COMPUTATIONAL EVALUATION OF A TRANSONIC LAMINAR-FLOW WING GLOVE DESIGN

A Thesis

by

MATTHEW WILLIAM ROBERTS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

May 2012

Major Subject: Aerospace Engineering

COMPUTATIONAL EVALUATION OF A TRANSONIC LAMINAR-FLOW WING GLOVE DESIGN

A Thesis

by

MATTHEW WILLIAM ROBERTS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Helen Reed
Committee Members,	William Saric
	Hamn-Ching Chen
Head of Department,	Dimitris Lagoudas

May 2012

Major Subject: Aerospace Engineering

ABSTRACT

Computational Evaluation of a Transonic Laminar-Flow Wing Glove Design. (May 2012)

Matthew William Roberts, B.S., Kansas State University

Chair of Advisory Committee: Dr. Helen Reed

The aerodynamic benefits of laminar flow have long made it a sought-after attribute in aircraft design. By laminarizing portions of an aircraft, such as the wing or empennage, significant reductions in drag could be achieved, reducing fuel burn rate and increasing range. In addition to environmental benefits, the economic implications of improved fuel efficiency could be substantial due to the upward trend of fuel prices. This is especially true for the commercial aviation industry, where fuel usage is high and fuel expense as a percent of total operating cost is high.

Transition from laminar to turbulent flow can be caused by several different transition mechanisms, but the crossflow instability present in swept-wing boundary layers remains the primary obstacle to overcome. One promising technique that could be used to control the crossflow instability is the use of spanwise-periodic discrete roughness elements (DREs). The Flight Research Laboratory (FRL) at Texas A&M University has already shown that an array of DREs can successfully delay transition beyond its natural location in flight at chord Reynolds numbers of 8.0×10^6. The next step is to apply DRE technology at Reynolds numbers between 20×10^6 and 30×10^6, characteristic of transport aircraft.

NASA’s Environmentally Responsible Aviation Project has sponsored a transonic laminar-flow wing glove experiment further exploring the capabilities of DRE technology. The experiment will be carried out jointly by FRL, the NASA Langley Research Center, and the NASA Dryden Flight Research Center. Upon completion of a wing glove design, a thorough computational evaluation was necessary to determine if the design can meet the experimental requirements. First, representative CAD models of the testbed aircraft and wing glove were created. Next, a computational grid was generated employing these CAD models. Following this step, full-aircraft CFD flowfield calculations were completed at a variety of flight conditions. Finally, these flowfield data were used to perform boundary-layer stability calculations for the wing glove. Based on the results generated by flowfield and stability calculations, conclusions and recommendations regarding design effectiveness were made, providing guidance for the experiment as it moved beyond the design phase.

ACKNOWLEDGMENTS

For as far back as I can remember I have had an intense fascination with aviation and when the time came to choose a path after completing my undergraduate degree, I could never have imagined how important and fulfilling my time here at Texas A&M would be. There are numerous people and organizations to thank for the contributions they made towards my development as a better engineer. I'll begin with my advisor Dr. Helen Reed. Without ever meeting me in person, she was willing to extend an invitation to join her research group. Any time I had a question, she always took time out of her busy schedule to help. Her guidance over the past two-and-a-half years is much appreciated. I would also like to extend thanks to the other two members of my committee: Dr. William Saric and Dr. Hamn-Ching Chen. Their considerable knowledge of subject matter related to my research was invaluable.

The work completed for this thesis was the product of collaboration or assistance from several organizations. First, the SARGE experiment is supported by NASA's Environmentally Responsible Aviation Project and conducted in partnership with the Dryden Flight Research Center and the Langley Research Center. From Dryden, I would like to thank Fletcher Hartshorn, Trong Bui, Ethan Baumann, Aaron Pahs, Dan Nolan, and Kurt Kloesel for their assistance throughout the project. From Langley, I would like thank the staff of ERA and Dr. Mujeeb Malik and his team. Second, the CFD simulations completed would not have been possible without the resources and expertise of the Texas A&M Supercomputing Facility and its staff. Third, I would like to thank ANSYS support for assisting me on questions related to the ICEM meshing software and FLUENT CFD solver.

I would also like to acknowledge the people I worked with on a daily basis here at A&M. First, thanks go out to the members of the SARGE experiment group: Michael Belisle, Maj. Aaron Tucker USAF, Matthew Tufts, Thomas Williams, and Matt Woodruff. This work simply would not have been possible without the collective efforts of the team. Next, I would like to thank former students Richard Rhodes and Tyler Neale for providing the foundation from which my research was built upon. Finally, special thanks

go to Colleen Leatherman for all that she does in keeping the machine that is the Saric-Reed research group well oiled.

Lastly, I would like to acknowledge the grants that made this work possible, including ViGYAN subrecipient grant C10-00350, ATK grant SP00029509, and AFOSR grant A4760.

NOMENCLATURE

α	x-direction wavenumber
β	z-direction (spanwise) wavenumber
ε	Thickness ratio of equivalent ellipse
λ	Spanwise wavelength
λ_{crit}	Critical wavelength
λ_{sub}	Subcritical control wavelength
$\mathit{\Lambda}$	Leading edge sweep angle
μ	Dynamic viscosity
μm	Micrometer
ρ	Density
ω	Angular frequency
A	Disturbance amplitude
A_0	Initial disturbance amplitude
C_L	Lift coefficient
C_p	Pressure coefficient
f	Frequency
ft	Foot
GB	Gigabyte
Hz	Hertz
in	Inch
kg	Kilogram
kN	KiloNewton
k-ε	k-epsilon turbulence model
k-ω	k-omega turbulence model

lb	Pound
m	Meter
M	Mach number
mm	Millimeter
N	Smith-Van Ingen N-factor
p	Pressure
r	Airfoil section nose radius
R	Boundary-layer Reynolds number
$Re_{\theta A.L}$	Attachment- line momentum thickness Reynolds number
Re_c	Chord Reynolds number
T	Temperature
t/c	Thickness-to-chord ratio
TB	Terabyte
u, v, w	Local orthogonal velocity components
u_t, v_t, w_t	Streamwise orthogonal velocity components
U, V, W	Global orthogonal velocity components
U_∞	Freestream velocity
V_0	Surface wall-normal velocity
x/c	Chord fraction
x/c_{tr}	Transition-location chord fraction
x, y, z	Local orthogonal coordinate system components
x_t, y_t, z_t	Streamwise orthogonal coordinate system components
X, Y, Z	Global orthogonal coordinate system components

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

I. INTRODUCTION

The research described in this thesis focuses on efforts to advance technologies related to obtaining laminar flow on transport aircraft. The primary benefit laminar flow presents is a reduced fuel burn rate. Significant fuel, and subsequent fuel cost, savings can be achieved by laminarizing portions of the aircraft, such as the wings and empennage. However, accomplishing this goal is a challenging problem. There are several mechanisms that cause transition from laminar to turbulent flow and a thorough understanding of their behavior is crucial to developing techniques to suppress or eliminate them.

A promising transition control technique using discrete roughness elements (DREs) has recently received considerable attention. They have been shown to delay transition on swept-wings in wind tunnel and low-Reynolds-number flight experiments. To increase the technological readiness level of DREs, a flight experiment using a wing glove on a transonic aircraft has been proposed. Early computational efforts showed that increasing laminar flow at higher Reynolds numbers and operational lift coefficients was feasible. This initiated the computational work covered in this thesis, with the overarching goal of showing that the wing glove design can achieve the project success requirements and effectively demonstrate transition delay using DRE technology.

A. Motivation

Laminar flow has long been a sought-after attribute in aircraft design because of potential gains in aerodynamic efficiency. Thibert *et al.* [1] found that nearly half of the total drag for an aircraft is due to skin friction. Laminar skin friction can be up to 90% less than turbulent skin friction, making laminarization an obvious path towards drag reduction [2]. For subsonic and transonic aircraft, the primary benefit achieved through drag reduction is a reduced fuel burn rate. This in itself has a number of desirable outcomes, including reduced fuel storage (allowing for more payload), increased range, fewer emissions, and perhaps most importantly decreased direct operating costs.

This thesis follows the style of the *AIAA Journal*.

Economy will always be a driving force in aircraft design, especially for the commercial airline industry where profit margins are increasingly narrow. Of all the factors driving an aircraft's operating cost, fuel cost is the most volatile. Data from the Bureau of Transportation Statistics [3] , shown in Fig. 1, clearly illustrate that aviation fuel expenses have risen significantly since 2000.

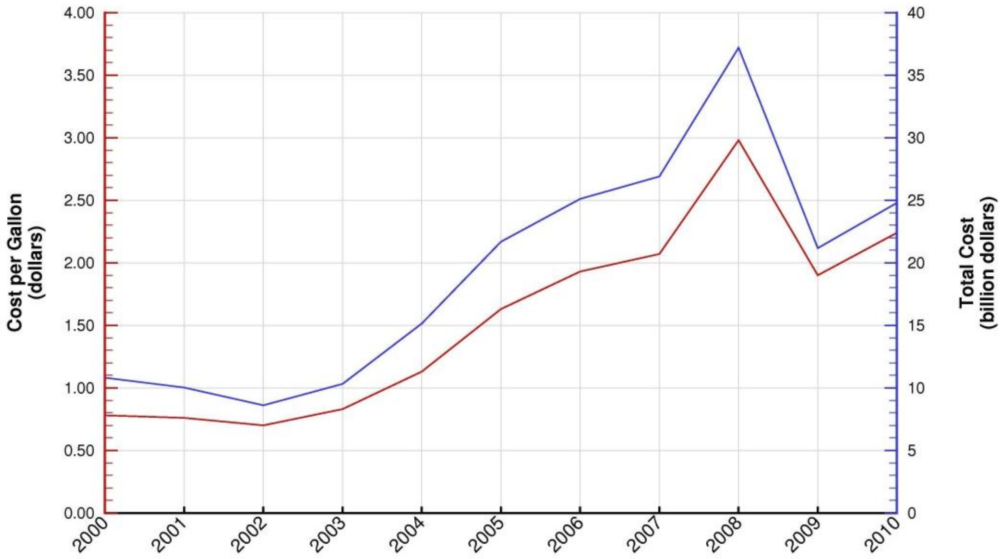


Fig. 1 Total cost and cost per gallon for domestic aviation fuel.

Statistical analysis performed by the Air Transport Association [4] found that over this same period of time, fuel represented a minimum of 12% of total operating expenses in 2002 and a maximum of 32% of total operating expenses in 2008. At the end of 2010, it was 25% of total operating expenses. These numbers are clearly connected to the three-fold increase in fuel prices over the past 11 years. While the future is always uncertain, the likelihood of further cost increases has been an impetus to improve fuel efficiency. Laminar flow is certainly a strong candidate to accomplish this based on the significant role skin friction drag plays in the total drag count. Modest gains in this area have the potential to save hundreds of millions of dollars in yearly fuel expenditures.

B. Transition Mechanisms

Boundary-layer transition from laminar to turbulent flow is a complex and nuanced process and a great deal of effort has been made to better understand the mechanisms that cause it. Reshotko [5] discusses the paths to transition, which are illustrated in Fig. 2.

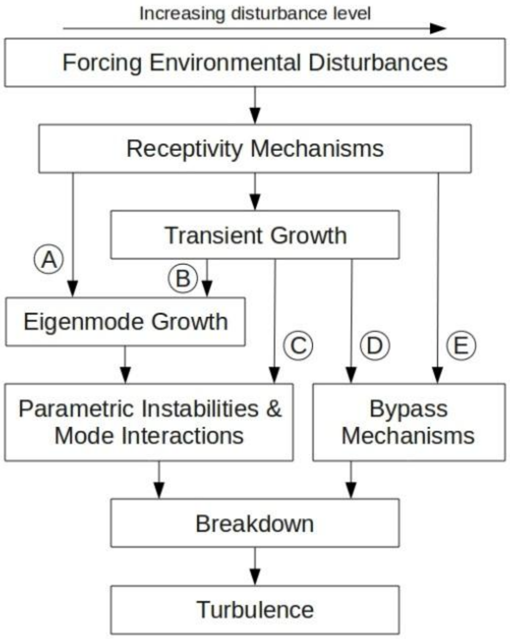

Increasing disturbance level
Forcing Environmental Disturbances
Receptivity Mechanisms
Transient Growth
A
B
Eigenmode Growth
C
D
E
Parametric Instabilities & Mode Interactions
Bypass Mechanisms
Breakdown
Turbulence

Fig. 2 Paths to transition [5].

The process begins with environmental disturbances, such as freestream sound or vorticity, which become entrained in the boundary layer through a process known as receptivity. Receptivity provides the initial conditions that eventually lead to turbulence. Fig. 2 is arranged left-to-right by disturbance level, with path A representing the lowest disturbance level and path E representing the highest disturbance level. Flight is considered to be a low-disturbance environment and since this thesis focuses on boundary-layer transition for aircraft, only path A will be considered. Eigenmode growth represents the initial growth of a boundary-layer disturbance and is described by linear stability theory (LST). As the disturbance continues to grow, 3-D effects and nonlinearities arise through parametric instabilities and mode interactions. Eventually, the mean flow will have been modified enough to form high-frequency secondary instabilities, at which point disturbance growth is very rapid over a short length scale and breakdown to turbulence occurs. The mechanisms that begin with eigenmode growth and may eventually lead to transition are described below.

Curvature-Induced Instability (Görtler Vortices)

Surfaces with slight concave curvature in external flows are susceptible to a curvature-induced instability in the form of Görtler vortices [6]. These steady, counter-rotating, and streamwise oriented vortices cause convection of streamwise momentum that eventually produce large changes in the boundary-layer mean flow. Secondary instabilities set in and breakdown to turbulence occurs. Fortunately, transition due to the Görtler instability can be avoided by eliminating concave surface curvature in the region of interest, *e.g.* the first 60% *x/c* of an airfoil.

Attachment-Line Contamination and Instability

Aircraft with swept wings are susceptible to attachment-line contamination because turbulence generated along the fuselage can travel chordwise and spanwise along the forward attachment line of the wing and trip the boundary layer [7]. Limiting the attachment-line momentum thickness Reynolds number, shown by Equation 1, below a critical value of 100 avoids contamination [8].

$$Re_{\theta A.L.} = 0.404\sqrt{\frac{\rho U_{\infty} r \sin^2 \Lambda}{\mu(1+\epsilon)\cos\Lambda}} \tag{1}$$

Contamination can also be prevented by starting a new boundary layer at the leading-edge junction of the wing and fuselage. This is accomplished actively by using a suction patch [7] or passively with a small bump that creates a stagnation point between the fuselage and wing [9]. Even if attachment-line contamination is eliminated, a shear-layer instability along the attachment line is still present. However, the $Re_{\theta A.L.}$ upper limit can now be increased to the suggested design criterion of 245 [8]. Controlling the sweep angle and the normal-to-the-LE nose radius is typically the easiest way to satisfy either of these design criteria.

Streamwise Instability (Tollmien-Schlichting Waves)

Streamwise instabilities in the form of Tollmien-Schlichting (T-S) waves are an important aspect of the transition process because they affect both unswept and swept wings at all flight conditions. Reviews of this mechanism can be found in Saric and Reed [10] and Reed and Saric [8]. T-S waves are traveling waves that appear in the mid-chord region of airfoils and are most unstable in regions with flat or adverse pressure gradients. LST characterizes the instability waves and reasonably predicts transition. However, the receptivity process has an appreciable influence on the T-S mechanism, which is sensitive to freestream sound and 2-D surface roughness such as steps and gaps, and as such should technically be governed by an initial value problem. For aircraft flying at higher subsonic Mach numbers, compressibility effects are stabilizing to T-S when compared to incompressible analysis. If supersonic flow occurs locally, *e.g.* a supercritical wing, the most unstable disturbance is 3-D rather than 2-D.

Crossflow Instability

A thorough overview of the crossflow instability is given by Saric *et al.* [11]. The presence of both swept geometry and a pressure gradient generate curved inviscid streamlines, causing a reduction in streamwise velocity and leading to an imbalance in centripetal acceleration and pressure gradient. This creates a boundary-layer component called crossflow that is perpendicular to the inviscid streamline. Fig. 3 illustrates this concept.

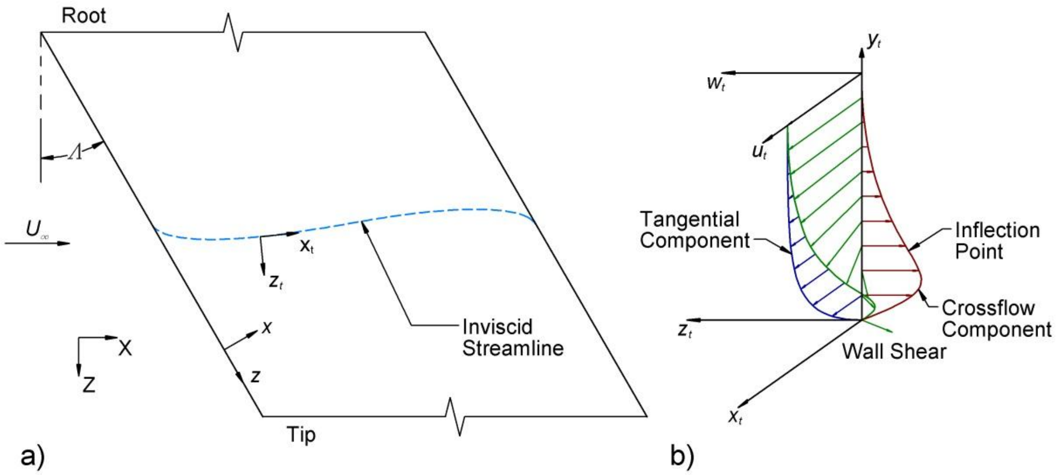


Fig. 3 Swept-wing flow field behavior. a) inviscid streamline for a swept wing and b) 3-D boundary-layer profile for swept wings.

The crossflow boundary-layer component contains an inflection point, providing a source for the strong inviscid instability described by Rayleigh and Fjortøft's criteria. Unlike T-S waves, crossflow exists as traveling and stationary waves, though transition is usually caused by one or the other and not both. Traveling waves are more important for high-turbulence environments while stationary waves dominate in low-disturbance environments [12]. The physical structure of the crossflow instability can be described as co-rotating vortices aligned within a few degrees of the inviscid streamlines. Crossflow vortices are responsible for the modification of the mean flow in which high momentum is convected down and low momentum fluid is convected up. This creates strong distortions in the streamwise velocity and initiates the development of nonlinear effects. As these distortions grow, the boundary-layer velocity profiles at particular locations become double inflected and inviscidly unstable, precipitating a high frequency secondary instability and eminent breakdown to turbulence. Experimental results from Reibert *et al.* [13] and supporting computational efforts by Haynes and Reed [14] showed that the crossflow mechanism is nonlinear in nature and verified that transition prediction for crossflow using linear techniques is not as accurate. This nonlinear behavior requires an understanding of how receptivity affects the crossflow instability. Radeztsky *et al.* [15] showed surface roughness plays a significant role in determining transition location. Crossflow is also sensitive to freestream turbulence and, contrary to T-S, can be further destabilized by a favorable pressure gradient.

C. Laminar Flow Control

By understanding the mechanisms responsible for transition to turbulence, strategies can be formed to control the process. The Gortlër and attachment line mechanisms are relatively simple to control and approaches to eliminate them were discussed above. Control of T-S and crossflow, however, is much more challenging. Knowledge gained from LST has shown that a fuller boundary-layer profile (more negative $\partial^2 U/\partial y^2$) generally results in lower disturbance growth. The boundary-layer equation near the wall, shown by Equation 2, gives insight as to how this can be achieved [16].

$$\mu \frac{\partial^2 U}{\partial y^2} = \frac{\partial p}{\partial x} + \rho V_0 \frac{\partial U}{dy} - \frac{\partial \mu}{\partial T}\frac{\partial T}{\partial y}\frac{\partial U}{\partial y}, \;\; y \approx 0 \tag{2}$$

Several possibilities exist to make the left hand side of (2) more negative: accelerating pressure gradients *(*$\partial p/\partial x < 0$), wall-normal suction ($V_0 < 0$), wall cooling in air *(*$\partial T/\partial y > 0$; $\partial \mu/\partial T > 0$), and wall heating in water ($\partial T/\partial y < 0$; $\partial \mu/\partial T < 0$). Using this information, the techniques described below have been developed to delay transition to turbulence.

Natural Laminar Flow

Natural laminar flow (NLF) is a passive technique generally used to suppress the T-S instability. T-S is stabilized by a favorable pressure gradient and can therefore be controlled over much of the wing by moving the pressure minimum as far aft as possible and avoiding any regions with an adverse pressure gradient ahead of that point [16]. However, an accelerated pressure gradient does not suppress crossflow and, in fact, will only destabilize it further. Therefore, a NLF swept wing designed to prevent T-S disturbance growth will be ineffective for crossflow-dominated transition characteristic of swept-wing aircraft [10].

Laminar Flow Control and Hybrid Laminar Flow Control

Active systems using wall-normal suction and wall cooling/heating represent another possibility for transition control. Laminar flow control (LFC) describes the use of an active control system over the whole-wing chord. Hybrid laminar flow control (HLFC) employs a LFC system near the leading edge in addition to NLF over much of the wing, reducing the complexity of a whole-wing LFC system. A great deal of attention has been given to active control systems over the years. Joslin [2] offers an excellent review of the numerous laminar flow experiments conducted from the 1930s to the 1990s. Suction is the primary method used in LFC and HLFC. In addition to other issues, thermal control has a limited effect on the crossflow instability and is therefore not a viable control technique for swept wings [10]. HLFC has certainly been shown to be an effective method at extending laminar flow on swept wings. However, there are expected manufacturing and operational concerns that detract from the possible benefits gained from implementing such a system.

Swept-Wing Laminar Flow Control

Crossflow represents the crux of the transition problem for swept-wing aircraft since techniques exist to limit all the other mechanisms. Until recently, it was thought that boundary-layer suction was the only method available to control crossflow. Wind tunnel experiments at Arizona State University showed that another possibility exists, one that uses passive nonlinear biasing of stationary crossflow wave growth near the attachment line [16]. This idea is the foundation of swept-wing laminar flow control (SWLFC). SWLFC is possible in low-disturbance environments, such as flight and low-turbulence wind tunnels, because the stationary crossflow instability dominates.

Wind tunnel experiments performed by Reibert *et al.* [13] on a swept wing showed that the crossflow waves could be made more uniform by placing spanwise arrays of micron-sized roughness elements near the attachment line at chord Reynolds numbers ranging from 1.6×10^6 to 3.2×10^6. The application of spanwise periodic DREs led to two important observations: (1) unstable waves occurred only at integer values of the primary disturbance wavenumber and (2) no subharmonic disturbances were destabilized. This means a DRE spacing of λ excites spanwise wavelengths of λ, $\lambda/2$, $\lambda/3$,…, λ /n, but not intermediate wavelengths or wavelengths greater than λ.

Saric *et al.* [17] continued this work, exploring DRE spacings that were smaller, or subcritical, to the most unstable wavelength that occurred naturally. A subcritical mode typically grows early and then decays, which by itself won't cause transition. The key outcome of initially forcing a subcritical wavelength was the modification of the mean flow, which suppressed all larger wavelengths, including the most unstable mode. The net effect was a significant increase in laminar flow on the wing.

Using these lessons, Saric and Reed [16] outlined design criteria for SWLFC. The primary goals are to eliminate streamwise and other instabilities and encourage crossflow, allowing shorter wavelengths to grow sufficiently and early enough for control of the most unstable wavelength. NLF principles of pressure-distribution shaping can be used to prevent the T-S instability. To implement DREs for laminar flow control, the most unstable stationary crossflow wavelength, λ_{crit}, must be found. LST accurately predicts the critical wavelength and the point it first becomes unstable, also called the Branch I neutral point. Next, the possibilities of subcritical control wavelengths, λ_{sub}, must be explored. A good control candidate should exhibit strong early growth but decay after O(40%) chord and would preferably be about half the critical wavelength. The pressure distribution can be tailored to accomplish this, under the condition that T-S is still sufficiently suppressed. Radeztsky et al. [15] showed that small roughness placed too far downstream had no influence on the boundary layer. Thus, to be most effective, the DRE array should be placed at the Branch I neutral point of the critical wavelength, which is typically within the first few percent of chord.

Direct numerical simulations by Wasserman and Kloker [18] have further verified the SWLFC technique under the afore-mentioned wind tunnel conditions. However, wind tunnel experiments are unable to exactly replicate the higher chord Reynolds number achieved during flight where SWLFC would ultimately be used. The next step is therefore to move beyond ground-based facilities and demonstrate DRE technology through a series of flight experiments.

D. Predecessors to Current Work

Much of the work described in this thesis builds upon past experimental and computational efforts. The Swept Wing In-Flight Testing (SWIFT) program and early transonic aircraft computations provided this framework and are described below.

SWIFT Program

Representing the next logical step in demonstrating the SWLFC technique after wind tunnel and computational successes, the SWIFT program was developed by the Texas A&M Flight Research Laboratory (FRL) to determine the effectiveness of DRE technology on a 30° swept-wing model at chord Reynolds numbers between 6.5×10^6 and 8.0×10^6. Carpenter *et al.* [19] details many of the experimental aspects of the program. The SWIFT model itself is mounted vertically under the port wing of a Cessna O-2A Skymaster, shown in Fig. 4.

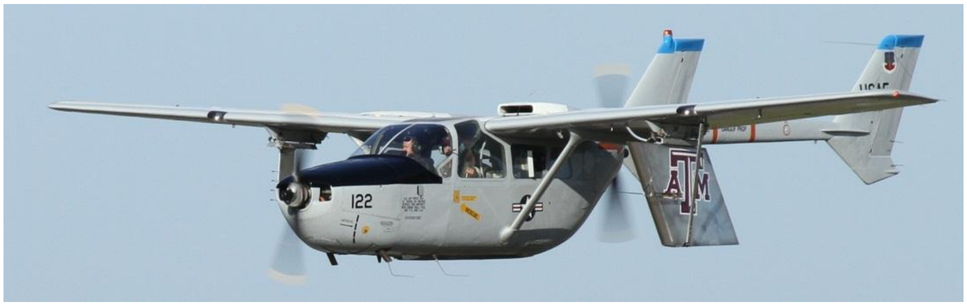


Fig. 4 The FRL O-2A Skymaster with the SWIFT model.

The SWIFT model has a host of instrumentation options, including infrared (IR) thermography, an air data probe, pressure ports, thermocouples, hotfilms, and Preston tubes. IR thermography is the primary method to determine the extent of laminar flow. Because turbulent flow has a higher heat transfer rate than laminar flow, the transition location can be visualized on a surface experiencing a temperature differential with the flow. This technique is advantageous because it gives a complete view of the surface, rather than a collection of discrete data points, and because it is non-obtrusive to the flow. The typical procedure for a test flight involves first climbing to an altitude of roughly 10,500 feet and cold-soaking the model until it is at a uniform temperature. The aircraft then performs a dive, using the temperature lapse rate of the atmosphere to create the temperature differential between the air and the wall. Turbulent regions of the SWIFT model's surface are slightly warmer than laminar regions and can be distinguished by the IR camera. The dive is also beneficial in that it increases the Reynolds number to the fit within the desired range. The air data probe provides freestream measurements near the SWIFT model. These measurements are critical to the pilot, who must hold the conditions constant during the test, the experimentalist, who must know the conditions for comparison to other data, and the computationalist, who must match the conditions when setting the farfield boundary conditions. Two rows of pressure ports are used to correlate experimental and computational pressure readings and serve as simulation validation prior to the completion of boundary-layer stability calculations. Thermocouples, hotfilms, and Preston tubes are also used depending on the test objectives.

Companion computational work for the SWIFT experiments is outlined in Rhodes [20] and Rhodes *et al.* [21]. Obtaining an accurate representation of the actual aircraft and test model is a critical first step in the CFD process. FRL generated 3-D CAD models of the 0-2A and the SWIFT test article, but this geometry was far too complex for CFD purposes. A CFD model was constructed, simplifying fine details, eliminating geometry with negligible influence on the test article, and ensuring regions of the aircraft near the test article were accurately modeled.

Once a suitable CAD model was generated, several computational grids were created. A hybrid meshing technique was implemented to properly capture the flowfield. This included a coarse, structured freestream region to efficiently capture the uniform flowfield, an unstructured region around the aircraft to capture the relatively complex aircraft geometry, and a fine, structured region around the SWIFT model to capture the boundary layer. Interfaces were used between different zones. Grids of varying refinement were used to judge grid independence.

Flowfield computations were completed, with post-processing attention focused on the region near the SWIFT model. CFD and experimental results for the pressure distribution on the test article had good agreement, validating the CFD process. It was determined during this comparison that the influence from nearby bodies, such as the fuselage and wing, needed to be considered. This had a direct impact on the location of the air data probe used to measure freestream conditions.

Boundary-layer stability calculations were also performed to determine stability characteristics of the SWIFT model, such as the most unstable mode, possible control wavelengths, and DRE array placement location. Boundary-layer information was extracted directly from a CFD solution using a user-defined function (UDF) within the CFD software. The UDF modified the existing mesh near the SWIFT model and exported the flowfield data in a useful format. Stability calculations were then completed and analyzed. The information obtained from this process was used in concert with experimental efforts.

The SWIFT program experiments focusing on the SWLFC technique produced some remarkable results, which are given in Saric *et al.* [22]. Using an operationally-realistic painted leading edge with no DREs at $Re_c = 8.0\times10^6$, transition was observed around 25-30% chord. When DREs were applied at a subcritical spacing of one-half the critical wavelength, transition moved back to 60% chord. A comparison is shown in Fig. 5, with the cooler laminar regions designated by darker colors.

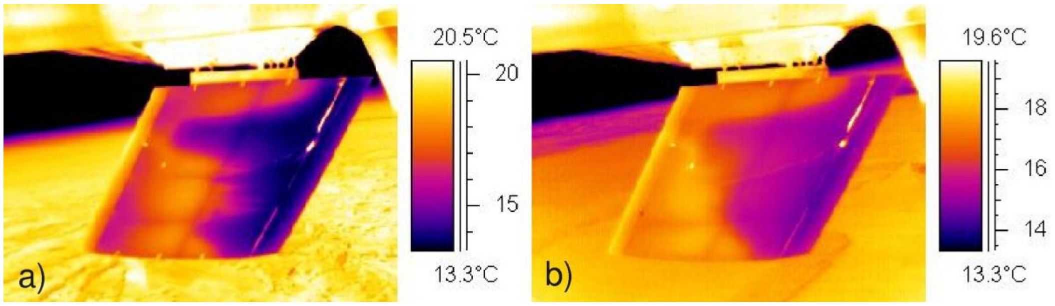


Fig. 5 SWIFT IR thermography, painted leading edge. a) no DREs at $Re_c = 8.00\times10^6$, b) DREs applied at $Re_c = 8.13\times10^6$. [22]

The DREs doubled the amount of laminar flow, demonstrating SWLFC at a chord Reynolds number of 8.0×10^6. While this is certainly a significant achievement, this technology is targeted for commercial aircraft, which typically cruise at chord Reynolds numbers between 20×10^6 and 30×10^6. The next step is therefore to design a flight experiment that can demonstrate SWLFC at similar transport-aircraft Reynolds numbers.

Transonic Aircraft Computations

A laminar-flow experiment using a transonic aircraft outfitted with a wing glove was chosen by FRL to demonstrate DRE technology at chord Reynolds numbers between 15×10^6 and 20×10^6. However, there was significant groundwork that was necessary prior to designing this experiment to demonstrate proof-of-concept, which can be found in Neale [23] and Belisle *et al.* [24]. The first step was to select a suitable testbed. Three aircraft were identified as potential experimental platforms based on chord and unit Reynolds numbers, wing leading-edge sweep, and fuselage-mounted engine requirements: the Gulfstream II (G-II), Gulfstream III (G-III), and Sabreliner 80 (S-80). Due to overall geometry and performance attributes, but primarily because of expected availability, the G-II (SP variant) was selected (Fig. 6).

Fig. 6 The NASA G-II/SP based in El Paso, TX. [24]

Next, steps similar to those taken by Rhodes [20] were completed. This started with the generation of a CAD model, which involved a laser scan of the G-II outer mold line (OML). A number of lessons were learned during this process, such as ensuring curvature continuity on the wings, and could be applied to future scans. A hybrid computational grid was created for the G-II geometry, focusing on properly resolving the flowfield near the wing. A primary concern for wing glove placement on the wing was the extent of engine effects. The engine inlet creates a high-pressure region in the flowfield, causing the pressure distribution on the wing to be adversely affected. Proper modeling of engine effects was achieved through an iterative process, which then allowed the wing's pressure distribution to be examined. It was determined that any unfavorable engine effects had dissipated beyond 45% semispan. The wing glove should therefore be placed outboard of this region.

Finally, to show that the SWLFC technique was feasible at larger Reynolds numbers, an initial glove airfoil section was designed, following the SWLFC guidelines described earlier. The pressure distribution of the airfoil section was then used with a boundary-layer solver to provide flowfield data for boundary-layer stability calculations. These calculations showed that it was indeed possible to generate a glove airfoil section with a transport-relevant lift coefficient and pressure distribution conducive to DRE control. With the proof-of-concept stage complete, FRL was ready to begin a laminar-flow wing glove experiment.

E. Overview

The Environmentally Responsible Aviation (ERA) Project was created by NASA to explore the feasibility, benefits, and technical risks associated with aircraft concepts that would reduce the impact of aviation on the environment. A laminar-flow wing glove experiment implementing DREs was one such concept that could contribute to the ERA's overall goal of reducing mission fuel burn by 50%. FRL at Texas A&M, NASA's Langley Research Center, and NASA's Dryden Flight Research Center (DFRC) at Edwards Air Force Base were tasked with designing and building the glove and performing the flight experiment. This work would focus on successfully demonstrating NLF and SWLFC techniques under the aerodynamic requirements shown in Table 1. These parameters can be readily achieved by the three business jets identified by Neale [23].

Table 1 Aerodynamic requirements for the laminar flow wing glove experiment.

Parameter	Value
NLF Chord Reynolds Number	15 - 22×10^6
SWLFC Chord Reynolds Number	22 - 30×10^6
Unit Reynolds Number	≥ 1.4×10^6 1/ft
Mach Number	≥ 0.72
Glove Section Lift Coefficient	≥ 0.5
Glove Leading Edge Sweep	≥ 30°

In January 2010, NASA DFRC acquired a G-III, designated the SubsoniC Research Aircraft Testbed (SCRAT), in anticipation for several projects, including the laminar-flow wing glove experiment. Because SCRAT would be a dedicated aircraft for this project and because it would be based at Dryden where the test flights would take place, the G-II originally identified as the experimental platform was no longer necessary. Though very similar to the G-II, the G-III is a generation newer and has a larger wing, both in span and in chord, actually making it a better candidate for the experiment. The glove experiment was named the Subsonic Aircraft Roughness Glove Experiment (SARGE) by FRL and is projected to be the first experiment to be flown on SCRAT. With the testbed aircraft set, SARGE design and analysis work could begin.

Due to the challenges associated with obtaining laminar flow, comprehensive measures must be taken during the design and analysis stages to ensure mission success. This thesis addresses the computational evaluation of the SARGE wing glove design in its installed configuration on the SCRAT testbed aircraft. Two principal areas will be investigated: 1) the aerodynamic effects of the flowfield created by both the aircraft and wing glove and 2) the boundary-layer stability characteristics of the wing glove under the influence of a full-aircraft flowfield. The central goal is to determine the effectiveness of the current glove design for the demonstration of DRE technology at operationally relevant Reynolds numbers.

This document is organized into sections by analysis subtasks. Section II describes the flight test configuration for the experiment, discussing both the aircraft testbed and the wing glove. Section III covers the development of the CAD models used in full-aircraft CFD simulations. Grid generation and grid independence studies are discussed in Section IV. A description of the full-aircraft CFD simulation process and subsequent results can be found in Section V. Section VI discusses the methods used to perform boundary-layer stability calculations and reviews the results. Finally, a summary of the work and conclusions drawn from the analysis are given in Section VII.

II. FLIGHT TEST CONFIGURATION

An understanding of the flight test configuration is important for the planning process of the entire project. This is certainly true for all computational analysis efforts since any detail overlooked at this stage could have major repercussions on the entire study. This section covers both the SCRAT and SARGE configurations.

A. SCRAT Configuration

The Gulfstream III is an executive jet built by Gulfstream Aerospace Corporation that made its first flight in December 1979 and received FAA certification in September 1980 [25]. A total of 206 G-IIIs were manufactured before production ended in 1986. The aircraft is powered by two Rolls-Royce Spey Mk 511-8 turbojet engines rated at 50.7 kN (11,400 lb static) each. This is the same engine model used on the G-II. Important geometric, operational, and performance specifications can be found in Table 2. The G-III that would become SCRAT was built in 1982 and had several previous owners before arriving at DFRC. SCRAT has a tail number of N804NA and is shown outside of its hangar at Dryden in Fig. 7.

Table 2 G-III specifications. [25]

Parameter	Value
Wing Span	23.72 m (77 ft 10 in)
Overall Length	25.32 m (83 ft 1 in)
Overall Height	7.43 m (24 ft 4½ in)
Gross Wing Area	86.83 m^2 (934.6 ft^2)
Wing Leading Edge Sweep	31.7°
Maximum Takeoff Weight	31,615 kg (69,000 lb)
Long-range Cruising Speed	Mach 0.77
Maximum Cruising Speed	Mach 0.85
Maximum Operating Altitude	13,720 m (45,000 ft)

Because SCRAT is intended to be used as a research testbed, it has undergone several modifications to meet service requirements such as replacing the original cabin interior with work stations, instrumentation racks, and camera mounts, adding additional exterior aircraft instrumentation, and installing a pilot flight display in the cockpit. SCRAT's test flight configuration will be project dependent since a variety of experiments are anticipated.

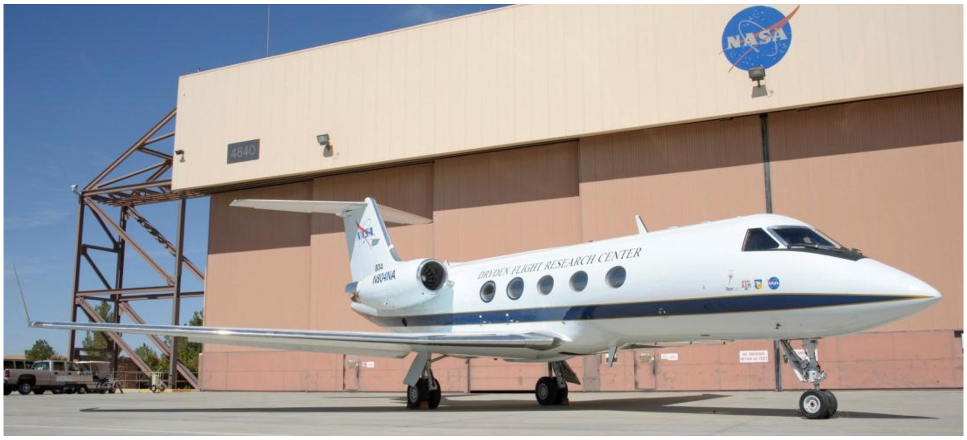


Fig. 7 SCRAT outside its hangar at Dryden Flight Research Center.

B. SARGE Configuration

The SARGE wing glove, which will also be referred to as SARGE, will be attached to the port wing of SCRAT as shown in Fig. 8.

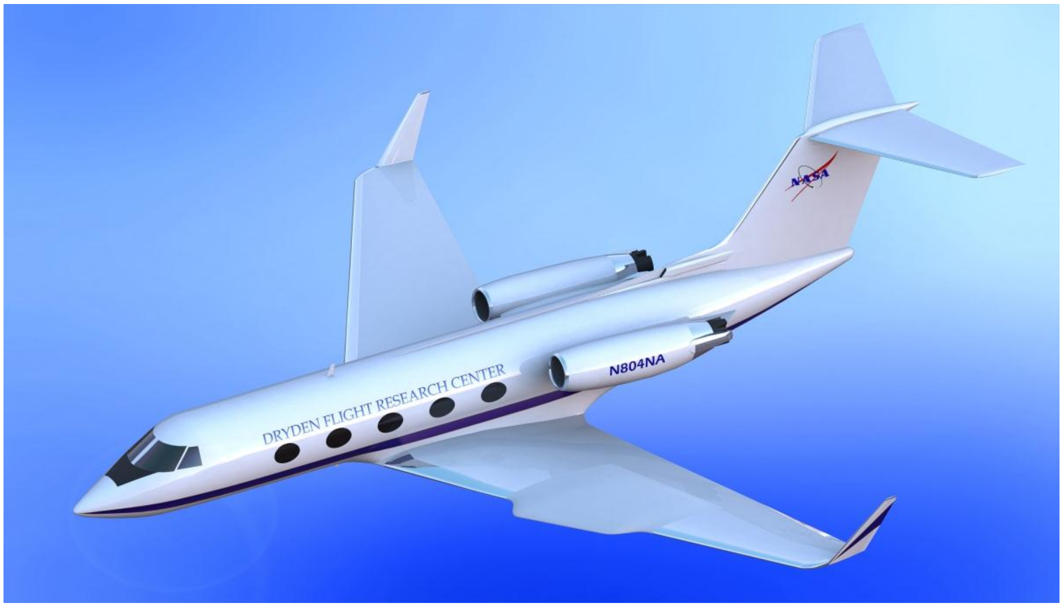


Fig. 8 Rendering of SCRAT with SARGE installed on the port wing.

Constructed from composite materials, SARGE will be offset from the outer mold line (OML) of the wing. Because much of the SCRAT wing serves as a fuel storage tank, all SARGE components must remain on the exterior of the wing. Fabrication considerations also require a minimum offset of 50.8 mm (2 in) from wing OML to SARGE test section OML on the suction side, providing clearance for mounting structure and instrumentation wire routing. SARGE will be made up of several distinct components: an interchangeable test section leading edge, a test section center panel, upper and lower aft blending sections, and inboard and outboard fairings (Fig. 9).

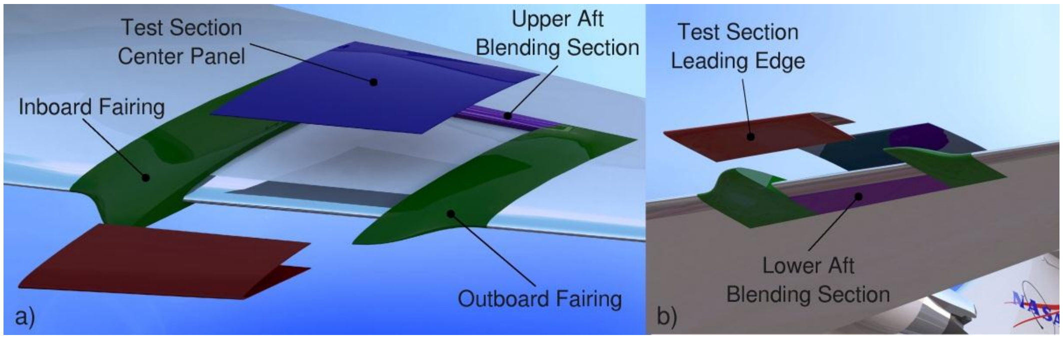


Fig. 9 SARGE components. a) suction side, b) pressure side.

The interchangeable leading edge allows for several configurations to be used, such as a leading edge outfitted with pressure ports, a polished leading edge, and an operationally-realistic painted leading edge. The upper and lower aft blending sections create a smooth chordwise transition from the test section leading edge and center panel back to the wing OML. The inboard and outboard fairings do the same in the spanwise direction. The inboard fairing also serves as a large Gaster bump, preventing attachment-line contamination. Maintaining continuous surface curvature at all component boundaries is critical because discontinuities in curvature cause undesirable pressure fluctuations which can introduce boundary-layer disturbances.

The OML of SARGE has undergone several design changes since the inception of this project. The information reported in this thesis is from the most recent configuration. The general size, shape, and location of SARGE has remained fairly constant through each design iteration due to various requirements and restrictions. Fig. 10 shows the placement and extent of SARGE.

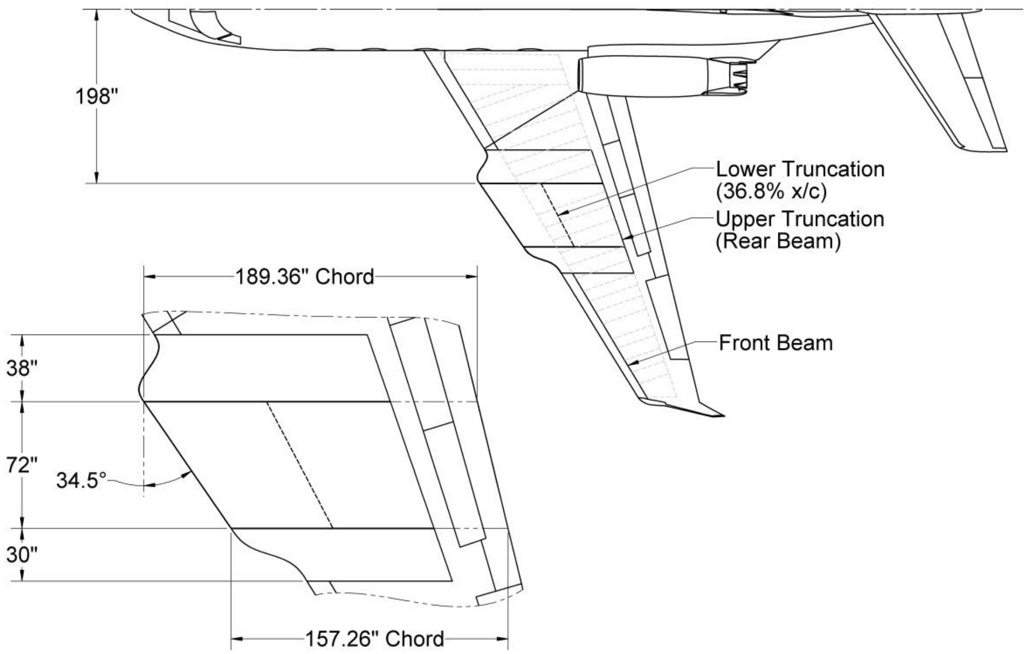


Fig. 10 Planform layout of SARGE.

The test section width was set to provide enough surface area to generate the required amount of laminar flow in the spanwise direction. Turbulent wedges form near the test section leading edge corners, effectively reducing the spanwise extent of laminar flow as the flow moves aft. Using an empirical method for turbulent flow propagation, this span allows for more than half the test section to be conducive to laminar flow back to 60% *x/c*. The glove design employs a conic layout, in that the leading edge line of the glove, the leading edge of the wing, and the trailing edge line of the wing all intersect at a focus. A balance between structural considerations and Reynolds numbers, both chord and unit, for the glove established the inboard glove chord, from which the outboard chord and leading edge sweep could be derived based on the conic design. Control surfaces, primarily the spoilers/speed brakes, limited the aft extent of the glove on the suction side. The rear beam of the wing was chosen as the truncation point (roughly 73% glove *x/c*). Thickness-to-chord considerations led to the truncation of the glove much earlier on the pressure side.

The spanwise location of the glove was determined by referencing the CFD work performed on the G-II. It was found that the engine effects could be accounted for inboard of 45% semispan for the G-II.

Even though the G-III has a larger wingspan, 45% semispan was conservatively chosen as the inboard extent of the glove. This corresponds to a buttock line (BL) of 198, *i.e.* 198 inches from the aircraft symmetry plane. All span locations will be reference by their BL number. The inboard fairing width was set to avoid the G-III wing break because the surface discontinuity would complicate fairing design and fabrication. The outboard fairing width was set to keep all portions of SARGE inboard of the port aileron so as not to adversely affect the flow over the control surface.

In early designs, the glove OML was generated by lofting SWLFC airfoils together. However, it was discovered that this method was inadequate due to a non-uniform spanwise pressure distribution on the test section surface. This issue is discussed further in Section V-B. To produce a uniform pressure distribution that accounted for all 3-D flowfield effects generated by the glove and aircraft, the full-potential CFD code TRANAIR was used to optimize to the glove OML. Fig. 11 shows optimized airfoil sections at three span locations: the inboard test section boundary at BL198, the middle of the test section at BL234, and the outboard test section boundary at BL270. Examination of the optimized glove airfoil profiles results in several important findings. First, the 2-inch clearance between the wing OML and glove OML is maintained on the suction side back to the upper aft blending. Second, there are no regions of concave curvature on the glove OML forward of 60% *x/c*, preventing curvature-induced instabilities. Third, the leading-edge nose radius across the entire glove is small enough to eliminate both attachment line contamination and instabilities.

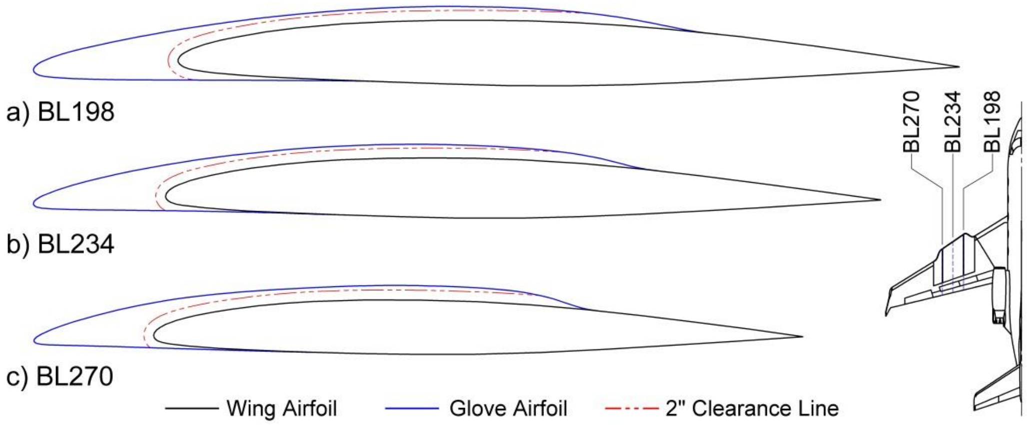


Fig. 11 SARGE airfoil sections. a) BL198, b) BL234, and c) BL270.

Like the SWIFT model, SARGE will be heavily instrumented. IR thermography will again serve as the primary laminar-flow detection method on the suction side. Pressure ports will be used to establish the envelope for the science test flights as well as provide verification for CFD efforts. An air data probe acts as a truth source for the local freestream conditions. Thermocouples and hotfilms will measure the glove wall temperature and frequency spectra respectively, which can be used to determine the transition location. This will be the only laminar-flow detection method on the pressure side and will supplement IR thermography on the suction side. Fig. 12 shows a conceptual instrumentation suite for SARGE. The domain of influence is also shown in this diagram, illustrating areas where laminar flow will most likely be found. Because the streamlines over the glove are deflected inboard, the turbulent wedges marking the sides of the domain of influence are also skewed inboard.

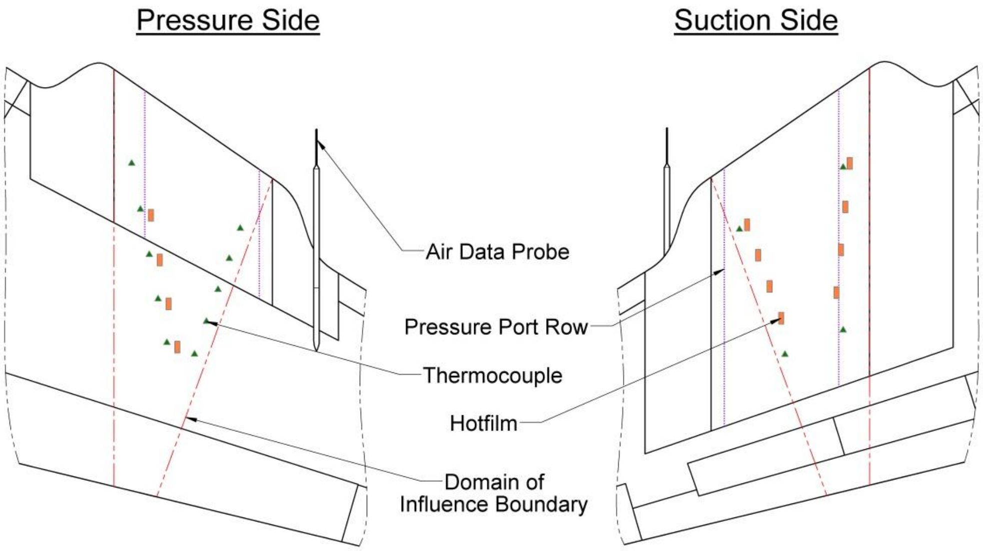


Fig. 12 Conceptual instrumentation suite for SARGE.

III. CAD MODEL DEVELOPMENT

CAD model development is the foundation to the CFD efforts described in this thesis. If the CAD model is not representative of the real geometry, there can be no hope in obtaining accurate computational results. Fortunately, the testbed aircraft is available to generate a well-defined model that can be used not only for this experiment, but for future projects as well. This section discusses the development of clean-wing, glove-wing, and CFD models.

A. Clean-Wing Models

The first step in creating a CAD model of SCRAT involved a laser scan of the aircraft, similar to the process taken with the G-II. Direct Dimensions, Inc. (DDI) was contracted to perform the scan. The equipment used had a single-point accuracy of less than 1 mm from a scanning distance of 15 m [26]. The scan produces a data point cloud from which surfaces can be fit. To reduce the scan time and model generation time, only the port side of the aircraft was scanned. Symmetry was assumed for the starboard side in the event that a model of the entire aircraft was required. The resulting geometry was delivered in a neutral CAD format so all major CAD software packages could be used for post-processing work.

The original SCRAT model created by DDI still required a great deal of work before it would be ready for use in grid generation. Several small issues, such as missing, duplicate, or overlapping faces, were present and needed to be corrected. To rectify these problems and to perform all subsequent CAD-related tasks, the SolidWorks CAD package [27] was used. Once an error-free baseline aircraft model was complete, the next step was to generate a series of clean-wing models that could later be outfitted with the SARGE wing glove. Past CFD experience with the O-2A and G-II showed that the empennage had a negligible effect on the SWIFT model and the G-II wing, leading to its exclusion from all SCRAT models used by FRL. This *a priori* assumption would later be validated by simulations completed by DFRC that included the empennage. Even though the port and starboard sides of the aircraft would differ due to the presence of the glove, it would be much less expensive computationally to assume symmetric flow and only model the port half of the aircraft. If the flow was asymmetric, *i.e.* nonzero sideslip, a model of the entire aircraft would be necessary. It was anticipated that a majority, if not all, simulations performed during the design analysis stage of this project would have symmetric flow. This eventually proved to be true. Fig. 13 shows a clean-wing CAD model of SCRAT.

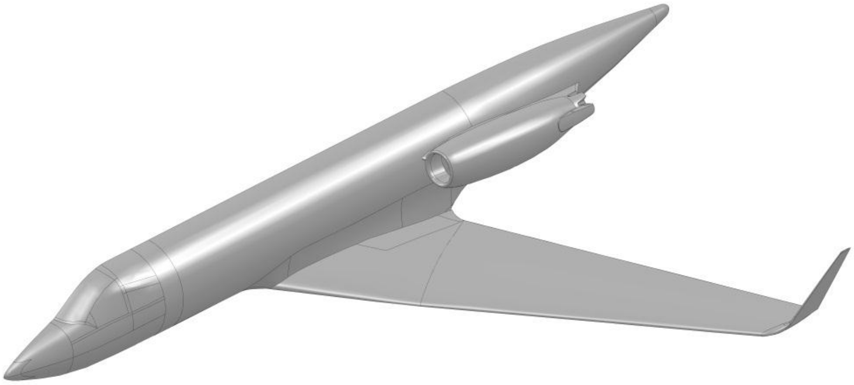


Fig. 13 SCRAT clean-wing CAD model.

Consideration of expected CAD and CFD tasks led to the identification and creation of several clean-wing model types. They include full-detail, engine-simplified, no-engine, control-surface, layout, and wing-section models. Once the SARGE OML CAD model was generated, it could simply be added to each of the clean-wing models to create glove-wing models. A detailed explanation of the roles each of these models would serve can be found in the following section.

B. Glove-Wing Models

With the clean-wing models in place, the next step was to incorporate the model of SARGE to create glove-wing models. Due to the expected use of various models over several glove design iterations, it became apparent that some form of configuration control would be necessary. A naming scheme was first created to distinguish SARGE configurations and aircraft model types from each other. The format, which is detailed by the SARGE Numbering System drawing found in Appendix A, consists of the originator, layout number, airfoil number, and model type suffix. For example, the SARGE configuration presented in this thesis is TAMU-06-05. It encompasses all the different model types, such as TAMU-06-05-FD (full-detail) or TAMU-06-05-ES (engine-simplified). Layout and airfoil numbers increase sequentially with each new design iteration, meaning the current design is the sixth planform layout and fifth airfoil set.

For each new SARGE configuration, a set of engineering drawings was generated for reference and record keeping purposes. All drawings underwent a typical engineering release process with the goal of increasing design assessment and adding accountability by not only FRL, but by DFRC as well. All involved parties were required to approve the design before the aircraft models would be released for design analysis. A drawing revision process was also implemented in the event that errors or omissions were discovered on released drawings. For each release cycle, two drawings were created: a Glove Planform Layout drawing and a Glove Airfoil Profiles drawing. The drawings for the TAMU-06-05 configuration expand upon the geometric information given in Fig. 10 and Fig. 11 and can be found in Appendix A.

Each glove-wing model serves a different role in the overall CAD development process. The full-detail model contains all geometric information except the empennage created from the laser scanning process. Using the clean-wing full-detail model as a base, the SARGE OML was generated and incorporated to form the glove-wing full-detail model. The TRANAIR optimization process used to define the TAMU-06-05 SARGE OML outputted a series of streamwise and spanwise curves that were lofted together in SolidWorks to form all glove surfaces. Once the glove OML was established, it could simply be combined with the existing aircraft surfaces. Unalterable versions of the glove OML were then copied into the other model types. The full-detail model was also used to create the drawing set of glove airfoil profiles.

Due to the level of geometric complexity found in the full-detail model, it was not suitable for CFD purposes. By eliminating fine details that would have no influence on the flowfield near the glove, an unnecessarily exorbitant cell count could be avoided. This led to the creation of the simplified-engine model, whose namesake comes from the location where the simplifications were made. The original engine model included details like the thrust reverser structure and engine hush kit. These features were removed and replaced with surfaces that blend smoothly into the existing engine nacelle while keeping the outlet location unchanged and the area constant. These modifications are illustrated in Fig. 14.

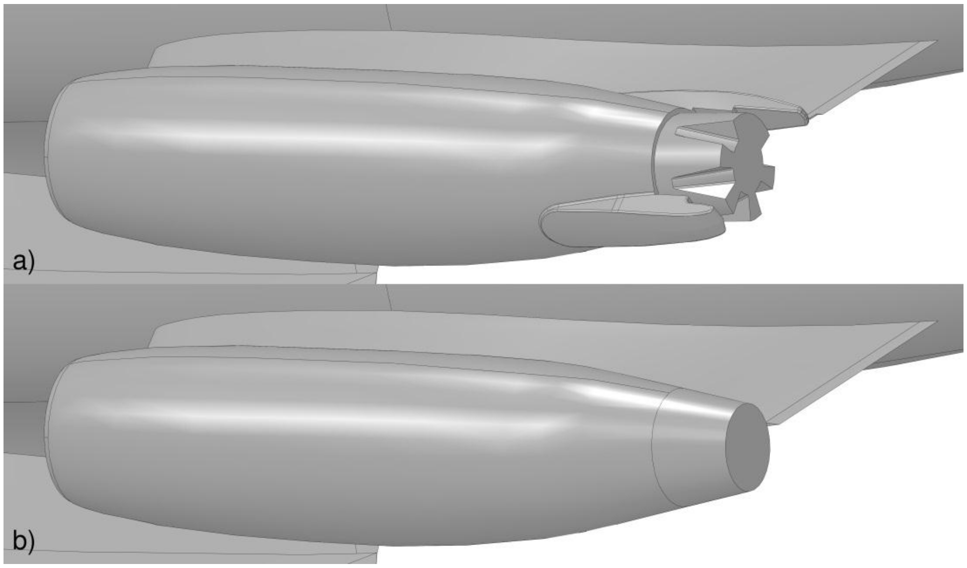


Fig. 14 Comparison of engine geometric detail. a) full-detail model, b) engine-simplified model.

Several other models with less prominent roles were also created. The no-engine model has, as its name implies, no engine nacelle or pylon. It was primarily used early on in the CFD process when modeling the engine was causing issues with simulation stability. Eventually, this problem was solved and the no-engine model was no longer necessary. The control-surface model was outfitted with a wing that included cavities for adjustable control surfaces. This model was used solely by DFRC and was only requested for the first few SARGE design iterations. The layout model was used to generate glove planform layout drawing sets. It did not contain the model of the SARGE OML, rather a layout sketch representing the SARGE planform. Finally, a wing-section model that only included the SCRAT wing and SARGE was used for CFD troubleshooting.

C. CFD Models

The motivation behind creating the engine-simplified model was to obtain geometry that was conducive to the meshing process while still representing the aircraft and glove properly. However, the geometry required to generate the computational grid will include more than just the aircraft model. Geometry defining different grid zones must also be created, and because geometry creation within meshing software is limited when compared to the capabilities of dedicated CAD software, it would be logical to perform these tasks during the CAD model development phase. This process is inherently intertwined with grid generation and CFD solution strategies because these steps dictate the supplemental geometry that is necessary. This section will focus on the different elements of the CFD geometry and how they were created. Further details on what roles the different grid zones serve and how they are meshed can be found in Section IV-A.

The standard computational grid used in full-aircraft CFD simulations is composed of three nested zones: a freestream zone, an aircraft zone, and a glove zone. Due to its size relative to the aircraft, the freestream zone was unable to be modeled in SolidWorks. The aircraft zone, whose boundary is displayed in Fig. 15 as semi-transparent surfaces, surrounds the aircraft. It is a simple rectangular prism with a cutout of the aircraft on the surface that is coplanar with the aircraft symmetry plane.

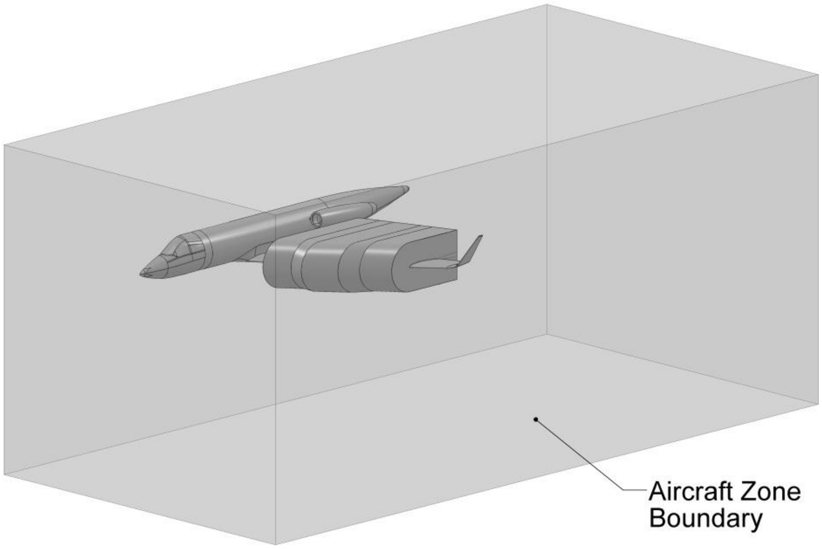


Fig. 15 Aircraft zone boundary used for grid generation.

The glove zone surrounds the glove and a large portion of the wing, as shown in Fig. 16. The zone boundary was created with a series of lofts. Forthcoming CFD simulation strategies would necessitate splitting the glove zone into laminar and turbulent subzones. The laminar subzone would eventually contain flowfield data used to perform boundary-layer stability calculations and as such requires a fine wall-normal mesh resolution. Even though the laminar subzone only surrounds the glove test section, the turbulent portions of the glove zone should also have a well resolved wall-normal mesh. A surface was offset a few inches to better control regions of the mesh near the wing and glove OMLs. Surfaces trailing the offset surface were also created to better manage the mesh in the wake region. The offset surface set is shown in Fig. 17.

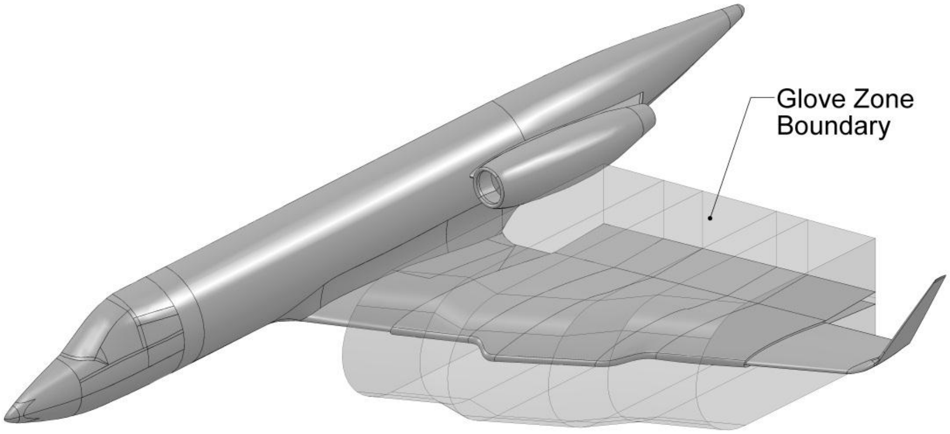


Fig. 16 Glove zone boundary used for grid generation.

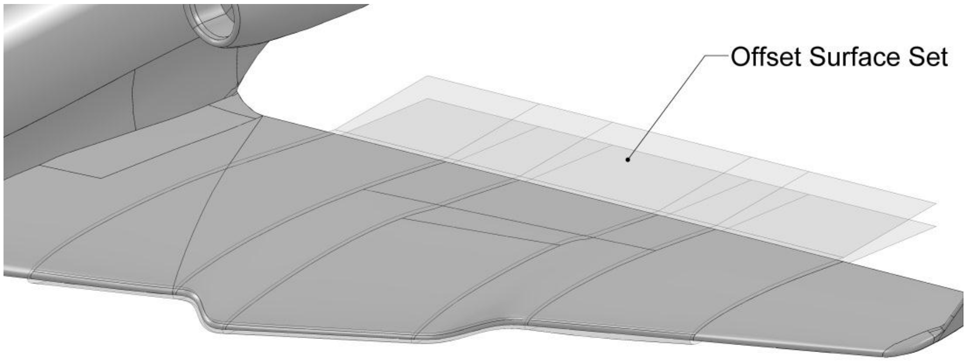


Fig. 17 Offset surface set used for grid generation.

Where necessary, split lines were added to both SCRAT and the supplemental geometry described above in order to help map the mesh to the geometry. This was typically the final step in the CAD development phase. Once all CAD models used for CFD purposes were completed, the next task was to begin generating computational grids in preparation for full-aircraft simulations.

IV. COMPUTATIONAL GRID GENERATION

Grid generation is sometimes described as more of an art than a science because it relies heavily on user experience and intuition. The grid setup is also highly dependent on the problem being solved and can require a significant investment in time and effort to ensure the flowfield will be properly captured. For this project, CAD model development, grid generation, and the full-aircraft CFD simulations were an integrated effort that required constant tuning as new information about the problem was obtained, eventually resulting in a converged meshing methodology. These grid generation techniques and studies that were performed to ensure grid-independent solutions are discussed in detail below.

A. Grid Setup

Full-aircraft grid generation for SARGE began with the examination of work that had already been completed on similar projects, primarily CFD efforts for the O-2A and G-II. Both offered initial guidance in formulating and implementing successful meshing techniques. While many aspects of past grid generation have remained unchanged for the SARGE project, the approach has been updated to align with new software capabilities and overall project CFD goals. The former of these two represents the largest departure in how grids are created. The aging meshing software GAMBIT that had been previously used was replaced with ICEM CFD 13.0. Both programs are part of a commercial package that includes the CFD solver FLUENT, all of which are developed by ANSYS, Inc. The latter considers CFD solution techniques, such as employing a laminar subzone, and flowfield data extraction techniques.

ICEM CFD is a grid generation tool that can parametrically create grids in a variety of formats, such as multiblock structured, mixed unstructured, and Cartesian [28]. For the SARGE project, a hybrid grid format that included both structured and unstructured cells was used. Each grid is composed of geometry, blocking, mesh, and attribute files. The keeps disparate elements of the entire grid separate, providing the user with more precise control. For example, several different meshes can easily be applied to the same geometry without the need to reproduce the geometry, blocking, or attributes. Typically, the grid generation process begins by creating or importing geometry. Next, the blocking and the mesh are generated. Cell and boundary types are then specified and saved in the attribute file. Upon completion of these steps, the grid can be exported from ICEM CFD and is ready for use in CFD simulations. All grid generation work, which may be computationally expensive for large grids, was completed on an Apple Mac Pro workstation outfitted with an open-source Linux operating system. The specifications of the workstation are given in Section V-A.

As mentioned in Section III-C, the grid was divided into freestream, aircraft, and glove zones with the intent of making the entire grid generation process more modular. Each of the zones for the baseline grid and the steps taken to create them are described below. A summary of the baseline grid parameters are given in Section IV-B.

Freestream Zone

Working inward, the freestream zone is the first level of the nested grid structure. It defines the extent of the computational domain and is the largest zone by volume. The freestream zone is blocked and meshed with structured hexahedral cells. This efficiently fills the zone's large volume, allowing for uniform cell size growth away from the aircraft zone boundary in all directions. The cell structure is also aligned with the flow, which would be impossible for unstructured tetrahedral or pyramidal cells. Furthermore, the level of control generally provided by a structured mesh is greater than an unstructured mesh but it comes at a cost of less automation. Fig. 18 displays the freestream zone mesh with the aircraft zone boundary labeled for perspective.

Because it was not included in the CFD model created in SolidWorks, the surfaces, curves, and points defining the boundaries of freestream zone were added in ICEM CFD. Correctly sizing the farfield is an important step for the entire CFD process. If the distances between the domain boundaries and the aircraft are too small, the farfield boundary conditions can improperly influence the flowfield generated by the aircraft and cause unphysical effects. Grid generation guidelines outlined by the Fourth AIAA CFD Drag Prediction Workshop (DPW) suggested that all farfield boundaries remain roughly 100 reference chord-lengths away from all aircraft surfaces [29]. The mean aerodynamic chord of the G-III, which is 4.201 m (165.4 in), was used as the reference chord-length. The contribution of the freestream zone to the total cell count, as will be made apparent in the following section, is relatively small. Therefore, a conservative farfield sizing was used since the extra computational expense would be negligible. An offset distance of

457.2 m (1,500 ft) was chosen forward, above, below, and to the port side of the aircraft. To provide a larger volume for the wake, an offset distance of 914.4 m (3,000 ft) was chosen aft of the aircraft. Section IV-B explores whether the computational domain is sufficiently large and will not affect the validity of the CFD solution.

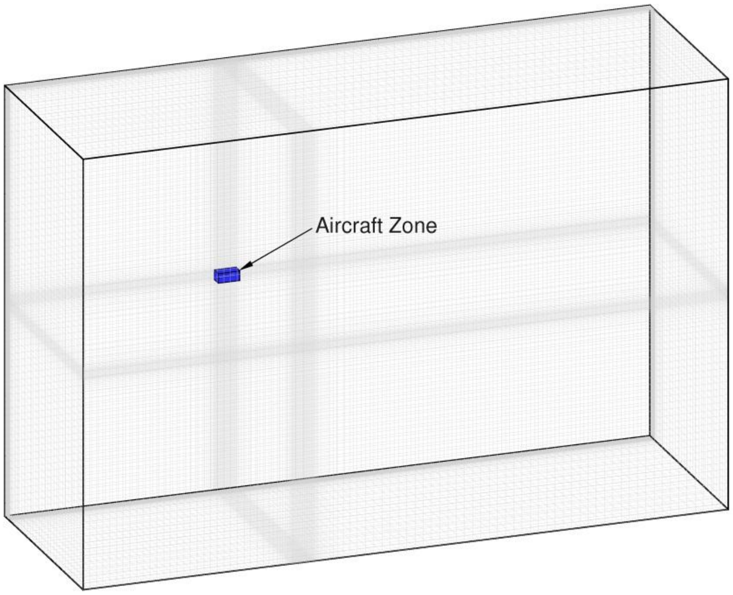


Fig. 18 Freestream zone mesh.

A symmetry boundary condition was assigned to the freestream zone face that is coplanar with the aircraft symmetry plane. The remaining five exterior faces were given a pressure far field boundary condition. An interior condition was set for the freestream zone faces that are shared with the aircraft zone boundary.

Aircraft Zone

The aircraft zone is the second level of the nested grid structure and contains a majority of the aircraft geometry. Due to the complex shape of the aircraft, this zone was meshed with unstructured tetrahedral, pyramidal, and prismatic cells. The typical balance between cell count and computational expense played a role in this decision, as did mesh creation difficulty, and zone purpose. In the end, the use of unstructured elements was preferred because they could provide reasonable mesh refinement, would be much simpler to create, and would sufficiently model the full-aircraft effects near the region of interest –

the glove test section. Fig. 19 shows the aircraft zone mesh and includes the glove zone boundary for reference.

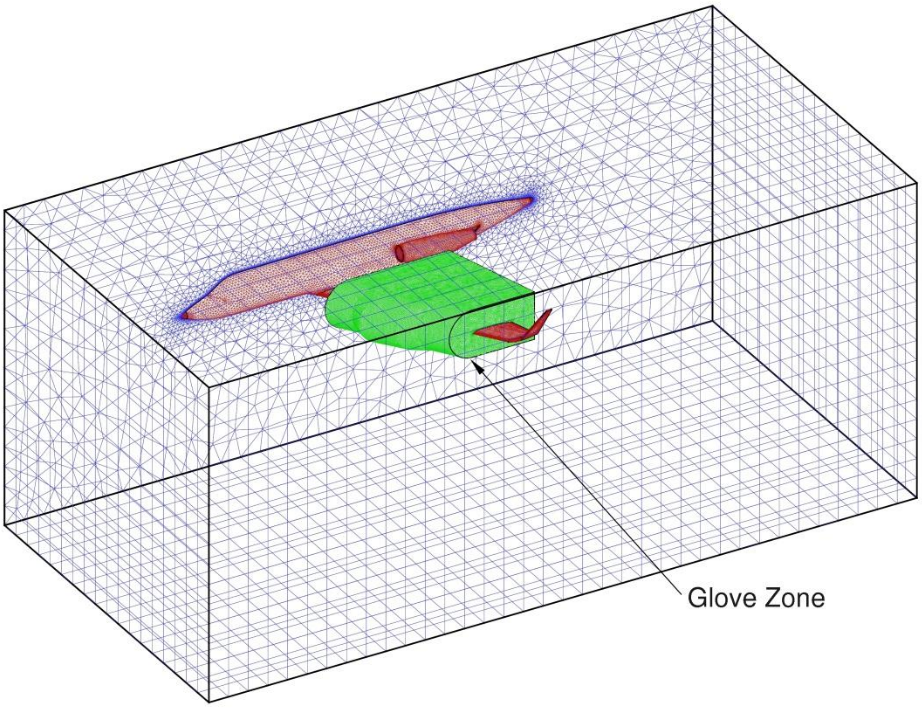


Fig. 19 Aircraft zone mesh.

The extent of the aircraft zone is not nearly as critical as the other two zones, but space should be given to allow the cells to grow away from the aircraft and smoothly transition to the freestream zone. The boundary was created in SolidWorks with a specified length, width, and height of 45.72 m (150 ft), 22.86 m (75 ft), and 20.32 m (67.7 ft) respectively. The aircraft is centered between the forward, aft, upper, and lower boundary faces.

While much of the aircraft zone is composed of tetrahedral cells, pyramidal and prismatic cells were implemented in select locations. The zone boundary is shared with the freestream zone and therefore has rectangular surface cells. The use of pyramidal cells to transition from hexahedral cells to tetrahedral cells eliminates the need for zonal interfaces. Prismatic cells, which are in essence wall-normal extruded surface cells, assist in resolving the flowfield near the aircraft surface where viscous effects are significant. This is very difficult with tetrahedron-only unstructured meshes because it would require an extremely fine surface mesh to place nodes within the boundary layer. Generating prismatic cells is not without its own challenges, however. It often required substantial trial and error to obtain an error-free and high-quality mesh. Prismatic cells were generated for the all aircraft faces in the aircraft zone except the set of engine faces. Prism layers near the engine inlet and outlet caused problems with CFD solution stability and were therefore omitted.

To exact more control over the meshing process in the aircraft zone, the aircraft surface was subdivided into sets of faces that would share the same meshing parameters. These parameters include maximum and minimum cell size, maximum deviation, and the option to generate prismatic cells. The maximum deviation parameter is used to control mesh refinement. Illustrated in Fig. 20, the maximum deviation limits the distance a planar surface cell can deviate from the surface it will represent. In regions of high curvature, a smaller maximum deviation distance leads to smaller surface cell size. This automatically refines the surface mesh in places like the wing leading edge and the nose and tail of the fuselage where surface curvature and flowfield gradients are larger.

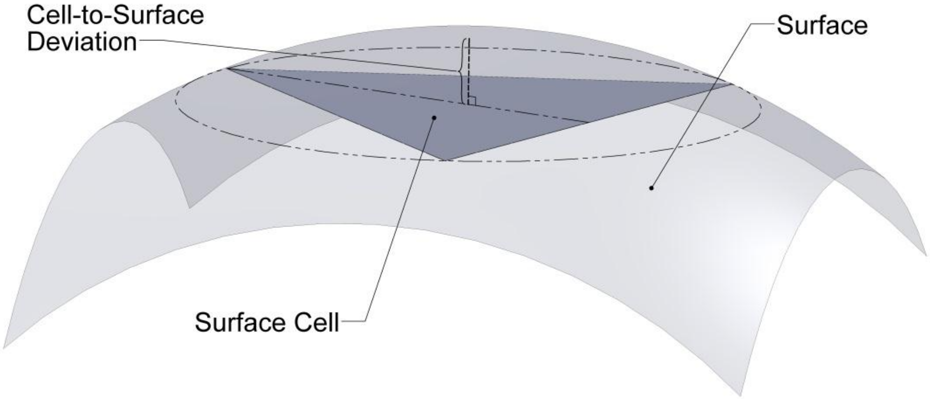


Fig. 20   Maximum deviation parameter for unstructured surface cells.

ICEM CFD also has the ability to create patch independent meshes which allows surface cells to overlap multiple faces, a useful feature if the geometry has surface gaps or contains small or oddly shaped faces. For example, there are a number of small or high-aspect-ratio faces near the cockpit of the SCRAT model that can be effectively combined using a patch-independent technique. Recommendations provided by the DPW again offered guidance for the mesh setup. Following an extensive parameter adjustment period, the aircraft zone mesh design converged to a satisfactory result. As a final step in the zone meshing process, built-in mesh smoothing algorithms were used to improve the mesh quality.

As mentioned earlier, an interior boundary condition was placed on the aircraft zone boundary. Similar to the freestream zone, the face coplanar with the aircraft symmetry plane was given a symmetry condition. All aircraft surfaces, with the exception of the engine inlet and outlet, were set as walls. Because mass is leaving the computational domain at the engine inlet, a pressure outlet with a specified mass flow rate was used. Vice versa, mass reenters the domain at the engine inlet, leading to the use of a pressure inlet boundary condition. Finally, the boundary between the aircraft and glove zone is designated as an interface. An interface is necessary because the use of pyramidal cells to connect the unstructured aircraft zone to the structured glove zone on a node-by-node basis would result in a very poor quality mesh.

Glove Zone

The glove zone is third and final major level of the nested grid structure. It contains a large portion of the wing, including SARGE. Structured hexahedral cells were the optimal choice for this zone for several reasons. First and foremost, unstructured cells would most likely not provide the level of refinement necessary in the boundary layer region for the boundary-layer stability calculations. Second, structured cells offered considerably more control over the mesh design than unstructured surface and prismatic cells. Finally, unstructured cells were not conducive to the flowfield data extraction techniques employed to complete boundary-layer stability calculations. The blocking required would be much more complex than that of the freestream zone, which initiated the creation of the supplemental geometry described in Section III-C for this zone to assist with mesh mapping. The blocking was also very user intensive and required a great deal of manual adjustment. The glove zone mesh is shown in Fig. 21.

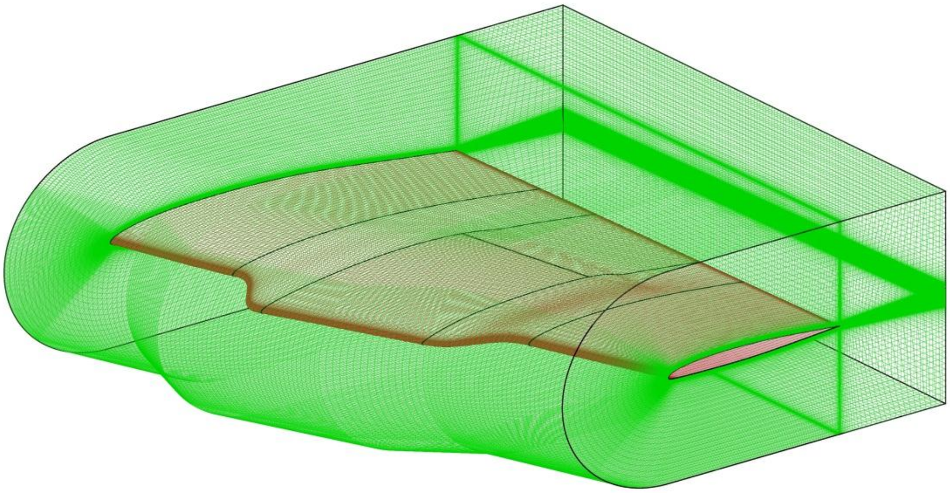


Fig. 21   Glove zone mesh.

The shape and size of the glove zone was tailored specifically to the meshing strategies that would be employed. A structured c-style mesh is an effective and often used technique for airfoil grid generation. Applying this approach across the wing span led to the creation of a 3-D c-style mesh. Establishing the location and extent of the zone boundary was not as clear-cut as the zone shape. Because an interface was required to pass flowfield information from the aircraft zone to the glove zone, there was a concern that placing the interfaces too close to the glove test section could skew the results there. The sizing that was chosen placed the boundary roughly 1.524 m (60 in) away from all wing and SARGE surfaces in the normal direction and 2.489 m (98 in) inboard and outboard of the test section in the spanwise direction. Whether or not this boundary placement produces any undesirable effects is discussed further in Section IV-B.

Early CFD simulations indicated that using a completely laminar model would cause solution convergence issues, with the likely culprit being flow separation. This is not entirely surprising because similar difficulties were encountered in past projects. Using a turbulence model corrects this problem; however, a turbulent boundary layer cannot be used for boundary-layer stability calculations. The solution involves creating a separate region near the test section where the flow will remain laminar while the rest of the domain uses a turbulence model. The specifics of this computational strategy are given in Section V-A. The glove zone was split into turbulent and laminar subzones in preparation for this technique.

The laminar subzone mesh, shown below in Fig. 22, will contain the flowfield information used in all boundary-layer stability calculations and therefore must be created in a careful and specific manner. The extent of the laminar subzone must first be established. The boundary layer above the test section is the primary concern so the spanwise boundaries of the laminar subzone were set to coincide with those of the test section. Viscous effects beyond the boundary layer are negligible whether the flow is modeled laminar or turbulent. The offset surface set discussed in Section III-C, which is 50.8 mm (2 in) from the wing and glove surfaces and well outside the test section boundary layer region, serves as the boundary parallel to glove surface. The location of the aft boundary of the laminar subzone was the most difficult to determine. Flow separation with a laminar model was occurring somewhere near the upper aft blending section. It was discovered that this could be avoided by ending the laminar subzone slightly before the suction-side pressure minimum and switching back to a turbulent model aft of this location. Through an iterative process, the aft boundary of the laminar subzone was placed at 60% *x/c*.

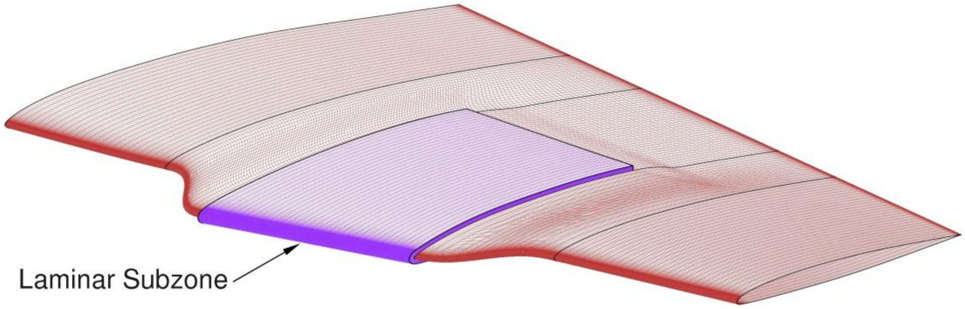


Fig. 22 Laminar subzone mesh.

The mesh within the laminar subzone must be generated in a particular manner to not only capture the flowfield at a proper resolution, but to satisfy the needs of boundary-layer data extraction performed after the CFD simulation. Surface mesh construction plays a large role in both these goals. Boundary-layer stability is sensitive to changes in the flowfield near the leading edge, making it important to place a large number of cells in this region to accurately resolve large gradients. Spanwise grid lines are tightly packed near the leading edge initially, but are allowed to expand downstream. They are also specifically arranged to lie at constant *x/c* locations across the glove span to facilitate the flowfield data extraction process. The streamwise grid lines are placed at constant span locations for similar extraction reasons. Moving away from the glove surface, it is critical to obtain a fine wall-normal mesh resolution because boundary-stability calculations require a well-resolved boundary-layer profile. The baseline mesh configuration in the laminar subzone used an initial cell layer height of 2.54 μm (0.0001 in) and a wall-normal growth rate of 1.06. This corresponds to 60 wall-normal points in the first 1.27 mm (0.05 in) and 122 total points in the laminar subzone. Fig. 23 shows a cross-section of the glove zone and highlights the glove leading edge, illustrating the level of refinement. The grid independence studies discussed in Section IV-B explores whether or not the baseline mesh configuration is properly refined.

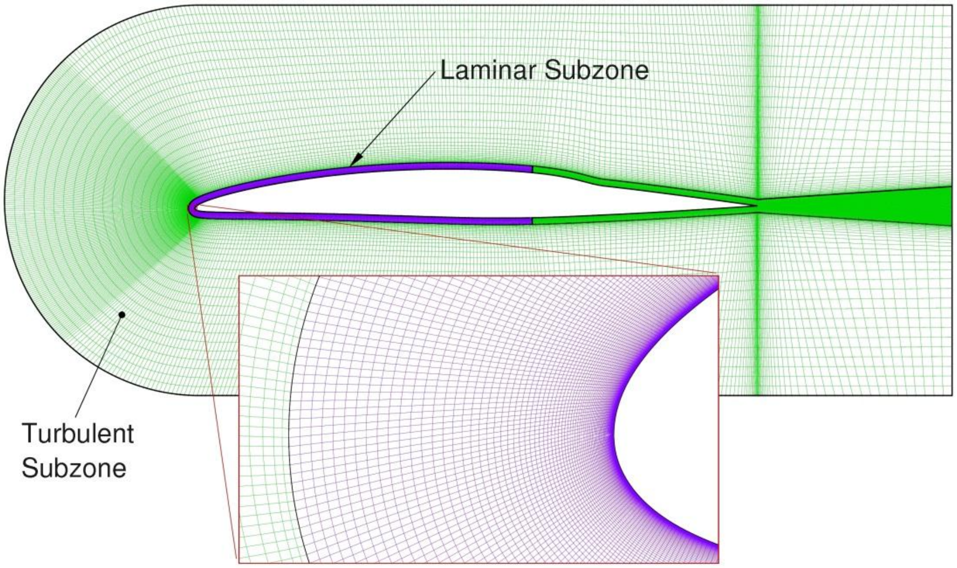


Fig. 23 Cross-section of glove zone.

The glove zone boundary was designated as the second half of an interface set between the aircraft and glove zones as previously stated. All SCRAT and SARGE surfaces in this zone are walls and all other supplemental geometry were given interior boundary conditions since they are used for mesh mapping only.

B. Grid Independence Studies

Assessing the credibility of CFD modeling and simulation is a vital step for all computational efforts. The two primary methods used to accomplish this are verification and validation. The AIAA Guide for Verification and Validation (V&V) [30] describes verification as providing evidence that the model is solved correctly, even though it may not represent the "real world" solution. Validation, on the other hand, evaluates whether the CFD model, meaning the mathematical and computer code representation of the physical system, is solved accurately with regard to experimental data and observations. Because the SARGE project is still in the design stage and no experimental data exists yet, validation will not be addressed in this thesis. Verification of the CFD model, however, can be performed.

The V&V Guide states that there are four predominant sources of error in CFD simulations: insufficient spatial discretization convergence, insufficient temporal discretization convergence, lack of iterative convergence, and computer programming. All simulations presented are steady state, making temporal discretization convergence unnecessary. Iterative convergence, which is discussed further in Section V-A, is easily monitored and controlled. Computer programming is also a nonissue because the commercial CFD solver FLUENT used for this project has been extensively verified over the years. This leaves spatial discretization convergence as the primary concern and will of course be highly dependent on the computational grid.

To determine how varying grid parameters affect the solution, CFD simulations were performed using different grids. The results obtained from these simulations could then be compared to results from a CFD simulation using the baseline grid described in Section IV-A. The flowfield solution in the laminar subzone serves as the area-of-interest for these grid independence studies. Test section C_p and boundary-layer stability results were used as the principal criteria to judge independence.

Three comparison grids were generated for this analysis: a farfield grid, an interface grid, and a refined grid. Except for the supplemental geometry or parameter changes under investigation, they are all nearly identical to the baseline grid. The farfield grid explores how modifying the extent of the computational domain affects the region of interest. For this study, the farfield boundary was reduced by roughly 45% in all three directions. The interface grid examines how moving the location of the aircraft/glove zone interface by altering the size of the glove zone changes the solution near the glove test section. The normal and spanwise test section offsets for the boundary were reduced by 38%. Finally, the refined grid involved refining the aircraft and glove zones. Most of the refinement effort was focused on the laminar subzone to ensure the boundary-layer stability behavior did not change. Table 3 gives the specific changes for these three grids (highlighted in gray), along with the baseline grid for comparison.

Table 3 Grid geometry and parameters.

		Baseline Grid	**Farfield Grid**	**Interface Grid**	**Refined Grid**
Domain Size	Spanwise	457.2 m [1,500 ft]	254.0 m [833.3 ft]	457.2 m [1,500 ft]	457.2 m [1,500 ft]
	Vertical	914.4 m [3,000 ft]	508 m [1,667 ft]	914.4 m [3,000 ft]	914.4 m [3,000 ft]
	Streamwise	1,372 m [4,500 ft]	825.5 m [2,708 ft]	1,372 m [4,500 ft]	1,372 m [4,500 ft]
Interface Offset	Normal	1.524 m [60 in]	1.524 m [60 in]	0.9525 m [37.5 in]	1.524 m [60 in]
	Spanwise	2.489 m [98 in]	2.489 m [98 in]	1.524 m [60 in]	2.489 m [98 in]
Laminar Subzone Refinement	First Layer Height	2.54 µm [1×10^{-4} in]	2.54 µm [1×10^{-4} in]	2.54 µm [1×10^{-4} in]	1.27 µm [5×10^{-5} in]
	Growth Rate	1.06	1.06	1.06	1.05
	Points on Airfoil	391	391	391	471
Cell Count	Freestream Zone	0.682×10^{6}	0.510×10^{6}	0.682×10^{6}	0.682×10^{6}
	Aircraft Zone	5.37×10^{6}	5.37×10^{6}	7.71×10^{6}	8.00×10^{6}
	Glove Zone	11.56×10^{6}	11.56×10^{6}	8.42×10^{6}	21.5×10^{6}
	Total	17.6×10^{6}	17.4×10^{6}	16.8×10^{6}	30.2×10^{6}

The solver options and solution strategies used for all grid independence study simulations are outlined in Section V-A. Boundary conditions from Test Point A, which can be found in Appendix B, were applied to the farfield, engine inlet, and engine outlet for all simulations. The section C_p plots chosen for these studies are taken from BL204, BL234, and BL264. Recall, BL198 and BL270 mark the inboard and outboard edges of the glove test section respectively. LST results for stationary crossflow were generated for the suction side of the airfoil section at BL234. Details on how these stability calculations were performed and how they should be interpreted are given in Section VI. Due to the sensitivity of the boundary-layer stability to small changes in the flowfield, the stability results will serve as the true litmus test whether grid independence has been achieved. For the purpose of showing grid independence, the focus should remain how the solution varies from grid to grid and not the behavior of the solution itself. Fig. 24 displays the C_p results and Fig. 25 gives the boundary-layer stability results in the form of an N-factor plot.

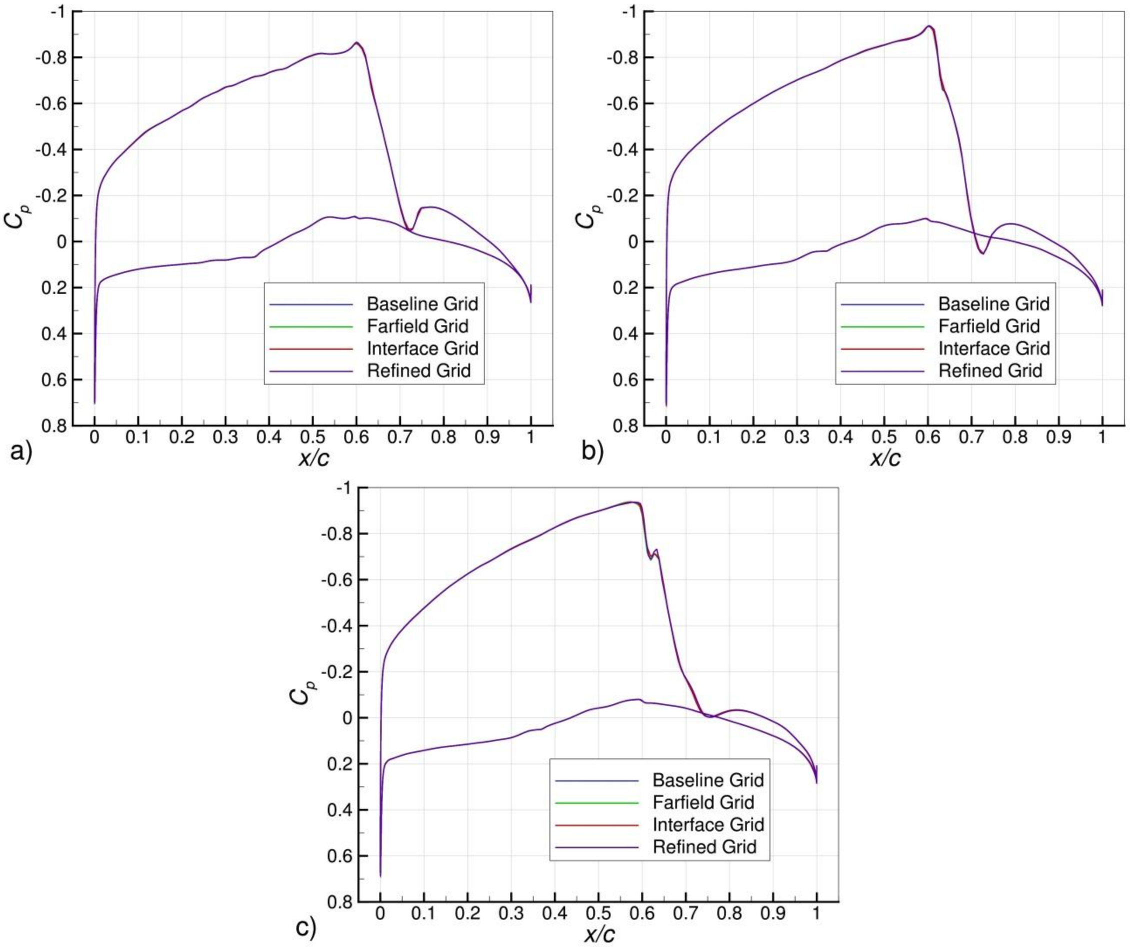


Fig. 24 Comparison of grid-independence study C_p results. a) BL204, b) BL234, and c) BL264.

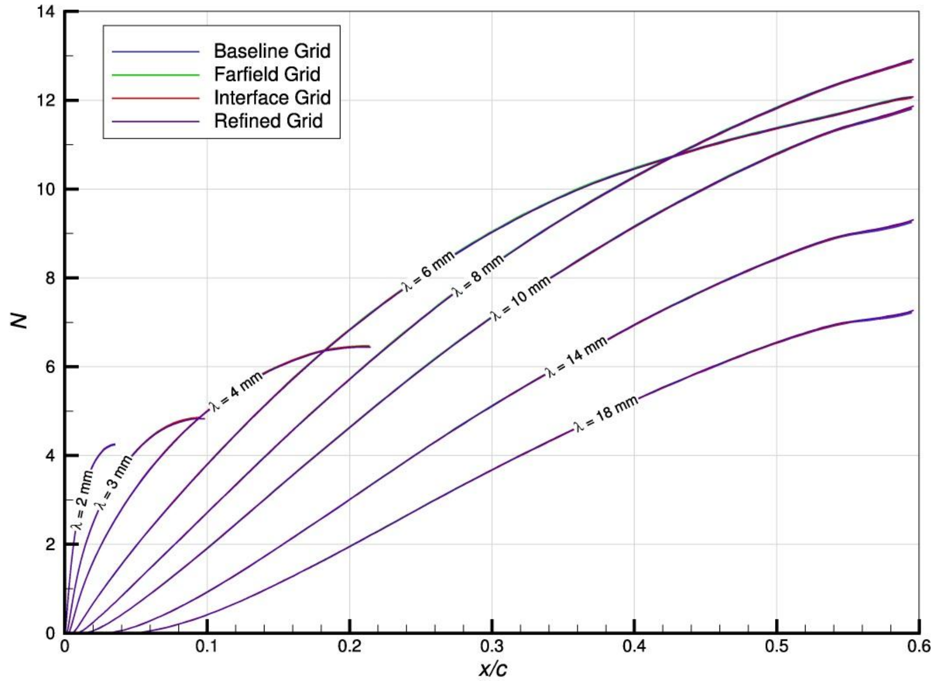


Fig. 25   Comparison of grid-independence study crossflow N-factor results at BL234.

The agreement between the results from all grids was excellent. There were essentially no differences in the test section C_p plots back to 60% *x/c* or the stability characteristics at BL234. This verified that the baseline grid sufficiently captures the flowfield effects desired for this project. It should be noted that a conservative approach was taken with the baseline grid as evidenced by the lack of variation between solutions using other grids. This was done primarily because of the uncertainty that similar behavior would occur with different boundary conditions. The test points examined for the SARGE design vary in angle of attack (AoA), Mach number, and Reynolds number – sometimes significantly. It is infeasible to perform grid independence studies for each test case in light of both the computational expense and level of analysis required. With the assurance that the baseline grid will provide grid independent solutions, CFD simulations using the various test point boundary conditions could now be performed.

V. FULL-AIRCRAFT CFD CALCULATIONS

All conclusions drawn about the effectiveness of the SARGE design originate from the full-aircraft simulations. They generate all the flowfield information used for standard aerodynamic analysis as well as detailed boundary-layer properties near the glove test section used for stability calculations. The solver options and solution strategies, SARGE pressure distribution analysis, and stability analysis simulation flowfield results are discussed below.

A. Solver Options and Solution Strategies

The commercial CFD package FLUENT 13.0 [31] was used for all project simulations. FLUENT can model a variety of physical phenomena including fluid flow, heat transfer, and chemical reactions for complex geometries using a finite volume method. Its parallel processing capability is an attractive feature for simulations employing grids containing millions cells.

FLUENT was used on two computer systems: a local workstation and a remote supercomputer cluster. The local workstation, as mentioned previously, is an Apple Mac Pro. Its technical specifications are shown in Table 4. No solution calculations were performed on this system due to the relatively small number of processors and shortage of available memory. Instead, the workstation served in a pre- and post-processing capacity for FLUENT case and data files. Solution calculations were completed through batch processing exclusively on the remote supercomputer cluster Eos which is operated by the Texas A&M Supercomputing Facility. Its specifications are also given in Table 4. Simulations performed on Eos made use of 32 processors acting in parallel over 8 nodes with a maximum of 176 GB of available memory. Some pre-processing activities, such as case setup, were also carried out with Eos.

Table 4 Technical specifications of computing resources.

	Apple Pro Mac Workstation	Eos Cluster
Operating System	Linux - Ubuntu 10.04 LTS	Linux - Redhat Enterprise
Number of Nodes	-	372
Number of Processing Cores	24	3,168
Total Memory	64 GB	9,056 GB
Total Disk Size	1 TB	120 TB

Once the computational grid has been loaded into FLUENT, there are a series of steps that must be completed prior to performing the simulation. The first step occurred outside the software realm and involved fully defining the type of problem being solved. The SARGE project involves a transonic aircraft flying at constant conditions with analysis focused on determining flowfield and boundary-layer stability characteristics. This simple statement provides the foundation from which all FLUENT settings will rest upon; the flow is external, steady, compressible, and viscous. From here, the steps that establish model and solution settings could begin. The FLUENT User's and Theory Guides provided direction on the various settings available through the solver.

FLUENT gives the user the option of using either a pressure-based or density-based solver. According to the Theory Guide, the pressure-based solver is suited for incompressible and mildly-compressible flows. It solves the continuity, momentum, and energy equations separately or through a coupling mechanism. Conversely, the density-based solver is suited for high-speed compressible flows and solves the continuity, momentum, and energy equations simultaneously. The density-based solver is the clear choice to use for these simulations based on this information. Because the flow is compressible and includes the energy equation, gas property modeling is required. The ideal gas law and Sutherland's law were used for air density and viscosity modeling respectively.

One of the major simulation setup decisions involved choosing the viscous flow model that would be employed throughout the domain. As earlier sections alluded to, a laminar model caused solution convergence difficulties, most likely due to flow separation, making the use of a turbulence model necessary. However, a laminar model would still be required near the glove surface in order to perform meaningful boundary-layer stability calculations. Initially, this was a formidable problem because it was not known whether a solution existed within FLUENT. Fortunately, a simple method existed that would accomplish this goal. The User’s Guide states that while a simulation may employ a turbulence model, it is possible to “turn off” turbulence modeling in a particular cell zone, such as the laminar subzone described in Section IV-A, effectively making it laminar. FLUENT achieves this by disabling turbulence production and turbulent viscosity while still transporting the turbulence quantities. This strategy was the impetus to creating the laminar subzone. Now that this hurdle was cleared, a specific turbulence model was chosen. Out of the available options presented by FLUENT, the two-equation *k*-*ω* shear-stress-transport (SST) turbulence model developed by Mentor [32] was selected because, as stated by the Theory Guide, it blends the robust and accurate formulation of the *k*-*ω* model near the wall with the freestream independence of the *k*-*ε* model in the farfield. The default model constants for FLUENT were used.

At this point in the setup process, boundary conditions were applied to the farfield, engine inlet, and engine outlet. The farfield used a pressure farfield type boundary condition, which requires a static pressure, Mach number, flow direction, and static temperature. These properties were based on the simulation test point parameters. Test points define the flight conditions that generate the near-aircraft flowfield under investigation. For the SARGE project, test points consist of a freestream Mach number, aircraft AoA, and chord Reynolds number at a given span station, *e.g.* BL270. The Mach and Reynolds numbers allow for the calculation of the freestream pressure and temperature, as well as the altitude, using the 1976 U.S. Standard Atmosphere model. Calculating the altitude ensures the test point is within the aircraft flight envelope. Further information about the test points investigated can be found in Section V-C. Farfield boundary conditions used for all test points are given in Appendix B.

The engine inlet and engine outlet were modeled with a pressure outlet and pressure inlet respectively. NASA DFRC provided the boundary conditions for the engine modeling, using a 1-D code to obtain pressure, temperatures, and mass flow rates. FLUENT allows the user to specify a target mass flow rate for a pressure outlet, which was employed for all CFD simulations. CFD results from the G-II showed that there was flowfield sensitivity on the wing due to engine effects, primarily the from engine inlet since it is positioned above the upper surface of the wing. By specifying a target mass flow rate at the engine inlet, it is hoped obtain more realistic engine effects. The engine outlet mass flow rate was monitored but no target was designated because the flowfield near the glove was fairly insensitive to the flowfield created by the engine outlet. Engine inlet and outlet boundary conditions for all test points are also given in Appendix B.

In order for the unstructured aircraft zone to “talk” with the structured glove zone, an interface group must be designated in FLUENT. Doing so simply involves specifying two interface zones, the first being the outer unstructured glove zone boundary and the second being the inner structured glove zone boundary. Once this is complete, the next step is to specify solution strategy settings.

Solution strategies define how the CFD simulation is solved and include solution methods, controls, monitors, initialization, and calculation activities. Beginning with solution methods, the density-based solver gives the user the option of using either an implicit or explicit formulation. The implicit formulation was chosen for this project because it has no solution stability limits, though it comes at a higher computational cost than the explicit formulation. The Roe-FDS flux type was chosen because the User’s Guide recommends it for most cases while the second option, the AUSM flux type, is recommended when the objective is to provide exact resolution of contact and shock discontinuities. The Green-Gauss node based scheme was selected for the spatial discretization gradient. Compared to the Green-Gauss cell based and least-squares cell based schemes, the Green-Gauss node based scheme is more accurate, particularly on irregular unstructured grids. It also comes at a higher computational cost however. Simulations were completed with second-order upwind schemes for flow, turbulent kinetic energy, and specific dissipation rate spatial discretization. The User’s Guide states that a second-order scheme decreases the numerical discretization error and subsequently provides a more accurate solution, especially on grids that are not aligned with the flow and for complex flows. The V&V Guide suggests that all simulations employ second-order accurate schemes to achieve proper solution accuracy. Higher order schemes were available, but the computational cost and solution divergence concerns associated with them discouraged their use.

Solution controls allow the user to specify a Courant number and various under-relaxation factors. The Courant number limits how much the solution changes with each iteration and can be increased to larger values to speed up the rate of convergence for steady-state simulations. However, care must be taken because rapid solution changes can cause divergence due to nonlinearities in the governing equations. The procedure for changing the Courant number as the simulation progresses is discussed below. Under-relaxation factors control how the calculated variables are updated for each iteration and can be used to prevent divergence. The default under-relaxation factors provided by FLUENT were used without issue for all simulations.

Solution monitors allow the user to observe the progress of the simulation as it iterates toward a converged solution. The V&V Guide [30] listed iterative convergence as a possible source in CFD simulation error, and as such should be examined for all simulations. Iterative convergence is achieved when the simulation solution experiences changes less than a user specified tolerance. FLUENT registers solution changes through scaled residuals of the governing equations. For every SARGE simulation, all scaled residuals decreased by more than four orders of magnitude. Another tool to judge iterative convergence involves monitoring specific flowfield results. The aircraft lift and drag coefficients and the engine inlet and outlet mass flow rates were calculated and displayed along with the residuals to assist with determining solution convergence. Typically, the residuals and engine mass flow rates would decrease and level off before the force coefficients had converged. This required additional iterations until both the lift and drag coefficients settled to constant values as well. Once this occurred, the solution was said to have converged and iteration would stop.

A standard solution initialization was used for the entire computational domain. The farfield boundary conditions were used as initial values. Upon completion of this step, the solution could be calculated. Because FLUENT has a text user interface (TUI) in addition to a graphical user interface, the entire setup described up to this point can be automated through the use of a journal file. A sample case setup journal file is given in Appendix C-1.

When a set of case and data files was ready to begin calculation activities, the simulation was submitted to Eos for batch processing. A second journal file was created that allowed for solution calculation activities. Samples of the simulation batch and journal files can also be viewed in Appendix C-2 and -3. For the SARGE project simulations, the activities included changes in the Courant number, discretization scheme, and specifying a target mass flow rate for the engine inlet. At the beginning of the simulation, a first-order discretization scheme was used and the Courant number was incrementally increased from 1 to 16. These measures avoided solution divergence, which will most likely occur at the beginning of a simulation. After 1,000 iterations, the discretization scheme was changed to second-order and the Courant number was reset back to 1. At the 1,100 iteration mark, after the second-order solution had settled slightly, the engine inlet mass flow rate was specified. From here, the Courant was again incrementally increased up to 64 where it remained until the solution had converged. For the baseline grid, the entire simulation was typically completed in 7,000 iterations, taking roughly 1,120 CPU hours or 35 wall-clock hours using 32 processors. This process was repeated for the ten test points performed for the glove design analysis.

B. SARGE Pressure Distribution Analysis

The pressure distribution on the glove surface was the first flowfield feature examined as initial full-aircraft simulations for the SARGE design were completed. Boundary-layer stability behavior is greatly influenced by the pressure distribution and any major spanwise variations across the glove would most likely result in very different stability characteristics. This was experienced firsthand with the previous glove OML design, designated TAMU-05-04, which is discussed in detail by Belisle *et al.* [26] and Malik *et al.* [33]. As mentioned briefly in Section II-B, the test section OML for the TAMU-05-04 design was created by lofting two airfoil profiles together. Preliminary calculations on these airfoils using infinite-swept-wing assumptions showed that the pressure distribution and stability characteristics were satisfactory for a DRE experiment. Unfortunately, this behavior did not translate to the 3-D glove OML, which is not entirely surprising considering SARGE employs a finite-span conical design and is located in a region of appreciable 3-D flowfield effects created by the aircraft and glove fairings. Suction-side C_p contour and section C_p results from a TAMU-05-04 simulation are shown in Fig. 26. The simulation test point conditions are AoA = 3.5°, $M = 0.75$, and $Re_c = 22\times10^6$ at BL234.

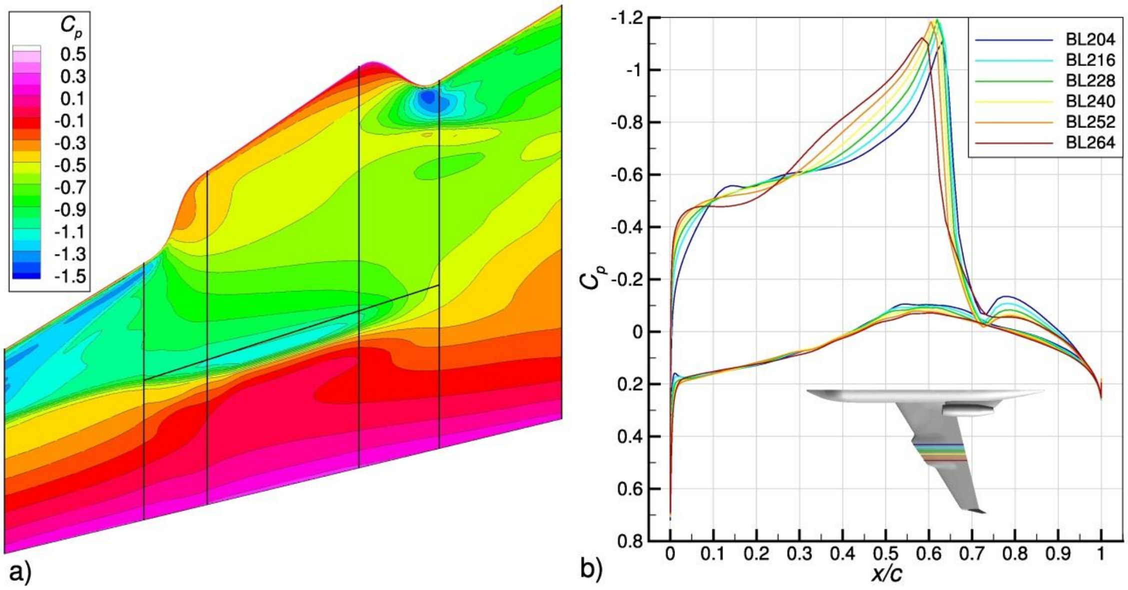


Fig. 26 C_p results from the TAMU-05-04 glove design. a) suction-side contours and b) section plots from BL204 to BL264 in 0.3048 m (12 in) increments.

