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

10.2514/6.2021-2459

VORLAX 2020: Benchmarking Examples of a Modernized Potential Flow Solver

Tyler J. Souders[1] and Timothy T. Takahashi[2]
Arizona State University, Tempe, AZ, 85287-6106

This paper demonstrates the evergreen utility of the 1970's vorte*x*-lattice method "*VORLAX*" supporting the modern design process. *VORLAX* can quickly and accurately develop aerodynamic force, moment and surface-pressure data to support rapid design synthesis. *VORLAX2020* is a dependable, lightweight, and fast computational aerodynamics tool featuring an improved and updated solver and a large number of bug fixes. This work shows how *VORLAX* can provide remarkably accurate aerodynamic results in a few seconds of run-time on a consumer-grade computer. This work also highlights the use and includes sample input files for a number of operational modes, in particular the "sandwich panel," "fusiform body" and "wake survey" options which have fallen out of general use.

Nomenclature

α	=	Angle of Attack (deg)
γ	=	Specific Heat Ratio of Air
C_D	=	Drag Coefficient
C_L	=	Lift Coefficient
C_{L_0}	=	Zero-Pitch Lifting Coefficient
Δ	=	Displacement Between Panels
Γ	=	Circulation Strength
ITRMAX	=	Maximum Number of Iterations
LAX	=	*X*-Direction Spacing Method
LAY	=	*Y*-Direction Spacing Method
M	=	Mach Number
N	=	Number of Control Points
NVOR	=	Spanwise Control Points
φ	=	leading edge sweep (deg)
R	=	Residual
RNCV	=	Chordwise Control Points
t/c	=	Thickness Normalized to Chord
x/c	=	*X*-Dimension Normalized to Chord
y/c	=	Camber Displacement Normalized to Chord

I. Introduction

VORLAX is the name of one specific computer code of a class of panel computational fluid dynamics (CFD) methods known as "vorte*x*-lattice methods." [1] A vortex lattice method solves for pressure distribution over a configuration of infinitesimally thin "flat" panels, allowing for the computation of aerodynamic quantities such as

[1] M.S. Candidate, Mechanical Engineering, School for Engineering of Matter, Transport, & Energy, P.O. Box 876106, Tempe, AZ. Student Member AIAA

[2] Professor of Practice, School for Engineering of Matter, Transport, & Energy, P.O. Box 876106, Tempe, AZ. Associate Fellow AIAA

lifting coefficients, drag coefficients, induced drag, and pressure coefficients. Using these values, one may quickly assess the viability of a proposed aircraft wing and body configuration in the High-Reynolds number flow regime.

VORLAX was developed by Lockheed-California Company in the early 1970's by Luis R. Miranda, Robert D. Elliot, and William M. Baker before being formally published in a NASA Contractor Report in 1977. [1] The original code was written in FORTRAN for use on the CDC 6600, IBM 360, and IBM 370 systems. Since then, various engineers have worked to maintain the code, allowing it to thrive and run quickly as a design software on modern machines.

This paper benchmarks a revised version of *VORLAX*, developed from the source code in the NASA Contractor Report [1]. This upgrade allows the code to handle larger, more complex models than the original (a maximum matrix of 5000x5000). It features a number of bug fixes: resolving problems with drag integration under certain gridding options, inconsistent panel geometry using the fusiform body option and run time errors using the wake survey option. We have also thoroughly revised its memory management, eliminating the need for "scratch files" while preserving its ability to run in 32-bit memory space.

We refer you to our companion paper, AIAA 2021-xxxx [2] for a more thorough description of the code changes.

In this paper, we showcase the use of *VORLAX2020* to analyze a variety of aerodynamic configurations using all of its geometrical modelling capabilities: flat panels, cambered panels, sandwich panels and fusiform bodies. We present grid-density studies, highlighting best practices. Where possible, we will compare the code output against wind tunnel data.

II. The Vorte*x*-Lattice Method

A. What is *VORLAX*?

VORLAX is a vorte*x*-lattice method that models inviscid, attached, shock-free flow over a define surface. Unlike lifting-line theory, which assumes that the bound vortices are all collinear, the vorte*x*-lattice method lets the use model any geometry as a series of discrete, finite panels. Each of these broad panels is subdivided into a grid. The code places mathematical representations of bound vortices on each grid element. A complete *VORLAX* model consists of hundreds, perhaps thousands of these bound vortex and control point pairs, each of which influences the flow over the modelled geometry. FIGURE 1 shows an example of an entire aircraft, complete with control surfaces and a fuselage, drawn up in *VORLAX*.

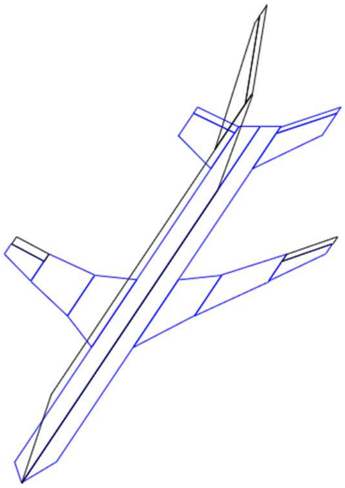


FIGURE 1 - *VORLAX* Representation of Boeing 737-300

VORLAX provides three primary modelling offerings: an infinitesimal thin panel, an infinitesimal thin panel with camber, and a "sandwich panel" – a combination of the two aforementioned methods. While this functionality existed from the inception, we realize that the more advanced camber and "sandwich panel" methods have fallen into disuse, despite proving very useful, in practice.

A large advantage of *VORLAX* relative to more complex methods is that the interfacing system with *VORLAX* is greatly simplified relative to other software suites, thereby existing as a system that is not only faster and much simpler, but also completely avoids reliance on any proprietary mesh generators of postprocessing utilities. *VORLAX* operates entirely on the basis of standard UTF-8 encoded text files. The user will generate file inputs, complete with geometric and flight configuration definitions, and will receive two UTF-8 output files. Of these files, one returns basic results regarding quantities such as the lift, drag, and stability derivatives, while the other offers precise pressure distribution information at each control point of the model.

Fortunately, it is easy to manipulate these files using other more modern coding platforms such as Visual Basic for Applications (VBA), MATLAB and Python. VBA is a great utility for this process because it is included as an underlying utility in the Microsoft Office Suite. Thus, it is possible to automate the reading and writing of these data files using a method that an employer or educator almost certainly already has access to. With VBA in conjunction with Microsoft Excel, it is simple to construct a "wrapper" for *VORLAX* that includes a geometric preview of the model, an input file builder, code execution automation, an output file parser and a plotting suite. We show here more powerful visualization using Python.

Thus, *VORLAX* exists as a highly accessible utility for the analysis of aerodynamics. The program is incredibly accessible, and in the subsequent chapters the theory and operation will be detailed, as well as its applications.

B. *VORLAX* Limitations

VORLAX models "compressibility corrected," inviscid, attached flows. It uses linear algebra techniques to determine the necessary circulation strength, Γ, of each bound vortex in order to impose a zero-flux, Neumann type boundary condition at each control point. However, this becomes computationally limiting, as this means the system of equations involves a dense, non-diagonally dominant, non-positive definite matrix of size [Total Points, Total Points] that must be solved. Typically, with modern finite-difference based computational fluids solvers, memory management is largely simplified by the usage of sparse, diagonall*y*-dominant matrices, of which the verte*x*-lattice method is inherently **unable** to take advantage. Nonetheless, with an appropriate solver *VORLAX* remains so efficient overall that its throughput and quality exceeds volume-grid CFD methods for many applications.

While the assumption that the flow is inviscid is undeniably inaccurate, this may be remedied by running a flat-plate drag buildup program such as *EDET* [3] in conjunction with *VORLAX.* These drag buildup methods also exist as efficient FORTRAN codes, so there is little overhead burden to such an addition, and the task of generating geometries in one of the programs based on the input file of another is just as simple and can be easily automated

VORLAX is a generalized subsonic and supersonic panel method code. To calculate the local pressure distribution, *VORLAX* operates by taking the user-defined panel shapes and breaking them into a continuous series of collinear control points. *VORLAX* superimposing normal-wash and axial-wash components on top of the freestream velocity., [1] At each of these points, the code will solve for a zero-mass-flux boundary condition (unless otherwise specified) by solving for the appropriate circulation strength at each of the discrete vortices distributed at each element. This is done using a small-perturbation approach, which allows the code to account for compressibility effects for the free-stream Mach number. By knowing the circulation strength for each of the vortices about the airframe, the calculation of the pressure coefficient at each control point becomes simple by means of classical potential flow theory. [4] Using the well-known Prandtl-Glauert correction: $\beta=1-M^2$, *VORLAX* produces local pressures as well as integrated forces and moments that are significantly more accurate than those obtained via a blind application of the Prandtl-Glauert scaling factor on a low-speed solution. [5] Later in this paper, we demonstrate how well the "sandwich panel" representation works.

While *VORLAX* may handle purely sub- or super-sonic flow conditions but inherently lacks the ability to resolve mid-span shock effects on a transonic surface. For example, we can use *VORLAX* to architect complex 3D aerodynamic shaping when we model the geometry using the "sandwich" and "fusiform" elements. Because such a model computes local surface pressures (*Cp*) rather than net pressures (Δ*Cp*), we can then post-process the results to inspect any given solution to determine if we are using *VORLAX* outside its "trust zone." We can verify whether a shock will form by inspecting the pressure field to see if the local surface pressure falls below a critical pressure coefficient. We prefer the swept-surface *Cp** prediction method from Küchemann [6]

$$Cp^* = \frac{2}{\gamma M_\infty^2}\left\{\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma}{\gamma-1}}\left(1+\frac{\gamma-1}{2}M_\infty^2(\cos\varphi)^2\right)^{\frac{\gamma}{\gamma-1}}-1\right\} \tag{1}$$

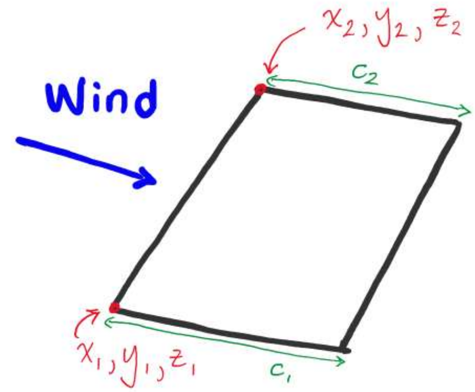


FIGURE 2 – Basic Quadrilateral Panel

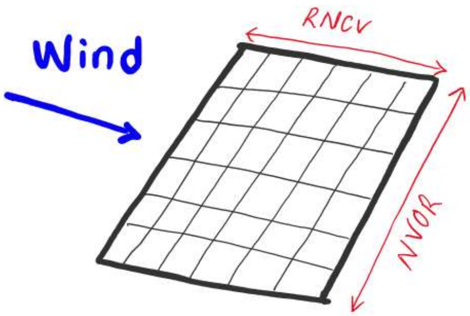


FIGURE 3 – Discretized Panel

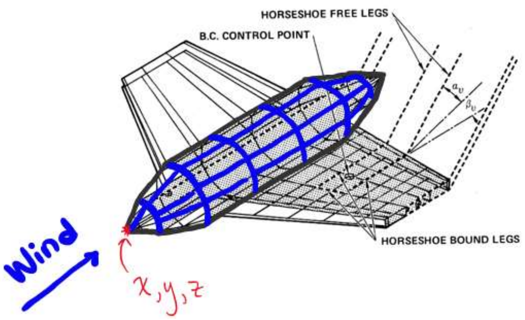


FIGURE 4 – Fusiform Body

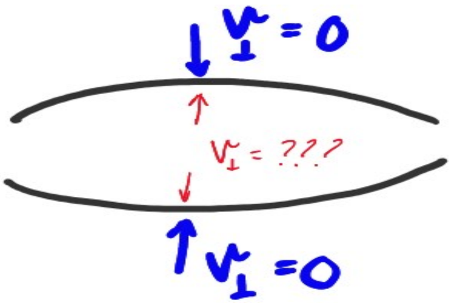


FIGURE 5 – Sandwich Panel Schematic

III. *VORLAX* Operating Modes

VORLAX defines basic geometry as a collection of quadrilateral panels governed by the *x*, *y* and *z* position and the wind referenced local chord at an "inboard" and "outboard" station; see FIGURE 2.

