NASA/TM-2004-213233

Langley Stability and Transition Analysis Code (LASTRAC) Version 1.2 User Manual

Chau-Lyan Chang
Langley Research Center, Hampton, Virginia

June 2004

NASA/TM-2004-213233

Langley Stability and Transition Analysis Code (LASTRAC) Version 1.2 User Manual

Chau-Lyan Chang
Langley Research Center, Hampton, Virginia

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

June 2004

Available from:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 605-6000

Contents

List of Figures

Abstract

Langley Stability and Transition Analysis Code (LASTRAC) is a general-purposed, physics-based transition prediction code released by NASA for Laminar Flow Control studies and transition research. While significant progress has been made in the past decades concerning instability wave evolution and transition flow physics both computationally and experimentally, transition prediction remains a nontrivial task for engineers due to the lack of a widely available, robust, and efficient prediction tool. The design and development of the LASTRAC code is aimed at providing one such engineering tool that is easy to use and yet capable of dealing with a broad range of transition related issues. LASTRAC was written from scratch based on the state-of-the-art numerical methods for stability analysis and modern software technologies. At low fidelity, it allows users to perform linear stability analysis and N-factor transition correlation for a broad range of flow regimes and configurations by using either the linear stability theory or linear parabolized stability equations method. At high fidelity, users may use nonlinear PSE to track finite-amplitude disturbances until the skin friction rise. Coupled with the built-in receptivity model that is currently under development, the nonlinear PSE method offers a synergistic approach to predict transition onset for a given disturbance environment based on first principles. This document describes the governing equations, numerical methods, code development, detailed description of input/output parameters, and case studies for the current release of LASTRAC. Concrete examples that guide users from instability wave parameter exploration to N-factor or amplitude-based transition prediction through a series of calculations are given in this manual.

Acknowledgments

The author would like to thank Dr. Meelan Choudhari for his great help throughout the course of LASTRAC development. Many helpful discussions with ex-colleagues while the author was employed at High Technology Corporation are also acknowledged.

Chapter 1

Introduction

In the late 1980s and early 1990s, laminar-turbulent transition research was one of the outstanding issues for the National Aerospace Plane (NASP), High Speed Research (HSR), and Advanced Subsonic Technology (AST) programs. The recent DARPA Quiet Supersonic Platform (QSP) project (ref. [1]), aimed at devising a low-boom laminar flow supersonic aircraft, rekindles interests in supersonic laminar flow control. The ongoing NASA Langley Supersonic Vehicle Technology (SVT) project also focuses on developing technologies required for designing such supersonic vehicles. Computational tool development as well as a series of wind-tunnel and flight experiments are planned.

Laminar flow control has been investigated extensively over the past decades. Experimental, theoretical, and computational efforts on various issues such as receptivity, linear stability, and nonlinear evolution of finite-amplitude waves significantly broaden our knowledge base of this intricate flow phenomenon. For low-speed flows, various stages of the transition process are now fairly well understood. Theory and numerical simulations using either parabolized stability equations (PSE) or direct numerical simulations (DNS) accurately describe disturbance generation and evolution until the transitional stage. Secondary instability in two-dimensional and swept-wing boundary layers observed in low-speed experiments may also be calculated accurately via the nonlinear PSE and a 2-D eigenvalue approach (refs. [2, 3]). Saric, Carrillo, and Reibert (ref. [4]) discovered that transition onset due to stationary crossflow instability can be controlled by introducing spanwise periodic roughness elements (SPRE) near the leading edge to alter the mean state and subsequently suppress or delay the growth of the most unstable disturbances. This control phenomenon may be simulated and parametrically studied by nonlinear PSE (refs. [3, 5]).

For supersonic to hypersonic boundary layers, the breakdown mechanism is not fully understood, largely due to the lack of experimental measurements. For 2-D boundary layers, computational results have indicated that the secondary instability mechanism observed in low-speed cases may still be present (ref. [2]). However, other mechanisms such as the oblique-mode breakdown (ref. [6]), are also possible and even more relevant because of the large linear growth of the oblique disturbances in low supersonic boundary layers. At hypersonic speed, large amplitude second-mode disturbances have been observed experimentally (refs. [7, 8]) but the breakdown mechanism is unclear, mainly because the wave orientation in the experiments is largely unknown. The recent experiment by Horvath et al. (ref. [9]) on a flared cone in a conventional Mach 6 tunnel indicated that transition may or may not be caused by second-mode disturbances because of a small N-factor at the transition onset. We need more detailed experimental measurements to clarify this issue. Federov and Tumin (ref. [10]) and Zhong and Ma (ref. [11]) studied the receptivity of hypersonic boundary layers to acoustic disturbance. Fedorov and Tumin (ref. [12]) investigated, both theoretically and numerically, the role of the entropy layer on hypersonic boundary layers. Saric and Reed (ref. [13]) extended the SPRE control technique for supersonic swept-wing boundary layers.

Despite the advances in transition research, the traditional way of transition correlation and prediction using the e^N method remains one of the mainstream tools used in the design phase due to its simplicity and availability. The N-factor method provides a simple way to correlate transition locations and predict transition onset by using a prescribed N-value. For a higher fidelity transition prediction based on first principles, one can start from the receptivity model and compute the disturbance initial amplitudes according to the given disturbance environment, such as a roughness or free-stream disturbance distribution. Receptivity calculations may be based on a simple linear theory (e.g., the finite Reynolds number approach by Choudhari and Streett, ref. [14], or Crouch, ref. [15]) or a more advanced linear Navier-Stokes (LNS) method (refs. [16, 17]). The evolution of the internalized disturbances with known amplitudes can then be calculated by using nonlinear PSE until the skin friction rise. Alternatively, a secondary instability calculation may be performed for a steady, highly nonlinear boundary layer with the presence of stationary crossflow vortices (ref. [3]). To further refine the transition region or to carry the perturbed field into a fully turbulent stage, DNS or large eddy simulation (LES) may be used.

The integrated transition prediction approach described previously encompasses several mature computational means to form a package of tools for high-fidelity predictions beyond the conventional e^N method. Our current research effort is geared toward this integrated approach. To this end, the disturbance environment must be modeled correctly, which remains a difficult task except in a forced transition situation where manually excited roughness or free-stream waves dominate the disturbance environment. In a natural environment, statistical means must be introduced to account for the randomness of the disturbances. Some efforts have been made in this direction (ref. [18]). A case study for supersonic swept wing using the integrated approach is also available (ref. [19]).

One of the main goals of the ongoing NASA Langley projects for transition flow physics and prediction is to devise a set of tools to enable traditional and integrated transition prediction. The Langley Stability and Transition Analysis Code (LASTRAC) was developed to meet two main objectives: to provide an easy-to-use engineering tool for routine use and to incorporate state-of-the-art computational and theoretical findings for integrated transition predictions. For the former objective, LASTRAC can perform linear calculations and transition correlation by using the N-factor method based on either LST or a more advanced linear PSE method. The user interface of LASTRAC code was designed to allow users to identify unstable parameter space in terms of disturbance frequency and wave numbers with minimum effort. Transition prediction or correlation thus can be made with the identified parameter space. In addition to the traditional LST-based N-factor calculations, LASTRAC allows users to perform linear PSE N-factor calculations, which sometimes give a more compact N-factor range and better correlation.

To achieve the second objective, LASTRAC provides transition simulation capability based on an absolute amplitude. The receptivity module computes the initial amplitudes near the neutral stability location. Nonlinear PSE then simulates disturbance evolution until skin friction rise. For further refinement near the transition region, a subgrid scale model based on nonlinear PSE, or an LES and DNS may be used. These modules are all currently under development. We will only discuss the nonlinear PSE method in this paper.

In the current LASTRAC release, LST and linear and nonlinear PSE options are implemented for 2-D, axisymmetric, and infinite swept wing boundary layers. In addition, 2-D mixing layer, axisymmetric jet, and vortex flows are also handled by the current release. Extension to general three-dimensional (3-D) boundary layers is underway. This manual documents governing equations, numerical solutions, code development, and case studies for LASTRAC Release 1.2.

Chapter 2

Governing Equations and Numerical Formulation

2.1. Governing Equations

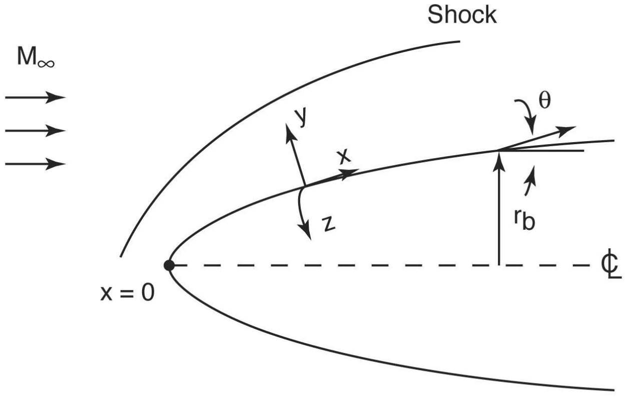

Figure 1: Coordinate system for 2-D or axisymmetric boundary layers.

The problem of interest is a compressible flow over a 2-D or axisymmetric body with a coordinate system depicted in figure 1 where x is the streamwise, y is the wall-normal, and z is the spanwise direction (or the azimuthal direction for axisymmetric cases). For the case of infinite swept wing boundary layers, the coordinate system used is shown in figure 2 where x is the normal-chord (perpendicular to the leading edge) direction and z is parallel to the leading edge. A nonorthogonal coordinate system can also be used for infinite swept wing boundary layers and will be adopted for future LASTRAC releases. We use a body-fitted coordinate to resolve the region adjacent to the wall. For 2-D or axisymmetric configurations, both streamwise curvature along x and transverse curvature along z may be included in the computation; only streamwise curvature is allowed for an infinite swept wing configuration. In the orthogonal body-fitted coordinate system, elements of length are h_1dx, dy, and h_3dz, in the x-, y-, and z-direction, respectively.

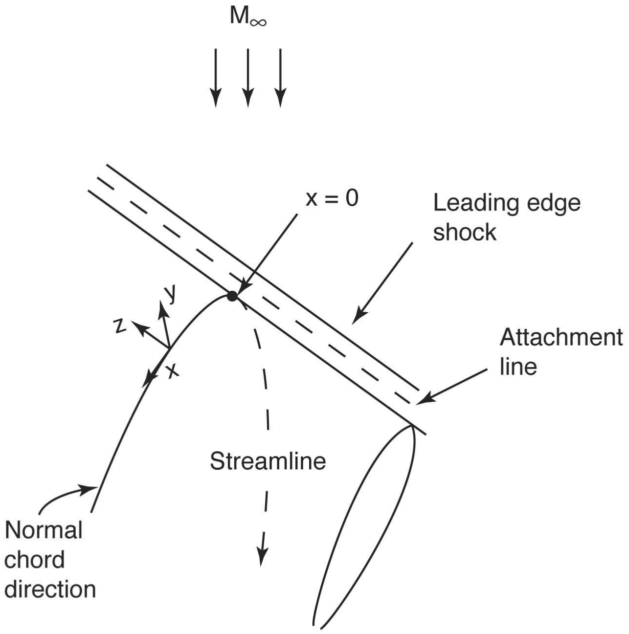

Figure 2: Coordinate system for infinite swept wing boundary layer.

The element length can be written as

$$dl = \sqrt{(h_1dx)^2 + dy^2 + (h_3dz)^2}$$

The metric h_1 is associated with the streamwise curvature κ and is defined as

$$h_1 = 1 + \kappa y$$

The metric h_3 is unity for nonaxisymmetric bodies; for axisymmetric configurations, it is defined as

$$h_3 = r_b + ycos(\theta)$$

where r_b is the local radius and θ is local half-angle along the axisymmetric surface.

The evolution of disturbances is governed by the compressible Navier-Stokes equations

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{V}) =& \quad 0 \\ \rho[\frac{\partial \vec{V}}{\partial t} + \vec{V} \cdot \nabla)\vec{V}] =& \quad - \quad \nabla p + \frac{1}{R}(\nabla[\lambda(\nabla \cdot \vec{V})] + \nabla \cdot [\mu(\nabla \vec{V} + \nabla \vec{V}^T)]) \\ \rho C_p[\frac{\partial T}{\partial t} + (\vec{V} \cdot \nabla)T] =& \quad \nabla \quad \cdot(k \nabla T) + \frac{\partial p}{\partial t} + (\vec{V} \cdot \nabla)p + \frac{\Phi}{R} \end{aligned} \tag{1}$$

where $\vec{V}$ is the velocity vector, ρ the density, p the pressure, T the temperature, C_p the specific heat, k the thermal conductivity, R the Reynolds number (to be defined later), and μ and λ the first and second coefficient of viscosity, respectively. The viscous dissipation function is defined as

$$\Phi = \lambda(\nabla \cdot \vec{V})^2 + \frac{\mu}{2}[\nabla \vec{V} + \nabla \vec{V}^T]^2 \tag{2}$$

The second coefficient of viscosity is related to the first one by the Stokes parameter s, defined as

$$\lambda = \frac{2}{3}(s-1)\mu \tag{3}$$

The perfect gas relation,

$$p = \rho \Re T \tag{4}$$

is used as the equation of state.

Equation (1) is in a nondimensional form, i.e., all lengths are scaled by a reference length scale ℓ, velocity by u_e, density by ρ_e, pressure by $\rho_e u_e^2$, temperature by T_e, viscosity by μ_e, and time by ℓ/u_e. For stability calculations, these normalization quantities are taken to be the boundary layer edge values. The length scale ℓ is the similarity boundary-layer length scale defined by

$$\ell = \sqrt{\nu_e x/u_e} \tag{5}$$

where ν_e is the dynamic viscosity. The corresponding Reynolds number is defined by

$$R = u_e \ell/\nu_e \tag{6}$$

The solution to the Navier-Stokes equations consists of two parts, the mean laminar flow solution and the disturbance fluctuation, i.e.,

$$\begin{aligned} u &= \bar{u} + u', v = \bar{v} + v', w = \bar{w} + w' \\ p &= \bar{p} + p', \rho = \bar{\rho} + \rho', T = \bar{T} + T' \\ \mu &= \bar{\mu} + \mu', \lambda = \lambda + \lambda', k = \bar{k} + k' \end{aligned} \tag{7}$$

Substituting equation (7) into equation (1) and subtracting the governing equations for the steady-state basic flow, we obtain the governing equations for the disturbances as

$$\begin{aligned} \Gamma \frac{\partial \phi}{\partial t} + \frac{A}{h_1}\frac{\partial \phi}{\partial x} + B\frac{\partial \phi}{\partial y} &+ \frac{C}{h_3}\frac{\partial \phi}{\partial z} + D\phi = \frac{1}{R_0}\left(\frac{V_{xx}}{h_1^2}\frac{\partial^2 \phi}{\partial x^2} + \frac{V_{xy}}{h_1}\frac{\partial^2 \phi}{\partial x \partial y} + V_{yy}\frac{\partial^2 \phi}{\partial y^2}\right. \\ &+ \left.\frac{V_{xz}}{h_3}\frac{\partial^2 \phi}{\partial x \partial z} + \frac{V_{yz}}{h_3}\frac{\partial^2 \phi}{\partial y \partial z} + \frac{V_{zz}}{h_3^2}\frac{\partial^2 \phi}{\partial z^2}\right) \end{aligned} \tag{8}$$

where ϕ is the disturbance vector defined as

$$\phi = (p', u', v', w', T')^T \tag{9}$$

The coefficient matrices, Γ, A, B, C, D, V_{xx}, V_{xy}, etc. are the Jacobians of the flux vectors, and $R_0 = u_e\ell_0/\nu_e$ is the Reynolds number used to normalize the equations. The reference length scale ℓ_0 depends on the flow configuration. For a boundary layer flow, the boundary layer length scale (defined in equation (5)) computed at the initial location of the stability calculations is used. For a jet or shear layer, the jet diameter or shear layer thickness may be used.

The Jacobian matrices can be further decomposed into two parts. The first part contains only mean quantities and the second part contains perturbation quantities. For example,

$$A = A^l + A^n$$

where $A^l = A^l(\bar{p}, \bar{u}, \bar{v}, \bar{w}, \bar{T}, ...)$ and $A^n = A^n(\bar{p}, \bar{u}, ...p', u', ...)$.

For a 2-D or infinite swept boundary layer, we assume the disturbance field is periodic in both temporal and spanwise directions. The disturbance vector ϕ can then be expressed as the following discrete Fourier series:

$$\phi(x, y, z, t) = \sum_{m=-M}^{M} \sum_{n=-N}^{N} \chi_{mn}(x, y) e^{i(n\beta z - m\omega t)} \tag{10}$$

where ω and β are the fundamental temporal and spanwise wave numbers of interest, respectively. M and N are the number of Fourier modes included in the calculation in time and space. The wave number ω is related to the physical frequency f by

$$\omega = \frac{2\pi\ell}{u_e} f \tag{11}$$

Another nondimensional frequency often used in stability calculations is defined by

$$F = \frac{2\pi\nu_e}{u_e^2} f \tag{12}$$

The spanwise wave number is defined by

$$\beta = 2\pi/\lambda_z \tag{13}$$

where λ_z is the spanwise wavelength normalized by the length scale ℓ_0. The fundamental mode (ω, β) determines the size of the computational domain ($1/f$ in time and λ_z in the spanwise direction) and M and N represent the numerical resolution ($2M + 1$ and $2N + 1$ discretized points in time and the spanwise direction) in the simulation. The mode shape of a mode (m, n) is represented by the complex vector χ_{mn}.

Equation (8) is the governing equation for the disturbance evolution. Direct solution of equation (8) is referred to as the direct numerical simulation (DNS) method. DNS requires a significant amount of computational time even for a simple linear case. Hence, a more efficient approximate solution to equation (8) is desirable. When the disturbance amplitude is very small relative to the mean quantities, each Fourier mode evolves independently. Nonlinear interaction among disturbance modes is negligible. The linearized version of equation (8) can be written as

$$\begin{aligned}\Gamma^l \frac{\partial \phi}{\partial t} + \frac{A^l}{h_1}\frac{\partial \phi}{\partial x} + B^l \frac{\partial \phi}{\partial y} \quad + \quad & \frac{C^l}{h_3}\frac{\partial \phi}{\partial z} + D^l \phi = \frac{1}{R_0}(\frac{V_{xx}^l}{h_1^2}\frac{\partial^2 \phi}{\partial x^2} + \frac{V_{xy}^l}{h_1}\frac{\partial^2 \phi}{\partial x \partial y} + V_{yy}^l \frac{\partial^2 \phi}{\partial y^2} \\ & + \frac{V_{xz}^l}{h_3}\frac{\partial^2 \phi}{\partial x \partial z} + \frac{V_{yz}^l}{h_3}\frac{\partial^2 \phi}{\partial y \partial z} + \frac{V_{zz}^l}{h_3^2}\frac{\partial^2 \phi}{\partial z^2})\end{aligned} \tag{14}$$

For a single disturbance mode (ω, β), ϕ can be expressed as

$$\phi = \chi(x, y)e^{i(\beta z - \omega t)} \tag{15}$$

Substituting equation (15) into equation (14), we obtain

$$\begin{aligned}(\frac{A^l}{h_1} - \frac{i\beta V^l_{xz}}{h_3 R_0})\frac{\partial \chi}{\partial x} \quad &+ \quad (B^l + \frac{i\beta V^l_{yz}}{h_3 R_0})\frac{\partial \chi}{\partial y} + (D^l - i\omega\Gamma^l + \frac{i\beta C^l}{h_3} + \frac{\beta^2 V^l_{zz}}{h_3^2 R_0})\chi = \\ &\frac{1}{R_0}(\frac{V^l_{xx}}{h_1^2}\frac{\partial^2 \chi}{\partial x^2} + \frac{V^l_{xy}}{h_1}\frac{\partial^2 \chi}{\partial x \partial y} + V^l_{yy}\frac{\partial^2 \chi}{\partial y^2})\end{aligned} \tag{16}$$

Equation (16) is the linearized Navier-Stokes (LNS) equation, which is a set of constant coefficient PDEs. To solve equation (16) numerically, we discretize it in both x- and y-directions; the equation then reduces to

$$A_{lns}\phi_{lns} = f_{lns} \tag{17}$$

where ϕ_{lns} is the solution vector for all discretized points in the field, f_{lns} is the forcing vector associated with the boundary conditions or external disturbance forcing such as wall blowing/suction or free-stream disturbances associated with acoustic waves or vorticity. The matrix A_{lns} is a banded or full block matrix depending on the discretization scheme used. If a fourth-order central difference scheme is used in x, A_{lns} is a block pentadiagonal matrix with the block size determined by the discretization scheme in the y-direction.

By using a disk-based caching technique, Streett (ref. [16]) and Guo, Malik, and Chang (ref. [17]) were able to achieve reasonable turn-around time for a moderate number of grid points in x and y. To enable routine engineering calculations, further approximations must be made for equation (16). We discuss several such approximate solutions in the following sections.

2.2. Quasi-Parallel Linear Stability Theory

In the linear stability theory, we assume that the mean flow evolves much more slowly in the streamwise direction than in the wall-normal direction. As a result, in the vicinity of a given location, the mean flow is quasi-parallel, i.e. the mean flow variation in x is negligible. The disturbance mode shape thus is assumed to have the following wave form:

$$\chi(x, y) = \psi(y)e^{i\alpha x} \tag{18}$$

The disturbance vector can be expressed as

$$\phi(x, y, z, t) = \psi(y)e^{i(\alpha x + \beta z - \omega t)} \tag{19}$$

where α is the local streamwise wave number at the given streamwise location. Substituting equation (19) into equation (16) and neglecting the mean wall-normal velocity component (following the quasi-parallel assumption) and all viscous terms that are of order $1/R_0^2$ or lower, we obtain

$$\begin{aligned}(B^l + \frac{i\beta V^l_{yz}}{h_3 R_0} - \frac{i\alpha V^l_{xy}}{h_1 R_0})\frac{d\psi}{dy} + (D^l - i\omega\Gamma^l + \frac{i\alpha A^l}{h_1} + \frac{\alpha^2 V^l_{xx}}{h_1^2 R_0} - \frac{i\beta V^l_{xz}}{h_3 R_0} \\ + \frac{i\beta C^l}{h_3} + \frac{\beta^2 V^l_{zz}}{h_3^2 R_0})\psi = \frac{V^l_{yy}}{R_0}\frac{d^2\psi}{dy^2}\end{aligned} \tag{20}$$

The viscous term in the wall-normal direction is retained in the above equation to handle viscous instability such as the Tollmien-Schlichting wave. Equation (20) is a set of ordinary differential equations (ODE). In conjunction with a set of homogeneous boundary conditions to be described later, it constitutes an eigenvalue problem with the following dispersion relation,

$$\alpha = \alpha(\omega, \beta) \tag{21}$$

Stability calculations in LASTRAC are formulated as a spatial stability problem, i.e., for a given disturbance frequency ω and spanwise wave number β, the complex eigenvalue $\alpha = \alpha_r + i\alpha_i$ is solved numerically. If $\alpha_i < 0$, the eigenmode is unstable. The corresponding solution of ψ is the eigenfunction of the mode under consideration.

2.3. Linear Parabolized Stability Equations

In the quasi-parallel linear stability theory, the nonparallel effect due to boundary layer growth is neglected. To account for this boundary layer growth, Herbert (ref. [20]) suggested decomposing the mode shape $\chi(x, y)$ into two parts; the wave part described by a complex wave number α, and the shape-function part ψ to track additional nonparallel effects, i.e.,

$$\chi(x, y) = \psi(x, y) e^{i \int_{x_0}^{x} \alpha(\xi) d\xi} \tag{22}$$

where $\psi = (\hat{p}, \hat{u}, \hat{v}, \hat{w}, \hat{T})$ is the shape-function vector. Here, $\alpha = \alpha(x)$ varies along the streamwise direction and its history effect is preserved in the integral. We assume that α does not vary along the wall-normal direction to simplify the decomposition, and the shape-function variation in y is captured in $\psi(x, y)$. In contrast to the linear stability theory, the shape-function ψ also depends on x. This shape-function dependency in x allows additional shape variation in x on top of the contribution from the wave number α. Hence, the nonparallel effect in the solution has contributions from both the streamwise wave number α and the variation of shape function in x. Substituting equation (22) into equation (16), we have

$$\hat{A}\frac{\partial \psi}{\partial x} + \hat{B}\frac{\partial \psi}{\partial y} + \hat{D}\psi = \frac{1}{R_0}(\frac{V^l_{xx}}{h_1^2}\frac{\partial^2 \psi}{\partial x^2} + \frac{V^l_{xy}}{h_1}\frac{\partial^2 \psi}{\partial x \partial y} + V^l_{yy}\frac{\partial^2 \psi}{\partial y^2}) \tag{23}$$

where matrices $\hat{A}$, $\hat{B}$, and $\hat{D}$ are defined by

$$\begin{aligned} \hat{A} &= \frac{A^l}{h_1} - \frac{2i\alpha V^l_{xx}}{h_1^2 R_0} - \frac{i\beta V^l_{xz}}{h_3 R_0} \\ \hat{B} &= B^l - \frac{i\alpha V_{xy}}{h_1 R_0} - \frac{i\beta V^l_{xz}}{h_3 R_0} \\ \hat{D} &= D^l - i\omega\Gamma^l + \frac{i\alpha A^l}{h_1} + \frac{i\beta C^l}{h_3} - (i\frac{d\alpha}{dx} - \alpha^2)\frac{V^2_{xx}}{h_1^2 R_0} + \frac{\alpha\beta V^l_{xz}}{h_1 h_3 R_0} + \frac{\beta^2 V_{zz}}{h_3^2 R_0}. \end{aligned} \tag{24}$$

This PDE differs from the linearized Navier-Stokes equations (eq. (16)) only in the terms associated with the wave part α. In fact, Guo, Malik, and Chang (ref. [17]), introduced a wave part in the LNS solver to reduce the number of grid points in x required to resolve the oscillatory wave and thereby increase the computational efficiency. The LNS solution procedure described in the previous section can be used for equation (23). A marching solution is more efficient if we can parabolize equation (23). To this end, we neglect the viscous derivatives in x and equation (23) reduces to

$$\hat{A}\frac{\partial\psi}{\partial x}+\hat{B}\frac{\partial\psi}{\partial y}+\hat{D}\psi=\frac{V^l_{yy}}{R_0}\frac{\partial^2\psi}{\partial y^2} \tag{25}$$

Chang et. al. (ref. [2]) pointed out that equation (25) is only "nearly parabolic" due to the streamwise pressure gradient term ($\partial p'/\partial x$) inherited from the original Navier-Stokes equation (1). Numerical instability similar to that observed in the parabolized Navier-Stokes equations (PNS) (ref. [21]) would occur if one attempted to solve equation (25) by using a small enough marching stepsize in x. Chang et al. (ref. [2]) further suggested that the Vigneron's approximation (ref. [21]) may be used to suppress numerical instability. Li and Malik (ref. [22]) used Fourier analysis to prove the existence of numerical instability and quantify the numerical instability bound.

In LASTRAC, equation (25) is solved by using a marching procedure. For small marching step sizes, the streamwise disturbance pressure gradient is treated as

$$\frac{\partial p'}{\partial x}=(i\alpha\hat{p}+\Omega\frac{\partial\hat{p}}{\partial x})e^{i(\int_{x_0}^{x}\alpha d\xi+\beta z-\omega t)} \tag{26}$$

where $\hat{p}$ is the pressure shape-function. The contribution from the wave part $i\alpha\hat{p}$ is completely retained in the $\hat{D}\psi$ term in equation (25), and the contribution from the shape-function part is approximated by the Vigneron parameter Ω determined by

$$\Omega=\sigma_p Min(1,M_x^2) \tag{27}$$

where σ_p is a coefficient to control the PNS approximation and M_x is the local streamwise Mach number. In the incompressible limit or $\sigma_p=0$, this approach is equivalent to neglecting the $\partial\hat{p}/\partial x$ part in equation (26) completely. For locally supersonic flow, equation (27) does not alter the governing equations since $\Omega=1$ (assuming $\sigma_p=1$). For an instability wave with $\alpha\neq 0$, part of the elliptic behavior is absorbed in the wave part; if the marching step size is large enough, then Vigneron's approximation is not necessary ($\Omega=1$). However, for a mode with $\alpha=0$, equation (27) must be used to compute Ω in equation (26). Examples are Görtler vortex mode, the mean flow distortion mode (the $(0,0)$ mode), or stationary vortex modes ($(0,n)$ and $n\neq 0$) in the nonlinear PSE calculations.