The first sign of a non-uniform spanwise pressure distribution is evidenced by the C_p contour lines. They do not follow the glove's constant-*x/c* lines very well, especially near the leading edge of the glove's inboard boundary and beyond 30% *x/c*. The pressure distribution variations are even more pronounced when comparing section C_p results at several span stations. Small C_p differences could be tolerated as long as the general shape of the C_p curves was maintained. This is certainly not the case, primarily on the suction side. Local pressure-minima also appear near the leading edge on both sides of the glove, which will lead to an adverse pressure gradient much earlier than desired and could possibly destabilize the T-S instability beyond acceptable levels. These results indicated that a redesign of the glove OML was necessary and that a different approach to define the glove OML should be taken.

The glove design presented in this thesis, TAMU-06-05, employed the CFD code TRANAIR to perform a 3-D optimization of the glove OML and was carried out through a collaborative effort between DFRC and FRL. Details on the process are given by Hartshorn *et al.* [34]. The glove optimization weighted the region between BL212 and BL264 the most heavily to ensure the center of the glove would produce the greatest uniformity. The TAMU-05-04 glove geometry was used as a starting point for the optimization, and actually only needed to be altered slightly (on the order of tenths of inches) to obtain the

TAMU-06-05 OML. Again, suction-side C_p contour and section C_p results were analyzed, as given by Fig. 27. This solution is from the Test Point A simulation, which is defined as AoA = 3.4°, $M = 0.75$, and $Re_c = 22\times10^6$ at BL270. These test point parameters were also used during the optimization process and define the “sweet spot” for the glove design. Further comments on Test Point A will be made in Section V-C.

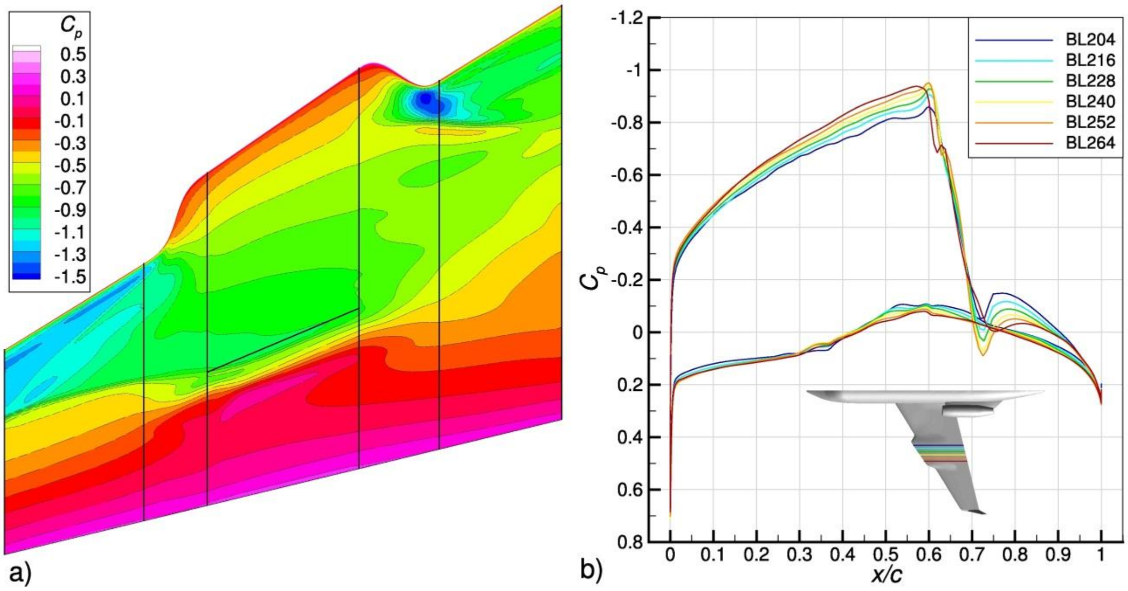


Fig. 27 C_p results from the TAMU-06-05 glove design. a) suction-side contours and b) section plots from BL204 to BL264 in in 0.3048 m (12 in) increments.

The spanwise uniformity of the pressure distribution on the test section was greatly improved with the optimized OML. The C_p contour lines follow the glove’s constant-*x/c* lines fairly well, especially near the leading edge. Further examination of section C_p results confirm that the overall behavior, such as the C_p chordwise gradient (given by the slope of the curve) and the location of the suction-side pressure minimum, is fairly consistent at all span stations on both sides of the glove. This is especially true near the leading edge, which plays a large role in the overall stability characteristics of the glove. The level of spanwise uniformity in the pressure distribution present in this design is certainly a step in the right direction towards generating uniform stability behavior across the glove.

C. Stability-Analysis Simulation Flowfield Results

The stability-analysis simulations are the central element in the investigation of the flowfield and boundary-layer stability attributes of the wing glove design. They consist of ten test point simulations that explore various combinations of aircraft AoA, Mach number, and chord Reynolds number using the full-aircraft configuration. The flowfield results discussed in this section will provide a first-order look into the performance of the wing glove design. In-depth boundary-layer stability results will follow in Section VI.

The stability analysis simulations began with the principal test point, Test Point A. As mentioned in Section V-B, its parameters were used for the OML optimization. The flight conditions of AoA = 3.4°, $M = 0.75$, and $Re_c = 22\times10^6$ at BL270 were chosen because they focused on meeting the requirements laid out in Section I-E. The angle of attack and Mach number combination were projected to satisfy the C_L and Mach number targets. The Reynolds number was chosen to match the minimum requirement for the DRE test flights, the primary objective of the experiment. All other test points were designed around Test Point A in an effort to better understand flowfield and boundary-layer stability sensitivity to condition changes.

The stability analysis simulations can be broken up into three test point groups: angle of attack simulations, Mach number simulations, and Reynolds number simulations. All groups include Test Point A. Simulations within a group share two of the three parameters from Test Point A, while the parameter under investigation is variable. For example, the angle of attack simulations use the same Mach and Reynolds number parameters as Test Point A, but the aircraft angle of attack varies. Table 5 lists the simulation test points and their parameters, arranging them by group.

Table 5 Simulation test point parameters.

<table>
<tr><td colspan="60">**Angle of Attack Simulations** (M = 0.75, $Re_c = 22\times10^6$ at BL270)</td></tr>
<tr><td colspan="0">**Test Point**
Angle of Attack</td><td colspan="12">B
3.2°</td><td colspan="12">A
3.4°</td><td colspan="12">C
3.6°</td><td colspan="12">D
3.8°</td><td colspan="12">E
4.0°</td></tr>
<tr><td colspan="60">**Mach Number Simulations** (AoA = 3.4°, $Re_c = 22\times10^6$ at BL270)</td></tr>
<tr><td>**Test Point**
Mach Number</td><td colspan="15">F
0.66</td><td colspan="15">G
0.72</td><td colspan="15">A
0.75</td><td colspan="15">H
0.76</td></tr>
<tr><td colspan="60">**Reynolds Number Simulations** (AoA = 3.4°, M = 0.75)</td></tr>
<tr><td>**Test Point**
Reynolds Number</td><td colspan="20">I
15×10^6 at BL270</td><td colspan="20">A
22×10^6 at BL270</td><td colspan="20">J
30×10^6 at BL198</td></tr>
</table>

Angle of Attack Simulations

The primary purpose of the angle of attack simulations was to examine flowfield sensitivity to different aircraft AoAs. The AoA was varied from 3.2° to 4.0° in 0.2° increments with the goal of finding the limiting AoA values based on C_L and C_p behavior. A similar investigation will be performed during the initial science test flights using pressure-port measurements. The lower bound of the AoA range comes from the glove C_L results, which is of course highly dependent on AoA. Table 6 gives a comparison of C_L values for the glove test section.

Table 6 Glove test section lift coefficients for the angle of attack simulations.

Angle of Attack Simulations (M = 0.75, $Re_c = 22\times10^6$ at BL270)					
Test Point	B	A	C	D	E
Angle of Attack	3.2°	3.4°	3.6°	3.8°	4.0°
Lift Coefficient	0.478	0.492	0.511	0.531	0.550

The AoA = 3.2° (Test Point B) and AoA = 3.4° (Test Point A) simulations do not meet the $C_L = 0.5$ requirement, but the remaining three do. The C_L of the target C_p that was used for the OML optimization was initially large enough to satisfy the requirement, but the small amounts of spanwise variation that occurred on the optimized glove pressure distribution caused the C_L for the principle test point simulation to be slightly lower than desired. An AoA of 3.48° would theoretically meet the $C_L = 0.5$ goal. This estimation was made using linear interpolation, which would be valid since the glove C_L values above change linearly with AoA. Further examination will show that this would only produce slight changes in the pressure distribution from the Test Point A and C results.

Section C_p plots were generated for the BL212, BL234, and BL264 span stations to determine the upper bound of the AoA range. The BL212 and BL264 sections were selected because they served as the boundaries of the heavily-weighted portion of the glove used in the optimization. The BL234 section was selected because it lies at the spanwise center of the glove. Fig. 28 gives a comparison of the section C_p results for all five angle of attack simulations at the three span stations.

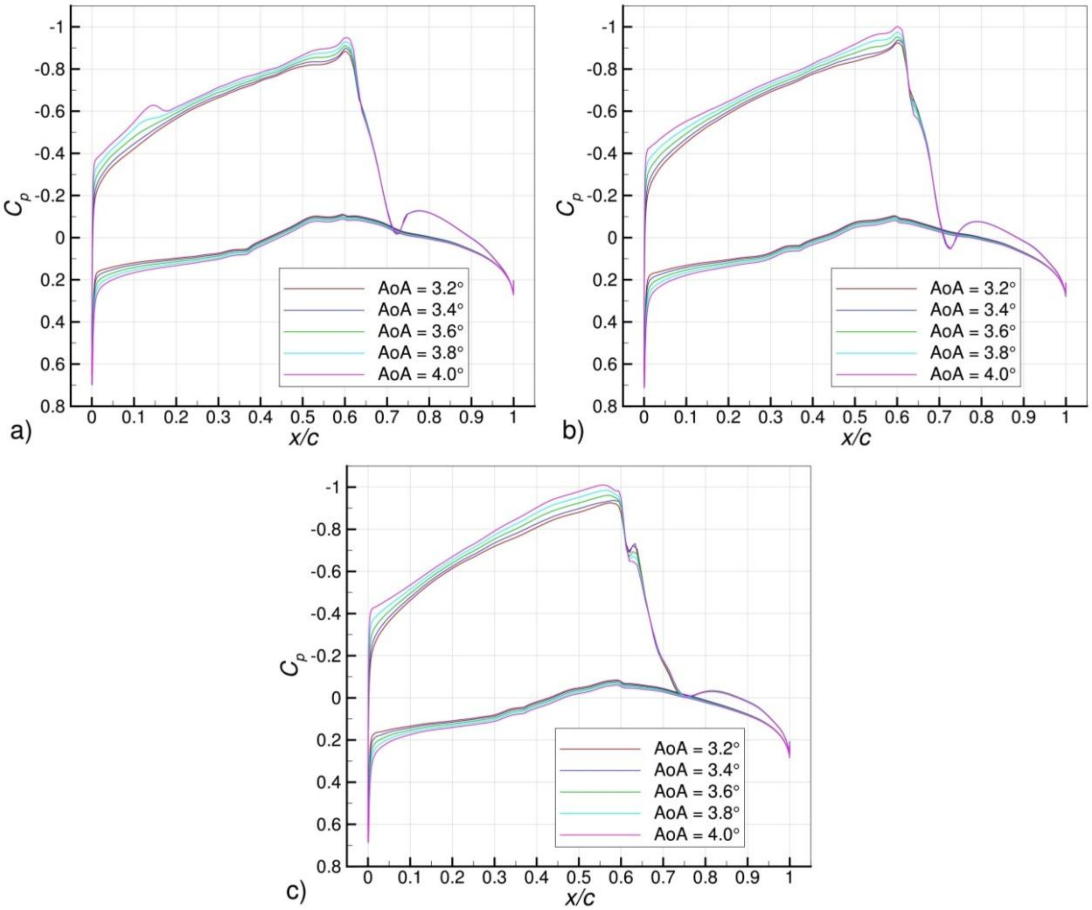


Fig. 28 Glove C_p comparison for angle of attack simulations. a) BL212, b) BL234, and c) BL264.

Several key conclusions can be drawn from the plots. First and most obviously, significant differences in the C_p behavior exist on the suction side of the BL212 section at higher AoAs. This will be addressed shortly. Excluding those results, all other C_p curves remain fairly consistent in shape. Second, the suction-side chordwise pressure gradient becomes less accelerated and the pressure-side chordwise pressure gradient becomes more accelerated as AoA increases. Recall that T-S is stabilized and crossflow is destabilized by an accelerating pressure gradient. Therefore, crossflow should become less unstable on the suction side and more unstable on the pressure side as AoA increases, while T-S should become more unstable on the suction side and less unstable on the pressure side as AoA increases. Third, the pressure minimum location for all three sections is relatively invariant with changing AoA, allowing for laminar flow back to roughly 60% *x/c*. It should also be noted that the difference between the AoA = 3.4° and AoA = 3.6° (Test Point C) C_p results are small and would serve as bounding cases for an AoA = 3.48° simulation, whose results would of course differ even less from the other two cases.

The loss of spanwise uniformity at BL212 on the suction side required further investigation. The results for AoA = 4.0° (Test Point E) was especially concerning because it contained a local pressure-minimum that would likely destabilize T-S and could cause early transition. Suction-side C_p contour results for the five AoAs, shown in Fig. 29, explain the cause of this behavior.

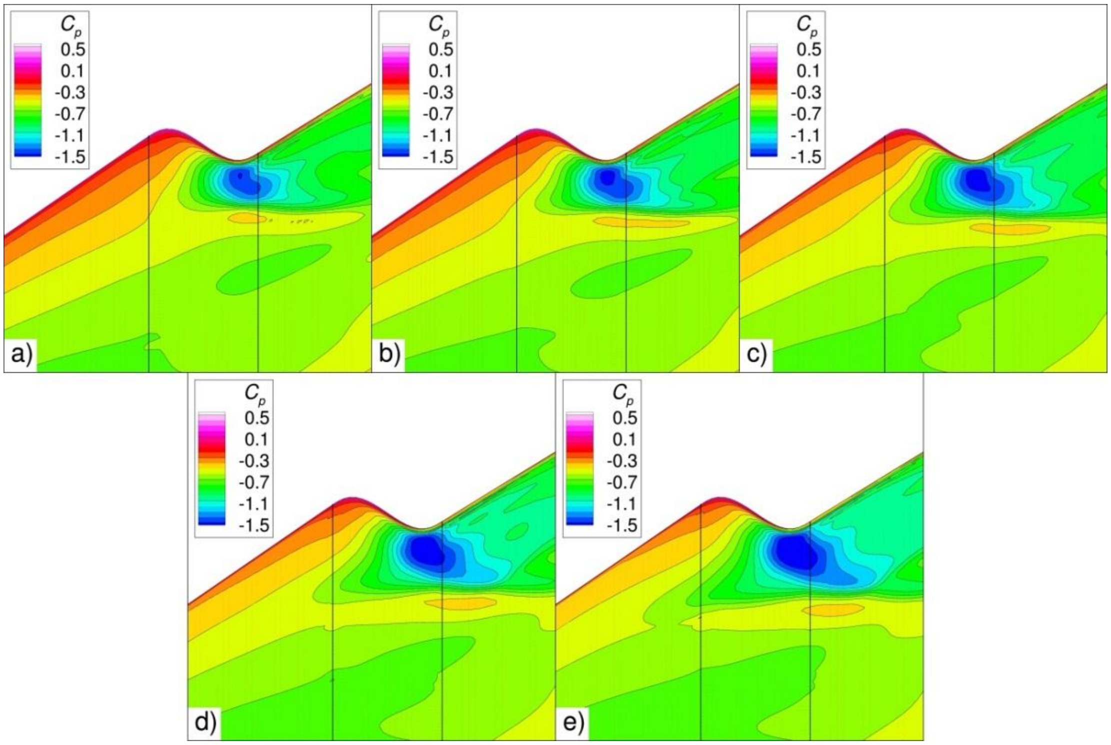


Fig. 29 C_p contours for the angle of attack simulations. a) 3.2°, b) 3.4°, c) 3.6°, d) 3.8°, and e) 4.0°.

As AoA increases, a region of low pressure near the leading edge of the inboard fairing grows in size, eventually spilling onto the glove test section and causing the local adverse pressure gradient observed at BL212 in Fig. 28. This flowfield phenomenon effectively sets the upper bound on the AoA range to 3.8°. Ideally, the upper and lower bounds of the AoA range would be established by the formation of local pressure minima within a few percent *x/c* from the leading edge of the test section. Instead, the AoA range for this project is determined by the glove C_L and a flowfield feature situated near the test section. The 3.5° - 3.8° band technically allows for the expected ±0.1° AoA tolerance specified in Belisle *et al.* [26]. However, this would likely be difficult to maintain in flight for the estimated 15 seconds required for data collection and as such should be addressed.

Mach Number Simulations

The Mach number simulations were motivated by the possibility of flying at lower Mach numbers during science test flights. The reasoning behind this can be traced to the presence of supercritical flow on the suction side of the glove. The *t/c* ratio, which is constrained by the 2-inch clearance requirement and glove chord length, is simply too large to avoid supercritical flow. Awareness of this behavior occurred during the design of the target C_p used for the optimization. By reducing the flight Mach number, it was hoped that the edge velocity near the glove would remain completely subsonic. This is important because there is some level of uncertainty as to how supercritical flow will affect NLF characteristics and DRE effectiveness.

M = 0.75 (Test Point A) serves as the primary target Mach number. M = 0.72 (Test Point G) was chosen because it marked the lower limit of the Mach requirement from Section I-E. Because supercritical flow was still anticipated at M = 0.72, M = 0.66 (Test Point F) was selected to guarantee a completely subsonic glove, even though it would not meet the Mach requirement. M = 0.75 was intended to serve as the nominal upper limit for this experiment, but an expected ±0.01 Mach number tolerance [26] provided the possibility for M = 0.76 (Test Point H). Section C_p results from BL212, BL234, and BL264 were again compared for the four Mach number simulations, as shown in Fig. 30.

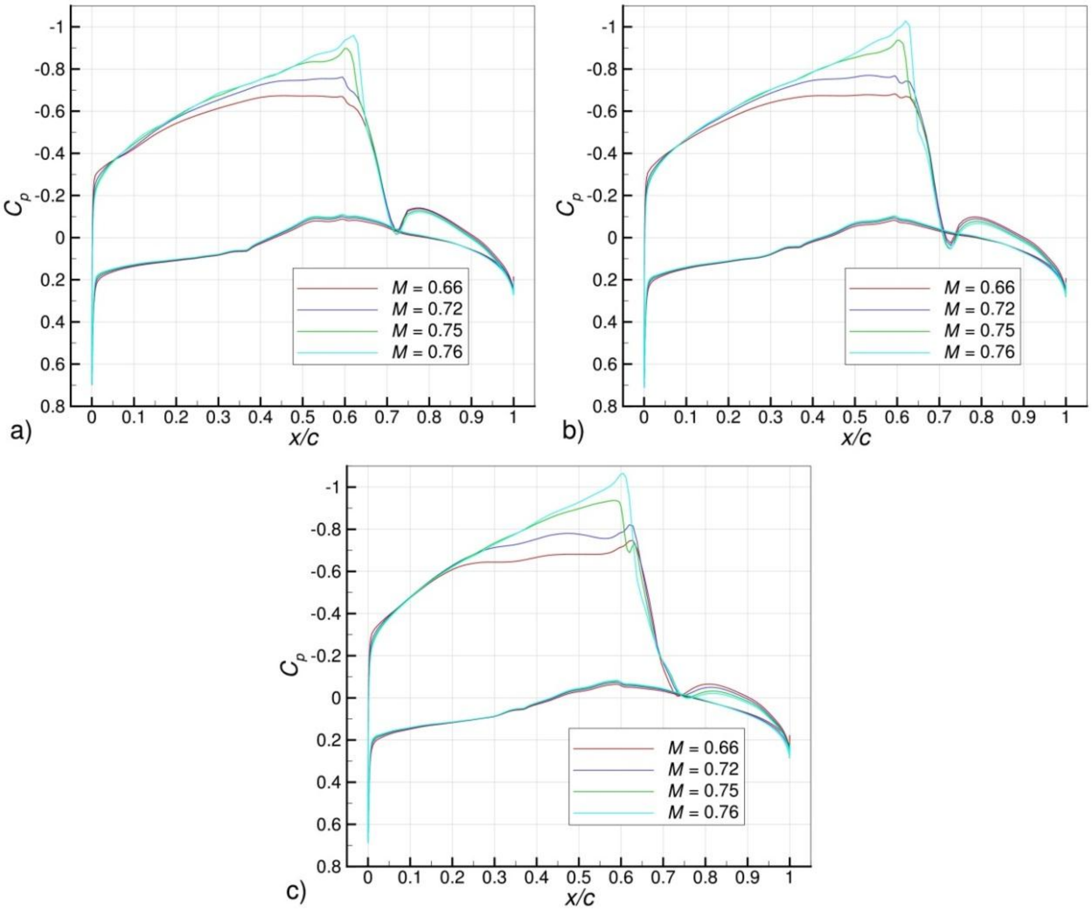


Fig. 30 Glove C_p comparison for Mach number simulations. a) BL212, b) BL234, and c) BL264.

The early hope that the general shape of the C_p curves would remain relatively invariant to Mach number changes was immediately dispelled. For the M = 0.72 and Test Point A simulations, the first 30% *x/c* on the suction side experienced similar behavior. From 30% to 60% *x/c*, however, the chordwise pressure gradient for the M = 0.72 C_p curves becomes less accelerated until they level off. The same effect occurs from 20% to 60% *x/c* with the M = 0.66 simulation in addition to different behavior near the suction-side leading edge. The M = 0.76 simulation conditions were nearly equivalent to Test Point A, but the C_p results differed from 50% to 62% *x/c*. The general trend observed shows that a decrease in Mach number leads to an appreciable decrease in then absolute suction-side pressure minimum. Due to this behavior, the C_L requirement for the glove cannot be met for Mach numbers less than M = 0.75 without significantly changing aircraft AoA. To show the extent of the supercritical flow, which was the motivation behind these simulations, Fig. 31 enlarges the BL234 C_p results and includes the sonic lines for each freestream Mach number.

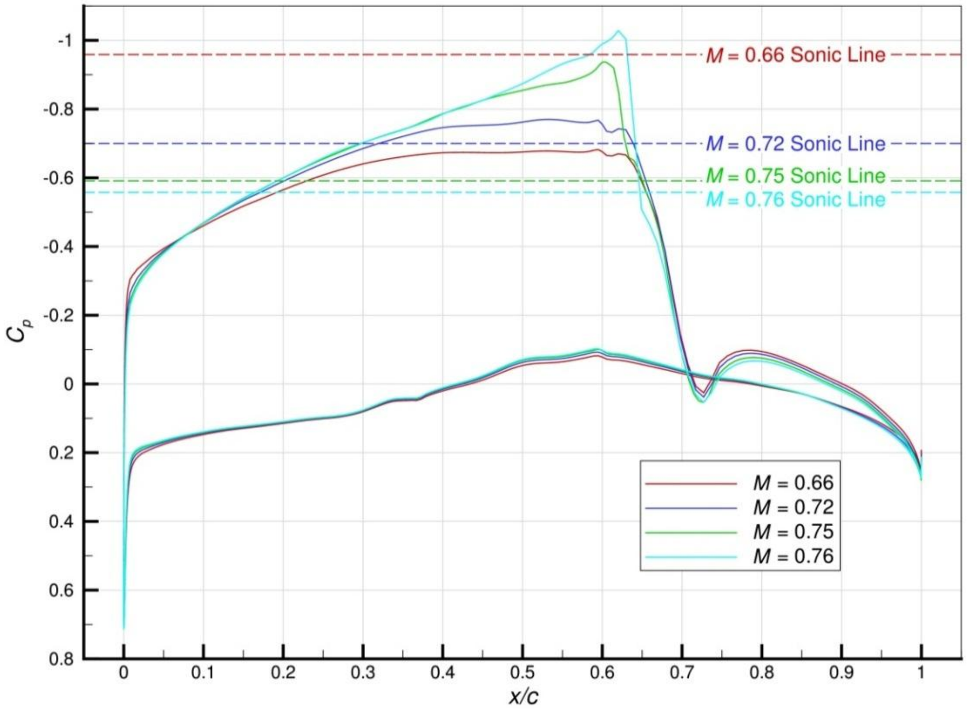


Fig. 31 Glove C_p results for Mach number simulations at BL234 with sonic lines.

Portions of the C_p curves above, *i.e.* more negative than, their respective sonic line show where the edge velocity of the airfoil section is supersonic. The supercritical region for Test Point A occurs from roughly 30% to 65% *x/c*. This range is increased slightly for *M* = 0.76 and lessened for *M* = 0.72. As expected, the *M* = 0.66 simulation generates entirely subsonic flow over the glove. In fact, had the shape of the C_p curve for *M* = 0.66 matched the Test Point A results, it would have still avoided supercritical flow. Unfortunately, supercritical flow significantly affects the glove pressure distribution. The regions of less-accelerated pressure gradients at the lower Mach number test points will no doubt lead to very different stability behavior as well.

Reynolds Number Simulations

The Reynolds number simulations were performed to explore solution sensitivity to Reynolds number changes. The values that were chosen were derived from the experiment's laminar flow requirements explained in greater detail below at the beginning of Section VI. NLF is explored from Re_c = 15 - 22×10^6 and SWLFC is explored from Re_c = 22 - 30×10^6. To investigate the NLF portion of the experiment, the lower limit of Re_c = 15×10^6 at BL270 (Test Point I) was chosen. To investigate the SWLFC portion of the experiment, the lower limit of Re_c = 22×10^6 at BL 270 (Test Point A) and the upper limit of Re_c = 30×10^6 at BL198 (Test Point J) were chosen. Because the glove has a non-constant chord across the glove, the span station was selected based on chord length to either keep all sections of the glove above the given Re_c (the smallest glove chord is at BL270) or keep all sections of the glove below the given Re_c (the largest chord is at BL198). Section C_p results from the three standard span stations are given in Fig. 32.

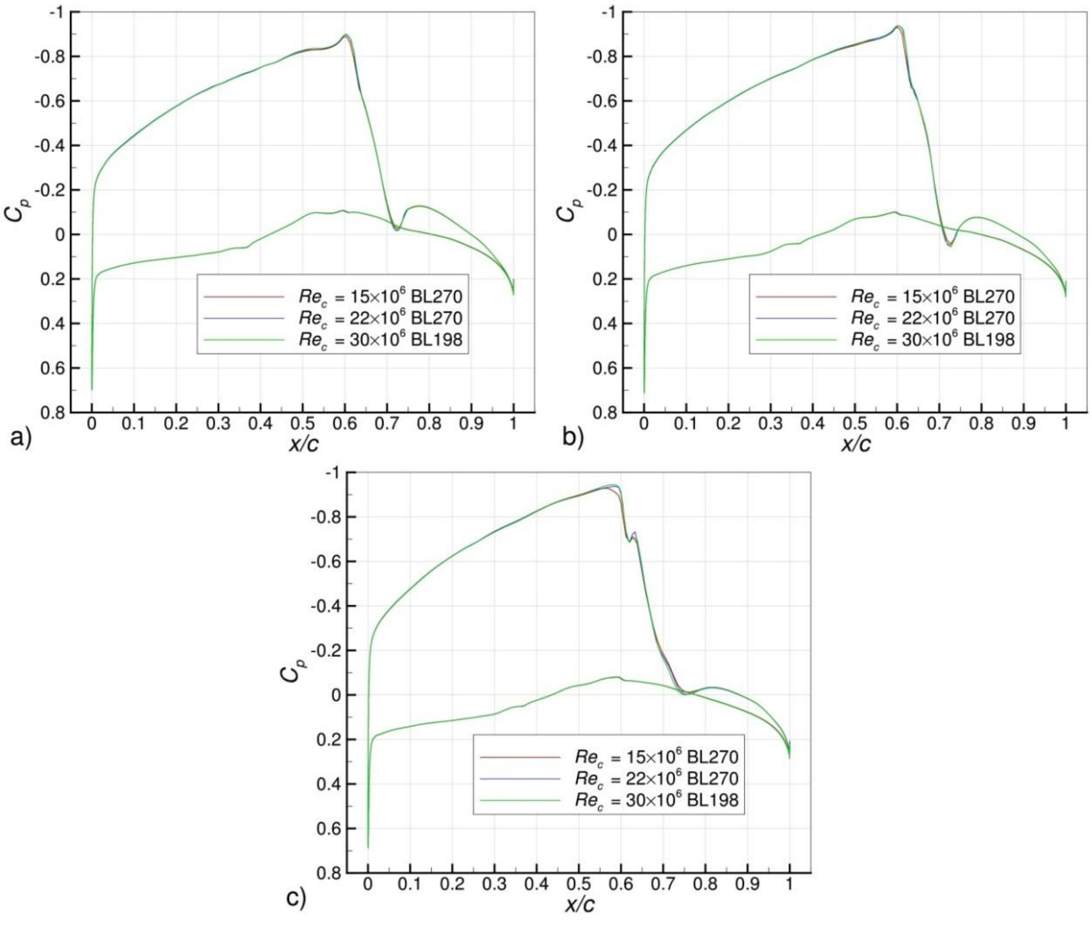


Fig. 32 Glove C_p comparison for Reynolds number simulations. a) BL212, b) BL234, and c) BL264.

Very little variation exists between the three Reynolds number simulations. The pressure minimum at BL264 for the Re_c = 15×10^6 simulation moves forward a few percent *x/c*, but this isn't particularly concerning because the location that this is observed is well outside the test section domain of influence. Stability analysis, covered in Section VI-C, provides a much clearer picture than the flowfield results on the differences that exist between these test points.

VI. FULL-AIRCRAFT STABILITY CALCULATIONS

Building on the full-aircraft simulations already completed, the results that are obtained from the full-aircraft stability calculations provide higher-order insights into the glove design effectiveness and serve as the principal arbiter on whether the design is acceptable or not. In addition to the aerodynamic requirements for the experiment given in Section I-E, two laminar-flow requirements were also specified. The first focused on the NLF portion of the experiment, requiring a suction-side transition location greater than or equal to 60% *x/c* and a transition front of at least 0.36 m (14 in) in the spanwise direction for $Re_c \geq 15\times10^6$. The second focused on the SWLFC portion of the experiment, requiring the suction-side transition location to be moved from its NLF location further aft by a factor of 1.5 when using the DREs for $Re_c \geq 22\times10^6$. For example, if the NLF transition location is at 40% *x/c*, the implementation of DREs must increase this location to 60% *x/c*. These requirements focus on the suction side because the primary transition detection tool, the IR camera, would be unable to view the pressure side. For the same reason, it is expected that DREs will not be used on the pressure side for this experiment. Table 7 summarizes the laminar flow requirements.

Table 7 Laminar flow requirements.

Laminar Flow Technique	Requirement
NLF (15 - 22×10^6)	$x/c_{tr} \geq 60\%$ *x/c*, transition front ≥ 0.36 m
SWLFC (22 - 30×10^6)	SWLFC $x/c_{tr} \geq 1.5 \times$ (NLF x/c_{tr})

The boundary-layer stability calculations performed on the flowfield data of all test point simulations will not only help determine if the laminar flow requirements can be met, but provide a greater sense of how the stability behavior is affected by different freestream conditions. Information important to the SWLFC technique, such as identifying possible control wavelengths, the most unstable wavelengths, and DRE array placement location, will also be obtained from the stability calculations.

The boundary-layer stability behavior is obtained from the full-aircraft simulations through a methodical process, starting with flowfield data extraction and conditioning. This step, which was divided into several tasks itself, required a bevy of codes to transform the flowfield information into a useful format. Next, LST calculations were performed and select results are given. Upon completion of the stability analysis process, a thorough understanding of glove design behavior will have been presented and conclusions about the design effectiveness can be made.

A. Flowfield Data Extraction and Conditioning

Precisely extracting boundary-layer data is a challenging problem to solve. Historically, the only attempt to do so for a FRL project, until recently, occurred during the O-2A computational efforts. The primary aim of the data extraction process is to obtain velocity-component, pressure, temperature, and density values along local, wall-normal rays for use in boundary-layer stability calculations. Rhodes [20] implemented a user-defined function (UDF) within FLUENT to generate a semi-wall-normal mesh in the boundary-layer region along a chordwise section. The required flowfield data could then be obtained directly at the mesh nodes. This method, however, wasn’t a true wall-normal approach because the grid was locally perpendicular to the airfoil section but not locally perpendicular to the complete 3-D surface of the SWIFT model. For this reason, a different data extraction process was necessary.

Prior to creating a new method to extract boundary-layer information, significant planning went into the procedure that would need to be executed in order to obtain the most meaningful results. Much of this relies on the way the stability calculations are carried out. The LAngley Stability and TRansition Analysis Code (LASTRAC) [35] was used for all boundary-layer stability calculations. LASTRAC was developed to serve as a general-purpose, physics-based transition prediction code and can perform calculations using LST, linear parabolized stability equations (LPSE), and nonlinear parabolized stability equations (NPSE). The stability characteristics of an infinite swept wing model is one of the possible types of problems LASTRAC was designed to solve, making it a suitable tool to complete the SARGE stability calculations. The coordinate system setup used by LASTRAC for the infinite swept wing model, given by the LASTRAC User Manual, is shown in Fig. 33.

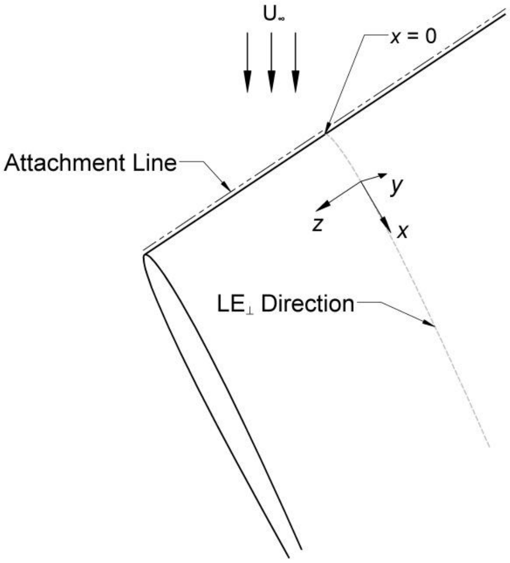


Fig. 33 Infinite swept wing coordinate system setup used by LASTRAC.

The important items to note in the diagram above is that LASTRAC marches along a section that is normal to the leading edge (LE⊥), starting at the attachment line location, and that the coordinate system along this section is always local, with the *x*-axis pointing away from the leading edge and parallel with the marching direction, the *y*-axis pointing away from the surface and perpendicular to the marching direction, and a right-handed *z*-axis. The idea behind this method is that the flow properties for an infinite swept wing are constant in the direction parallel to the leading edge, meaning the stability characteristics are also constant for any LE⊥ airfoil section. SARGE, however, is a conical swept wing design, which required adjustments to be made to the standard method.

In order to mimic the LASTRAC infinite swept wing marching method, a marching direction should be chosen so that the boundary-layer properties are constant in the direction perpendicular to it. Because some level of spanwise uniformity was achieved across the test section, it is conceivable that the flow properties in the directions parallel to constant-*x/c* lines experience minimal change. Therefore, the marching direction should remain locally perpendicular to constant-*x/c* lines to best approximate the infinite swept wing method. This marching scheme is not possible in a literal sense, unfortunately, because of the limited test section span. Doing so would result in marching off the test section for starting points near the inboard boundary. However, this problem is solved by using the "constant properties along constant-*x/c* lines" assumption, which makes a marching direction along a constant span location, *e.g.* BL234, equivalent to a marching direction that is locally perpendicular to constant-*x/c* lines. Fig. 34 illustrates the conical swept wing marching (CSWM) method.

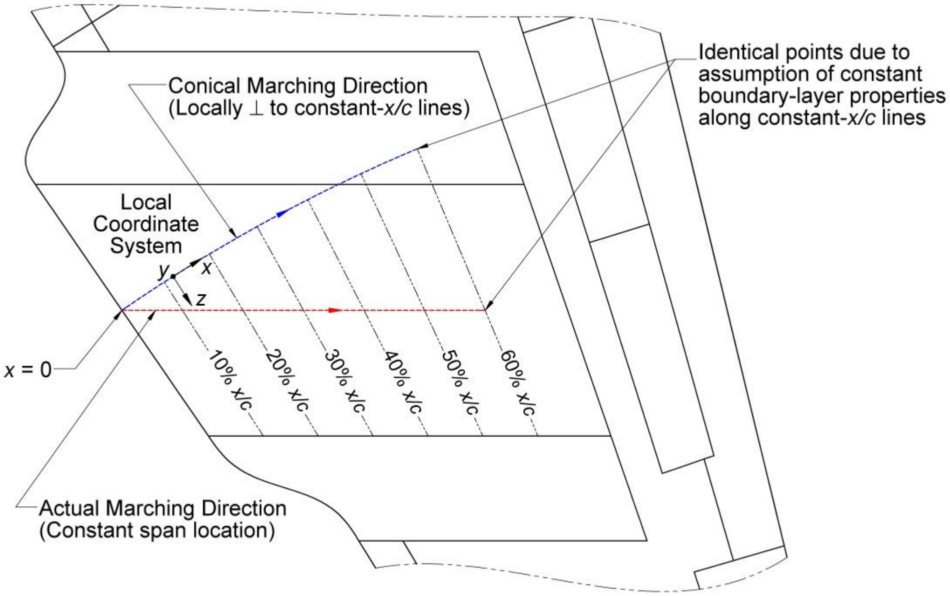

Conical Marching Direction
(Locally ⊥ to constant-*x/c* lines)
Identical points due to assumption of constant boundary-layer properties along constant-*x/c* lines
Local Coordinate System
y
x
z
x = 0
10% x/c
20% x/c
30% x/c
40% x/c
50% x/c
60% x/c
Actual Marching Direction
(Constant span location)

Fig. 34 Conical swept wing marching method.

The coordinate system shown in Fig. 34 above was modified slightly from the infinite swept wing marching method to match CSWM method. The *y*-axis remains wall normal, but the *z*-axis is now aligned with constant-*x/c* lines and the *x*-axis is derived from the other two. The positive *x*-axis direction always points away from the leading edge, which defines the positive *z*-axis direction using a right-handed system. This coordinate system will be referred to as the conical coordinate system from this point forward.

Flowfield data extraction and conditioning can be broken into four tasks: surface grid export, extraction journal file creation, boundary-layer data export, and LASTRAC mean flow file creation. A flowchart of the process, which will be described in detail below, is shown in Fig. 35.

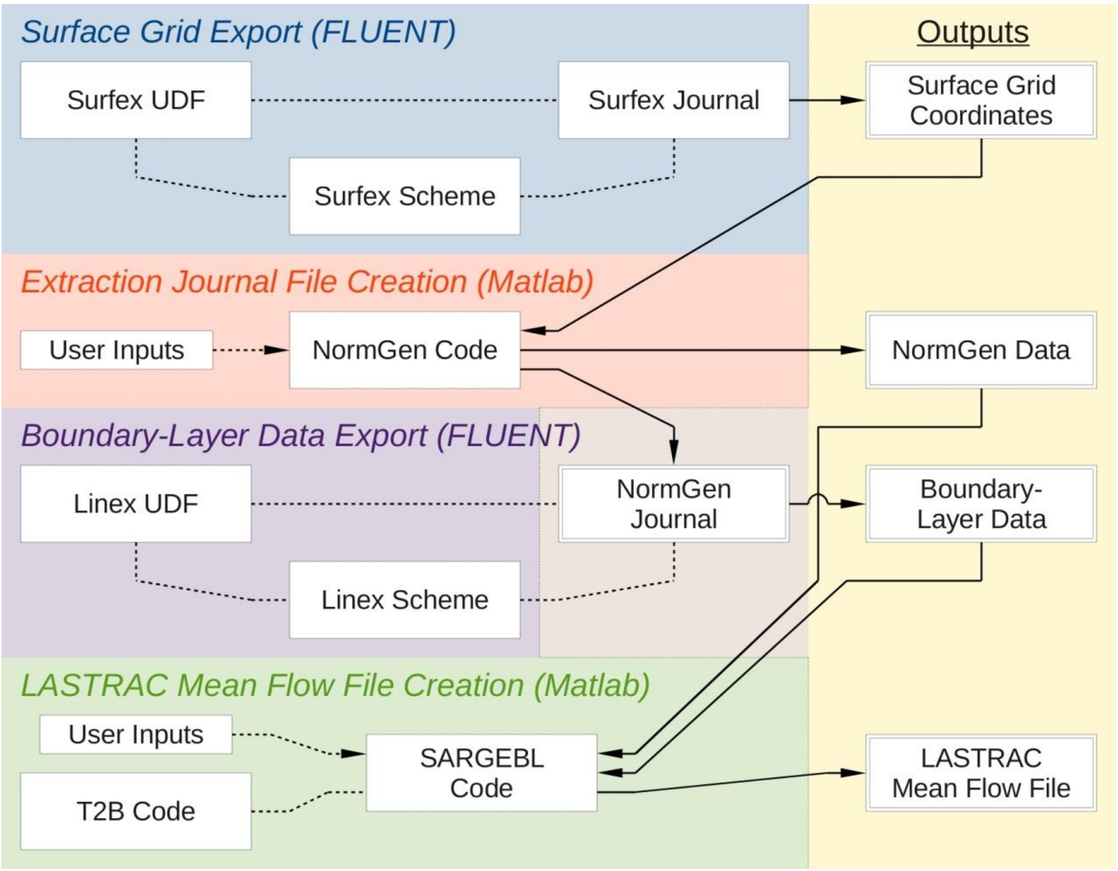


Fig. 35 Flowchart of the flowfield data extraction and condition process.

Surface Grid Export

The first data extraction and conditioning step involves exporting the coordinates of the test section surface grid. The grid nodes are used to set the span location for marching and establish the local conical coordinate system directions along a given span location. With the assistance of ANSYS Customer Support, a UDF called Surfex was written to perform an ASCII-format coordinate export with a user-specified number of significant digits. All surface grid coordinates exported for this project contained sixteen significant digits.

To increase functionality and automate this task, Surfex scheme and journal files were also created to work in concert with the Surfex UDF. The scheme file allows the user to pass information entered through FLUENT’s TUI to the UDF, rather than change the UDF source code each time it is used. In this case, the specified information includes the test section boundary name and the desired export file name.

The journal file, which was edited to update the scheme inputs for each test point simulation, carries out the entire procedure automatically through the TUI by performing UDF and scheme loading and execution duties. The end result is a text file output that includes global *X*, *Y*, and *Z* coordinates for all grid nodes that are part of the test section boundary. The Surfex journal, scheme, and UDF files can be viewed in Appendix D-1, -2, and -3.

Extraction Journal File Creation

The second step in the flowfield data extraction and conditioning process involves creating a journal file that automatically performs the boundary-layer data export for a given span location in FLUENT. A utility code called NormGen was written in MATLAB [36] to accomplish this task. The surface grid coordinate file and several user-specified parameters serve as inputs to NormGen, while the NormGen journal file and NormGen data serve as outputs.

The NormGen code begins with surface grid coordinate import and conditioning. As stated in Section IV-A, the chordwise grid lines were placed at constant span locations. Once the glove coordinates are read into MATLAB, they are placed into sets according to their span location. The user can then enter the desired span location, *e.g.* BL234, for which the data extraction will be performed along.

The next step NormGen takes is to establish local conical coordinate systems at all nodes along the given airfoil section. The *z*-axis is defined using the spanwise grid lines, which were deliberately oriented along constant-*x/c* lines for this purpose. In order to determine the *y*-axis, an intermediate step involving the creation of a temporary axis that is locally parallel to the airfoil section is necessary. Though they are not mutually perpendicular, both the temporary axis and *z*-axis are parallel to the test section surface. By performing a cross-product of these two directions, the wall-normal *y*-axis could be defined. Finally, the *x*-axis was obtained by performing a second cross-product with the already established *y*- and *z*-axes. Fig. 36 displays several local coordinate systems near the leading edge of the test section to better illustrate the conical coordinate system concept.

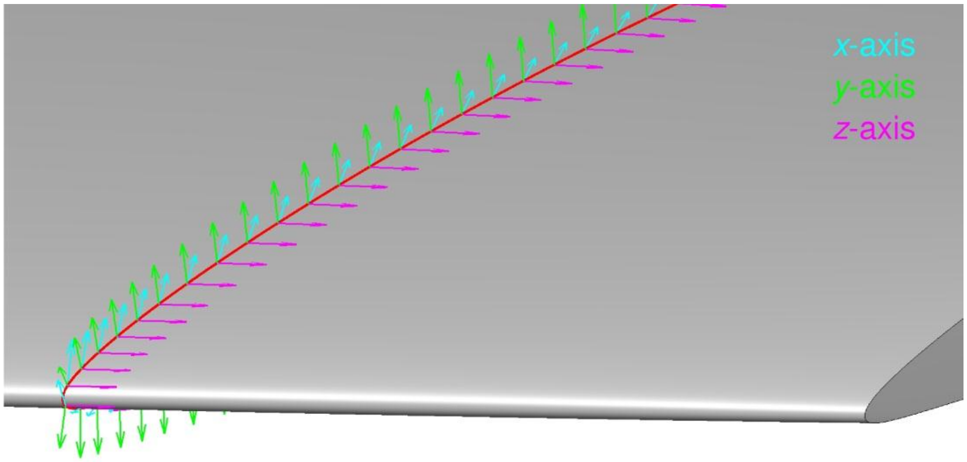


Fig. 36   Local conical coordinate system examples near the test section leading edge.

Due to the use of the CSWM method, the coordinates for the given airfoil section must be transformed to be consistent with the assumed marching path. These adjusted coordinates will be used to find the body-fitted distance from the attachment line location as well as for curvature calculations. All airfoil section coordinates are in essence translated along their respective constant-*x/c* lines until they match the CSWM path shown previously in Fig. 34. This is simple to implement since the marching path planform is a circular arc that begins at the leading edge, establishing its radius, and is centered at the conic focus of the glove and wing design. A comparison of airfoil section and adjusted coordinates is given in Fig. 37.

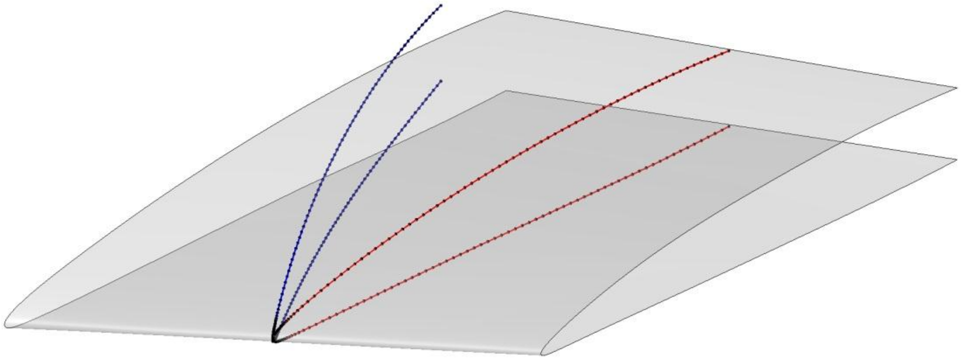


Fig. 37   Airfoil section and adjusted coordinate comparison.

NormGen allows the user to filter nodes that are too close together by specifying a tolerance on point spacing for the adjusted coordinates. The filtering occurs primarily near the leading edge, where a high density of grid nodes was necessary to capture the rapidly changing flow properties. However, a smaller density can be used for stability calculations without altering the solution. Filtering nodes near the leading edge also assists in providing a smooth curvature distribution for the adjusted coordinates. Performing the curvature calculation with unfiltered nodes leads to significant curvature variance near the leading edge. This behavior could affect stability calculations that employ curvature and lead to erroneous results. The curvature calculation performed on the adjusted coordinates is quite straightforward and uses a parametric method to avoid numerical discontinuities.

At this point in the NormGen process, the NormGen journal file can be written. The journal file itself is designed to carry out three major groups of tasks within FLUENT. The first is the creation of wall-normal lines from which the data will be extracted from, the second is the extraction itself, and the third is the deletion of the wall-normal lines, which will be called data lines henceforth. To create the data lines in FLUENT, start and end points must be specified. The start points come from the nodes along the given airfoil section, which of course reside on the glove surface. The end points exist in the freestream and are calculated using the local *y*-axis, *i.e.* wall-normal, direction of the conical coordinate system and a user-specified data line length. All sets of start and end points are written into data line creation commands in the journal file. The next group of commands employs a second UDF and scheme called Linex that are responsible for the data extraction. Both of these files are covered in greater detail in the boundary-layer data export task below. The final group of NormGen journal file commands serve as a cleanup step, deleting all data lines after the boundary-layer data is exported. This returns the FLUENT case and data files back to the state they existed in prior to the export. The finished product is a complete journal file that will execute all commands automatically through FLUENT’s TUI.

Much of the information calculated by the NormGen code is needed to create the LASTRAC mean flow file, which is also further discussed below. The NormGen data saved for use in steps following the data extraction process includes the local conical coordinate system directions, the coordinates for the given airfoil section, the adjusted coordinates, the adjusted coordinate curvature, and the flight conditions for the particular test point simulations.

The NormGen process was completed for six span locations for all ten test point simulations. The locations selected were BL204, BL212, B222, BL234, BL246, and BL264. The decision process behind these choices considered the extent of the domain of influence for the test section, the areas of the glove with the most spanwise uniformity, and limiting locations near the test section boundaries. The NormGen code and its functions are given in Appendix D-4.

Boundary-Layer Data Export

The third step in the flowfield data extraction and conditioning process is to export the boundary-layer data from a full-aircraft simulation. The inputted NormGen journal file automates this process in conjunction with the Linex UDF and scheme. A significant reduction in the extraction time can be achieved by saving a copy of the completed FLUENT simulation that keeps only the laminar subzone portion of the solution active. The output of the extraction is a text file containing all the pertinent boundary-layer data needed to create the LASTRAC mean flow file.

After the NormGen journal file has finished creating the data lines in FLUENT, the Linex UDF and scheme are implemented. ANSYS Customer Support also assisted in writing these codes. The Linex UDF uses a recording and writing process similar to the Surfex UDF, but has added functionality. It has the capability of not only outputting geometric information, but flowfield information as well. Velocity components, pressure, temperature, and density along data lines are required to create the mean flow file used by LASTRAC. The Linex UDF employs built-in functions within FLUENT to discretize the data lines and interpolate the flowfield solution. The discretization is performed by placing a point at every data line and cell wall intersection location. Since the mesh near the glove surface is very refined, the data lines will contain a large number of points near the wall as well. FLUENT uses a weighted cell-node interpolation for points not coincident with cell nodes where the flowfield data is known. This was eventually found to be an effective technique. The Linex UDF internally records all geometric and flowfield data at all data line points and then writes it to an ASCII-format text file.

The Linex scheme performs the same function as the Surfex scheme, passing information specified in the journal file through FLUENT’s TUI to the UDF. In this case, the data line name, which is designated during the line creation process, and boundary-layer data output file name are given. Once all data line information is extracted and written to the export file, the NormGen journal deletes all data lines. The Linex UDF and scheme files can be viewed in Appendix D-5 and -6.

LASTRAC Mean Flow File Creation

The fourth and final step in the flowfield data extraction and conditioning process involves creating the mean flow file used by LASTRAC to complete boundary-layer stability calculations. A utility code called SARGEBL was written in MATLAB to condition the data exported from FLUENT to fit the LASTRAC mean flow file format. The NormGen and boundary-layer data serve as inputs while the mean flow file will serve as the output for this task.

SARGEBL begins by loading the NormGen data created earlier and reading the boundary-layer data into MATLAB. A filtering algorithm is implemented to clean up the boundary-layer data and ensure an error-free mean flow. From here, the conditioning process begins, starting with the calculation of the wall-normal distance for all points along their respective data lines, with the wall serving as $y = 0$. This is accomplished with simple coordinate translations and vector magnitude calculations. Next, the global *U*-, *V*-, and *W*-velocity components from FLUENT are projected onto the local conical coordinate systems, creating local *u*-, *v*-, and *w*-velocity components. No conditioning is required for pressure, temperature, and density data. Wall and boundary-layer-edge parameters are recorded at this point in preparation for the next step. Groups of boundary-layer data corresponding to particular airfoil section nodes will be referred to as stations, following the notation used by LASTRAC. Examples of conditioned boundary layer velocity profiles at several *x/c* locations are displayed in Fig. 38.

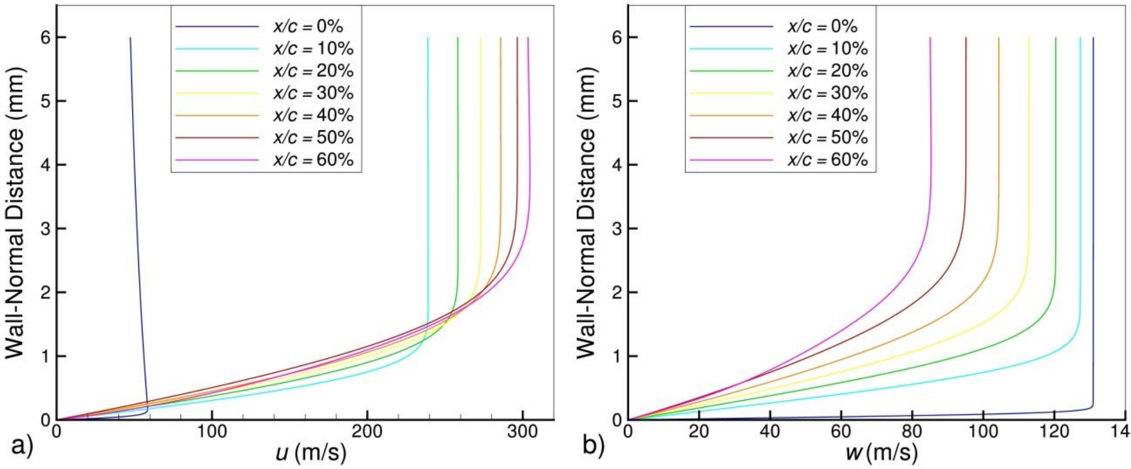


Fig. 38 Conditioned boundary-layer data along the suction side of the BL234 airfoil section. a) *u*-velocity profiles and b) *w*-velocity profiles.

Once the boundary-layer data is properly conditioned, SARGEBL determines the attachment line location by finding the point at which the *u*-velocity near the surface of the test section reverses direction. Suction-side data is above and downstream from this point while pressure-side data is below and downstream from this point. Next, station properties for each side of the airfoil section are calculated. The required information, which is listed and explained in the LASTRAC User Manual, includes the station number, data point count, body-fitted distance from the attachment line location, length scale, Reynolds number, curvature, spanwise radius, *x/c* location, and edge *u*-velocity, temperature, and density. The adjusted coordinates determined previously are used for the station body-fitted distance calculation while the airfoil section coordinates are used for the station *x/c* location calculation. Because the station spanwise radius parameter will not be used in the stability calculations, it was set to zero. The curvature calculated earlier was included in all the mean flow files, but was always disabled for LST calculations.

The final step in SARGEBL is to write the LASTRAC mean flow file. The user is given the option of choosing which side of the test section to write the mean flow file for. Following the file setup given by the LASTRAC User Manual, the boundary-layer data and station properties for the given airfoil section side, as well as several global properties, are written to an ASCII-formatted text file. However, LASTRAC ultimately requires an unformatted, or binary, mean flow file. A FORTRAN code called T2B was written to covert the text file to a binary file. This step is executed within the SARGEBL code to automate the process. The end result is a LASTRAC mean flow file for the given side and airfoil section. SARGEBL was used for both sides of all six span locations listed earlier and for all ten simulations. The SARGEBL and T2B codes are given in Appendix D-7 and -8.

B. Boundary-Layer Stability Calculation Procedure

Once the LASTRAC mean flow files were created, boundary-layer stability calculations could begin. The primary focus of these calculations is to explore the stationary crossflow and T-S disturbance behavior for the test section. The results generated will give a good sense as to whether the laminar flow requirements can be met with the current glove design. For each calculation, the procedure used to run LASTRAC and analyze the stability results remained fairly constant. Important shared concepts are covered in this section.

In addition to the flowfield data stored in the mean flow file, LASTRAC requires an input file that contains all the settings used in the stability calculations. Input files are typically very similar from calculation to calculation, generally only requiring updates to parameters like the mean flow file name, the starting and ending stations, and the wave properties of the disturbance. Examples of LASTRAC input files for LST crossflow and T-S calculations are given in Appendix E-1 and -2. Because of the numerous setting options available within LASTRAC, which are detailed in the User Manual, independence studies were performed to ensure the stability results didn't change. Numerous tests individually checking particular parameters verified that the settings predominantly used would produce independent solutions. The local workstation computer discussed in Section V-A was used to perform all stability calculations presented in this thesis.

Once a stability calculation is completed for a given mean flow file and input file combination, LASTRAC outputs a text file containing important stability characteristics. The User Manual again provides comprehensive descriptions of the data produced by LASTRAC. Included in the output file are N-factor results, which are calculated using the Smith-Van Ingen e^N method [37]-[38] and serve as a key stability analysis tool for the glove design. Developed as a transition prediction tool, the e^N method involves the calculation of an N-factor using the disturbance amplitude ratio, A/A_0, between two given points along the marching path. For LST, the Branch I neutral point marks the location where disturbance growth begins and as such is typically used as the location for the initial amplitude, A_0. The Branch II neutral point marks where the disturbance ceases to grow and begins to decay. This would correspond to the maximum *N*-factor. Exponential growth is assumed to occur between the Branch I and II points, leading to the relationship given by Equation 3. Fig. 39 illustrates how the amplitude ratio typically changes with *R*, the boundary-layer Reynolds number which is a function of the body-fitted *x*-location.

$$N = \ln(A/A_0) \tag{3}$$

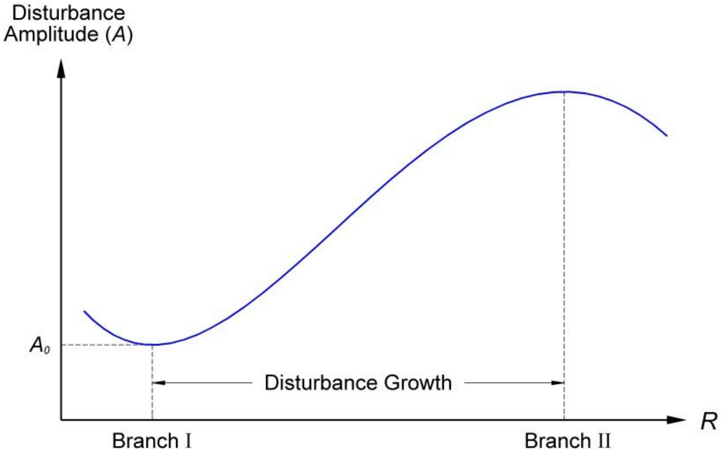


Fig. 39 Illustration of typical boundary-layer disturbance growth.

The e^N method is a particularly useful design tool because empirical correlations between *N* and transition location can be made in the absence of experimental data or more complex calculations. This allows for a quick and simple comparison of the different disturbance modes defining the stability behavior. However, it must be used with caution because the initial amplitude, which is governed by environmental initial conditions, is usually unknown and could lead to large errors. To help avoid these errors, transition *N*-factors are typically tailored to fit problem at hand using past experimental experience as a guide.

For the SARGE project, several limiting *N*-factor criteria were designated for crossflow and T-S disturbances. Recall that a SWLFC design calls for the suppression of T-S instabilities, making crossflow the dominant transition mechanism. With this in mind, two transition N-factor limits were specified for crossflow. The first represents an operationally-realistic painted leading edge and assumes transition will occur at $N = 9$. The second represents a highly-polished leading edge and assumes transition will occur at $N = 14$. In the event that neither *N*-factor criterion is reached before the end of the laminar subzone, transition will be assumed to occur at this location: 60% *x/c*. Due to the steep adverse pressure gradient beyond this point, the T-S mechanism would likely lead to turbulence quickly. Because transition due to T-S is not expected forward of 60% *x/c*, an *N*-factor design limit rather than transition limit is used. This limit was set at $N = 6$ and represents the maximum allowable *N*-factor due to the T-S mechanism. Using a lower *N*-factor criterion will ensure that the glove design is not operating on the margins of T-S dominated transition. These criteria were applied to all boundary-layer stability calculations described in the next section.

C. LST Calculations and Results

Linear stability theory is governed by the Orr-Sommerfeld equation. Its derivation begins with the simplified Navier-Stokes equations using the parallel-flow assumption, where all flowfield variables are a function of y. The equation variables are then broken into basic state (mean flow) and disturbance quantities. Because the pure basic state quantities satisfy the Navier-Stokes equations by themselves, they can be eliminated, leaving mixed and pure disturbance quantities. From here, the nonlinear terms, such as the product of two or more disturbance quantities, are removed under the assumption that all disturbances are small. Because the disturbance quantities are functions of y only, the separation of variables technique can be performed using a normal mode wave to convert the current partial differential equations into ordinary differential equations. The wave definition is dependent upon the x-direction wavenumber, α, the z-direction (spanwise) wavenumber, β, and the angular frequency, ω. After implementing the separation of variables approach, the equations are combined and rearranged to arrive at the Orr-Sommerfeld equation, which is solved using an eigenvalue approach. Often, it is easier to describe the disturbance wave in terms of spanwise wavelength, λ, and frequency, f. The conversions from β and ω, respectively, are given in Equations 4 and 5 below.

$$\lambda = \frac{2\pi}{\beta} \tag{4}$$

$$f = \frac{\omega}{2\pi} \tag{5}$$

Two primary sets of LST stability calculations were performed. The first set examined stationary crossflow. These calculations require user-specified spanwise wavelengths with a constant zero frequency. The second set examined T-S with a wave propagation direction parallel to the marching path (2-D). This requires user-specified frequencies with a constant zero spanwise wave number. These two standard LST calculations were performed for all LASTRAC mean flow files created using the process described in Section VI-A. In total, 240 calculations completed and analyzed. Select results are presented below, while the comprehensive set of the results can be found in Appendix F. Two supplemental sets of LST calculations were also performed for specific cases to provide a better understanding of the stability characteristics of the glove. The first set explored 3-D T-S, which has a nonzero spanwise wavenumber, while the second set explored traveling crossflow, which has a nonzero frequency.

While not as accurate as its LPSE and NPSE counterparts due to simplifications in its formulation, LST calculations are both relatively quick and straightforward while still providing a great deal of useful information about the boundary-layer stability characteristics. This section will first present the stability results from the principal test point, followed by a comparison of results for the angle of attack, Mach number, and Reynolds number simulations. Finally, results from the 3-D T-S and travelling crossflow calculations will be discussed.

Stability Characteristics for the Principal Test Point Simulation

The stability characteristics of the glove under the conditions that define Test Point A were the most heavily scrutinized of all the stability calculations because this test point was chosen to serve as the primary target for the SWLFC portion of the test flight phase of the experiment. As noted previously, the main focus rests on stationary crossflow, observing how it varies across the glove. T-S was of course checked to ensure no unexpected behavior was present. Furthermore, more attention was placed on the suction side of the glove due to the laminar flow requirements outlined in the beginning of Section VI.

The pressure-side stability behavior will also be examined, but under the assumption that the transition location will be dependent upon its NLF characteristics only. As stated earlier, all *N*-factor results not shown directly in this section can be found in Appendix F.

Beginning with the suction-side stationary crossflow results, several key points should be drawn from the *N*-factor plots for this and all other simulation stability calculations. First and foremost, the consistency of the stability behavior should be noted. Large deviations, especially in the center of the test section, would invalidate the CSWM method and could signal serious design problems. If no significant variations are present, the expected transition location for the two crossflow *N*-factor criteria should be studied. These *x/c* values approximate the amount of laminar flow that would occur naturally. This is important for both the laminar flow requirements outlined earlier in Table 7. Finally, the stability characteristics important to the SWLFC technique should be examined, including the critical wave, possible subcritical control wavelengths, and the Branch I neutral point of the critical wavelength which will dictate the DRE array placement location. Beginning with the suction-side stationary crossflow results, *N*-factor plots for the airfoil sections at BL212, BL222, BL234, and BL246 from the principal test point simulation are given in Fig. 40 below.

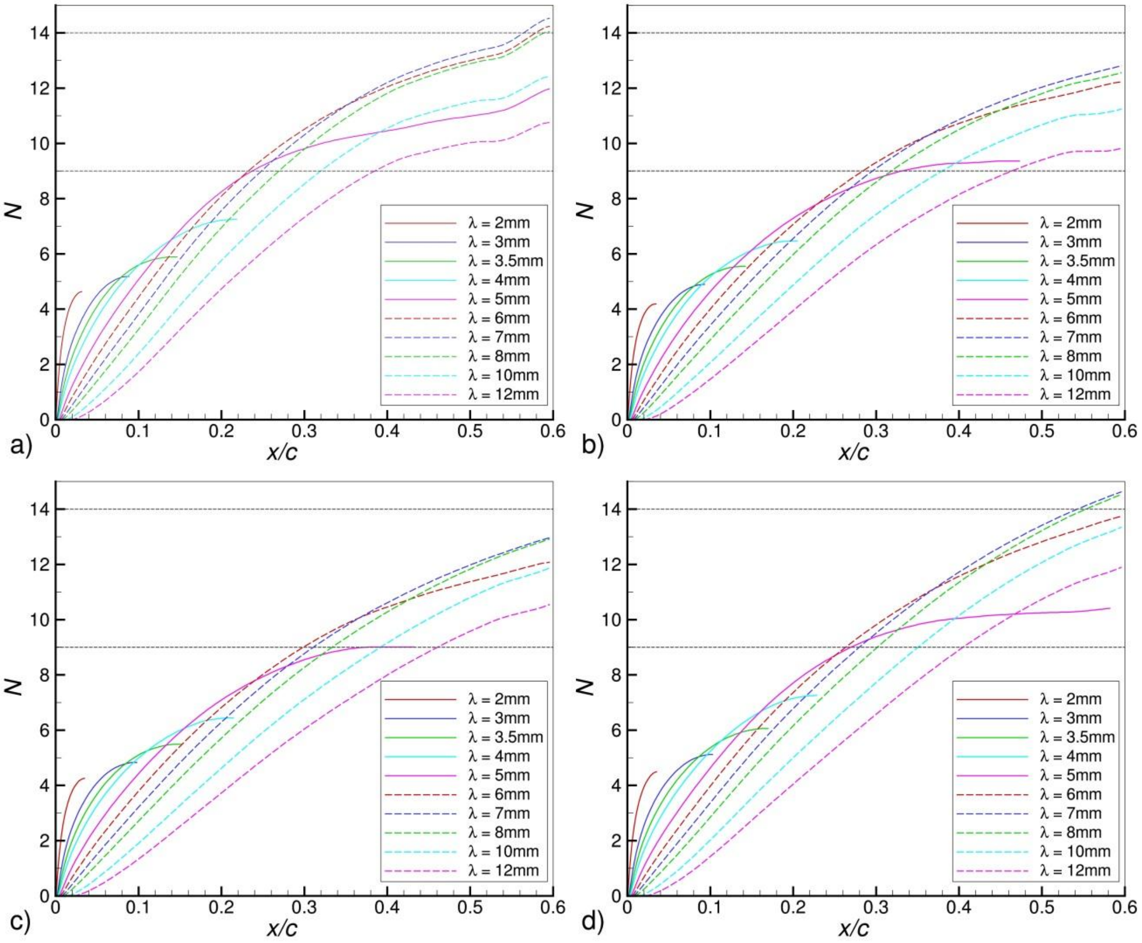


Fig. 40 Suction-side crossflow *N*-factor results from Test Point A. a) BL212, b) BL222, c) BL234, and d) BL246.

The stability characteristics for all four sections are fairly consistent considering the sensitivity of boundary-layer stability calculations. It shows that little variation in the stability behavior exists across more than half of the test section span. This is a good sign because it verifies the methods described in Section VI-A. The predicted transition locations for both transition criteria differed by no more than 6% *x/c* from BL212 to BL246, with the most laminar flow occurring in the center of the test section (BL234). A clearer graphical approach used to analyze the predicted transition location in the form of *N*-factor contour plots is discussed and presented shortly below. The stability behavior features important to the SWLFC technique were also consistent. The 7 mm wave was the most unstable wave for all four sections. The 3.5 mm wave could be used for control, since it grows to *N*-factors between 5 and 6 and downstream roughly 15% *x/c*. The 4 mm wave is another possible candidate. It grows further downstream but is closer to the critical wavelength and comes with slightly larger *N*-factors. The 7 mm wave consistently begins growing at approximately 0.64% *x/c*, or 44.2 mm (1.74 in) downstream of the attachment line, across the entirety of the test section. The DRE array would be positioned at this location to force the subcritical wavelength and subsequently suppress the critical wavelength.

While the center portion of the test section returned consistent results, the airfoil sections near the boundaries, BL204 and BL264, experienced larger disturbance growth, causing the predicted transition locations to move much further forward. This is not entirely unexpected due to these sections' proximity to the test section boundaries, where conical-swept-wing flow would likely be spoiled by the nearby fairings.

The T-S calculations that were performed for the suction side were also very consistent, in that they exhibited little growth over the first 60% *x/c*. The $N = 6$ criterion was never reached and the frequencies examined typically grew no larger than $N = 1$. However, because supercritical flow is present over a significant portion of the test section's suction side, 3-D T-S waves, which have wave propagation directions that are not parallel to the marching direction, could experience larger growth – possibly enough to cause transition. Nearly all other test points that were examined exhibited similar supercritical behavior. Further discussion on whether or not 3-D T-S on the suction side will pose a serious problem can be found below.

The pressure-side stability calculations were primarily used for predicting transition since the implementation of the SWLFC technique for this side was unlikely. To better convey transition locations across the glove for crossflow, *N*-factor contour plots were generated. *N*-factor data for all studied wavelengths is used to find the *x/c* location that a particular *N*-factor criterion is first met. Performing this algorithm at each of the six airfoil sections defines the transition front for a particular *N*-factor criterion and shows how transition moves aft as *N* increases. Only the domain-of-influence region is plotted since areas outside of this are expected to be turbulent. Fig. 41 shows suction-side and pressure-side *N*-factor contours on the test section for the principal test point simulation stationary crossflow calculations.

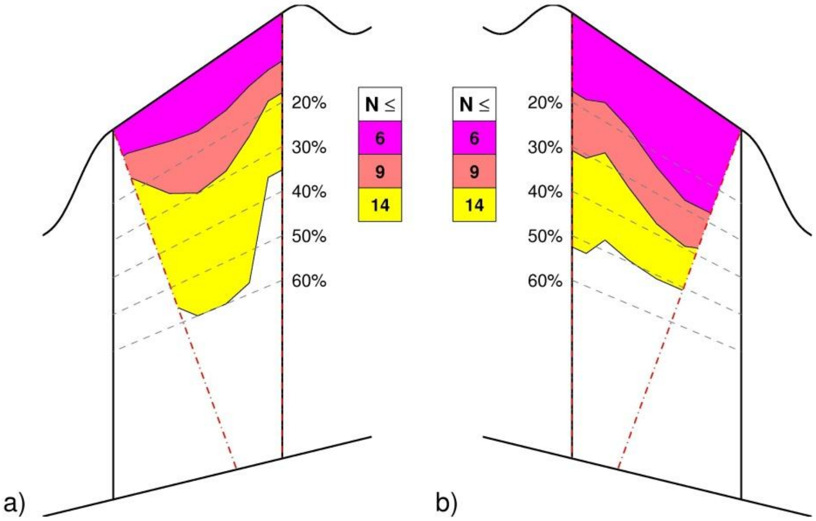


Fig. 41 Crossflow *N*-factor contours from Test Point A. a) suction side and b) pressure side.

For the suction side, the *N* = 9 contour line predicts a transition location of roughly 25% *x/c* across the middle portion of the glove, leaving ample room to increase laminar flow by a factor of 1.5 through the use of DREs without saturating at the 60% *x/c* location. This would allow the SWLFC requirement to be met. For the *N* = 14 contour line, demonstrating DRE effectiveness would be difficult because the transition region is already approaching 60% *x/c* in the center of the glove, preventing gains in laminar flow to be made. In order to best demonstrate the DRE technology, a transition location of roughly 30% - 40% would be preferred, meaning that the surface roughness would need to be slightly smoother than the operationally-realistic painted surface given the current predictions. For the pressure side, laminar flow can be achieved naturally anywhere between 32% *x/c* for *N* = 9 and 50% *x/c* for *N* = 14.

The pressure-side T-S calculations exhibited larger disturbance growth than the suction-side results. This was expected because the pressure gradient for the pressure side is less accelerated than the suction side. The maximum *N*-factor for all sections never reached the *N* = 6 design limit and were typically less than *N* = 4. Because the pressure-side boundary layer is completely subsonic, 3-D T-S should not be a concern.

Building upon the stability results from the principal test point, the different simulation groups can now be analyzed using the above results as a reference to judge how adjustments in the different parameters affect the overall stability characteristics.

Stability Results for Angle of Attack Simulations

Of all the flowfield characteristics for the angle of attack simulations, the feature that has the greatest influence on stability behavior is the chordwise pressure gradient. It was discussed in Section V-C how the suction-side and pressure-side pressure gradients changed as AoA increased as well as the implications for crossflow and T-S. These predictions were confirmed following the analysis of the LST calculations completed for the five angle of attack simulations. For suction-side crossflow, the predicted transition location moved further aft as AoA increased, as shown by *N*-factor contour plots for the AoA = 3.2°, 3.6°, and 4.0° (Test Points B, C, and E) simulations in Fig. 42. This was expected because a less accelerated pressure gradient stabilizes crossflow. Transition moved back an additional 10% *x/c* in the center of the test section for the *N* = 9 criterion through an increase from AoA = 3.2° to AoA = 3.6°. Even larger amounts of laminar flow were obtained through an increase to AoA = 4.0°, though it should be noted that the laminar flow region narrowed in the spanwise direction for higher AoAs. The SWLFC parameters did not experience much change from the principle test point, with the exception perhaps of the 4 mm wave becoming a more viable candidate for control than the 3.5 mm wave, again, at higher AoAs.

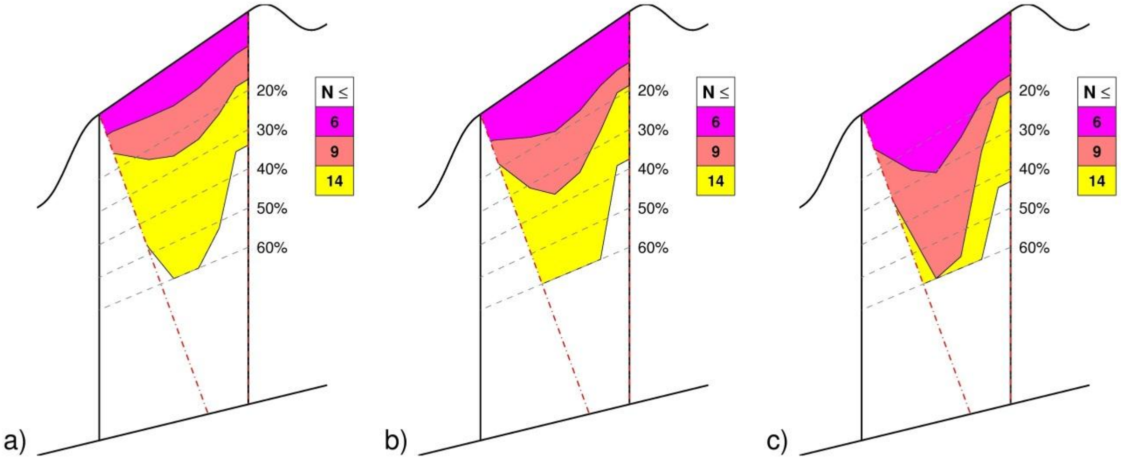


Fig. 42 Suction-side crossflow *N*-factor contours for the angle of attack simulations. a) 3.2°, b) 3.6°, and c) 4.0°.

The suction-side T-S became less stabilized with increasing aircraft AoA, also as expected. Even with this destabilization, the T-S *N*-factors typically remained well below the *N* = 6 limit for all sections at all AoAs with perhaps one exception: the BL204 section for the AoA = 4.0° simulation. Recall from Section V-C that an adverse pressure gradient was observed from 14% - 17% *x/c* on the suction side of the BL212 airfoil section. The BL204 section mirrors this C_p behavior which results in strong T-S growth, as shown in Fig. 43. While the *N* = 6 criterion was not exceeded, the nearness to the limit and rapid growth over such a short distance should be concerning, especially since T-S has typically been benign, as evidenced by other calculations. This reaffirms that all decelerating pressure gradients forward of 60% *x/c* should be avoided.

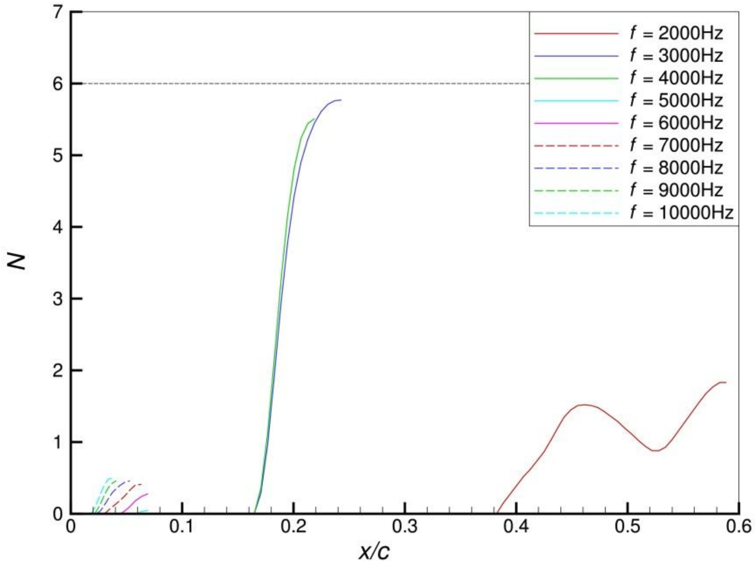


Fig. 43 Suction-side T-S *N*-factor results at BL204 for AoA = 4.0° (Test Point E).

The pressure side exhibited stability behavior is reverse of the suction side because its pressure gradient became more accelerated as AoA increased. Crossflow became less stable, moving transition forward, while T-S became more stable.

Stability Results for Mach Number Simulations

The flowfield results from Mach number simulations showed that the chordwise pressure gradient was significantly affected by off-design Mach numbers. Additionally, supercritical flow was present on the test section for all simulations except the *M* = 0.66 (Test Point F) simulation. The flowfield results have already indicated that flying at lower Mach numbers would not be possible, but stability calculations were nonetheless completed to see how the test section stability behavior was influenced.

Following the same analysis procedure described above, it was discovered that crossflow was stabilized considerably for the lower Mach number cases, no doubt due to the large region of a near-zero pressure gradient in the center of the glove. The SWLFC parameters given above for the principal test point were somewhat affected by a decrease in Mach number. The critical wavelength ranged from 6 - 8 mm and possible control wavelengths increased to 4.5 mm as Mach number decreased. However, the stabilization of crossflow makes it more difficult for the SWLFC technique to be implemented from a control standpoint because the difference in disturbance growth between the most unstable wave and the possible control wave has been diminished. Furthermore, any gains in laminar flow that could be made would be limited due to the expectation of turbulent flow beyond the suction-side pressure minimum (60% *x/c*). Fig. 44 shows how the predicted transition location for crossflow was affected by varying Mach number.

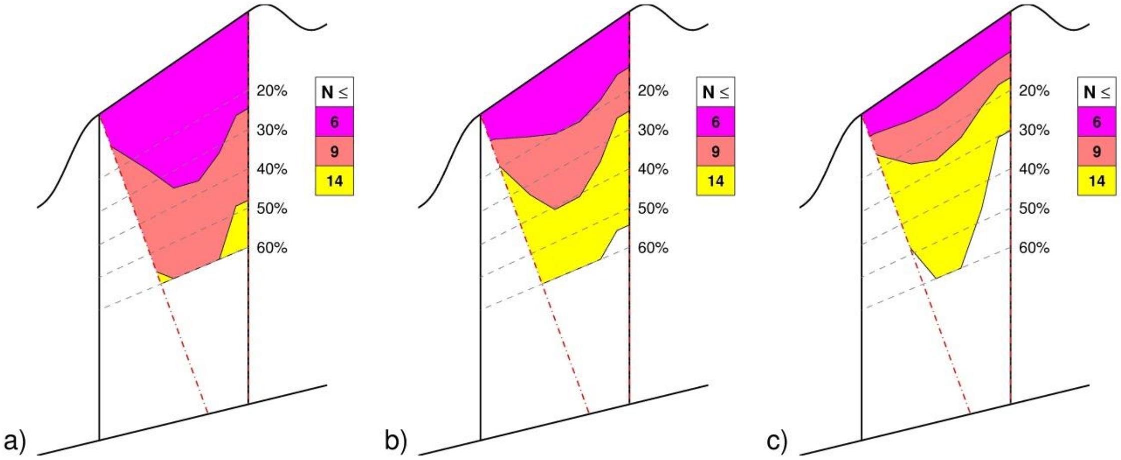


Fig. 44 Suction-side crossflow *N*-factor contours for the Mach number simulations. a) *M* = 0.66, b) *M* = 0.72, and c) *M* = 0.76.

For *M* = 0.66, nearly the entire test section within the domain of influence is laminar for the *N* = 9 crossflow criterion, leaving no room for improvement through the use of DREs. The same trends are present for the M = 0.72 (Test Point G) simulation results, though to a lesser extent. The M = 0.76 (Test Point H) stability behavior mimics Test Point A.

Compounding on the issues already touched on, the stabilization of crossflow comes at the cost of destabilized suction-side T-S. For all airfoil sections from the M = 0.66 simulation, the N = 6 limiting criterion was met for the f = 2000 Hz wave. The T-S behavior for the M = 0.72 simulation experienced slightly less growth, but it also flirts with or crosses the N = 6 limit.

The pressure-side chordwise pressure gradient experienced minimal change from the principal test point. The crossflow and T-S results subsequently remained constant with changing Mach number.

Stability Results for Reynolds Number Simulations

The flowfield results from the Reynolds number simulations given in Section V-C showed that the pressure distribution on the test section remained constant with varying the Reynolds number. The stability calculations instead would serve as the primary analysis tool for the comparison of these simulations. Transition location is strongly dependent on the Reynolds number, which is why chord Reynolds numbers are tied directly to the laminar flow requirements summarized in Table 7. The three Reynolds number simulations represented limiting cases for the NLF and SWLFC portions of the experiment. The $Re_c = 15\times10^6$ simulation (Test Point I) has direct implications on the NLF portion and its laminar flow requirement. The $Re_c = 22\times10^6$ simulation (Test Point A) serves as the lower bound to the SWLFC portion of the experimental flight envelope and of course has already been covered. The $Re_c = 30\times10^6$ simulation (Test Point J) serves as the upper bound to the SWLFC portion of the flight envelope and will be used to compare directly against the principal test point to better understand the sensitivity of expected transition for this Reynolds number range. The suction-side N-factor contour plots given in Fig. 45 show the predicted amounts of laminar flow for all three simulations.

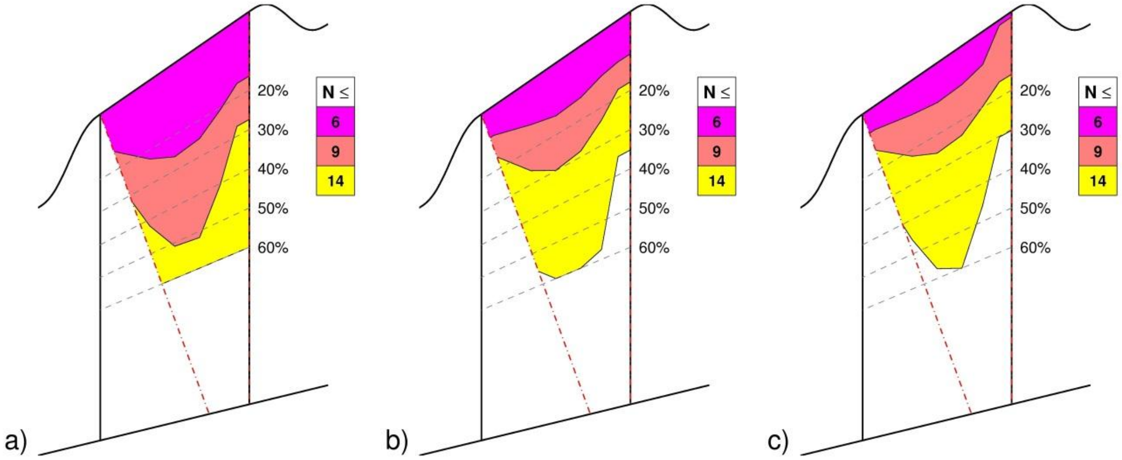


Fig. 45 Suction-side crossflow *N*-factor contours for the Reynolds number simulations. a) Re_c = 15×10^6, b) Re_c = 22×10^6, and c) Re_c = 30×10^6.

According to the LST calculations performed for the Re_c = 15×10^6 simulation, it appears as though the NLF requirement can be met for N = 14 criterion, but not for the N = 9 criterion. This indicates a polished leading edge will likely be necessary to meet the NLF requirement. The amounts of laminar flow obtained through the use of an operationally-realistic painted leading edge are still encouraging because transition occurs naturally at 50% x/c in the center of the test section.

For the SWLFC Reynolds number range, the behavior of note is the change in the amount of laminar flow for the Re_c = 22×10^6 and the Re_c = 30×10^6 simulations. The sensitivity is not nearly as strong as the change between the Re_c = 15×10^6 and the Re_c = 22×10^6 simulations. For the center portion of the test section, the transition front only moved forward a few percent *x/c* across the entire SWLFC Reynolds number range. Larger changes occurred near the test section boundaries, which effectively narrows the laminar flow region. Based on the discussion already given for Test Point A, the framework to meet the SWLFC requirement is in place and will depend on DRE performance. For the Re_c = 30×10^6 simulation, the SWLFC parameters changed from the principal test point. The critical wavelength and the possible control wavelength decreased to 5 mm and 3 mm respectively and the Branch I neutral point moved forward to 0.12% *x/c*. Any concerns that may have existed about the suction-side T-S for this simulation due to the increased Reynolds number were relieved by results that showed that T-S was still sufficiently suppressed.

The pressure-side crossflow stability results for the Reynolds number simulations exhibited trends similar to that of the suction side with regard to predicted transition location, moving forward with increasing Reynolds number. Pressure-side T-S again remained below the N = 6 criterion at all sections for all three simulations, though the 2000 Hz wave N-factors for the $Re_c = 30\times10^6$ simulation came close to exceeding this limit near the inboard edge of the test section.

Stability Results from 3-D T-S Studies

As mentioned previously, the presence of supercritical flow in regions of the test section make 3-D T-S waves more unstable than their standard 2-D counterpart. How much more unstable was the primary question these studies aimed to answer. 3-D T-S waves are characterized by nonzero spanwise wavenumber, leading to a propagation direction that is not parallel to the usual marching direction. Due to the effort required to complete a set of 3-D T-S calculations for a single airfoil section, it was simply not feasible to perform a comprehensive investigation that included all span locations for all ten simulations. Instead, select sections and simulations were chosen to better understand the sensitivity of 3-D T-S waves.

The first set of calculations was performed using the mean flow from the suction side of the center section (BL234) extracted from the principal test point simulation. The standard 2-D T-S results showed that for all wavelengths investigated, disturbance growth was minimal and was contained within the first 8% *x/c*. No T-S growth was predicted in the region of supercritical flow. Because 3-D T-S is expected only when the flow is supersonic, the LASTRAC calculations completed for this study were performed only over this region. Doing so prevents LASTRAC from finding a 3-D mode forward of the supercritical flow region. For this airfoil section, supercritical flow occurs aft of 19.3% *x/c*. To identify the scope of frequencies and spanwise wavenumbers that would experience growth, an iterative process was employed. Once these limits were found, the N-factor results for each frequency series across a range of spanwise wavenumbers were examined. For comparison purposes, the maximum N-factor for each mode was recorded and then plotted as N versus β for each frequency, as shown in Fig. 46.

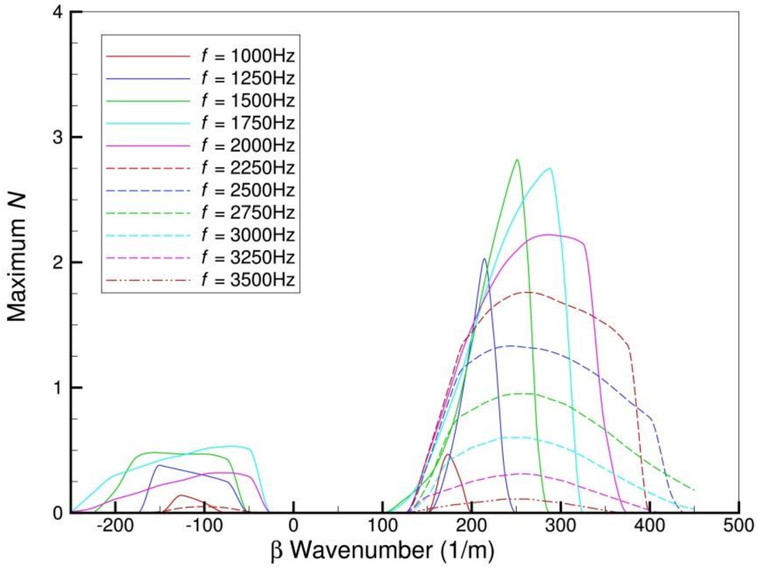


Fig. 46 Suction-side 3-D T-S maximum *N*-factor results at BL234 from Test Point A.

Several observations can be made from this plot, but the primary takeaway should be that maximum *N*-factor for all modes analyzed never reached $N = 3$. This means that while 3-D T-S does indeed experience larger growth than 2-D T-S, it is still well below the $N = 6$ criterion and as such should not cause transition. *N*-factor growth was observed for both negative and positive wavenumbers, while the 2-D T-S modes, which corresponds to $\beta = 0$ 1/m, experienced no growth. For all frequencies, the growth of the positive wavenumber portion was always larger than the negative wavenumber portion. The decay observed on the boundaries of the spanwise wavenumber range and the trends exhibited by the frequency series show that the study is well bounded.

Because the 2-D T-S growth for the BL234 airfoil section of the principal test point was practically negligible to begin with, a second study was completed. The suction side of the center section from the AoA = 3.8° simulation was chosen because it exhibited larger T-S growth and remained a likely scientific flight condition, unlike the AoA = 4.0° or lower Mach number simulations which also had elevated T-S growth. The 2-D T-S results from this airfoil section showed not only increased *N*-factors, but growth over a wider range of frequencies and growth further downstream from the attachment line. The 3-D T-S calculations were carried out using a procedure similar to the one described above, this time noting that the supercritical flow for the section began aft of 16.1% *x/c*. The maximum *N*-factor results were again plotted as *N* versus *β* for each frequency and are given by Fig. 47.

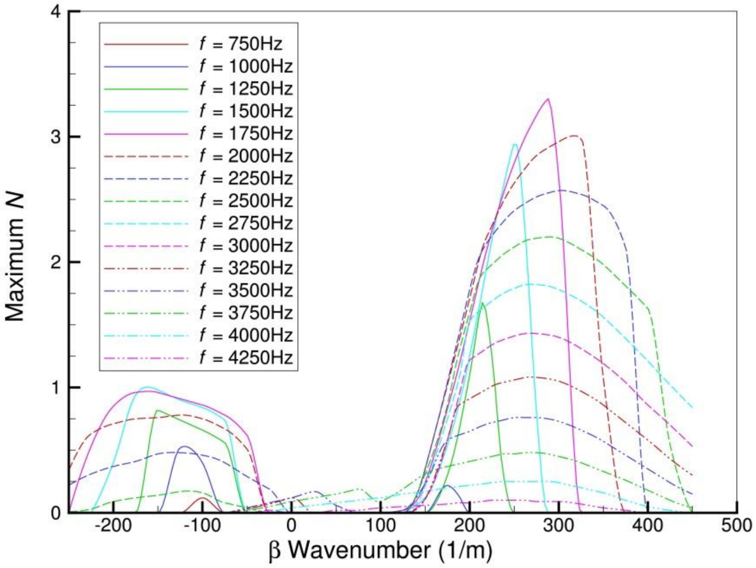


Fig. 47 Suction-side 3-D T-S maximum *N*-factor results at BL234 from Test Point D.

When compared to the Test Point A 3-D T-S study, the maximum *N*-factor results shown above echo the differences between the 2-D T-S behavior from the two studies. But even with larger growth, no 3-D T-S mode exceeded the $N = 6$ criterion. Many of the same trends exhibited earlier are also present in these results. This time, the demarcation point between the negative and positive wavenumber mode groups is clouded somewhat by modes found at and around $\beta = 0$ 1/m, but the existence of the two groups is still obvious. Also, some of the lowest frequencies, such as $f = 750$ Hz and $f =1000$ Hz, experience larger *N*-factors for negative rather than positive wavenumbers while, conversely, the other frequencies exhibit behavior consistent with the principal test point study.

Since these two studies really only scratch the surface of 3-D T-S stability characteristics, further exploration into this topic may be necessary in the future. However, the results presented here should help alleviate concerns that 3-D T-S disturbances in regions of supercritical flow may lead to early transition.

Traveling Crossflow Study Stability Results

Travelling crossflow occurs, as the name implies, when the crossflow disturbance wave propagates through space rather than remain in one location as stationary crossflow does. This is described mathematically with a nonzero frequency. As mentioned in Section I-B, travelling crossflow is typically dominant in high-turbulence environments, which is not expected in flight. Nonetheless, an understanding of traveling crossflow is important because of the implications its presence would have on this experiment. The SWLFC technique is first and foremost reliant on the existence of stationary crossflow and would be rendered useless if traveling crossflow dominated instead. Also, traveling crossflow modes are generally responsible for the largest possible disturbance growth, meaning that the natural transition front would likely move forward from its stationary crossflow location. Traveling crossflow calculations are similar to the 3-D T-S calculations in that the wave definition contains nonzero spanwise wavenumber and frequency. Because these are crossflow calculations, spanwise wavelength will be used in lieu of spanwise wavenumber to describe the wave.

Due to the numerous calculations required, performing multiple studies was again impractical. The suction side of the center airfoil section from the principal test point simulation was chosen to explore the sensitivity of the glove design to traveling crossflow. To provide a useful comparison, the wavelengths studied mirrored those used for the stationary crossflow calculations. A process very similar to the one used in the 3-D T-S studies was taken. Contrary to the 3-D T-S calculations, however, each wavelength was examined across a range of frequencies. Upon completion of all calculations, the maximum N-factors for all wavelength and frequency combinations were recorded and plotted as N versus f for each wavelength, as shown in Fig. 48.

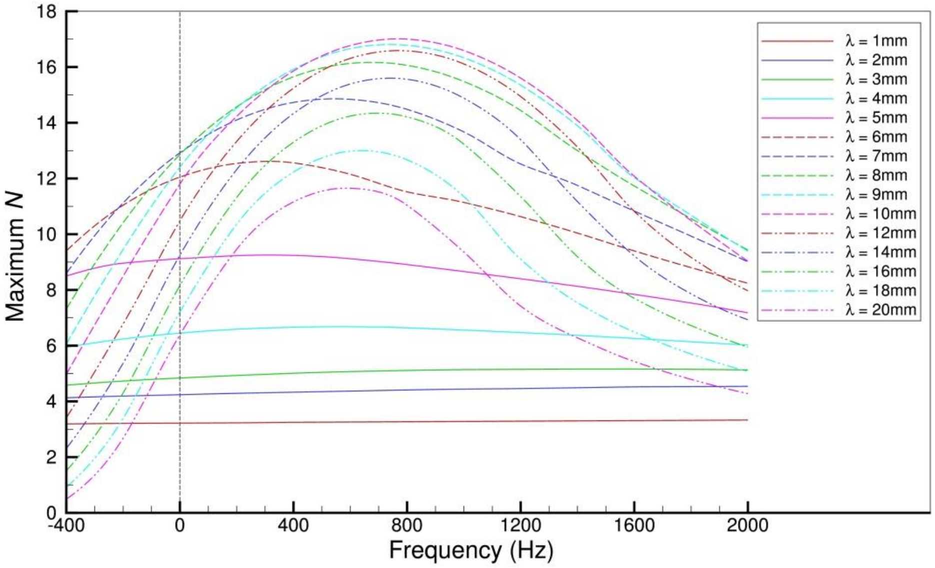


Fig. 48 Suction-side traveling crossflow maximum *N*-factor results at BL234 from Test Point A.

Several key points can be drawn from this plot. First, the largest *N*-factors for each wavelength occur at nonzero frequencies with the maximum *N*-factors of the larger wavelengths varying more over the frequency range than the smaller wavelengths. To better illustrate the differences between the maximum *N*-factors of the traveling and stationary crossflow modes, the stationary crossflow location is represented by the dashed gray line. Second, each wavelength series curve has a clear maximum with varying frequency. Third, but perhaps less obvious, the same behavior occurs for a constant frequency and varying wavelength. This confirms that the problem is well bounded. The differences in disturbance growth between the stationary crossflow and traveling crossflow of a particular wavelength can be pronounced and have a significant impact on predicted transition location. For example, *f* = 800 Hz represents the approximate *N*-factor maximums for wavelengths between 9 mm and 14 mm and likely serves as the worst-case scenario for laminar flow loss due to travelling crossflow. To illustrate how this compares to the stationary crossflow *N*-factors, Fig. 49 shows both *N*-factor plots.

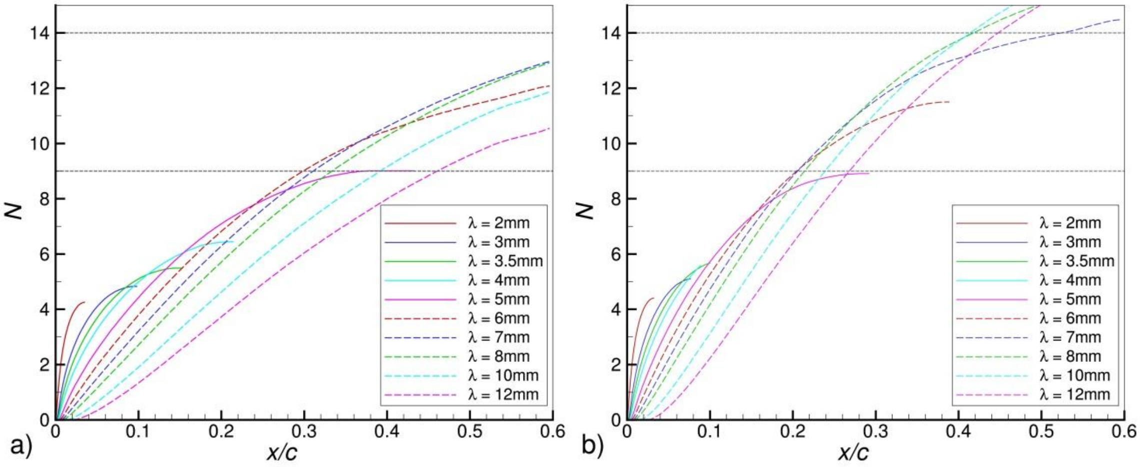


Fig. 49 Suction-side crossflow *N*-factor results from Test Point A at BL234. a) *f* = 0 Hz (stationary) and b) *f* = 800 Hz (traveling).

The *N*-factors for the *f* = 800 Hz calculation show larger growth than those for stationary crossflow. For the *N* = 9 transition criterion, the transition front moves from 29.9% *x/c* to 20.5% *x/c* – a loss of 9.5% *x/c*. The change for the *N* = 14 transition criterion is more pronounced. The stationary crossflow did not cross the *N* = 14 line, meaning transition would be expected at 60% *x/c*. The *f* = 800 Hz *N*-factors reach the *N* = 14 line at 41.4% *x/c* resulting in a loss in laminar flow of 18.6% *x/c*.

Currently, there is no indication that traveling crossflow will be the dominant transition mechanism for this experiment. However, should it be present during the science test flights, both laminar flow requirements would be placed in jeopardy because the SWLFC technique would no longer be applicable and because larger disturbance growth rates would likely occur, resulting in earlier transition. The *f* = 800 Hz example shows that the loss of laminar flow can be substantial. Until further research is completed on understanding the traveling crossflow mechanism in flight and at large Reynolds numbers, some level of uncertainty will exist if or how it will impact laminar flow control techniques.

VII. SUMMARY AND CONCLUSIONS

A. Summary

Obtaining laminar flow on large portions of an aircraft's wings and empennage could result in considerably less drag and subsequent fuel burn. As a result, the financial implications could also be substantial, especially for commercial aviation, based on the large percentage fuel represents in total operating costs. The SWLFC technique could provide significant gains in laminar flow, but its technology readiness level is still relatively low. While it has been shown to be effective at chord Reynolds numbers of roughly 8.0×10^6, this technique is still unproven at the larger Reynolds numbers characteristic of transport aircraft. The SARGE experiment will attempt to extend DRE technology to relevant Reynolds numbers. The work presented in this thesis focused on computationally evaluating the effectiveness of the SARGE wing glove design, both from aerodynamic and boundary-layer stability standpoints.

The first major step towards determining design effectiveness was obtaining and processing representative CAD models of both the aircraft and wing glove. Following a laser scan of the testbed aircraft, simplifications were made, such as detail removal around the aircraft engines, to better suit CFD purposes. Supplementary geometry was also created to assist with the grid generation process.

Once the CAD model adapted for CFD purposes was completed, it was used to generate a computational grid. A great deal of iteration was required to obtain a suitable product. The end result produced a grid that employed a nested structure containing both structured and unstructured cell zones that were specifically tailored to properly capture the flowfield. Particular attention was placed on the mesh on and around the glove test section, where proper refinement was critical in order to resolve the boundary layer. Upon completion of the baseline grid, several other grids were generated in order to check grid independence. The flowfield and stability results using these grids showed that grid independence was indeed achieved.

Using the baseline grid, a series of full-aircraft CFD calculations were carried out. The level of spanwise uniformity present in the test section pressure distribution, which is very important to the overall stability characteristics, was the first flowfield feature examined. The design discussed in this thesis showed clear improvements over the previous glove design. The CFD calculations completed were centered around the principal test point, which represents the “sweet spot” of the design. Sensitivity to angle of attack, Mach number, and Reynolds number was explored. In total, ten simulations were completed and analyzed. Important findings will be discussed below.

Boundary-layer stability calculations using flowfield data were then performed. The major obstacle for this step was extracting and processing boundary-layer data generated by the CFD simulations. A series of codes were written to perform these tasks, resulting in the creation of a mean flow file that would be used to perform stability calculations. Several span locations for each test point simulation were examined to determine both stationary crossflow and T-S disturbance characteristics across the test section. These efforts again centered around the principal test point, with other test points providing information about boundary-layer stability sensitivity with respect to varying conditions. Predicted transition location and SWLFC parameters were analyzed for all stability calculations performed. 3-D T-S and travelling crossflow were also briefly investigated. Major findings with respect to boundary-layer stability are also given below.

B. Conclusions

The purpose of this thesis was to computationally evaluate the SARGE wing glove design in its installed configuration and provide conclusions and recommendations based on aerodynamic and boundary-layer stability results. The work completed and presented above delivers valuable insight into the design effectiveness towards demonstrating the SWLFC technique at operationally relevant Reynolds numbers. It should be noted that in addition to the major conclusions specifically addressed in this section, myriad “lessons learned” can be found throughout the document. While many of these small discoveries may appear trivial due to their casual reference in text, they were often the product of significant trial and error and collectively played an enormous role in the overall analysis process. For convenience and consistency purposes, the major conclusions are separated into aerodynamic and boundary-layer stability groups.

Aerodynamic Conclusions

1) The spanwise uniformity of the test section pressure distribution for the current glove design shows marked improvements over the previous design and as such should result in more consistent stability behavior over the span of the test section. While the spanwise uniformity is most likely acceptable based on the stability results, there is still room for improvement.
2) The glove design under the principal test point conditions (AoA = 3.4°, M = 0.75, and Re_c = 22×10^6 at BL270) meets all the aerodynamic requirements outlined in Table 1 except the lift coefficient requirement. This is due to differences between the target C_p used for the glove OML optimization, which maintained C_L = 0.5, and the final pressure distribution over the entire test section. A simple solution to this problem would be to fly at a slightly higher aircraft AoA. A more involved solution would involve redesigning the target C_p to add margin to the C_L value while simultaneously improving spanwise uniformity of the test section pressure distribution to ensure all airfoil sections maintained C_L = 0.5.
3) The angle of attack simulations showed that the pressure gradient is a strong function of aircraft AoA, which is directly tied to test section C_L and stability behavior, and that the flowfield over the test section can be significantly influenced by flowfield effects generated elsewhere, specifically the region of high pressure near the inboard fairing leading edge. These results predict the limits for acceptable aircraft AoA during data collection to be 3.5° - 3.8°.
4) The Mach number simulations exposed large changes in the test section pressure distribution at off-design Mach numbers. The spirit of these simulations was to find a limiting Mach number that would avoid supercritical flow on the suction side of the test section. This was done because the presence of supersonic flow could possibly have a considerable effect on the boundary-layer stability behavior. While it was shown that obtaining completely subsonic flow over the test section is indeed possible, the undesirable modification of the pressure distribution would not meet the C_L requirement and likely undermine the laminar flow goals of the experiment. Off-design Mach numbers should therefore be avoided.

5) The Reynolds number simulations showed a nearly invariant test section pressure distribution over a chord Reynolds number range of 15×10^6 to 30×10^6. This behavior confirmed that when holding the other test point parameters constant, the stability characteristics of the test section would depend only on the Reynolds number and not be altered by changes in the test section pressure distribution.

Boundary-Layer Stability Conclusions

1) The principal test point exhibits consistent stability characteristics over much of the center of the glove test section. This not only validates many of the assumptions and techniques used in the stability calculation process, but will improve the experiment effectiveness because the parameters important to the SWLFC technique are also consistent. This behavior is not continued all the way to the test section boundaries, however, where flow effects detrimental to laminar flow are more significant due to the proximity of the glove fairings.
2) The principal test point stability results near the center of the test section have the following SWLFC parameters: a critical wavelength of 7 mm, a possible control wavelength of 3.5 mm for DRE spacing, and a DRE array location at 0.64% *x/c*.
3) The predicted transition fronts from the principal test point LST stationary crossflow calculations show that the SWLFC laminar flow requirement in Table 7 can be met for the $N = 9$ transition criterion, representing a operationally realistic painted leading edge. However, there would be little room for increases in laminar flow through the use of DREs for the $N = 14$ criterion based on its predicted transition front.
4) For the angle of attack simulations, aircraft AoA played a large role in the general stability behavior due to changes in the chordwise pressure gradient. Across the experimentally relevant AoA range of 3.5° to 3.8°, the predicted natural transition location was altered on both the pressure and suction sides of the test section, but the SWLFC control parameters for the suction side remained relatively unaffected.
5) The $Re_c = 15\times10^6$ simulation (Test Point I), which represented the lower bound of the NLF portion of the experiment, showed that the NLF laminar flow requirement in Table 7 could be met for the

N = 14 transition criterion based on the predicted transition front. Assuming transition at N = 9 would result in laminar flow back to 50% x/c in the center of the glove.

6) The $Re_c = 30\times10^6$ simulation (Test Point J), which represented the upper bound of the SWLFC portion of the experiment, showed that the SWLFC technique could still be employed with slightly adjusted SWLFC parameters. The predicted amounts of laminar flow experienced only slight decreases from the principal test point.

7) The 2-D T-S calculations for all experimentally relevant test points typically showed marginal growth. The N = 6 design limit imposed on the experiment was never reached. All of these test points experienced supercritical flow on the suction side of the test section, leading to the investigation of 3-D T-S in these regions. The N = 6 limit was not crossed for the 3-D T-S studies that were completed, though they showed that larger disturbance growth was possible.

Based on the conclusions listed above, all indications point to the current glove design fulfilling both the aerodynamic and laminar flow requirements. However, an opportunity exists to improve some aspects of this design. As of the writing of this thesis, the project has not moved beyond the preliminary design phase. Should the glove undergo a redesign, a process very similar to the steps described throughout this thesis would need to be repeated. While this would require a substantial commitment in both time and effort, the tools and methods already in place would streamline the process. Areas that could be addressed include:

1) Obtaining a more uniform spanwise pressure distribution on the test section through improved optimization.

2) Modifying the pressure distribution to increase the test section C_L and further tailor the stability characteristics using the experience gained from the current design analysis.

3) Increasing the aircraft AoA range for the science envelope either through OML changes during the glove optimization or through a pressure distribution redesign.

In addition to the future work that a glove redesign would entail, there were some aspects of the design analysis process that would be benefit from further exploration and some that were not completed that would add to the overall understanding of the design. For areas that were addressed in this thesis, the full-aircraft simulations could include test points with nonzero aircraft angle of side slip, which would represent more realistic flight conditions, and the 3-D T-S and travelling crossflow studies could be expanded beyond the few cases examined. For areas not explicitly addressed, the boundary-layer stability calculations could move beyond LST and include LPSE and NPSE calculations, which would provide a more accurate picture of the crossflow stability behavior, and stability behavior differences observed between mean flow data obtained through boundary-layer solvers and from Navier-Stokes solutions could be studied.

The SARGE experiment has the potential to take laminar flow technology a giant leap forward through the implementation of the SWLFC technique at larger Reynolds numbers. Based on the results generated from this design evaluation, meeting all experimental requirements is well within the realm of possibility. If successful, this experiment would move the industry one step closer towards realizing laminar flow for transonic aircraft.

REFERENCES

[1] Thibert, J. J., Reneaux, J., and Schmitt, R. V., "ONERA Activities on Drag Reduction," *Proceedings of the 17th Congress of the International Council of the Aeronautical Sciences*, Sept. 1990, pp. 1053–1059.

[2] Joslin, R. D., "Overview of Laminar Flow Control," NASA Langley Research Center, NASA/TP-1998-208705, Hampton, VA, Oct. 1998.

[3] Bureau of Transportation Statistics, "Airline Fuel Cost and Consumption (U.S. Carriers – Scheduled)," [online database], http://www.transtats.bts.gov/fuel.asp [retrieved 6 October 2011].

[4] Air Transport Association "ATA Quarterly Cost Index: U.S. Passenger Airlines," [online database], http://www.airlines.org/Economics/DataAnalysis/Pages/QuarterlyCostIndex.aspx [retrieved 6 October 2011].

[5] Reshotko, E., "Paths to Transition in Wall Layers," *Advances in Laminar Turbulent Transition Modeling*, VKI Lecture Series, von Karmen Institute. for Fluid Dynamics, Rhode St. Genèse, Belgium, June 2009, pp. 1-1 – 1-8.

[6] Saric, W. S., "Görtler Vortices," *Annual Review of Fluid Mechanics*, Vol. 26, Jan 1994, pp. 379-409.

[7] Pfenninger, W., "Laminar Flow Control – Laminarization," *Special Course on Drag Reduction*, AGARD Report 654, von Karman Institute, Rhode St. Genèse, Belgium, 1977, pp. 3-1 – 3-75.

[8] Reed, H. L., and Saric, W. S., "Transition Mechanisms for Transport Aircraft," AIAA Paper 2008-3743, June 2008.

