[illegible]

NASA Contractor Report 172200

Specification for a Program for an Iterative Aeroelastic Solution (PIAS)

Marjorie E. Manro, Michael J. Donahue
Rodney L. Dreisbach, and John E. Bussoletti

Boeing Commercial Airplane Company
Seattle, Washington

Contract NAS1-16740
December 1983

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

NASA Contractor Report 172200

Specification for a Program for an Iterative Aeroelastic Solution (PIAS)

Marjorie E. Manro, Michael J. Donahue
Rodney L. Dreisbach, and John E. Bussoletti

Boeing Commercial Airplane Company
Seattle, Washington

Prepared for
NASA Langley Research Center
under contract NAS1-16740

National Aeronautics and
Space Administration

December 1983

N84-13153#

This Page Intentionally Left Blank

CONTENTS

Page

CONTENTS (Concluded)

TABLES

FIGURES

Page

1.0 SUMMARY

The majority of existing computer programs for calculating aeroelastic loads rely on linearized aerodynamic theories. As aircraft configurations become more complex and high-technology airfoils are used, the real flow exhibits several kinds of nonlinear behavior, one of which is a separation-induced leading-edge vortex.

The aeroelastic codes that are based on linearized aerodynamic theories incorporate the structural flexibility terms directly in the closed-form solution. Generally, a closed-form solution of an aeroelastic problem using a nonlinear aerodynamic theory is difficult, if not impossible. A second approach to solving the aeroelastic problem is the alternate execution of a code to calculate aerodynamic loads on a specific shape and a code to calculate the deflected shape under load. This iterative approach has been reported by other authors as noted in section 2.0 and is the approach selected for this program specification.

The proposed computer Program for an Iterative Aeroelastic Solution (PIAS), as described herein, will provide the capability for calculating aeroelastic loads on wings, with the option of including the effect of a separation-induced leading-edge vortex. PIAS combines the use of two computer codes that have been developed by The Boeing Company; an aerodynamic module to provide the airload distribution and a finite-element structural analysis module to calculate the deformed shape of the wing under load. The aerodynamic code can calculate pressure distributions either for attached flow or for separated flow with a separation-induced leading-edge vortex. The finite-element structural analysis module will allow for modeling of simple and complex structural configurations.

Rapid advances are currently being made in computational fluid dynamics and the ability to predict nonlinear flow phenomena is progressing at an unprecedented rate. One advantage of the proposed PIAS procedure is that the aerodynamic module initially used in the solution may be easily replaced by other linear or nonlinear aerodynamic theories.

2.0 INTRODUCTION

The majority of existing computer programs for calculating aeroelastic loads rely on linearized aerodynamic theories. As aircraft configurations become more complex and high-technology airfoils are used, the real flow exhibits several kinds of nonlinear behavior, one of which is a separation-induced leading-edge vortex.

The aeroelastic codes that use linearized aerodynamic theories incorporate the structural flexibility terms directly in the solution. This approach is generally difficult to apply in aeroelastic solutions using nonlinear aerodynamic theories.

A second approach to solving the aeroelastic problem is the alternate execution of a code to calculate aerodynamic loads on a specific shape and a code to calculate the deflected shape under load. This approach has been used with good results by Chipman, Waters, and MacKenzie (ref. 1) and by Whitlow and Bennett (ref. 2) for specific types of flow phenomena. The method developed by Chipman, Waters, and MacKenzie uses only a beam structural representation, which limits its use to high aspect ratio wings with chordwise rigidity. In addition, as the authors note, small disturbance theory as used in their method may predict shock waves at incorrect locations for some supercritical Mach numbers, causing errors in the loads and in the final shape. The method of Whitlow and Bennett uses the full potential aerodynamic computer code of Jameson and Caughey (ref. 3), and describes the use of a structural representation that could be either a finite-element structural model or a beam model. In reference 2, the two wings analyzed are represented by beams, and the examples shown are all for very low angles of attack, which are typical of cruise conditions but not typical of structural design conditions.

The proposed Program for an Iterative Aeroelastic Solution (PIAS), as described herein, will provide the capability for calculating aeroelastic loads on wings, with options for including the effect of a leading-edge vortex. PIAS is designed to iterate between an aerodynamic module that provides the airload distribution (The Boeing Company's Leading-Edge Vortex Program, refs. 4 and 5) and a finite-element structural analysis module that determines the deformed shape of the wing under load (The Boeing Company's ATLAS Integrated Structural Analysis and Design System, ref. 6). The PIAS procedure will provide a versatile method for calculating aeroelastic loads and will allow for modeling of simple and very complex structural configurations. Another advantage is that it will allow for simple substitution or addition of other linear or nonlinear aerodynamic theories into the solution procedure.

2.1 LEADING-EDGE VORTEX (LEV) PROGRAM

The Leading-Edge Vortex (LEV) Program is a fully three-dimensional, inviscid, potential flow formulation that models steady subsonic (Mach number less than 1.0) flow about wing or wing-body configurations with separation-induced leading- and/or side-edge vortex flow. The compressible flow is assumed to be governed by the Prandlt-Glauert equation and a higher-order panel method approach is used to solve the problem. The basis of this program is the Free Vortex Sheet (FVS) theory developed by The Boeing Company for the NASA Langley Research Center.

Formulation of the aerodynamic solution in the LEV Program results in a problem governed by the linear subsonic-flow differential equation for velocity potential and by nonlinear boundary conditions. The boundary conditions require that the flow be parallel to the wing surface and that the free vortex sheet, springing from the leading and side edges of the wing (see fig. 1), be aligned with the local flow and support no pressure jump. The Kutta condition is imposed along all wing edges. The actual configuration is represented by quadrilateral panels on all surfaces, with quadratically varying doublet singularities distributed on the panels. The vortex core is modeled as a simple line vortex that receives vorticity from the free sheet through a connecting kinematic sheet (fed sheet). The set of nonlinear equations is solved by an iterative procedure, starting with an assumed initial geometry of the vortex sheet.

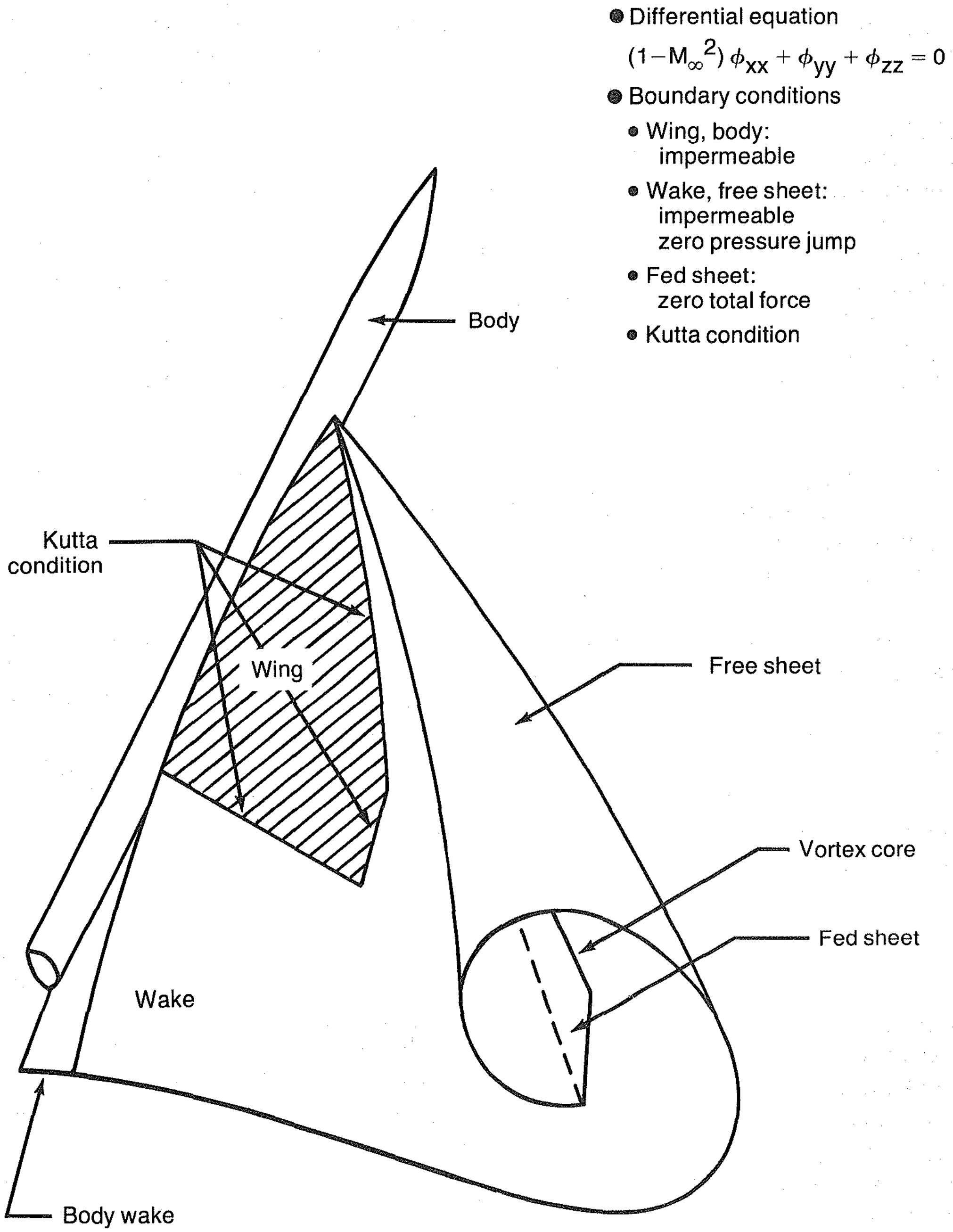
In the iterative procedure, a solution is obtained for the surface singularities that describe the flow over the configuration. The influence of these singularities produces a flow that is not, in general, aligned with the shape of the vortex sheets. Residuals are defined to measure deviations between the current position of the vortex sheet and the position of a stream surface. The geometry of the vortex sheet is then modified to minimize the residuals, and the process is repeated until convergence is achieved in the shape and location of the vortex sheets.

The LEV Program is a versatile tool for calculating flows about a class of lifting surface configurations with a separation-induced leading-edge vortex. The wing geometry may be quite general, including curved or cranked leading edges and trailing edges. The wing may also have camber and/or twist. However, if a strong, well-developed vortex does not exist in the real flow, difficulties will probably be encountered in obtaining a converged solution in the LEV Program. Cases included in this category are configurations with less than 60-degree leading-edge sweep, configurations with discontinuities in the leading edge that will promote the formation of more than one vortex system*, and solutions at angles of attack where a well-defined vortex has not yet formed in the real flow.

There is an often overlooked capability of the LEV Program that can be used to analyze attached-flow models. The setup of such models is the same as for separated-flow cases except the modeling of the vortex is omitted.

The LEV Program has been extensively evaluated, both by NASA Langley (ref. 7) and by The Boeing Company under NASA contract NAS1-15678 (ref. 8). The comparisons of data in reference 8 are for a highly swept, low-aspect-ratio wing for which a large experimental data base is available. Some examples from this evaluation are shown in appendix A. Comparisons of theory-to-experiment are shown for both a full-span vortex and a partial-span vortex. These data show that the program has the capability to predict pressure distributions for a vortex that extends over only the outboard portion of the wing as well as for a vortex that extends over the entire span. The LEV theory is generally good at predicting the spanwise location of the vortex and the peak pressure level, with the predictions being a definite improvement over those of linear codes such as FLEXSTAB (refs. 9 and 10).

*The capability for multiple vortex systems was recently added to the LEV Program but requires that the vortex systems never coalesce (NASA contract NAS1-16449, document in preparation).



- Differential equation

  $(1-M_\infty^2)\,\phi_{xx} + \phi_{yy} + \phi_{zz} = 0$
- Boundary conditions
  - Wing, body:
    impermeable
  - Wake, free sheet:
    impermeable
    zero pressure jump
  - Fed sheet:
    zero total force
  - Kutta condition

Figure 1.—Flow Model for the Three-Dimensional Leading-Edge Vortex (LEV) Program

2.2 ATLAS INTEGRATED STRUCTURAL ANALYSIS AND DESIGN SYSTEM

The ATLAS System provides an extensive set of general purpose technical modules based on the stiffness finite element method with some of its capabilities being particularly suited for the aeroelastic design of aerospace vehicles. ATLAS was chosen as the structural analysis module for PIAS because of its following attributes:

- A software architecture composed of modular functional components that is consistent with the requirements of the proposed aeroelastic solution capability

- A common executive module to provide control over a sequence of problem solution steps including logic decisions

- A data management system consistent with the solution restart requirements

- An extensive library of finite elements for effective structural and mass modeling of metallic and advanced composite structures, with either simple or complex configurations

- Procedures for interpolating data associated with one set of grid points to another set of grid points

- Options for calculating only the deformed shape of the loaded structure during the iterative aeroelastic solution process, and for calculating displacements and stresses when the solution has converged

A high level of confidence has been established in the use of ATLAS on many and varied projects of The Boeing Company.

2.3 PROPOSED AEROELASTIC SOLUTION PROCEDURE

Structural design conditions are defined by a combination of flight parameters (e.g., Mach number, equivalent airspeed) and configuration parameters (e.g., load factor, gross weight, center of gravity, mass distribution); the angle of attack is related to these parameters but is not specified. For nonlinear aerodynamic analysis methods the angle of attack, rather than the vertical load factor, is used to define the condition. This requires an iterative solution procedure in which the vertical load factor is calculated after each aerodynamic solution.

There are three levels of iteration in PIAS as illustrated in figure 2. The Execution Control Monitor (ECM) directs the overall solution process using the LEV and ATLAS codes as required. The first level of iteration is performed by the LEV Program to obtain a converged aerodynamic solution for a particular combination of geometry and angle of attack. In the next level of iteration, alternate executions of the LEV and ATLAS codes are required to obtain a converged aeroelastic solution to the proper deformed shape of the wing at each selected angle of attack. Finally, it is generally necessary to have converged solutions at three angles of attack, with the associated vertical load factors, and then to use a curve-fitting procedure to compute the angle of attack consistent with the desired load factor.

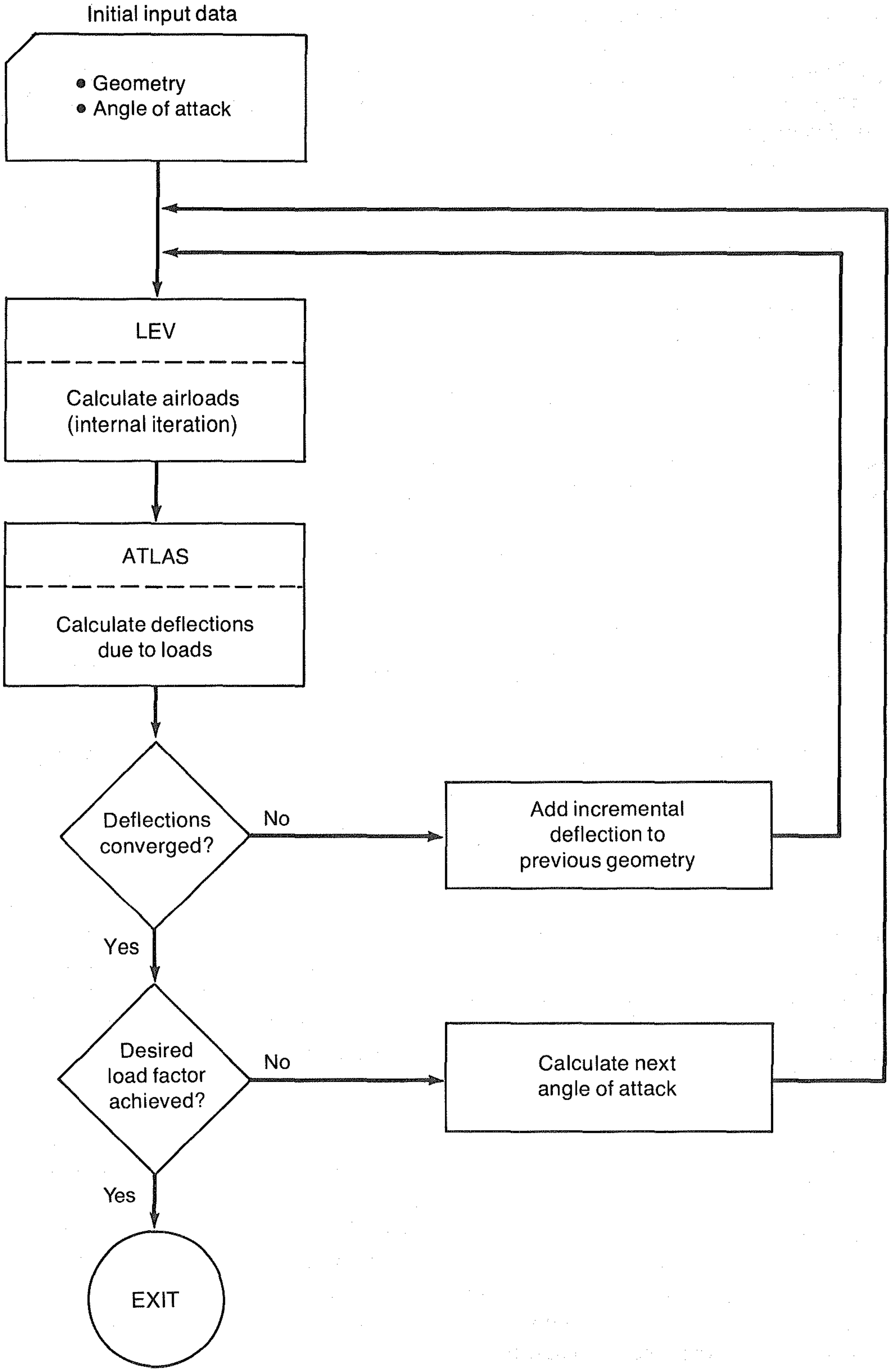

Initial input data
• Geometry
• Angle of attack
LEV
Calculate airloads (internal iteration)
ATLAS
Calculate deflections due to loads
Deflections converged?
No
Add incremental deflection to previous geometry
Yes
Desired load factor achieved?
No
Calculate next angle of attack
Yes
EXIT

Figure 2. – Iteration Levels of PIAS With the ECM Providing Execution Control and Data Management

2.4 RECOMMENDATIONS

A detailed description of the design of PIAS is presented in sections 3.0 and 4.0 of this document. The functions are modular in the planned product, with sufficient data written to files so that a restart of the solution can be made from any logical break in the sequence. It is highly recommended that PIAS be developed.

3.0 PIAS FUNCTIONAL SPECIFICATIONS

The proposed Program for an Iterative Aeroelastic Solution (PIAS) is based on the combined use of two existing computer programs. These two programs have been developed by The Boeing Company and are the Three-Dimensional Leading-Edge Vortex (LEV) Program and the ATLAS Integrated Structural Analysis and Design System. The Execution Control Monitor (ECM) will provide data transfer between the two existing codes and control the logical sequence of their execution to obtain a solution at a specified load factor. A brief description of the current capabilities of these codes, the software architecture required to provide a combined solution, and the statement of work required to implement this solution are presented in the following sections.

3.1 CURRENT TECHNICAL CAPABILITIES

The capabilities of the existing programs that will be used in PIAS are described in the following sections.

3.1.1 LEADING-EDGE VORTEX (LEV) PROGRAM

The Leading-Edge Vortex (LEV) Program is a fully three-dimensional, inviscid, potential flow formulation that models steady subsonic flow about wing or wing-body configurations with separation-induced leading- and/or side-edge vortex flow. It is assumed that the separation line, a viscous flow phenomenon, is known and that boundary layer effects are negligible. Experimental studies (refs. 11 and 12) show that the shape and strength of the principal vortex are only weakly dependent on Reynolds number, i.e., viscous effects. The compressible flow is assumed to be governed by the Prandtl-Glauert equation and a higher-order panel method approach is used to solve the problem. Either quadratic doublet or linear source distributions are used on the panels. Doublet panels are used to model thin wings, wakes, and vortex networks, whereas source panels are used to model thickness effects such as fuselages. Thick wings may be modeled with surface source networks and internal mean plane doublet networks.

The method differs from conventional higher-order panel methods by virtue of the boundary conditions that must be satisfied to model the vortex flow case. The wing must satisfy the usual zero mass flux boundary condition locally (no flow), but with the Kutta condition imposed along network edges having either vortex separation or wake flow. The Kutta condition is implemented by matching not only doublet strength, but also the component of the vorticity along the appropriate edges. This approach, in conjunction with the vortex sheet updating procedure, results in all the usual flow phenomena associated with the Kutta condition, i.e., zero pressure jump at the wing edge, finite flow at the wing edge, no flow through the vortex sheet, etc.

Wakes must satisfy the zero pressure jump boundary condition locally (no load), but with the near wake doing so to second-order accuracy in perturbation velocity quantities. (The trailing wake is formulated in the usual sense and persists far downstream). This higher-order formulation is necessary due to the large spanwise velocities induced on the wake by the primary vortices, thus rendering the linearized calculation of pressure differences invalid. A consequence of this higher-order formulation is that the wake vorticity will skew laterally in the process of satisfying the no-load boundary condition and thereby influence the Kutta condition at the wing trailing edge. Inclusion of the near wake (which is a user option) has been shown in reference 4 to significantly improve correlations between theoretical and experimental lift coefficients for delta wings over a range of aspect ratios.

The free vortex sheet must have no local flow through it (like the wing) and simultaneously support no local pressure jump across it (like the near wake). This duality in boundary conditions is balanced by the fact that both the geometry of the free sheet and its singularity strength are unknown. However, by virtue of these boundary conditions the resultant integral solution equation is nonlinear. The solution results in a boundary value problem for which both the location and strength of a portion of the boundary, the vortex sheets, is part of the solution. Alternatively, one may think of the free sheet as a design doublet surface that must be aligned with the local flow and that must satisfy the condition that the pressure differences, calculated with a second-order equation, are zero.

As in conical flow (ref. 13), the fed vortex sheet represents an approximation to the inner wrappings of the free vortex sheet and serves the purpose of terminating free sheet roll-up as well as condensing free sheet vorticity into the vortex core. The fed sheet's strength is dictated by the free sheet, whereas its size and position are determined by imposing a boundary condition "consistent with those that would be applicable to an infinitely rolled-up vortex sheet as well, namely, that the total force normal to the core be zero" (ref. 4) at each chord station. Such an approach is analogous to the conical flow formulation. In practice, free sheet roll-up is terminated at approximately one half of a revolution, near the crest of the vortex; additional quantities of free sheet roll-up do not change the solution appreciably but do increase the computational expense.

Because of the nonlinearity associated with the vortex sheets, an iterative solution procedure must be employed. The basic approach is to initiate singularity values for a fixed, initial geometry by satisfying some but not all of the boundary conditions. The total problem is then solved by one of two iterative schemes. The primary approach is a quasi-Newton scheme with controlled step size that incorporates an approximate Jacobian updating procedure to reduce computational expense. The Jacobian is recomputed every three iterations. An alternate approach is a least-squares scheme with controlled step size that incorporates a weighted free sheet panel-twist constraint. This technique was developed to help with problem cases for which it was difficult to obtain solution convergence. With this technique, solution instabilities due to local flow anomalies tend to be damped out, but the method satisfies the boundary conditions only in the least-squares sense, employs a small yet arbitrary weighting value on the penalty function, and is more expensive per iteration than the quasi-Newton method.

A description of the aerodynamic model and the LEV Program solution options is presented in the following sections.

3.1.1.1 Aerodynamic Modeling

Solution of the boundary value problem in the LEV Program is accomplished with the basic panel method of reference 4. The boundary surface is divided into networks. Each network is defined as a smooth portion of the boundary that is divided into panels and on which source and/or doublet splines are defined (accompanied by properly posed boundary conditions). The networks are assumed to be logically independent in that each network contributes as many equations as unknowns to the overall boundary value problem, i.e., networks can be added or deleted without total reformulation of the problem. The network independence is particularly useful for PIAS because it allows the aerodynamic problem to be formulated only once. Subsequent execution of the LEV Program for the attached-flow case simply requires the deletion of the vortex-related networks.

As previously stated, a network is defined as a portion of the boundary surface on which a distribution of source or doublet strengths is specified, together with properly posed analysis (Neumann) or design (Dirichlet) boundary conditions. The true surface is assumed to have continuous position, slope, and curvature. Discontinuities in these quantities, therefore, are limited to network edges. Each network is defined by the coordinates of an array of grid points consisting of M rows of points each containing N points or columns. A triangular-shaped network is produced when one side collapses into a single point, i.e., one grid point belongs to all the rows or columns of the network.

Figure 3 illustrates the numbering scheme for grid points, network corners, and network sides. The outward direction of the network is defined by the vector cross product NxM. This outward direction is important for a source network since its outward side must always bound the flow. The numbering sequence of the network edges is also important in defining the geometry because specific orientations are required to ensure proper continuity across network boundaries.

Proper edge matching is dependent on the abutment of adjacent networks. The network grid points must be coincident along adjacent edges and must abut along complete edges. This rule places some restrictions on the modeling, e.g., a thin wing, which could be one network for a full-span vortex, must be at least two networks for a partial-span vortex. In addition, there are rules for proper orientation of the networks, both as independent entities and relative to each other. A maximum of 20 networks may be used to define a model for the LEV Program. The available types of networks showing the use and applied boundary conditions for each are shown in table 1; locations of the discrete values of singularity strength and the control points at which boundary conditions are applied are shown in figure 4. The maximum number of singularities allowed for a configuration depends on whether the solution is for attached flow or separated flow with a quasi-Newton or least-squares method. These limits are shown in table 2. Detailed descriptions of networks and paneling requirements are presented in reference 5.

Wing Paneling.—Two general styles of paneling can be used for the wing: conical and streamwise. With conical paneling, the primary panel edges radiate from the apex and bend as necessary to accommodate the planform. The other panel edges are usually at constant body stations. With streamwise paneling, the primary panel edges are parallel to the centerline of the vehicle. The other edges are arbitrary, but typically they either coincide with the hinge lines or are at constant-percent chord lines. Doublet/analysis networks are used to define the camber surface of the wing; source/analysis networks are used to define the upper and lower surfaces if thickness is represented (see table 1).

Vortex Paneling.—The leading-edge vortex is represented by the combination of a free sheet, a fed sheet (terminating in a line vortex), and associated wakes. Doublet networks are used and, even though the geometry is allowed to change during the solution, the initial geometry must be defined. This geometry can be generated by the program based on the conical solution of Smith (ref. 13), in which case the size and position of the free and fed sheets are dependent on the

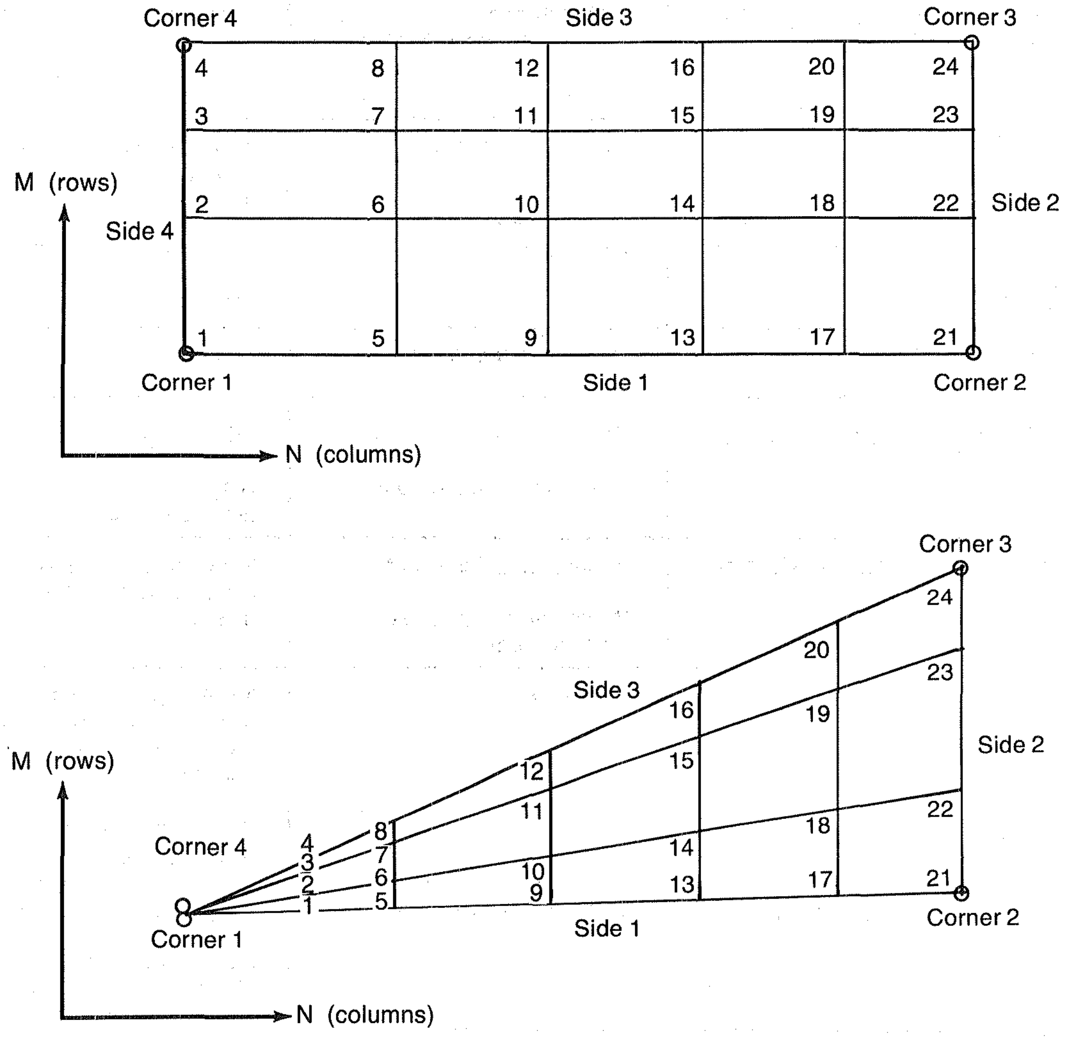


Figure 3. – Numbering Sequence for Grid Points, Network Corners, and Network Sides for LEV Program

Table 1. – LEV Program Network Types, Uses, and Boundary Conditions

Network type	Use	Boundary conditions
NT = 1 Source/analysis	Exterior surface of thick wings Exterior surface of bodies	Impermeable
NT = 2 Doublet/analysis	Camber surface of wing	Kutta condition imposed on appropriate leading, side, and trailing edges.
NT = 4 Doublet/design no. 1	Free sheet	ΔC_p = 0 (second order formula) Stream surface
NT = 6 Doublet/design no. 2	Near wake (behind wing)	ΔC_p = 0 (second order formula)
NT = 8 Doublet/analysis wake no. 1	Simple wake Carryover lifting system	ΔC_p = 0 (linearized formula) Singularity strength constant along columns
NT = 10 Constant doublet/wake no. 4	Wake behind fed sheet Wake behind carryover lifting system	Singularity strength constant throughout, therefore, produces no shed vorticity
NT = 14 Doublet/design wake no. 2	Fed sheet	Total force induced on fed sheet and vortex core by the rest of the configuration is parallel to the core Singularity strength constant along columns
NT = 16 Doublet/analysis wake no. 3	Wake behind free sheet	ΔC_p = 0 (linearized formula) Singularity strength constant along columns

Note: See figure 4 for locations of discrete values of singularity strength and locations of control points at which boundary conditions are applied.