The code subdivides each panel into a grid. The user may control the grid density on a panel-b*y*-panel basis. NVOR defines the number of grid points in the spanwise direction; RNCV defines the number of grid points in the chordwise direction; see FIGURE 3. We will show later in this paper that more points do not necessarily correlate to better quality results.

The user may define the fuselage as either a flat panel or as a "fusiform body." If the user chooses the "fusiform body" option, the shape will be defined by an *x*, *y*, *z* and leading-edge wind-referenced length of the body, as well as a longitudinal body cross-sectional shape given by a radius distribution; see FIGURE 4.

For the quadrilateral panels, *VORLAX* gives the end user three ways to panel geometry:

- Infinitesimall*y*-thin, flat panel
- Infinitesimall*y*-thin, cambered panel
- Two infinitesimall*y*-thin panels "sandwiched" together to make a panel with finite thickness

If the user does not choose to define a camber line, the quadrilateral panel is assumed to be flat. The user can specify camber magnitudes given at several stations along the wind-axis length of the wing panel. *VORLAX* interpolates the coordinates to draw a shape.

Each *VORLAX* panel has a control flag, *ITS*, which can signify whether the flow is not to permeate through either side of the panel. *ITS*=0 represents the "impermeable" option typically used for flat or cambered panels. *ITS*=+1 or -1 defines a "semi-impermeable" panel. A "sandwich" of two semi-impermeable cambered panels mimics the behavior of a "thick" panel. We use these to model real wings and analyze the pressure distributions over the surface. For a sandwich panel, the user commands *VORLAX* to set the normal velocity on the outside surfaces to zero (*ITS*= +1 for upper surface, *ITS*= -1 for lower surface); we don't really know (or care) what the internal velocity would be. See FIGURE 5.

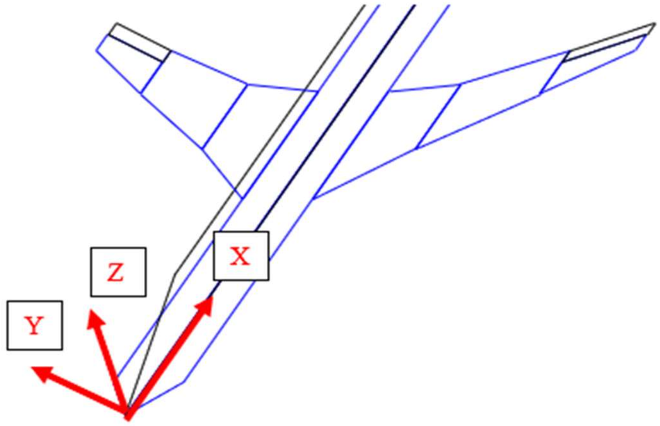
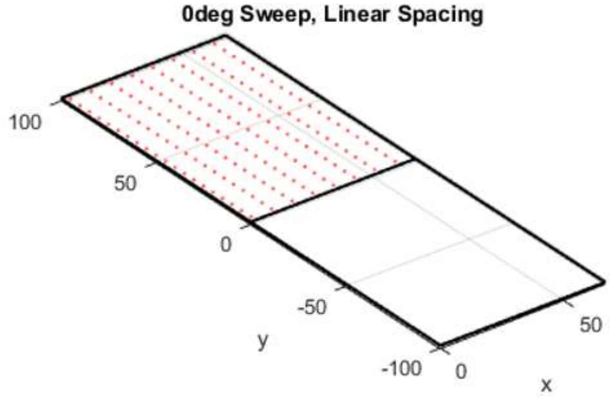


FIGURE 6 – Coordinate Reference



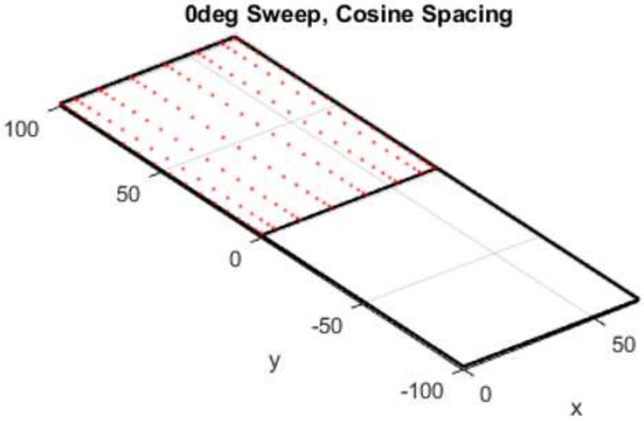


FIGURE 7 – Self Gridding Options – linear vs cosine spacing

B. Details of the Flat Panel Mode

When the user selects *VORLAX* "flat panels," the code resolves zero-mass flux condition on each side of the infinitesimally thin panels which comprise the aerodynamic structure of interest. In *VORLAX2020*, the user is at liberty to define up to 20 panels, with mirrored panels (i.e. those which are symmetric about the centerline) counted only once. Thus, a user may define a full aircraft shape, including the body (both horizontal and vertical), main wing and ailerons, horizontal tail and elevator, and vertical tail and rudder. Furthermore, the user is not restricted to a single panel for any of the surfaces and may find it preferential to build the main wing as a series of 3-4 individual panels when analyzing macroscopic aerodynamic quantities of the aircraft.

The *VORLAX* geometric coordinate frame is consistent with modern CAD standards. Positive x represents distances further downstream from the origin; increasing Frame Station. Positive y represents a lateral displacement from the line of symmetry; increasing Butt Line. Positive z represents a vertical displacement above the origin; increasing Water Line.

VORLAX self-gridding may employ either linear or cosine spacing in both the longitudinal and transverse reference frames. FIGURE 7 shows linear/linear and cosine/cosine gridding options applied to a simple "Hershe*y*-bar" wing planform.

To simulate the drag characteristics of thick, blunt-leading-edge wings, *VORLAX* has built-in functionality to apply an analytical "leading edge suction" to its drag data. *VORLAX* computes leading-edge suction using Lan's method. [7] One condition of this method which was formally thought to be a bug is that the chordwise spacing along the panel must be done using the cosine method. In prior compiles, the user could specify linear grid spacing in conjunction with the analytical leading-edge-suction correction; this permits *VORLAX* to output erroneous drag values. Thus, to avoid future confusion and inaccurate results, *VORLAX2020* error traps and rejects runs if the user specifies a nonzero SPC value with linear chordwise spacing.

C. Details of Cambered Panel Mode

The user may wish to define the main lifting structures as a series of smaller panels because it affords the ability to define unique characteristics at each point. *VORLAX* may include camber effects as a series of discrete finite-element panels which are set at a certain incidence, thereby altering the direction of the unit normal when used in the zero-mass-flux boundary condition. Thus, the code can capture effects such as the zero-pitch lifting coefficient and its associated drag.

To define a camber line, the user will input a series of up to 90 chord stations and displacement values normalized to the chord length, *x/c*. FIGURE 8 shows a representative input camber line along the two-dimensional airfoil "cut" representing a panel edge. A cambered *VORLAX* panel, therefore will have two defining geometry lines; one along the inboard and one along the outboard element.

Internally, *VORLAX* uses the "control points" to determine the appropriate panel slope for each grid element using bi-linear interpolation; see FIGURE 9.

The *VORLAX* solution, thus incorporates the aggregate slope, and panel normal directions implied by the geometry. Thus, the entire structure is "seen" by the program as a group of normal vectors in an arbitrary Cartesian 3-D space. Because the potential flow solver satisfies a zero-mass flux boundary condition, changing the direction of the normal vector will directly affect the coefficient resulting from the inner product of the velocity flow field with the normal vector. Thus, a change which initially seems rather basic leads to accurate representation of camber effects on C_{L_0}. A cambered wing, for example, will develop non-zero pitching moments at zero lift.

D. Details of the Sandwich Panel Mode

VORLAX may capture thickness effects using "sandwich" panel representations. Much like in the case of the cambered panels, *VORLAX* represents these two values as a series of discrete, angled panels, to which the code solves for the zero-normal flow condition. To build a sandwich, the user locates two cambered panels in close proximity to one another and enforcing the zero-mass flux boundary condition only on the surfaces wetted to external flow; see FIGURE 10.

Miranda [1] suggests good results having the panels spaced at a distance of:

$$\Delta \approx \frac{2}{3}\cdot\left(\frac{t}{c}\right)_{\max}(y)$$

is sufficient for accurate panel loading.

.

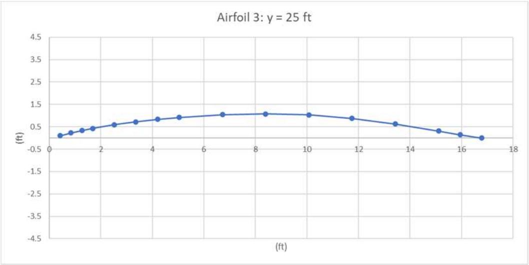


FIGURE 8 – Definition of a Camber Line

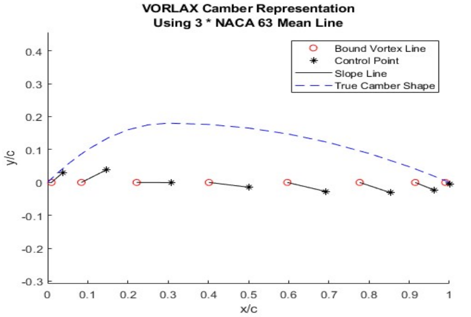


FIGURE 9 – Cambered Representation at the control points.

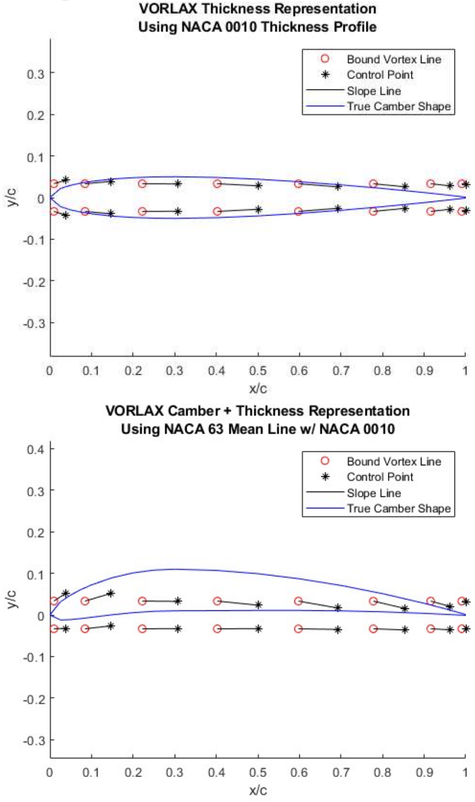


FIGURE 10 – NPP=0 - Sandwich Panel Representation

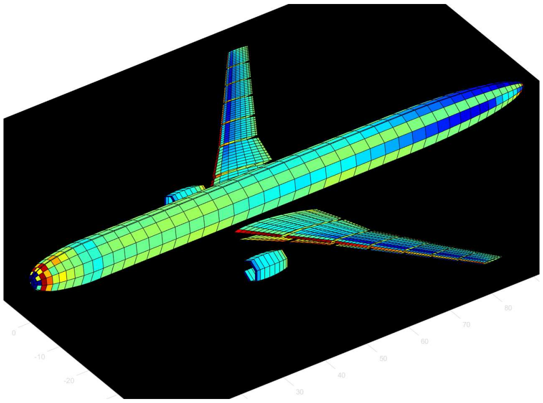


FIGURE 11 – Use of the Sandwich Panel and Fusiform Body Option to Model a Fuselage and Nacelles.

E. Details of the Fusiform Body Mode

VORLAX may represent cylindrical "fusiform" configurations and include them in the overall computation of the pressure distribution over the aircraft. These are particularly useful when computing wing-body interactions or nacelle interference. To generate this cylinder, the code reads user-input station/area pairs and draws a ring of flat panels at those coordinates. This ring contains bound vortex and control point pairs, much like the other flat panels. However, in the fusiform case, there is a generated semi-circular ring of trailing vortices. By representing the vorte*x*-control point pairs in this fashion, one may obtain higher fidelity pressure distributions on the fuselage of the aircraft; see FIGURE 11.

IV. Care and Feeding

A. Model Creation

VORLAX follows old FORTRAN conventions. It parses through an input file line-b*y*-line in a card-style fashion to input the user defined values. It does not implement name lists, XML or other free-from inputs.