The wave decomposition in equation (22) is nonunique because we can place the wave part in either ψ or α. Ideally, we should place as much oscillatory wave in the α as possible to minimize the shape-function (ψ) variation in x. Therefore, a low-order finite difference scheme in x is sufficient. Assume a disturbance quantity ψ_n (e.g., peak mass flow rate, peak temperature, or disturbance kinetic energy) is selected to "normalize" the decomposition and obtain a wave number α_n. Then the "effective" wave number α_e should include the part from the normalized wave number α_n and the variation of the shape-function, i.e.,

$$\alpha_e=\alpha_n-i\frac{1}{\psi_n}\frac{\partial\psi_n}{\partial x} \tag{28}$$

The real part of α_e represents the phase of the disturbance and the imaginary part gives the growth rate, both corresponding to the variable ψ_n chosen. For a given location x, equation (28) may be applied repeatedly to the selected ψ_n until the change in α_e is small. Hence most of the oscillatory motion is contained in α for the variable ψ_n. The solution then marches to the next x location. Typically, three to five iterations are enough for the linear PSE.

In references [2, 23], various flow variables ψ_n were used to normalize the wave decomposition and the PSE results were shown to be insensitive to the variable selected. In LASTRAC, the kinetic energy integral

$$\psi_{KE} = \int_0^{y_{max}} (\hat{u}^*\hat{u} + \hat{v}^*\hat{v} + \hat{w}^*\hat{w})dy \tag{29}$$

is used to normalize α via the following normalization

$$\alpha_e = \alpha_n - \frac{1}{\psi_{KE}} \int_0^{y_{max}} (\hat{u}^*\frac{\partial \hat{u}}{\partial x} + \hat{v}^*\frac{\partial \hat{v}}{\partial x} + \hat{w}^*\frac{\partial \hat{w}}{\partial x})dy \tag{30}$$

In equations (29) and (30), the superscript $*$ represent the complex conjugate.

In a nonparallel (growing) boundary layer or shear layer, the streamwise wave number α obtained from the PSE solution does not represent the true physical wave number and growth rates. The physical growth rates and wave numbers depend on the flow quantity and location inside the boundary layer. The effective streamwise wave number is evaluated using

$$\alpha_e = \alpha - i\frac{1}{\psi_s}\frac{\partial \psi_s}{\partial x} \tag{31}$$

where ψ_s is a disturbance quantity evaluated at certain y location. The imaginary part of $-\alpha_e$ represents the disturbance growth rate. In general, numerical results indicate that α_i varies slowly near the critical layer where disturbance reaches a maximum value. Therefore, nonparallel growth rate is usually evaluated at the location where a chosen disturbance quantity peaks. Alternatively, one can use an integral quantity such as kinetic energy (eq. (29)) to evaluate the "averaged" nonparallel growth rate. LASTRAC computes growth rates for mass flow rate, five dependent flow variables, and kinetic energy and writes them in the output.

2.4. Nonlinear Parabolized Stability Equations

Similar to the derivation in linear PSE, for a Fourier mode $(m\omega, n\beta)$, the mode shape is decomposed into two parts:

$$\begin{aligned} \chi_{mn} &= \psi_{mn}(x,y)A_{mn}(x) \qquad (32) \\ A_{mn}(x) &= e^{i\int_{x_0}^{x} \alpha_{mn}(\xi)d\xi} \qquad (33) \end{aligned}$$

where $\psi_{mn} = (\hat{p}_{mn}, \hat{u}_{mn}, \hat{v}_{mn}, \hat{w}_{mn}, \hat{T}_{mn})$ is the shape-function for the Fourier mode (m, n) and α_{mn} is the associated streamwise wave number. Substituting equation (32) into equation (8) and neglecting the viscous x derivatives, we have the nonlinear parabolized stability equations for mode (m, n), i.e.,

$$\hat{A}_{mn}\frac{\partial \psi_{mn}}{\partial x} + \hat{B}_{mn}\frac{\partial \psi_{mn}}{\partial y} + \hat{D}_{mn}\psi_{mn} - \frac{V_{yy}^l}{R_0}\frac{\partial^2 \psi_{mn}}{\partial y^2} = \frac{F_{mn}}{A_{mn}} \tag{34}$$

where matrices $\hat{A}_{mn}$, $\hat{B}_{mn}$, and $\hat{D}_{mn}$ are defined by

$$
\begin{aligned}
\hat{A}_{mn} &= \frac{A^l}{h_1} - \frac{2i\alpha_{mn}V^l_{xx}}{h_1^2 R_0} - \frac{in\beta V^l_{xz}}{h_3 R_0} \\
\hat{B}_{mn} &= B^l - \frac{i\alpha_{mn}V_{xy}}{h_1 R_0} - \frac{in\beta V^l_{xz}}{h_3 R_0} \\
\hat{D}_{mn} &= D^l - im\omega\Gamma^l + \frac{i\alpha_{mn}A^l}{h_1} + \frac{in\beta C^l}{h_3} - (i\frac{d\alpha_{mn}}{dx} - \alpha^2_{mn})\frac{V^2_{xx}}{h_1^2 R_0} + \frac{n\alpha\beta V^l_{xz}}{h_1 h_3 R_0} + \frac{n^2\beta^2 V_{zz}}{h_3^2 R_0}
\end{aligned}
\tag{35}
$$

The left-hand side of equation (34) contains only linear coefficient matrices; all nonlinear terms are included in the forcing function F_{mn}, which is the Fourier component of the total forcing F^n defined as

$$
\begin{aligned}
F^n = -\Gamma^n\frac{\partial\phi}{\partial t} &- \frac{A^n}{h_1}\frac{\partial\phi}{\partial x} - B^n\frac{\partial\phi}{\partial y} - \frac{C^n}{h_3}\frac{\partial\phi}{\partial z} - D^n\phi + \frac{1}{R_0}(\frac{V^n_{xx}}{h_1^2}\frac{\partial^2\phi}{\partial x^2} + \frac{V^n_{xy}}{h_1}\frac{\partial^2\phi}{\partial x\partial y} + V^n_{yy}\frac{\partial^2\phi}{\partial y^2} \\
&+ \frac{V^n_{xz}}{h_1 h_3}\frac{\partial^2\phi}{\partial x\partial z} + \frac{V^n_{yz}}{h_3}\frac{\partial^2\phi}{\partial y\partial z} + \frac{V^n_{zz}}{h_3^2}\frac{\partial^2\phi}{\partial z^2})
\end{aligned}
\tag{36}
$$

where the disturbance vector ϕ is defined in physical space (see eq. (9)). Nonlinear forcing for each Fourier mode is defined by the following Fourier transform:

$$
F^n(x, y, z, t) = \sum_{m=-M}^{M}\sum_{n=-N}^{N} F_{mn}(x, y)e^{i(n\beta z - m\omega t)} \tag{37}
$$

The forcing term F_{mn} for a given Fourier mode (m, n) can be evaluated by collecting all nonlinear terms contributing to it. For instance, the $(0, 1)$ mode interacts quadratically with the $(1, 0)$ mode to influence the $(1, 1)$ mode. In LASTRAC, nonlinear forcing terms are evaluated by computing F^n in physical space and then transforming F^n back into the wave number space F_{mn}.

The nonlinear forcing term on the right-hand side of equation (34) is a function of various Fourier modes. Solving this coupled PDE in an implicit fashion would require a time-consuming solution of a large full matrix that is only feasible when a small number of Fourier modes are included in the calculation. In LASTRAC, we use an iterative method to solve the system of coupled equations. Each Fourier mode is solved independently by using the forcing computed from the previous iteration. The iterative procedure is repeated until the L2 norm of the shape-function is less than a predetermined tolerance. The iterative method is robust for small to moderate disturbance amplitudes and is easier to parallelize than the implicit method. When strong nonlinear interaction takes place, the right-hand side forcing carries more weight than the left-hand side linear operator in equation (34). Consequently, numerical instability may occur for the iterative procedure. Numerical experiments indicate that including the mean flow distortion (the $(0, 0)$ Fourier mode) in the left-hand side linear operator can stabilize the iterative procedure. Alternatively, when strong nonlinear interaction takes place, the nonlinear operator may be stabilized by adding a subgrid scale model, similar to that in large eddy simulations, in the nonlinear iterative procedure. This subgrid scale model will be implemented in the future release.

2.5. Boundary Conditions

To close the problem, we need boundary conditions at both ends of the domain in the wall-normal direction. At the wall, we use the no slip conditions

$$\hat{u} = \hat{v} = \hat{w} = \hat{T} = 0 \tag{38}$$

For nonlinear PSE, equation (38) is applied for all Fourier modes. For the solution of a set of coupled ODEs involved in the linear stability theory, or when a two-point compact scheme (ref. [24]) is used for the PDEs involved in the PSE approach, only four boundary conditions are necessary at the wall boundary and equation (38) suffices. In LASTRAC, a central-difference scheme is used; thus, we need one more condition. It has been found that either the continuity equation or normal momentum equation can be used as the fifth auxiliary condition. In LASTRAC, the continuity equation is used.

In the free-stream, the Dirichlet boundary conditions

$$\hat{u} = \hat{v} = \hat{w} = \hat{T} = 0, \quad y \to \infty. \tag{39}$$

are used. Equation (39) works well for subsonic modes. For supersonic modes, the disturbance has a non-decaying oscillatory structure in the free-stream; hence, the use of a Dirichlet condition will cause spurious reflection. In this case, we can use the nonreflecting boundary condition at the far field. These nonreflecting boundary conditions are derived, as suggested by Thompson (ref. [25]), based on the inviscid Euler equations because viscous effects are negligible in the farfield. The inviscid disturbance equations at the farfield can be written as

$$\Gamma \frac{\partial \phi}{\partial t} + \frac{A}{h_1} \frac{\partial \phi}{\partial x} + B^+ \frac{\partial \phi}{\partial y} + \frac{C}{h_3} \frac{\partial \phi}{\partial z} + D\phi = 0 \tag{40}$$

In equation (40), all coefficient matrices are identical to those given in equation (8), except B^+, which is defined as

$$B^+ = \Gamma(L \Lambda^+ L^{-1})$$

where L is the left eigenvector matrix of the product of the coefficient matrix $\Gamma^{-1}B$ given in equation (8), and Λ^+ is the modified diagonal eigenvalue matrix defined by

$$\Lambda^+ = diag(max(0, \lambda_j))$$

in which λ_j are the corresponding eigenvalues of the matrix $\Gamma^{-1}B$.

Equation (40) contains only the outgoing characteristic equations, and it suppresses numerical reflections from the far field. Numerical experiments indicate that the use of these nonreflecting boundary conditions allows a smaller computational domain; thus more grid points can be used to resolve the important disturbance structure inside the shear layer. However, a large number of grid points must be used and the far field conditions can only be applied at a large wall-normal distance if the Dirichlet conditions (eq. (39)) are used.

For nonlinear PSE, the above Dirichlet or nonreflecting condition may be applied for each Fourier mode with the exception of the mean flow distortion mode $(0, 0)$. For incompressible boundary layers, the mean flow distortion mode should be allowed to compensate for the change in displacement thickness due to nonlinear interaction. In this case, the following equation

$$\hat{u}_{00} = \frac{\partial \hat{v}_{00}}{\partial y} = \hat{w}_{00} = \hat{T}_{00} \qquad y \to 0$$

is used. For supersonic boundary layers, the nonreflecting condition 40 works equally well.

If a shock is present in the free-stream, the Rankine-Hugoniot shock condition for disturbances suggested by reference [26] may be used. When the shock is located far away from the boundary layer (e.g., five or more times the boundary thickness), the effect on the instability wave is very small. The shock boundary condition, as described in reference [27], is implemented in the LASTRAC code for both linear and nonlinear PSE solvers.

2.6. Discretized Equations

Numerical solutions of the LST or PSE require discretization of the governing equations in both x- and y-directions. We transform the governing equation to a generalized coordinate system defined by

$$\xi = \xi(x, y)$$
$$\eta = \eta(x, y)$$

After this transformation, the nonlinear PSE becomes

$$\tilde{A}_{mn} \frac{\partial \psi_{mn}}{\partial \xi} + \tilde{B}_{mn} \frac{\partial \psi_{mn}}{\partial \eta} + \tilde{D}_{mn} \psi_{mn} - \tilde{V}_{\eta\eta} \frac{\partial^2 \psi_{mn}}{\partial \eta^2} = \frac{F_{mn}}{A_{mn}} \tag{41}$$

where the transformed coefficient matrices are defined by

$$\begin{aligned} \tilde{A}_{mn} &= \xi_x \hat{A}_{mn} + \xi_y \hat{B}_{mn} \\ \tilde{B}_{mn} &= \eta_x \hat{A}_{mn} + \eta_y \hat{B}_{mn} - J \frac{V_{yy}^l}{R_0} \frac{\partial}{\partial \eta} (\frac{\eta_y^2}{J}) \\ \tilde{D}_{mn} &= \hat{D}_{mn} \\ \tilde{V}_{\eta\eta} &= \frac{\eta_y^2 V_{yy}^l}{R_0} \end{aligned} \tag{42}$$

with the Jacobian of transformation defined as

$$J = \xi_x \eta_y + \xi_y \eta_x$$

Using this transformation, we map a computational grid with arbitrary clustering into a uniform mesh with constant increments in ξ and η coordinates. The metrics ξ_x, η_x, etc. may be calculated numerically by using a finite-difference scheme. The main advantage of using a general coordinate system over more traditional analytical mapping is the flexibility of grid clustering.

Three streamwise grid distributions may be chosen in LASTRAC. The first one adopts the same grid used in the basic flow calculations. The second and third ones use constant increments of the streamwise coordinate

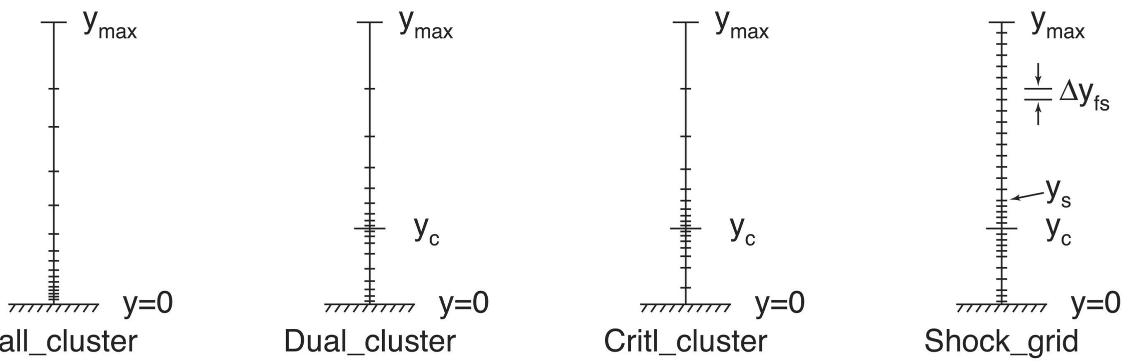

Figure 3: Four normal grid distributions for stability calculations.

x and the Reynolds number R (defined in eq. (6)), respectively. Interpolations are used for streamwise locations not coinciding with the grid used in the basic flow.

As depicted in figure 3, there are four different types of grids in the wall-normal direction in LASTRAC. In the following discussion, we use a parameter ζ $(0 \leq \zeta \leq 1)$ to denote the mapped coordinate. It has a constant increment ζ/N with N representing the number of points used. The first grid clusters the grid points near the wall according to the following algebraic mapping:

$$y = \frac{a\zeta}{b-\zeta} \tag{43}$$

where $b = 1 + a/y_{max}$. The parameter a controls the clustering near the wall. The second grid is designed for high supersonic and hypersonic boundary layers. It clusters grid points both near the wall and the critical layer. The critical layer location varies with the phase speed of the instability wave; for hypersonic boundary layers, it is located close to the boundary layer edge. Let y_c denote the critical layer location. The algebraic mapping defined in two regions can be written as

$$\begin{aligned} y &= y_c(1-\cos\pi\zeta_1)/2; \quad 0 \leq y \leq y_c \\ y &= y_c + \frac{a\zeta_2}{b-\zeta_2}; \quad y_c \leq y \leq y_{max} \end{aligned} \tag{44}$$

where ζ_1 and ζ_2 represent the mapped coordinate in the two regions. The parameters a and b are

$$\begin{aligned} b &= 1 + a/(y_{max} - y_c) \\ a &= y_{max} - y_c \end{aligned}$$

Half of the grid points are used in both regions. These mappings were suggested by reference [28].

The third grid is similar to the second one except that it clusters points only near the critical layer. The interior hyperbolic mapping suggested by reference [29] is used for this grid type. For stability problems such as a shear layer, good resolution is needed only near the critical layer and this grid type is well suited.

If a shock is present in the free-stream boundary, or if the eigenfunction exhibits an oscillatory structure in the free-stream as in a supersonic mode (see ref. [27]), then a uniform grid in the free-stream better resolves the disturbance eigenfunction structure. In the fourth grid, we prescribe the increment in the free-stream h_s and let y_s denote the location where the grid increment in the second region in equation (44) exceeds h_s. The mapping becomes

$$\begin{aligned} y &= y_c(1-\cos\pi\zeta_1)/2; \qquad 0 \le y \le y_c \\ y &= y_c + \frac{a\zeta_2}{b-\zeta_2}; \qquad y_c \le y \le y_s \\ y &= y_s + \zeta_3(y_{max} - y_s); \qquad y_s \le y \le y_{max} \end{aligned} \tag{45}$$

and the number of points in the third region is $N_3 = (y_{max} - y_s)/h_s$. This "shock grid" works very well for supersonic modes.

We use indices i and j to denote the grid index along the streamwise and wall-normal directions, respectively. In the streamwise direction, we use either the first-order scheme

$$\frac{\partial\psi}{\partial\xi} = \frac{\psi_{i,j} - \psi_{i-1,j}}{\Delta\xi} \tag{46}$$

or the second-order scheme

$$\frac{\partial\psi}{\partial\xi} = \frac{3\psi_{i,j} - 4\psi_{i-1,j} + \psi_{i-2,j}}{2\Delta\xi} \tag{47}$$

where $\Delta\xi = 1$ for convenience. Numerical calculations indicate that a first-order scheme works well for most of the cases when more than three or four points per wave length are used in the streamwise direction.

In the wall-normal direction, we employ the fourth-order central-difference scheme

$$\begin{aligned} \frac{\partial\psi}{\partial\eta} &= \frac{-\psi_{i,j+2} + 8\psi_{i,j+1} - 8\psi_{i,j-1} + \psi_{i,j-2}}{12\Delta\eta} \\ \frac{\partial^2\psi}{\partial\eta^2} &= \frac{-\psi_{i,j+2} + 16\psi_{i,j+1} - 30\psi_{i,j} + 16\psi_{i,j-1} - \psi_{i,j-2}}{12\Delta\eta^2}. \end{aligned} \tag{48}$$

For the grid points next to the boundaries, the five-point scheme (eq. (48)) is replaced with the two-point second-order scheme

$$\begin{aligned} \frac{\partial\psi}{\partial\eta} &= \frac{\psi_{i,j+1} - \psi_{i,j-1}}{2\Delta\eta} \\ \frac{\partial^2\psi}{\partial\eta^2} &= \frac{\psi_{i,j+1} - 2\psi_{i,j} + \psi_{i,j-1}}{\Delta\eta^2} \end{aligned} \tag{49}$$

Substituting the above discretized equations into the governing equations, equation (34), we have

$$L^i_{mn}\Psi^i_{mn} = K_{mn} + F^i_{mn} \tag{50}$$

where $\Psi^i_{mn} = (\psi_{i,1}, \psi_{i,2}, ...\psi_{i,N_y})$ is the solution vector for N_y points in the wall normal direction. the matrix operator L^i_{mn} is a block pentadiagonal matrix with a length of N_y and a block size of 5×5. It is a function of linear Jacobian matrices such as A^l, B^l, etc. and the wave numbers ω, α_{mn}, and β. The first vector on the right-hand side, K_{mn}, is due to the discretization of the term involving $\partial\psi/\partial\xi$ and can be expressed as a function of the solution at the previous station, Ψ^{i-1}_{mn}. The vector F^i_{mn} is due to nonlinear forcing.

For linear stability theory, the ξ derivative vanishes and there is no forcing; the discretized equation becomes

$$L\Psi = 0 \tag{51}$$

where the subscripts mn and superscripts i have been dropped for clarity. The L operator contains only η derivatives for station i. This homogeneous equation coupled with the homogeneous boundary conditions constitutes an eigenvalue problem. Its solution procedure will be discussed in the following sections.

2.7. Eigenvalue Solution Procedure

In the linear stability theory, the governing equations reduce to a set of ODEs, equation (20), or its discretized form, equation (51). Note that the operator L in equation (51) is a function of wave numbers:

$$L = L(\omega, \alpha, \beta)$$

In the spatial stability problem, we are looking for the spatial eigenvalue α for a given disturbance frequency ω and spanwise wave number β. In LASTRAC, only the spatial stability problem described by the dispersion relation, $\alpha = \alpha(\omega, \beta)$ is implemented.

We note that in equation (51), L is linear in ω and contains terms of α^2 due to the viscous terms. If we neglect the viscous terms in L, equation (51) can be recast to

$$P\Psi = \alpha Q\Psi \tag{52}$$

where P and Q are block matrices. Q is only a block diagonal matrix and thus can be easily inverted to give

$$|Q^{-1}P - \alpha I| = 0 \tag{53}$$

in which I is the identity matrix. This eigenvalue problem can be solved by a conventional eigenvalue algorithm such as QR or LR. In LASTRAC, two options are provided: the complex eigenvalue solver from the LAPACK (http://www.netlib.org/lapack/) or a built-in complex LR eigenvalue solver. A global eigenvalue spectrum can then be obtained from these eigenvalue solvers. This global spectrum will contain all discrete modes and the continuous spectrum. Discrete modes can be easily identified if they are unstable. However, these unstable eigenvalues are not exact due to the neglected viscous terms.

We can solve the exact eigenvalue problem by using a local eigenvalue search. Here, *local* is used in contrast to the global eigenvalue spectrum. An initial guess of the eigenvalue α is provided before the local search starts. Let $\tau = 0$ denote the target equation that needs to be satisfied. Newton's method gives

$$\alpha^{n+1} = \alpha^n - \tau^n / (\frac{\partial \tau}{\partial \alpha})^n \tag{54}$$

in which superscript n denotes the iteration count. The quantity $\partial\tau/\partial\alpha$ can be evaluated numerically by using a small increment of α. When the iteration converges, the target equation $\tau = 0$ is satisfied. We can use the second-order Newton's method as suggested by reference [28] to increase the convergence radius. The target equation is usually chosen to be one of the boundary conditions at the wall (either $\hat{u} = 0$ or $\hat{T} = 0$). We replace this chosen equation by a normalization equation (usually by setting $\hat{p} = 1$). Equation (51) is solved with the modified boundary conditions. Equation (54) is then employed to update α until the replaced wall boundary condition is satisfied. Because we use the exact equation, at convergence, the correct α is obtained without any approximation. Newton's method does not guarantee convergence. Therefore, the eigenvalue search procedure usually begins with a global eigenvalue search. The resulting global eigenvalue spectrum provides a good initial guess for the local eigenvalue search. In addition, a second-order Newton's method is used to increase the radius of convergence, and hence the robustness of the eigenvalue search procedure. We note that instead of replacing one of the wall conditions, we can relax any discretized equation inside the domain and replace it with a normalization equation at the point under consideration and then iterate on α using the Newton's method until the relaxed discretized equation is satisfied. Usually the normal momentum or energy equation near the critical layer is relaxed. This procedure is quite robust in tracking eigenvalues, especially in the stable region. For jet or shear layers, we need to use a field point equation as the target because the eigenfunction variation diminishes near the center line.

In summary, the eigenvalue solution thus involves two main steps. For a given spatial location and (ω, β), a global eigenvalue search is performed and unstable modes are selected from the global spectrum. These selected modes are then fed into the local eigenvalue solvers as initial guesses. Converged modes from the local eigenvalue search are probably good discrete modes. When multiple unstable modes are present, LASTRAC selects the most unstable mode automatically. With this combined procedure, the eigenvalue search becomes much more robust as compared to other numerical methods based on shooting techniques.

2.8. Nonparallel Eigenvalue Solution Procedure

The PSE formulation requires an initial condition to start the streamwise marching procedure. A simple way to initiate marching is to use the eigenfunction obtained from the quasi-parallel linear stability theory (LST) solution. However, because of the neglected nonparallel terms, the marching PSE solution obtained this way would have a "transient" region where the PSE solution attempts to slowly adjust the inconsistent parallel eigenfunctions. This transient effect, more noticeable for supersonic boundary layers, is a result of the difference in governing equations between LST and PSE. We can use the multiple-scale method (refs. [30, 31]) to account for the nonparallel effect, and the eigenfunction generated may be used to initiate the PSE marching procedure in order to alleviate the transient effect.

LASTRAC offers an alternative nonparallel eigensolution (NES) procedure based on the discretized parabolized governing equations. The governing equation is locally transformed to a general coordinate system by using

$$\xi = \xi(x, y)$$
$$\eta = \eta(x, y)$$

Numerical calculation is then performed in the (ξ, η) plane. Using first-order finite difference in ξ, the governing linear PSE at two consecutive solution locations, i and $i+1$, may be written as

$$\begin{aligned} \tilde{D}_i \psi_i + \tilde{A}_i(\psi_{i+1} - \psi_i) + \tilde{B}_i \frac{\partial \psi_i}{\partial \eta} &= \tilde{V}_{\eta\eta,i} \frac{\partial^2 \psi_i}{\partial \eta^2} \\ \tilde{D}_{i+1} \psi_{i+1} + \tilde{A}_{i+1}(\psi_{i+1} - \psi_i) + \tilde{B}_{i+1} \frac{\partial \psi_{i+1}}{\partial \eta} &= \tilde{V}_{\eta\eta,i+1} \frac{\partial^2 \psi_{i+1}}{\partial \eta^2} \end{aligned} \tag{55}$$

For this equation, we assume that the second derivative in ξ is negligible and $\partial\psi/\partial\xi$ is the same for both point i and $i+1$. This assumption is equivalent to taking a forward difference for point i and a backward difference for point $i+1$. The streamwise wave number α_i and α_{i+1} is related by

$$\alpha_{i+1} = \alpha_i + \frac{d\alpha}{dx}\Delta x \tag{56}$$

in which Δx is the streamwise distance between points i and $i+1$.

If we define the combined shape-function vector Ψ as

$$\Psi = \begin{pmatrix} \psi_i \\ \psi_{i+1} \end{pmatrix}$$

then equation (55) may be rewritten as

$$\bar{D}\Psi + \bar{B}\frac{d\Psi}{d\eta} = \bar{V}\frac{d^2\Psi}{d\eta^2} \tag{57}$$

where matrices $\bar{D}$, $\bar{B}$, and $\bar{V}$ are

$$\begin{aligned} \bar{D} &= \begin{pmatrix} \tilde{D}_i - \tilde{A}_i & \tilde{A}_i \\ -\tilde{A}_{i+1} & \tilde{D}_{i+1} + \tilde{A}_{i+1} \end{pmatrix} \\ \bar{B} &= \begin{pmatrix} \tilde{B}_i & 0 \\ 0 & \tilde{B}_{i+1} \end{pmatrix} \\ \bar{V} &= \begin{pmatrix} \tilde{V}_{\eta\eta,i} & 0 \\ 0 & \tilde{V}_{\eta\eta,i+1} \end{pmatrix} \end{aligned} \tag{58}$$

At the boundaries, homogeneous boundary conditions (e.g., no slip at the wall and Dirichlet or nonreflecting conditions at free-stream) may be applied for both i and $i+1$ stations. Equation (57) is a set of homogeneous ordinary differential equations (ODEs). After discretization in η, Equation (57) can be cast into the following nonparallel eigenvalue problem:

$$L_{np}(\Psi) = 0 \tag{59}$$

where $L_{np} = L_{np}(\omega, \alpha, d\alpha/dx, \beta)$. For a given disturbance (ω, β), eigenvalues α and $d\alpha/dx$ need to be found. Unlike the LST eigenvalue problem, the NES has one more unknown: $d\alpha/dx$. Because the variation of α is generally small, our first approximation is that $d\alpha/dx = 0$. Thus the eigenvalue problem is similar to the LST case, except that the L_{np} operator contains solutions at two consecutive streamwise stations and requires more time to solve (for both local and global eigenvalue searches). In fact, obtaining the global spectrum with both α and $d\alpha/dx$ as unknowns is nontrivial. In LASTRAC, we assume $d\alpha/dx = 0$ for the solution of the global eigenvalue spectrum.