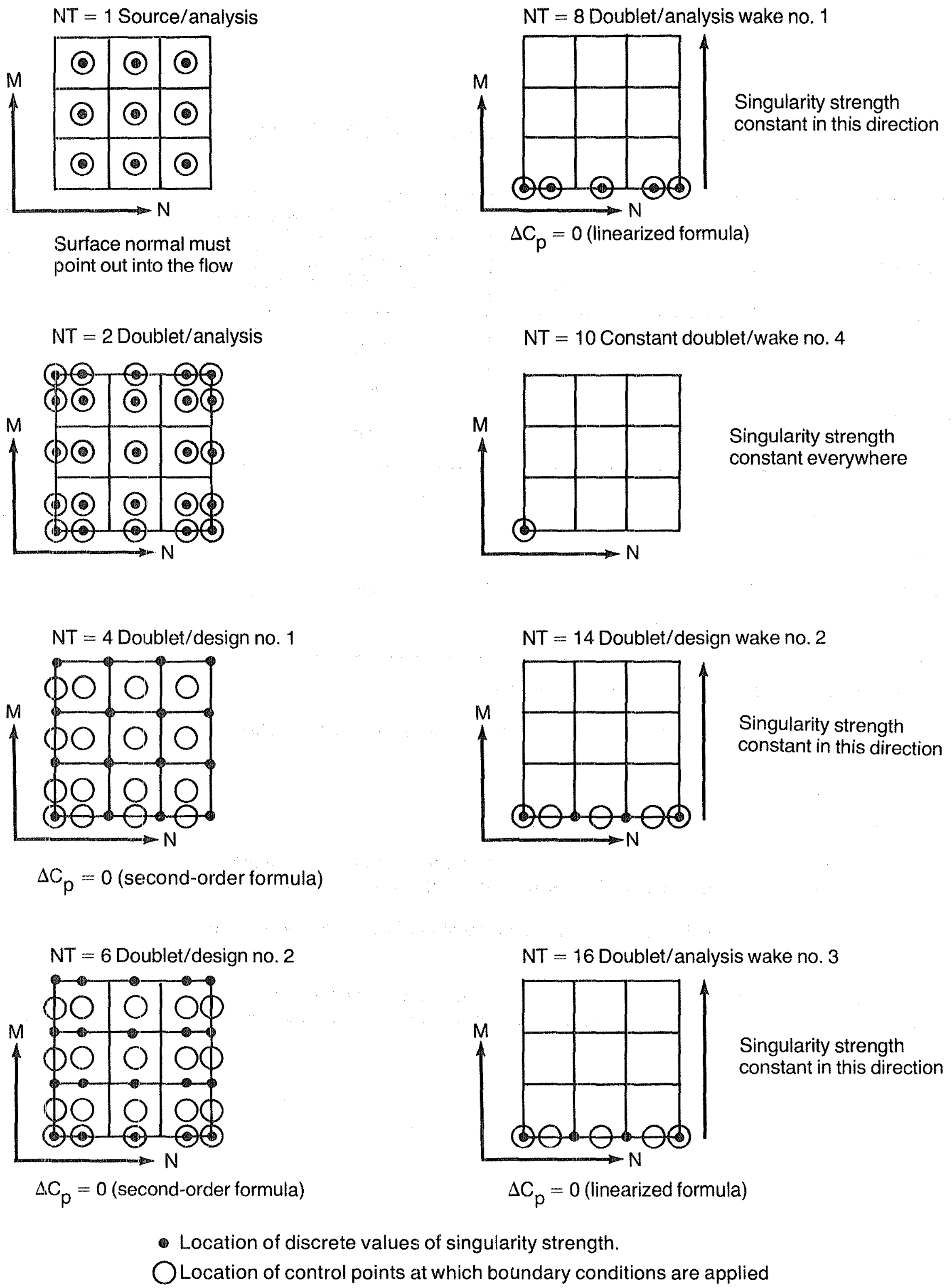

NT = 1 Source/analysis
M
N
Surface normal must point out into the flow
NT = 8 Doublet/analysis wake no. 1
M
N
Singularity strength constant in this direction
ΔCp = 0 (linearized formula)
NT = 2 Doublet/analysis
M
N
NT = 10 Constant doublet/wake no. 4
M
N
Singularity strength constant everywhere
NT = 4 Doublet/design no. 1
M
N
ΔCp = 0 (second-order formula)
NT = 14 Doublet/design wake no. 2
M
N
Singularity strength constant in this direction
NT = 6 Doublet/design no. 2
M
N
ΔCp = 0 (second-order formula)
NT = 16 Doublet/analysis wake no. 3
M
N
Singularity strength constant in this direction
ΔCp = 0 (linearized formula)
● Location of discrete values of singularity strength.
○ Location of control points at which boundary conditions are applied

Figure 4.— Locations of Discrete Values of Singularity Strength and of the Control Points at Which Boundary Conditions Are Applied for LEV Networks

Table 2. – Maximum Size Capabilities of the LEV Program*

Parameter	Attached-flow	Separated-flow	
	$SUBITERATION	Quasi-Newton $ITERATION	Least-squares $LEAST SQUARES
NF	750	750	750
NG	—	300	300
NH	—	50	50
NG + NH	—	350	—
2 x NG + NH	—	—	650
NG + NH + NF	—	500	—
2 x NG + NH + NF	—	—	500

The maximum number of networks (NNETT) allowed is 20.

The maximum number of additional points for which flow parameters can be evaluated is 500.

Definitions:

NF is the number of singluarity strength parameters.

NG is the number of free sheet panels; therefore, the number of panel orientation angles and also, the number of twist function equations for the least-squares separated-flow solution.

NH is the number of fed sheet panels; therefore, the number of geometry parameters.

*Discrepancies in maximum size shown in this table are due to decisions made when the code was initially written. Consistent size limitations will be implemented as part of this proposed program development.

parameter "a" that is calculated using the angle of attack and the leading-edge sweep (fig. 5). The vortex may be moved closer to, or farther from, the wing by modifying the calculated value of "a". This is accomplished by adding the user-specified parameter "Δa" and/or by rotating the vortex about the attachment edge.* In general, convergence is faster if the size of the initial vortex is too large rather than too small. In addition to these options, the geometry of the vortex networks may be specified directly by the user although the method just described is generally satisfactory.

Wakes.—Wake networks are included to ensure satisfaction of the boundary conditions. Each type of wake network is designed to attach to other specific types of networks. The locations of the discrete values of singularity strength, the locations of the control points at which boundary conditions are applied, and the boundary conditions on these doublet networks are compatible with the networks to which they are attached. The wake network that uses the second-order formula for the zero pressure jump boundary condition is placed directly behind the wing to enforce the Kutta condition on the wing trailing edge. Additional benefits of this wake are discussed in reference 7.

Body.—To complete the configuration, the body may be included. The body surface is modeled as one or more source networks. A carryover lifting system (doublet network) must be defined inboard of the wing doublet networks to provide continuity of the doublet networks to the configuration centerline.

3.1.1.2 Solution Options

The solution options of the LEV Program allow for both attached-flow and separated-flow problems. As stated earlier, the option for an attached-flow solution is often overlooked, even though the formulation is the same as that in PANAIR (ref. 14).

In the separated-flow option, the extent of the vortex along the wing span is a user option. In PIAS, the vortex location will be determined by using an algorithm that relates the leading-edge suction to the parabolic nose drag. When the vortex networks are placed only where vortex flow actually occurs, the results for cases with a partial-span vortex are as good as for cases with a full-span vortex. This overall capability allows for the analysis of a single configuration with either attached flow or separated flow (a separation-induced leading-edge vortex over all or part of the span). Thus, one program can be used throughout the flight regime even for configurations on which a separation-induced vortex does not always exist.

3.1.2 ATLAS INTEGRATED STRUCTURAL ANALYSIS AND DESIGN SYSTEM

ATLAS is a system of integrated computer program modules with common executive and data management facilities for the analysis and design of general structural configurations. ATLAS can be used to perform a variety of structural analysis and design tasks ranging from the small single-discipline problem to the very large multidiscipline problem. The system provides an

*The capability to rotate the vortex was added as a temporary modification during the LEV Program evaluation under NASA contract NAS1-15678. This provided an effective way of moving the vortex core away from the wing surface without making the vortex excessively large. It is planned to implement this feature in the LEV Program as part of this proposed program development (see sec. 3.3.2.1).

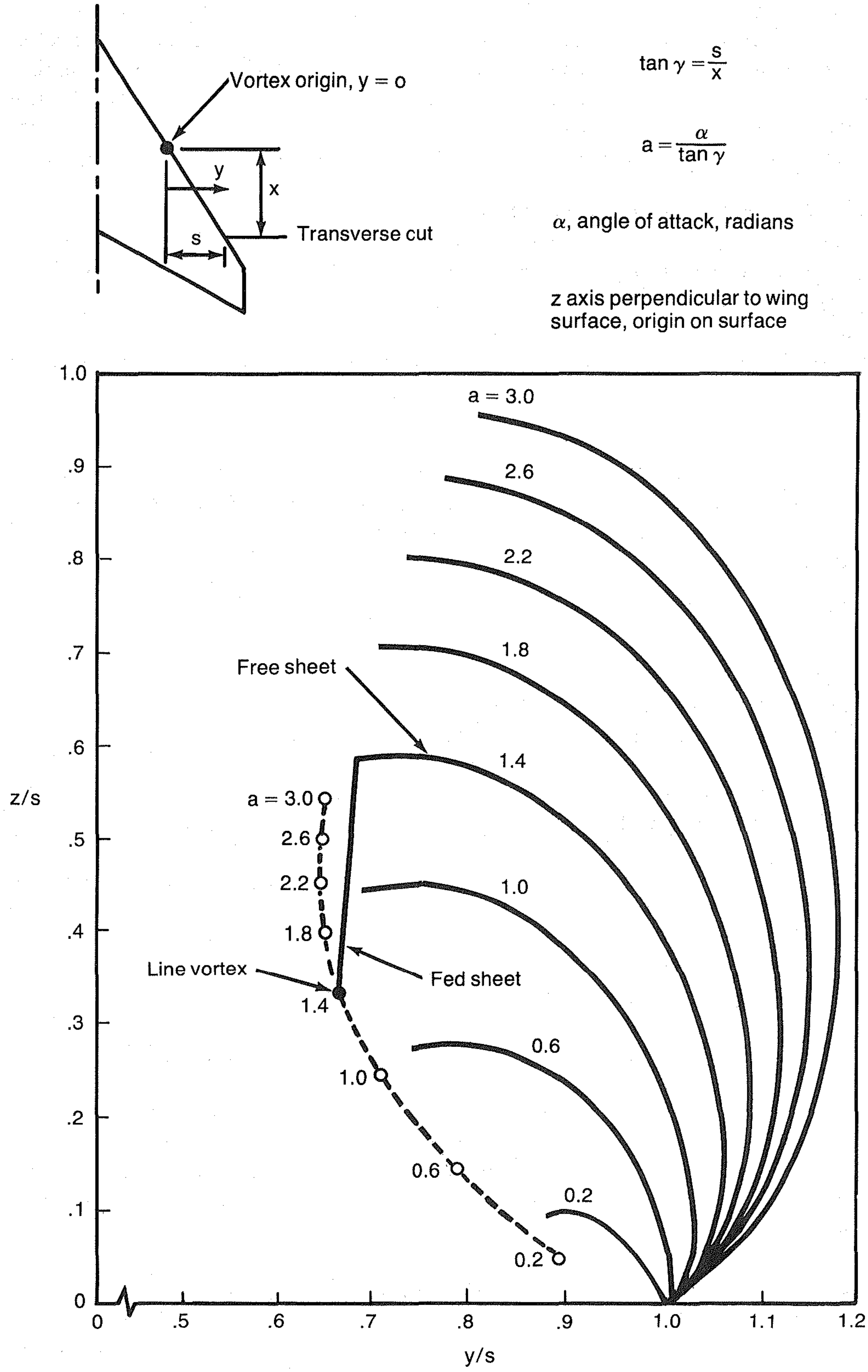


Figure 5.– Initial Free Sheet Geometry and Size of Fed Sheet for Various Values of "a" (ref. 13); Transverse Cuts Normal to the Wing Surface

extensive set of technical modules including stiffness, stress, buckling, loads, mass, substructuring, strength design, unsteady aerodynamics, vibration, and flutter analyses. ATLAS is based on the stiffness finite-element structural analysis method in which the physical properties of the structure are represented by a finite number of idealized elements.

A common executive language is used to control the sequence of program module executions, including the selection of module options needed to perform a particular analysis. This selective execution of modules gives the user considerable versatility in solving a wide range of problems. Communication of data between program modules is performed by the ATLAS data base manager, which also can be used to manipulate data for communication between ATLAS and external programs.

Input data are written in an engineering oriented language using a free field format. The preprocessors minimize the amount of data and flowtime that are required for problem definition and verification by providing default values for input data, routines to generate detailed data from simple user input, and quality checks on the input data. Postprocessors allow selected input and calculated data to be extracted, manipulated, and displayed.

All of the ATLAS input and execution data are described in detail in reference 6. The ATLAS structural and mass modeling capabilities, and one of the ATLAS data interpolation procedures pertinent to the proposed aeroelastic solution program are summarized below.

3.1.2.1 Structural Modeling

The ATLAS structural model definition data include: a) structural geometry defined as nodal points, b) the finite elements with their geometric, elastic material, and mass density properties, and c) the boundary conditions that identify which nodal freedoms are constrained and unconstrained. A single structural model can be defined by as many as 4095 nodes and 32,767 elements. In addition, any number of substructure component models can be interacted to define a total structural model by using the ATLAS multilevel substructured analysis capabilities. Thus, large problems with many degrees of freedom can be handled.

Options are provided to define nodal points using any combination of local rectangular, cylindrical, and spherical coordinate systems. Any combination of these types of coordinate systems can also be used for the output (analysis) reference frames for the nodes. Analysis reference frames are those coordinate systems relative to which the kinematics of the nodes are expressed, e.g., the calculated displacements. Any number of local reference frames can be used.

Capabilities are provided for automatically generating grids of nodes and elements, based on a minimal amount of input data. Capabilities for defining element cross section property tables further reduce the amount of input data required when multiple elements have the same properties.

The ATLAS library of structural finite elements is summarized in table 3 in terms of the geometry and the number of nodes used to define each type of element. Characteristics of these elements are shown in table 4 according to:

o Material and geometric properties

o Elastic and geometric (stress stiffening) stiffnesses

o Types of element loading

o Type of elemental mass distribution

As can be noted from tables 3 and 4, both metal and layered-composite material structures can be modeled by this wide selection of finite elements. Rigid offsets of the elastic and mass c.g. axes from the structural nodes are provided for the beam element, as well as rigid offsets for the plate-like elements. Constant-strain triangle and quadrilateral membrane plate elements, linear and higher-order isoparametric quadrilateral membrane plates, a shear panel, plus membrane/bending plates for metal and composite structures are provided, with the stacking sequence of the composite laminas being accounted for.

The built-up SPAR, COVER, and CCOVER elements provide effective means to model wing-like structures for preliminary design studies, wherein the change in distance between the skin surfaces can be ignored. The SPAR element is composed of a shear plate and two linearly tapered axial rods with continuous shear transfer between the shear plate and the rods. The COVER and CCOVER elements are made up of two plate elements separated by rigid posts.

3.1.2.2 Mass Modeling

ATLAS provides capabilities for modeling structural, nonstructural, fuel, and payload mass distributions. The overall mass distribution condition for an aerospace vehicle can be defined as any combination of these distributions.

Structural mass is defined by the structural finite element model (sec. 3.1.2.1). Variable factors, which are functions of the element geometry (e.g., thickness and area), can be used to provide the model-to-actual design correlation required to evaluate the "as built" structure.

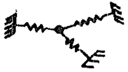
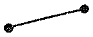
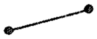
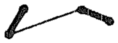
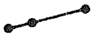
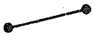
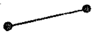
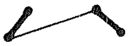
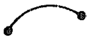
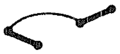
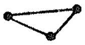
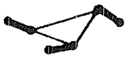
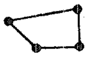
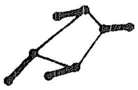
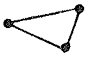
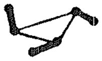
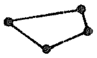
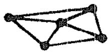
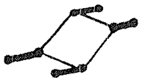
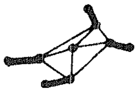
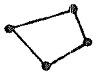
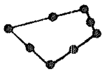
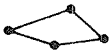
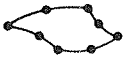
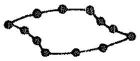
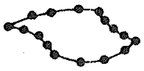
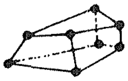
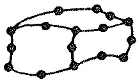
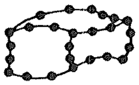
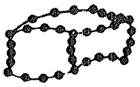
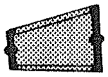
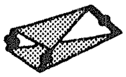
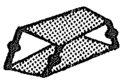
Nonstructural mass, including fuel and payload, is defined by mass finite elements or by concentrated masses at nodes. The ATLAS library of mass finite elements, with density and weight property input options, is shown in table 5.

Calculated mass properties include the weight, the center of gravity, and the mass moments of inertia. These properties are calculated for each load condition for the total modeled vehicle, as well as for any identified subsets. These subsets generally include the various fuel and payload distributions, the primary and secondary structure, and the nonstructural items, but they may also include user-defined subsets of the structural elements.

Options are provided for generating airload panel mass matrices as a function of load case, lumped diagonal and nondiagonal mass matrices, as well as options for generating Guyan-reduced mass matrices that are based on the elemental and concentrated masses.

Concentrated masses are typically used to model a component or region of structure that is concentrated at a point such as a nacelle, landing gear, or external store. Convenient and cost effective capabilities are provided specifically for defining and managing multiple fuel and payload distributions.

Table 3. – ATLAS Library of Stiffness Finite Elements

Element		Number of nodes ①		Geometric Shape ②
Type	Name	Structural	Auxiliary	
Scalar	SCALAR	1	(2)	
Rod/ Beam	ROD	2	0	
	BEAM	2	(1)	
		2	(2), (3)	
	SROD	3	0	
	SBAR	2	(1)	
Tube/Pipe	PIPE (straight)	2	(1)	
		2	(2), (3)	
	PIPE (curved)	2	1	
		2	3	
Plate	PLATE and CPLATE (mem-brane)	3	0	
		3	3	
		4	0	
		4	4	
	GPLATE (mem-brane and/or bending)	3	0	
		3	3	
		4	0	
		4	1	
		4	4	
		4	5	
Plate (Con-tinued)	SPLATE (shear only)	4	0	
		4	4	
	QIPLATE (mem-brane) and QCPLATE (mem-brane/ bending)	4	0	
		8	0	
		12	0	
		16	0	
Solid	BRICK	8	(3)	
		20	(3)	
		32	(3)	
		44	(3)	
Built-up	SPAR	2 (mid-surface nodes)	0	
	COVER and CCOVER	3 (mid-surface nodes)	0	
		4 (mid-surface nodes)	0	

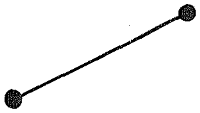
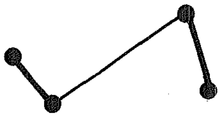
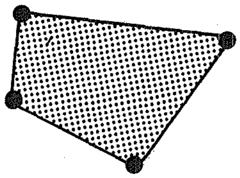
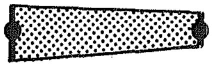
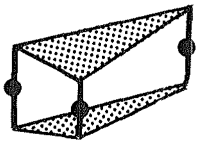
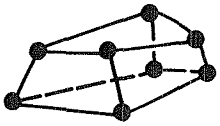
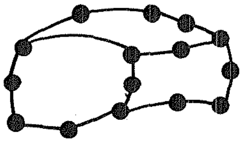
① Auxiliary nodes denoted by (N) are optionally used for orientating the element or its material axes.

② Rigid element components (offsets and rigid connections) are illustrated by /

Table 4. – Characteristics of the ATLAS Stiffness Finite Elements

Element name	Properties						Stiffness						Loads			Mass	
	Orthotropic material	Composite material	Orthotropic smeared stiffness	Element offset	Warped element	Variable cross section	Axial	Torsional	Shear	Moment	Membrane	Geometric	Distributed	Thermal	Inertia	Lumped	Consistent
SCALAR							●	●	●	●							
ROD						●	●					●	●	●	●	●	
BEAM				●		●	●	●	●	●		●	●	●	●	●	
SROD						●	●						●	●	●	●	
SBAR							●		●				●	●	●	●	
PIPE				●			●	●	●	●			●	●	●	●	
PLATE	●		●	●	●				●		●	●	●	●	●	●	
CPLATE	●	●	●	●	●				●		●		●	●	●	●	
GPLATE	●			●	●	●			●	●	●	●	●	●	●	●	
SPLATE					●				●				●		●	●	
QIPLATE	●				●	●			●		●		●		●		●
QCPLATE	●	●	●		●				●	●	●		●		●	●	
BRICK	●					●	●		●			●	●	●	●		●
SPAR						●	●		●	●	●		●	●	●	●	
COVER	●		●		●				●	●	●		●	●	●	●	
CCOVER	●	●	●		●				●	●	●		●	●	●	●	

Table 5. – ATLAS Library of Mass Finite Elements

Element name	Number of nodes	Geometric shape ②	Option	Property input						
				Density	Thickness	Cross-sectional areas	Area moments of inertia	Weight	Weight moments of inertia	Center of gravity
SCALAR	1		a					●	●	
ROD	2		a	●		●				
			b			①		●		
			c					●		●
BEAM	2-5		a	●		●	①			
			b			①	①	●		
			c					●		●
PLATE	3-9		a	●	●					
			b		①			●		
			c					●		●
SPAR	2		a	●	●	●				
			b		①	①		●		
COVER	3-9		a	●	●					
			b		①			●		
			c					●		●
BRICK	8		a	●						
	9-20		b					●		

① Default value may be used or property may be input

② Rigid element components (offsets and rigid connections) are illustrated by:

Note:
For some elements, multiple combinations of property definitions are shown. Only those properties marked in a single row are inputs for a selected option a, b, or c.

The overall strategy is that of modeling the vehicle gross weight as a three-dimensional array of basic mass elements. This array of elements is then cut into polyhedral regions corresponding to the user-defined mass lumping regions or airload panels. Fuel mass properties are calculated by automatically dividing the user-defined fuel tanks into a series of mass volume elements and summing the properties of those elements that are needed to obtain the desired fuel level.

Airload panel mass properties are computed by summing the properties of the elements or parts of elements within the defined airload panel. This procedure is applied separately to the primary structure, secondary structure, fuel, payload, and other nonstructural element subsets. The result is a series of airload panel and lumped mass matrices corresponding to these element subsets.

The mass analysis approach is based on a finite mass element representation for all of the mass distribution condition components. This concept is consistent with the finite element structural analysis approach, and is convenient from a user point of view in that once the mass elements are defined, changes in the airload panel geometry or in the location of the nodes retained for lumping masses require no changes to the basic mass model.

3.1.2.3 ATLAS Surface Spline Interpolation

The geometry grid input to the LEV Program defines the aerodynamic panel corner points; the centers of these aerodynamic panels define the output grid for pressure coefficients. The structural grid defined for ATLAS is representative of the structural configuration of the wing. The scalar data associated with one of these sets of grid points can be interpolated to the other set of grid points by using routines already available in the ATLAS system.

Of the various types of interpolation methods provided by the INTERPOLATION Processor of the ATLAS system, it is proposed that the surface spline method be used in PIAS. In this method, scalar data associated with one set of points are interpolated to a second set of points by using an interpolation function that is based on the small-deflection bending equation for a circular, uniform thickness, thin plate of infinite radius simply-supported along its periphery (ref. 15).

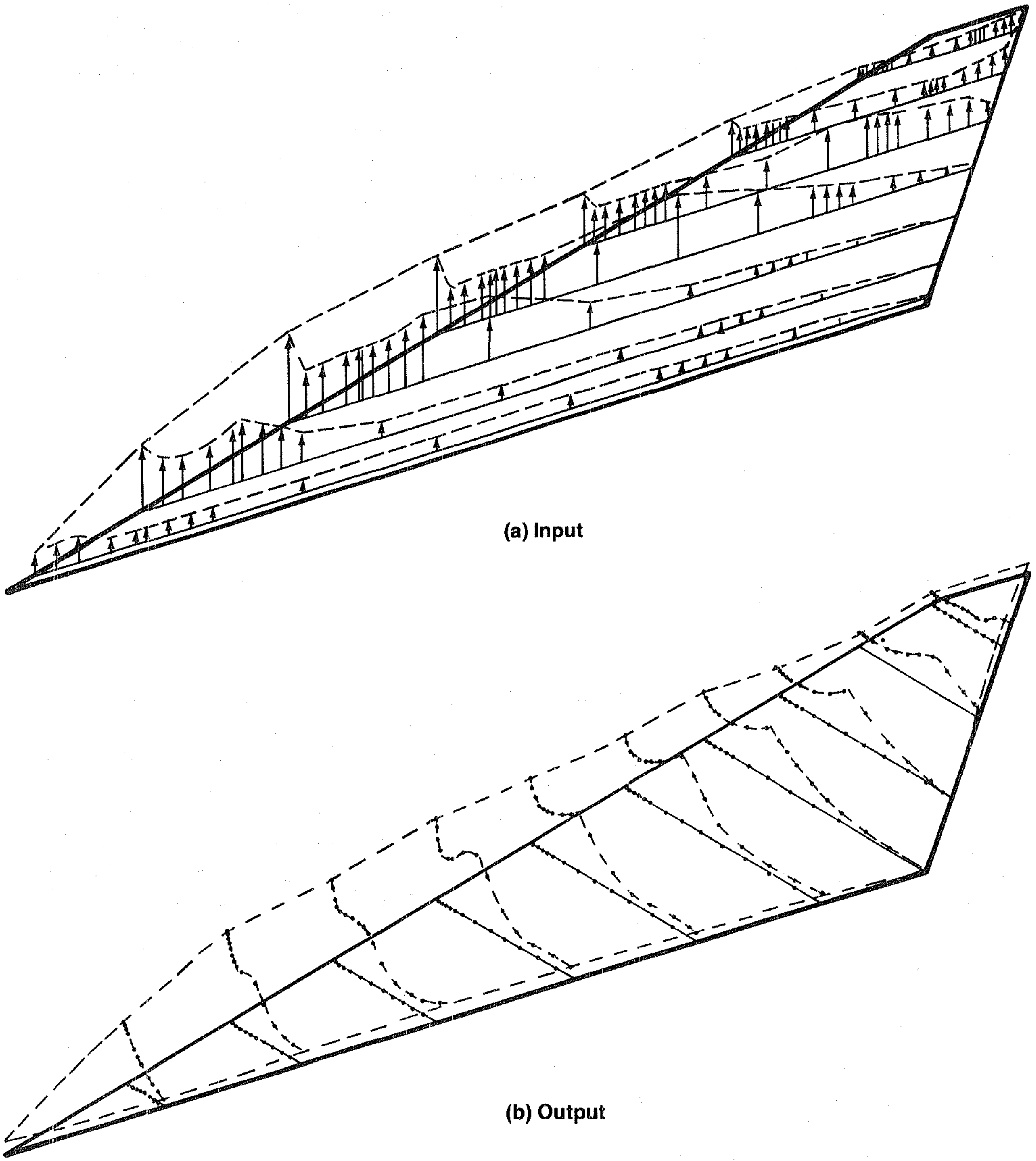
The surface spline interpolation process is performed by ATLAS in two steps:

1. The surface spline interpolation coefficients are generated as a function of the input grid points and the scalar data associated with the input grid points.

2. Corresponding data values for the output grid points are calculated by using the interpolation coefficients and the output grid points.

The input grid points must not all lie along a straight line, but should define a planar or nearly planar surface. For best results, the input points should be distributed so that extrapolation to the output points is not required. This interpolation method works best when the input grid points are uniformly distributed. For cases where the input distribution is less dense in one direction than the other, the coordinates are transformed so that the spacing is more uniform, thus improving the results. The output grid points may have an arbitrary distribution.

An example of the results obtained using the surface spline interpolation method for experimental pressure data is shown in figure 6. The experimental data are for a low-aspect ratio



(a) Input

(b) Output

Figure 6. – Example of the ATLAS Surface Spline Interpolation Method Using an Experimental Upper Surface Pressure Distribution on an Arrow Wing

wing at an angle of attack of 16 degrees and Mach = 0.40. The distribution of points in the chordwise direction is quite dense relative to the distribution in the spanwise direction (pressure points are at arrows). The data are interpolated to a grid consisting of rows that extend to the wing tip and that are located at constant body stations. The results are very good. It should be noted that the most outboard streamwise station for the input data is at 0.93 of the semispan.

3.2 SOFTWARE ARCHITECTURE

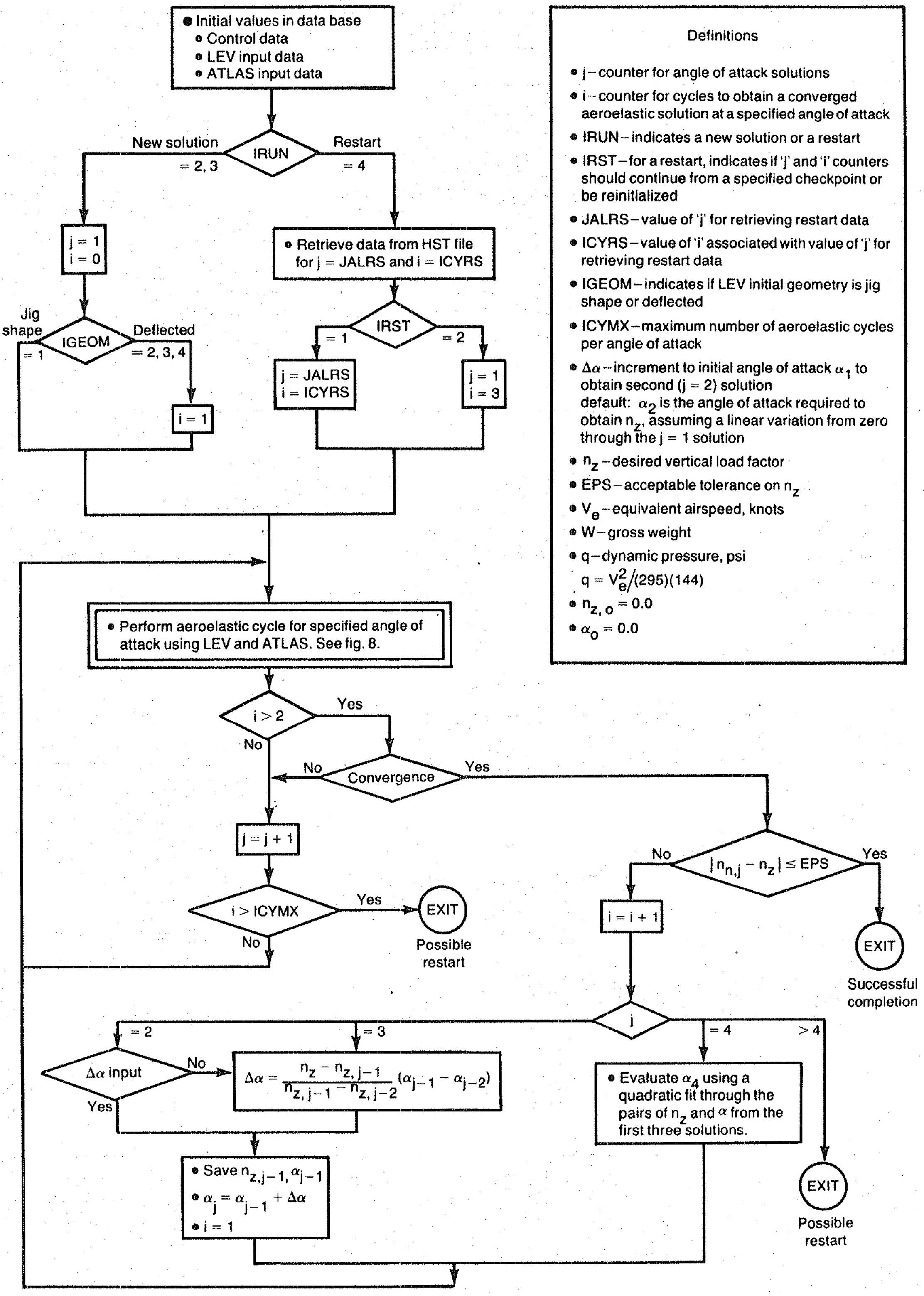
The Execution Control Monitor (ECM) of PIAS will be an ATLAS Control Program that implements an aeroelastic solution algorithm, by using the LEV Program and selected modules of the ATLAS system. The LEV Program will be used to provide pressure distributions on a fixed wing geometry either for attached flow, or for separated flow of the separation-induced leading-edge vortex type. The input geometry data define the wing and body, and an initial representation of the vortex sheet and wakes for the separated-flow solution. During the LEV iterative process for the separated-flow solution, only the vortex sheet and associated wakes are allowed to move.