[9] Gaster, M., "On the Flow Along Swept Leading Edges," *Aeronautical Quarterly*, Vol. 18, May 1967, pp. 165-184.

[10] Saric, W. S., and Reed, H. L., "Toward Practical Laminar Flow Control – Remaining Challenges," AIAA Paper 2004-2311, June 2004.

[11] Saric, W. S., Reed, H. L., and White, E. B., "Stability and Transition of Three-Dimensional Boundary Layers," *Annual Review of Fluid Mechanics*, Vol. 35, Jan. 2003, pp. 413-440.

[12] Bippes H., "Environmental Conditions and Transition Prediction in 3-D Boundary Layers," AIAA Paper 1997-1906, June 1997.

[13] Reibert, M. S., Saric, W. S., Carillo, R. B., and Chapman, K. L., "Experiments in Nonlinear Saturation of Stationary Crossflow Vortices in a Swept-Wing Boundary Layer," AIAA Paper 1996-0184, Jan. 1996.

[14] Haynes, T. S., and Reed, H. L., "Simulation of Swept-Wing Vortices Using Nonlinear Parabolized Stability Equations," *Journal of Fluid Mechanics*, Vol. 405, Feb. 2000, pp. 325-349.

[15] Radeztsky, R. H., Reibert, M. S., and Saric W. S., "Leading-Edge Roughness as a Transition Control Mechanism," *AIAA Journal*, Vol. 37, No. 11, Nov. 1999, pp. 1370-1377.

[16] Saric, W. S., and Reed, H. L., "Crossflow Instabilities – Theory and Technology," AIAA Paper No. 2003-0771, Jan. 2003.

[17] Saric, W. S., Carrillo, R. B., and Reibert, M. S., "Leading-Edge Roughness as a Transition Control Mechanism," AIAA Paper No. 1998-0781, Jan. 1998.

[18] Wasserman, P., and Kloker, M., "Mechanisms and Control of Crossflow-Vortex-Induced Transition in a 3-D Boundary Layer," *Journal of Fluid Mechanics*, Vol. 456, April 2002, pp. 49-84.

[19] Carpenter, A. L., Saric, W. S., and Reed, H. L., "In-Flight Receptivity Experiments on a 30-Degree Swept-Wing using Micron-Sized Discrete Roughness Elements," AIAA Paper 2009-590, Jan. 2009.

[20] Rhodes, R. G., "Computations of Laminar Flow Control on Swept Wings as a Companion to Flight Test Research," Masters Thesis, Aerospace Engineering Department, Texas A&M University, College Station, TX, Dec. 2008.

[21] Rhodes, R. G., Carpenter, A. L., Reed, H. L., and Saric, W. S., "CFD Analysis of Flight-Test Configuration for LFC on Swept Wings," AIAA Paper 2008-7336, Aug. 2008.

[22] Saric, W. S., Carpenter, A. L., and Reed, H. L., "Passive Control of Transition in Three-Dimensional Boundary Layers, with Emphasis on Discrete Roughness Elements," *Philosophical Transactions of the Royal Society A*, Vol. 369, April 2011, pp. 1352-1364.

[23] Neale, T. P., "CFD Investigations of a Transonic Swept-Wing Laminar Flow Control Flight Experiment," Masters Thesis, Aerospace Engineering Department, Texas A&M University, College Station, TX, May 2010.

[24] Belisle, M. J., Neale, T. P., Reed, H. L., and Saric, W. S., "Design of a Swept-Wing Laminar Flow Control Experiment for Transonic Aircraft," AIAA Paper 2010-4381, June 2010.

[25] HIS Jane's "Jane's Aircraft Upgrades," [online database], http://jau.janes.com/public/jau/index.shtml [retrieved August 19 2011].

[26] Belisle, M. J., Roberts, M. W., Tufts, M. W., Tucker, A. A., Williams, T. C., Saric, W. S., and Reed, H. L., "Design of the Subsonic Aircraft Roughness Glove Experiment," AIAA Paper 2011-3524, June 2011.

[27] SolidWorks, *Software Package*, 2011 x64 Edition, Dassault Systemes SolidWorks Corp., Concord, MA, 2011.

[28] ICEM CFD, *Software Package*, Version 13.0, ANSYS, Inc., Canonsburg, PA, 2011.

[29] Vassberg, J. C., Tinoco, E. N., Mani, M., Rider, B., Zickuhr, T., Levy, D. W., Brodersen, O. P., Eisfeld, B., Crippa, S., Wahls, R. A., Morrison, J. H., Mavriplis, D. J., Murayam, M., "Summary of the Fourth AIAA CFD Drag Prediction Workshop," AIAA Paper 2010-4547. June 2010.

[30] Oberkampf, W. L., Sindir, M. M., Conlisk, A. T., "Guide for the Verification and Validation of Computational Fluid Dynamics Simulations," AIAA G-077-1999, Jan. 1998.

[31] FLUENT, *Software Package*, Version 13.0, ANSYS, Inc., Canonsburg, PA, 2011.

[32] Menter, F. L., "Two-Equation Eddy-Viscosity Turbulence Models for Engineering Applications," *AIAA Journal*, Vol. 32, No. 8, Aug. 1994, pp. 1598-1605.

[33] Malik, M., Liao, W., Lee-Rausch, E., Li, F., Choudhari, M., Chang, C-L., "Computational Analysis of the G-III Laminar Flow Glove," AIAA Paper 2011-3525, June 2011.

[34] Hartshorn, F., Belisle, M. J., Reed, H. L., “Computational Optimization of a Natural Laminar Flow Experimental Wing Glove,” AIAA Paper 2012-870, Jan. 2012.

[35] LASTRAC, *Software Package*, Version 2.0, NASA Langley Research Center, Hampton, VA, 2009.

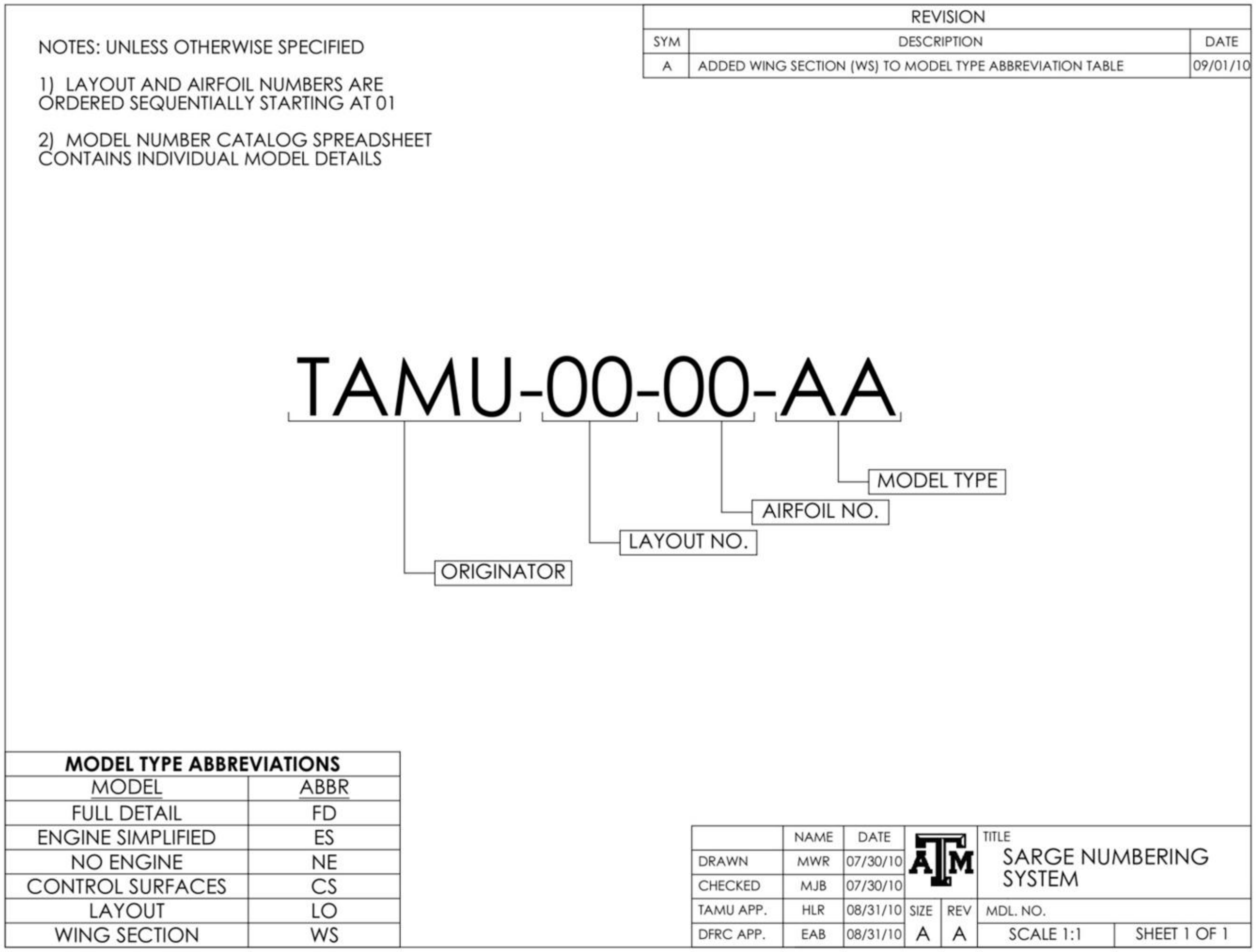
[36] MATLAB, *Software Package*, Version R2010b, MathWorks, Inc., Natick, MA, 2010.

[37] Smith, A., “Transition, Pressure Gradient, and Stability Theory,” *Proceedings of the IX International Congress of Applied Mechanics*, Vol. 4, pp. 234 –244, 1956.

[38] Van Ingen, J., “A Suggested Semi-Emperical Method for the Calculation of the Boundary-Layer Region,” Report VTH71, VTH74, Delft, Holland, 1956.

APPENDIX A

CAD DRAWINGS



NOTES: UNLESS OTHERWISE SPECIFIED

1) LAYOUT AND AIRFOIL NUMBERS ARE ORDERED SEQUENTIALLY STARTING AT 01

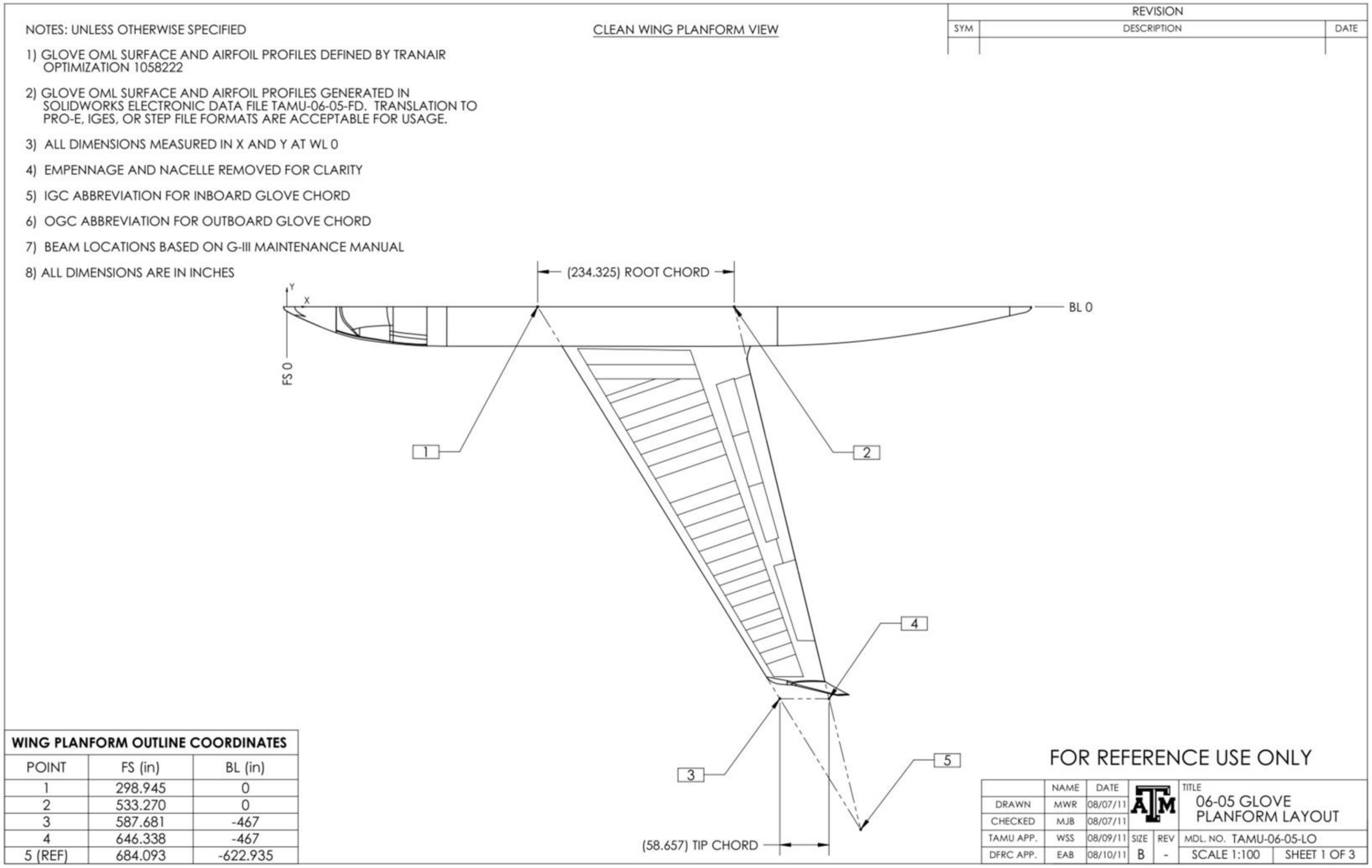
2) MODEL NUMBER CATALOG SPREADSHEET CONTAINS INDIVIDUAL MODEL DETAILS

REVISION		
SYM	DESCRIPTION	DATE
A	ADDED WING SECTION (WS) TO MODEL TYPE ABBREVIATION TABLE	09/01/10

MODEL TYPE ABBREVIATIONS	
MODEL	ABBR
FULL DETAIL	FD
ENGINE SIMPLIFIED	ES
NO ENGINE	NE
CONTROL SURFACES	CS
LAYOUT	LO
WING SECTION	WS

	NAME	DATE			TITLE
DRAWN	MWR	07/30/10			SARGE NUMBERING SYSTEM
CHECKED	MJB	07/30/10			
TAMU APP.	HLR	08/31/10	SIZE	REV	MDL. NO.
DFRC APP.	EAB	08/31/10	A	A	SCALE 1:1 SHEET 1 OF 1

Fig. A-1 SARGE Numbering System drawing.



NOTES: UNLESS OTHERWISE SPECIFIED

1) GLOVE OML SURFACE AND AIRFOIL PROFILES DEFINED BY TRANAIR OPTIMIZATION 1058222

2) GLOVE OML SURFACE AND AIRFOIL PROFILES GENERATED IN SOLIDWORKS ELECTRONIC DATA FILE TAMU-06-05-FD. TRANSLATION TO PRO-E, IGES, OR STEP FILE FORMATS ARE ACCEPTABLE FOR USAGE.

3) ALL DIMENSIONS MEASURED IN X AND Y AT WL 0

4) EMPENNAGE AND NACELLE REMOVED FOR CLARITY

5) IGC ABBREVIATION FOR INBOARD GLOVE CHORD

6) OGC ABBREVIATION FOR OUTBOARD GLOVE CHORD

7) BEAM LOCATIONS BASED ON G-III MAINTENANCE MANUAL

8) ALL DIMENSIONS ARE IN INCHES

REVISION		
SYM	DESCRIPTION	DATE

CLEAN WING PLANFORM VIEW

WING PLANFORM OUTLINE COORDINATES

POINT	FS (in)	BL (in)
1	298.945	0
2	533.270	0
3	587.681	-467
4	646.338	-467
5 (REF)	684.093	-622.935

FOR REFERENCE USE ONLY

	NAME	DATE			TITLE 06-05 GLOVE PLANFORM LAYOUT	
DRAWN	MWR	08/07/11				
CHECKED	MJB	08/07/11				
TAMU APP.	WSS	08/09/11	SIZE	REV	MDL. NO. TAMU-06-05-LO	
DFRC APP.	EAB	08/10/11	B	-	SCALE 1:100	SHEET 1 OF 3

Fig. A-2 Sheet 1 of the 06-05 Glove Planform Layout drawing: clean wing planform view.

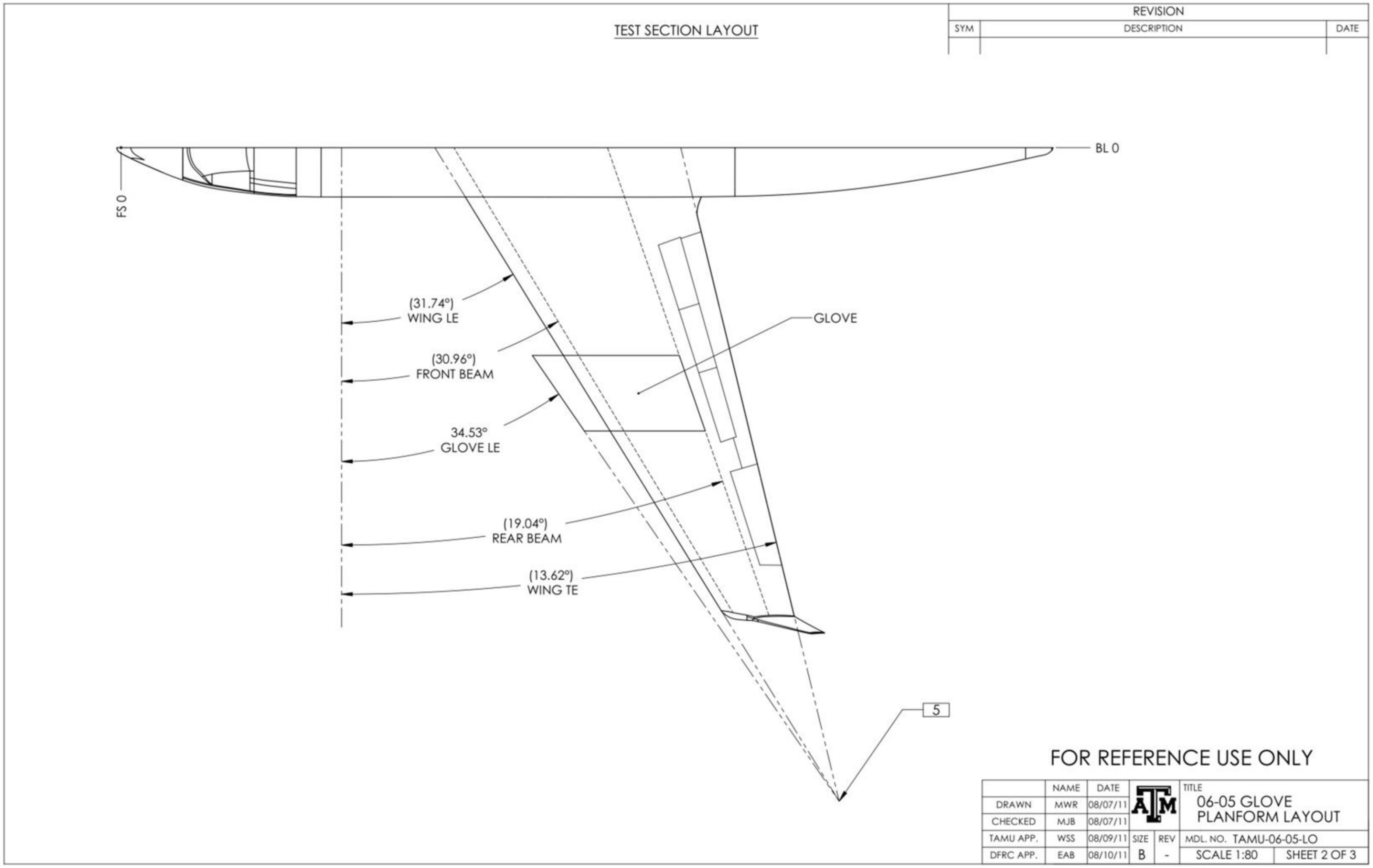
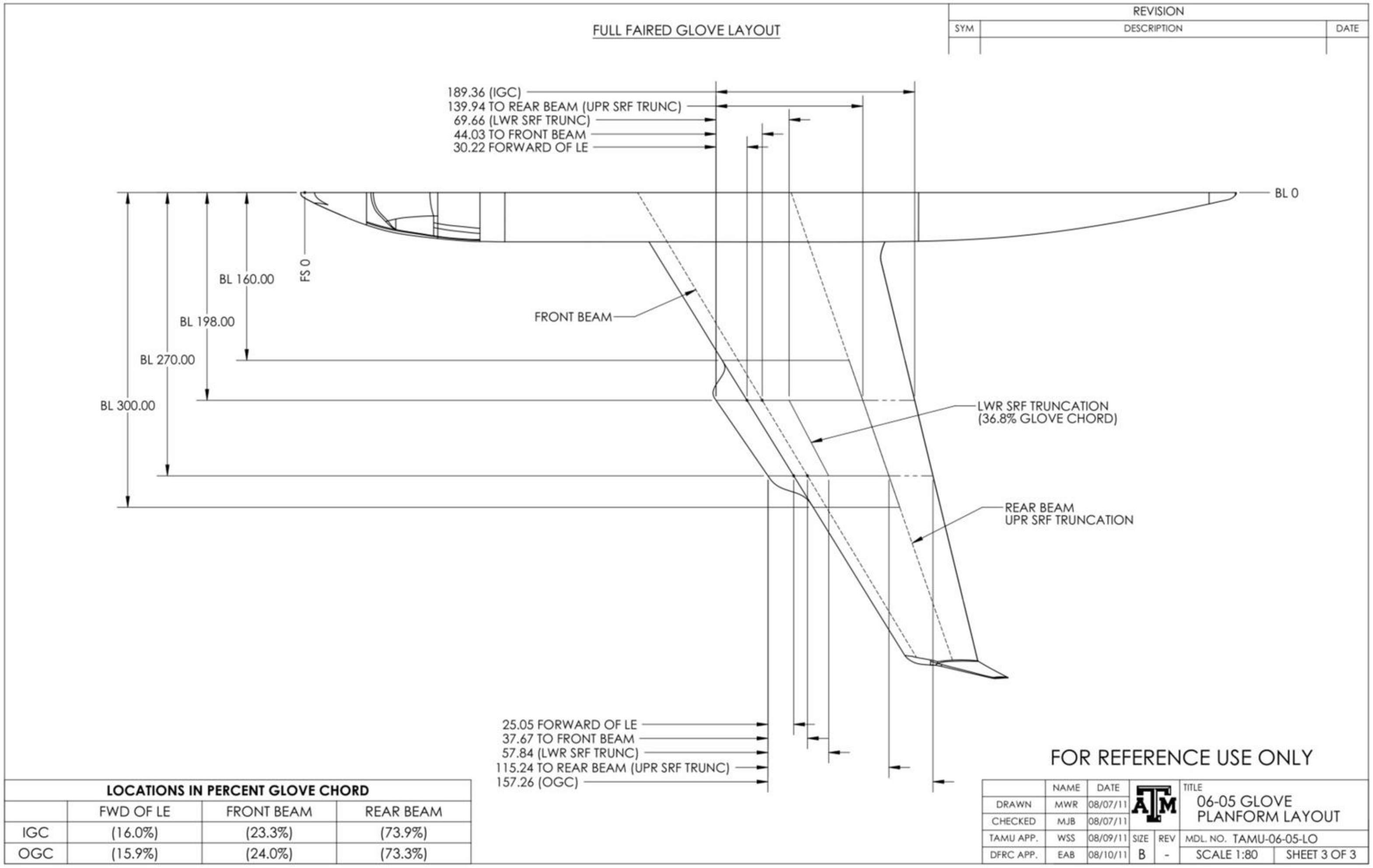


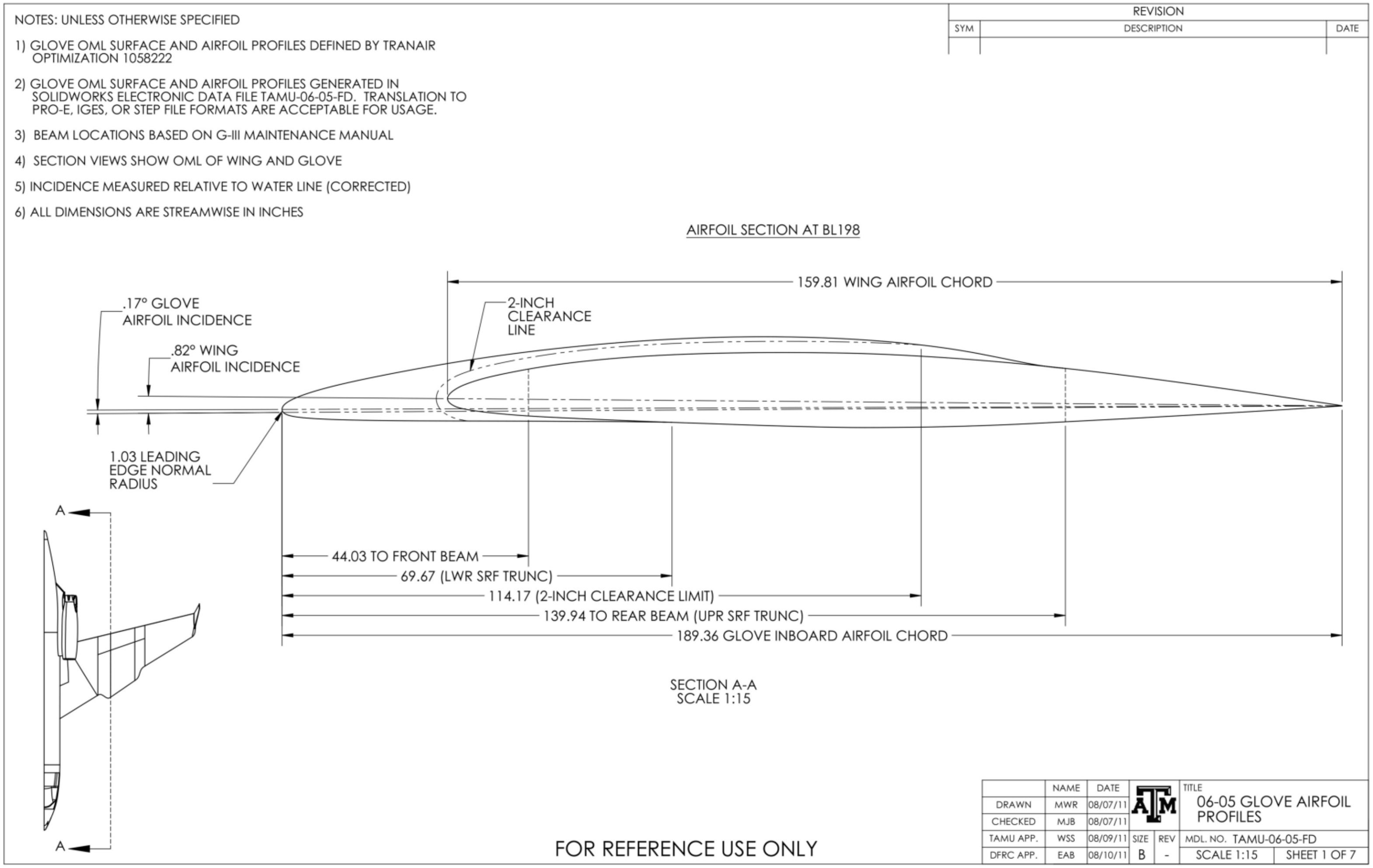
Fig. A-3 Sheet 2 of the 06-05 Glove Planform Layout drawing: test section layout.



LOCATIONS IN PERCENT GLOVE CHORD

	FWD OF LE	FRONT BEAM	REAR BEAM
IGC	(16.0%)	(23.3%)	(73.9%)
OGC	(15.9%)	(24.0%)	(73.3%)

Fig. A-4 Sheet 3 of the 06-05 Glove Planform Layout drawing: full faired glove layout.



NOTES: UNLESS OTHERWISE SPECIFIED

1) GLOVE OML SURFACE AND AIRFOIL PROFILES DEFINED BY TRANAIR OPTIMIZATION 1058222

2) GLOVE OML SURFACE AND AIRFOIL PROFILES GENERATED IN SOLIDWORKS ELECTRONIC DATA FILE TAMU-06-05-FD. TRANSLATION TO PRO-E, IGES, OR STEP FILE FORMATS ARE ACCEPTABLE FOR USAGE.

3) BEAM LOCATIONS BASED ON G-III MAINTENANCE MANUAL

4) SECTION VIEWS SHOW OML OF WING AND GLOVE

5) INCIDENCE MEASURED RELATIVE TO WATER LINE (CORRECTED)

6) ALL DIMENSIONS ARE STREAMWISE IN INCHES

REVISION		
SYM	DESCRIPTION	DATE

AIRFOIL SECTION AT BL198
159.81 WING AIRFOIL CHORD
.17° GLOVE AIRFOIL INCIDENCE
2-INCH CLEARANCE LINE
.82° WING AIRFOIL INCIDENCE
1.03 LEADING EDGE NORMAL RADIUS
A
A
44.03 TO FRONT BEAM
69.67 (LWR SRF TRUNC)
114.17 (2-INCH CLEARANCE LIMIT)
139.94 TO REAR BEAM (UPR SRF TRUNC)
189.36 GLOVE INBOARD AIRFOIL CHORD
SECTION A-A
SCALE 1:15
FOR REFERENCE USE ONLY

	NAME	DATE			TITLE
DRAWN	MWR	08/07/11			06-05 GLOVE AIRFOIL PROFILES
CHECKED	MJB	08/07/11			
TAMU APP.	WSS	08/09/11	SIZE	REV	MDL. NO. TAMU-06-05-FD
DFRC APP.	EAB	08/10/11	B	-	SCALE 1:15 SHEET 1 OF 7

Fig. A-5 Sheet 1 of the 06-05 Glove Airfoil Profiles drawing: airfoil section at BL198.

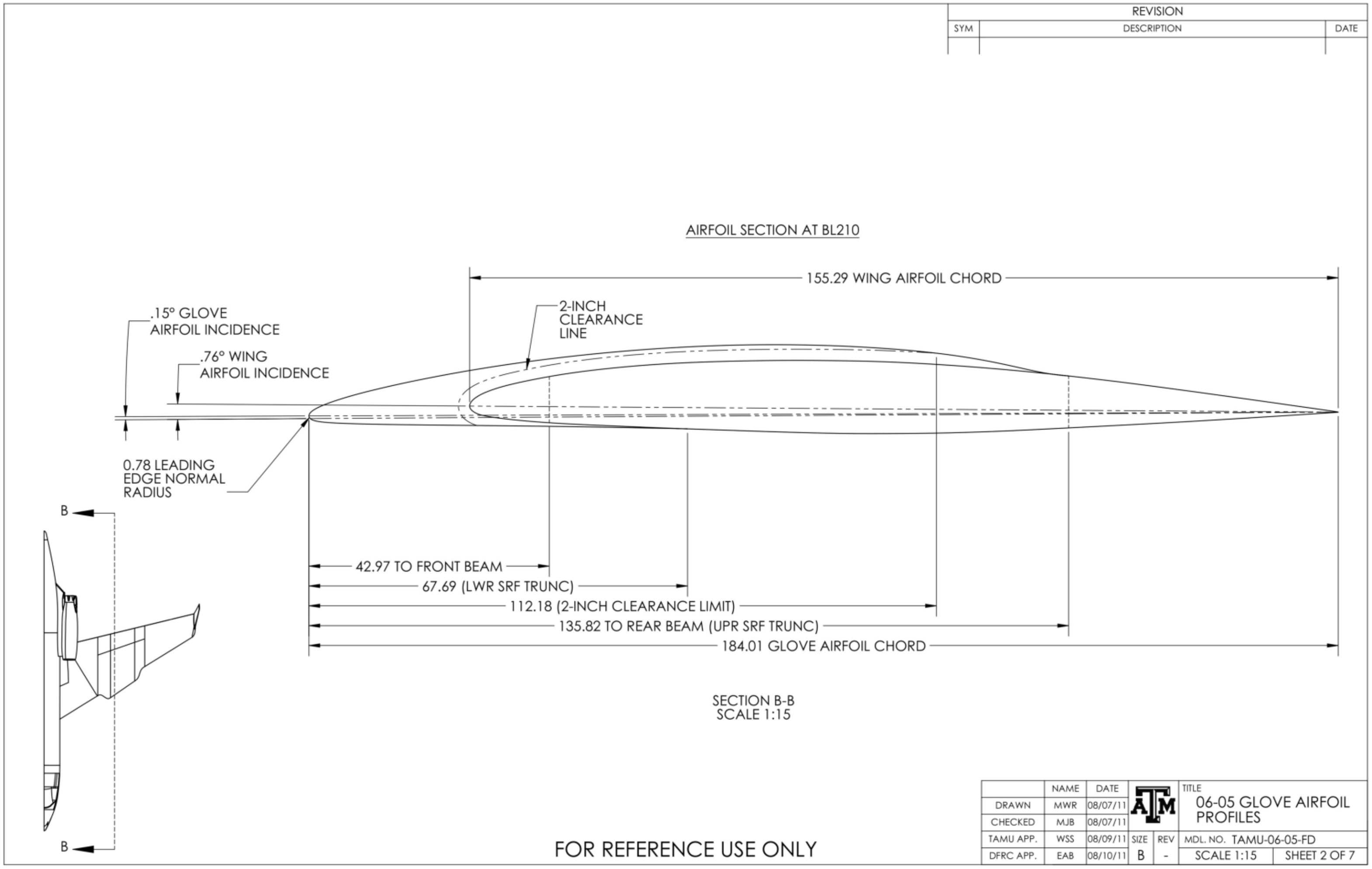


Fig. A-6 Sheet 2 of the 06-05 Glove Airfoil Profiles drawing: airfoil section at BL210.

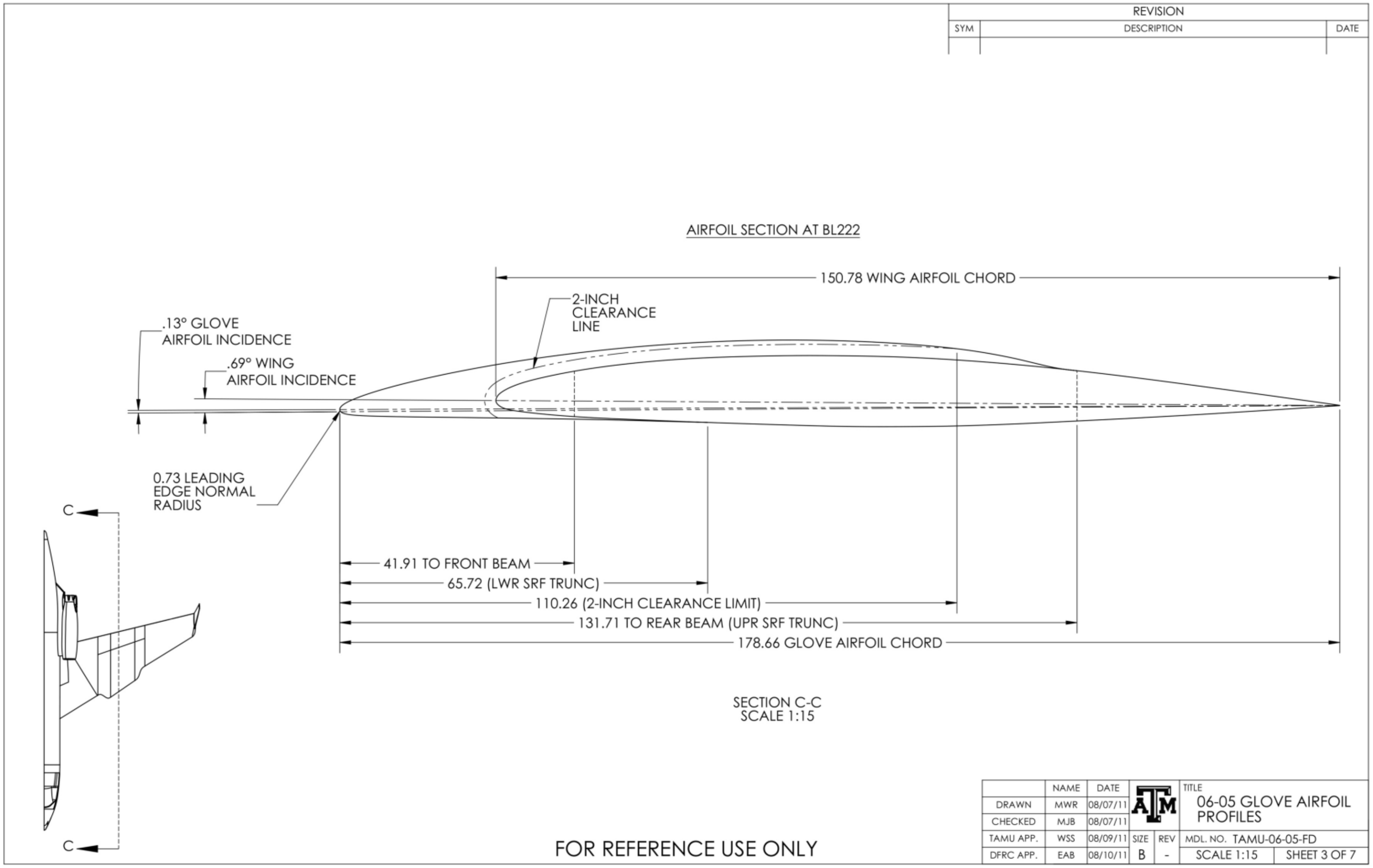


Fig. A-7 Sheet 3 of the 06-05 Glove Airfoil Profiles drawing: airfoil section at BL222.

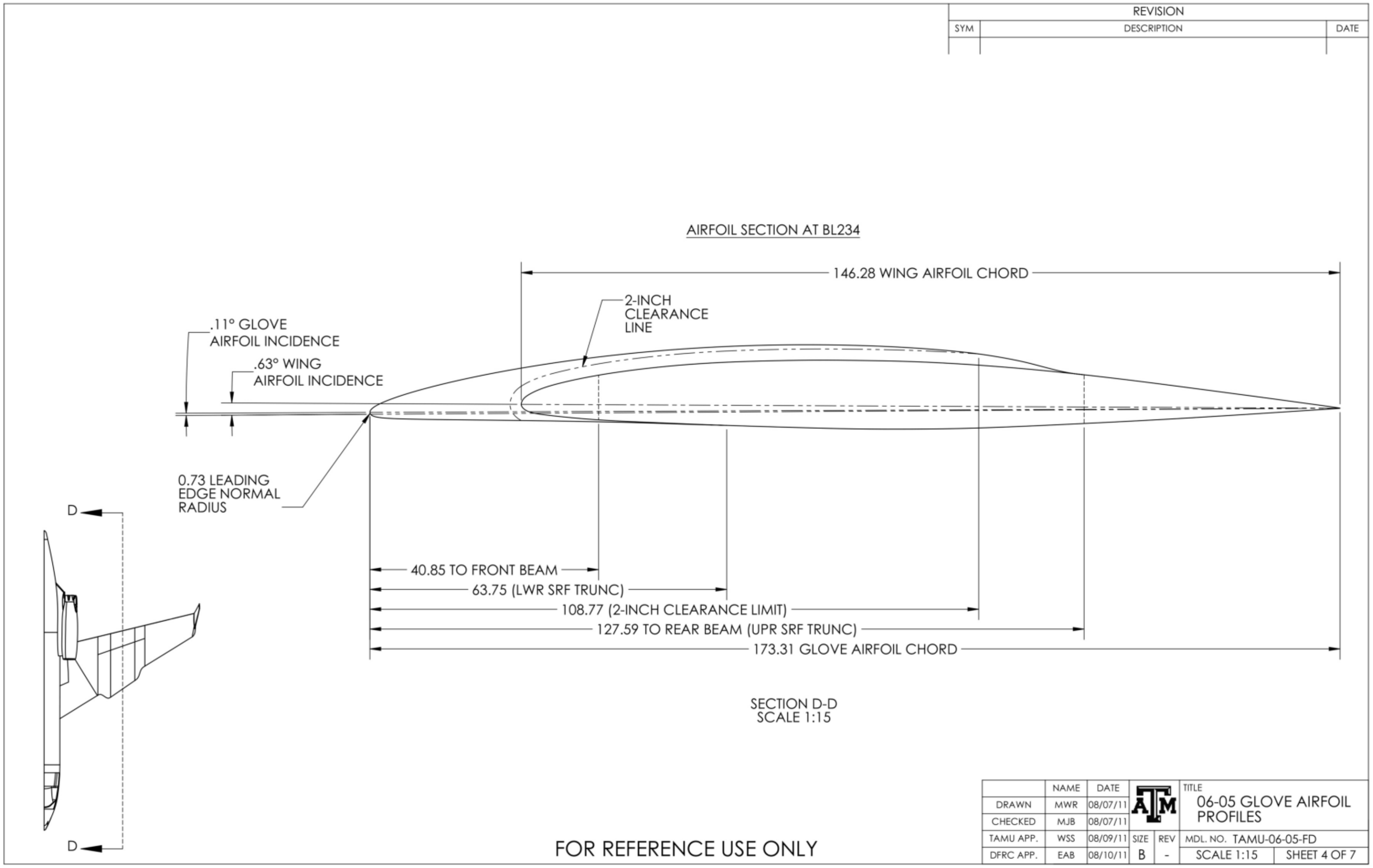


Fig. A-8 Sheet 4 of the 06-05 Glove Airfoil Profiles drawing: airfoil section at BL234.

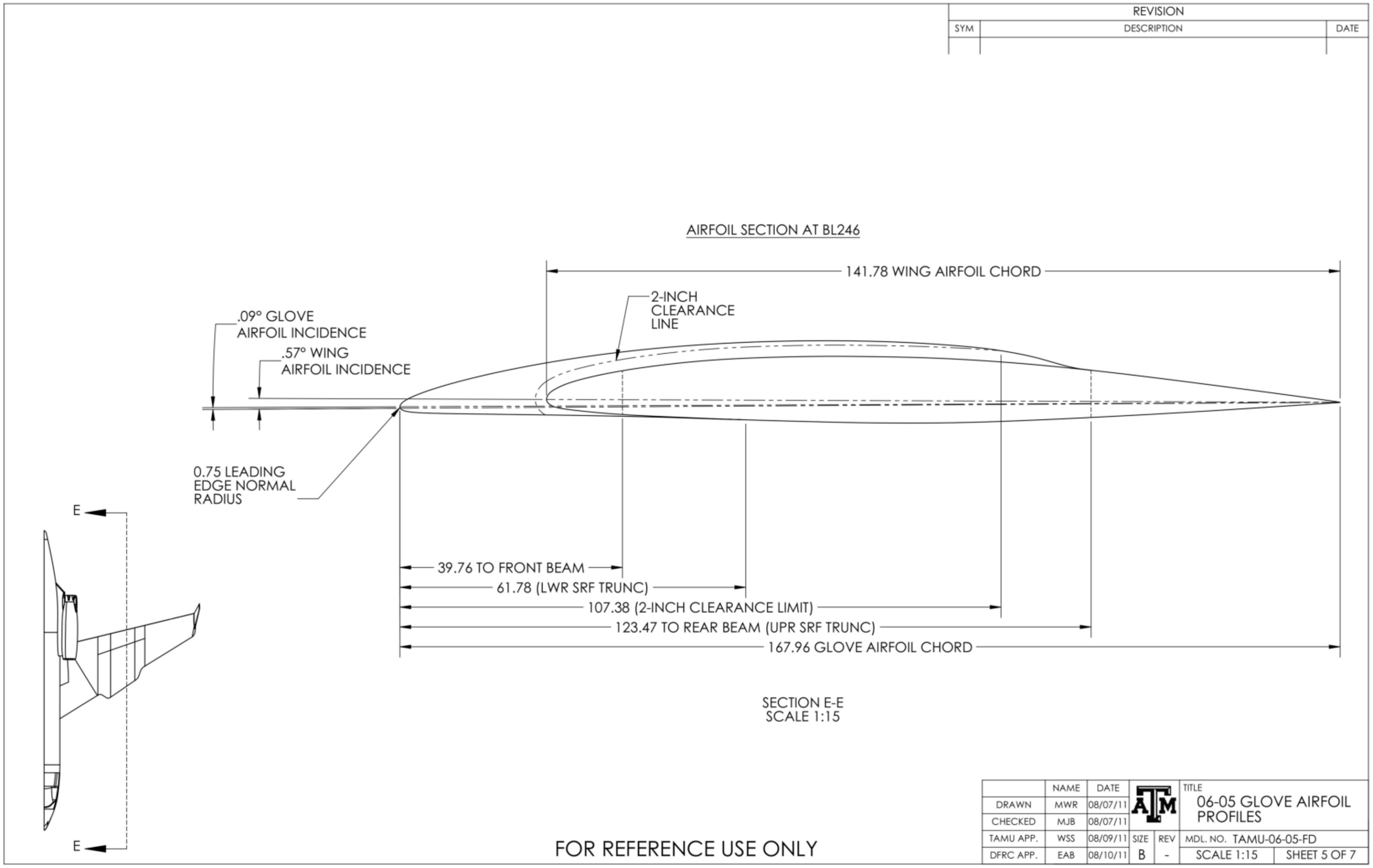


Fig. A-9 Sheet 5 of the 06-05 Glove Airfoil Profiles drawing: airfoil section at BL246.

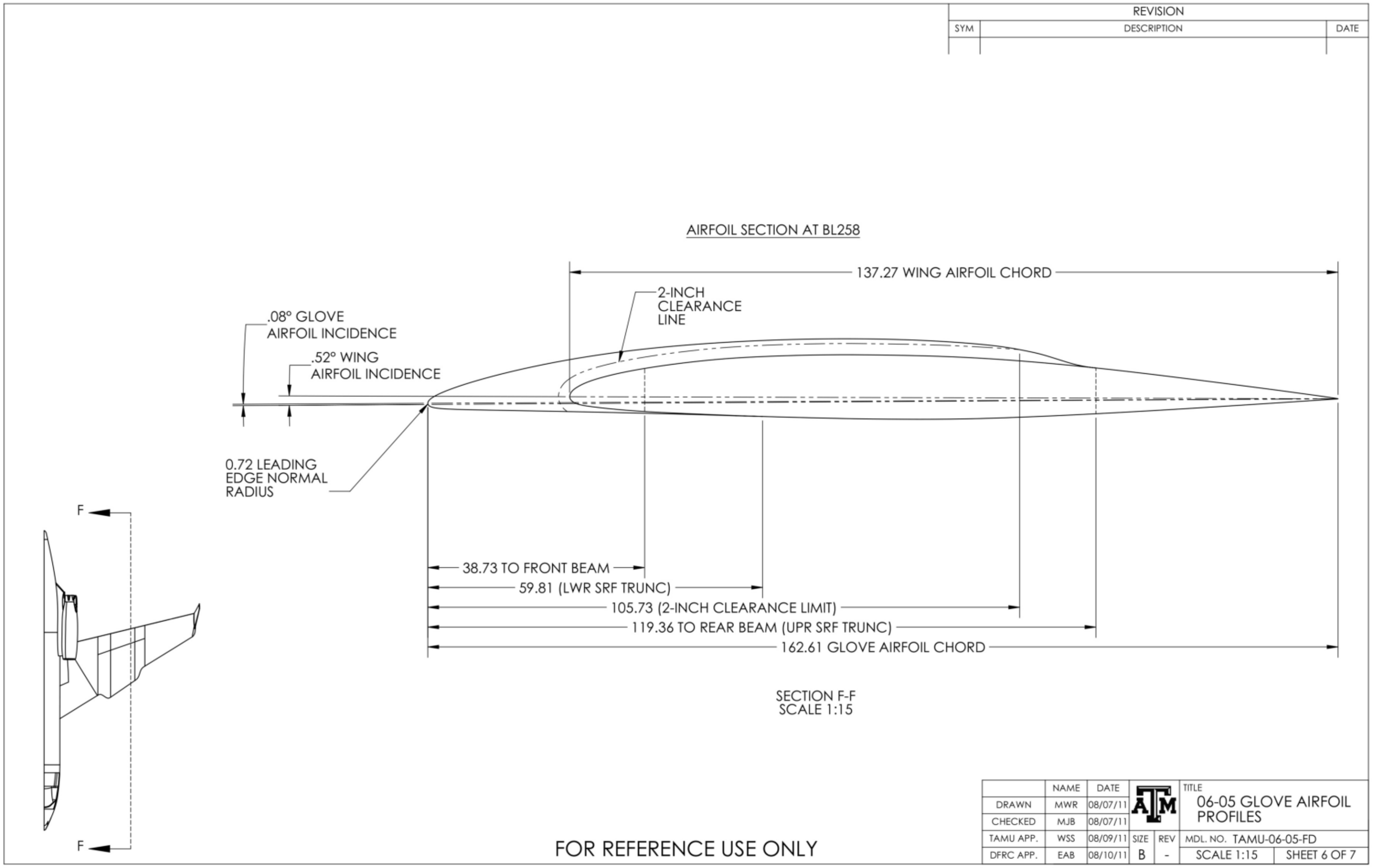


Fig. A-10 Sheet 6 of the 06-05 Glove Airfoil Profiles drawing: airfoil section at BL258.

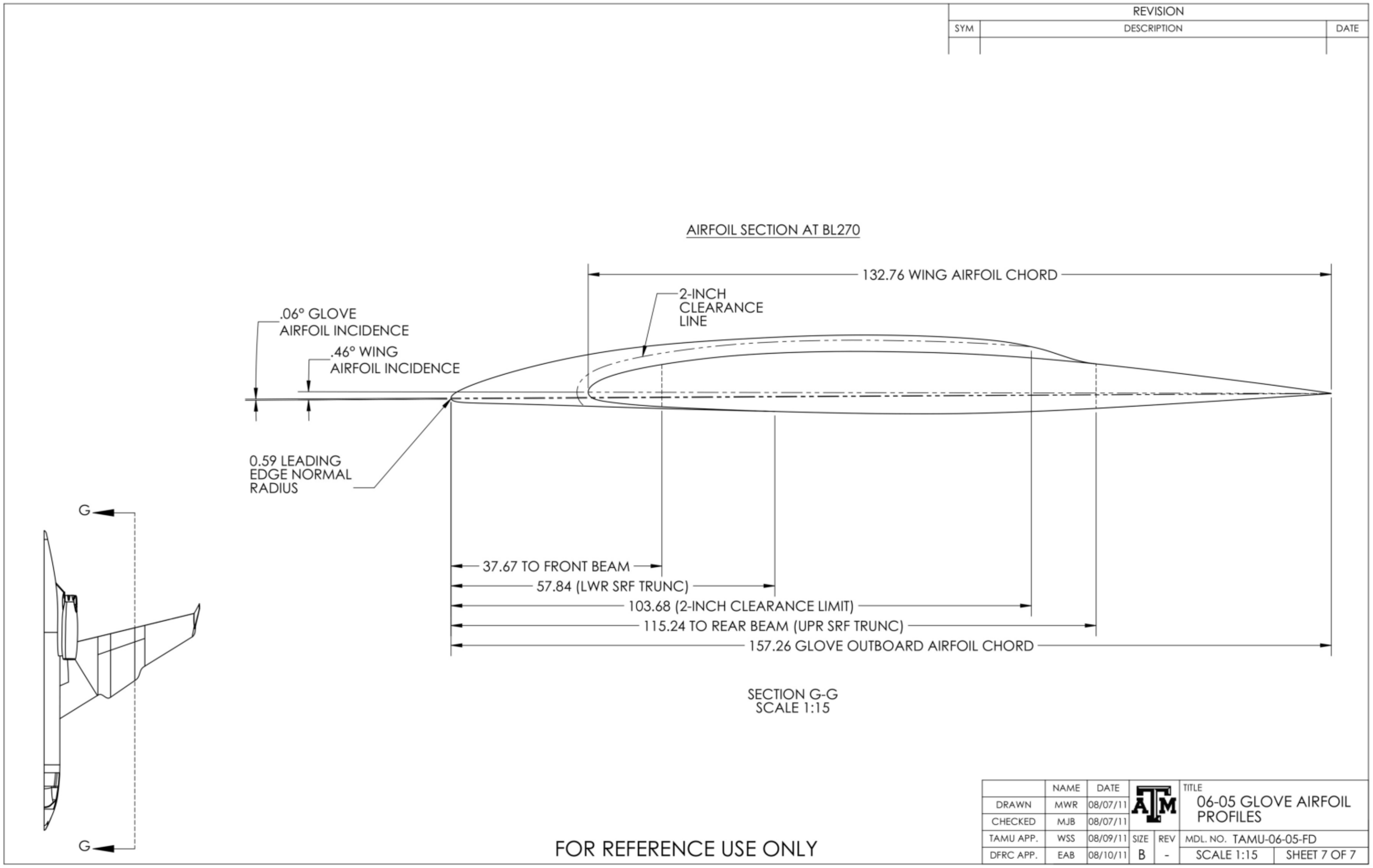


Fig. A-11 Sheet 7 of the 06-05 Glove Airfoil Profiles drawing: airfoil section at BL270.

APPENDIX B

CFD SIMULATION BOUNDARY CONDITIONS

Table B-1 Boundary conditions for the angle of attack simulations.

Test Point	B	A	C	D	E
M	0.75	0.75	0.75	0.75	0.75
AoA	3.2°	3.4°	3.6°	3.8°	4.0°
Re_c	22×10^{6} at BL270	22×10^{6} at BL270	22×10^{6} at BL270	22×10^{6} at BL270	22×10^{6} at BL270
H (ft)	36,730	36,730	36,730	36,730	36,730
Farfield Boundary Conditions					
M	0.75	0.75	0.75	0.75	0.75
AoA	3.2°	3.4°	3.6°	3.8°	4.0°
p (Pa)	22,014	22,014	22,014	22,014	22,014
T (K)	216.65	216.65	216.65	216.65	216.65
Engine Inlet Boundary Conditions					
p_{static} (Pa)	27,539	27,539	27,539	27,539	27,539
T_{total} (K)	231.03	231.03	231.03	231.03	231.03
MFR (kg/s)	28.118	28.118	28.118	28.118	28.118
Engine Outlet Boundary Conditions					
p_{total} (Pa)	68,421	68,421	68,421	68,421	68,421
p_{static} (Pa)	36,592	36,592	36,592	36,592	36,592
T_{total} (K)	522.66	522.66	522.66	522.66	522.66

Table B-2 Boundary conditions for the Mach number simulations.

Test Point	**F**	**G**	**A**	**H**
M	0.66	0.72	0.75	0.76
AoA	3.4°	3.4°	3.4°	3.4°
Re_c	22×10^6 at BL270	22×10^6 at BL270	22×10^6 at BL270	22×10^6 at BL270
H **(ft)**	33,320	35,790	36,730	37,010
Farfield Boundary Conditions				
M	0.66	0.72	0.75	0.76
AoA	3.4°	3.4°	3.4°	3.4°
p **(Pa)**	25,876	23,027	22,014	21,721
T **(K)**	222.24	217.36	216.65	216.65
Engine Inlet Boundary Conditions				
p_{static} **(Pa)**	31,075	28,565	27,539	27,545
T_{total} **(K)**	241.65	239.96	231.03	241.74
MFR (kg/s)	26.902	27.152	28.118	27.672
Engine Outlet Boundary Conditions				
p_{total} **(Pa)**	62,021	64,221	68,421	69,799
p_{static} **(Pa)**	33,151	34,336	36,592	35,911
T_{total} **(K)**	472.47	496.48	522.66	519.59

Table B-3 Boundary conditions for the Reynolds number simulations.

Test Point	I	A	J
M	0.75	0.75	0.75
AoA	3.4°	3.4°	3.4°
Re_c	15×10^6 at BL270	22×10^6 at BL270	30×10^6 at BL198
H (ft)	44,740	36,730	33,420
Farfield Boundary Conditions			
M	0.75	0.75	0.75
AoA	3.4°	3.4°	3.4°
p (Pa)	15,002	22,014	25,755
T (K)	216.65	216.65	222.04
Engine Inlet Boundary Conditions			
p_{static} (Pa)	18,757	27,539	32,834
T_{total} (K)	241.08	231.03	247.08
MFR (kg/s)	19.187	28.118	30.836
Engine Outlet Boundary Conditions			
p_{total} (Pa)	46,694	68,421	74,176
p_{static} (Pa)	24,973	36,592	39,665
T_{total} (K)	523.00	522.66	512.74

APPENDIX C

SIMULATION JOURNAL AND BATCH FILES

C-1: Case Setup Journal File

The case setup journal file is used to automate the simulation setup process by carrying out the given FLUENT commands through the TUI. An example is given below.

```
;Read in mesh
/file/read-case
"TAMU0605-GRID06.msh"

;Set solver
/define/models/solver/density-based-implicit
yes

;Set mesh scale
/mesh/scale
.0254
.0254
.0254

;Turn energy equation on
/define/models/energy
yes

;Set model to viscous k-omega SST
/define/models/viscous/kw-sst
yes

;Set air material properties
/define/materials/change-create
air
air
yes
ideal-gas
no
no
yes
sutherland
three-coefficient-method
1.716e-05
273.11
110.56
no
no
no

;Set lam to laminar zone
/define/boundary-conditions/fluid
lam
no
no
no
no
no
0
no
0
```

```
no
0
no
0
no
0
no
1
no
no
yes
yes
no

;Set operating pressure
/define/operating-conditions/operating-pressure
0

;Set farfield boundary conditions
/define/boundary-conditions/pressure-far-field
ff
no
22014     ;static pressure
no
.75       ;Mach number
no
216.65    ;temperature
yes
no
0         ;x-component
no
.065577287   ;y-component
no
-.997847493  ;z-component
no
no
yes
.1
1

;Set engine inlet boundary conditions (MFR turned off)
/define/boundary-conditions/pressure-outlet
eng/eng_in
no
27538.96     ;static pressure
no
231.03    ;total temperature
no
yes
no
no
yes
.1
1
no
no
no

;Set engine outlet boundary conditions
/define/boundary-conditions/pressure-inlet
```

```
eng/eng_out
yes
no
68420.68     ;total pressure
no
36592.36     ;static pressure
no
522.66    ;total temperature
no
yes
no
no
yes
.1
1

;Create mesh interface
/define/mesh-interfaces/create
struct
struct_in
()
struct_out
()
no
no

;Set reference values
/report/reference-values/compute/pressure-far-field
ff
/report/reference-values/area
43.4139
/report/reference-values/length
4.2012
/report/reference-values/viscosity
1.4216e-5



;Set solution gradient scheme
/solve/set/gradient-scheme
yes

;Set discretization schemes (All first order)
/solve/set/discretization-scheme/amg-c
0
/solve/set/discretization-scheme/k
0
/solve/set/discretization-scheme/omega
0

;Set initial Courant number
/solve/set/courant-number
1

;Set residual monitor
/solve/monitors/residual/criterion-type
3
/solve/monitors/residual/n-display
10000
/solve/monitors/residual/n-save
```

```
10000

;Set drag monitors
/solve/monitors/force/drag-coefficient
yes
*
()
yes
no
no
no
0
.065577287
-.997847493

;Set lift monitor
/solve/monitors/force/lift-coefficient
yes
*
()
yes
no
no
no
0
.997847493
-.065577287

;Set engine inlet monitor
/solve/monitors/surface/set-monitor
eng_in
"Mass Flow Rate"
eng/eng_in
()
no
yes
no
1

;Set engine outlet monitor
/solve/monitors/surface/set-monitor
eng_out
"Mass Flow Rate"
eng/eng_out
()
no
yes
no
1

;Reorder zones and domain
/mesh/reorder/reorder-zones
/mesh/reorder/reorder-domain

;Smooth and swap mesh
/mesh/smooth-mesh
"quality based"
4
.1
```

```
;Repair wall distance
/solve/initialize/repair-wall-distance

;Check mesh
/mesh/check

;Initialize the flow
/solve/initialize/compute-defaults/pressure-far-field
ff
/solve/initialize/initialize-flow

;Create iso-surface at BL212
/surface/iso-surface
x-coordinate
bl212
glve_lam
glve_turb
()
()
5.3848
()

;Create iso-surface at BL234
/surface/iso-surface
x-coordinate
bl234
glve_lam
glve_turb
()
()
5.9436
()

;Create iso-surface at BL264
/surface/iso-surface
x-coordinate
bl264
glve_lam
glve_turb
()
()
6.7056
()

;Get mesh size, get mesh memory, list zones, list surfaces
mesh/size-info
mesh/memory-usage
mesh/modify-zones/list-zones
surface/list-surfaces

;Save case and data files (gunzipped)
/file/write-case-data
"TAMU0605-07R.cas.gz"

exit
yes
```

C-2: Simulation Batch File

The simulation batch file is used by the Eos supercomputing cluster to carry out a FLUENT simulation. An example is given below.

```
#PBS -l nodes=8:ppn=4,walltime=48:00:00,mem=160gb
#PBS -W x=NACCESSPOLICY:SINGLEJOB
#PBS -N output
#PBS -S /bin/bash
#PBS -j oe
#PBS -M robertsm@tamu.edu

# Initialize Fluent for running Job
module load fluent

cd $PBS_O_WORKDIR

# Run Fluent with 32 cpus using the infiniband interconnect and intel mpi
fluent 3ddp -g -t32 -pinfiniband -mpi=intel -ssh < inputlong

## Some job POST-processing may go here
cp ./files.*  $PBS_O_WORKDIR
```

C-3: Simulation Journal File

The simulation journal file is used to perform solution calculation activities with the given FLUENT commands through the TUI. An example is given below.

```
;Load case and data files
rc TAMU0605-07R.cas.gz
rd TAMU0605-07R.dat.gz

/solve/set/courant-number
1
/solve/iterate
100
wc TAMU0605-07R-1.cas.gz
wd TAMU0605-07R-1.dat.gz

/solve/set/courant-number
2
/solve/iterate
200
wc TAMU0605-07R-1.cas.gz
yes
wd TAMU0605-07R-1.dat.gz
yes

/solve/set/courant-number
4
/solve/iterate
300
wc TAMU0605-07R-1.cas.gz
yes
wd TAMU0605-07R-1.dat.gz
yes

/solve/set/courant-number
8
/solve/iterate
400
wc TAMU0605-07R-1.cas.gz
yes
wd TAMU0605-07R-1.dat.gz
yes

;Change to 2nd order discretization
/solve/set/discretization-scheme/amg-c
1
/solve/set/discretization-scheme/k
1
/solve/set/discretization-scheme/omega
1

/solve/set/courant-number
1
/solve/iterate
100
wc TAMU0605-07R-2.cas.gz
```

```
wd TAMU0605-07R-2.dat.gz

;Turn on specified mass flow rate for engine inlet
/define/boundary-conditions/pressure-outlet
eng/eng_in
no
27538.96  ;static pressure
no
231.03 ;total temperature
no
yes
no
no
yes
.1
1
no
yes
no
no
28.1182   ;mass flow rate
no
5000000
no
1

/solve/set/courant-number
2
/solve/iterate
200
wc TAMU0605-07R-2.cas.gz
yes
wd TAMU0605-07R-2.dat.gz
yes

/solve/set/courant-number
4
/solve/iterate
300
wc TAMU0605-07R-2.cas.gz
yes
wd TAMU0605-07R-2.dat.gz
yes

/solve/set/courant-number
8
/solve/iterate
400
wc TAMU0605-07R-2.cas.gz
yes
wd TAMU0605-07R-2.dat.gz
yes

/solve/set/courant-number
16
/solve/iterate
500
wc TAMU0605-07R-3.cas.gz
wd TAMU0605-07R-3.dat.gz
```

```
/solve/set/courant-number
32
/solve/iterate
500
wc TAMU0605-07R-3.cas.gz
yes
wd TAMU0605-07R-3.dat.gz
yes

/solve/set/courant-number
48
/solve/iterate
1000
wc TAMU0605-07R-4.cas.gz
wd TAMU0605-07R-4.dat.gz

/solve/set/courant-number
64
/solve/iterate
1000
wc TAMU0605-07R-5.cas.gz
wd TAMU0605-07R-5.dat.gz

/solve/set/courant-number
64
/solve/iterate
1000
wc TAMU0605-07R-6.cas.gz
wd TAMU0605-07R-6.dat.gz

/solve/set/courant-number
64
/solve/iterate
1000
wc TAMU0605-07R-7.cas.gz
wd TAMU0605-07R-7.dat.gz

exit
yes
```

APPENDIX D

BOUNDARY-LAYER EXTRACTION AND CONDITIONING CODES

D-1: Surfex Journal File

The Surfex journal file automatically performs the coordinate extraction of the test section surface grid. An example is given below.

```
;Compile Surfex UDF
define/user-defined/compiled-functions
compile
"libudf"
yes
"y"
"surfex.c"
""
"n"

;Load Surfex UDF
define/user-defined/compiled-functions
load
"libudf"

;Load Surfex scheme file
(load 'surfexscheme.scm)

;EDIT THIS LINE ACCORDINGLY - enter correct surface name and correct ##X-
glovelam name
(idnumb 'glve_lam '07R-glovelam)

;Unload Surfex UDF
define/user-defined/compiled-functions
unload
"libudf"
```

D-2: Surfex Scheme File

The Surfex scheme file passes information specified in the Surfex journal file to the Surfex UDF and then runs the UDF.

```
(if (not (rp-var-object 'surf_id))(rp-var-define 'surf_id 999 'int #f))
(if (not (rp-var-object 'surf_name))(rp-var-define 'surf_name 'axis →
   string #f))
(if (not (rp-var-object 'file_name))(rp-var-define 'file_name 'axis →
   'string #f))

(define (idnumb idname filename)
   (let ()
       (rpsetvar 'surf_id (surface-name->id idname))
       (rpsetvar 'surf_name idname)
       (rpsetvar 'file_name filename)
       (ti-menu-load-string "define/user-defined/execute-on-demand →
          \"surfex::libudf\"")
))
```

D-3: Surfex UDF

The Surfex UDF writes the surface-grid node coordinates of the glove test section to a text file.

```
#include "udf.h"
#include "surf.h"
#include "cxsurf.h"
#include "cxiface.h"
#include "dx.h"

DEFINE_ON_DEMAND(surfex)
{
   Surface *s;
   Surf_Point *p;
   Domain *d = Get_Domain(1);
   int i, j, k, l,index, nw,nfp;
   real value, x, y, z;
   char *what;
   char *datafn;
   char *filename;
   int line_id;
   FILE *fp;
   double *x_arr,*y_arr,*z_arr,*xv_arr;

   /* Store information from Surfex scheme */
   filename = RP_Get_String("file_name");
   filename[2] = 'R';
   datafn = RP_Get_String("surf_name");
   line_id = RP_Get_Integer("surf_id");

   /* Open text file to write to */
   fp = fopen(filename,"w+");

   /* Specify surface */
   s = SurfaceList+line_id;

   /* Number of points on surface */
   i = s->np;

   /* Allocate memory to store all the variables */
   x_arr = (double *) malloc(i*sizeof(double));
   y_arr = (double *) malloc(i*sizeof(double));
   z_arr = (double *) malloc(i*sizeof(double));
   xv_arr = (double *) malloc(i*sizeof(double));

   /* String name */
   what = "x-velocity";

   /* Fill all the nodes with values of "what" */
   Node_Function_Values(d,what);
   p = s->points;

   /* Surface_Value(p) returns value of "what" at the node */
   /* Extract coordinates (x-velocity not needed) */
   for(i=0;i<s->np;i++)
   {
      x_arr[i] = Get_Surface_Point_Coord(p,X_DIM);
      y_arr[i] = Get_Surface_Point_Coord(p,Y_DIM);
      z_arr[i] = Get_Surface_Point_Coord(p,Z_DIM);
      value = Surface_Value(p);
      xv_arr[i] = value;
      p += 1;
   }

   /* Write the values to a output file */
   for(i=0;i<s->np;i++)
   {
      fprintf(fp,"% 22.15E % 22.15E % 22.15E",x_arr[i],y_arr[i], →
         z_arr[i]);
      fprintf(fp,"\n");
   }

   /* Close text file */
   fclose(fp);
}
```

D-4: NormGen

The NormGen code uses the surface grid coordinates to eventually write a FLUENT journal file that will perform the boundary-layer data extraction along wall-normal rays. The code is broken up into a number of functions, all of which are given below in the order with which the main NormGen code calls them. The NormGen journal file format is given within the fluentexport function code and will not be included in this appendix.

```
%% NormGen - MWR 6 July 2011
%This code takes the surface mesh information from the laminar portion of
%the glove (exported from FLUENT using the surfex UDF) and performs the
%following tasks:
    %1)  Condition the surface data into a useful format
    %2)  Correctly order the surface data
    %3)  Pull section data from a user-specified span location
    %4)  Calculate the surface-normal vectors for the given section
    %5)  Create body-fitted coordinate system for all points in the section
    %6)  Create adjusted coordinates based on tapered wing approximation
    %7)  Filter stations based on node-to-node distance
    %8)  Calcuate section curvature
    %9)  Calculate section chord length
    %10)  Write a FLUENT journal file to extract wall-normal flowfield data
    %11) Save data generated by this program
%The tasks performed by this program and the FLUENT journal file directly
%feed into SARGEBL, which will process the flowfield data and write the
%meanflow file used by LASTRAC.

%Variables:
    %workdir = working directory for all imports and exports related to run
    %surfname = name of glove-laminar-subzone surface coordinate file
    %blin = buttock line input
    %chordtable = matrix with inboard and outboard locations and chord lengths
    %yorn = yes or no string
    %chord = chord length of section found using chordtable and span location
    %csnum = unique suffix file directory and all generated files
    %tol = station filter tolerance (if used)
    %normdist = allowable normal distance from surface for data extraction
    %H = flight condition altitude
    %M = flight condition Mach number
    %savename = name of filename for saved NormGen data

%(See functions for all other variable definitions)

%Units (unless specified otherwise):
    %length = m
    %velocity = m/s
    %pressure = Pa
    %density = kg/m^3
    %temperature = K
    %kinematic viscosity = m^2/s

%% Clear, clc, and close figures
```

```
clear;
clc;
close all;

%% Start Log
[~,~]=system('rm NormGen.txt');
diary('NormGen.txt');
Start Message
fprintf('-----------------\n---- NormGen ----\n-----------------\n');

fprintf('Matt Roberts 2011\n\n\n\n');
pause(1);

%% Working Directory - Change if desired
workdir='~/MATLAB/LASTRACFILES/';

%% Import Data - dataimport.m
fprintf('--- IMPORT DATA ---\n\n');

%Display the working directory
fprintf('Working directory ---> %s\n\n',workdir);

%Prompt user to enter the filename for import
%NOTE - The filename's first three characters must contain the job number.
surfname=input('Enter the name of the file for import:\n-->> ','s');
fprintf('\n');

%Import glove surface data - see description in dataimport.m
%NOTE - The glove data must be placed in the SURFEXDATA folder in the home
%directory
[graw]=dataimport([workdir 'SURFEXDATA/'],surfname);

fprintf('-------------------\n\n\n');
pause(1);

%% Condition Data - datacondition.m
fprintf('--- CONDITION DATA ---\n\n');

%Condition raw data - see description in datacondition.m
[gsec,nps,spg]=datacondition(graw);

fprintf('----------------------\n\n\n');
pause(1);

%% Order Data - dataorder.m, sectionorder.m
fprintf('--- ORDER DATA ---\n\n');

%Order conditioned data - see description in dataorder.m
[gord,listxavg,listLE]=dataorder(gsec,nps,spg);

%Prompt user to look at Figure 1
fprintf('See Figure 1.\n\n');

fprintf('------------------\n\n\n');
pause(1);

%% Find closest section - sectionfind.m
fprintf('--- CHOOSE SECTION ---\n\n');

%Prompt user to input desired span location in inches
```

```
fprintf('Enter desired span location (from %0.1f" to %0.1f"):\n-->> ',...
    ceil(10*listxavg(2)/.0254)/10,floor(10*listxavg(end-1)/.0254)/10);
blin=input('');
fprintf('\n');

% Record section coordinates and section number nearest to blin - see
% description in sectionfind.m
[seccor,secid]=sectionfind(blin,gord,listxavg);

fprintf('----------------------\n\n\n');
pause(1);

%% Find section surface-normal vectors - sectionnorm.m
fprintf('--- CALCULATE NORMALS ---\n\n');

%Generate section surface-normal vectors - see description in sectionnorm.m
[ydir]=sectionnorm(seccor,secid,gord);

fprintf('-------------------------\n\n\n');
pause(1);

%% Create Body-Fitted Coordinate Systems - bfcoorsys.m
fprintf('--- GENERATE BODY-FITTED COORDINATE SYSTEMS ---\n\n');

%Determine body-fitted coordinate system - see description in bfcoorsys.m
[xdir,zdir,offsc]=bfcoorsys(gord,secid,ydir);

%Prompt user to look at Figure 2
fprintf('See Figure 2.\n\n');

fprintf('-----------------------------------------------\n\n\n');
pause(1);

%% Create Adjusted Coordinates - pntadj_tsw.m
fprintf('--- CREATE ADJUSTED SECTION COORDINATES ---\n\n');

%Adjust section coordinates to fit a tapered swept - see description in
%pntadj_tsw.m
[adjseccor]=pntadj_tsw(gord,listLE,seccor,secid);

%Prompt user to look at Figure 3
fprintf('See Figure 3.\n\n');

fprintf('-------------------------------------------\n\n\n');
pause(1);

%% Filter Stations - stationfilter.m
fprintf('--- FILTER STATIONS ---\n\n');

%Ask use if they would like to filter stations
yorn=input('Filter stations? (y/n):\n-->> ','s');
fprintf('\n');

if yorn=='y'
    %Prompt user to enter point spacing tolerance
    tol=input('Enter point spacing tolerance (in):\n-->> ');
    fprintf('\n');

    %Filter points - see description in stationfilter.m
    [seccor,adjseccor,xdir,ydir,zdir,offsc]=stationfilter(tol,seccor, ...
   adjseccor,xdir,ydir,zdir,offsc);
elseif yorn=='n'
    %Remove fi0rst and last points of seccor since no normals are generated
    %for those points
    seccor(1,:)=[];
    seccor(end,:)=[];

    %Print number of stations
    fprintf('There are %d stations.\n\n',size(seccor,1));
else
    error('ERROR - Enter y or n');
end

fprintf('-----------------------\n\n\n');
pause(1);

%% Section Curvature Calculation - curvature.m
fprintf('--- CALCULATE SECTION CURVATURE ---\n\n');

%Calculate the curvature of the adjusted coordinates - see description in
%curvature.m
[K,t,adjseccor]=curvature(adjseccor);

fprintf('-----------------------------------\n\n\n');

%% Calculate Section Chord Length
fprintf('--- CALCULATE SECTION CHORD LENGTH ---\n\n');

%Calculate total chord length
%NOTE - The user should specify the glove's inboard and outboard edge chord
%lengths so an interpolation can be performed (assuming a straight loft
%between the two profiles) to determine the section chord length.  Updates
%to these numbers must be performed here.  Span locations and chord lengths
%are given in meters.  The number order is as follows:
    %chordtable(1,:) = [inboard span location, inboard chord]
    %chordtable(2,:) = [outboard span location, outboard chord].
chordtable=[5.0292 4.809744;6.8580 3.994404];

%Print chord table information (in inches)
fprintf('          Span Location    Chord Length\n');
fprintf(' Inboard:%9.0f" %16.2f"\n',chordtable(1,:)/.0254);
fprintf('Outboard:%9.0f" %16.2f"\n\n',chordtable(2,:)/.0254);

%Ask user if these numbers are correct
yorn=input('Are the span locations and chord lengths correct? (y/n):\n ','s');
fprintf('\n');

if yorn=='n'
    error('ERROR - Enter correct chord information in NormGen.m');
elseif yorn~='y'
    error('ERROR - Enter y or n');
end

%Calculate chord length
chord=interp1(chordtable(:,1),chordtable(:,2),mean(seccor(:,1)));

%Print the chord length (in inches)
fprintf('The chord length is %0.2f.\n\n',chord/.0254);

fprintf('--------------------------------------\n\n\n');
```

```
pause(1);

%% Export data to FLUENT - fluentexport.m
fprintf('--- CREATE FLUENT JOURNAL FILE ---\n\n');


%Create unique suffix for particular run using the case and section numbers
%NOTE - The case number is derived from the original surfname (the first
%three characters) and the section number is the rounded inch value of the
%section span location
csnum=[surfname(1:3) '-' int2str(round(blin))];

%Create a file directory for all imports and exports to go in
%NOTE - This directory is initially named after the case and section number
%(csnum).  It's name is later appended to inlude the section side.
[~,~,~]=mkdir(workdir,csnum);

%Display the file directory for all NormGen and SARGEBL files
fprintf('NormGen file directory ---> %s\n\n',[workdir csnum '/']);

%User input allowable normal distance in laminar subzone
normdist=input('Enter the allowable normal distance (in):\n-->> ');
fprintf('\n');

%User input altitude (in feet)
H=input('Enter the altitude (from 25kft to 50kft):\n-->> ');
fprintf('\n');

%User input Mach number
M=input('Enter the Mach number:\n-->> ');
fprintf('\n');

%Generate FLUENT journal file and record section surface-normal (SSN) line
%endpoints - see description in fluentexport.m
[fluentex,airprop]=fluentexport(gord,seccor,ydir,normdist,H,M, ...
    chord,offsc,workdir,csnum);

%Prompt user to look at Figure 4
fprintf('See Figure 4.\n\n');

fprintf('----------------------------------\n\n\n');
pause(1);

%% Save data for import into SARGEBL
fprintf('--- SAVE NORMGEN DATA ---\n\n')

%Save body-fitted coordinate system data, section coordinates, adjusted
%section cooridnates, section curvature, chord length, altitude, Mach
%number, standard atmospheric conditions at altitude, and FLUENT journal
%file name
savename=['NormGendata-' csnum '.mat'];
save([workdir csnum '/' savename],'xdir','ydir','zdir','seccor',...
    'adjseccor','K','chord','H','M','airprop','fluentex');

%Display the NormGen data filename
fprintf('NormGen data filename ---> ');
fprintf(savename);
fprintf('\n\n');

fprintf('-------------------------\n\n\n');
```

```
pause(1);

% Finish message
fprintf('-----------------\n--- GO STATE! ---\n-----------------\n');

%% Store Log
diary off
[~,~]=system(['cp NormGen.txt ' workdir csnum '/NormGen-' csnum '.log']);
[~,~]=system('rm NormGen.txt');
```

%% FUNCTION: NormGen.dataimport - MWR 6 July 2011

```
%This function imports glove surface data (that was exported from FLUENT)
%into MatLab. The file 'surfname' must be an ASCII file, containing only
%the surface nodes of the laminar glove subzone (no other properties).  If
%one were to open the ASCII file in a word editor, the following columns
%(in this order) should be listed: nodenumber, x-coordinate, y-coordinate,
%and z-coordinate.  The node number is unnecessary and will be removed in
%this fuction.

%Variables:
    %directory = directory containing surface data (workdir/SURFEXDATA/...)
    %import = structure containing all data brought in by the import
    %import.data = numerical data in the structure
    %graw = last 3 cols of import.data (x, y, z surface coordinates's only)

function [graw]=dataimport(directory,surfname)

fprintf('Importing data from %s...\n',surfname);

%Import data based on given surfname
import=importdata([directory surfname]);

%Check to make sure the imported data contains only node number and x, y, z
%coordinates
if size(import,2) ~= 3
    error('ERROR - Incorrect number of variables, check imported data')
end

%Keep only x, y, z coordinates
graw=import;

fprintf('Done.\n\n');
```

%% FUNCTION: NormGen.datacondition - MWR 6 July 2011

```
%This function takes the raw data and sorts it into individual, unordered
%airfoil sections.  The section coordinates are then stored in a 3D matrix
%(dim1=node, dim2=xyz-coordiate, dim3=section).  The coordinate deviation
%in the x-direction and number of nodes per section and number of sections
%per glove are recorded.

%Variables:
    %nps = nodes per section
    %spg = sections per glove
    %gsec = matrix of glove surface data arranged in sections

function [gsec nps spg]=datacondition(graw)

fprintf('Conditioning the surface data...\n')

%Sort matrix initially by x-coordinate values (span locations)
graw=sortrows(graw,1);

%Find number of nodes per section
nps=1;
while abs(graw(nps+1,1)-graw(nps,1)) < .005 && nps<size(graw,1)
    nps=nps+1;
end
%NOTE - .005 serves as a limit since there is some slight deviation
%between section x-coordinates.  The spanwise section spacing should always
%be greater than 5mm and the x-coordinate deviation should always be less
%than 5mm

%Find number of sections
spg=size(graw,1)/nps;

%Check to make sure nps divides evenly and gives integer spg
if spg-round(spg)~=0
    error('ERROR - Number of nodes per section not uniform')
end

%Create 3D matrix that contains coordinates of one section per level
%NOTE - The sections are arranged inboard to outboard
for i=1:spg
    gsec(:,:,i)=graw(nps*(i-1)+1:i*nps,:);
end

fprintf('Done.\n\n');

%Display grid information to user - nps and spg
fprintf('There are %d nodes per section.\n\n',nps);
fprintf('There are %d sections per glove.\n\n',spg);
```

%% FUNCTION: NormGen.dataorder - MWR 6 July 2011

```
%This function stores all ordered section cooridnates as well as all LE
%information (id and coordinates), the average section x-coodinate, and the
%x-coordinate deviation.  All sections are plotted together for inspection
%purposes. The user will be informed of the maximum deviation in the
%x-coordinates as a further check. The data format for gord is the same as
%gsec.

%Variables:
    %colormap = the colormap for the section plot
    %scrnsize = size of screen (in pixels)
    %position = position matrix used for plot
    %listLE = list of LE information
    %listxavg = list of average x-coordinate per section
    %listxdev = list of maximum x-coordinate devation per section
    %gord = 3D matrix containing ordered coordinates for all sections
```

```
function [gord,listxavg,listLE]=dataorder(gsec,nps,spg)

fprintf('Ordering surface data...\n');

%Define colormap for plot
colormap=jet(spg);

%Initialize figure
figure(1);

%Figure settings
scrnsize=get(0,'ScreenSize');
position=[.4*scrnsize(3) .6*scrnsize(4) .3*scrnsize(3) .4*scrnsize(4)];
set(gcf,'Units','pixels','OuterPosition',position,'Color','w');
axes('Units','pixels','Xdir','reverse','Ydir','reverse',...
    'Position',get(gcf,'Position').*[0 0 1 1]+[5 5 -10 -10]);
axis equal;
axis off;

hold on;

%Reorder each section in gsec
for i=1:spg
    %Send each section to the fuction sectionorder.m - see description in
    %sectionorder.m
    [tempord LEid LExyz xavg xdev]=sectionorder(gsec(:,:,i),nps);

    %Plot each section as it is reordered as a check
    plot3(tempord(:,3),tempord(:,1),tempord(:,2),'Color',colormap(i,:));

    %Append list of LE information (LEid and LExyz)
    %Note - The format is [LEid LEx LEy LEz] with each row corresponding to
    %one section
    listLE(i,:)=[LEid LExyz];

    %Append lists of xavg, and xdev for all sections
    listxavg(i)=xavg;
    listxdev(i)=xdev;

    %Append gord
    gord(:,:,i)=tempord;

end

hold off;

view(-45,20);
axis tight;

fprintf('Done.\n\n');

%Show user maximum x-coordinate deviation as a check, should be low (<1E-4)
fprintf('The maximum deviation in the x-coordinates is %0.4Em.\n\n',...
    max(abs(listxdev)));
```

%% FUNCTION: NormGen.sectionorder - MWR 6 July 2011

```
%This function reorders the nodes for one section of the glove data,
%starting with the upper, aft-most point, working its way around to the LE
%point, and ending with the lower, aft-most point.  It first locates the LE
%point and then assigns all points with higher y-coordinates to the upper
%side and all points with lower y-coordinates to the lower side.  The
%location of the LE point and its ID are recorded for comparison purposes.
%The deviation in the x-coordinates and the average of the x-coordinates
%are computed and recorded as well.

%Variables:
    %tempsec = 2D matrix containing unordered coordinates for a given section
    %LEid = gives the row number of the LE point
    %LExyz = xyz-coordinates corresponding to the LE point
    %upper = coordinates located above LE point
    %lower = coordinates located below LE point
    %countupper = number of points on the upper portion of the section
    %tempord = 2D matrix containg ordered coordinates for a given section
    %xavg = average of x-coordinates for given section
    %xdev = maximum deviation in x-coordinates for given section

function [tempord LEid LExyz xavg xdev]=sectionorder(tempsec,nps)

%Locate LE point by finding forward-most z-coordinate
[~,LEid]=max(tempsec(:,3));

%Store LE point coordinates
LExyz=tempsec(LEid,:);

%Preallocate upper and lower side matrices
upper=zeros(nps,3);
lower=zeros(nps,3);

%Sort coordinates by pressure or suction side criteria (above or below LE)
for i=1:nps
    if tempsec(i,2) > LExyz(1,2)
        upper(i,:)=tempsec(i,:);
    end
    if tempsec(i,2) < LExyz(1,2)
        lower(i,:)=tempsec(i,:);
    end
end

%Sort sides by z-coordinate (upper TE to LE, lower LE to TE)
upper=sortrows(upper,3);
lower=sortrows(lower,-3);

%Count total #non-zero elements and divide by #cols to correctly place LE
%coordinate in the tempord matrix
countupper=nnz(upper)/3;

%The LE is one after the upper side ends
LEid=countupper+1;

%Generate ordered coordinate matrix - Upper TE to LE to lower TE
tempord=upper+lower;
tempord(LEid,:)=LExyz;

%Calculate average of the x-coordinates
```

```
xavg=mean(tempord(:,1));

%Calculate the maximum deviation in the x-coordinate
xdev=range(tempord(:,1));
```

%% FUNCTION: NormGen.sectionfind - MWR 7 July 2011

```
%This function takes a user inputted span location and finds the nearest
%existing span section (from gord).  The options for span section input are
%limited so the user must choose a section that can be evaluated by the
%code. The section coordinates and section number are recorded (same
%standard format).

%Variables:
    %gib = glove inboard boundary in the spanwise direction
    %gob = glove outboard boundary in the spanwise direction
    %xdiff = temporary matrix giving differences between listxavg and blin
    %secid = ID of section with closest x-coordinates to blin
    %seccor = section coordinates corresponding to secid

function [seccor,secid]=sectionfind(blin,gord,listxavg)

%% Set limits on span location
%Set inboard limit rounded up to the nearest 0.1" (.0254 = conversion)
%NOTE - This limit is set to be outboard of the second section
gib = ceil(10*listxavg(2)/.0254)/10;

%Set outboard limit rounded down to the nearest 0.1" (.0254 = conversion)
%NOTE - This limit is set to be inboard of the second-to-last section
gob = floor(10*listxavg(end-1)/.0254)/10;

%% Check input span location
%Check to see if span location is outside of boundary
if blin<gib || blin>gob
    error(sprintf('ERROR - the input should be between %d and %d',gib,gob))
    return
end

%% Find and record nearest section to input BL
%Convert span location to meters
blin=blin*.0254;

%Determine difference between specified BL and existing BL values and
%record the section ID
xdiff=listxavg-blin;
[~, secid]=min(abs(xdiff));

%Inform user what BL location (to the nearest .1") was closest to the
%input and will subsequently be used
%NOTE - This is not the exact x-coordinate of the section
fprintf('The closest span location (~%0.1f") will be used.\n\n',...
    round(10*listxavg(secid)/.0254)/10);

%Record section coordiates
seccor=gord(:,:,secid);
```

%% FUNCTION: NormGen.sectionnorm - MWR 8 July 2011

```
%This code calculates the section surface-normal vectors by crossing
%spanwise and streamwise vectors, which are found using a central
%differencing scheme around the node of interest.  The SSN vectors are then
%normalized (for future scaling purposes).

%Variables:
    %secib = coordinates of section immediately inboard of seccor
    %secob = coordinates of section immediately outboard of seccor
    %spannorm = spanwise vectors for each node of seccor (central diff)
    %streamnorm = streamwise vectors for each node of seccor (central diff)
    %ydir = section surface-normal vector for each node

function [ydir]=sectionnorm(seccor,secid,gord)

fprintf('Calculating section surface-normal vectors...\n');

%% Determine spanwise vectors
%Store coordinates from section immediately inboard of primary section
secib = gord(:,:,secid-1);

%Store coordinates from section immediately outboard of primary section
secob = gord(:,:,secid+1);

%Determine spanwise vectors based on secib and secob
spannorm=secob-secib;

%Delete first and last row (eliminate normals near the boundaries)
spannorm(1,:)=[];
spannorm(end,:)=[];

%% Determine streamwise vectors
%Pre-allocate streamwise vector matrix
streamnorm=zeros(size(spannorm));

%Determine streamwise vectors for each node using the -1 and +1 nodes.  The
%vectors for the first and last nodes are not included
for i=2:size(seccor,1)-1
    streamnorm(i-1,:)=seccor(i+1,:)-seccor(i-1,:);
end

%% Determine section surface-normal vectors
%Perform cross product, ensuring vectors point outward, and normalize
ydir=normr(cross(streamnorm,spannorm,2));

fprintf('Done.\n\n');
```

%% FUNCTION: NormGen.bfcoorsys - MWR 12 July 2011

```
%This function creates local coordinate systems (LCS) for each node around
%a given section.  The section surface-normal vectors define the LCS
%y-direction.  The LCS z-direction is chosen to point outboard (a product
%of placing the glove on the port wing) along iso-x/c lines on the glove.
%This direction is found using the surface node locations of the glove.  As
%noted below, care must be taken to ensure the grid is generated properly
```

```
%to assume this direction.  The LCS x-direction is found by taking the
%cross product of the LCS y- and z-directions.  Its direction is assumed to
%point (more or less) aft on the suction side and forward on the pressure
%side.  A plot of the three LCS components at their surface node locations
%is created to check the calculations.

%Variables:
    %xdir = local coordinate system x-direction
    %ydir = local coordinate system y-direction
    %zdir = local coordinate system z-direction
    %ipnorm = section in-plane normal
    %offsc = scaling for flowfield data lines
    %tempx = temporary x-coordinate used for plotting purposes
    %tempy = temporary y-coordinate used for plotting purposes
    %tempz = temporary z-coordinate used for plotting purposes
    %scrnsize = size of screen (in pixels)
    %position = position matrix used for plot

function [xdir,zdir,offsc]=bfcoorsys(gord,secid,ydir)

fprintf('Generating body-fitted coordinate systems...\n');

%Calculate z-direction (spanwise) vectors based on section location,
%skipping the first and last points since BLP's are not extracted for them
%NOTE - This step assumes spanwise gridline run at constant x/c positions.
%Gridding practices should reflect this requirement as closely as possible.
%The spanwise direction is assumed to point outboard, matching the
%standard suction-side x-direction for boundary layer stability
%calculations
zdir=normr(gord(2:end-1,:,secid+1)-gord(2:end-1,:,secid-1));

%Cross zdir and ydir to obtain x-direction (streamwise) vectors
%NOTE - the order is (ydir x zdir) to ensure the x-direction is pointing in
%the correct direction on the suction side.  Once the attachment line is
%located, the x-direction (and subsequently the z-direction) will need to
%be switched for any pressure side calculations.
xdir=cross(ydir,zdir,2);

%Create section in-plane normals (x-dir set to 0)
ipnorm=normr([zeros(size(ydir,1),1) ydir(:,2:3)]);

%Calculate the offset scaling factors
%NOTE - This calculation is simply a dot product of ydir and ipnorm, which
%equals the |ydir|*|ipnorm|*cos(theta).  Since both vectors are normalized,
%all that is left is cos(theta).  This is inversed to make scaling for
%flowfield data lines larger near the leading edge (where the included
%angle between ipnorm and ydir is larger).  The reason for doing this is to
%have flowfield data lines cross through roughly the same number of cells
%during data extraction.
offsc=(ydir(:,1).*ipnorm(:,1)+ydir(:,2).*ipnorm(:,2)+ydir(:,3).*ipnorm(:,3)).^-
1;

fprintf('Done.\n\n');

%% Plot BF Coordinate System Vectors
%Generate reformatted matrices from gord - plotting purposes only
for i=1:size(gord,3)
    tempx(:,i)=gord(:,1,i);
    tempy(:,i)=gord(:,2,i);
    tempz(:,i)=gord(:,3,i);
```

```
end

%Initialize figure
figure(2);

%Figure settings
scrnsize=get(0,'ScreenSize');
position=[.7*scrnsize(3) .6*scrnsize(4) .3*scrnsize(3) .4*scrnsize(4)];
set(gcf,'OuterPosition',position,'Color','w');
axes('Units','pixels','Xdir','reverse','Ydir','reverse', ...
    'Position',get(gcf,'Position').*[0 0 1 1]+[5 5 -10 -10]);
axis equal
axis off;
grid off;

hold on;

%Plot glove surface
surf(tempz,tempx,tempy,'FaceColor',[.9 .9 .9],'EdgeColor','none');

%Plot glove surface boundaries
plot3(gord(:,3,1),gord(:,1,1),gord(:,2,1),'k');
plot3(gord(:,3,end),gord(:,1,end),gord(:,2,end),'k');
plot3(tempz(1,:),tempx(1,:),tempy(1,:),'k');
plot3(tempz(end,:),tempx(end,:),tempy(end,:),'k');

%Plot section of interest
plot3(gord(:,3,secid),gord(:,1,secid),gord(:,2,secid),'r');

%Plot section local coordinate system vectors (xdir, ydir, and zdir)
quiver3(gord(2:end-1,3,secid),gord(2:end-1,1,secid),gord(2:end-1,2,secid)...
    ,xdir(:,3),xdir(:,1),xdir(:,2),'c');
quiver3(gord(2:end-1,3,secid),gord(2:end-1,1,secid),gord(2:end-1,2,secid)...
    ,ydir(:,3),ydir(:,1),ydir(:,2),'g');
quiver3(gord(2:end-1,3,secid),gord(2:end-1,1,secid),gord(2:end-1,2,secid)...
    ,zdir(:,3),zdir(:,1),zdir(:,2),'m');

hold off;

view(-45,20);
axis tight;
```

%% FUNCTION: pntadj_tsw - MWR 27 July 2011

```
%This function takes the section coordinates and adjusts them as if they
%were taken from a tapered swept wing.  The assumption is that flowfield
%properties do not vary significantly along iso-x/c lines.  The body-fitted
%coordinate system already takes this into account by orienting the
%z-direction with spanwise grid lines, which should be long iso-x/c lines
%if the grid was generated correctly.  The adjusted coordinates are only
%used to calculate body-fitted distance and curvature, a step that occurs
%during the BLP property generation section of SARGEBL. Leading and
%trailing (for the glovelam surface) edge points are found and used to find
%the focus of the tapered glove.  The tranformation is based on the
%distance from the focus to the section leading edge point. Dihedral is
%taken into account and the projection is performed along a line running
%vertically through the focus, rather than just from the focus itself.  A
```

```
%plot for the adjusted coordinates is shown for reference.

%Variables:
    %ibLEp = inboard section leading edge point coordinates
    %ibTEp = inboard section trailing edge point coordinates
    %secLEp = section of interest leading edge point coordinates
    %obLEp = outboard section leading edge point coordinates
    %obTEp = outboard section trailing edge point coordinates
    %mLEy = y-slope of leading edge line
    %mLEz = z-slope of leading edge line
    %mTEz = z-slope of trailng edge line
    %c = streamwise range for section of interest
    %eqpline = matrix solving set of equations to find location of focus
    %focus = intersection of leading and trailing edge lines
    %dist = distance from the section leading edge point to the focus
    %tempfocus = temporary focus for y-coordinate shift, unique to each point
    %dir = direction defined by section point and tempfocus
    %adjseccor = adjusted coordinates using the tapered swept wing
approximation
    %tempx = temporary x-coordinate used for plotting purposes
    %tempy = temporary y-coordinate used for plotting purposes
    %tempz = temporary z-coordinate used for plotting purposes
    %scrnsize = size of screen (in pixels)
    %position = position matrix used for plot

function [adjseccor]=pntadj_tsw(gord,listLE,seccor,secid)

fprintf('Generating adjusted coordinates using tapered-wing
approximation...\n');

%Store the leading and trailing edge points for the inboard and outboard
%glove section, as well as the section leading edge point
%NOTE - the TE points are defined as the aft-most point on the suction side
%of the glovelam data
ibLEp=listLE(1,2:4);
ibTEp=gord(1,:,1);
secLEp=listLE(secid,2:4);
obLEp=listLE(end,2:4);
obTEp=gord(1,:,end);

%Calculate the slopes of the leading and trailing edge lines
mLEy=(obLEp(2)-ibLEp(2))/(obLEp(1)-ibLEp(1));
mLEz=(obLEp(3)-ibLEp(3))/(obLEp(1)-ibLEp(1));
mTEz=(obTEp(3)-ibTEp(3))/(obTEp(1)-ibTEp(1));

%Calculte the length between the maximum and minimum z-coordinate for the
%inboard section
c=abs(max(gord(:,3,1))-min(gord(:,3,1)));

%Solve set of equations to locate focus
%NOTE - the inboard section leading edge coordinates serve as the origin in
%this calculation
eqline=rref([-mLEy 1 0 0;-mLEz 0 1 0;-mTEz 0 1 -c]);

%Store the location of the focus using the global coordinate system
focus=ibLEp+eqline(1:3,end)';

%Calcualte the distance from the section leading edge point to the focus
dist=norm(secLEp-focus);
```

```
%Calculate the adjusted coordinates
%NOTE - tempfocus is shifted up and down from the focus based on the
%y-coordinate of the section point with respect to the section leading edge
%point
for i=1:size(seccor,1)
    tempfocus(i,:)=focus+[0 seccor(i,2)-secLEp(2) 0];
    dir(i,:)=(seccor(i,:)-tempfocus(i,:))/norm(seccor(i,:)-tempfocus(i,:));
    adjseccor(i,:)=dir(i,:)*dist+tempfocus(i,:);
end

fprintf('Done.\n\n');

%% Plot Adjusted Coordinates
%Generate reformatted matrices from gord - plotting purposes only
for i=1:size(gord,3)
    tempx(:,i)=gord(:,1,i);
    tempy(:,i)=gord(:,2,i);
    tempz(:,i)=gord(:,3,i);
end

%Initialize figure
figure(3);

%Figure settings
scrnsize=get(0,'ScreenSize');
position=[.4*scrnsize(3) .2*scrnsize(4) .3*scrnsize(3) .4*scrnsize(4)];
set(gcf,'OuterPosition',position,'Color','w');
axes('Units','pixels','Xdir','reverse','Ydir','reverse', ...
    'Position',get(gcf,'Position').*[0 0 1 1]+[5 5 -10 -10]);
axis equal
axis off;
grid off;

hold on;

%Plot glove surface boundaries
plot3(gord(:,3,1),gord(:,1,1),gord(:,2,1),'k');
plot3(gord(:,3,end),gord(:,1,end),gord(:,2,end),'k');
plot3(tempz(1,:),tempx(1,:),tempy(1,:),'k');
plot3(tempz(end,:),tempx(end,:),tempy(end,:),'k');

%Plot section of interest
plot3(seccor(:,3),seccor(:,1),seccor(:,2),'-r.')

%Plot taper lines
plot3(focus(3),focus(1),focus(2),'*g');
plot3([focus(3) ibLEp(3)],[focus(1) ibLEp(1)],[focus(2) ibLEp(2)],'g');
plot3([focus(3) ibTEp(3)],[focus(1) ibTEp(1)],[focus(2) ibLEp(2)],'g');

%Plot temporary focus points
plot3([tempfocus(1,3) tempfocus(end,3)], ...
    [tempfocus(1,1) tempfocus(end,1)], ...
    [tempfocus(1,2) tempfocus(end,2)],'-c*');

%Plot adjusted coordinated for section of interest
plot3(adjseccor(:,3),adjseccor(:,1),adjseccor(:,2),'-b.');

hold off;

view(-45,45); %Isometric view
```

```
%view(270,0); %Front view outboard to right
%view(90,90); %Top view, outboard to left
axis tight;
```

%% FUNCTION: NormGen.stationfilter - MWR 15 Aug 2011

```
%This function filters the number of stations based on node-to-node
%distance.  Should the user employ this function in NormGen, a tolerance
%will be given that sets the minimum allowable distance between nodes.
%Since there are a large number of nodes near the leading edge, the filter
%should affect that area only.  The filtering process is necessary for two
%reasons.  First, LASTRAC LPSE calculations have difficulty when stations
%are closely spaced.  Second, the curvature calculation is very noisy near
%the leading edge when all points are including.  Several data sets are
%affected by this function: seccor, adjseccor, xdir, ydir, zdir, and offsc.
% Upon completion of the filtering process, the new number of nodes is
% printed along with the old number of nodes for comparison purposes and
% all modified data sets are returned to NormGen.

%Variables:
    %orgstat = pre-filter number of stations
    %c1 = counter that records number of nodes that were removed
    %c2 = counter guarding against an infinite loop
    %dist = distance between consecutive nodes (backward difference)

function [seccor,adjseccor,xdir,ydir,zdir,offsc]=stationfilter(tol,seccor, ...
   adjseccor,xdir,ydir,zdir,offsc)

%Remove first and last points of seccor since no normals are generated for
%those points
seccor(1,:)=[];
seccor(end,:)=[];

%Record original number of stations
orgstat=size(seccor,1);

c1=0;
c2=0;
i=2;

fprintf('Filtering stations...\n');

%Filter all surface data based on inputted tolerance
while i<size(seccor,1) && c2<orgstat
    %Compute distance between consecutive points
    dist=norm(seccor(i,:)-seccor(i-1,:));

    %Remove rows below minimum spacing tolerance
    %NOTE - Since the adjusted section coordinates have two extra points
    %(one at the upper aft-most location and one at the lower aft-most
    %location), it must be offset by 1 to match all the other data sets
    if dist<tol*.0254
        seccor(i,:)=[];
        adjseccor(i+1,:)=[];
        xdir(i,:)=[];
        ydir(i,:)=[];
        zdir(i,:)=[];
        offsc(i,:)=[];
        c1=c1+1;
    else
        i=i+1;
    end
    c2=c2+1;
end

fprintf('Done.\n\n');

%Print station filter statistics
fprintf('               Pre-Filtered   Filtered\n');
fprintf('Station Count      %3d           %3d\n\n',orgstat,size(seccor,1));
```

%% FUNCTION: NormGen.curvature - MWR 4 August 2011

```
%This function calculates the curvature of the adjusted coordinates.  It
%uses a non-constant-spacing truncated Taylor series for 1st and 2nd
%derivative calculations.  The curve that represents the adjusted
%coordinates is parametrized by arc length and all derivatives
%are taken with respect to the differential length dt, e.g. dx/dt,
%d^2x/dt^2, etc.  The curvature, arc length, and truncated adjusted
%coordinates (the first and last points are removed since no normals are
%generated for those points) are returned.

%Variables:
    %mxp = x-coordinates at i-1, i, and i+1
    %myp = y-coordinates at i-1, i, and i+1
    %mzp = z-coordinates at i-1, i, and i+1
    %dtm = differential arc length at i using a backward difference
    %dtp = differential arc length at i using a forward difference
    %dx = 1st derivative of x-coordinates at i with respect to arc length
    %dy = 1st derivative of y-coordinates at i with respect to arc length
    %dz = 1st derivative of z-coordinates at i with respect to arc length
    %ddx = 2st derivative of x-coordinates at i with respect to arc length
    %ddy = 2st derivative of y-coordinates at i with respect to arc length
    %ddz = 2st derivative of z-coordinates at i with respect to arc length
    %t = cumlative arc length along curve generated by adjusted coordinates
    %K = curvature of adjusted coordinates

function [K,t,adjseccor]=curvature(adjseccor)

fprintf('Calculating section curvature...\n');

%Calculate the 1st and 2nd derivatives
for i=2:size(adjseccor,1)-1
    %(ft-1) f(t) f(t+1) for x, y, and z
    mxp=[adjseccor(i-1,1) adjseccor(i,1) adjseccor(i+1,1)];
    myp=[adjseccor(i-1,2) adjseccor(i,2) adjseccor(i+1,2)];
    mzp=[adjseccor(i-1,3) adjseccor(i,3) adjseccor(i+1,3)];

    %Backward and forward deltas along curve
    dtm(i)=norm(adjseccor(i,:)-adjseccor(i-1,:));
    dtp(i)=norm(adjseccor(i+1,:)-adjseccor(i,:));

    %1st derivatives
    dx(i)=(mxp(3)-mxp(1))/(dtm(i)+dtp(i));
    dy(i)=(myp(3)-myp(1))/(dtm(i)+dtp(i));
    dz(i)=(mzp(3)-mzp(1))/(dtm(i)+dtp(i));

    %2nd derivatives
    ddx(i)=(2*mxp(3)-4*mxp(2)+2*mxp(1)-2*dx(i)*(dtp(i)- ...
        dtm(i)))/(dtp(i)^2+dtm(i)^2);
    ddy(i)=(2*myp(3)-4*myp(2)+2*myp(1)-2*dy(i)*(dtp(i)- ...
        dtm(i)))/(dtp(i)^2+dtm(i)^2);
    ddz(i)=(2*mzp(3)-4*mzp(2)+2*mzp(1)-2*dz(i)*(dtp(i)- ...
        dtm(i)))/(dtp(i)^2+dtm(i)^2);
end

%Record cumlative arc length
t=cumsum(dtm);

%Delete the first rows from dx, dy, dz, ddx, ddy, ddz, and t
dx(1)=[];
dy(1)=[];
dz(1)=[];
ddx(1)=[];
ddy(1)=[];
ddz(1)=[];
t(1)=[];

%Re-zero arc length
t=t-t(1);

%Calculate curvature
K=(sqrt((ddz.*dy-ddy.*dz).^2+(ddx.*dz-ddz.*dx).^2+(ddy.*dx-ddx.*dy).^2)./ ...
    ((dx.^2+dy.^2+dz.^2).^(3/2)))';

%Remove first and last points from adjseccor to match previously generated
%section surface normals
%NOTE - No normals are generated for the first and last points of the
%section coordinates.  The adjusted section coordinates should reflect this
%as well before being exported to SARGEBL.
adjseccor(1,:)=[];
adjseccor(end,:)=[];

fprintf('Done.\n\n');
```

%% FUNCTION: NormGen.fluentexport - MWR 9 July 2011

```
%This function takes the coordinates for a given section and its
%surface-normal vectors and outputs a journal file that will used in FLUENT
%to obtain boundary layer information.  The user inputted normal distance
%entered should be based on grid mesh knowledge, i.e. how far away from
%surface the laminar subzone extends.  An approximate boundary layer height
%is calculated by using flight conditions (determined by US Standard
%Atmosphere 1976) and a Blasius boundary layer model for comparison
%purposes.  The start points of flowfield data lines begin on the glove
%surface and terminate useing the normal distance and the offset scaling
%factors calculated earlier.  Once this information is computed, all
%flowfield data line information is written to the journal file that can be
%read directly by FLUENT.

%Variables:
    %satm = US Standard Atmosphere 1976 table
    %airprop = Standard atmosphere properties at a given altitude (H)
    %Rec = section Reynolds number based on chord freestream properties
    %delta = approximate boundary layer height
    %p0 = start points of the surface-normal lines (on the glove surface)
    %p1 = end points of the surface-normal lines
    %tempx = temporary x-coordinate used for plotting purposes
    %tempy = temporary y-coordinate used for plotting purposes
    %tempz = temporary z-coordinate used for plotting purposes
    %scrnsize = size of screen (in pixels)
    %position = position matrix used for plot
    %matlabex = name of FLUENT journal file to be exported
    %fluentex = name of ASCII FLUENT file containing boundar layer data

function [fluentex,airprop]=fluentexport(gord,seccor,ydir,normdist,H,M, ...
    chord,offsc,workdir,csnum)

%% Load US Standard Atmosphere 1976 table
%The format is as follows: H(ft) p(Pa) rho(kg/m^3) T(K) mu(kg/ms) c(m/s)
%Altitude ranges from 25000ft to 50000ft (within SARGE science envelope)
%NOTE - Small errors (< 1%) have been found in this data
load satm.mat

%% Check user inputted altitude
%Check to make sure altitude is within range
if H<min(satm(:,1)) || H>max(satm(:,1))
    error('ERROR - The specified altitude is outside the table range');
end

%% Interpolate atmospheric data and calculate Blasius boundary layer height
%Interpolate on H to get atmospheric property values
airprop=[H interp1(satm(:,1),satm(:,2:end),H)];

%Calculate the chord Reynolds number based on the section chord length and
%the freestream properties
Rec=(airprop(3)*airprop(6)*M*chord)/airprop(5);

%Calculate Blasius boundary layer height (eta = 5)
%The reference length is z-coordinate range of seccor
delta=5*sqrt((airprop(5)*range(seccor(:,3)))/(airprop(3)*airprop(6)*M));

%Display the chord Reynolds number
fprintf('The section chord Reynolds number is %0.2fM.\n\n',Rec/1E6);

%Display boundary layer height, normal distance, and ratio of the two
fprintf('The approximate boundary layer height is %0.2fmm.\n\n',1000*delta);
fprintf('The wall-normal distance for flowfield data extraction is
%0.2fmm.\n',...
    25.4*normdist);
fprintf('This is %0.1f times the boundary layer height.\n\n',...
    normdist*.0254/delta);

%% Record start and end points of normals
%Start points are original section coordinates corresponding to generated
%normals
p0=seccor;

%End points are original section coordinates added to scaled ydir
%NOTE - ydir is scaled by normdist (converted to meters) and offsc. The
%number that the user inputs should be based on the offset distance of the
```

```
%laminar subzone. The scaling provided by offsc increases the normal
%distance based on the angle the section surface normal makes with the
%in-plane (y-z plane for these simulations) section normal.  The larger the
%angle (as seen near the leading edge for example), the larger the normal
%distance.  This is done because the grid spacing is in the section normal
%direction, rather than the surface normal direction. Flow field data lines
%with large in-plane angles cross fewer cells and offscshould correct this.
%In the end, the primary goal is for all the boundary layers to have
%roughly then same height for a given number of points.
p1=seccor+((normdist*.0254)*offsc*ones(1,3)).*ydir;

%% Plot start and end points for the flowfield data lines
%Generate reformatted matrices from gord - plotting purposes only
for i=1:size(gord,3)
    tempx(:,i)=gord(:,1,i);
    tempy(:,i)=gord(:,2,i);
    tempz(:,i)=gord(:,3,i);
end

%Initialize figure
figure(4);

%Figure settings
scrnsize=get(0,'ScreenSize');
position=[.7*scrnsize(3) .2*scrnsize(4) .3*scrnsize(3) .4*scrnsize(4)];
set(gcf,'OuterPosition',position,'Color','w');
axes('Units','pixels','Xdir','reverse','Ydir','reverse', ...
    'Position',get(gcf,'Position').*[0 0 1 1]+[5 5 -10 -10]);
axis equal
axis off;
grid off;

hold on;

%Plot glove surface
surf(tempz,tempx,tempy,'FaceColor',[.9 .9 .9],'EdgeColor','none');

%Plot glove surface boundaries
plot3(gord(:,3,1),gord(:,1,1),gord(:,2,1),'k');
plot3(gord(:,3,end),gord(:,1,end),gord(:,2,end),'k');
plot3(tempz(1,:),tempx(1,:),tempy(1,:),'k');
plot3(tempz(end,:),tempx(end,:),tempy(end,:),'k');

%Plot p0 coordinates (start points)
plot3(p0(:,3),p0(:,1),p0(:,2),'-r.');

%Plot p1 coordinates (end points)
plot3(p1(:,3),p1(:,1),p1(:,2),'-b.');

hold off;

view(-45,20); %Isometric view
axis tight;

%% Write FLUENT Journal File
fprintf('Writing FLUENT journal file...\n');

%The FLUENT journal file has to be written in a particular way to properly
%generate the line-surfaces and export the correct data.  Examples of each
%section are given below
```

```
%Specify the Matlab and Fluent export names based on dataname
matlabex=['NormGenex-' csnum '.jou'];
fluentex=['SARGEBLin-' csnum];

%Create file to write based on above filename
fid=fopen([workdir csnum '/' matlabex],'w');

%FLUENT Command: compile linex.c UDF in libudf
%NOTE - The linex.c UDF must be in the same folder as the case and data
%files for it to properly compile
fprintf(fid,'define/user-defined/compiled-functions\n');
fprintf(fid,'compile\n');
fprintf(fid,'"libudf"\n');
fprintf(fid,'yes\n');
fprintf(fid,'"y"\n');
fprintf(fid,'"linex.c"\n');
fprintf(fid,'""\n');
fprintf(fid,'"n"\n\n');

%FLUENT Command: load linex.c UDF
fprintf(fid,'define/user-defined/compiled-functions\n');
fprintf(fid,'load\n');
fprintf(fid,'"libudf"\n\n');

%FLUENT Command: load linexscheme.scm scheme
%NOTE - The linexscheme.scm scheme must be in the same folder as the case
%and data files for it to properly load
fprintf(fid,'(load ''linexscheme.scm)\n\n');

%FLUENT Command: create line-surface
%EX:    /surface/line-surface
%       {line name} x1 y1 z1 x0 y0 z0 (repeat as necessary)
for i=1:size(p0,1)
    fprintf(fid,'/surface/line-surface\n');
    fprintf(fid,'line%d %0.15E %0.15E %0.15E %0.15E %0.15E %0.15E\n\n', ...
        i,p0(i,:),p1(i,:));
end

%FLUENT Command: export flow field data (ASCII format) along line-surfaces
%using the linename scheme
%NOTE - This scheme calls a UDF (linex.c) that writes the location and
%flowfield data of the points along the line-surface in addition to the
%number of points along the line.  Every new set of line-surface data is
%appended to the same text file.
%EX:     (idnumb '{line name})

for i=1:size(p0,1)
    fprintf(fid,'(idnumb ''line%d ''%s)\n',i,fluentex);
end

fprintf(fid,'\n');

%FLUENT Command: delete line-surface
%EX:    /surface/delete-surface
%       {line names}
for i=1:size(p0,1)
    fprintf(fid,'/surface/delete-surface\n');
    fprintf(fid,'line%d\n\n',i);
end
```

```
%FLUENT Command: unload linex.c UDF
fprintf(fid,'define/user-defined/compiled-functions\n');
fprintf(fid,'unload\n');
fprintf(fid,'"libudf"');

%Close the file
fclose('all');

fprintf('Done.\n\n');

%Display the FLUENT journal filename
fprintf('FLUENT journal filename --------> %s\n',matlabex);

%Display the FLUENT ASCII export filename
fprintf('FLUENT ASCII export filename ---> %s\n\n',fluentex);
```