For the local nonparallel eigenvalue search, as mentioned previously, we may assume $\alpha_i = \alpha_{i+1}$. The solution procedure is then similar to the LST case. Alternatively, we may use a local Newton's iteration to solve for α_i and α_{i+1} simultaneously. To this end, we need two target conditions for the iteration procedure. If we use u-velocity at the wall, then the wall boundary condition $\hat{u}_i(0) = \hat{u}_{i+1}(0) = 0$ is replaced by $\hat{p}_i(0) = \hat{p}_{i+1}(0) = 1$. This condition implies that we will normalize the eigenfunctions such that the pressure eigenfunction is unity at the wall. The relaxed u-velocity condition at the wall becomes the target of the Newton iterations. In this case, we further assume that the pressure shape-function at the wall does not vary from station i to $i+1$. For most cases, the wall pressure variation $d\hat{p}/dx$ is small and the approximation given previously is valid. If $t_1 = \hat{u}_i(0)$ and $t_2 = \hat{u}_{i+1}(0)$, thus by using first-order expansion, we have

$$\begin{aligned} 0 &= t_1 + (\frac{\partial t_1}{\partial \alpha_i})\Delta\alpha_i + (\frac{\partial t_1}{\partial \alpha_{i+1}})\Delta\alpha_{i+1} \\ 0 &= t_2 + (\frac{\partial t_2}{\partial \alpha_i})\Delta\alpha_i + (\frac{\partial t_2}{\partial \alpha_{i+1}})\Delta\alpha_{i+1} \end{aligned} \tag{60}$$

where $\Delta\alpha_i = \alpha_i^{n+1} - \alpha_i^n$ and $\Delta\alpha_{i+1} = \alpha_{i+1}^{n+1} - \alpha_{i+1}^n$ denote the change of eigenvalues from iteration n to iteration $n+1$. Solution of equation (60) provides two eigenvalues: α_i and α_{i+1}. The eigenvalue iteration is repeated until $\Delta\alpha_i$ and $\Delta\alpha_{i+1}$ are both smaller than a prescribed tolerance. We have also used the wall temperature instead of the u-velocity as the target condition for Newton's iterations; numerical experiments indicate that both targets worked well. For hypersonic boundary layers, relaxing wall temperature tends to be more robust than wall velocity.

This new nonparallel eigenvalue formulation has several distinct characteristics. First, for every eigensolution, we obtain eigenfunctions at two consecutive locations. Therefore, streamwise derivatives and thus the effective nonparallel growth rates accounting for this streamwise variation may be evaluated. Second, local eigenvalue search using only one alpha ($d\alpha/dx = 0$) is more robust because the global eigenvalue solver is a good approximation to the local solver. However, due to the neglected $d\alpha/dx$, this approximation produces a small but noticeable transient effect (smaller than that produced by using the LST solutions) if we use the obtained eigenmode to initiate the PSE marching procedure. Again, this transient effect is more pronounced in supersonic than in low-speed boundary layers for 2-D flows. For crossflow instability, the variation of α near the leading edge is usually large and the one-alpha procedure yields a noticeable transient effect even at low speed. Furthermore, even though this transient effect is noticeable in the disturbance growth rate, it is much less so for the disturbance amplitude that represents the integrated effect of the growth rate. Figure 4 compares linear PSE solutions obtained by starting with the one-alpha, two-alpha, and quasi-parallel LST eigenfunctions. The solution initiated far upstream is also plotted as a baseline solution. Clearly, the two-alpha eigenfunction produces the least transient effect.

On the other hand, the two-alpha iteration procedure is less robust because it requires convergence of two eigenvalues and only a first-order Newton's method is employed. Deriving a second-order method is possible but would require introducing more target conditions. The lack of robustness also stems from inconsistency

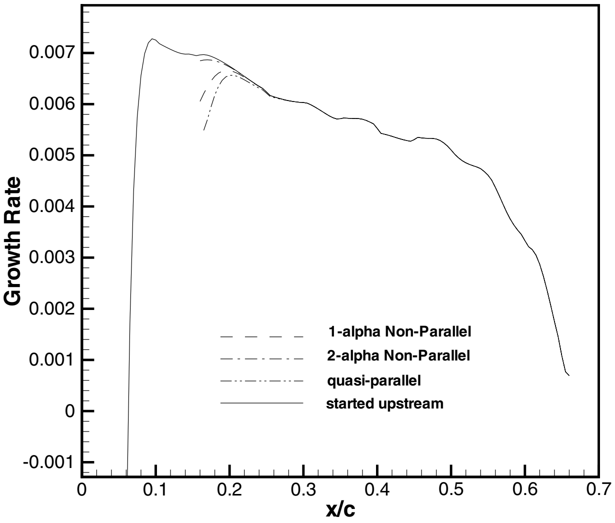

Figure 4: Linear PSE solution obtained by using three different initial eigenfunctions, along with solution initiated far upstream to assess transient effect.

between global and local eigenvalue solvers (noted that in the global solver, we assume $d\alpha/dx = 0$). The initial guess of $d\alpha/dx$ is important for the two-alpha procedure. We found that for 2-D boundary layers, using the guess from the global solver and imposing $d\alpha/dx = 0$ generally leads to a converged solution. However, for crossflow instability near the leading edge of a swept wing, providing a good guess of $d\alpha/dx$ is crucial to convergence of the eigenvalue search. LASTRAC uses several built-in $d\alpha/dx$ guesses for the two-alpha eigenvalue search. However, in some cases, users need to provide a good guess for $d\alpha/dx$ to get a converged eigenvalue. We also found that, in some cases, the eigensolution obtained with the two-alpha procedure gives smoother pressure eigenfunction in the wall normal direction.

As discussed in the previous section, the determination of α is nonunique within the nonparallel framework. The above nonparallel eigenvalue formulation typically generates more discrete eigenmodes than its quasi-parallel LST counterpart. Therefore, selecting eigenmodes of interest from the global eigenvalue spectrum becomes more difficult. Users may need to provide certain additional criteria to filter out spurious modes. Interestingly, one quasi-parallel eigenmode split into two modes in the corresponding nonparallel global spectrum, mainly because we solved two consecutive eigensolutions simultaneously and assumed that $\alpha_i = \alpha_{i+1}$. These two eigenmodes correspond to solutions at two consecutive stations and appear as the common α solutions. We found that with the two-alpha local eigenvalue search, two split modes from the global spectrum converge to the same value during the local eigenvalue search. Mathematically, the nonparallel eigenvalue formulation introduces more degrees of freedom in the system and thus allows more discrete eigenmodes (other than the continuous spectrum). The mode closely associated with the unstable mode given by the parallel LST still appears as an eigenmode in most cases. Near the leading edge of the boundary layer, instability wave formation and evolution is governed by the receptivity process; therefore, the nonparallel eigenmode may or may not be relevant from the theoretical standpoint. LASTRAC selects the most unstable mode that gives the largest growth rate at the location of interest to initiate the marching procedure. Numerical experiments indicate that the eigenvalue and eigenfunction generated by the NES (both one- and two-alpha approaches) produce a smaller transient effect than those from the quasi-parallel LST solution.

Chapter 3

Code Development

The LASTRAC code was developed from scratch. Several guiding principles were set forth in the initial stage of LASTRAC development. They are:

- Object-oriented (OO) design and implementation
- Generic programming using templates
- Optimization to avoid abstraction penalties and ensure efficiency
- Multithread and message passing interface (MPI) based parallelization
- Source code control and some configuration management

The first two items limit the choice of programming language to C++ because it is the only mature language that supports both object-oriented and generic programming. The OO paradigm has been recognized by the software world as a mainstream approach for software development. An OO software system may take longer to develop, but it is much easier to maintain and extend as the project progresses. The concepts of data abstraction, inheritance, and polymorphism are the themes of an OO language. The principle of separation of interface and implementation makes software modules more independent of one another.

The common concept for modern software development is to decompose the software into objects. Each object is derived from a software structure called class. Objects derived from the same class share a common interface, which defines the data structure and operations, usually called functions, that operating on the data structure. The clients of these objects only need to know about the interface–not the implementation details. In this way, new objects may be derived and added to the system without changing the common interface or any existing object client's code. Code reuse is thus achieved via reciting common interfaces for various class objects–not through cut-and-paste of existing codes.

The OO design procedure typically employs a formal requirement analysis along with use case studies. The Unified Modeling Language (UML) (ref. [32]) is a widely used tool for OO analysis and design. Due to the lack of resources, we did not use UML for analysis and design. Instead, the structure of LASTRAC was laid out based on numerical formulation; then, different design patterns were applied when appropriate. The design patterns used in LASTRAC came mostly from Gamma et al. (ref. [33]) with some modifications to cope with numerical efficiency. A design pattern is a set of objects and interfaces that occur over and over again in a software system; hence, it is also a solution to a software problem in a context. Many patterns proposed by Gamma et al. are used in LASTRAC. For example, the Singleton pattern is used for input and control parameters, a Factory Method is used to generate objects on the fly, and the Observer pattern is used to let the metric class, coordinate transformation class, and Jacobian matrices class update their contents when the solution marches to the next station. The Template Method and Command patterns are used to implement a multithread class that allows a nonstatic class member function to be called within a C-style thread routine.

Generic programming (ref. [34]) is a new paradigm that allows a programmer to introduce a parameterized class member or member function in a function or class definition. The introduced parameter is called a template. A routine designed this way allows unrelated classes or algorithms to be plugged in as a template parameter as long as the plugged-in classes fulfill the compilation requirement. For example, one can design a numerical integration class that allows various approximation rules be specified as a template parameter. The C++ programming language includes a powerful library called standard template library (STL) based on the generic programming concept. LASTRAC uses templates extensively in the class design. The STL containers and algorithms are used to ensure efficiency and flexibility.

3.1. LASTRAC Code Design

Discussing the details of the LASTRAC code design and structure is beyond the scope of this paper. Here, we only describe important elements of the code structure. Users of LASTRAC need to prepare two files: one contains the mean flow, and the other has the input parameters instructing which computation (LST, linear or nonlinear PSE, etc.) needs to be performed. The input parameter file is read by LASTRAC through a parser. The parser takes an input format similar to the namelist format in FORTRAN but allows C++ style comments. LASTRAC performs a "sanity check" for the input parameter file and flags errors if any user mistake is found. The mean flow reader was designed to be a hierarchy of classes that can be extended for perfect gas or reacting flow, and 2-D or 3-D through inheritance of the abstract class.

The core part of the code uses the observer pattern (see reference [33]) where two subjects, LocalSubject and MarchingSubject, perform a local eigenvalue solution and a marching PSE solution, respectively. The observers are the coordinate transformation, metric and derivatives, and Jacobian classes. These classes update their states when the LocalSubject and MarchingSubject finish their calculations and move on to the next location. The nonlinear marching subject communicates with LocalSubject and MarchingSubject to obtain a local eigenvalue for initiating the marching and to solve for each individual mode during the nonlinear iteration. It has several logistic classes, such as a Fourier container class, to perform fast Fourier transform and to handle Fourier-mode related operations.

The finite difference operation and boundary condition treatment is handled by using a template class and the generic programming paradigm. These generic classes would work for different orders of differentiation and variable types such as real or complex. The left-hand side operator of the PSE and eigenvalue solver were cast in an operator form to facilitate future maintenance.

To achieve computational efficiency in the LASTRAC code design, we tried to avoid using many fine-grained objects. The use of templates in generic programming allows substitution of template parameters at compile time; consequently, the code can be optimized for run-time performance. In LASTRAC, the template meta-programming technique is used to optimize the matrix multiplication procedure and the block matrix solvers. Extensive use of generic programming and template utilities makes LASTRAC as efficient as an earlier code written in FORTRAN 77.

Parallelization for the nonlinear PSE option in LASTRAC is implemented by using the multithread technique in conjunction with the more traditional MPI approach. The Posix thread is used in LASTRAC to guarantee portability. Multithread and MPI options are only implemented for nonlinear PSE calculations. Linear options, including LST and linear PSE, can only be run as a single-thread application. Because linear calculations are mode independent, users can launch many LASTRAC runs concurrently on a multiprocessor or clustered environment.

For nonlinear PSE calculations, each Fourier mode is solved by a thread or a process under MPI; however, the nonlinear forcing terms have to be computed collectively. Parallelization of nonlinear forcing is only performed through computing terms in physical space simultaneously. Each MPI process may spawn multiple threads. For a multiprocessor node in a cluster, this combined thread and MPI approach makes LASTRAC run more efficiently.

Source code control and configuration management are reinforced by using the public domain software CVS. Unit testing was accomplished for major modules during the initial development stage. Due to lack of resources, we do not follow unit and integration testing procedures rigorously. A semiautomated integration testing procedure is also implemented.

LASTRAC code has been ported to all major Unix platforms. In the release, we include executables for Linux Intel, SUN Solaris, and SGI. We are planning to port the code for Linux Alpha and Windows in the near future.

3.2. Code Validation

The LASTRAC code was validated by comparing its results with those obtained in the literature and by other publicly available codes. The first case used for validation was the quasi-parallel LST results published by Malik (ref. [28]). The test cases range from low-speed (incompressible) to hypersonic conditions. Linear and nonlinear PSE results were compared with those given in references [2, 23, 35]. For swept wing boundary layers, the Arizona State University test configuration (ref. [36]) was used for validation, and results were compared with Malik et al. (ref. [3]).

We also compared solutions from LASTRAC with experimental data, including the incompressible subharmonic breakdown (ref. [37]) and Stetson's Mach 8 sharp cone experiments (ref. [7]). A recent comparison with a DNS code developed by Liu & Jiang (ref. [38]) for both a flat plate and an infinite swept wing exhibits excellent agreements between PSE and DNS. Those results will be published in a separate paper.

We also compare the LASTRAC solutions with those obtained from previously developed codes such as EMALIK2D and ECLIPSE. The agreement is very good. Further validations are planned when new experimental data become available.

Chapter 4

Basic Flow Computation

LASTRAC does not include a module for basic flow computation. Users are responsible for preparing a mean flow file in a FORTRAN unformatted binary format. An interface for reading in a C style binary mean flow file will be implemented in the future release.

4.1. Binary Mean Flow File Format

The mean flow file should be written by using the FORTRAN unformatted output (assumed to be unit 10) as follows:

```
character*64 title

...
open(unit=10, file = ’meanflow.out’, form = ’unformatted’)
...
```

where `title` is a title string with a length of 64. The mean flow is then written as

```
...
write(10) title
write(10) n_station

write(10) igas, iunit, prandtl, pfs, nsp

do n = 1, n_station

  write(10) istat, npts, x, rl, Rel, xcurvt, radius, drdx
  write(10) t_e, u_e, rho_e
  write(10) (y(k), k = 1, npts)
  write(10) (u(k), k = 1, npts)
  write(10) (v(k), k = 1, npts)
  write(10) (w(k), k = 1, npts)
  write(10) (t(k), k = 1, npts)
  write(10) (p(k), k = 1, npts)
```

```
enddo
...
```

Definitions of the FORTRAN parameters are as follows:

`title`: Title of the mean flow with a length of 64 characters.

`n_station`: Total number of stations in the profiles output.

`igas`: Use 1 for perfect gas. In the future release, equilibrium and finite-rate options will be added.

`iunit`: Use 0 for imperial (U.S. customary) units and 1 for SI units. Unit only applies to dimensional quantities. The units for a length are meter and foot, for SI and imperial units, respectively. Velocity units are m/s and ft/s, temperature K and ^{o}R, and density kg/m^3 and lb$_f$ - sec^2/ft^4, respectively.

`prandtl`: Prandtl number of the mean flow.

`pfs`: Free-stream static pressure (dimensional in units defined by `iunit`). This input is ignored in the current release.

`nsp`: Number of species if non-equilibrium mean flow. Use a value of 0 for perfect gas.

`istat`: Station number for each group of profiles (for display only, not used).

`npts`: Number of points in the profiles following.

`x`: Streamwise coordinate (using a body-fitted coordinate system) of each station (dimensional). Note that for axisymmetric cones, this is the coordinate along the cone surface, not the axial coordinate in a cylindrical coordinate system.

`rl`: Local length scale used to compute `Rel` and normalize the wall normal coordinate `y(k).` For boundary layer flows, the boundary-layer length scale defined by equation (5) is usually used. For jet flow, the jet diameter (a constant value) may be used. Note that this length scale will be used by LASTRAC to normalize the growth rate and wave number output; therefore, the choice should be such that the normalized growth rate value falls in the default growth rate range (described in the next chapter). If not, the growth rate filters should be adjusted accordingly.

`Rel`: Local Reynolds number based on `rl` (see eq. (6)).

`xcurvt`: Dimensional streamwise curvature (inverse of the radius of curvature).

`radius`: Local dimensional transverse cone radius (in cylindrical coordinates).

`drdx`: Nondimensional local derivatives of cylindrical radius with respect to the streamwise coordinate (`x`). For an axisymmetric cone, this value would be $sin\theta$ where θ is the local cone half angle. If the mean flow is for an infinite-swept boundary layer, input x/c (where x is the Cartesian coordinate and c is the chord length) instead of radius. This x/c input is not used for calculations but will be written to various output files as a location indicator.

t_e: Dimensional local boundary-layer edge temperature and the value used to normalize the temperature profile.

u_e: Dimensional local boundary-layer edge velocity along the body surface and the velocity scale used to normalize all three velocity profiles. For a jet flow, this value could be the centerline velocity. For an infinite swept-wing boundary layer, this velocity is along the normal chord direction and diminishes to a value of zero at the attachment line.

rho_e: Dimensional local boundary-layer edge density and the value used to normalize the pressure profile.

y(k): Nondimensional wall-normal coordinate (normalized by rl).

u(k): Nondimensional streamwise velocity normalized by u_e. For infinite swept-wing boundary layers, this is the velocity component along the normal-chord (normal to the leading edge) direction.

v(k): Nondimensional wall-normal velocity normalized by u_e.

w(k): Nondimensional spanwise velocity normalized by u_e. For infinite swept-wing boundary layers, this is the velocity component along the spanwise (attachment line) direction.

t(k): Nondimensional temperature normalized by t_e.

p(k): Nondimensional pressure normalized by rho_e * u_e^2.

4.2. Boundary-Layer Codes

Two publicly available boundary layer codes have been used to generate most of the test cases described in the following chapters. The output interface to generate a mean flow file according to the format described in the previous section has been implemented in these codes.

For general 2-D or axisymmetric boundary layers, we use the bl2d code written by Harris & Blanchard (ref. [39]). The bl2d code employs the Levy-Lees transformation and computes the solution on the transformed plane by using a second-order finite difference method. Both planar 2-D and axisymmetric compressible flow boundary layers can be computed.

For infinite swept-wing boundary layers, we have tested the wingbl2 code by Pruett (ref. [40]). It is a compressible boundary layer code written specifically for infinite swept wing flows. Governing equations are also transformed by using the Levy-Lees transformation. A spectral method is used to solve the equations; therefore, this code is, in general, less efficient than the bl2d code mentioned above. Code validation was performed for several configurations by comparing results with other swept-wing boundary-layer codes such as the BLSTA code (ref. [41]).

Chapter 5

Input and Output Description

To run LASTRAC, users must provide an input file in addition to the aforementioned basic flow file. The input file consists of several lines in the form of:

`parameter1 = value1`, `parameter2 = value2` ...,

A white space or a comma is used to separate tokens. C++ comments such as "//" are allowed anywhere in the input file. LASTRAC will ignore anything following "//" until the end of the line. Parameters are not case sensitive. The parameter and value pair can be in different lines as long as they appear consecutively. For array-type parameters, multiple values separated by comma or space can be used, and a wild card is also allowed. Some sanity check is provided for the parameters specified. If important errors are found, the code will flag the error and stop. For less important mistakes, the code simply flags a warning message, uses the default value, and then continues.

Most of the input parameters have a default value. Users do not need to specify the parameter value if the default value is to be used. For a numeric parameter, the value is a number. For a boolean parameter, the value is either true or false. If an input value is of a complex number type, use the format $(n1, n2)$ in the input file. The string type input should be contained in a double quote (e.g., `''eigenfunction.dat''`). An input parameter may be specified more than once. The parameter specified later in the input file overwrites the previous one.

5.1. Input Parameters

We divide the input parameters into several groups according to their functionality. The first group of input parameters include the major input that is necessary for every run.

flow_type: Identifies the type of flow to be computed. There are three options: `bound_layer`, `jet_vortex`, and `shear_layer`. The first option (`bound_layer`) is for a boundary-layer mean flow that has a wall boundary at $y(0) = 0$ and a maximum local streamwise velocity at $y(n) = y_{max}$ in the mean flow profile. The `bound_layer` mean flow could be either two-dimensional or axisymmetric. A `jet_vortex` mean flow is the one that has a local maximum streamwise velocity at $y(0) = 0$ and zero velocity in the free-stream. It can be either a planar or axisymmetric jet. A vortex flow is similar to a jet except that the mean azimuthal velocity component is non-zero. The `shear_layer` option assumes that the local maximum and minimum streamwise (either positive or negative) velocities are at either end of the domain. The default value is `flow_type = bound_layer`.

mflow_filename: The mean flow file name with a path relative to the location of the LASTRAC executable. Using the absolute path is recommended. The file name must be included in the double quote (e.g., `mflow_filename` = `‘‘/h/u/case1.mflow’’`). No default value is assumed.

nonl_pse_calc: A boolean parameter to determine whether a nonlinear PSE will be run or not. It takes a value of `true` or `false`. If `true`, the nonlinear PSE parameters described later should also be included. The default is `nonl_pse_calc = false`.

solution_type: Two stability solution methods are available. The first one, `solution_type = local_eig_solution`, performs the quasi-parallel or nonparallel local eigenvalue search at each mean flow station specified by the user. N-factor integration is performed whenever an unstable eigenmode is found. The second method, `marching_pse_solution`, starts with a similar nonparallel eigenvalue search for each mean flow station. When an unstable mode is found, LASTRAC switches to the linear PSE marching process and marches downstream to the end location specified by the user. No default value.

marching_method_2d: Users can choose from three marching methods constructed by different streamwise grids. The option `along_station` marches from the `init_station` station until the station specified in `final_station` for every station read in from the mean flow file. If a local eigenvalue calculation is requested, users may use `step_station` to skip stations in the marching process. For PSE marching, the `step_station` input is ignored and no station skip is allowed. The option `along_xc` marches the solution downstream from $x =$ `init_xc` until $x =$ `final_xc` with a step size of `step_xc`. Similarly, the option `along_re` (`re` is based on the Reynolds number defined in equation (6) marches the solution from `init_re` until `final_re` with a step size of `step_re`. For the latter two options, mean flow quantities and profiles are interpolated from the station values read in from the mean flow file. For all three options, the marching step may be negative if `solution_type = local_eig_solution`. This allows both forward and backward marching to be performed for a local eigenvalue analysis. Default is `along_station`.

init_station: Initial marching station for `marching_method_2d = along_station`. No default value.

final_station: Final marching station for `marching_method_2d = along_station`. No default value.

step_station: Stability solution is performed for every `step_station` when the parameter `marching_method_2d` is equal to `along_station` and a local eigenvalue calculation is intended. Default is `step_station` = 1.

init_xc: Initial x coordinate in meters for `marching_method_2d = along_xc`. No default value.

final_xc: Final x coordinate in meters for `marching_method_2d = along_xc`. No default value.

step_xc: Marching step size in meters for `marching_method_2d = along_xc`. The value can be positive or negative. No default value.

init_re: Initial Reynolds number for `marching_method_2d = along_re`. No default value.

final_re: Final Reynolds number for `marching_method_2d = along_re`. No default value.

step_re: Marching step size in Reynolds number if `marching_method_2d = along_re`. The value can be positive or negative. No default value.

freq_unit: Determines the unit of frequency or temporal wave number ω being inputted. If `freq_unit = non_dim_freq`, the value specified for the parameter `freq` is the nondimensional wave number ω defined in equation (11). If `freq_unit = capf_freq`, `freq` is a nondimensional frequency defined in equation (12) and for `freq_unit = in_hertz_freq`, `freq` is in Hertz. Default is `freq_unit = non_dim_freq`.

beta_unit: Determines the unit of spanwise wave number being inputted in the parameter `beta`. For a nondimensional spanwise wave number as defined in equation (13), `beta_unit = non_dim_beta` is used. The option `in_mm_beta` is used when the inputting `beta` represents the spanwise wavelength in mm. The second option `nwave_axisym_beta` is used for `beta` to represent the integer azimuthal wave number n in an axisymmetric boundary layer or jet. The corresponding nondimensional wave number is then $\beta = n\ell/r_b$ where r_b is the local cylindrical radius and ℓ is the length scale. For axisymmetric configurations, the use of an integer azimuthal wave number is more physical than an arbitrary real wave number and thus is recommended.

The previous three options for `beta_unit` are for a distinctive spanwise wave number. For cases where an optimized (maximum growth rate with respect to β) wave number is to be found, users do not provide the spanwise wave number. Instead, other information such as wave angle is specified and the code will attempt to find the spanwise wave number that has a maximized growth rate. For 2-D boundary layers, we enter `beta_unit = wave_angle_beta` and provide a guess of wave angle with respect to the mean flow direction in `beta`. A good guess for the oblique first mode in compressible boundary layer is around 60^o.

This guess need not be very accurate because LASTRAC will attempt to find instability modes around the value provided. For infinite swept wing boundary layers, we enter `beta_unit = l_over_d_beta` and provide a guess for the total wavelength to boundary layer thickness ratio(λ/δ) in `beta` and a guess of wave angle in `wave_angle`. This λ/δ ratio is about 4 (and a wave angle around 87^o) for stationary crossflow instability mode and increases to about 12 or larger (and a smaller, or even negative wave angle) for traveling crossflow waves. Based on both guesses, LASTRAC would attempt to find a spanwise wave number that gives a maximum growth rate at a given location. Therefore, the last two options are ideal when the parameter range for flow instability is sought. Default is `beta_unit = non_dim_beta`.

freq: Disturbance frequency or temporal wave number, depending on the input value of `freq_unit`. Users may specify either a single value or an array of values for `freq`. Users may use the character "*" to represent successive appearances of a value, e.g., `freq = 1.e-4, 5* 0.002, 100`. Number of `freq` values must match that of the `beta` values. No default value.

beta: Spanwise wave number, wavelength, wave angle, or the value of λ/δ, depending on `beta_unit`. Similar to `freq`, users may specify an array of values to be computed in `beta`. Users may use the character "*" to represent successive appearances of a value, e.g. `beta = 10*0.05, 0.05`. The number of `beta` values must match that of the `freq` values. No default value.

wave_angle: An array of wave angles in degrees corresponding to the `freq` and `beta` pair. The wave angle is the angle with respect to the free-stream mean flow direction. Users only need to specify wave angles when `beta_unit = l_over_d_beta`. A good guess for a stationary crossflow mode is around 87 degree. Traveling crossflow modes can have a smaller or even negative wave angle when crossflow reversal occurs. No default value is assumed.

qp_approx: A boolean to determine whether to apply quasi-parallel LST approximation at each marching station. This parameter only affects the local eigenvalue search mode (`solution_type = local_eig_solution`). For `solution_type = marching_pse_solution`, this parameter is ignored. Default is `false`.