Pressures are output from the LEV Program at aerodynamic panel centroids. The ATLAS surface spline interpolation method will be used to obtain pressures at the nodes that define the structural finite element model. ATLAS modules will be used to calculate the nodal deformations of the wing due to the pressure loading and associated inertia loads. Using the nodal deformation, the ATLAS surface spline interpolation method will again be used to obtain the coordinates of the aerodynamic panel corner points for input to the LEV Program. The alternate execution of these two codes will be repeated until convergence on a deflected shape is obtained.

The ECM contains an outer loop pertaining to the angle of attack calculations required to obtain a solution at a specified load factor. This logic flow is illustrated in figure 7. For each angle of attack, an inner loop performs an iterative solution using the LEV Program to calculate panel pressures on the rigid wing shape and the ATLAS system to compute the resulting deflections at the structural nodes. For each aeroelastic cycle, the ECM invokes the LEV Program, executes the required ATLAS processing modules, provides the data management and other functions required to interface these two programs, and controls the sequence of these tasks. This logic flow is illustrated in figure 8. The areas outlined with solid lines indicate processes and decisions, whereas those areas bordered by dashed lines identify data flow.

The framework provided by this proposed program can be used to incorporate other aerodynamic programs. The LEV Program currently included in PIAS can either be replaced, or options can be added to provide multiple aerodynamic theories in PIAS. This open-ended design of the proposed software architecture will allow for the use of different types of existing theories, or the use of new theories as they are developed and validated.

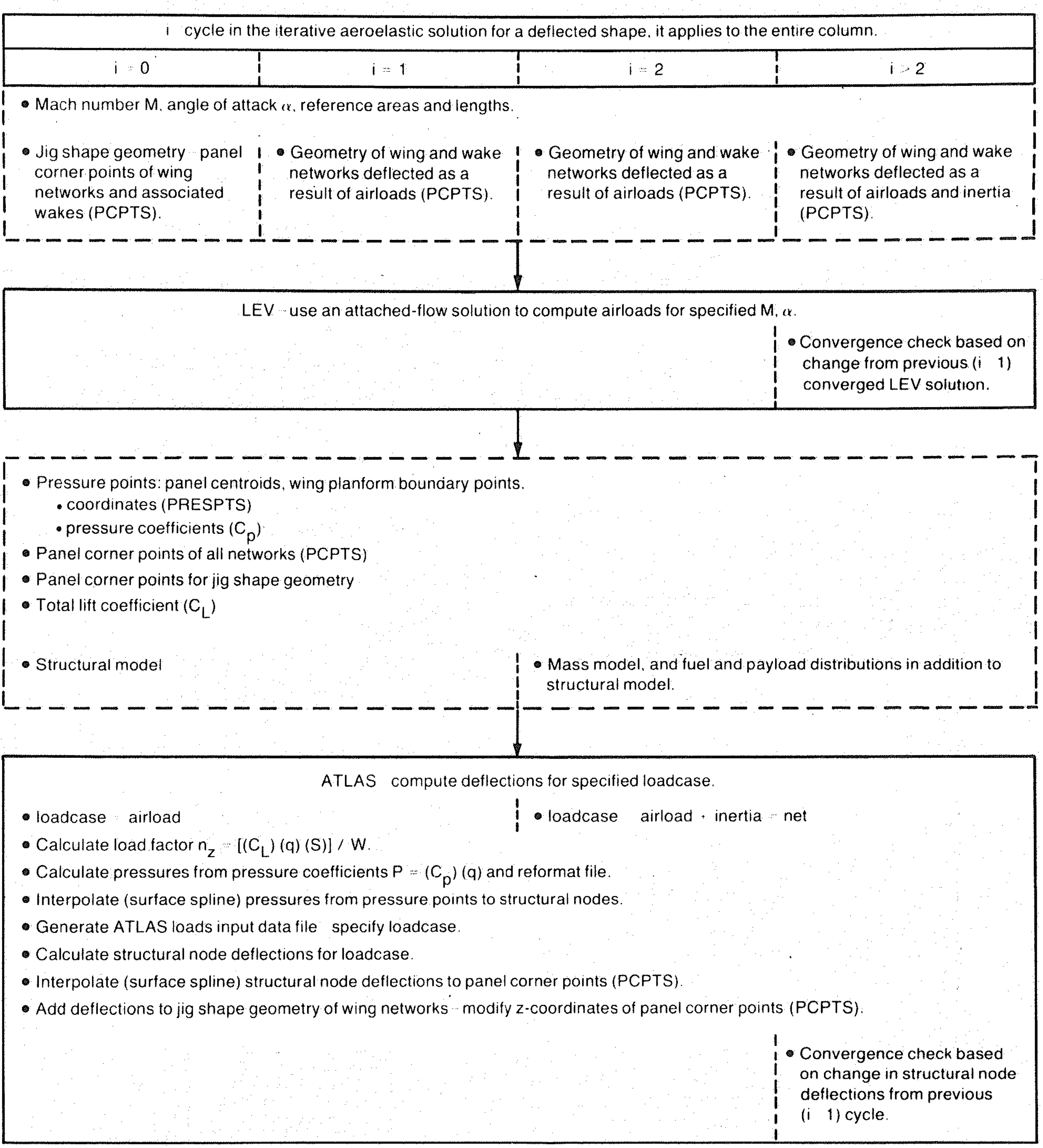
ATLAS execution directives, SNARK statements, and FORTRAN language statements will be used in constructing the ECM. SNARK is an engineering oriented language that provides standard matrix and scalar mathematical capabilities, as well as management functions for matrices and computer core allocation (ref. 16). The ECM will use the ATLAS library routine, LODAREC, for preprocessing its input data. This routine reads and decodes free field input data records, identifying the type of each data item in a data record for validity checks.


● Initial values in data base
● Control data
● LEV input data
● ATLAS input data
New solution
= 2, 3
IRUN
Restart
= 4
j = 1
i = 0
● Retrieve data from HST file for j = JALRS and i = ICYRS
Jig shape
= 1
IGEOM
Deflected
= 2, 3, 4
i = 1
IRST
= 1
= 2
j = JALRS
i = ICYRS
j = 1
i = 3
● Perform aeroelastic cycle for specified angle of attack using LEV and ATLAS. See fig. 8.
i > 2
Yes
No
No
Convergence
Yes
j = j + 1
i > ICYMX
Yes
EXIT
Possible restart
No
No
$|n_{n,j} - n_z| \leq$ EPS
Yes
i = i + 1
EXIT
Successful completion
j
= 2
= 3
= 4
> 4
Δα input
No
$\Delta\alpha = \frac{n_z - n_{z,j-1}}{n_{z,j-1} - n_{z,j-2}} (\alpha_{j-1} - \alpha_{j-2})$
Yes
● Evaluate α_4 using a quadratic fit through the pairs of n_z and α from the first three solutions.
● Save $n_{z,j-1}, \alpha_{j-1}$
● $\alpha_j = \alpha_{j-1} + \Delta\alpha$
● i = 1
EXIT
Possible restart

Definitions

- j – counter for angle of attack solutions
- i – counter for cycles to obtain a converged aeroelastic solution at a specified angle of attack
- IRUN – indicates a new solution or a restart
- IRST – for a restart, indicates if 'j' and 'i' counters should continue from a specified checkpoint or be reinitialized
- JALRS – value of 'j' for retrieving restart data
- ICYRS – value of 'i' associated with value of 'j' for retrieving restart data
- IGEOM – indicates if LEV initial geometry is jig shape or deflected
- ICYMX – maximum number of aeroelastic cycles per angle of attack
- $\Delta\alpha$ – increment to initial angle of attack α_1 to obtain second (j = 2) solution
  default: α_2 is the angle of attack required to obtain n_z, assuming a linear variation from zero through the j = 1 solution
- n_z – desired vertical load factor
- EPS – acceptable tolerance on n_z
- V_e – equivalent airspeed, knots
- W – gross weight
- q – dynamic pressure, psi
  $q = V_e^2/(295)(144)$
- $n_{z,o} = 0.0$
- $\alpha_o = 0.0$

Figure 7. – ECM Logic to Obtain a Converged Solution at a Desired Load Factor



i = cycle in the iterative aeroelastic solution for a deflected shape, it applies to the entire column.

i = 0	i = 1	i = 2	i > 2

- Mach number M, angle of attack α, reference areas and lengths.

i = 0	i = 1	i = 2	i > 2
• Jig shape geometry – panel corner points of wing networks and associated wakes (PCPTS).	• Geometry of wing and wake networks deflected as a result of airloads (PCPTS).	• Geometry of wing and wake networks deflected as a result of airloads (PCPTS).	• Geometry of wing and wake networks deflected as a result of airloads and inertia (PCPTS).

LEV – use an attached-flow solution to compute airloads for specified M, α.

- Convergence check based on change from previous (i – 1) converged LEV solution.

- Pressure points: panel centroids, wing planform boundary points.
  - coordinates (PRESPTS)
  - pressure coefficients (C_p)
- Panel corner points of all networks (PCPTS)
- Panel corner points for jig shape geometry
- Total lift coefficient (C_L)

- Structural model
- Mass model, and fuel and payload distributions in addition to structural model.

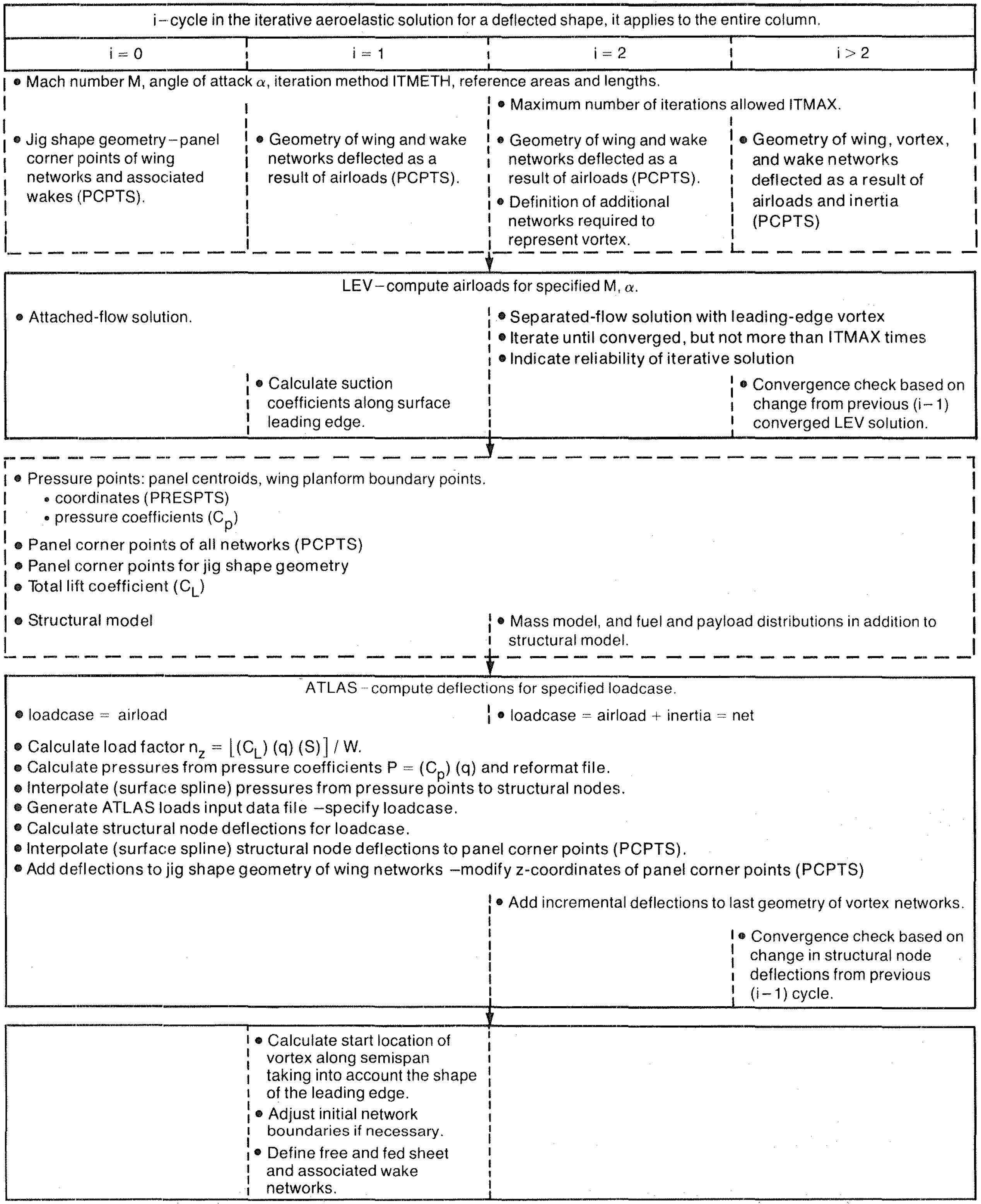
ATLAS – compute deflections for specified loadcase.

- loadcase = airload
- loadcase = airload + inertia = net
- Calculate load factor $n_z = [(C_L)(q)(S)] / W$.
- Calculate pressures from pressure coefficients $P = (C_p)(q)$ and reformat file.
- Interpolate (surface spline) pressures from pressure points to structural nodes.
- Generate ATLAS loads input data file – specify loadcase.
- Calculate structural node deflections for loadcase.
- Interpolate (surface spline) structural node deflections to panel corner points (PCPTS).
- Add deflections to jig shape geometry of wing networks – modify z-coordinates of panel corner points (PCPTS).
- Convergence check based on change in structural node deflections from previous (i – 1) cycle.

Note: Comments that apply to more than one column (i.e., multiple values of i) are not repeated: vertical dashed lines are used to define the limits of their applicability.

(a) Attached-Flow Solution

Figure 8.—ECM Logic to Perform an Aeroelastic Cycle for One Angle of Attack



i – cycle in the iterative aeroelastic solution for a deflected shape, it applies to the entire column.

i = 0	i = 1	i = 2	i > 2
• Mach number M, angle of attack α, iteration method ITMETH, reference areas and lengths.			
		• Maximum number of iterations allowed ITMAX.	
• Jig shape geometry – panel corner points of wing networks and associated wakes (PCPTS).	• Geometry of wing and wake networks deflected as a result of airloads (PCPTS).	• Geometry of wing and wake networks deflected as a result of airloads (PCPTS). • Definition of additional networks required to represent vortex.	• Geometry of wing, vortex, and wake networks deflected as a result of airloads and inertia (PCPTS)

LEV – compute airloads for specified M, α.

- Attached-flow solution.
- Separated-flow solution with leading-edge vortex
- Iterate until converged, but not more than ITMAX times
- Indicate reliability of iterative solution
- Calculate suction coefficients along surface leading edge.
- Convergence check based on change from previous (i – 1) converged LEV solution.

- Pressure points: panel centroids, wing planform boundary points.
  - coordinates (PRESPTS)
  - pressure coefficients (C_p)
- Panel corner points of all networks (PCPTS)
- Panel corner points for jig shape geometry
- Total lift coefficient (C_L)
- Structural model
- Mass model, and fuel and payload distributions in addition to structural model.

ATLAS – compute deflections for specified loadcase.

- loadcase = airload
- loadcase = airload + inertia = net
- Calculate load factor $n_z = \lfloor (C_L)\,(q)\,(S) \rfloor / W$.
- Calculate pressures from pressure coefficients $P = (C_p)\,(q)$ and reformat file.
- Interpolate (surface spline) pressures from pressure points to structural nodes.
- Generate ATLAS loads input data file – specify loadcase.
- Calculate structural node deflections for loadcase.
- Interpolate (surface spline) structural node deflections to panel corner points (PCPTS).
- Add deflections to jig shape geometry of wing networks – modify z-coordinates of panel corner points (PCPTS)
- Add incremental deflections to last geometry of vortex networks.
- Convergence check based on change in structural node deflections from previous (i – 1) cycle.

- Calculate start location of vortex along semispan taking into account the shape of the leading edge.
- Adjust initial network boundaries if necessary.
- Define free and fed sheet and associated wake networks.

Note: Comments that apply to more than one column (i.e., multiple values of i) are not repeated; vertical dashed lines are used to define the limits of their applicability.

(b) Separated-Flow Solution

Figure 8. – (Concluded)

The ECM will be processed by the ATLAS system precompilers to translate the ATLAS and SNARK language statements into equivalent FORTRAN code defining a primary overlay. This primary overlay will be compiled, and the results appended to a dummy relocatable main overlay. This relocatable program will then be loaded to create the absolute Control Program.

3.3 STATEMENT OF WORK

This section describes the analytical functions that must be developed to perform an iterative aeroelastic solution with leading-edge vortex flow. The data management requirements are discussed in section 4.0.

The Execution Control Monitor (ECM) will direct the proper sequence of the aeroelastic solution steps and will provide the following additional functions:

- Manage data (see section 4.0)
- Determine the spanwise location of the apex of the vortex
- Calculate the load factor
- Calculate the angle of attack
- Determine if convergence to a deformed shape has been achieved

The necessary modifications to the LEV Program are grouped into the following two categories:

- Improvements to the solution convergence characteristics
  - The use of the Marquardt algorithm in the iterative process
  - An improved procedure for calculating the total residual
  - A method to rotate the initial vortex about its attachment edge
  - A revised application of the twist function used in the least-squares solution method to reduce the twist of the free and fed sheets
  - Evaluation of the calculated results to stop the LEV iterative solution when it has converged or when it appears that it will not converge

- Requirements for PIAS
  - Calculate the leading-edge suction coefficients when an attached-flow solution is requested
  - Add data to current output files

The only change to ATLAS that is currently planned is related to the significant differences between the aerodynamic and structural grids. The pressures will be integrated over a local area to obtain equivalent nodal loads.

3.3.1 EXECUTION CONTROL MONITOR (ECM)

While iterating between the LEV and ATLAS codes to obtain the deflected wing shape, the ECM will define the vortex networks for the initial separated-flow cycle and calculate a vertical load factor for each cycle. For each outer level iteration to the desired load factor, the ECM will compute a new angle of attack for input to the LEV Program.

3.3.1.1 Determination of Vortex Start Location

A portion of the output from the LEV Program for an attached-flow solution will be the distribution of the leading-edge suction coefficient along the leading edge (ref. 17). This theoretical leading-edge suction coefficient is used in the leading-edge suction analogy of Polhamus for predicting the gross effects of leading-edge vortices on sharp, thin, plate-type wind tunnel model wings (ref. 18). For these wings, the vortex exists on the full wingspan.

Experimental data have shown that wing thickness affects the growth and development of leading-edge vortices. Furthermore, the effects of wing thickness on the leading-edge vortex development depend on the bluntness of the leading edge. A procedure to account for these effects was developed by R. M. Kulfan of The Boeing Company under contract NAS1-15678 and is discussed in references 8, 19, and 20. In these cases, the vortex generally starts at the wingtip and moves inboard as the angle of attack increases.

Based on Kulfan's concept for a round-nose airfoil, the leading-edge vortex starts at the most inboard point along the leading edge at which the leading-edge suction is greater than the parabolic nose drag (the local nose is approximated by a parabola). For typical wings, the values of leading-edge suction and parabolic nose drag vary smoothly along the wing semispan. The parabolic nose drag is a function of the ratio of the leading-edge radius to the streamwise chord length of the airfoil section. The user provides this ratio, as a function of wing semispan, as part of the ECM input data.

The free and fed sheet doublet/design networks (Dirichlet boundary conditions) that represent the vortex must always attach to the leading edge and/or side edge of the doublet/analysis networks (Neumann boundary conditions) that represent the camber surface of the wing. These edges of the camber surface define the vortex separation line for any thick wing as well. In addition, adjacent doublet networks must abut along complete edges with the corner points coinciding. For the proposed aeroelastic solution, the user-defined grid points will not be changed to match the origin of the vortex. The apex of the free and fed sheets representing the vortex will be located at the panel corner point on the leading edge that is next outboard of the vortex origin as determined above. To satisfy the abutment requirement, it may be necessary to move one or more rows (or columns) of corner points from one network to another. If there is a potential for a partial-span vortex, the user must model the configuration with an adequate number of networks on the wing as new networks will not be added by the program. A pair of free and fed sheet networks will be attached to each wing doublet network with a leading or side edge outboard of the vortex apex. The actual panel corner points for the initial position of the free and fed sheet networks and associated wake networks will be generated by the LEV Program.

3.3.1.2 Calculation of Vertical Load Factor

One of the parameters that defines a design condition is the vertical load factor. However, the load factor is not a parameter associated with either the input or the output of the LEV Program. For a solution at a specified angle of attack, the output from the LEV Program does include the pressure distribution, and the total lift and lift per network obtained by summing these pressures. Based on the total lift on the configuration, the vertical load factor $n_{z,j}$ for each cycle will be calculated by the ECM as follows:

$$n_{z,j} = \frac{C_L\, q\, S}{W} \tag{1}$$

The following notation is used:

C_L, total lift coefficient calculated by the LEV Program

q, dynamic pressure (psi); $q = V_e^2/(295.0)(144.0)$

V_e, equivalent airspeed (knots)

S, reference area (in^2)

W, gross weight (lb)

3.3.1.3 Calculation of a New Angle of Attack

The final aeroelastic solution must be at the desired vertical load factor n_z. To accomplish this task, the proposed procedure will obtain converged solutions at three angles of attack. Because the load factor as a function of angle of attack is nonlinear, a curve fit of the data for the first three solutions will be used to determine the angle of attack corresponding to the desired load factor for the fourth aeroelastic solution. At each angle of attack, the solution continues until the deflection under load has converged to the user-defined tolerance.

The user specifies the initial angle of attack α_1 as part of the LEV input data. The second angle of attack can be determined in one of two ways. First, the user can specify an incremental angle (ECM input data) to be added to the first angle of attack as shown in figure 9(a). Second, the program default procedure can be used, in which the second angle of attack is determined by linear interpolation. This method is based on the assumption that a load factor of zero is obtained at zero angle of attack (identified by a zero $[0]$ subscript) as shown in figure 9(b). Using this method, the new angle of attack is calculated using equations (2) and (3).

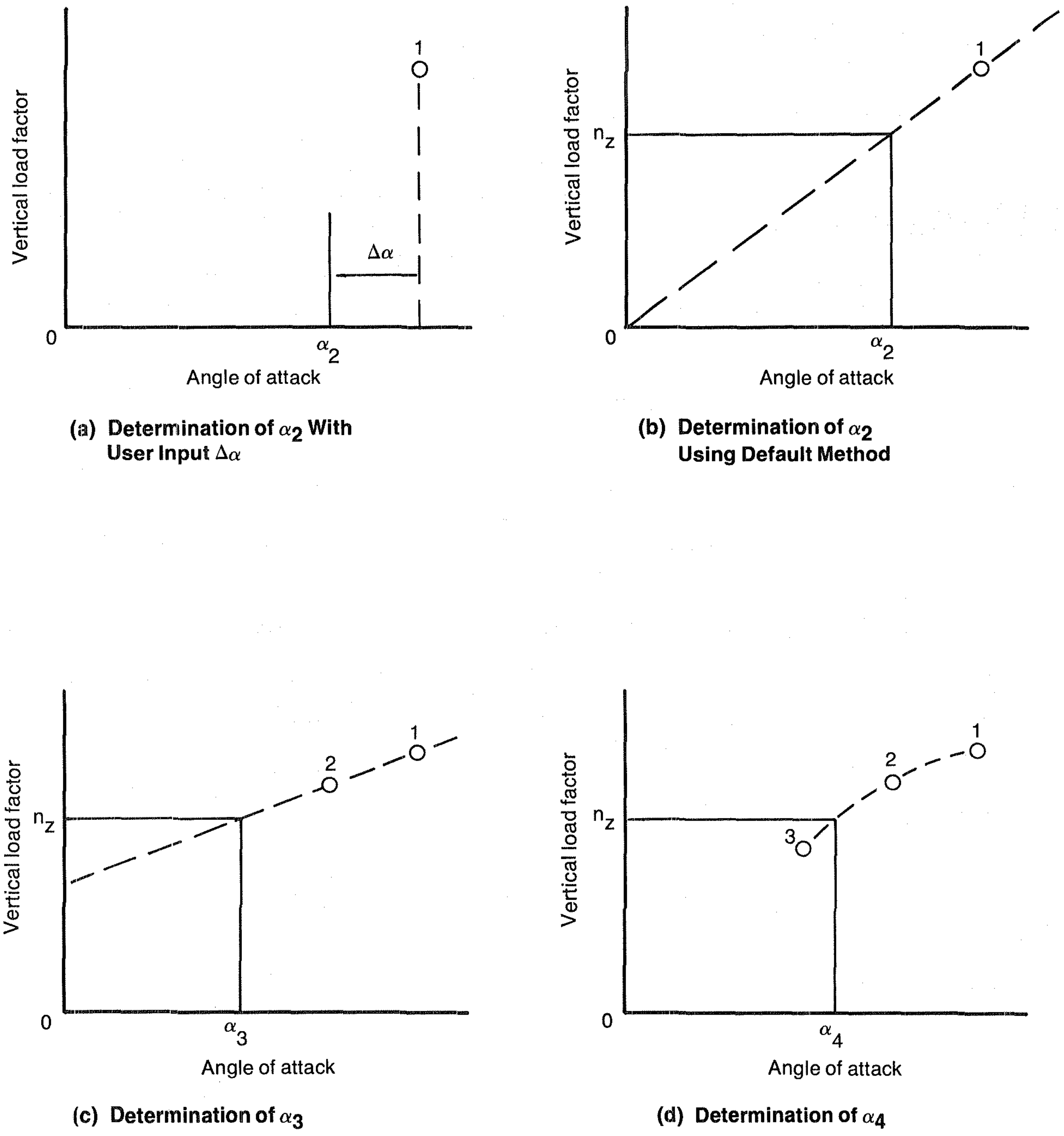
$$\alpha_j = \alpha_{j-1} + \Delta\alpha_j \tag{2}$$

$$\Delta\alpha_j = \frac{n_z - n_{z,j-1}}{n_{z,j-1} - n_{z,j-2}}\left(\alpha_{j-1} - \alpha_{j-2}\right) \tag{3}$$

The third angle of attack is always determined by linear interpolation of the first two solutions (fig. 9(c)) by using equations (2) and (3).

To evaluate the final angle of attack α_4, a quadratic curve is fit through the results from the first three solutions as shown in figure 9(d):

$$n_{z,j} = a + b\,\alpha_j + c\,\alpha_j^2, \quad j = 1, 2, 3 \tag{4}$$



(a) **Determination of α_2 With User Input $\Delta\alpha$**

(b) **Determination of α_2 Using Default Method**

(c) **Determination of α_3**

(d) **Determination of α_4**

Figure 9.—Procedure for Determining the Angle of Attack in the Iterative Aeroelastic Solution

where the values of $n_{z,j}$ are obtained from equation (1) for each angle of attack α. The following equation is then solved for α_4:

$$n_z = a + b\,\alpha_4 + c\,\alpha_4^2 \tag{5}$$

3.3.1.4 Convergence to a Deformed Shape

As shown in figure 2, the LEV and ATLAS programs are executed as many times as necessary to obtain a deformed shape for a specific angle of attack solution. Each cycle produces a set of deflections at the aerodynamic panel corner points defining the wing surface. Convergence will be achieved when the difference between the deflections at each point for two consecutive cycles (i and i + 1) is less than a user-specified percentage of the deflection for cycle "i" at that point. The user also provides a limit on the number of cycles, so that if the solution is not converging, the execution will stop, and the data will be saved for a possible solution restart.

3.3.2 LEADING-EDGE VORTEX (LEV) PROGRAM

The required modifications to the LEV Program (see sec. 3.1.1) are grouped into two categories. The first group includes modifications that will improve the solution convergence characteristics. These are essential if the aeroelastic solution is to be economically feasible, but will also be of benefit to users of the standalone LEV Program. The second group of modifications are required to use the LEV Program as part of PIAS.

3.3.2.1 Improvements to the LEV Program Solution Convergence Characteristics

Marquardt Algorithm.—The Marquardt algorithm (ref. 21) is a technique that allows the method of solution to change adaptively during the iterative solution of a system of nonlinear equations. Initially (when the residuals are large) the algorithm uses a method of steepest descent. As the residuals approach a minimum, the method reverts to a Gauss-Newton procedure. This allows a solution with minimum residuals to be attained more rapidly. When the residuals are large, the primary emphasis is placed on the effect of each panel on itself. As the residuals become smaller, this effect and the effect of each panel on the other panels become equally important. This procedure for solving nonlinear least-squares problems is described below.

Marquardt's procedure for solving nonlinear least-squares problems of the form:

Minimize $\|f(x)\|^2$ with respect to x.

computes a sequence of estimates $\{x_n\}$ for convergence to a minimum point x* by solving a sequence of constrained linear least squares problems. The basic approach is an extension of the Gauss-Newton minimization procedure (ref. 22, ch. 6) in which the function f(x), whose norm is to be minimized, is approximated by a one term Taylor's series expanded about a linearization point x_n:

$$f(x) \sim f(x_n) + A_n(x - x_n).$$

Here, A_n is the Jacobian matrix of f at x_n. Because this approximation can only be regarded as accurate for x in some neighborhood of x_n, $\| x - x_n \| \leq c_n$, the next estimate of the minimum point is generated by solving the constrained linear least squares problem.

Minimize $\| f(x_n) + A_n(x - x_n) \|^2$ with respect to x

subject to the constraint: $\| x - x_n \| = c_n$

The solution x_{n+1} of this problem is given by

$$x_{n+1} = x_n - (A_n^T A_n + \lambda_n I)^{-1} A_n^T f(x_n)$$

where λ_n is a parameter that depends in a complicated way on the value of c_n. For $\lambda_n = 0$, this is precisely the Gauss-Newton procedure, whereas in the limit of large λ_n ($\lambda_n \gg \| A_n^T A_n \|$), a variant of the method of steepest descent is obtained:

$$x_{n+1} \simeq x_n - (1/\lambda_n) A_n^T f(x_n)$$

In practice, the value of λ_n is determined by a heuristic procedure that decreases λ toward zero as long as $\| f(x_{n+1}) \|$ decreases, and increases λ whenever $\| f(x_{n+1}) \|$ fails to be smaller than $\| f(x_n) \|$. This procedure for step size limitation is more cost-effective than the step size limited Newton method currently used in the LEV Program:

$$x_{n+1} = x_n - \rho_n A_n^{-1} f(x_n)$$

where $\rho_n \in [0,1]$ is chosen to be as large as possible (in the interval [0,1]) subject to the restriction that $\| f(x_{n+1}) \| \leq \| f(x_n) \|$.

