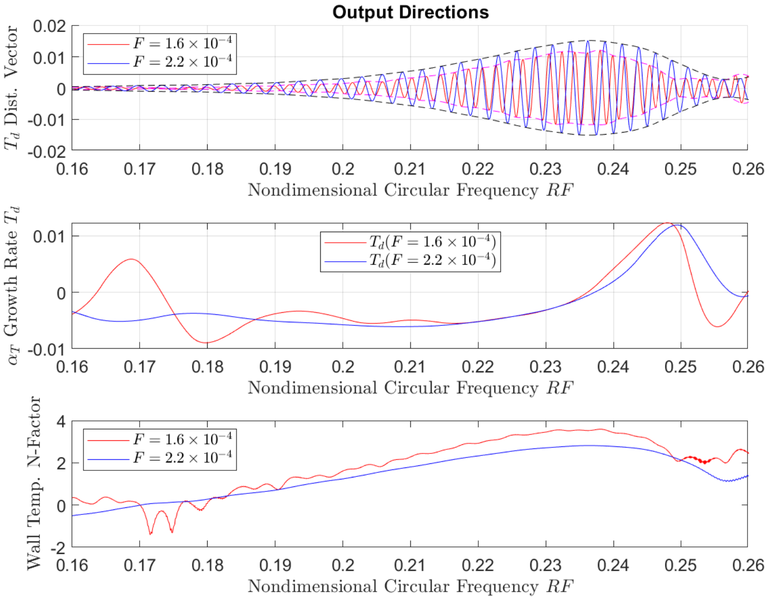


Fig. 8 (*Top*) Optimal Temperature Perturbation Output Direction vs. Nondimensional Circular Frequency with a 100×2400 grid. (*Middle*) Nondimensional Temperature Growth Rate vs. Nondimensional Circular Frequency. (*Bottom*) N-factor vs. Nondimensional Circular Frequency.

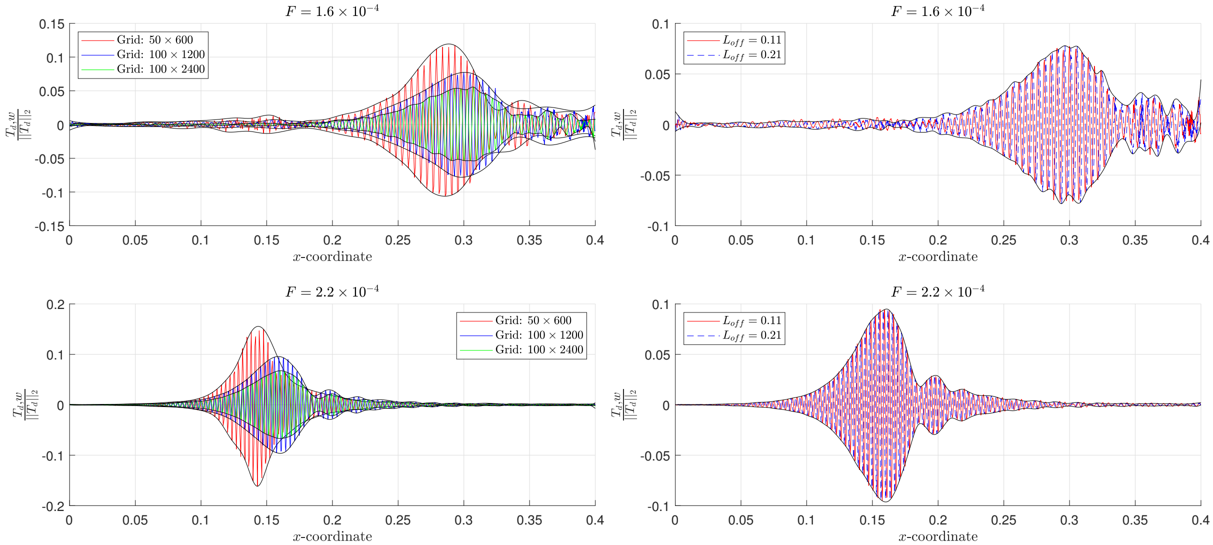


Fig. 9 Grid Independence of Input-Output Analysis Results, embodied via Optimal Wall Temperature Perturbation Directions at $F = 1.6 \times 10^{-4}$ and $F = 2.2 \times 10^{-4}$. (*Left*) Variation of grid sizing. Note that results have different normalization, mainly due to the size of the statespace vector, and therefore convergence in mode shape is important. (*Right*) Variation of Upper Bound Geometry Parameter L_{off}.

governing equations and maximized the energy-weighted input-to-output amplifcation gain of the flow system. The occurrence of an unattenuated second-mode instability at $F = 1.6 \times 10^{-4}$ for $L = 0.4$ m further validated the use of the generalized inverse in the numerical SVD algorithm.