D-5: Linex Scheme File

The Linex scheme file passes information specified in the NormGen journal file to the Linex UDF and then runs the UDF.

```
(if (not (rp-var-object 'line_id))(rp-var-define 'line_id 999 'int #f))
(if (not (rp-var-object 'line_name))(rp-var-define 'line_name 'axis →
   'string #f))
(if (not (rp-var-object 'file_name))(rp-var-define 'file_name 'axis →
   'string #f))

(define (idnumb idname filename)
   (let ()
      (rpsetvar 'line_id (surface-name->id idname))
      (rpsetvar 'line_name idname)
      (rpsetvar 'file_name filename)
      (ti-menu-load-string "define/user-defined/execute-on-demand →
         \"linex::libudf\"")
))
```

D-6: Linex UDF

The Linex UDF writes the boundary-layer data, including the spatial coordinates, velocity components, pressure, temperature, and density, to a text file. This code is run for each data line generated by the NormGen journal file. The number of points along the current data line is also written to the text file.

```
#include "udf.h"
#include "surf.h"
#include "cxsurf.h"
#include "cxiface.h"
#include "dx.h"

DEFINE_ON_DEMAND(linex)
{
   Surface *s;
   Surf_Point *p;
   Domain *d = Get_Domain(1);
   int i, j, k, l,index, nw,nfp;
   real value, x, y, z;
   char *what;
   char *datafn;
   char *filename;
   int line_id;
   FILE *fp;
   double *x_arr,*y_arr,*z_arr,*xv_arr,*yv_arr,*zv_arr,*p_arr,*t_arr,*r_arr;

   /* Store information from Linex scheme (capitalize 'sargebl' and →
      'r'  in filename */
   filename = RP_Get_String("file_name");
   filename[0] = 'S';
   filename[1] = 'A';
   filename[2] = 'R';
   filename[3] = 'G';
   filename[4] = 'E';
   filename[5] = 'B';
   filename[6] = 'L';
   filename[12] = 'R';
   datafn = RP_Get_String("line_name");
   line_id = RP_Get_Integer("line_id");

   /* Open text file to write to (append file so as not to overwrite →
      existing data */
   fp = fopen(filename,"a+");

   /* Specify data line */
   s = SurfaceList+line_id;

   /* Number of points along data line */
   i = s->np;

   /* Allocate memory to store all the variables */
   x_arr = (double *) malloc(i*sizeof(double));
   y_arr = (double *) malloc(i*sizeof(double));
   z_arr = (double *) malloc(i*sizeof(double));
```

```
xv_arr = (double *) malloc(i*sizeof(double));
yv_arr = (double *) malloc(i*sizeof(double));
zv_arr = (double *) malloc(i*sizeof(double));
p_arr = (double *) malloc(i*sizeof(double));
t_arr = (double *) malloc(i*sizeof(double));
r_arr = (double *) malloc(i*sizeof(double));

/* String name */
what = "x-velocity";

/* Fill all the nodes with values of "what" */
Node_Function_Values(d,what);
p = s->points;

/* Surface_Value(p) returns value of "what" at the node */
/* Extract coordinates and x-velocity */
for(i=0;i<s->np;i++)
{
   x_arr[i] = Get_Surface_Point_Coord(p,X_DIM);
   y_arr[i] = Get_Surface_Point_Coord(p,Y_DIM);
   z_arr[i] = Get_Surface_Point_Coord(p,Z_DIM);
   value = Surface_Value(p);
   xv_arr[i] = value;
   p += 1;
}

/* Extract y-velocity */
what = "y-velocity";
Node_Function_Values(d,what);
p = s->points;
for(i=0;i<s->np;i++)
{
   value = Surface_Value(p);
   yv_arr[i] = value;
   p += 1;
}

/* Extract z-velocity */
what = "z-velocity";
Node_Function_Values(d,what);
p = s->points;
for(i=0;i<s->np;i++)
{
   value = Surface_Value(p);
   zv_arr[i] = value;
   p += 1;
}

/* Extract pressure */
what = "pressure";
Node_Function_Values(d,what);
p = s->points;
for(i=0;i<s->np;i++)
{
   value = Surface_Value(p);
   p_arr[i] = value;
   p += 1;
}

/* Extract density */
```

```
   what = "density";
   Node_Function_Values(d,what);
   p = s->points;
   for(i=0;i<s->np;i++)
   {
      value = Surface_Value(p);
      r_arr[i] = value;
      p += 1;
   }

   /* Extract temperature */
   what = "temperature";
   Node_Function_Values(d,what);
   p = s->points;
   for(i=0;i<s->np;i++)
   {
      value = Surface_Value(p);
      t_arr[i] = value;
      p += 1;
   }

   /* Write the number of points to the output file*/
   fprintf(fp,"%5d\n",s->np);

   /* Write the values to the output file*/
   for(i=0;i<s->np;i++)
   {
      fprintf(fp,"% 22.15E % 22.15E % 22.15E",x_arr[i],y_arr[i], →
         z_arr[i]);
      fprintf(fp,"% 22.15E",xv_arr[i]);
      fprintf(fp,"% 22.15E",yv_arr[i]);
      fprintf(fp,"% 22.15E",zv_arr[i]);
      fprintf(fp,"% 22.15E",p_arr[i]);
      fprintf(fp,"% 22.15E",r_arr[i]);
      fprintf(fp,"% 22.15E",t_arr[i]);
      fprintf(fp,"\n");
   }
   fprintf(fp,"\n\n");

   /* Close text file */
   fclose(fp);
}
```

D-7: SARGEBL

The SARGEBL code conditions the extracted boundary-layer data and writes the results to a mean flow file that will be used by LASTRAC to complete boundary-layer stability calculations. Like the NormGen code, SARGEBL was broken into functions. They will again be given in the order that they are called by SARGEBL.

```
%% SARGEBL - MWR 11 July 2011
%This code takes boundary layer data exported from FLUENT, processes it,
%and writes a meanflow file necessary to run boundary layer calculations in
%LASTRAC.  The following tasks are performed:
    %1) Load necessary data generated by NormGen
    %2) Read FLUENT flowfield data
    %3) Generate boundary layer profile data necessary for LASTRAC
    %4) Generate boundary layer profile properties necessary for LASTRAC
    %5) Write LASTRAC meanflow file
%The tasks performed by this program feeds directly into LASTRAC, which
%will complete all stability calculations.  A LASTRAC input file will need
%to be generated seperately.

%Variables:
    %workdir = working directory for all imports and exports related to run
    %csnum = case and section numbers (should match SBL input file suffix)
    %normgendata = name of NormGen data to import
    %pfs = simulation free stream pressure
    %Tfs = simulation free stream temperature
    %side = variable signifying the suction or pressure side of the section

%(See functions for all other variable definitions)

%Units (unless specified otherwise):
    %length = m
    %velocity = m/s
    %pressure = Pa
    %density = kg/m^3
    %temperature = K
    %kinematic viscosity = m^2/s

%% Clear and clc
clear;
clc;
close all;

%% Start Log
[~,~]=system('rm SARGEBL.txt');
diary('SARGEBL.txt');

%% Start Message
fprintf('-----------------\n---- SARGEBL ----\n-----------------\n');

fprintf('Matt Roberts 2011\n\n\n\n');
pause(1);
Working Directory
```

```
workdir='~/MATLAB/LASTRACFILES/';

%% Load NormGen Data File
fprintf('--- LOAD NORMGEN DATA ---\n\n');

fprintf('Working directory ---> %s\n\n',workdir);

%Prompt user to enter the case and section number
csnum=input('Enter the case and section number (e.g. 31R-234):\n-->> ','s');
fprintf('\n');

fprintf('File directory ------> %s\n\n',[workdir csnum '/']);

%Create NormGen data filename
normgendata=['NormGendata-' csnum '.mat'];

fprintf('Loading %s...\n',normgendata);

%Load NormGen data - see NormGen.m for contents of normgendata
load([workdir csnum '/' normgendata]);

fprintf('Done.\n\n');

fprintf('-------------------------\n\n\n');
pause(1);

%% Read FLUENT flowfield data - readdata.m
fprintf('--- READ FLOWFIELD DATA ---\n\n');

%Read BLP data from FLUENT - see description in readdata.m
[blcon]=readdata(seccor,workdir,csnum,fluentex);

fprintf('---------------------------\n\n\n');
pause(1);

%% Generate BLP Data - bldata.m
fprintf('--- GENERATE BLP DATA ---\n\n');

%Generate BLP data by reformatting conditioned data - see description in
%bldata.m
[bl,wparam,eparam]=bldata(blcon,xdir,ydir,zdir);

fprintf('See Figure 1.\n\n');

fprintf('-------------------\n\n\n');
pause(1);

%% Generate BLP Properties - blprop.m
fprintf('--- GENERATE BLP PROPERTIES ---\n\n');

%Generate BLP properties using the BLP edge parameters - see description in
%blprop.m
[blp,blpropp,bls,blprops]=blprop(bl,wparam,eparam,chord,adjseccor,K);

fprintf('See Figure 2 for pressure side BLP velocity profiles.\n');
fprintf('See Figure 3 for suction side BLP velocity profiles.\n');
fprintf('See Figure 4 for reference length and Reynolds number plots.\n\n');

fprintf('-------------------------------\n\n\n');
pause(1);
```

```
%% Write LASTRAC Meanflow File - statdelete.m, lastracmff.m
fprintf('--- WRITE LASTRAC MEANFLOW FILE ---\n\n');

%Prompt user to specify pressure or suction side
side=input('Enter the section side (s=suction, p=pressure):\n-->> ','s');

fprintf('\n');

%Prompt user to specify the station(s) immediately after an x array
%SplineFit error message
badstat=input('Enter the bad stations ([1 2 3]):\n-->> ');

if side=='p'
    %Delete stations if necessary - see description in statdelete.m
    [blp,blpropp]=statdelete(badstat,blp,blpropp);
elseif side=='s'
    %Delete stations if necessary - see description in statdelete.m
    [bls,blprops]=statdelete(badstat,bls,blprops);
else
    error('ERROR - Enter either s for suction of p for pressure');
end

fprintf('\n');

%Use standard atmosphere conditions to set free stream pressure (Pa)
%NOTE - The standard atmosphere pressure was set in NormGen
pfs=airprop(2);
fprintf('The free stream pressure is %0.0f Pa.\n\n',pfs);

%Use standard atmosphere conditions to set free stream temperature (K)
%NOTE - The standard atmosphere temperature was set in NormGen
Tfs=airprop(4);
fprintf('The free stream pressure is %0.2f K.\n\n',Tfs);

if side=='p'
    %Write pressure side meanflow file - see description in lastracmff.m
    [newfd]=lastracmff(blp,blpropp,pfs,Tfs,csnum,side,workdir);
elseif side=='s'
    %Write suction side meanflow file - see description in lastracmff.m
    [newfd]=lastracmff(bls,blprops,pfs,Tfs,csnum,side,workdir);
else
    error('ERROR - Enter either s for suction of p for pressure');
end

fprintf('------------------------------------\n\n\n');
pause(1);

%% Finish message
fprintf('-----------------\n--- GO STATE! ---\n-----------------\n');

%% Store Log
diary off
[~,~]=system(['cp SARGEBL.txt ' workdir csnum '/' newfd ...
    '/SARGEBL-' csnum '.log']);
[~,~]=system('rm SARGEBL.txt');
```

%% FUNCTION: SARGEBL.readdata - MWR 29 July 2011

```
%This function reads the FLUENT flowfield data extracted along flowfield
%data lines and stores it in a structure.  Before being returned, the data
%is filtered and the end of the data lines are checked for inconsistent
%information (jumps in the flowfield data).  A number of statistics are
%reported for user knowledge and can be found below in print statements.

%Variables:
    %fid = ID of file that is being read from
    %c1 = missing surface point counter
    %curr = current line being read
    %ppbl = points per boundary layer
    %blcon = structure containing boundary layer data
    %yn = wall-normal distance
    %newrow = appending data if surface point is missing
    %c2 = BLP end point removal counter
    %velend = total velocity at last point of a BLP (furthest from wall)
    %velendm1 = total velocity at second-to-last point of a BLP
    %newppbl = updated ppbl after data filtering and endpoint removal
    %ynmax = maximum wall-normal distance for a given BLP

function [blcon]=readdata(seccor,workdir,csnum,fluentex)

fprintf('Reading and conditioning flowfield data from FLUENT...\n');

%Open the FLUENT flowfield data file
fid=fopen([workdir csnum '/' fluentex],'r');

%Loop counter
i=1;

%Missing surface point counter
c1=0;

while ~feof(fid) && i<500
    %Scan number of points in BLP
    curr=fscanf(fid,'%d',1);

    if ~isempty(curr)
        %Store number of points in BLP
        ppbl(i)=curr;

        %Read BLP data based on above number of points
        %The column order is as follows:
        %[x-coor y-coor z-coor Ug Vg Wg p rho T]
        blcon(i).data=fscanf(fid,'%E %E %E %E %E %E %E %E %E',[9 ppbl(i)])';

        %Calculate wall-normal distance
        yn=sqrt((blcon(i).data(:,1)-seccor(i,1)).^2+ ...
            (blcon(i).data(:,2)-seccor(i,2)).^2+ ...
            (blcon(i).data(:,3)-seccor(i,3)).^2);

        %Append wall-normal distance and sort the data based on wall-normal
        %distance (starting at the wall and moving away)
        blcon(i).data=sortrows([yn blcon(i).data],1);

        %Remove wall normal distance for consistency
        blcon(i).data(:,1)=[];
        %Check and see if first point contains zero-velocity information
        %NOTE - It is necessary to check if the wall point was not in the
        %flowfield domain for whatever reason.  If it is not, the
        %corresponding wall point from NormGen is used, the velocity
        %components are set to zero, and p, T, and rho from the next point
        %are used for the wall point.
        if blcon(i).data(1,4:6)~=[0 0 0]
            newrow=[seccor(i,:) 0 0 0 blcon(i).data(1,7:9)];
            blcon(i).data=[newrow;blcon(i).data];
            c1=c1+1;
        end

        %Remove points near the end since it may contain "incorrect"
        %information (depending on if neighboring zones were deactivated)
        %NOTE - The tolerance has been set to 2.5 m/s difference between the
        %velocity magnitudes of the last and second-to-last points

        %End point removal counter
        c2=0;

        %Velocity magnitude at last point
        velend=norm(blcon(i).data(end,4:6));

        %Velocity magnitude at second to last point
        velendm1=norm(blcon(i).data(end-1,4:6));

        while abs(velend-velendm1)>2.5 && c2<5
            %Remove last row
            blcon(i).data(end,:)=[];

            %Recalculate velocity magnitudes of last and 2nd-to-last points
            velend=norm(blcon(i).data(end,4:6));
            velendm1=norm(blcon(i).data(end-1,4:6));

            %Update counter
            c2=c2+1;

            %Protect against too many points being removed
            if c2 == 5
                error('ERROR - too many points being removed from BLP');
            end
        end

        %Filter points - see description in pntfilter.m
        blcon(i).data=pntfilter(blcon(i).data,i);

        %Calculate new ppbl for comparison and troubleshooting reasons
        newppbl(i)=size(blcon(i).data,1);

        %Calculate the max BLP wall-normal distance
        ynmax(i)=norm(blcon(1).data(end,1:3)-blcon(1).data(1,1:3));

        i=i+1;
    end
end

fprintf('Done.\n\n');

%Print number of missing surface points
if c1==1
```

```
    fprintf(['There was %d missing surface point ’ ...
        ‘that had to be replaced.\n\n'],c1);
else
    fprintf(['There were %d missing surface points ’ ...
        ‘that had to be replaced.\n\n'],c1);
end

%Print unfiltered and filtered minimum, maximum, and average points per BLP
fprintf('      BLP Statistics     \n\n');
fprintf('    Unfiltered   Filtered\n');
fprintf('Min: %6.d %10.d\n',min(ppbl),min(newppbl));
fprintf('Avg: %6.1f %10.1f\n',mean(ppbl),mean(newppbl));
fprintf('Max: %6.d %10.d\n\n',max(ppbl),max(newppbl));

fprintf('Wall-Normal Distance Statistics\n\n');
fprintf('Min: %15.2fmm\n',1000*min(ynmax));
fprintf('Avg: %15.2fmm\n',1000*mean(ynmax));
fprintf('Max: %15.2fmm\n\n',1000*max(ynmax));
```

%% FUNCTION: SARGEBL.pntfilter - MWR 26 July 2011

```
%This function takes the data for a single BLP and filters out information
%based on a proximity test with regard to wall-normal distance.  Ideally,
%points exported from FLUENT would only cross faces of cells that are more
%or less parallel to the surface.  However, the data lines sometime cross
%cell faces that are more or less perpendicular to the surface and
%introduce data in between the parallel cell faces.  This "extra" data
%ruins the uniform growth rate of the data spacing.  By removing this
%"extra" data, the uniform growth of the point spacing can be
%re-established.  This generates a much cleaner data set, where BLP's have
%more similar point counts and wall-normal distance is more suitable for
%LASTRAC.

%Variables:
    %tempbl = matrix containing temporary BLP data inputted by readdata.m
    %tol1 = point filter tolerance that sets how much data is removed
    %tol2 = point count tolerance that limits on how much data is removed
    %c1 = counter for number of points removed
    %c2 = counter used to guard against an infinite while loop
    %pcount = number of data points before filter is employed
    %yn = wall-normal distance
    %dm1 = distance between wall-normal points i and i-1
    %dm2 = distance between wall-normal points i-1 and i-2
    %message = error message displayed if too many point are removed

function [tempbl]=pntfilter(tempbl,station)

%Specify tolerance 1, used to filter points
tol1=.5;

%Specify tolerance 2, used big changes in the size of tempbl
tol2=.6;

%Pre-allocate counter 1, used for troubleshooting filtering process
c1=0;

%Pre-allocate counter2, used to guard against an infinite while loop
```

```
c2=0;

%Record pre-filtered point count
pcount=size(tempbl,1);

%Calculate wall normal distance from xyz-coordinates
yn=sqrt(sum((tempbl(:,1:3)-(ones(size(tempbl,1),1)*tempbl(1,1:3))).^2,2));

%Append tempbl with yn
tempbl=[yn tempbl];

%Begin the filtering process at the third point from the wall
i=3;

%Filter points
while i<size(tempbl,1) && c2<1000
    %Length difference between i and i-1
    dm1=tempbl(i,1)-tempbl(i-1,1);
    %Length difference between i-1 and i-2
    dm2=tempbl(i-1,1)-tempbl(i-2,1);

    %Filter, using tolerance 1
    if dm1<tol1*dm2
        tempbl(i,:)=[];
        c1=c1+1;
    else
        i=i+1;
    end

    c2=c2+1;
end

%Remove wall normal distance from tempbl before returning it
tempbl(:,1)=[];

%Check number of points in tempbl
%NOTE - if the number of points in the BLP is too small compared to the
%unfiltered count (pcount), an error message stops the code and displays
%the station with the problem
if size(tempbl,1)<tol2*pcount
    message = ['ERROR - Too many points were removed from station '...
        int2str(station) '.'];
    error(message);
end
```

%% FUNCTION: SARGEBL.bldata - MWR 14 July 2011

```
%This function take the conditioned data and reformats it for a LASTRAC
%meanflow file.  For each profile, the point coordinates are reformatted
%into wall-normal distance and the global velocities are reformatted into
%local velocities (given by xdir, ydir, and zdir).  These and the other
%properties (pressure, density, and temperature) are stored in 3D matrix
%with the same basic format as blcon. Index 2 variable order is listed
%below.  All BLP data is still dimensional.  All wall parameters (from
%blcon) are returned for use in BLP property calculations.  Some sample
%velocity profiles are plotted as a check using the function plotuvw.
```

```
%Variables:
    %yn = wall-normal distance, calculated from BLP point coordinates
    %U = local x-velocity, calculated from global velocities and xdir
    %V = local y-velocity, calculated from global velocities and ydir
    %W = local z-velocity, calculated from global velocities and zdir
    %p = pressure, copied from blcon
    %rho = density, copied from blcon
    %T = temperature, copied from blcon
    %bl = structure containing BLP data from all above variables
    %wparam = matrix of wall parameters for all BLP's
    %eparam = matrix of edge parameters for all BLP's
    %nue = edge kinematic viscosity for all BLP's (Sutherland's Law)
    %LEid = leading edge profile number
    %profiles = list of profile numbers to plot

function [bl,wparam,eparam]=bldata(blcon,xdir,ydir,zdir)

%Reformat data message
fprintf('Reformat boundary layer profile data.\n');

%Progress message (followed by periods which are printed below)
fprintf('Reformatting');

%Data counter
c1=1;

%Progress period distribution (10 total)
progpnt=round(linspace(0,size(blcon,2),11));
progpnt(1)=[];

%Reformat data for all BLP's
for i=1:size(blcon,2)
    %Reformat coordinates into wall-normal distance. This is accomplished
    %by setting the wall node to [0 0 0] and finding the magnitude of the
    %coodinates for each point
    yn=sqrt((blcon(i).data(:,1)-blcon(i).data(1,1)).^2+ ...
        (blcon(i).data(:,2)-blcon(i).data(1,2)).^2+ ...
        (blcon(i).data(:,3)-blcon(i).data(1,3)).^2);

    %Reformat global velocities into local velocities.  This is
    %accomplished by dotting the global velocities with the local
    %coordinate system directions.
    U=dot(blcon(i).data(:,4:6),ones(size(blcon(i).data,1),1)*xdir(i,:),2);
    V=dot(blcon(i).data(:,4:6),ones(size(blcon(i).data,1),1)*ydir(i,:),2);
    W=dot(blcon(i).data(:,4:6),ones(size(blcon(i).data,1),1)*zdir(i,:),2);

    %Copy the pressure, density, and temperature
    p=blcon(i).data(:,7);
    rho=blcon(i).data(:,8);
    T=blcon(i).data(:,9);

    %Create reformatted BLP data
    %The column order is as follows:
    %[wall-normal-distance U V W temperature pressure density]
    bl(i).data(:,:)=[yn U V W T p rho];

    %Print progress period
    if i==progpnt(c1);
        fprintf('.');
        c1=c1+1;
    end

end

fprintf('\n\n');

%% Record BLP wall and edge parameters
fprintf('Storing wall and edge parameters...\n');

%Store wall parameters
%The column order is as follows:
%[x-coor y-coor z-coor Ug Vg Wg p rho T]
for i=1:size(blcon,2)
    wparam(i,:)=blcon(i).data(1,:);
end


%Store edge parameters
%The column order is as follows:
%[wall-normal-distance U V W T p rho nu]
for i=1:size(bl,2)
    eparam(i,:)=bl(i).data(end,:);
end

%Calculate edge kinematic viscosity using Sutherland's Law and edge density
%NOTE - The coefficients used are as follows:
    %muref = 1.716*10^-5 kg/m*s
    %Tref = 273.11 K
    %S = 110.56 K
nue=(1.716E-5*(eparam(:,5)/273.11).^(3/2)...
    .*((273.11+110.56)./(eparam(:,5)+110.56)))./eparam(:,7);

%Append edge kinematic viscosity to existing edge parameters
eparam=[eparam nue];

fprintf('Done.\n\n');

%% Plot some boundary layer velocity profiles
%Set up locations on section to plot
%NOTE - Change the coefficients in the 'round' statements to modify the
%intermediate locations.  By default, the aft-most points on the upper and
%lower sides and the leading edge point of the section are plotted.
LEid=find(wparam(:,3)==max(wparam(:,3)));
profiles=[1 ...
    round(.2*LEid)...
    round(.5*LEid)...
    LEid...
    LEid+round(.45*(size(bl,2)-LEid))...
    LEid+round(.8*(size(bl,2)-LEid))...
    size(bl,2)];

%Allocate figure
figure(1);

%Plot velocity profiles - see description in plotuvw.m
plotuvw(profiles,wparam,bl,1);

%Figure settings
scrnsize=get(0,'ScreenSize');
position=[.4*scrnsize(3) .6*scrnsize(4) .3*scrnsize(3) .4*scrnsize(4)];
```

```
set(gcf,'Units','pixels','OuterPosition',position,'Color','w');
```

%% FUNCTION: SARGEBL.plotuvw - MWR 14 July 2011

```
%This function takes BLP data and plots the U, V, and W velocity profiles
%for specifed profiles.  The positions of these profiles on the section are
%shown.

%Variables:
    %cmnum = color map number - sets desired colormap
    %colormap = color map for all subplots
    %maxyn = maximum wall-normal distance out of all BLP's

function plotuvw(profiles,wparam,bl,cmnum)

%Define color map
if cmnum==1
    colormap=lines(length(profiles));
elseif cmnum==2
    colormap=jet(length(profiles));
end

%Find maximum wall-normal distance for y-axis scale
for i=1:length(profiles)
    maxyn(i)=bl(profiles(i)).data(end,1);
end
maxyn=max(maxyn);

%Plot locations of BLP's on section
subplot(2,2,1)
plot(wparam(:,3),wparam(:,2),'Color','k');
hold on
for i=1:length(profiles)
    plot(wparam(profiles(i),3),wparam(profiles(i),2),'.',...
        'MarkerEdgeColor',colormap(i,:),'MarkerFaceColor',colormap(i,:));
end
axis equal;
axis 'auto x';
set(gca,'Xdir','reverse');
xlabel('Boundary Layer Profile Locations');
hold off;

%Plot U profiles
subplot(2,2,2)
hold on;
for i=1:length(profiles)
    plot(bl(profiles(i)).data(:,2),bl(profiles(i)).data(:,1), ...
        'Color',colormap(i,:));
end
xlim('auto');
ylim([0 1.05*maxyn]);
axis 'auto x';
xlabel('Streamwise Velocity (m/s)');
hold off;

%Plot V profiles
subplot(2,2,3)
```

```
hold on;
for i=1:length(profiles)
    plot(bl(profiles(i)).data(:,3),bl(profiles(i)).data(:,1), ...
        'Color',colormap(i,:));
end
xlim('auto');
ylim([0 1.05*maxyn]);
axis 'auto x';
xlabel('Wall-Normal Velocity (m/s)');
hold off;

%Plot W profiles
subplot(2,2,4)
hold on;
for i=1:length(profiles)
    plot(bl(profiles(i)).data(:,4),bl(profiles(i)).data(:,1), ...
        'Color',colormap(i,:));
end
xlim('auto');
ylim([0 1.05*maxyn]);
axis 'auto x';
xlabel('Spanwise Velocity (m/s)');
hold off;
```

%% FUNCTION: SARGEBL.blprop - MWR 14 July 2011

```
%This function returns BLP data and BLP properties for the pressure and
%suction sides of the glove section.  The first task is to find the
%location of the attachment line.  Once that is complete, all original BLP
%data stored in the bl structure, the BLP wall and edge parameters, the
%section adjusted coordinates, and the section curvature values can be
%seperated accordingly.  From here, the BLP properties can be calculated
%and stored.  Plots of the velocity components vs wall-normal distance and
%a plot of the reference length and Reynolds number vs body-fitted distance
%are included for user inspection of the process.

%Variables:
    %Unw = Matrix with BLP ID and streamwise (U) velocity near the wall
    %ALsid = Attachment line BLP ID (towards the suction side)
    %tempdiff = temporary adjusted coordinates node-to-node difference

    %PRESSURE/SUCTION SIDE VARIABLES (Replace _ with p or s respectively)
    %bl_ = BLP data
    %wparam_ = wall parameters
    %eparam_ = edge parameters
    %adjseccor_ = adjusted section coordinates
    %K_ = curvature of adjusted section coordinates
    %profiles_ = profile ID's to plot BLP velocities
    %istat_ = station number
    %npts_ = number of points per profile
    %h1_ = streamwise curvature metric used in the x_ calculation
    %x_ = body-fitted streamwise distance
    %rl_ = reference length
    %Rel_ = Reynolds number based on reference length
    %drdx_ = x/c location of airfoil (not body-fitted)
    %blprop_ = matrix of all BLP properties (see order below)
```

```
function [blp,blpropp,bls,blprops]=blprop(bl,wparam,eparam,chord,adjseccor,K)

%% Locate the attachment line and separate BLP's
%Record the profile number and the near-wall (closest point not already on
%the wall) streamwise velocity (U)
for i=1:size(bl,2)
    Unw(i,:)=bl(i).data(2,2);
end

%Find BLP ID of first suction-side point
%Check velocity sign
velsign=sign(Unw);

%NOTE - The attachment line is found by determining when the near-wall
%streamwise velocity switches sign.  A positive U near the LE signals the
%profile is on the suction side and a negative U near the LE signals the
%profile is on the pressure side. The minimum function will find the first
%-1 it comes to (the first point on the pressure side).  This check does
%not include the 10 aft-most points on either side since they will never be
%near the LE and the flow may be seperated in that region, leading to a
%spurious attachment line location.

for i=1:length(velsign)
    if i<10 && velsign(i)==-1
        fprintf('WARNING - Flow seperation on suction side.\n\n');
    elseif i>=10 && i<=length(velsign)-9 && velsign(i)-velsign(i-1)==-2
        ALsid=i-1;
        fprintf('The attachment line is at wall-point number %d.\n\n',ALsid);
    elseif i>length(velsign)-9 && velsign(i)==1
        fprintf('WARNING - Flow seperation on pressure side.\n\n');
    end
end

%Check ALsid by finding the ID corresponding to the maximum wall pressure
%If the difference between these two ID's is too large, there is most
%likely an error
if abs(find(wparam(:,7)==max(wparam(:,7)))-ALsid)>1
    error('ERROR - The attachment line location is not correct')
end

%% Store BLP data, wall and edge parameters - Pressure Side
fprintf('Storing data and calculating properties for the pressure side...\n');

%Store existing pressure side data and parameters
blp=bl(ALsid+1:end);
wparamp=wparam(ALsid+1:end,:);
eparamp=eparam(ALsid+1:end,:);
adjseccorp=adjseccor(ALsid+1:end,:);
Kp=K(ALsid+1:end,:);

%Reverse streamwise and spanwise velocities so that U points aft (more or
%less) and W points inboard (to keep coordinate system right handed)
for i=1:size(blp,2)
    blp(i).data(:,2)=-blp(i).data(:,2);
    blp(i).data(:,4)=-blp(i).data(:,4);
end
eparamp(:,2)=-eparamp(:,2);
eparamp(:,4)=-eparamp(:,4);

%Plot some velocity profiles
```

```
profilesp=1:1:size(blp,2);

%Allocate figure
figure(2);

%Send information to plotuvw - see description in plotuvw.m
plotuvw(profilesp,wparamp,blp,2);

%Figure settings
scrnsize=get(0,'ScreenSize');
position=[.7*scrnsize(3) .6*scrnsize(4) .3*scrnsize(3) .4*scrnsize(4)];
set(gcf,'Units','pixels','OuterPosition',position,'Color','w');

%% Calculate BLP properties - Pressure Side
%Create station number list for pressure side
istatp=[1:size(blp,2)]';

%Create 'number of points' list for pressure side
for i=1:size(blp,2)
    nptsp(i,:)=size(blp(i).data,1);
end

%Calculate streamwise coordinates (body-fitted) for the pressure side
%NOTE - This calculation is performed on the adjusted coordinates
xp=cumsum(sqrt(sum([0 0 0;diff(adjseccorp)].^2,2)));

%Calculate the normalized length scales for the pressure side
%NOTE - The first length scale is artificially determined by  halving the
%second length scale on the list. This prevents dividing by zero later
rlp=sqrt((eparamp(:,8).*xp(:,:))./abs(eparamp(:,2)));
rlp(1,:)=.5*rlp(2,:);

%Calculate the Reynolds numbers for the pressure side using the pressure
%side length scales
Relp=(abs(eparamp(:,2)).*rlp)./eparamp(:,8);

%Calculate the x/c location for the pressure side
%NOTE - The user needs to make sure the chord length is correct.  This step
%is performed in NormGen.
drdxp=abs(wparamp(:,3)-max(wparamp(:,3)))/chord;

%Combine all properties into a single matrix
%The property order is as follows:
%[station number, number of points, body-fitted streamwise location,
%reference length, Reynolds number, streamwise curvature, transverse cone
%radius, x/c location, edge temperature, edge velocity, and edge density]
%NOTE - The transverse cone radius is set to zero since it is not
%applicable for swept wing cases
blpropp=[istatp nptsp xp rlp Relp Kp zeros(size(blp,2),1) drdxp...
    eparamp(:,5) eparamp(:,2) eparamp(:,7)];

fprintf('Done.\n\n');

%% Store BLP data, wall and edge parameters - Suction Side
fprintf('Storing data and calculating properties for the suction side...\n');

%Store suction side data and parameters
%NOTE - The profile order has been flipped so the first profile starts at
%the leading edge
bls=flipdim(bl(1:ALsid),2);
```

```
wparams=flipud(wparam(1:ALsid,:));
eparams=flipud(eparam(1:ALsid,:));
adjseccors=flipud(adjseccor(1:ALsid,:));
Ks=flipud(K(1:ALsid,:));

%Plot some velocity profiles
profiless=1:1:size(bls,2);

%Allocate figure
figure(3);

%Send information to plotuvw - see description in plotuvw.m
plotuvw(profiless,wparams,bls,2);

%Figure settings
scrnsize=get(0,'ScreenSize');
position=[.4*scrnsize(3) .2*scrnsize(4) .3*scrnsize(3) .4*scrnsize(4)];
set(gcf,'Units','pixels','OuterPosition',position,'Color','w');

%% Calculate BLP properties - Suction Side
%Create station number list for suction side
istats=[1:size(bls,2)]';

%Create 'number of points' list for suction side
for i=1:size(bls,2)
    nptss(i,:)=size(bls(i).data,1);
end


%Calculate streamwise coordinates (body-fitted) for the pressure side
%NOTE - This calculation is performed on the adjusted coordinates
xs=cumsum(sqrt(sum([0 0 0;diff(adjseccors)].^2,2)));

%Calculate the normalize length scales for the suction side
%NOTE - The first length scale is artificially determined by  halving the
%second length scale on the list. This prevents dividing by zero later
rls=sqrt((eparams(:,8).*xs(:,:))./abs(eparams(:,2)));
rls(1,:)=.5*rls(2,:);


%Calculate the Reynolds numbers for the suction side using the suction
%side length scales
Rels=(abs(eparams(:,2)).*rls)./eparams(:,8);

%Calculate the x/c location for the pressure side
%NOTE - The user needs to make sure the chord length is correct.  This step
%is performed in NormGen.
drdxs=abs(wparams(:,3)-max(wparams(:,3)))/chord;

%Combine all properties into a single matrix
%The property order is as follows:
%[station number, number of points, body-fitted streamwise location,
%reference length, Reynolds number, streamwise curvature, transverse cone
%radius, x/c location, edge temperature, edge velocity, and edge density]
%NOTE - The transverse cone radius is set to zero since it is not
%applicable for swept wing cases
blprops=[istats nptss xs rls Rels Ks zeros(size(bls,2),1) drdxs...
    eparams(:,5) eparams(:,2) eparams(:,7)];

fprintf('Done.\n\n');
```

```
%% Plot reference length and Reynolds number against BF-distance
%Allocate figure
figure(4);

%Plot reference length
subplot(2,1,1);
plot(blpropp(:,3),blpropp(:,4),'r',blprops(:,3),blprops(:,4),'b');
xlabel('Body-fitted Streamwise Distance (m)');
ylabel('Reference Length (m)');
legend('Pressure Side','Suction Side');

%Plot Reynolds number
subplot(2,1,2);
plot(blpropp(:,3),blpropp(:,5),'r',blprops(:,3),blprops(:,5),'b');
xlabel('Body-fitted Streamwise Distance (m)');
ylabel('Reynolds Number');
legend('Pressure Side','Suction Side');

%Figure settings
scrnsize=get(0,'ScreenSize');
position=[.7*scrnsize(3) .2*scrnsize(4) .3*scrnsize(3) .4*scrnsize(4)];
set(gcf,'Units','pixels','OuterPosition',position,'Color','w');
```

%% FUNCTION: SARGEBL.statdelete - MWR 12 September 2011

```
%This function deletes user specified data before writing the mean flow
%file.  Occasionally, LASTRAC will give the error "The input x array in
%SplineFit has duplicated elements".  By removing the station information
%immediately downstream of the stations with the error messages (there are
%generally two stations with this problem), this problem can be avoided.

%Variables:
    %newnum = list of new station numbers accounting for deleted stations

function [bl,blprop]=statdelete(badstat,bl,blprop)

%Sort bad station list to ensure they are in sequential order
badstat=sort(badstat);

%Remove bad stations
%NOTE - This loop takes into account the deletion of stations in the
%previous step, hence the -(i-1).
for i=1:length(badstat);
    bl(badstat(i)-(i-1))=[];
    blprop(badstat(i)-(i-1),:)=[];
end

%Renumber stations in blprop
newnum=(1:size(blprop,1))';
blprop(:,1)=[];
blprop=[newnum blprop];
```

%% FUNCTION: SARGEBL.lastracmff - MWR 15 July 2011

```
%The function writes a text file containing all the information necessary
%to create a LASTRAC meanflow file.  The parameters that are used for the
%entire simulation are generated first.  A text-based input file is then
%created containing all the data used to generate a LASTRAC meanflow
%file.  This input uses 64 significant-digit numbers.  LASTRAC requires a
%binary (unformatted) meanflow file, so the text-based file must first be
%converted by the FORTRAN code m2f.  Once this step has been completed, the
%meanflow file can be inputted into LASTRAC.  To review the meanflow file
%data, the formatted data is decoded by the FORTRAN code decode.f90.  Some
%system tasks are performed to store all files in a unique location.

%Variables:
    %title = filename of the formatted meanflow file
    %n_station = number of stations in BLP data for secton side
    %id = ID of first non-negative edge velocity
    %igas = gas type
    %iunit = specified units
    %Pr = free stream Prandtl number
    %nps = number of gas species
    %fid = ID of the file being written to
    %curdir = current directory
    %suffix = user-specified suffix for new directory
    %newfd = name of new file directory all files are stored in

function [newfd]=lastracmff(bl,blprop,pfs,Tfs,csnum,side,workdir)

%Create meanflow file title based on case, section, and section side
title=['SARGEBLmff-' csnum side '.out'];

fprintf('LASTRAC Meanflow file name ---> %s\n\n',title);

%Determine total number of stations
n_station=size(bl,2);

%Print station number of first non-negative edge velocity
[~,id]=max(sign(blprop(:,10)));
fprintf('The first non-negative edge velocity is at station %d.\n',id);

%Print number of stations
fprintf('There are %d stations.\n\n',n_station);

%Set gas type (1=perfect gas)
igas=1;

%Set units (1=SI and 0=imperial)
%NOTE - All stored data uses SI units
iunit=1;

%Calculate Prandtl number
%Use the free stream temperatue and interpolate off of data set shown below
%T=200, Pr=0.737
%T=250, Pr=0.720
Pr=((Tfs-200)/(250-200))*(.737-.720)+.720;

if Tfs>250 || Tfs<200
    error('ERROR - enter the correct temperature');
end
```

```
%Set number of gas species (0=perfect gas)
nps=0;

%% Write formatted meanflow file
%NOTE - See Chapter 4 of the LASTRAC User Manual for meanflow file format
%guide

%Write meanflow file message
fprintf('Write meanflow file.\n');

%Progress message (followed by periods which are printed below)
fprintf('Writing');

%Data counter
c1=1;

%Progress period distribution (10 total)
progpnt=round(linspace(0,size(bl,2),11));
progpnt(1)=[];

%Open precise file to write using the title as the filename
fid=fopen([workdir csnum '/' title],'w');

%Write global parameters
fprintf(fid,'%s\n',title);
fprintf(fid,'%d\n',n_station);
fprintf(fid,'%d %d %73.63E %73.63E %d\n\n',igas,iunit,Pr,pfs,nps);

%Write parameters and data for all stations
for i=1:size(bl,2)
    %Write station parameters
    fprintf(fid,'%d %d %73.63E %73.63E %73.63E %73.63E %73.63E %73.63E\n',...
        blprop(i,1),blprop(i,2),blprop(i,3),blprop(i,4),blprop(i,5),...
        blprop(i,6),blprop(i,7),blprop(i,8));
    fprintf(fid,'%73.63E %73.63E %73.63E\n',...
        blprop(i,9), blprop(i,10), blprop(i,11));

    %Write station data
    %NOTE - As per LASTRAC requirements, station data must be normalized in
    %the following manner: y(j)/rl, U(j)/u_e, V(j)/u_e, W(j)/u_e, T(j)/t_e,
    %and p(j)/(rho_e*u_e^2). See LASTRAC User Manual, Chpater 4 for a more
    %detailed explanation.
    for j=1:size(bl(i).data,1)

        fprintf(fid,'%73.63E %73.63E %73.63E %73.63E %73.63E %73.63E\n',...
            bl(i).data(j,1)/blprop(i,4),bl(i).data(j,2)/blprop(i,10),...
            bl(i).data(j,3)/blprop(i,10),bl(i).data(j,4)/blprop(i,10),...
            bl(i).data(j,5)/blprop(i,9),bl(i).data(j,6)/(blprop(i,11)*...
                blprop(i,10)^2));

    end

    %Print progress period to screen
    if i==progpnt(c1);
        fprintf('.');
        c1=c1+1;
    end

end
```

```
fprintf('\n\n');

%Close file
fclose(fid);

%% System commands
%Change to the file directory
curdir=pwd;

%Change to the working directory
cd([workdir csnum]);

%Ask user to add suffix to folder if desired
%NOTE - The user must add a dash at the beginning of the suffix if they
%want it to appear between the side variable and the suffix.  Hitting enter
%will store nothing in the suffix.
suffix=input('Enter a suffix for folder if desired:\n-->> ','s');
fprintf('\n');

%Create SARGEBL file directory name that includes the case and section
%numbers and suffix
newfd=[csnum side suffix];

%Create SARGEBL file directory, removing folder already named newfd first
[~,~]=system(['rm -r ' newfd]);
[~,~]=system(['mkdir ' newfd]);

%Copy meanflow file into SARGEBL directory and remove old file
[~,~]=system(['scp ' title ' ' newfd]);
[~,~]=system(['rm ' title]);

%Change to SARGEBL file directory
cd([workdir csnum '/' newfd]);

%Run T2B, converting formatted meanflow file to unformatted (binary) mean
%flow file
[~,~]=system(['T2B ' title]);

%Run decode, outputing unformatted meanflow file into a readable text file
[~,~]=system(['decode ' title ' decode-' csnum side]);

%Change back to SARGEBL directory
cd(curdir);

%Print file directory name
fprintf(['SARGEBL file directory ---> ' workdir csnum '/' newfd '\n\n']);
```

D-8: T2B

The T2B code is used internally by SARGEBL to convert the mean flow file from an ASCII format to a binary format. LASTRAC requires this format for all mean flow files.

```
program T2B
implicit none


! Declare variables
character*64 filename
character*64 trashchar
integer trashint, i1, maxi2
integer, allocatable :: i2(:)
real*8 trashreal
character*64 title
integer n_station, igas, iunit, nsp
real*8 prandtl, pfs
integer n, k
integer, allocatable :: istat(:), npts(:)
real*8, allocatable :: x(:), rl(:), Rel(:), xcurvt(:), radius(:), drdx(:), →
   t_e(:), u_e(:), rho_e(:)
real*8, allocatable :: y(:,:), u(:,:), v(:,:), w(:,:), t(:,:), p(:,:)

! User input for files
call getarg(1,filename)

! Open and read through file to get n_station and npts information
open(unit=10,file=filename,status='old')
read(10,*) trashchar
read(10,*) i1
allocate(i2(i1))
read(10,*) trashint
do n = 1, i1
   read(10,*) trashint, i2(n)
   read(10,*) trashreal
   do k = 1, i2(n)
      read(10,*) trashreal
   enddo
enddo
close(10)

! Set new maximum number of points
maxi2=maxval(i2)

! Open and read ASCII data (formatted)
open(unit=10,file=filename,status='old')

!Allocate matrices
allocate(istat(i1), npts(i1), x(i1), rl(i1), Rel(i1), xcurvt(i1), radius(i1), →
   drdx(i1), t_e(i1), u_e(i1), rho_e(i1))
allocate(y(i1,maxi2), u(i1,maxi2), v(i1,maxi2), w(i1,maxi2), t(i1,maxi2), →
   p(i1,maxi2))

! Read global parameters
read(10,*) title
read(10,*) n_station
read(10,*) igas, iunit, prandtl, pfs, nsp

! Read station parameters
do n = 1, n_station
   read(10,*) istat(n), npts(n), x(n), rl(n), Rel(n), xcurvt(n), radius(n), →
      drdx(n)
   read(10,*) t_e(n), u_e(n), rho_e(n)
   print*, i2(n)

   ! Read station data
   do k = 1, npts(n)
      read(10,*) y(n,k), u(n,k), v(n,k), w(n,k), t(n,k), p(n,k)
   enddo
   print*, ""
enddo

close(10)

! Open and write binary data (unformatted)
open(unit=20,file=filename,form='unformatted',status='replace')

! Write global parameters
write(20) title
write(20) n_station
write(20) igas, iunit, prandtl, pfs, nsp

! Write station parameters
do n = 1, n_station

   write(20) istat(n), npts(n), x(n), rl(n), Rel(n), xcurvt(n), radius(n), →
      drdx(n)

   ! Write station data
   write(20) t_e(n), u_e(n), rho_e(n)
   write(20) (y(n,k), k=1, npts(n))
   write(20) (u(n,k), k=1, npts(n))
   write(20) (v(n,k), k=1, npts(n))
   write(20) (w(n,k), k=1, npts(n))
   write(20) (t(n,k), k=1, npts(n))
   write(20) (p(n,k), k=1, npts(n))

enddo

close(20)

end
```

APPENDIX E

LASTRAC INPUT FILES

E-1: LST Crossflow Input File

The example LASTRAC input file given below contains the settings used to complete LST crossflow calculations.

```
// LST Crossflow Input File

   //General Parameters
   flow_type = bound_layer
   mflow_filename   = "SARGEBLmff-07R-234s.out"
   nonl_pse_calc    = false
   solution_type    = local_eig_solution

   //Marching and Station Parameters
   marching_method_2d     = along_station
   init_station           = 2
   final_station          = 151

   //Wave Parameters
   freq_unit = in_hertz_freq
   beta_unit = in_mm_beta
   freq      = 14*0.
   beta      = 1,2,3,3.5,4,4.5,5,6,7,8,9,10,12,14
   qp_approx = true

   //Grid Parameters
   grid_type              = wall_cluster
   ymax                   = 200
   ymax_glob_search       = 60
   num_normal_pts         = 101
   num_normal_pts_geig    = 41

   //Curvature Parameters
   strm_curvt       = false
   transv_curvt     = false

   //Order of Accuracy Parameters
   strm_order             = second_order
   wall_normal_order      = fourth_order

   //Memory Storage Parameter
   mflow_storage_type     = file_storage

   //Extrapolation Parameter
   use_extrap_mprof = true

   //Eigenvalue Search Parameter
   relax_type       = wall_temp

   //Wave Angle Parameters
   wave_ang_min = 65
   wave_ang_max = 99

   //Growth Rate Limit Parameter
   np_growth_rate_min     = -1.e-4
```

E-2: LST T-S Input File

The example LASTRAC input file given below contains the settings used to complete LST T-S calculations.

```
// LST Tollmien-Schlichting Input File

   //General Parameters
   flow_type = bound_layer
   mflow_filename   = "SARGEBLmff-07R-234s.out"
   nonl_pse_calc    = false
   solution_type    = local_eig_solution

   //Marching and Station Parameters
   marching_method_2d     = along_station
   init_station           = 2
   final_station          = 151

   //Wave Parameters
   freq_unit = in_hertz_freq
   beta_unit = non_dim_beta
   freq      = 1.e3,2.e3,3.e3,4.e3,5.e3,6.e3,7.e3,8.e3,9.e3,10.e3
   beta      = 10*0.
   qp_approx = true

   //Grid Parameters
   grid_type              = wall_cluster
   ymax                   = 200
   ymax_glob_search       = 60
   num_normal_pts         = 101
   num_normal_pts_geig    = 41

   //Curvature Parameters
   strm_curvt       = false
   transv_curvt     = false

   //Order of Accuracy Parameters
   strm_order             = second_order
   wall_normal_order      = fourth_order

   //Memory Storage Parameter
   mflow_storage_type     = file_storage

   //Extrapolation Parameter
   use_extrap_mprof = true

   //Eigenvalue Search Parameter
   relax_type       = wall_temp

   //Growth Rate Limit Parameter
   np_growth_rate_min     = -1.e-4
```

APPENDIX F

LST CALCULATION RESULTS

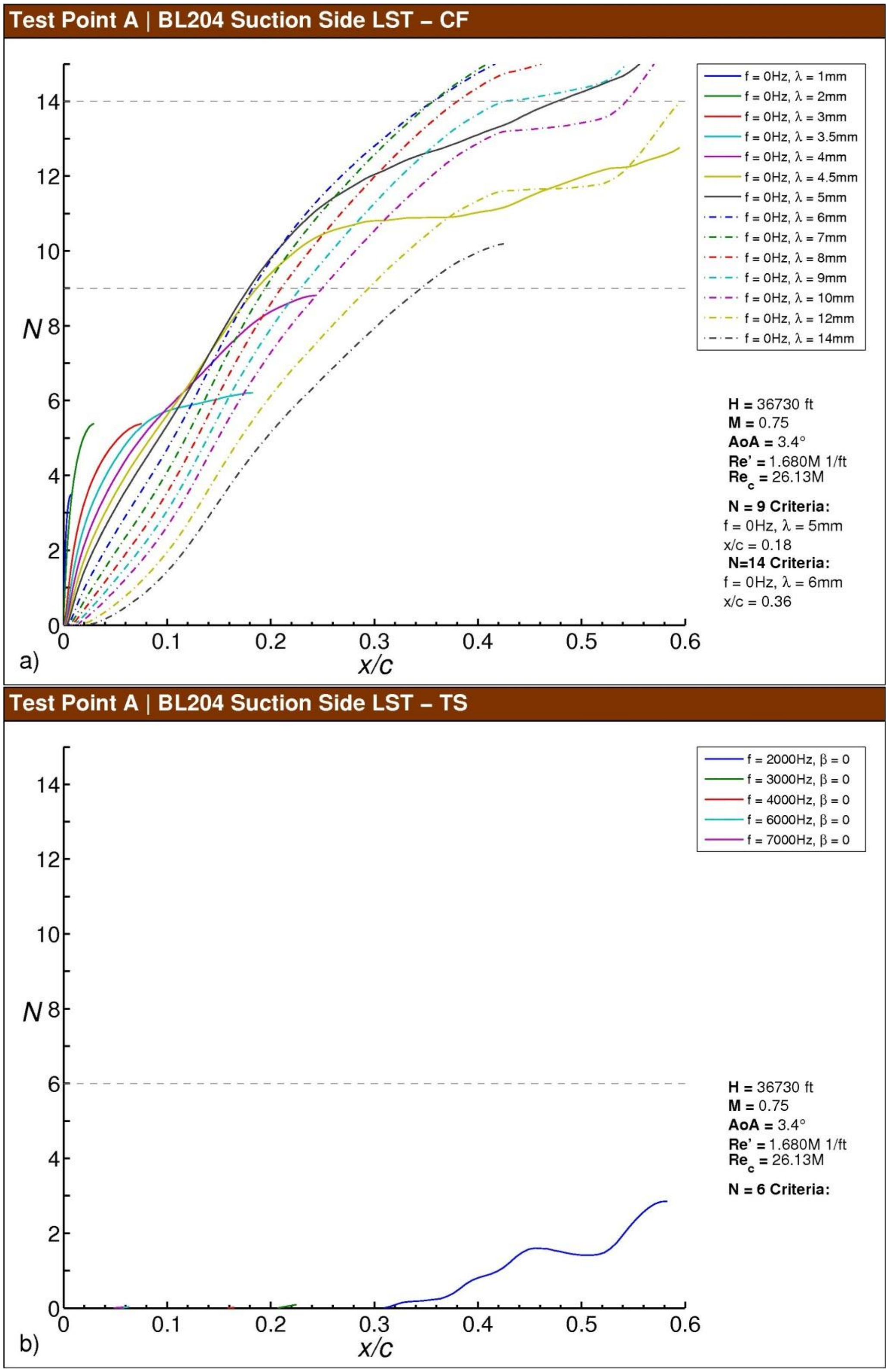


Fig. F-1 Test Point A *N*-factor results for BL204, suction side: a) stationary crossflow, b) T-S.

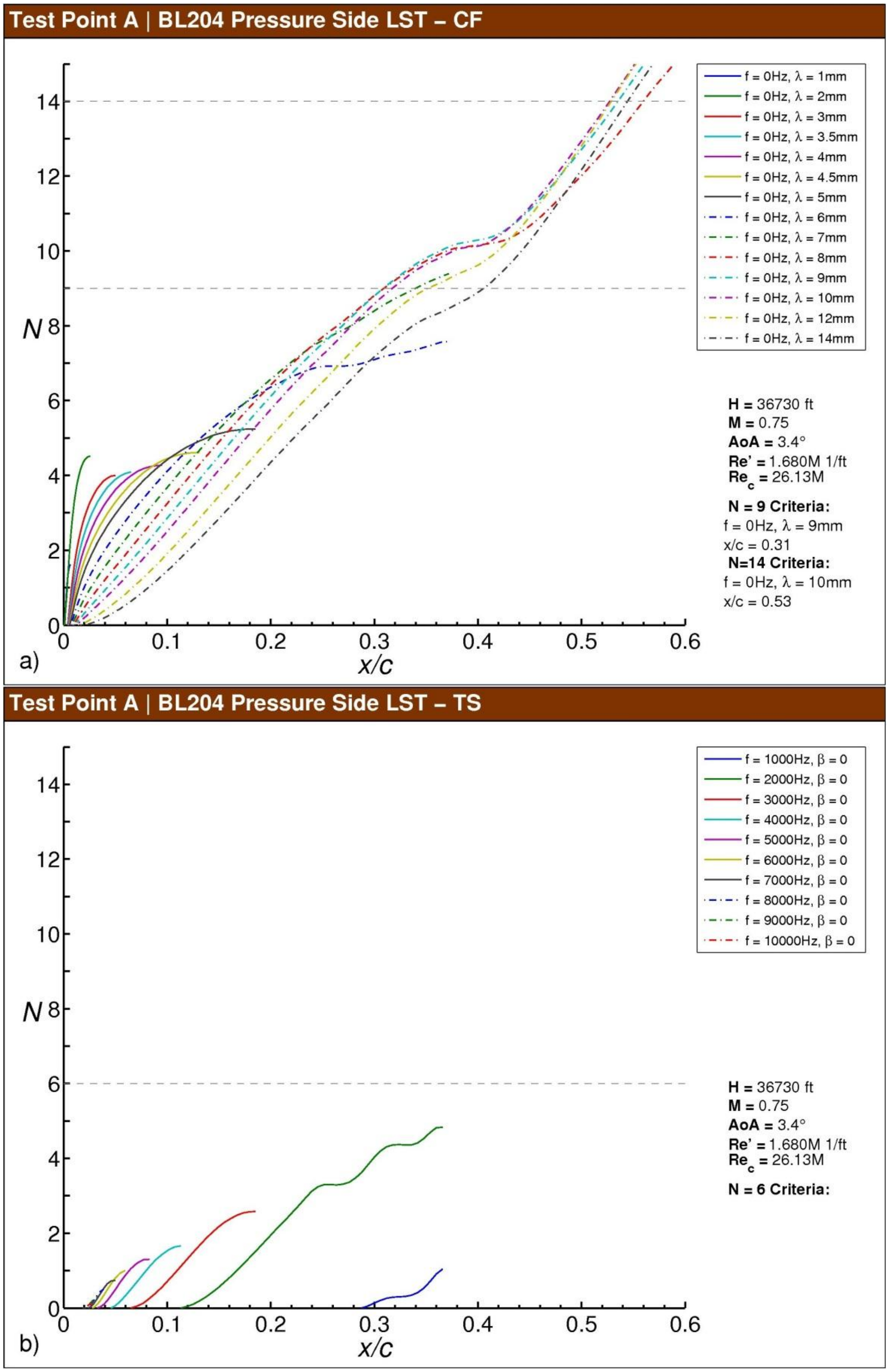


Fig. F-2 Test Point A *N*-factor results for BL204, pressure side: a) stationary crossflow, b) T-S.

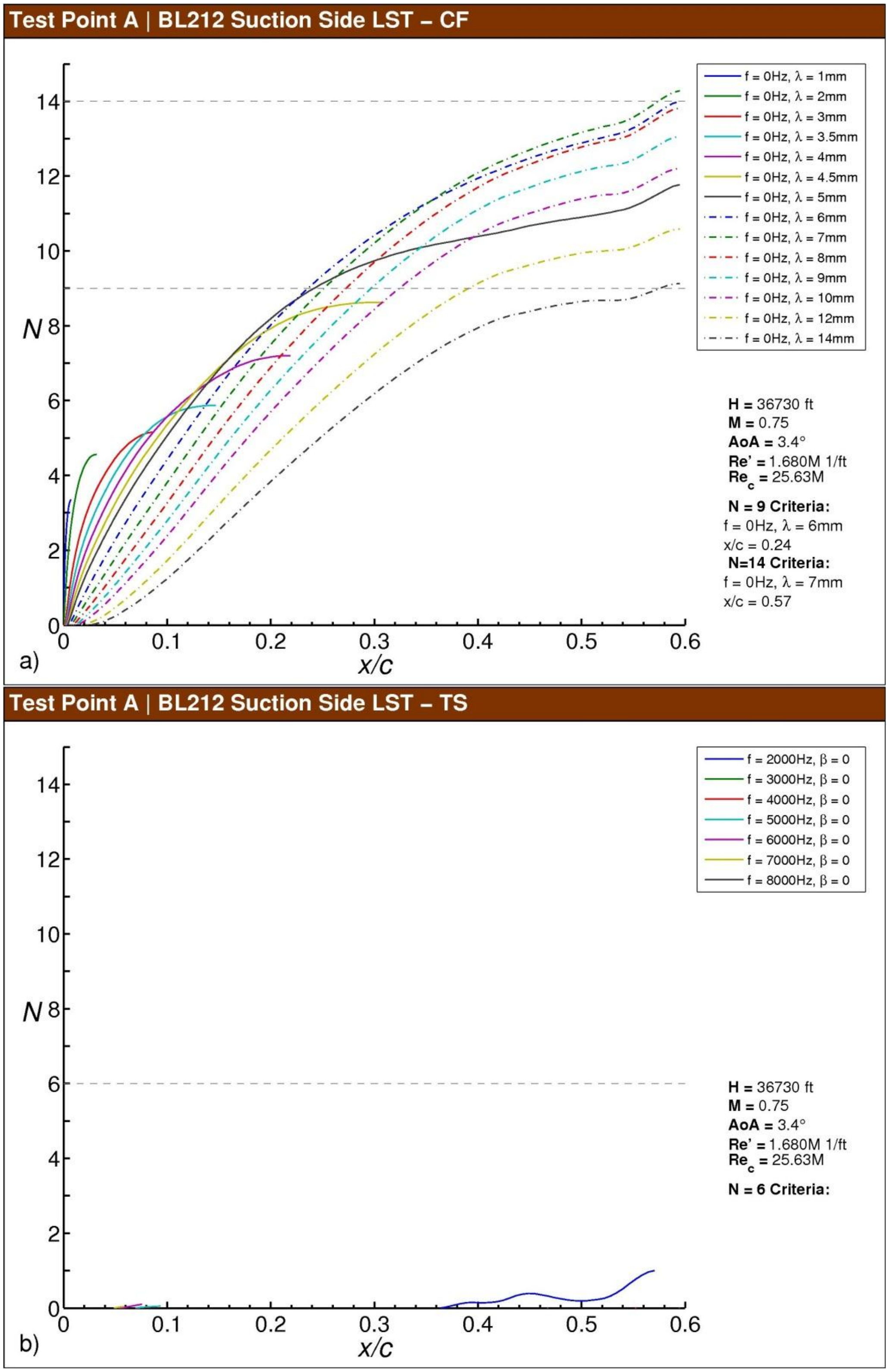


Fig. F-3 Test Point A *N*-factor results for BL212, suction side: a) stationary crossflow, b) T-S.

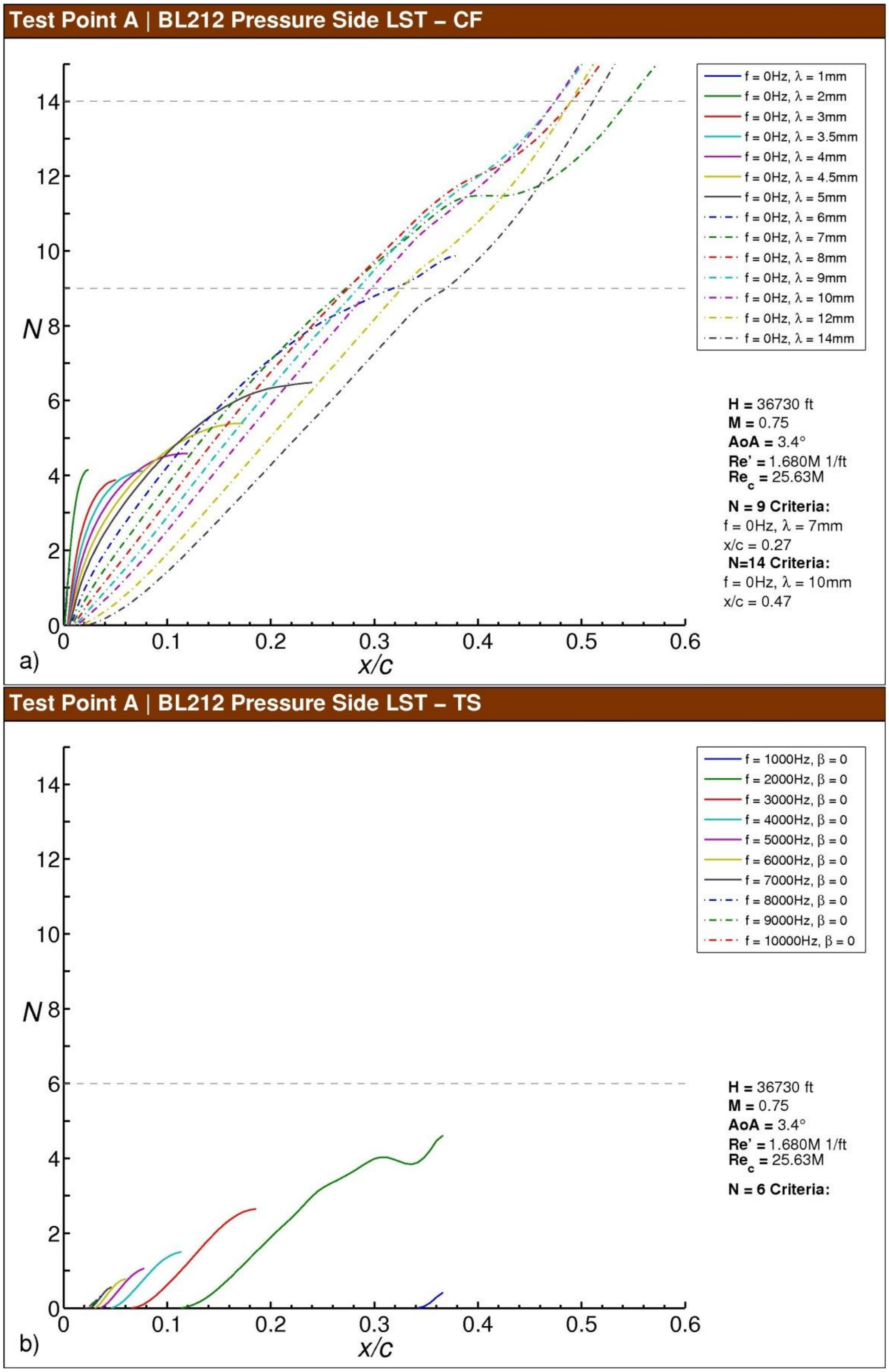
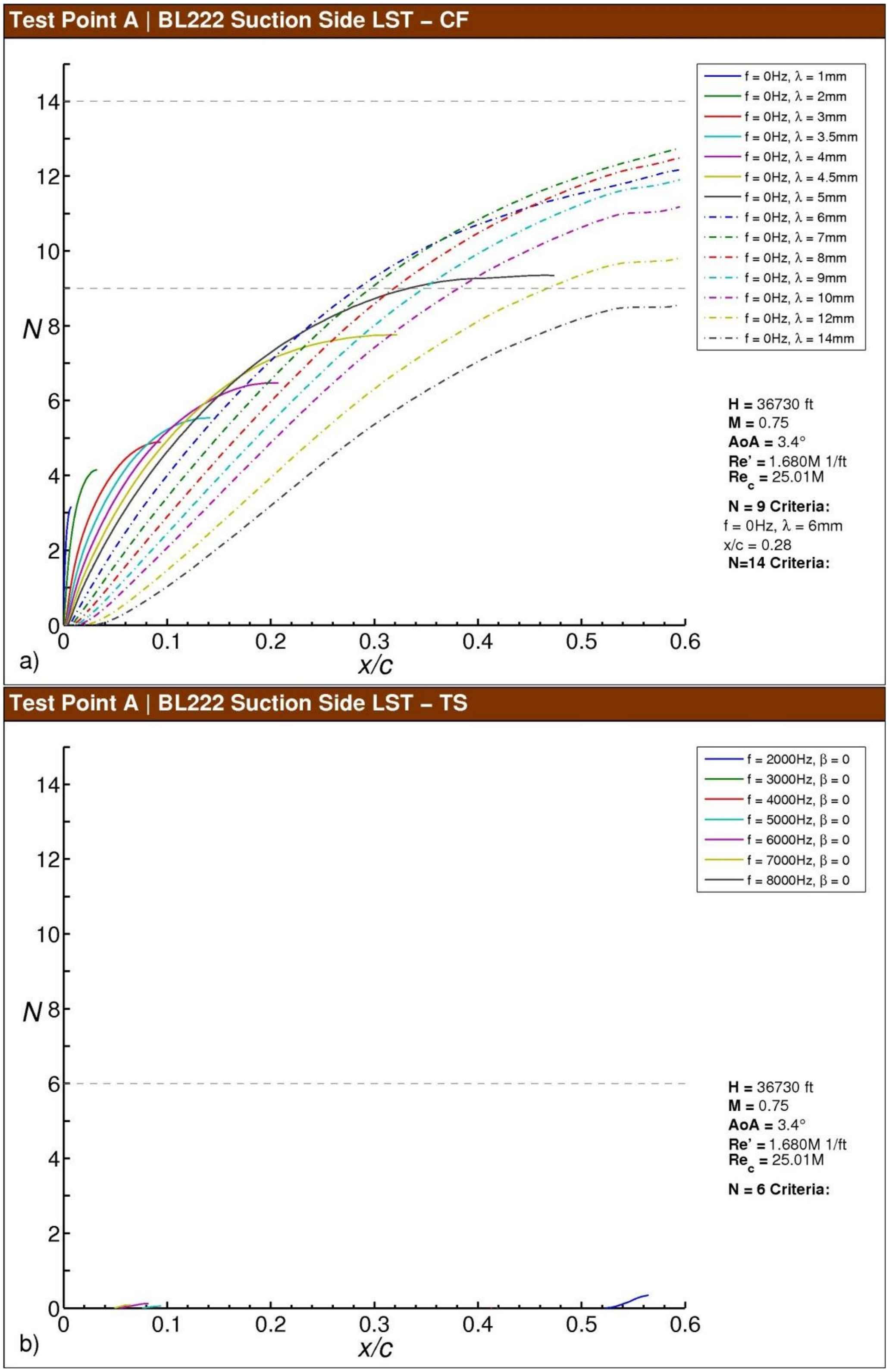


Fig. F-4 Test Point A *N*-factor results for BL212, pressure side: a) stationary crossflow, b) T-S.



Test Point A | BL222 Suction Side LST – CF

H = 36730 ft
M = 0.75
AoA = 3.4°
Re' = 1.680M 1/ft
Re$_c$ = 25.01M

N = 9 Criteria:
f = 0Hz, λ = 6mm
x/c = 0.28
N=14 Criteria:

Test Point A | BL222 Suction Side LST – TS

H = 36730 ft
M = 0.75
AoA = 3.4°
Re' = 1.680M 1/ft
Re$_c$ = 25.01M

N = 6 Criteria:

Fig. F-5 Test Point A *N*-factor results for BL222, suction side: a) stationary crossflow, b) T-S.

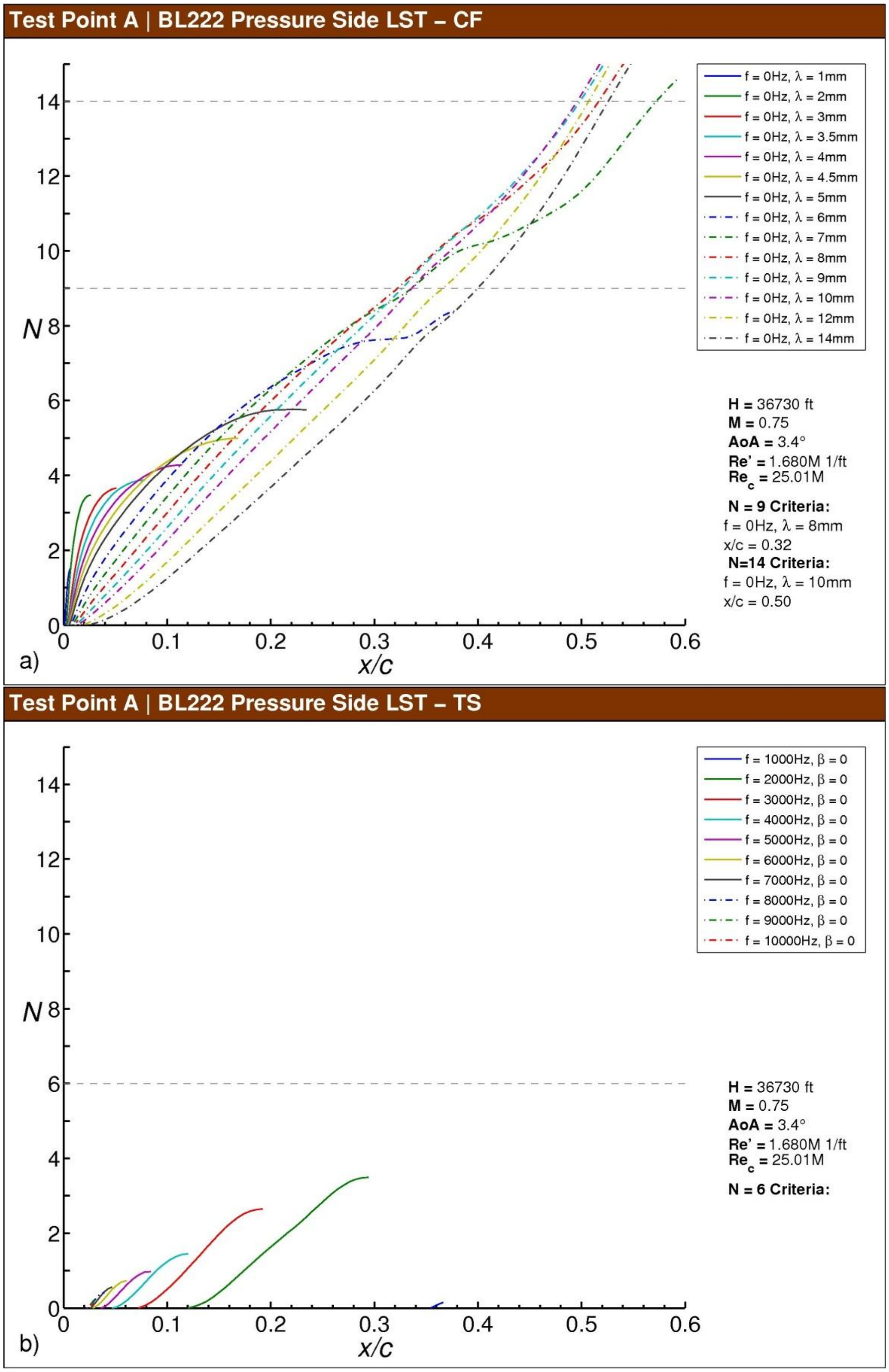


Fig. F-6 Test Point A *N*-factor results for BL222, pressure side: a) stationary crossflow, b) T-S.

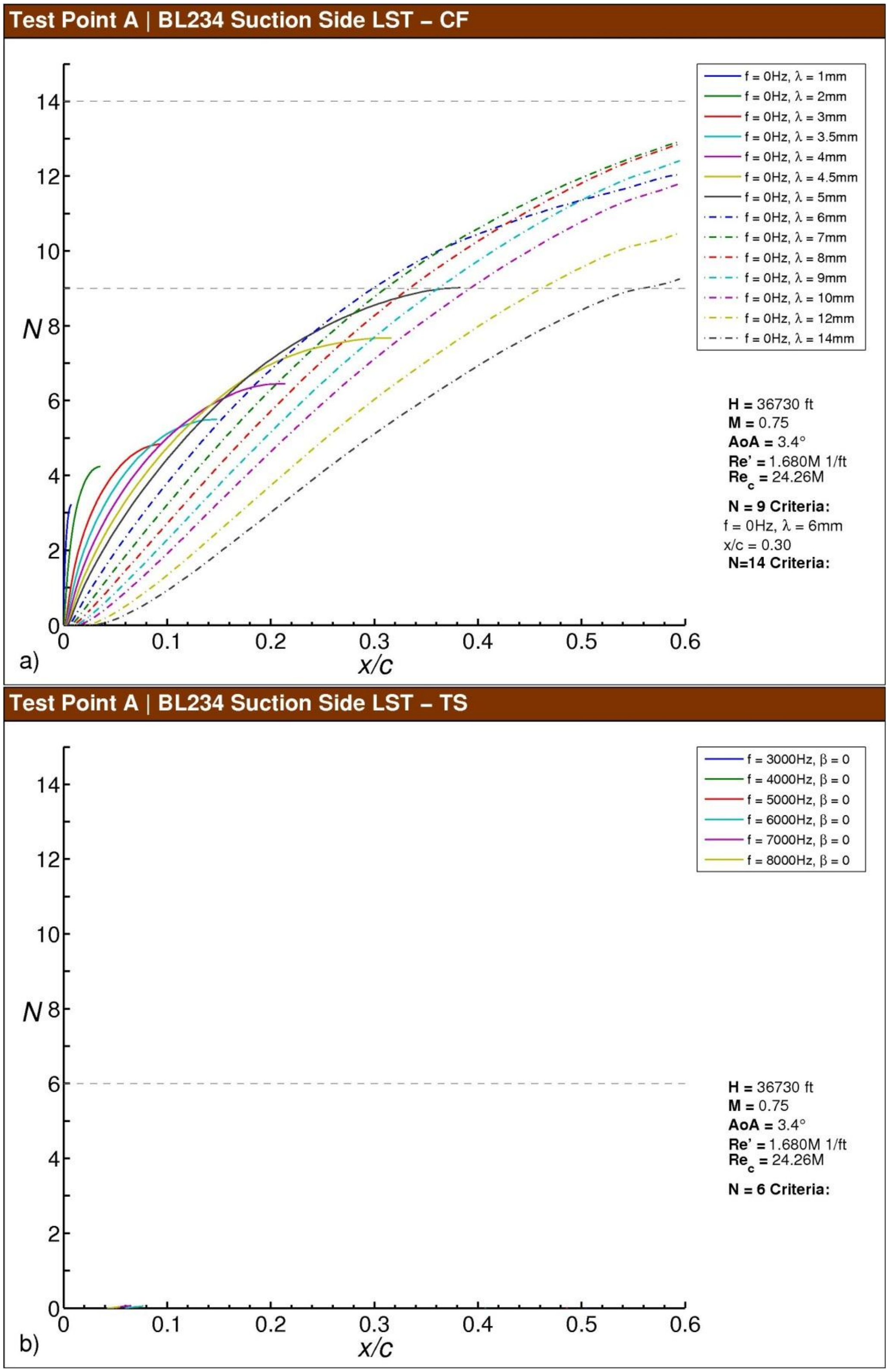


Fig. F-7 Test Point A *N*-factor results for BL234, suction side: a) stationary crossflow, b) T-S.

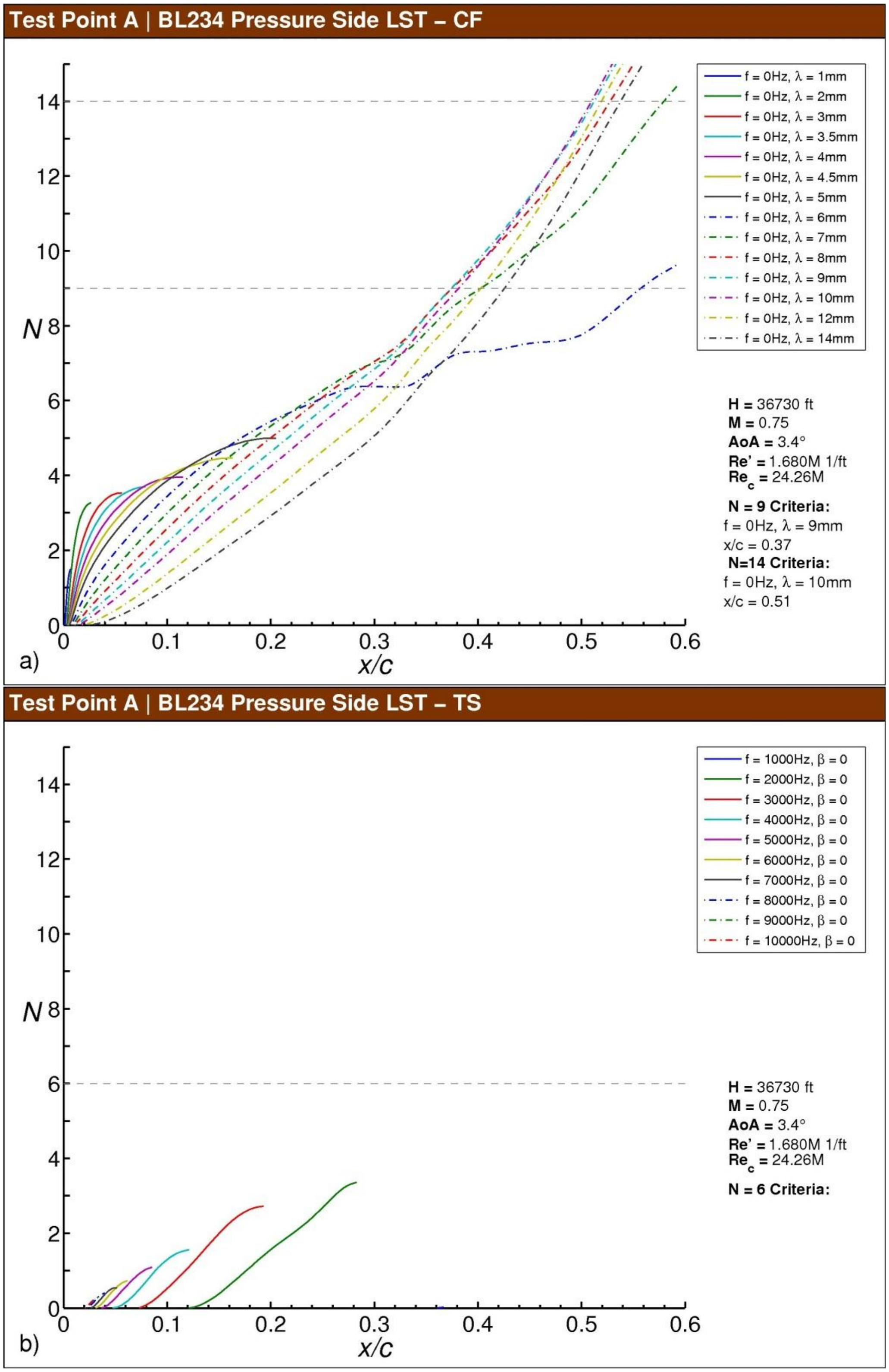


Fig. F-8 Test Point A *N*-factor results for BL234, pressure side: a) stationary crossflow, b) T-S.

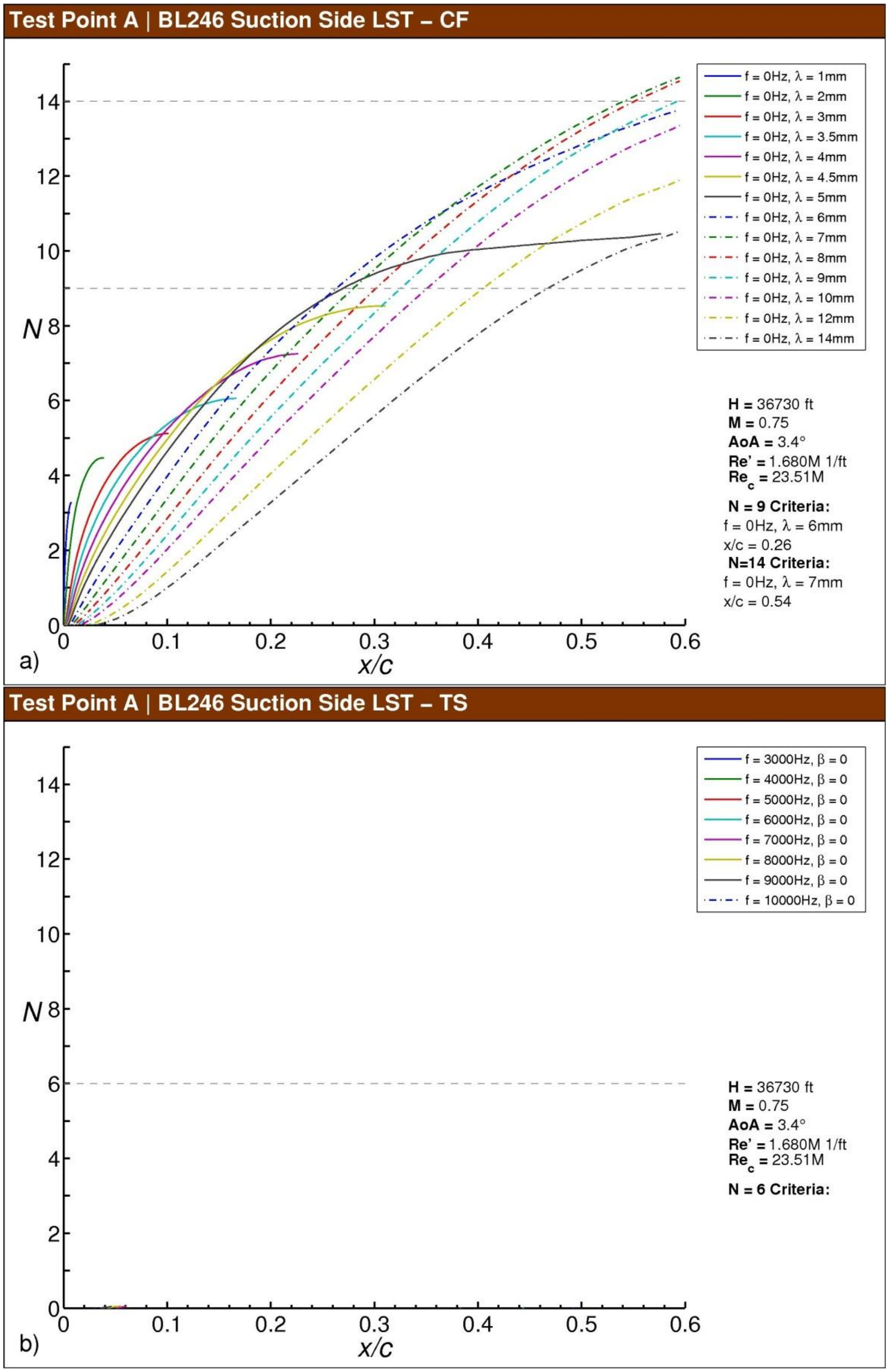


Fig. F-9 Test Point A *N*-factor results for BL246, suction side: a) stationary crossflow, b) T-S.

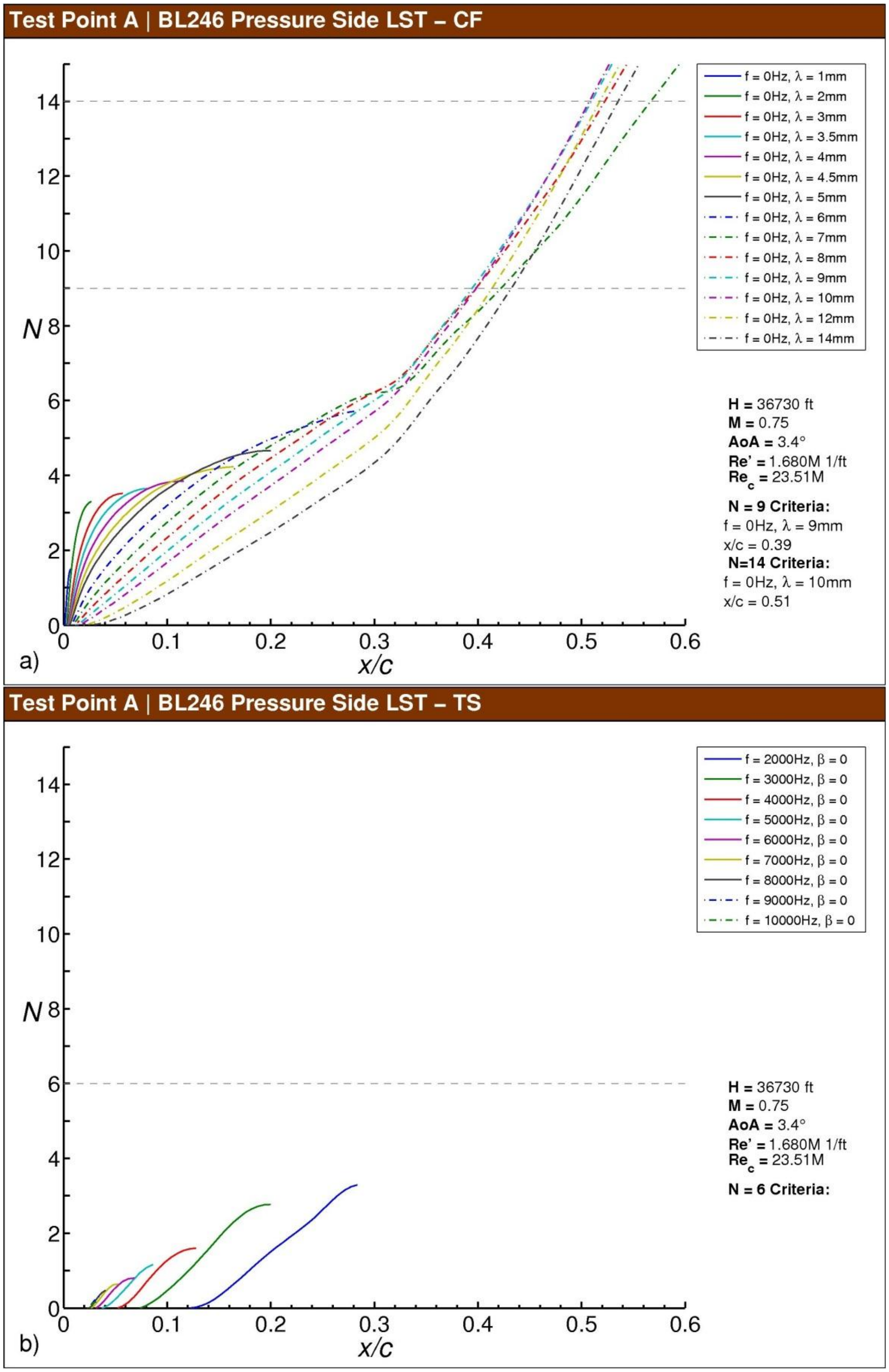


Fig. F-10 Test Point A *N*-factor results for BL246, pressure side: a) stationary crossflow, b) T-S.

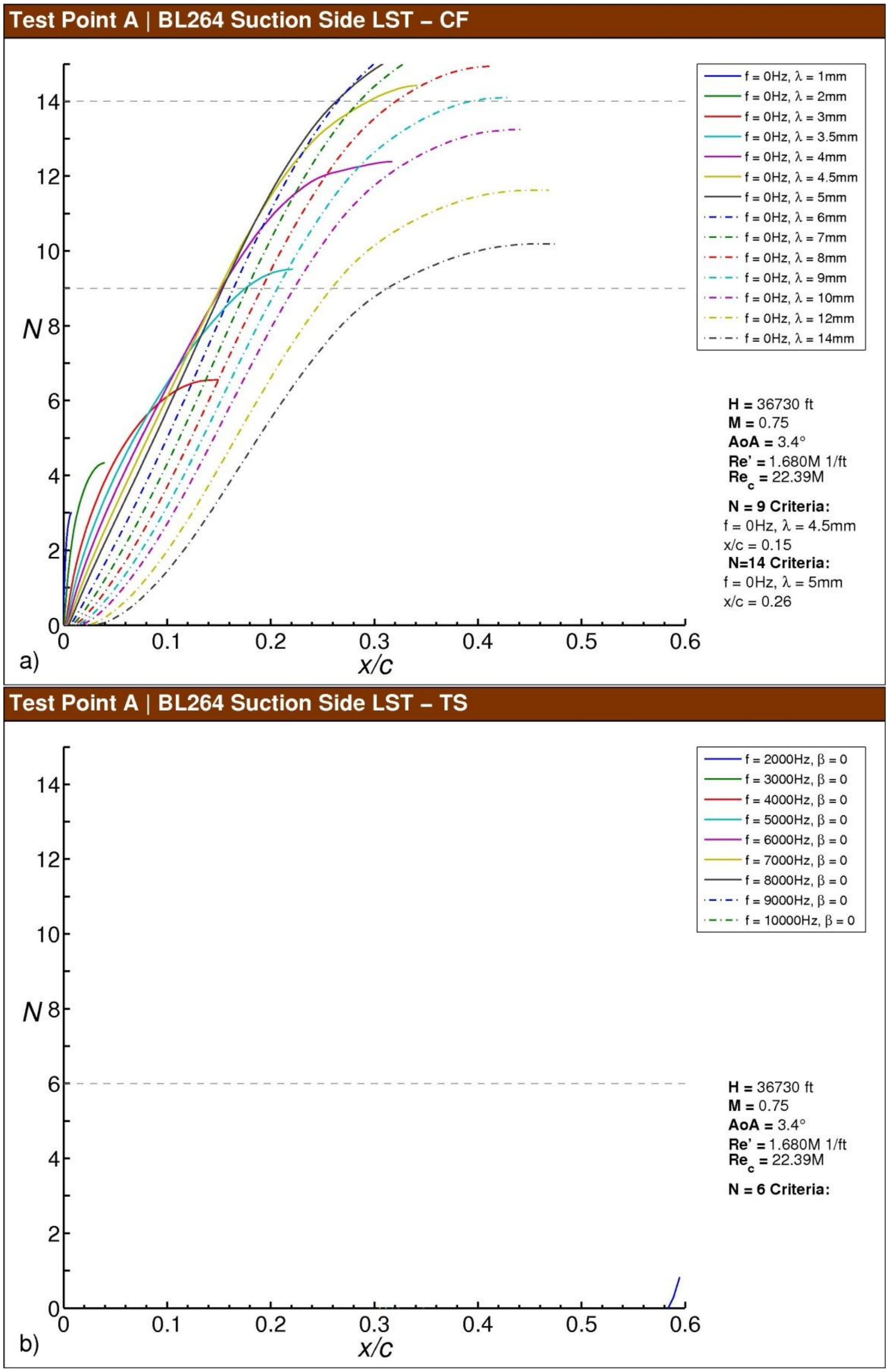


Fig. F-11 Test Point A *N*-factor results for BL264, suction side: a) stationary crossflow, b) T-S.

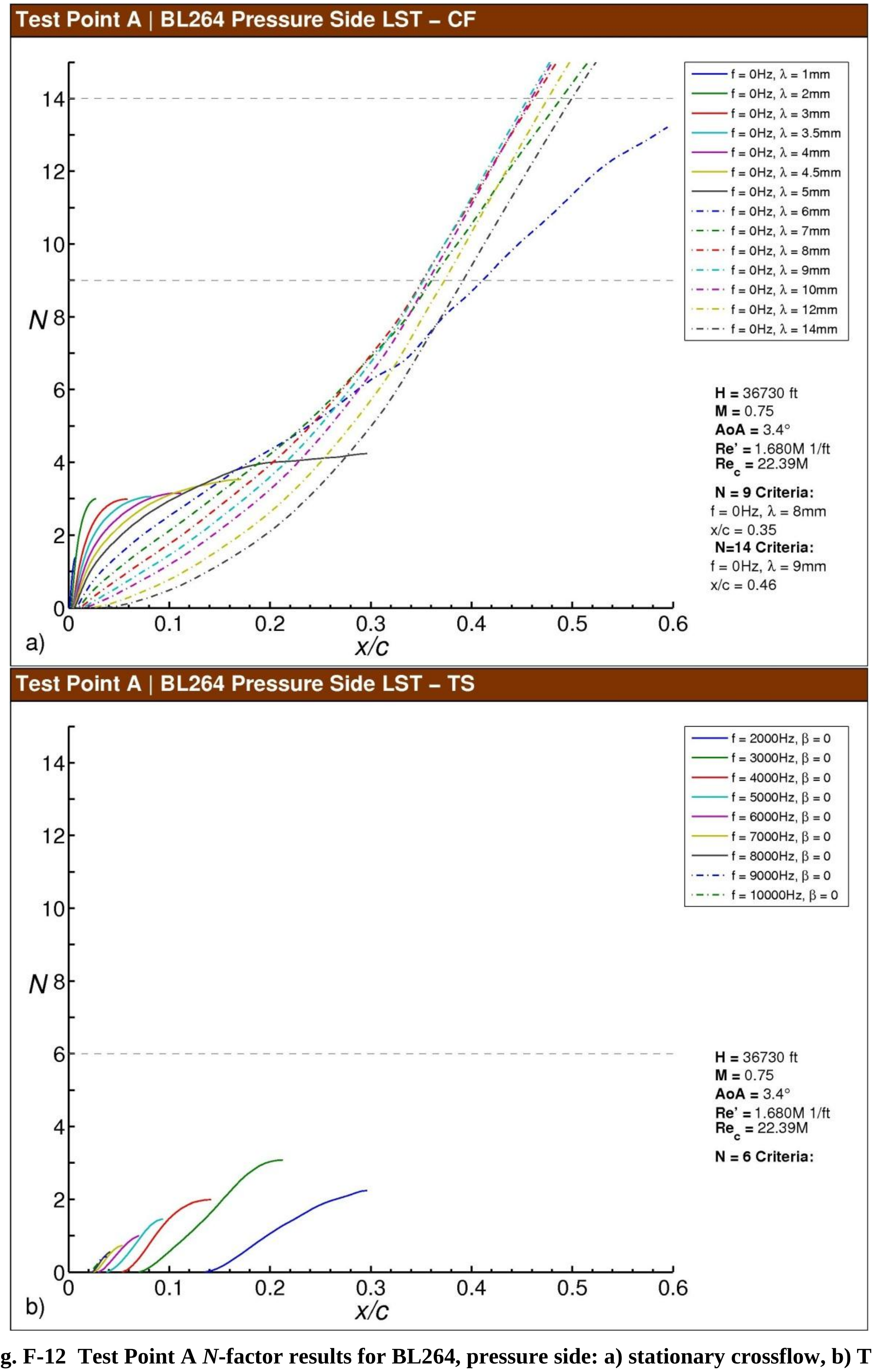


Fig. F-12 Test Point A *N*-factor results for BL264, pressure side: a) stationary crossflow, b) T-S.

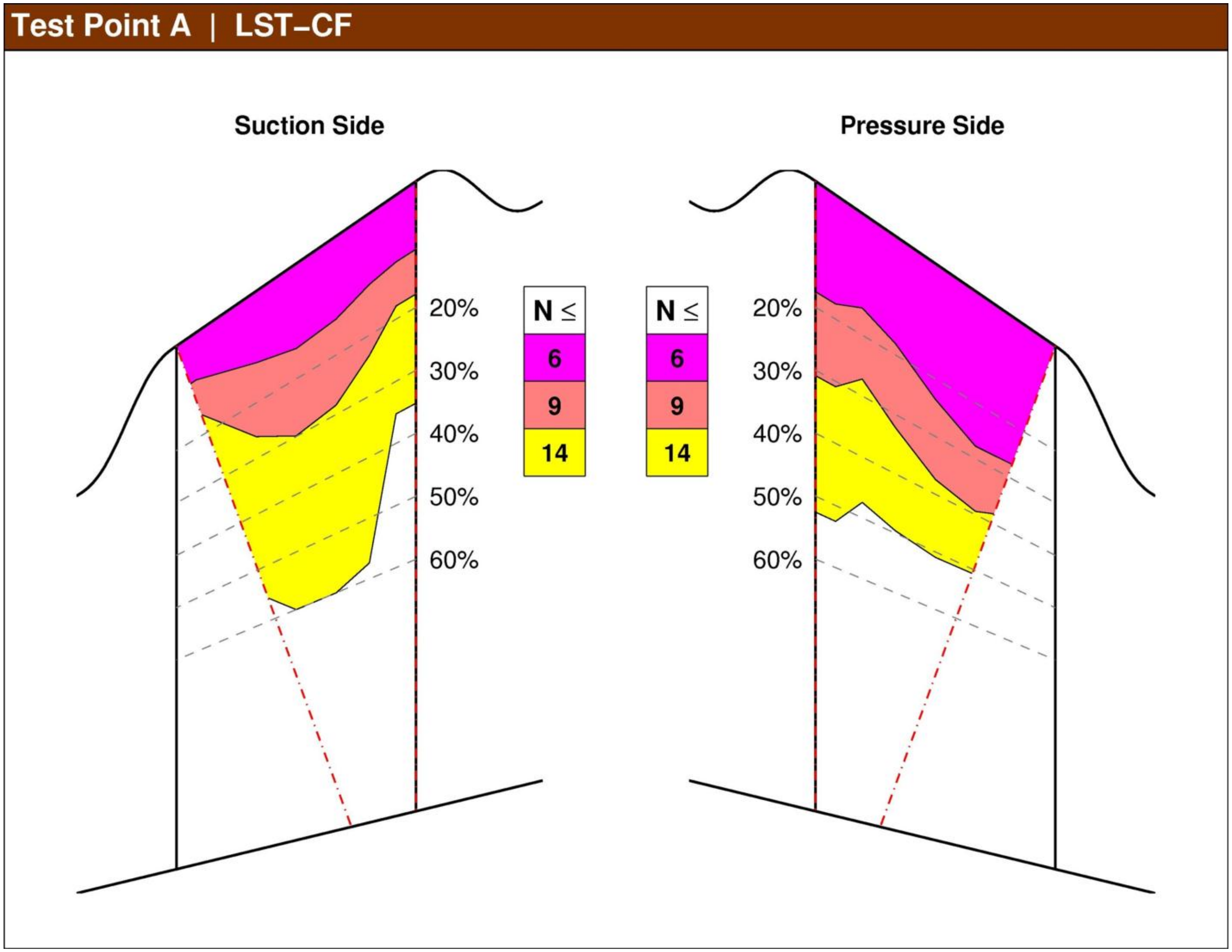


Fig. F-13 Test Point A *N*-factor contours for the test section suction and pressure sides.

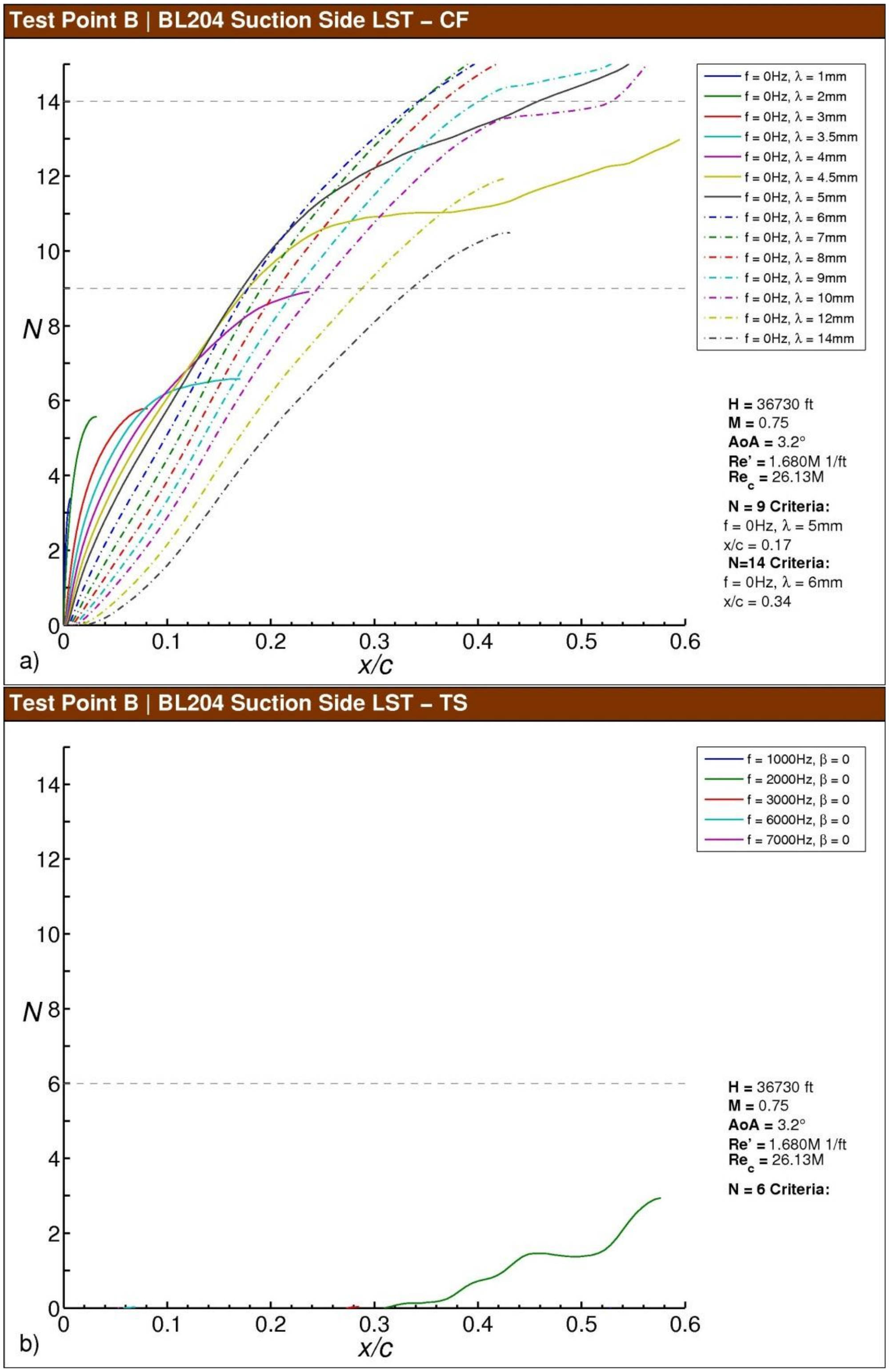


Fig. F-14 Test Point B *N*-factor results for BL204, suction side: a) stationary crossflow, b) T-S.

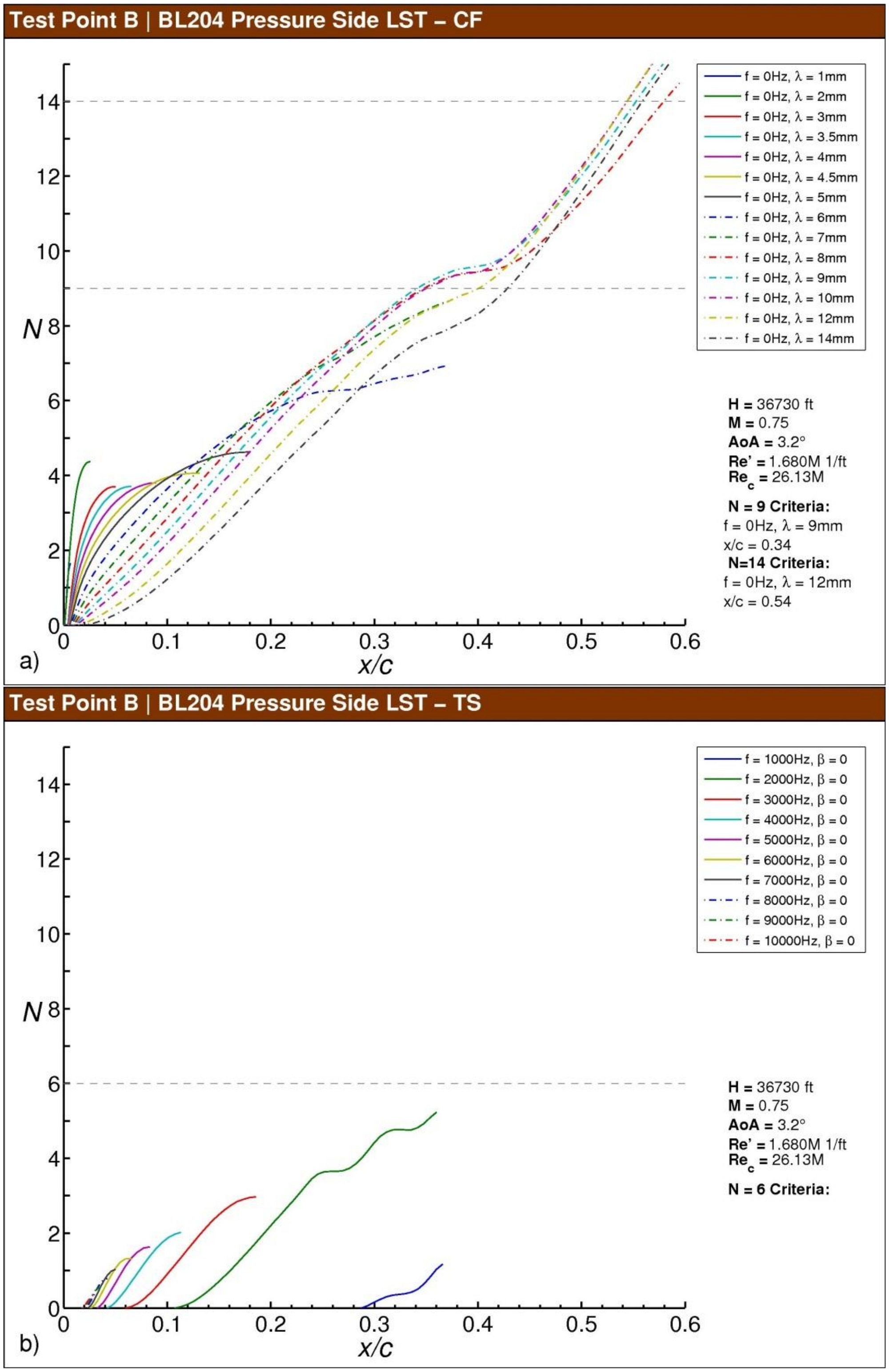


Fig. F-15 Test Point B *N*-factor results for BL204, pressure side: a) stationary crossflow, b) T-S.