The second group of input parameters is associated with the stability grid and other geometrical values.

grid_type: Four types of wall-normal stability grid distribution may be selected (see fig.3). The `wall_cluster` option clusters the grid points near the wall according to equation (43). The `dual_cluster` option constructs the grid in two zones and clusters the points both near the wall and the critical layer by using equation (44). For hypersonic boundary layers, disturbances peak at the critical layer located near the boundary-layer edge. The `dual_cluster` option is necessary to properly resolve the disturbance eigenfunction. For low-speed flows, the `wall_cluster` option suffices. For jets or vortex flows, users may choose to cluster the grid only near the critical layer with the `critl_cluster` option. This option uses a hyperbolic tangent mapping suggested in reference [29]. The `shock_grid` option has a two-zone clustering similar to `dual_cluster` but it has a uniform grid in the free-stream with an increment specified in the parameter `fstrm_incr`. If `grid_type = critl_cluster`, users may need to provide `tau_s` to further control the grid stretching. For `grid_type = shock_grid`, users may need to provide additional input parameters such as `base_grid_type`, `yshock_bound`, `fstrm_incr`, or `use_shockgrid_global`. Default is `dual_cluster`.

tau_s: Stretching parameter used when `grid_type = critl_cluster` (see eq. (5-221) in ref. [29]). The larger the value is, the stronger the clustering will be. Default is `20`.

base_grid_type: Determines the base grid type for `grid_type = shock_grid`. Two choices can be made, `dual_cluster` or `critl_cluster`, for a boundary layer or a shear layer, respectively. Default is `base_grid_type = dual_cluster`.

yshock_bound: Defines the domain size for stability calculations when the parameter `grid_type` is `shock_grid`. If the input value is zero, the local y_{max} value read in from the mean flow file (which may vary along x) will be used. Non-zero values specified here will be regarded as y_{max}. Default is 0.

fstrm_incr: Nondimensional constant free-stream increment used for the shock grid when the parameter `grid_type` is `shock_grid`. Default value is 1.

use_shockgrid_global: Boolean to determine whether to use the shock grid for global eigenvalue search. This parameter is used only when the parameter `grid_type` is `shock_grid`. Default is `false`.

ymax: Maximum wall-normal coordinate for any stability calculations except the global eigenvalue search. The input is a nondimensional value normalized by the local boundary layer length scale ℓ defined in equation (5) or the length scale specified by the user in the mean flow file. Default is 200.

ymax_glob_search: Maximum wall-normal coordinate for the global eigenvalue search. For certain hypersonic flows, reducing this value makes nonparallel eigenvalue spectrum cleaner and thus easier to select unstable modes. Default is 60.

num_normal_pts: Number of wall-normal points for local (parallel or nonparallel) eigenvalue search and linear or nonlinear PSE calculations. A value of 101 is sufficient for most cases. For higher resolution of the eigenfunction, a larger value can be used. Default is 81.

num_normal_pts_geig: Number of wall-normal points for the global eigenvalue search. This value can be different from `num_normal_pts`. Increasing this value increases the computational time. A value of 41 is sufficient for most cases. When the mean flow profile is under-resolved, we may need to increase this value to 61 or 81. Default is 41.

strm_curvt: Boolean to determine whether to include streamwise curvature in the stability calculations. Default is `false`.

transv_curvt: Boolean to determine whether to include transverse curvature for axisymmetric geometries in the stability calculations. Default is `false`.

strm_order: Determines the order of accuracy along the marching direction. Two choices are available: `first_order` or `second_order`.

wall_normal_order: Determines the order of accuracy along the wall-normal direction. Two choices are available: `second_order` or `fourth_order`.

The following parameters are associated with the input/output, boundary conditions and physical properties of the calculation. For all stability calculations, the no-slip condition (eq. (38)) is used at the wall boundary. In the far field, three boundary conditions (Dirichlet, equation (39), nonreflecting, equation (40), or the Rankine-Hugoniot shock condition) may be chosen. The parameters `fs_mach, fs_pr, fs_temp, shock_angle,` and `init_shock_disp` are used only if a shock boundary condition is chosen.

farfield_bc: Three far-field boundary conditions are available; `dirichlet, non_reflect,` and `rh_shock`. Default is `farfield_bc = non_reflect`.

fs_mach: Free-stream Mach number used for `farfield_bc = rh_shock`. No default value.

fs_pr: Free-stream pressure used for `farfield_bc = rh_shock` (in N/m^2). No default value.

fs_temp: Free-stream temperature used for `farfield_bc = rh_shock` (in K). No default value.

shock_angle: Constant shock angle with respect to the surface for `farfield_bc = rh_shock`. This parameter is used only if a constant shock angle exists, otherwise the read-in mean flow boundary is used to compute the local shock angle. No default value.

init_shock_disp: Initial shock perturbation displacement (a complex number) at the starting location of the PSE marching. Usually a zero value is appropriate. Default is `(0, 0)`.

variable_Prandtl: Boolean to determine whether to use variable Prandtl number in the stability calculations or not. The values of Cp and k will be computed using a curve-fit function. Default is `false`.

gamma: The specific heat ratio. Default is `1.4`.

stokes: The Stokes parameter as defined in equation (3). Default is 0 (Stokes' hypothesis).

mflow_storage_type: There are two ways to internally store the read-in mean flow in LASTRAC. The `memory_storage` option will read in all mean flow profiles and store them in physical memory for future use. The `file_storage` option scans through the mean flow file and stores the file position index database for each mean flow station. The mean flow profiles then are extracted on the fly when necessary. The latter option is useful when the mean flow file is large. To output the mean profiles read in for debugging, the `memory_storage` option must be used. Default is `file_storage`.

use_extrap_mprof: Stability calculations typically employ a domain larger than that used in meanflow calculations. Therefore, extrapolation is necessary in the free-stream. This parameter controls whether to use linear extrapolation based on the data near the boundary of the mean flow calculations. While extrapolation generally works well, it must be used with caution when Navier-Stokes codes are used to generate the meanflow. The Navier-Stokes meanflow typically has a non-vanishing derivative at the boundary. When extrapolated to a large y value in the stability grid, the free stream values may become unphysical. In this case, we need to turn off the extrapolation and freeze the free-stream value at the domain boundary of the mean flow calculation. Default is `true`.

output_eigenfunction: Boolean to determine whether to write eigenfunction output for stability calculations or not. This parameter applies to both options of `solution_type`. Default is `false`.

print_rhou_eigf: Boolean to determine whether to print the mass flow rate instead of the pressure eigenfunction in the eigenfunction output file. This parameter is applicable when `output_eigenfunction = true`. If false, the pressure eigenfunction will be printed instead of the mass flow rate. This parameter applies to both linear and nonlinear runs. Default is `false`.

pns_approx: Determines whether the PNS approximation (eq. (26)) will be applied to the streamwise disturbance pressure gradient. For small marching step size, the PNS approximation is necessary to maintain numerical stability. Default is `false`.

pns_sigma: The coefficient σ_p in the Vigneron's PNS approximation, equation (27). To completely neglect the pressure shape-function gradient, this value is set to zero. Default value is 1.

The following parameters are related to the eigenvalue calculation or initial conditions of the PSE marching. For a local eigenvalue search, one of the wall conditions must be replaced with the normalized pressure to facilitate the Newton's iteration. Usually, wall temperature or streamwise velocity component is replaced and used as the target of Newton's iteration. Alternatively, an interior equation near the critical layer can be used as a target. These options are provided through `relax_type`.

global_solver: Two global eigenvalue search methods are provided. The `use_cmpxlr` option employs a built-in eigenvalue routine using an LR algorithm. The `use_lapack` option calculates the global eigenvalue spectrum by calling the CGEEV routine in the LAPACK. Default is `use_cmpxlr`.

relax_type: Determines how the wall boundary conditions are treated in a local eigenvalue search. Three choices are available: `wall_uvel, wall_temp,` and `critlayer`. For low-speed flows, the option `wall_uvel` works well. For hypersonic flow, `wall_temp` is more robust. To track stable modes, `critlayer` generally works better. For jets or shear layers, there is no wall and the `critlayer` option must be used. Default is `relax_type = wall_temp`.

alpha_init:

This parameter is only used for linear PSE marching. If this parameter is left unspecified, a global eigenvalue search will be performed until a good unstable mode is found downstream of the initial location. Otherwise, the value provided by `alpha_init` will be used as an initial guess in the local eigenvalue search. If the local eigenvalue search does not converge, the global search will start automatically. The input value is of a complex type. No default value.

read_in_eigf:

For PSE calculations, the initial eigenfunction (or shape-functions in the context of PSE) can be generated by a local nonparallel eigenvalue search as discussed in the previous section, or users can provide it from external sources through reading a file. This parameter determines whether to read in eigenfunction from a file. If true, the file name must be provided by `eigf_file_name`, and the number of profiles to be read should be given by the parameter `num_eigf_readin`. Default is `false`.

num_eigf_readin:

Determines how many eigenfunction profiles are to be read in the file given by `eigf_file_name`.

eigf_file_name:

File name of the eigenfunction file (specified between quotation marks, e.g. `"eigfunc.dat"`) to be used when `read_in_eigf = true`. This file constains columns of data in the order of y, $\hat{p}$, $\hat{u}$, $\hat{v}$, $\hat{w}$, $\hat{T}$ for perfect gas (y is real and the rest are complex numbers). In this case, `num_eigf_readin = 5`. No default value.

update_alpha:

Boolean to determine whether to update streamwise wave number α during the linear PSE marching. Default is `true`.

use_l2alpha:

Boolean to determine whether to use the local two-alpha approach for nonparallel eigenvalue search by using equation (60). The two-alpha iteration needs a guess for $d\alpha/dx$. LASTRAC will perform this two-alpha search for $d\alpha/dx = (0,0)$ and $d\alpha/dx = (1.e-5, 1.e-5)$ repeatedly at every location where a local nonparallel search is requested for a 2-D or axisymmetric boundary layer. If crossflow exists as in an infinite swept-wing boundary layer, eight guesses of $d\alpha/dx$ will be tried: $(0,0)$, $(2.e-4, 1.e-5)$, $(1.e-4, 1.e-5)$, $(4.e-4, 1.e-5)$, $(6.e-4, 1.e-5)$, $(8.e-4, 1.e-5)$, $(1.e-3, 1.e-5)$, $(2.e-3, 1.e-5)$. Numerical experiments indicate that for 2-D or axisymmetric boundary layers, setting $d\alpha/dx$ to zero in general works well for almost all cases. For crossflow instability cases, the guesses attempted by LASTRAC give satisfactory solutions except near the leading edge where α varies rapidly. If LASTRAC fails to find a good unstable mode near the leading edge, users have two options. First, users can turn off two-alpha iteration and revert to the single-alpha iteration (`use_l2alpha =`

`false`). Alternatively, a better guess for $d\alpha/dx$ can be specified in `dadx` or a shift can be applied to the internal $d\alpha/dx$ values using `dadx_shift`. These parameters (`use_l2alpha`, `dadx`, and `dadx_shift`) should be disregarded in a quasi-parallel LST eigenvalue search. Default is `true`.

dadx: Provides a guess of $d\alpha/dx$ to perform a local nonparallel two-alpha eigenvalue search. If `dadx` is specified in the input file and not equal to $(0, 0)$, then the internally stored guesses for $d\alpha/dx$ are ignored and only this $d\alpha/dx$ value will be used to search for nonparallel eigenvalues. Therefore, users supply `dadx` only if the internal guesses fail to identify an expected unstable mode, or if only one value of $d\alpha/dx$ is intended. The value provided should be a complex number even though the real part is more important than the imaginary part (setting imaginary part to $1.e - 5$ usually suffices). Default value is $(0, 0)$.

dadx_shift: Shift to the internally stored guesses of $d\alpha/dx$ as described for the parameter `use_l2alpha`. The value provided should be a complex number. Default value is $(0, 0)$.

lod_max: Determines the maximum value of allowable λ/δ (where λ is the total wavelength of the disturbance and δ is the boundary layer thickness) for the local eigenvalue search. Any converged mode during the eigenvalue search with a λ/δ exceeding this value will be regarded as a spurious mode. Normally, the λ/δ value for instability wave is less than 20, except very near the leading edge where the boundary layer thickness is small and a transient instability mode may have a λ/δ value of around 30–40. Default is `40`.

cr_max: Determines the maximum allowable phase speed for the selection of eigenmodes from the global eigenvalue spectrum. All eigenmodes from the global eigenvalue spectrum with a phase speed higher than this value will be filtered out. Default value is `0.99`.

cr_min: Determines the minimum allowable phase speed for the selection of eigenmodes from the global eigenvalue spectrum. All eigenmodes from the global eigenvalue spectrum with a phase speed lower than this value will be filtered out. Default value is `1.e-4`.

wave_ang_max: Determines the maximum allowable wave angle with respect to the local free-stream mean flow direction for the selection of eigenmodes from both the global eigenvalue spectrum and the local eigenvalue search. All eigenmodes with a wave angle larger than this value will be filtered out. Default value is `90`.

wave_ang_min: Determines the minimum allowable wave angle with respect to the local free-stream mean flow direction for the selection of eigenmodes from both the global eigenvalue spectrum and the local eigenvalue search. All eigenmodes with a wave angle less than this value will be filtered out. Default value is `-90`.

alpha_i_max: Determines the value of a maximum acceptable growth rate. The eigenvalue search sometimes generates unphysical spurious modes, and this parameter filters out modes that have very large growth rates, such as the upstream propagating mode. This filter applies to both global and local eigenvalue searches. Users may need to increase this value if the local length scale provided in the mean flow file is such that the physical unstable mode may have a growth rate larger than the default value. Default is 0.1.

np_growth_rate_min: Determines the value of a minimum acceptable growth rate during the local eigenvalue search. The input value is not used to filter modes from the global spectrum. This parameter applies to both parallel and nonparallel eigenvalue searches. A negative value defines the lower bound of a stable mode being tracked. Therefore, to track a mode deeply into the stable region, users need to decrease this value (make it more negative). Default is $-1.e-4$.

wall_dpdy_ratio_min: A spurious mode may occasionally be identified by using the pressure eigenfunction derivative near the wall. This parameter determines the lower bound of the ratio (rate of change) in the wall pressure eigenfunction for an acceptable eigenmode. Lowering this value allows more modes to be included. In cases where the mean flow is poorly resolved, the near wall solution is often noisy; using this parameter may help filter out spurious modes. However, it may also filter out a good unstable mode. Therefore, users must use this parameter with caution. Default value is $1.e-4$.

delta_for_eig_iteration: Determines $\Delta\alpha$ used for computing derivatives of the target with respect to α during the local eigenvalue search. Default value is $1.e-5$.

local_eig_convg_tolerance: Convergence criterion for local eigenvalue search. Default value is $1.e-8$.

march_eig_convg_tolerance: Convergence criterion for the PSE marching process. Default value is $1.e-8$.

max_value_eig_correction: During the local eigenvalue search, if the correction of the eigenvalue in each Newton's iteration exceeds this specified value, the search process will stop and a non-convergence is flagged. Default value is `1`.

num_max_eig_iteration: Determines the maximum number of iterations to be employed for Newton's iteration in the local eigenvalue search. Default is `30`.

The following parameters are related to nonlinear PSE calculations and are ignored if `nonl_pse_calc = false`. The Fourier series of the disturbance is defined in equation (10). The fundamental frequency and spanwise wave number are determined by the first pair of `freq` and `beta`, and the corresponding values of `freq_unit` and `beta_unit` as in the linear PSE calculations.

use_mthread: Boolean to determine whether LASTRAC will be run in the multi-threaded mode. For the MPI version of LASTRAC, multi-threaded execution may be turned on or off for each process by using this parameter. Default is `false`.

nthread: Determines total maximum number of concurrent threads to be spawned for nonlinear PSE calculations. The optimum number depends on the system configuration and is usually more than the number of central processing units (CPU) available. Default value is `64`.

n_mode_omega: Specifies the number of temporal modes used (M in the Fourier series defined in equation (10). Note that this number is not the total number of Fourier modes used and can be any nonnegative integer. The fundamental wave number is determined by the first element in the `freq` array and its associated `freq_unit` value. Default is `0`.

n_mode_beta: Specifies the number of spanwise modes used (N in the Fourier series defined in equation (10). Note that this number is not the total number of Fourier modes used and can be any nonnegative integer. The fundamental wave number is determined by the first element in the `beta` array and its associated `beta_unit` value. Default is `0`.

mode_symm: Determines the symmetry condition in the wave number space (ω, β). The option `OMEGA_BETA_SYMM`, valid for 2-D and axisymmetric boundary layers where no crossflow is present in the mean flow, assumes symmetry conditions along both the ω and the β axes (thus, only one quadrant, $0 \leq m \leq M$ and $0 \leq n \leq N$, is calculated). The other option `BETA_SYMM_ONLY`, assumes a symmetry condition only along the ω axis. Therefore, two quadrants in the wave number space ($-M \leq m \leq M$ and $0 \leq n \leq N$) are calculated. This latter option is used for crossflow instability. Default is `OMEGA_BETA_SYMM`.

n_init_modes: Specifies the number of Fourier modes to be initiated in the nonlinear PSE calculation. For each excited mode, users must provide a value for α_{mn} in `nonl_alpha_init`, an amplitude in `nonl_amp_init`, and a Fourier mode number in `nonl_mode_init`. No default value.

nonl_alpha_init: This input array of complex numbers provides an initial guess of the value of α_{mn} for all the excited modes. LASTRAC takes the input value and performs a local eigenvalue search to find a converged eigenmode and then uses that mode eigenfunction and eigenvalue to initiate the corresponding Fourier mode. Users must provide at least one good eigenmode initially. For the rest of the excited Fourier modes, a $(0, 0)$ value may be specified in this array if a converged eigenmode is not expected or difficult to find. The code will then scale the initial value of α_{mn} based on the one good eigenmode provided. To avoid transient effect, users should always provide a good guess from a linear calculation. No default value.

nonl_dadx_init: This complex array provides initial guesses of the value of $d\alpha/dx$ for the default nonparallel two-alpha local eigenvalue search that will be needed to initialize the eigenmode. If the single-alpha approach is chosen (`use_l2alpha = false`), users don't need to input this array. Default is 0 for all elements.

nonl_amp_init:

This complex array provides initial amplitudes of all excited modes. The use of a complex value here enables a phase shift among the excited modes. These amplitudes are multiplied by the eigenfunctions to determine the absolute amplitude of the disturbance. For user-supplied eigenfunctions, the total disturbance amplitude thus is determined by the product of this value and the eigenfunction. The eigenfunction generated internally is normalized by the peak streamwise velocity disturbance for a mean flow Mach number less than 3, and by the peak temperature disturbance otherwise. Therefore, the maximum streamwise velocity or temperature disturbance amplitude is the value specified in this parameter, depending on the flow Mach number. The amplitude value read in here is further scaled by the `nonl_amp_vscale` defined below. No default value.

nonl_amp_vscale:

The dimensional velocity scale (m/s) used to scale the initial amplitude of the excited mode. If left unspecified, a value of 1 is used internally for 2-D/axisymmetric boundary layers, and the total free-stream velocity at the initial station is used for infinite swept wing boundary layers.

nonl_mode_init:

This complex integer array provides the mode number of each excited mode. For example, in a subharmonic breakdown simulation with only two modes, users may specify `nonl_mode_init = (2, 0), (1, 1)`. No default value.

nonl_eigfn_file:

If the initial eigenfunction of the excited Fourier modes is not of the eigenmode type (e.g., obtained from other codes), users may specify the eigenmodes in a separate file by using this parameter. The input file name must be inside the quotes (e.g., `nonl_eigfn_file = ‘‘nonl_eigf.dat’’`). This ASCII file has the following format:

```
npts1 m1 n1
y p_r p_i u_r u_i v_r v_i w_r w_i T_r T_i
...
...total of npts1 lines
...
npts2 m2 n2
y p_r p_i u_r u_i v_r v_i w_r w_i T_r T_i
...
...total of npts2 lines
...
npts3 m3 n3
y p_r p_i u_r u_i v_r v_i w_r w_i T_r T_i
...
...total of npts3 lines
...
```

where `nptsi, mi,` and `ni` represent the number of points, and the mode number for the *i*th mode inputted. For each mode, there should be `nptsi` lines of wall-normal coordinate (`y`) and complex eigenfunction values. In addition to this file, users should still specify amplitudes and values of α_{mn}. No default value.

max_nonl_iteration: Determines the maximum number of iterations for nonlinear iteration. Default is `30`.

nonl_alpha_scaling_mode_t: Mode number of the Fourier mode in the spectrum used to scale the value of α_{mn} for all non-stationary modes (i.e.. a mode (m, n) such that $m \neq 0$) used in the calculations. This parameter is necessary because for non-excited modes with a zero initial eigenfunction, strong transient effects may prevent the wave number α_{mn} from being determined by the normalization procedure given in equation (28). Scaling these modes by using the α value of a more dominant mode does not incur any inaccuracy provided that the amplitude of the modes being scaled is small. For instance, a value of `(2, 0)` can be used in a subharmonic breakdown simulation. Default value is $(1, 0)$.

nonl_alpha_scaling_mode_s: Mode number of the Fourier mode in the spectrum used to scale the value of α_{mn} for all stationary modes (i.e. a mode $(0, n)$) used in the calculations. This parameter is necessary because for non-excited modes with a zero initial eigenfunction, strong transient effect may prevent the wave number α_{mn} from being determined by the normalization procedure given in equation (28). Scaling these modes using the α value of a more dominant mode does not incur any inaccuracy provided that the amplitude of the modes being scaled is small. For instance, `nonl_alpha_scaling_mode_s = (0, 1)` can be used in a stationary crossflow instability simulation. Default value is $(0, 1)$.

no_nonl_alpha_scaling: Boolean, when set true, forces the nonlinear calculation to bypass the scaling of α_{mn} according the primary modes and update α_{mn} for all Fourier modes according to equation (28). This option is not to be used initially due to the strong transient effects for non-initialized modes. However, after all Fourier modes settle down in the modal structure, updating α_{mn} for every Fourier mode sometimes provides more accurate solutions when the disturbance amplitudes for all participating modes are reasonably large. Default is `false`.

nonl_inc_mfd_jac: Boolean to determine whether to include the mean flow distortion mode $(0, 0)$ on the left-hand side operator of the governing equations (eq. (34)). At moderately to highly nonlinear stages, putting the $(0, 0)$ mode on the left-hand side may increase the diagonal dominance of the linear operator and thus enhance numerical stability for nonlinear iterations. For most calculations, selecting `nonl_inc_mfd_jac = true` is helpful. Default is `false`.

nonl_relax_coef: Under-relaxation factor for nonlinear iterations. Lowering this factor may help numerically stabilize nonlinear iterations when the disturbance field reaches the moderately to highly nonlinear stage. Default is 1.

nonl_convg_tolerance: Determines the tolerance of the L2 norm of the changes of shape-functions during nonlinear iterations. Lowering this value increases the number of nonlinear iterations required at each marching station; thus, it's more costly. Because the L2 norm is evaluated by using normalized quantities, a tolerance of around $1.e-3$ or $1.e-4$ is sufficient. Default is $1.e-6$.

nonl_dealiasing: The nonlinear forcing on the right-hand side equation (34) is evaluated in the physical space and then transformed back to the wave number space. De-aliasing can be performed during this process. In LASTRAC, we pad one half of modes with zero eigenfunctions in both temporal and spanwise directions before transforming to the physical space. Only the modes kept in the Fourier spectrum are used after transforming back to the wave number space. When only a small number of modes are kept in the nonlinear calculations, this padding process keeps the spectrum from being contaminated due to the aliasing error that arises when the tail of the spectrum becomes energetic. The parameter `nonl_dealiasing` is a boolean to determine whether de-aliasing will be performed. Default is `false`.

nonl_restart: Boolean to determine whether a run is a restart for nonlinear PSE calculations. When LASTRAC runs, a restart file is repeatedly written to a binary file (called `Nonl_restart.dat`, if the `-t` option is not used in the command line) for every marching station. When `nonl_restart` is turned on, the code will read in the restart file and continue nonlinear iteration at the location where last run was stopped. Default is `false`.

nonl_restart_fn: A string contains the name of the nonlinear restart file to be read in when the parameter `nonl_restart = true`. The file path can be either absolute or relative to the LASTRAC executable. If the `-t` command line option is not used, the default restart file name is `''Nonl_restart.dat''`. In this case, users don't need to specify the file name in this parameter. Default is `''Nonl_restart.dat''`.

5.2. Command-line Options

To execute LASTRAC, the following command is issued:

```
lastrac [-hvdt] input_file
```

Several command line options are available for LASTRAC. The `-v` option prints the LASTRAC version number and the build date and then exits. The `-d` option is for debugging. With this option, LASTRAC will print the mean flow information to the standard output. Detailed history of the eigenvalue iterations is also printed. In addition, the mean flow profiles as interpolated to the stability grid will be printed in the file `mprofile_interpd.dat`. If the input parameter `mflow_storage_type = memory_storage`, then the `-d` option also prints all the mean flow profiles read in from the unformatted binary file in a file called `mprofile_readin.dat`. Both mean flow profile files are written in Tecplot format and may be plotted for visual inspection. If a global eigenvalue search is involved in the calculations, the `-d` option also prints the global eigenvalue spectrum in a file called Global_eig_spectrum.dat. The `-h` option prints help information on the standard output. The `-t` option is used to specify a file name “tag” for all output files. We will discussed this option in the next section.

5.3. Output File Formats

Most of output files of LASTRAC are written in Tecplot format. By default, all output files are overwritten when a new LASTRAC run is executed. Users must back up all output files before a new run is issued. Meanwhile, users may specify a file name tag by using the `-t` command line option. The output files then will be tagged according to the file tag provided in the command line. Users can use different tags for different runs to avoid the output files being overwritten. For example, a run with this command

```
lastrac -t .run1234 m0.in
```

would run LASTRAC using an input script file `m0.in` and a file tag `.run1234`. For a linear run, the output files are then `nfact.dat.run1234`, `npgrowth.dat.run1234`, etc.; for a nonlinear run, this command generates files such as `Nonl_u_amp.dat.run1234''` and `Nonl_wallshear.dat.run1234`.

The tag name following the `-t` option may start with a period as in the previous example and this tag name is appended to the default file names. If the tag name does not start with a period, the tag name is inserted before the file extension. For example, the command

```
lastrac -t _run1234 m0.in
```

generates output files such as `nfact_run1234.dat` and `Nonl_u_amp_run1234.dat`.

We describe the content of each output file as follows. File names given here are the default file names without a user specified tag.

nfact.dat: The major output for any linear runs. It summarizes the N-factor, growth rate, and other important information such as the physical quantities used to normalize the quantities listed in the output. Each row of data contains output values of x, Re, N, ‘a_r, ‘s, ‘s_T, lscl, u_e, T_e, C_r, f, ‘l, ‘q, ‘y, ‘b, ‘l/d, and ist at each mean flow station being computed. Definitions of these output values are as follows.

x: Dimensional x coordinate in meters for 2D or axisymmetric cases. For infinite swept-wing boundary layers with crossflow, this is the nondimensional x/c value as read in from the mean flow file.

Re: Local Reynolds number as defined in equation (6).

N: N-factor computed by integrating the growth rate starting from the onset of instability wave (i.e., the first output station in this file).

‘a_r: Tecplot symbol ‘a represents the Greek letter alpha. This quantity is the real part of α at each location. For nonparallel calculations (including local eigenvalue or PSE), this is the α value before any nonparallel correction. It is non-dimensionalized by the length scale output in the column lscl.

's: Tecplot symbol 's represents the Greek letter sigma. This quantity is the nondimensional physical growth rate of the disturbance. For parallel computations, this is the negative of the imaginary part of the eigenvalue α. For nonparallel cases, this is the effective growth rate (eq. (31)) based on the total kinetic energy (eq. (29)). It is normalized by the length scale output in the column lscl.