When starting a *VORLAX* study, the user will define the solver type, the automatic gridding style (LAX, LAY) (linear or cosine) and the iteration limit for the Gaussian CSOR solver. The user also inputs up to 20 Mach numbers and 16 angles-of-attack as well as a fixed sideslip angle, pitch rate, yaw rate and/or roll rate. Thus, *VORLAX* may be used to compute laterall*y*-symmetric forces and moments, forces and moments at sideslip as well as *p*, *q,* and *r* dynamic derivatives.

The detailed geometry includes the quadrilateral panel position and chord, local incidence angles and camber profiles. Panels may be isolated or "reflected" about the aircraft's line of symmetry. A typical stability & control model employs a mix of elements; the main wing, horizontal tail and elevators are "reflected" elements while the ailerons are individual so that they can be deflected opposite one another. A centerline vertical tail and rudder are modelled as a pair of isolated elements.

When drawing a *VORLAX* panel, the user must take care to ensure none of the panels overlap, or else the solution will diverge or incorrect values.

Another area of caution involves overall panel spacing. Because *VORLAX* is a simple program, there are not checks in place for common errors – that is the duty of the user to correct. *VORLAX* sees heavy usage in both of Professor Takahashi's aerodynamic and design courses at Arizona State. There are cases where students mistakenly define panel locations incredibly far from the body of the aircraft, thereby causing stability moments to be massive in magnitude. Another common mistake is having incorrect zero-flux directionality specified, which will present incorrect values. Finally, it is common to define incorrect spacing for the sandwich panels, thereby causing the panels to lie coplanar with each other, or to have a vast gap between them. Because of the fact that *VORLAX* does not capture shock nor separation effects, this will cause the program to see the airfoil as very thick, and thus the user will obtain the appropriate lift coefficient value for a very thick airfoil with no shock.

Some of these usage errors are quickly remedied via visualization and analysis tools, while others require input file debugging, like a typical computational fluid dynamics program. In the case of student usage, some errors are obvious, as obtaining $C_L = 100.94$ is obviously wrong. Problems arise more typically when debating whether $C_L = 0.621$ or $C_L = 0.644$ are more accurate. Thus, attention to detail when using *VORLAX* is very important, as is good instruction when showing others how to use the program.

B. Grid Spacing and Density Study

VORLAX allows the user to adjust grid density, offering two spacing methods and density parameters. If the user chooses longitudinal cosine spacing the grid is clustered in the leading and trailing edge than in the center. If the user chooses transverse cosine spacing the wing will feature higher grid density near the centerline and wingtips.

To evaluate the grid behavior, we validated it against wind tunnel results on a symmetric, uncambered, low aspect ratio wing, following geometric guidelines from *NACA RM A50K28a [8]* see FIGURES 12 & 13. Note that the test configuration featured only a half-span wing section mounted close to the tunnel wall, thus introducing aerodynamic interference via the testing setup.

Our original intention was to examine *VORLAX* performance for both the cosine and linear chordwise spacing techniques. Upon further examination of the program and theory guide, it was discovered that linear spacing was incompatible with the chosen method of resolving leading-edge suction effects. Thus, the linear spacing was disabled for runs involving subsonic freestream Mach numbers (it was left available for supersonic freestream Mach numbers, as leading-edge suction is a strictly subsonic phenomenon). Thus, only the cosine chordwise spacing was used to compare spanwise spacing methods.

The run was simulated for both a flat plate and sandwich panel model. The *VORLAX* model was run at 20 different half-span grid densities ranging from 5 to 100 spanwise points. The chord density varied from 4 to 20 for each span density, generating 180 different runs. There was a single freestream Mach number of interest, $M_\infty = 0.21$, which was run alongside a complete range of pitching angles, $-10° \leq \alpha \leq 16°$. This test was repeated four times for each configuration, varying the spacing methods as described in Table 1 (overleaf).

Despite the wide range of tests, the most computationall$y$-intensive of configurations took only 1.95 seconds to run for a single Mach number and 14 angles of attack on a consumer-grade PC with an Intel Core i5-9400F running at 4.1GHz. On the upgraded Intel Core i9-9900K, this was further reduced to only 1.55 seconds.

FIGURE 15 show the results of the comparisons for both of the flat panel configurations – comparing limits of grid des – one which has cosine spacing both chordwise and spanwise, and another with cosine spacing in the chordwise direction and linear spacing in the spanwise direction. It is clear that the *VORLAX* output results are not substantially impacted by grid density. Our trials were unbiased, some of our attempts featured incredibly coarse grids or grids with very badly

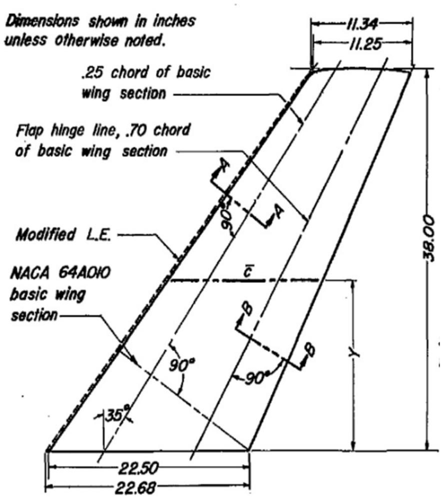

Dimensions shown in inches unless otherwise noted.
11.34
11.25
.25 chord of basic wing section
Flap hinge line, .70 chord of basic wing section
A
90°
A
Modified L.E.
c̄
NACA 64A010 basic wing section
B
90°
B
90°
35°
38.00
y
22.50
22.68

FIGURE 12 – Validation case from NACA RMA50K28a [8]

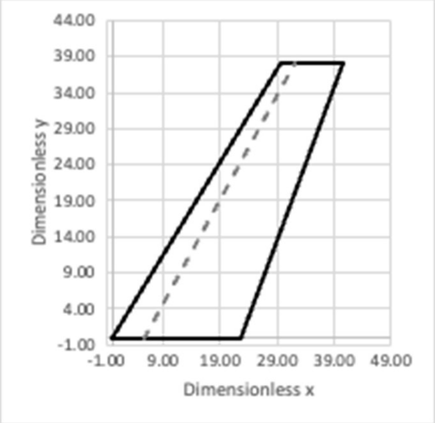


FIGURE 13 – *VORLAX* Model

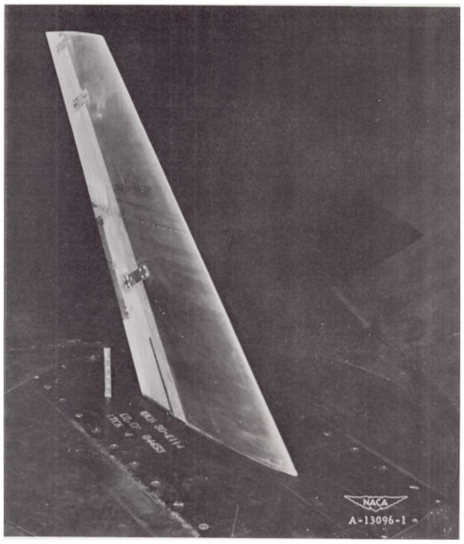


FIGURE 14 – Installation Photograph [8]

skewed grid aspect ratios. Nonetheless, the results all agree comparably – it is difficult upon visual inspection to point out a "best" model. Upon closer inspection, for the cosine spanwise grid we suggest that the best grid for the flat plate model has NVOR = 60 and RNCV = 20; for the linear spanwise grid we suggest NVOR = 50 and RNCV = 20. [9]

Table 1. Trial Configurations

Configuration	SPANWISE SPACING	CHORDWISE SPACING
LAX=0; LAY=1	COSINE	LINEAR
LAX=0; LAY=0	COSINE	COSINE

FIGURE 16, overleaf, demonstrates one of the specific challenges faced when calibrating inviscid *VORLAX* data to wind tunnel; *VORLAX* lacks any capacity to model skin friction. We can see here, comparing our preferred cosine-spaced flat panel model to the NACA report data how at small angles of attack, the CL vs ALPHA plots align almost perfectly. The correlation with the "true" tested CL begins to fall off once the incidence reaches stall; α>12-degrees. To match the drag coefficients, it was beneficial to add an approximation for the friction drag. On this FIGURE, we add a zero-lift-drag increment to the *VORLAX* CD data to represent skin friction based upon a *Cf* and form-factor approach; see Reference [10]. This method uses an empirical Reynolds number approach as a function of the mean chord length to approximate the friction drag. We see excellent drag correlation for α <10-degrres, with the wind tunnel registering additional drag as the wing approaches stall.

The results of the grid density testing are very complimentary to the intended usage case for *VORLAX*. This program is meant to be something that an engineer may use on their work machine without a ton of preprocessing work or library updates. It is meant to work efficiently, and that begins at the human operating the program. The results returned

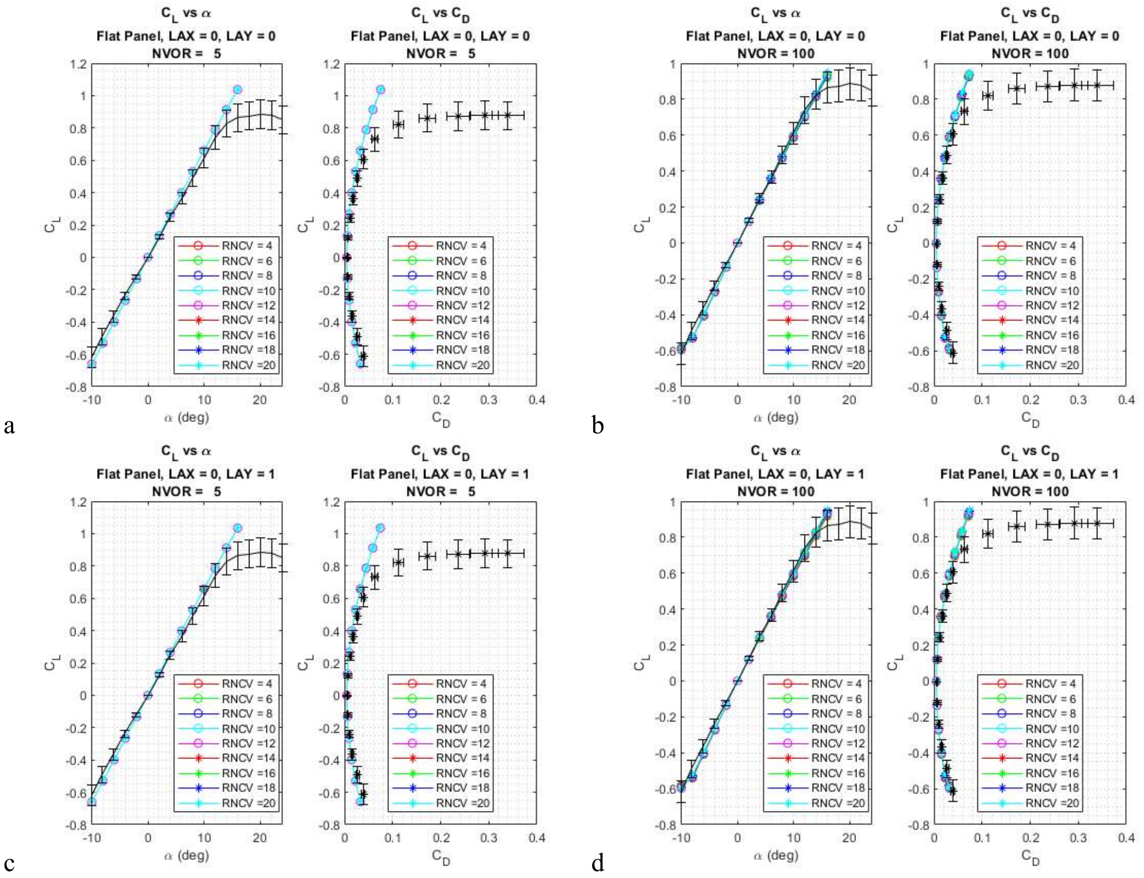


FIGURE 15 – *NACA RM A50K28a* Validation Study. [8] a) coarse grid cosine/cosine spacing, b) fine grid cosine/cosine spacing, c) coarse grid cosine/linear spacing, d) fine grid cosine/linear spacing