The shapes of the wall temperature perturbation vector also reflects that of Ma *et al.* [1], which gives further confidence in our work. In Fig. 8, temperature perturbation growth rates are displayed in agreement with Ma *et al.* [1], and the shape of the temperature growth rates agree with those of Dwivedi *et al.* [2] at both frequencies. N-factors are also displayed for both frequency cases, showing that the $F = 2.2 \times 10^{-4}$ case is slightly more stable. Instability is shown to start in the vicinity of $RF = 0.17$, which corresponds to $x = 0.15$ m for $F = 1.6 \times 10^{-4}$ and $x = 0.083$ m for $F = 2.2 \times 10^{-4}$. Low N-factors of approximately 2 indicate that the flow is not transitional, but for a short plate, second mode disturbances do not die away for $F = 1.6 \times 10^{-4}$. This can inform experimental design about the flat plate length necessary to capture a mode shape at a hypersonic flow condition.

In Fig. 9, a grid convergence study was carried out to (1) give confidence in the presented results for grid 100 × 2400, (2) confirm the stability of the SVD solver algorithm, considering the large number of sparse matrices involved, and (3) present another avenue for further research. The offset parameter L_{off}, which largely determines the height of the upper domain boundary y_u, is important. The authors observed that for sufficiently large y_u, the solution converges to results featured in Fig. 7. For too small a y_u, the stability of the Input-Output analysis breaks down entirely, since the domain is not large enough. The Neumann and Dirichlet boundary conditions on the upper boundary behave as far-field conditions and should therefore be far enough away from the flat wall to guarantee physical results. Therefore, context for grid and domain sizing is obtained, along with their computation times.

Table 1 Computation Times and Memory Requirements with respect to grid size and nondimensional forcing frequency F for Input-Output Analysis in the MATLAB programming environment. The time for the assembly of the analytical Jacobian is t_A, and solution times are indicated by t_s. RAM stands for read-access-memory.

Grid	Size of A	Iterations	t_A (s)	t_s at $F = 1.6 \times 10^{-4}$ (s)	t_s at $F = 2.2 \times 10^{-4}$ (s)	Max. RAM (GB)
50 × 600	114816^2	8	75.7	164	163	4
100 × 1200	469616^2	8	320	1600	1620	16
100 × 2400	940016^2	5	633	2840	2960	48

In Table 1, the computational speed of this paper's input-output algorithm is tabulated, which includes times for the Jacobian matrix construction and SVD power iteration. Due to the large size of the Jacobian matrix A, the sparse data type, native to MATLAB [24], is used to conserve RAM and decrease computation time while preserving accuracy. The data type stored only the indices and values of nonzero elements, which must each be supplied in vector form for efficient construction (a minor inconvenience) [25]. The sparse data type allows for quick mathematical operations and is accommodated by the linear equation solver *lsqnormmin*. In Table 1, the small Jacobian assembly times reflect the effectiveness of the sparse data type, and the reasonable solution times indicate the efficiency of the sparse least-squares-equation solver for obtaining the input-output directions at maximum amplification. For comparison, solution times for the same excitation-frequency cases and grids could amount to hours to days. Even just the construction of the A matrix of size $(100 \cdot 1200 \cdot 4)^2 = 46616^2$ results a 1643 GB matrix, necessitating the sparse datatype. Large solution times would preclude grid-independence analyses, which is why it is important to have a code that could solve the input-output SVD problem in a reasonable amount of time.

IX. Conclusions

A sparse Jacobian matrix was constructed and used in the stability analysis of a flat plate flow. The second mode shapes of Ma *et al.* [1] and Dwivedi *et al.* [2] were captured. The coding scripts for 2D flow in a generalized, domain-mapped grid were wholly written in the MATLAB programming environment and took advantage of sparse least-squares minimization to solve the rank-deficient, full-state input-output analysis, SVD problem. A proof is presented to show how the Lagrangian correctly reduces to the singular value decomposition of a matrix. Low-level details, such as finite differencing formulas and the application of boundary conditions, are provided for the construction of the Jacobian matrix from the linearized Navier-Stokes equations.

Acknowledgments

Nick Pariale and David Shekhtman were supported by the ONR Grants N00014-19-1-2523, N00014-20-1-2549, and N00014-20-1-2637 for which Eric Marineau is the Program Manager.

References

[1] Ma, Y., and Zhong, X., "Receptivity of a supersonic boundary layer over a flat plate. Part 1. Wave structures and interactions," *Journal of Fluid Mechanics*, Vol. 488, 2003, pp. 31–78. doi: 10.1017/S0022112003004786.