Procedure for Calculating the Total Residual.—The residuals in the LEV Program measure the extent to which boundary conditions at control points fail to be satisfied. The total residual for the shape and location of the vortex sheet is defined as:

$$R = \sum_n \sum_c \left[G_{nc}(\Lambda, \Theta, \lambda, \nu) + H_{nc}(\Lambda, \Theta, \lambda, \nu) \right]$$

where:

Λ, singularity strength parameters

Θ, free sheet orientation angles

λ, vortex system scale factor

ν, fed sheet scale factor

In this equation, the functions G_{nc} relate to the satisfaction of the impermeability boundary condition on the free sheet, and the functions H_{nc} relate to the satisfaction of the global boundary condition of zero net force acting on the fed sheet and line vortex. The current LEV nonlinear solution process minimizes this residual. This is not necessarily an optimum (or even satisfactory) definition of the residual for the equations. Note that as the number of control points is increased or as the number of networks is increased, the total residual (assuming a uniform distribution of values) must also increase, even though there is no essential change in the geometry of the configuration. This implies that the value of the total residual is not as meaningful a quantity as is desired. For example, the number of control points in a problem might be doubled. This would require the residual at each control point in the more finely paneled case to be reduced by a factor of two over those in the coarsely defined case to obtain the same level of total residual. It is proposed that the residuals be defined in a manner that makes them less sensitive to the number of panels on the free and fed sheets. Panel area normalization, as described below, accomplishes such a definition.

The value of the boundary condition at each control point is assumed to be constant over the entire panel. Thus, the residual at each control point is weighted by the area of the panel. These residuals are summed and normalized by dividing by the total area of the free and fed sheet networks. That is, the total residual is defined by:

$$R = \sum_{n} \sum_{c} \left[G_{nc}(\Lambda, \Theta, \lambda, \nu) * A_{nc} + H_{nc}(\Lambda, \Theta, \lambda, \nu) * A_{nc} \right] / \sum_{n} \sum_{c} \left[A_{nc} \right]$$

The sums are over the networks that define the free and fed sheets in the problem and over the control points on these networks. This definition of the residual has the advantage that a uniform distribution of the residuals over all the control points on the free and fed sheets will give the same value of total residual regardless of the number of panels that are used to define the sheets.

Incorporation of this normalization scheme into the LEV Program will be straightforward since the G and H terms are currently computed separately and are added to form the total residual.

User-Specified Rotation of the Vortex.—The better the initial position of the vortex, the fewer cycles should be required to obtain convergence. The current options for defining the initial geometry of the free and fed sheet networks representing the vortex sheet are:

- Direct input of all the panel corner points defining the free and fed sheet networks.

- Use of the conical solution of Smith (ref. 13) to determine the size and position of the sheet. This method uses a calculated parameter that is based on the angle of attack and the leading-edge sweep angle. The user may increment this parameter if it is believed that a larger or smaller vortex sheet will be more appropriate.

During the LEV Program evaluation under NASA contract NAS1-15678, it was established that failure of the program to converge for a wing modeled to include thickness was caused by the intersection of the vortex sheet and the upper surface of the wing. As the size of the vortex sheet appeared to be adequately large, the vortex was rotated outward about the attachment line. With this change to the initial geometry, the solution did converge. This modification was a temporary option of the LEV Program at the time of the evaluation; it is proposed that this option be made a permanent part of the LEV Program.

Twist Function Used in Least-Squares Method.—Local flow anomalies on the free sheet are caused when the first control point on the wing is farther from the apex than the first control points on the vortex sheet (typical of streamwise wing paneling). In this case, flow through the wing near the apex is not prevented and consequently the free and fed sheet control points in the first row encounter an environment different from those farther aft. To satisfy the boundary conditions on the free and fed sheet, the vortex system near the apex is moved substantially inboard, causing errors in wing pressures at the first wing control point. Even though some error is expected near the apex, the flow anomaly there tends to propagate to other areas of the sheet. This situation is caused by the attempt to satisfy exactly the boundary conditions on the free and fed sheets. Limiting panel twist is a simple method for damping this instability and is the method chosen for use in the LEV Program with the least-squares solution option. Although this method is designed to keep the individual panels flat, the overall sheet can become quite warped. A different approach, recently developed by Forrester Johnson of The Boeing Company, was formulated to minimize the change in orientation angle between adjacent panels. This approach is more successful in minimizing the kinking of the free sheet and it is proposed that this method be implemented in the LEV Program.

LEV Solution Convergence.—As a standalone program, the LEV Program is executed for a user-specified number of iterations with the final geometry and singularity parameters saved on a file for possible solution restart. The user then examines the solution convergence characteristics, including the total and individual residuals, the step size, the fraction of Newton step taken, and the total forces and moments. Based on this examination, it is determined that one of the following situations has resulted:

- The solution has converged.

- The solution is converging and should be restarted for a specific number of iterations.

- The solution will not converge in a reasonable number of additional iterations, therefore, some reformulation of the problem is required.

It is necessary to develop a procedure for evaluating the convergence characteristics during the LEV solution process. Logic will be developed for the LEV Program to evaluate the data and make the decisions that the user currently makes using these results. The previously discussed modifications will enhance this evaluation.

Code will be added to determine, after each iteration, whether the solution has converged to the desired accuracy, whether progress is still being made toward convergence, or whether little progress toward convergence is being made so that user intervention is required. There is currently a summary of the solution convergence characteristics printed for each execution. It is proposed that a procedure be developed to evaluate these characteristics for each iteration of the LEV solution process. This will help to reduce the overall cost of execution. In PIAS, this is essential to determine whether the aeroelastic cycle should be continued.

Document Guidelines for Aerodynamic Modeling.—The LEV Program provides many user choices in formulating the problem. Available guidelines for modeling the configuration to calculate airload distributions are not adequate. The evaluation of this computer program under NASA contract NAS1-15678 provided some guidelines for the aerodynamic modeling, however, it was evident that some additional study was required to understand the effects of certain input parameters. The extensive use of the LEV Program by the NASA Langley Research Center (LaRC) personnel has been reported in many papers and documents (refs. 7, 23, and 24). A combination of the experience gained from both the NASA studies and those of The Boeing Company would provide a much needed guide for the cost-effective use of the LEV Program. It is recommended that a composite of that experience be a part of the final documentation for PIAS, and that additional studies be made if necessary.

3.3.2.2 LEV Program Modifications Required for PIAS

Calculate Leading-edge Suction.—In order to use the method developed by Kulfan (see sec. 3.3.1.1) to determine the point along the leading edge at which a vortex starts, it is necessary to calculate the leading-edge suction. The distribution of leading-edge suction is available when an attached-flow solution is requested ($SUBITERATION with no vortex sheets defined). A study performed by The Boeing Company for the NASA Langley Research Center under contract NAS1-16740 identified a method for calculating leading-edge suction (ref. 17). This method utilizes the current capability in the LEV Program for evaluating flow field properties at points other than the control points. For that study, which was performed using code for a two-dimensional model, the following parameters were investigated.

o Panel spacing along the chord

o Type of summation
  o Trapezoidal rule
  o Simpson's rule
  o Gaussian quadrature, using an equation that specifies a logarithmic singularity at the panel edge

o Number of points, and number of half-panels included in the summation

Whereas, each type of summation gave satisfactory results for some specific paneling, the Gaussian quadrature using seven points per half-panel to the control point of the fourth panel gave consistently good results for all the panelings used in the study.

For each row of panels, the additions required to the LEV Program are as follows:

o A routine to determine the locations (x, y, and z) of points to be evaluated using the current capability of the program, and the weighting factors to be applied to the data at each point in calculating the leading-edge suction

o A routine to calculate the leading-edge suction after the solution is completed

This procedure was not evaluated in the three-dimensional environment of the LEV Program under contract NAS1-16740. The evaluation will be a part of this proposed program.

Add Data to Current TAPE14.—Depending on the type of analysis selected (see sec. 3.1.1.2), some additional data are required on file TAPE14 that is generated by the LEV Program.

o Data check: TAPE14 is a formatted file. Add the type and update index of each network.

o Attached-flow solution when it is part of a user-selected separated-flow solution: TAPE14 is an unformatted file. Write records of the leading-edge suction coefficients and the location of each.

- Separated-flow or attached-flow solution: TAPE14 is an unformatted file. Add the panel centroid coordinates and surface pressure coefficients at each panel. Add the lift coefficient for each network.

3.3.3 ATLAS INTEGRATED STRUCTURAL ANALYSIS AND DESIGN SYSTEM

The best choice of aerodynamic paneling places a dense array of panels in areas where the pressure gradients are large. The structural grid is representative of the actual structural configuration of the wing. This inherent difference in the aerodynamic and structural grids can cause problems. Even with the surface spline interpolation method that will be used in PIAS, the effect of large pressures extending over only a small area of the wing may not be included in the calculated nodal loads because of this difference in grids. To some extent, this problem can be alleviated by user coordination during the paneling process.

In general, it is not possible to completely cure the problem with careful paneling because the pressure peaks do not always occur in the same region for all cases. Two approaches are proposed to address this problem:

1. Develop an overall check that compares the total lift calculated by the LEV Program with a summation of the equivalent nodal loads calculated by ATLAS. This would not cure the problem, but it would help identify the problem and it would provide an additional check on the solution.

2. Interpolate the pressures to a grid that is more dense than the structural node grid. These interpolated pressures would be integrated so that the equivalent nodal loads would represent more accurately the distributed pressure loading on the structural finite elements.

3.3.4 CONVERGENCE CRITERIA

There are three levels of solution convergence in PIAS (fig. 2): the convergence of the LEV solution, the convergence to a deformed shape in the aeroelastic cycle, and the convergence to the specified load factor.

3.3.4.1 Convergence of LEV Solution

The procedure for determining covergence of a solution in a standalone execution of the LEV Program is described in section 3.3.2.1 and occurs external to the program. It is necessary to develop a procedure for evaluating the convergence characteristics during the LEV solution process in PIAS. Logic will be added to the LEV Program to make the decisions that the user currently makes. Other planned modifications will improve the quality of the parameters examined and will reduce the number of iterations required to obtain convergence.

3.3.4.2 Convergence to a Deformed Shape

The ECM will monitor the deflected shape of the wing after consecutive aeroelastic cycles to determine convergence to the user-defined tolerance. This procedure is described in section 3.3.1.4.

3.3.4.3 Convergence to the Desired Vertical Load Factor

The first three angle of attack solutions, in most cases, will not provide a solution at the desired vertical load factor. The relationship of vertical load factor to angle of attack is nonlinear. A curve fit of these first three sets of data will be used to determine the angle of attack for the fourth solution as described in section 3.3.1.3. The load factor obtained for each of these angle of attack solutions will be compared to the desired load factor and the iteration will stop as soon as the load factor is within the user-defined tolerance.

3.3.5 CHECKPOINT-RESTART CAPABILITIES

The solution restart capabilities proposed for PIAS are of two kinds. The first is for planned stopping and restarting to allow qualitative reviews of the results at various stages of the aeroelastic solution. The second kind is to allow for a solution restart after the ECM has detected a situation for which a course of action has not been provided. In both cases, the data from the most recent cycle are saved for use in a solution restart. For any restart, the user may continue the solution from any previous checkpoint. Through the use of the PIAS procedure call statement described in section 4.2.1, values of some execution parameters may be changed prior to restarting the solution.

4.0 PIAS DATA MANAGEMENT

Data management is critical to the design of PIAS. Large amounts of data are required to perform each of the types of analysis available in PIAS—data check, new attached-flow or separated-flow solution, and solution restart.

In addition to the functions described in section 3.3.1, the Execution Control Monitor (ECM) of PIAS will direct reading of the input data, control the data flow through the iterative aeroelastic solution, and provide the results needed for postprocessing and solution restart. Several user data files and temporary storage files are used. The management of these files provides for the transfer of data between the LEV Program and the ATLAS system.

4.1 DATA FLOW

The ECM controls the data necessary to execute the LEV Program and the ATLAS system at each level of the aeroelastic solution. The input data that are necessary for each program depend on the type of solution and the level of iteration within PIAS. A detailed data flow diagram is shown in figure 10. There are several sources of data used by the ECM to direct the execution of PIAS. These are:

o Keyword parameters in the PIAS procedure call statement

o Input to the ECM (file ECMUIN, formatted)

o Optional input to the ECM (file JIGFILE or LEVCHK, formatted)

o Input to the LEV Program (file LEVUIN, formatted)

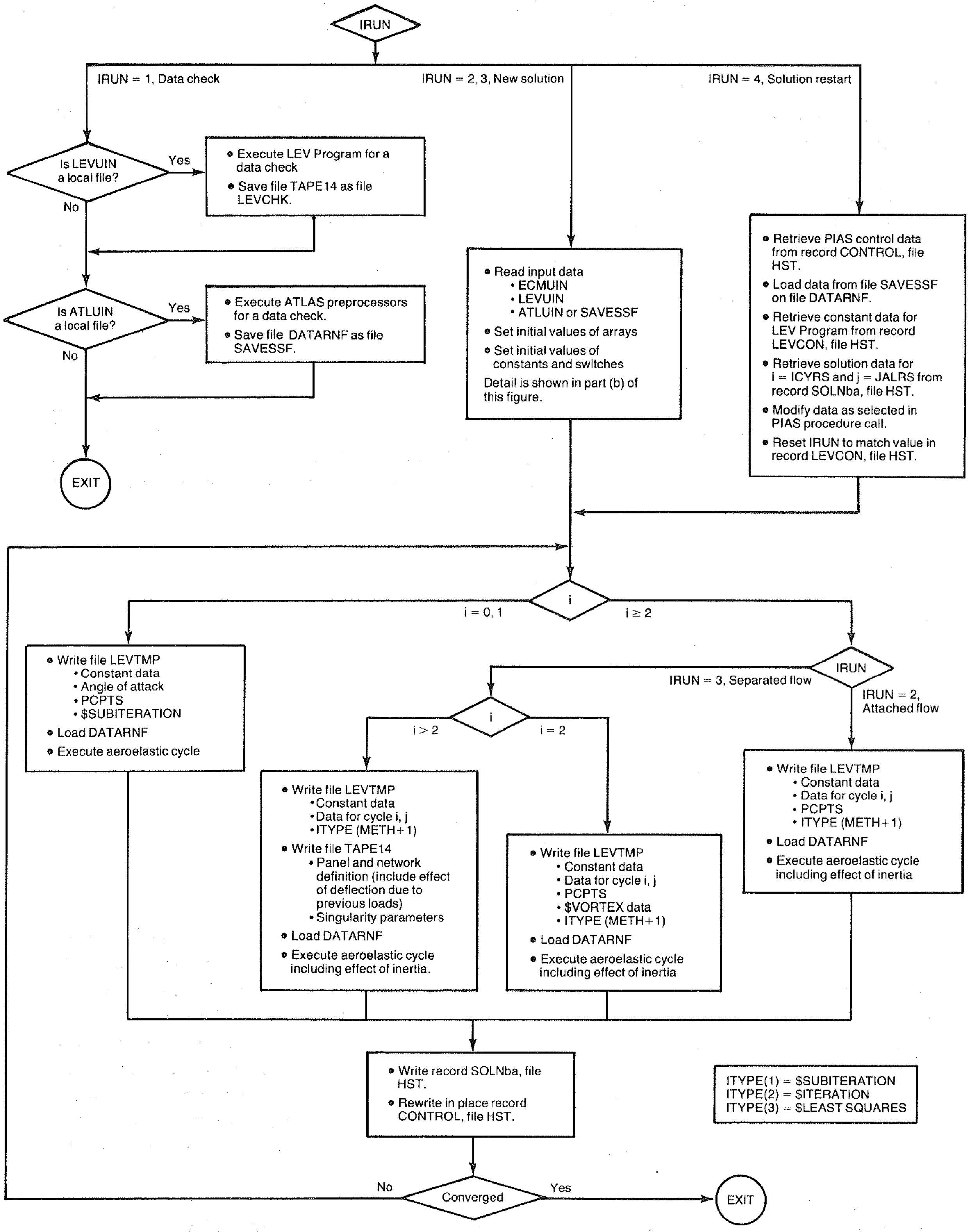
o Input to ATLAS (file ATLUIN, formatted)

o Preprocessed input to ATLAS (file SAVESSF, unformatted)

o Solution history file (file HST = aaaaaa, unformatted)

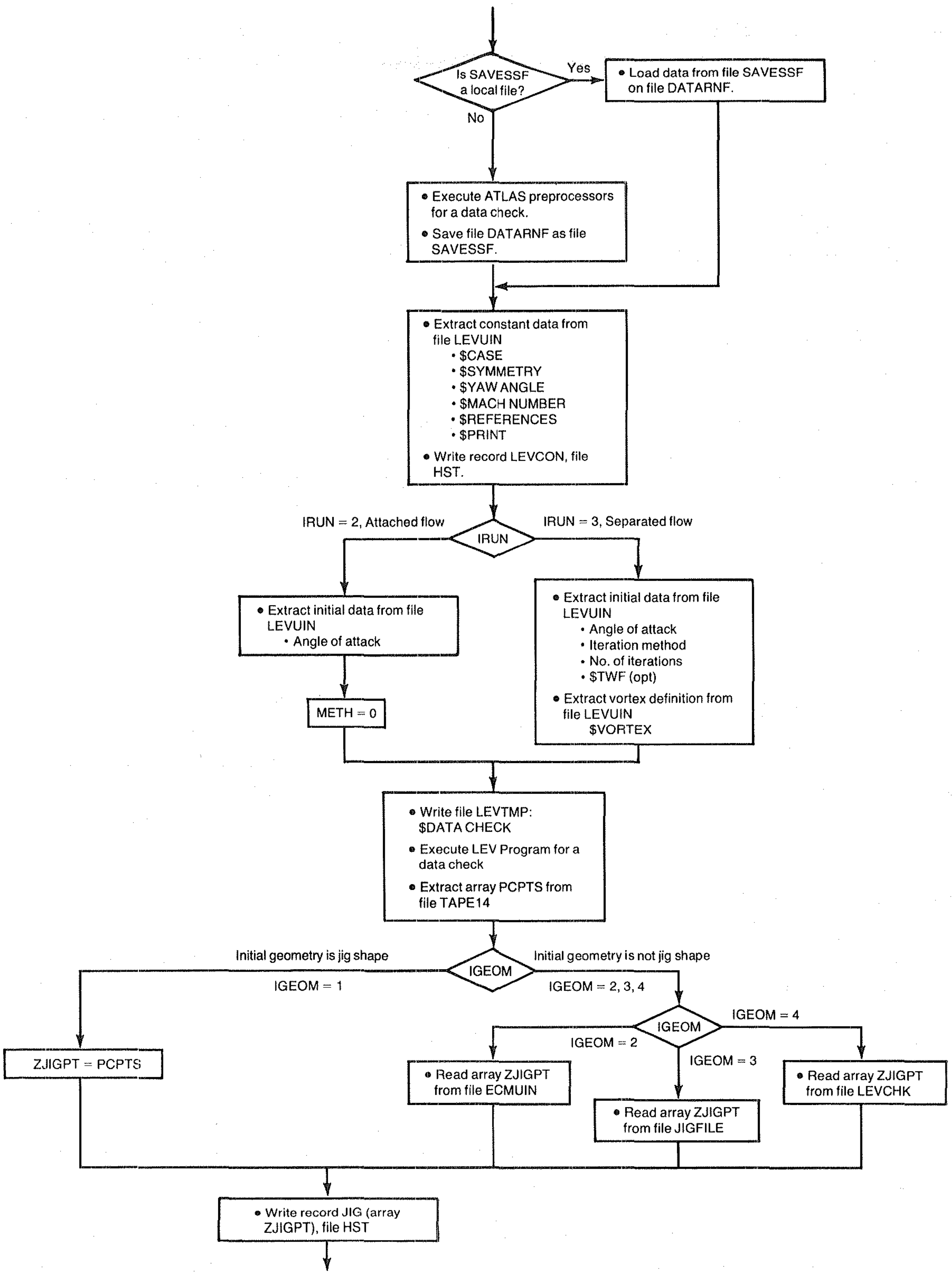
The majority of the input data for executing the LEV Program is the geometry defining the boundary surface of the configuration (panel corner points). Generally, these geometry data remain constant for all design conditions, although the vertical coordinates of the panel corner points are changed during the iterative process as a result of the wing deflection under load. Other input by the user includes the Mach number, initial angle of attack, definition of the initial vortex networks, the solution method to use, and the maximum number of iterations. Throughout the iterative solution, the angle of attack and the position of the vortex networks change in accordance with the solution algorithm. In addition, the user may change the iteration method to be used and the maximum number of iterations to be performed for a solution restart.

The input data to the ATLAS modules define the finite element structural and mass distribution models and the applied loadcases. Depending on the level of iteration within PIAS, the loadcase includes airloads only or a combination of airloads and inertia loads. The ECM compiles the appropriate loadcase data for each cycle.


IRUN
IRUN = 1, Data check
IRUN = 2, 3, New solution
IRUN = 4, Solution restart
Is LEVUIN a local file?
Yes
No
• Execute LEV Program for a data check
• Save file TAPE14 as file LEVCHK.
Is ATLUIN a local file?
Yes
No
• Execute ATLAS preprocessors for a data check.
• Save file DATARNF as file SAVESSF.
EXIT
• Read input data
• ECMUIN
• LEVUIN
• ATLUIN or SAVESSF
• Set initial values of arrays
• Set initial values of constants and switches
Detail is shown in part (b) of this figure.
• Retrieve PIAS control data from record CONTROL, file HST.
• Load data from file SAVESSF on file DATARNF.
• Retrieve constant data for LEV Program from record LEVCON, file HST.
• Retrieve solution data for i = ICYRS and j = JALRS from record SOLNba, file HST.
• Modify data as selected in PIAS procedure call.
• Reset IRUN to match value in record LEVCON, file HST.
i
i = 0, 1
i ≥ 2
• Write file LEVTMP
• Constant data
• Angle of attack
• PCPTS
• $SUBITERATION
• Load DATARNF
• Execute aeroelastic cycle
IRUN
IRUN = 3, Separated flow
IRUN = 2, Attached flow
i
i > 2
i = 2
• Write file LEVTMP
• Constant data
• Data for cycle i, j
• ITYPE (METH+1)
• Write file TAPE14
• Panel and network definition (include effect of deflection due to previous loads)
• Singularity parameters
• Load DATARNF
• Execute aeroelastic cycle including effect of inertia.
• Write file LEVTMP
• Constant data
• Data for cycle i, j
• PCPTS
• $VORTEX data
• ITYPE (METH+1)
• Load DATARNF
• Execute aeroelastic cycle including effect of inertia
• Write file LEVTMP
• Constant data
• Data for cycle i, j
• PCPTS
• ITYPE (METH+1)
• Load DATARNF
• Execute aeroelastic cycle including effect of inertia
• Write record SOLNba, file HST.
• Rewrite in place record CONTROL, file HST.
ITYPE(1) = $SUBITERATION
ITYPE(2) = $ITERATION
ITYPE(3) = $LEAST SQUARES
No
Converged
Yes
EXIT

(a) Overall Solution

Figure 10.—Diagram of Detailed Data Flow in PIAS


Is SAVESSF a local file?
Yes
No
• Load data from file SAVESSF on file DATARNF.
• Execute ATLAS preprocessors for a data check.
• Save file DATARNF as file SAVESSF.
• Extract constant data from file LEVUIN
• $CASE
• $SYMMETRY
• $YAW ANGLE
• $MACH NUMBER
• $REFERENCES
• $PRINT
• Write record LEVCON, file HST.
IRUN = 2, Attached flow
IRUN
IRUN = 3, Separated flow
• Extract initial data from file LEVUIN
• Angle of attack
METH = 0
• Extract initial data from file LEVUIN
• Angle of attack
• Iteration method
• No. of iterations
• $TWF (opt)
• Extract vortex definition from file LEVUIN
$VORTEX
• Write file LEVTMP: $DATA CHECK
• Execute LEV Program for a data check
• Extract array PCPTS from file TAPE14
Initial geometry is jig shape
IGEOM
Initial geometry is not jig shape
IGEOM = 1
IGEOM = 2, 3, 4
IGEOM
IGEOM = 4
IGEOM = 2
IGEOM = 3
ZJIGPT = PCPTS
• Read array ZJIGPT from file ECMUIN
• Read array ZJIGPT from file JIGFILE
• Read array ZJIGPT from file LEVCHK
• Write record JIG (array ZJIGPT), file HST

(b) Input of Data for New Solutions

Figure 10.—(Concluded)

The ATLAS surface spline interpolation module (as explained in sec. 3.1.2.3) will be used for the following two tasks in each aeroelastic cycle, both for new solutions and for solution restarts.

1. Interpolate the pressures calculated by the LEV Program at the centroids of the aerodynamic panels (both the upper and lower surfaces) to the corresponding grid of nodes used in the structural finite element model.

   These interpolated pressures at the nodes of the structural plate-like finite elements are used directly by ATLAS to calculate equivalent nodal loads in performing displacement and stress analyses.

2. Interpolate the calculated deflections of the wing from the structural grid points to the aerodynamic panel corner points.

   The wing deflections calculated by ATLAS are associated with a user-selected loadcase composed of airload pressures and inertia loads. The vertical deflections of the wing are added to the jig shape panel corner point coordinates for use as input to the LEV Program for the next cycle.

The processing of data for the sequence of functions pertaining to ATLAS interpolation is presented in more detail in appendix B.

4.1.1 DATA CHECK

A data check execution may be used to validate the input for the LEV Program and/or the ATLAS system. This type of execution requires only that the proper input data are available to PIAS in local files—LEVUIN for LEV and ATLUIN for ATLAS. The output files from a data check execution—LEVCHK for LEV and SAVESSF for ATLAS—may be used as the input files for subsequent aeroelastic solutions. The use of these files as input reduces the expenditure of computing resources.

4.1.2 NEW SOLUTIONS

Whether the new solution is for attached flow or for separated flow, the data flow is very similar. The data that are typically required for the LEV Program and the ATLAS system are prepared as usual. Data that are required for an aeroelastic solution using LEV and ATLAS, but not needed by either separately, are provided directly to the ECM or as parameters in the PIAS procedure call.

The user-supplied LEV input data are in file LEVUIN and in the parameter list in the PIAS procedure call statement. As the aeroelastic solution progresses, there are changes in the angle of attack, as well as changes in the vertical coordinates of the panel corner points as the surface deforms under load. In addition, for the separated-flow solution as described in section 4.1.2.2, the total number of networks changes as the vortex networks are added to the model. To incorporate these changes in the LEV input, the ECM writes the file LEVTMP for use as input to the LEV Program, allowing the user-supplied file to remain intact. As explained later (sec. 4.2.3), the file LEVUIN may not include all of the panel corner point coordinates. This requires that the first execution of LEV for any new solution be a data check so that the coordinates of all the corner points are available for subsequent cycles.

The preferred source of ATLAS data for new solutions is the file SAVESSF that is the result of a data check execution. If local file SAVESSF is not available when a new aeroelastic solution (either attached flow or separated flow) is requested, the ECM will use file ATLUIN and perform a data check prior to proceeding with the requested solution.

When each cycle for an angle of attack-wing shape combination is completed, all of the results that are needed to continue the solution are written to a random access file. This file is used to restart a solution either from the point at which it was stopped or from a previous point in the solution as explained in section 4.1.3.

4.1.2.1 Attached-Flow Solution

As described above, the file LEVTMP is generated for each cycle in PIAS. When executing an attached-flow solution, few modifications to the LEV input data in the file LEVTMP are required. For a specific angle of attack, only the vertical coordinate of each panel corner point changes with each cycle. The angle of attack is the only other LEV input data item that changes during a PIAS execution.

4.1.2.2 Separated-Flow Solution

When a separated flow solution is desired, the first two cycles for each angle of attack are attached-flow solutions so that the location of the vortex can be determined (sec. 3.3.1.1). The LEV input data written to the file LEVTMP for these cycles do not include the networks that define the vortex, even though the vortex networks must be defined by the user in file LEVUIN. In the second cycle, the wing is deflected as a result of the airload distribution calculated in the first cycle and the leading-edge suction is obtained. The ECM calculates the spanwise location of the vortex apex and writes the input file LEVTMP including the definition of the vortex networks for the first separated-flow cycle.

After the first separated-flow solution for each angle of attack, cost-saving benefits are realized because both the network geometry and the discrete values of singularity strength at each control point from the previous iteration, are read by the LEV Program from the separate file TAPE14. This reduces the number of calculations required and the amount of data needed in file LEVTMP. This option is not currently available for attached-flow solutions. This feature is useful even when the angle of attack is changed, particularily to a smaller angle, since the solution convergence is faster.

4.1.3 PROBLEM SOLUTION RESTART

When the user chooses a solution restart analysis, ECM, LEV, and ATLAS data from a previous execution of PIAS must be provided. The PIAS solution history file is generated during each new solution execution and this file is added to during each solution restart execution. It is a random access file containing all of the ECM, LEV, and ATLAS data from the previous execution that are required to perform a solution restart. The name of this random access file is defined by the HST parameter in the PIAS procedure call described in section 4.2.1. The definition of the structural and mass modeling for ATLAS must be available in file SAVESSF. Changes may be made in the solution process by the use of parameters specified in the PIAS procedure call statement.

4.2 INPUT DESCRIPTIONS

The user input to PIAS consists of ECM data, LEV data, ATLAS data, and the data transmitted by the keyword parameters in the PIAS procedure call. These input data are described in the following sections.

4.2.1 PIAS PROCEDURE CALL

Execution procedures for performing various types of ATLAS analyses are available in the file named ATLAS. The particular procedure in this file that is designed to access and initiate the execution of PIAS is named PIAS. Keyword parameters are included within the parentheses in the procedure call statement for user selection of analysis options. Each keyword and its assigned value is separated from the others by commas. The keywords are shown in capital letters, and are joined by an equal sign to the assigned value that is represented by the lower case letters. The keyword parameters, their default values, and the types of analyses for which they are required are summarized in table 6 and are described in the following paragraphs.

A procedure call to PIAS, with any combination of parameters, must not contain more than eighty (80) characters. When the ATLAS procedure file is available as a permanent file, the CDC job control statements required to execute PIAS are as follows:

```
GET,ATLAS/UN = UATLASU.
CALL,ATLAS,PIAS(RUN=irun,CYC=icymx,HST=nameh,MTH=meth,GEO=igeom,
                RST=irst,JRS=jalrs,IRS=icyrs,ITM=itmx,TWF=twfwt)
```

RUN=irun Type of analysis (integer). The options are:
1, LEV and/or ATLAS data check only
2, New attached-flow solution
3, New separated-flow solution
4, Problem solution restart, will continue with an attached-flow or separated-flow solution
Default: 1

NOTE: CALL,ATLAS,PIAS. is sufficient when executing a data check (RUN=1) since none of the remaining parameters are applicable.

The following two parameters are applicable for both a new solution (RUN=2 or 3) and a solution restart (RUN=4):

CYC=icymx Maximum number of aeroelastic cycles to be performed per angle of attack (integer).
Default: 3

HST=nameh Name of local file containing the solution history data, see table 7 for content (1 to 7 characters). Generated as an output file except for data check (RUN=1). Also used as input for a solution restart (RUN=4).
Default: HISTORY

Table 6.— PIAS Procedure Call Parameters Associated With the Available Types of Analyses

Parameter keyword	Data check	New solution		Solution restart		Default value
		Attached flow	Separated flow	Attached flow	Separated flow	
RUN	1	2	3	4	4	1, data check, type of analysis
CYC		X	X	X	X	3, number of aeroelastic cycles
HST		X	X	X	X	HISTORY, solution history file name
MTH			X		X	1, Quasi-Newton for new solution; uses last method (from HST file) for solution restart; solution method in LEV
GEO		X	X			1, LEVUIN, source of jig shape geometry
RST				X	X	1, solution counters continue from values at solution restart
JRS				X	X	Last angle of attack solution on HST
IRS				X	X	Last aeroelastic cycle on HST for JRS angle of attack
ITM					X	Same as number of iterations last used for LEV, from HST
TWF					X	Same as last used weighting factor on twist function, from HST; only applicable for least-squares solution method in LEV (MTH = 2)

Table 7.– Format of the PIAS Solution History File (named random access file)

Record name	Description	
CONTROL	Data input to ECM either in the ECMUIN input file or in the PIAS procedure call. Also the current values of the counters 'i' and 'j.'	This record is rewritten in place for each angle of attack solution 'j' and aeroelastic cycle 'i.'
JIG	Jig shape geometry in terms of aerodynamic panel corner points.	
LEVCON	All constant data required for input to the LEV program.	