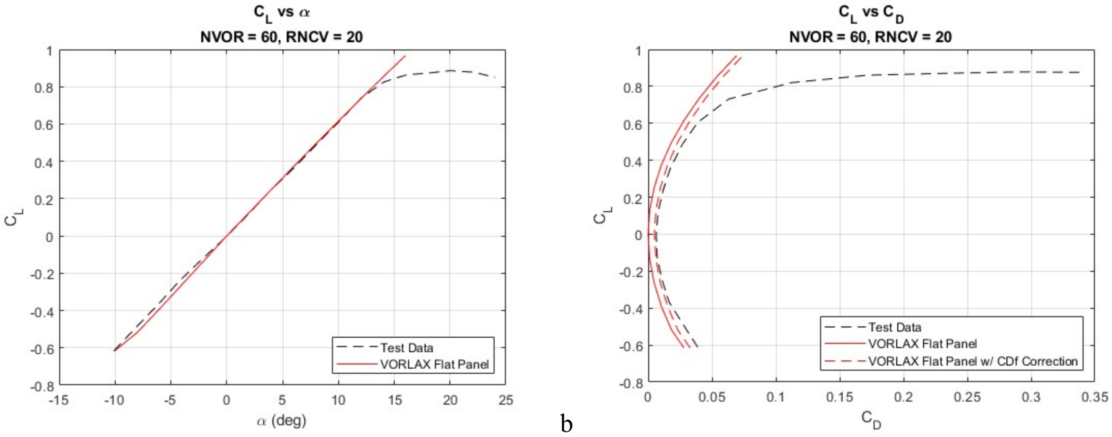


FIGURE 16 – Flat Plate *VORLAX* correlation to test data from *NACA RM A50K28a* [8]

are enough to help the user estimate the lift of a configuration, estimate the drag of the configuration, as well as other things such as the stability and elliptical loading of the design. While the above outlines the scenarios for "ideal" grid configurations, it is important to note that almost none of the configurations give results that are completely incorrect to the point of destroying intuition. For those configurations that do give remarkably poor results (namely those with the sandwich panel method), the results behave in a manner where any engineer with an understanding of aerodynamics should be able to identify that they are incorrect. That is, in essence, the "point" of the vortex-lattice method – it exists as an underappreciated tool that is immensely useful for first-order design considerations.

C. Sandwich Panels and Drag

We next turned to repeat these same tests using the sandwich panel configuration. However, this test presented more interesting results. For this study, we use *VORLAX* in the NPP=0 *ITS*=+1/-1 mode to produce a simple sandwich panel.

To incorporate both thickness and camber, the designated coordinates are superimposed atop one another, given as

$$\left(\frac{y}{c}\right)_{\text{Top}} = \frac{1}{2}\left(\frac{y}{c}\right)_{\text{Thickness}} + \left(\frac{y}{c}\right)_{\text{Camber}} \qquad (4a)$$

$$\left(\frac{y}{c}\right)_{\text{Bottom}} = -\frac{1}{2}\left(\frac{y}{c}\right)_{\text{Thickness}} + \left(\frac{y}{c}\right)_{\text{Camber}} \qquad (4b)$$

which allows for incredibly precise representation of nearly any airfoil shape. Thus, it becomes possible to use *VORLAX* in conjunction with scripting to build a very powerful aerodynamic design tool. With such a tool, hundreds of wing designs can be tested by making very small changes via the combination of geometric positioning, thickness profiles, camber lines, and wing twist.

FIGURE 17 (overleaf) shows the results of the testing for the planar sandwich panels modeling the geometry from NACA RM A50K28a. [8]. From these plots, it is clear that the lift curves are more-or-less spot-on relative to the tests. At the same time, the drag polar plots are blatantly incorrect; they are the byproduct of computing drag by pressure integration over a coarse grid. This was no surprise to Professor Takahashi given his experience, but students do not realize that even a 100x20x2 (i.e. a 4000 element panel method model) completely misses drag. The "minor problem" notwithstanding, we will show later in this paper why the sandwich panel has great utility – it can predict accurate and detailed surface pressure distributions useful for everything but drag.

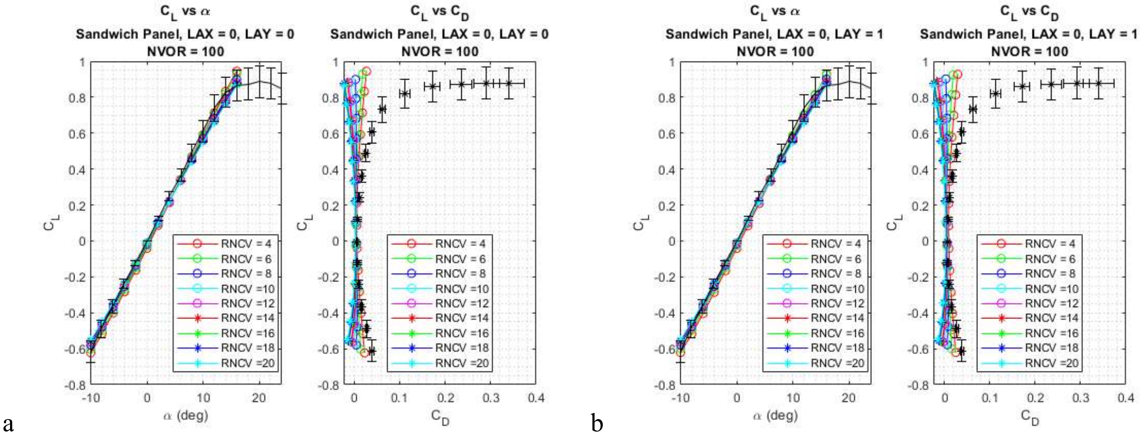


FIGURE 17 - Fine Spanwise Spacing comparison of linear versus cosine methods, Sandwich Panel (NPP=0)

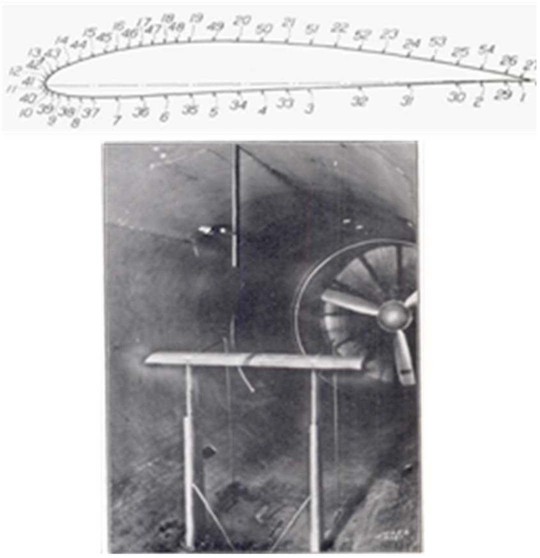


FIGURE 18 – Wind Tunnel Model for NACA 4412 pressure validation [11]

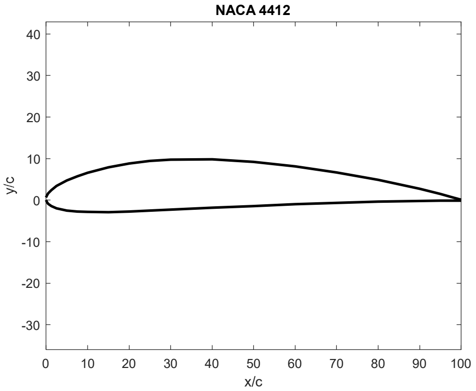


FIGURE 19 - NACA 4412 Airfoil Ordinates [12]

D. Pressures Over a Thick Cambered Wings

Code validation is an important part of any CFD program, with no exceptions afforded to the vortex-lattice method. One reliable metric of comparing a CFD program to real-world results is to compare a localized pressure distribution, preferably one that was completed in a 3-D wind tunnel. A *VORLAX* model for an incompressible NACA 4412 wing was run to compare the pressure distribution to the results given by Pinkerton. [11] This study presented high-quality pressure distributions over the surface of the wing when tested in the Langley variable-density wind tunnel, thereby providing an accurate dataset to compare the output results from *VORLAX*.

FIGURE 18 shows the airfoil profile and its pressure orifice locations (top) along with a photo of the wing mounted in the wind tunnel (bottom). The large number of orifices provided very detailed pressure measurements that were tabulated in the reference document, therefore removing a source of error associated with any need to "trace" the pressures from the image of a plot.

FIGURE 19 shows the shape of the NACA 4412 profile [12] as input into *VORLAX*. FIGURE 20, overleaf, shows the results of the pressure distribution comparison. While the results are not identical, they are remarkably close at the centerline of the wing. The differences arise primarily from the lack of viscous effects (and therefore the lack of a boundary layer), so in the context of the vortex-lattice method, the correlation is quite good. The magnitudes of the pressure coefficients are comparable and occur in approximately the same locations, thereby providing excellent insight into the nature of the flow behavior about the finite wing

.

E. *VORLAX* Wake Survey

VORLAX offers wake survey functionality; it is activated by simple flags placed at the footer of the input file. This capability can be very useful due to its ability to present wash components due to a body in passing airflow. Some may wish to use the wake survey to resolve blunt body drag, for example. Another technique useful via the wake survey option is for engine design and placement. In the case of a propeller-driven aircraft, the user may utilize the wake survey feature for a plane directly in front of the nose of the aircraft in order to align the propellors in a manner that preserves the integrity of the flow distribution over the main wing and body.

To access the wake survey, the user specifies the number of desired transverse planes to "cut" across the solution. This *y*-*z* plane is placed at an *x*-coordinate station in a user-defined location in the proximity of the aircraft and will return an output featuring the cartesian components and the velocity components at those locations. The user may also employ a finite-difference method to calculate the vorticity at each point. For the following example, this was done using 2nd-order central difference schemes:

$$f'(x) \approx \frac{f(x+h)-f(x-h)}{2h} \quad (5)$$

for the inner points of the grid and using 1st-order forward and backwards differences at the edges at the grid.

$$f'(x) \approx \frac{\big(f(x+h)-f(x)\big)}{h} \quad (6)$$

$$f'(x) \approx \frac{f(x)-f(x-h)}{h} \quad (7)$$

While the quantities at the edge of the plane domain contain slightly larger errors than those in the inner portion, the wake survey can be specified in such a way that the edges are considered "far field", and do not pertain to the vortex behavior in the immediate proximity of the body.

Recall that vorticity is given via

$$\vec{\omega} \equiv \nabla \times \vec{u} \; (8)$$

and because *VORLAX* gives the exact velocities over the grid, the finite-difference equations are perfectly adequate to approximate the derivatives of the velocities in space necessary to obtain the vorticity. Furthermore, because this is a matter of a 2D planar "cut" in space, the only vorticity of interest is that in the *x*-direction, given as

$$\omega_x \equiv \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z} \; (9)$$

Thus, the user may determine the vorticity concentrations within the cut.

Turn next to FIGUREs 21, 22 and 23, where we show a cross-plane surveys for the flow ahead of and behind an AR = 2 wing analyzed at M=0.3 and α=5-degrees. The grid used to compute the flow-field survey has 75 stations in the spanwise

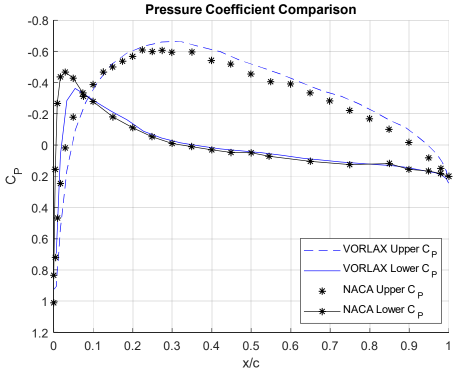


FIGURE 20. Pressure Coefficient about NACA 4412 Wing of Finite Span [11]

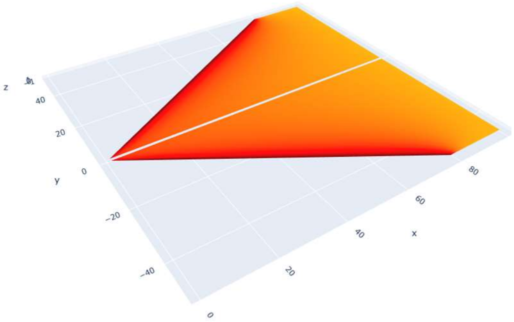


FIGURE 21. AR = 2 Wing in *VORLAX* for Wake Survey Analysis

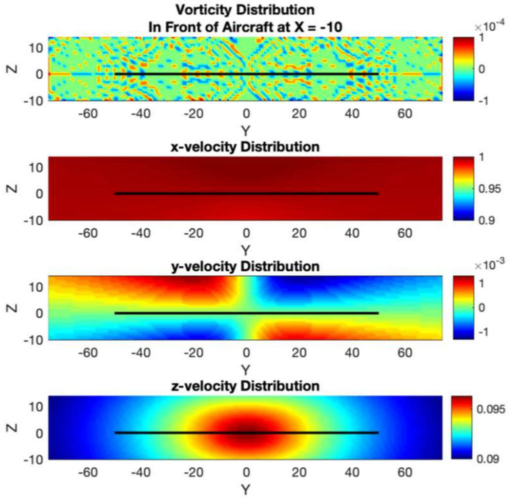


FIGURE 22 - Wake Survey Results ahead of a finite wing.

direction and 25 stations in the vertical direction. We looked at the cross-plane flow directly in front of the wing (at X=-10) and slightly behind the rearmost trailing edge coordinate of the wing (at X=+105). As expected, the plane just aft of the trailing edge had much more occurring than the plane before the nose, however, there was still some interesting behavior.