's_T: Physical growth rate based on the peak temperature disturbance for nonparallel (either local or PSE marching) computations using equation (31). This output is ignored for parallel calculations.

lscl: Dimensional local length scale in meters used to normalize α and growth rate, and to compute Reynolds number.

u_e: Dimensional boundary layer edge streamwise velocity in m/sec.

T_e: Dimensional boundary layer edge temperature in K.

C_r: Local phase speed ω/α_r. It is non-dimensionalized by the velocity scale printed out in the beginning of `nfact.dat`. For nonparallel computations, this phase speed is based on the value of α before any nonparallel correction (the value of 'a_r).

f: Disturbance frequency in Hz.

'l: Represents the Greek letter lambda (λ) in Tecplot and is the disturbance spanwise wavelength in mm.

'q: Represents the Greek letter theta (θ)in Tecplot and is the angle (in degrees) between the free-stream mean flow direction and the x axis.

'y: Represents the Greek letter Psi (Ψ) in Tecplot and is the wave angle (in degrees) with respect to the free-stream mean flow direction.

'b: Represents the Greek letter beta (β) in Tecplot and is the nondimensional spanwise wave number. It uses the same normalization as 'a_r. If `beta_unit = nwave_axisym_beta` or the transverse curvature is accounted for and `beta_unit = wave_angle_beta`, the integer azimuthal wave number ($n = \beta * r_b$, where r_b is the local cone radius) is written here. For the latter case, n is the optimized wave number which gives rise to a maximum growth rate.

'l/d: Ratio of the total disturbance wavelength to the boundary-layer thickness. The total disturbance wavelength includes contributions from both α and β.

ist: Local station number in the calculation.

npgrowth.dat: Contains the output of nonparallel growth rates at each mean flow station being computed. Each row of data contains values of x, Re, 's_'r_u, 's_'r, 's_u, 's_v, 's_T, 's_E, 's_w. The symbol 'r represents the Greek letter rho (ρ) in Tecplot. Nonparallel effective growth rates based on equation (31) are computed for peak ρu, u, v, T, E, and w disturbances. At each location, the local length scale written in `nfact.dat` is used to normalize the growth rate.

npnfact.dat: Contains the output of nonparallel N factors at each mean flow location being computed. Each row of data contains values of x, Re, ‘N_‘r_u, ‘N_‘r, ‘N_u, ‘N_v, ‘N_T, ‘N_E, ‘N_w. These N factors are computed based on the nonparallel growth rates (e.g., ‘s_‘r_u, ‘s_‘r, etc.) written in `npgrowth.dat`. The integration begins at the first station in this file.

eigenfunc.dat: This output is written only if the input parameter `output_eigenfunction = true`. The eigenfunction or shape-function computed at each mean flow location is written as a zone in Tecplot. Each zone contains the output of the wall-normal y-coordinate normalized by the corresponding local length scale written in the output `nfact.dat`, and the complex eigenfunction values of $\hat{p}$, $\hat{u}$, $\hat{v}$, $\hat{w}$, and $\hat{T}$. If the input parameter `print_rhou_eigf` is set to true, the mass flow rate

$$\widehat{\rho u} = \rho \hat{u} + \hat{\rho} u$$

will be printed instead of $\hat{p}$ in the output file even though the Tecplot variable name still shows pressure.

Global_eig_spectrum.dat: This output is written only if the `-d` option is issued in the command line and a global eigenvalue search is performed in the calculation. The file contains real and imaginary parts of the eigenvalue α computed.

For nonlinear PSE runs, results of the complete Fourier spectrum are written in a C binary file. A separate post-processing code is currently being developed to extract perturbed flow field details. In addition, disturbance amplitudes, phase information, and phase speed are computed and printed for each Fourier mode at a wall-normal location where the corresponding disturbance is maximum. These peak values represent disturbance energy distribution for various Fourier modes. Depending on the symmetry conditions in the Fourier wave number space, only part of the Fourier modes are written in these files. For 2-D or axisymmetric boundary layers, only the first quadrant ($0 \leq m \leq M$ and $0 \leq n \leq N$ in equation (10) is written. For crossflow instability in an infinite swept wing boundary layer, both the first and second quadrants are written ($-M \leq m \leq M$ and $0 \leq n \leq N$). All dimensional quantities in these files are normalized with the scales printed out in the file `nfact.dat`. Unlike the linear runs, `nfact.dat` contains only the scales and no further information.

Nonl_u_amp.dat: Contains x (or x/c for infinite swept wing boundary layers), R, and the streamwise velocity u' amplitude at each mean flow location computed.

Nonl_rhou_amp.dat: Contains x (or x/c for infinite swept wing boundary layers), R, and the streamwise mass flow rate ($(\rho u)' = \bar{\rho} u' + \rho' \bar{u}$) amplitude at each mean flow location computed.

Nonl_T_amp.dat: Contains x (or x/c for infinite swept wing boundary layers), R, and the streamwise velocity T' amplitude at each mean flow location computed.

Nonl_u_phase.dat: Contains x (or x/c for infinite swept wing boundary layers), R, and the streamwise velocity phase information (real part of the effective α_{mn} calculated by applying equation (31) for the wave number α_{mn}) at each mean flow location computed.

Nonl_rhou_phase.dat: Contains x (or x/c for infinite swept wing boundary layers), R, and the streamwise mass flow rate phase information (real part of the effective α_{mn} calculated by applying equation (31) for the wave number α_{mn}) at each mean flow location computed.

Nonl_T_phase.dat: Contains x (or x/c for infinite swept wing boundary layers), R, and the temperature phase information (real part of the effective α_{mn} calculated by applying equation (31) for the wave number α_{mn}) at each mean flow location computed.

Nonl_u_cr.dat: Contains x (or x/c for infinite swept wing boundary layers), R, and the streamwise velocity phase speed at each mean flow location computed.

Nonl_rhou_cr.dat: Contains x (or x/c for infinite swept wing boundary layers), R, and the streamwise mass flow rate phase speed at each mean flow location computed.

Nonl_T_cr.dat: Contains x (or x/c for infinite swept wing boundary layers), R, and the temperature phase speed at each mean flow location computed.

Nonl_KE.dat: Total disturbance kinetic energy as defined in equation (29) is written for every Fourier mode at each mean flow location computed.

Nonl_KE_gr.dat: Contains the computed nonlinear disturbance growth rates based on the disturbance total kinetic energy (eq. (29)) for every Fourier mode at each mean flow location computed.

Nonl_eigenfunc.dat: If the input parameter `output_eigenfunction = true`, shape-functions of each Fourier mode at all locations are written to this file. Each Fourier mode is written as a zone in Tecplot. The absolute amplitude for p, u, v, w, and T is written as a real number. If `print_rhou_eigf` is set to true, p is replaced by the mass flow rate disturbance defined as

$$(\rho u)' = \rho u' + \rho' u + \rho' u'$$

Please note that the wall-normal coordinate y in this file is normalized with a fixed length scale (printed out in `nfact.dat`).

Nonl_wallshear.dat: A good quantity to measure transition is the wall shear stress. LASTRAC computes the average streamwise wall shear ($\partial u/\partial y$) which includes contributions from the mean flow as well as the mean flow distortion mode. The file contains x (or x/c for infinite swept wing boundary layers), R, laminar wall shear, perturbed wall shear, laminar friction coefficient, and the perturbed friction coefficient. Note that when the mean flow distortion mode is included on the left-hand side operator (input parameter `nonl_inc_mfd_jac = true`), the mean laminar values actually already include part of the mean flow distortion.

Nonl_restart.dat: Binary restart file written during the execution of LASTRAC. It is used to restart the nonlinear run at the previous leftover location.

Nonl_flowfld.dat: Includes all Fourier modes computed at all mean flow locations. The post-processing code to read and write specific outputs such as time probing or plot3d files for a snapshot is being developed.

Chapter 6

Case Study I: Linear Calculations

Instead of showing the usage of each input parameter discussed in the previous chapter by a series of unrelated examples, we demonstrate procedures of running LASTRAC to perform stability calculations and transition correlations through case studies that cover all speed regimes. We concentrate on linear LST or PSE calculations in this chapter, nonlinear PSE calculations will be discussed in the next chapter.

Several steps are necessary to perform LASTRAC calculations for a given configuration The first step is to prepare the mean flow file according to the previously described format. One common mistake in preparing the mean flow is to run the mean flow code on a platform different from the one on which LASTRAC is running. Binary files have different byte orders on different machines; as a result, they are not interchangeable. For example, a SUN or SGI machine shares the same binary format, but neither machine can share binary files with Linux Intel machines. When a binary file mismatch happens, LASTRAC will flag error messages like:

```
Error reading title from the mean flow file : ./mflow.m0fp
Error reading number of stations from the mean flow file : ./mflow.m0fp
Error reading gas model, code_unit, etc. from the mean flow file : ./mflow.m0fp
lastrac: MF2d_Reader.cpp:970: Assertion ‘num_stations > 0’ failed.
```

and quit.

The other common mistake is that the mean flow code was run with single precision while LASTRAC expects a double precision binary mean flow file. As a result, a core dump may happen. Moreover, on some machine platforms, compiling the mean flow code using `-r8` would make an integer 8 bytes by default; thus, a `-i4` flag must be used simultaneously.

Running LASTRAC using the `-d` option will print out the parameter values being read from the binary mean flow file. Users may use this option to make sure that the mean flow file was prepared in a correct manner. In addition, the `-d` option together with `mflow_storage_type = memory_storage` in the input script file would generate a file called `mprofile_readin.dat` which contains all mean profiles in the FORTRAN binary mean flow file for debugging.

In the following sections, we present several test cases that cover a broad range of flow speeds. In every case, we guide the users through necessary steps to identify the unstable disturbance frequency and spanwise wave number range. After the unstable range is identified, a transition correlation by using LST or linear PSE may be performed using the N-factor method. Alternatively, a finite amplitude may be given to the dominate unstable mode to perform nonlinear PSE calculations. Transition locations may be computed directly from these nonlinear simulations. Nonlinear calculations will be discussed in the next chapter.

6.1. Flat Plate Boundary Layers

The first test case is the Blasius boundary layer. The first attempt of the input script file may look like this:

```
//
//  mach 0 flat plate
//
      //mflow_storage_type = MEMORY_STORAGE
      grid_type = wall_cluster
      mflow_filename = "../../bl2d/mflow_lastrac"
      marching_method_2d    = along_station
      init_station         = 30
      final_station         =  45

      solution_type      = local_eig_solution

      freq_unit          =  capf_freq

      freq               =  0.5e-4,
      beta               =  0.

      qp_approx          = true
```

Most of the parameter values here except that of `mflow_filename` are irrelevant because we just want to identify the parameter range and make sure that the binary mean flow file is in a correct format. The `wall_cluster` grid specified here is suitable for incompressible and low supersonic flows. The disturbance frequency is $F = 0.5 \times 10^{-4}$. A quasi-parallel LST eigenvalue calculation is intended in the above input. Also note that several lines are commented out in the input file. When LASTRAC has successfully read in the mean flow file, a banner like the following appears on the stdout:

```
   **  **      /*  Mean Flow Parameters Reading in *\   **  **

      Title                 :   MACH 0 FLAT PLATE
      GasModel              : Perfect gas model
      Reference Mach Number : 0.00999951
      Prandtl Number        : 0.7
      No. of Stations       : 200
      X Coordinate Range(m) :      0.038 -      7.620
      Re Range              :    155.434 -   2198.164
      B.L. Length Scale(m)  :   2.4512059E-04 -   3.4665286E-03
      U_inf Velocity (m/s)  :      3.535 -      3.535
      T_inf Temperature (K) :    311.105 -    311.105
   **   **      \*  .............................  */    **   **

  %%%%%%%% disturbance reference frequency (c_r = 0.5) in Hz %%%%%%%%%%%%
    ranging from 81.9492 to 47.4315 for l/d = 2
        or  from 13.6582 to 7.90526 for l/d = 12
```

This banner summarizes the mean flow information for inspection. It also suggests a range of the disturbance frequency in Hz. This frequency estimation is based on a phase speed of 0.5 and a λ/δ value of 2 and 12. These two values cover the usual range of instability waves. In this case, LASTRAC suggests a frequency range of 8–80 Hz. This frequency range is quite low because the mean flow has a very thick boundary layer.

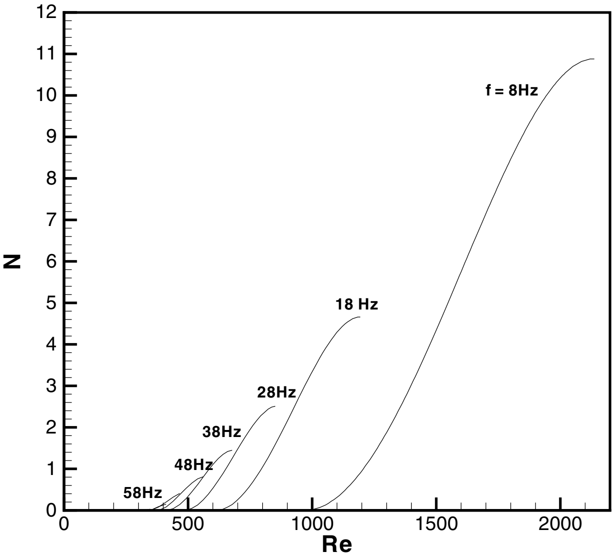

Figure 5: N-factor versus Reynolds number for Blasius boundary layer.

The next step is to confirm the unstable frequency range. Using the suggested frequency band, we may modify the input to:

```
grid_type = wall_cluster

mflow_filename = "../bl2d/mflow_lastrac"

marching_method_2d    = along_station
init_station         = 2
final_station         =  199

solution_type      = local_eig_solution

freq_unit          =  in_hertz_freq

freq               =  8, 18, 28, 38, 48, 58, 68, 78, 88
```

```
beta               =  9*0.

qp_approx          = true
```

Here, we intend to perform a quasi-parallel LST eigenvalue search for 2-D ($\beta = 0$) Tollmien-Schlichting (TS) waves (they are more unstable than the oblique (3-D) waves for Blasius boundary layers) at several frequencies ranging from 8 to 88 Hz. Out of the total 200 stations, we scan from station 2 to 199. The resulting N-factor versus Reynolds number plot is shown in figure 5. The 8 Hz mode gives an N-factor of about 11. Based on these results, we can further refine the N factor calculation by including more frequencies around 8 Hz in the calculation. Alternatively, we can perform linear PSE calculations by using

```
grid_type = wall_cluster

mflow_filename = "../../bl2d/mflow_lastrac"

marching_method_2d    = along_station
init_station         = 10
final_station         =  199

solution_type      = marching_pse_solution
dadx = (0, 1.e-10)

freq_unit          =  in_hertz_freq

freq               =  4, 8, 12, 16, 20
beta               =  5*0.

qp_approx          = false
```

where we have changed the `init_station` to 10 based on the results shown in figure 5 to save computational time. In addition, the $d\alpha/dx$ (`dadx`) value is set to $(0, 1.e - 10)$ because by default the nonparallel eigenvalue search will be done for at least two (`dadx`) values. Setting `dadx = (0, 1.e-10)` is equivalent to $d\alpha/dx = 0$, but it bypasses the other internal guess of dadx and saves computational time. The resulting linear PSE results resemble that shown in figure 5 with the most amplified mode around 8Hz.

The second case is a Mach 2 flat plate boundary layer. Using a simple input file (similar to the Blasius case), we have the following banner output:

```
**  **      /*  Mean Flow Parameters Reading in *\   **  **

   Title                 :  Compressible similar profiles
   GasModel              : Perfect gas model
   Reference Mach Number : 1.99986
   Prandtl Number        : 0.7
   No. of Stations       : 401
   X Coordinate Range(m) :      0.000 -      1.951
   Re Range              :      0.100 -   8000.000
   B.L. Length Scale(m)  :   3.0483704E-09 -   2.4386963E-04
   U_inf Velocity (m/s)  :    590.065 -    590.065
   T_inf Temperature (K) :    216.667 -    216.667
   **  **      \*  ............................  */   **  **



  %%%%%%%% disturbance reference frequency (c_r = 0.5) in Hz %%%%%%%%%%%%
    ranging from 153150 to 88310.4 for l/d = 2
        or  from 25524.9 to 14718.4 for l/d = 12
```

At this Mach number, the most amplified first mode wave is oblique; thus, we need to search for the range of spanwise wavelength that would give rise to unstable first mode disturbances. The following input script computes the maximized N-factor (with respect to the spanwise wave number) for various disturbance frequencies:

```
//
//  Mach 2 flat plate
//
      num_normal_pts       =101

      mflow_filename = "../../meanflow/mflow.m2fp.linux"

      marching_method_2d    = along_re
      init_re =  500,  final_re = 7900, step_re = 100,

      solution_type      = local_eig_solution

      lod_max = 25

      freq_unit          =  in_hertz_freq
      beta_unit          =  wave_angle_beta

      freq = 2e3, 4e3, 6e3, 8e3, 10e3, 12e3, 15e3, 20e3, 25e3, 30e3, 35e3, 40e3, 45e3,
      beta = 13*60,

      qp_approx          = true
```

The option `beta_unit = wave_angle_beta` is used to optimize the growth rate with respect to the spanwise wave number at each station using a guess of wave angles of 60^o (`beta = 60`). This value is typical for the most unstable oblique first-mode waves. We also increase the grid resolution by using 101 points in the wall-normal direction. Instead of going along the stations, we choose to march with a constant Reynolds number increment of 100. This increment is quite large because we are interested only in the unstable spanwise wave number range, not in accurate solutions. Notice the use of the `lod_max = 25` other than the default value of 40. Without this filter, an upstream mode would appear initially for $f = 25$ kHz. For 2-D boundary layers, a value of 25 is reasonable. The resulting N-factor distribution is shown in figure 6 and the corresponding spanwise wavelength distribution is shown in figure 7. Transition correlation may be based on these maximized N-factors. However, N-factor correlation based on individual modes rather than maximized modes represents the physics better; therefore, it gives better transition correlation with respect to experimental data.

The resulting unstable frequency range appears to be closer to the lower values (i.e., from `l/d = 12`) suggested by the LASTRAC banner output (frequencies suggested are around 100 kHz for `l/d = 2`). This

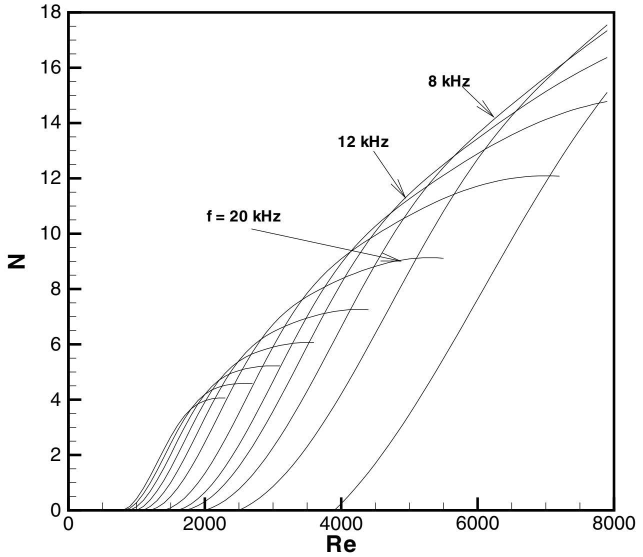

Figure 6: Maximized N-factor versus Reynolds number for $f = 2 - 45$ kHz for a Mach 2 flat plate boundary layer.

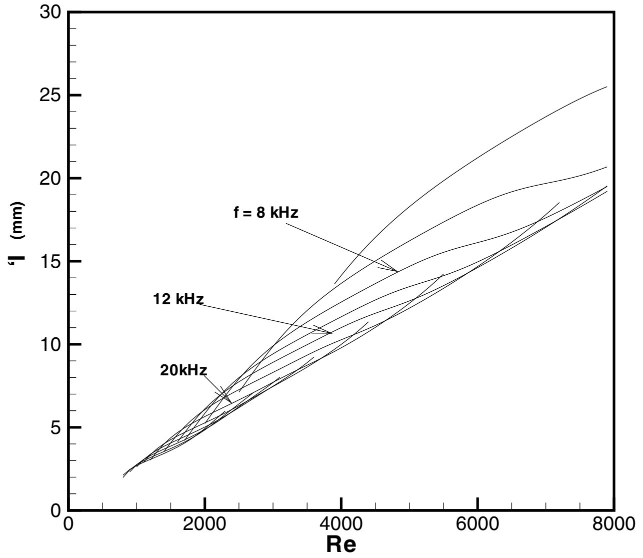

Figure 7: Maximized spanwise wavelength versus Reynolds number for $f = 2 - 45$ kHz for a Mach 2 flat plate boundary layer.

is because a value of λ/δ around 10-12 is typical for oblique first mode waves, Users should always verify the unstable frequency range by quasi-parallel LST calculations to avoid missing important unstable waves. After identifying the unstable parameter range, we may now run linear PSE N-factor correlations in that range.

```
num_normal_pts       =101

mflow_filename = "../../meanflow/mflow.m2fp.linux"

marching_method_2d         =    along_re
init_re =  600,  final_re = 7900, step_re = 100,

dadx = (0, 1.e-10)

solution_type = marching_pse_solution

freq_unit = in_hertz_freq
beta_unit = in_mm_beta
lod_max = 25

freq = 4*8e3, 4*10e3, 4*12e3, 4*15e3, 4*20e3,
beta = 4, 8, 12, 16, 4, 8, 12, 16, 4, 8, 12, 16,
      4, 8, 12, 16, 4, 8, 12, 16,

qp_approx          = false
```

We have narrowed the parameter range to a frequency around $8-20$ kHz and a spanwise wavelength around $4-16$ mm according to the results shown in figures 6 and 7. The nonparallel N-factors computed are plotted in figure 8. The envelope formed by these unstable modes may be used for transition correlation.

In summary, the previous example illustrates how to use LASTRAC for a new problem. First, the frequency range is suggested in the stdout. Users then perform quasi-parallel LST calculations to identify the spanwise wavelength range that gives the most unstable first mode disturbances, and to validate the frequency range suggested by LASTRAC. The maximized N-factor distribution may be used directly for transition correlations. Alternatively, users may track individual unstable modes by using additional LST or linear PSE calculations. The N-factor envelope formed by these calculations provides additional results for transition correlations or predictions.

One of the outstanding issues in stability calculations is the presence of the “upstream modes”. These modes are a discrete eigenmode that travels along the upstream direction with a very large growth rate. Because of its traveling direction, this eigenmode, in fact, decays very rapidly in the downstream direction. As a result, it does not have any physical significance. Nevertheless, upstream modes are present in many eigenvalue calculations. They are usually filtered out in the global spectrum by the preset filters. To demonstrate the upstream mode, we can open up the filter range to allow more modes in the following example:

```
num_normal_pts    =  101

mflow_filename = "../../meanflow/mflow.m2fp.linux"

marching_method_2d    = along_re
```

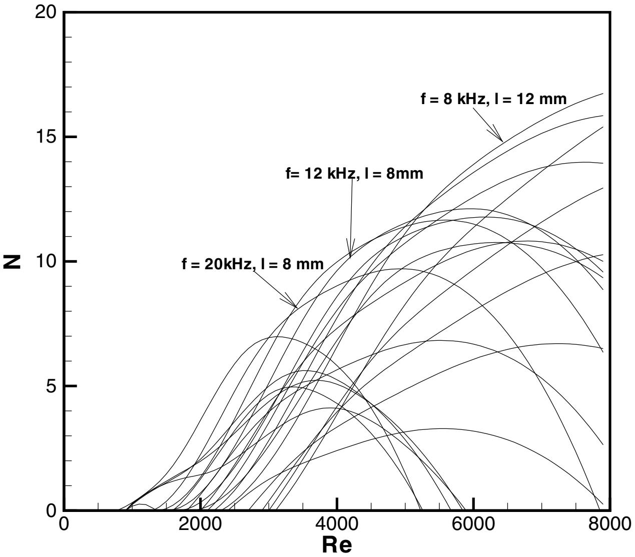

Figure 8: N-factor versus Reynolds number for various oblique first mode waves using linear PSE for $f = 8 - 20$ kHz and $\lambda_z = 4, 8, 12, 16$.

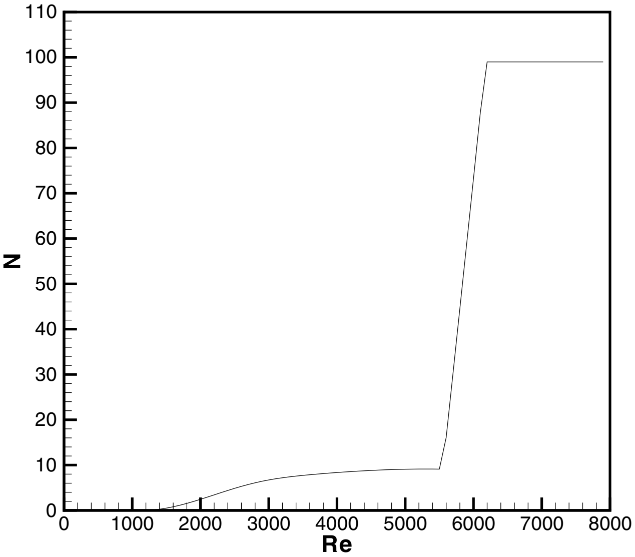

Figure 9: N factor versus Reynolds number for the Mach 2 flat plate boundary layer with $f = 20$ kHz, showing the presence of upstream modes.

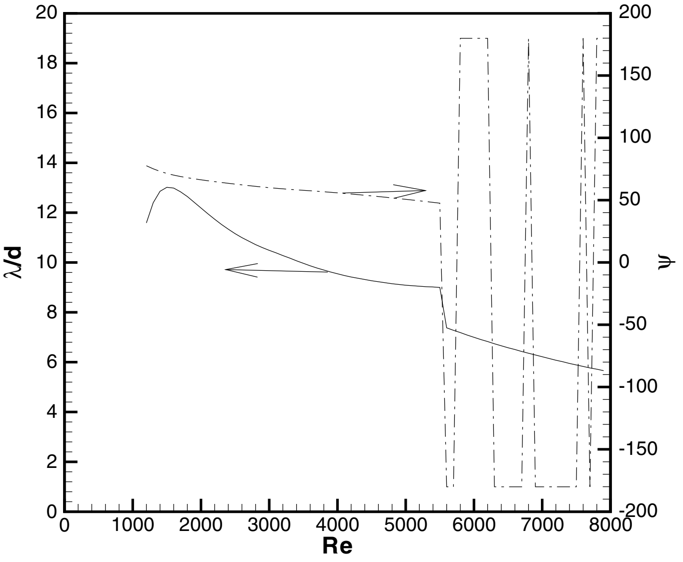

Figure 10: Wave angle and λ/δ versus Reynolds number for the Mach 2 flat plate boundary layer with $f = 20$ kHz, showing the presence of upstream modes.