SOLNba*	Data describing this particular step in the solution cycle.
	● Description of flight condition.
	● Network definition, including panel corner points.
	● LEV pressure distributions, upper and lower surface, and the location of each.
	● Mass distribution
	● Deflected shape
	● LEV grid
	● ATLAS grid
		This record is repeated for:
		j = 1, i = 0 to ICYMX (GEO = 1)
		1 to ICYMX (GEO >1)
		j = 2, i = 1 to ICYMX
		j = 3, i = 1 to ICYMX
		j = 4, i = 1 to ICYMX

*b = Display code equivalent of angle of attack solution 'j', e.g., if j = 1, b = A
a = Display code equivalent of aeroelastic cycle 'i', e.g., if i = 10, a = J

The following parameter is applicable for a new or restart separated-flow solution (RUN=3 or 4):

MTH=meth Type of iterative solution method to be used in the LEV Program (integer). The options are:
1, Quasi-Newton method
2, Least-squares method
Default: 1, if new separated-flow solution (RUN=3),
same as on HST file, if solution restart (RUN=4)

The following parameter is applicable only for a new solution (RUN=2 or 3):

GEO=igeom Type of initial geometry for the LEV Program (integer). The options are:
1, Jig shape geometry.
2, Deflected geometry, jig shape geometry is read in free field format from file ECMUIN.
3, Deflected geometry, jig shape geometry is read in free field format from file JIGFILE.
4, Deflected geometry, jig shape geometry is read in fixed field format (as generated by an LEV Program data check) from file LEVCHK.
Default: 1

The following three parameters are applicable only for a solution restart (RUN=4):

RST=irst Indicator for counters (j and i) on a solution restart (integer). The options are:
1, Counters should continue from a specified angle of attack solution (j=JRS=jalrs) and aeroelastic cycle (i=IRS=icyrs).
2, Counters should be reinitialized to the first angle of attack solution (j=1) and the first aeroelastic cycle with a separated-flow solution in LEV (i=3).
Default: 1

JRS=jalrs Identifies the angle of attack (value of j) for retrieving restart data from file HST=nameh (integer).
Default: the last angle of attack solution for which data were previously saved in the solution history file

IRS=icyrs Identifies the aeroelastic cycle (value of i) associated with JRS=jalrs for retrieving restart data from file HST=nameh (integer).
Default: the last aeroelastic cycle associated with the last angle of attack solution for which data were previously saved in the solution history file

The following parameter is applicable only for a separated-flow solution restart (RUN=4):

ITM=itmx Maximum number of iterations to be performed in the LEV Program (integer). This data value is included in the LEV user input file LEVUIN when RUN=3.
Default: the most recent value in the solution history file

The following parameter is applicable only for a separated-flow solution restart (RUN=4) and only when the least-squares solution method (MTH=2) is used in the LEV Program:

TWF=twfwt Multiplier for weighting of the twist functions ($TWF option) (decimal).
Default: the most recent value in the solution history file

4.2.2 EXECUTION CONTROL MONITOR (ECM) INPUT DATA

Input data for the ECM are only required for a new solution (RUN=2 or 3) and must be in file ECMUIN. This file contains airplane condition data, convergence tolerances for the iterative aeroelastic solution, and parabolic nose drag information. The format of these user data is illustrated in table 8.

For a new solution (RUN=2 or 3), if the wing geometry input to LEV is a deflected shape (GEO=2, 3, 4), the user must supply a file containing the jig shape geometry. The name of this file depends on the value specified for the GEO parameter in the PIAS procedure call.

o If GEO=2, this file is the ECM user input file ECMUIN (see table 8).

o If GEO=3, this is a separate data file JIGFILE, with the jig shape data in free field format (see card sets 5 through 7 in table 8).

o If GEO=4, this file is a separate data file LEVCHK, in the fixed field format generated by the LEV Program during a data check (see ref. 5).

4.2.3 LEADING-EDGE VORTEX (LEV) PROGRAM INPUT DATA

The input data for the LEV Program are summarized in table 9; a complete description of all the variables is presented in reference 5. The LEV user input must be in a local file named LEVUIN when using PIAS with RUN=1, 2, or 3. Some of the input is optional regardless of the type of analysis. Input file LEVUIN is not required for a solution restart (RUN=4) because the solution history file will contain the necessary LEV data, although certain data items can be modified through the PIAS procedure call.

The input to define the network geometry may be very detailed, i.e., the coordinates of all panel corner points may be defined explicitly, or only a general description may be input and the LEV Program preprocessors generate the coordinates of the panel corner points. There is a required sequence for defining networks for the LEV Program. The definition of those networks for which geometry will be updated (the free and fed sheet networks representing the vortex and their associated wakes) must be input after the definition of those networks to which they are attached (doublet networks representing the camber surface of the wing).

Some additional data will be required when the LEV Program is executed as part of PIAS. In particular, it is necessary to request the leading-edge suction calculation and subsequent data transfer for the attached-flow solution if executed in conjunction with a separated-flow solution. The specific data format will be determined as part of the detailed specification.

Table 8. – Input Data for the Execution Control Monitor

Input	Record No./Item	Variable	Description
If RUN = 2 or 3	1/1	V_e	Equivalent airspeed (knots)
	/2	W	Gross weight (lb)
	/3	n_z	Desired vertical load factor
	/4	EPS	Acceptable tolerance on n_z
	/5	FRACDIS	The change in structural node displacement between consecutive cycles (as a fraction of total displacement) that is acceptable for a converged solution (recommended value 0.04).
	/6	$\Delta\alpha$	Increment added to initial angle of attack (α_1 from LEV user input) to obtain second solution (α_2). Default: α_2 is the angle of attack required to obtain n_z, assuming a linear variation from zero through the initial solution.
If RUN = 2 or 3	2/1	NNOSE	Number of spanwise sections at which the ratio of nose radius to streamwise chord is defined (integer).
If RUN = 2 or 3	3/	YNOSE	Spanwise coordinate of section nose midplane, NNOSE values.
If RUN = 2 or 3	4/	RNOSE	Ratio of nose radius to streamwise chord. Used to obtain the parabolic nose drag coefficient. NNOSE values in same sequence as YNOSE.
If RUN = 2 or 3 and GEO = 2	5/1	NNETT	Total number of networks for jig shape geometry (integer).
If RUN = 2 or 3 and GEO = 2	6/1	NROW	Number of rows in this network (integer).
	/2	NCOL	Number of columns in this network (integer).
If RUN = 2 or 3 and GEO = 2	7/	ZJIGPT	X, Y, Z coordinates of panel corner points, NROW points.
			Repeat record 7, NCOL times.
			Repeat records 6 and 7 NNET times.

Note: All data items are input in free field format and, unless specified otherwise, they are decimal numbers.

Table 9. – Summary of Input Data for the LEV Program

	$CASE	Followed by two card images: title, user information
	$ANGLE OF ATTACK	Followed by value on next card image (deg)
	$YAW ANGLE	Followed by value on next card image (deg)
	$SYMMETRY	Followed by indicator on next card image: if = 1.0, symmetry about x-z plane; otherwise, not symmetrical
	$MACH NUMBER	Followed by value on next card image, must be less than 1.0
	$REFERENCES	Followed by x, y, z coordinates of the moment center on next card image, followed by reference area and lengths on next card image
	Type of solution desired, choose one of the following four:	
	$DATA CHECK	Verifies network abutments
	$SUBITERATION	If free and fed sheets are not included in the model, will give attached-flow solution
	$ITERATION	Iterative separated-flow solution using the quasi-Newton scheme, followed by maximum number of iterations on next card image
	$LEAST SQUARES	Iterative separated-flow solution using the least-squares method, followed by maximum number of iterations on next card image
*	opt $TWF	Followed on next card image by a weighting factor for the twist function that is used to minimize kinking of the free sheet, default = 1.0. Applies only to $LEAST SQUARES
	$PRINT	Followed by selectors for various print options (12) on next two card images
*	$NETWORK	Followed by number of networks on next card image. Max 20 Data defining the networks follow. A network for which geometry will be updated ($VORTEX) must be input after the network to which it is attached. Each network is preceded by one of the following:
*	$POINTS	
*	$QUADRILATERAL	
*	$GOTHIC	
*	$VORTEX (each call creates 2 or 4 networks)	
*	$TRAILING WAKE	
	opt $FLOW VIZ	Followed by card images with data to define the points to be evaluated
	$END OF CASE	Always last card in input

* Identifies data that are not required for a restart of LEV.
opt Identifies optional input.

4.2.4 ATLAS INPUT DATA

All of the ATLAS input data and execution control statements that can be used to perform a wide variety of structural analysis tasks are described in detail in reference 6. The ATLAS input data pertinent to the proposed aeroelastic solution program are summarized in table 10 to illustrate the type of data sets—Nodal, Stiffness, Boundary Condition, and Mass—that are required to define the finite element structural and mass distributions. The Mass data are used in defining inertial loads that may be combined with the static airloads that are generated by the LEV Program and preprocessed for ATLAS by the ECM.

The ATLAS user input to PIAS for a data check (RUN=1) must be in a local file named ATLUIN. When read by ATLAS, these data are stored in a random access file that is used by the various ATLAS modules. The user can save this random access file after any program execution. For a PIAS execution that does not include modifications to the ATLAS user input data, the previously processed ATLAS input data can be retreived as a local file that must be named SAVESSF.

It can be noted in table 10 that many default values, optional formats, and generation features for input data are provided by ATLAS to minimize the amount of information that a user must provide. Many additional facilities are provided by ATLAS for convenient and versatile data preprocessing and postprocessing, as described in reference 6. For example, the geometry input data prepared for either the NASA-LaRC airplane configuration plot program (ref. 25) or the NASA supersonic-aircraft aerodynamic design and analysis system (ref. 26) can be used directly in defining nodal coordinates for the structural and mass models (see sec. 320.9, ref. 6).

4.3 DATA FILE DEFINITIONS

The files used by PIAS include user input files, temporary storage files (preceded by an *), and output files. The output files may be used as input files for subsequent executions (table 11). All of the file names are fixed except for the name of the PIAS solution history file HST. These files are defined as follows:

ECMUIN formatted file. Card image input file for the ECM.

LEVUIN formatted file. Standard card image input data for the LEV Program. Provided by the user with all network data included.

*LEVTMP formatted, temporary storage file. Standard card image input file for the LEV Program. Created by the ECM and tailored for the specific solution, i.e., data check, attached-flow solution, or separated-flow solution.

*TAPE14 formatted file, from a data check execution of the LEV Program. Contains all network corner point coordinates.

unformatted file, from an attached-flow solution of the LEV Program if part of a user-selected separated-flow solution. Contains leading-edge suction coefficients and the locations, along the semispan, of these coefficients.

unformatted file, from a separated-flow or attached-flow solution of the LEV Program. Contains the coordinates of the grid corner points and the panel

Table 10. – Summary of the Pertinent ATLAS Structural and Mass Model Input Data

(a) Nodal Data (sec. 146, ref. 6)

```
BEGIN NODAL DATA
--------------------------------------------------------------------------------
SET n
--------------------------------------------------------------------------------
 ⎧REC⎫                                  ⎧ X-Axis  Z-Axis    ⎫
 ⎨CYL⎬  label < Label1 > Origin         ⎨ X  deg < Label2 > ⎬
 ⎩SPH⎭                                  ⎪ Y  deg < Label2 > ⎪
                                        ⎩ Z  deg < Label2 > ⎭
RESUME  ⎧Label  ⎫
        ⎩GLOBAL ⎭
                        ⎧Label  ⎫
ANALYSIS FRAME          ⎨GLOBAL ⎬   < ATlist(n) >
                        ⎩INPUT  ⎭
node1 < list1 >< TO  node2 < list2 ><< BY inc >< OF d1  d2. . .>>>
                        / INPUT    \
node1  FROM  Node2     <  ANALYSIS  >   dc1  dc2  dc3
                       \  GLOBAL   /
                        \ Label   /
node1  FROM  Node2  Node3  length
--------------------------------------------------------------------------------
REORDER FROM Nn
--------------------------------------------------------------------------------
END NODAL DATA
```

(b) Stiffness Data (sec. 152, ref. 6)

```
BEGIN STIFFNESS DATA
--------------------------------------------------------------------------------
SET  n
  / BEAMXY \
  \ BEAMXZ /
--------------------------------------------------------------------------------
BEGIN PROPERTY DATA
      Pcode <* "text" > Plist
END PROPERTY DATA
--------------------------------------------------------------------------------
BEGIN ELEMENT DATA
    Eltype < Mcode >< Tcode >< Userid1 > Nlist1  Plist < TO < Userid2 >
      Nlist2 < BY < Inc1 > inc2 >>
END ELEMENT DATA
--------------------------------------------------------------------------------
END STIFFNESS DATA
```

Note:

{ } – Data items enclosed by braces have optional input formats. One of the indicated options must be selected.

<> – Data items enclosed by brackets have default values. If the default value, as shown in reference 4, is acceptable for defining the problem data, the bracketed item or items need not be input.

Table 10.–(Concluded)

(c) Boundary Condition (BC) Data (sec. 106, ref. 6)

```
------------------------------------------------------------------------
BEGIN BC DATA
------------------------------------------------------------------------
< SET Se > < STAGE st >
------------------------------------------------------------------------
                   { AS LISTED     }
ORDER RETAIN       { BY INTERNALID }
                   { BY USERID     }
{ FREE    }  { Nf1 Nf2 . . . }
{ RETAIN  }  { ALL           }    FOR ATlist(n)
{ SUPPORT }
{ FREE    }  { Nf1 Nf2 . . . }                 { 1 }
{ RETAIN  }  { ALL           }    IN SURFACE   { 2 } < THROUGH Nn>
{ SUPPORT }  { SYMM          }                 { 3 }
             { ASYM          }
------------------------------------------------------------------------
END BC DATA
------------------------------------------------------------------------
```

(d) Mass Data (sec. 138, ref. 6)

```
------------------------------------------------------------------------
BEGIN MASS DATA
------------------------------------------------------------------------
SET n
------------------------------------------------------------------------
BEGIN CONDITION DATA
   { STAGE St        }
   { PANEL DATA Sub  }   CONDITION cond << Fuel Payload > Conmass >
END CONDITION DATA
------------------------------------------------------------------------
BEGIN MASS ELEMENT DATA
   / BEAMXY \
   \ BEAMXZ /
                / F1 \
    Eltype     <  F2  >  < Userid > Nlist1 plist < TO < Userid2 > Nlist2 < BY < Inc1 > inc2 >>
                \ F3 /
END MASS ELEMENT DATA
------------------------------------------------------------------------
BEGIN CONCENTRATED MASS DATA n
    < name > Node1 < Node2 Node 3 > / < CG > Cnode \ < O = Label1 >< I = Label2 >
      wt < i11 i22 i33 i12 i13 i23 > \  CG   x y z /
END CONCENTRATED MASS DATA
------------------------------------------------------------------------
BEGIN LUMPING DATA
   / LUMP1 \
   \ LUMP2 /
           { STIFFNESS SUBSETS ATlist }
    LUMP   { MASS SUBSETS ATlist      }   AT NODE SUBSET Num
           { FUEL                     }
           { PAYLOAD                  }
END LUMPING DATA
------------------------------------------------------------------------
END MASS DATA
------------------------------------------------------------------------
```

Table 11.– Input and Output Data Files Used by PIAS

Type of PIAS analysis	RUN	Code	Local input data files	Local output data files
Data check	1	ECM	none	none
		LEV	LEVUIN	LEVCHK
		ATLAS	ATLUIN	SAVESSF
New attached-flow solution	2	ECM	ECMUIN JIGFILE* LEVCHK*	HST(HISTORY)
		LEV	LEVUIN	none
		ATLAS	ATLUIN or SAVESSF	SAVESSF none
New separated-flow solution	3	ECM	ECMUIN JIGFILE* LEVCHK*	HST(HISTORY)
		LEV	LEVUIN	none
		ATLAS	ATLUIN or SAVESSF	SAVESSF none
Problem solution restart	4	ECM	HST(HISTORY)	HST(HISTORY)
		LEV	none	none
		ATLAS	SAVESSF	none

A file name followed by another name enclosed in parentheses shows that the user may either select a file name or use the default name shown in parentheses.

*Note: The use of this file for jig shape geometry is optional and depends on the value of GEO. See PIAS procedure call description (sec. 4.2.1) for options.

centroids, the discrete values of singularity strength, and the pressure coefficients.

LEVCHK formatted file. Copy of TAPE14 as written during a data check execution of the LEV Program.

ATLUIN formatted file. Standard card image input data for ATLAS. Provided by the user with all structural and mass data included.

SAVESSF unformatted file. Contains the results of the ATLAS module executions. It is used instead of ATLUIN for a solution restart.

Jig Shape Geometry File formatted file, the name depends on the value of the GEO parameter in the PIAS procedure call statement. Contains panel corner point coordinates for the jig shape. Free field format on file ECMUIN if GEO=2, or on file JIGFILE if GEO=3. Fixed field format (as written on TAPE14 during a data check execution of the LEV Program) on file LEVCHK, if GEO=4.

HST=nameh unformatted file. (Default name is HISTORY). Contains the solution history data needed for a solution restart (see table 7). An alternate name may be defined by the HST parameter in the PIAS procedure call.

5.0 COMPUTING SPECIFICATIONS

Computer programs are dependent, to some extent, on the host computing hardware and operating system. Therefore, descriptions of the computing hardware and operating system for which PIAS is designed, and the construction of the application software, are presented below.

5.1 HARDWARE AND OPERATING SYSTEM

The Program for an Iterative Aeroelastic Solution (PIAS) is designed for the current hardware and operating system at the NASA Langley Research Center (LaRC). The hardware consists of Control Data Corporation (CDC) Cyber mainframes with a NOS 1.4, level 552 operating system. This system has been modified relative to the standard CDC operating system.

5.2 APPLICATION SOFTWARE

The Execution Control Monitor (ECM) of PIAS will be written specifically for the foregoing combination of computer and operating system. The LEV Program, level 40, is already resident on the LaRC computer. The LaRC operating system and the subroutine libraries available at LaRC (refs. 27 through 30) have been reviewed in preparation for installing Version 4.1 of ATLAS at the LaRC. In addition, several routines used in ATLAS are known to be peculiar to the Boeing Computer Services (BCS) computing installation. Some of these routines are available in libraries and will be delivered to the LaRC with the ATLAS code. Others have been written to take advantage of the BCS computing installation and must be either rewritten or replaced with routines currently resident at the LaRC.

As discussed in section 3.2, the ECM will be an ATLAS Control Program. The ATLAS execution language, SNARK statements (ref. 16), and FORTRAN extended Version 4 language statements (ref. 31) will be used in constructing the ECM. FORTRAN language syntax is allowed within an ATLAS Control Program.

The ATLAS execution statements are those available in Version 4.1 of ATLAS and are summarized in table 200-1 of reference 6. These user-oriented directives identify the ATLAS modules whose functions are needed to solve the aeroelastic problem. They also include parameters to select options associated with the modules, reset default values, and specify data items.

SNARK is a high-level language for the analysis and solution of matrix related problems. SNARK statements display simplicity and economy of usage in a matrix- oriented syntax. Use of the SNARK language in the ECM will allow for algebraic matrix manipulation, matrix input and output on random access files, and management of matrix storage in blank common consistent with the PIAS requirements. Blank common will be that area of computer core between the end of the ECM program and the random file indices. The SNARK statements will be translated by the SNARK precompiler into equivalent FORTRAN code.

FORTRAN statements will be intermixed with the ATLAS directives and SNARK statements to perform those additional calculations required of the ECM to obtain a converged aeroelastic solution. The in-line FORTRAN code plus the FORTRAN code generated by the SNARK and ATLAS language precompilers will be further processed by the Control Data Corporation FTN IV, Version 4.8, compiler for subsequent loading and execution.

6.0 PIAS CERTIFICATION

The certification of PIAS will be accomplished in three stages.

1. Standalone checkout of the LEV Program as modified

2. Checkout of PIAS using an attached-flow case

3. Checkout of PIAS using a separated-flow (leading-edge vortex) case

For each of the above stages, simple test cases will be used during the software development prior to using test cases that are more representative of design problems. Throughout this procedure, the rationale for any changes to the specification will be to attain a more rapid convergence to the desired solution and therefore, a less costly execution.

6.1 DEVELOPMENT TEST PLAN

6.1.1 LEV PROGRAM

Validation of the LEV Program will be conducted so that the effect of each modification can be evaluated. The aerodynamic models, primarily delta wings, that have been used throughout the development of the LEV Program (ref. 4) will be used for standalone checkout of the modified LEV Program. For completeness, the clipped-tip arrow wing used for evaluating the LEV Program under NASA contract NAS1-15678 (ref. 8) will be added to this set of checkout problems.

6.1.2 ATLAS SYSTEM

Validation of the ATLAS Version 4.1 system installation at the LaRC computing complex will be accomplished by executing standard ATLAS test cases.

6.1.3 PIAS

Initial checkout of PIAS will utilize the checkpoint-restart capability to step through typical, but simple, aeroelastic solutions.

6.2 CERTIFICATION TEST PLAN

Certification and demonstration of the aeroelastic analysis capabilities provided by PIAS will be performed at the LaRC in conjunction with the final delivery and installation of the system. The selected analytical options to be exercised during the demonstration of PIAS will emphasize use of the newly developed capabilities and will address their applicability to as many analysis situations as is practical. In addition, the aeroelastic structural problems proposed for use in validating the system have been selected from those problems for which structural and weight mathematical models are readily available. This approach will allow for the practical analytical capabilities of the system to be verified and demonstrated with a minimum contract cost.

It is proposed that the two problems identified as the NACA delta wing and the NASA SCAT-15 flutter model be used to verify the new aeroelastic analysis capabilities provided by PIAS. These two problems, as described in the following sections, have been previously subjected to rather extensive experimental testing and analysis. Furthermore, refined and proven ATLAS structural and mass finite element models and previous static and dynamic analysis results for these problems are available. The aerodynamic paneling for these two problems will be prepared as part of the certification.

6.2.1 NACA DELTA WING

The small-scale structural test model of a 45-degree delta wing, as described in reference 32, is proposed as the first problem for verifying PIAS. It is unlikely that a vortex would develop on this configuration because of the moderate leading-edge sweep. Therefore, it is intended to use this wing to verify the convergence to a deformed shape and load factor without involving the complications of convergence on the vortex geometry. The ATLAS structural model is comprised of mid-surface nodes with SPAR and COVER finite elements. Because the structure is symmetric about the plane Y=0, only one-half of the structure is modeled. A plan view of the structural model is shown in figure 11.

The ATLAS mass model is defined by the structural finite elements plus SCALAR mass finite elements to represent the concentrated weights described in reference 32. Existing analytical results show good correlation of the ATLAS calculated mass distribution and vibration natural frequencies with the experimentally obtained data previously documented for the test model.

An aerodynamic model of the NACA delta wing is not currently available. It is planned that streamwise paneling will be used to define the configuration boundary surface for the LEV Program.

6.2.2 NASA SCAT-15 FLUTTER MODEL

It is proposed that the SCAT-15 flutter test model shown in figure 12 also be used to verify the analytical capabilities of PIAS. This model was built by The Boeing Company in 1968 during the national supersonic transport program to validate supersonic flutter analysis methods (ref. 33). The existing ATLAS structural finite element representation shown in figure 13 includes the wing, the horizontal tail, the vertical tail and its ventral, the upper and lower wing fin, the inboard and outboard nacelles, and a "stick" model of the fuselage. The wing structure was modeled by using ATLAS midsurface nodes with SPAR and COVER finite elements. Because of structural symmetry, only one-half of the total structure was modeled.

The existing ATLAS mass distribution model is such that excellent correlation has been obtained between the vibration and flutter analysis results and the experimental test data.

Only the wing and body of this model will be used in the validation. The wing is highly swept and it is expected that a vortex will form at moderate angles of attack. This case will be used to verify the overall aeroelastic capabilities of PIAS. Conical paneling will be used for the wing in the LEV Program because this formulation generally provides more rapid convergence on the vortex geometry.

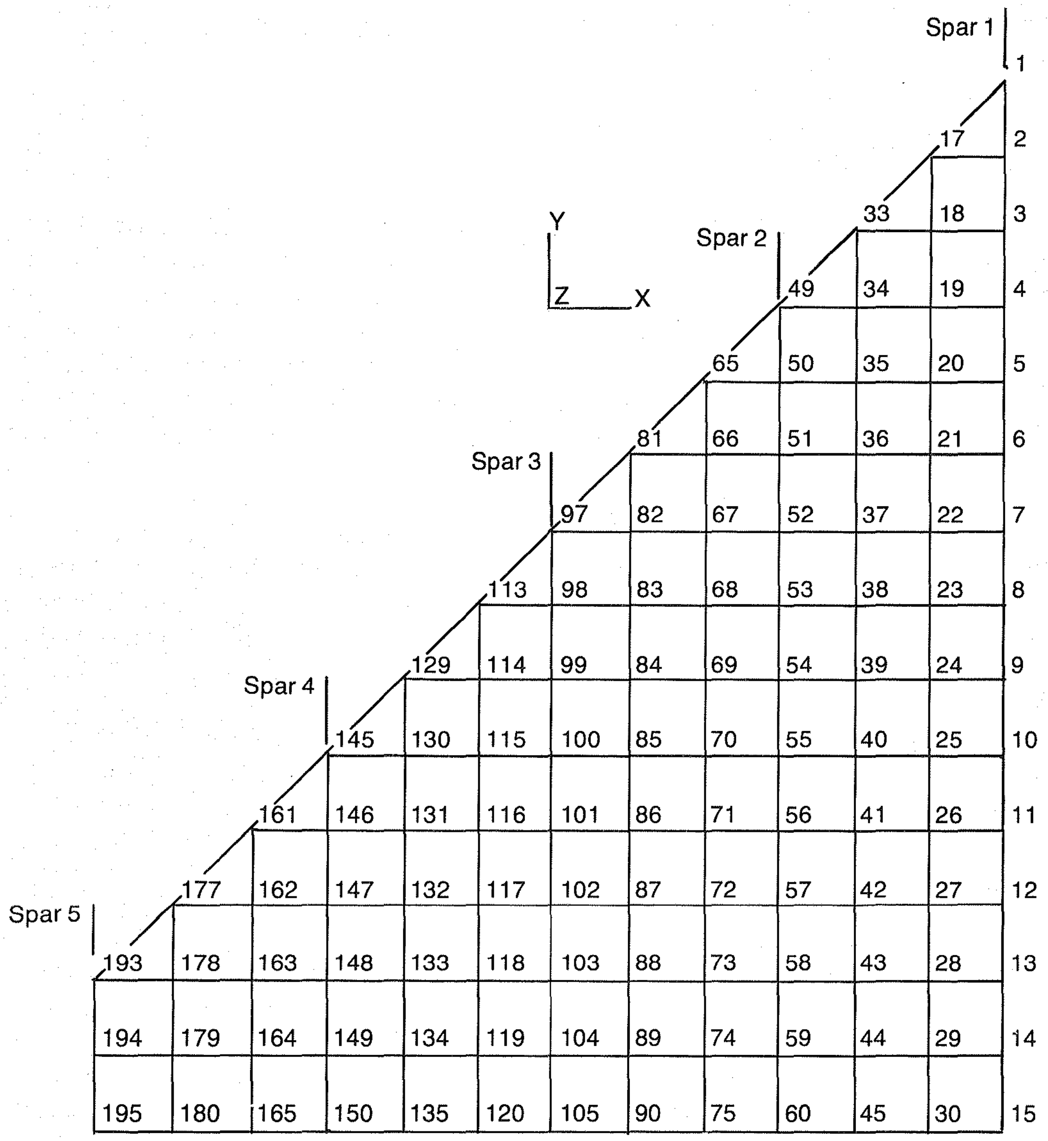


Figure 11.– Delta Wing Structural Grid With Node Numbers

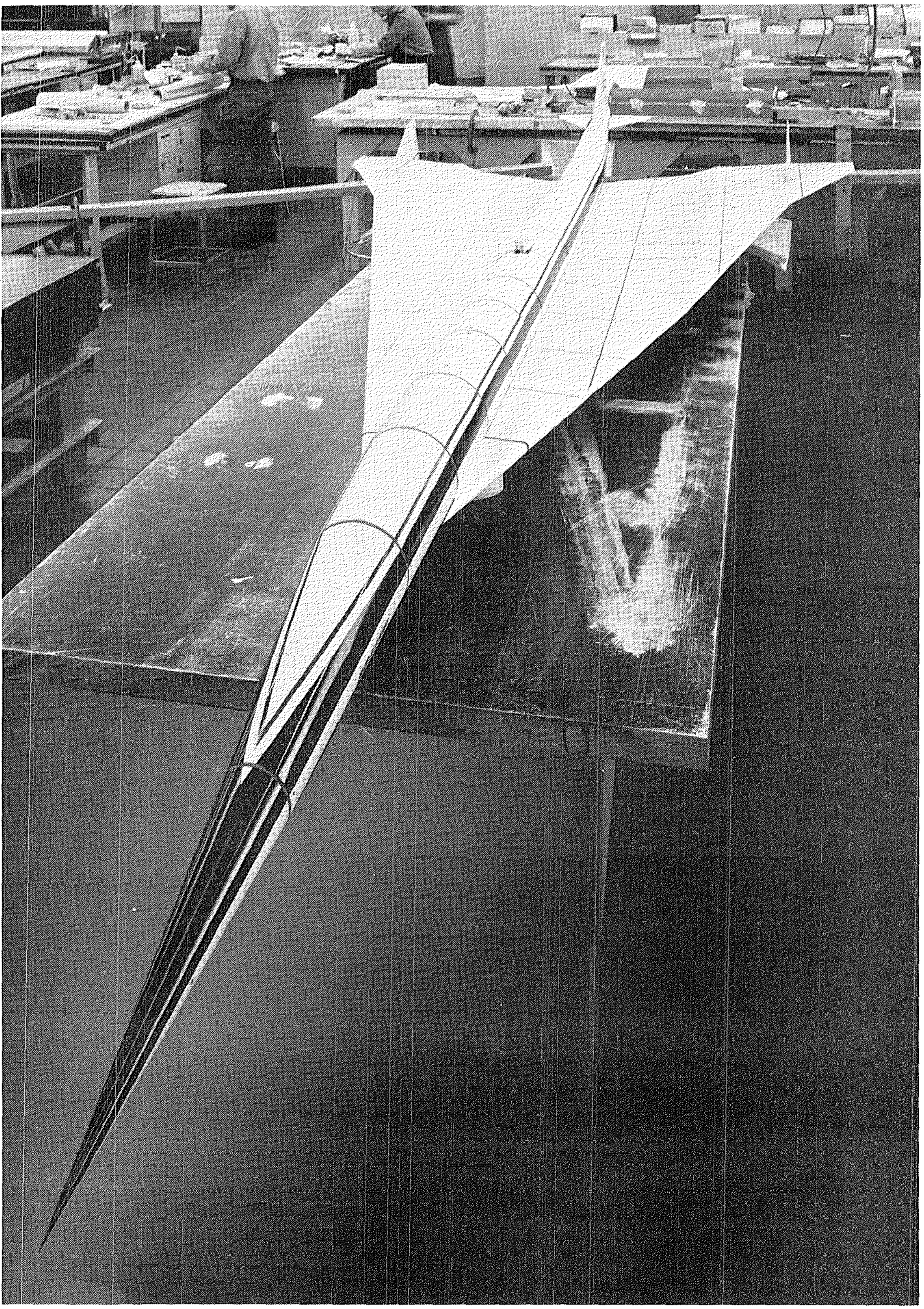


Figure 12. – The SCAT-15 Flutter Test Model

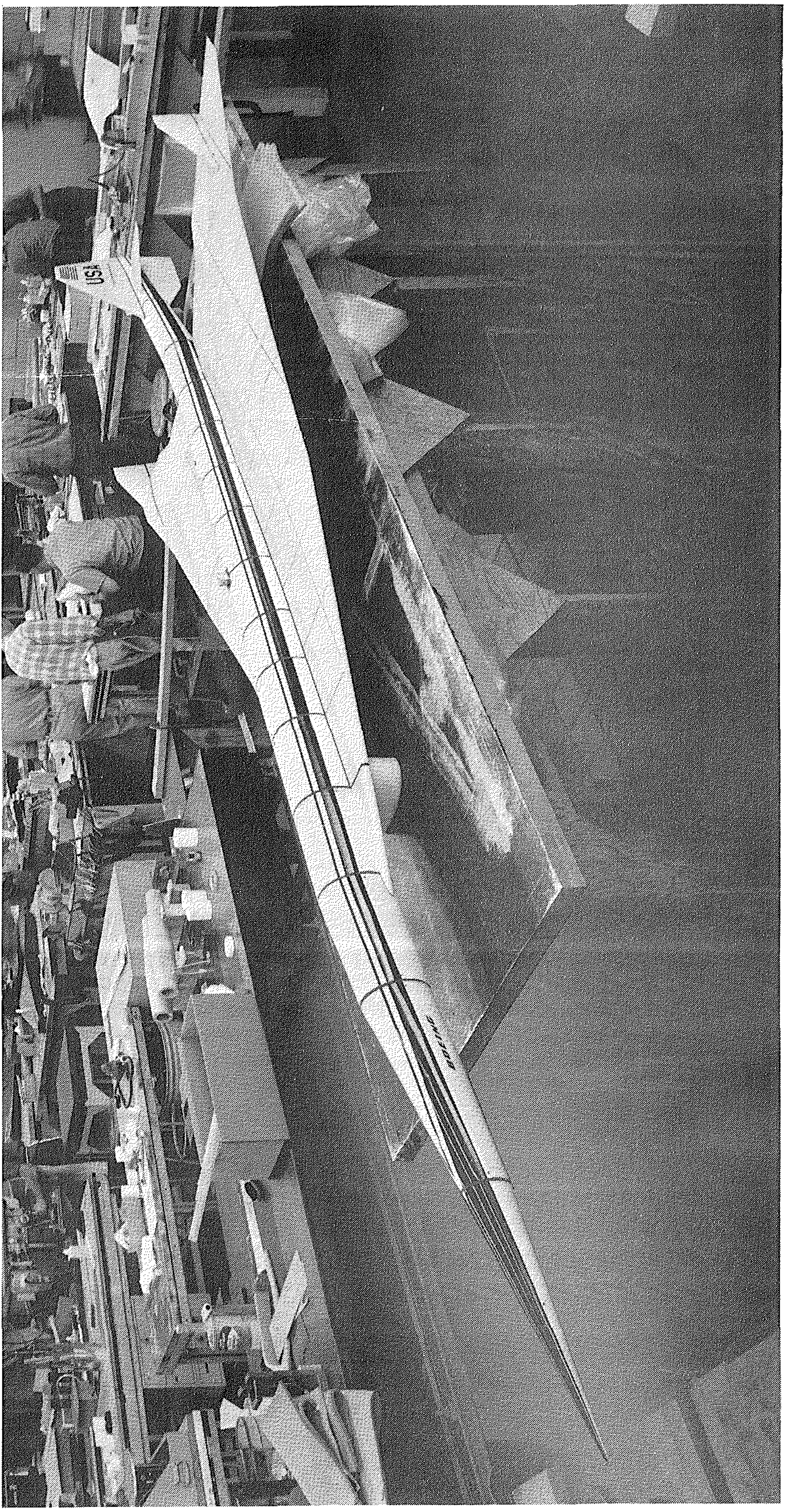


Figure 12. – (Concluded)