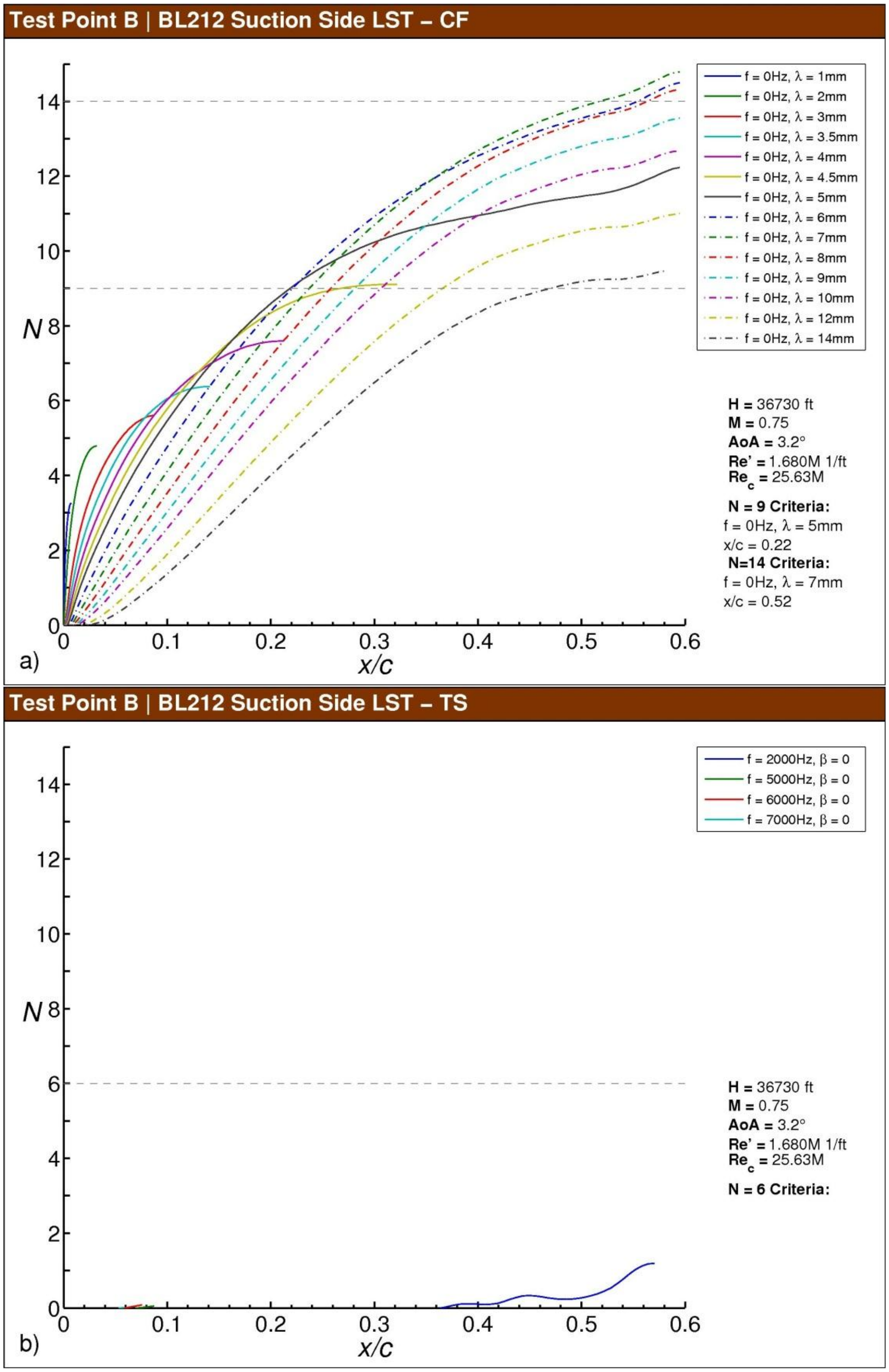


Fig. F-16 Test Point B *N*-factor results for BL212, suction side: a) stationary crossflow, b) T-S.

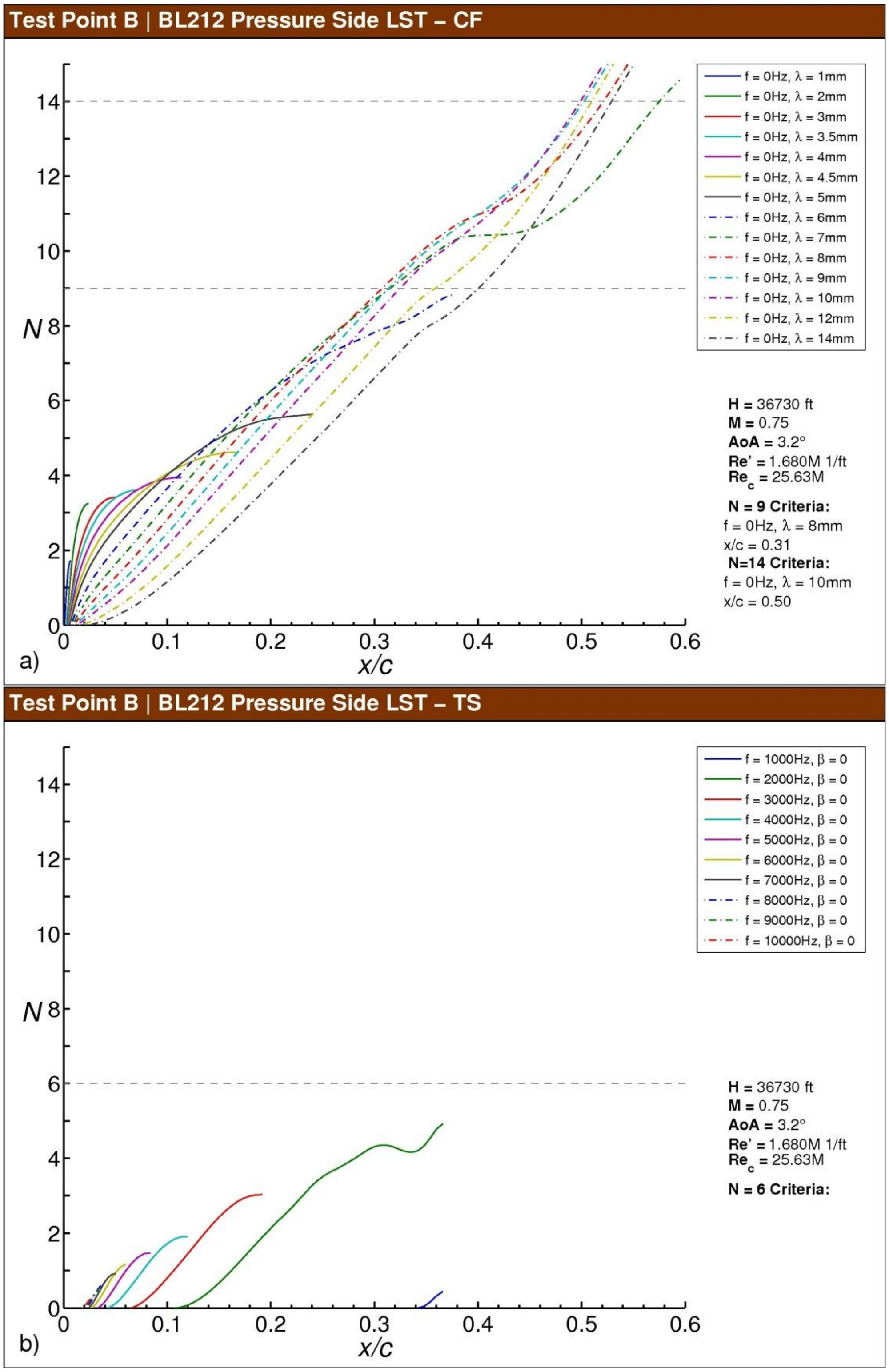


Fig. F-17 Test Point B *N*-factor results for BL212, pressure side: a) stationary crossflow, b) T-S.

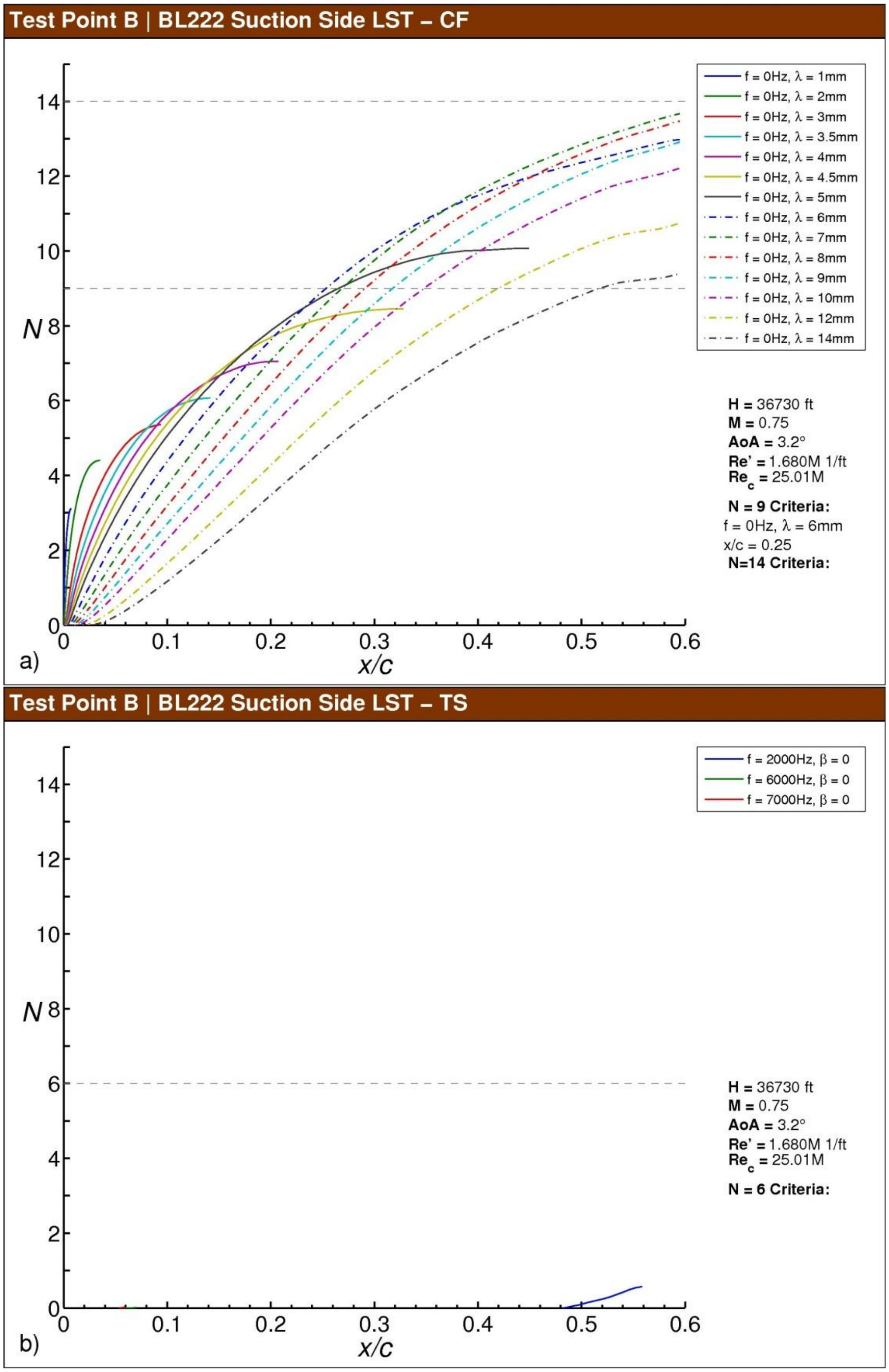


Fig. F-18 Test Point B *N*-factor results for BL222, suction side: a) stationary crossflow, b) T-S.

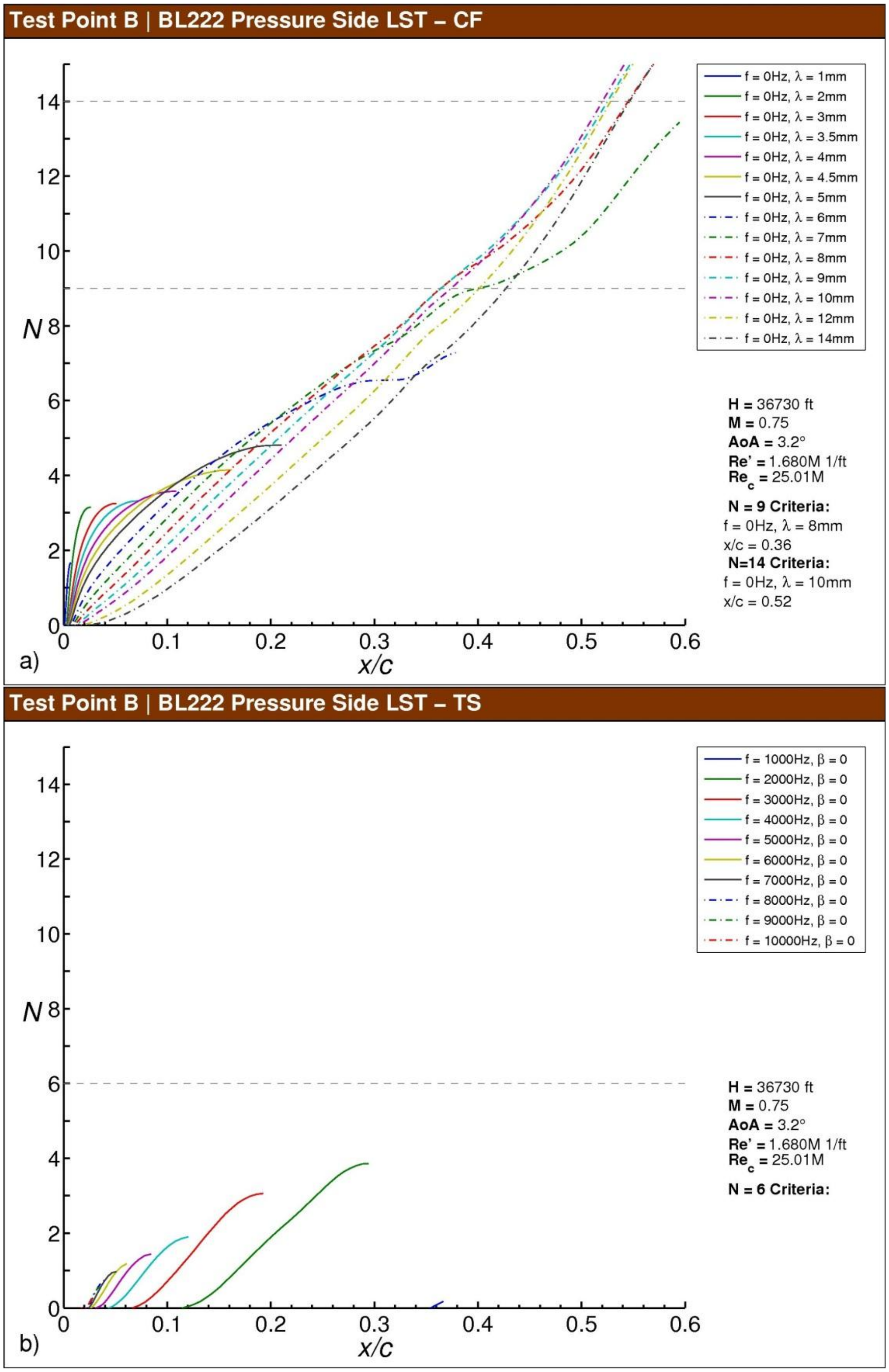


Fig. F-19 Test Point B *N*-factor results for BL222, pressure side: a) stationary crossflow, b) T-S.

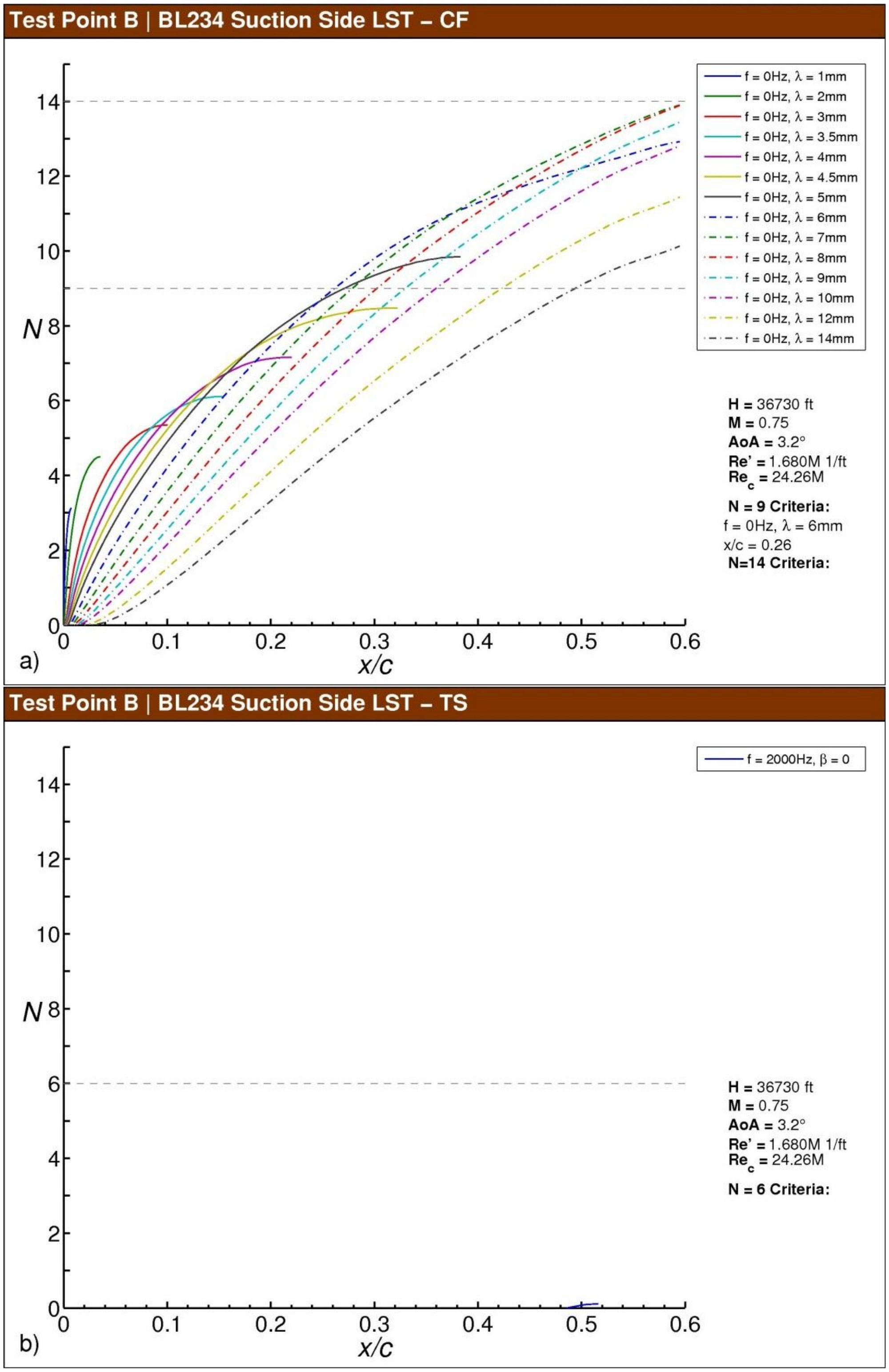


Fig. F-20 Test Point B *N*-factor results for BL234, suction side: a) stationary crossflow, b) T-S.

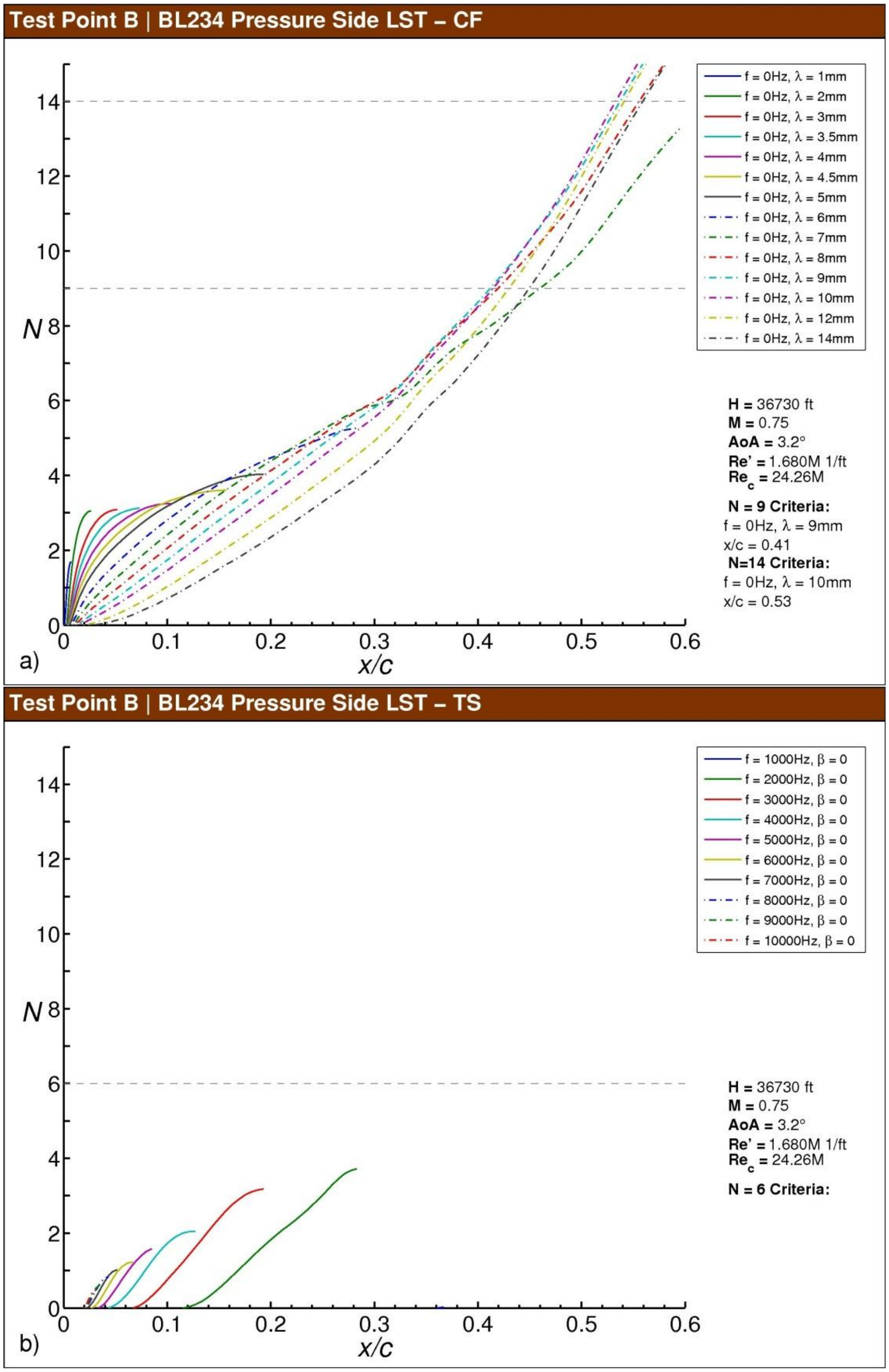


Fig. F-21 Test Point B *N*-factor results for BL234, pressure side: a) stationary crossflow, b) T-S.

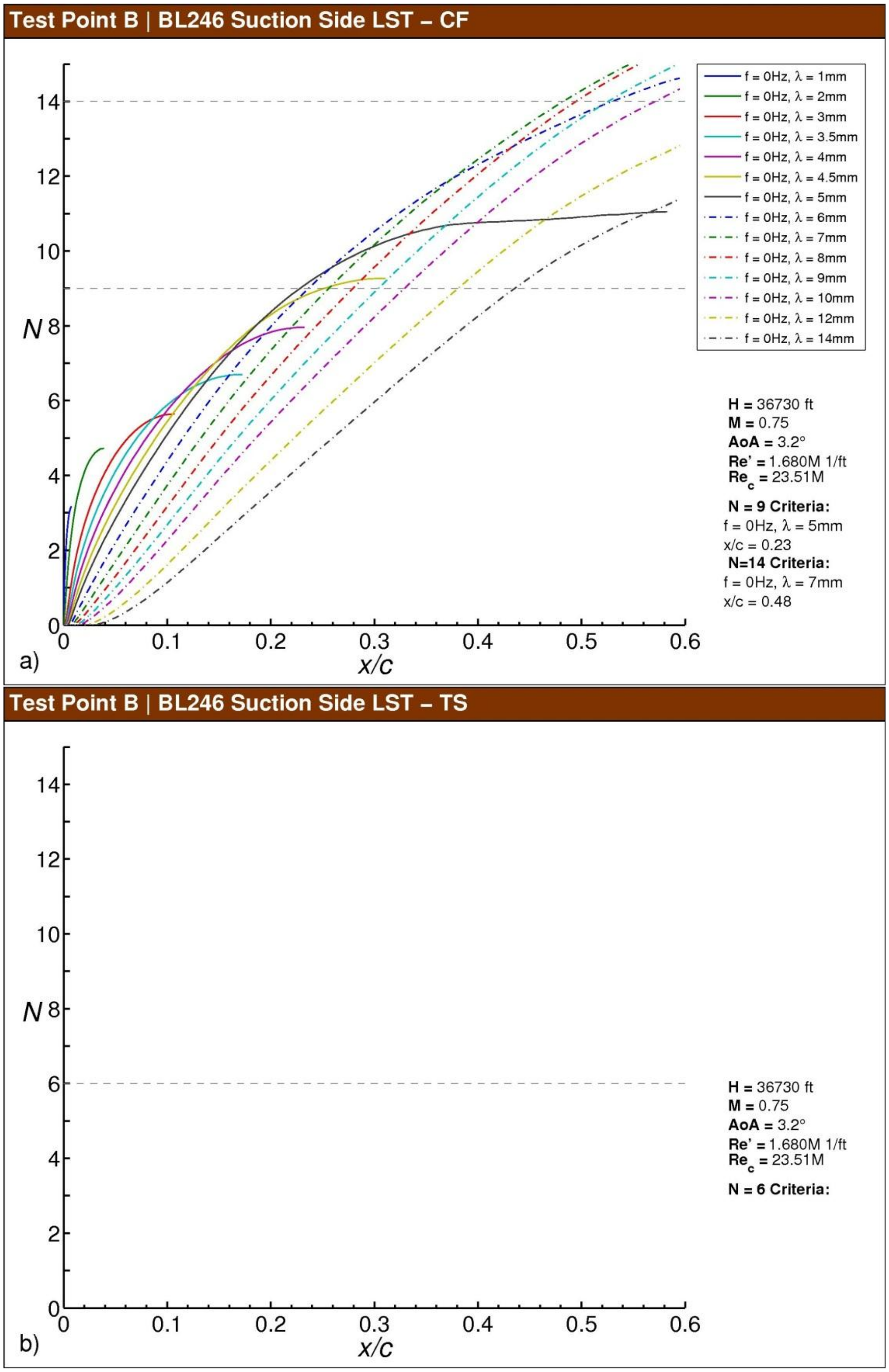


Fig. F-22 Test Point B *N*-factor results for BL246, suction side: a) stationary crossflow, b) T-S.

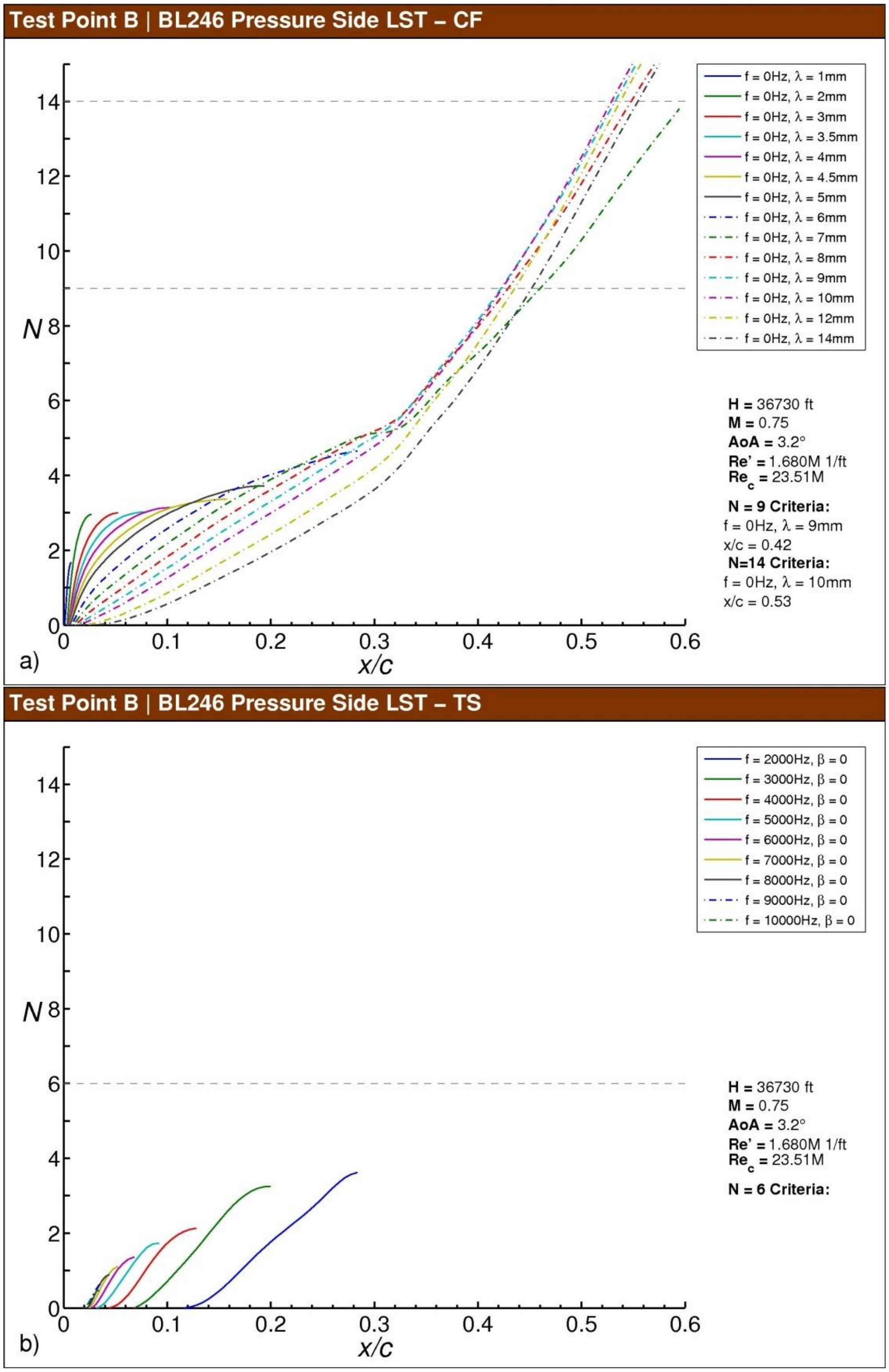


Fig. F-23 Test Point B *N*-factor results for BL246, pressure side: a) stationary crossflow, b) T-S.

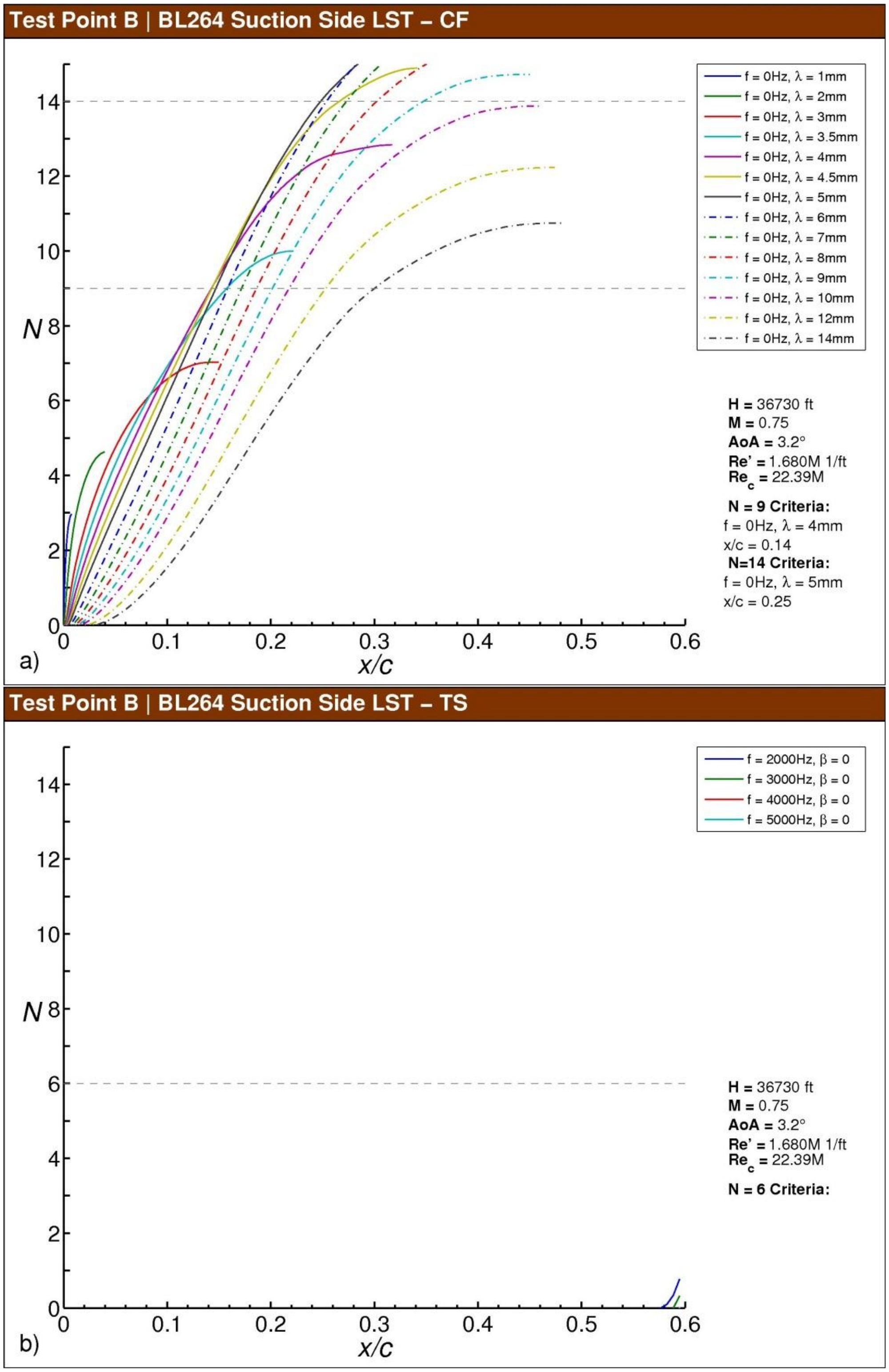


Fig. F-24 Test Point B *N*-factor results for BL264, suction side: a) stationary crossflow, b) T-S.

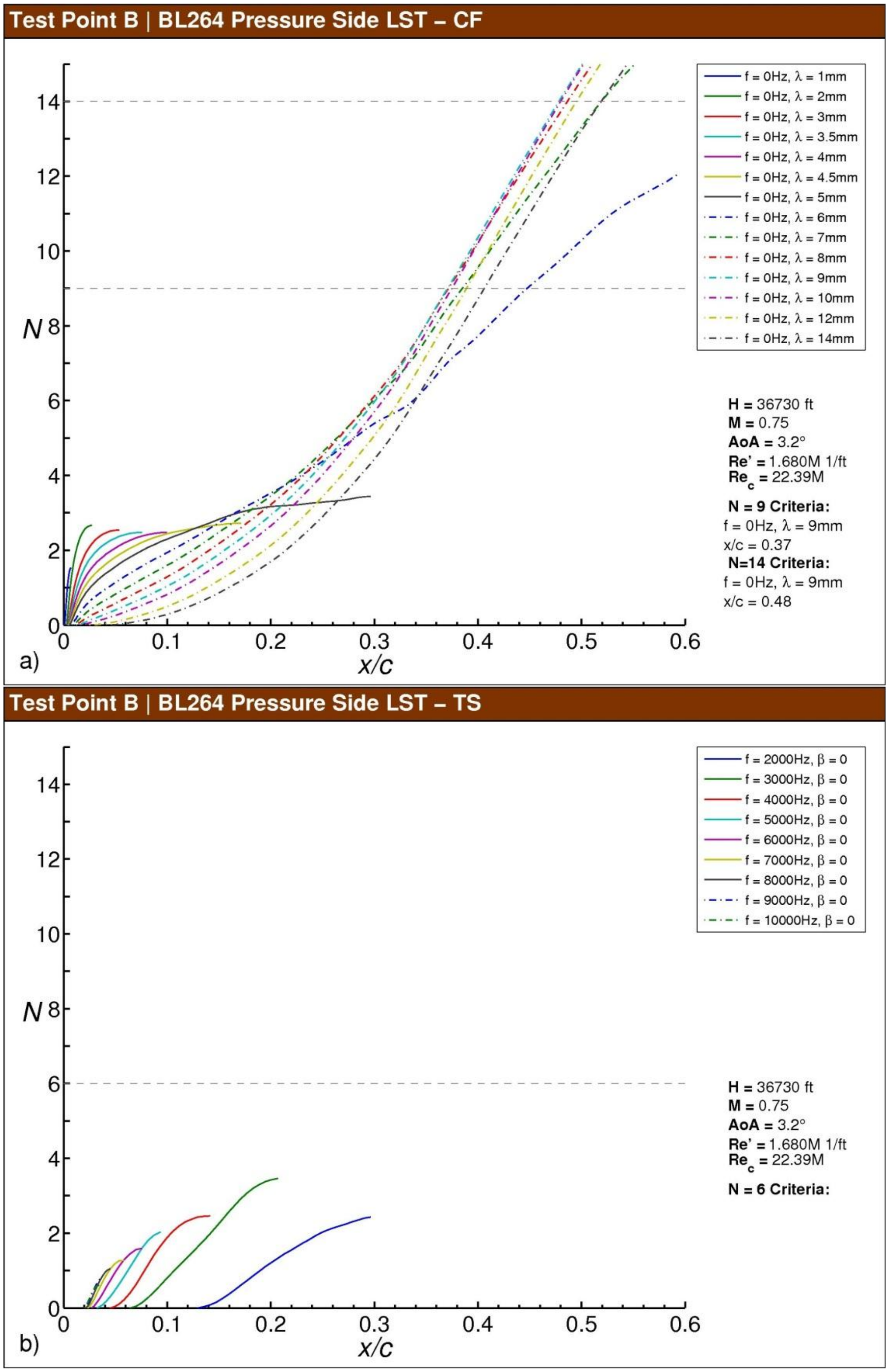


Fig. F-25 Test Point B *N*-factor results for BL264, pressure side: a) stationary crossflow, b) T-S.

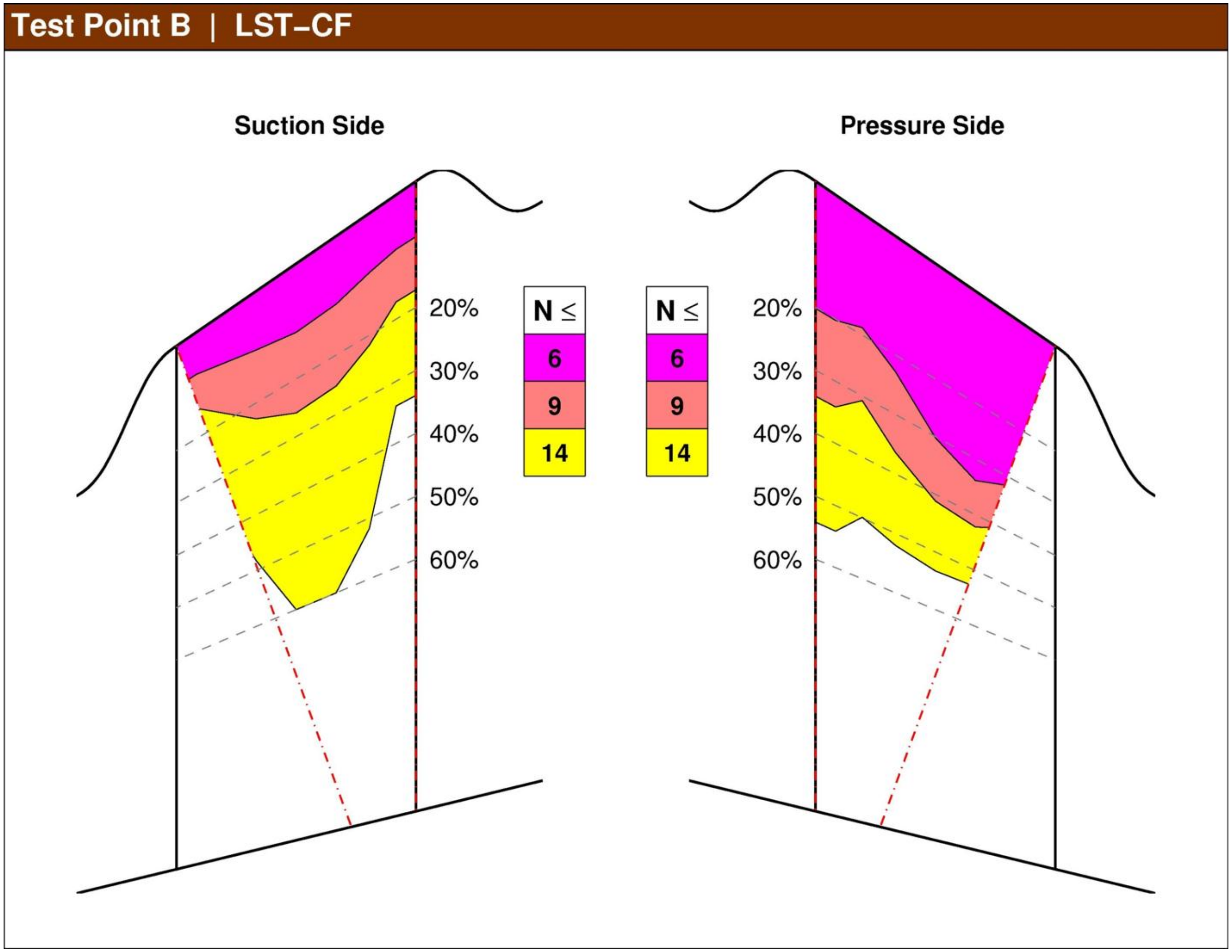


Fig. F-26 Test Point B *N*-factor contours for the test section suction and pressure sides.

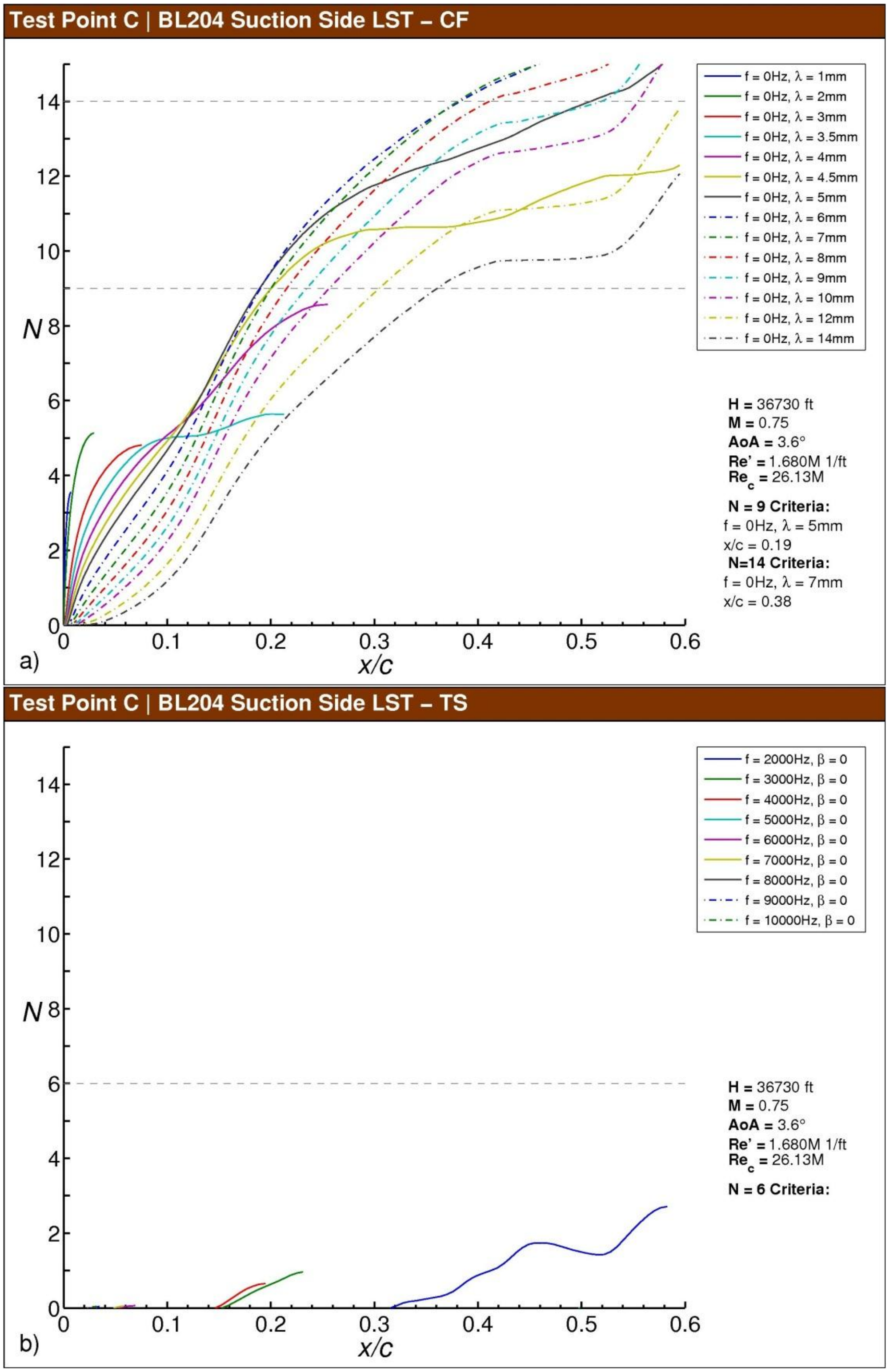
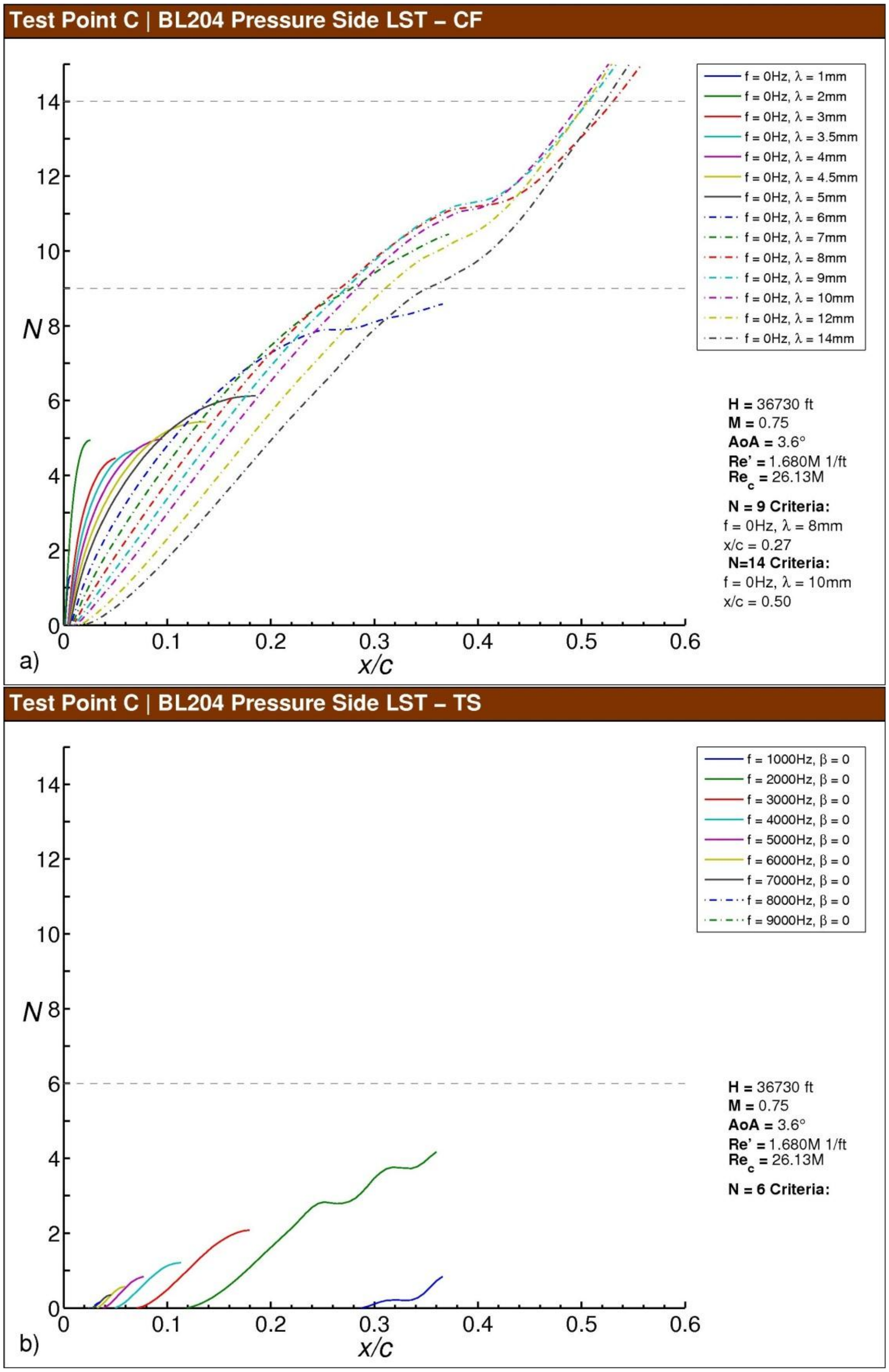


Fig. F-27 Test Point C *N*-factor results for BL204, suction side: a) stationary crossflow, b) T-S.


Test Point C | BL204 Pressure Side LST – CF

H = 36730 ft
M = 0.75
AoA = 3.6°
Re' = 1.680M 1/ft
Re_c = 26.13M

N = 9 Criteria:
f = 0Hz, λ = 8mm
x/c = 0.27

N=14 Criteria:
f = 0Hz, λ = 10mm
x/c = 0.50

Test Point C | BL204 Pressure Side LST – TS

H = 36730 ft
M = 0.75
AoA = 3.6°
Re' = 1.680M 1/ft
Re_c = 26.13M

N = 6 Criteria:

Fig. F-28 Test Point C *N*-factor results for BL204, pressure side: a) stationary crossflow, b) T-S.

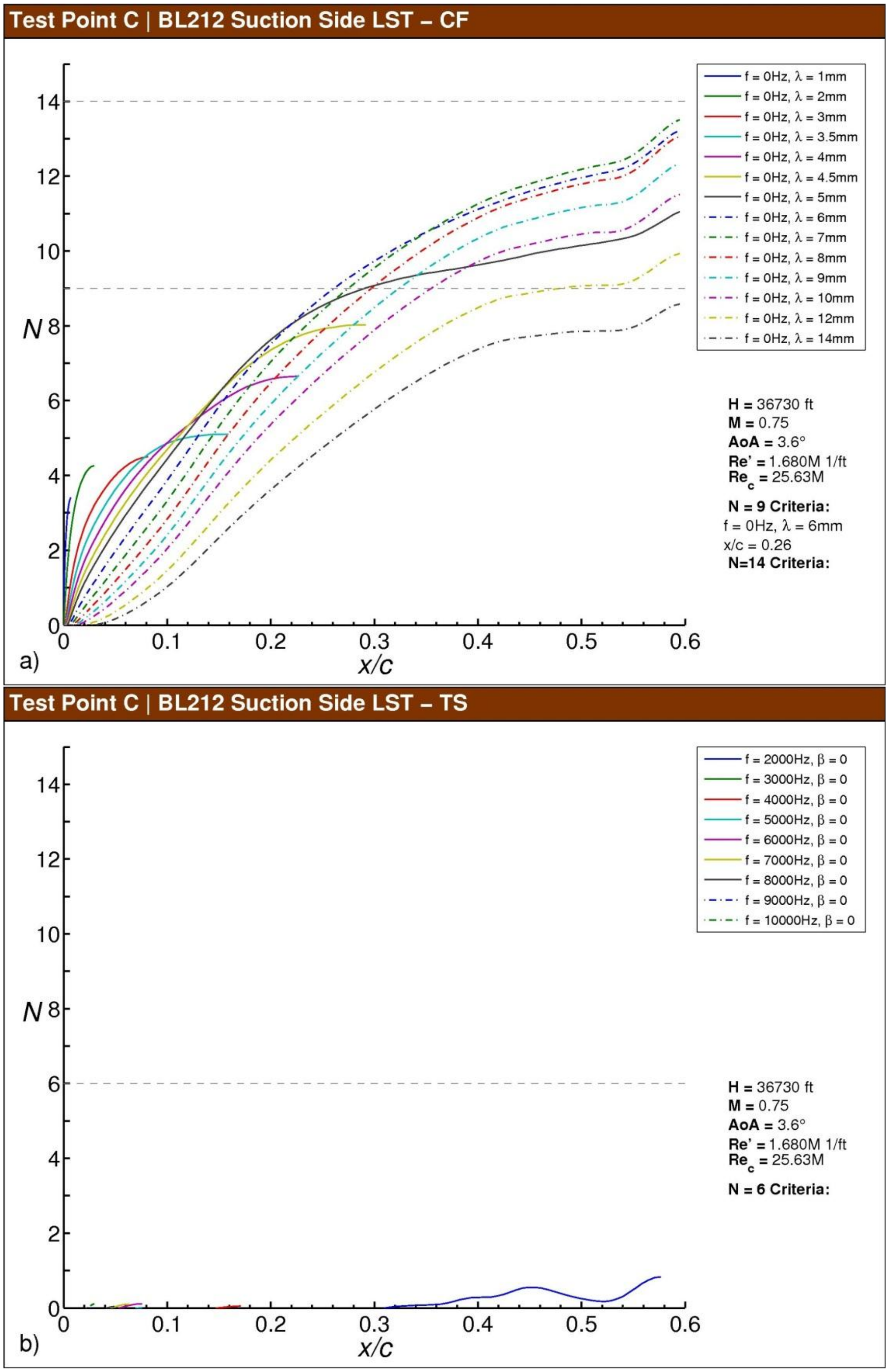


Fig. F-29 Test Point C *N*-factor results for BL212, suction side: a) stationary crossflow, b) T-S.

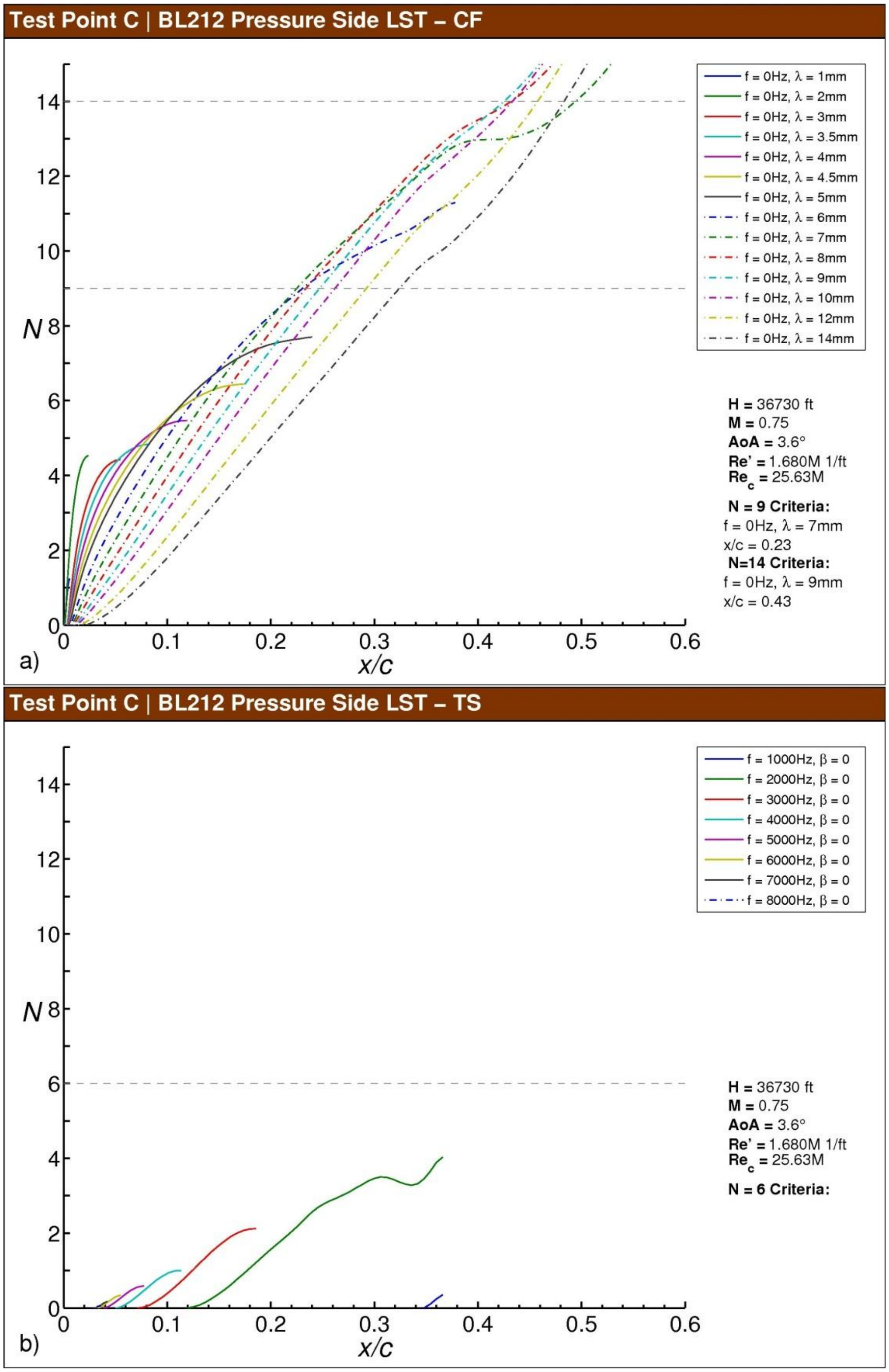


Fig. F-30 Test Point C *N*-factor results for BL212, pressure side: a) stationary crossflow, b) T-S.

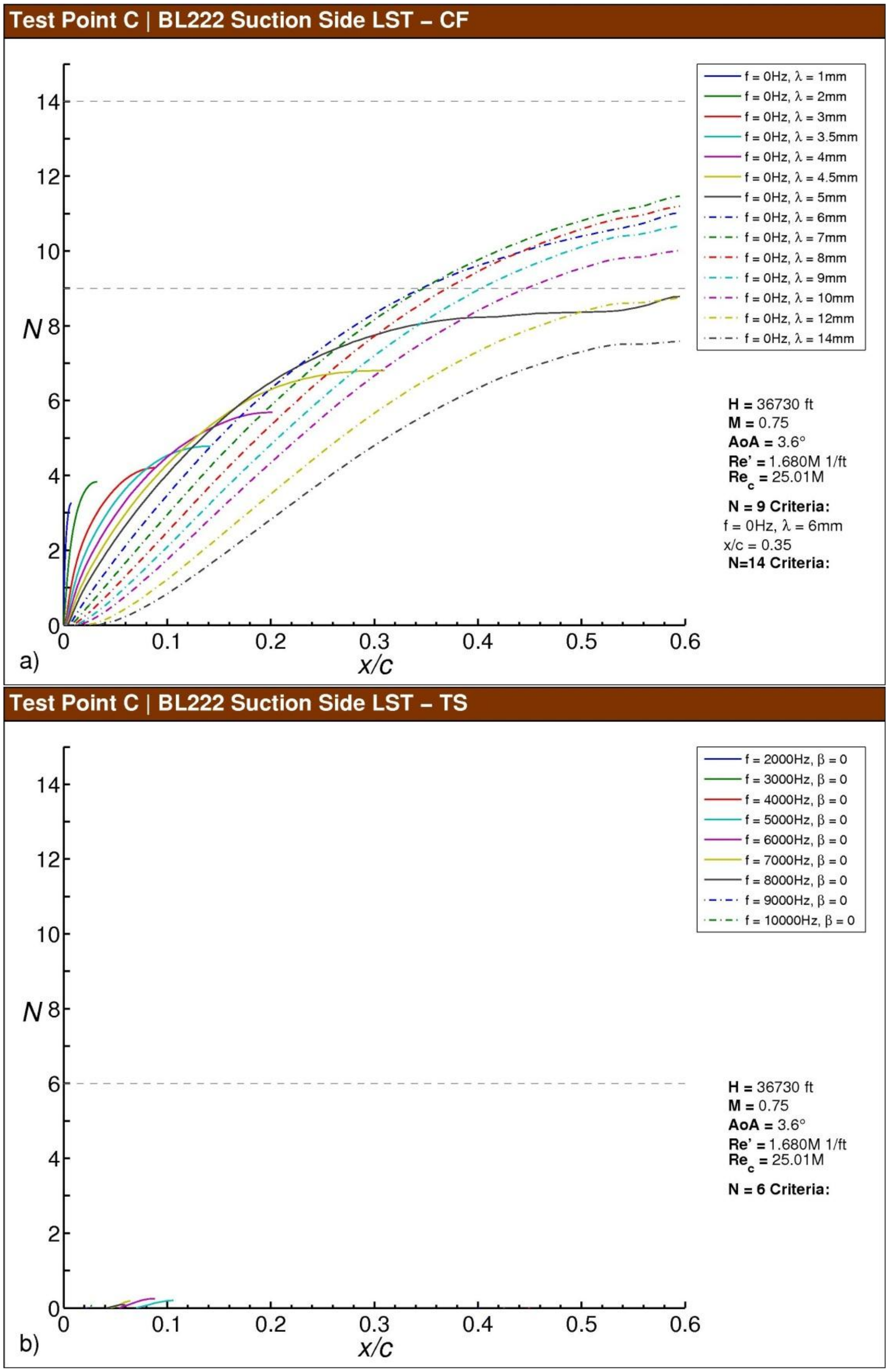


Fig. F-31 Test Point C *N*-factor results for BL222, suction side: a) stationary crossflow, b) T-S.

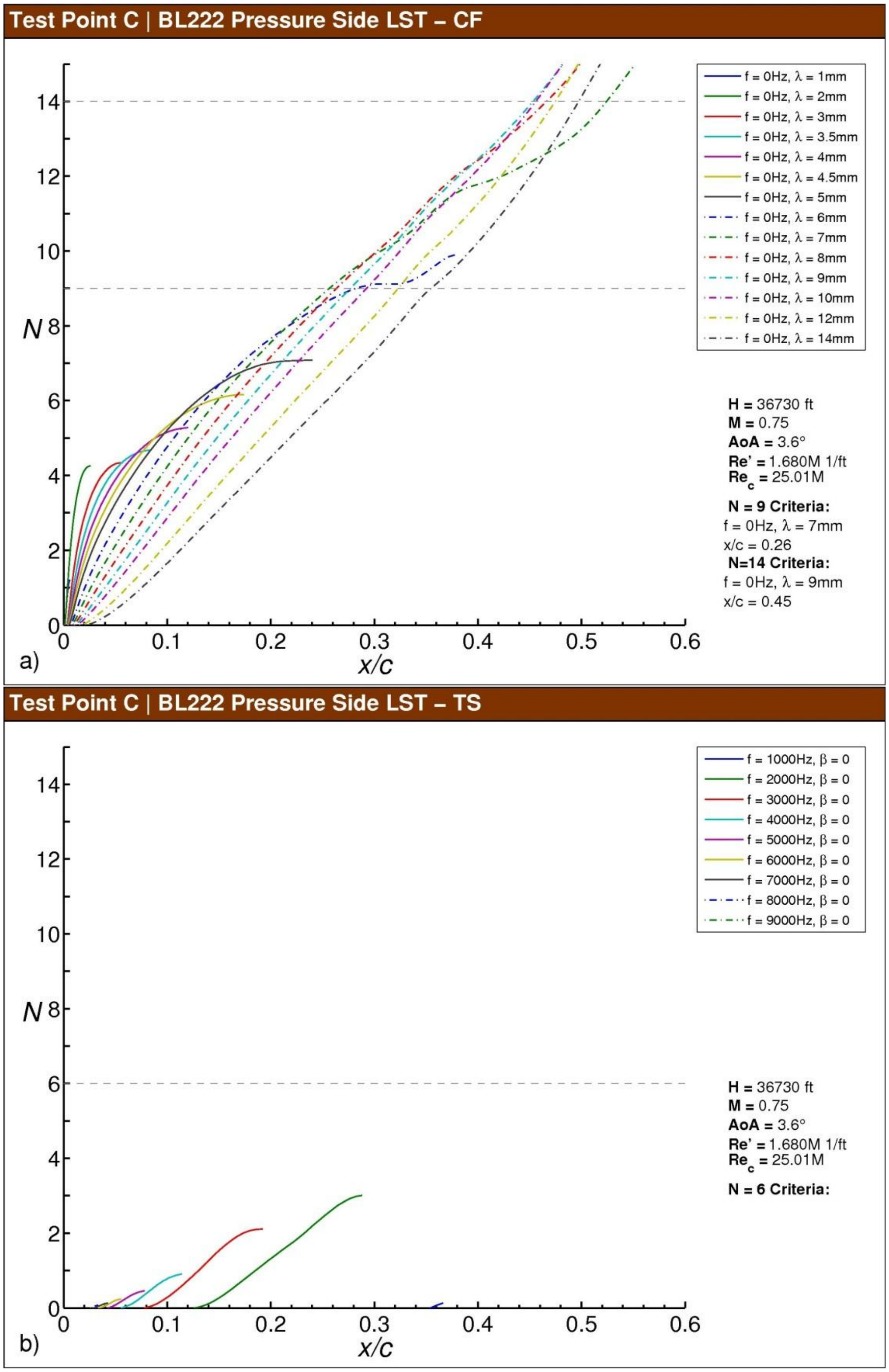
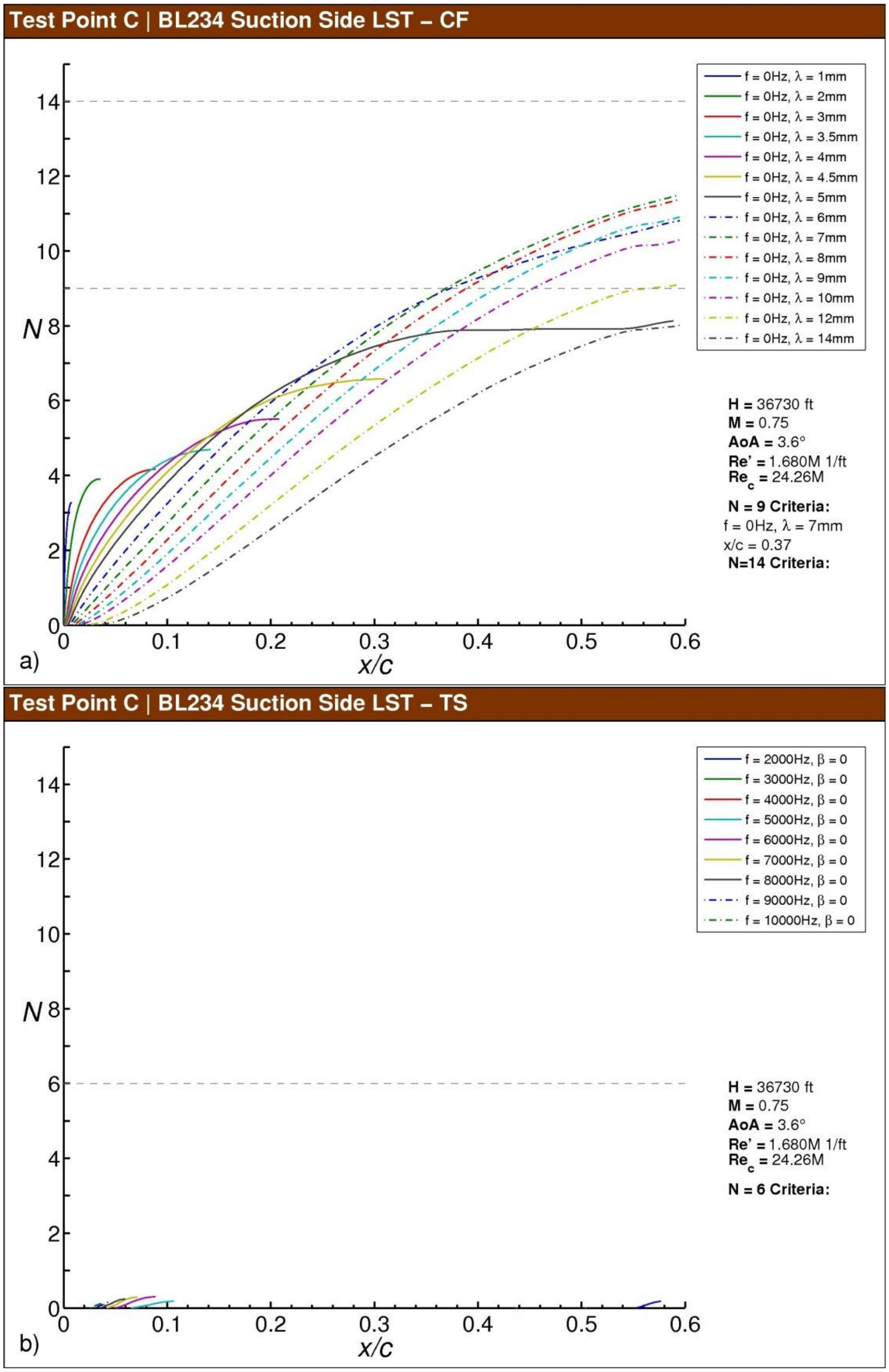


Fig. F-32 Test Point C *N*-factor results for BL222, pressure side: a) stationary crossflow, b) T-S.



Test Point C | BL234 Suction Side LST – CF

Test Point C | BL234 Suction Side LST – TS

Fig. F-33 Test Point C *N*-factor results for BL234, suction side: a) stationary crossflow, b) T-S.

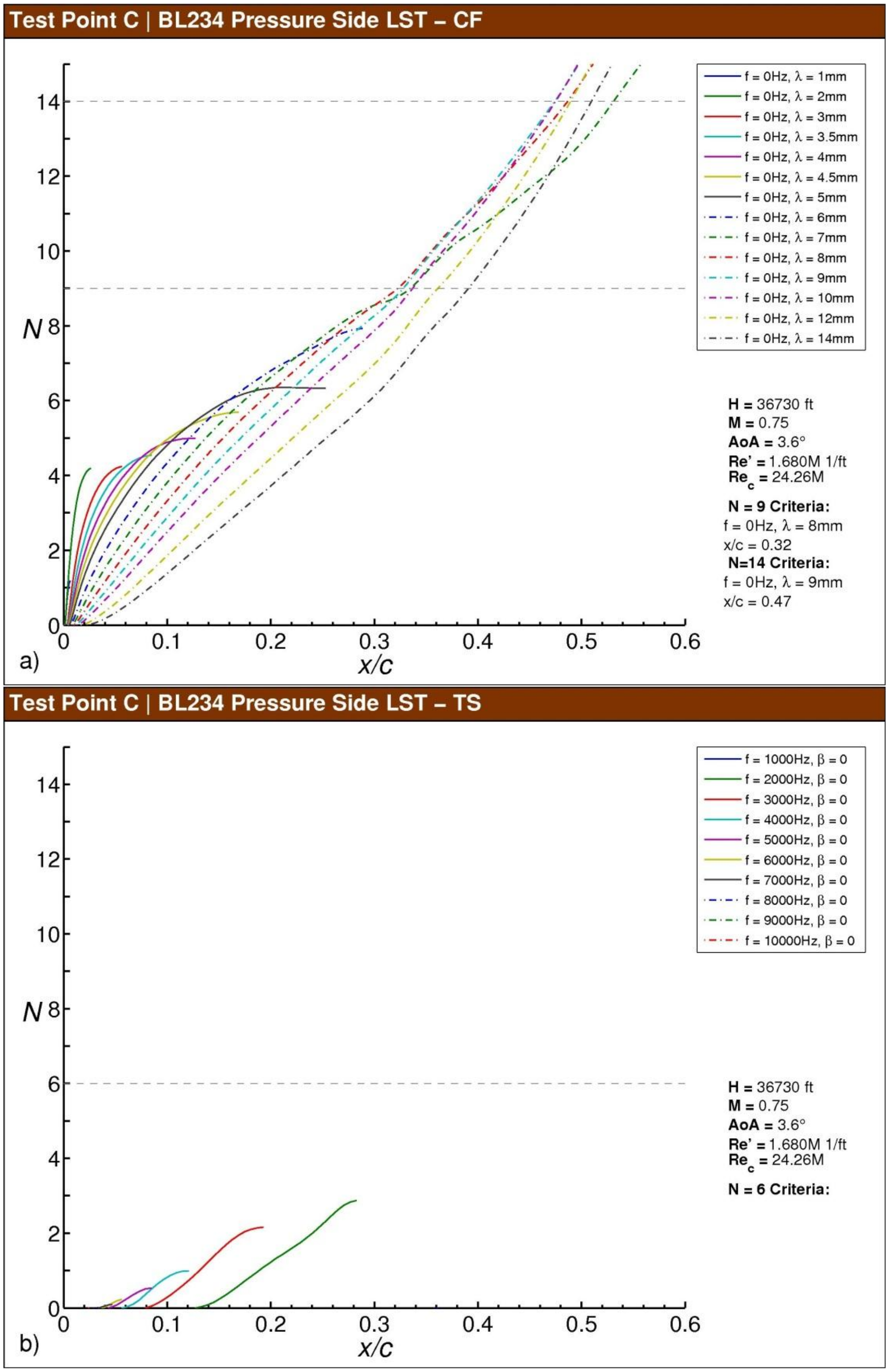


Fig. F-34 Test Point C *N*-factor results for BL234, pressure side: a) stationary crossflow, b) T-S.

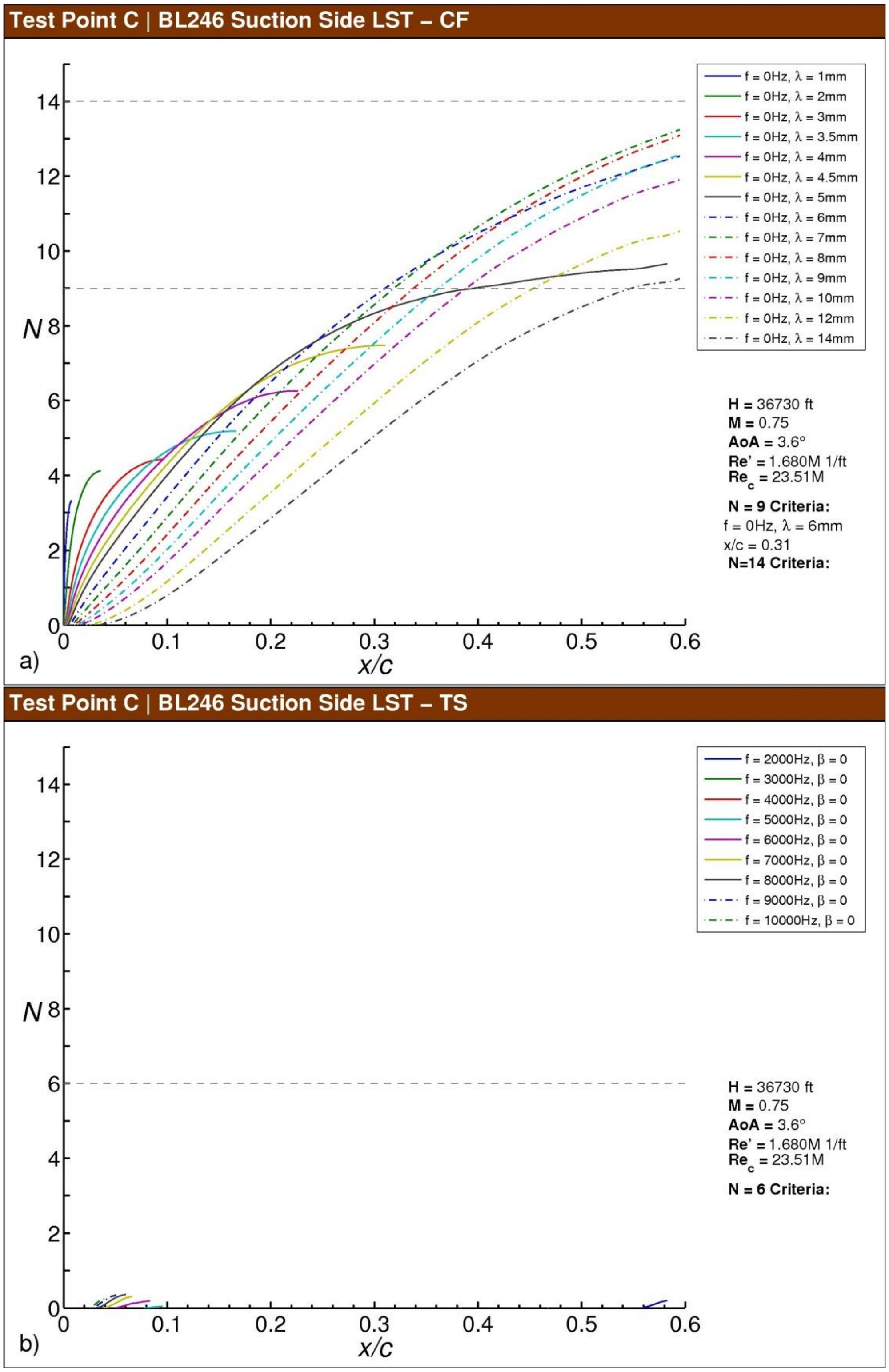
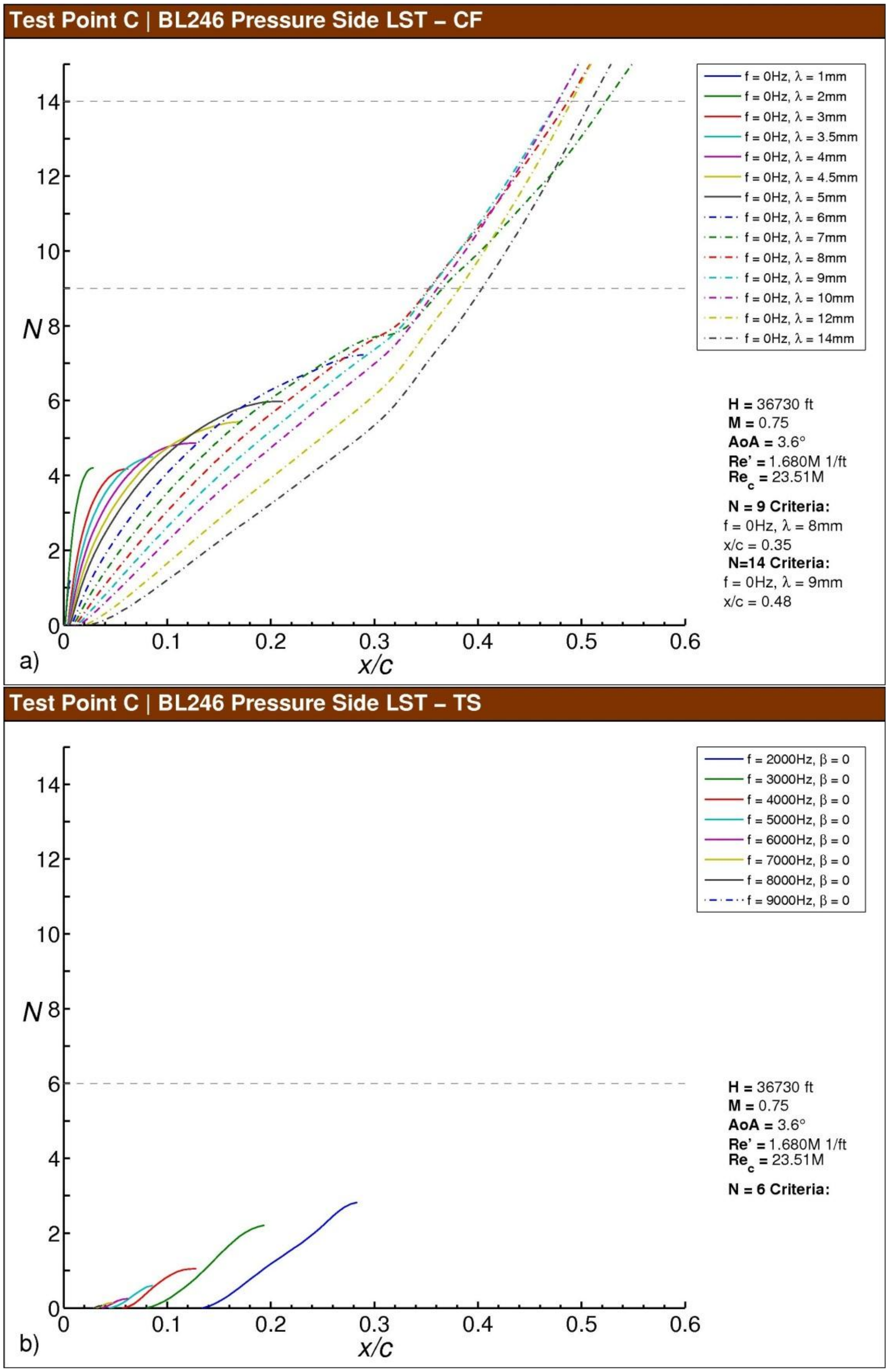


Fig. F-35 Test Point C *N*-factor results for BL246, suction side: a) stationary crossflow, b) T-S.



Test Point C | BL246 Pressure Side LST – CF

Test Point C | BL246 Pressure Side LST – TS

Fig. F-36 Test Point C *N*-factor results for BL246, pressure side: a) stationary crossflow, b) T-S.

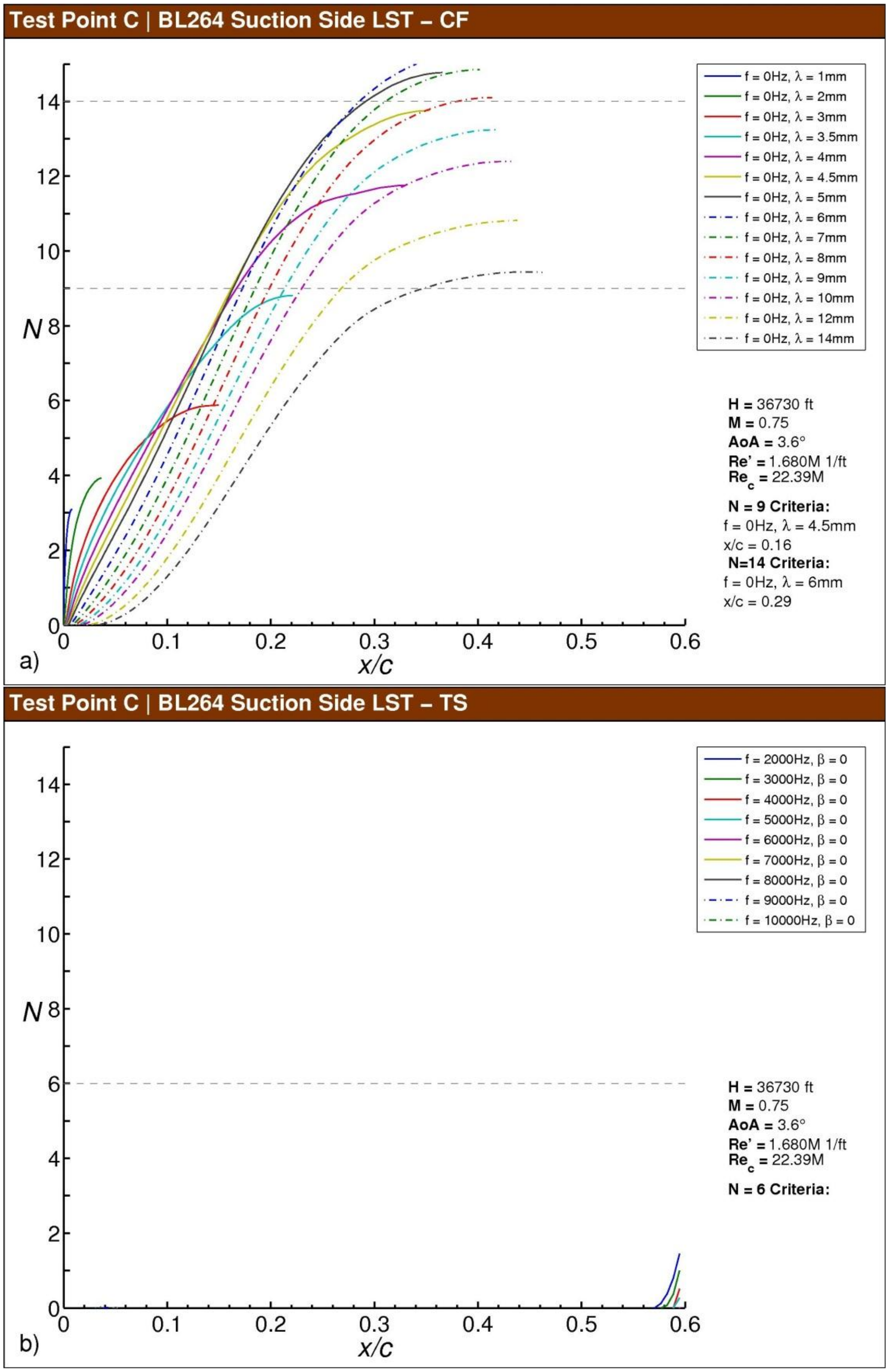


Fig. F-37 Test Point C *N*-factor results for BL264, suction side: a) stationary crossflow, b) T-S.

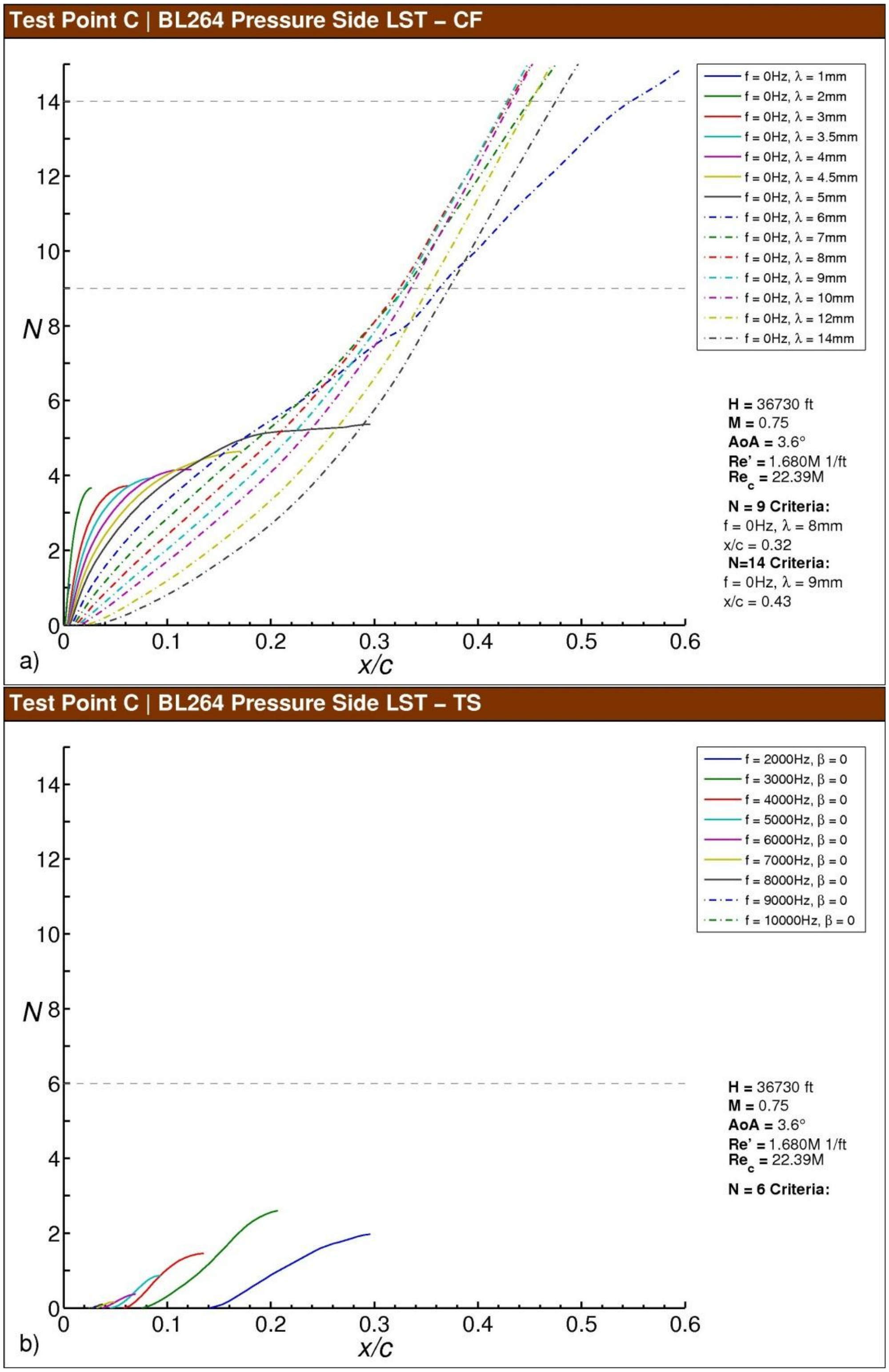


Fig. F-38 Test Point C *N*-factor results for BL264, pressure side: a) stationary crossflow, b) T-S.

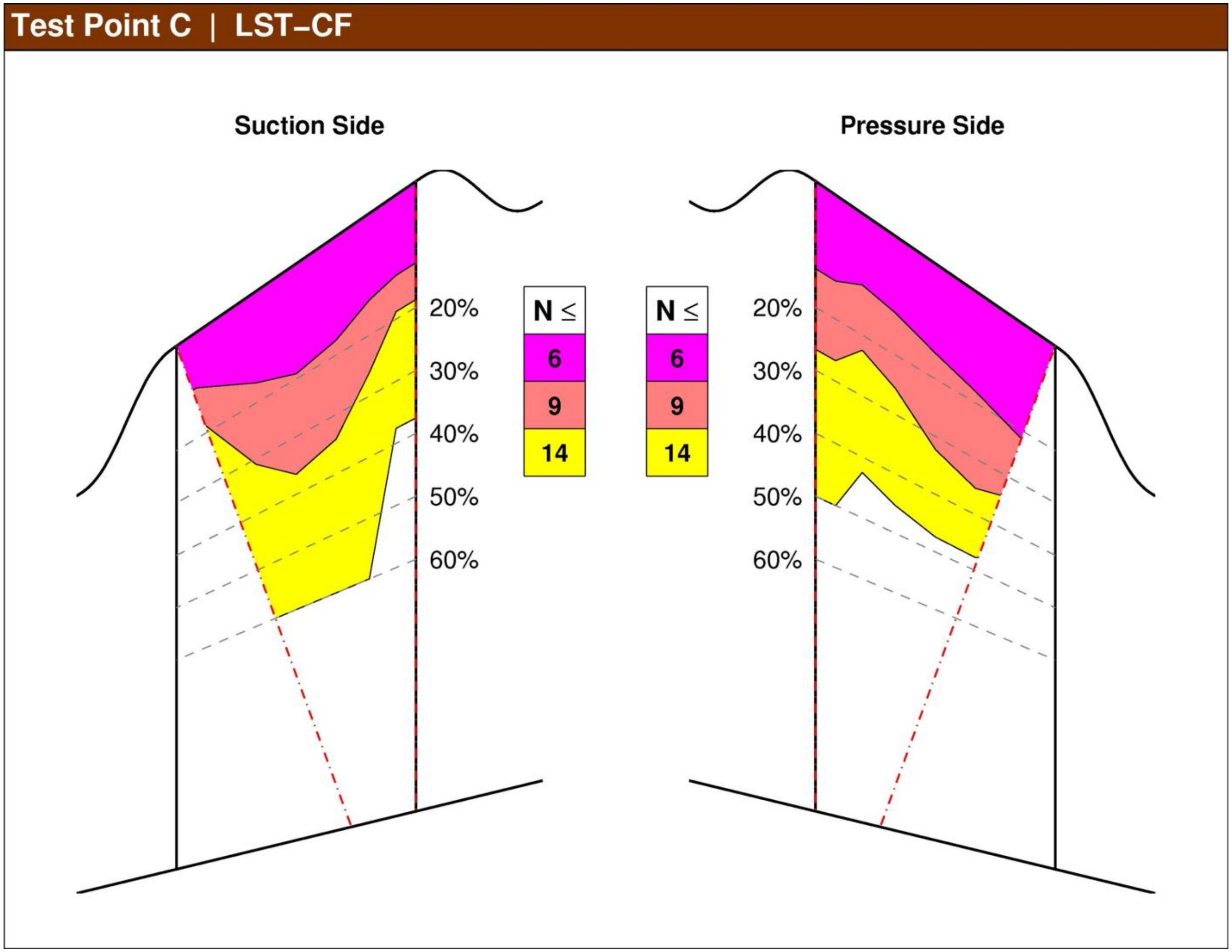


Fig. F-39 Test Point C *N*-factor contours for the test section suction and pressure sides.

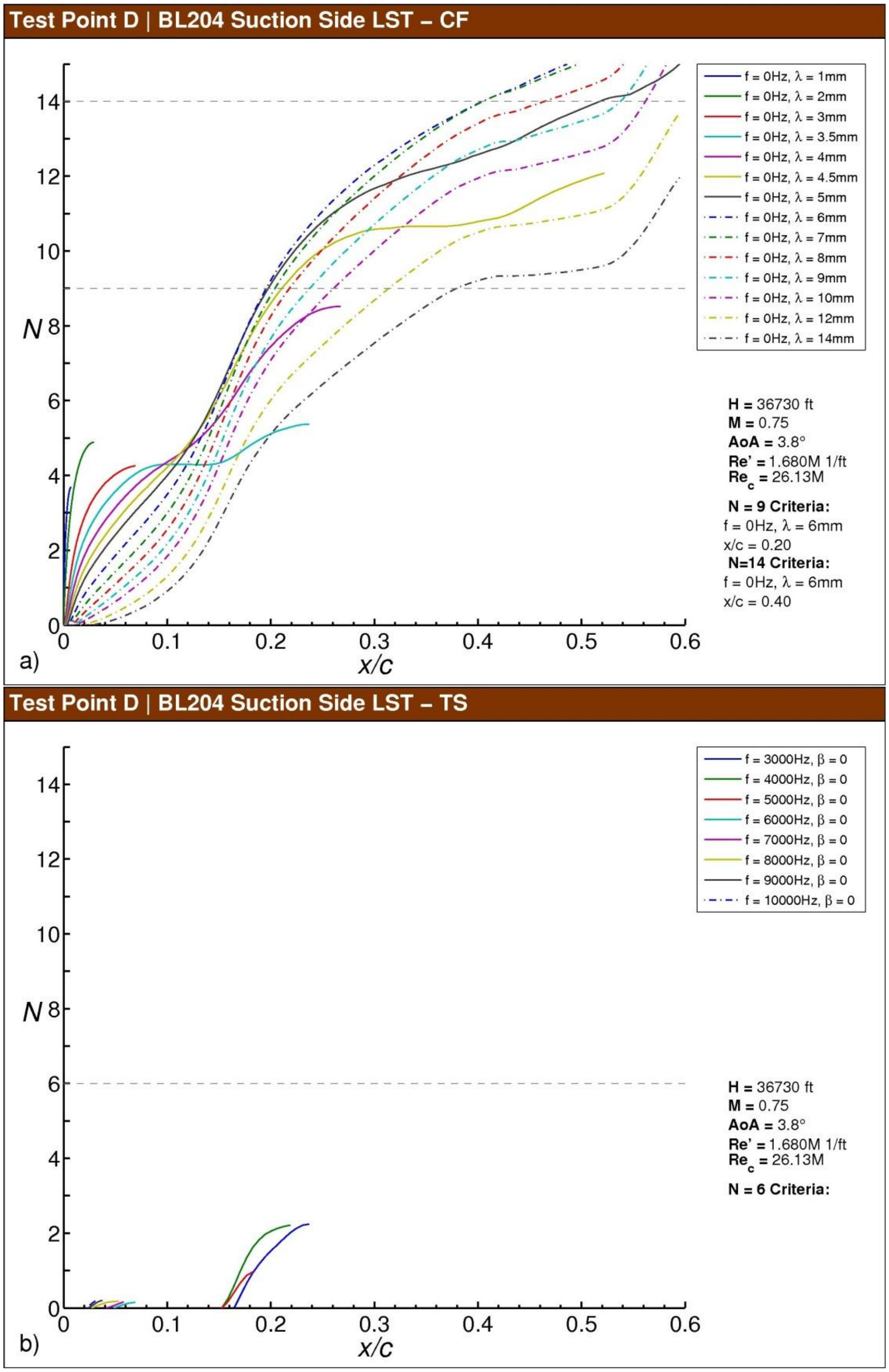


Fig. F-40 Test Point D *N*-factor results for BL204, suction side: a) stationary crossflow, b) T-S.

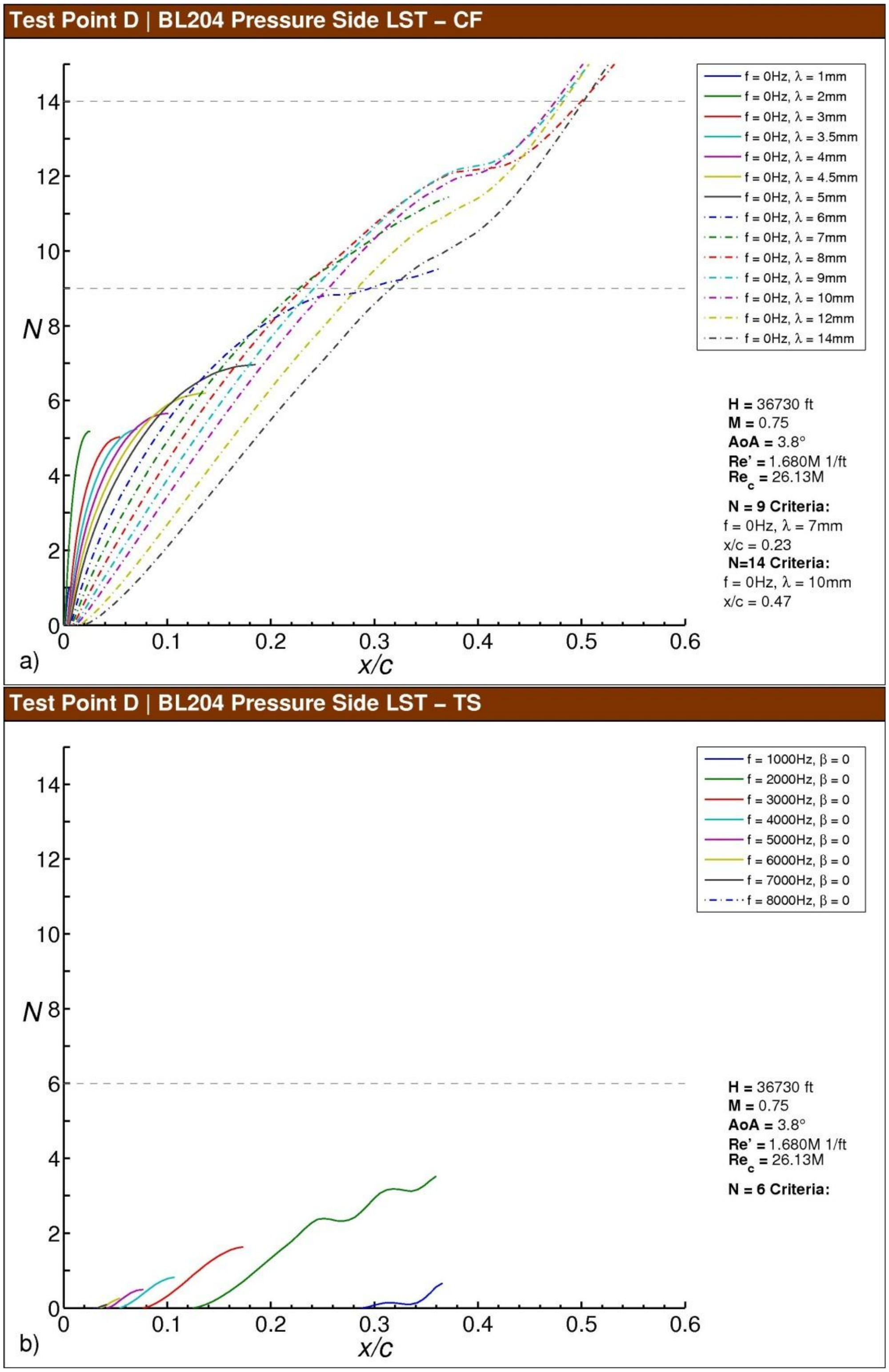


Fig. F-41 Test Point D *N*-factor results for BL204, pressure side: a) stationary crossflow, b) T-S.

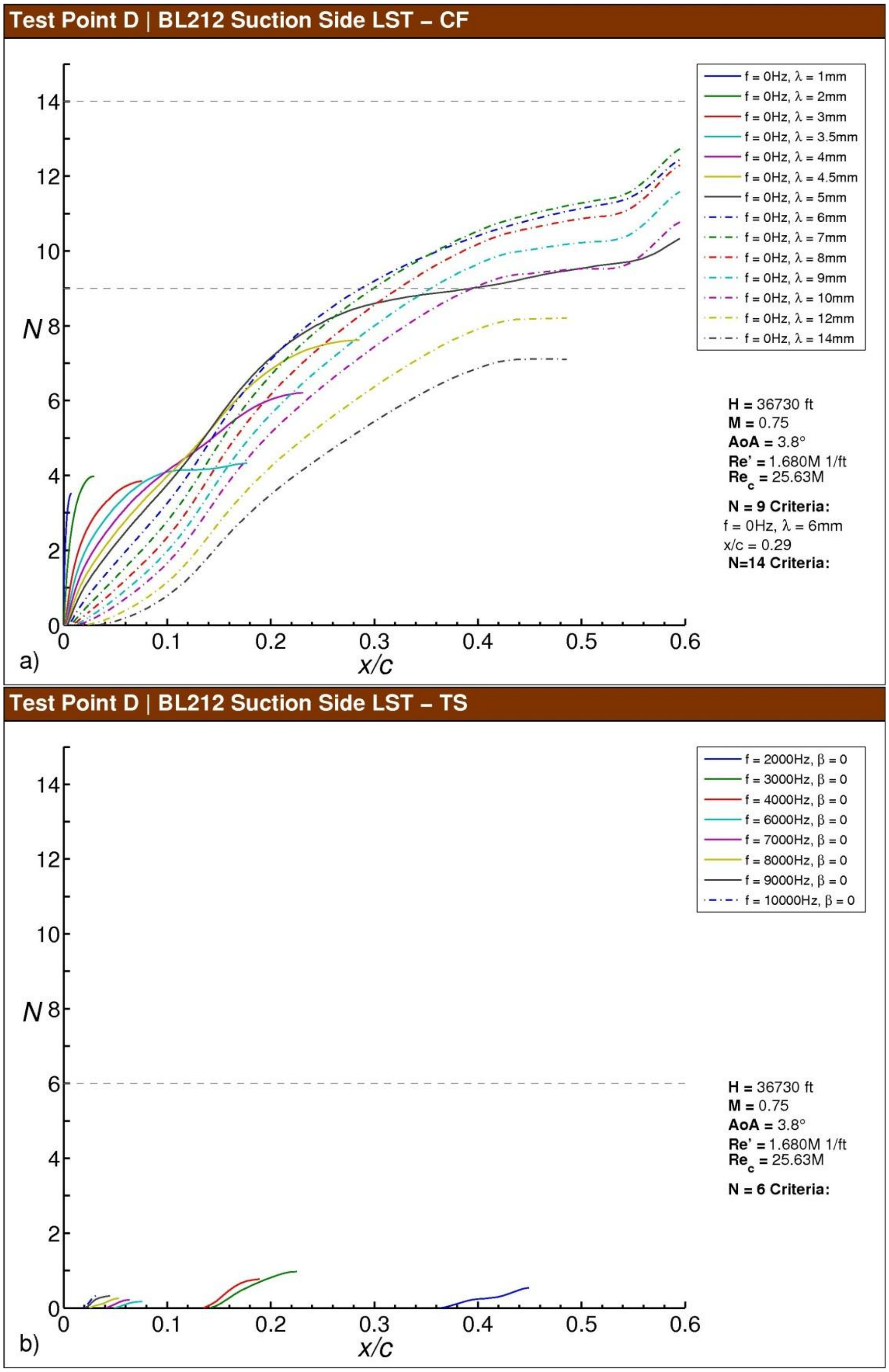
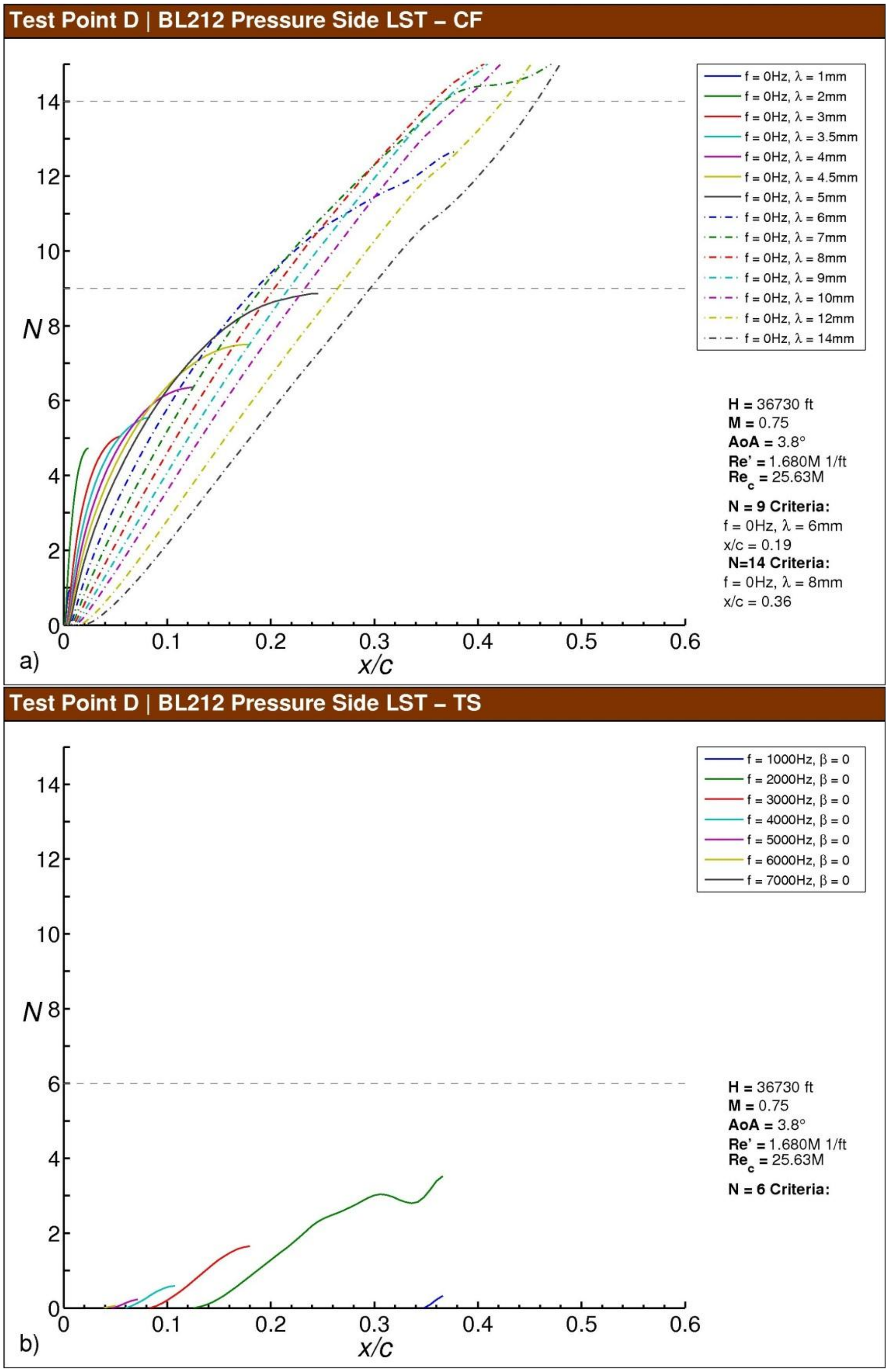


Fig. F-42 Test Point D *N*-factor results for BL212, suction side: a) stationary crossflow, b) T-S.



Test Point D | BL212 Pressure Side LST – CF

H = 36730 ft
M = 0.75
AoA = 3.8°
Re' = 1.680M 1/ft
Re_c = 25.63M

N = 9 Criteria:
f = 0Hz, λ = 6mm
x/c = 0.19
N=14 Criteria:
f = 0Hz, λ = 8mm
x/c = 0.36

Test Point D | BL212 Pressure Side LST – TS

H = 36730 ft
M = 0.75
AoA = 3.8°
Re' = 1.680M 1/ft
Re_c = 25.63M

N = 6 Criteria:

Fig. F-43 Test Point D *N*-factor results for BL212, pressure side: a) stationary crossflow, b) T-S.

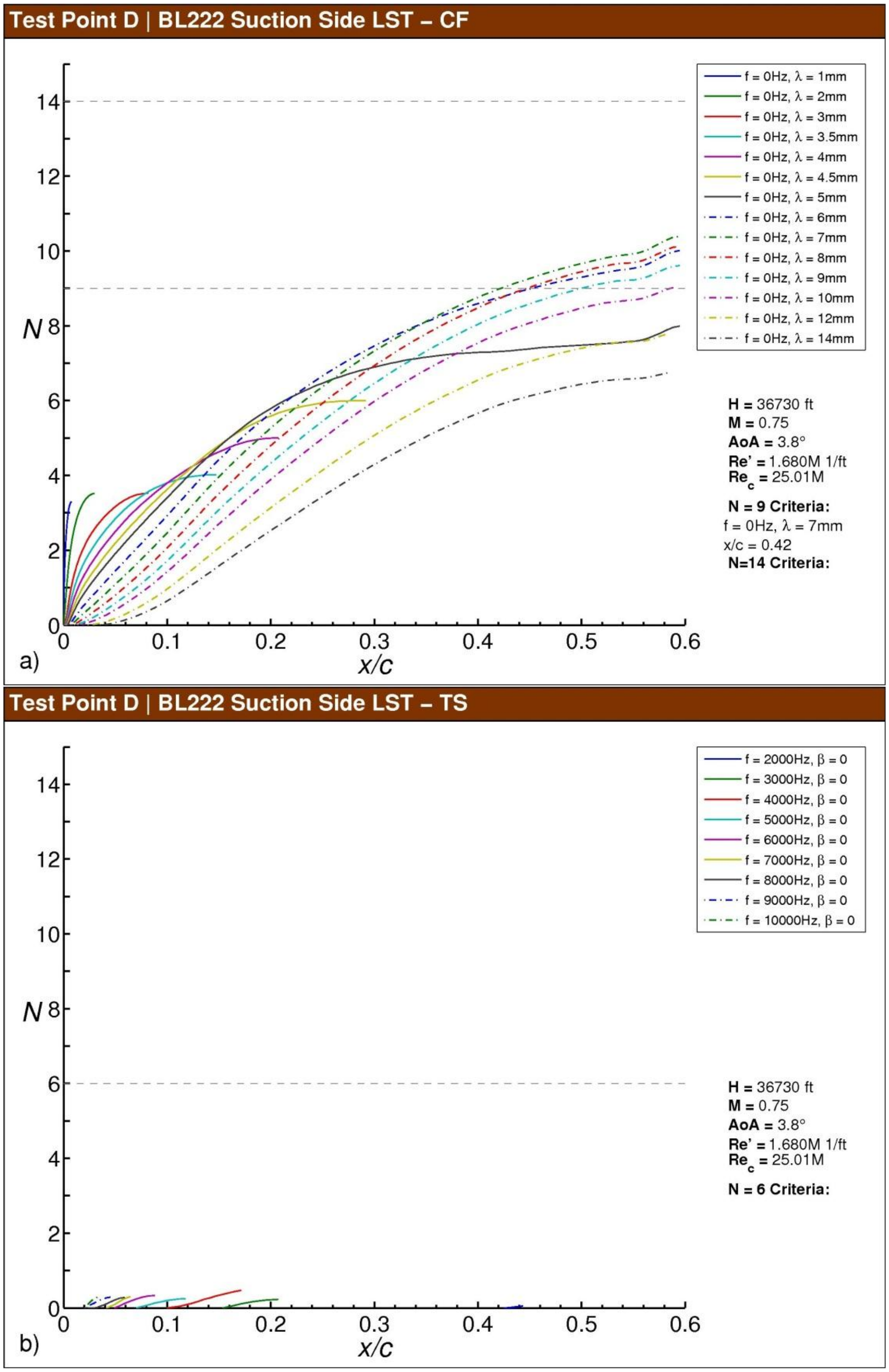


Fig. F-44 Test Point D *N*-factor results for BL222, suction side: a) stationary crossflow, b) T-S.