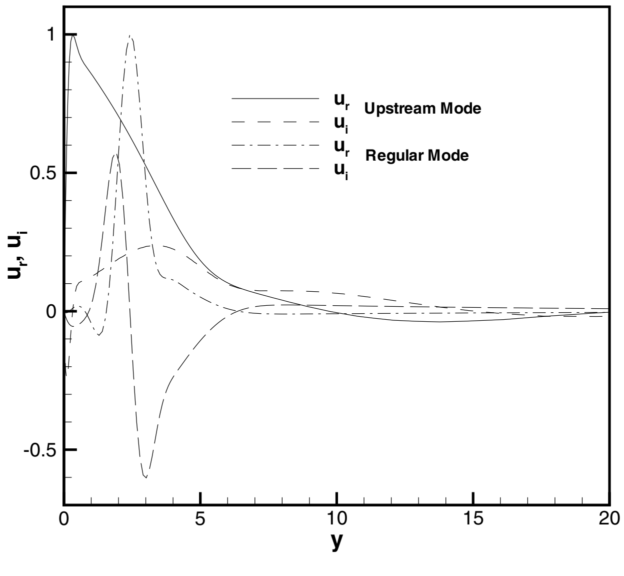

Figure 11: Comparison of streamwise velocity eigenfunctions for upstream and regular instability modes.

```
init_re =  500,  final_re = 7900, step_re = 100,

solution_type      = local_eig_solution

freq_unit          =  in_hertz_freq
beta_unit          =  wave_angle_beta

freq = 20e3  beta = 60,

lod_max = 100
wave_ang_min = -180.   wave_ang_max = 180.

qp_approx          = true
```

The lod_max is increased to 100 and the wave angle filter allows all wave angles to be included. The resulting N-factor is shown in figure 9. A sudden jump near $R = 5500$ is evident; beyond that, the N-factor reaches 100 rapidly. The corresponding λ/δ and wave angles are shown in figure 10. It appears that after the sudden jump, the wavelength falls in the normal range but the wave angle is $\pm 180^o$ with respect to the mean flow direction. This peculiar wave angle indicates that this mode is indeed traveling upstream. Figure 11 compares the streamwise velocity eigenfunction of this upstream mode with that of a regular unstable mode. The upstream mode structure is extremely close to the wall in contrast to the regular mode for which the velocity perturbation peaks near the critical layer.

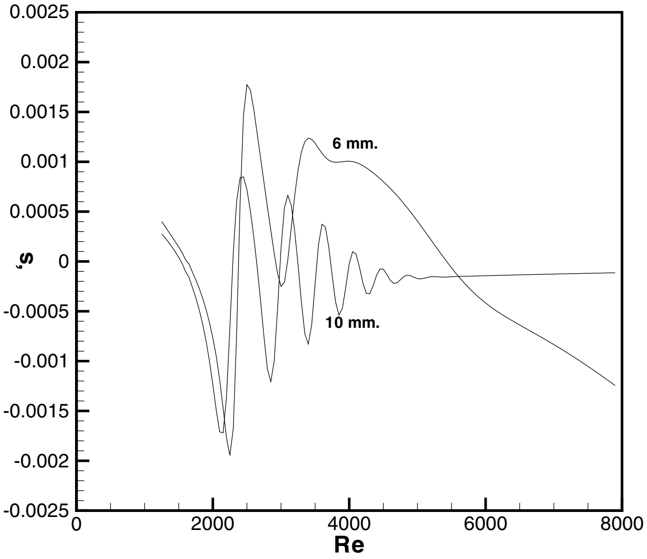

Figure 12: Nonparallel growth rates versus Reynolds number for spanwise wavelength of 10 and 6 mm and $f = 5$ kHz.

During linear PSE calculations, initializing the nonparallel eigenmodes is another important issues. The following shows an example for the same Mach 2 mean flow:

```
num_normal_pts       =101

mflow_filename = "../../meanflow/mflow.m2fp.linux"

marching_method_2d    = along_re
init_re =  1200,  final_re = 7900, step_re = 100,

solution_type      = marching_pse_solution

grid_type = wall_cluster

freq_unit          =  in_hertz_freq
beta_unit          =  in_mm_beta

cr_min = 0.2,    cr_max = 0.6
np_growth_rate_min = -.01

dadx = (0., 1.e-10)
freq = 4*5e3,
beta = 18, 14, 10, 6,

qp_approx          = false
```

Similar to the Blasius case, we use `dadx = (0, 1.e-10)` to save time. In addition, the window for the unstable mode is restricted to a phase speed between 0.2 and 0.6. From linear stability results shown previously, the phase speed is around 0.4; thus, this window will not filter out relevant modes but will prevent irrelevant spurious modes from being captured. The parameter `np_growth_rate_min` is lowered to -0.01 to allow neutrally stable modes in the nonparallel eigenvalue selection process. For PSE calculations, lowering this value allows the code to lock on a nonparallel mode faster and thus save computational time. The starting Reynolds number is increased to 1200 based on the linear stability results. Using this input, we find that the results were good for $\lambda_z = 18, 14$; however, two modes with $\lambda_z = 10, 6$ gave growth rates as shown in figure 12. These modes exhibit oscillatory growth rates and eventually settle with a neutrally stable growth rate. The reason these neutral modes were selected by LASTRAC is that we have a lower threshold for the growth rate in the input. Furthermore, the nonparallel eigenvalue approach has more degree of freedom than its parallel counterpart. Consequently, the eigenvalue procedure may generate modes that are transiently unstable but would degenerate to a neutral mode downstream. From a receptivity standpoint, this kind of mode does exist in the boundary layer but is not relevant. To avoid locking onto these modes, we may restart the calculation later downstream where the unstable eigenmodes stand out and may be easily selected. For this example, if we use `init_Re = 2000`, the results are given in figure 13. In doing linear PSE calculations, users must be aware of situations similar to this example.

6.2. Axisymmetric Cone Boundary Layers

In this example, we are interested in computing stability characteristics for a Mach 6 flared cone investigated experimentally by Horvath et al. (ref. [9]). The mean flow was computed by using the compressible Navier-Stokes code CFL3D (http://fmad-www.larc.nasa.gov/~rumsey/CFL3D/cfl3d.html). Several wind tunnel conditions were studied. The example shown here is associated with the quiet tunnel condition with a unit Reynolds number of 2.89×10^6/ft. The front part of the cone ($0 \leq x \leq 10$ in) is a 5^o half-angle straight cone followed by a circular arc flare section for $10 \leq x \leq 20$ in. The header output for this case is:

```
  **  **      /*  Mean Flow Parameters Reading in *\    **  **

     Title                : Re=2.89Million
     GasModel             : Perfect gas model
     Reference Mach Number : 5.73009
     Prandtl Number       : 0.72
     No. of Stations      : 128
     X Coordinate Range(m) :      0.004 -      0.512
     Re Range             :    200.073 -   2713.197
     B.L. Length Scale(m)  :   1.9912235E-05 -   1.8858289E-04
     U_inf Velocity (m/s)  :    881.857 -    862.361
     T_inf Temperature (K) :     58.947 -     76.948

  **  **      \*  .............................  */    **  **


 %%%%%%%% disturbance reference frequency (c_r = 0.5) in Hz %%%%%%%%%%%%
   ranging from 167695 to 158897 for l/d = 2
       or  from 27949.2 to 26482.8 for l/d = 12
```

The reference Mach number 5.73 is the value behind the shock near the leading edge, not the free-stream value of 6. At this Mach number, the most amplified disturbance is the two-dimensional second mode, which has a wavelength to boundary layer thickness (λ/δ) ratio around 2. Therefore, the frequency range of interest should be in the order of hundreds of kilohertz. The LASTRAC output suggests a frequency around 160 kHz. We perform linear stability calculation for a broader range:

```
//
//  Mach 6 flare cone
//
      mflow_filename = "../../meanflow/mflow.m6fc_qt"

      num_normal_pts       101

      strm_curvt     =   true
      transv_curvt   =   true
      use_extrap_mprof = false


      marching_method_2d   = along_station
      init_station         = 2
      final_station        = 127

      solution_type      = local_eig_solution

      freq_unit          =  in_hertz_freq

      freq = 300e3, 290e3, 280e3, 260e3, 240e3, 220e3, 200e3, 180e3,
            160e3, 140e3, 120e3, 100e3, 80e3, 60e3, 40e3, 20e3
```

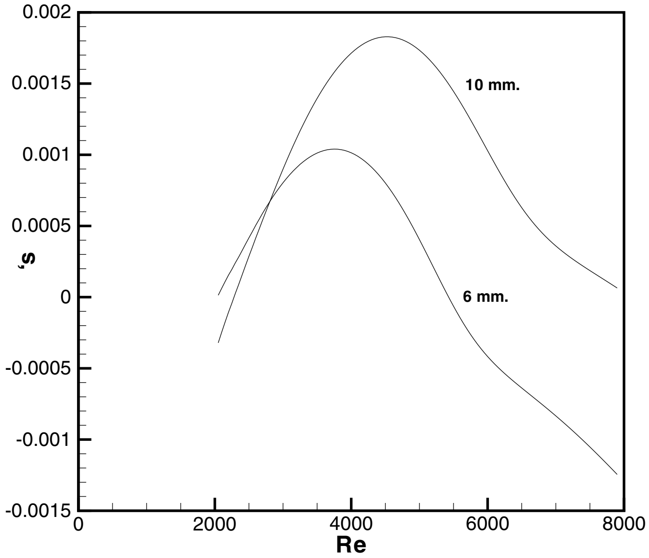

Figure 13: Nonparallel growth rates versus Reynolds number for spanwise wavelength of 10 and 6 mm and $f = 5$ kHz, starting at $R = 2000$

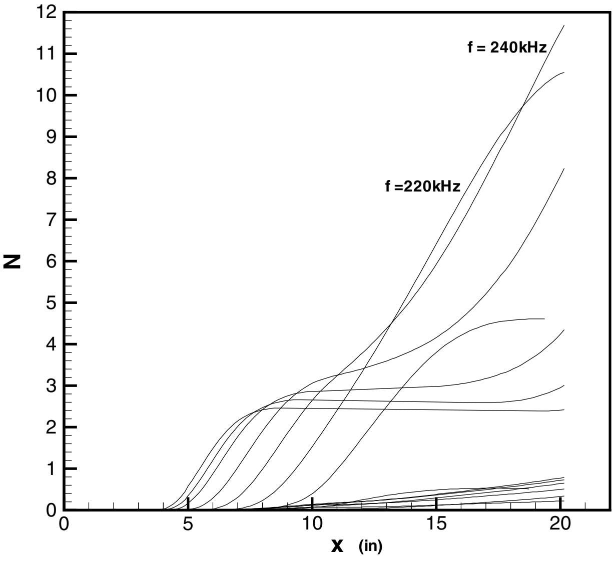

Figure 14: Second mode N-factor versus Reynolds number for Mach 6 flared cone using LST.

```
beta = 16*0,

qp_approx          = true

output_eigenfunction = true

print_rhou_eigf = true
```

We turn on the curvature terms both along the streamwise and transverse (azimuthal) direction because of the flared cone geometry. The meanflow extrapolation is turned off (`use_extrap_mprof = false`) to avoid over-estimating the mean quantities at the far field because of the non-vanishing derivatives in the freestream caused by the Navier-Stokes meanflow solver. Stability calculations start from station 2 to exclude possible noisy solutions right near the inlet of the computational domain in the Navier-Stokes calculations. We also turn on the printing eigenfunctions option and print the mass flow rate instead of the pressure eigenfunction. For hypersonic flows, the mass flow rate eigenfunction information is sometimes more useful than the disturbance pressure. The results are shown in figure 14. The most amplified wave is around $220 - 240$ kHz. The corresponding linear PSE calculations may be carried out by simply replacing `local_eig_solution` with `marching_pse_solution`. The results are shown in figure 15.

The next calculation is for the oblique first mode disturbance at a lower frequency ($f = 60$ kHz). Unlike in the second-mode case, we must first find out the range of unstable integer azimuthal wave numbers (denoted by n). To this end, we optimize the growth rates with respect to n by using the following input:

```
mflow_filename = "../../meanflow/mflow.m6fc_qt"

num_normal_pts       101

strm_curvt     =   true
transv_curvt   =   true
use_extrap_mprof = false

marching_method_2d   = along_station
init_station         = 2
final_station        = 127

solution_type      = local_eig_solution

freq_unit =  in_hertz_freq
beta_unit = wave_angle_beta

freq = 60e3, beta = 65

qp_approx          = true
```

The parameter `beta_unit` is set to `wave_angle_beta` and the `beta` value represents a guess of 65^o. Figure 16 depicts the N-factor and corresponding integer azimuthal wave number. The kink on the wave number curve near $x = 10$ in. is due to the curvature discontinuity in the presence of the flare. The results show that the first mode is most unstable for n ranging from 10 to 50. We may use this information to further perform single mode calculations (i.e., without maximizing the growth rate) by using either LST or PSE. The N-factor envelope formed by these single-mode calculations may be used for transition correlation.

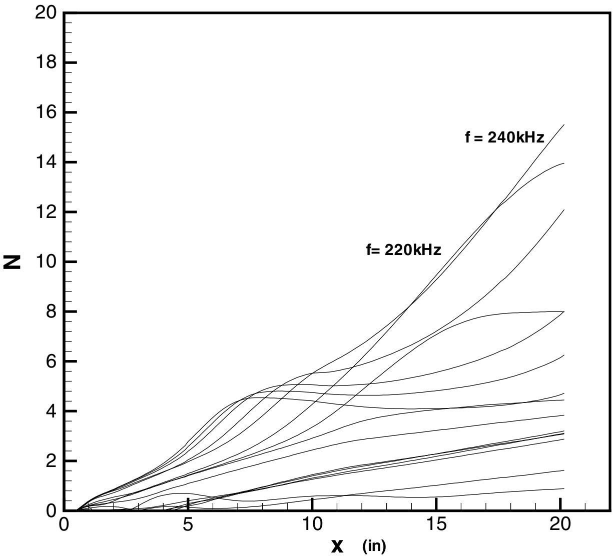

Figure 15: Second mode N-factor versus Reynolds number for Mach 6 flared cone using linear PSE.

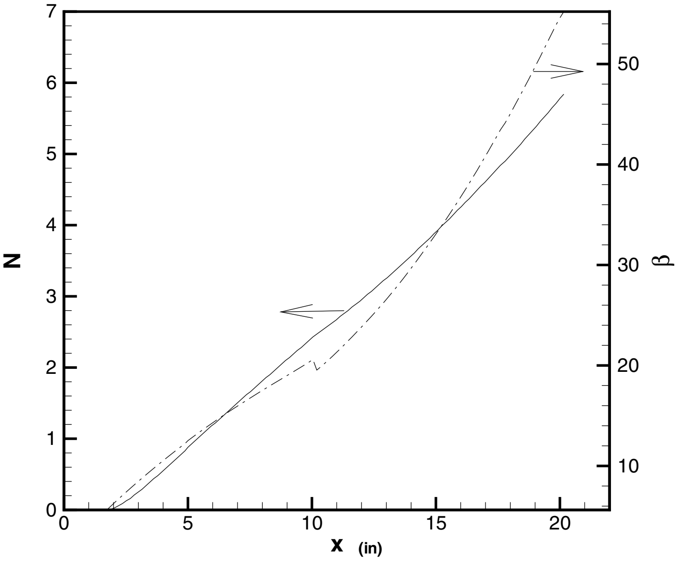

Figure 16: Optimized first mode N factor versus Reynolds number for Mach 6 flared cone, along with corresponding integer azimuthal wave number.

6.3. Infinite Swept Wing

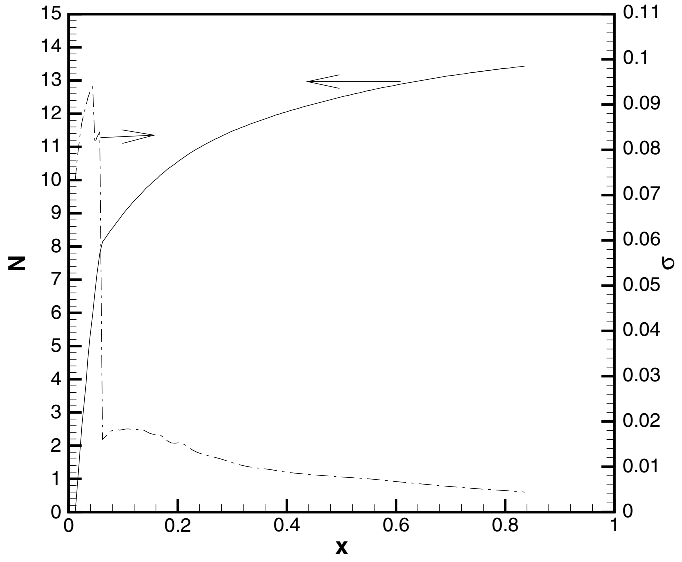

Figure 17: Optimized N-factor versus x/c at $f = 0$ for Mach 2 swept wing, along with corresponding growth rates.

As an example for crossflow instability, we investigate a Mach 2 flow over an 80^o backward swept wing. The LASTRAC output header is:

```
  **  **      /*  Mean Flow Parameters Reading in *\    **  **

     Title                : Mach 2 80 degree swept wing
     GasModel             : Perfect gas model
     Reference Mach Number : 0.043396
     Prandtl Number       : 0.72
     No. of Stations      : 70
     X Coordinate Range(m) :      0.001 -      0.038
     Re Range             :      8.697 -    272.687
     B.L. Length Scale(m)  :   6.3083707E-05 -   1.4083816E-04
     U_inf Velocity (m/s)  :      9.169 -    151.118
     T_inf Temperature (K) :    111.106 -     99.779

  **  **      \*  ..............................  */    **  **

 %%%%%%%% disturbance reference frequency (c_r = 0.5) in Hz %%%%%%%%%%%%%
   ranging from 109511 to 37646 for l/d = 2
       or  from 18251.9 to 6274.33 for l/d = 12
```

The reference Mach number 0.043 is based on the velocity component along the normal-chord direction near the leading edge. We study the stationary crossflow instability first. The first step is to identify the unstable range of spanwise wavelength for stationary crossflow disturbances, we use the following input:

```
//
//  Mach 2 80 degree swept wing
//
      mflow_filename = "../../meanflow/mflow.f16dave.linux"

      marching_method_2d    = along_station
      init_station         = 2
      final_station         =  139

      strm_curvt = true

      solution_type  = local_eig_solution

      freq_unit          = in_hertz_freq
      beta_unit          = l_over_d_beta

      freq = 0,  beta =  4, wave_angle = 87

      qp_approx          = true
```

Here, we optimize the growth rate for a disturbance frequency of 0. For the option of `l_over_d_beta`, it is necessary to input a guess of the wave angle in the `wave_angle` array in addition to the guess of λ/δ provided by `beta`. For stationary crossflow, a guess of 87^o and $\lambda/\delta = 4$ works for almost all cases. The results are shown in figure 17. The computed growth rate near the leading edge is very large (around 0.1) and eventually drops to 0.02 or smaller. The value of λ/δ is around $25 - 34$ near the leading edge and rapidly decreases to below 6 for $x/c > 0.06$. This wavelength to boundary layer thickness ratio near the leading edge is a little too large but still within the acceptable range for crossflow instability. However, further investigations show that the corresponding wave angle is around -25^o near the leading edge and rapidly changes to around 88^o. This wave angle information suggests that the large growth rate mode near the leading edge is an upstream mode and should be filtered out. Therefore, we add the following line to the input file:

```
wave_ang_min = 60, wave_ang_max = 99
```

The new results, shown in figure 18, do not have any large growth rate near the leading edge and both λ/δ and the wave angle fall in the normal range. For crossflow instability calculations, we suggest that a correct wave angle range be specified all the times.

In the previous stationary mode calculations, the most amplified spanwise wavelength ranges from 7 to 14 mm. We then proceed to perform linear PSE calculations based on this information.

```
      mflow_filename = "../../meanflow/mflow.f16dave.linux"

      marching_method_2d    = along_station
      init_station         = 5
      final_station         =  139

      strm_curvt = true

      solution_type  = marching_pse_solution
```

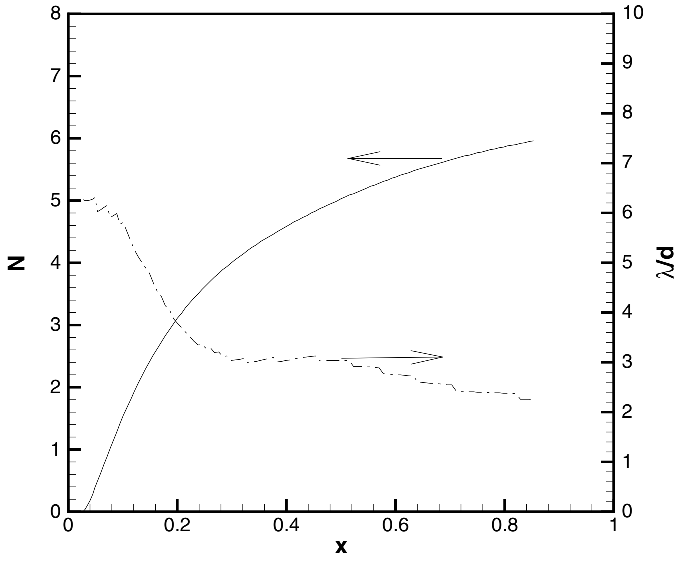

Figure 18: Optimized N factor versus x/c at $f = 0$ for Mach 2 swept wing, along with λ/δ.

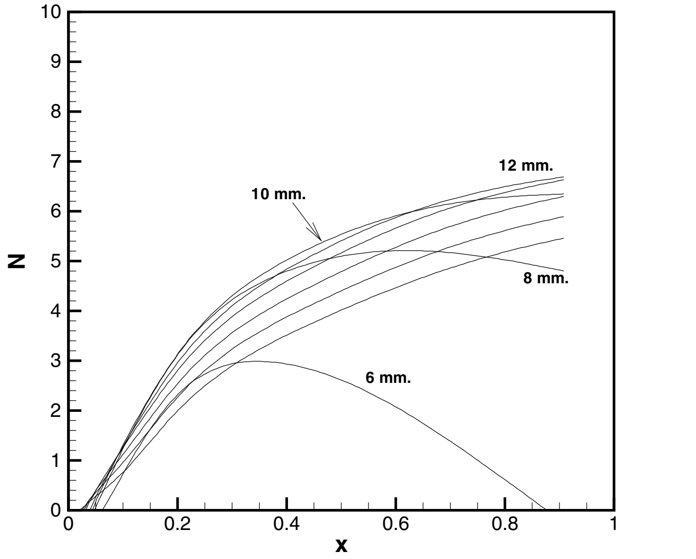

Figure 19: Linear PSE N-factor versus x/c for stationary crossflow modes with spanwise wavelength from 6 to 20 mm.

```
wave_ang_min  = 60,    wave_ang_max = 99
np_growth_rate_min = -1.e-2

freq_unit           = in_hertz_freq
beta_unit           = in_mm_beta

freq = 8*0
beta = 6, 8, 10, 12, 14, 16, 18, 20

qp_approx           = false
```

Here, the starting station is changed to 5 to save some computational time. The use of np_growth_rate_min = -1.e-2 allows linear PSE marching to lock on a converged mode sooner. The resulting N-factor plot is shown in figure 19.

Traveling crossflow instability is also important for swept wing boundary layers. We obtain the unstable spanwise wave number range by using the following input:

```
mflow_filename = "../../meanflow/mflow.m2sw80.linux"

marching_method_2d    = along_station
 init_station         = 5
final_station         =  139
strm_curvt = true

lod_max = 20
wave_ang_min  = 40,    wave_ang_max = 90,

solution_type  = local_eig_solution

freq_unit           = in_hertz_freq
beta_unit           = l_over_d_beta

freq = 12e3, 15e3, 18e3, 21e3, 24e3, 27e3, 30e3, 33e3, 36e3, 39e3,
      42e3, 45e3,
beta = 12*4,  wave_angle = 12*80,

qp_approx           = true
```

Here, the frequency band is based on the LASTRAC output. Note that we use lod_max = 20 and a narrow wave angle range (from 40^o to 90^o). These two parameters offer a properly narrowed filter window that allows traveling crossflow modes to be captured.

Figure 20 shows the resulting growth rate distribution using the previous input file. For $f > 33$ kHz, spurious modes appear near the leading edge. These modes have normal values of λ/δ around 20 and the wave angles are around 60^o. However, further inspection of the eigenfunctions reveals that they are indeed spurious modes. LASTRAC did not filter out these modes because of their about normal wave angle and values of λ/δ. To remedy this, we may use the the growth rate filter

```
alpha_i_max = 0.05
```

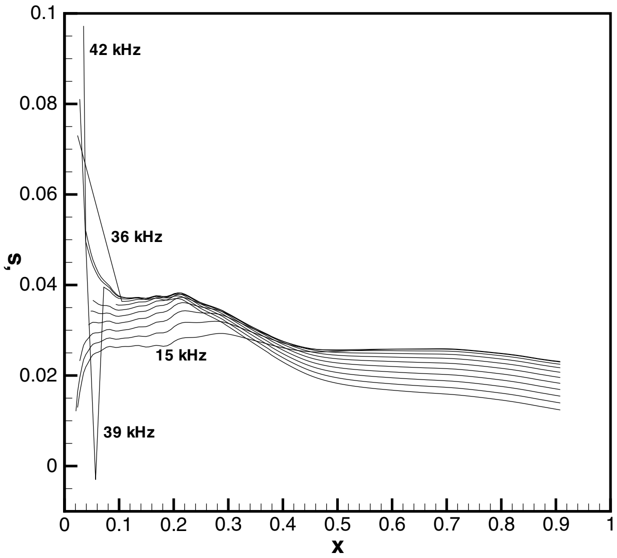

Figure 20: Growth rate versus x/c for traveling crossflow modes with frequency of 15 to 45 kHz with spurious modes near leading edge.

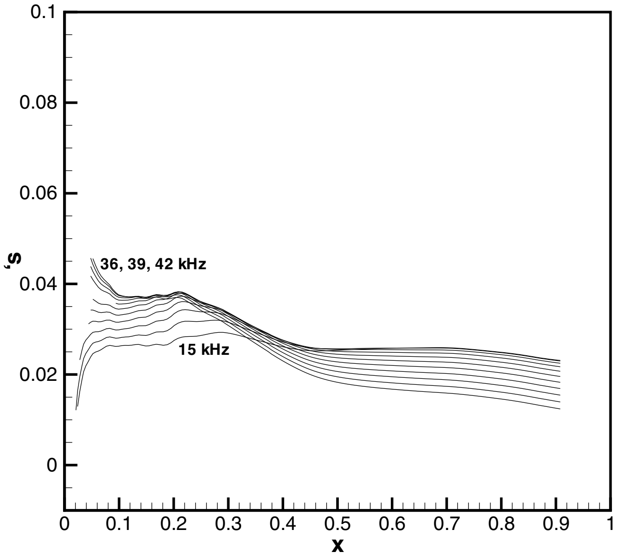

Figure 21: Growth rate versus x/c for traveling crossflow modes with frequency of 15 to 45 kHz, without spurious modes

to force LASTRAC to discard these modes from the global spectrum. Figure 21 shows the results obtained when we include the growth rate filter. The corresponding N-factor results are shown in figure 22. The unstable spanwise wavelength ranges from 6 to 20 mm. We may further pursue linear PSE calculations by using this information.

The mean flow for the above swept wing case was obtained with a spectral code which employed a relatively coarse grid. As a result, the mean flow grid does not have enough resolution inside the boundary layer. The low-order (relative to spectral interpolation) spline interpolated mean flow used by LASTRAC thus makes the stability calculations more difficult than that from a regular finite-difference based mean flow code. However, with proper filters, users can still obtain good and accurate solutions.