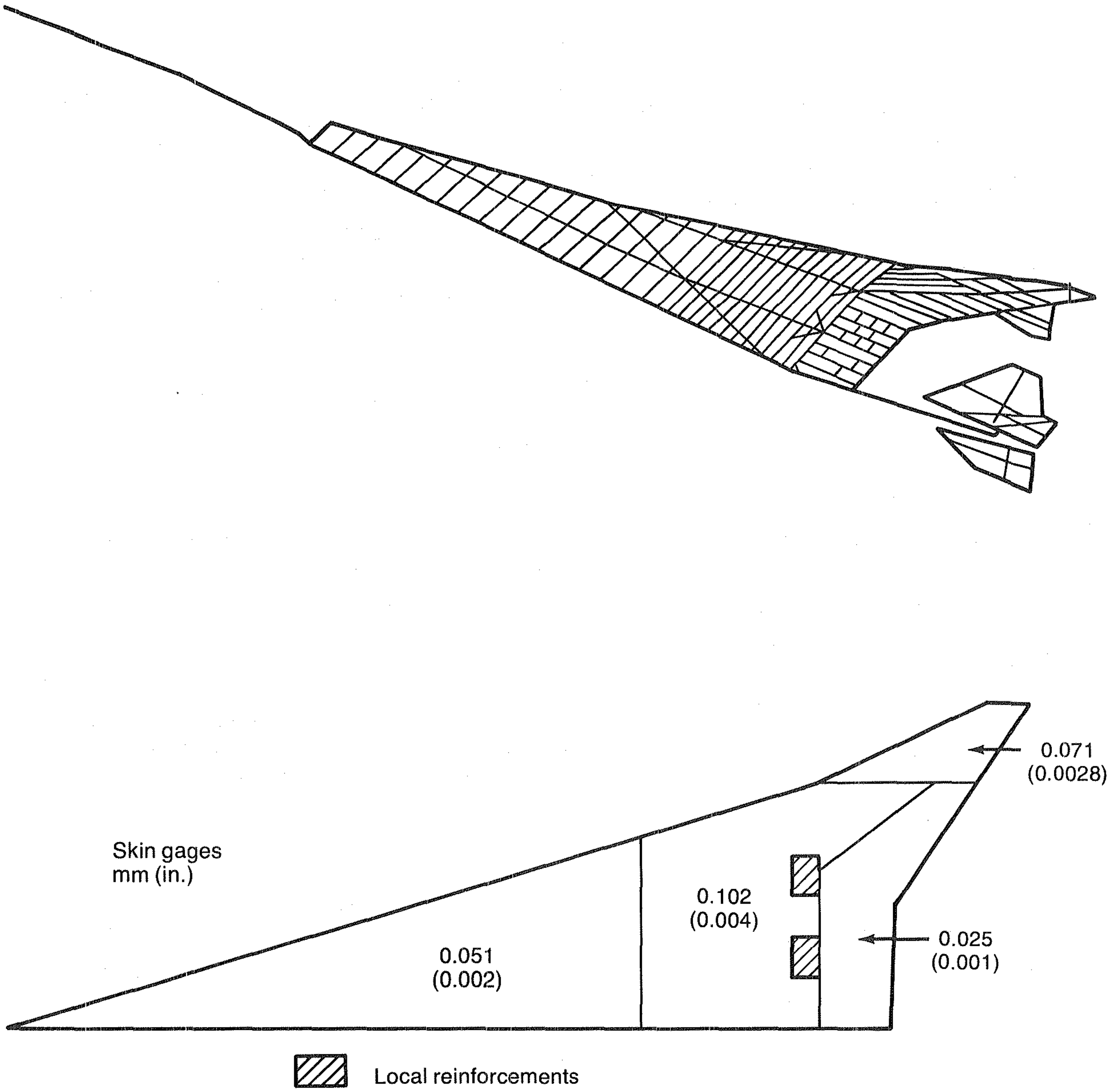


Figure 13.—ATLAS Finite Element Structural Representation of the NASA SCAT-15 Flutter Model

7.0 DELIVERABLE ITEMS

7.1 SOFTWARE

The PIAS computing software to be delivered for installation at the LaRC will include the ECM and ATLAS codes, and updates to the LEV Program. All of the PIAS software will be maintained by using UPDATE, Version 1.3, which is a CDC utility for creating and manipulating a file of card images in a compressed format known as a program library (ref. 34). UPDATE assigns a unique identifier and sequence number to each card image so that each one can be referred to individually. Therefore, any amount of text may be easily deleted or added. This capability makes code modification and maintenance relatively simple tasks, and provides a history of the code changes as well.

7.1.1 EXECUTION CONTROL MONITOR (ECM)

The ECM code will be delivered in source form on magnetic tape. It will be maintained as an UPDATE program library. At the LaRC, UPDATE will be used to generate the original Control Program statements. The ATLAS 4.1 software maintenance procedures will be used to precompile the Control Program into FORTRAN source code. These procedures will also be used to initiate compilation of the FORTRAN code and loading of the relocatable binary files to produce the absolute Control Program.

7.1.2 UPDATES TO THE LEV PROGRAM

The LEV Program, level 40, currently resides at the LaRC as an UPDATE program library. Therefore, the proposed modifications to the code, as discussed in section 3.3.2, will be delivered as UPDATE correction sets.

7.1.3 ATLAS SYSTEM

The code for the ATLAS Version 4.1 system, along with the proposed modifications as described in section 3.3.3, will be delivered in source form on magnetic tape. It also will be maintained in UPDATE library format. The Boeing Company will generate an executable version of ATLAS on the LaRC computing system.

7.2 DOCUMENTATION

The documents prepared specifically in support of PIAS and the documents describing the separate LEV and ATLAS systems will be delivered to the LaRC. The LEV Program documents are the final reports to the LaRC describing Level 40 of the LEV Program (NASA contracts NAS1-15169, NAS1-15275, and NAS1-16449). These documents will be updated to reflect the modifications described in section 3.3.2. The ATLAS documents describe Version 4.1 of the ATLAS system and are internal Boeing Company documents. Twenty copies of each of the following documents will be delivered to the LaRC.

o Program for an Iterative Aeroelastic Solution (PIAS) User Manual

o LEV Program Theory Document; reference 4

- LEV User's Guide and Programmer's Document; reference 5

- ATLAS System User Manual; reference 6

- ATLAS System Demonstration Problem Document; reference 35

The User Manual for the Program for an Iterative Aeroelastic Solution (PIAS) will include the following topics:

- Description of the PIAS procedure in detail, and general descriptions of the LEV Program and the ATLAS modules used in PIAS

- Guidelines for the use of PIAS, including the input data for the ECM, the LEV Program, and the ATLAS system, and the parameters in the PIAS procedure call statement

- Description of the overall system software design in terms of its engineering technical functions, its data management capabilities including the solution restart procedures, and its executive functions

- Guidelines for aerodynamic, structural, and mass modeling

- Documentation of the NACA delta wing and NASA SCAT-15 demonstration problems used for certification of PIAS (see sec. 6.3). These two demonstration problems, structured to emphasize the new PIAS aeroelastic solution analysis capabilities, will be documented in terms of the following:

  - Description of the problem and analysis steps (the aerodynamic and structural models, the input data, the analysis performed, etc.)

  - Presentation and discussion of the analysis results, including selected print and plot displays of the data.

  - Listings of all input data used to perform the analyses (the PIAS procedure call with its parameters, the ECM input data, the LEV input data, and the ATLAS input data)

7.3 MAINTENANCE OF PIAS

The Boeing Company proposes maintenance of PIAS for a period of one year following the delivery and installation of PIAS at the LaRC Computer Complex. Maintenance will include:

- Investigation of all program errors identified and reported by the LaRC personnel

- Correction of the errors

- Necessary revisions of the PIAS User Manual to account for error corrections

- Software modifications that may be required as a result of any CYBER computer operating changes at the LaRC Computer Complex

- Transmission of corrections to the LaRC

Any program or document errors identified by The Boeing Company will be reported immediately to the LaRC. In all cases, the LaRC will initiate any maintenance efforts to be performed by The Boeing Company.

8.0 PROPOSED IMPLEMENTATION PLAN

The tasks required to implement the proposed Program for an Iterative Aeroelastic Solution (PIAS) are grouped into the following three categories:

Execution Control Monitor (ECM)

- o Write code to execute the LEV and ATLAS programs and perform the additional functions required
- o Checkout with small problems
- o Checkout with larger, more realistic problems

LEV Program, Level 40

- o Write additional code
  - o Enhancements to improve solution convergence
  - o Additions required for PIAS
- o Validate as a standalone program
- o Validate in PIAS

ATLAS Version 4.1

- o Write additional code
  - o Required to install at the LaRC
  - o Modifications to capabilities
- o Install at the LaRC
- o Validate as standalone program
- o Validate in PIAS

These tasks are treated separately throughout this specification. Figure 14 shows an initial estimate of the flow time required to accomplish these tasks.

Boeing Commercial Airplane Company
P.O. Box 3707
Seattle, Washington 98124
May 23, 1983

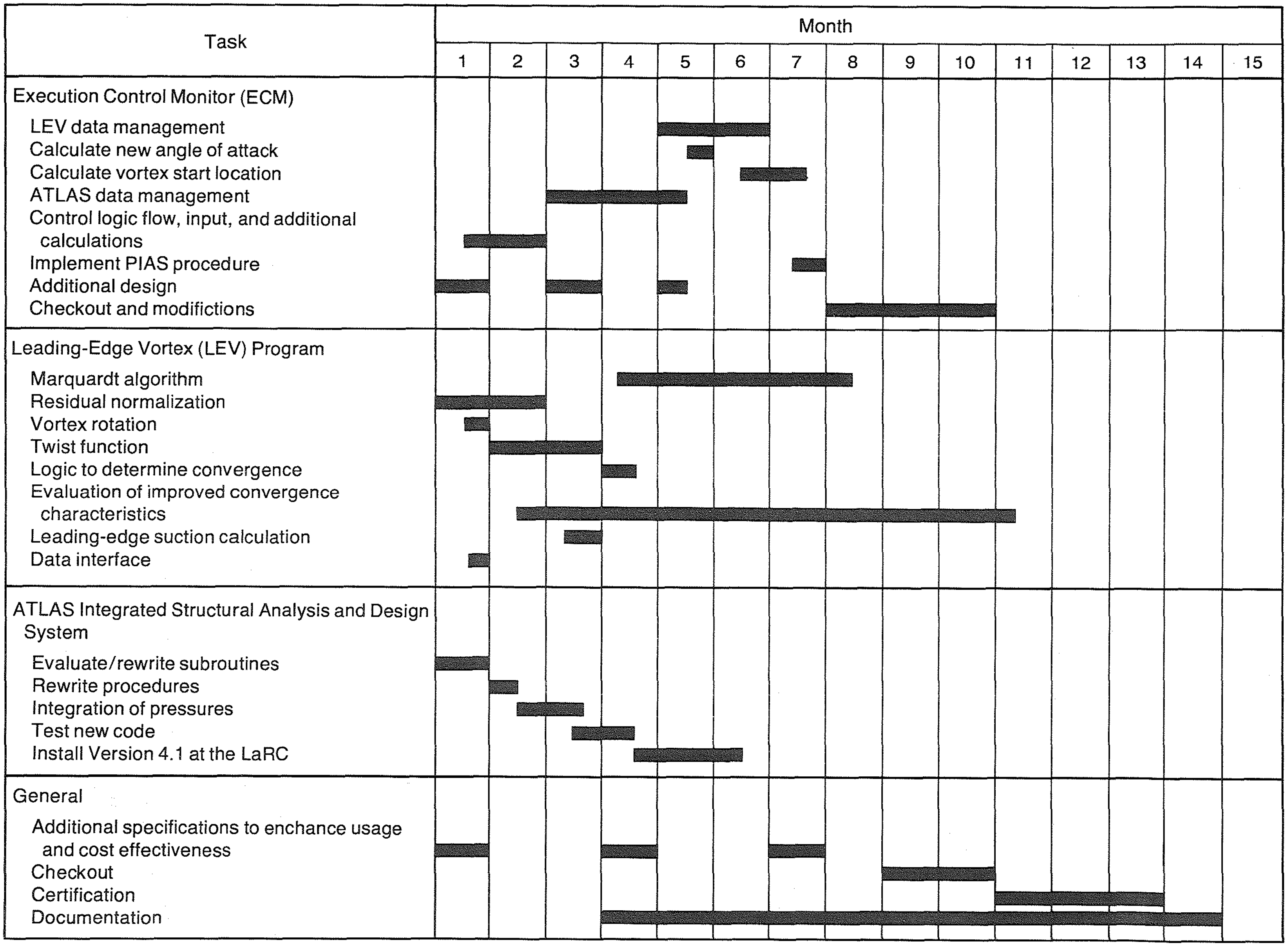


Figure 14.—Initial Estimate of Implementation Schedule for PIAS

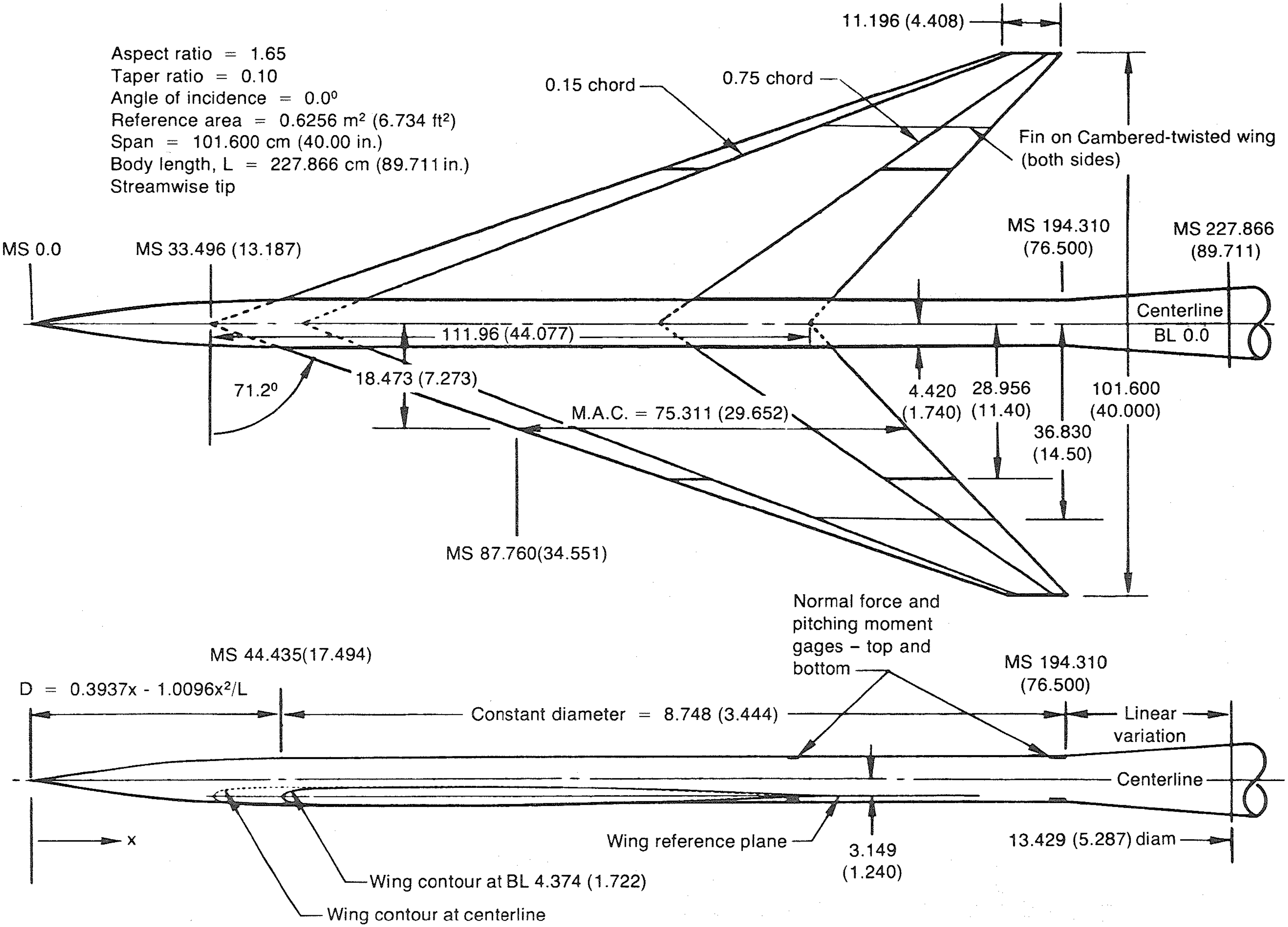
APPENDIX A
EXAMPLES OF THE LEADING-EDGE VORTEX PROGRAM CAPABILITIES

The LEV Program has been extensively evaluated, both by the LaRC personnel (ref. 7) and by The Boeing Company under NASA contract NAS1-15678 (ref. 8). The model built and tested under NASA contracts NAS1-12875, NAS1-14141, and NAS1-14962 was used in the evaluation by The Boeing Company. This model was a highly swept, low-aspect ratio wing that was mounted on a high slenderness ratio body. Three wings were constructed, all with the same planform and thickness (maximum 3.36%). One of the wings was flat, one had a twist that was typical of aeroelastic distortion, and the third had the same twist with camber (leading- and trailing-edges up) superimposed. General characteristics of this model are shown in figures A-1 through A-4.

Some examples from reference 8 of the theory-to-experiment comparisons of wing upper surface pressure distributions are shown in figures A-5 through A-7. The pressure distribution on the flat wing with zero thickness and a full-span vortex is shown in figure A-5. The predicted spanwise location of the vortex and the pressure level are good as compared to the experimental data, certainly a vast improvement over the predictions of linear theories.

The data in figure A-6 show upper surface pressure distributions for the three wing shapes—flat, twisted, and cambered-twisted. For each wing section in the figure, the data are arranged so that the experimental data are shown in a grid at the top of the page and the predictions of the LEV Program in the grid below it. The change in the pressures predicted by the LEV Program for a change in wing shape is similar to that seen in the experimental data, indicating that the proper influence would be seen in an aeroelastic solution.

Figure A-7 shows data for a partial-span vortex on a flat wing that was modeled to include thickness. The spanwise location of the vortex is good although the level of predicted pressure is too high and covers a very narrow area of the wing.



All dimensions in centimeters (inches)

Figure A-1. —General Arrangement and Characteristics

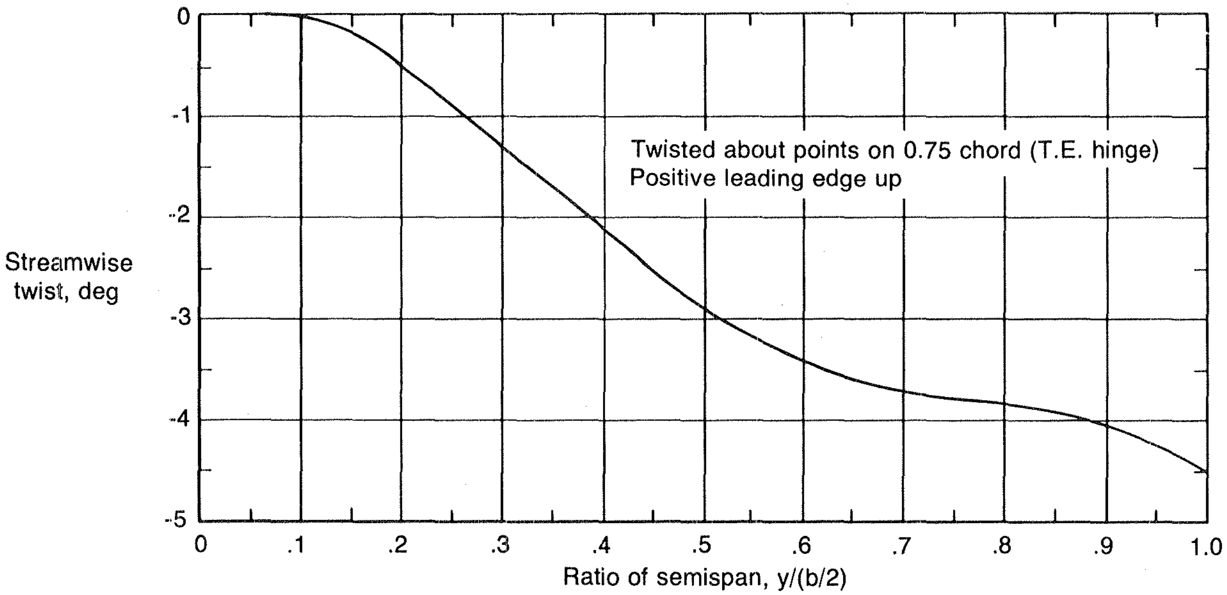


Figure A-2 —Spanwise Twist Distribution for the Model Wing

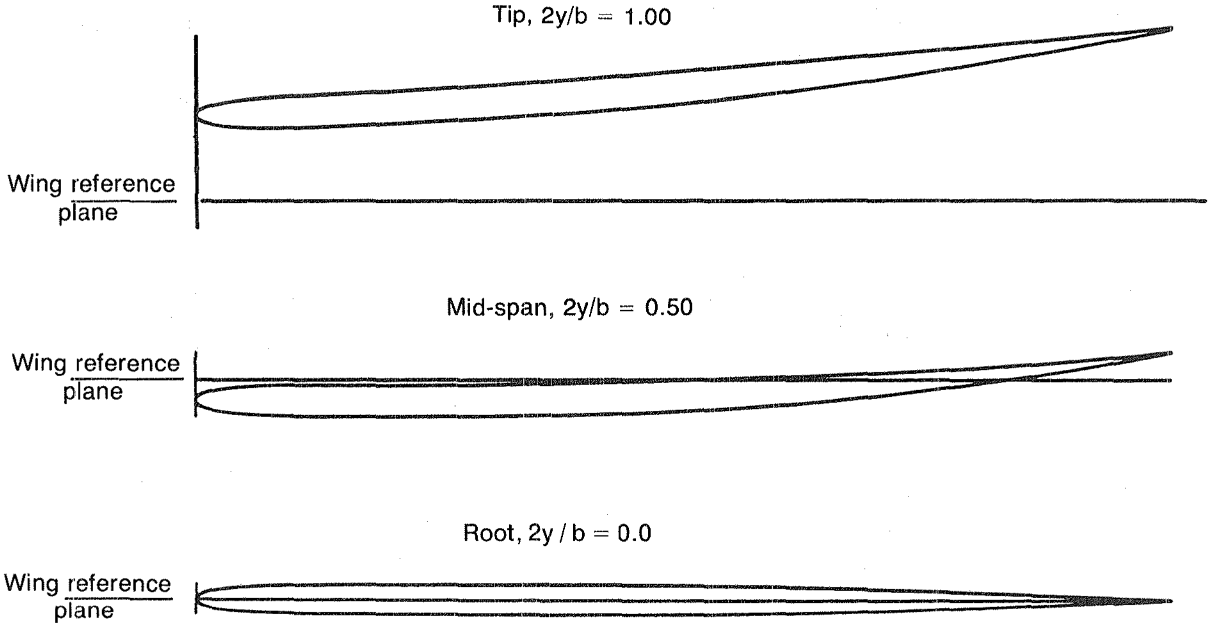


Figure A-3.—Cambered-Twisted Wing Section Geometry

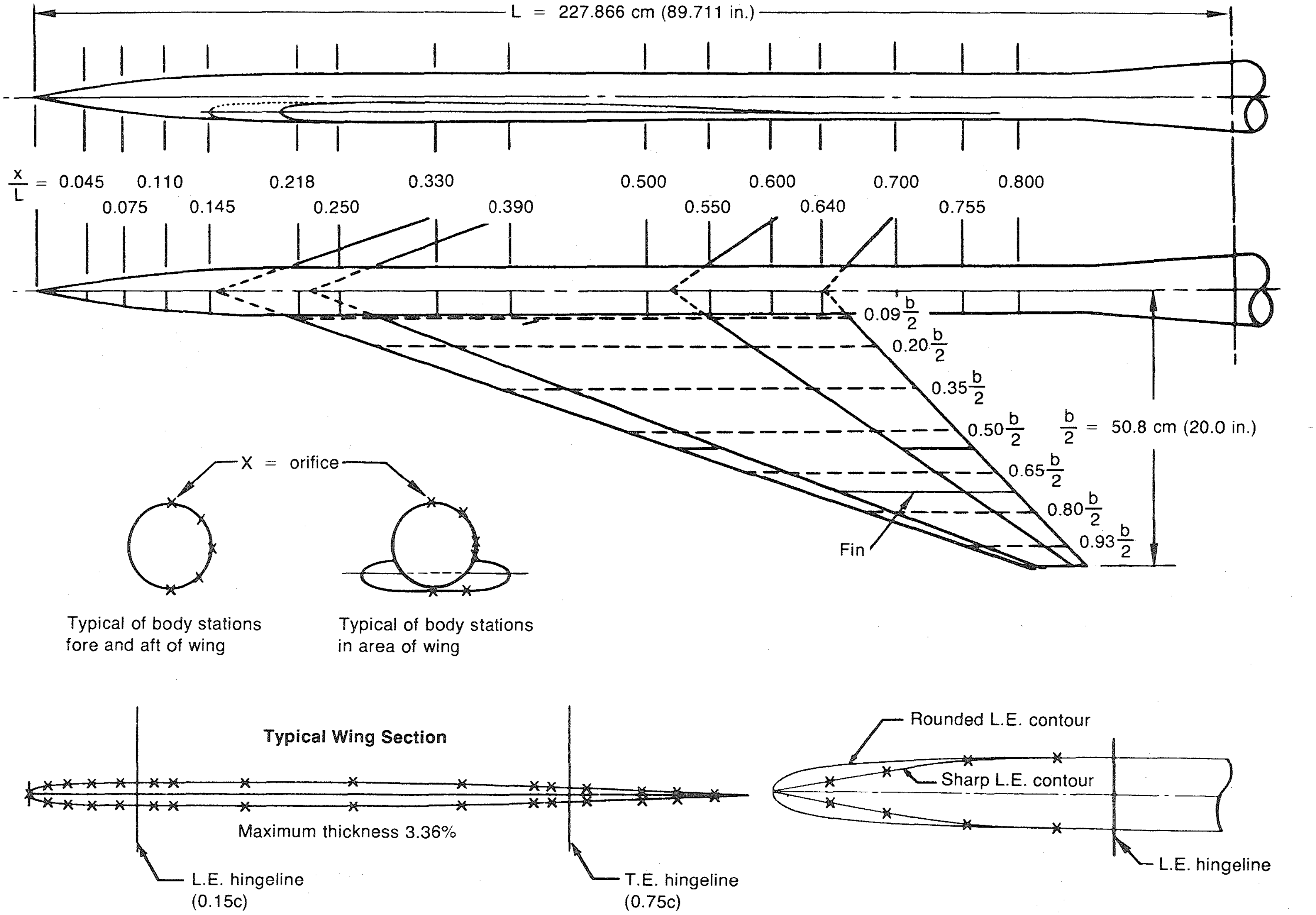


Figure A-4.—Pressure Orifice Locations

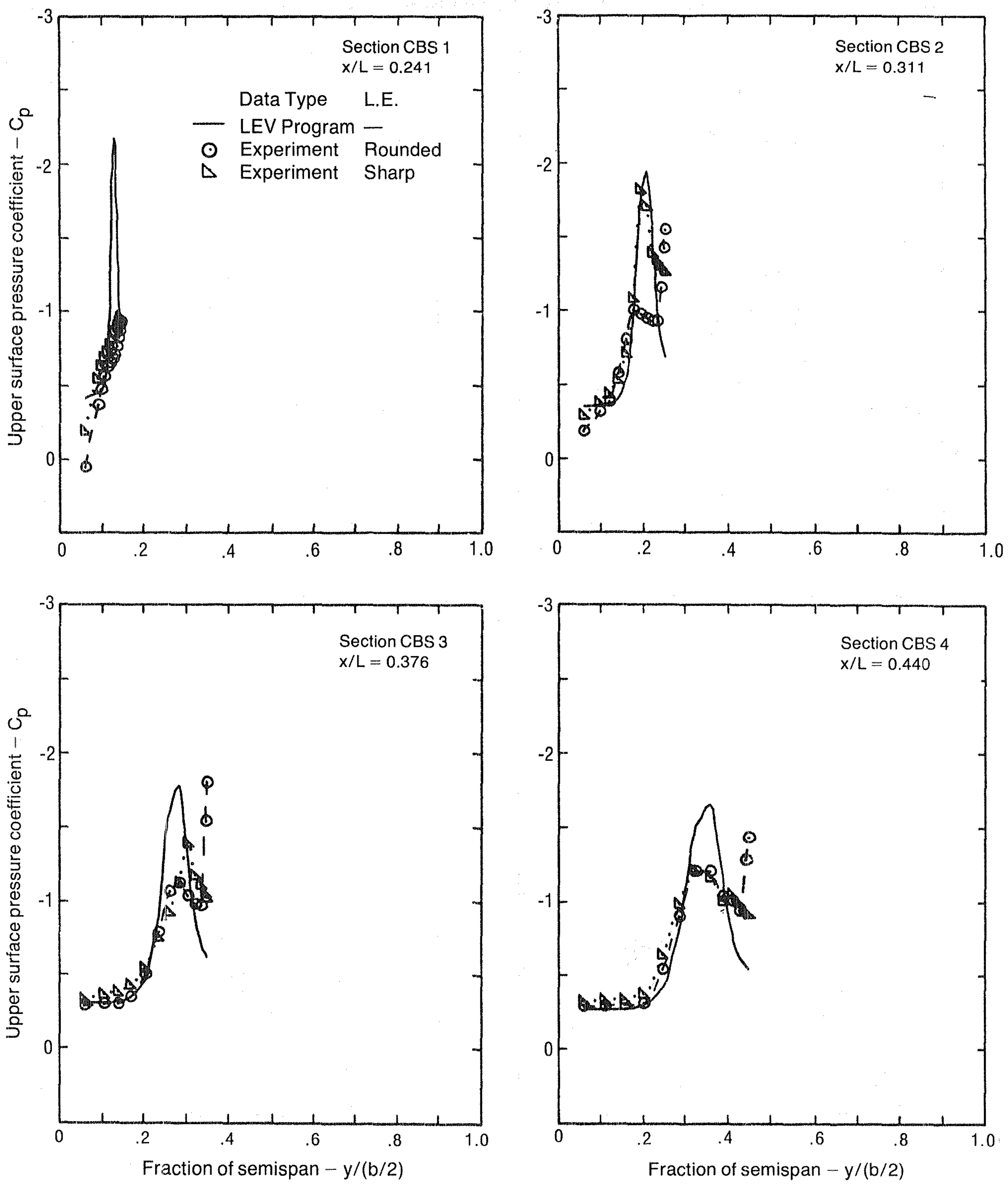


Figure A-5.—Upper Surface Pressure Distributions, Flat Wing, Zero Thickness, Δa = 0.0, M = 0.40, α = 16 Degrees