Turning to FIGURE 22, we see that the upstream vorticity distribution shows nothing other than extremely small fluctuations. While small and essentially negligible, the values are nonzero and are split between positive and negative values of vorticity. Recall that the sign notation for vorticity follows the right-hand rule, so those values greater than zero indicate a vorticity concentration that would act counter-clockwise in this view, while those less than zero indicate a clockwise direction. The black line visible in the contour plot represents the location of the wing panel (this was a planar configuration) and was included for reference.

Continuing with FIGURE 22, we see that the incoming freestream *x*-component of velocity (the primary direction of the freestream flow) is largely constant. One takeaway from this plot is that the velocity in the region above the wing is slightly higher in magnitude than that below the wing. This implies that there are some other velocities that exist to take away from the energy in the *x*-direction velocity. A portion of this change comes from the induced velocities associated with the upwash component of the flow.

The *y*-velocity profile shows that there are incredibly small velocities occurring in the spanwise direction, even prior to the airflow reaching the aircraft. This corresponds with the change to the *x*-velocities. On the lower portion of the plot, the velocities indicate that the small flows are moving away from the aircraft center plane on the bottom portion while they move towards the center plane on the upper surface. This is consistent with the contours for vorticity, in which there were small yet visible vortices visible near the wingtips.

Finally, the contours for the *z*-velocity components are some of the most interesting. Although this "slice" of the flow occurs prior to the wing passing through the location in space, there is a considerable magnitude of air with an induced vertical velocity. Namely, there is an upwash component equal to ~9% of the *x*-velocity visible in the region just before the aircraft. Thus, the true direction of the flow normal to the wing does not occur head-on, but rather it occurs with a small upward component. This information, while interesting to look at, also serves a practical purpose in aircraft design applications. By understanding the direction of the airflow that the aircraft "sees", it becomes easier to design propeller systems and even turbofan systems optimally. For propellers, this means that the effects due to the propellers on the over-wing airflow can be minimized and accounted for in design. As for turbofan applications, this becomes useful when arranging the diffuser, in which it is desirable to have a geometry that maintains shock-free flow during engine operation.

For the survey occurring in the region aft of the wing, see FIGURE 23 the contours show airflow conditions that are considerably different than those in the region before the wing. The vortices are now much more concentrated, with large vorticity magnitudes occurring at the wingtips. These are the well-understood vortices that lead to increases in induced drag, and they are largely due to the large pressure imbalance between the upper and lower surface of the wingtips. Much like the region before the aircraft, the *x*-velocity (out of the page) is largely uniform, but with small differences in the regions above and below the wing. Likewise, the *y*-velocity shows the same behavior as in FIGURE 22, however there are now much larger magnitudes that are concentrated at the wingtips, rather than small magnitudes occurring in the entire proximity near the wing. It should be noted that these *y*-direction velocities on the surface of the wing are largely the reason that two-dimensional approximations of airflow on a wing are hugely inaccurate. The velocities in this model show that the

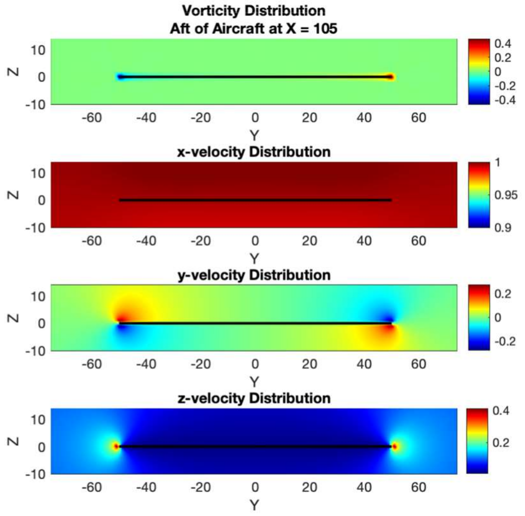


FIGURE 23 - Wake Survey Results behind a finite wing.

magnitudes of the spanwise velocity components are of the order of 20% the freestream flow velocity, which is considerably large.

Finally, the components of velocity in the *z*-direction have shifted towards the wingtips, becoming much more concentrated. This is a large difference from the region causing upwash before the aircraft, as the *z*-components of the velocities are now acting in a much tighter region, but are doing so at a magnitude four times those before the aircraft. In both cases, the *z*-velocities are entirely positive. This leads to a decreased effective angle of attack for the wing of the aircraft, again agreeing with the commonly understood behavior of induced drag.

F. Non-Planar Modeling of Camber – Thin & Sandwich Panels

VORLAX offers two options to model cambered surface; this is controlled by the NPP parameter which may impact the geometry of both thin and "sandwich" panel models. As shown above in FIGURE 10, *VORLAX* typically models cambered surfaces as a "venetian-blind." If we set NPP=1, *VORLAX* will vertically displace the panels; see FIGURES 23 and 24.

When contrasting FIGURES 24 and 25 to FIGURES 9 and 10, it is clear that when NPP=1, the locations of the points are changed drastically. By plotting these values, some of the nuances of this method become apparent. For instance, in FIGURE 23, the slope at the quarter-chord is not particularly well-aligned with the camber profile. This is due to the "cosine" method of drawing chordwise control points, where areas away from the leading- and trailing-edge are comparably coarse. FIGURE 24 shows again that the *VORLAX* drawings are well-aligned with the general profile of the section, however there is some disagreement at the leading edge regarding the slope of the section with the first control point.

In FIGURE 26, we compare the results of running the same basic wing as *NACA RM A50K28* as previously, however this time modified with a uniform application of the NACA 63 camber line. [8][9] While the original tests did not include camber, the tests were run with the expectation that there be a near-linear increase in lift for the entire range of pitch angles. While the modification presented the correct assumed change, we questioned the accuracy of the lift magnitudes as they seemed slightly high. FIGURE 25 clearly shows that the nonplanar example of the lift test is of a greater magnitude than both the flat panel with camber and the planar sandwich panel, which indicates that the nonplanar sandwich panels overpredict lift. Likewise, FIGURE 26 shows that the correlation of the drag coefficients is also poor, which tells us

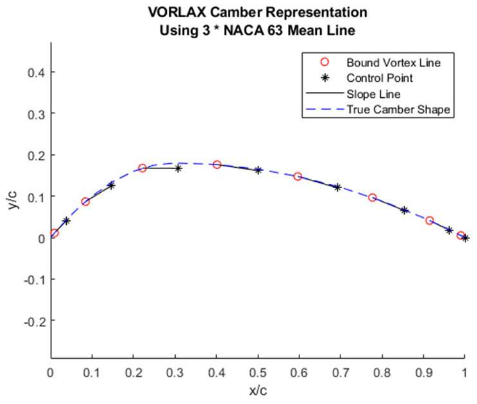


FIGURE 24 – NPP=1; Modeling of non-planar cambered thin panel

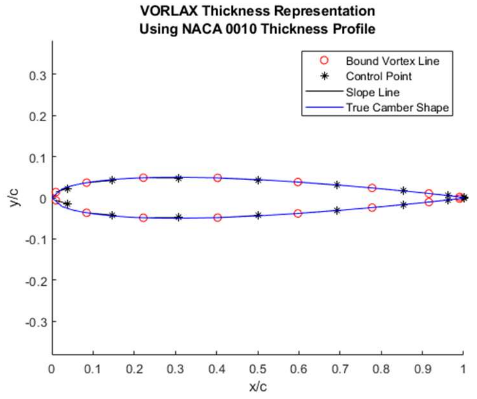


FIGURE 25 – NPP=1; Modeling of a non-planar "sandwich panel"

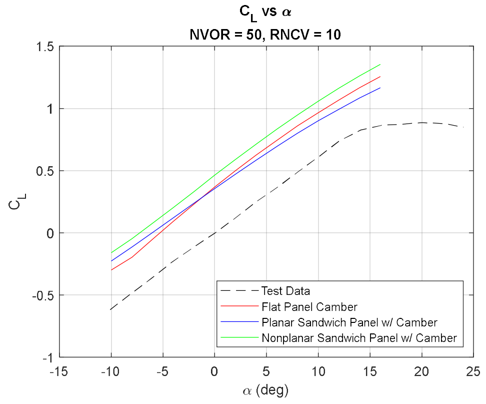


FIGURE 26 –CL vs alpha NPP=1; Modeling of non-planar cambered thin panel

overall that there are bugs with the nonplanar mode that prevent it from being a completely feasible method of measuring lift and drag.

Within the *VORLAX* code, there are a few places where small-angle approximations appear. One of these was tied to the nonplanar parameter (NPP). In an attempt to reconcile the difference between the planar and nonplanar case, a value for NPP was used to avoid the small-angle approximations. While the estimate lift closely matches between the various methods, large discrepancies remain with drag computations. This implies that there is more to the difference than simply the small angle approximation. However, the nonplanar parameter does not actually appear often in the *VORLAX* source code, and so the differences in the drag may be arising due to conflicts in other portions of the code, such as those related to the induced drag. Because the linear chordwise spacing was shown to significantly impact the quality of the drag polar measurements, there is a chance that the displacement of the points due to nonplanarity may lead to errors, as well.

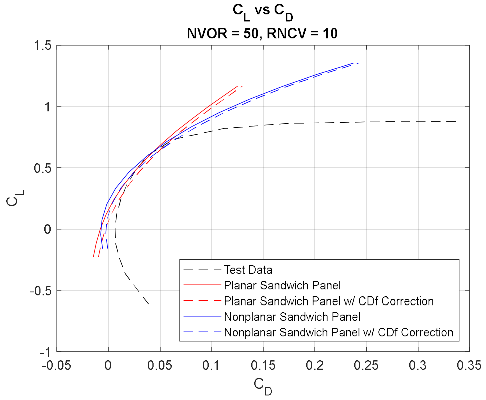


FIGURE 27 – CD vs CL NPP=1; Modeling of a non-planar “sandwich panel”

We conclude that neither NPP=0 nor NPP=1 solutions for “sandwich” panel aerodynamic models are appropriate to estimate drag. The method of drag prediction via flat plates remains the best option within *VORLAX* for computing drag.

The nonplanar case is included in *VORLAX2020* but with caveats. In the event that the user invokes this mode, the code will add a disclaimer asterisk to the LOG and output files.

G. Complex Loft Analysis – Detailed Surface Pressures

Finally, the most useful mode in aircraft design involves imposing design changes due to both the camber and the thickness of the wing. Because the aerodynamic properties of the wing relate to the combination of camber, thickness, and twist, the only way to accurately resolve the effects (where $c_p < c_p^*$) is to include all three. Recall, the accuracy of *VORLAX* holds only until flow separation due to stall and in shock-free regions. Thus, we have taken great care to streamline and document the process of analyzing wing pressure loading in order to understand the tendencies of the flow over a 3-D wing.

We see in FIGURE 28 (overleaf) another example of the *NACA RM A50K28*, this time with contours at a low Mach number.[8] To demonstrate three-dimensional effects of the wing airflow, this model was run at four angles of attack, thereby giving a clear progression from no pitch to a moderate pitch angle. It is immediately clear that the progression of the pressures do not behave in a way that may be construed as 2-D. For instance, between $\alpha = 0°$ and $\alpha = 2°$, the overall magnitude of the pressure coefficients become more negative, however this does not occur uniformly, rather it occurs in a region concentrated towards the wingtip. Likewise, the higher angles of attack show even more drastic 3-D effects. Another area that demonstrates this is the wingtip region. For the two smaller angles of attack, the pressure contours align in parallel to the wingtip, however the higher angles of attack present contours at an angle relative to the wingtip. This can only happen if there is some force acting along the span to change the shape of the loading, for if the load was truly 2-D one would expect changes in magnitude while maintaining the same shape.