6.4. Jet and Shear Layers

A shear layer with hyperbolic tangent velocity and temperature profiles as shown in figure 23 is used as a test case. The basic flow file was written by repeating these profiles for a few downstream locations via artificially increasing the x coordinate but fixing the rest of the mean flow values. The LASTRAC output header is:

```
   **  **      /*  Mean Flow Parameters Reading in *\    **  **

      Title                : Shear Layer Mean Flow
      GasModel             : Perfect gas model
      Reference Mach Number : 0.300142
      Prandtl Number       : 0.71
      No. of Stations      : 10
      X Coordinate Range(m) :      0.000 -      0.274
      Re Range             : 170000.000 - 170000.000
      B.L. Length Scale(m)  :   2.5400000E-02 -   2.5400000E-02
      U_inf Velocity (m/s)  :    106.080 -    106.080
      T_inf Temperature (K) :    310.887 -    310.887

   **  **      \*  ..............................  */    **  **


  %%%%%%%% disturbance reference frequency (c_r = 0.5) in Hz %%%%%%%%%%%%%
    ranging from 8429.69 to 8429.69 for l/d = 2
        or  from 1404.95 to 1404.95 for l/d = 12
```

Based on the suggested frequency, we use this input file:

```
//
//  shear layer test case
//
      mflow_filename = "../../meanflow/mflow.slayer"
      flow_type = shear_layer

      grid_type = critl_cluster
```

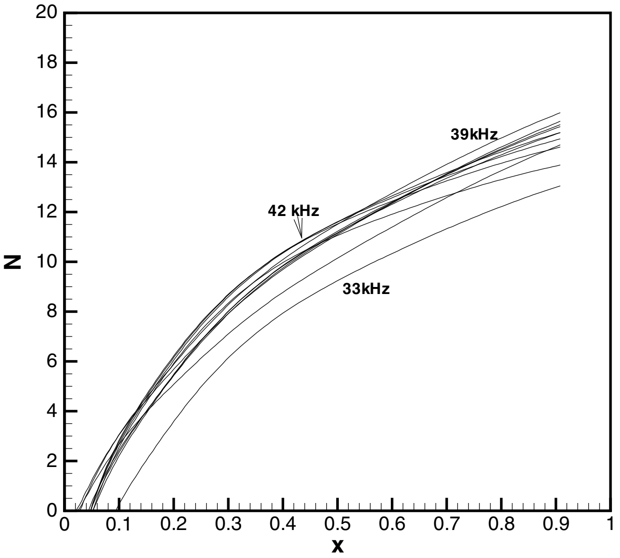

Figure 22: N-factor versus x/c for traveling crossflow modes with a frequency of 15 to 45 kHz, without spurious modes

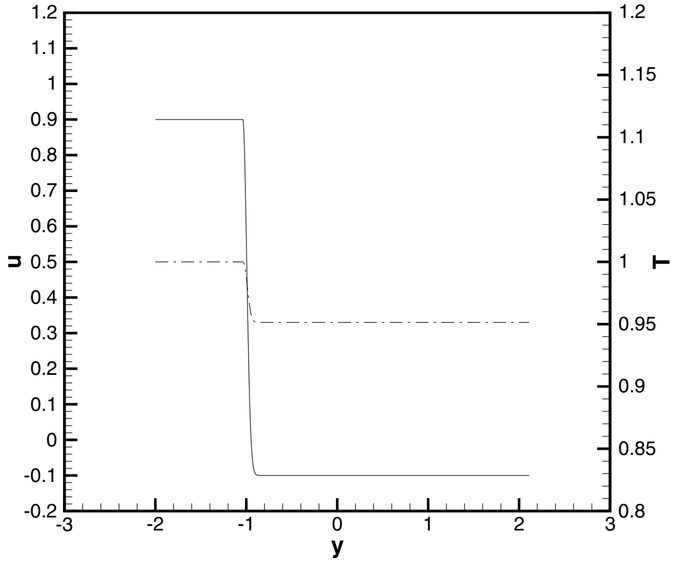

Figure 23: Velocity and temperature profiles for compressible shear layer.

```
    tau_s = 20
    relax_type = critlayer

    cr_min = -2, cr_max = 2

    num_normal_pts       = 201
    num_normal_pts_geig = 61

    alpha_i_max = 10000.
    max_value_eig_correction = 10000.

    ymax = 2
    ymax_glob_search = 2

    marching_method_2d   =  along_station
    init_station = 5,  final_station = 5

    solution_type      = local_eig_solution

    freq_unit = in_hertz_freq,
    beta_unit = non_dim_beta

    freq = 8000, 7800, 7600, 7400, 7200, 7000, 6800, 6600, 6400, 6200,
          6000, 5800, 5600, 5400, 5200, 5000, 4800, 4600, 4400, 4200,
          4000, 3800, 3600, 3400, 3200, 3000, 2800, 2600, 2400, 2200,
          2000, 1800, 1600, 1400, 1200, 1000, 0800, 0600, 0400, 0200,
    beta = 40*0

    qp_approx          = true

    output_eigenfunction = true
```

Notice the use of the shear_layer flow type. A grid that clusters grid points in the middle of the shear layer is suitable for this case. Thus, we use the critl_cluster option and a stretching parameter of 20. We use 201 grid points to resolve this thin shear layer. The phase speed range covers both positive and negative values. Because the scale used to normalize the length is the shear layer thickness, we must allow large growth rates and eigenvalues. Therefore, we significantly increase the parameters alpha_i_max and max_value_eig_correction. The ymax value of 2 is sufficient to cover the entire domain of interest. Employing the default value of ymax=200 renders the near center line grid inappropriate to resolve the shear layer. Adjusting these growth rate and grid related parameters is also important for boundary layer type calculations in which the length scale used is not the similarity boundary layer scale. Notice that the use of relax_type = critlayer for eigenvalue iteration is necessary because no wall is involved in this calculation. Multiple unstable modes are found by using the above input. Figure 24 shows the velocity and temperature eigenfunctions of one such mode.

Our next example is a Mach 2.5 supersonic jet. The mean flow characteristics are summarized as:

```
 **  **      /*  Mean Flow Parameters Reading in *\   **  **

    Title                 :  Mach 2.5 Jet mean flow from Balakumar code
    GasModel              : Perfect gas model
    Reference Mach Number : 1.66667
```

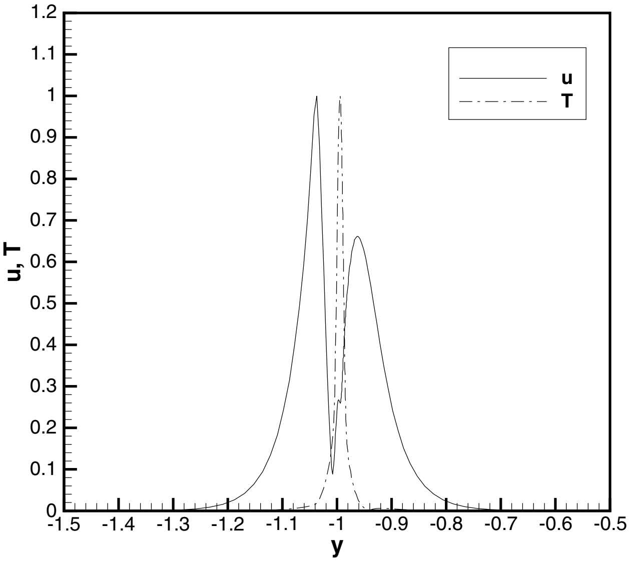

Figure 24: Velocity and temperature eigenfunctions of unstable mode for compressible shear layer at $f = 8000$ Hz

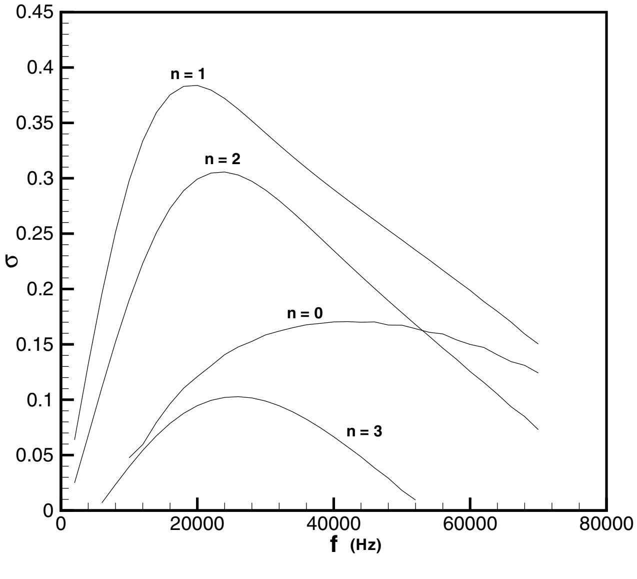

Figure 25: Growth rates versus disturbance frequency at $x = 0.02$ m for different azimuthal wave numbers.

```
      Prandtl Number        : 0.72
      No. of Stations       : 257
      X Coordinate Range(m) :      0.001 -      0.159
      Re Range              :   1924.578 -   1924.578
      B.L. Length Scale(m)  :   6.1600000E-03 -   6.1600000E-03
      U_inf Velocity (m/s)  :    572.832 -    572.832
      T_inf Temperature (K) :    294.000 -    294.000

   **  **      \*  ..............................  */   **  **



  %%%%%%%% disturbance reference frequency (c_r = 0.5) in Hz %%%%%%%%%%%%%
    ranging from 1635.28 to 1626.27 for l/d = 2
        or  from 272.547 to 271.045 for l/d = 12
```

The reference Mach number is based on the free-stream temperature and thus is smaller. We use the following input to study possible unstable modes at a downstream location of $x = 0.02$:

```
//
//  Mach 2.5 jet
//
      mflow_filename = "../../meanflow/mflow.m2.5jet"

      flow_type = jet_vortex

      grid_type = critl_cluster
      tau_s = 10.

      num_normal_pts       = 101

      alpha_i_max = 1.

      ymax = 10
      ymax_glob_search = 5

      transv_curvt = true

      relax_type = critlayer

      marching_method_2d   =  along_xc
      init_xc = 0.02,  final_xc = 0.02, step_xc = 0.001

      solution_type      = local_eig_solution

      freq_unit = in_hertz_freq,
      beta_unit = nwave_axisym_beta,

      freq = 0.2e4, 0.4e4, 0.6e4, 0.8e4, 1.e4, 1.2e4, 1.4e4,
            1.6e4, 1.8e4, 2.e4, 2.2e4, 2.4e4, 2.6e4, 2.8e4,
            3.0e4, 3.2e4, 3.4e4, 3.6e4, 3.8e4, 4.0e4, 4.2e4,
            4.4e4, 4.6e4, 4.8e4, 5.0e4, 5.2e4, 5.4e4, 5.6e4,
            5.8e4, 6.0e4, 6.2e4, 6.4e4, 6.6e4, 6.8e4, 7.0e4,
      beta = 35*1,

      sqp_approx          = true
```

Notice the use of the `jet_vortex` flow type and the `critl_cluster` grid type. The scaling in the mean flow is based on the effective jet diameter; therefore, we must allow modes with large growth rate by setting `alpha_i_max = 1`. The use of `ymax = 10` is also quite different from the boundary layer cases. In the cylindrical coordinate, we must also account for the azimuthal curvature (`transv_curvt = true`). Similar to the shear layer cases, the `relax_type` should be `critlayer`. In addition, the appropriate spanwise wave number option is an integer azimuthal wave number; therefore, we use `beta_unit = nwave_axisym_beta`. Here, the frequency range is from $f = 2$ to 70 kHz. We make this frequency range broader than that suggested in the banner output to cope with possible errors in the boundary layer based frequency estimation invoked in LASTRAC. We repeat the above calculations from $n = 0$ to $n = 3$; the results are shown in figure 25.

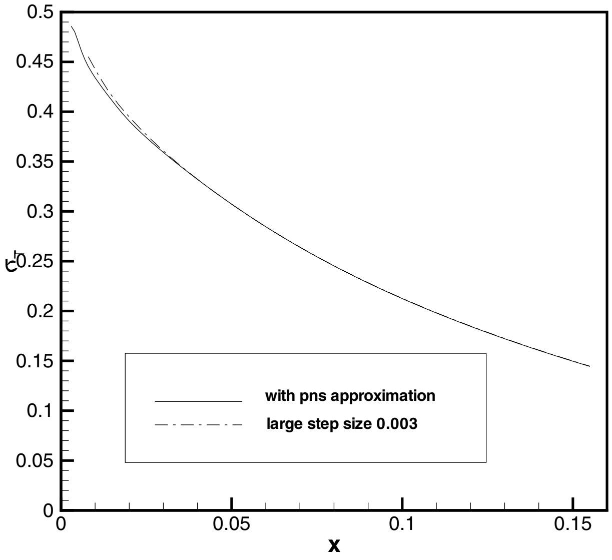

Figure 26: Growth rates vs. x from linear PSE marching with pns approximation or a small step size.

Based on the linear stability results shown in figure 25, we perform linear PSE calculations for the most amplified mode ($f = 19$ kHz, $n = 1$):

```
    init_xc = 0.002,  final_xc = 0.156, step_xc = 0.001
    solution_type      = marching_pse_solution
    freq = 19e3, beta = 1,
```

The code runs, but it stops at $x = 0.014$ and an error message indicates that the PSE iteration did not converge. This error is associated with the elliptic behavior of the linear PSE approximation discussed in the previous section. To maintain numerical stability, we need to add the following line to the input file:

```
    pns_approx = true
```

Alternatively, we can increase the marching step size by `step_xc = 0.003`. Figure 26 compares the growth rates from these two approaches. Except near the beginning of the domain, the results are almost identical.

The examples discussed in this section show that LASTRAC may also be used for jet and shear layer flows. However, because the default parameters are tuned for boundary layer flows, users must pay attention to the eigenvalue filters when applied to shear layer calculations.

Chapter 7

Case Study II : Nonlinear Calculations

All calculations presented in the previous chapter are linear, i.e., the amplitude of disturbances scales directly with the initial amplitude. The absolute amplitude of the disturbance does not play a role in the disturbance evolution. Each instability wave grows or decays independently. In reality, this linear assumption is only valid when the absolute amplitude of the disturbance is sufficiently small. When a disturbance grows to large enough amplitude, nonlinear interaction causes energy exchange among all Fourier harmonics. As a result, the mean flow profile becomes distorted in response to nonlinear effects. Stability characteristics of this distorted mean flow are quite different from the original laminar state. In this chapter, we discuss examples in which a finite amplitude is imposed on disturbances.

The amplitude of any Fourier mode used herein is defined as the root mean square (rms) value, i.e.,

$$A_{mn} = \left(\frac{1}{\tau} \int_0^{\tau} \phi_{mn}^2 dt \right)^{1/2}$$

where $\tau = 2\pi/\omega$ is the time period associated with the computational domain. The quantity ϕ_{mn} is the absolute value of the corresponding Fourier component of the total disturbance (as defined in equation (10),

$$\phi_{mn} = |\chi_{mn}(x, y) e^{i(n\beta z - m\omega t)}|$$

measured at the corresponding peak location in the wall-normal direction. For low ($M < 4$) and high Mach number flows, the amplitude is based on peak streamwise velocity and temperature, respectively. The amplitude parameter that users provide in the input file is interpreted as the rms amplitude of each Fourier mode. The amplitude parameter has a complex value to allow phase disparity among different modes.

7.1. Nonlinear Disturbance Evolution

The first case is a Mach 4.5 flat plate boundary layer. We pick a disturbance frequency of $F = 0.8 \times 10^{-4}$. Using the linear PSE procedure described in the previous chapter, we find an unstable mode at $R = 2000$ with an eigenvalue of $(0.176936, -0.000330943)$. The nonlinear PSE calculation is performed with the following input:

```
//
//  input file for nonlinear PSE
//
      use_mthread = true
      nthread = 8

      nonl_pse_calc  = true
      n_mode_omega = 7,  n_mode_beta = 0,
      n_init_modes = 1,
      nonl_alpha_init = (0.176936,-0.000330943)
      nonl_amp_init = (0.06, 0)
      nonl_mode_init = (1, 0)
      nonl_dealiasing = true
      nonl_relax_coef = 0.5

      nonl_convg_tolerance = 1.e-4
      max_nonl_iteration = 200

      mflow_filename = "../../meanflow/mflow.m4.5fp.linux"

      marching_method_2d         =    along_re
      init_re = 2000,  final_re = 3000, step_re = 20

      np_growth_rate_min = -0.01

      freq_unit          =  capf_freq
      freq               =  0.80e-4,
      beta               =  0.
```

Here, we turn on the multi-thread option and set the maximum number of concurrent threads to 8. The optimal number for performance depends on the hardware configuration of the machine. To make a nonlinear PSE calculation, we need to set `nonl_pse_calc = true` and then specify number of Fourier modes (in this case, 7 positive temporal modes and 0 spanwise modes, $M = 7$ and $N = 0$). We only initiate one eigenmode (mode $(1, 0)$) with the eigenvalue specified and a disturbance amplitude of 6% based on the peak temperature disturbance (because the free-stream Mach number is 4.5). The de-aliasing option is on and an under-relaxation factor of 0.5 help stabilize nonlinear iterations. Other parameters needed are mean flow file name and the marching step. The use of `np_growth_rate_min = -0.01` allows a relatively stable eigenmode to be included in the initial location. The disturbance frequency specified is the fundamental frequency of the Fourier series expansion. We use $F = 0.8 \times 10^{-4}$, and only higher harmonics will be calculated.

The computed mass flow rate amplitude for each Fourier mode is shown in figure 27. In addition to the input $(1, 0)$ mode, other higher harmonics and the mean flow distortion $(0, 0)$ modes all grow to larger amplitudes and then decay with the primary mode. The peak mass flow rate amplitude in this case reaches about 13% and temperature about 31%. The growth rate of the primary mode is compared with that from the corresponding linear PSE calculation in figure 28. The nonlinear case saturates and decays faster than the linear case.

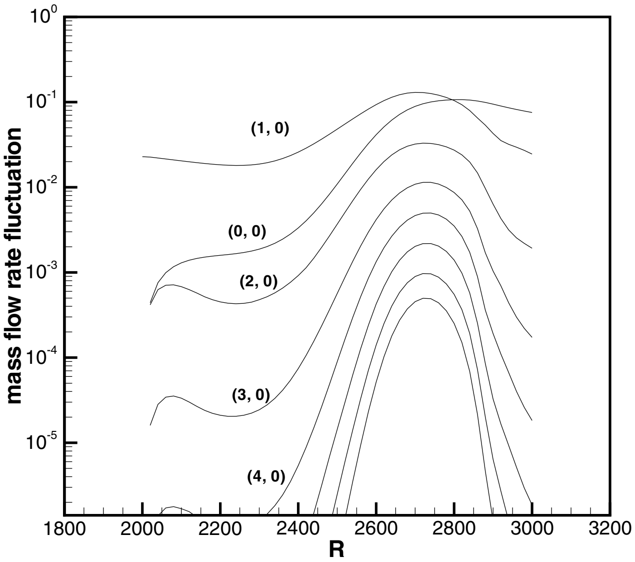

Figure 27: Mass flow rate amplitude for various Fourier modes for Mach 4.5 flat plate boundary layer with 2-D primary disturbance of $F = 0.8 \times 10^{-4}$.

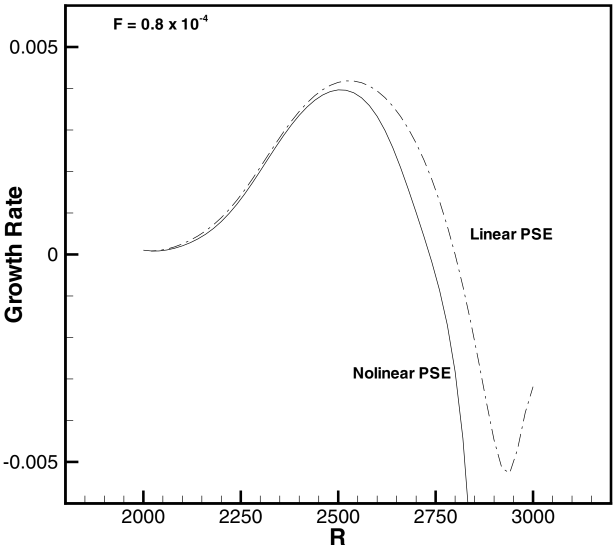

Figure 28: Growth rates versus Reynolds number for linear and nonlinear PSE calculations for Mach 4.5 flat plate boundary layer.

7.2. Secondary Instability of 2-D Boundary Layers

In this section, we use LASTRAC to compute the secondary instability of an incompressible flat plate boundary layer. The primary disturbance has a frequency of $F = 1.24 \times 10^{-4}$. The subharmonic secondary instability with a spanwise wave number of $\beta = 0.14$ at $R = 390$ is of interest. We first ran LASTRAC with the linear PSE option to get a nonparallel eigenvalue $\alpha = (0.132594, -0.000704844)$ with $\beta = 0$ at $R = 390$. The oblique subharmonic mode is with a spanwise wave number β of 0.14 and a frequency of $F = 0.62 \times 10^{-4}$. There are two different ways to initialize this mode in LASTRAC. We can use the primary mode $((2, 0))$ eigenfunction directly. However, this leads to transient effects for the subharmonic mode. Alternatively, we can use the associated stable eigenmode at this Reynolds number. However, identifying a stable discrete eigenmode poses a problem. A good strategy is to begin with an easy to find unstable 2-D mode ($\beta = 0$) and gradually increase β to the desired value. The following input file shows how to do it,

```
      np_growth_rate_min = -1.e-1

      marching_method_2d         =    along_re
      init_re = 390,    final_re = 390, step_re = 1.0

      solution_type = local_eig_solution
      qp_approx          = false
      freq_unit = capf_freq
      beta_unit = non_dim_beta

      freq = 8*0.62e-4,
      beta = 0, 0.02, 0.04, 0.06, 0.08, 0.1, 0.12, 0.14
```

In this input, a global nonparallel eigenvalue search will be performed for the 2-D mode ($\beta = 0$). The converged mode is then used as an initial guess for the next calculation ($\beta = 0.02$ in this case). In this way, we avoid the difficulties associated with picking a stable mode in the spectrum. Note that we also lower the bound of stable growth rate `np_growth_rate_min = -1.e-1` to make it more inclusive for the eigenvalue selection. Using this input, we find $\alpha = (0.0569691, 0.00620216)$ for the oblique subharmonic mode.

Having both primary and subharmonic eigenmodes, we are ready to simulate transition with a prescribed initial amplitude. As a first attempt, we may use:

```
//
//  Subharmonic breakdown for an incompressilbe boundary layer
//
      num_normal_pts       =101

      grid_type = wall_cluster

      np_growth_rate_min = -1.e-1

      use_mthread = true
      nthread = 4

      nonl_pse_calc  = true
```

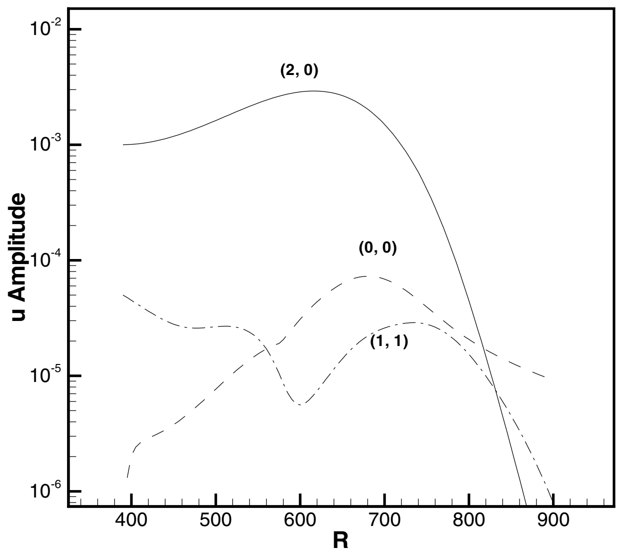

Figure 29: Streamwise velocity disturbance amplitude versus Reynolds number for primary $(2,0)$, subharmonic $(1,1)$, and mean flow distortion $(0,0)$ modes; initial amplitude of the primary mode is .1%.

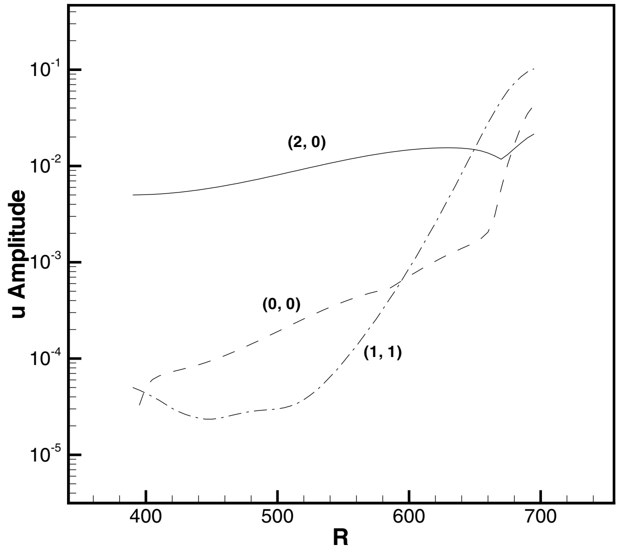

Figure 30: Streamwise velocity disturbance amplitude versus Reynolds number for primary $(2,0)$, subharmonic $(1,1)$, and mean flow distortion $(0,0)$ modes; initial amplitude of the primary mode is .5%

```
n_mode_omega = 4,  n_mode_beta = 4,
n_init_modes = 2,
nonl_alpha_init = (0.132594,-0.000704844), (0.0569691,0.00620216)
nonl_amp_init = (0.001, 0), (5.e-5, 0)
nonl_mode_init = (2, 0), (1, 1)
nonl_alpha_scaling_mode_t = (2, 0)
nonl_dealiasing = true

nonl_inc_mfd_jac = true

nonl_convg_tolerance = 1.e-4
max_nonl_iteration = 100

pns_approx = true

mflow_filename = "../../meanflow/mflow.klexp.linux"

marching_method_2d    = along_re
init_re = 390,    final_re = 900, step_re = 5.0

freq_unit          =  capf_freq

freq               =  0.62e-4,
beta               =  0.14

qp_approx          = false
```

Here, we use $M = 4$ and $N = 4$. The fundamental frequency is $F = 0.62e - 4$. The primary and subharmonic modes are $(2, 0)$ and $(1, 1)$, respectively. The α_{mn} scaling for the non-initialized Fourier modes is based on the $(2, 0)$ mode (`nonl_alpha_scaling_mode_t = (2, 0)`). The initial amplitudes are 0.1% and $5.e - 5$, respectively. To stabilize the nonlinear iterations, we include the mean flow distortion mode on the left-hand side operator by setting `nonl_inc_mfd_jac = true`. The computed disturbance amplitude is plotted in figure 29. The primary disturbance grows initially but never reaches an amplitude large enough to trigger secondary instability of the subharmonic mode.