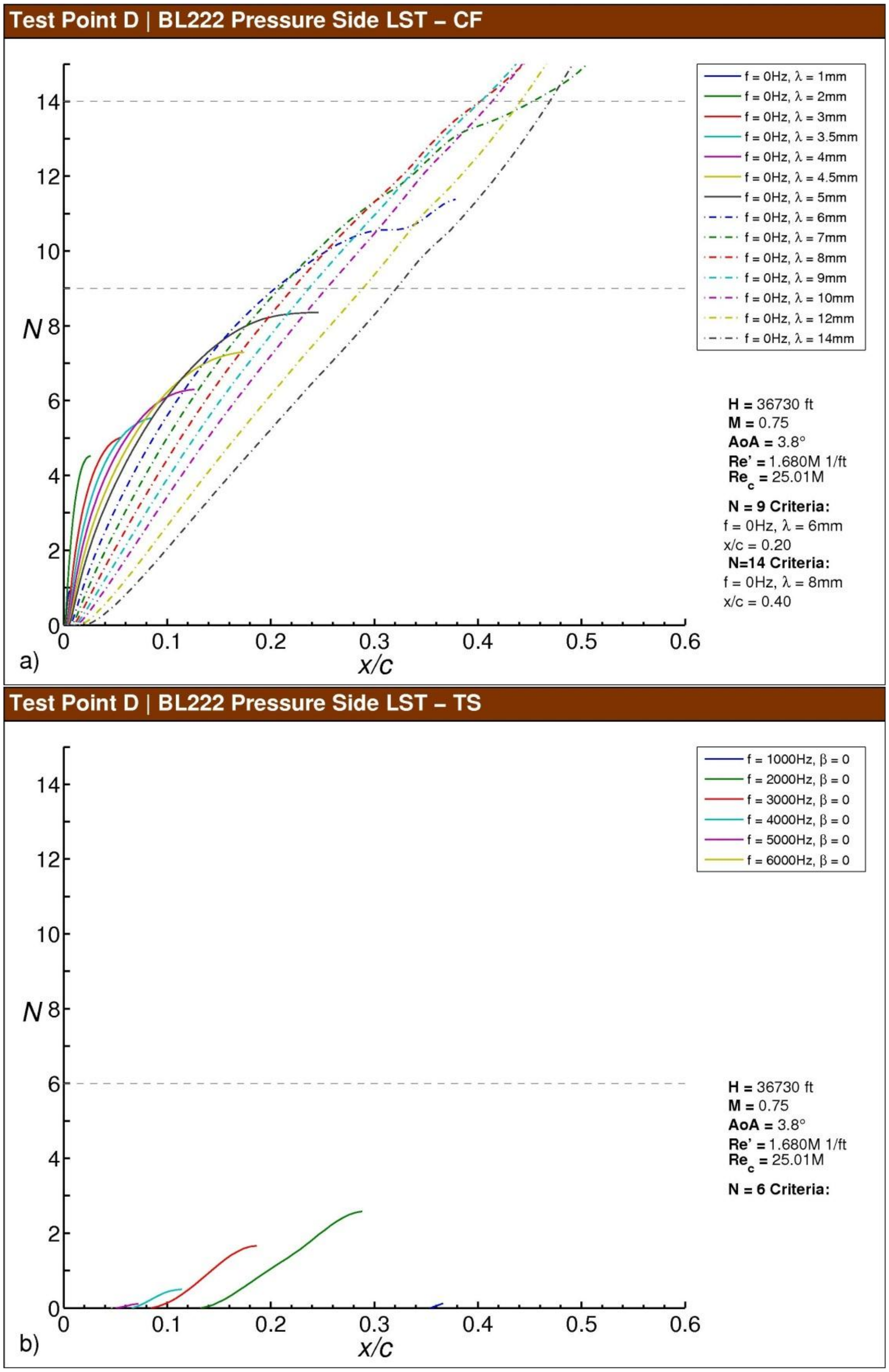


Fig. F-45 Test Point D *N*-factor results for BL222, pressure side: a) stationary crossflow, b) T-S.

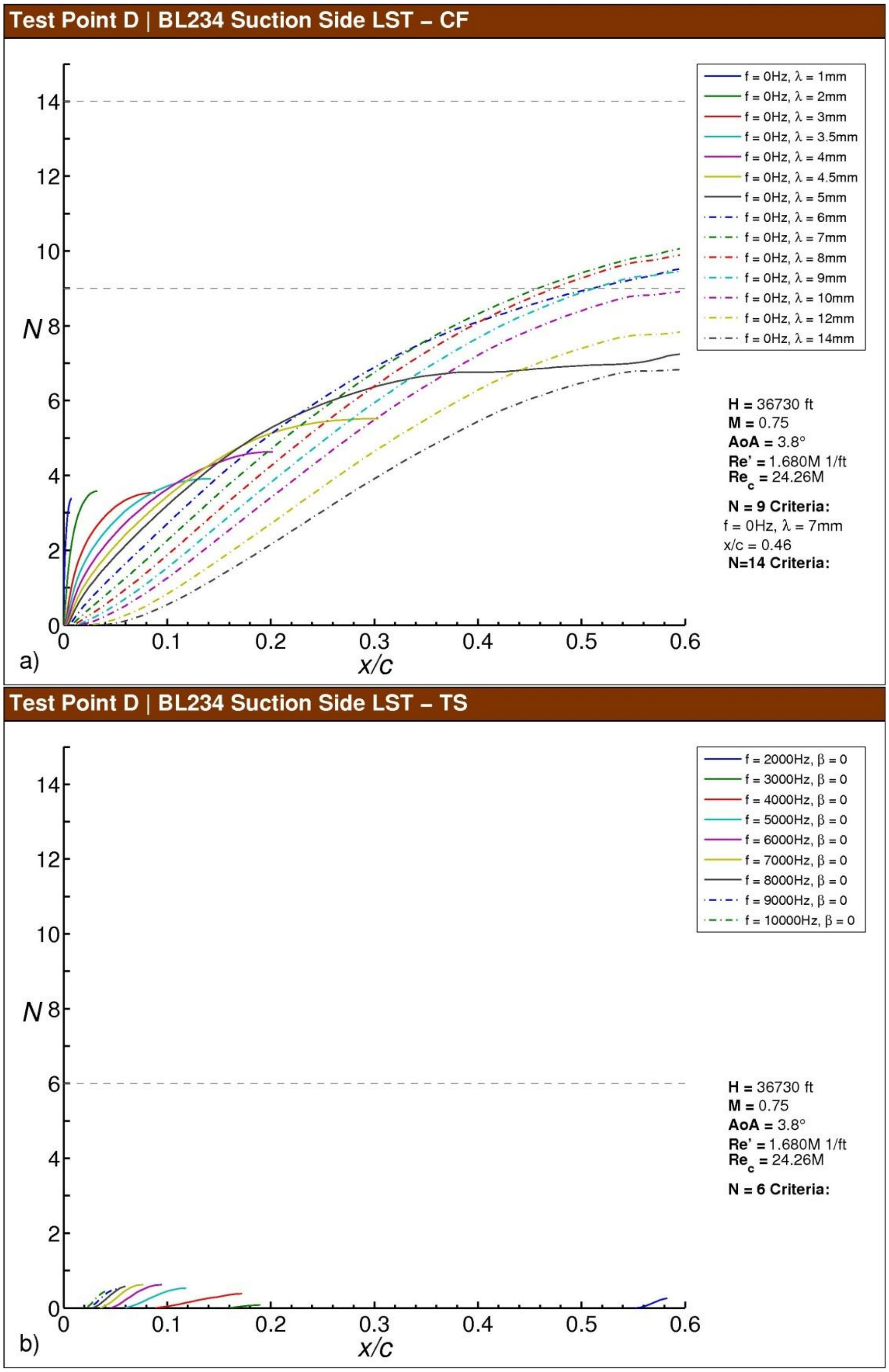


Fig. F-46 Test Point D *N*-factor results for BL234, suction side: a) stationary crossflow, b) T-S.

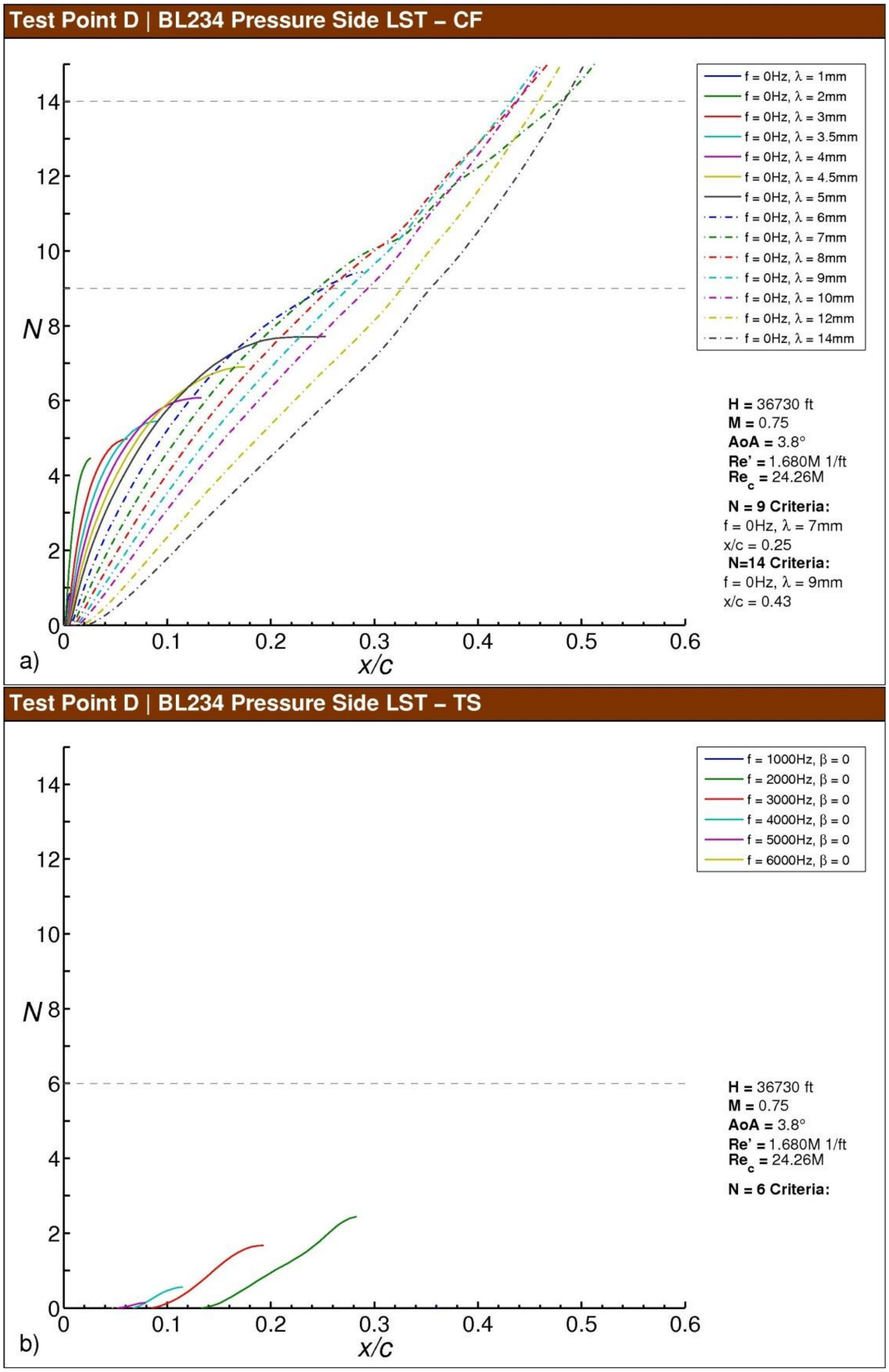


Fig. F-47 Test Point D *N*-factor results for BL234, pressure side: a) stationary crossflow, b) T-S.

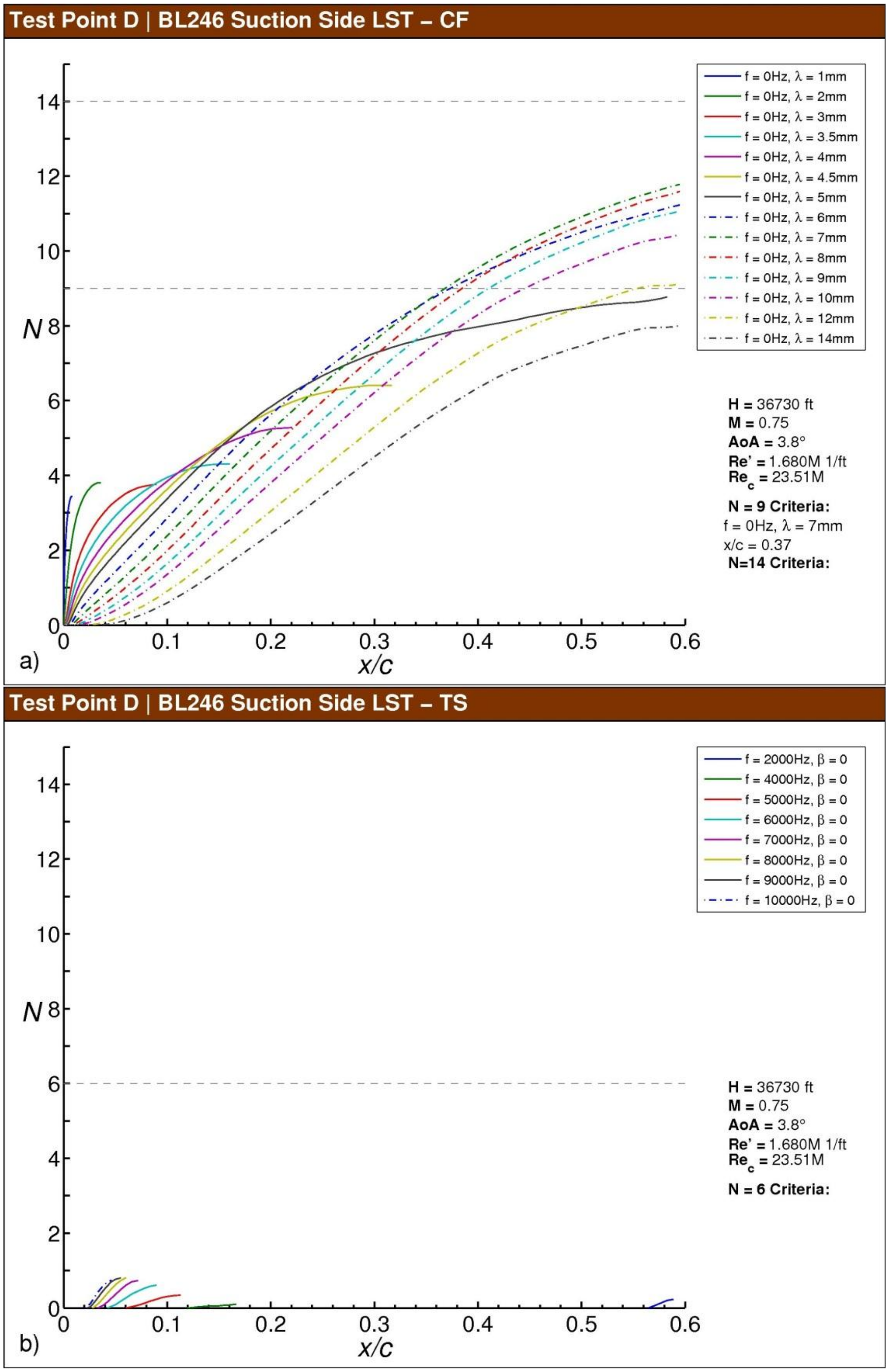


Fig. F-48 Test Point D *N*-factor results for BL246, suction side: a) stationary crossflow, b) T-S.

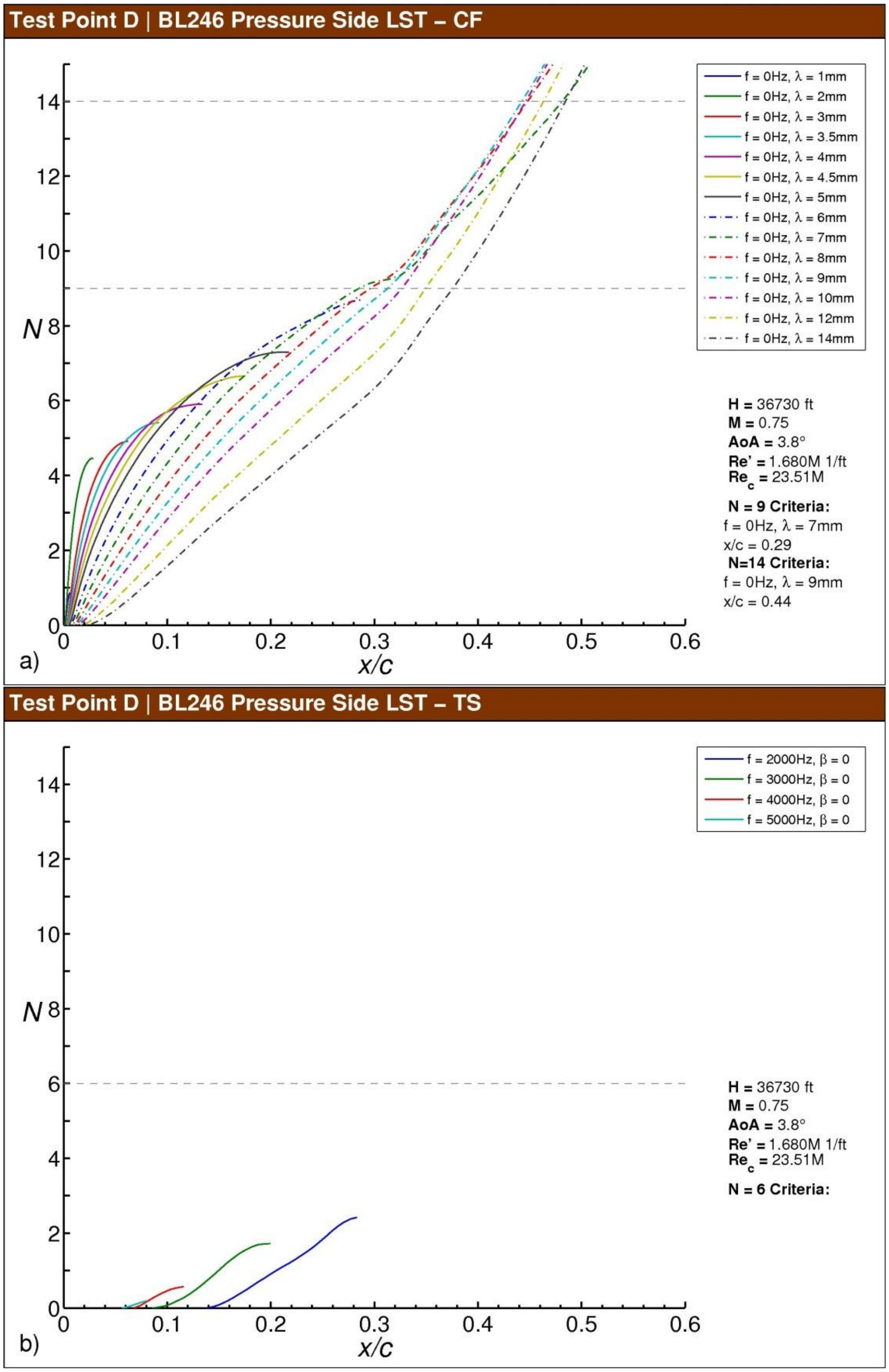


Fig. F-49 Test Point D *N*-factor results for BL246, pressure side: a) stationary crossflow, b) T-S.

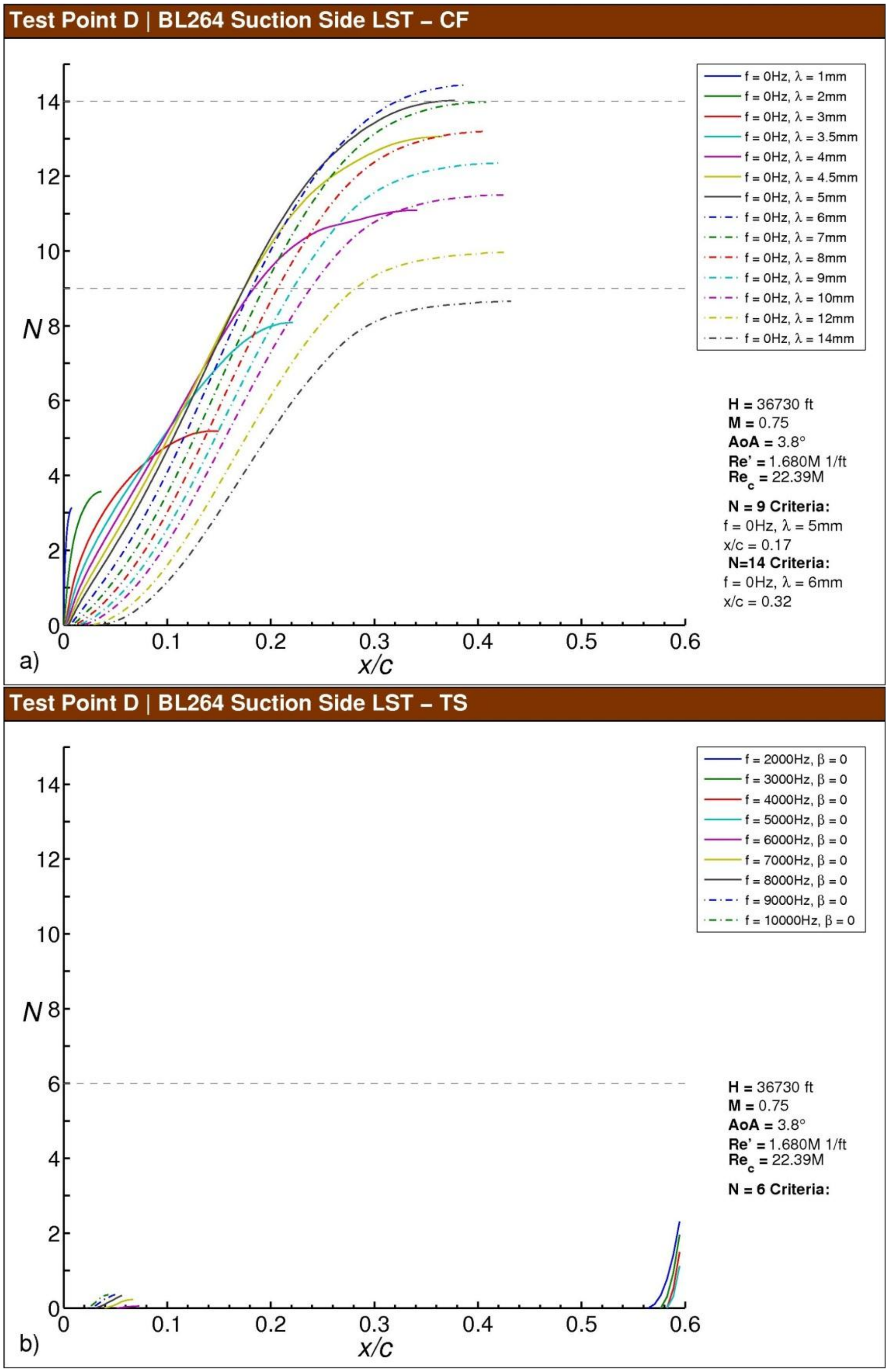


Fig. F-50 Test Point D *N*-factor results for BL264, suction side: a) stationary crossflow, b) T-S.

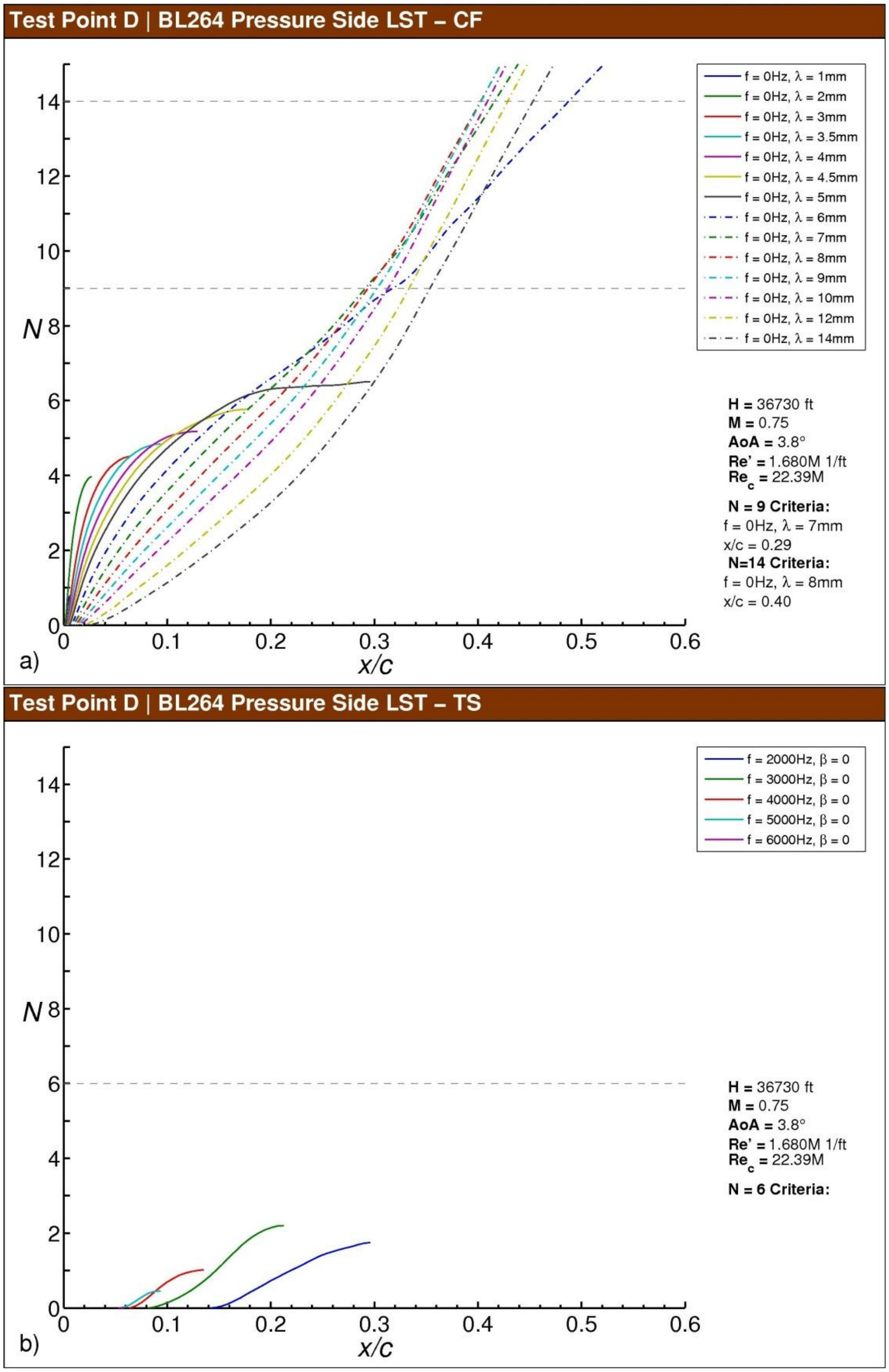


Fig. F-51 Test Point D *N*-factor results for BL264, pressure side: a) stationary crossflow, b) T-S.

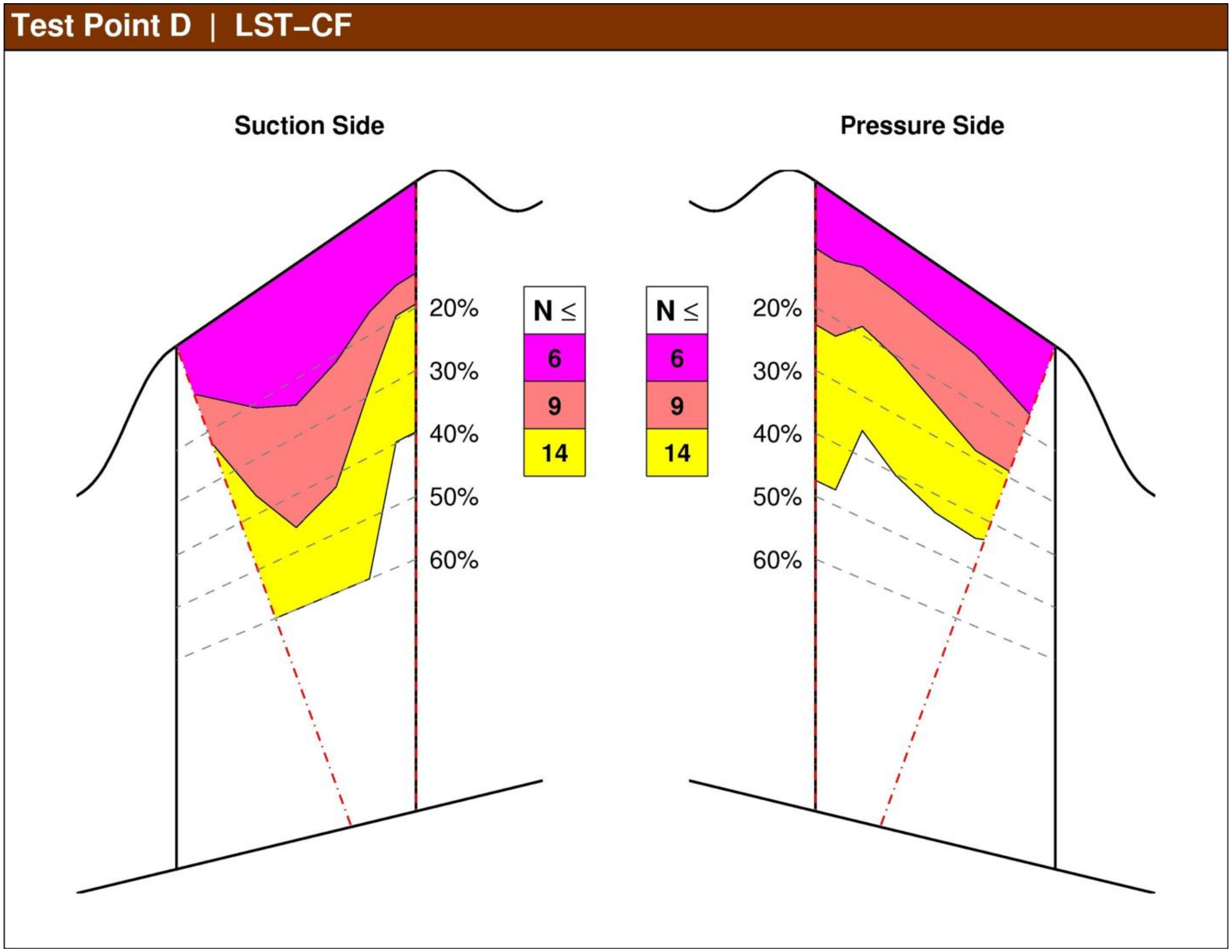


Fig. F-52 Test Point D *N*-factor contours for the test section suction and pressure sides.

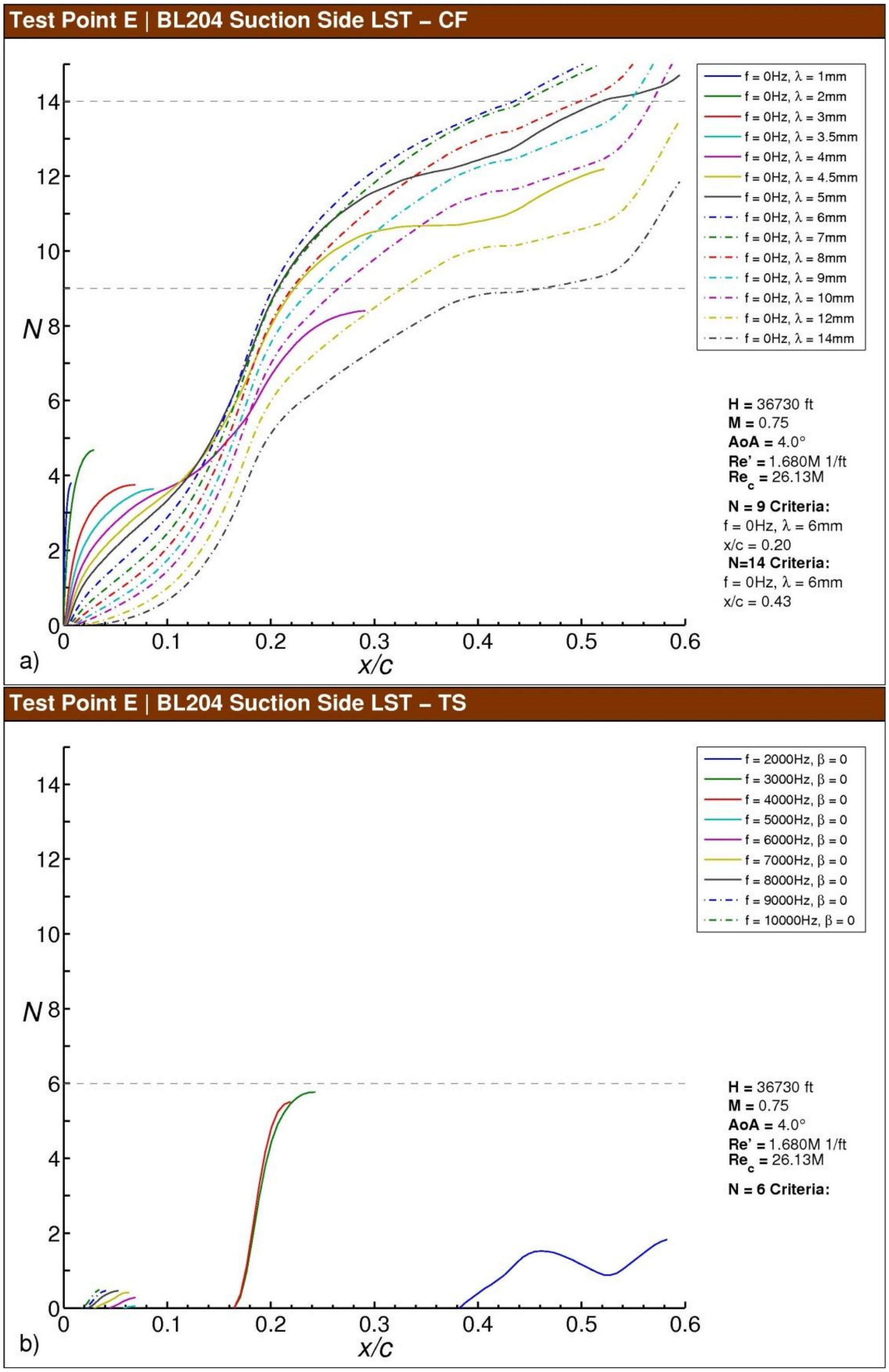


Fig. F-53 Test Point E *N*-factor results for BL204, suction side: a) stationary crossflow, b) T-S.

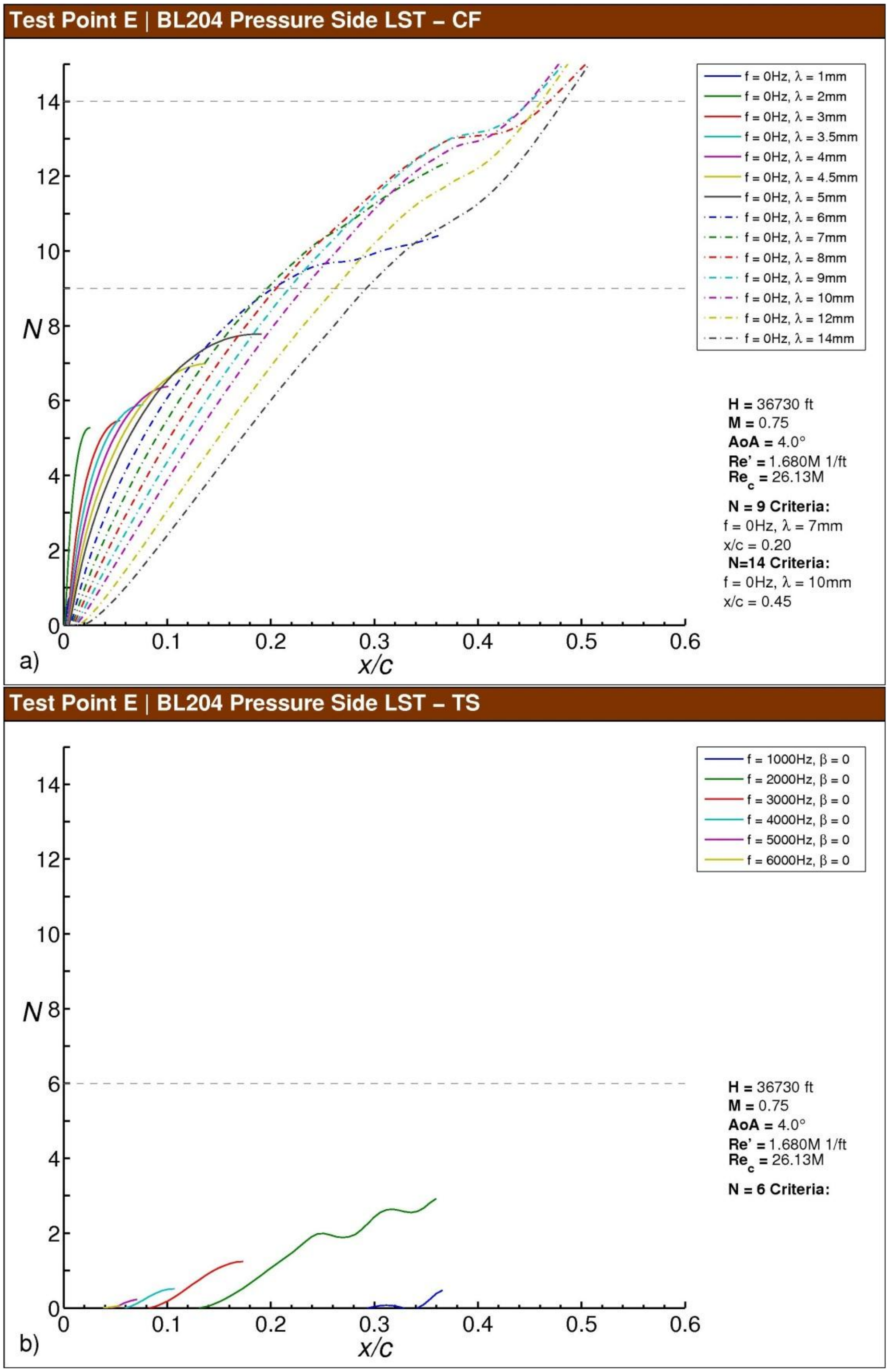


Fig. F-54 Test Point E *N*-factor results for BL204, pressure side: a) stationary crossflow, b) T-S.

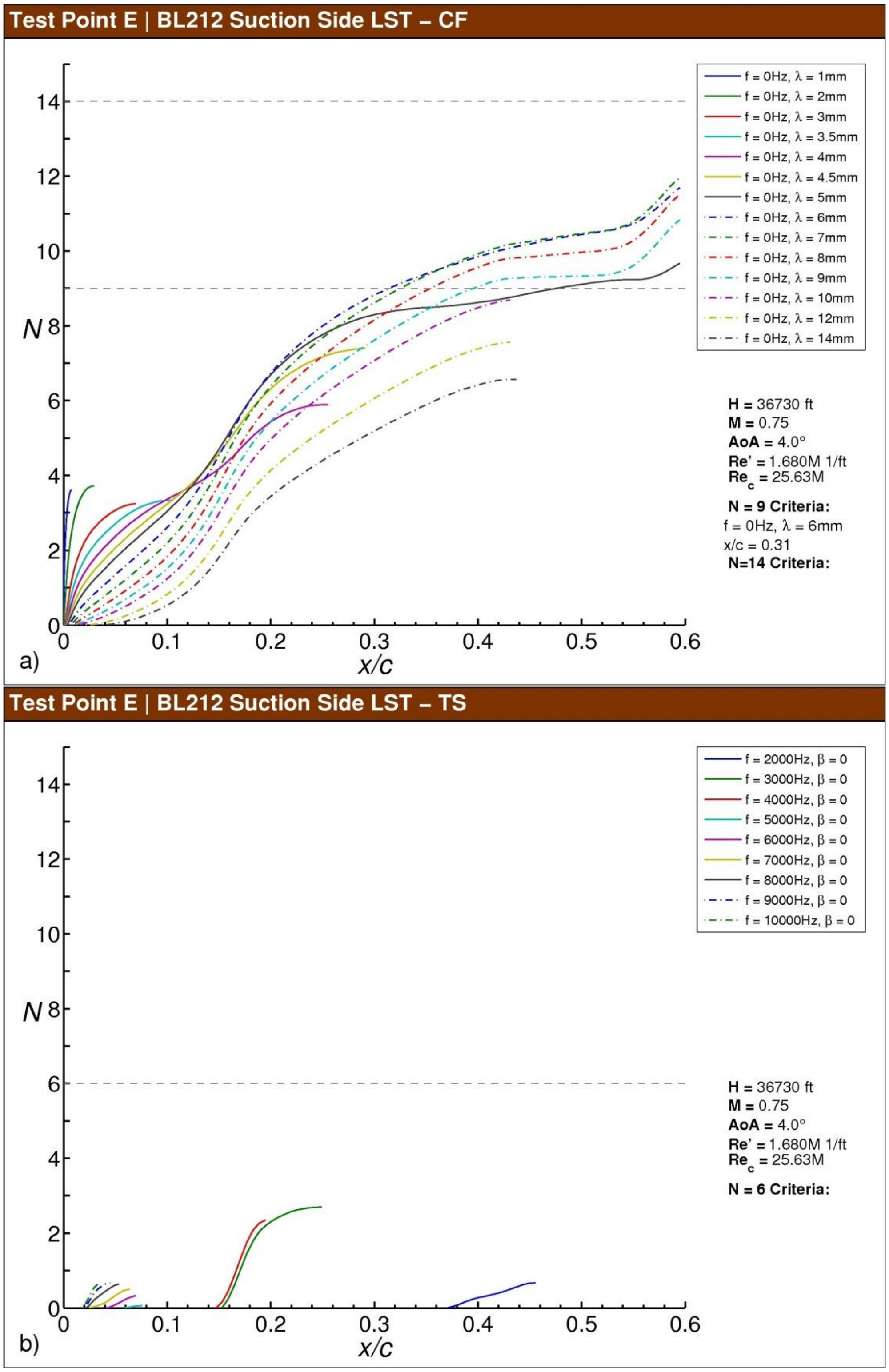


Fig. F-55 Test Point E *N*-factor results for BL212, suction side: a) stationary crossflow, b) T-S.

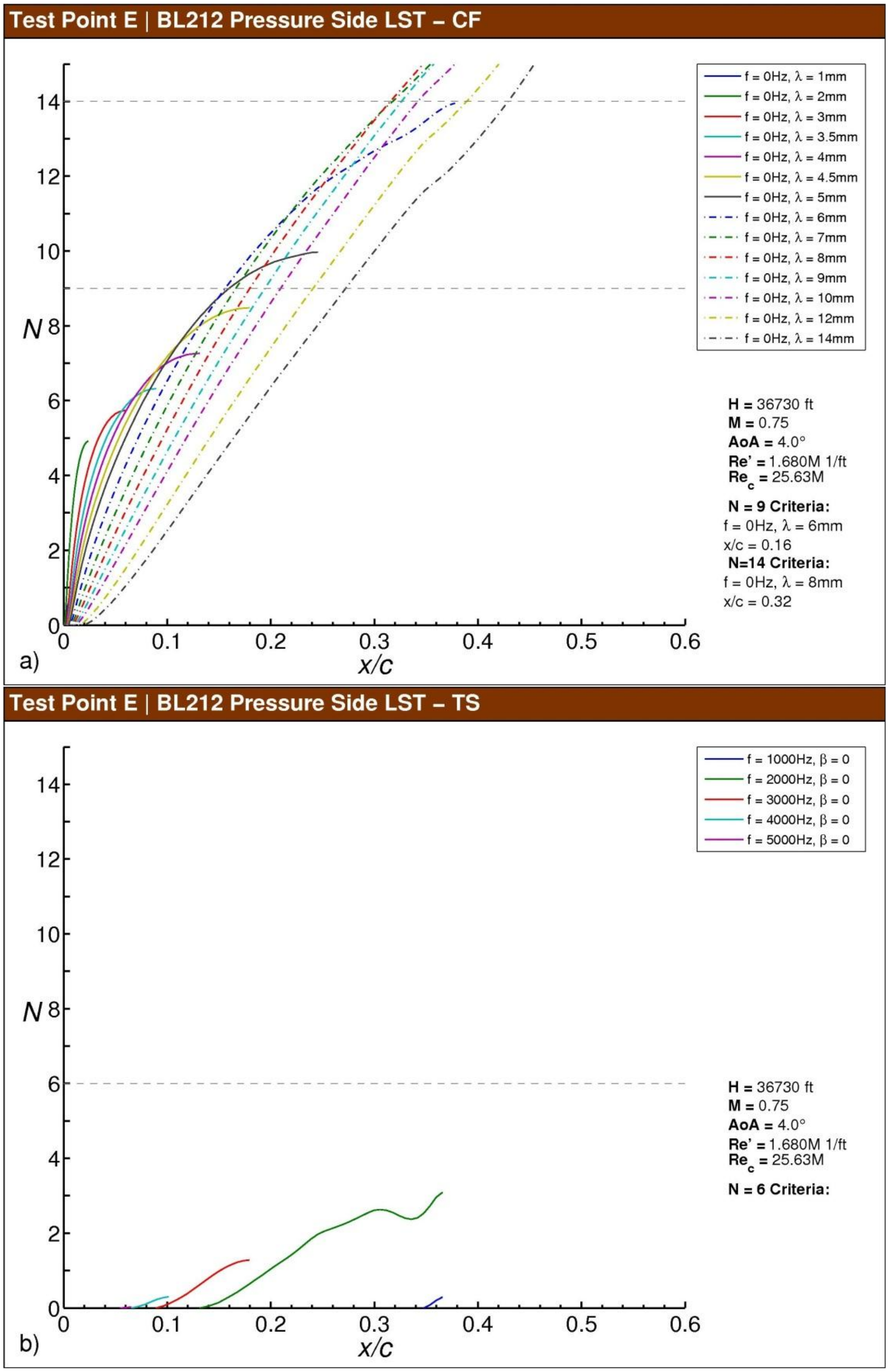


Fig. F-56 Test Point E *N*-factor results for BL212, pressure side: a) stationary crossflow, b) T-S.

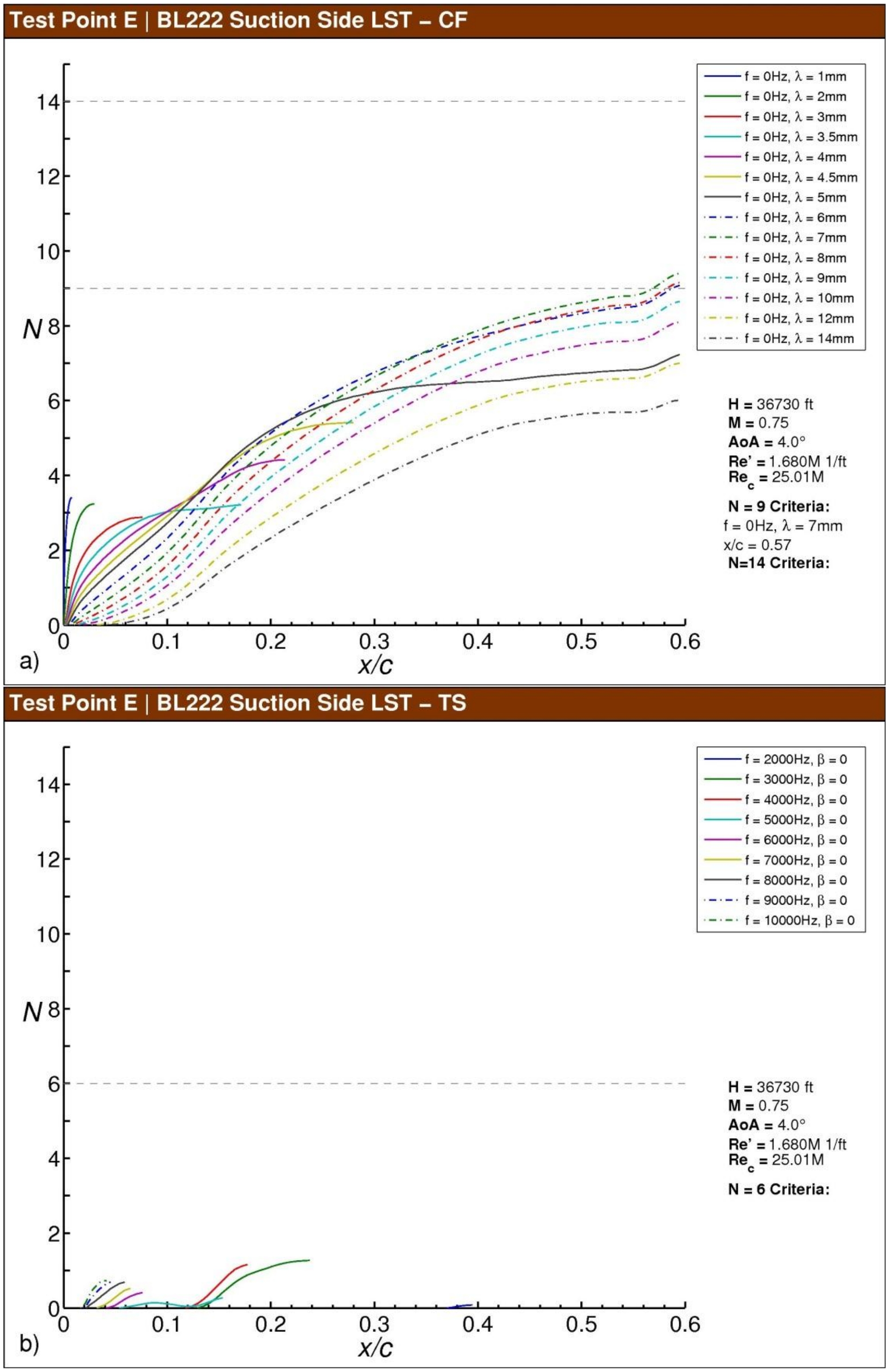


Fig. F-57 Test Point E *N*-factor results for BL222, suction side: a) stationary crossflow, b) T-S.

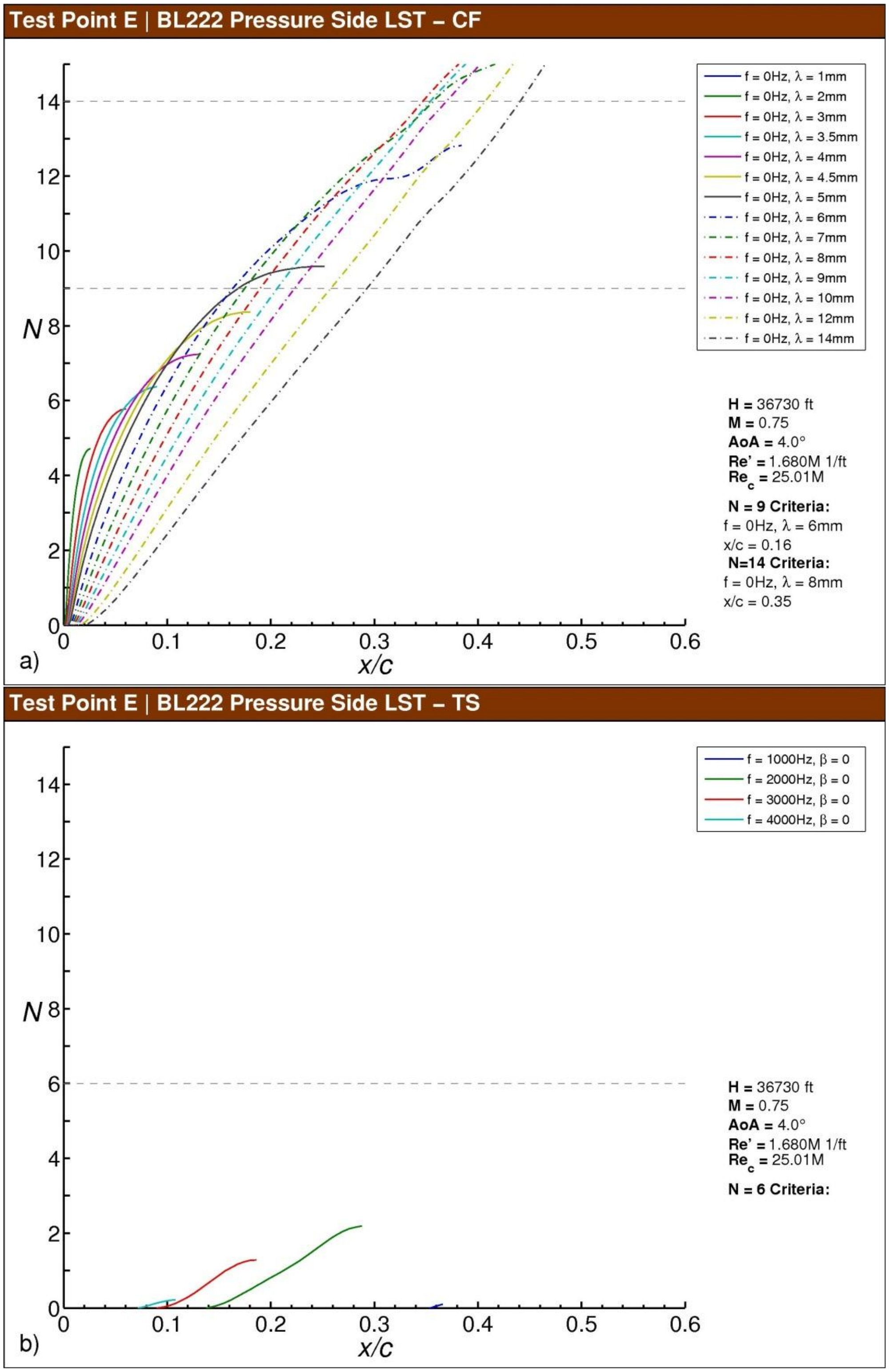


Fig. F-58 Test Point E *N*-factor results for BL222, pressure side: a) stationary crossflow, b) T-S.

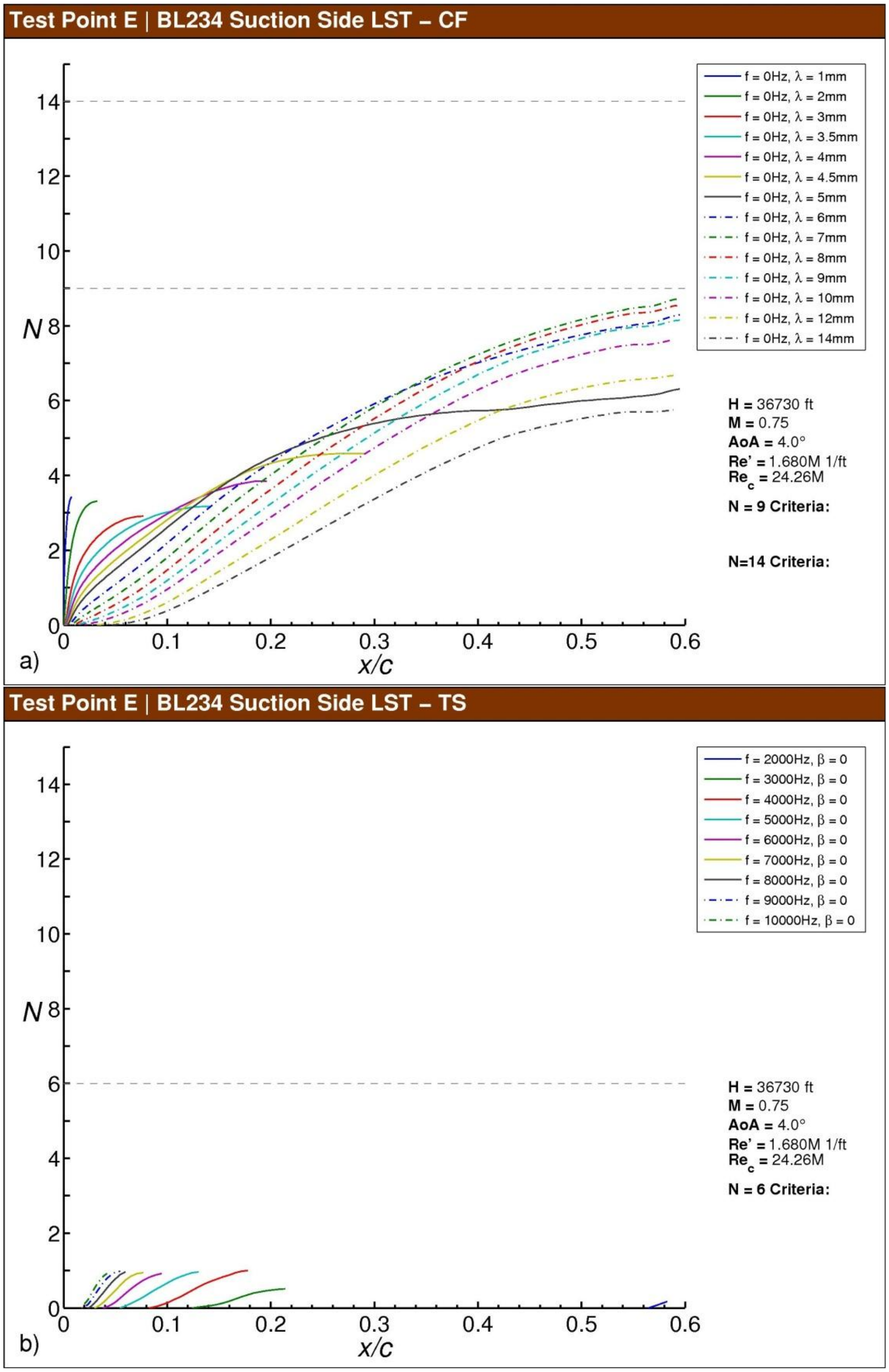


Fig. F-59 Test Point E *N*-factor results for BL234, suction side: a) stationary crossflow, b) T-S.

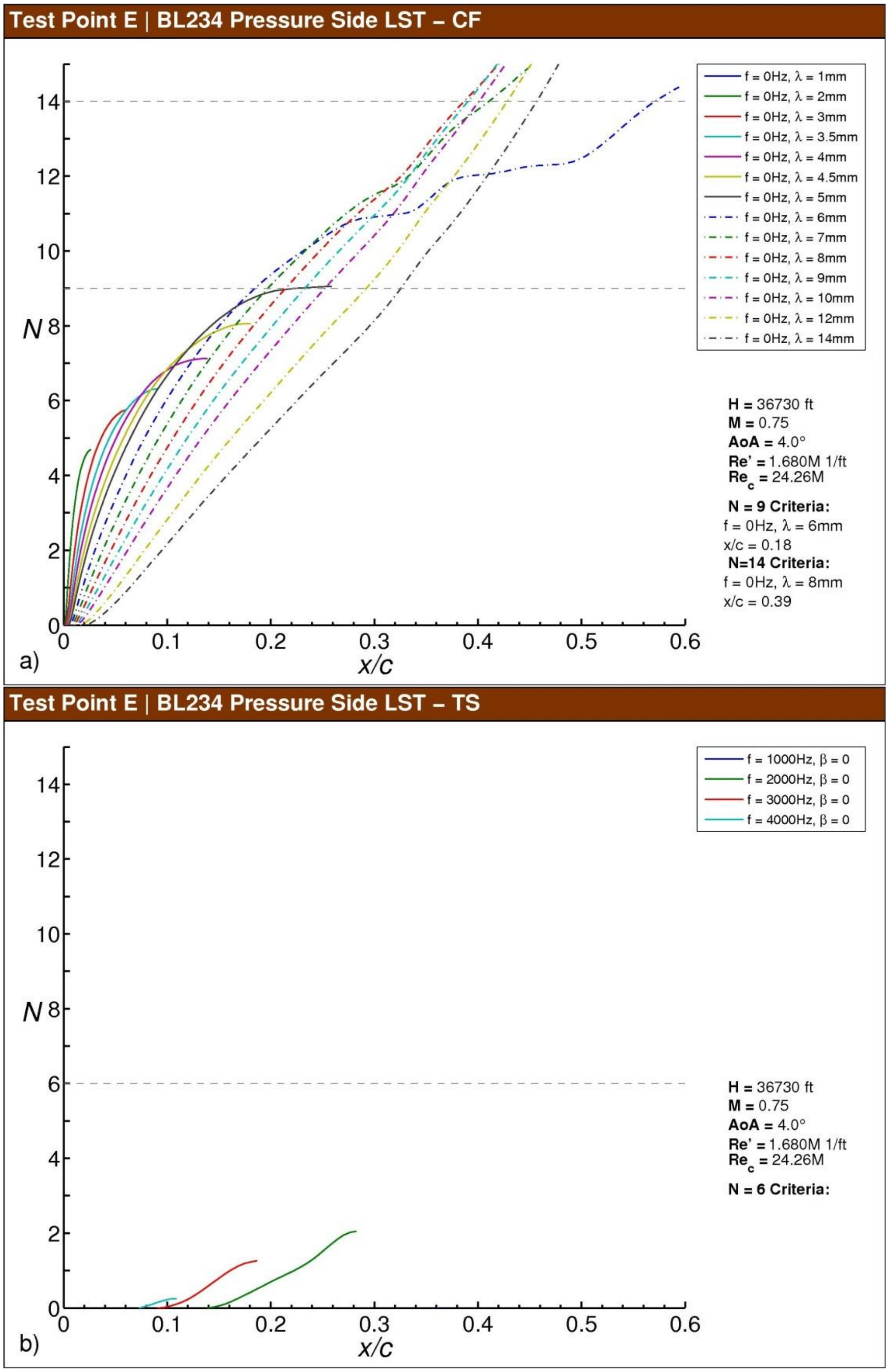


Fig. F-60 Test Point E *N*-factor results for BL234, pressure side: a) stationary crossflow, b) T-S.

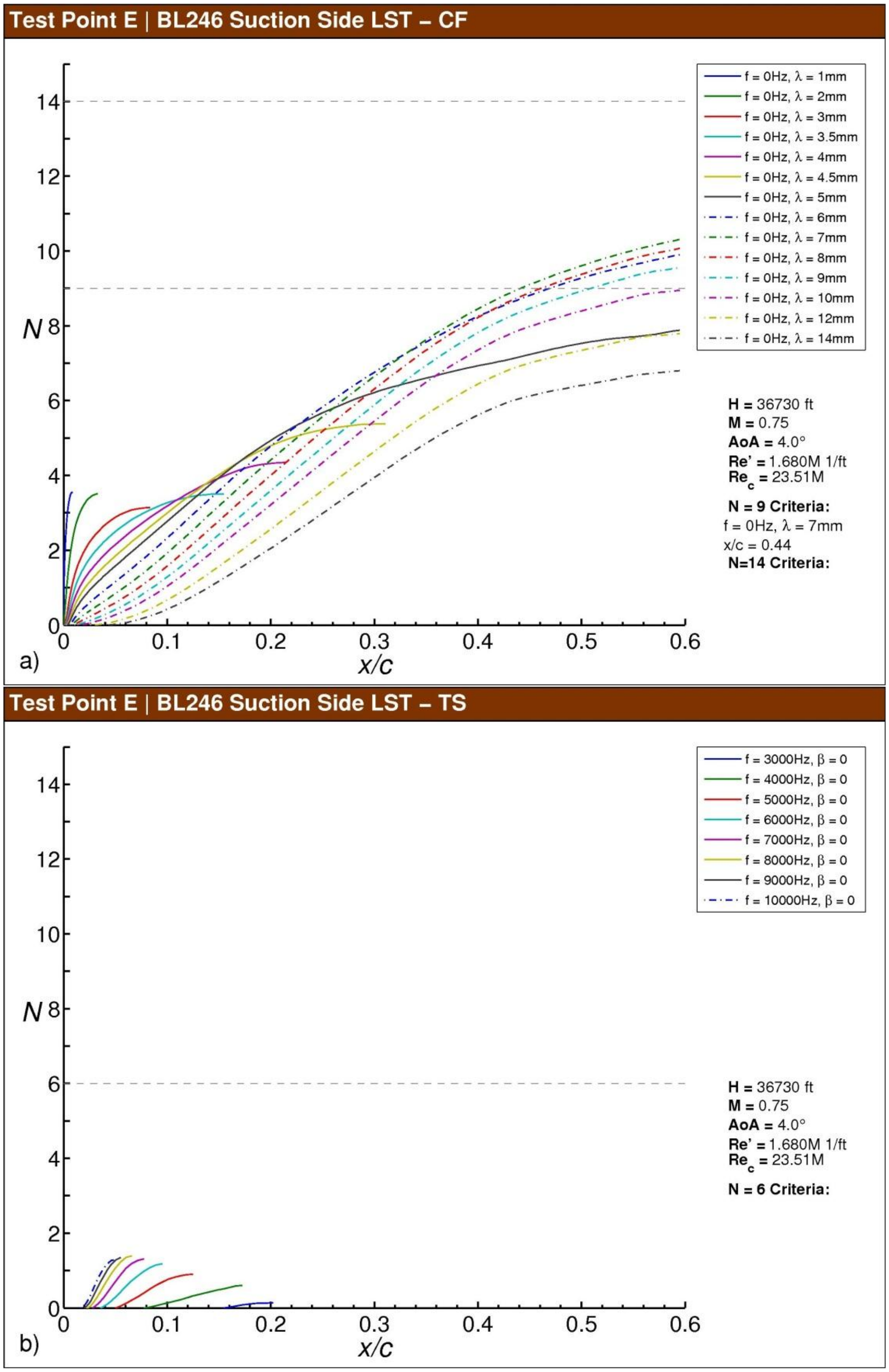


Fig. F-61 Test Point E *N*-factor results for BL246, suction side: a) stationary crossflow, b) T-S.

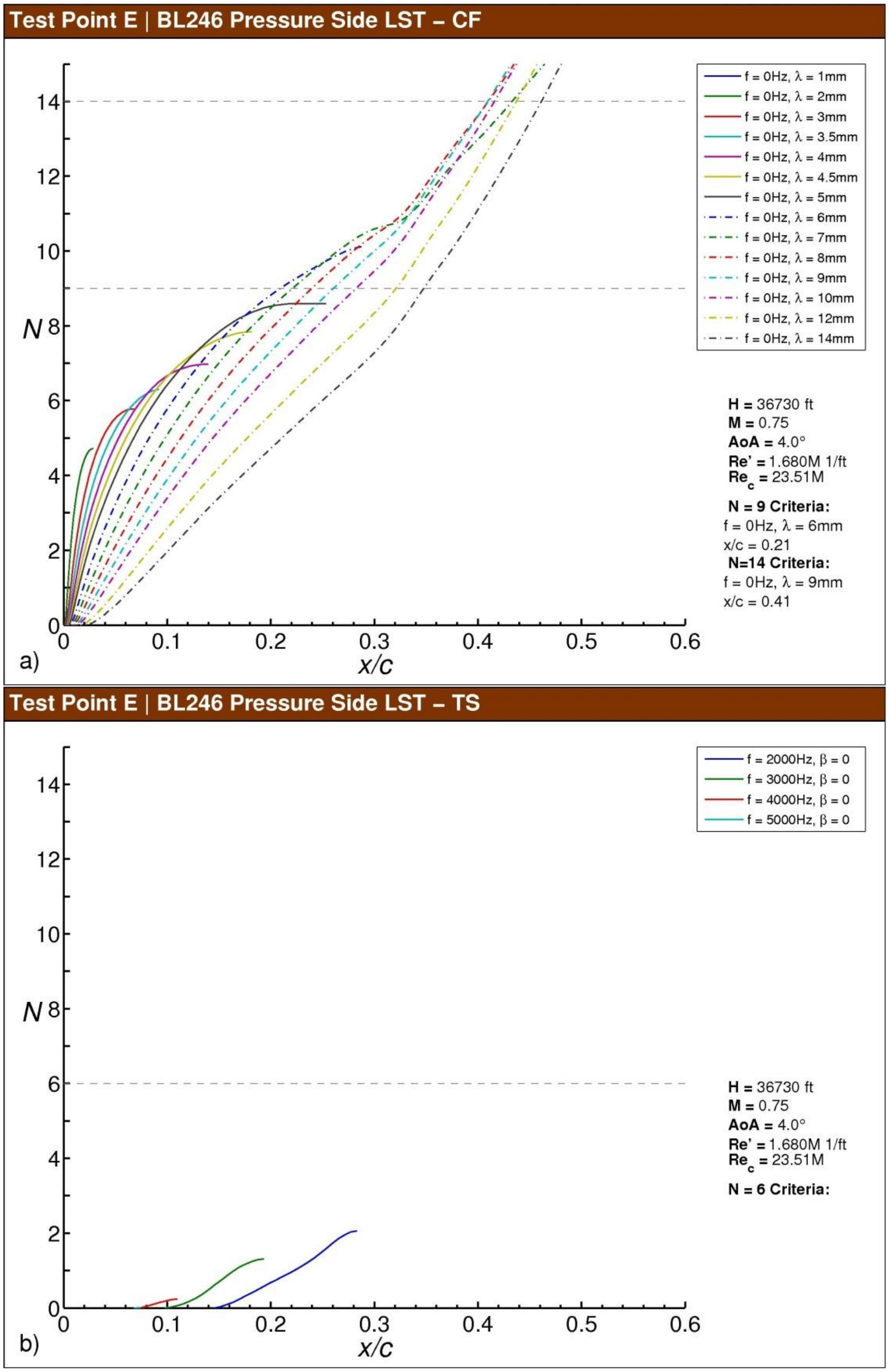
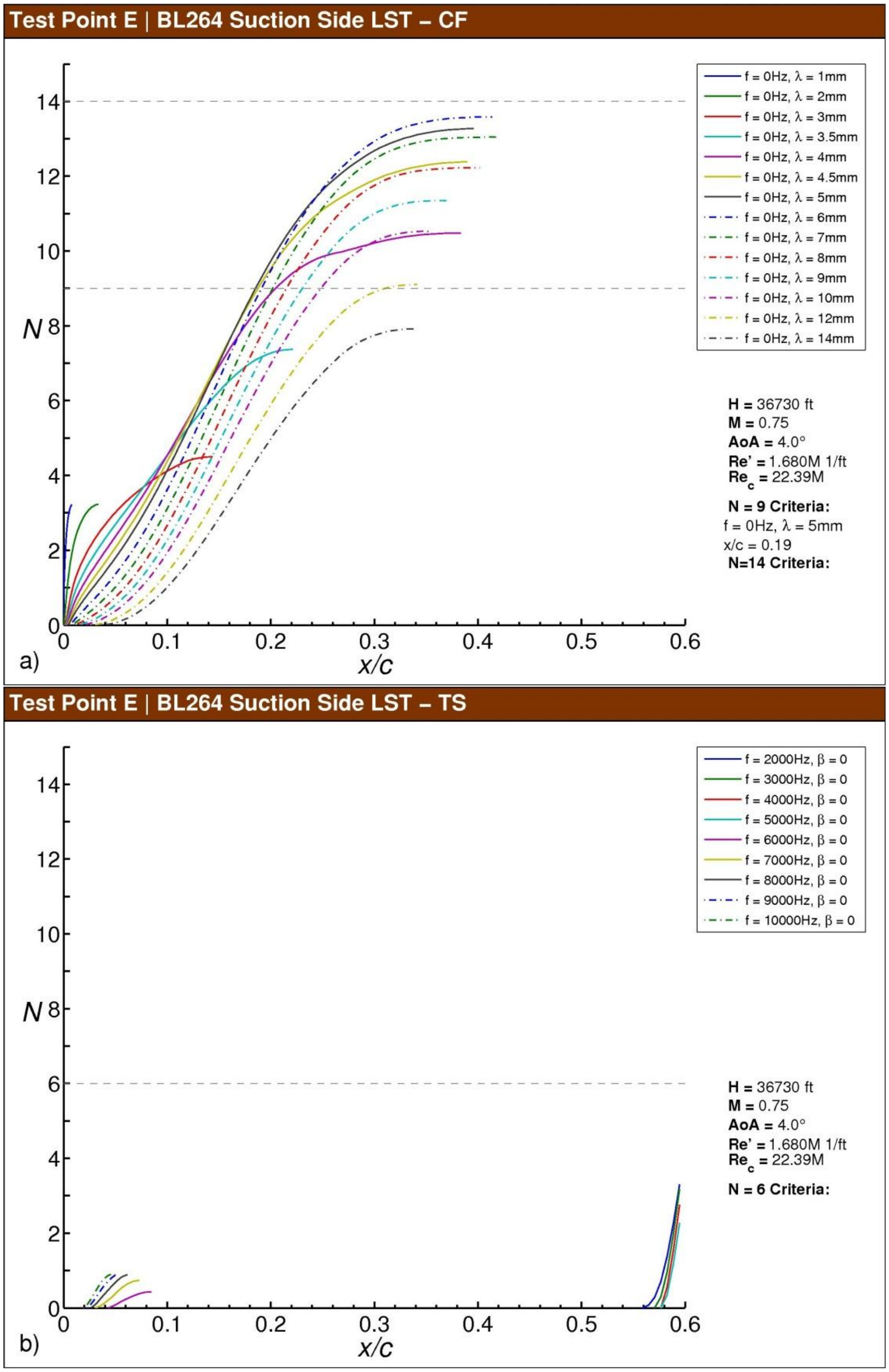


Fig. F-62 Test Point E *N*-factor results for BL246, pressure side: a) stationary crossflow, b) T-S.


Test Point E | BL264 Suction Side LST – CF

Test Point E | BL264 Suction Side LST – TS

Fig. F-63 Test Point E *N*-factor results for BL264, suction side: a) stationary crossflow, b) T-S.

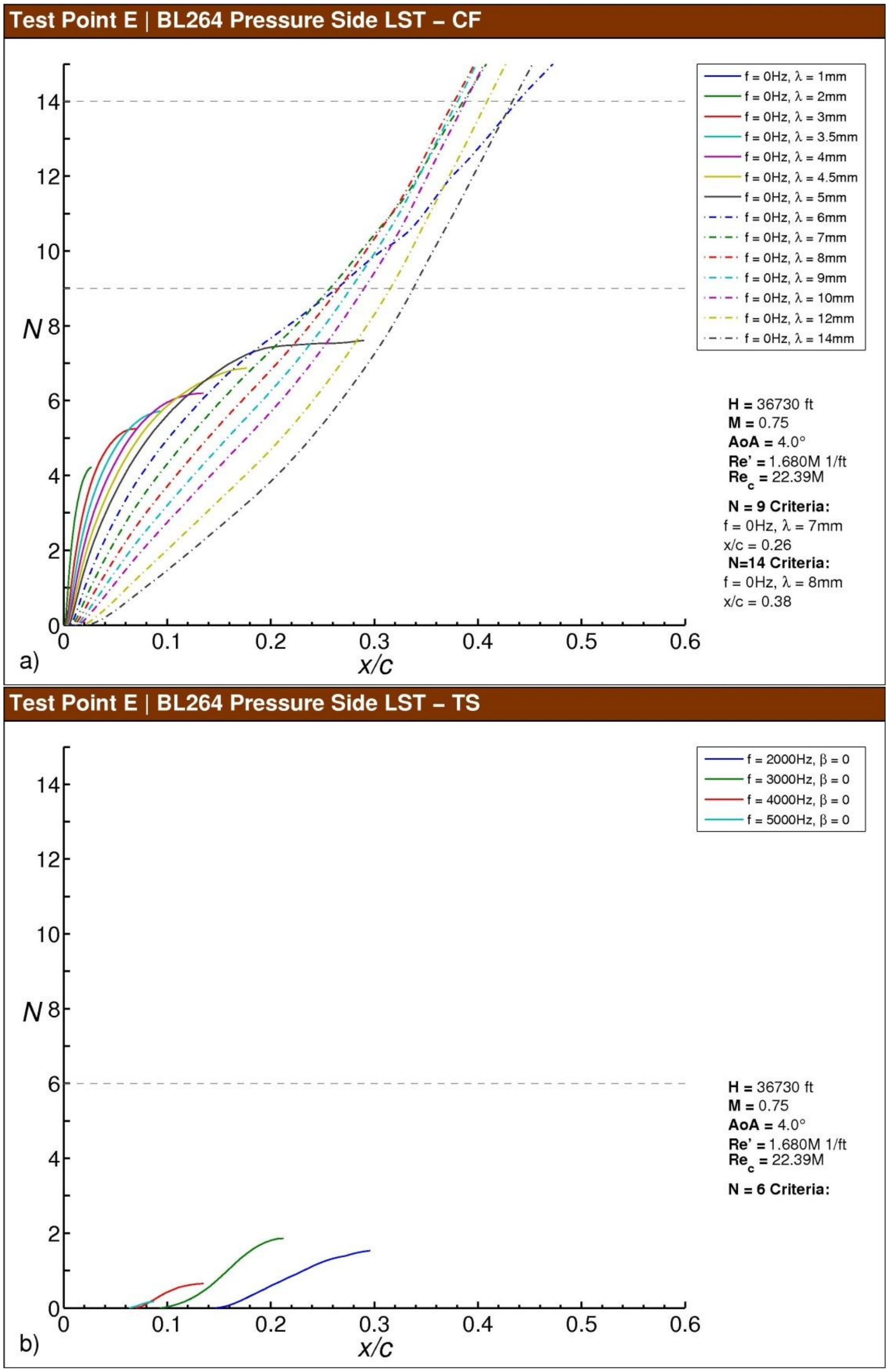


Fig. F-64 Test Point E *N*-factor results for BL264, pressure side: a) stationary crossflow, b) T-S.

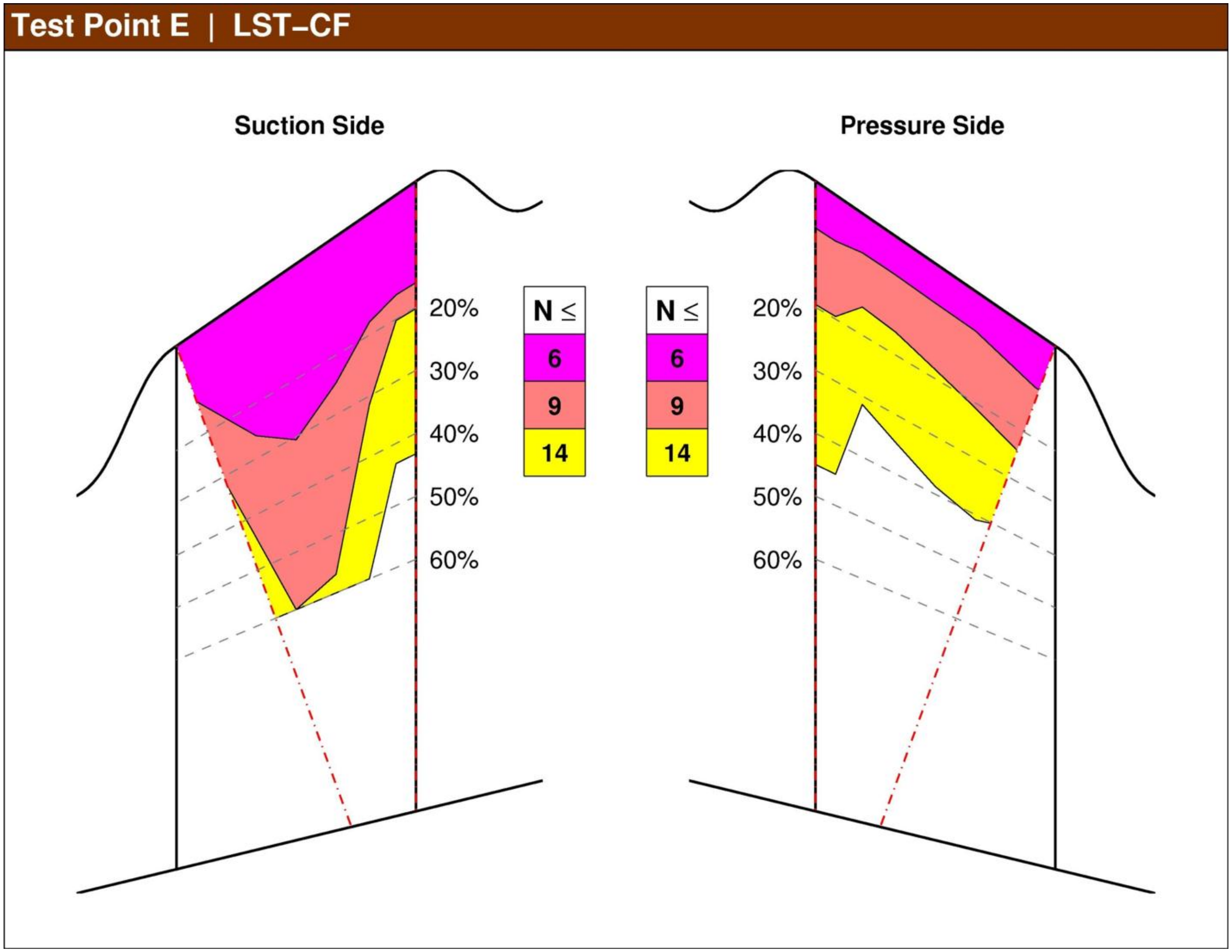


Fig. F-65 Test Point E *N*-factor contours for the test section suction and pressure sides.

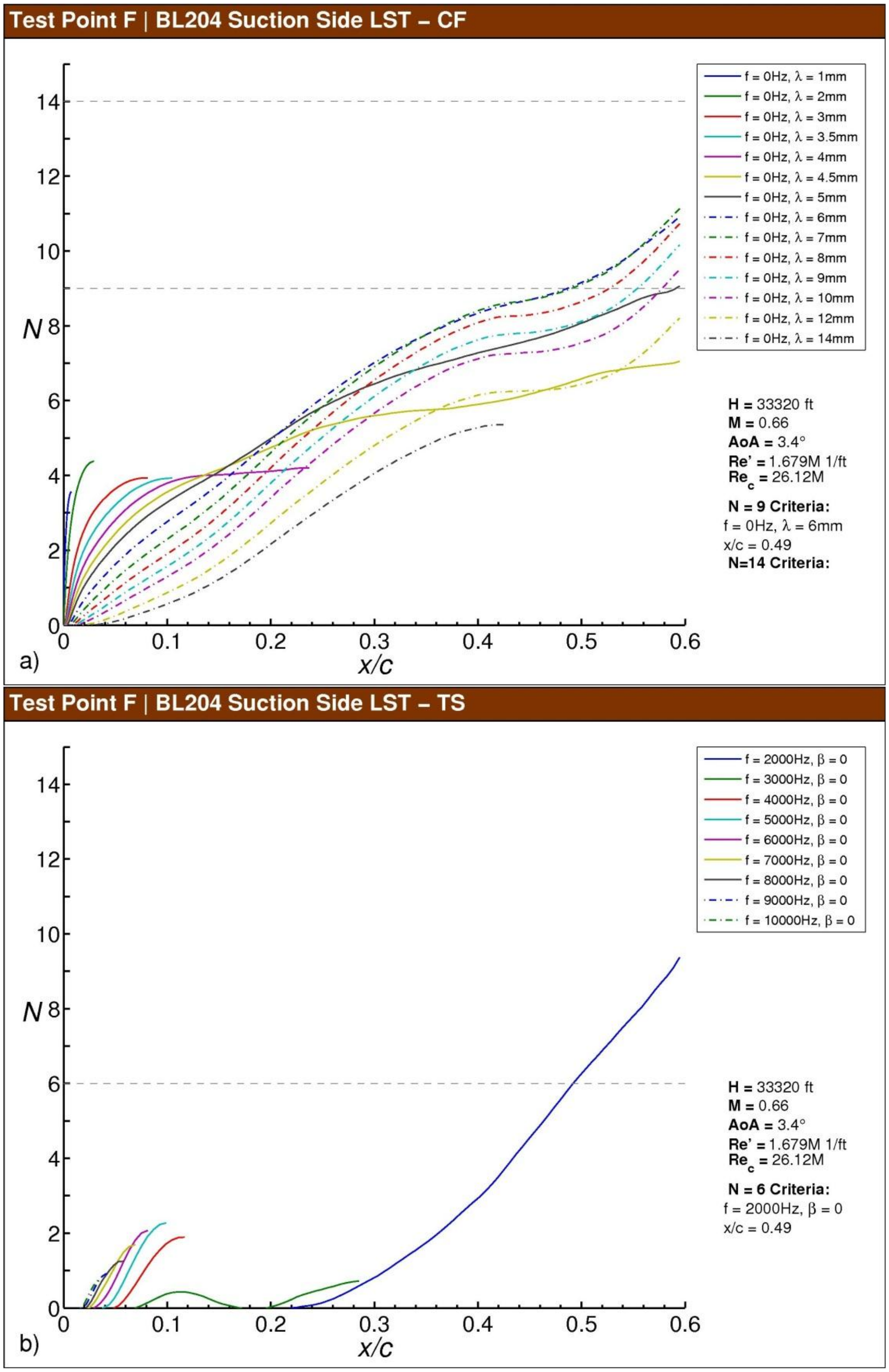


Fig. F-66 Test Point F *N*-factor results for BL204, suction side: a) stationary crossflow, b) T-S.

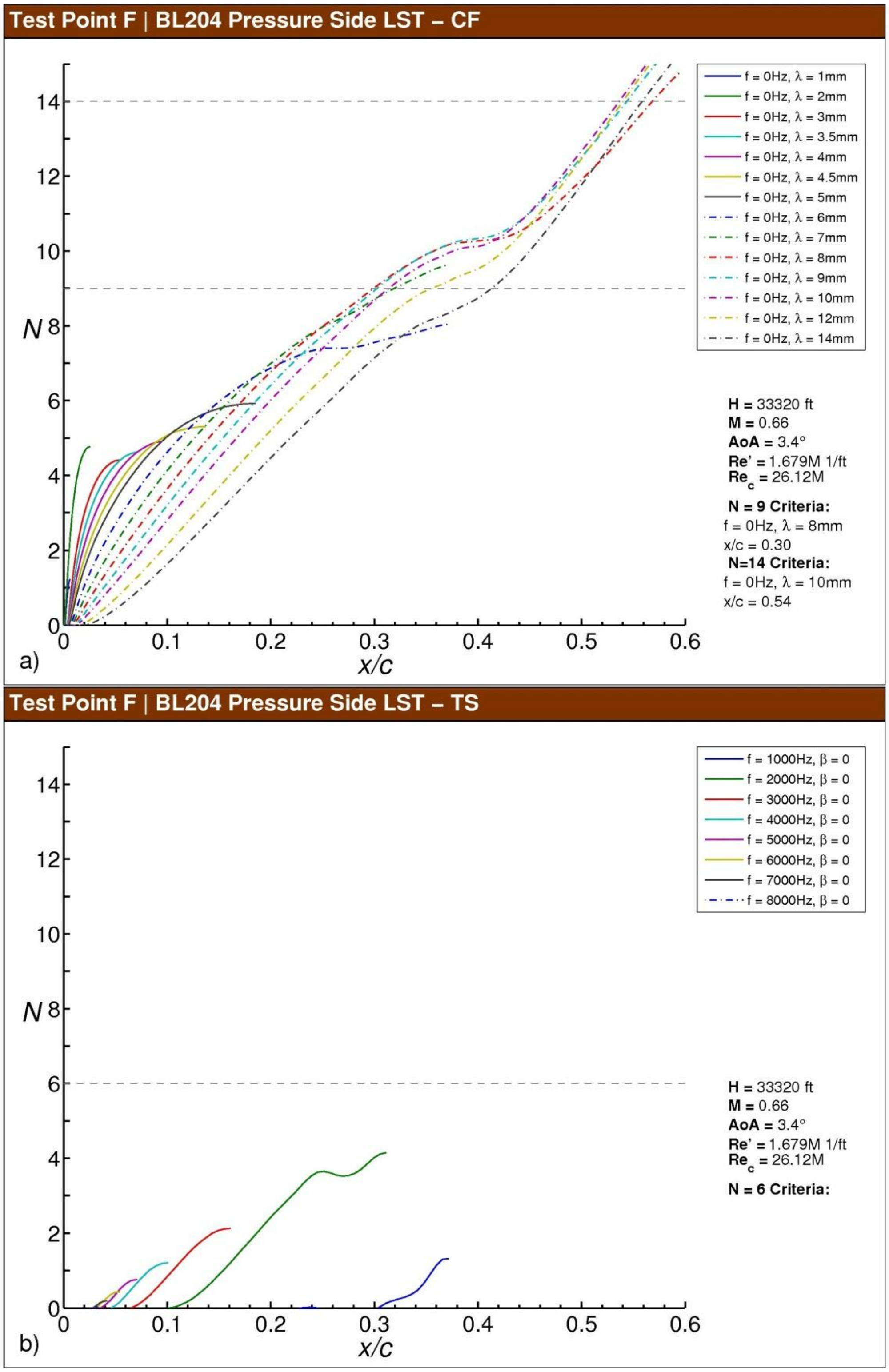


Fig. F-67 Test Point F *N*-factor results for BL204, pressure side: a) stationary crossflow, b) T-S.

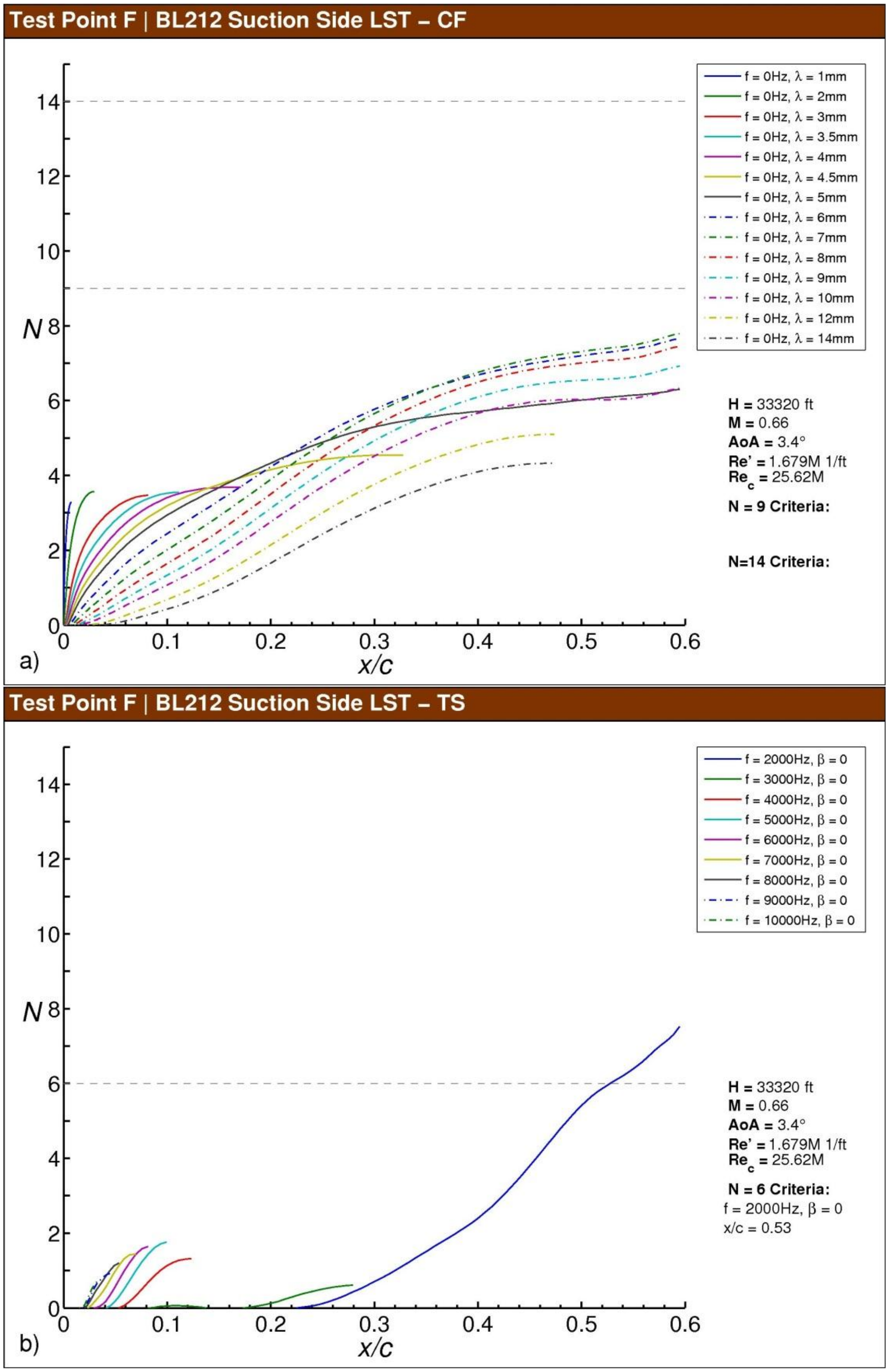


Fig. F-68 Test Point F *N*-factor results for BL212, suction side: a) stationary crossflow, b) T-S.

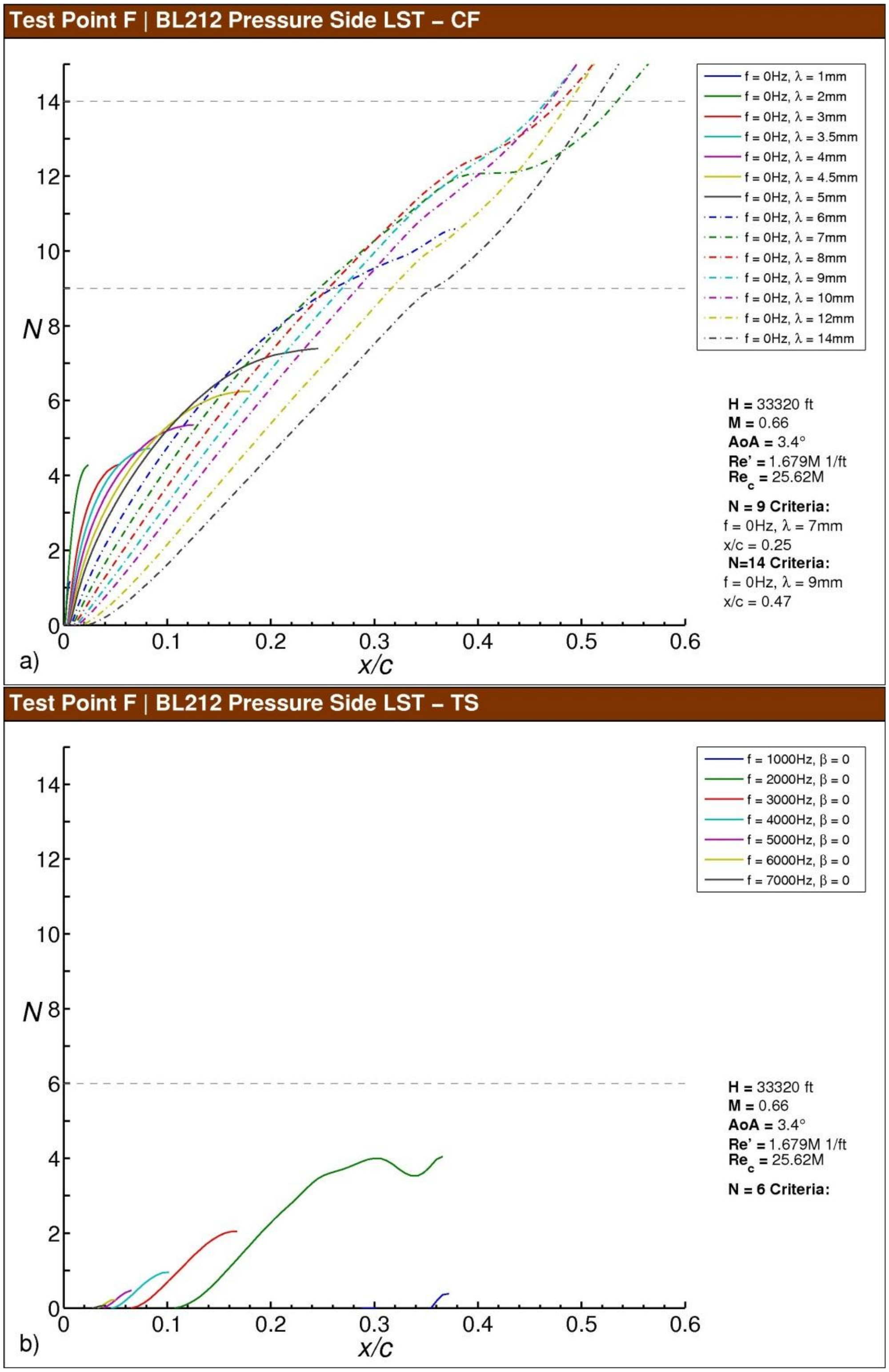


Fig. F-69 Test Point F *N*-factor results for BL212, pressure side: a) stationary crossflow, b) T-S.

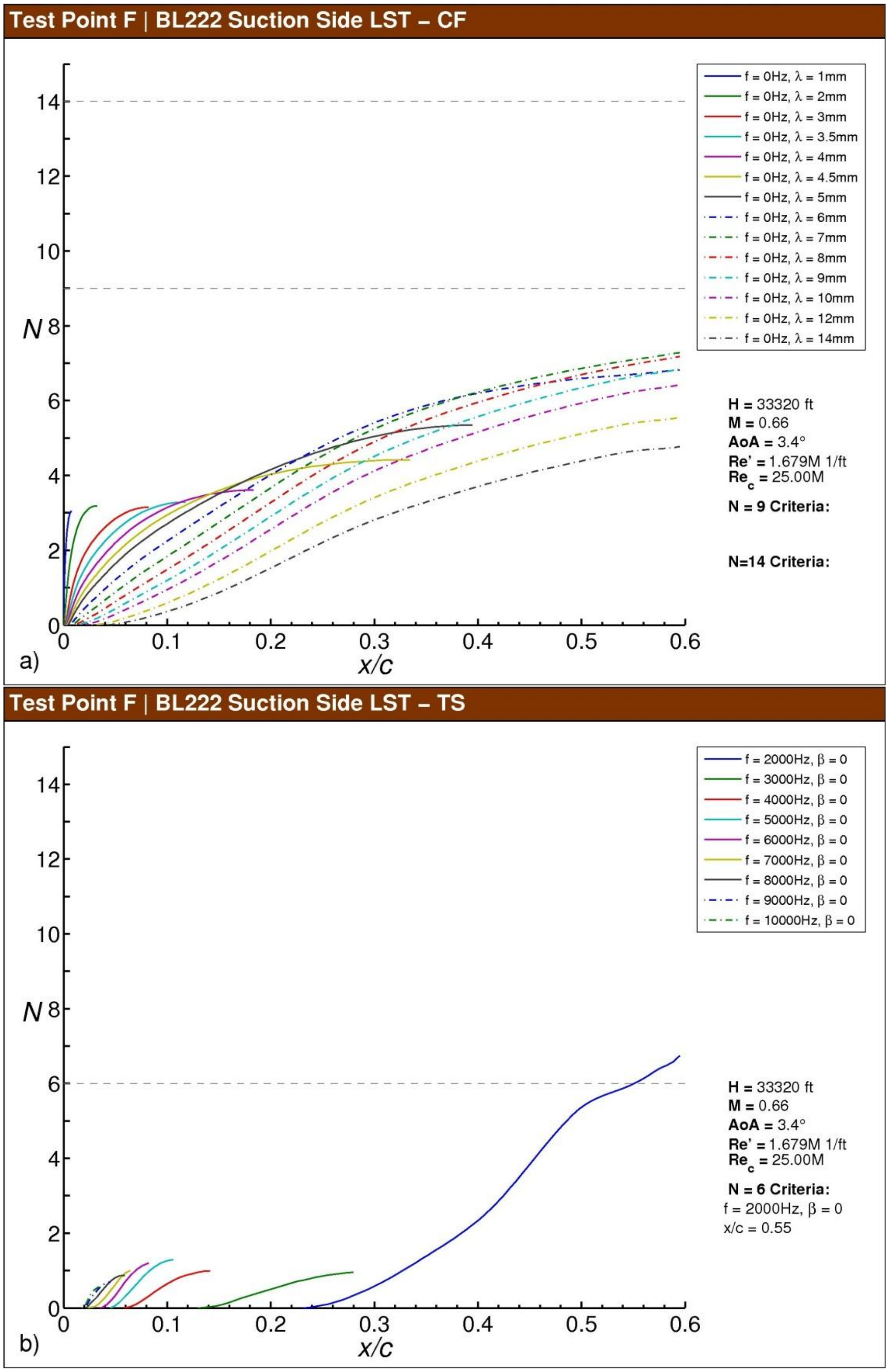
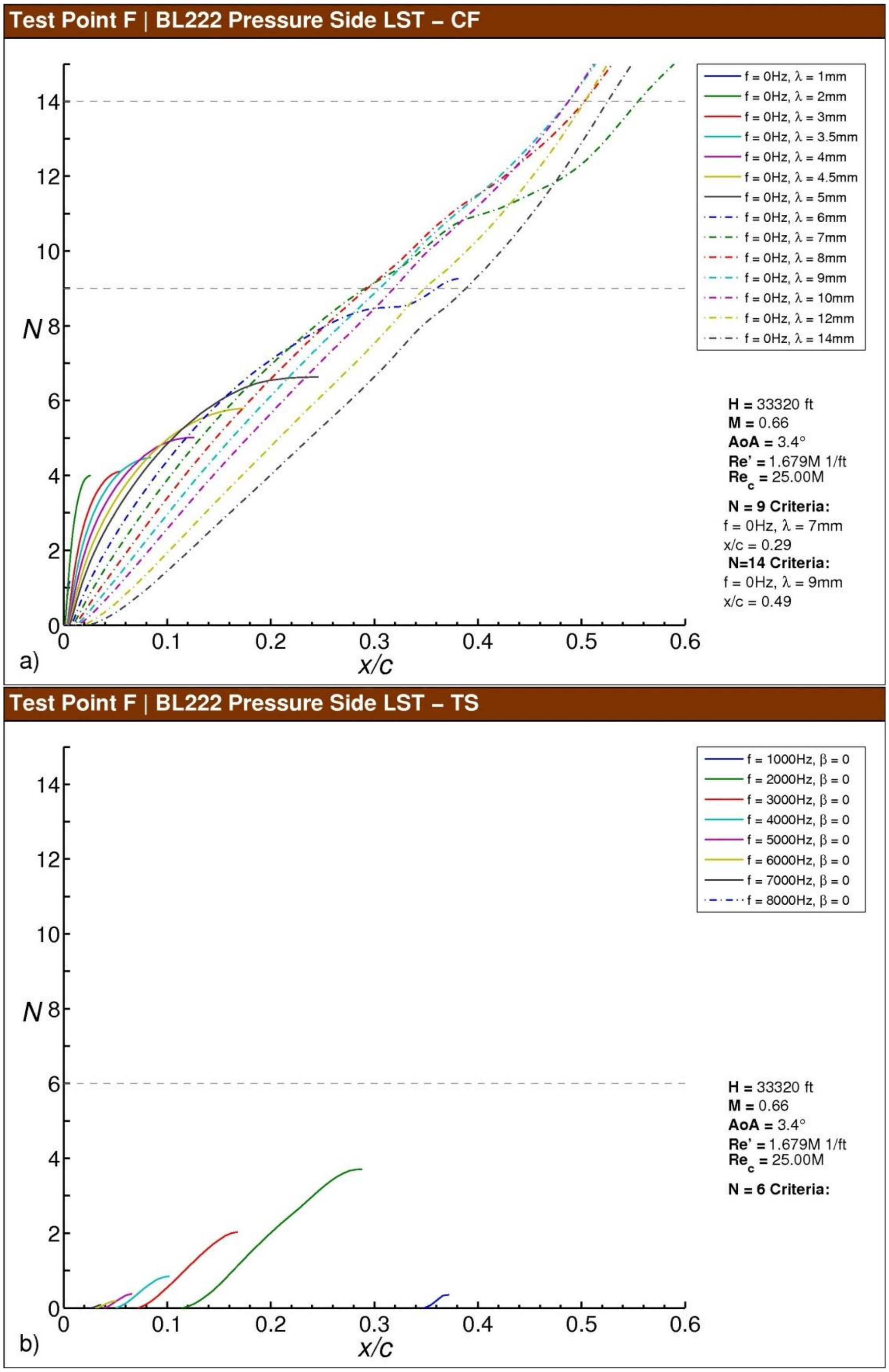


Fig. F-70 Test Point F *N*-factor results for BL222, suction side: a) stationary crossflow, b) T-S.



Test Point F | BL222 Pressure Side LST – CF

Test Point F | BL222 Pressure Side LST – TS

Fig. F-71 Test Point F *N*-factor results for BL222, pressure side: a) stationary crossflow, b) T-S.

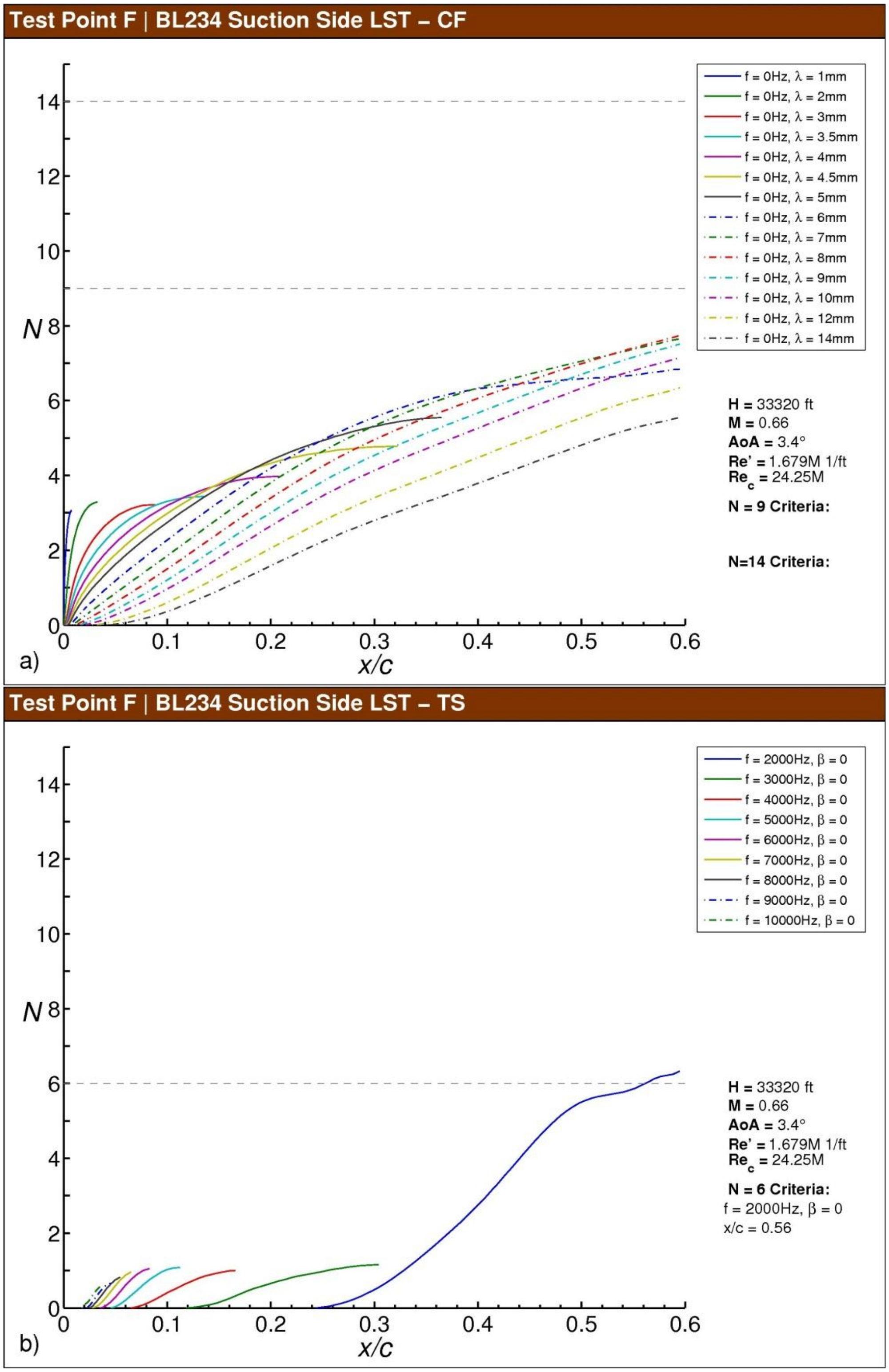
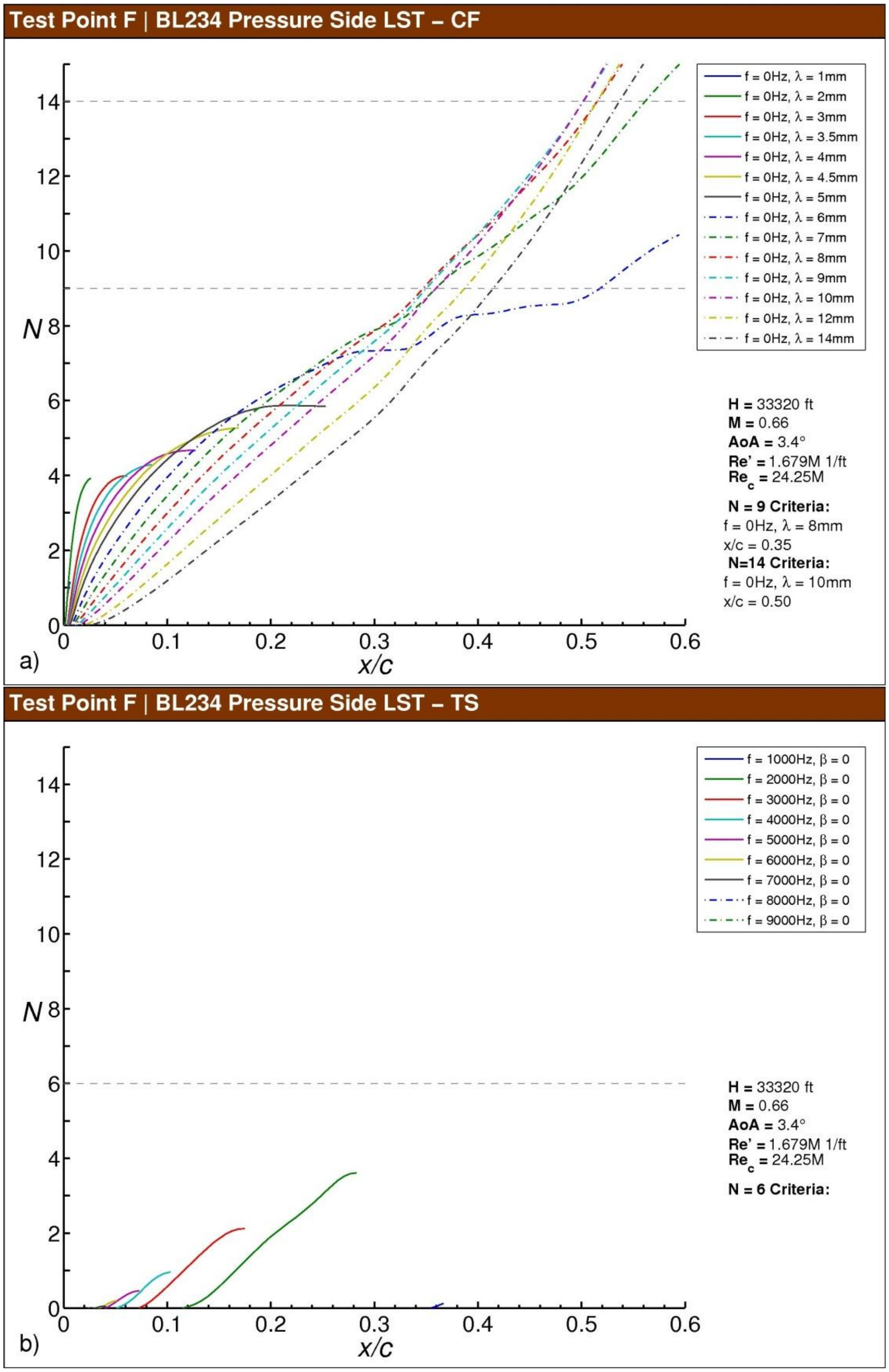


Fig. F-72 Test Point F *N*-factor results for BL234, suction side: a) stationary crossflow, b) T-S.