FIGURE 29 (overleaf) shows the same geometry help at a single angle of attack while the Mach number is increased. Unlike the previous case with increase pitch angles, the pressure contours stay relatively consistent in this example. There are small decreases in the local pressure coefficients, however at a magnitude much lower than those seen by pitching the wing. We completed this test to demonstrate that by using the critical pressure coefficient given in Equation (1), it is possible to use *VORLAX* to indicate the location where a shock is likely to form.

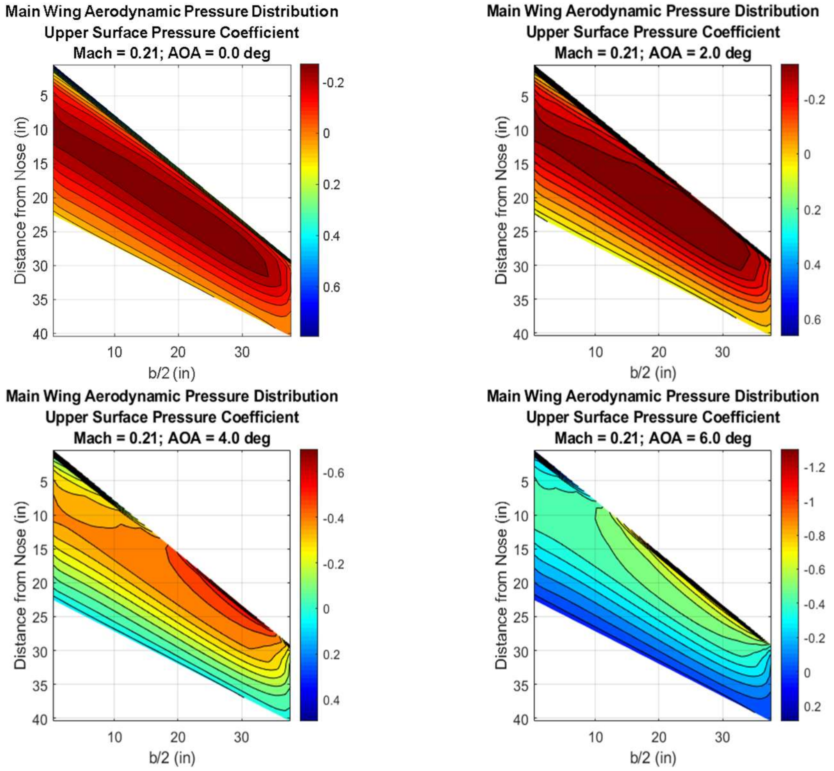


FIGURE 28 - Low Mach Pressure Loading Characteristics

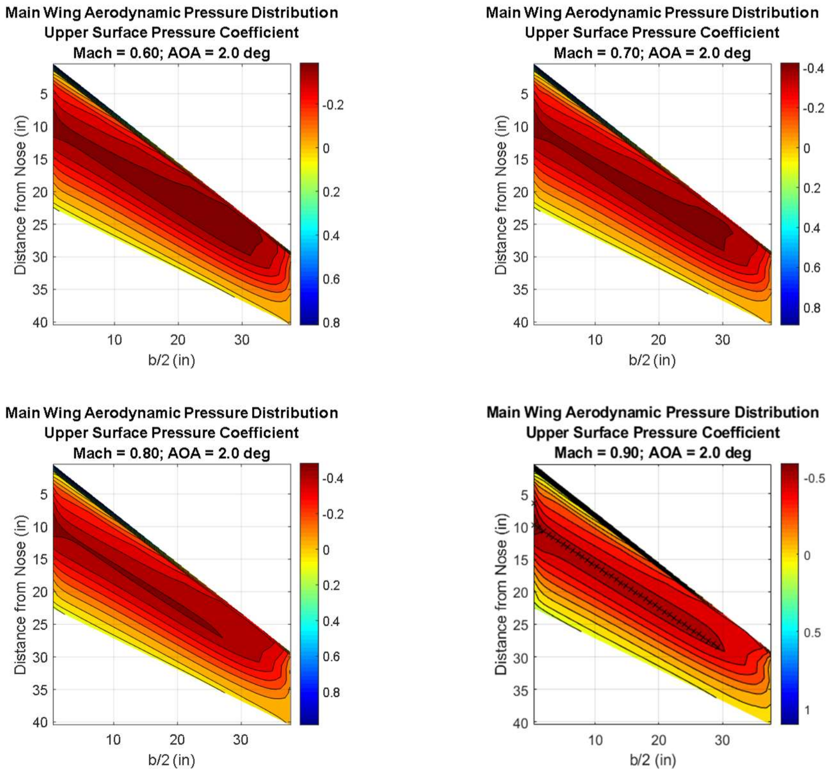


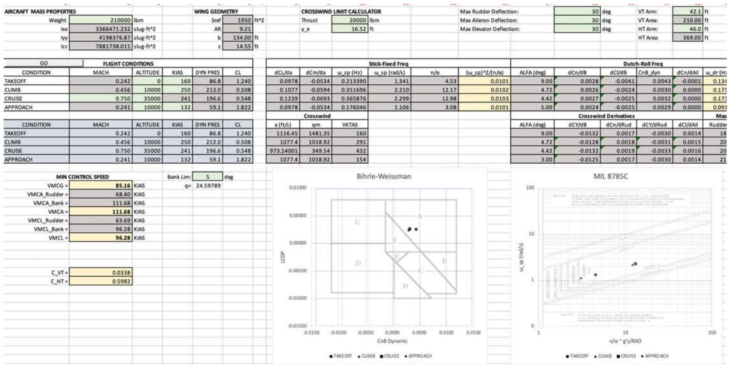
FIGURE 29. Progression of Pressure Distribution, Increasing Mach Number

In the case of FIGURE 29, the crosses imposed on the FIGURE at Mach 0.90 indicate that those control points have breached the *Cp** limit prescribed by Equation (1). The location of the shock formation is exactly where the lower Mach number examples indicate. As the Mach number increases, there is a region concentrated towards the quarter-chord with the most negative pressure coefficients for all of the configurations, and it is this region which is likely to produce a shock.

It should be noted that much like FIGURE 29, this example demonstrates 3-D loading on the wing. We see that the lowest pressure coefficient does not occur over the entire length of the wing, instead it concentrates towards the center of the wing. There is also the same behavior towards the wingtip as before, where the contour lines begin at a slight angle to the wingtip with fairly uniform spacing before forming a much more aggressive angle with the wingtip at the higher Mach numbers. Finally, there is a region of slightly lower pressure that persists near (b/2)=25 for the tests which does not reflect any kind of "clean" or "simple" progression of the pressure over the wing. Thus, we see that the flow over a wing is not easily described, especially not in a pattern that claims the pressure behaves in a manner that is at all rooted in 2-D flow behavior.

H. Complex Loft Analysis – Stability & Control Data Base

VORLAX is also able to provide the information necessary to compute stability and control parameters of an aircraft. The panel method is very versatile for representing control surfaces. Ailerons, elevators, and rudders may be modeled using a standard flat panel, and its deflection is represented using the panel twist parameter in its input card. Thus, it is possible to fully automate a basic stability and control spreadsheet for use in early design stages.



AIRCRAFT MASS PROPERTIES

Weight	210000	lbm
Ixx	3366471.232	slug-ft^2
Iyy	4198376.87	slug-ft^2
Izz	7881738.011	slug-ft^2

WING GEOMETRY

Sref	1950	ft^2
AR	9.21	
b	134.00	ft
c	14.55	ft

CROSSWIND LIMIT CALCULATOR

Thrust	20000	lbm
y_e	16.52	ft

Max Rudder Deflection:	30	deg
Max Aileron Deflection:	30	deg
Max Elevator Deflection:	30	deg

VT Arm:	42.1	ft
VT Area:	210.00	ft
HT Arm:	46.0	ft
HT Area:	369.00	ft

GO

FLIGHT CONDITIONS

CONDITION	MACH	ALTITUDE	KIAS	DYN PRES	CL
TAKEOFF	0.242	0	160	86.8	1.240
CLIMB	0.456	10000	250	212.0	0.508
CRUISE	0.750	35000	241	196.6	0.548
APPROACH	0.241	10000	132	59.1	1.822

Stick-Fixed Freq

dCL/da	dCm/da	ω_sp (Hz)	ω_sp (rad/s)	n/α	(ω_sp)^2/(n/α)
0.0978	-0.0534	0.213390	1.341	4.53	0.0101
0.1077	-0.0594	0.351696	2.210	12.17	0.0102
0.1239	-0.0693	0.365876	2.299	12.98	0.0103
0.0978	-0.0534	0.176046	1.106	3.08	0.0101

Dutch-Roll Freq

ALFA (deg)	dCn/dB	dCl/dB	CnB_dyn	dCn/dAil	ω_dr (H
9.00	0.0028	-0.0041	0.0043	-0.0001	0.134
4.72	0.0026	-0.0024	0.0030	0.0000	0.175
4.42	0.0027	-0.0025	0.0032	0.0000	0.173
5.00	0.0024	-0.0025	0.0029	0.0000	0.091

CONDITION	MACH	ALTITUDE	KIAS	DYN PRES	CL
TAKEOFF	0.242	0	160	86.8	1.240
CLIMB	0.456	10000	250	212.0	0.508
CRUISE	0.750	35000	241	196.6	0.548
APPROACH	0.241	10000	132	59.1	1.822

Crosswind

a (ft/s)	qm	VKTAS
1116.45	1481.35	160
1077.4	1018.92	291
973.14001	349.54	432
1077.4	1018.92	154

Crosswind Derivatives

ALFA (deg)	dCY/dB	dCn/dRud	dCY/dRud	dCl/dAil	Max Rudder
9.00	-0.0132	0.0017	-0.0030	0.0014	18
4.72	-0.0128	0.0018	-0.0031	0.0015	20
4.42	-0.0132	0.0019	-0.0033	0.0016	20
3.00	-0.0125	0.0017	-0.0030	0.0014	21

MIN CONTROL SPEED

VMCG =	85.16	KIAS
VMCA_Rudder =	68.40	KIAS
VMCA_Bank =	111.68	KIAS
VMCA =	111.68	KIAS
VMCL_Rudder =	63.69	KIAS
VMCL_Bank =	96.28	KIAS
VMCL =	96.28	KIAS

Bank Lim: 5 deg

q= 24.59789

C_VT =	0.0338
C_HT =	0.5982

FIGURE 30 – Example *VORLAX* S&C post-processing tool developed in EXCEL/VBA

To automate the process, a scripting language such as VBA will run through five *VORLAX* input models, including a neutral configuration, a model with one degree of sideslip, and then one model for each of the control surfaces deflected to their maximum angle. By doing so, one may recover all major stability derivatives, and with nothing more than basic cell arithmetic, can present stability and controllability outcomes to a reasonable degree of certainty. We can estimate crosswind trim limts, minimum engine inoperative control speed (VMCA), Dutch-Roll frequency, Short-Period Frequency, and automate the construction of Bhirle-Weissman and 8785C controllability plots; see FIGURE 20. [13]

Prior to the update to *VORLAX*, the slower execution time made stability and control analysis tedious. If we ran each of the five models across 14 angles of attack and three Mach numbers, we experienced aggregate runtimes around 10 minutes. With the *VORLAX* updates, this time was reduced drastically, now taking only a matter of seconds per run. This greatly streamlines the stability and control analysis of an aircraft using *VORLAX*.

I. Complex Loft Analysis – High Lift Systems

As another demonstration for the wake surveying feature of *VORLAX*, a wake survey was completed for a model of an entire airframe including flaps, courtesy of Gabino Martinez-Rodriguez, another one of Dr. Takahashi's graduate students. [14] This model involves a flat representation of a fuselage, complete with wings including twist, thickness, and camber. The noteworthy feature of this model is that includes flat panel high-lift devices at the leading- and trailing-edge of the main wing of the aircraft.

Downloaded by Timothy Takahashi on August 9, 2021 | http://arc.aiaa.org | DOI: 10.2514/6.2021-2459

Mr. Martinez modeled the aircraft at Mach 0.2 with an aggressive angle of attack of 12 degrees, representing an aircraft landing configuration for a narrow body aircraft comparable to a Boeing 757 in size; see FIGURE 31.