[2] Dwivedi, A., GS, S., Candler, G. V., and Nichols, J. W., "Input-Output Analysis of Shock Boundary Layer Interaction," *Proceedings of the AIAA Aviation Forum*, AIAA 2018-3220, Atlanta, Georgia, 2018, pp. 1–17. doi: 10.2514/6.2018-3220.

[3] White, F., *Viscous Fluid Flow*, 3rd ed., McGraw-Hill, 2006.

[4] Skogestad, S., and Postlethwaite, I., *Multivariable Feedback Control*, 2nd ed., Wiley, 2005.

[5] Cook, D. A., Knutson, A. L., Nichols, J. W., and Candler, G. V., "Matrix methods for input-output analysis of 2D and 3D hypersonic flows," *Proceedings of AIAA Scitech 2020 Forum*, AIAA 2020-1820, Orlando, FL, 2020, pp. 1–17. doi: 10.2514/6.2020-1820.

[6] Kuehl, J. J., "Thermoacoustic Interpretation of Second-Mode Instability," *AIAA Journal*, Vol. 56, No. 9, 2018, pp. 3585–3592. doi: 10.2514/1.J057015.

[7] Anderson, J. D., *Hypersonic and High Temperature Gas Dynamics*, 2nd ed., AIAA, 2011.

[8] Khalil, H. K., *Nonlinear Systems*, Pearson, Chap. 4 and 13, 2001. Print.

[9] Hameed, A., Parziale, N. J., Paquin, L., Butler, C., and Laurence, S. J., "Hypersonic Slender-Cone Boundary Layer Instability in the UMD HyperTERP Shock Tunnel," *Proceedings of AIAA SciTech 2020*, AIAA-2020-0362, Orlando, Florida, 2020, pp. 1–14. doi: 10.2514/6.2020-0362.

[10] Mustafa, M. A., Parziale, N. J., Smith, M. S., and Marineau, E. C., "Amplification and structure of streamwise-velocity fluctuations in compression-corner shock-wave/turbulent boundary-layer interactions," *Journal of Fluid Mechanics*, Vol. 863, 2019, pp. 1091–1122. doi: 10.1017/jfm.2018.1029.

[11] Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., *Computational Fluid Mechanics and Heat Transfer*, 1st ed., McGraw-Hill Book Company, 1984.

[12] Anderson, J. D., *Computational Fluid Mechanics: The Basics with Applications*, McGraw-Hill, Inc., 1995.

[13] Monin, A. S., and Yaglom, A. M., *Statistical Fluid Mechanics: Mechanics of Turbulence*, Vol. I, Dover Publications, Inc., 2007.

[14] Cerulus, N., Quintanilha Jr., H., and Theofilis, V., "Global Linear Stability Analysis of the Supersonic Flows Over a Hollow Cylinder Flare Model," *Proceedings of AIAA Scitech 2021*, AIAA 2021-0052, Virtual Event, 2021, pp. 1–14. doi: 10.2514/6.2021-0052.

[15] Taylor, C., "Finite Difference Coefficients calculator," https://web.media.mit.edu/ crtaylor/calculator.html, 2021. Accessed: 2021-06-16.

[16] Bellaard, G., "Finite difference coefficient calculator," https://bellaard.com/apps/Finite%20difference%20coefficient%20calculator/, 2021. Accessed: 2021-06-16.

[17] Ogata, K., *System Dynamics*, 4th ed., Pearson Education, Inc., 1997.

[18] MathWorks, "Get Started with Symbolic Math Toolbox," www.mathworks.com, 2020. Accessed: 2020-09-11.

[19] MathWorks, "MATLAB Operators and Special Characters," www.mathworks.com, 2020. Accessed: 2020-09-11.

[20] MathWorks, "lsqminnorm," https://www.mathworks.com/help/matlab/ref/lsqminnorm.html, 2021. Accessed: 2021-06-17.

[21] Kamien, M. I., *Dynamic Optimization: The Calculus of Variations and Optimal Control in Economics and Management*, 2nd ed., Dover Publications, Inc., 2012.

[22] Murray, R. M., *Optimazation-Based Control*, 2nd ed., California Institute of Technology, 2010.

[23] Strang, G., *Linear Algebra and Its Applications*, 4th ed., Cengage Learning, 2006.

[24] MathWorks, "sparse," https://www.mathworks.com/help/matlab/ref/sparse.html, 2021. Accessed: 2021-06-17.

[25] MathWorks, "Accessing Sparse Matrices," https://www.mathworks.com/help/matlab/math/accessing-sparse-matrices.html, 2021. Accessed: 2021-06-18.