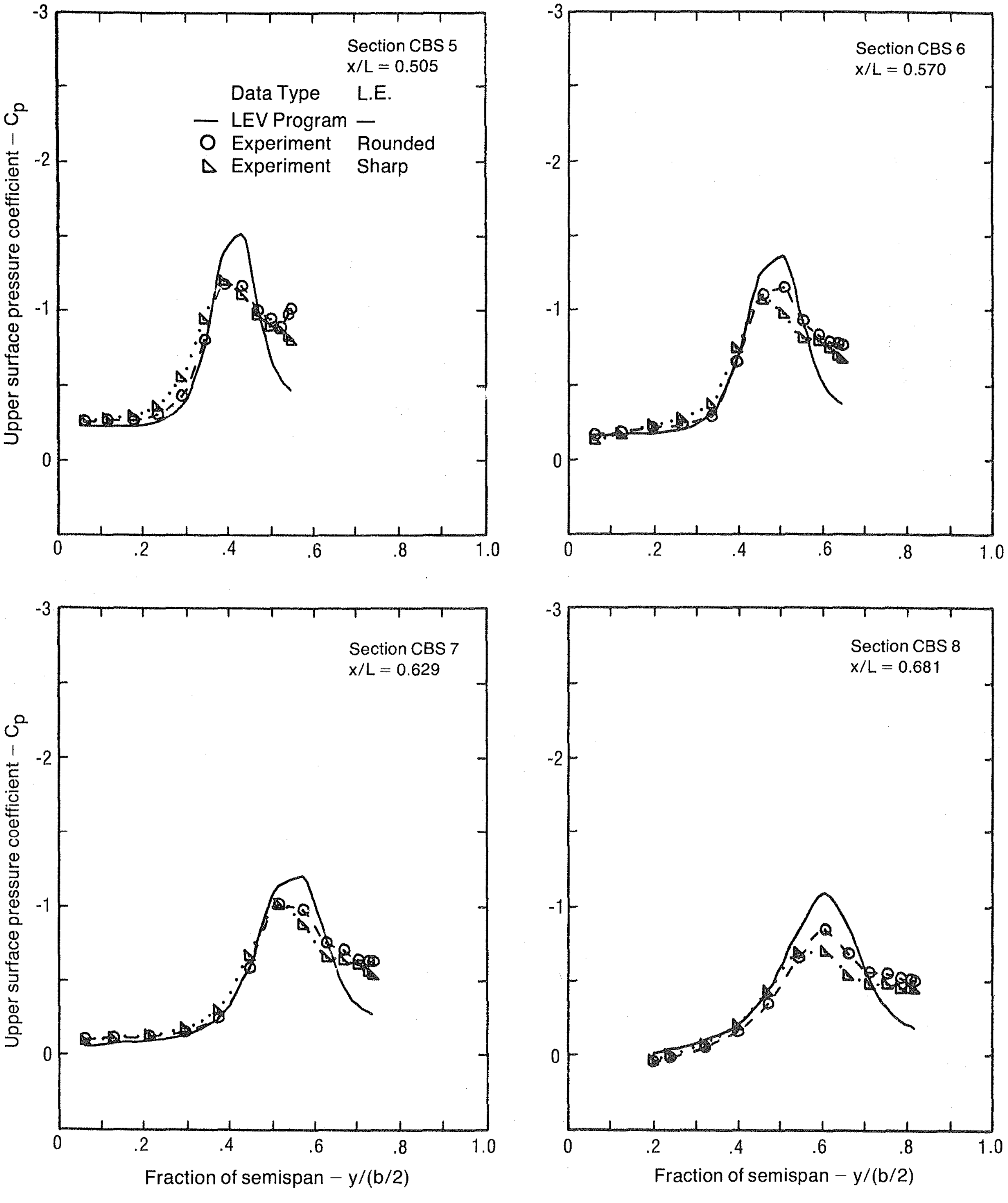
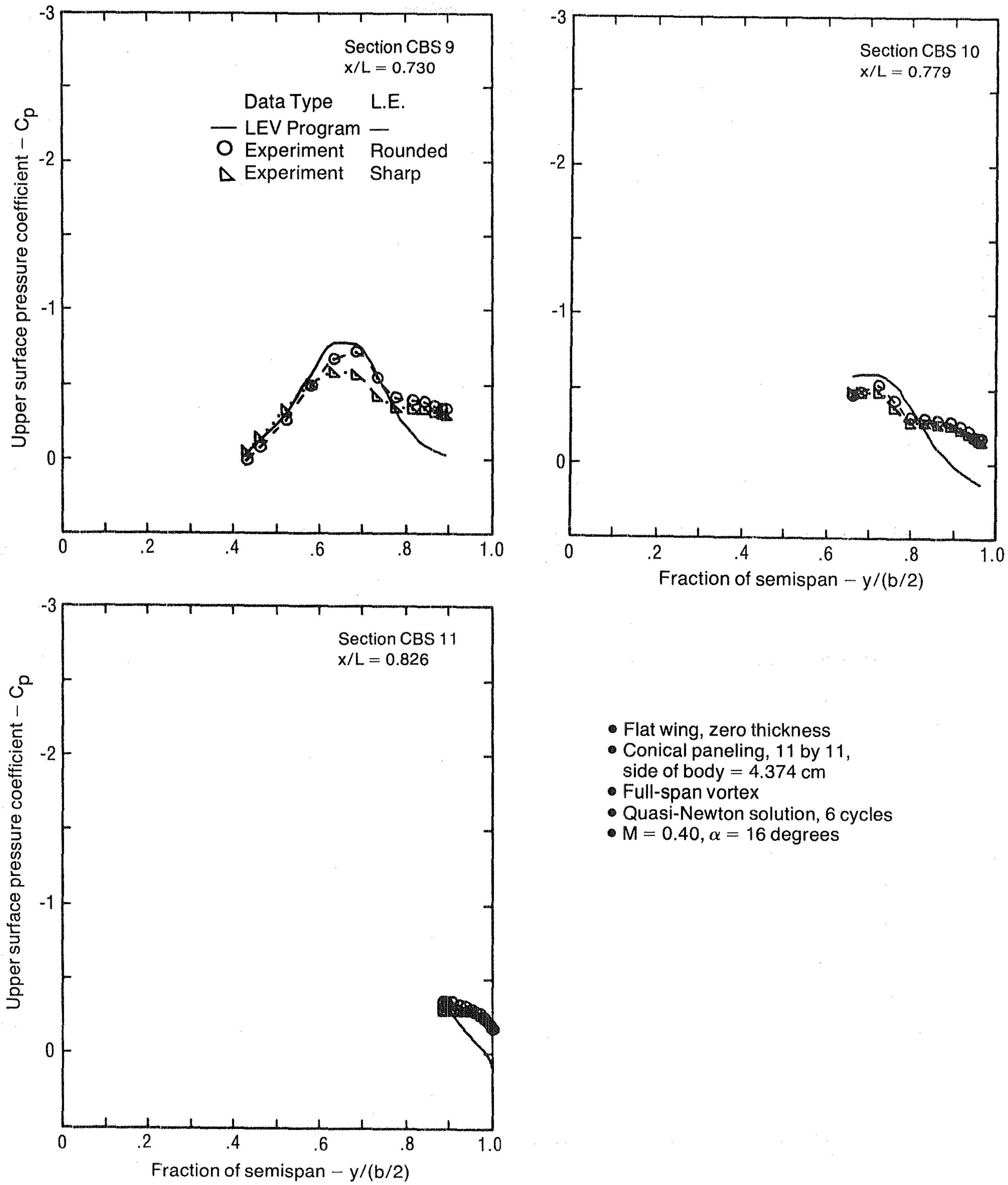


Figure A-5.—(Continued)



- Flat wing, zero thickness
- Conical paneling, 11 by 11, side of body = 4.374 cm
- Full-span vortex
- Quasi-Newton solution, 6 cycles
- M = 0.40, α = 16 degrees

Figure A-5.—(Concluded)

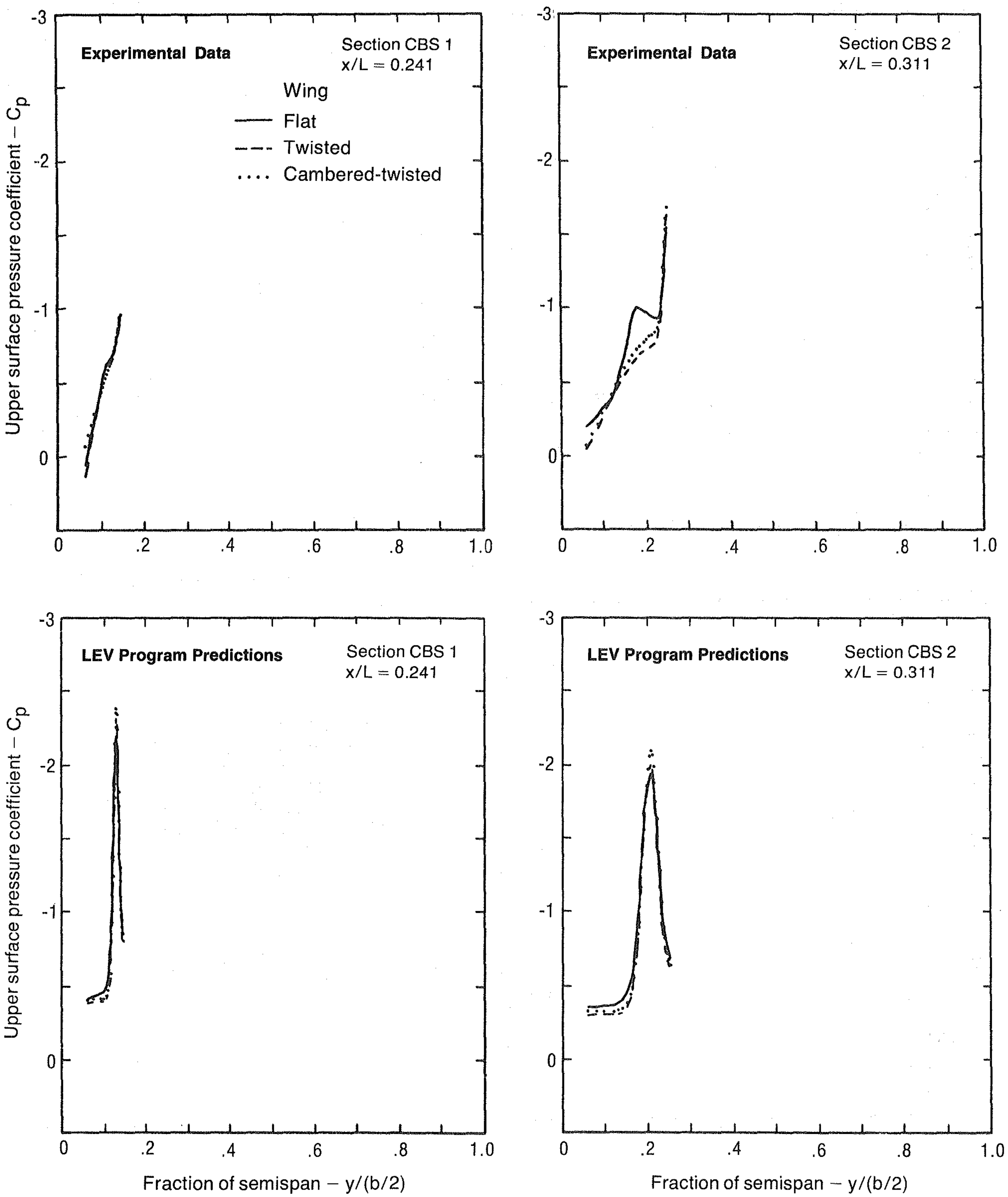


Figure A-6.—Comparison of Upper Surface Pressure Distributions for Three Wing Shapes, Experimental and LEV Program Predictions, Zero Thickness, $\Delta a = 0.0$, $M = 0.40$, $\alpha = 16$ Degrees

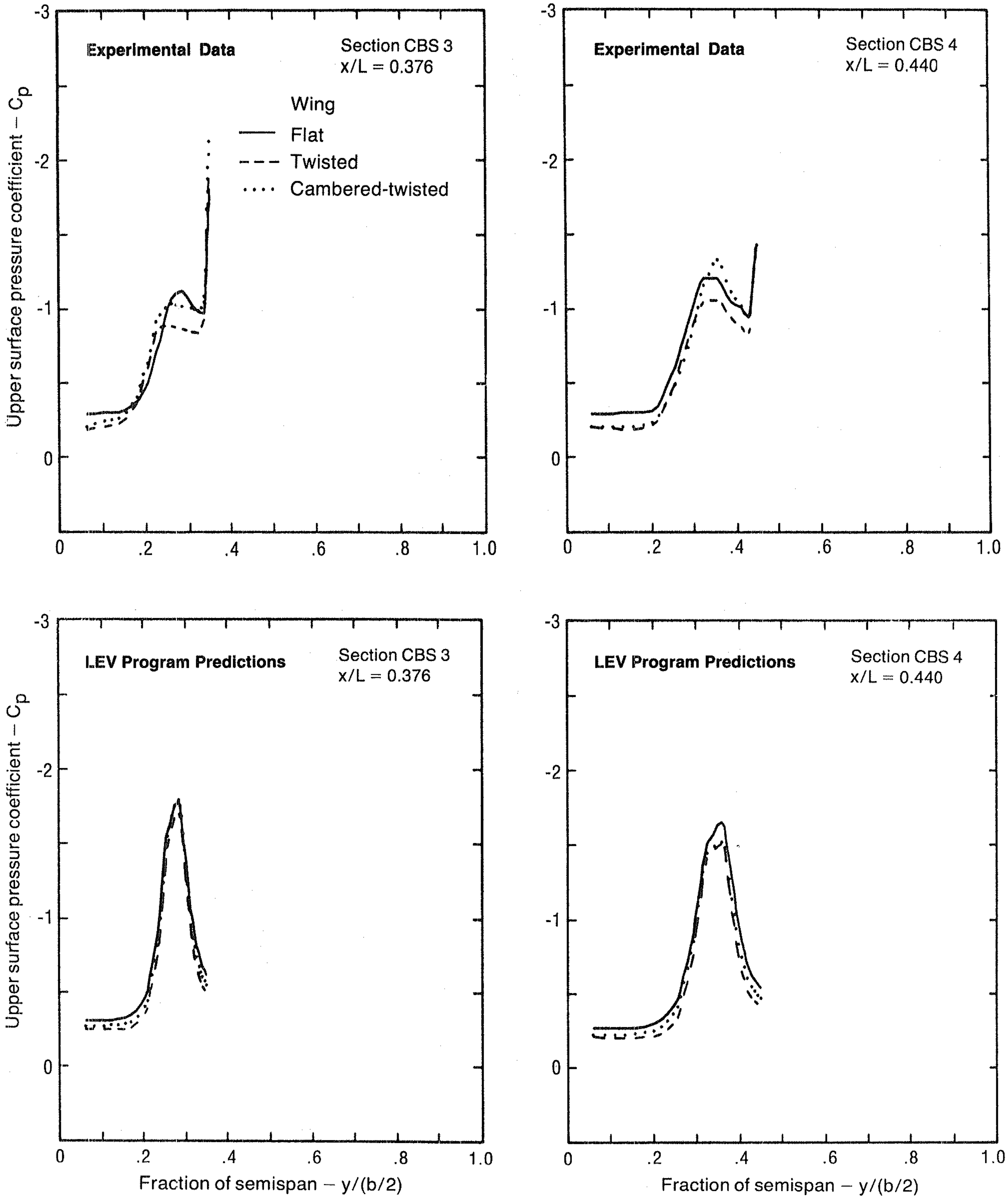


Figure A-6.—(Continued)

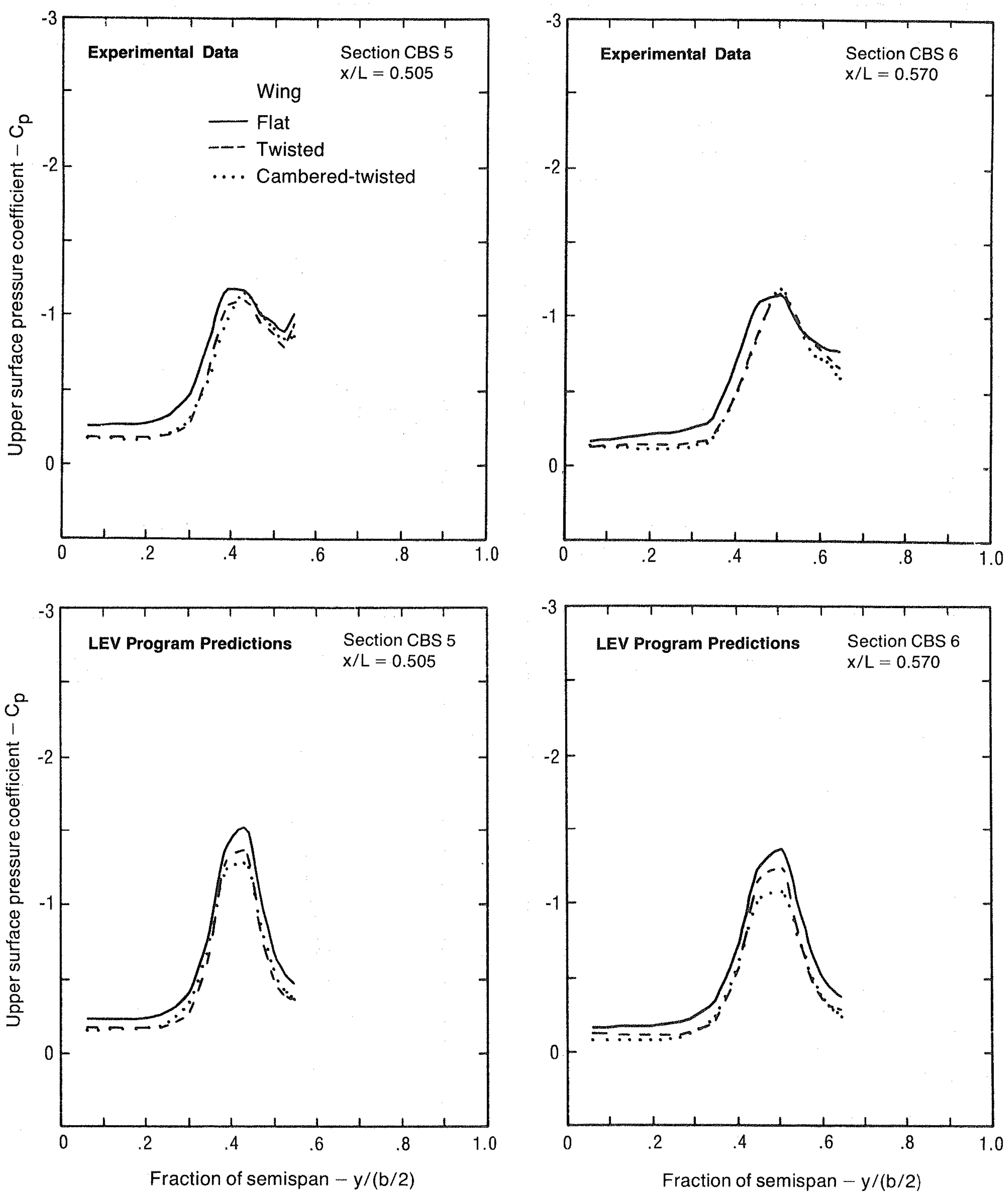


Figure A-6.–(Continued)

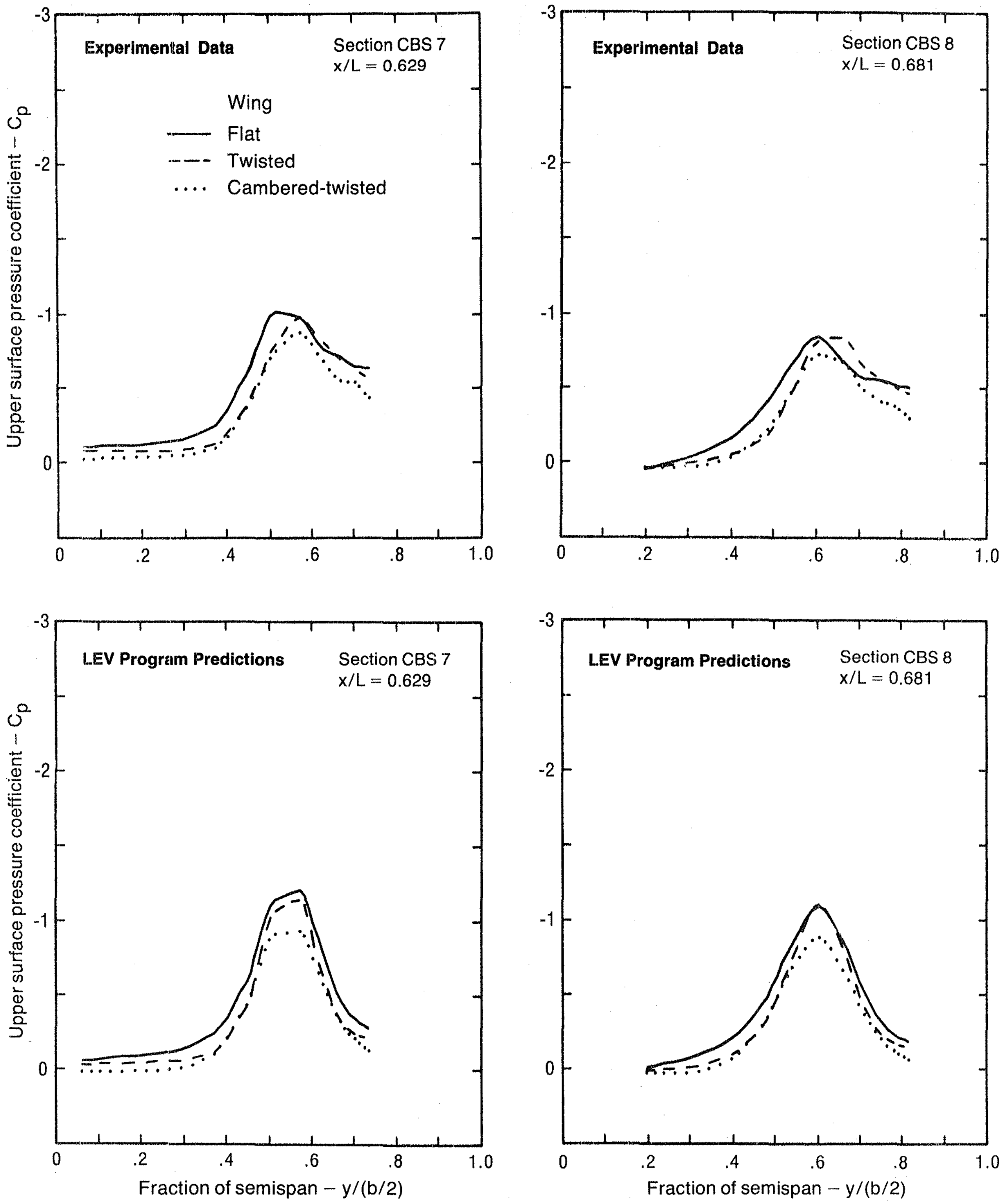


Figure A-6.–(Continued)

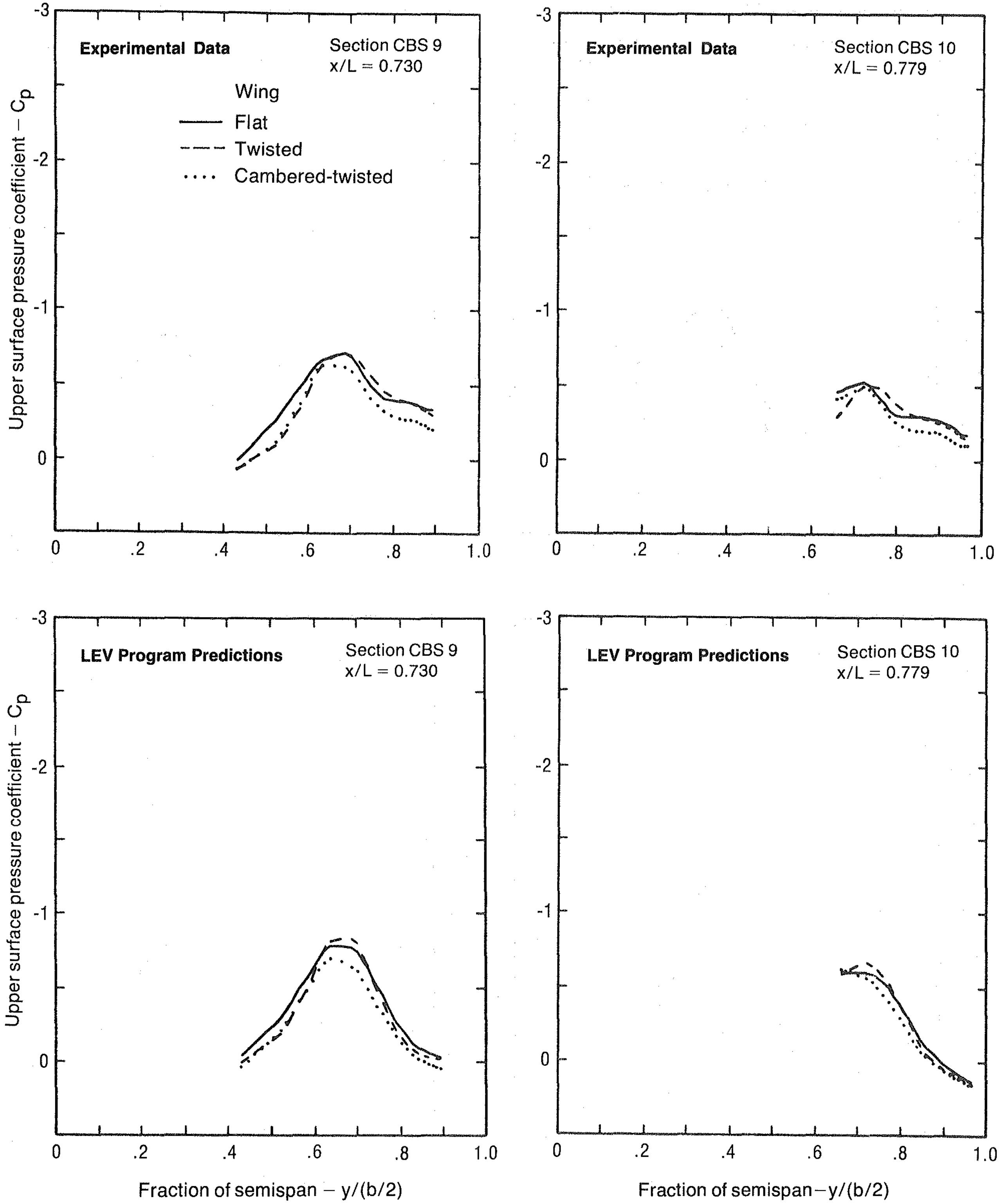
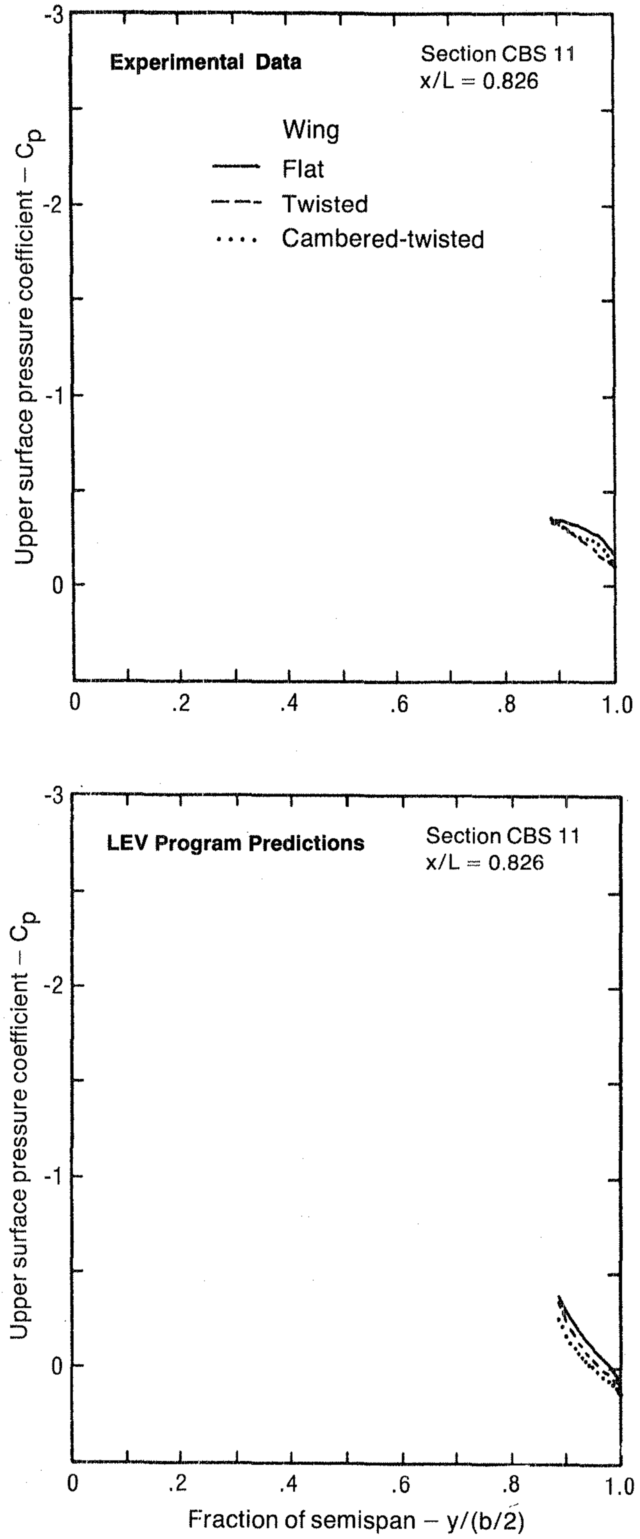