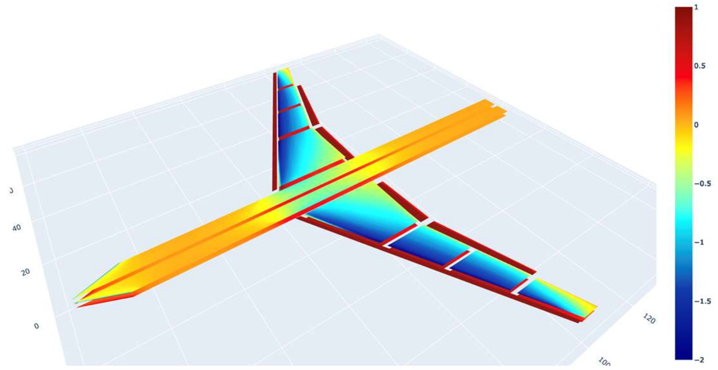


FIGURE 31 – *VORLAX* Model of Aircraft Flap Configuration [23]

FIGURE 32 shows the results for the wake aft of the aircraft. It is noticeable that the vorticity for this configuration occurs at a slightly higher concentration at more specific locations relative to the simple "wing in a flow" configuration. The locations of this vorticity concentration occur at expected regions, namely the wingtips and the edges of the trailing-edge flaps. Recall that the vorticity is understood to arise from pressure imbalance with no physical device existing to maintain that imbalance, and thus it makes sense that the vorticity concentration is in regions supporting high levels of lift near edges of surfaces.

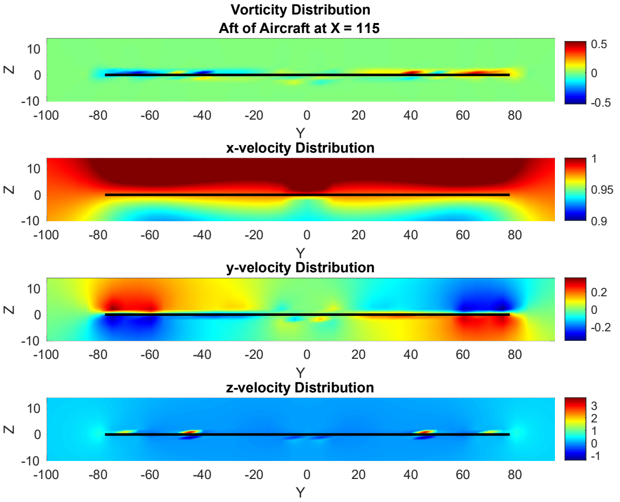


FIGURE 32 – Trailing Wake Survey of Aircraft with a deployed High-Lift System [14]

The *x*-velocity components are significantly different than those in the earlier example (FIGURE 23). While the simple wing only showed the velocity decreasing to ~97-98% of its freestream magnitude in the region under the wing, this model with flaps shows a decrease to ~90-97% in the regions below, much slower than the other case. For an aircraft in a landing configuration, the wing is making much more lift, which leads to a much larger component of flow acting downwards in the region below the aircraft. With this, there is also a large accumulation of pressure under the wing, further causing a decrease in the *x*-component of the velocity.

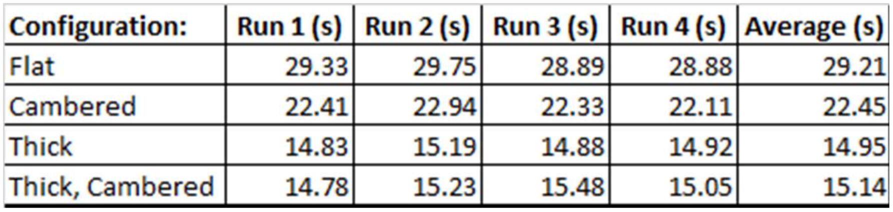
The *y*-velocity contours share similar information as those in the single wing configuration. In the flow aft of the wing, there is spanwise velocity induced towards the center plane of the aircraft above the wing, while the velocity points away from the center plane of the aircraft below the wing. While the direction and magnitudes of these velocities remains the same for the aircraft in the landing configuration, the regions where these magnitudes occur were restricted largely to the wingtips, with considerably smaller spanwise components moving in towards the center plane. However, in the landing configurations, the spanwise components stay at higher magnitudes covering a significantly larger portion of the wing. These components still become zero towards the center, but this happens over a much smaller spanwise distance.

Finally, the *z*-velocities show that there are incredibly large upwards components occurring aft of the aircraft above the main wing near the gap between trailing edge flaps. Likewise, there is a slightly smaller downward component of flow below the wing in the same region. This is considerably different than the behavior for the single wing case, as there was very little *z*-velocity in the wake behind the wing. In this case, the components at the wingtips are smaller than those for the lone wing, and there are large but concentrated components near the flap gaps under the primary wing.

V.Computation Time

To further demonstrate the utility of *VORLAX*, benchmarks were run for each of the aforementioned usage cases. Because of the nature of this tool and its intended application, the computer was not run in a "sterile" environment, but rather in a manner that mimics a "realistic usage" scenario. For clarity, this included one instance of Microsoft Edge running, two Microsoft Word files open, MATLAB open, and Slack open and running. Coupled with all of the miscellaneous background tasks, this put the average *CP*U utilization at 4%, running at 4.66±0.1 GHz, and the RAM utilization was 43%, using 13.6GB of 31.9GB available.

Table 3 *VORLAX* Runtime Benchmarks [2][9]



Configuration:	Run 1 (s)	Run 2 (s)	Run 3 (s)	Run 4 (s)	Average (s)
Flat	29.33	29.75	28.89	28.88	29.21
Cambered	22.41	22.94	22.33	22.11	22.45
Thick	14.83	15.19	14.88	14.92	14.95
Thick, Cambered	14.78	15.23	15.48	15.05	15.14

The testing consisted of running the same files used for the descriptions above in their same configuration, except maximizing the code's capabilities and running for 16 freestream Mach numbers and 16 angles of attack, totaling 256 cases. To return the most accurate runtime measurement, a feature was added into the source code in FORTRAN to implement a timing function. It takes scripting languages such as VBA time to spawn the command prompt shell and execute the program, and thus it was inappropriate to rely on error-prone time tracking methods within the scripts. When running the test, the batch file used to call *VORLAX* was rearranged in order to ensure that there was no behind-the-scenes behavior with Windows 10 giving an advantage based on the order in which the file was run.

Table 3 shows that on average, each of the cases takes under 30 seconds to run – a remarkable speed for that many configurations. At this speed, there still remains the functionality to generate accurate plots for pressure loading, aerodynamic coefficients, and even calculate stability derivatives, which can be extended to overall aircraft stability and control. Naturally, as the complexity of the system increases and its grid increases in size, the runtimes will increase. Fortunately, in the case of *VORLAX*, the "worst case" runtimes are in the order of minutes, not hours nor days. This makes *VORLAX* a remarkably powerful tool for the engineering design process, as it provides the engineer with validating figures in a very short time.

VI.Known Bugs and "Undocumented Features"

We realize that *VORLAX* has a few bugs that have remained in the code since its development in the 1970's. Of these, most have reasonable workarounds that do not compromise the overall capability of the program.

Formerly, the fusiform geometry feature was misunderstood and considered to be buggy. *VORLAX* was written in a manner where fusiform bodies are generated in a manner that is counterintuitive, however it is very consistent. The fusiform generation strategy has been documented better and example input files have been stored for future reference. Generally, representing the fuselage as an inverted "T" shape panel or as a "U" shape will provide reasonable side-of-body flow resolution for the majority of cases. However, the fusiform function is preferable for accuracy of the fuselage, so it is imperative that it is functional in order to provide the most realistic pressure disturbances possible.

The original *VORLAX* also featured a "Synthesis Mode." The user should be able to define a target pressure distribution and the code would iteratively determine the camber profiles leading to such a distribution. However, this mode has fallen out of use dating back 30+ years, and there exists minimal documentation regarding its usage. At the present time we do not understand how to make it function reliably.

Finally, the wake survey feature was formerly inoperative. This has been repaired by Professor Takahashi.

VII. Conclusion

Many people see old software and mistakenly consider it to be "bad" software. This belief does not do *VORLAX* justice. This paper demonstrates many of its powerful abilities in order to demonstrate its value in a modern engineering workflow. The pioneers of engineering computational methods were not only intelligent, but remarkably creative and resourceful. What results is a program that performs very well in a very small package. While many choose to push the bounds of what a computer can accomplish, *VORLAX* is an example of how efficiently modern technology can execute yesterday's cutting-edge.

VORLAX is an immensely powerful program relative to its computational footprint. Although many features had fallen into disuse, this paper demonstrates many of the fantastic features offered by the program. With relatively minimal training, *VORLAX* allows a user to represent an aircraft via one of many methods in order to generate their desired data however they see fit. Given the ever-increasing overhead of computing, a lean program capable of providing "90% final" data is invaluable.

Acknowledgements

This revised version of *VORLAX* is currently in use in Professor Takahashi's senior- and graduate-level courses beginning in the Fall 2020 semester. The final version released at AIAA AVIATION 2021.

This manuscript derives from work Mr. Souders performed in partial fulfillment of the degree requirements for obtaining his M.S. in Mechanical Engineering from Arizona State University.

References

1. Miranda, L. R., Elliot, R. D., and Baker, W. M. "A Generalized Vortex Lattice Method for Subsonic and Supersonic Flow Applications." NASA CR 2865, 1997.
2. Souders, T.J. and Takahashi, T.T. "VORLAX 2020: Making a Potential Flow Solver Great Again," AIAA 2021-2458, 2021.
3. Feagin, R.C. and Morrison, W.D. "Delta Method, An Empirical Drag Buildup Technique," NASA CR 151971, December 1978.
4. Bertin, J.J. and Cummings, R.M., "Dynamics of an Incompressible, Inviscid Flow Field", *Aerodynamics for Engineers,* 5th ed., Pearson Prentice-Hall, New Jersey, 2009, pp 113-127
5. Anderson, J.D., "Linearized Flow", *Modern Compressible Flow with Historical Perspective,* 3rd ed., McGraw Hill Education (India) Private Limited, Chennai, 2012, pp 327
6. Küchemann, D., *The Aerodynamic Design of Aircraft*, AIAA, Virginia, 2012
7. Lan, C. E. "A Quasi-Vortex-Lattice Method in Thin Wing Theory." Journal of Aircraft, Vol. 11, No. 9, 1974, pp. 518--527.
8. Demele F.A. and Sutton F.B., "The Effects of Increasing the Leading-Edge Radius and Adding Forward Camber on the Aerodynamic Characteristics of a Wing with 35° of Sweepback", NACA RM A50K28a, February, 1951.
9. Souders, T.J., "Moderization of a Vortex-Lattice Method with Aircraft Design Applications," M.S. Thesis, Mechanical Engineering Department, Arizona State University, Tempe, AZ, 2021.
10. Takahashi, T.T., *Aircraft Performance and Sizing, Vol. I: Fundamentals of Aircraft Performance*, Momentum Press, 2016.
11. Pinkerton, R.M., "Calculated and Measured Pressure Distributions Over the Midspan Section of the N.A.C.A. 4412 Airfoil", NACA Report No. 563, 1936.
12. https://ntrs.nasa.gov/api/citations/19930091638/downloads/19930091638.pdf
13. Abbott, I., Von Doenhoff, A. & Stivers, L., "Summary of Airfoil Data," NACA Report 824, National Advisory Committee for Aeronautics, 1945.
14. Takahashi, T.T., *Aircraft Performance and Sizing, Vol. II: Applied Aerodynamic Design*, Momentum Press, 2016.
15. Martinez-Rodriguez, G., "A Study and Design of Multi-Element High Lift Systems for Commercial Transport Aircraft", M.S. Thesis, Aerospace Engineering Department, Arizona State University, Tempe, AZ, 2021.