Test Point F | BL234 Pressure Side LST – CF

Test Point F | BL234 Pressure Side LST – TS

Fig. F-73 Test Point F *N*-factor results for BL234, pressure side: a) stationary crossflow, b) T-S.

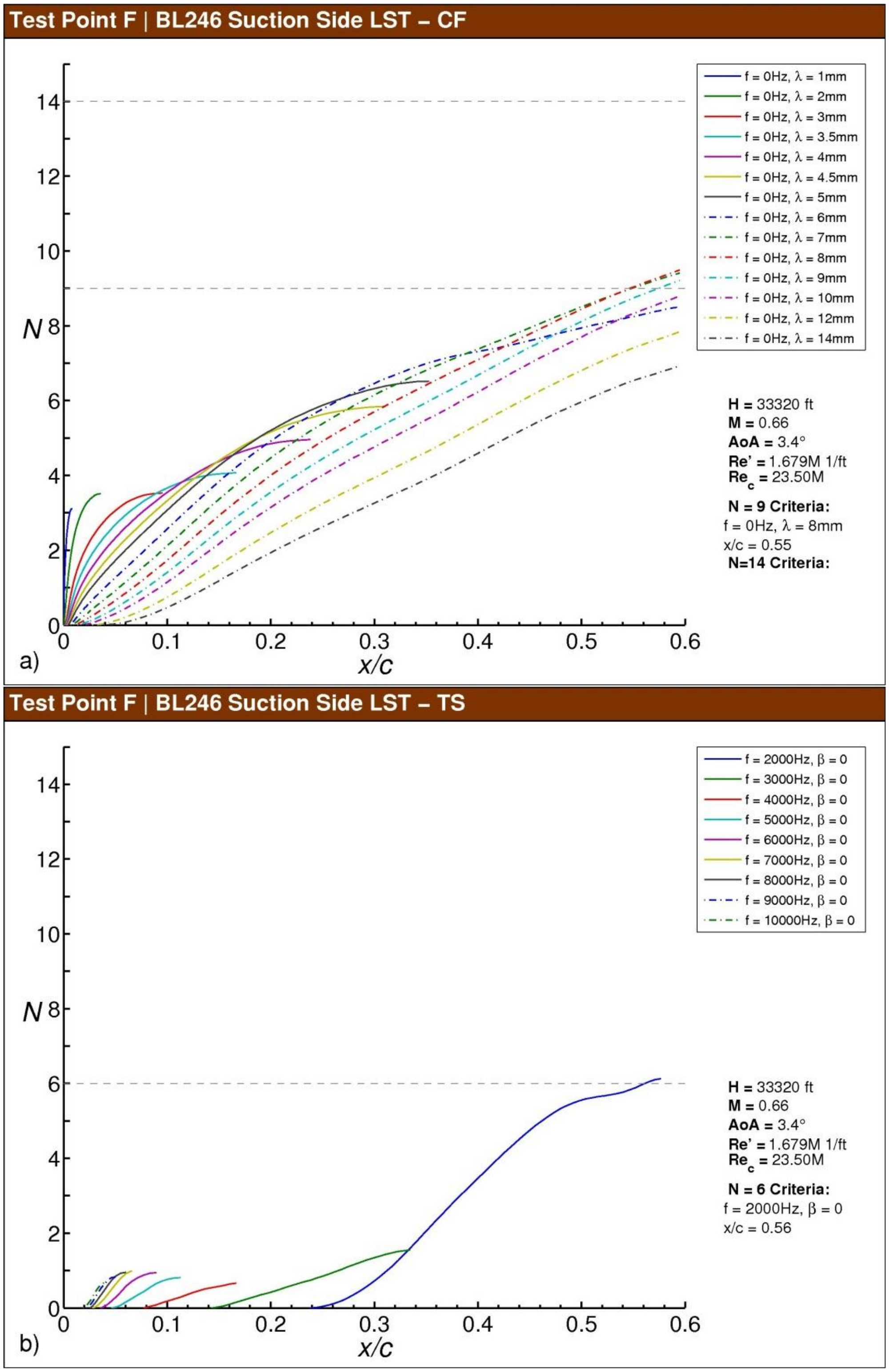


Fig. F-74 Test Point F *N*-factor results for BL246, suction side: a) stationary crossflow, b) T-S.

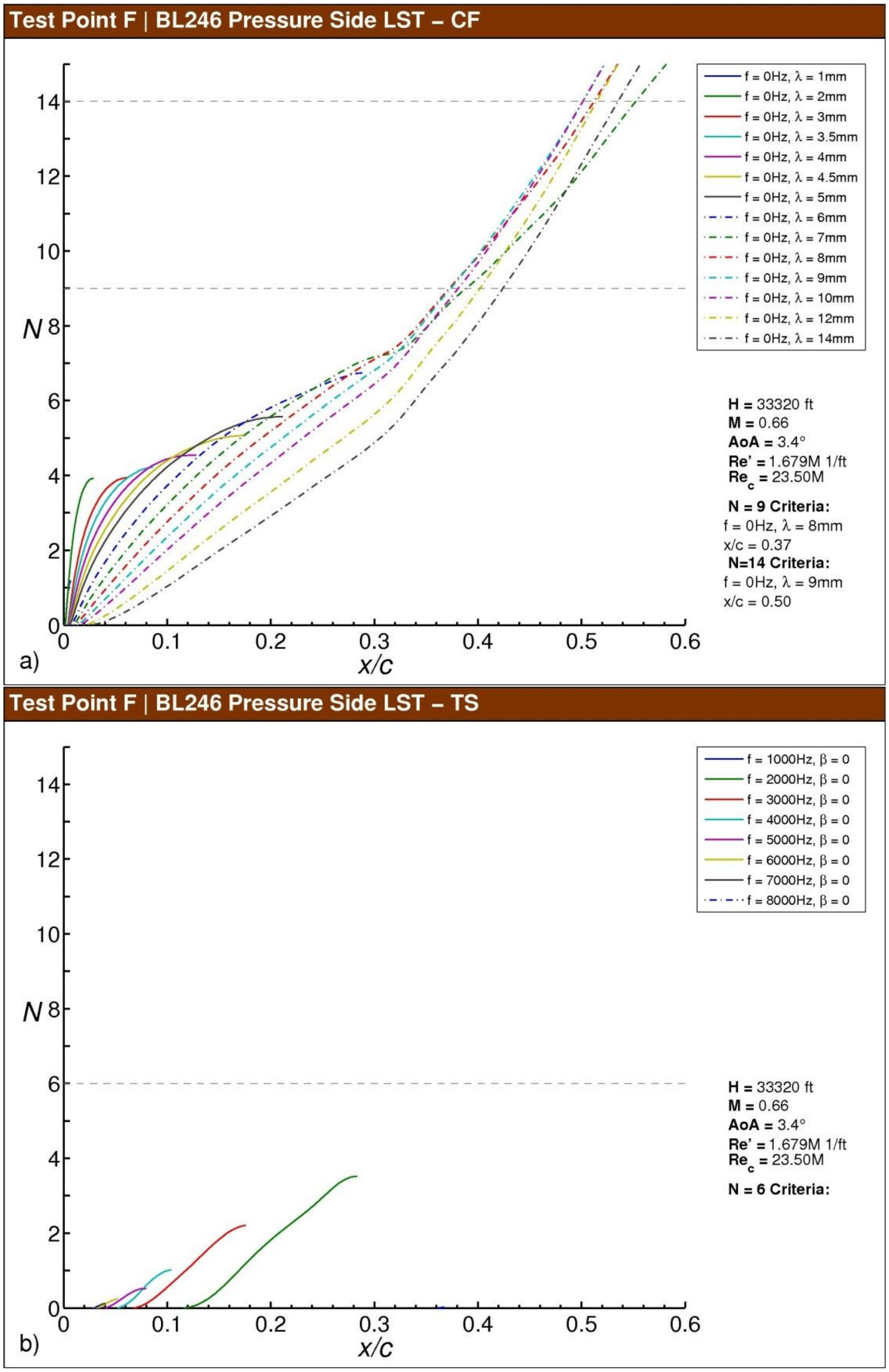


Fig. F-75 Test Point F *N*-factor results for BL246, pressure side: a) stationary crossflow, b) T-S.

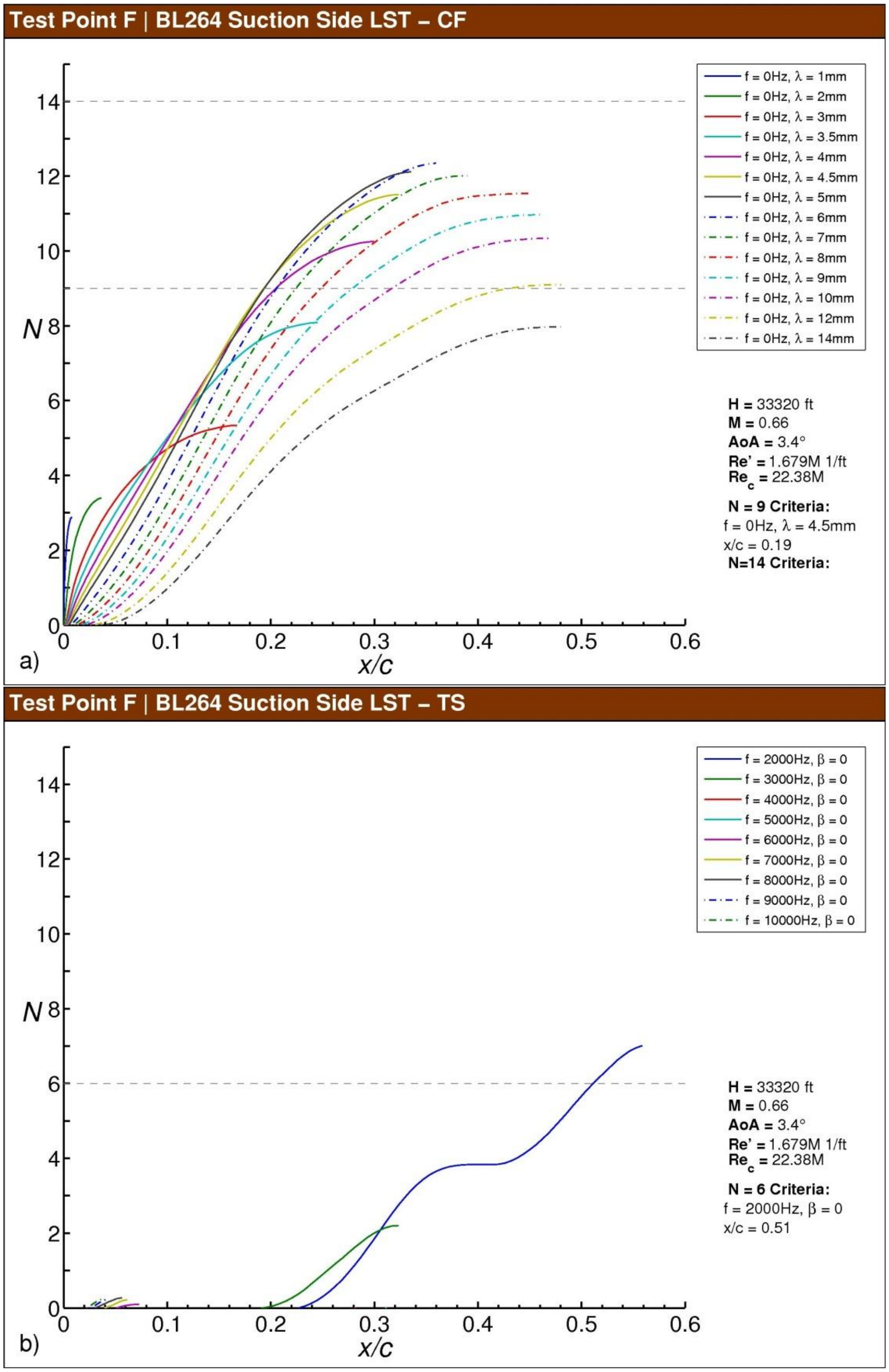


Fig. F-76 Test Point F *N*-factor results for BL264, suction side: a) stationary crossflow, b) T-S.

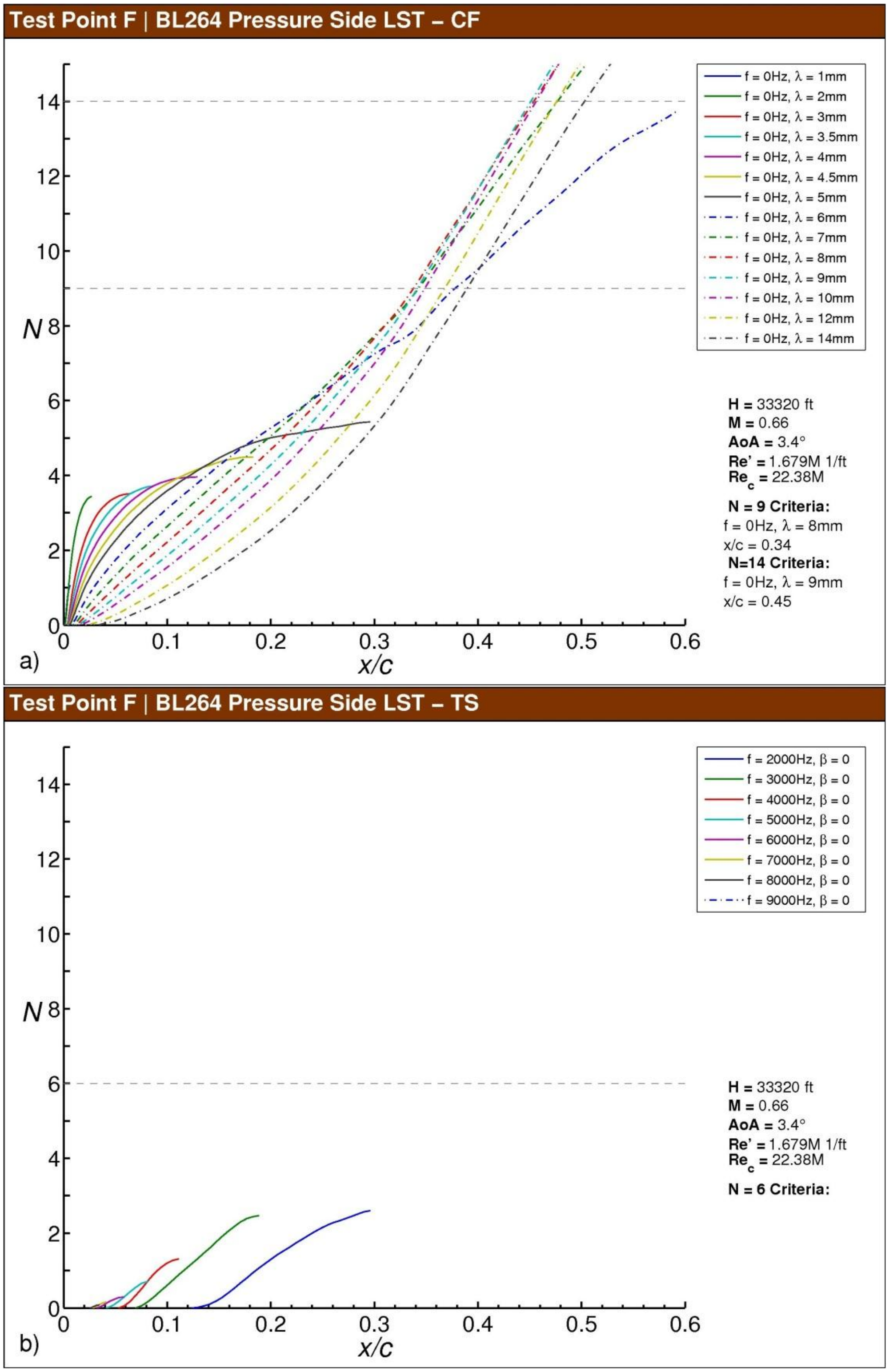
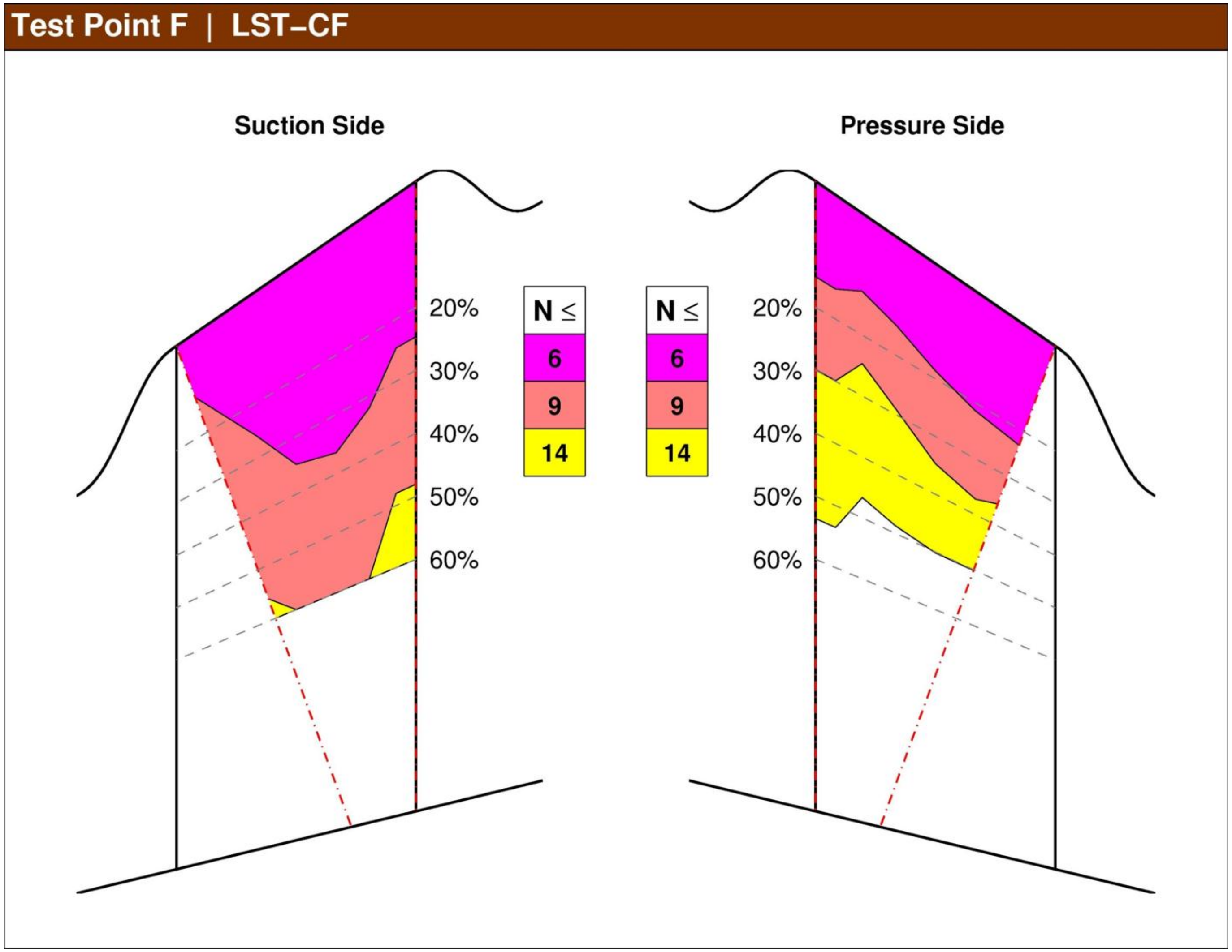


Fig. F-77 Test Point F *N*-factor results for BL264, pressure side: a) stationary crossflow, b) T-S.



Test Point F | LST–CF

Fig. F-78 Test Point F *N*-factor contours for the test section suction and pressure sides.

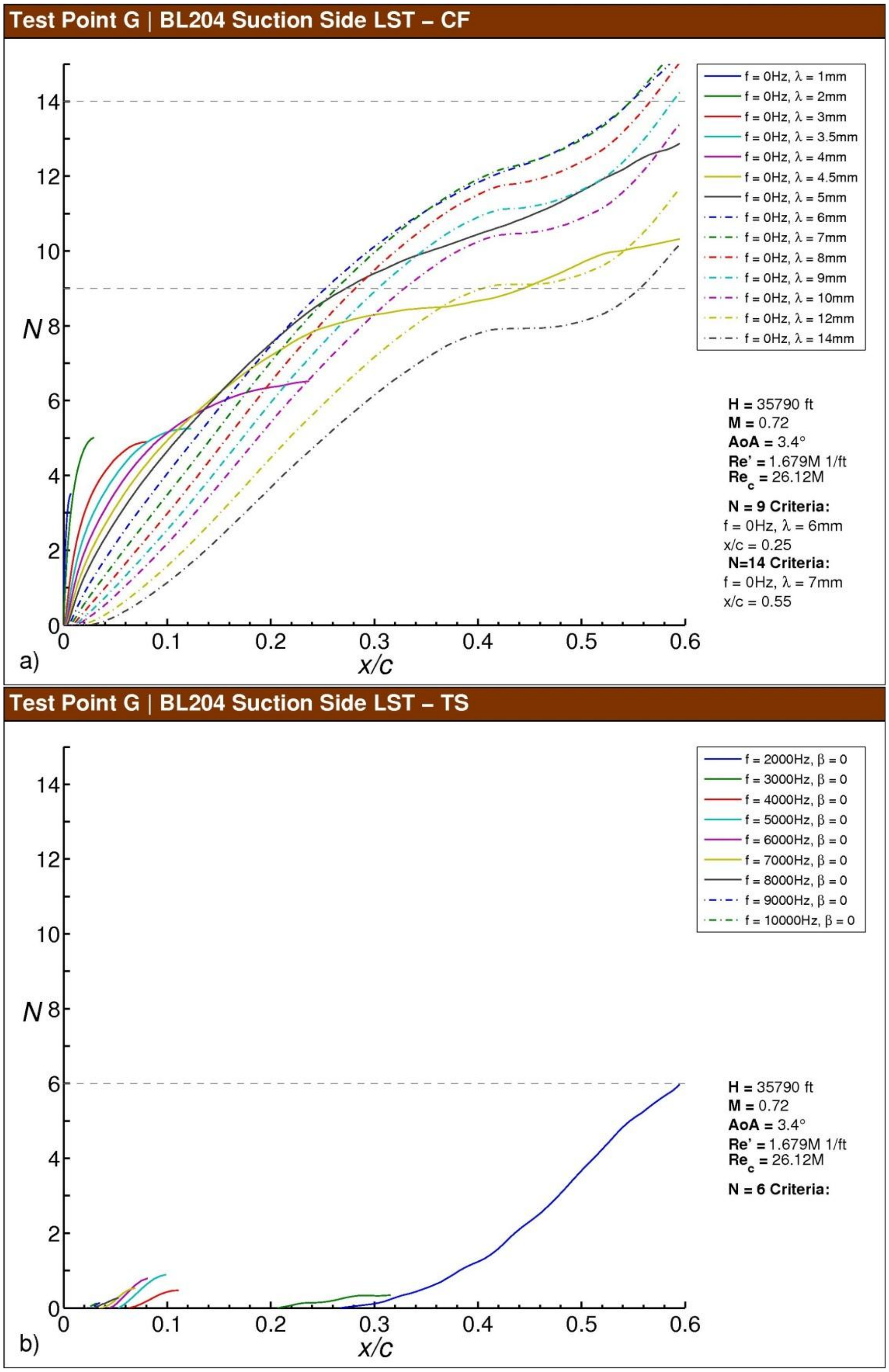
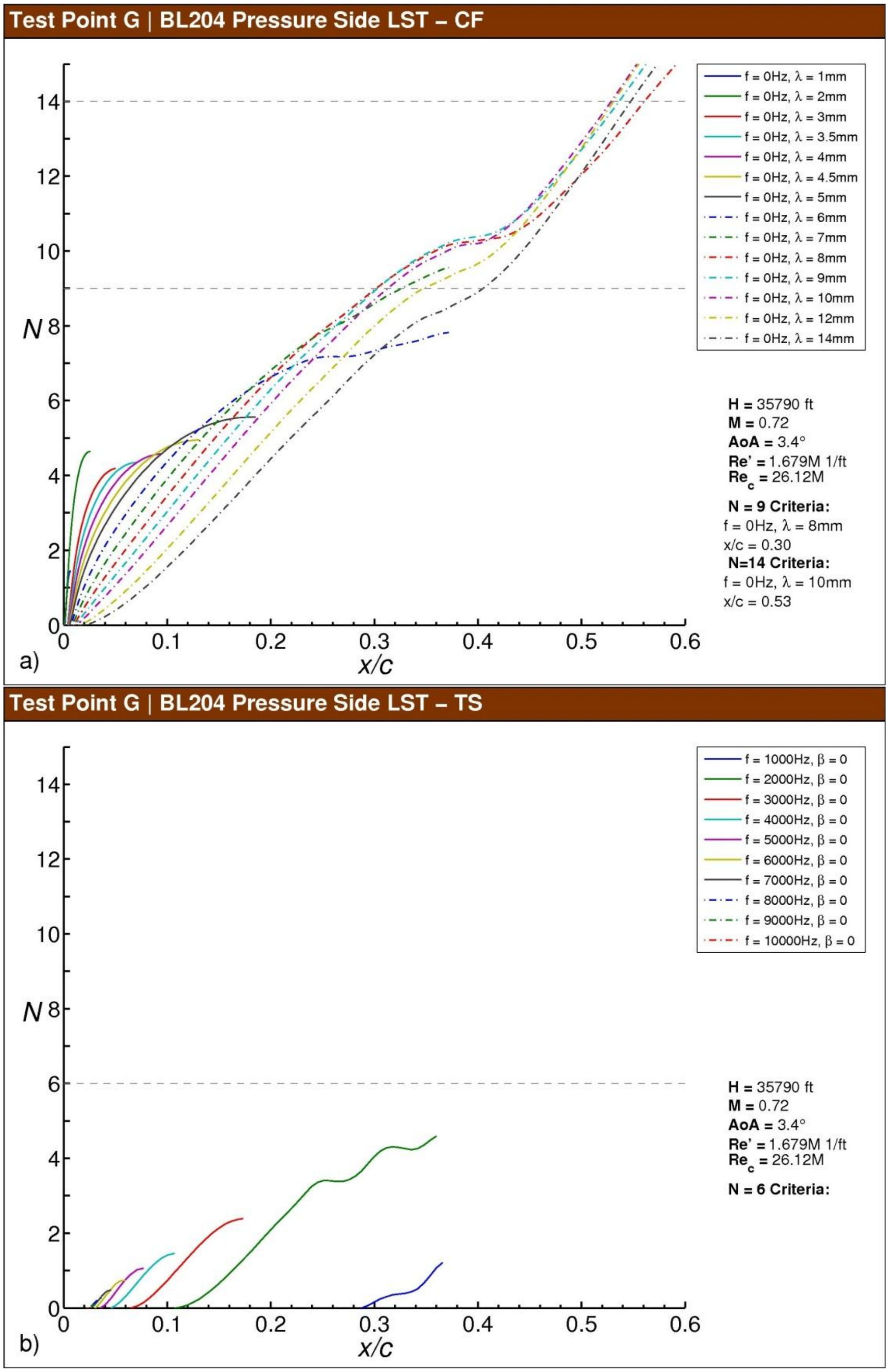


Fig. F-79 Test Point G *N*-factor results for BL204, suction side: a) stationary crossflow, b) T-S.



Test Point G | BL204 Pressure Side LST – CF

Test Point G | BL204 Pressure Side LST – TS

Fig. F-80 Test Point G *N*-factor results for BL204, pressure side: a) stationary crossflow, b) T-S.

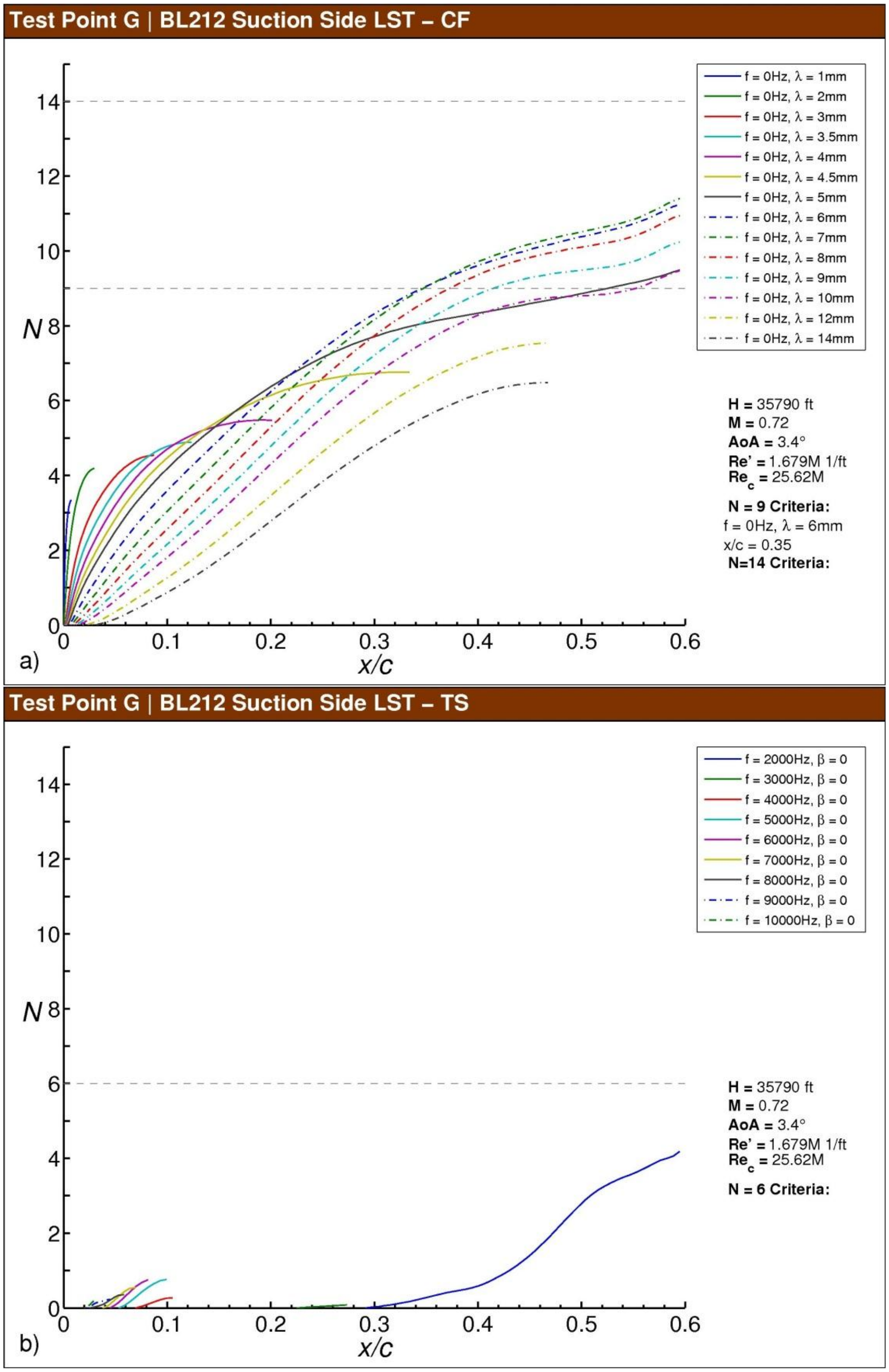


Fig. F-81 Test Point G *N*-factor results for BL212, suction side: a) stationary crossflow, b) T-S.

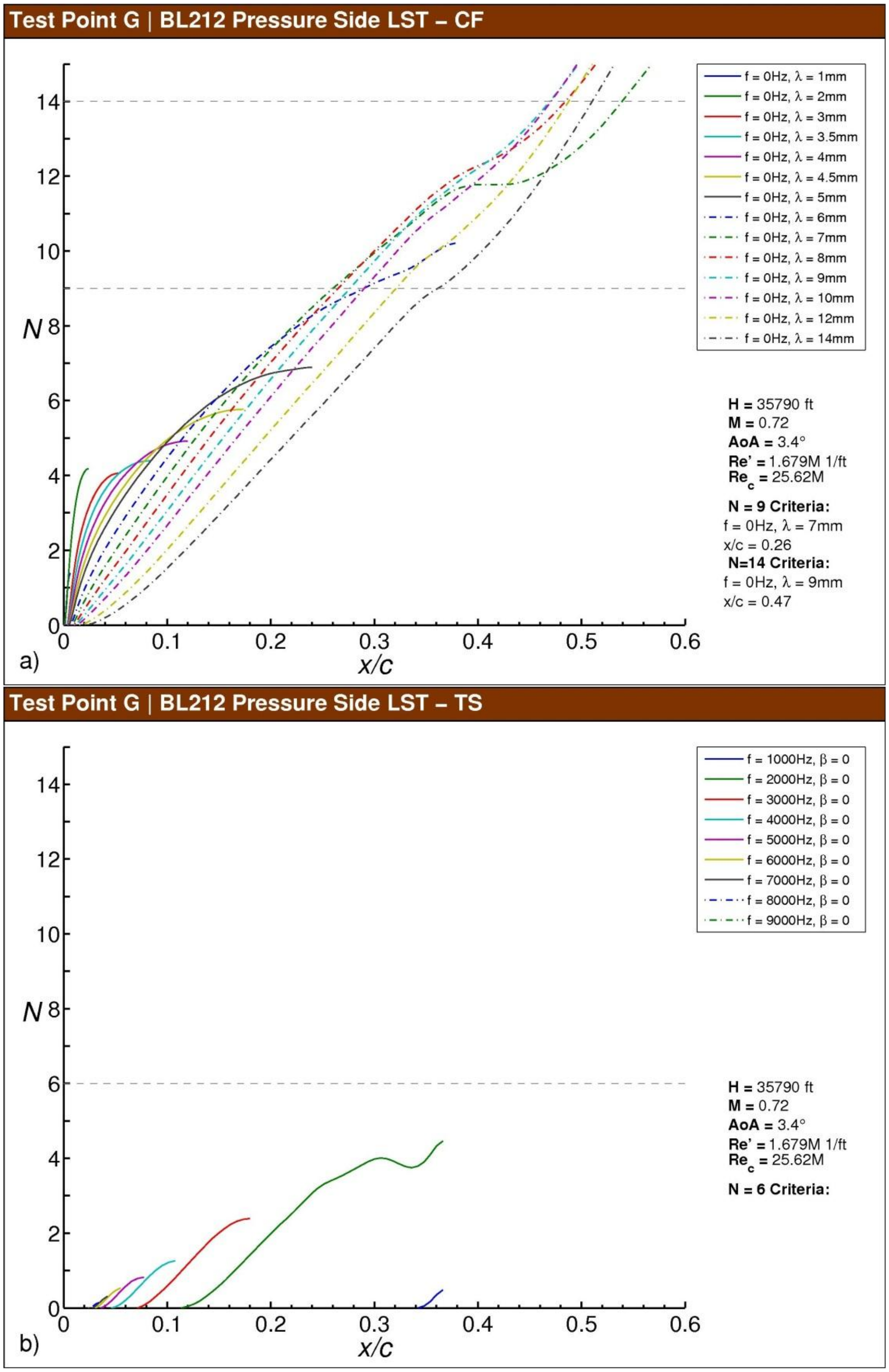


Fig. F-82 Test Point G *N*-factor results for BL212, pressure side: a) stationary crossflow, b) T-S.

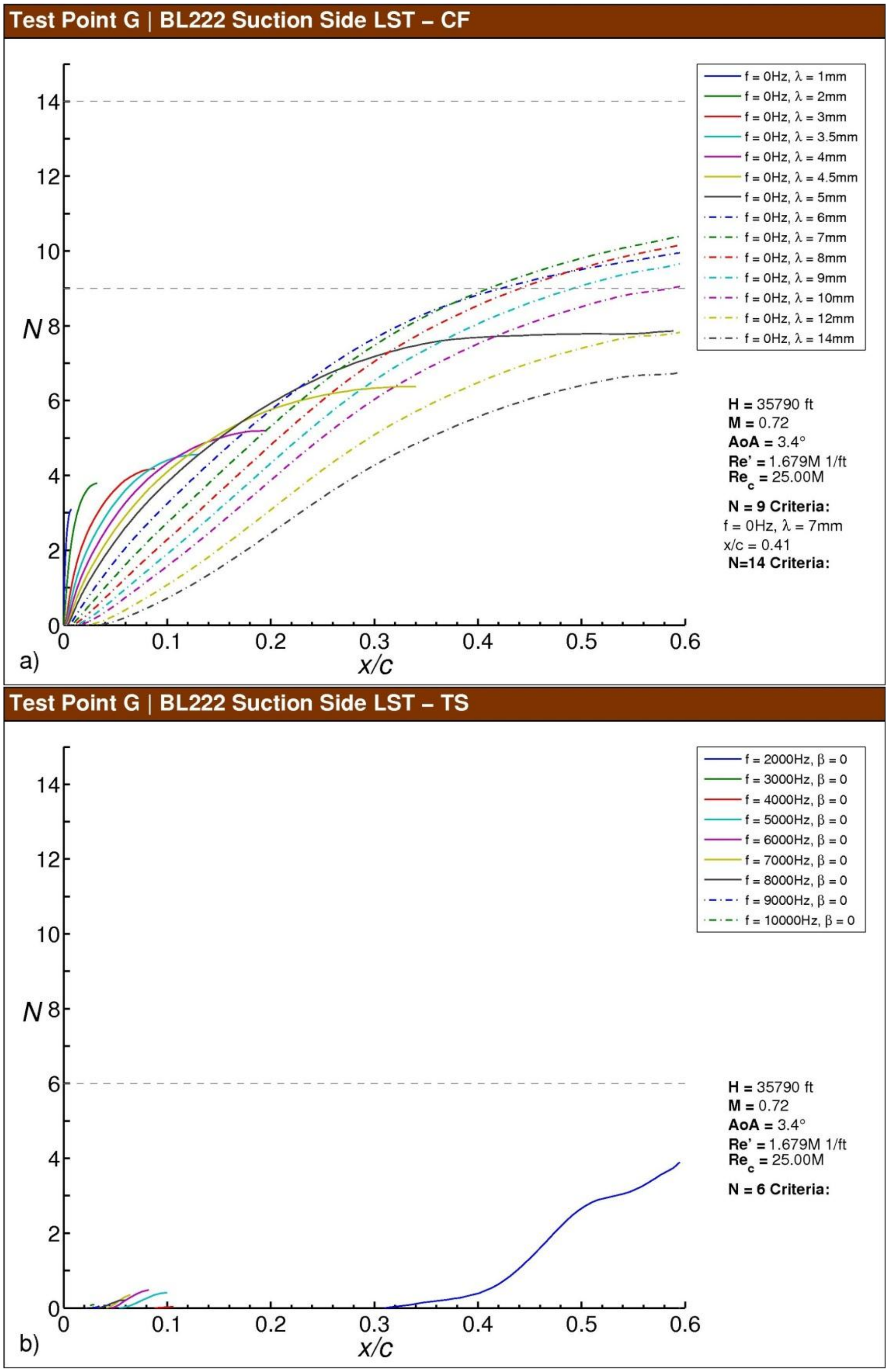


Fig. F-83 Test Point G *N*-factor results for BL222, suction side: a) stationary crossflow, b) T-S.

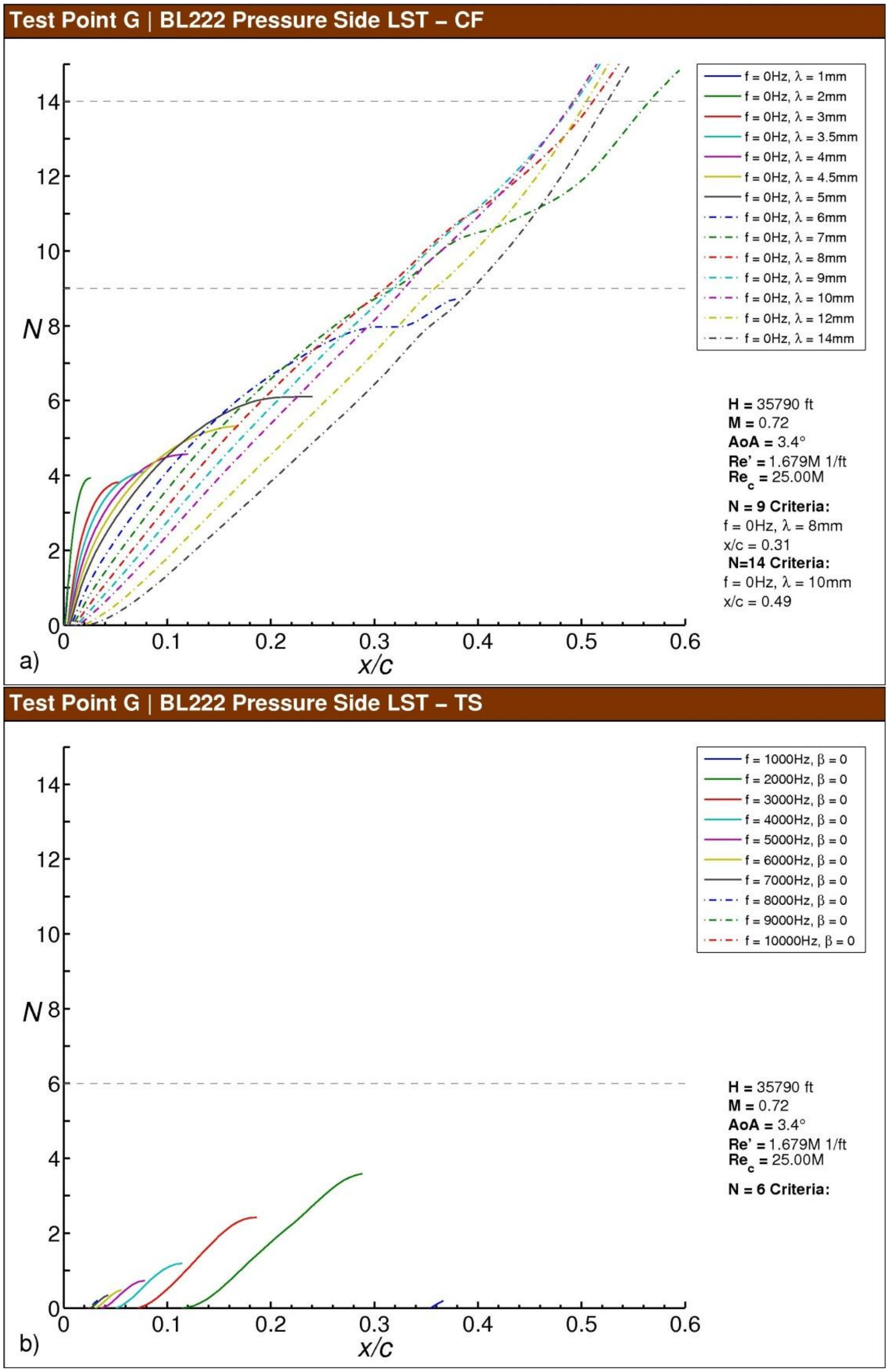


Fig. F-84 Test Point G *N*-factor results for BL222, pressure side: a) stationary crossflow, b) T-S.

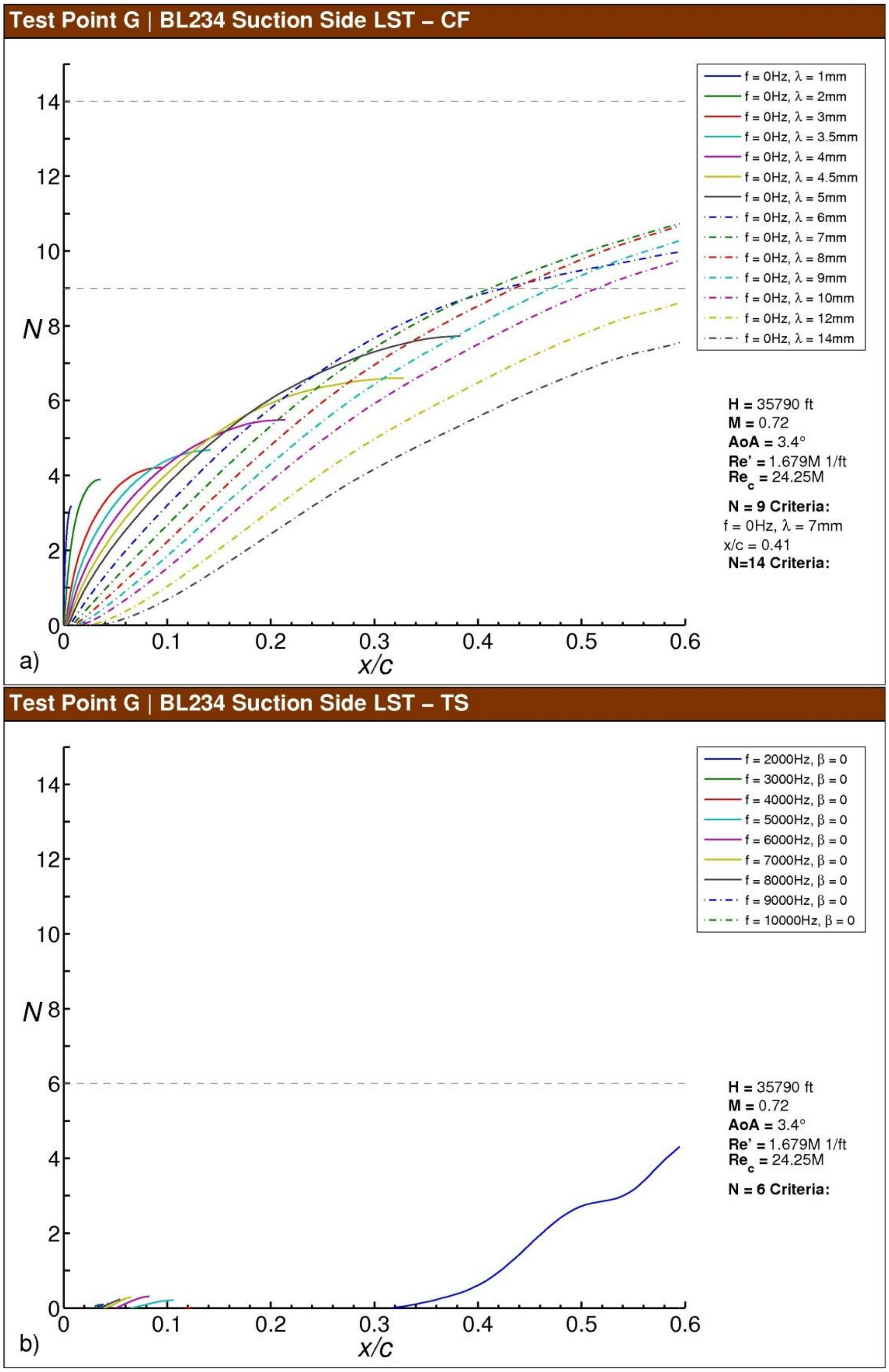


Fig. F-85 Test Point G *N*-factor results for BL234, suction side: a) stationary crossflow, b) T-S.

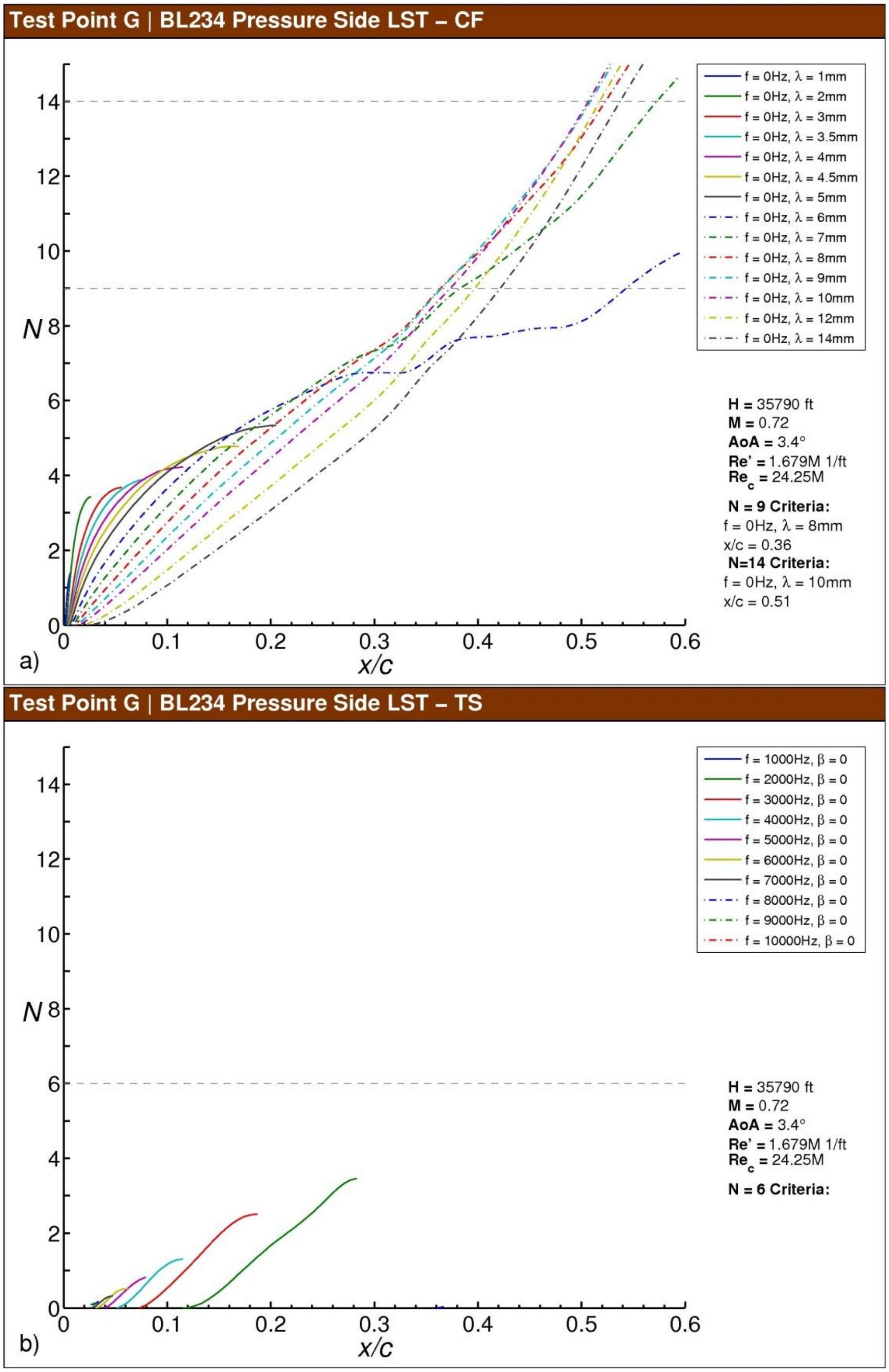
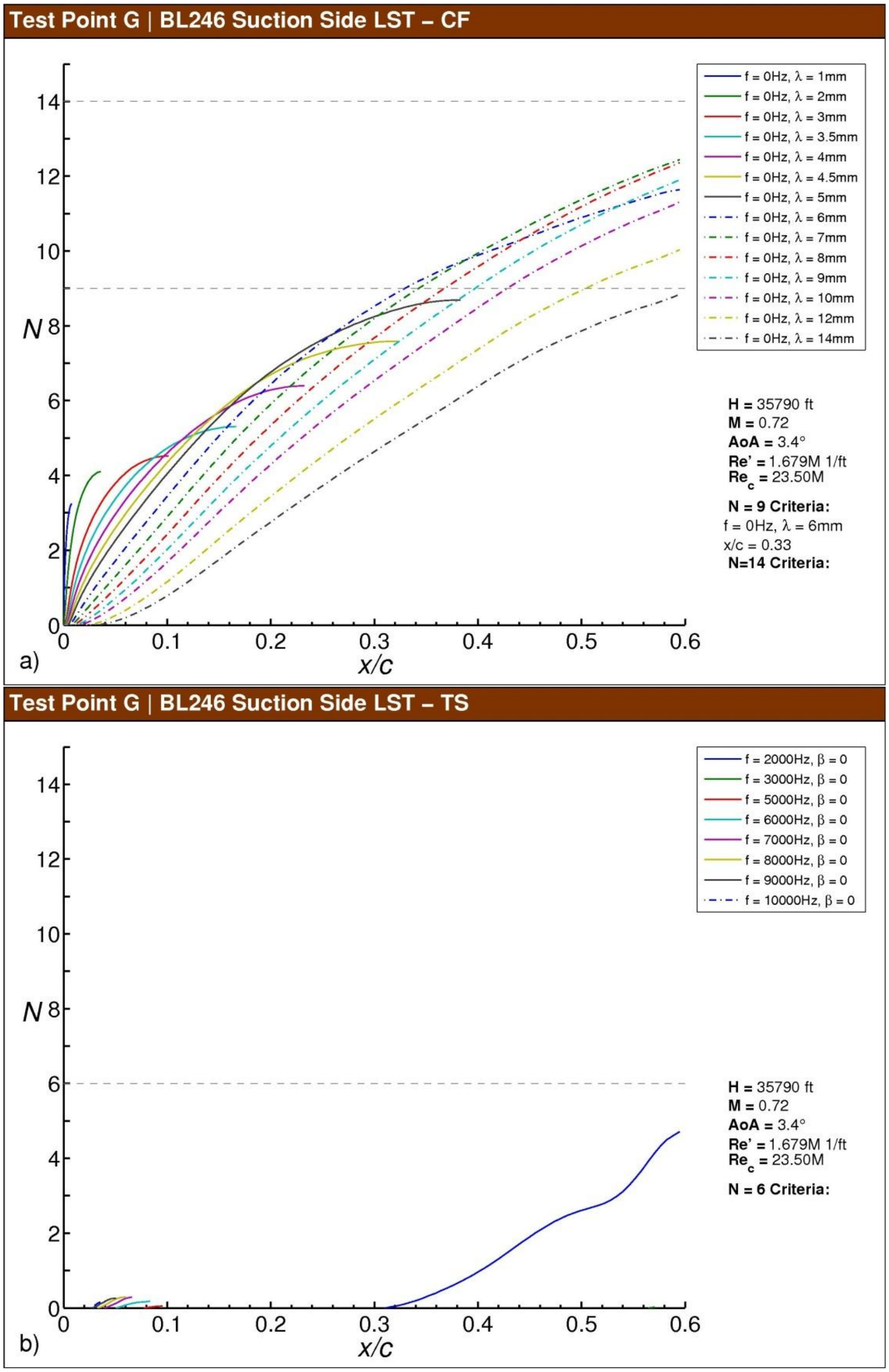


Fig. F-86 Test Point G *N*-factor results for BL234, pressure side: a) stationary crossflow, b) T-S.



Test Point G | BL246 Suction Side LST – CF

H = 35790 ft
M = 0.72
AoA = 3.4°
Re' = 1.679M 1/ft
Re$_c$ = 23.50M

N = 9 Criteria:
f = 0Hz, λ = 6mm
x/c = 0.33
N=14 Criteria:

Test Point G | BL246 Suction Side LST – TS

H = 35790 ft
M = 0.72
AoA = 3.4°
Re' = 1.679M 1/ft
Re$_c$ = 23.50M

N = 6 Criteria:

Fig. F-87 Test Point G *N*-factor results for BL246, suction side: a) stationary crossflow, b) T-S.

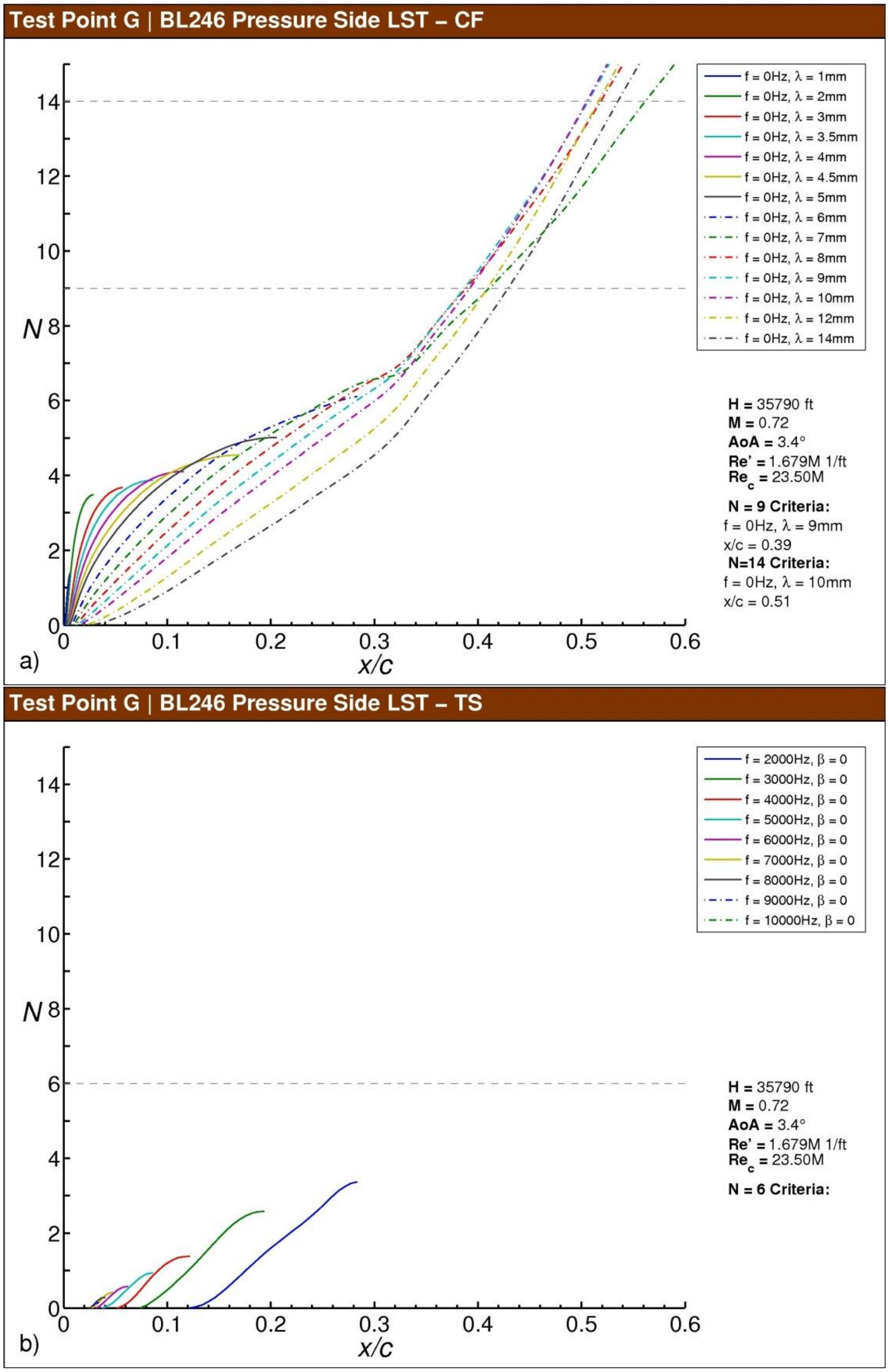
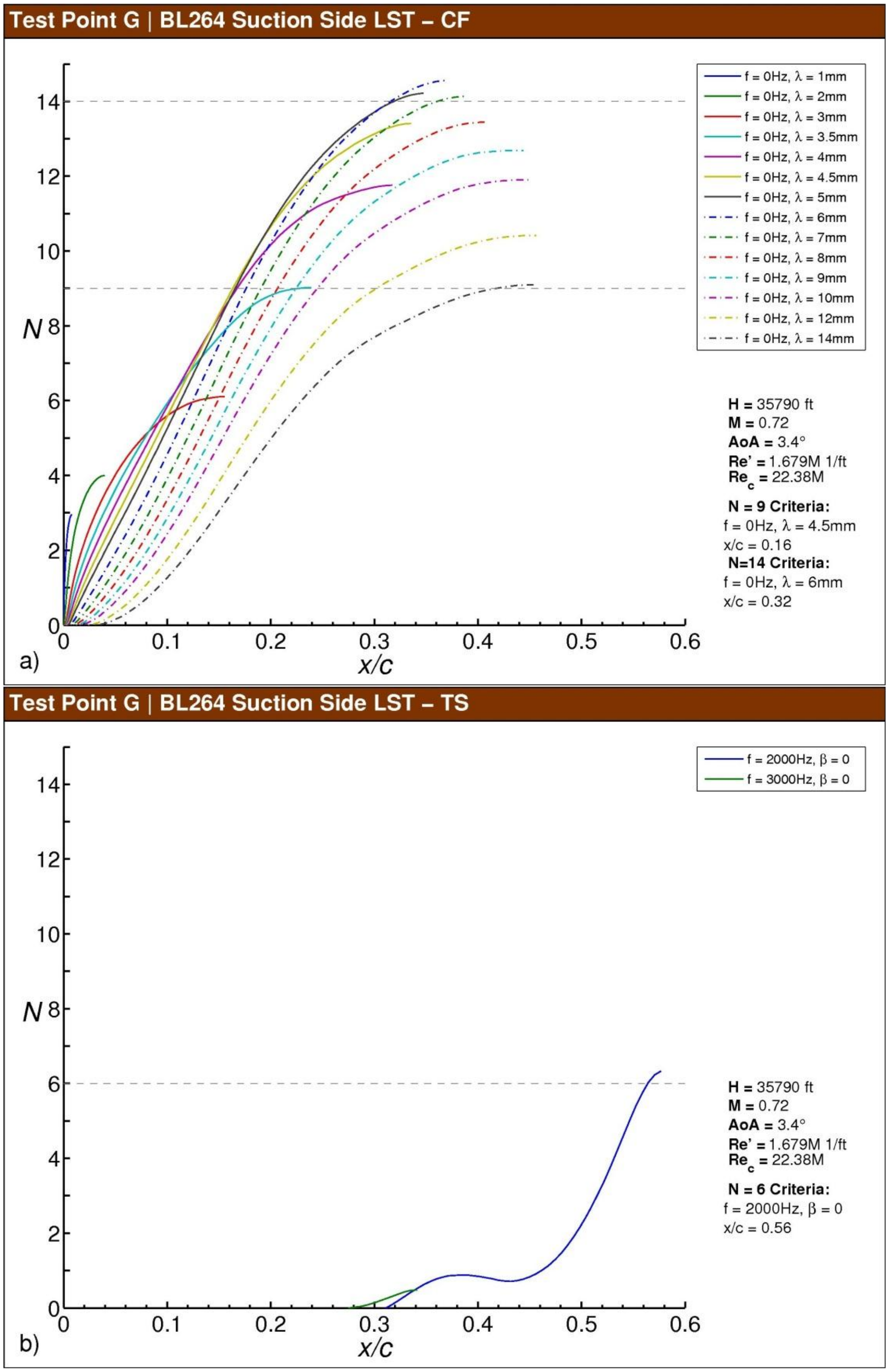


Fig. F-88 Test Point G *N*-factor results for BL246, pressure side: a) stationary crossflow, b) T-S.


Test Point G | BL264 Suction Side LST – CF

H = 35790 ft
M = 0.72
AoA = 3.4°
Re' = 1.679M 1/ft
Re_c = 22.38M

N = 9 Criteria:
f = 0Hz, λ = 4.5mm
x/c = 0.16

N=14 Criteria:
f = 0Hz, λ = 6mm
x/c = 0.32

Test Point G | BL264 Suction Side LST – TS

H = 35790 ft
M = 0.72
AoA = 3.4°
Re' = 1.679M 1/ft
Re_c = 22.38M

N = 6 Criteria:
f = 2000Hz, β = 0
x/c = 0.56

Fig. F-89 Test Point G *N*-factor results for BL264, suction side: a) stationary crossflow, b) T-S.

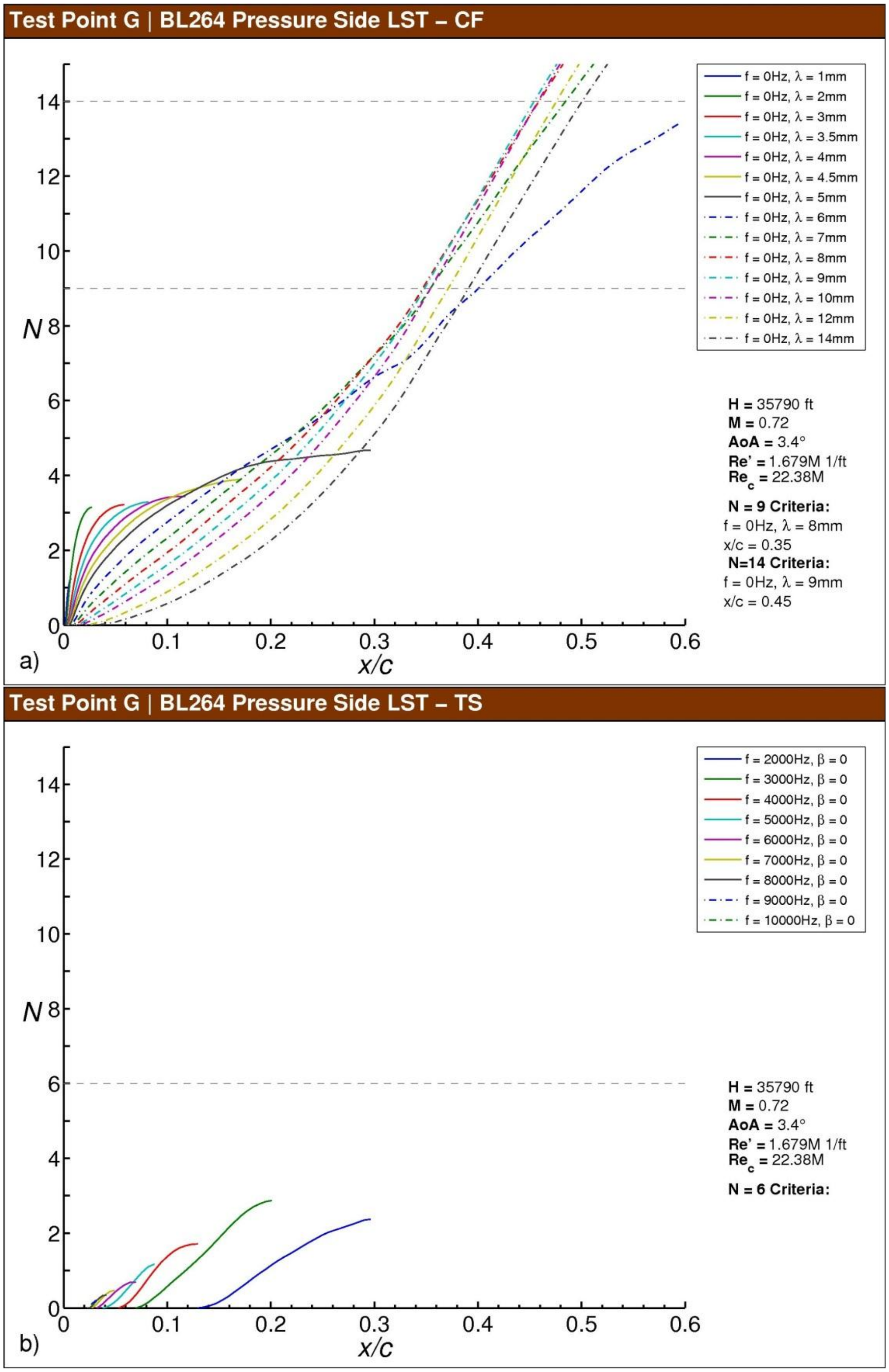


Fig. F-90 Test Point G *N*-factor results for BL264, pressure side: a) stationary crossflow, b) T-S.

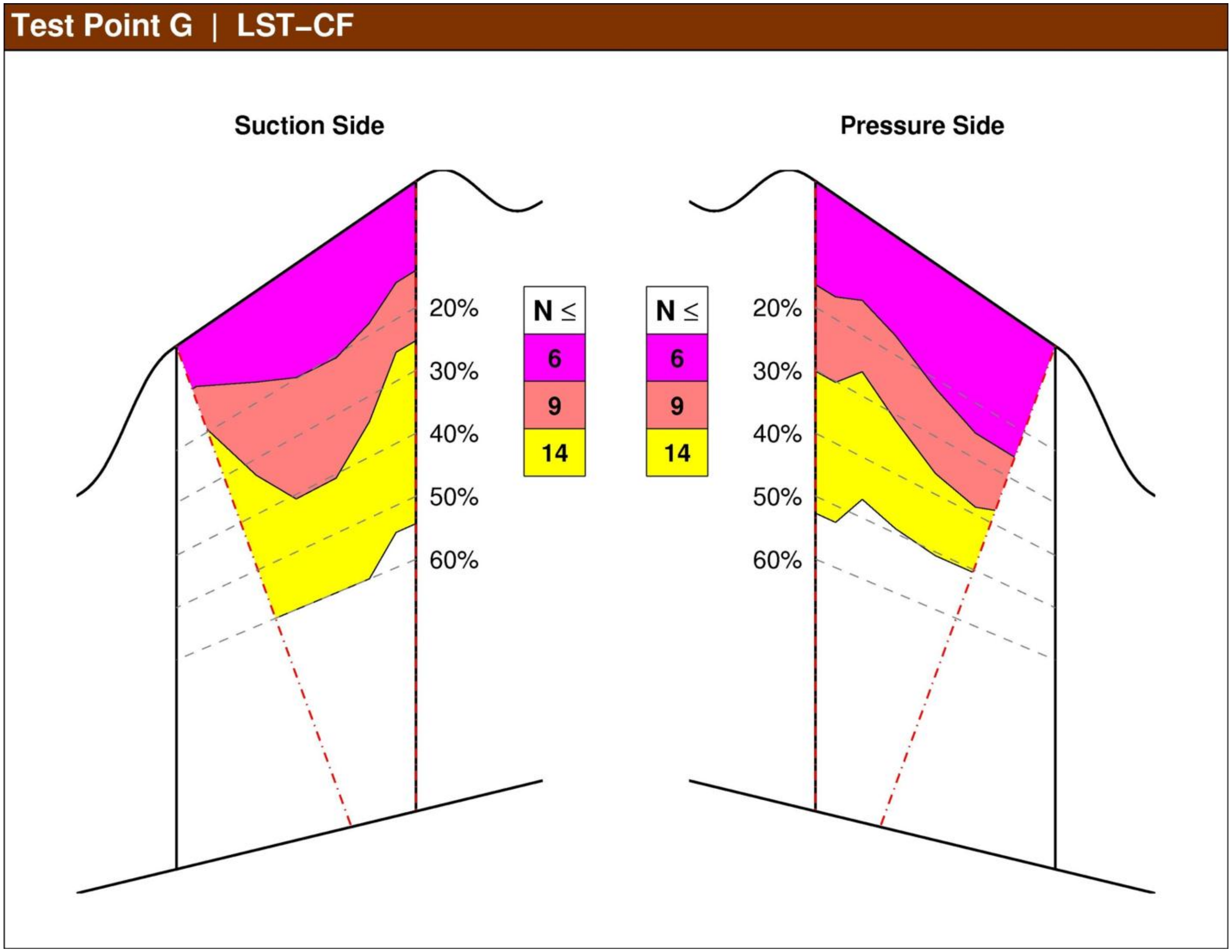


Fig. F-91 Test Point G *N*-factor contours for the test section suction and pressure sides.

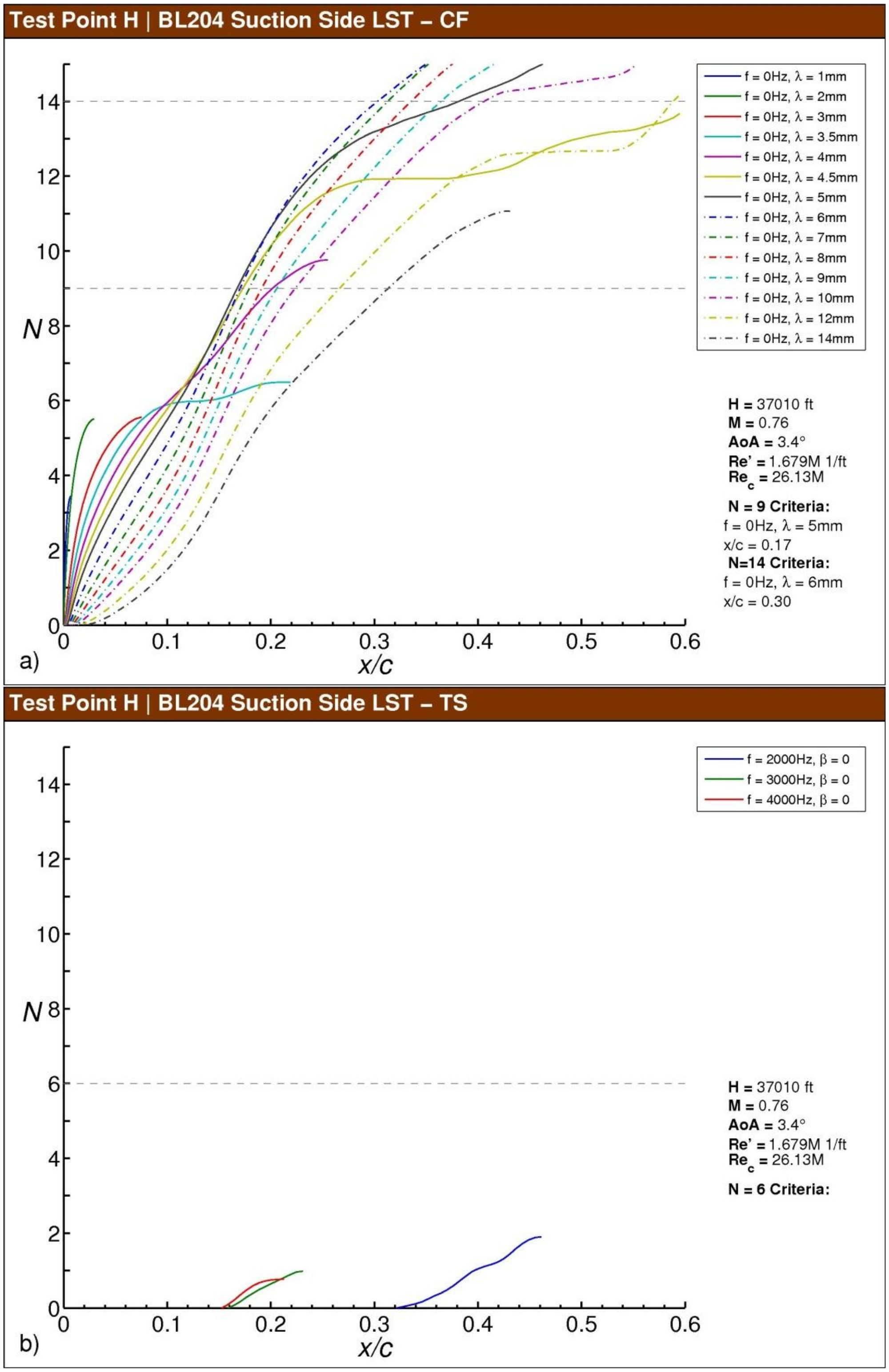
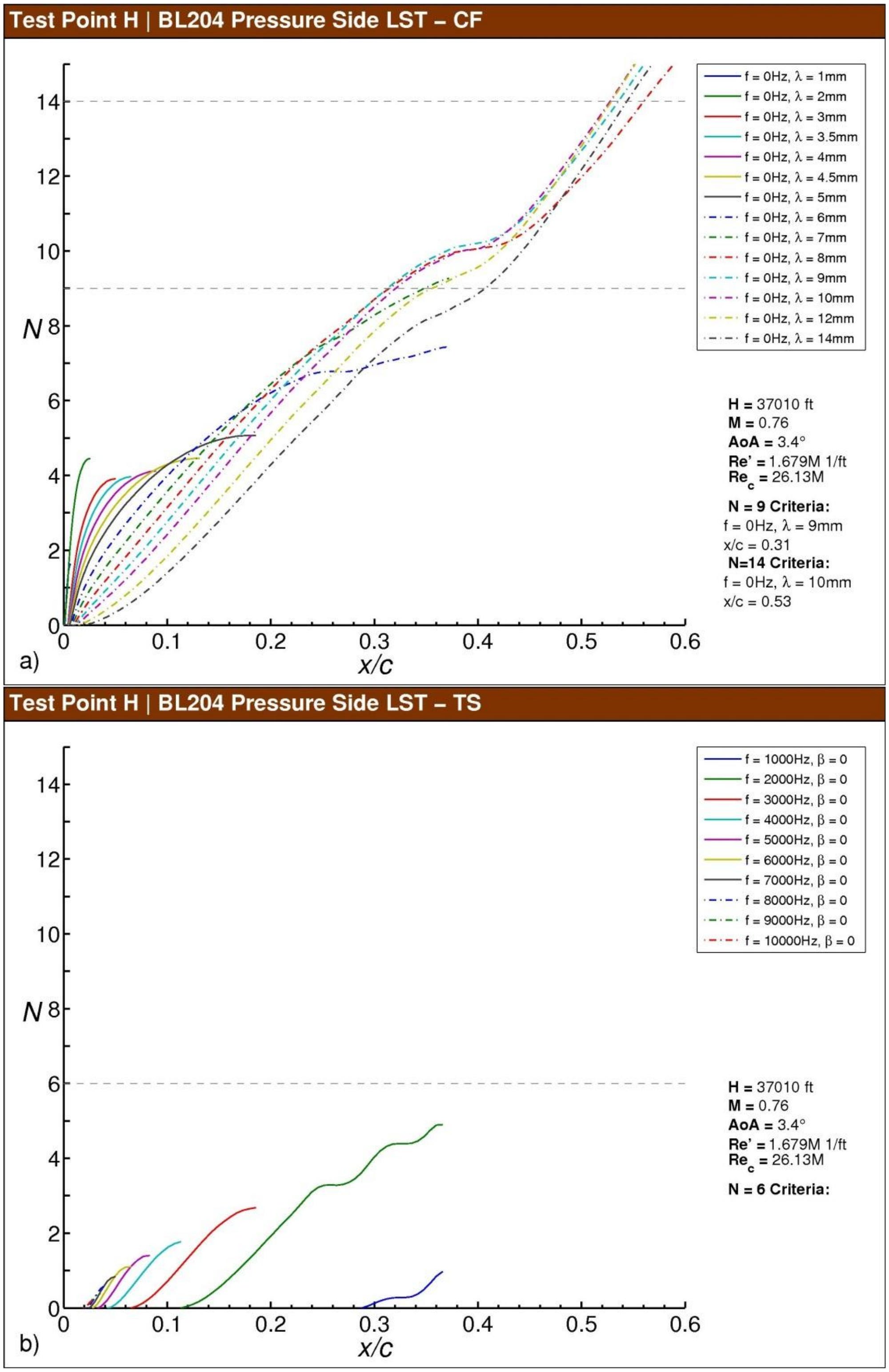


Fig. F-92 Test Point H *N*-factor results for BL204, suction side: a) stationary crossflow, b) T-S.



Test Point H | BL204 Pressure Side LST – CF

Test Point H | BL204 Pressure Side LST – TS

Fig. F-93 Test Point H *N*-factor results for BL204, pressure side: a) stationary crossflow, b) T-S.

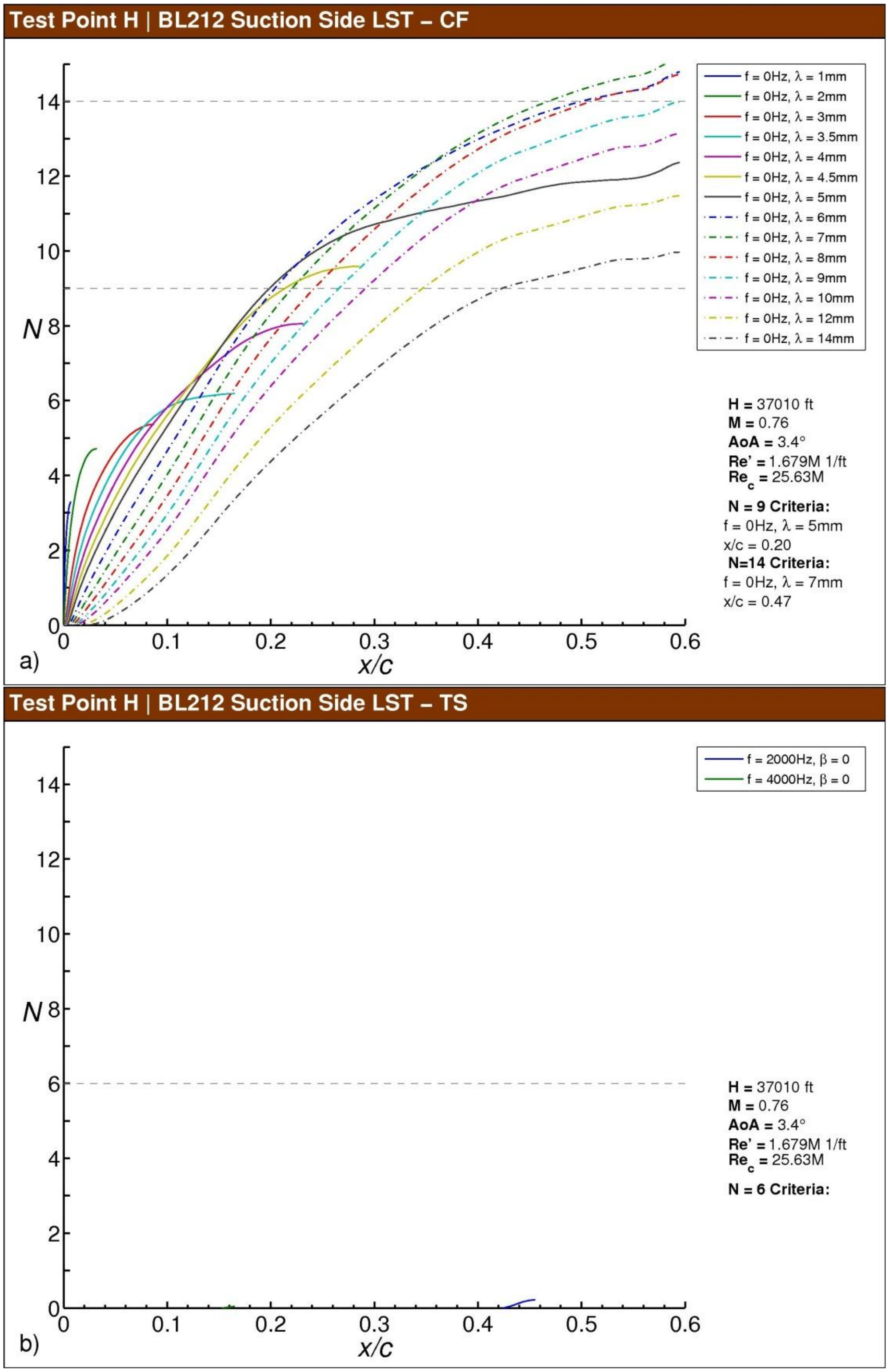


Fig. F-94 Test Point H *N*-factor results for BL212, suction side: a) stationary crossflow, b) T-S.

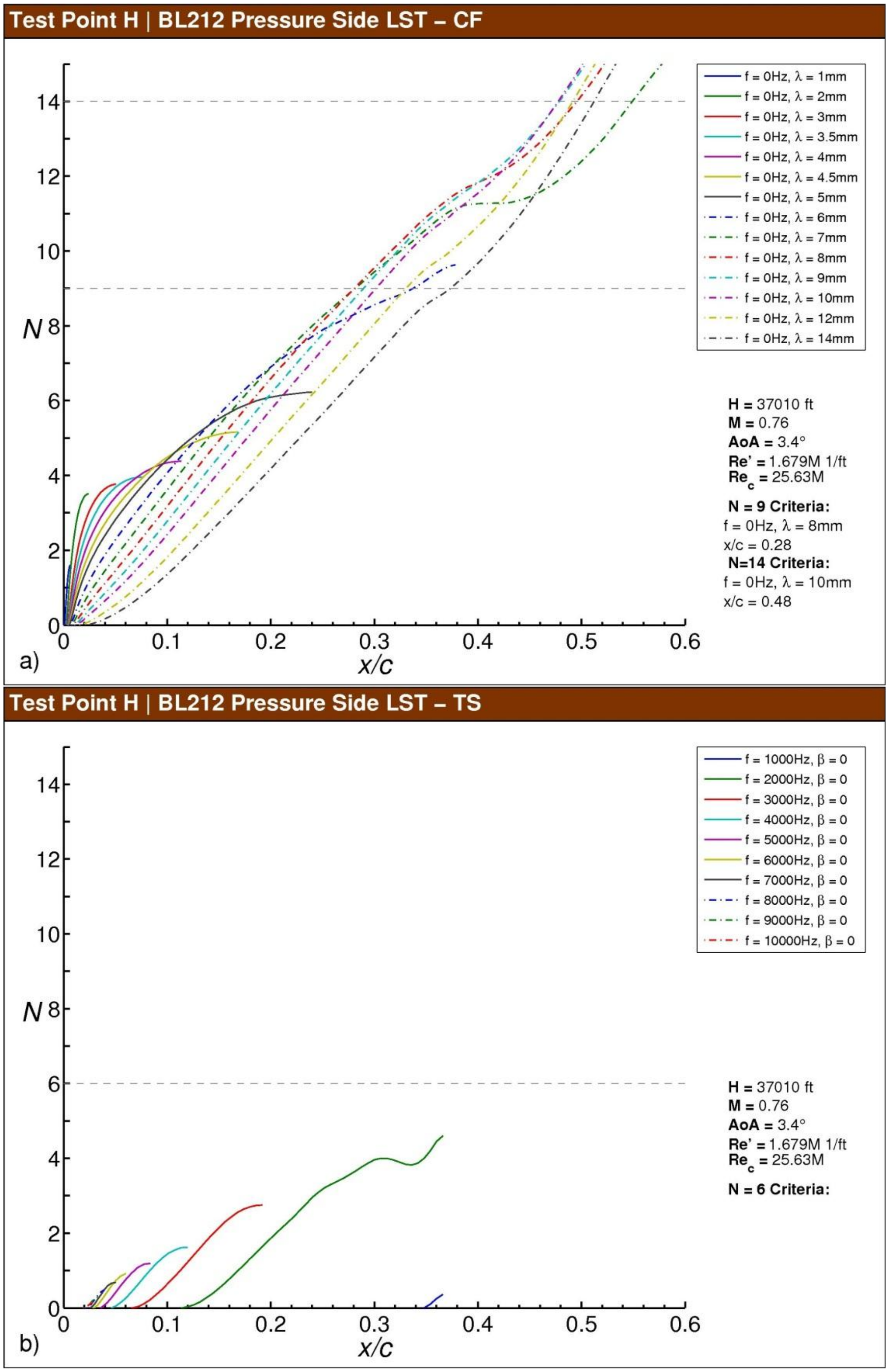
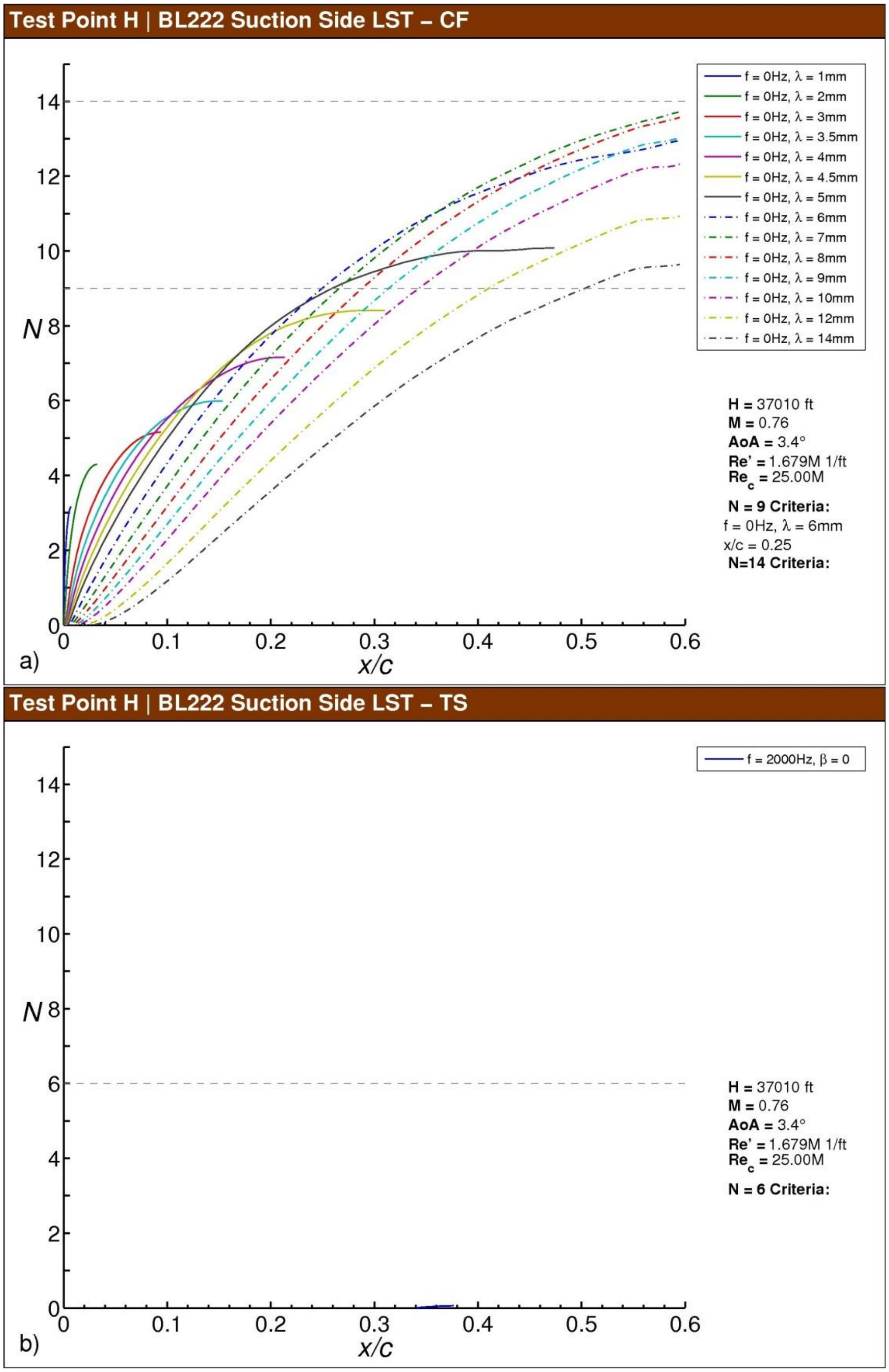


Fig. F-95 Test Point H *N*-factor results for BL212, pressure side: a) stationary crossflow, b) T-S.


Test Point H | BL222 Suction Side LST – CF

Test Point H | BL222 Suction Side LST – TS

Fig. F-96 Test Point H *N*-factor results for BL222, suction side: a) stationary crossflow, b) T-S.

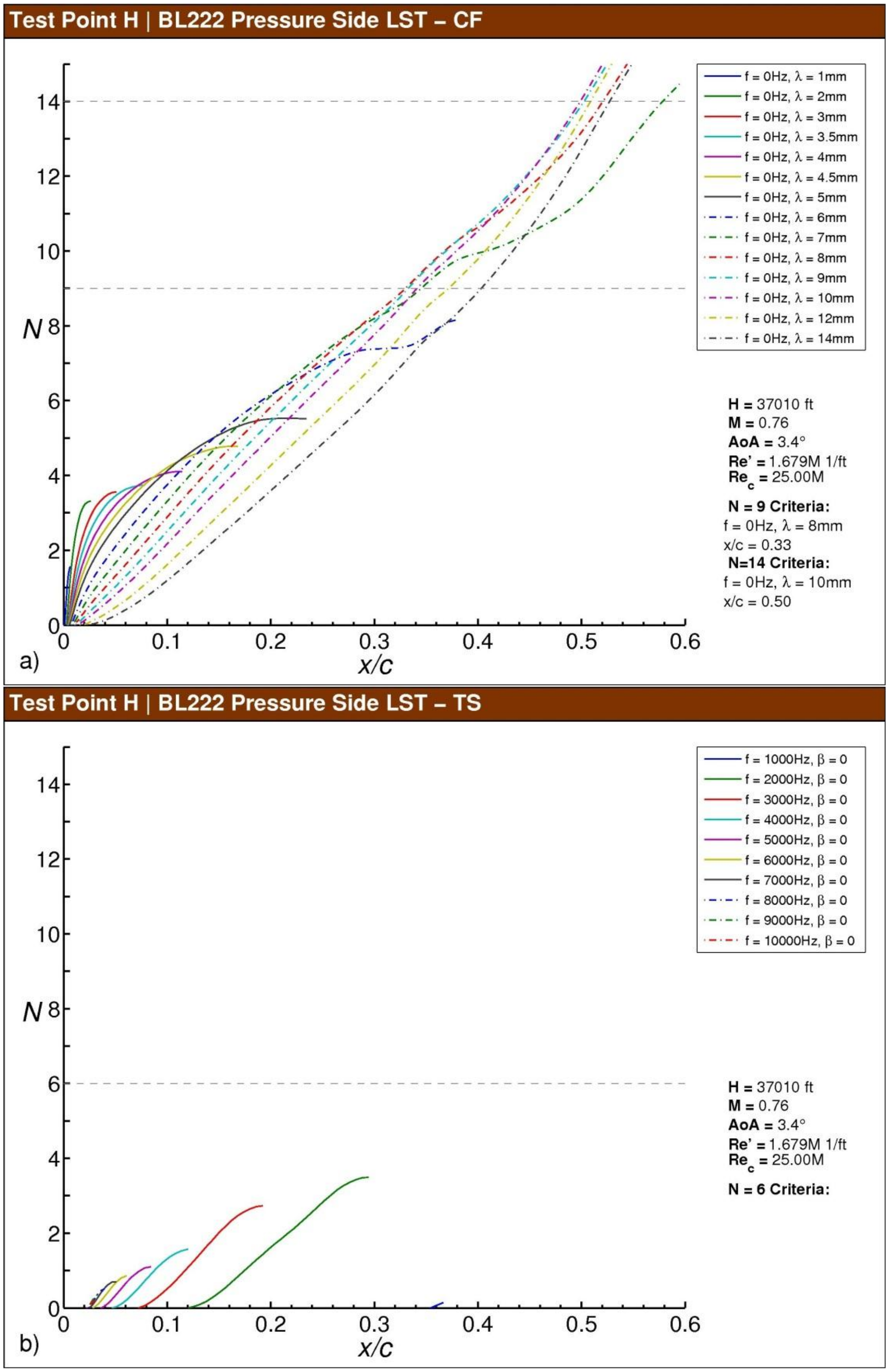


Fig. F-97 Test Point H *N*-factor results for BL222, pressure side: a) stationary crossflow, b) T-S.

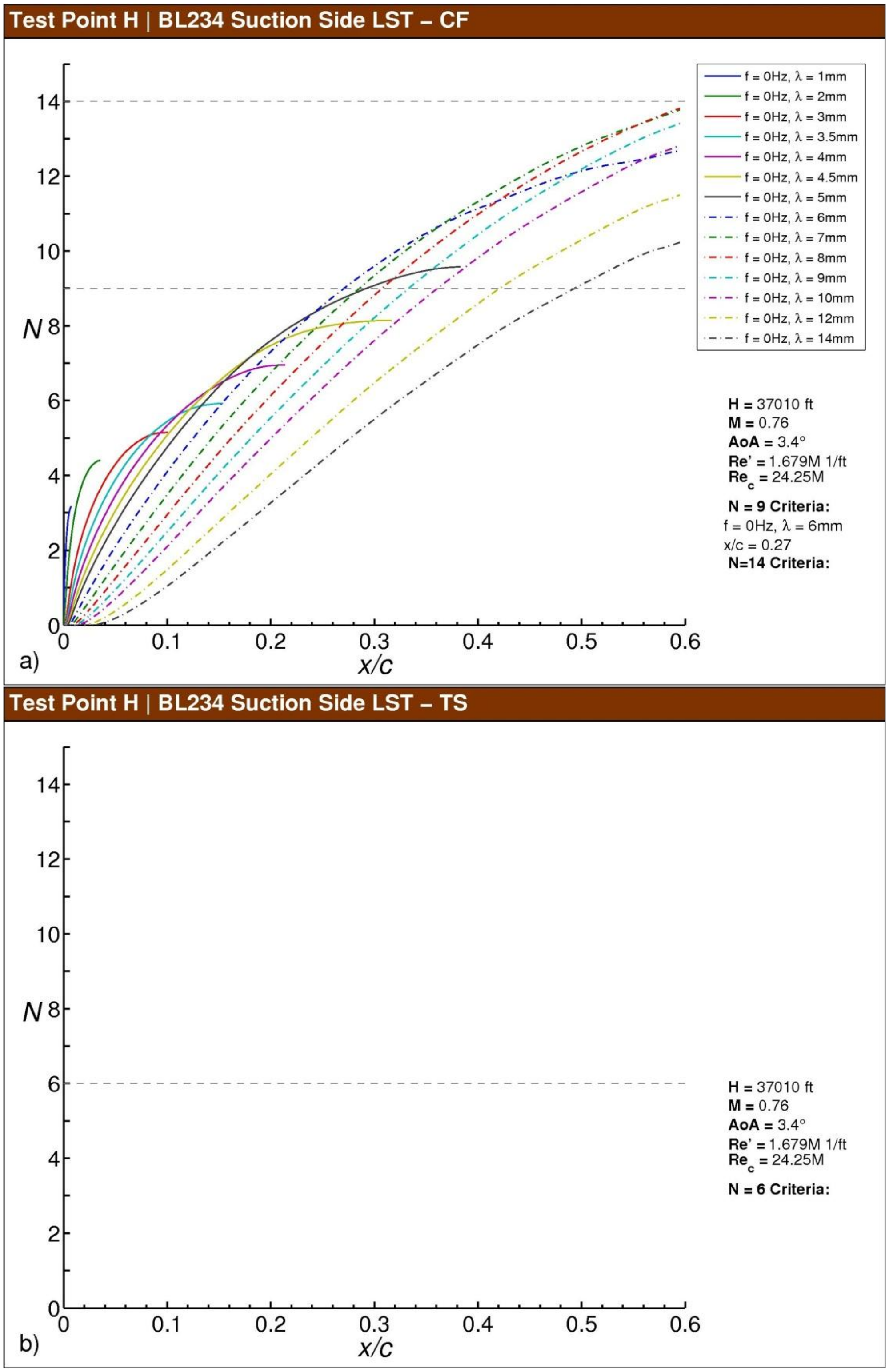


Fig. F-98 Test Point H *N*-factor results for BL234, suction side: a) stationary crossflow, b) T-S.

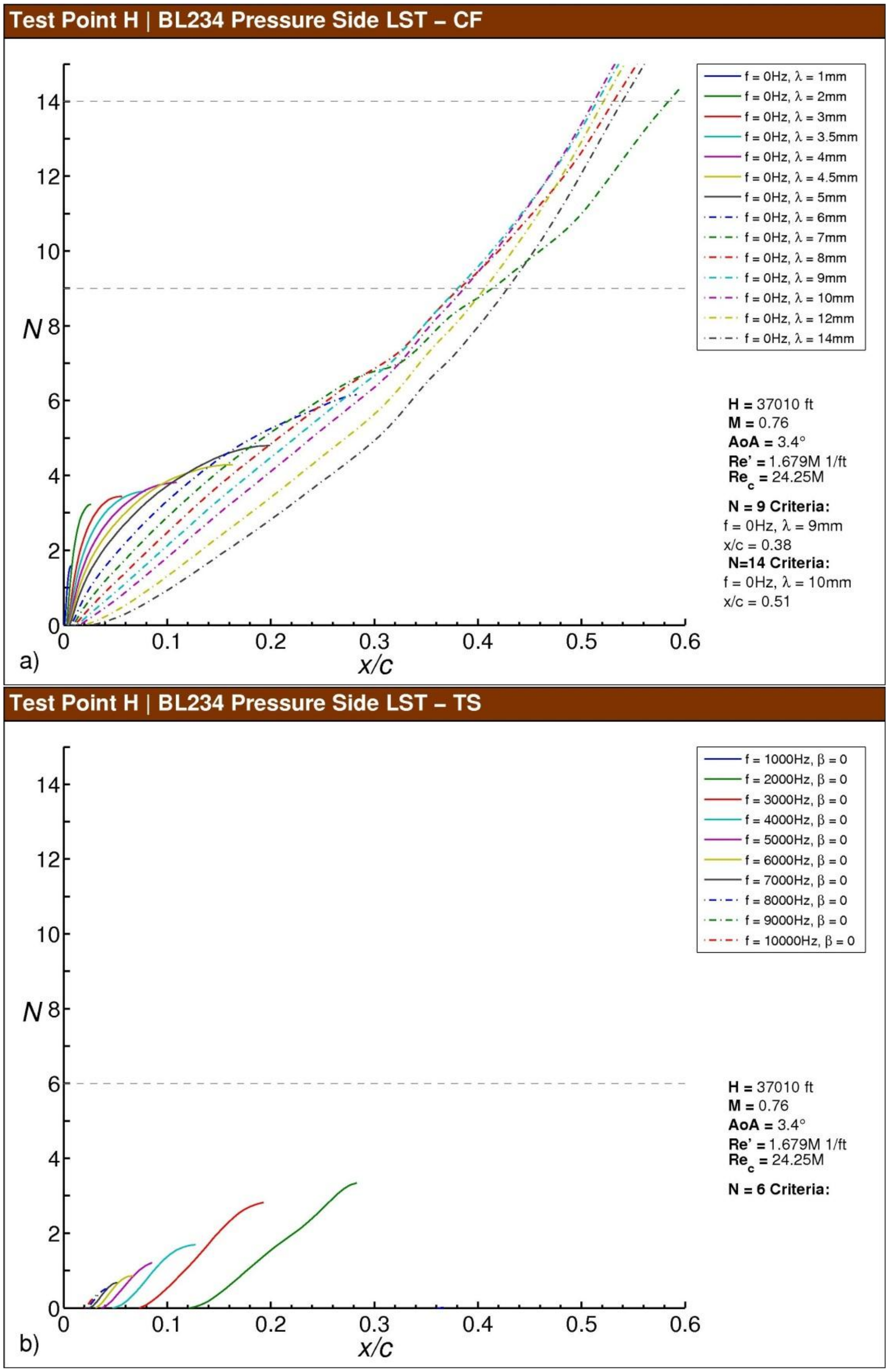


Fig. F-99 Test Point H *N*-factor results for BL234, pressure side: a) stationary crossflow, b) T-S.

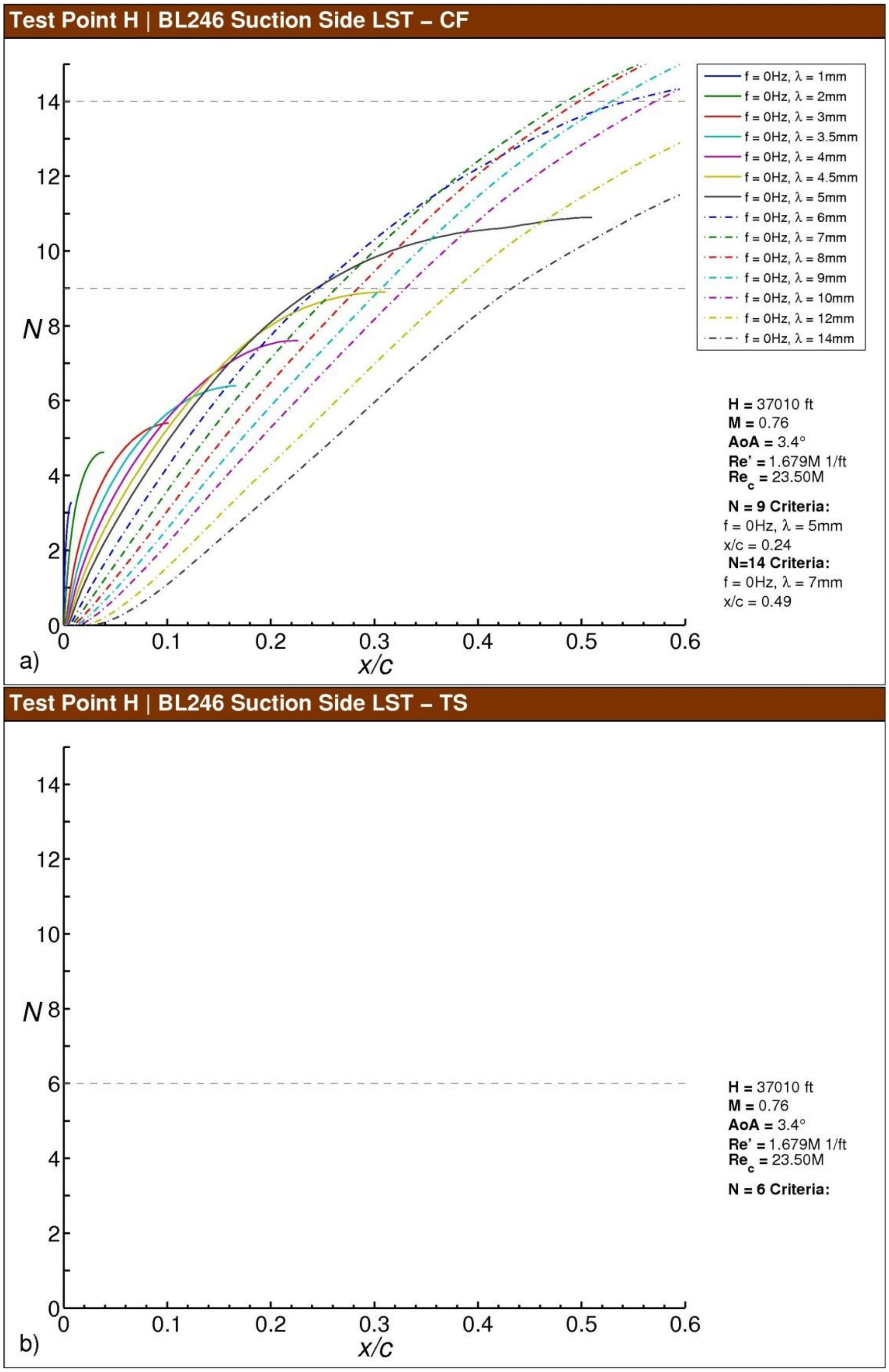


Fig. F-100 Test Point H *N*-factor results for BL246, suction side: a) stationary crossflow, b) T-S.

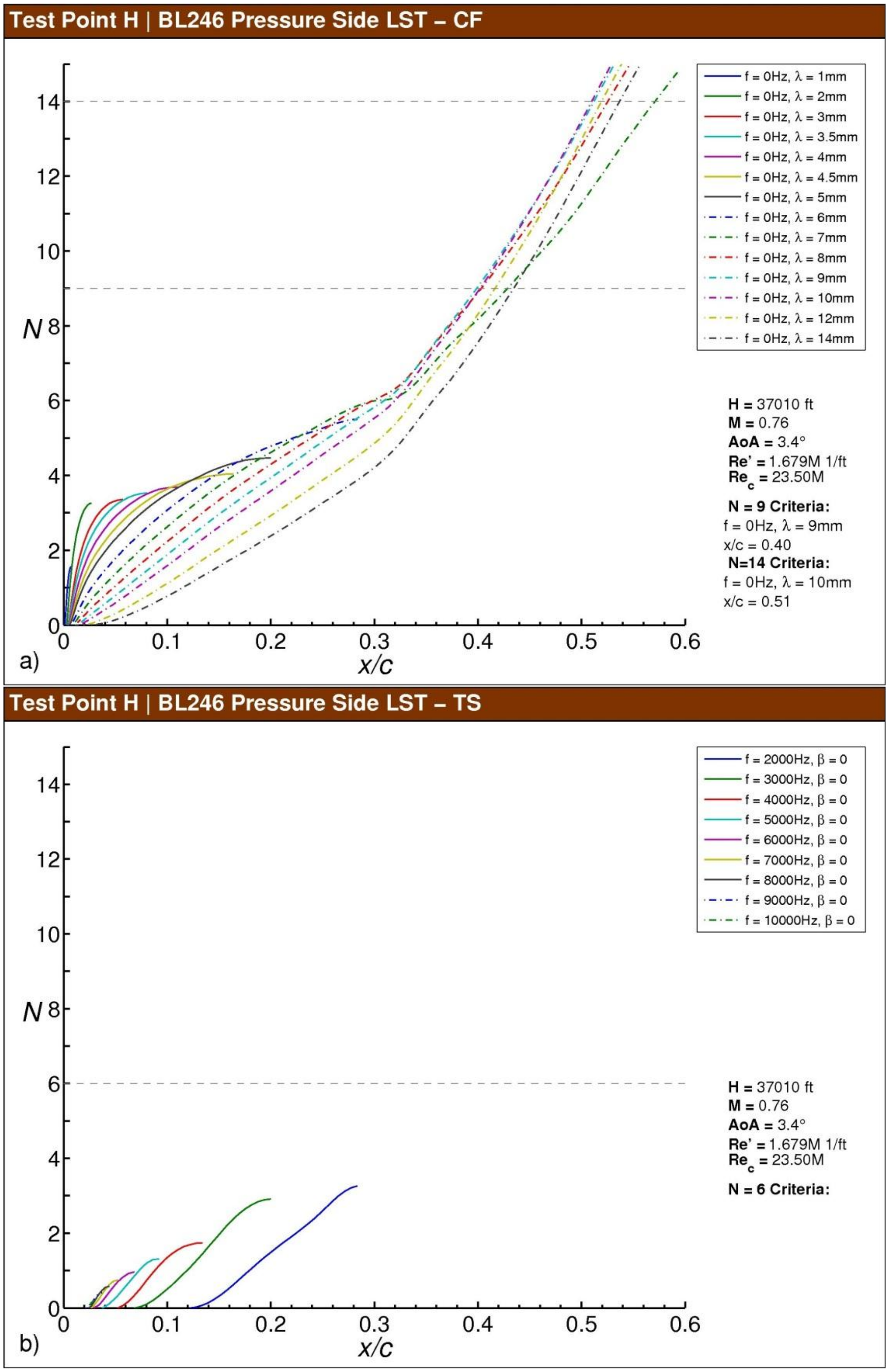


Fig. F-101 Test Point H *N*-factor results for BL246, pressure side: a) stationary crossflow, b) T-S.