Figure A-6.–(Continued)



- $M = 0.40$, $\alpha = 16$ degrees

- Conical paneling, 11 by 11, side of body = 4.374 cm
- Full-span vortex
- Quasi-Newton solution, 6 cycles
- $M = 0.40$, $\alpha = 16$ degrees

Figure A-6.—(Concluded)

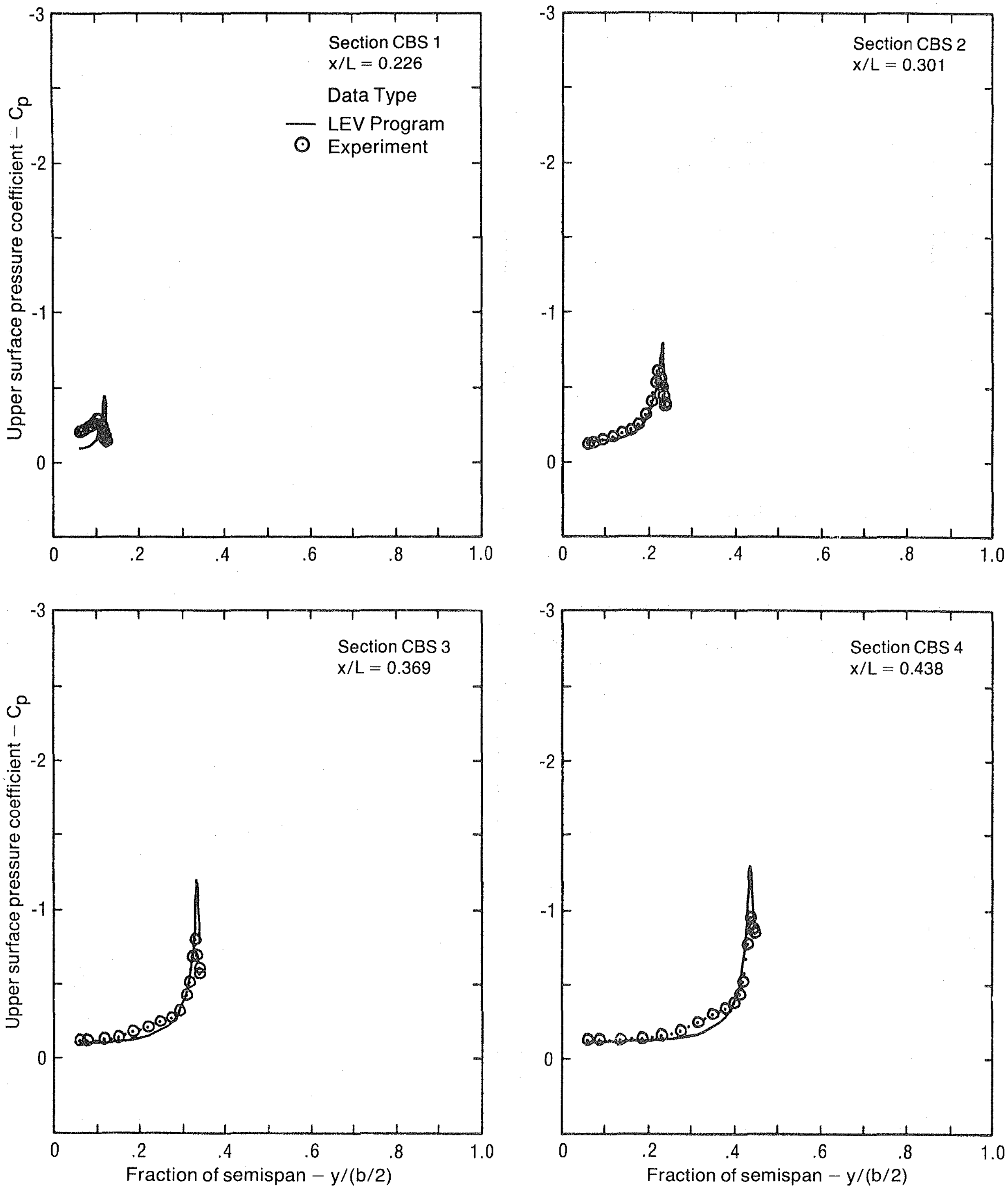


Figure A-7.—Upper Surface Pressure Distributions, Flat Wing With Thickness, Δa = 0.6, Initial Vortex Rotated Outward, M = 0.40, α = 8 Degrees

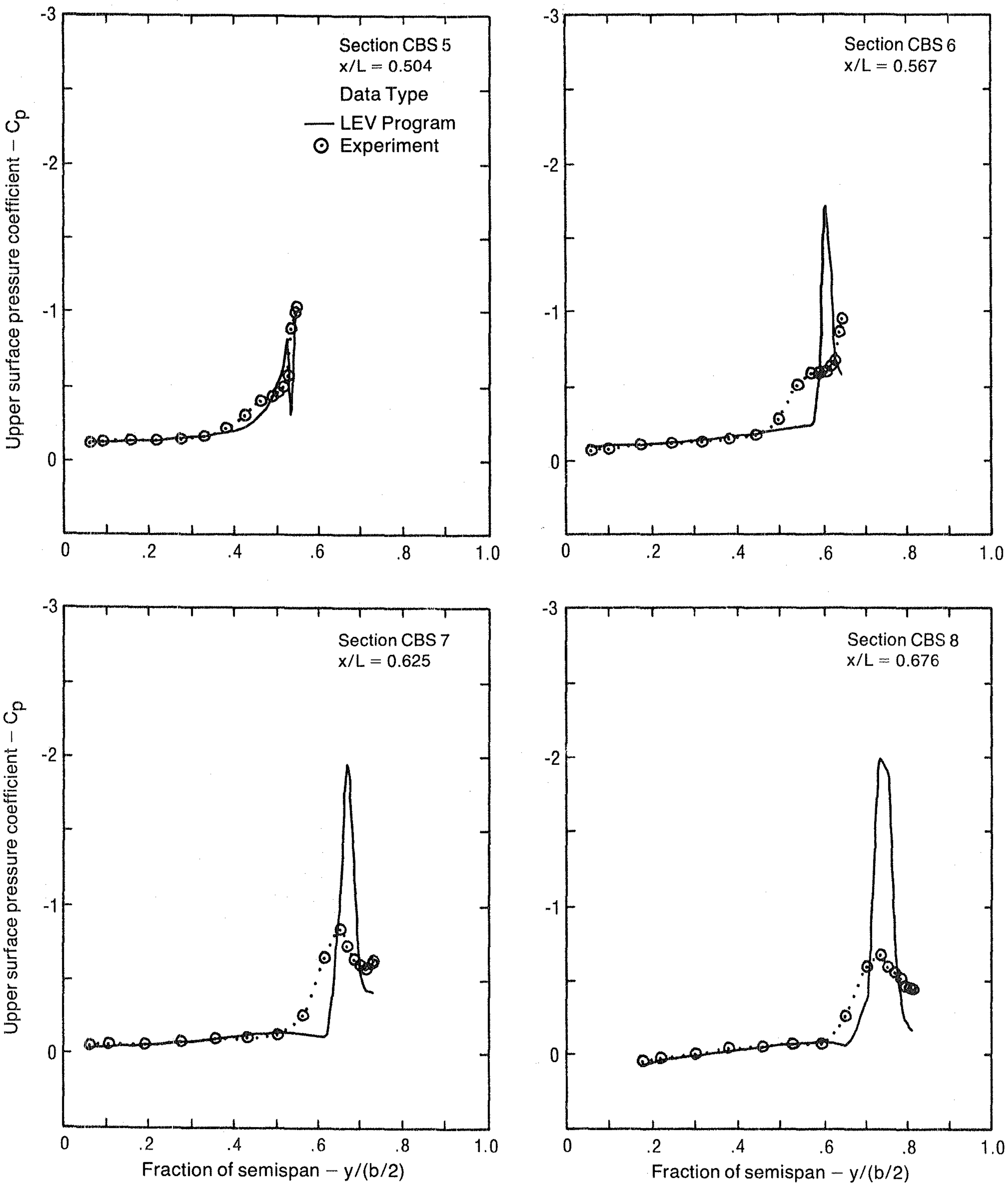
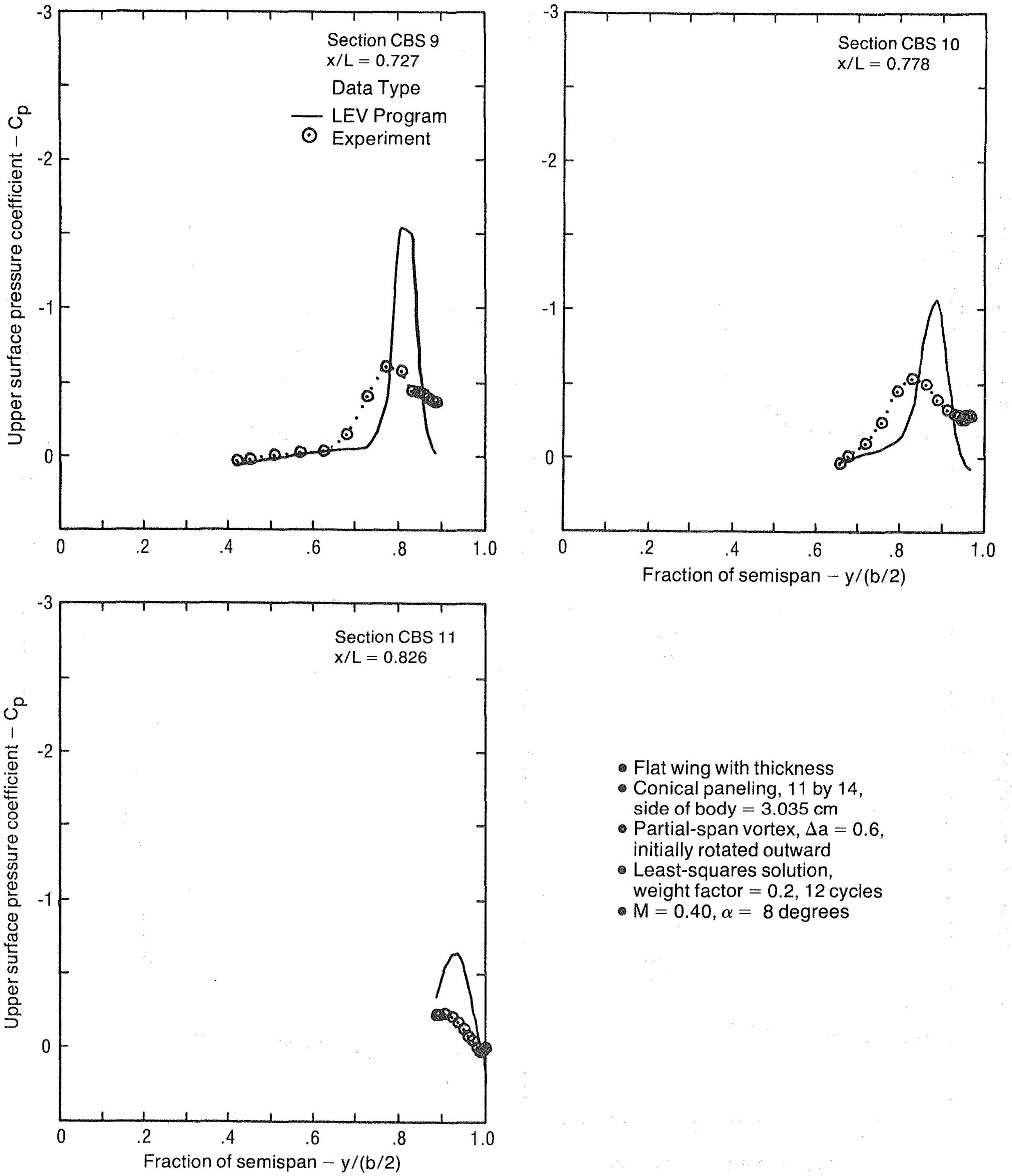


Figure A-7.—(Continued)



- Flat wing with thickness
- Conical paneling, 11 by 14, side of body = 3.035 cm
- Partial-span vortex, $\Delta a = 0.6$, initially rotated outward
- Least-squares solution, weight factor = 0.2, 12 cycles
- $M = 0.40$, $\alpha =$ 8 degrees

Figure A-7.—(Concluded)

APPENDIX B
DATA MANAGEMENT FOR ATLAS INTERPOLATION SOLUTION SEQUENCE

The following text describes the sequence of operations necessary to interpolate data from the aerodynamic panel centroids to the structural nodes and also the operations necessary to interpolate the deflections from the structural nodes to the aerodynamic panel corner points. The data sets referred to are defined in table B-1.

B.1 INTERPOLATE PRESSURES FROM AERODYNAMIC PANEL PRESSURE POINTS TO STRUCTURAL NODES

- SNARK read parameter matrix KPARMS1
  - data SET number is first non-zero column
  - number of nodes (M) is item 9 in column corresponding to SET number

- SNARK read local coordinate systems matrix KLOCOOa where "a" is the display code equivalent of the SET number
  - get number of columns in matrix
  - for each local coordinate system (column)
    - user ID is in item 1
    - if user ID = "LEV"—for lower surface and upper surface
      - transform pressure point coordinates of the doublet networks from the LEV local system to the ATLAS global system

- SNARK read nodal data matrix KNOALTa
  - for each nodal data row
    - analysis coordinate system (J) is in column 1
    - global x, y, z coordinates are in columns 2, 3, and 4

- SNARK read nodal correspondence table matrix KNC100a
  - for each internal node number (1 through M)
    - get pointer (K) to corresponding row of nodal data matrix

- SNARK read stiffness element nodal matrix KMELNOa
  - number of elements (L) is in item 1
  - for each internal element number (items 2 through L + 1)
    - get number of nodes (N) defining element
    - get user element number
    - get pointer to nodes
    - get the N internal node numbers defining element beginning at item given by pointer

- Initialize indicator for internal node usage (INODE (M, 2))

Table B-1.—SNARK Data Definition by the ECM

Position	Matrix name	Origin	Description
1	ZJIGPT	TAPE14, ECMUIN, JIGFILE, LEVCHK	Panel corner points of jig shape geometry
2	PCPTS	TAPE14	Panel corner points of all deflected wing, wake, and vortex networks
3	PRESPTS	TAPE14	Pressure point coordinates
4	CP	TAPE14	Pressure point coefficients
5	KPARMS1	DATARNF	SET parameters
6	KLOCOOa	DATARNF	Local coordinate systems
7	KNOALTa	DATARNF	Nodal data
8	KNC100a	DATARNF	Nodal correspondence table
9	KMELNOa	DATARNF	Stiffness element nodes
10	INODE	local	Internal node usage indicator
11	SEKddda	DATARNF	Element subset definitions
12	$X_{\text{SURF NODES}}$	local	Surface node global x-coordinates
13	$Y_{\text{SURF NODES}}$	local	Surface node global y-coordinates
14	$P_{\text{SURF NODES}}$	local	Surface nodal pressures
15	P_{NODES}	local	Nodal pressures
16	KUFRTOa	DATARNF	User freedom reference table
17	UDCO1ba	STRERNF	Nodal displacement control
18	UDzzzba	STRERNF	Active nodal displacements in ATLAS local X, Y, Z
19	$Txyz_{\text{SURF NODES}}$	local	Upper surface nodal translational displacements in LEV X, Y, Z
20	$X_{\text{DBLT PCPTS}}$	local	Panel corner point x-coordinates of doublet/analysis networks
21	$Y_{\text{DBLT PCPTS}}$	local	Panel corner point y-coordinates of doublet/analysis networks
22	$Tz_{\text{DBLT PCPTS}}$	local	Translational displacements in Z at panel corner points of doublet/analysis networks
23	$X_{\text{SORC PCPTS}}$	local	Panel corner point x-coordinates of source/analysis networks
24	$Y_{\text{SORC PCPTS}}$	local	Panel corner point y-coordinates of source/analysis networks
25	$Tz_{\text{SORC PCPTS}}$	local	Translational displacements in Z at panel corner points of source/analysis networks

```
o For lower surface subset and then upper surface subset
  · initialize number of unique nodes on surface ($M_{SURF\ NODES} = 0$)
  · SNARK read element subset matrix SEKddda where "ddd" is the SUBSET number
    · get subset vector
    · for each internal element number
      · if corresponding element is included in subset—for each internal node number "n"
        defining element (1 to N)
        · if node is the type to be loaded, i.e., structural
          · if node has *not* been used for a previous element ($INODE_{n,1} = INODE_{n,2} = 0$)
            · $M_{SURF\ NODES} = M_{SURF\ NODES} + 1$
            · fill next location ($M_{SURF\ NODES}$) of surface node arrays ($X_{SURF\ NODES}$, $Y_{SURF\ NODES}$) with global x, y coordinates in row K of nodal data matrix
              corresponding to internal node number
            · $INODE_{n,1}$ = "ddd"
            · $INODE_{n,2} = M_{SURF\ NODES}$
  · CALL PLATEI (. . .)
  · CALL PLATEO (. . .) for $P_{SURF\ NODES}$
  · for each internal node number "m" (1 through M)
    · if $INODE_{m,1}$ = "ddd"—fill location "m" of pressure array ($P_{NODES}$) with value in
      location $INODE_{m,2}$ of surface pressure array ($P_{SURF\ NODES}$)
```

B.2 GENERATE ATLAS LOADS INPUT DATA

```
o for all elements (1 through L) included on lower surface and upper surface
  · i = 0,1 (airloads only)
    BEGIN LOADS DATA
      BEGIN ELEMENT LOAD DATA
        CASE NET
        user element number;  pressures ($P_{NODES}$) at internal nodes to be loaded
          .
          .
          .
      END ELEMENT LOAD DATA
    END LOADS DATA

  · i ≥ 2 (net)
    BEGIN LOADS DATA
      CASE NET COMBINE 1. AIRLOAD 1. INERTIA
      BEGIN ELEMENT LOAD DATA
        CASE AIRLOAD
        user element number;  pressures ($P_{NODES}$) at internal nodes to be loaded
          .
          .
          .
      END ELEMENT LOAD DATA
      BEGIN INERTIA LOAD DATA
        CASE INERTIA
        0.0 0.0 ($N_z$) (386.04)
      END INERTIA LOAD DATA
    END LOADS DATA
```

- READ INPUT—store loads input in file DATARNF, but not in file SAVESSF

- EXECUTE MASS (. . .) for i = 2 (first aeroelastic cycle with inertia loads)

- Perform procedure STRESS for loadcase NET
  - EXECUTE STIFFNESS (. . .) for i = 0 or 1 (first aeroelastic cycle)
  - EXECUTE LOADS (. . .)
  - EXECUTE ALGEBRA (. . .)
  - EXECUTE CHOLESKY (. . .) decomposition for i = 0 or 1 (first aeroelastic cycle)
  - EXECUTE CHOLESKY (. . .) forward and back substitution
  - EXECUTE STRESS (. . .)
  - EXECUTE ALGEBRA (. . .)

B.3 INTERPOLATE STRUCTURAL NODE DEFLECTIONS TO AERODYNAMIC PANEL CORNER POINTS

- SNARK read user freedom reference table matrix KUFRTOa
  - Boundary condition STAGE number is item 1 in column 1

- SNARK read nodal displacement control matrix UDCO1ba where "b" is the display code equivalent of the STAGE number
  - for each node in user number order (items 1 through M)
    - get internal node number "m"
    - if node is on upper surface—
      - get code describing which freedoms are active
      - get PARTITION number of matrix containing displacement data
      - get pointer within partition to start of displacement data
      - if node begins a new PARTITION number—
SNARK read next displacement matrix UDzzzba where "zzz" is the PARTITION number
      - the displacements of active freedoms begin in current partition at item given by pointer. Put active translational displacements in X, Y, Z for internal node number in array Txyz(3). Initialize inactive translational displacements in X, Y, Z to 0.0
      - transform translational displacements in array Txyz from ATLAS local to ATLAS global coordinates for analysis system J given in row K of nodal data matrix corresponding to internal node number
      - fill array of translational displacements in X, Y, Z for all internal node numbers ($\mathrm{Txyz}_{\mathrm{SURF\ NODES}}$ ($\mathrm{M}_{\mathrm{SURF\ NODES}}$,3)) at row $\mathrm{INODE}_{\mathrm{m},\,2}$, the row of surface node arrays ($\mathrm{X}_{\mathrm{SURF\ NODES}}$, $\mathrm{Y}_{\mathrm{SURF\ NODES}}$) corresponding to internal node number
  - transform nodal translational displacements in array $\mathrm{Txyz}_{\mathrm{SURF\ NODES}}$ from ATLAS global system to the LEV local system

- For ATLAS global x, y coordinates in upper surface node arrays $\mathrm{X}_{\mathrm{SURF\ NODES}}$, $\mathrm{Y}_{\mathrm{SURF\ NODES}}$ from previous pressure point interpolation
  - CALL PLATEI (. . .)

- Set up LEV x, y coordinate arrays $\mathrm{X}_{\mathrm{DBLT\ PCPTS}}$, $\mathrm{Y}_{\mathrm{DBLT\ PCPTS}}$ using panel corner points from doublet/analysis networks *only*
  - CALL PLATEO (. . .) for $\mathrm{Tz}_{\mathrm{DBLT\ PCPTS}}$

```
o If LEV user input contains source/analysis networks for the exterior surfaces of a thick
  wing—assume wing thickness is maintained by interpolating ATLAS nodal translational
  displacements in Z for upper surface to source networks also
    · Set up LEV x, y coordinate arrays $X_{SORC\ PCPTS}$, $Y_{SORC\ PCPTS}$ using panel corner points
      from source/analysis networks only
        · CALL PLATEO (. . .) for $Tz_{SORC\ PCPTS}$
```

REFERENCES

1. Chipman, R.; Waters, C.; and MacKenzie, D.: Numerical Computation of Aeroelastically Corrected Transonic Loads. AIAA Paper No. 79-0766, April 1979. Presented at the AIAA/ASME/ASCE/AHS 20th Structures, Structural Dynamics and Materials Conference, St. Louis, MO.

2. Whitlow, Woodrow, Jr.; and Bennett, Robert M.: Application of a Transonic Potential Flow Code to the Static Aeroelastic Analysis of Three-Dimensional Wings. AIAA Paper No. 82-0689, May 1982.

3. Jameson, A.; and Caughey, D. A.: Numerical Calculation of the Transonic Flow Past a Swept Wing. NASA CR-153297, 1977.

4. Johnson, F. T.; Lu, P.; Tinoco, E. N.; and Epton, M. A.: An Improved Panel Method for the Solution of Three-Dimensional Leading-Edge Vortex Flows, Volume I—Theory Document. NASA CR-3278, 1980.

5. Tinoco, E. N.; Lu, P.; and Johnson, F. T.: An Improved Panel Method for the Solution of Three-Dimensional Leading-Edge Vortex Flows, Volume II—User's Guide and Programmer's Document. NASA CR-3279, 1980.

6. Dreisbach, Rodney L.: ATLAS—An Integrated Structural Analysis and Design System, "User's Manual—Input and Execution Data, (ATLAS 4.1)." Boeing Document No. D6-25400-0003, 1981.

7. Luckring, J. M.; Schoonover, W. E., Jr.; and Frink, N. T.: Recent Advances in Applying Free Vortex Sheet Theory for the Estimation of Vortex Flow Aerodynamics. AIAA Paper No. 82-0095, January 1982.

8. Manro, Marjorie E.: Aeroelastic Loads Prediction for an Arrow Wing, Task III—Evaluation of the Boeing Three-Dimensional Leading-Edge Vortex Code. NASA CR-3642, 1983.

9. Tinoco, E. N.; and Mercer, J. E.: FLEXSTAB—A Summary of the Functions and Capabilities of the NASA Flexible Airplane Analysis Computer System. NASA CR-2564, October 1975.

10. Dusto, A. R.; et al.: A Method for Predicting the Stability Characteristics of an Elastic Airplane. Volume 1—FLEXSTAB Theoretical Description. NASA CR-114712, 1974.

11. Maskell, E. C.: Some Recent Developments in the Study of Edge Vortices. Proceedings of 3rd Congress of Int. Council Aero. Sci., 1962, pp. 737-749, Spartan Books, Inc., Washington, 1964.

12. Peckham, D. H.: Low-Speed Wind-Tunnel Tests on a Series of Uncambered Slender Pointed Wings with Sharp Edges. ARC RM No. 3186, 1961.

13. Smith, J. H. B.: Improved Calculations of Leading-Edge Separation from Slender Delta Wings. RAE Tech. Report 66070, March 1966.

14. Ehlers, F. Edward; Epton, Michael A.; Johnson, Forrester T.; Magnus, Alfred E.; and Rubbert, Paul E.: A Higher Order Panel Method for Linearized Supersonic Flow. NASA CR-3062, 1979.

15. Harder, R. L.; and Desmarais, R. N.: Interpolation Using Surface Splines. Journal of Aircraft, vol. 9, no. 2, pp. 189-191, Feb. 1972.

16. Erickson, W. J.: SNARK User's Manual. BCS-G0686, June 1975.

17. Ehlers, F. Edward; and Manro, M. E.: A Method for Computing the Leading-Edge Suction in a Higher-Order Panel Method. NASA CR-3730, 1984.

18. Polhamus, E. C.: Predictions of Vortex Lift Characteristics by a Leading-Edge Suction Analogy. Journal of Aircraft, vol. 8, no. 4, April 1971, pp. 193-199.

19. Kulfan, R. M.: Wing Airfoil Shape Effects on the Development of Leading-Edge Vortices. AIAA Paper No. 79-1675, August 1979.

20. Kulfan, R. M.: Wing Geometry Effects on Leading-Edge Vortices. AIAA Paper No. 79-1872, August 1979.

21. Marquardt, D. W.: An Algorithm for Least Squares Estimation of Nonlinear Problems. SIAM Journal of Numerical Analysis, vol. 11, 1963, pp. 431-441.

22. Dahlquist, Germund; and Bjorck, Ako (Anderson, Ned, transl.): Numerical Methods. Prentice-Hall, Inc., 1974.

23. Lamar, J. E.; and Luckring, J. M.: Recent Theoretical Developments and Experimental Studies Pertinent to Vortex Flow Aerodynamics—With a View Towards Design. High Angle of Attack Aerodynamics, AGARD CP-247, Paper No. 24, Jan. 1979.

24. Frink, N. T.: Analytical Study of Vortex Flaps on Highly Swept Delta Wings. ICAS 82-6.7.2, 1982.

25. Craidon, C. B.: Description of a Digital Computer Program for Airplane Configuration Plots. NASA TM X-2074, Sept. 1970.

26. Middleton, W. D.; Lundry, J. L.; and Coleman, R. G.: Aerodynamic Design and Analysis System for Supersonic Aircraft. NASA CR-2521, March 1975.

27. NOS Version 1 Reference Manual, Volume 1. Control Data Corporation Publication 60435400, Rev. N, annotated for use at LaRC, 1982.

28. NOS Version 1 Reference Manual, Volume 2. Control Data Corporation Publication 60445300, Rev. N, 1981.

29. Computer Programing Manual, Volume II, Subroutine Library. NASA Langley Research Center, 1980 (rev.).

30. Computer Programing Manual, Volume IV, Special Capabilities. NASA Langley Research Center, 1983.

31. FORTRAN Extended Version 4 Reference Manual, Control Data Corporation Publication 60497800, Rev. D, 1975.

32. Kordes, E. E.; Kruszewski, E. T.; and Weidman, D. J.: Experimental Influence Coefficients and Vibration Modes of a Built-Up 45° Delta-Wing Specimen. NACA TN 3999, May 1957.

33. Advanced Concept Studies for Supersonic Vehicles. NASA CR-159244, section 6.0, May 1980.

34. Update Version 1 Reference Manual. Control Data Corporation Publication 60449900, Rev. E, 1982.

35. Dreisbach, Rodney L.: ATLAS—An Integrated Structural Analysis and Design System, "System Demonstration Problems, (ATLAS 4.1)." Boeing Document No. D6-25400-0005, 1981.

1. Report No. NASA CR-172200	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle SPECIFICATION FOR A PROGRAM FOR AN ITERATIVE AEROELASTIC SOLUTION (PIAS)	5. Report Date December 1983 6. Performing Organization Code
7. Author(s) Marjorie E. Manro, Michael J. Donahue, Rodney L. Dreisbach, and John E. Bussoletti	8. Performing Organization Report No. D6-52134 10. Work Unit No.
9. Performing Organization Name and Address Boeing Commercial Airplane Company P.O. Box 3707 Seattle, Washington 98124	11. Contract or Grant No. NAS1-16740 13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code 743-01-12-02

15. Supplementary Notes

Technical monitor—Percy J. Bobbitt, Chief, Transonic Aerodynamics Division, NASA Langley Research Center, Hampton, Virginia.
Principal investigator—Marjorie E. Manro, Boeing Commercial Airplane Company.

16. Abstract

An engineering and software specification has been written for a computer program to calculate aeroelastic structural loads including the effects of nonlinear aerodynamics. This computer program will be called the Program for an Iterative Aeroelastic Solution (PIAS). The procedure used in PIAS is to alternately execute two computer codes: one to calculate aerodynamic loads for a specific wing shape, and another to calculate the deflected shape caused by this loading. A significant advantage to the design of PIAS is that the initial aerodynamic module can be replaced with others.

The Boeing Company's Leading-Edge Vortex (LEV) Program will be used as the aerodynamic module in PIAS. This provides the capability to calculate aeroelastic loads, including the effects of a separation-induced leading-edge vortex.

The finite element method available in The Boeing Company's ATLAS Integrated Structural Analysis and Design system will be used to determine the deflected wing shape for the applied aerodynamics and inertia loads. The data management capabilities in ATLAS will be used by the Execution Control Monitor (ECM) of PIAS to control the solution process.

17. Key Words (Suggested by Author(s)) Aeroelasticity Experimental pressure distributions Aerodynamic theory Arrow-wing configuration Aeroelastic solution	18. Distribution Statement Unclassified-unlimited Subject category 02

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 100	22. Price

For sale by the National Technical Information Service, Springfield, Virginia 22151

End of Document