We increase the initial amplitude of the primary disturbance to 0.5% while keeping the subharmonic mode the same by using `nonl_amp_init = (0.005, 0), (5.e-5, 0)`. The results are shown in figure 30. Now, the $(1, 1)$ mode grows rapidly starting from $R = 520$. At around $R = 650$, the subharmonic mode overtakes the primary mode. These results indicate that the flow undergoes a subharmonic transition process. To further resolve this flow phenomena, we change the input file to:

```
num_normal_pts       =101
np_growth_rate_min = -1.e-1

use_mthread = true
nthread = 4

nonl_pse_calc  = true

n_mode_omega = 6,  n_mode_beta = 6,
n_init_modes = 2,
nonl_alpha_init = (0.132594,-0.000704844), (0.0569691,0.00620216)
```

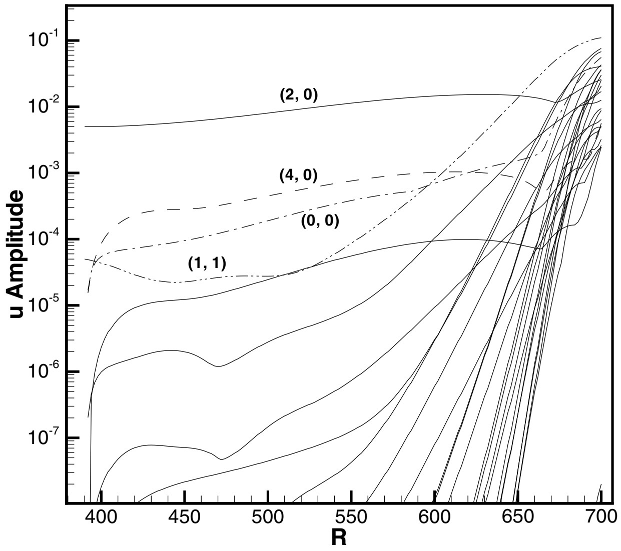

Figure 31: Evolution of disturbance amplitudes for incompressible subharmonic breakdown for all participating Fourier modes

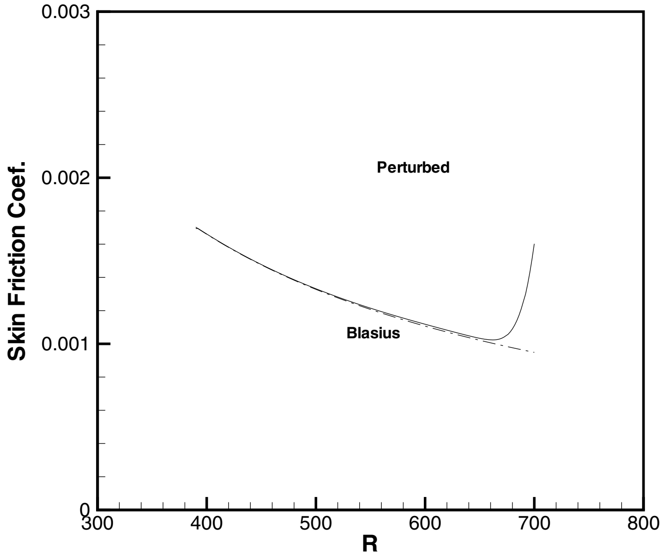

Figure 32: Comparison of skin friction coefficients for aminar (Blasius) and subharmonic transitioning cases.

```
nonl_amp_init = (0.005, 0), (5.e-5, 0)
nonl_mode_init = (2, 0), (1, 1)
nonl_alpha_scaling_mode_t = (2, 0)
nonl_dealiasing = true

nonl_inc_mfd_jac = true

nonl_convg_tolerance = 1.e-4
max_nonl_iteration = 100
pns_approx = true

mflow_filename = "../mflow/mflow.klexp.linux"

marching_method_2d        =    along_re
init_re = 390,    final_re = 900, step_re = 2.0

freq_unit          =  capf_freq

freq               =  0.62e-4,
beta               =  0.14

qp_approx          = false
```

We increase the number of Fourier modes to $M = 6$ and $N = 6$. The streamwise grid resolution is also increased by using a constant Reynolds number increment of 2 (`step_re = 2`). This new grid gives better resolution near the transition region. The resulting amplitude evolution is depicted in figure 31 which clearly shows the secondary instability process. Compared with the previous run with less resolution (fig. 30), the transition region in figure 31 is better resolved; near the end, the dominating subharmonic mode saturates and back reaction forces the primary mode to grow again. These important features of the subharmonic breakdown process were previously only observed in DNS calculations. Nonlinear iterations failed to converge at $R = 700$ for this calculation. Decreasing the under-relaxation factor (`nonl_relax_coef`) to about 0.1, we can continue nonlinear iterations beyond $R = 700$ and go deeper into the transition region. Alternatively, a subgrid scale model that will be implemented in the future release would also help stabilize nonlinear iterations in the transition region. The corresponding friction coefficient plot is shown in figure 32 where a rapid rise in skin friction is evident. This phenomenon further verifies that the flow has entered the laminar-turbulent transition stage.

If a converged eigensolution cannot be found for the oblique mode, we can use the primary $(2, 0)$ eigenfunction in place of the $(1, 1)$ mode by using

```
nonl_alpha_init = (0.132594,-0.000704844), (0, 0)
```

and keeping everything else the same. This alternative initialization procedure worked similar to the original one except that noticeable transient effect was observed for the $(1, 1)$ mode.

This case exemplifies how transition may be predicted in a forced environment with a prescribed amplitude for relevant instability modes. Of course, the initial amplitude still needs to be determined by the receptivity process, this issue is a future topic for LASTRAC development.

7.3. Oblique-Mode Breakdown in Supersonic Boundary Layers

For low supersonic boundary layers with a free-stream Mach number less than 4, the most dominant instability is the oblique first mode. One pair of oblique modes could interact and produce the streamwise vortex mode. Triad interaction among these three modes gives rise to the growth of other Fourier harmonics and eventually leads to transition (for details, see reference [6]). Here, we demonstrate this breakdown process for the Mach 2 flat plate boundary layer investigated previously.

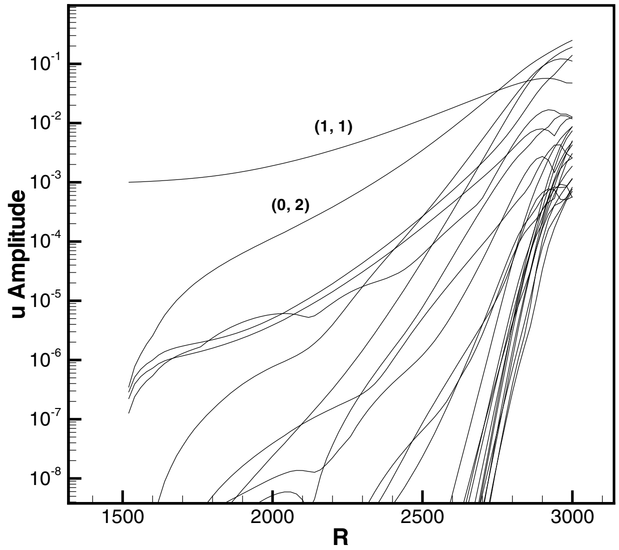

Figure 33: Evolution of disturbance amplitudes for Mach 2 flat plate that undergoes oblique-mode breakdown for all participating Fourier modes.

Based on the linear PSE results shown in the previous chapter, we pick the most amplified disturbance with a frequency of 12 kHz and a spanwise wavelength of 8 mm for the oblique-mode breakdown simulation. By running LASTRAC with the linear PSE option, we found a nonparallel eigenvalue (α) of $(0.0149649, 8.58088e - 05)$ at $R = 1500$. We impose an initial amplitude of $.1\%$ for this oblique mode. The input file is:

```
use_mthread = true
nthread = 8

nonl_pse_calc  = true
n_mode_omega = 4,  n_mode_beta = 4,
n_init_modes = 1,
nonl_alpha_init = (0.0149649,8.58088e-05)
nonl_amp_init = (0.001, 0)
nonl_mode_init = (1, 1)
nonl_alpha_scaling_mode_t = (1, 1)
nonl_dealiasing = true
```

```
      nonl_relax_coef = 0.5

      nonl_convg_tolerance = 1.e-4
      max_nonl_iteration = 200

      nonl_inc_mfd_jac = true

      num_normal_pts       = 101

      mflow_filename = "../../meanflow/mflow.m2fp.linux"

      marching_method_2d         =    along_re
      init_re =  1500,  final_re = 5000, step_re = 20,

      freq_unit = in_hertz_freq
      beta_unit = in_mm_beta

      freq = 12e3,
      beta = 8

      qp_approx          = false
```

Notice that the scaling mode for this case is the excited $(1, 1)$ mode. Only the $(1, 1)$ mode needs to be specified; the $(1, -1)$ mode is also excited internally based on the default symmetry condition. Only four Fourier modes are kept in both temporal and spanwise directions. The resulting streamwise velocity amplitude spectrum versus Reynolds number is shown in figure 33. The skin friction is plotted in figure 34. The rapid rise near the end of the run illustrates that the flow is transitional.

7.4. Nonlinear Development of Crossflow Instability

We use the Mach 2, 80^o swept wing boundary layer studied in the previous chapter to demonstrate nonlinear crossflow instability calculations. For stationary crossflow instability, the most amplified mode has a spanwise wave length of $\lambda_z = 10$ mm. In this example, we excite two stationary modes with spanwise wavelengths of 8 and 10 mm.

```
//
//  Mach 2 80 degree swept wing
//
      use_mthread = true
      nthread = 8
      nonl_pse_calc  = true

      n_mode_omega = 0,  n_mode_beta = 20,
      n_init_modes = 2,
      nonl_alpha_init = (-0.292164,-0.0298771), (-0.373376,-0.028974)
      nonl_dadx_init = (0.00233852,-0.000142725), (0.00295343,-0.000225542)

      nonl_amp_init = (0.001, 0), (0.001, 0)
      nonl_mode_init = (0, 4), (0, 5)
      nonl_alpha_scaling_mode_s = (0, 5)
      mode_symm = beta_symm_only
      nonl_dealiasing = true
      nonl_inc_mfd_jac = true

      nonl_convg_tolerance = 1.e-3
      max_nonl_iteration = 200

      mflow_filename = "../../meanflow/mflow.m2sw80.linux"
      grid_type = wall_cluster

      marching_method_2d    = along_station
       init_station         = 12
      final_station         =  139

      pns_approx = true
      pns_sigma = 0.

      strm_curvt = true
      freq_unit  = in_hertz_freq
      beta_unit = in_mm_beta

      freq = 0  beta = 40
      qp_approx = false
```

Here, the fundamental wavelength is set to 40 to include both the 8- and 10-mm modes in the same Fourier series. Initial values of α and $d\alpha/dx$ are obtained from the corresponding linear PSE runs. We keep only 20 positive spanwise modes ($N = 20$). Initial amplitudes are .1% for both excited modes. Notice the use of `beta_symm_only` for nonlinear crossflow instability. We also turn on the PNS approximation `pns_approx = true` and `pns_sigma = 0` to suppress any elliptic behavior. These two parameter settings result in neglecting the streamwise pressure shape-function gradient variation entirely. They are not always necessary for nonlinear runs; however, these two parameters are used in this case to stabilize nonlinear iterations for the insufficiently resolved mean flow associated with the spectral grid used by the boundary layer code. For well-resolved mean flows, setting `pns_omega = 0` is not necessary. The scaling mode for this case could be either the $(0, 4)$ or $(0, 5)$.

Figure 35 shows the resulting streamwise velocity amplitudes. The 8-mm mode dominates for a short distance and then decays. Nonlinear interaction of the two excited modes feeds energy to other harmonics such as $(0, 1)$, $(0, 2)$, $(0, 3)$, and $(0, 8)$; these harmonics become very energetic near the end of the simulation.

We can further add a traveling crossflow mode by changing a few lines in the input file:

```
...
      n_mode_omega = 4,  n_mode_beta = 20,
      n_init_modes = 3,
      nonl_alpha_init = (-0.292164,-0.0298771), (-0.354517,-0.0271069)
       (-0.0991061,-0.0673744)
      nonl_dadx_init = (0.00233852,-0.000142725), (0.00232006,-4.67329e-05)
              (0.00180321,-0.00110323)
      nonl_amp_init = (0.001, 0), (0.001, 0), (1.e-6, 0)
      nonl_mode_init = (0, 4), (0, 5), (1, 4)
      nonl_alpha_scaling_mode_s = (0, 5)
      nonl_alpha_scaling_mode_t = (1, 4)
```

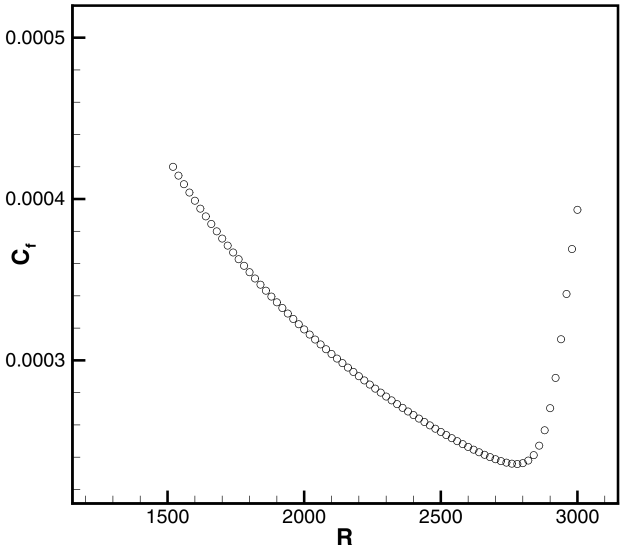

Figure 34: Skin friction coefficient versus Reynolds number for Mach 2 flat plate boundary layer that undergoes oblique-mode breakdown.

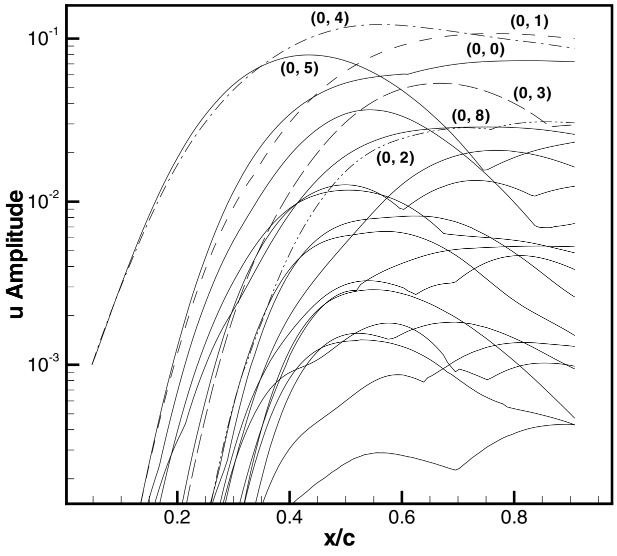

Figure 35: Evolution of disturbance amplitudes for Mach 2, 80^o swept wing boundary layer; nonlinear interaction of stationary crossflow instability with $\lambda_z = 8$ for $(0, 5)$ mode and $\lambda_z = 10$ for $(0, 4)$ mode.

```
...
      freq = 25e3  beta = 40
```

Here, we have $M = 4$ and $N = 20$. The fundamental frequency is no longer zero. The input traveling mode has a frequency of 25 kHz and a wavelength of 10 mm; thus, it is the $(1, 4)$ mode. We also need to set the scaling mode for unsteady Fourier modes to be the $(1, 4)$ mode, which is the only unsteady mode excited. In this simulation, the Fourier modes in the range of $-4 \leq m \leq 4,\ 0 \leq n \leq 20$ will be computed, and the rest of the modes are assumed to follow the symmetry condition. Nonlinear iterations failed to converge at a location where the maximum amplitude reached about 6% when we ran LASTRAC with this input file. We restarted LASTRAC by adding the following:

```
        nonl_relax_coef = 0.2
        nonl_restart = true
        nonl_restart_fn = "Nonl_restart.dat.f0n1_25k"
```

The under-relaxation factor is reduced to `0.2`; and the restart flag is turned on together with a prescribed binary restart file name. This input file carried the run further into the nonlinear regime. The results are shown in figure 36 for various Fourier harmonics. The traveling wave ($(1, 4)$ mode) grows along with the excited stationary modes. Nonlinear interactions cause all three excited modes to evolve differently as compared to the pure stationary case shown in figure 35.

This example shows that LASTRAC can handle nonlinear interactions of multiple eigenmodes. Coupled with the receptivity model, we may simulate transition by using a rather efficient nonlinear PSE procedure. To initiate the spectrum, we can use eigenmodes from a linear PSE calculation as given in the previous example. Only a few important modes need to be specified, and other participating modes may be initiated by setting `nonl_alpha_init = 0` and a small amplitude. Alternatively, users may feed in disturbance mode profiles from external sources, such as experimental data, to initiate the excited modes.

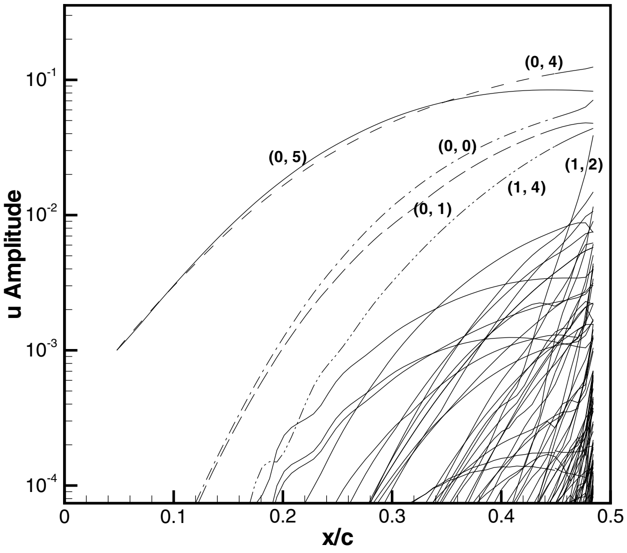

Figure 36: Evolution of disturbance amplitudes for Mach 2, 80^o swept wing boundary layer; nonlinear interaction of stationary and traveling crossflow instability with $\lambda_z = 8$ for $(0, 5)$ mode, $\lambda_z = 10$ for $(0, 4)$ mode, and $f = 25$ kHz, $\lambda_z = 10$ for (14) mode.

Chapter 8

Summary

Predicting transition onset remains a daunting task even with state-of-the-art computing facilities. At the lowest fidelity, transition may be predicted by using a prescribed N-factor and a simple linear stability or parabolized stability theory. Depending on the accuracy requirement, the N value may come from empirical correlations with existing wind tunnel or flight experiments, or from a commonly used value such as 10. The major issue with this approach, in addition to the prediction tool, lies in the N-value itself. According to past experiences, the transition N-value may vary from 2 to 3 in a noisy facility to 10 or higher in flight. Thus, for a new configuration, determining the N-value is itself a difficult task. Despite this uncertainty, N-factor correlation remains the most viable method for transition prediction to date because of its simplicity. The Langley Stability and Transition Analysis Code (LASTRAC) provides both linear stability theory (LST) and parabolized stability equations (PSE) based N-factor correlation capabilities.

For a given mean flow, a possible unstable frequency range is suggested by LASTRAC. By using the maximizing N-factor option, users may obtain an unstable parameter range of disturbance frequency and spanwise wavelength with minimum effort and little prior knowledge of the mean flow under consideration. Further linear calculations using either quasi-parallel LST or nonparallel PSE may be launched to obtain an envelope of N-factors formed by a broad range of unstable modes. The N-factor envelope can then be used for transition correlations or predictions.

LASTRAC can also compute disturbance evolution based on an absolute amplitude. Nonlinear PSE calculations may be performed for a number of unstable modes with a finite amplitude. We demonstrated several test cases in which the boundary layer was perturbed with a given amplitude, and transition was captured by using the nonlinear PSE option. With a specified initial amplitude for all participating Fourier modes, LASTRAC offers a transition prediction tool that may be used to compute transition onset without any modeling or N-factor assumptions. The initial disturbance amplitude can be obtained by using a linearized Navier-Stokes (LNS) or direct numerical simulation (DNS) solver, or by incorporating a receptivity model that will be implemented in LASTRAC in the near future.

LASTRAC is currently used by various industrial organizations and academia for stability calculations and transition predictions. Parametric studies for several supersonic laminar flow control concepts were also investigated using LASTRAC. The test cases shown in this manual cover a broad range of flow configurations. In addition to the traditional N-factor method, LASTRAC offers a comprehensive set of options based on the state-of-the-art numerical methods that may be used for stability calculations and transition predictions in an integrated fashion.

Bibliography

[1] Wlesien, R. W. and Veitch, L., "Quiet Supersonic Platform Program Advanced Algorithms for Design and Optimization of Quiet Supersonic Platforms," AIAA Paper-2002-0143, 2002.

[2] Chang, C.-L., Malik, M. R., Erlebacher, G., and Hussaini, M. Y., "Compressible Stability of Growing Boundary Layers Using Parabolized Stability Equations," AIAA Paper-1991-1636, 1991.

[3] Malik, M., Li, F., Choudhari, M., and Chang, C.-L., "Secondary Instability of Crossflow Vortices and Swept-wing Boundary-Layer Transition," *J. Fluid Mech.*, Vol. 399, 1999, pp. 85–115.

[4] Saric, W. S., Carrillo, R. B., and Reibert, M. S., "Leading-Edge Roughness as a Transition Control Mechanism," AIAA Paper, 98-0781, 1998.

[5] Chang, C.-L., "Thermal Effect on Crossflow Instability in Supersonic Boundary Layers," 14th US National Congress of Theoretical and Applied Mechanics, 2002.

[6] Chang, C.-L. and Malik, M. R., "Oblique-Mode Breakdown and Secondary Instability in Supersonic Boundary Layers," *J. Fluid Mech.*, Vol. 273, 1994, pp. 323–359.

[7] Stetson, K. F., Thompson, E. R., Donaldson, J. C., and Siler, L. G., "Laminar Boundary Layer Stability Experiments on a Cone at Mach 8 Part I: Sharp Cone," AIAA Paper-83-1761, 1983.

[8] Lachowicz, J. T., Chokani, N., and Wilkinson, S. P., "Boundary-Layer Stability Measurements in a Hypersonic Quiet Tunnel," *AIAA J.*, Vol. 34, 1996, pp. 2496–2500.

[9] Horvath, T. J., Berry, S. A., Hollis, B. R., Chang, C.-L., and Singer, B. A., "Boundary Layer Transition On Slender Cones In Conventional and Low Disturbance Mach 6 Wind Tunnels," AIAA Paper-2002-2743, 2002.

[10] Fedorov, A., "Receptivity of High Speed Boundary Layer to Acoustic Disturbances," AIAA Paper-2002-2846, 2002.

[11] Zhong, X. and Ma, Y., "Receptivity and Linear Stability of Stetson's Mach 8 Blunt Cone Stability Experiments," AIAA Paper-2002-2849, 2002.

[12] Fedorov, A. and Tumin, A., "Evolution of Disturbances in Entropy Layer on a Blunted Plate in Supersonic Flow," AIAA Paper-2002-2847, 2002.

[13] Saric, W. S. and Reed, H. L., "Supersonic Laminar Flow Control on Swept Wings Using Distributed Roughness," AIAA Paper-2002-0147, 2002.

[14] Choudhari, M. M. and Streett, C. L., "A Finite Reynolds-number Approach for the Prediction of Boundary-layer Receptivity in Localized Regions," *Phys. Fluids A*, Vol. 4, No. 11, 1992, pp. 2495–2514.

[15] Crouch, J. D., "Receptivity of Three-dimensional Boundary Layers," AIAA Paper-1993-0074, 1993.

[16] Streett, C. L., "Direct Harmonic Navier-Stokes Methods for Efficient Simulation of Weakly-Nonlinear Wave Packets," AIAA Paper-1998-0784, 1998.

[17] Yan, G., Malik, M. R., and Chang, C.-L., "A Solution Adaptive Approach for Computation of Linear Waves," AIAA Paper-1997-2072, 1997.

[18] Rubinstein, R. and Choudhari, M., "Statistical Prediction of Laminar-Turbulent Transition," ICASE Report, 2000-50, 2000.

[19] Choudhari, M., Chang, C.-L., Streett, C. L., and Balakumar, P., "Integrated Transition Prediction: A Case Study in Supersonic Laminar Flow Control," AIAA Paper-2003-0973, 2003.

[20] Herbert, T., "Boundary-Layer Transition – Analysis and Prediction Revised," AIAA Paper-2002-0147, 2002.

[21] Vigneron, Y. C., Rakich, J. V., and Tannehill, J. C., "Calculation of Supersonic Viscous Flow Over Delta Wings with Sharp Supersonic Leading Edges," AIAA Paper, 78-1337, 1978.

[22] Li, F. and Malik, M. R., "On the Nature of PSE Approximation," *Theoret. and Comput. Fluid Dynamics*, Vol. 8, 1996, pp. 253–273.

[23] Chang, C.-L. and Malik, M. R., "Non-Parallel Stability of Compressible Boundary Layers," AIAA Paper-1993-2912, 1993.

[24] Malik, M. R., Chuang, S., and Hussaini, M. Y., "Accurate Numerical Solution of Compressible, Linear Stability Equations," *ZAMP*, Vol. 33, 1982, pp. 189.

[25] Thompson, K. W., "Time Dependent Boundary Conditions for Hyperbolic Systems," *J. Comput. Phys.*, Vol. 68, 1987, pp. 1–24.

[26] Chang, C.-L., Malik, M. R., and Hussaini, M. Y., "Effects of Shock on the Stability of Hypersonic Boundary Layers," AIAA Paper-1990-1448, 1990.

[27] Chang, C.-L., Vinh, H., and Malik, M. R., "Hypersonic Boundary-Layer Stability with Chemical Reactions using PSE," AIAA Paper-1997-2012, 1997.

[28] Malik, M. R., "Numerical Methods for Hypersonic Boundary Layer Stability," *J. Comput. Physics*, 1989, pp. 376–413.

[29] Anderson, D. A., Tannehill, J. C., and Pletcher, R. H., "Computational Fluid Mechanics and Heat Transfer," *McGraw-Hill Book Company*, 1984, pp. 250–251.

[30] El-Hady, N. M., "Non-Parallel Instability of Supersonic and Hypersonic Boundary Layers," AIAA Paper-1991-0324, 1991.

[31] Saric, W. S. and Nayfeh, A. H., “Non-Parallel Stability of Boundary-Layer Flows,” *Phys. Fluids*, Vol. 18, 1975.

[32] Rumbaugh, J., Jacobson, I., and Booch, G., *The Unified Modeling Language Reference Manual*, Addison Wesley, 1999.

[33] Gamma, E., Helm, R., Johnson, R., and Vlissides, J., *Design Patterns*, Addison Wesley, 1995.

[34] Stepanov, A. and Lee, M., “The Standard Template Library,” Technical Report HPL-95-11 (R.1), Hewlett-Packard Laboratories, 1995.

[35] Chang, C.-L., Malik, M. R., and Vinh, H., “Linear and Nonlinear Stability of Compressible Swept-Wing Boundary Layers,” AIAA Paper-1995-2278, 1995.

[36] Reibert, M., Saric, W. S., Carrillo, R. B. J., and Chapman, K. L., “Experiments in Nonlinear Saturation of Stationary Crossflow Vortices in a Swept-Wing Boundary Layer,” AIAA Paper, 96-0184, 1996.

[37] Kachanov, Y. S. and Levchenko, V., “The Resonant Interaction of Disturbances at Laminar-turbulent Transition in a Boundary Layer,” *J. Fluid Mech.*, Vol. 138, 1984, pp. 209.

[38] Jiang, L., Chang, C.-L., Choudhari, M. M., and Liu, C., “Cross-Validation of DNS and PSE Results for Instability-Wave Propagation,” AIAA Paper-2003-3555, 2003.

[39] Harris, J. E. and Blanchard, D. K., “Computer Program for Solving Laminar, Transitional, or Turbulent Compressible Boundary-Layer Equations for Two-Dimensional and Axisymmetric Flow,” NASA Technical Memorandum 83207, 1982.

[40] Pruett, C. D., “A Spectrally Accurate Boundary-Layer Code for Infinite Swept Wings,” NASA Contractor Report 195014, 1994.

[41] Wie, Y. S., “BLSTA - A Boundary Layer Code for Stability Analysis,” NASA Contractor Report 4481, 1992.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 01- 06 - 2004	**2. REPORT TYPE** Technical Memorandum	**3. DATES COVERED** *(From - To)*

4. TITLE AND SUBTITLE Langley Stability and Transition Analysis Code (LASTRAC) Version 1.2 User Manual	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Chang, Chau-Lyan	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER 23-762-45-ME
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) NASA Langley Research Center Hampton, VA 23681-2199	**8. PERFORMING ORGANIZATION REPORT NUMBER** L-19022
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) National Aeronautics and Space Administration Washington, DC 20546-0001	**10. SPONSOR/MONITOR'S ACRONYM(S)** NASA
	11. SPONSOR/MONITOR'S REPORT NUMBER(S) NASA/TM-2004-213233

12. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 02
Availability: NASA CASI (301) 621-0390 Distribution: Nonstandard

13. SUPPLEMENTARY NOTES
An electronic version can be found at http://techreports.larc.nasa.gov/ltrs/ or http://ntrs.nasa.gov

14. ABSTRACT

LASTRAC is a general-purposed, physics-based transition prediction code released by NASA for Laminar Flow Control studies and transition research. The design and development of the LASTRAC code is aimed at providing an engineering tool that is easy to use and yet capable of dealing with a broad range of transition related issues. It was written from scratch based on the state-of-the-art numerical methods for stability analysis and modern software technologies. At low fidelity, it allows users to perform linear stability analysis and N-factor transition correlation for a broad range of flow regimes and configurations by using either the linear stability theory or linear parabolized stability equations method. At high fidelity, users may use nonlinear PSE to track finite-amplitude disturbances until the skin friction rise. This document describes the governing equations, numerical methods, code development, detailed description of input/output parameters, and case studies for the current release of LASTRAC.

15. SUBJECT TERMS
Stability; Transition Prediction; N-factor; PSE; Direct numerical simulation

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** STI Help Desk (email: help@sti.nasa.gov)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** ***(Include area code)***
U	U	U	UU	105	(301) 621-0390