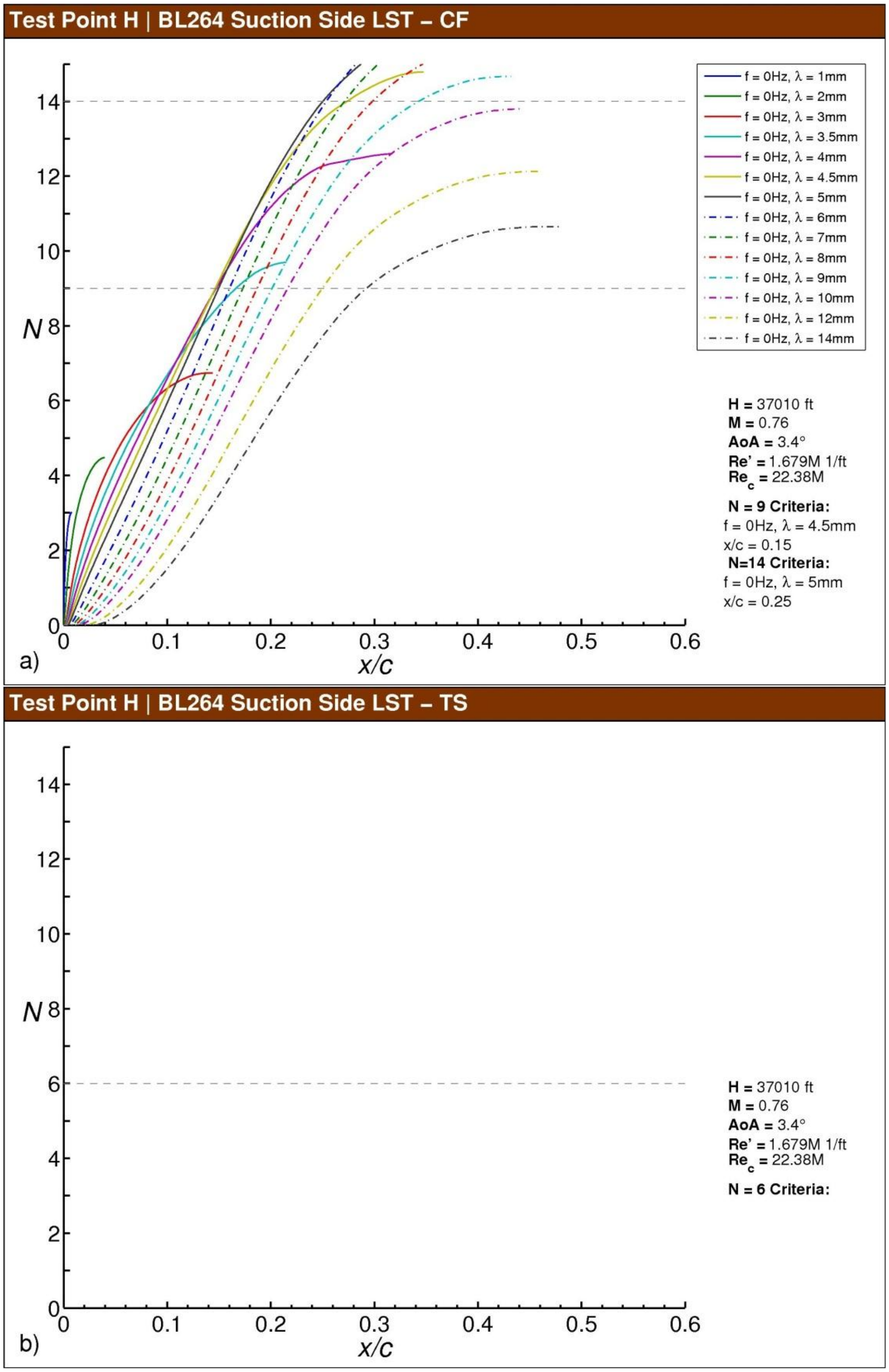


Fig. F-102 Test Point H *N*-factor results for BL264, suction side: a) stationary crossflow, b) T-S.

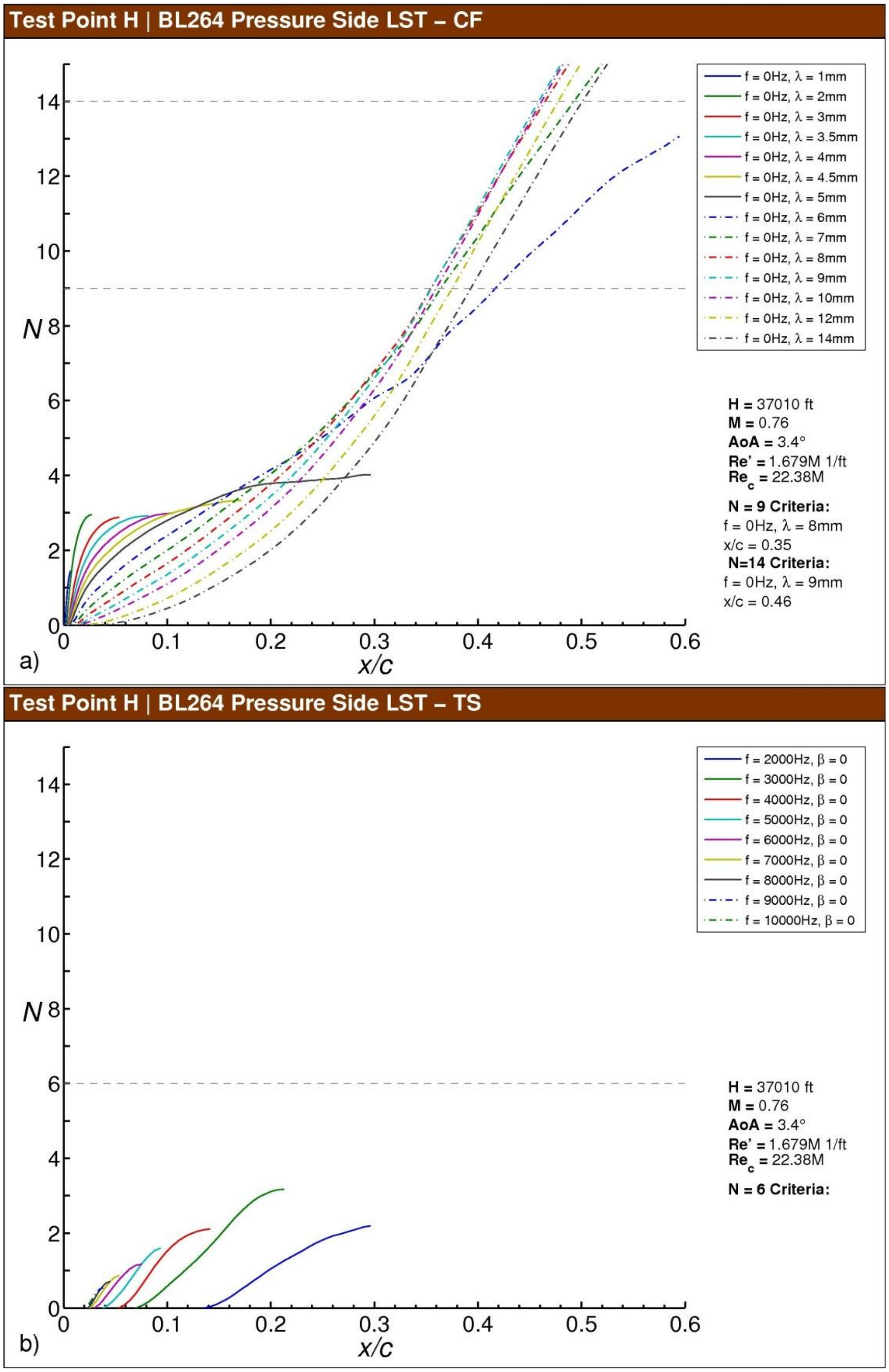
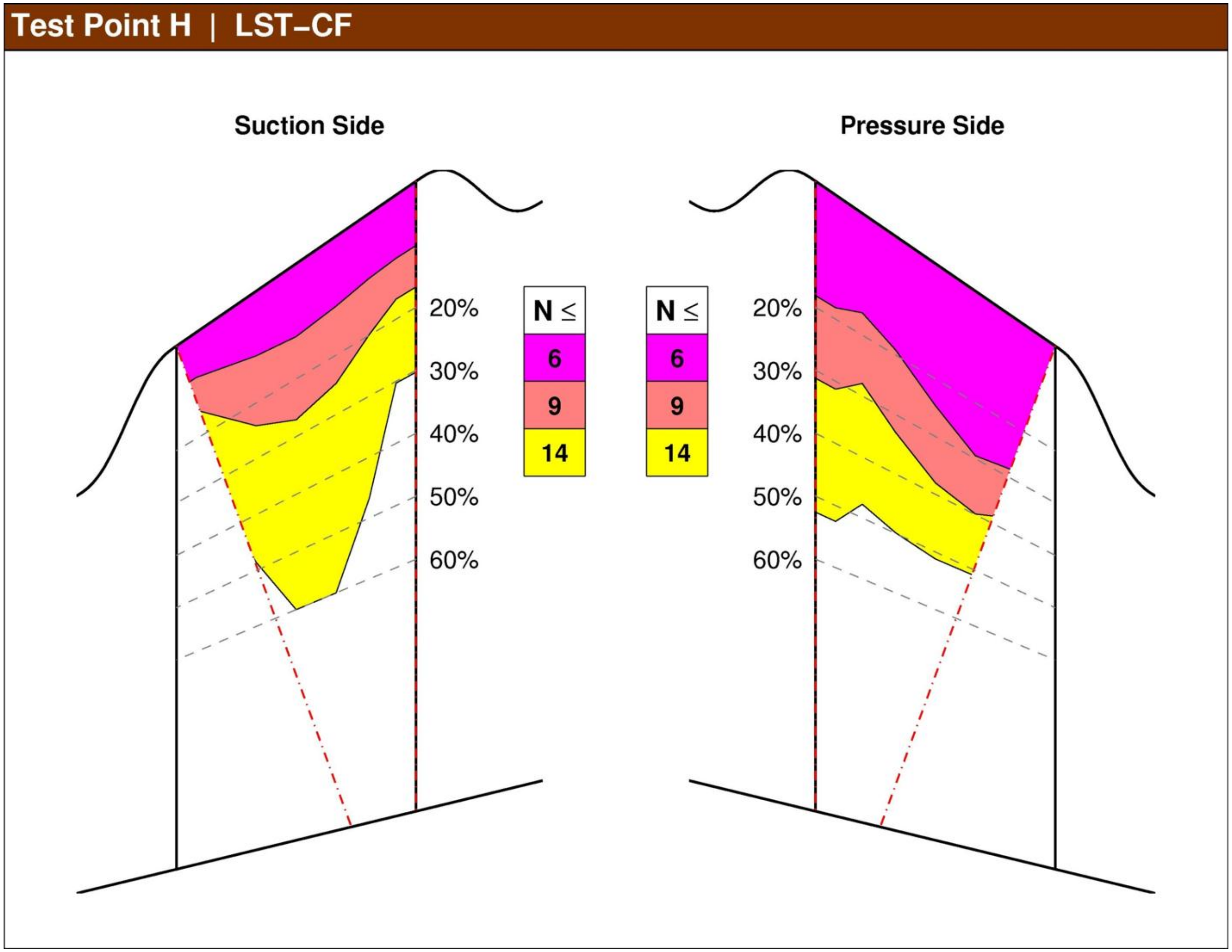


Fig. F-103 Test Point H *N*-factor results for BL264, pressure side: a) stationary crossflow, b) T-S.



Test Point H | LST–CF

Fig. F-104 Test Point H *N*-factor contours for the test section suction and pressure sides.

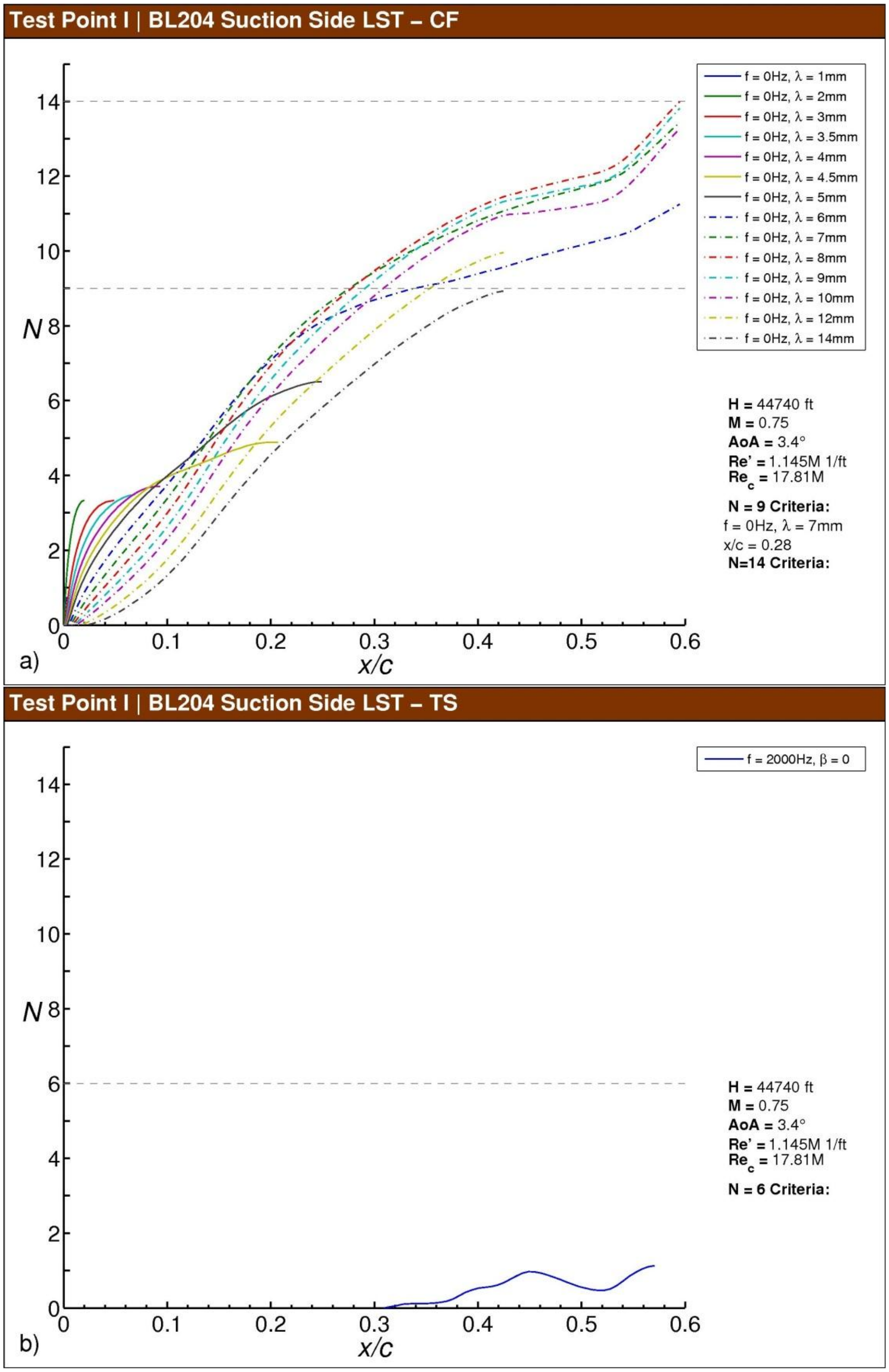


Fig. F-105 Test Point I *N*-factor results for BL204, suction side: a) stationary crossflow, b) T-S.

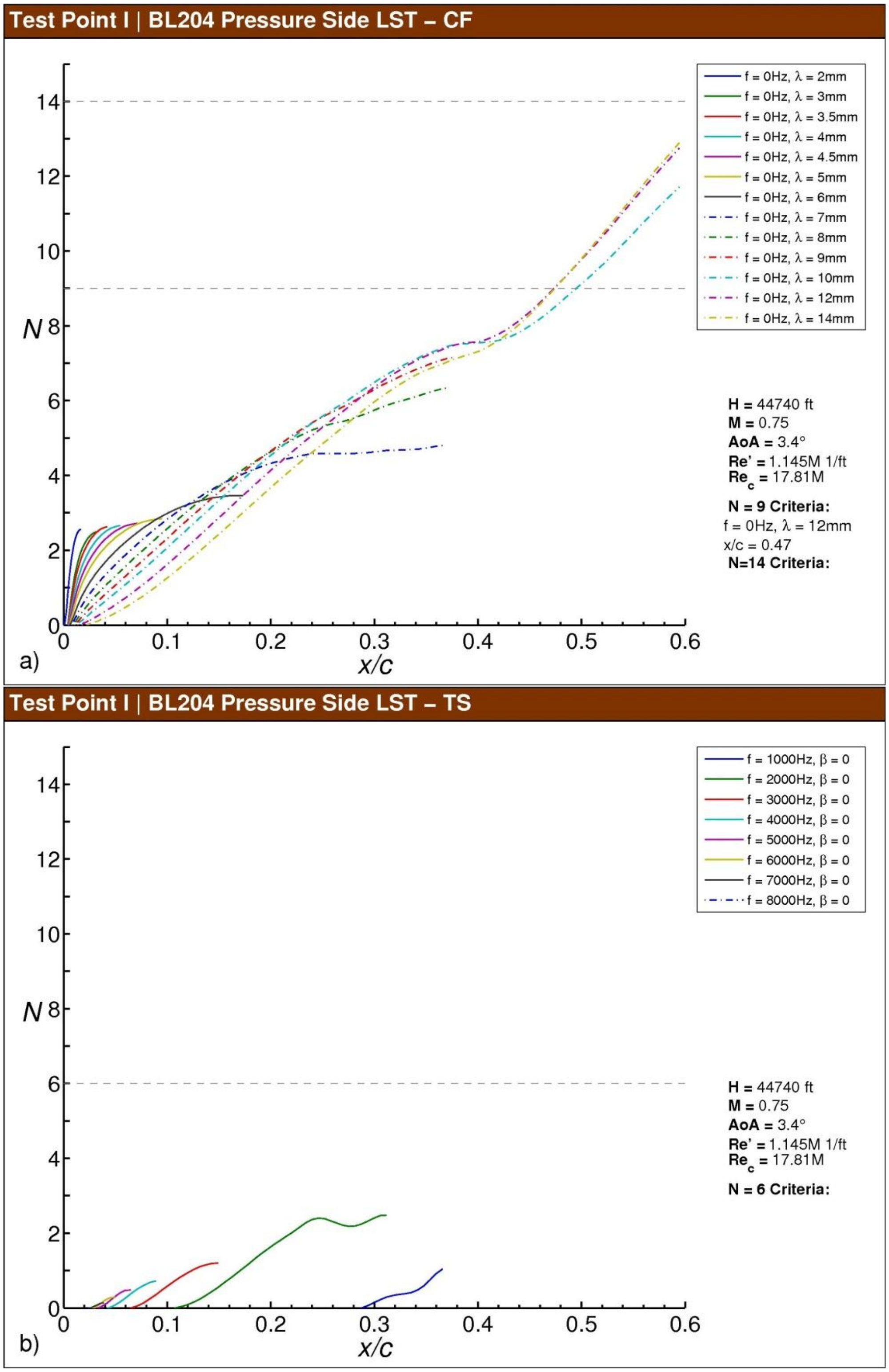


Fig. F-106 Test Point I *N*-factor results for BL204, pressure side: a) stationary crossflow, b) T-S.

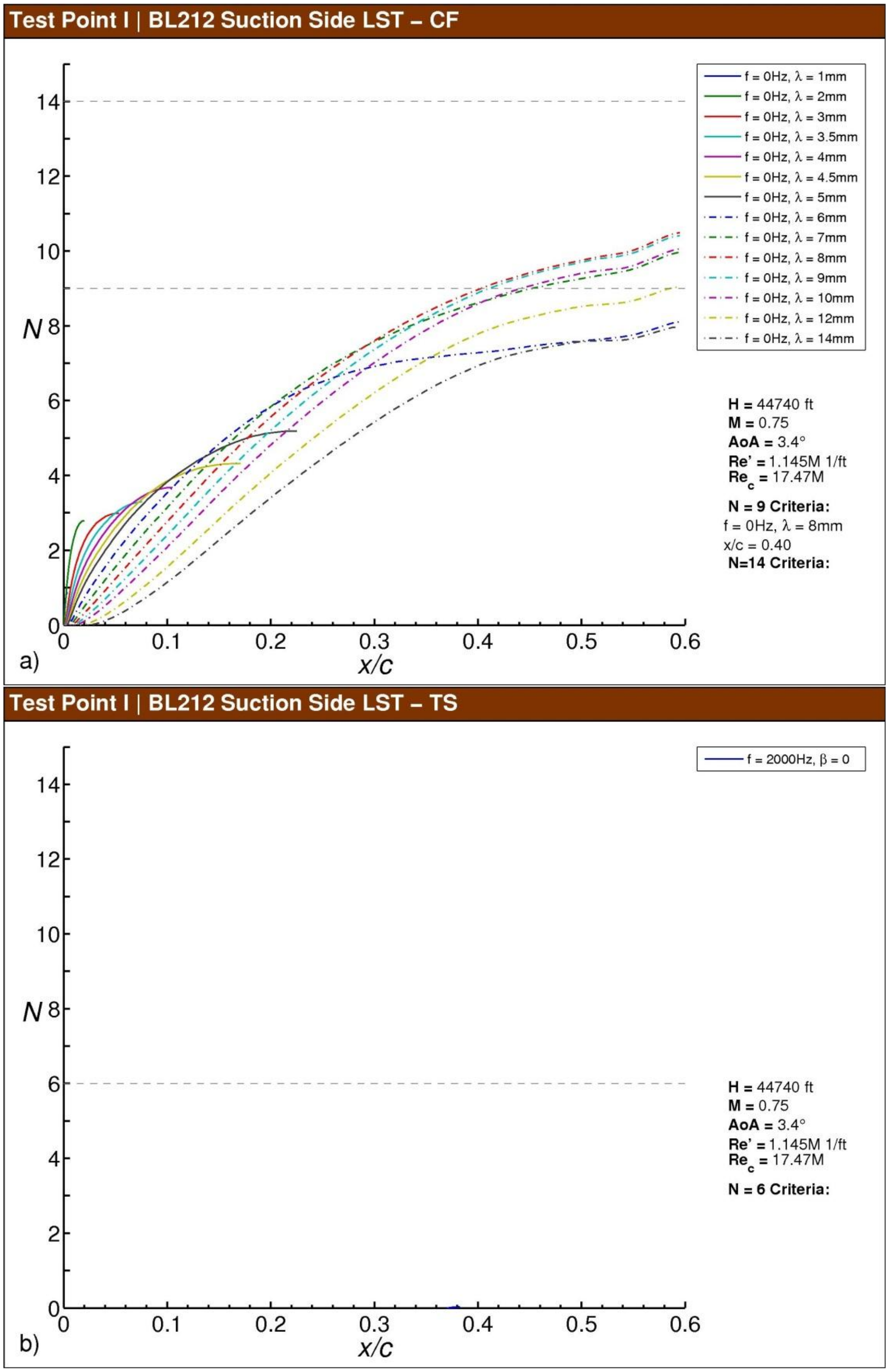


Fig. F-107 Test Point I *N*-factor results for BL212, suction side: a) stationary crossflow, b) T-S.

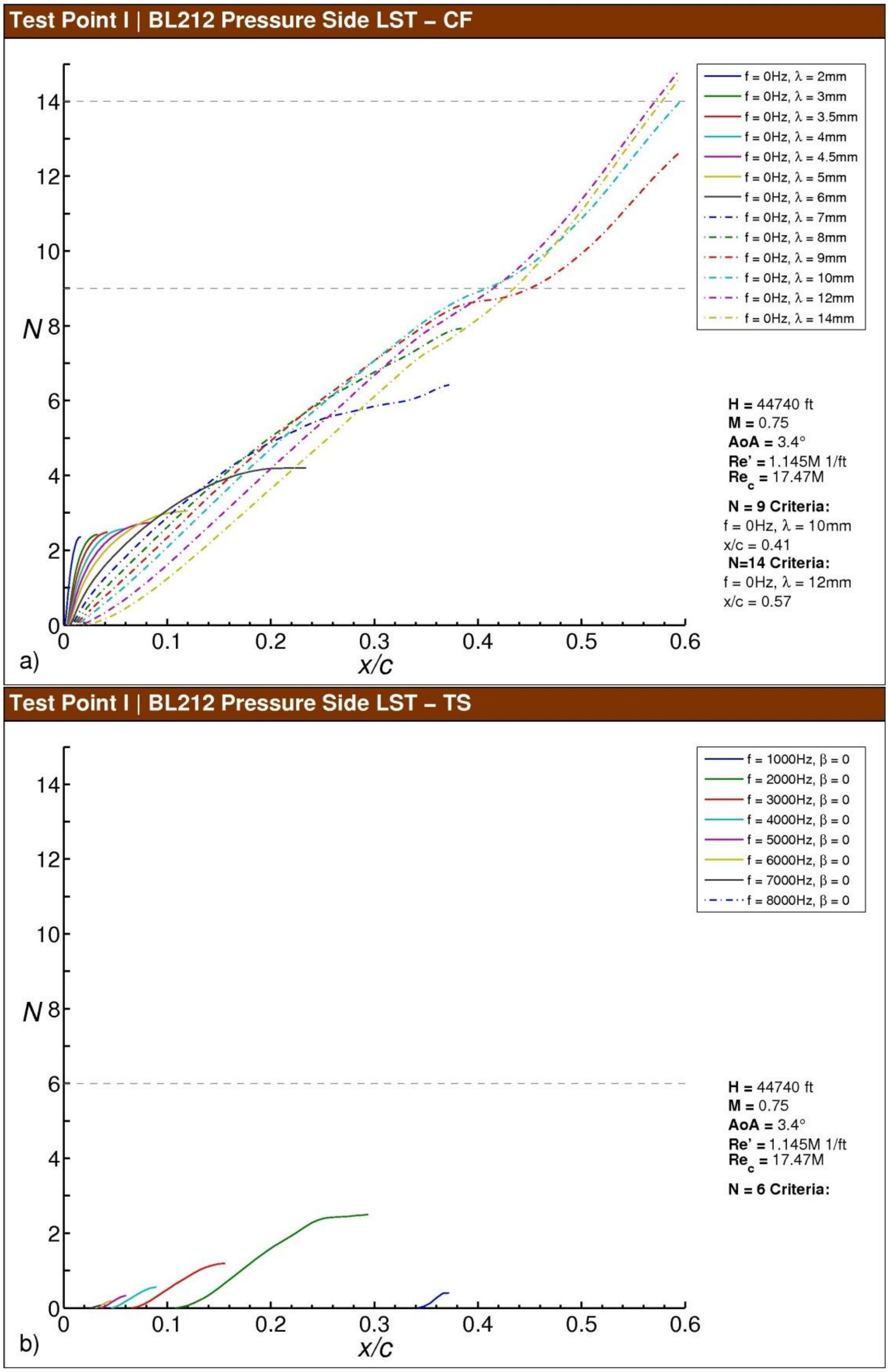


Fig. F-108 Test Point I *N*-factor results for BL212, pressure side: a) stationary crossflow, b) T-S.

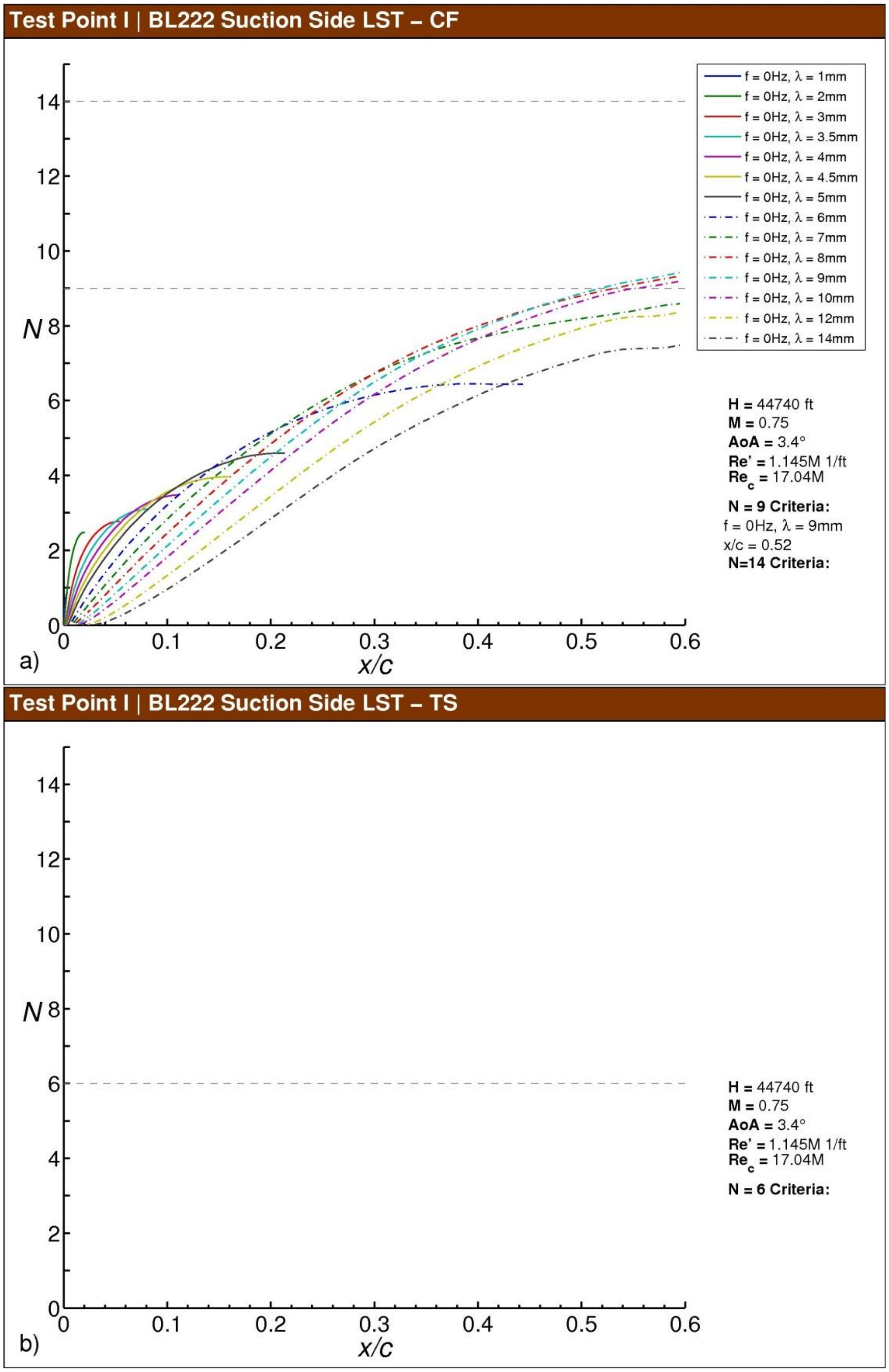


Fig. F-109 Test Point I *N*-factor results for BL222, suction side: a) stationary crossflow, b) T-S.

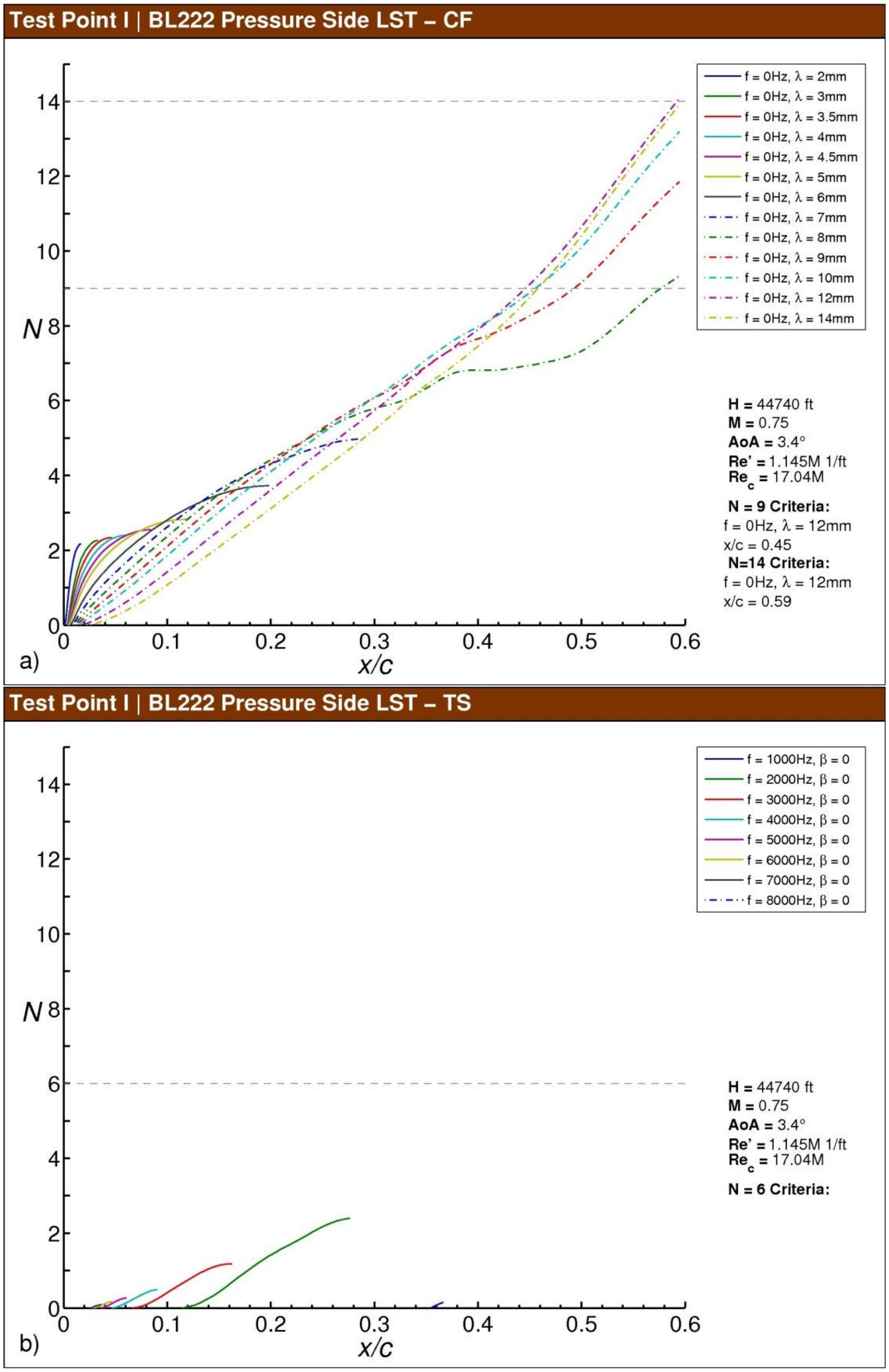


Fig. F-110 Test Point I *N*-factor results for BL222, pressure side: a) stationary crossflow, b) T-S.

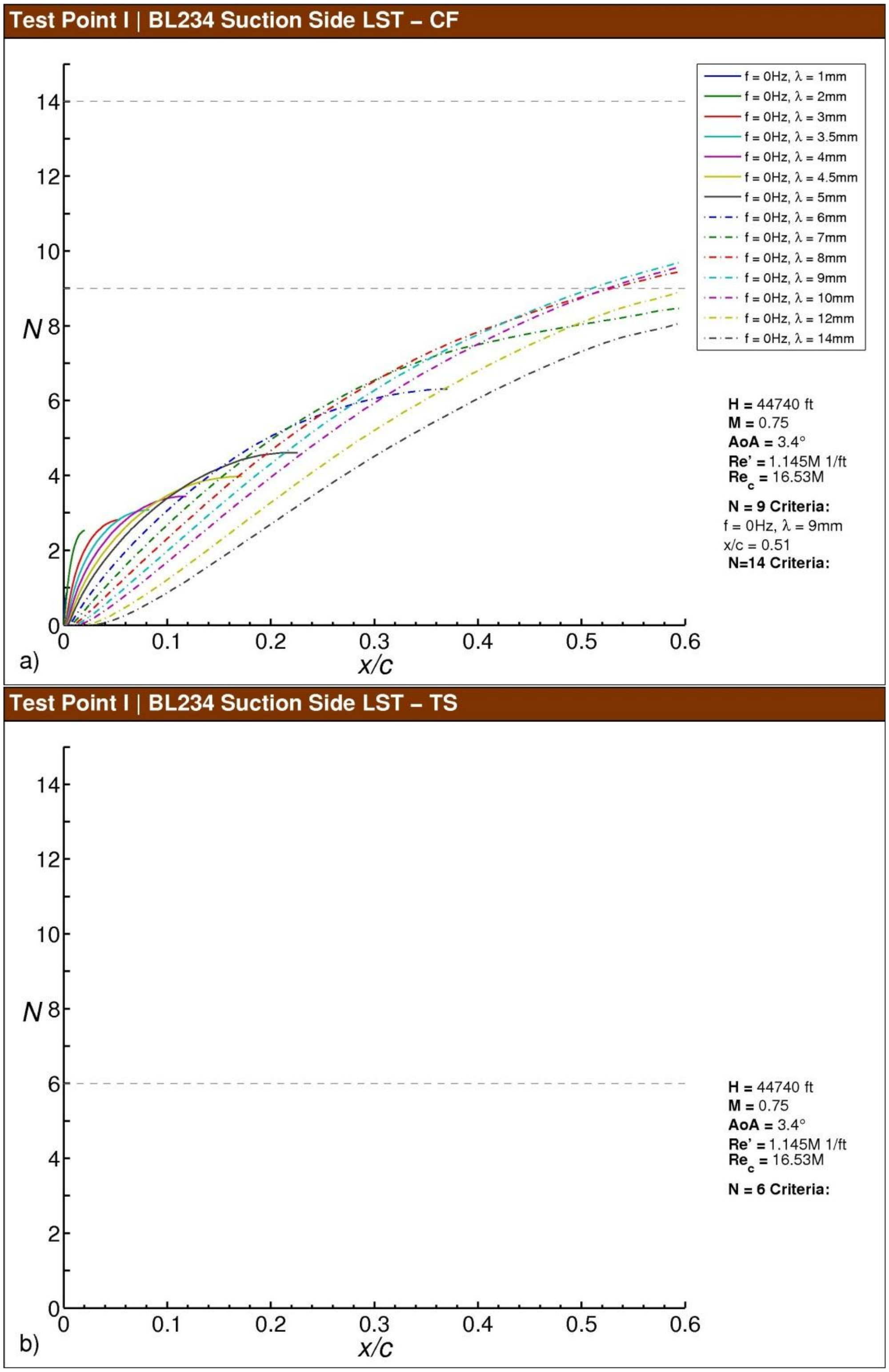


Fig. F-111 Test Point I *N*-factor results for BL234, suction side: a) stationary crossflow, b) T-S.

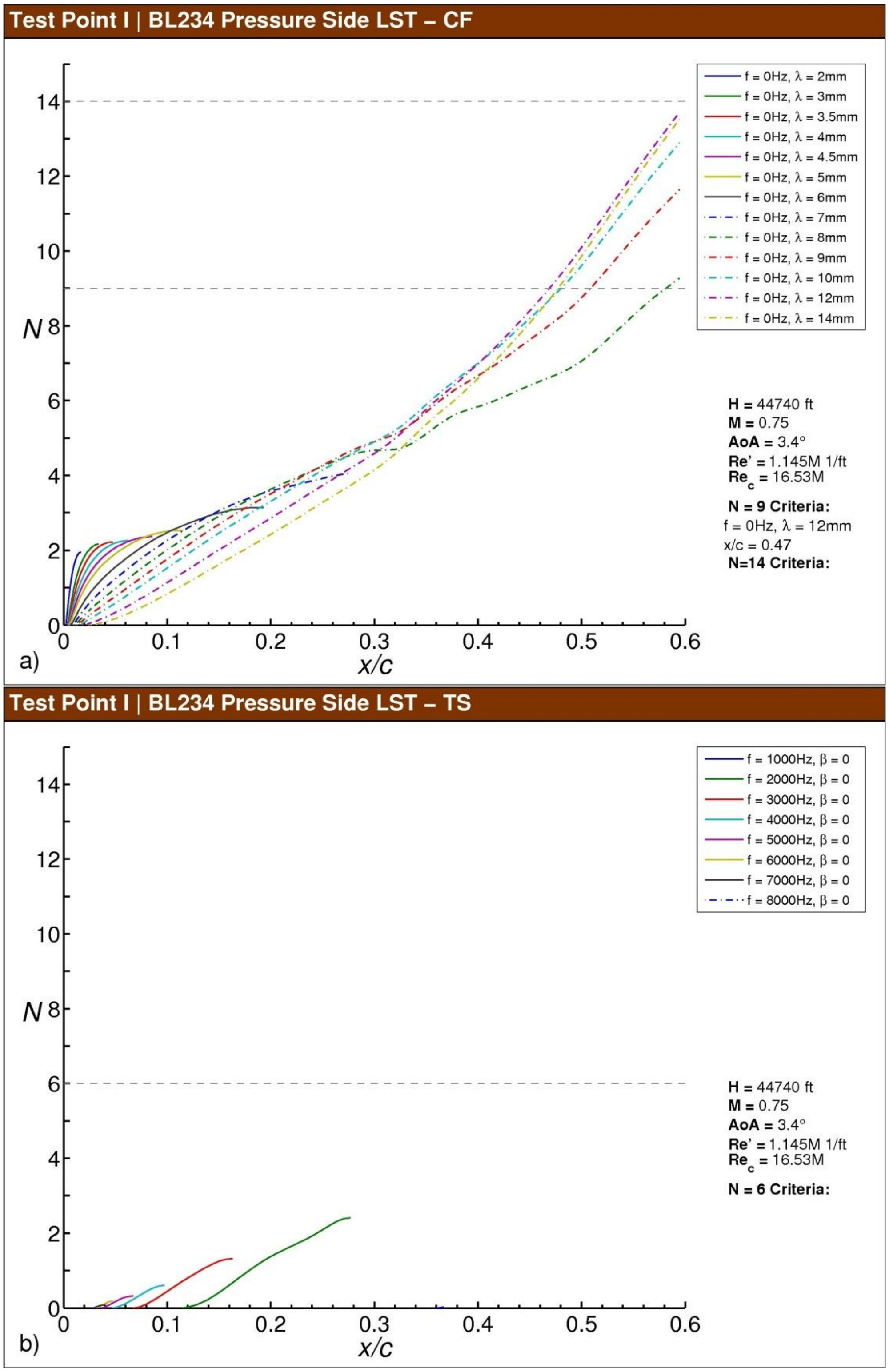


Fig. F-112 Test Point I *N*-factor results for BL234, pressure side: a) stationary crossflow, b) T-S.

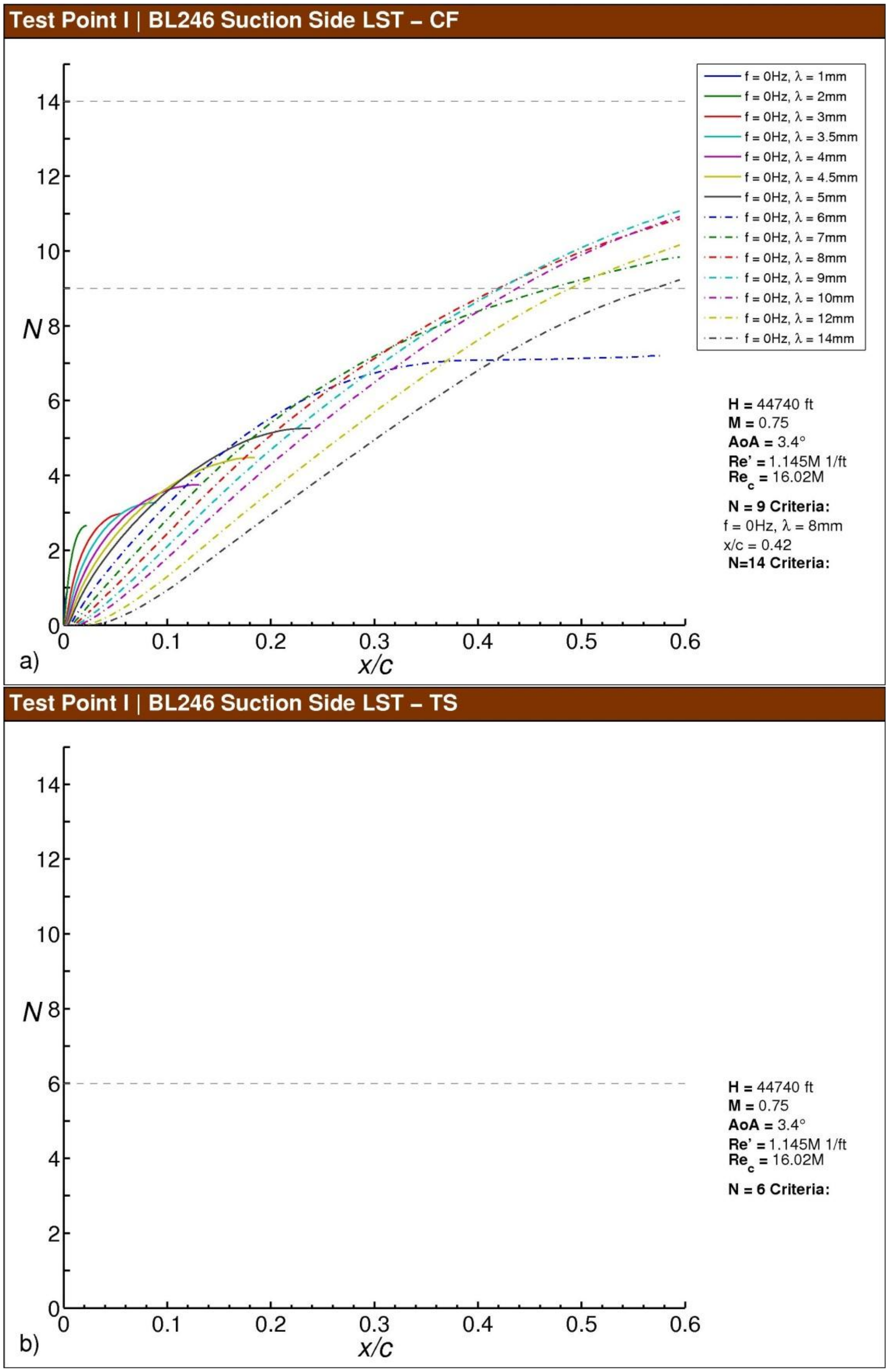


Fig. F-113 Test Point I *N*-factor results for BL246, suction side: a) stationary crossflow, b) T-S.

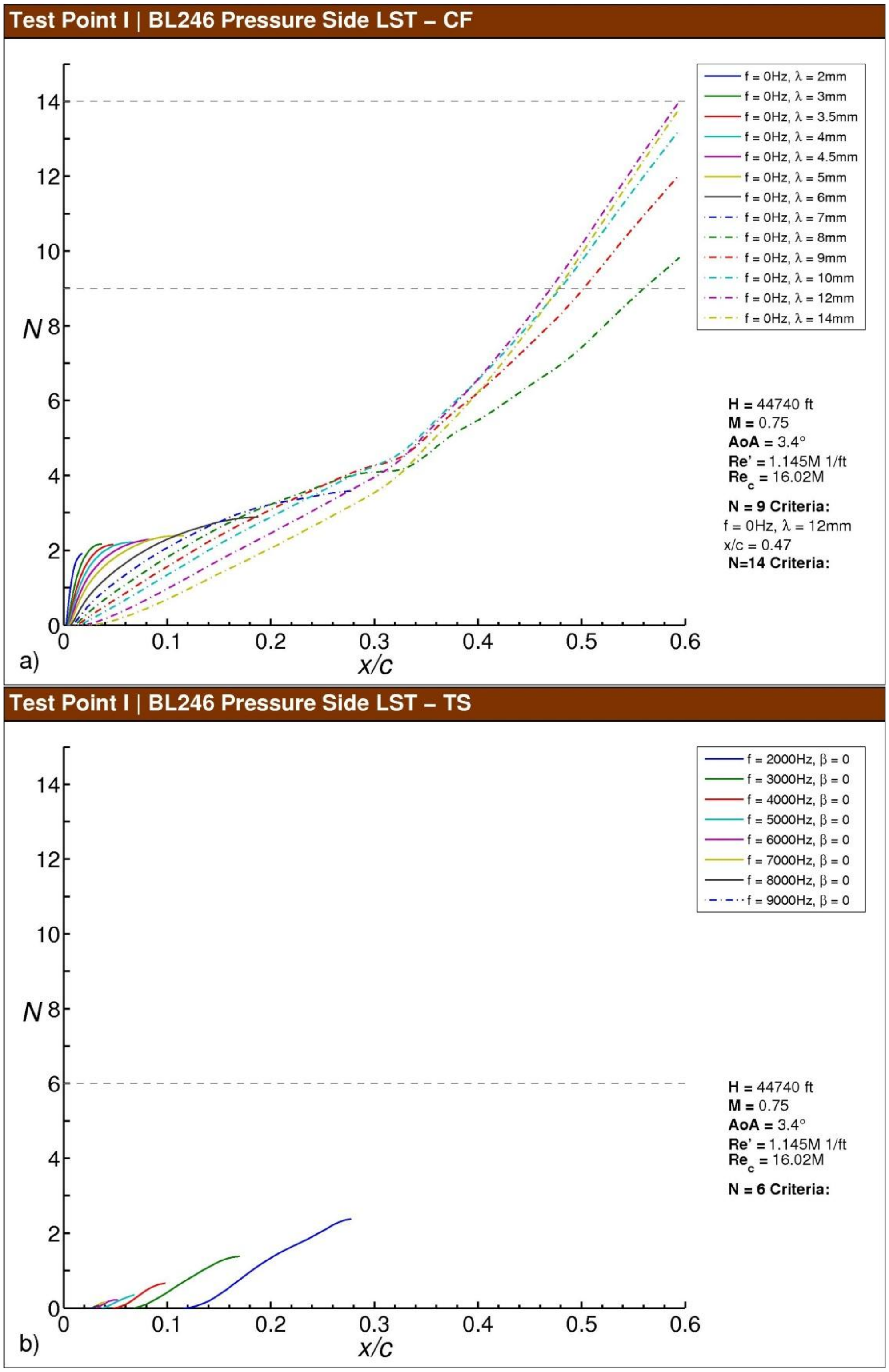


Fig. F-114 Test Point I *N*-factor results for BL246, pressure side: a) stationary crossflow, b) T-S.

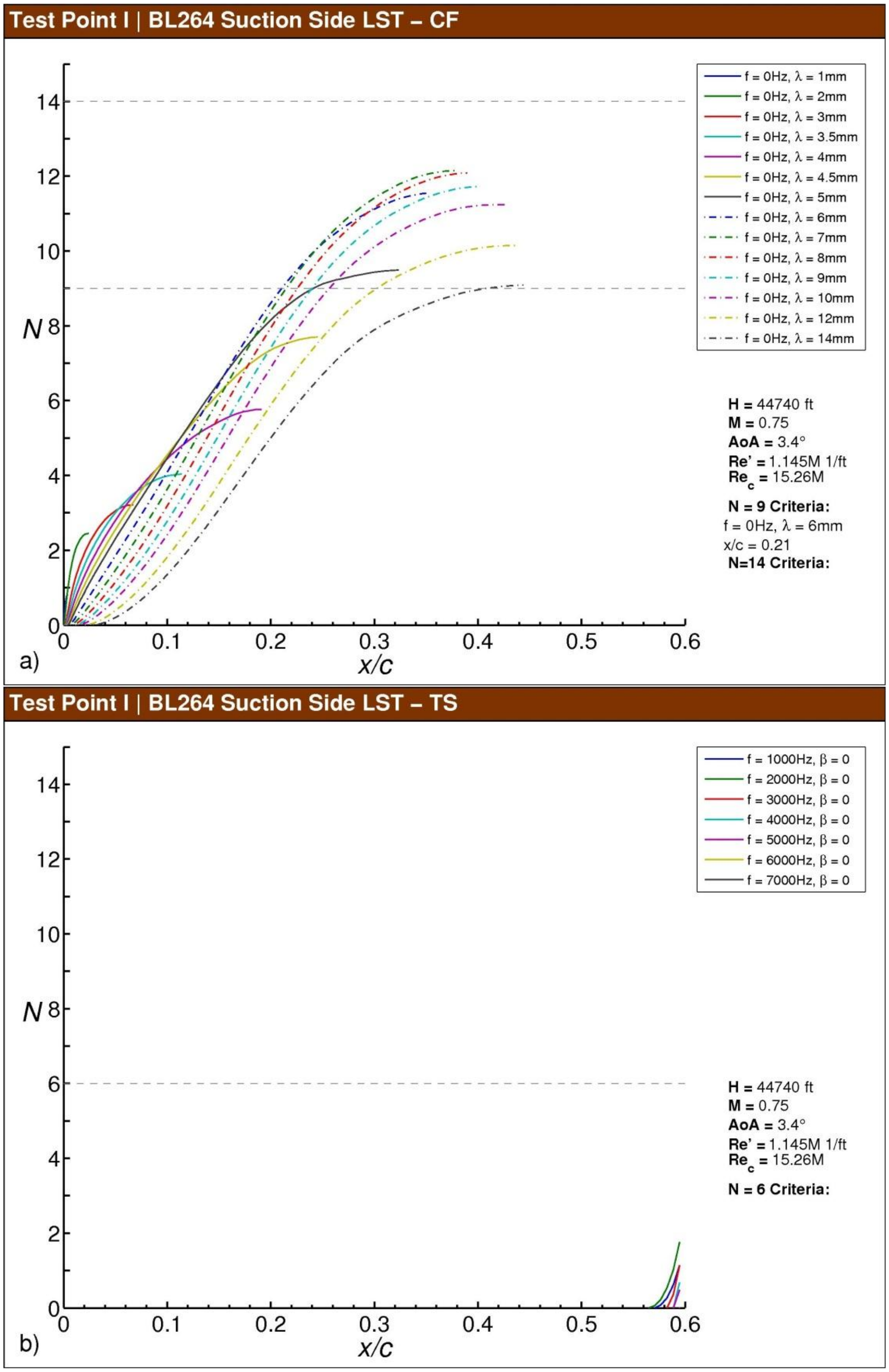


Fig. F-115 Test Point I *N*-factor results for BL264, suction side: a) stationary crossflow, b) T-S.

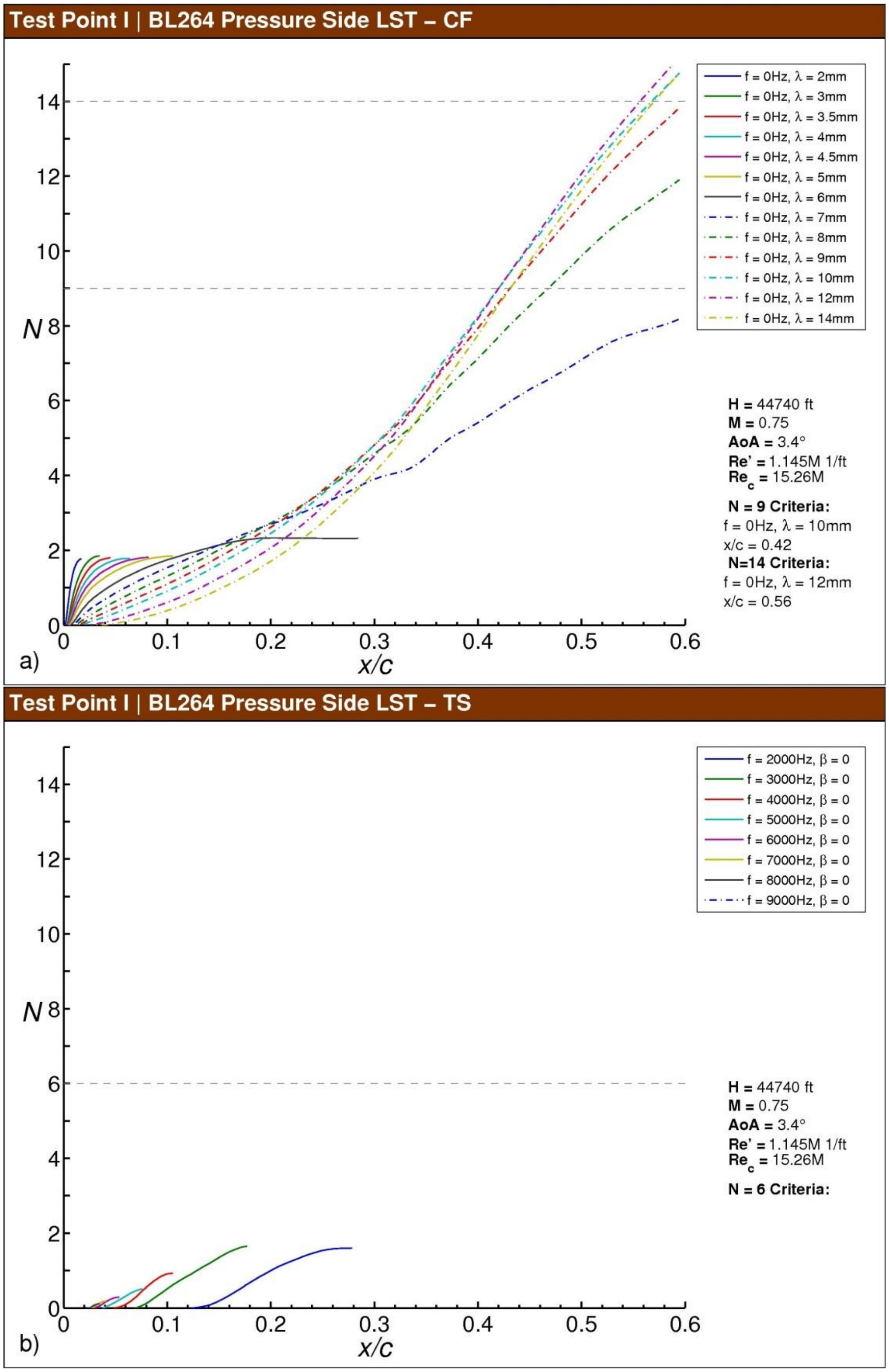
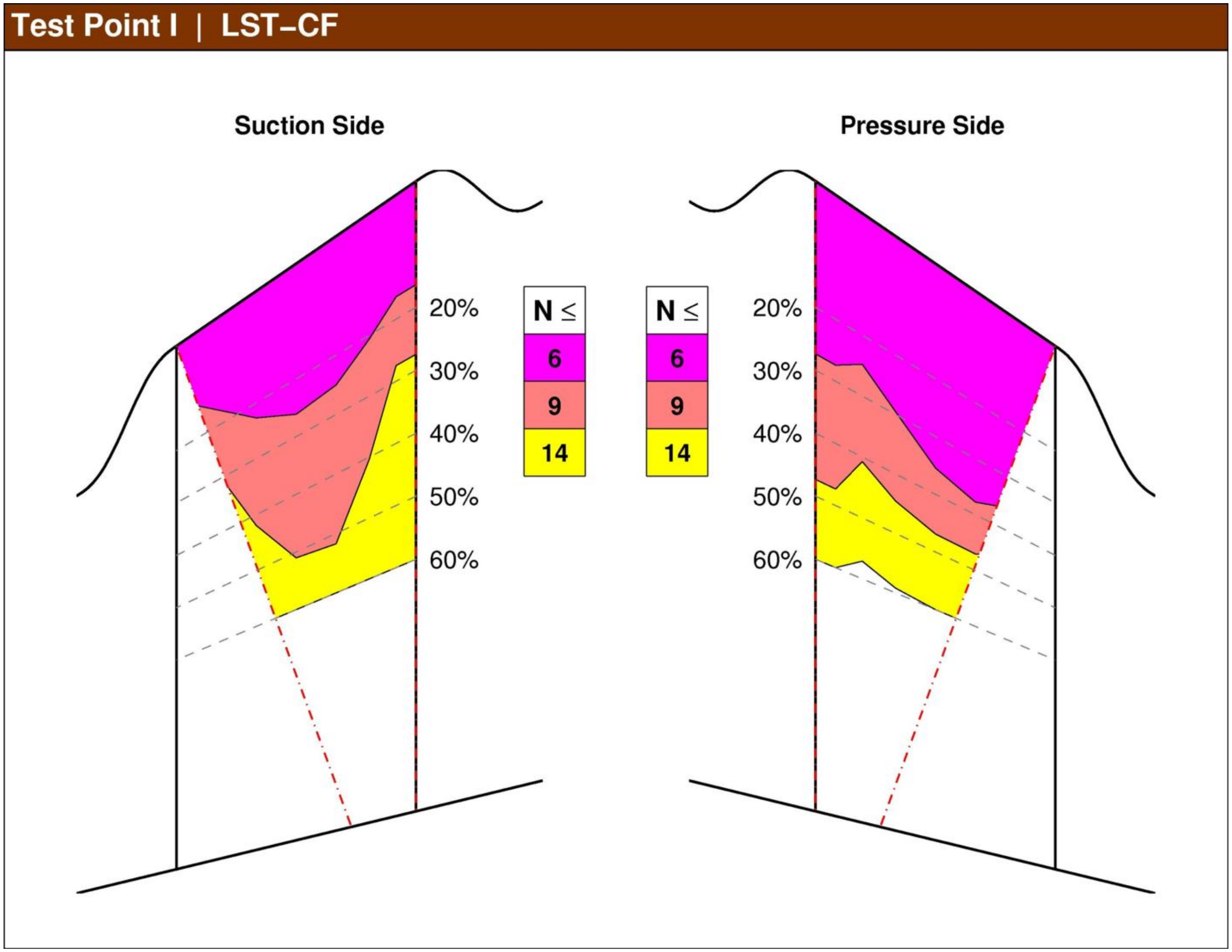


Fig. F-116 Test Point I *N*-factor results for BL264, pressure side: a) stationary crossflow, b) T-S.



Test Point I | LST–CF

Fig. F-117 Test Point I *N*-factor contours for the test section suction and pressure sides.

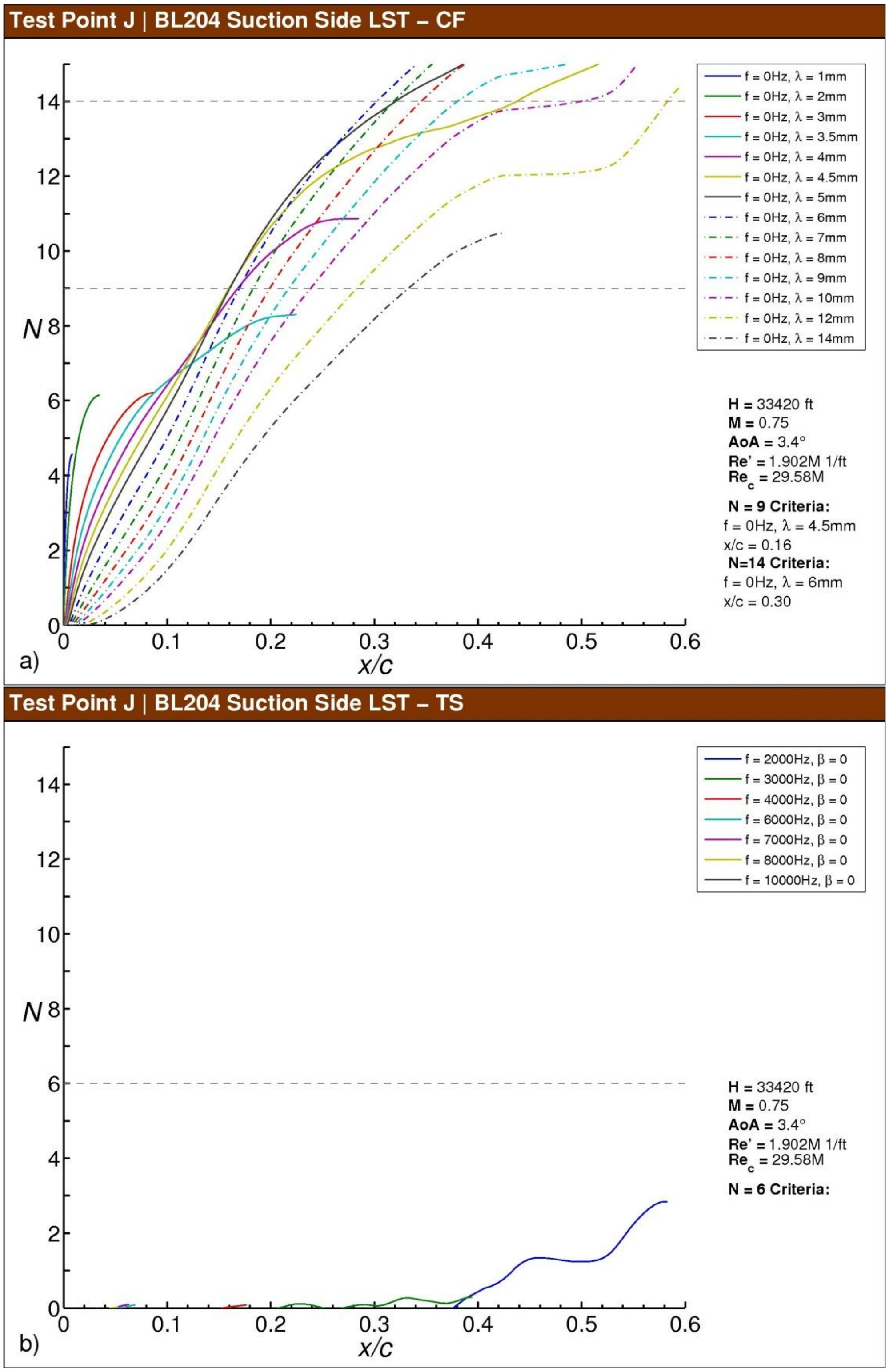


Fig. F-118 Test Point J *N*-factor results for BL204, suction side: a) stationary crossflow, b) T-S.

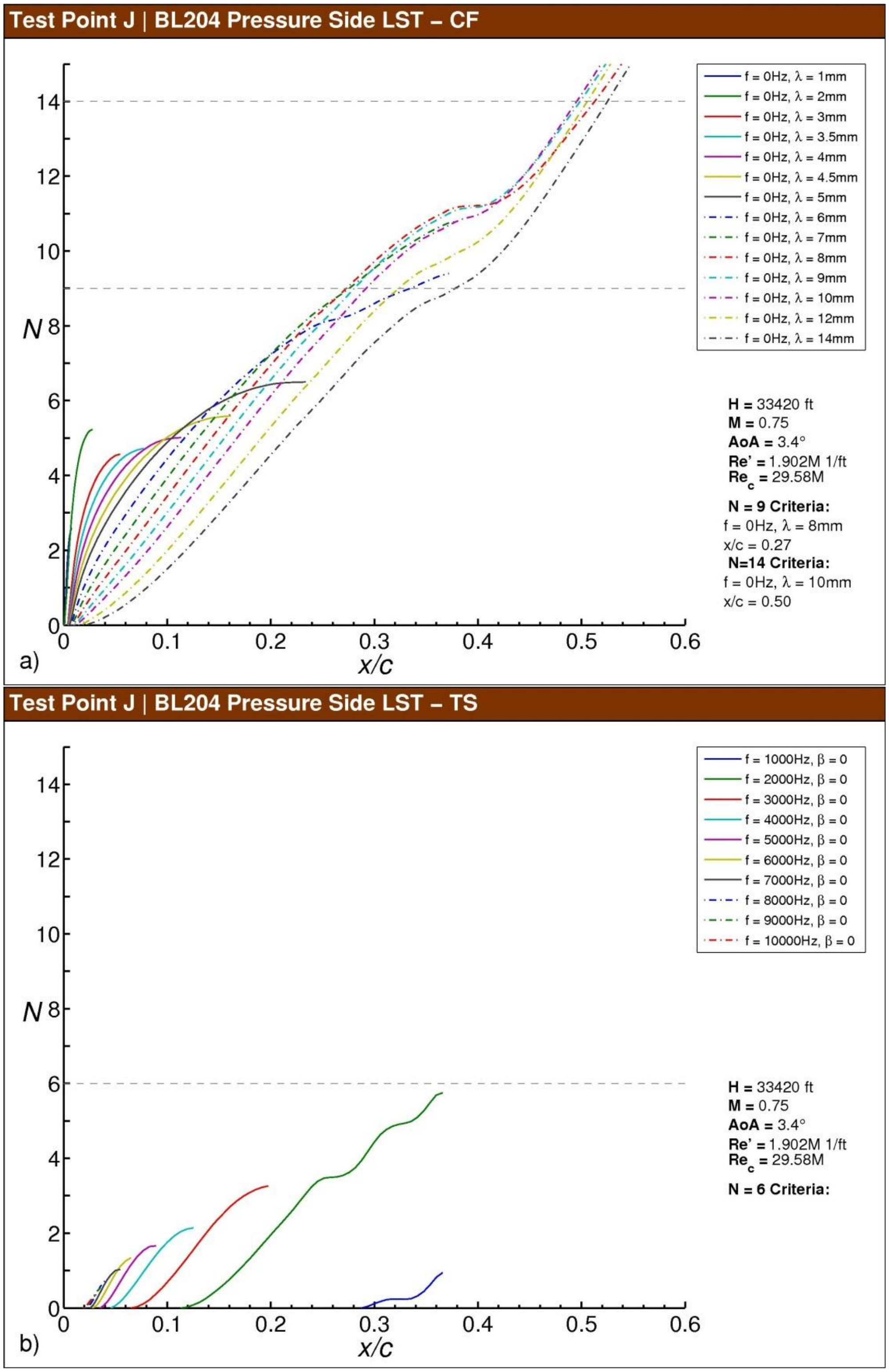


Fig. F-119 Test Point J *N*-factor results for BL204, pressure side: a) stationary crossflow, b) T-S.

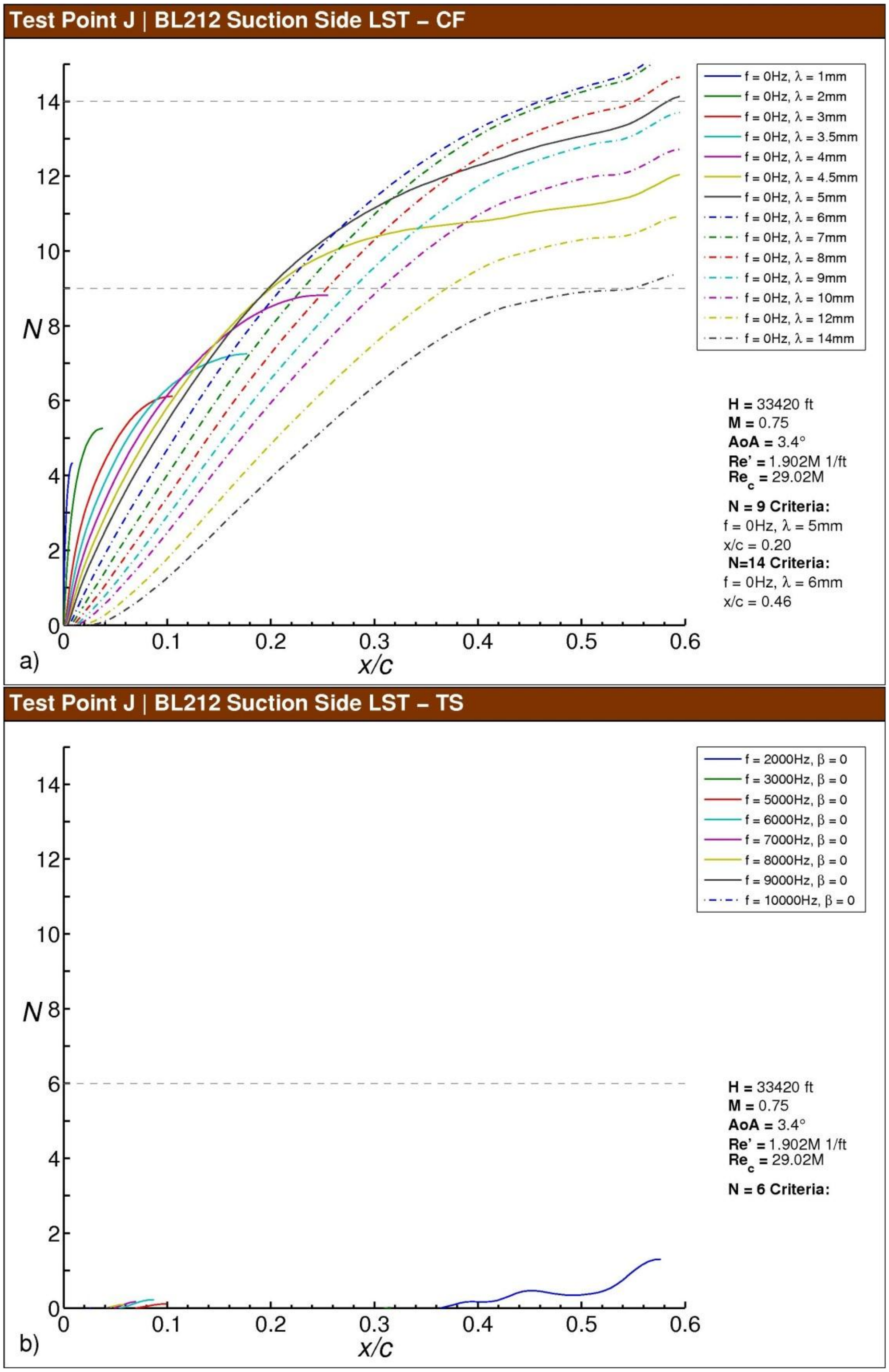


Fig. F-120 Test Point J *N*-factor results for BL212, suction side: a) stationary crossflow, b) T-S.

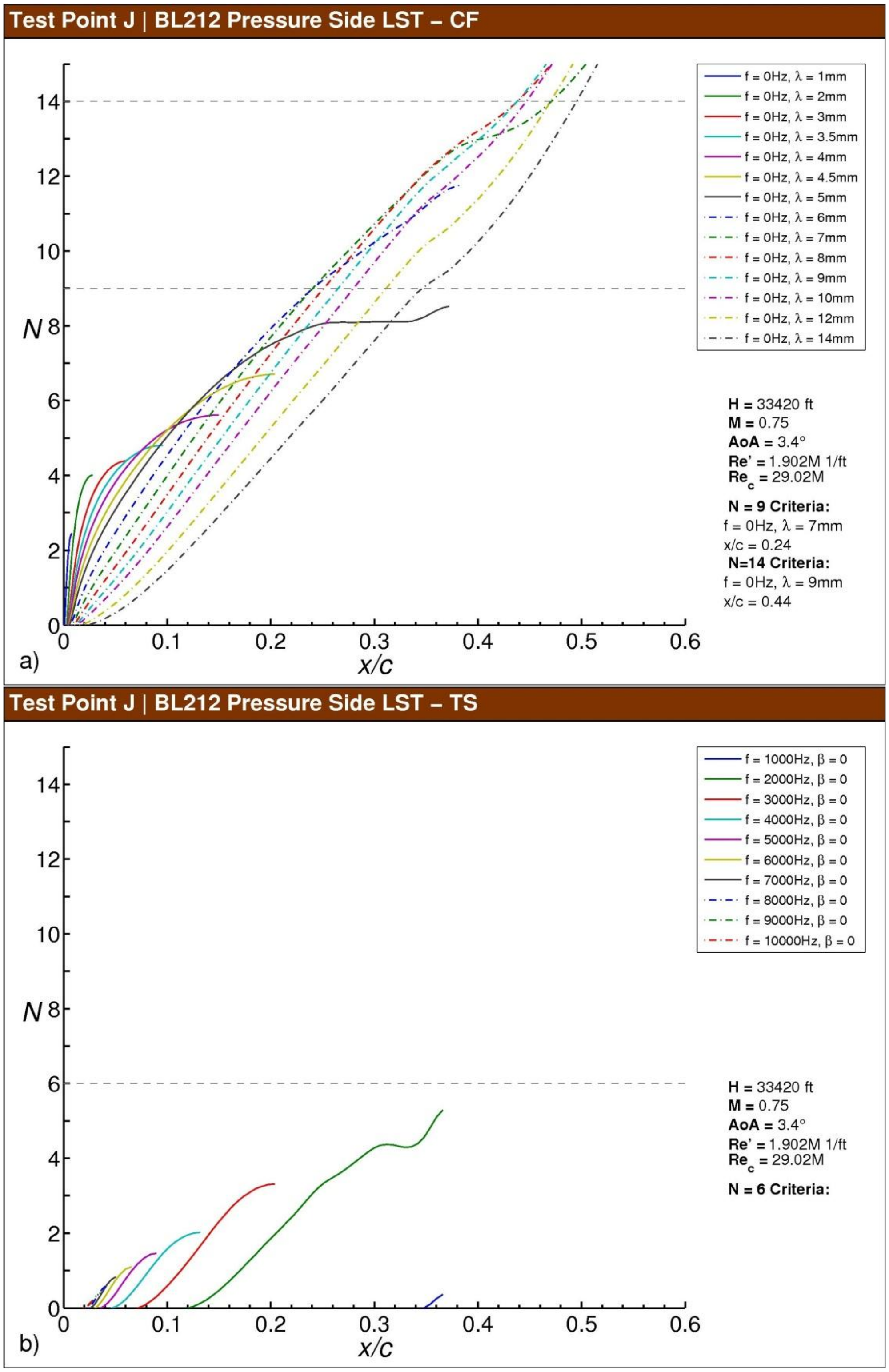


Fig. F-121 Test Point J *N*-factor results for BL212, pressure side: a) stationary crossflow, b) T-S.

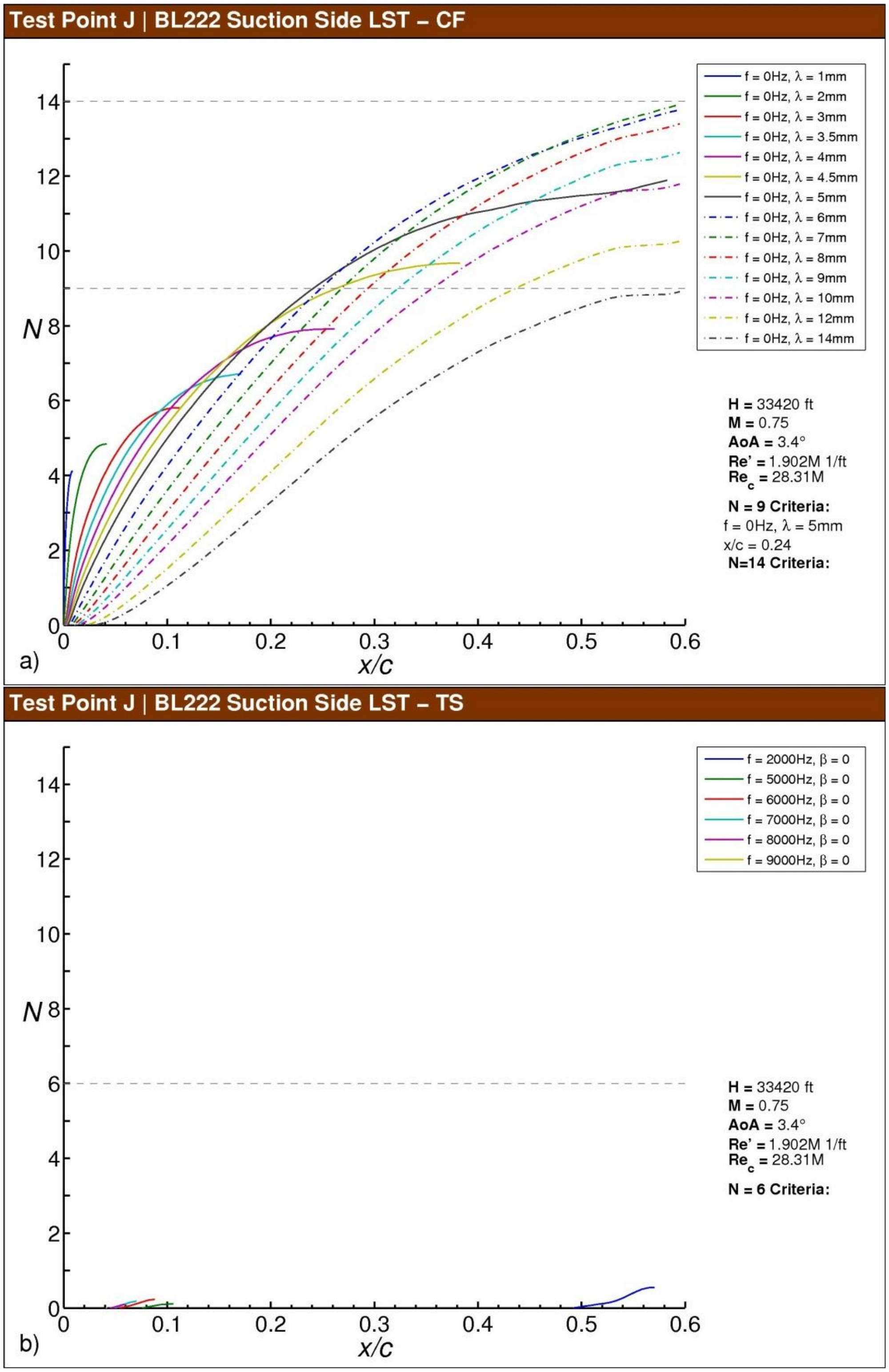


Fig. F-122 Test Point J *N*-factor results for BL222, suction side: a) stationary crossflow, b) T-S.

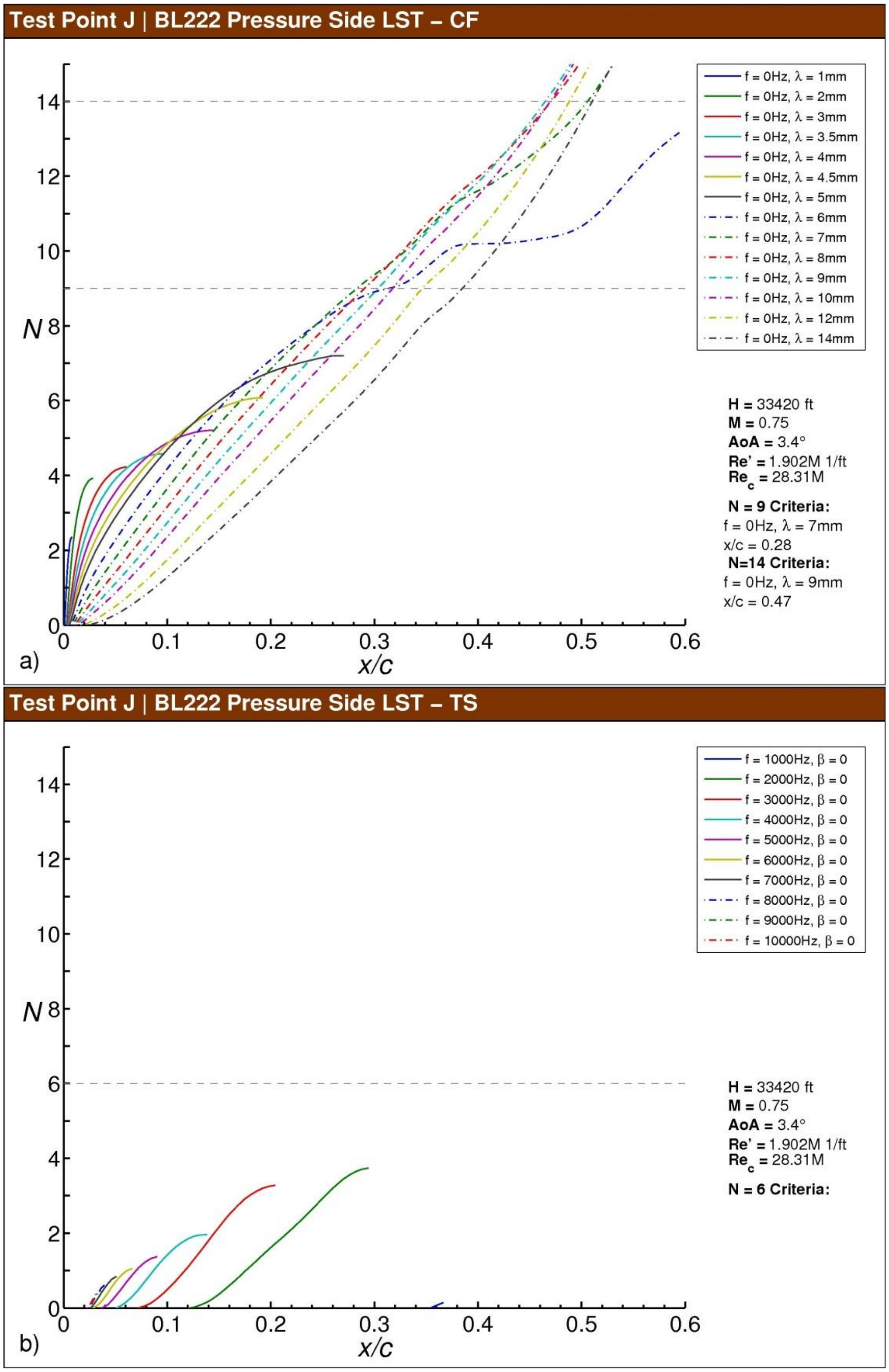


Fig. F-123 Test Point J *N*-factor results for BL222, pressure side: a) stationary crossflow, b) T-S.

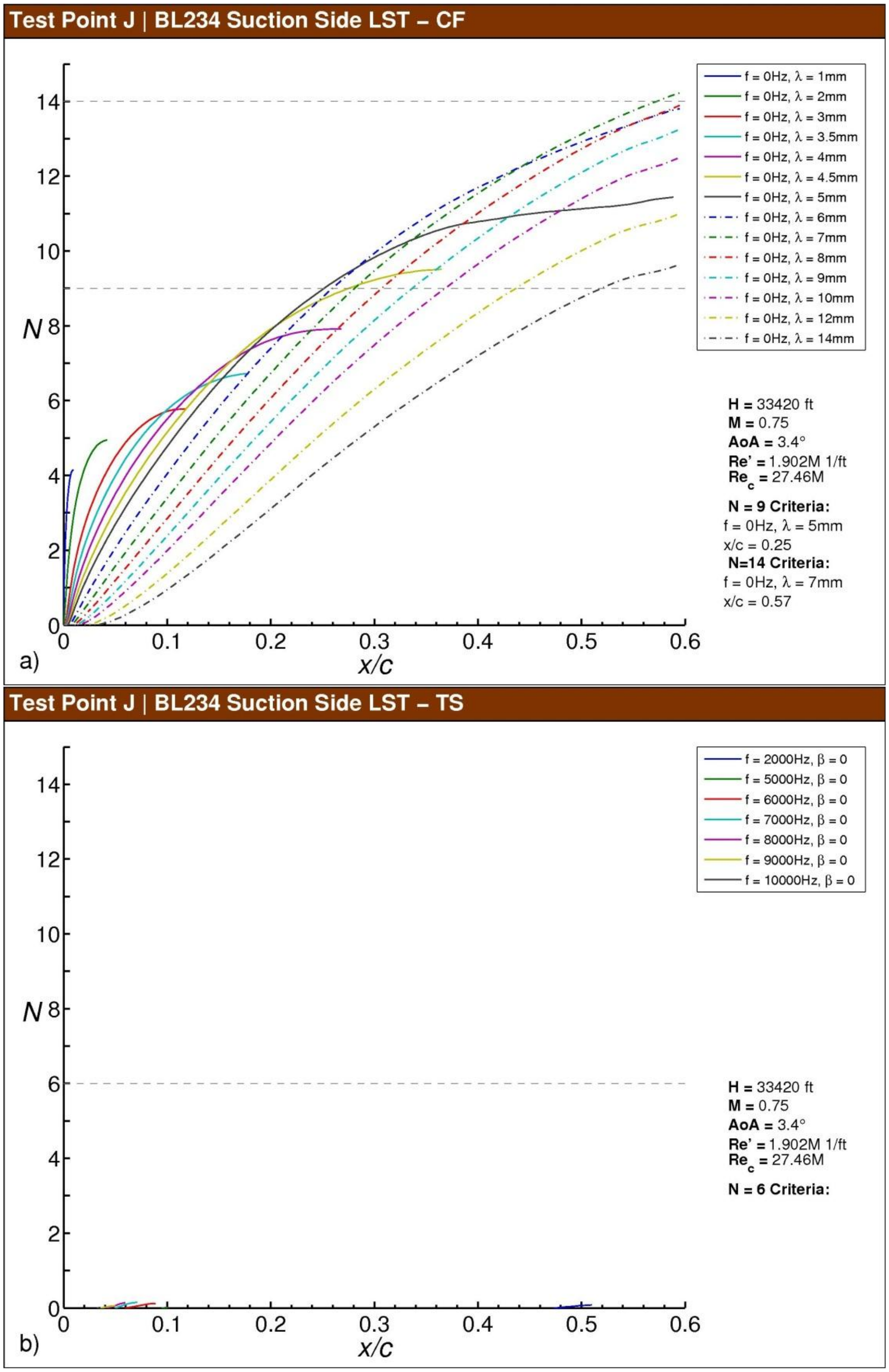


Fig. F-124 Test Point J *N*-factor results for BL234, suction side: a) stationary crossflow, b) T-S.

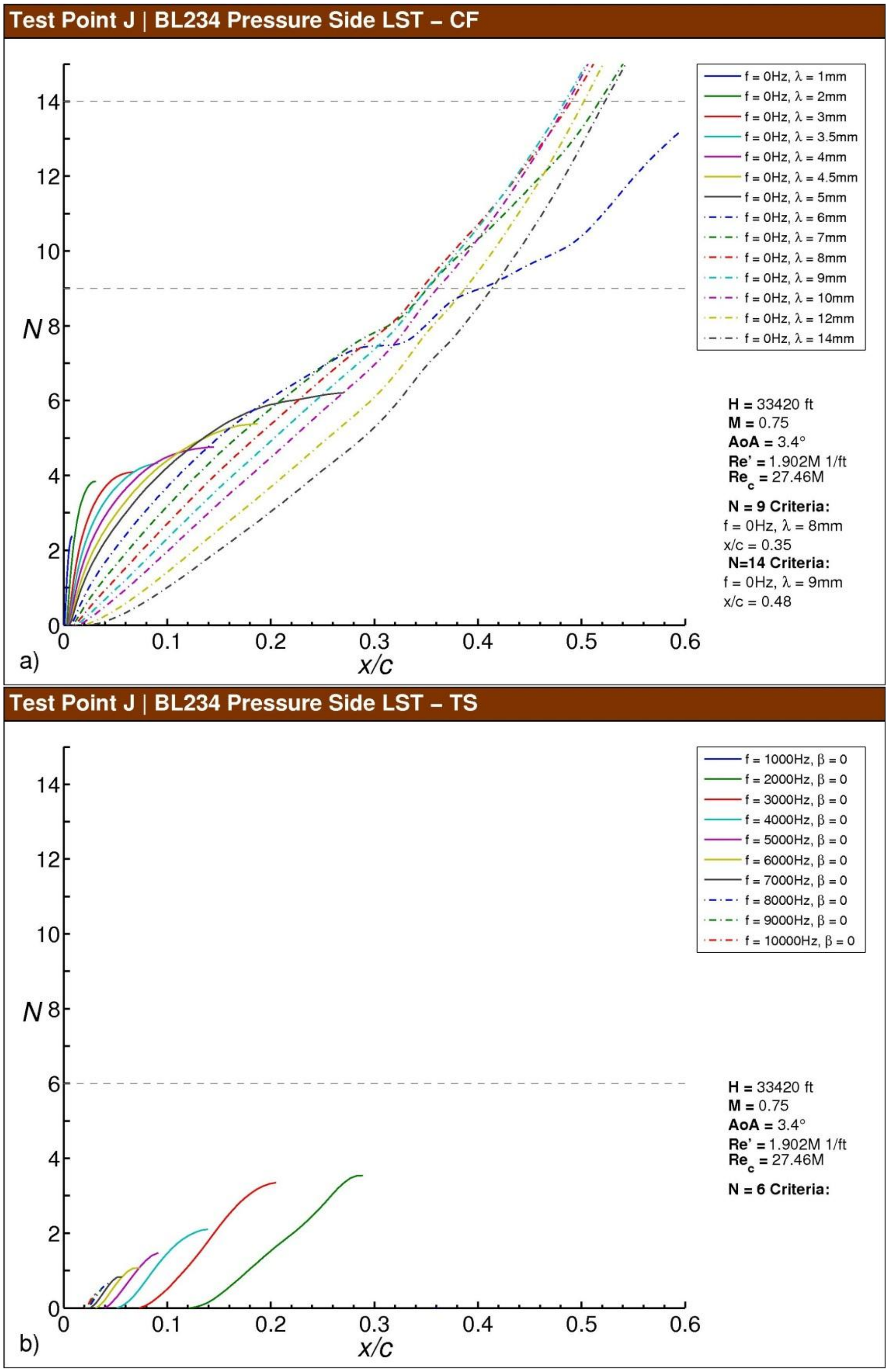


Fig. F-125 Test Point J *N*-factor results for BL234, pressure side: a) stationary crossflow, b) T-S.

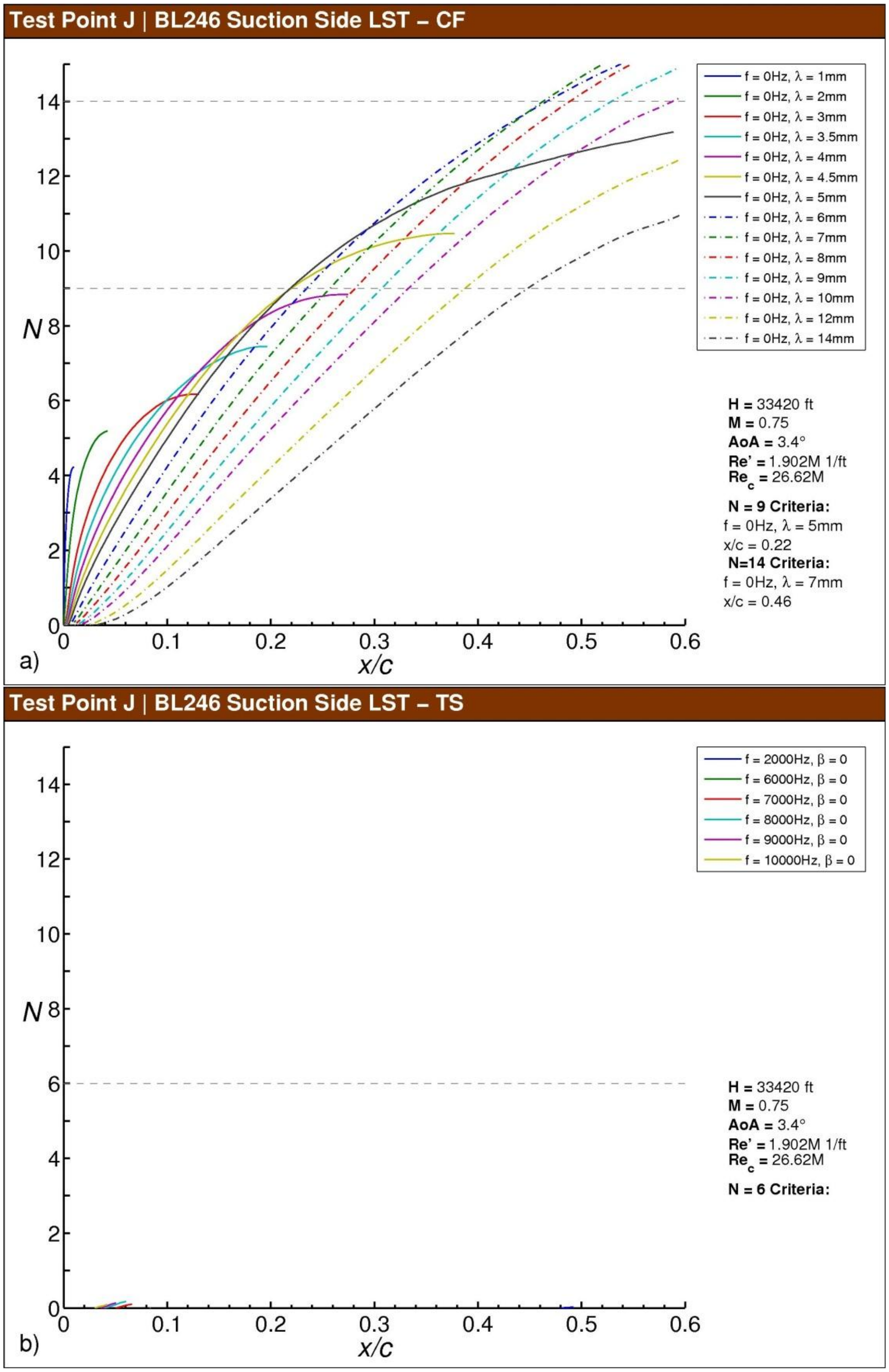


Fig. F-126 Test Point J *N*-factor results for BL246, suction side: a) stationary crossflow, b) T-S.

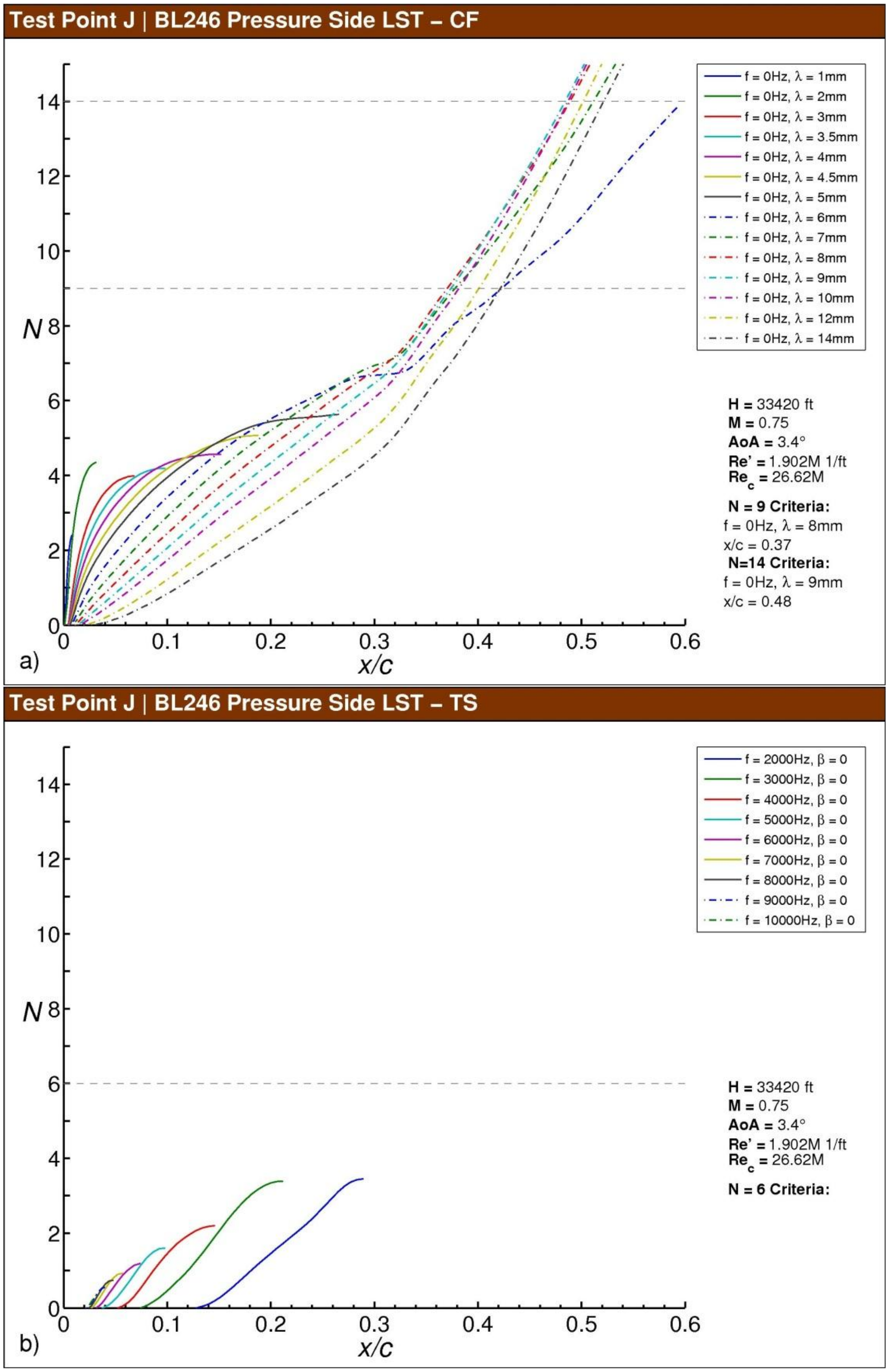


Fig. F-127 Test Point J *N*-factor results for BL246, pressure side: a) stationary crossflow, b) T-S.

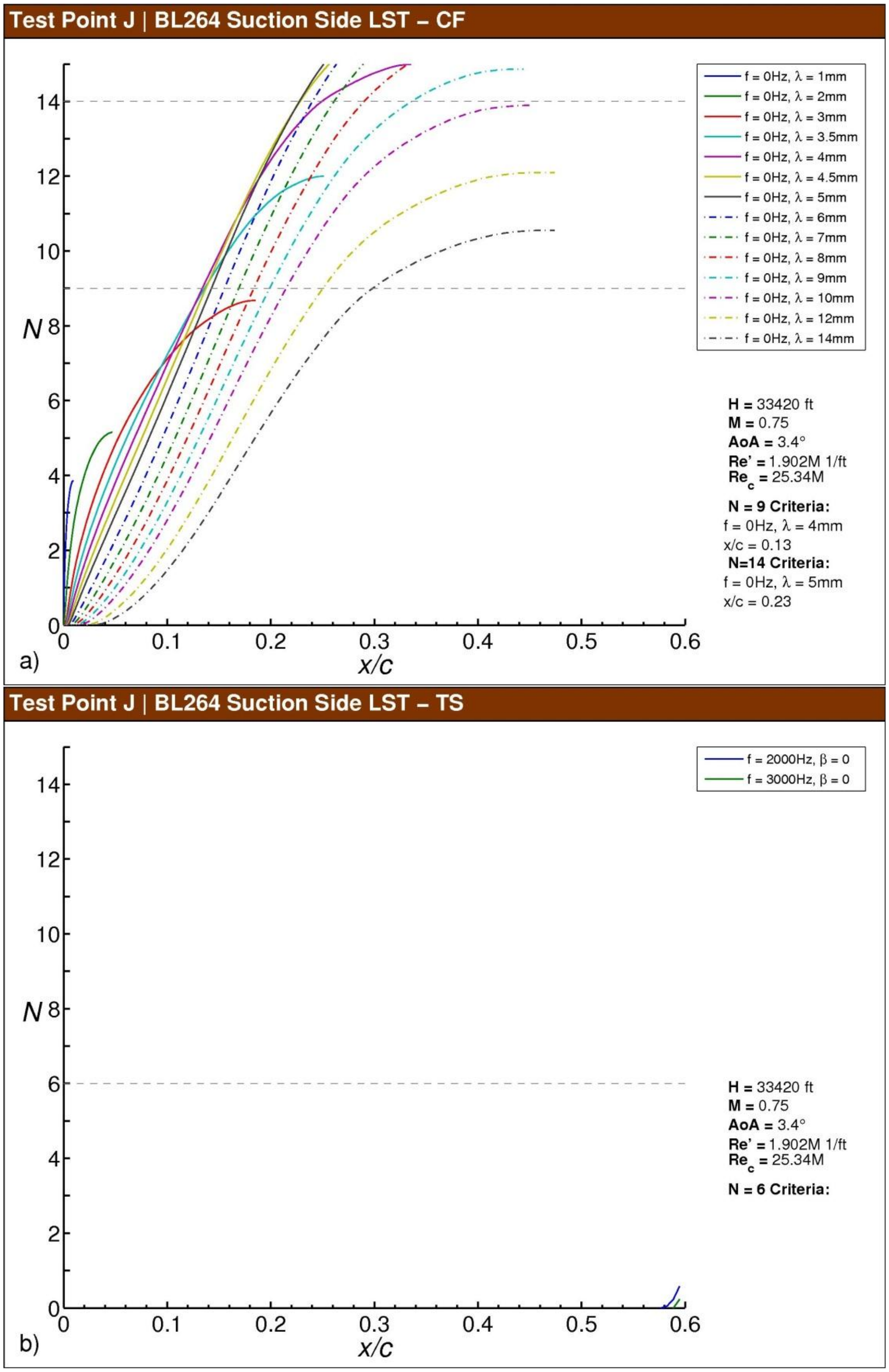


Fig. F-128 Test Point J *N*-factor results for BL264, suction side: a) stationary crossflow, b) T-S.

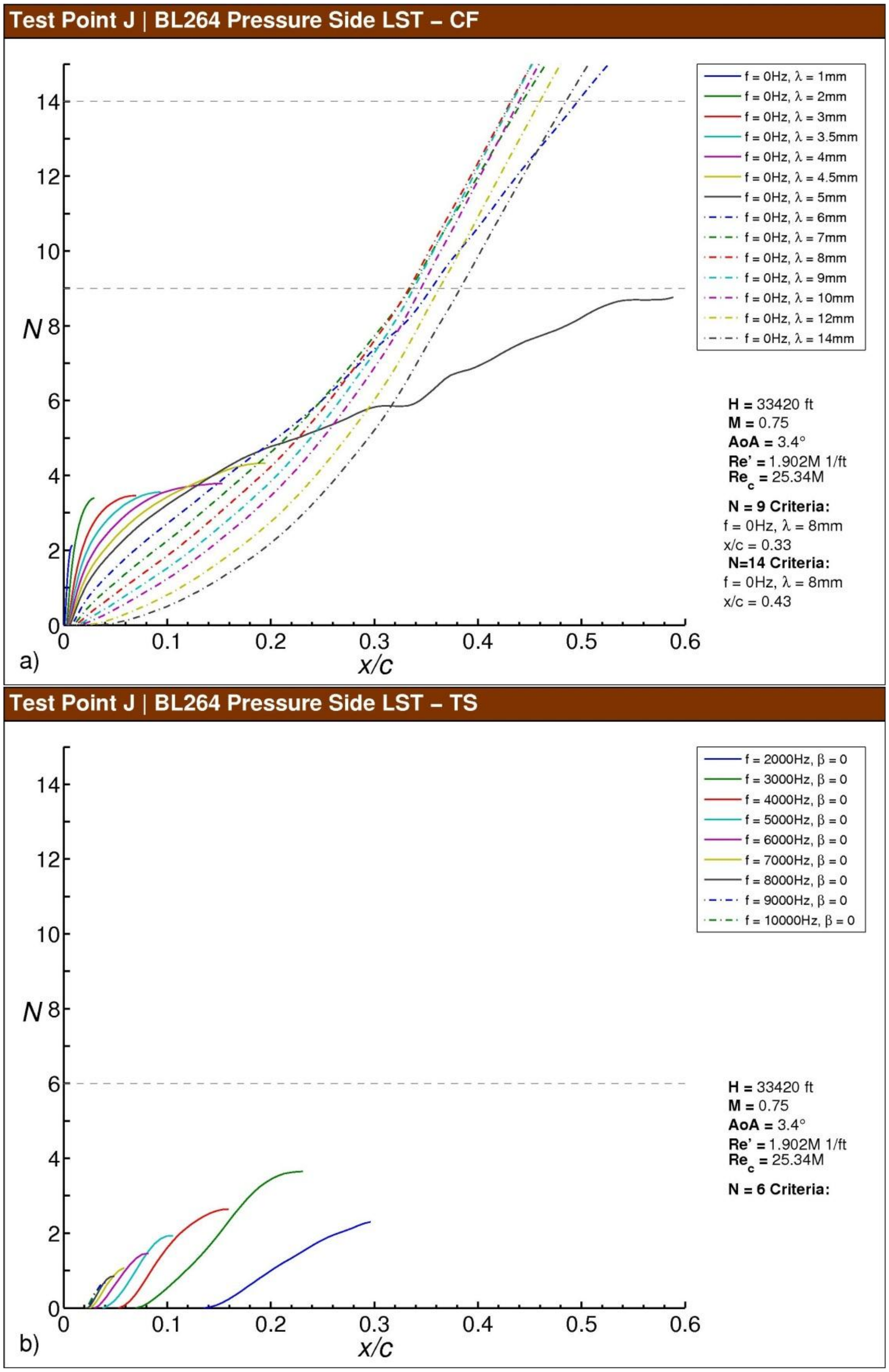


Fig. F-129 Test Point J *N*-factor results for BL264, pressure side: a) stationary crossflow, b) T-S.

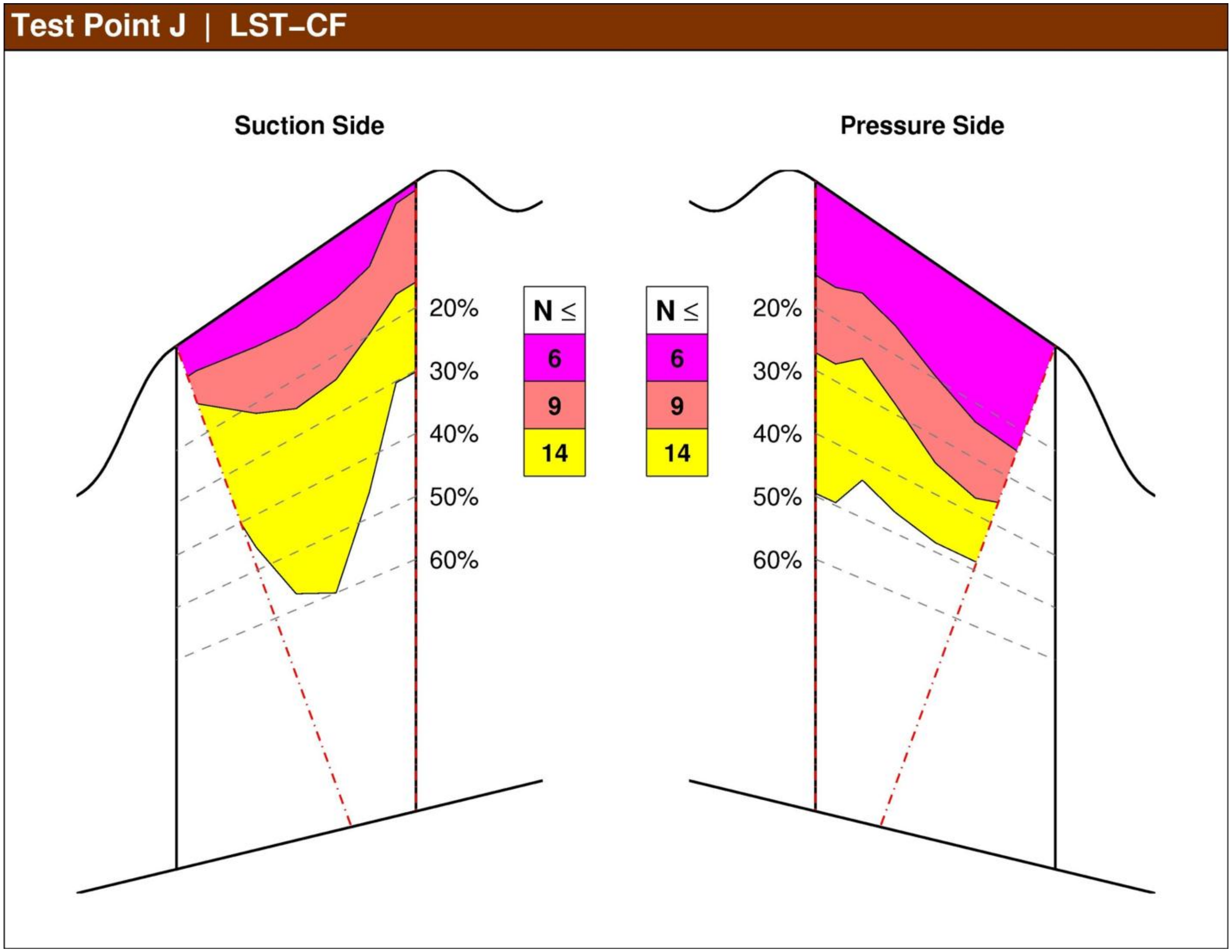


Fig. F-130 Test Point J *N*-factor contours for the test section suction and pressure sides.

VITA

Name: Matthew William Roberts

Address: Texas A&M University
Department of Aerospace Engineering
H. R. Bright Building, Room 701, Ross Street – TAMU 3141
College Station, TX 77843-3141

Email Address: matt.william.roberts@gmail.com

Education: B.S., Mechanical Engineering, Kansas State University, 2009
M.S., Aerospace Engineering, Texas A&M University, 2012