APPENDIX

A. Flat Panel Input File

```
Flat Panel Input
*00000000011111111112222222222333333333344444444445555555555666666666677777777778
*23456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
*ISOLV    LAX       LAY       REXPAR    HAG       FLOATX    FLOATY    ITRMAX
0.0       0.0       1.0       0.2       0.0       0.0       0.0       399.0
*NMACH    MACH
1.0        0.21
*NALFA    ALPHA
14.0       -10 -8 -6 -4 -2 0 2 4 6 8 10 12 14 16
*LATRL    PSI       PITCHQ    ROLLQ     YAWQ      VINF
0.0       0.0       0.0       0.0       0.0       1.0
*NPAN     SREF      CBAR      XBAR      ZBAR      WSPAN
1.0       1280.0    16.84     17.456    0.0       76.0
*--------------------------------------------------------------------------------
* Horizontal Tail
*X1       Y1        Z1        CORD1
0.0       0.0       0.0       22.5
29.43     38.0      0.0       11.25
*NVOR     RNCV      SPC       PDL
100.0     20.0      1.00      0.0
*AINC1    ANINC2    ITS       NAP       IQUANT    ISYNT     NPP
0.0       0.0       0.0       0.0       0.0       0.0       0.0
*--------------------------------------------------------------------------------
* NXS     NYS       NZS
0.0       0.0       0.0
*
*
* END
```

B. Cambered Panel Input File

```
NACA 63 Mean Line Input
*00000000111111111122222222223333333333444444444455555555556666666666777777777778
*23456789012345678901234567890123456789012345678901234567890123456789012345678901234567890
*ISOLV    LAX       LAY       REXPAR    HAG       FLOATX    FLOATY    ITRMAX
0.0       0.0       1.0       0.2       0.0       0.0       0.0       399.0
*NMACH    MACH
1.0        0.21
*NALFA    ALPHA
14.0       -10 -8 -6 -4 -2 0 2 4 6 8 10 12 14 16
*LATRL    PSI       PITCHQ    ROLLQ     YAWQ      VINF
0.0       0.0       0.0       0.0       0.0       1.0
*NPAN     SREF      CBAR      XBAR      ZBAR      WSPAN
1.0       1280.0    16.84     17.456    0.0       76.0
*-------------------------------------------------------------------------------
* Horizontal Tail
*X1       Y1        Z1        CORD1
0.0       0.0       0.0       22.5
29.43     38.0      0.0       11.25
*NVOR     RNCV      SPC       PDL
100.0     20.0      0.81      0.0
*AINC1    ANINC2    ITS       NAP       IQUANT    ISYNT     NPP
0.0       0.0       0.0       18.0       0.0       0.0       0.0
*-------------------------------------------------------------------------------
*
* X/C
 0.0000
 1.2500
 2.5000
 5.0000
 7.5000
 10.0000
 15.0000
 20.0000
 25.0000
 30.0000
 40.0000
 50.0000
 60.0000
 70.0000
 80.0000
 90.0000
 95.0000
 100.0000
* CAMBER ROOT
 0.0000
 0.4890
 0.9580
 1.8330
 2.6250
 3.3330
 4.5000
 5.3330
 5.8330
 6.0000
 5.8780
 5.5100
 4.8980
 4.0410
 2.9390
 1.5920
 0.8270
 0.0000
* CAMBER TIP
 0.0000
 0.4890
 0.9580
 1.8330
 2.6250
 3.3330
 4.5000
 5.3330
 5.8330
 6.0000
 5.8780
 5.5100
 4.8980
 4.0410
 2.9390
 1.5920
 0.8270
 0.0000
*
* NXS     NYS       NZS
0.0       0.0       0.0
* END
```

C. NACA 64A010 Input File

```
NACA 64A010 Input File
*00000000111111111122222222223333333333444444444455555555556666666666777777777778
*23456789012345678901234567890123456789012345678901234567890123456789012345678890
*ISOLV    LAX       LAY       REXPAR    HAG       FLOATX    FLOATY    ITRMAX
0.0       0.0       1.0       0.2       0.0       0.0       0.0       399.0
*NMACH    MACH
1.0        0.21
*NALFA    ALPHA
14.0       -10 -8 -6 -4 -2 0 2 4 6 8 10 12 14 16
*LATRL    PSI       PITCHQ    ROLLQ     YAWQ      VINF
0.0       0.0       0.0       0.0       0.0       1.0
*NPAN     SREF      CBAR      XBAR      ZBAR      WSPAN
2.0       1280.0    16.84     17.456    0.0       76.0
*--------------------------------------------------------------------------------
* UPPER SANDWICH PANEL
*X1       Y1        Z1        CORD1
0.0       0.0       2.25      22.5
29.43     38.0      1.125     11.25
*NVOR     RNCV      SPC       PDL
50.0      20.0      0.81      0.0
*AINC1    ANINC2    ITS       NAP       IQUANT    ISYNT     NPP
0.0       0.0       1.0       19.0      0.0       0.0       0.0
*--------------------------------------------------------------------------------
* X/C
 0.0000
 0.5000
 0.7500
 1.2500
 2.5000
 5.0000
 7.5000
 10.0000
 15.0000
 20.0000
 30.0000
 40.0000
 50.0000
 60.0000
 70.0000
 80.0000
 90.0000
 95.0000
 100.0000
* XLE1
 0.6870
* CAMBER ROOT
 0.0000
 0.8000
 0.9700
 1.2300
 1.6900
 2.3300
 2.8100
 3.2000
 3.8100
 4.2700
 4.8400
 5.0000
 4.6800
 4.0200
 3.1300
 2.1000
 1.0600
 0.5400
 0.0200
* XLE2
 0.6870
* CAMBER TIP
 0.0000
 0.8000
 0.9700
 1.2300
 1.6900
 2.3300
 2.8100
 3.2000
 3.8100
 4.2700
 4.8400
 5.0000
 4.6800
 4.0200
 3.1300
 2.1000
 1.0600
 0.5400
 0.0200
* -------------------------------------------------------------------------------
* LOWER SANDWICH PANEL
*X1       Y1        Z1        CORD1
```

```
0.0       0.0       -2.25     22.5
29.43     38.0      -1.125    11.25
*NVOR     RNCV      SPC       PDL
50.0      20.0      0.81      0.0
*AINC1    ANINC2    ITS       NAP       IQUANT    ISYNT     NPP
0.0       0.0       -1.0      19.0      0.0       0.0       0.0
*-------------------------------------------------------------------------------
* X/C
 0.0000
 0.5000
 0.7500
 1.2500
 2.5000
 5.0000
 7.5000
 10.0000
 15.0000
 20.0000
 30.0000
 40.0000
 50.0000
 60.0000
 70.0000
 80.0000
 90.0000
 95.0000
 100.0000
* XLE1
 0.6870
* CAMBER ROOT
 0.0000
 -0.8000
 -0.9700
 -1.2300
 -1.6900
 -2.3300
 -2.8100
 -3.2000
 -3.8100
 -4.2700
 -4.8400
 -5.0000
 -4.6800
 -4.0200
 -3.1300
 -2.1000
 -1.0600
 -0.5400
 -0.0200
* XLE2
 0.6870
* CAMBER TIP
 0.0000
 -0.8000
 -0.9700
 -1.2300
 -1.6900
 -2.3300
 -2.8100
 -3.2000
 -3.8100
 -4.2700
 -4.8400
 -5.0000
 -4.6800
 -4.0200
 -3.1300
 -2.1000
 -1.0600
 -0.5400
 -0.0200
*-------------------------------------------------------------------------------
* NXS     NYS       NZS
0.0       0.0       0.0
*
* END
```

D. NACA 64A010 with NACA 63 Mean Line

```
NACA 64A010 with NACA 63
*00000000111111111122222222223333333333444444444455555555556666666666777777777778
*2345678901234567890123456789012345678901234567890123456789012345678901234567890
*ISOLV    LAX       LAY       REXPAR    HAG       FLOATX    FLOATY    ITRMAX
0.0       0.0       1.0       0.2       0.0       0.0       0.0       399.0
*NMACH    MACH
1.0        0.21
*NALFA    ALPHA
14.0       -10 -8 -6 -4 -2 0 2 4 6 8 10 12 14 16
*LATRL    PSI       PITCHQ    ROLLQ     YAWQ      VINF
0.0       0.0       0.0       0.0       0.0       1.0
*NPAN     SREF      CBAR      XBAR      ZBAR      WSPAN
2.0       1280.0    16.84     17.456    0.0       76.0
*-------------------------------------------------------------------------------
* UPPER SANDWICH PANEL
*X1       Y1        Z1        CORD1
0.0       0.0       2.25      22.5
29.43     38.0      1.125     11.25
*NVOR     RNCV      SPC       PDL
50.0      20.0      0.81      0.0
*AINC1    ANINC2    ITS       NAP       IQUANT    ISYNT     NPP
0.0       0.0       1.0       19.0      0.0       0.0       0.0
*-------------------------------------------------------------------------------
* X/C
 0.0000
 0.5000
 0.7500
 1.2500
 2.5000
 5.0000
 7.5000
 10.0000
 15.0000
 20.0000
 30.0000
 40.0000
 50.0000
 60.0000
 70.0000
 80.0000
 90.0000
 95.0000
 100.0000
* XLE1
 0.6870
* CAMBER ROOT
0.022262962
1.009109775
1.27239718
1.718427984
2.637822049
4.153853558
5.431775719
6.533393901
8.310879229
9.602482272
10.84490492
10.88235938
10.18908231
8.918141771
7.170962057
5.039094835
2.651880622
1.363824989
0.023217466
* XLE2
 0.6870
* CAMBER TIP
0.022262962
1.009109775
1.27239718
1.718427984
2.637822049
4.153853558
5.431775719
6.533393901
8.310879229
9.602482272
10.84490492
10.88235938
10.18908231
8.918141771
7.170962057
5.039094835
2.651880622
1.363824989
0.023217466
* ------------------------------------------------------------------------------
* LOWER SANDWICH PANEL
*X1       Y1        Z1        CORD1
```

```
0.0       0.0       -2.25     22.5
29.43     38.0      -1.125    11.25
*NVOR     RNCV      SPC       PDL
50.0      20.0      0.81      0.0
*AINC1    ANINC2    ITS       NAP       IQUANT    ISYNT     NPP
0.0       0.0       -1.0      19.0      0.0       0.0       0.0
*-------------------------------------------------------------------------------
* X/C
 0.0000
 0.5000
 0.7500
 1.2500
 2.5000
 5.0000
 7.5000
 10.0000
 15.0000
 20.0000
 30.0000
 40.0000
 50.0000
 60.0000
 70.0000
 80.0000
 90.0000
 95.0000
 100.0000
* XLE1
 0.6870
* CAMBER ROOT
0.022262962
-0.590890225
-0.66760282
-0.741572016
-0.742177951
-0.506146442
-0.188224281
0.133393901
0.690879229
1.062482272
1.164904921
0.882359379
0.829082311
0.878141771
0.910962057
0.839094835
0.531880622
0.283824989
-0.016782534
* XLE2
 0.6870
* CAMBER TIP
0.022262962
-0.590890225
-0.66760282
-0.741572016
-0.742177951
-0.506146442
-0.188224281
0.133393901
0.690879229
1.062482272
1.164904921
0.882359379
0.829082311
0.878141771
0.910962057
0.839094835
0.531880622
0.283824989
-0.016782534
*-------------------------------------------------------------------------------
* NXS     NYS       NZS
0.0       0.0       0.0
*
* END
```

E. Wake Survey Input

```
Swept Wing Test Case - for Wake Survey - March 21, 2021 - TTT
*
*
*ISOLV       LAX       LAY    REXPAR       HAG    FLOATX    FLOATY    ITRMAX
     0         0         1      0.10      0.00      0.00      0.00       399
*
*NMACH      MACH
     1       0.3
*
*NALPHA    ALPHA
     1       5.0
*
*LATRL       PSI    PITCHQ     ROLLQ      YAWQ      VINF
     0      0.00      0.00      0.00      0.00      1.00
*
*NPAN       SREF      CBAR      XBAR      ZBAR     WSPAN
     1   5000.00     30.00       0.0      0.00    100.00
*
*       X1        Y1        Z1     CORD1   COMMENT
    0.00      0.00      0.00    100.00    WING
   80.00     50.00      0.00     20.00
*     NVOR      RNCV       SPC       PDL
     50.00     15.00      1.00      0.00
*    AINC1     AINC2       ITS       NAP    IQUANT     ISYNT       NPP
     0.000     0.000         0         0         2         0         0
*
*
*  NXS   NYS   NZS
     1   075    25
*
* X SURVEY STATION === (ex. -10 = ahead of apex, 105 right behind trailing edge)
*23456789!123456789!123456789!23456789!
  135.0
* Y & Z SURVEY STATIONS
* YN0T     DELTAY      ZNOT    DELTAZ
*23456789!123456789!123456789!23456789!
   -075       2.0        -10.     1.
```